

Received 17 March 2025; revised 7 November 2025; accepted 12 November 2025; date of publication 24 November 2025; date of current version 6 January 2026.

Digital Object Identifier 10.1109/TQE.2025.3636049

Feynman Meets Turing: Computability Aspects of Exact Circuit Synthesis, Gate Efficiency, and the Spectral Gap Conjecture

YANNIK N. BOECK¹ , HOLGER BOCHE^{1,2,3,4}  (Fellow, IEEE),
AND FRANK H.P. FITZEK⁵  (Fellow, IEEE)

¹Chair of Theoretical Information Technology, Technical University of Munich, D-80333 Munich, Germany

²BMBF Research Hub 6G-life, D-80333 Munich, Germany

³Munich Center for Quantum Science and Technology, D-80799 Munich, Germany

⁴Munich Quantum Valley (MQV), D-80807 Munich, Germany

⁵Telekom Chair of Communication Networks, BMBF Research Hub 6G-life, Cluster of Excellence “Centre for Tactile Internet with Human-in-the-Loop (CeTI),” and the 5G Lab Germany, Technische Universität Dresden, D-01062 Dresden, Germany

Corresponding author: Yannik N. Boeck (e-mail: yannik.boeck@tum.de).

This work was supported in part by the German Federal Ministry of Education and Research (BMBF) through the program “Souverän. Digital. Vernetzt.” – Joint project 6G-life under Grant 16KISK001K, in part by the project QUARKS under Grant 16KIS1998K, in part by the project Q-TREX under Grant 16KISR027, and in part by project QUIET under Grant 16KISQ092. The work of Yannik N. Böck and Holger Boche was supported in part by the BMBF within the National Initiative on 6G Communication Systems through the Research Hub 6G-life under Grant 16KISK002. The work of Yannik N. Böck was additionally supported in part by the BMBF Quantum Programm QD-CamNetz under Grant 16KISQ077, in part by the QuaPhySI under Grant 16KIS1598K, and in part by QUIET under Grant 16KISQ093. The work of Frank H.P. Fitzek was supported in part by the German Research Foundation (DFG) as part of Germany’s Excellence Strategy – EXC2050/1 – Project ID 390696704 – Cluster of Excellence “Centre for Tactile Internet with Human-in-the-Loop” (CeTI) of Technische Universität Dresden.

ABSTRACT We consider exact quantum circuit synthesis, quantum gate efficiency, and the spectral gap conjecture from the perspective of computable analysis. Circuit synthesis, in both its exact and its approximate variant, is fundamental to the circuit model of quantum computing. As an engineering problem, however, the practical and theoretical aspects of quantum circuit synthesis are far from being fully understood. Particularly, this concerns explicit methods for gate-agnostic circuit synthesis and questions of gate efficiency. More than 20 years ago, Harrow et al. published their famous spectral gap theorem: given a suitable family of quantum gates, it is possible to approximate any unitary transformation by means of a quantum circuit whose length is proportional to the required accuracy’s logarithm. Moreover, Harrow et al. suspected that all universal gate families allow for this type of approximation, a hypothesis that became known as the spectral gap conjecture and remains unproven until today. Being an entirely classical task, quantum circuit synthesis must be considered in the context of digital computing, that is, in the context of Turing computability and computable analysis. Using the relevant mathematical framework, we establish no-go results concerning exact quantum circuit synthesis and quantum big-O analysis. Our findings relate to the theory of approximate t-designs, which has recently received notable attention through the literature. Moreover, as follows from our findings, the existence of an algorithm that computes leading big-O coefficients would prove the spectral gap conjecture true within the computable special unitary group.

INDEX TERMS Approximate t-design, circuit synthesis, computable analysis, gate efficiency, quantum computing, quantum gate, spectral gap, turing machine, big-O theory.

I. INTRODUCTION

Throughout the past decades, the prospect of unprecedented computational capacities—allegedly dating back to comments by Feynman [1], [2] on the efficient simulation of physical processes—has lead to remarkable research

efforts in the field of quantum computing. Despite tremendous amounts of resources, however, building quantum computers has proven notoriously challenging from both the practical and theoretical perspective. Regarding the formalism of quantum circuits—the arguably best established

mathematical model of quantum computing—major open questions primarily concern exact and approximate circuit synthesis. Published almost 30 years ago, the Solovay–Kitaev theorem [3], [4] seminally delineated a synthesis method of feasible computational complexity. Soon after, Harrow et al. [5] published their famous spectral gap theorem on efficient quantum compiling [5]: Given a gate family that exhibits a spectral gap, random quantum circuits of increasing length rapidly approach the uniform Haar-measure in their distribution. When this is the case, we can approximate any unitary transformation by means of a quantum circuit whose length is proportional to the required accuracy’s logarithm.

Ever since, quantum circuit synthesis has been an active field of research. Nevertheless, its core problems are far from being solved satisfactorily (see, e.g., [6]). To the best of the authors’ knowledge, the spectral gap conjecture—the hypothesis that all universal gate families exhibit a spectral gap—remains open. Moreover, quantum big-O analysis—the standard mathematical framework for assessing a gate family’s asymptotic compiling performance—characterizes efficiency only up to multiplicative coefficients that depend on each individual gate family. As indicated by Harrow et al. [5], however, these admissible big-O coefficients are decisive concerning the relevant gate family’s suitability for practical circuit synthesis. The ability to determine admissible big-O coefficients for arbitrary gate families would likely provide major improvements in gate-agnostic quantum compiling.

Quantum circuit synthesis is an entirely classical task. Accordingly, we must consider its contemporary challenges in the context of digital computing. Almost all relevant publications implicitly rely on the blum-shub-smale (BSS)/real-RAM framework of computability [7], [8], which builds upon infinitely precise real-valued arithmetic operations and computer memory. First and foremost, the BSS/real-RAM framework thus serves as a heuristic approach to practical numerics [9] (see for [10, Ch. 1.4, p. 26ff] for an explicit example). Notably, Dawson and Nielsen [4] hinted toward this issue in their seminal publication on the Solovay–Kitaev “algorithm”. Much in contrast, *Turing’s* framework of computability provides the definitive exact mathematical characterization of digital computing. If it turns out infeasible to solve some mathematical problem by means of a Turing machine, it is definitely infeasible to solve that problem by means of a real-world digital computer.

In prior publications [11], [12], we have investigated approximate quantum circuit synthesis with regards to Turing computability. The present article extends our studies to exact quantum circuit synthesis and quantum big-O theory. Upon applying Turing’s computability framework to the efficiency analysis of gate families, direct links to the spectral gap conjecture and the theory of approximate t-designs emerge. The rest of this article is organized as follows.

Section II introduces exact and approximate circuit synthesis from the perspective of quantum computer engineering and provides a high-level summary of our contributions.

Sections III and IV are dedicated to preliminaries from computable analysis and establishing an effective quantum big-O formalism, respectively. Particularly, this includes formally defining the notion of computable unitary matrices and gate families. For this purpose, we reiterate some of the definitions and structural results previously given in [11] and [12].

Section V provides our results on exact circuit synthesis: Computing the length of a quantum circuit—or even an upper bound thereof—using only the circuit’s matrix representation is a mathematically infeasible task. Moreover, the relevant theorem immediately provides a strong no-go corollary for the explicit task of exact quantum circuit synthesis.

Sections VI and VII establish our results on approximate circuit synthesis, including a no-go result on algorithmic universality checking. We prove that concerning efficiency in the (computable) general unitary group $U_\mu(N)$, $N \in \mathbb{N} \setminus \{0\}$, computing admissible big-O coefficients using only the relevant gate family’s matrix representation is an infeasible task as well. Within the (computable) special unitary group $SU_\mu(N)$, the same result applies if there exists at least one inefficient (but still universal) computable gate family. Thus, the existence of an algorithm that computes admissible big-O coefficients (concerning $SU_\mu(N)$ -efficiency) would prove the spectral gap conjecture true for computable gate families. In addition, we discuss our findings’ relationship to the theory of approximate t-designs and locally symmetric groups.

Finally, Section VIII concludes this article. Aside from the usual subsumption, it discusses some technical remarks.

Appendices A and B provide additional mathematical auxiliaries and the formal proofs of our results, respectively.

A. NOTATION, NOMENCLATURE, AND GENERAL REMARKS

The letters n , N , m , M , l , L , k , K , j , and J denote generic natural-number variables. We do not reserve any of them for specific quantities; their purpose is always with respect to and unambiguous from the relevant context. In some cases, for example, N will refer to the *dimension* of a Hilbert space, while in other cases, it will refer to the number of members of some gate family (cf., Section IV). For *tuples* of natural numbers, we use $\mathbf{n}, \dots, \mathbf{J}$ (bold font), e.g., $\mathbf{m} = (m_0, m_1, \dots, m_n) \in \mathbb{N}^{n+1}$, $n \in \mathbb{N}$.

Given generic sets \mathcal{A} and \mathcal{B} , a *partial function* is a pair (f, \mathcal{A}) such that $f : D(f) \rightarrow \mathcal{B}$ is a function and $D(f) \subseteq \mathcal{A}$ is a set. With some abuse of notation, we do not distinguish between f and (f, \mathcal{A}) —we merely write $f : \mathcal{A} \rightrightarrows \mathcal{B}$ to indicate the function’s partiality. Moreover, we call $D(f)$ the *domain* of f ; if $D(f) = \mathcal{A}$, we call f a total function. Note that every total function is also partial, but not vice versa.

Consider again a generic set \mathcal{A} and let \mathcal{B} be a subset thereof. Occasionally, we abbreviate $\mathcal{A} \setminus \mathcal{B}$ by $\neg\mathcal{B}$. Specifically, if (f, \mathcal{A}) is a partial function, $\neg D(f)$ always denotes $\mathcal{A} \setminus D(f)$. To avoid ambiguity, we define each other instance of this notational principle explicitly.

We define the modulo division of $n \in \mathbb{N}$ by $m \in \mathbb{N} \setminus \{0\}$ through $n - \max\{ml : l \in \mathbb{N}, ml \leq n\} =: \text{mod}(n, m)$.

If an expression $\text{expr}(\cdot)$ of some form defines the elements of a sequence $(a_n)_{n \in \mathbb{N}}$, we employ the notation

$$(a_n)_{n \in \mathbb{N}} : a_n = \text{expr}(n)$$

for the sequence's definition. For example, if $\text{expr}(\cdot) \equiv \frac{1}{\sqrt{\cdot}}$, we may write $(a_n)_{n \in \mathbb{N}} : a_n = \frac{1}{\sqrt{n}}$.

We employ Ξ and Φ as placeholder symbols. The reader may consider

$$\Xi \equiv \text{U and } \Phi \equiv \phi \quad \Xi \equiv \text{SU and } \Phi \equiv \varphi. \quad (1)$$

within each relevant context (see Section IV for the formal definitions of U, SU, ϕ , and φ). Note that the substitution must be consistent within each context. Particularly, the two different cases in (1) are mutually exclusive.

In the context of formal statements, we highlight conditions or consequences through enumerated lists. Alphabetical labels indicate *conditions*, numerical labels indicate *consequences* (otherwise, we use normal bullet points).

II. MOTIVATION AND PROBLEM FORMALIZATION

Following the engineering approach to (universal) quantum computing, we consider an arbitrary but fixed finite-dimensional Hilbert-space $\mathcal{H} := \text{span}\{|e_1\rangle, \dots, |e_N\rangle\}$ over \mathbb{C} with $\dim \mathcal{H} := N \in \{2^n : n \in \mathbb{N}, n > 0\}$. Further, we require the vectors $|e_1\rangle, \dots, |e_N\rangle$ to form a distinguished orthonormal basis of \mathcal{H} , which we refer to as the *computational basis*. The relevant literature often considers the computational basis analogous to the N different bit strings we may feed to a classical Boolean circuit of width $\log_2 N$, see e.g. [13, Ch. 2, p. 53ff] or [14, Sect. 1.2, p. 13ff]. Particularly, for $1 \leq l \leq \log_2 N$, the linear subspaces

$$\mathbb{O}_l := \text{span}\{|e_m\rangle : N \geq m \geq 1, \dots$$

$$\dots 2^{l-1} > \text{mod}(m-1, 2^l)\}$$

$$\mathbb{I}_l := \text{span}\{|e_m\rangle : N \geq m \geq 1, \dots$$

$$\dots 2^{l-1} \geq \text{mod}(m-1, 2^l)\}$$

correspond to the l th bit being equal to zero or one, respectively. Formally, a quantum computation then realizes an unitary input–output relation

$$\sum_{m=1}^N z_m |e_m\rangle \mapsto \sum_{m=1}^N z_m \underline{u}(\omega) |e_m\rangle, \quad \omega \in \mathfrak{u}^*$$

with $z_1, \dots, z_N \in \mathbb{C}$ satisfying $|z_1|^2 + \dots + |z_N|^2 = 1$ and where $\{\underline{u}(\omega) : \omega \in \mathfrak{u}^*\} =: G(\underline{u})$ is the set of quantum algorithms the hardware is able to execute on the physical layer. Accordingly, the hardware accepts inputs consisting of the following two distinct components.

- A *program* $\omega \in \mathfrak{u}^*$, where \mathfrak{u}^* depends on the hardware under consideration.

- A piece of *data* in the form of a vector $(z_m)_{m=1}^N$ which is subject to the computation.

Note that the set $G(\underline{u})$ equally depends on the chosen hardware. However, as a consequence of the laws of quantum mechanics, it is ultimately a subset of the *general unitary group* U or—depending on the context—a subset of the *special unitary group* SU. To avoid redundancy, we will employ the symbol Ξ as a placeholder for U and SU (cf., (1) in Section I-A). That is, we may consider $\Xi \equiv \text{U}$ or $\Xi \equiv \text{SU}$. In those parts of our analysis that are valid for only one of the two cases, we will continue to use U or SU.

Among mathematical formalizations of quantum computing that fit the above scheme, the quantum circuit model is arguably the most prominent. In this model, the set of programs \mathfrak{u}^* is the set of finite words—each of which we call a quantum circuit—over some finite alphabet \mathfrak{u} . The individual letters of \mathfrak{u} each correspond to an *atomic* physical-layer algorithm, and we refer to such atomic algorithms as *quantum gates*. In other words, a quantum gate is an atomic quantum circuit, a word over \mathfrak{u} consisting of only one letter. Common examples include

$$\begin{aligned} \underline{u}_{\text{HT}}(\text{'Id'}) &:= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & \underline{u}_{\text{HT}}(\text{'H'}) &:= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ \underline{u}_{\text{HT}}(\text{'T'}) &:= \begin{bmatrix} 1 & 0 \\ 0 & \exp(\frac{1}{4}j\pi) \end{bmatrix} \end{aligned}$$

see, e.g., [15], [16], [17], and [18]. The string concatenation of quantum circuits corresponds to the successive execution of physical-layer algorithms, which in turn corresponds to the multiplication of unitary matrices. Formally, we require

$$\underline{u}(\omega_1 \circ \omega_2) = \underline{u}(\omega_1) \underline{u}(\omega_2) \quad \text{for all } \omega_1, \omega_2 \in \mathfrak{u}^*$$

in which case $G(\underline{u})$ forms a monoid. In summary, every word $\omega \in \mathfrak{u}^*$ corresponds to a symbolic quantum circuit, while $\underline{u}(\omega)$ corresponds to the resulting unitary transformation's matrix representation (see, e.g., [14, Sect. 2.1.2, p. 63ff] or [6]). In more abstract terms, we can think of symbolic quantum circuits as the purely syntactic part of quantum computing, whereas matrix representations assign semantic meaning to the relevant symbols [19]. The distinction often omitted, sometimes explicitly (see, e.g., [4]).

In the relevant literature, the term “quantum algorithm” usually does not merely refer to a specific physical-layer program in the above sense. Rather, it refers to (more or less) abstract methods of solving specific mathematical problems, potentially involving a multitude of physical-layer quantum algorithms embedded in classical preprocessing and postprocessing (cf., Fig. 1) [6]. Ultimately, any such method builds upon a matrix-valued mapping $\mathcal{Q} : \mathbb{N} \supseteq \rightarrow \Xi$, $n \mapsto \mathcal{Q}(n)$, that encodes the relevant problem instance $n \in \mathbb{N}$ —for example, a specific sequence of amino acids in the analysis of protein folding [20]—into some unitary transformation $\mathcal{Q}(n)$. The necessity of “compiling” $\mathcal{Q}(n)$ into an executable

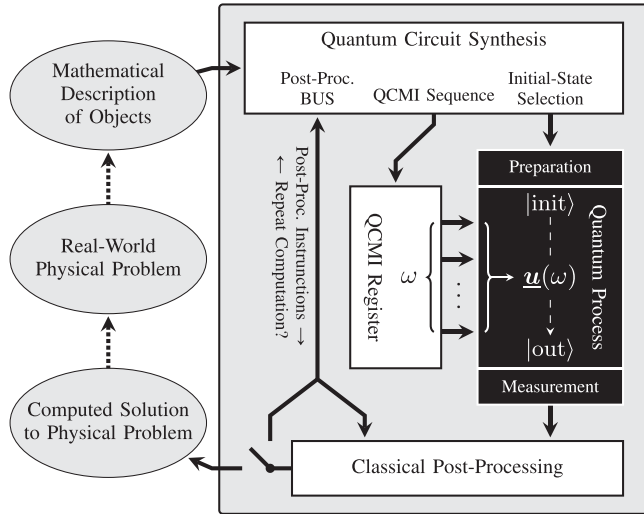


FIGURE 1. Schematics of a gate-based quantum computer. The circuit-synthesis handler receives the description of a quantum algorithm intended to solve some real-world physical problem. The description employs a high-level quantum programming language that hides the underlying computations from the user. Based on the given description, the handler determines suitable post-processing steps, selects an initial state for the available qubit memory, and extracts a chain of quantum-circuit machine instructions (QCMI). The instruction chain corresponds to the successive application of some sequence of quantum gates to the qubit memory, evolving the selected initial state accordingly. Note that the resulting quantum process is the only nonclassical part in the circuit model of quantum computing.

physical layer quantum algorithm leads to exact and approximate quantum circuit synthesis.

Consider \underline{u} , u^* , and \mathcal{Q} . Assume there exists some set $\mathcal{A} \subseteq D(\mathcal{Q})$ such that for all $n \in \mathcal{A}$, we have $\mathcal{Q}(n) \in G(\underline{u})$. Given $n \in \mathcal{A}$, exact circuit synthesis then concerns finding a physical layer program $\omega \in u^*$ that satisfies $\mathcal{Q}(n) = \underline{u}(\omega)$. Note that much in contrast to the unitary group, the set of finite words over a finite alphabet is always countably infinite. Except for the case of application-specific quantum hardware, where \underline{u} is specifically engineered with regards to \mathcal{Q} , we will thus have

$$\{\mathcal{Q}(n) : n \in D(\mathcal{Q})\} \not\subseteq G(\underline{u})$$

for most quantum algorithms. Therefore, one usually chooses \underline{u} universal, i.e., such that $G(\underline{u})$ is topologically dense in Ξ , together with some method of approximate circuit synthesis. Given $M \in \mathbb{N}$ and any $U \in \Xi$, approximate circuit synthesis concerns finding $\omega \in u^*$ such that $\|U - \underline{u}(\omega)\| \leq \frac{1}{2M}$, where $\|\cdot\|$ denotes a suitable operator norm.

On the one hand, exact and approximate circuit synthesis are entirely classical tasks (cf., Fig. 1). On the other hand—as exemplified by the quantum gates $\underline{u}_{\text{HT}}(\text{'H'})$ and $\underline{u}_{\text{HT}}(\text{'T'})$ —the entries of unitary matrices are generally irrational. In fact, as the solution to Schrödinger’s equation involves the matrix exponential, the matrix entries will most likely even be transcendental. Processing arbitrary unitary matrices directly thus requires an infinite amount of memory and

infinitely precise arithmetic operations, both of which are available only within the BSS/real-RAM model of classical computing. Within the Turing framework, we require “machine-readable” counterparts of U and SU : the computable general unitary group U_μ and the computable special unitary group SU_μ , respectively. At the time being, it is sufficient to think of Ξ_μ as comprised of exactly those unitary matrices we can “meaningfully” represent on digital hardware. Note that to the best of the authors’ knowledge, all practically relevant quantum gates correspond to elements of Ξ_μ . This includes (but is not limited to) all gates listed in [14, Nomenclature and Notation, p. xxixff] and [6]. A quantum algorithm \mathcal{Q} then corresponds to a Turing-computable function

$$\text{Qlg}_{\mathcal{Q}} : D(\mathcal{Q}) \rightarrow \Xi_\mu, n \mapsto \text{Qlg}_{\mathcal{Q}}(n) := \mathcal{Q}(n)$$

or, in case we have $D(\mathcal{Q}) = \mathbb{N}$, a computable sequence

$$(U_n)_{n \in \mathbb{N}} : U_n = \mathcal{Q}(n)$$

of (computable) unitary matrices. In other words, for all $n \in D(\mathcal{Q})$, we can algorithmically generate the (computable) matrix representation of $\mathcal{Q}(n)$.

Quantum circuit theory commonly formalizes exact and approximate circuit synthesis via “compiler functions”: abstract mathematical functions that evaluate to a suitable element of u^* or, depending on the context, $G(\underline{u})$ when given $U \in \Xi$. As indicated in Section I, the arguably best-known function of this kind (cf., [4]) builds upon the Solovay–Kitaev theorem. In [11] and [12], we have analyzed compiler functions for approximate circuit synthesis regarding their algorithmic feasibility in Turing’s model of classical computing. The present work extends our analysis to exact circuit synthesis and quantum big-O theory, in addition to some extra results on the problem of universality checking (see Theorem 2). We summarize our core finding on exact circuit synthesis as follows.

Summary 1—Theorem 1: Consider any universal \underline{u} and, for $U \in G(\underline{u})$, denote the length of the shortest ω such that $U = \underline{u}(\omega)$ by $\Omega(U|\underline{u})$. There exists a total quantum algorithm $\mathcal{Q} : \mathbb{N} \rightarrow \Xi_\mu$ that satisfies the following:

- 1) for all $n \in \mathbb{N}$, we have $\mathcal{Q}(n) \in G(\underline{u})$;
- 2) there is no (classical) algorithm $\text{Alg}_{\Omega, \mathcal{Q}} : \mathbb{N} \rightarrow \mathbb{N}$ such that for all $N \in \mathbb{N}$, we have $\Omega(\mathcal{Q}(n)|\underline{u}) \leq \text{Alg}_{\Omega, \mathcal{Q}}(n)$.

Assume \underline{u} and \mathcal{Q} satisfy Theorem 1. Then, given any $n \in \mathbb{N}$, there must exist some symbolic circuit ω such that we have $\mathcal{Q}(n) = \underline{u}(\omega)$. Moreover, $\Omega(\mathcal{Q}(n)|\underline{u})$ must, by definition, be smaller than or equal to the length of any such ω . Accordingly, Theorem 1 provides a strong no-go corollary (see Corollary 1) concerning the explicit task of exact circuit synthesis: even though an *exact* symbolic circuit representation of $\mathcal{Q}(n)$ exists for all $n \in \mathbb{N}$, we cannot generate it algorithmically.

If \underline{u} is universal, $\Omega(\cdot|\underline{u})$ may become arbitrarily large. In contrast, the circuit length required to realize any unitary

transformation in the approximate framework is bounded from above by functions of the required precision. If, up to a *leading coefficient*, some *growth function* $f : \mathbb{N} \rightarrow \mathbb{Q}$ forms an upper bound of this kind, \underline{u} is an element of the corresponding *big-O class* $\mathcal{O}_{\Xi}(f)$. As indicated by Harrow et al. [5], the “admissible” leading coefficients are decisive concerning the suitability of \underline{u} for practical circuit synthesis. We summarize the relevant implications of our core finding on quantum big-O theory as follows.

Summary 2—Theorem 3: Let $f : \mathbb{N} \rightarrow \mathbb{Q}$ be any (computable) growth function such that $\mathcal{O}_{\Xi}(f)$ is nonempty and denote the set of computable real numbers (cf., Section III) by \mathbb{R}_{μ} .

- 1) U-Variant—Given any algorithm $\text{Alg}_{\phi} : D(\text{Alg}_{\phi}) \rightarrow \mathbb{R}_{\mu}$ that satisfies $\mathcal{O}_{\text{U}}(f) \subseteq D(\text{Alg}_{\phi})$, there exists $\underline{u} \in \mathcal{O}_{\text{U}}(f)$ such that $\text{Alg}_{\phi}(\underline{u})$ is not an admissible big-O coefficient for the pair (\underline{u}, f) .
- 2) SU-Variant—Assume there exists an SU-universal \underline{u} that is not an element of $\mathcal{O}_{\text{SU}}(f)$. Then, given any algorithm $\text{Alg}_{\phi} : D(\text{Alg}_{\phi}) \rightarrow \mathbb{R}_{\mu}$ that satisfies $\mathcal{O}_{\text{SU}}(f) \subseteq D(\text{Alg}_{\phi})$, there exists $\underline{v} \in \mathcal{O}_{\text{SU}}(f)$ such that $\text{Alg}_{\phi}(\underline{v})$ is not an admissible big-O coefficient for the pair (\underline{v}, f) .

Note that Summary 1 is, in principle, an accurate description of Theorem 1. In contrast, Summary 2 is a rather imprecise account of Theorem 3, primarily intended to highlight the theorem’s implications for practical quantum computer engineering. A more accurate description would be formally similar to Summary 1, requiring some additional build-up. For brevity, we leave the details to Section VII. However, Summary 2 is sufficient to delineate one of the major consequences of Theorem 3. Consider the linear growth function

$$\chi : \mathbb{N} \rightarrow \mathbb{Q}, n \mapsto \chi(n) := n$$

(note that the growth function’s argument corresponds to the required precision’s *logarithm*). The spectral gap conjecture by Harrow et al. [5] states that \underline{u} is SU-universal if and only if it is computationally efficient, i.e., if and only if we have $\underline{u} \in \mathcal{O}_{\text{SU}}(\chi)$. Thus, if there *does* exist an algorithm that computes admissible big-O coefficients (with respect to computational efficiency), the spectral gap conjecture must be correct within the computable special unitary group.

III. TURING MACHINES, RECURSION THEORY, AND EFFECTIVE ANALYSIS

Subsequently, we provide a brief overview of recursion theory and computable analysis, both of which are fundamental to this article’s contributions. For comprehensive treatments, we refer to [21], [22], [23], [24], [25].

The set of μ -recursive functions [26] is the smallest set of partial functions $g : \mathbb{N} \times \dots \times \mathbb{N} \rightrightarrows \mathbb{N}$ such that

- a) the successor function, all constant functions, and all projection functions are μ -recursive;

- b) composition, primitive recursion, and unbounded search preserve μ -recursivity.

For explicit definitions, see, e.g., [21, Def. 2.1, p. 8, Def. 2.2, p. 10]. Equivalently, a function of the form $g : \mathbb{N} \times \dots \times \mathbb{N} \rightrightarrows \mathbb{N}$ is μ -recursive if and only if it is computable by means of a universal Turing machine [27], [28] (see [29] for the equivalence proof). Turing machines form a mathematical abstraction of digital computing, i.e., they formalize our intuitive understanding of what we call an algorithm. The Church–Turing thesis [21, Author’s Preface, p. ixff], which is widely accepted in the community of computer science, asserts that this formalization is definitive: If, in theory, we cannot solve a mathematical problem by means of a Turing machine, we cannot solve it on a real-world digital computer. Applying the framework of Turing machines and μ -recursive functions to the real and complex numbers leads to the discipline of computable analysis. Often, computable analysis is synonymously referred to as effective analysis.

Due to the very nature of μ -recursive functions, computable analysis requires representing abstract analytical objects by natural numbers. Specifically, we will consider different sets that are each equipped with “formal definitions” of computable sequences. This statement will remain vague to some extent, since we will not go into further detail on the properties such definitions must satisfy. In general, numberings [24, Ch. 1.4, p. 12] and naming systems [24, Def. 2.3.1, p. 33] provide a thorough way of formalizing computable sequences for arbitrary analytically well-defined sets. In modern type theory, a pair consisting of an arbitrary set and an associated numbering is referred to as modest set, see, e.g., [23]. In the context of the present article, none of these concepts will be required. Instead, we will explicitly define the term “computable sequence,” for each of the sets that are relevant to us. Statements about sets that are “equipped with a formal definition of computable sequences” then simply refer to those sets for which we provide such a definition. The first statement of this type will follow right below.

Let \mathcal{A}_{μ} be equipped with a formal definition of computable sequences and consider $m \in \mathbb{N}$ arbitrary. Moreover, consider computable sequences $(a_n)_{n \in \mathbb{N}}$, $(a_n)_{n \in \mathbb{N}^m} \subset \mathcal{A}_{\mu}$ and total μ -recursive functions $g : \mathbb{N}^m \rightarrow \mathbb{N}$, $g_1, \dots, g_m : \mathbb{N} \rightarrow \mathbb{N}$. Then, defining $(\tilde{a}_n)_{n \in \mathbb{N}^m}$, $(\tilde{a}_n)_{n \in \mathbb{N}} \subset \mathcal{A}_{\mu}$ through

$$\tilde{a}_n := a_{g(n)} \quad \tilde{a}_n := a_{g_1(n_1), \dots, g_m(n_m)}$$

provides computable sequences again. Whenever we establish some statement or construction with respect to m -fold computable sequences, we might want to apply a similar statement or construction with respect to l -fold computable sequences in a different context (for $l \in \mathbb{N}$ with $m \neq l$). To this end, we will subsequently make regular implicit use of the above principle throughout the rest of this article.

We will now start introducing the background from computable analysis relevant to our work.

Definition 1: For $m \in \mathbb{N}$, consider an m -fold sequence $(q_n)_{n \in \mathbb{N}^m} \subset \mathbb{Q}$. If there exist (total) μ -recursive functions

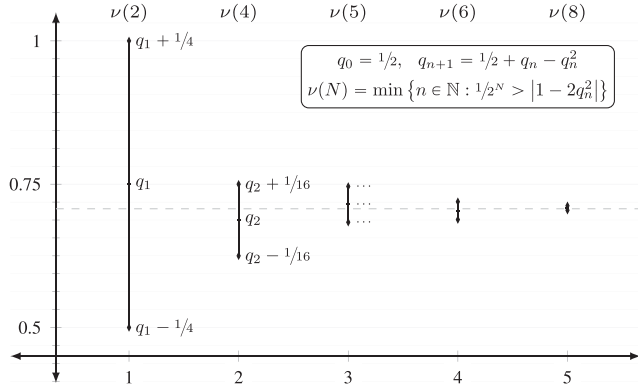


FIGURE 2. Computable number $\frac{1}{\sqrt{2}}$. With $(q_n)_{n \in \mathbb{N}}$ and $v : \mathbb{N} \rightarrow \mathbb{N}$, $N \mapsto v(N)$, as shown, the pair $((q_n)_{n \in \mathbb{N}}, v)$ determines $\frac{1}{\sqrt{2}}$ exactly, i.e., there exists exactly one number $x \in \mathbb{R}$ such that for all $n, N \in \mathbb{N}$, $n \geq v(N)$, we have $|x - q_n| < \frac{1}{2^N}$, and this number satisfies $x^2 = \frac{1}{2}$.

$g_1, g_2, h : \mathbb{N}^m \rightarrow \mathbb{N}$ such that we have

$$q_n = \exp(j\pi g_1(\mathbf{n})) \cdot \frac{g_2(\mathbf{n})}{h(\mathbf{n}) + 1}$$

for all $\mathbf{n} \in \mathbb{N}^m$, we call $(q_n)_{n \in \mathbb{N}^m}$ a computable sequence of rational numbers.

Note that we have $\exp(j\pi n) \in \{-1, 1\}$ for all $n \in \mathbb{N}$. Next, we extend Definition 1 to (general) computable real numbers and computable sequences thereof.

Definition 2: We call $(x_n)_{n \in \mathbb{N}^m} \subset \mathbb{R}$, $m \in \mathbb{N}$, a computable sequence of real numbers if there exists an $(m + 1)$ -fold computable sequence $(q_n)_{n \in \mathbb{N}^{m+1}} \subset \mathbb{Q}$ and a (total) μ -recursive function $v : \mathbb{N}^{m+1} \rightarrow \mathbb{N}$ such that for all $n_0, N \in \mathbb{N}$, $(n_1, \dots, n_m) \in \mathbb{N}^m$, $n_0 \geq v(N, n_1, \dots, n_m)$, we have

$$|x_{n_1, \dots, n_m} - q_{n_0, n_1, \dots, n_m}| < \frac{1}{2^N}.$$

A number $x \in \mathbb{R}$ is called computable real number, written $x \in \mathbb{R}_\mu$, if there exists a computable sequence $(x_n)_{n \in \mathbb{N}} \subset \mathbb{R}$ such that we have $x \in \{x_n\}_{n \in \mathbb{N}}$.

Fig. 2 provides an exemplary account of Definition 2. The depicted sequence is a variation of Heron’s method (after Hero of Alexandria) [30]. To transfer the idea of computable numbers to the complex domain, we introduce the notion of elementary computable sequences of complex numbers.

Definition 3: For $m \in \mathbb{N}$, consider an m -fold sequence $(q_n)_{n \in \mathbb{N}^m} \subset \mathbb{C}$. If there exist (total) μ -recursive functions $g_1, g_2, h : \mathbb{N}^m \rightarrow \mathbb{N}$ such that we have

$$q_n = \exp\left(\frac{j\pi g_1(\mathbf{n})}{h(\mathbf{n}) + 1}\right) \cdot \frac{g_2(\mathbf{n})}{h(\mathbf{n}) + 1}$$

for all $\mathbf{n} \in \mathbb{N}^m$, we call $(q_n)_{n \in \mathbb{N}^m}$ an elementary computable sequence of complex numbers.

We extend Definition 3 to (general) computable complex numbers and computable sequences thereof.

Definition 4: We call $(z_n)_{n \in \mathbb{N}^m} \subset \mathbb{C}$, $m \in \mathbb{N}$, a computable sequence of complex numbers if there

exists an $(m + 1)$ -fold elementary computable sequence $(q_n)_{n \in \mathbb{N}^{m+1}} \subset \mathbb{C}$ and a (total) μ -recursive function $v : \mathbb{N}^{m+1} \rightarrow \mathbb{N}$ such that for all $n_0, N \in \mathbb{N}$, $(n_1, \dots, n_m) \in \mathbb{N}^m$, $n_0 \geq v(N, n_1, \dots, n_m)$, we have

$$|z_{n_1, \dots, n_m} - q_{n_0, n_1, \dots, n_m}| < \frac{1}{2^N}.$$

A number $z \in \mathbb{C}$ is called computable complex number, written $z \in \mathbb{C}_\mu$, if there exists a computable sequence $(z_n)_{n \in \mathbb{N}} \subset \mathbb{C}$ such that we have $z \in \{z_n\}_{n \in \mathbb{N}}$.

As hinted at above, Turing-computable functions conform to our intuitive understanding of algorithms: rule-based sequential procedures that accept inputs of some type and (potentially) return outputs of some other type. Nevertheless, we will instead primarily consider Banach–Mazur computability [22] throughout this article. Rather than asking if some function itself is computable, Banach–Mazur computability concerns the question of whether the function preserves (sequential) computability. Turing and Banach–Mazur computability are closely related, which we discuss in detail further below. Moreover, we will define (Banach–Mazur) semi-decidability, which concerns the sequential verifiability of mathematical properties. The relation between Banach–Mazur (semi-)decidability and its traditional counterpart [24] is analogous to the relation between Banach–Mazur and Turing computability.

Definition 5: For $m \in \mathbb{N}$, let $\mathcal{A}_{1, \mu}, \dots, \mathcal{A}_{m, \mu}$, and \mathcal{B}_μ be equipped with a formal definition of computable sequences. A function $f : \mathcal{A}_{1, \mu} \times \dots \times \mathcal{A}_{m, \mu} \supseteq \rightarrow \mathcal{B}_\mu$ is called Banach–Mazur computable if, for all computable sequences

$$(a_{1, n})_{n \in \mathbb{N}} \subset \mathcal{A}_{1, \mu}, \dots, (a_{m, n})_{n \in \mathbb{N}} \subset \mathcal{A}_{m, \mu}$$

there exists an m -fold computable sequence $(b_n)_{n \in \mathbb{N}^m} \subset \mathcal{B}_\mu$ such that for all $\mathbf{n} = (n_1, \dots, n_m) \in \mathbb{N}^m$, we have

$$b_n = \begin{cases} f(a_{1, n_1}, \dots, a_{m, n_m}), \dots \\ \dots \text{ if } (a_{1, n_1}, \dots, a_{m, n_m}) \in D(f) \\ \text{arbitrary, otherwise.} \end{cases}$$

Note that due to its applicability to partial functions, Definition 5 is somewhat more general than its traditional counterpart employed in [22]. This indeed introduces an inconvenient artifact, since we do require b_n to be well-defined for all $\mathbf{n} \in \mathbb{N}^m$: Even if the partial function $f : \mathcal{A}_{1, \mu} \times \dots \times \mathcal{A}_{m, \mu} \supseteq \rightarrow \mathcal{B}_\mu$ with domain $D(f)$ is not Banach–Mazur computable, its restriction to $D(f)$ —the total function

$$\tilde{f} : D(\tilde{f}) \rightarrow \mathcal{B}_\mu, \mathbf{a} \mapsto \tilde{f}(\mathbf{a}) := f(\mathbf{a})$$

with $D(\tilde{f}) = D(f)$ —might be. However, this artifact will neither be problematic nor of other relevance to us. Note that if f is a total function to begin with, our notion of Banach–Mazur computability coincides with its traditional variant.

For the sake of completeness and for maintaining a uniform nomenclature and notation, we will define computable sequences of natural numbers at this point. We call $(m_n)_{n \in \mathbb{N}} \subset \mathbb{N}$ a computable sequence of natural numbers

if there exists a total μ -recursive function $g : \mathbb{N} \rightarrow \mathbb{N}$ such that we have $m_n = g(n)$ for all $n \in \mathbb{N}$. The reader may convince themselves that according to this definition, a function $f : \mathbb{N} \rightarrow \mathcal{A}_\mu$ is Banach–Mazur computable if and only if $(a_n)_{n \in \mathbb{N}} : a_n = f(n)$ is a computable sequence in \mathcal{A}_μ .

Definition 6: Let \mathcal{A}_μ be equipped with a formal definition of computable sequences and consider $\mathcal{A}_{1,\mu} \subseteq \mathcal{A}_\mu$, $\mathcal{A}_{2,\mu} \subseteq \mathcal{A}_\mu$, $\mathcal{A}_{1,\mu} \cap \mathcal{A}_{2,\mu} = \emptyset$, such that the following holds true.

- a) For all computable sequences $(a_n)_{n \in \mathbb{N}} \subset \mathcal{A}_{1,\mu} \cup \mathcal{A}_{2,\mu}$, there exists a μ -recursive function $g : \mathbb{N} \supseteq \rightarrow \mathbb{N}$ satisfying $D(g) = \{n \in \mathbb{N} : a_n \in \mathcal{A}_{1,\mu}\}$.

We call $\mathcal{A}_{1,\mu}$ (Banach–Mazur) semi-decidable versus $\mathcal{A}_{2,\mu}$, written $\mathcal{A}_{1,\mu} \xrightarrow{\mu} \mathcal{A}_{2,\mu}$. If $\mathcal{A}_{1,\mu}$ is not (Banach–Mazur) semi decidable versus $\mathcal{A}_{2,\mu}$, we write $\mathcal{A}_{1,\mu} \not\xrightarrow{\mu} \mathcal{A}_{2,\mu}$.

In other words, we have $\mathcal{A}_{1,\mu} \xrightarrow{\mu} \mathcal{A}_{2,\mu}$ if, for all computable sequences $(a_n)_{n \in \mathbb{N}} \subset \mathcal{A}_{1,\mu} \cup \mathcal{A}_{2,\mu}$, the set $\{n \in \mathbb{N} : a_n \in \mathcal{A}_{1,\mu}\}$ is recursively enumerable. Accordingly, whenever we have $\mathcal{A}_\mu = \mathbb{N}$ and $\mathcal{A}_{2,\mu} = \mathbb{N} \setminus \mathcal{A}_{1,\mu}$, the notions of recursive enumerability (cf., Appendix A1) and Banach–Mazur semi decidability coincide.

Next, we introduce the concepts of effective closedness and effective denseness. This this end, we consider Banach–Mazur computable (pseudo)metrics $d_A : \mathcal{A}_\mu \times \mathcal{A}_\mu \rightarrow \mathbb{R}_\mu$. Note that we always assume $D(d_A) = \mathcal{A}_\mu \times \mathcal{A}_\mu$, as properly partial (pseudo)metrics would lead to inconsistencies.

Definition 7: Let \mathcal{A}_μ be equipped with a formal definition of computable sequences and a Banach–Mazur computable metric $d_A : \mathcal{A}_\mu \times \mathcal{A}_\mu \rightarrow \mathbb{R}_\mu$. Further, assume the following for every computable double sequence $(a_{n,m})_{n,m \in \mathbb{N}} \subset \mathcal{A}_\mu$.

- a) If there exists a (total) μ -recursive function $\nu : \mathbb{N}^2 \rightarrow \mathbb{N}$ such that for all $n, m, l, K \in \mathbb{N}$, we have

$$d_A(a_{n,\nu(n,K)+m}, a_{n,\nu(n,K)+l}) < 2^{-K}$$

there also exists a computable sequence $(a_{n,*})_{n \in \mathbb{N}} \subset \mathcal{A}_\mu$ satisfying $a_{n,*} = \lim_{m \rightarrow \infty} a_{n,m}$ for all $n \in \mathbb{N}$.

Then, we call \mathcal{A}_μ effectively closed with respect to d_A . Further, we say that sequence $(a_{n,m})_{n,m \in \mathbb{N}}$ converges effectively toward $(a_{n,*})_{n \in \mathbb{N}}$ for $m \rightarrow \infty$, and we call ν a μ -recursive modulus of convergence for $(a_{n,m})_{n,m \in \mathbb{N}}$.

Definition 8: Let \mathcal{A}_μ be equipped with a formal definition of computable sequences and a Banach–Mazur computable (pseudo)metric $d_A : \mathcal{A}_\mu \times \mathcal{A}_\mu \rightarrow \mathbb{R}_\mu$. Further, consider sets $\mathcal{A}_{1,\mu} \subseteq \mathcal{A}_\mu$, $\mathcal{A}_{2,\mu} \subseteq \mathcal{A}_\mu$ that satisfy the following.

- a) We have $\mathcal{A}_{1,\mu} \cap \mathcal{A}_{2,\mu} = \emptyset$ and for all $a_1 \in \mathcal{A}_{1,\mu}$, $a_2 \in \mathcal{A}_{2,\mu}$, we have $d_A(a_1, a_2) > 0$.
- b) For every computable sequence $(a_{n,*})_{n \in \mathbb{N}} \subset \mathcal{A}_{2,\mu}$, there exists a computable double sequence $(a_{n,m})_{n,m \in \mathbb{N}} \subset \mathcal{A}_{1,\mu}$ and a (total) μ -recursive function $\nu : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ such that for all $n, m, M \in \mathbb{N}$, $m \geq \nu(M)$, we have $d_A(a_{n,*}, a_{n,m}) < 2^{-M}$.

We call $\mathcal{A}_{1,\mu}$ effectively dense in $\mathcal{A}_{2,\mu}$ with respect to d_A . If, in addition, $\mathcal{A}_{2,\mu}$ is the largest such set and \mathcal{A}_μ is effectively closed with respect to d_A , we call $\text{efc}\mathcal{A}_{1,\mu} := \mathcal{A}_{1,\mu} \cup \mathcal{A}_{2,\mu}$ the effective closure of $\mathcal{A}_{1,\mu}$.

Note that if \mathcal{A}_μ is effectively closed with respect to d_A and $\mathcal{A}_{1,\mu}$ is any subset of \mathcal{A}_μ , there necessarily exists a unique (but possibly empty) set $\mathcal{A}_{2,\mu} \subseteq \mathcal{A}_\mu$ such that we have $\text{efc}\mathcal{A}_{1,\mu} = \mathcal{A}_{1,\mu} \cup \mathcal{A}_{2,\mu}$. Furthermore, $\text{efc}\mathcal{A}_{1,\mu}$ is always effectively closed with respect to d_A , i.e., we have $\text{efc}(\text{efc}\mathcal{A}_{1,\mu}) = \text{efc}\mathcal{A}_{1,\mu}$. Henceforth, if the relevant (pseudo)metric is unambiguous from the context, we will do without its explicit mention in statements on effective closedness and effective denseness.

The subsequent Summary 3 fulfills two distinct purposes. On the one hand, it aims to help the reader establish an intuition concerning the computable real and complex numbers' mathematical structure. On the other hand, we will explicitly use most of the listed properties in the proofs of our results.

Summary 3—Properties of \mathbb{R}_μ and \mathbb{C}_μ : *Part 1.* The set \mathbb{R}_μ contains all algebraic numbers, as well as the numbers e and π . For all $x, y \in \mathbb{R}_\mu$, we have $x + jy \in \mathbb{C}_\mu$. Given $z \in \mathbb{C}_\mu$, there exist $x, y \in \mathbb{R}_\mu$ satisfying $z = x + jy$.

Part 2. Assume $(q_n)_{n \in \mathbb{N}} \subset \mathbb{Q}$ is a computable sequence of rational numbers. Then, $(q_n)_{n \in \mathbb{N}}$ is also a computable sequence of real numbers. Analogously, assume $(q_n)_{n \in \mathbb{N}} \subset \mathbb{C}$ is an elementary computable sequence of complex numbers. Then, $(q_n)_{n \in \mathbb{N}}$ is also a computable sequence of complex numbers.

Part 3. The sets \mathbb{R}_μ and \mathbb{C}_μ are effectively closed with respect to the metric induced by the absolute-value norm.

Part 4. Real-valued addition and real-valued multiplication are Banach–Mazur computable (with domain $\mathbb{R}_\mu \times \mathbb{R}_\mu$). Likewise, complex-valued addition and complex-valued multiplication are Banach–Mazur computable (with domain $\mathbb{C}_\mu \times \mathbb{C}_\mu$).

Part 5. The function $\exp : \mathbb{C}_\mu \rightarrow \mathbb{C}_\mu$, $z \mapsto \exp(z) := e^z$ is Banach–Mazur computable (with domain \mathbb{C}_μ).

Part 6. Let $(x_{n,m})_{n,m \in \mathbb{N}} \subset \mathbb{R}_\mu$ be a computable sequence of real numbers. Then, $(y_{n,m})_{n,m \in \mathbb{N}} : y_{n,m} = \min_{0 \leq l \leq m} x_{n,l}$ is a computable sequence of real numbers as well. Upon substituting max for min, the statement holds analogously.

Part 7. For $(x_n)_{n \in \mathbb{N}} \subset \mathbb{R}$, let $(q_{n,m})_{n,m \in \mathbb{N}}$, $(p_{n,m})_{n,m \in \mathbb{N}} \subset \mathbb{Q}$ be computable sequences that satisfy $q_{n,m} \leq q_{n,m+1} < x_n < p_{n,m+1} \leq p_{n,m}$ and $p_{n,m} \leq q_{n,m} + \frac{1}{2^m}$ for all $n, m \in \mathbb{N}$. Then, and only then is $(x_n)_{n \in \mathbb{N}}$ a computable sequence of computable real numbers.

We conclude Section III by a brief discussion on the relationship between Turing and Banach–Mazur computability. Specifically, Turing compatibility implies Banach–Mazur computability (but not vice versa). If some function $f : \mathcal{A}_\mu \rightarrow \mathcal{B}_\mu$ is not Banach–Mazur computable, it is not Turing computable either. Each no-go result on Banach–Mazur computability we establish in this

article thus implies the analogous no-go result on Turing computability.

Both Turing and Banach–Mazur computability emerge from the model of Turing machines. Henceforth, we will use the terms ‘Turing machine’ and ‘algorithm’ synonymously. Let \mathcal{A}_μ , and \mathcal{B}_μ be equipped with a formal definition of computable sequences and let $f : \mathcal{A}_\mu \rightarrow \mathcal{B}_\mu$ be a function. Intuitively, a computable sequence $(a_n)_{n \in \mathbb{N}} \subset \mathcal{A}_\mu$ corresponds to an algorithm that successively generates the objects $a_0, a_1, a_2, \dots \in \mathcal{A}_\mu$. That is, a Turing machine

$$\text{Alg}_A : \mathbb{N} \rightarrow \mathcal{A}_\mu, n \mapsto \text{Alg}_A(n) := a_n.$$

Accordingly, we call f Banach–Mazur computable if, for each such sequence, there exists a second algorithm

$$\text{Alg}_B : \mathbb{N} \rightarrow \mathcal{B}_\mu, n \mapsto \text{Alg}_B(n) := f(a_n)$$

that successively generates the corresponding function values $f(a_0), f(a_1), f(a_2), \dots \in \mathcal{B}_\mu$. In contrast, call the function f Turing computable if it is possible to realize f itself through an algorithm, i.e., if there exists a Turing machine

$$\text{Alg}_f : \mathcal{A}_\mu \rightarrow \mathcal{B}_\mu, a \mapsto \text{Alg}_f(a) := f(a)$$

that directly accepts objects $a \in \mathcal{A}_\mu$ as input and returns the corresponding function values $f(a) \in \mathcal{B}_\mu$ as output. Note that formally, Turing machines cannot accept abstract objects natively. Thus—identifying Turing machines with μ -recursive functions—realizing f through an algorithm requires encoding the elements of \mathcal{A}_μ and \mathcal{B}_μ into the natural numbers in a suitable way. For brevity, we leave the details to, e.g., [23], [24], and [25]. With some degree of informality, we define

$$\text{TM}(\mathcal{A}_\mu, \mathcal{B}_\mu) := \{f : \text{there exists a Turing machine} \dots$$

$$\dots \text{Alg}_f \text{ that accepts any } a \in \mathcal{A}_\mu \text{ and returns } f(a) \in \mathcal{B}_\mu \}$$

i.e., $\text{TM}(\mathcal{A}_\mu, \mathcal{B}_\mu)$ is the set of Turing-computable functions that map the set \mathcal{A}_μ to the set \mathcal{B}_μ . If $f : \mathcal{A}_\mu \rightarrow \mathcal{B}_\mu$ is not an element of $\text{TM}(\mathcal{A}_\mu, \mathcal{B}_\mu)$, no algorithmic implementation of f can exist: for any Turing machine supposedly realizing f , there exist values $a \in \mathcal{A}_\mu$ such that the machine’s output either differs from $f(a)$ or is not well defined at all.

As indicated above, Turing computability implies Banach–Mazur computability (the converse is not generally true). In the following, we will provide a brief intuitive explanation for this relationship. Let \mathcal{A}_μ , $\mathcal{B}_{1,\mu}$, and $\mathcal{B}_{2,\mu}$ be equipped with a formal definition of computable sequences. Furthermore, consider $g_1 \in \text{TM}(\mathcal{A}_\mu, \mathcal{B}_{1,\mu})$, $f \in \text{TM}(\mathcal{B}_{1,\mu}, \mathcal{B}_{2,\mu})$, and

$$g_2 : \mathcal{A}_\mu \rightarrow \mathcal{B}_{2,\mu}, a \mapsto g_2(a) := f(g_1(a))$$

arbitrary. Then, conforming to our intuition, we also have $g_2 \in \text{TM}(\mathcal{A}_\mu, \mathcal{B}_{2,\mu})$. Given an algorithm that implements g_1 and an algorithm that implements f , passing the first algorithm’s output to the second algorithm results in a third algorithm that implements the composition of g_1 and f . Next, let $(b_n)_{n \in \mathbb{N}} \subset \mathcal{B}_{1,\mu}$ be any sequence and consider $\mathcal{A}_\mu \equiv \mathbb{N}$.

From above, recall that any computable sequence $(b_n)_{n \in \mathbb{N}} \subset \mathcal{B}_{1,\mu}$ corresponds to an algorithm that successively generates the objects $b_0, b_1, b_2, \dots \in \mathcal{B}_{1,\mu}$. In other words, the sequence $(b_n)_{n \in \mathbb{N}} \subset \mathcal{B}_{1,\mu}$ is computable if and only if there exists $g_1 \in \text{TM}(\mathbb{N}, \mathcal{B}_{1,\mu})$ such that for all $n \in \mathbb{N}$, we have $g_1(n) = b_n$. In this case, if f is Turing computable, we also have $g_2 \in \text{TM}(\mathbb{N}, \mathcal{B}_{2,\mu})$ for

$$g_2 : \mathbb{N} \rightarrow \mathcal{B}_{2,\mu}, n \mapsto g_2(n) := f(g_1(n)) = f(b_n).$$

In other words, the sequence $(f(b_n))_{n \in \mathbb{N}} \subset \mathcal{B}_{2,\mu}$ is computable in turn. Since we chose $(b_n)_{n \in \mathbb{N}} \subset \mathcal{B}_{1,\mu}$ arbitrarily, we conclude that f must not only be Turing but also Banach–Mazur computable.

IV. COMPUTABLE UNITARY MATRICES, COMPUTABLE GATE FAMILIES, AND COMPUTATIONAL UNIVERSALITY

To analyze gate-based quantum computing in the context of Turing machines, we must provide an effective-analysis framework of quantum circuits, the unitary group, and universality. As indicated in Section II, we consider a fixed Hilbert-space $\mathcal{H} := \text{span}_{\mathbb{C}}\{|e_1\rangle, \dots, |e_N\rangle\}$, $N \in \mathbb{N} \setminus \{0\}$, with computational basis $|e_1\rangle, \dots, |e_N\rangle$. Since \mathcal{H} is finite dimensional, it is isometric to the set

$$\mathbb{C}^N := \{(z_n)_{n=1}^N : \{z_n\}_{n=1}^N \subset \mathbb{C}\}$$

with the usual component wise addition and multiplication. Accordingly, we will tacitly consider all elements of \mathcal{H} complex-valued N -tuples throughout this article. Likewise, the set $\mathbb{L}(\mathcal{H})$ of linear operators acting on \mathcal{H} is isometric to the set of complex-valued $(N \times N)$ -matrices

$$\mathbb{C}^{N \times N} := \{(z_{n,m})_{n,m=1}^N : \{z_{n,m}\}_{n,m=1}^N \subset \mathbb{C}\}$$

and we will tacitly consider all elements of $\mathbb{L}(\mathcal{H})$ complex-valued $(N \times N)$ -matrices throughout this article. Furthermore, all of our analysis is with respect to the standard matrix norm

$$\|\cdot\| : \mathbb{C}^{N \times N} \rightarrow \mathbb{R}, \mathbf{Z} \mapsto \|\mathbf{Z}\|.$$

Observe, however, that this is primarily due to convention; for any equivalent matrix norm, our results are equally valid.

By \mathbf{Z}^\dagger and \mathbf{Z}^{-1} , we denote the hermitian adjoint and, provided it exists, the *inverse* of $\mathbf{Z} \in \mathbb{C}^{N \times N}$, respectively. If we have $\text{Id} = \mathbf{U}^\dagger \mathbf{U}$ for $\mathbf{U} \in \mathbb{C}^{N \times N}$ —where

$$\text{Id} := \sum_{n=1}^N |e_n\rangle\langle e_n| \in \mathbb{C}^{N \times N}$$

is the identity matrix—we call \mathbf{U} a unitary matrix. In other words, \mathbf{U} is a unitary matrix if and only if it satisfies $\mathbf{U}^\dagger = \mathbf{U}^{-1}$. Denoting the determinant of $\mathbf{U} \in \mathbb{C}^{N \times N}$ by $\det \mathbf{U}$, we define

$$\text{U}(N) := \{\mathbf{U} \in \mathbb{C}^{N \times N} : \mathbf{U}^\dagger = \mathbf{U}^{-1}\}$$

$$\text{SU}(N) := \{\mathbf{U} \in \text{U}(N) : \det \mathbf{U} = 1\}.$$

We refer to the sets $\text{U}(N)$ and $\text{SU}(N)$ as the general unitary group and the special unitary group, respectively. If

$N = \dim \mathcal{H}$ is either clear from the context or irrelevant, we simply write U and SU for $U(N)$ and $SU(N)$. Moreover, recall that whenever the distinction between U and SU is unnecessary, we will use Ξ as a placeholder referring to both variants of the unitary group [cf., (1) in Section I-A].

As a fundamental quantum–physical principle, it is impossible to determine a quantum gate’s global phase $\theta \in [0, 1)$. The global phase relates to the determinant of the corresponding unitary matrix $U \in U$ by $\det U = \exp(j2\pi\theta)$. Thus, the quantum gates $U \in U$ and $(\det U)^{-\frac{1}{N}} \cdot U \in SU$ are physically indistinguishable. Accordingly, large parts of the theory on efficient universal quantum computing consider the special unitary group SU to begin with, see, e.g. [4], [5], [31], [32], [33], [34], [35], and [36]. In contrast, the general unitary group U is often considered for rather mathematical purposes or mere convenience, see, e.g., [18], [37], [38], and [39]. In the engineering context, the general unitary group does exhibit a physical interpretation nevertheless, which we will briefly discuss in Section VII-B. Finally, note that the distinction between U and SU is sometimes implicit in the relevant literature.

Analyzing gate-based quantum computing in the context of Turing machines requires an effective variant of U and SU : the computable general unitary group U_μ and the computable special unitary group $SU_\mu := U_\mu \cap SU$, respectively. In the finite-dimensional setting, defining U_μ and SU_μ on the basis of computable sequences of computable numbers is straightforward and in line with M. B. Pour-El and J. I. Richards’ fundamental method of computability structures [22, Ch. 2.1, p. 80ff] for Banach- and Hilbert-spaces.

Definition 9: For $m, N \in \mathbb{N}$, let $(U_n)_{n \in \mathbb{N}^m} \subset \Xi(N)$ be any sequence and consider $|e_1\rangle, \dots, |e_N\rangle$ as above. If each of the sequences $(z_{l,k,n})_{n \in \mathbb{N}^m} : z_{l,k,n} = \langle e_l | U_n | e_k \rangle$, $1 \leq l, k \leq N$, is computable, we call $(U_n)_{n \in \mathbb{N}^m}$ a computable sequence of unitary matrices. Moreover

- we call $U \in \Xi(N)$ a computable unitary matrix if there exists a computable sequence $(U_n)_{n \in \mathbb{N}}$ of unitary matrices such that we have $U \in \{U_n\}_{n \in \mathbb{N}}$, and
- we denote the set of such matrices by $\Xi_\mu(N)$.

Much like Definitions 2 and 4, Definition 9 provides a method of making abstract mathematical objects—in this case, unitary matrices—accessible to Turing machines. The justification of Definition 9 is operational: Ξ_μ is the unique subset of Ξ that satisfies the axioms listed in Summary 4 below, and thus maintains the algebraic structure of Ξ within the computable domain. Considering any other unitary matrices in the context of Turing’s computability theory comes at the price of violating at least one of the listed axioms. For details, we refer to [11], [12].

Summary 4—Axioms of U_μ and SU_μ . Part 1. Set $d_\Xi(U, V) := \|U - V\|$. Then, $d_\Xi : \Xi_\mu \times \Xi_\mu \rightarrow \mathbb{R}_\mu$ shall be a Banach–Mazur computable metric (see also §3).

Part 2. For $1 \leq l, k \leq \dim \mathcal{H}$, define $\beta_{l,k}(U) := \langle e_l | U | e_k \rangle$. Then, $\beta_{l,k} : \Xi_\mu \rightarrow \mathbb{C}_\mu$ shall be Banach–Mazur computable.

Part 3. The group operation of Ξ_μ (matrix multiplication) shall be iteratively computable. That is, given any computable sequence $(V_n)_{n \in \mathbb{N}} \subset \Xi_\mu$, there shall exist another computable sequence $(U_n)_{n \in \mathbb{N}} \subset \Xi_\mu$ and μ -recursive functions $g_1 : \mathbb{N} \rightarrow \mathbb{N}$, $g_2 : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$, such that for all $n, m, l \in \mathbb{N}$, we have $V_n = U_{g_1(n)}$ and $U_m U_l = U_{g_2(m,l)}$.

Part 4. The set Ξ_μ shall be effectively closed: All computable sequences $(U_n)_{n \in \mathbb{N}} \subset \Xi_\mu$, shall satisfy $\text{efc}\{U_n\}_{n \in \mathbb{N}} \subseteq \Xi_\mu$. Moreover, we shall have $\text{efc}\{U_n\}_{n \in \mathbb{N}} = \Xi_\mu$ if and only if $\{U_n\}_{n \in \mathbb{N}}$ is dense (in the traditional sense of topology and with respect to d_Ξ) in Ξ .

Despite computable unitary matrices conceptually concerning classical computing theory, they are fundamental to the idea of “general” quantum algorithms as considered in [13] and [14]. In what follows, we provide a detailed account of this relationship. To this end, recall Section II and let u^* denote a set of symbolic quantum circuits. Specifically, assume u^* be the set of finite words over some alphabet $u = \{\mathcal{U}_0, \dots, \mathcal{U}_N\}$, $N \in \mathbb{N}$. Moreover, let

$$G(\underline{u}) := \{\underline{u}(\omega) : \omega \in u^*\} \subset \Xi$$

characterize the corresponding set of physical-layer quantum algorithms some idealized quantum hardware platform is capable of executing. Then, $G(\underline{u})$ is the smallest set such that

- $\underline{u}(\mathcal{U}_0), \dots, \underline{u}(\mathcal{U}_N)$ are elements of $G(\underline{u})$;
- for all $U, V \in G(\underline{u})$, we have $UV \in G(\underline{u})$.

In mathematical terminology, the matrices $\underline{u}(\mathcal{U}_0), \dots, \underline{u}(\mathcal{U}_N)$ generate the monoid $G(\underline{u})$. From a practical perspective, $G(\underline{u})$ corresponds to the unitary transformations the hardware platform implements exactly. Given any unitary transformation $U \in G(\underline{u})$, the theory of exact circuit synthesis—which we will primarily consider in Section V—is concerned with finding a word $\omega \in u^*$ such that $U = \underline{u}(\omega)$. Note that $G(\underline{u})$ depends on the matrices $\underline{u}(\mathcal{U}_0), \dots, \underline{u}(\mathcal{U}_N)$. In other words, the set of available physical-layer quantum algorithms is specific to the given hardware platform.

Limiting the scope of gate-based quantum computing to exact circuit synthesis is, in most cases, neither feasible nor necessary. Assume for some $U \in \Xi$ and all $n \in \mathbb{N}$, we are able to (classically) compute a symbolic quantum circuit $\omega_n \in u^*$ that satisfies $\|U - \underline{u}(\omega_n)\| < \frac{1}{2^n}$. Accordingly, the computable sequence $(\omega_n)_{n \in \mathbb{N}} \subset u^*$ (cf., Section V) uniquely determines the matrix U , regardless of whether we have $U \in G(\underline{u})$ or not. Since no realistic computing scenario will require us to implement U with infinite accuracy, we may consider U implementable for all practical means. Given any $U \in \Xi$, the theory of approximate circuit synthesis is concerned with finding (computable) sequences $(\omega_n)_{n \in \mathbb{N}} \subset u^*$ that determine the matrix U in the above sense.

Consider $U \in \Xi$ and $(\omega_n)_{n \in \mathbb{N}} \subset u^*$ once more. In addition, assume we have $\underline{u}(\mathcal{U}_0), \dots, \underline{u}(\mathcal{U}_N) \in \Xi_\mu$. Then, the sequence $(U_n)_{n \in \mathbb{N}} : U_n = \underline{u}(\omega_n)$ is a computable sequence of computable unitary matrices (cf., Corollary 1). Essentially, this is a consequence of Summary 4, Part 3. Moreover, by

Summary 4, Part 4, we must then have $\underline{U} \in \Xi_\mu$. More generally, whenever we have $\underline{u}(\mathfrak{U}_0), \dots, \underline{u}(\mathfrak{U}_N) \in \Xi_\mu$, we also have $G(\underline{u}) \subset \text{efc}G(\underline{u}) \subseteq \Xi_\mu$. In other words, if the quantum gates specific to some quantum hardware platform correspond to computable unitary matrices, all unitary transformations we can meaningfully implement by means of that hardware are necessarily computable themselves. As indicated in Section II, all practically relevant quantum gates correspond to elements of Ξ_μ , considering the computable unitary group is thus not only a question of mathematical axioms, but also a consequence of contemporary quantum engineering.

Analogously to their purely analytic counterparts, the computable special unitary group and the computable general unitary group entail a notion of (possibly efficient) computational universality and (possibly efficient) strong universality, respectively. Note that the term computational universality is common in the relevant literature (see, e.g., [5]), but unrelated to the notion of computable gate families we will introduce below. Informally speaking, we call a gate family computationally universal if its members allow for approximating any element of the special unitary group by means of a quantum circuit, up to any desired precision. In contrast, a computable gate family simply consists of computable unitary operators. Neither of the two conditions implies the other.

Let $N \in \mathbb{N}$ be arbitrary. Up to this point, our treatment of alphabets $\mathfrak{u} = \{\mathfrak{U}_0, \dots, \mathfrak{U}_N\}$ and monoids $G(\underline{u})$ has been slightly informal. Our subsequent analysis will require a more precise approach, for which it is convenient to start with the set $[N] := \{0, \dots, N\}$ instead of the alphabet \mathfrak{u} . We consider

$$\Xi_{\mu,N} := \{\underline{u} : [N] \rightarrow \Xi_\mu, \underline{u}(0) = \text{Id}\}$$

and call $\underline{u} \in \Xi_{\mu,N}$ a computable gate family consisting of $|\underline{u}| := N$ (not necessarily different) members. In other words, \underline{u} is a mapping from the (finite) set $[N]$ to the computable unitary matrices Ξ_μ . Observe that setting

$$\underline{u}(n) := \mathfrak{U}_n \quad \text{for } n \in [N]$$

provides a one-to-one correspondence between the letters of \mathfrak{u} and the numbers $0, \dots, N$. Accordingly, we define

$$\underline{u}(\omega) := \underline{u}(n_1) \cdots \underline{u}(n_m) \quad \text{for } \omega = (u(n_1), \dots, u(n_m))$$

where $(n_1, \dots, n_m) \in [N]^m$ and $m \in \mathbb{N}$, and refer to the mapping $\omega \mapsto \underline{u}(\omega)$ with domain \mathfrak{u}^* as canonic. Each word $\omega \in \mathfrak{u}^*$ represents a symbolic quantum circuit, while $\underline{u}(\omega) \in \Xi_\mu$ corresponds to the resulting unitary transformation's matrix representation. Evidently, the specific choice of the letters $\mathfrak{U}_0, \dots, \mathfrak{U}_N$ is irrelevant for the analysis of the circuit $\underline{u}(\omega)$. Thus, we may equivalently remove the alphabet \mathfrak{u} altogether. For $m \in \mathbb{N}$, we define

$$\langle N; m \rangle := \{0\} \times \underbrace{[N] \times \cdots \times [N]}_{m\text{-times}}$$

where the leading zero is required for consistency with later definitions. We have $|\langle N; m \rangle| = (N + 1)^m$. Analogously to ω

and $\underline{u}(\omega)$, the tuple $\mathbf{n} = (0, n_1, \dots, n_m) \in \langle N; m \rangle$ represents a (symbolic) quantum circuit, while

$$\underline{u}(\mathbf{n}) := \underline{u}(0)\underline{u}(n_1) \cdots \underline{u}(n_m) = \underline{u}(n_1) \cdots \underline{u}(n_m)$$

corresponds to the resulting unitary transformation in matrix representation. In accordance with Section II, we obtain

$$G(\underline{u}) = \bigcup_{m=0}^{\infty} \{\underline{u}(\mathbf{n}) : \mathbf{n} \in \langle N; m \rangle\}$$

and call \underline{u} a generator family of $G(\underline{u})$. Note that the family \underline{u} uniquely determines the monoid $G(\underline{u})$. In contrast, unless some monoid of unitary matrices is finite, there exists an infinite number of corresponding generator families.

Consider $N \in \mathbb{N}$ and $\underline{u} \in \Xi_{\mu,N}$ once again. The formalism of approximate circuit synthesis provides an operational interpretation of universality: If, for all $\underline{U} \in \Xi_\mu$ and any prescribed accuracy, we can find $m \in \mathbb{N}$ and a quantum circuit $\mathbf{n} \in \langle N; m \rangle$ such that the resulting unitary transformation $\underline{u}(\mathbf{n})$ approximates \underline{U} up to the prescribed accuracy, we call \underline{u} a universal gate family. However, since we ultimately want to assess the gate family's efficiency, we follow a slightly different approach. Define $\eta_\Xi : \Xi_{\mu,N} \times \mathbb{N} \rightarrow \mathbb{R}_\mu$ through

$$\eta_\Xi(\underline{u}, m) := \sup_{\underline{U} \in \Xi_\mu} \inf_{\mathbf{n} \in \langle N; m \rangle} \|\underline{U} - \underline{u}(\mathbf{n})\|. \quad (2)$$

In the scope of approximate circuit synthesis, the function η_Ξ measures the smallest achievable worst-case approximation error in dependence of the underlying gate family and some maximally admissible circuit length. Given \underline{u} , we may thus—somewhat broadly—refer to the sequence $(\eta_\Xi(\underline{u}, m))_{m \in \mathbb{N}}$ as the gate family's compiling performance. At the time being, we are primarily interested in the asymptotic behavior of η_Ξ . To this end, define $\eta_\Xi^\infty : \Xi_{\mu,N} \rightarrow \mathbb{R}_\mu$ through

$$\eta_\Xi^\infty(\underline{u}) := \lim_{l \rightarrow \infty} \eta_\Xi(\underline{u}, l) = \inf_{l \in \mathbb{N}} \eta_\Xi(\underline{u}, l).$$

If the worst-case approximation error as measured by η_Ξ decreases to an arbitrarily low value for an increasing maximum circuit length,—that is, if η_Ξ^∞ equals zero—we call \underline{u} a universal gate family.

Definition 10: For $N \in \mathbb{N}$, we call $\underline{u} \in \Xi_{\mu,N}$ (strongly) universal if we have $\eta_\Xi^\infty(\underline{u}) = 0$, and we define

$$\mathbb{U}_{\mu,N}^\diamond := \{\underline{u} \in \Xi_{\mu,N} : \eta_\Xi^\infty(\underline{u}) = 0\}.$$

Analogously, we call $\underline{u} \in \text{SU}_{\mu,N}$ (computationally) universal if we have $\eta_{\text{SU}}^\infty(\underline{u}) = 0$, and we define

$$\text{SU}_{\mu,N}^\diamond := \{\underline{u} \in \text{SU}_{\mu,N} : \eta_{\text{SU}}^\infty(\underline{u}) = 0\}.$$

Definition 10 ultimately builds upon the standard (or any equivalent) matrix norm $\|\cdot\| : \Xi_\mu \rightarrow \mathbb{R}_\mu$, with large parts of the relevant literature formalizing universality along the same lines [4], [5], [33], [34], [37]. It is, however, worth mentioning that other possible similarity measures in this context exist, such as the gate fidelity [14, Sect. 9.3, p.416ff]. Likewise, it is possible to consider operator spaces other than the special and the general unitary group as mathematical

models for quantum gates. Notable examples include the projective unitary group (cf., e.g., [40], [41], and [42]), which we will briefly discuss in Section VII-B. While the mentioned approaches are largely similar, their differences are often subtle.

Consider $N \in \mathbb{N}$ arbitrary and let $\underline{u} \in \Xi_{\mu,N}$ be any computable gate family. Then, there exists a computable sequence $(U_n)_{n \in \mathbb{N}} \subset \Xi_\mu$ such that for all $m \in \mathbb{N}$, we have

$$\{U_0, \dots, U_{|\langle N; m \rangle| - 1}\} = \{\underline{u}(n) : n \in \langle N; m \rangle\}.$$

For the sake of maintaining a consistent structure, we defer formally discussing the claimed existence of $(U_n)_{n \in \mathbb{N}}$ to Appendix B1. Specifically, Lemma 6 will establish a more general variant of the claim, and we provide a proof afterward. At the time being, it is sufficient to observe that $(U_n)_{n \in \mathbb{N}} \subset \Xi_\mu$ enumerates the monoid $G(\underline{u})$. In informal terms, $(U_n)_{n \in \mathbb{N}} \subset \Xi_\mu$ lists all unitary matrices we can synthesize *exactly* through a finite quantum circuit using gates from the family \underline{u} . Then, in accordance with Definition 10, if $\{U_n\}_{n \in \mathbb{N}}$ is dense (in the usual topological sense) in Ξ_μ , we call the family \underline{u} universal. In consequence, we have

$$\underline{u} \in \Xi_{\mu,N}^\diamond \Leftrightarrow \text{efc}G(\underline{u}) = \Xi_\mu$$

by Summary 4, Part 4. That is, a gate family is universal if and only if it generates a monoid whose effective closure is the computable unitary group.

Observe that for some explicit gate families, the behavior of the worst-case approximation error—that is, the behavior of η_Ξ —for increasing circuit length is known (see [5] and Appendix A2 for details). Hereafter, we will introduce the big-O formalism of quantum circuits: a subject dedicated to studying the asymptotic behavior of η_Ξ for arbitrary gate families $\underline{u} \in \Xi_{\mu,N}^\diamond$. To this end, we consider functions $f : \mathbb{N} \rightarrow \mathbb{Q}$. From Definition 5, it follows that f is (Banach–Mazur) computable if and only if the sequence $(q_n)_{n \in \mathbb{N}} : q_n = f(n)$ is a computable sequence of rational numbers. In order to align with the relevant literature’s standards, we will consider computable functions rather than computable sequences in the big-O formalism’s context. Accordingly, let $(q_n)_{n \in \mathbb{N}}$ be any computable sequence of rational numbers that satisfies the following.

- For all $n \in \mathbb{N}$, we have $0 \leq q_n \leq q_{n+1}$ (i.e., $(q_n)_{n \in \mathbb{N}}$ is nonnegative and nondecreasing).
- For all $q \in \mathbb{Q}$, there exists $n \in \mathbb{N}$ such that we have $q < q_n$ (i.e., $(q_n)_{n \in \mathbb{N}}$ is unbounded).

Then, we call $f : \mathbb{N} \rightarrow \mathbb{Q}$, $n \mapsto f(n) := q_n$, a (Banach–Mazur) computable growth function.

Definition 11: Consider $L \in \mathbb{N}$, $\underline{u} \in \Xi_{\mu,L}^\diamond$, and a computable growth function $f : \mathbb{N} \rightarrow \mathbb{Q}$. Assume the following.

- There exist $q \in \mathbb{Q}$, $M \in \mathbb{N}$, such that for all $n, N \in \mathbb{N}$, $n \geq qf(N + M)$, we have $\eta_\Xi(\underline{u}, n) \leq \frac{1}{2^{(N+M)}}$.

We call $f : \mathbb{N} \rightarrow \mathbb{Q}$ an asymptotic upper bound of \underline{u} in Ξ_μ and write $\underline{u} \in \mathcal{O}_{\Xi,L}(f)$.

Equivalently, we have $\underline{u} \in \mathcal{O}_{\Xi,L}(f)$ if there exist $M \in \mathbb{N}$, $q \in \mathbb{Q}$, such that for all $N \in \mathbb{N}$, $N \geq M$, we have

$$qf(N) > E_\Xi(\underline{u}, N) := \dots$$

$$\dots \max \{n \in \mathbb{N} : -\log_2 \eta_\Xi(\underline{u}, n) < N\}.$$

In the context of traditional big-O analysis, the notation $\underline{u} \in \mathcal{O}_{\Xi,L}(f)$ commonly refers to E_Ξ rather than η_Ξ .

For the rest of Section IV, consider $L \in \mathbb{N}$ arbitrary. Given some computable growth function $f : \mathbb{N} \rightarrow \mathbb{Q}$, we call $\mathcal{O}_{\Xi,L}(f)$ a big-O class. Further, given $\underline{u} \in \mathcal{O}_{\Xi,L}(f)$ and an offset $M \in \mathbb{N}$, we call a prefactor $q \in \mathbb{Q}$ in the sense of Definition 11 an admissible big-O coefficient. Up to linear scaling by an admissible big-O coefficient, f forms a “global” recursive modulus of convergence for the class $\mathcal{O}_{\Xi,L}(f)$. Note that the admissibility of $q \in \mathbb{Q}$ as big-O coefficient for the pair (\underline{u}, f) depends on the chosen offset M .

Definition 11 analogously extends to general nonnegative nondecreasing total functions $F : \mathbb{N} \rightarrow \mathbb{R}$. Assume we can choose the computable growth function $f : \mathbb{N} \rightarrow \mathbb{Q}$ such that for some (total) μ -recursive function $\nu : \mathbb{N} \rightarrow \mathbb{N}$ and all $n, N \in \mathbb{N}$, $n \geq \nu(N)$, we have

$$|F(n) - f(n)| < \frac{1}{2^N}. \quad (3)$$

Then, we also have $\mathcal{O}_{\Xi,L}(F) = \mathcal{O}_{\Xi,L}(f)$. In this case, F does not have to be Banach–Mazur computable— F may even attain uncomputable values for some $n \in \mathbb{N}$, provided it is “asymptotically computable” in the sense of (3).

Finally, consider a computable growth function $f : \mathbb{N} \rightarrow \mathbb{Q}$ once more and let $\underline{u} \in \Xi_{\mu,L}$ be arbitrary but fixed. Observe that \underline{u} is an element of $\mathcal{O}_{\Xi,L}(f)$ if and only if for all offsets $M \in \mathbb{N}$, the set of admissible big-O coefficients is nonempty. Hence, if admissible big-O coefficients exist for some offset $M \in \mathbb{N}$, they must exist for all offsets $M \in \mathbb{N}$. We define $\phi_{f,M} : \mathcal{O}_{U,L}(f) \rightarrow \mathbb{R}$ and $\varphi_{f,M} : \mathcal{O}_{SU,L}(f) \rightarrow \mathbb{R}$ through

$$\phi_{f,M}(\underline{v}) := \inf \left\{ q \in \mathbb{Q} : \eta_U(\underline{v}, n) \leq \frac{1}{2^{(N+M)}} \text{ for all } \dots \right.$$

$$\dots n, N \in \mathbb{N} \text{ that satisfy } n \geq qf(N + M) \left. \right\} \text{ and}$$

$$\varphi_{f,M}(\underline{v}) := \inf \left\{ q \in \mathbb{Q} : \eta_{SU}(\underline{v}, n) \leq \frac{1}{2^{(N+M)}} \text{ for all } \dots \right.$$

$$\dots n, N \in \mathbb{N} \text{ that satisfy } n \geq qf(N + M) \left. \right\}$$

respectively. To avoid redundancy, we will hereafter employ the symbol Φ as a placeholder for ϕ , and φ . As indicated in Section I-A, the reader may either consider $\Xi \equiv U$ and $\Phi \equiv \phi$ or $\Xi \equiv SU$ and $\Phi \equiv \varphi$ within each relevant context. Then, if we have $\underline{u} \in \mathcal{O}_{\Xi,L}(f)$, $\Phi_{f,M}(\underline{u})$ is well defined for all offsets $M \in \mathbb{N}$ and equals the largest lower bound for the relevant set of admissible big-O coefficients.

Given $N, M \in \mathbb{N}$ and some computable growth function $f : \mathbb{N} \rightarrow \mathbb{Q}$, quantum big-O analysis concerns the behavior of $\Phi_{f,M} : \mathcal{O}_{\Xi,N}(f) \rightarrow \mathbb{R}$ for varying arguments. We will thus require a formal definition of computability for sequences $(\underline{u}_n)_{n \in \mathbb{N}} \subset \Xi_{\mu,N}$ and a compatible distance measure

for gate families $\underline{u}, \underline{v} \in \Xi_{\mu, N}$. Assume there exist a computable sequence $(U_n)_{n \in \mathbb{N}} \subset \Xi_{\mu}$ and a μ -recursive function $g : \mathbb{N} \times \mathbb{N} \rightrightarrows \mathbb{N}$ such that for all $(n, m) \in \mathbb{N} \times [N]$, we have

$$(n, m) \in D(g) \quad \underline{u}_n(m) = U_{g(n, m)}.$$

Then, we call $(\underline{u}_n)_{n \in \mathbb{N}}$ a computable sequence of gate families. With some abuse of notation, we define the Banach–Mazur computable metric $d_{\Xi} : \Xi_{\mu, N} \times \Xi_{\mu, N} \rightarrow \mathbb{R}_{\mu}$ through

$$d_{\Xi}(\underline{u}, \underline{v}) := \max_{1 \leq m \leq N} \|\underline{u}(m) - \underline{v}(m)\|.$$

Informally speaking, d_{Ξ} naively extends the $\|\cdot\|$ -induced distance measure of the unitary group to the set $\Xi_{\mu, N}$.

For $N \in \mathbb{N}$, consider gate families $\underline{u}, \underline{v} \in \Xi_{\mu, N}$ once more. From the perspective of engineering and numerical methods, we would like to consider \underline{u} and \underline{v} “practically the same” if their distance does not exceed what we would expect from manufacturing variations, material imperfections, environmental influences, or, in general, any expectable noise-like disturbance of arbitrary origin. The robustness of a gate family’s relevant properties toward such noise-like disturbances is crucial for the predictability of the physical hardware’s behavior *and* the numeric stability of any attempted method of computing the family’s characteristics.

Albeit formally meaningful, the metric d is not a suitable distance measure in the above regard. The families’ practically relevant properties ultimately depend on the corresponding gate sets $\{\underline{u}(0), \dots, \underline{u}(N)\}$ and $\{\underline{v}(0), \dots, \underline{v}(N)\}$, respectively, rather than the literal families themselves. However, even if we have

$$\{\underline{u}(0), \dots, \underline{u}(N)\} = \{\underline{v}(0), \dots, \underline{v}(N)\} \quad (4)$$

we may still have $d(\underline{u}, \underline{v}) > 0$ if, for example, \underline{v} is a permutation of \underline{u} . To provide a suitable modification of d , define

$$d_{\bar{\Delta}}(\underline{u}, \underline{v}) := \min \left\{ \delta \in \mathbb{Q}^+ : \underline{u}(0), \dots, \underline{u}(N) \in \dots \right. \\ \left. \dots \underbrace{\bigcup_{0 \leq m \leq N} \{V \in \Xi_{\mu} : \|\underline{v}(m) - V\| \leq |\delta|\}}_{:= \Delta_{\Xi}(\underline{v}, \delta)} \right\}$$

—if, for $\delta \in \mathbb{Q}^+$, we have $\Delta_{\Xi}(\underline{v}, \delta) = \Xi_{\mu}$, the set $\{\underline{v}(0), \dots, \underline{v}(N)\}$ forms a δ -covering of Ξ_{μ} —and, analogously, $d_{\bar{\Delta}}(\underline{v}, \underline{u})$. Then, with

$$d_{\Delta}(\underline{u}, \underline{v}) := \max \{d_{\bar{\Delta}}(\underline{u}, \underline{v}), d_{\bar{\Delta}}(\underline{v}, \underline{u})\}$$

the mapping $d_{\Delta} : \Xi_{\mu, N} \times \Xi_{\mu, N} \rightarrow \mathbb{R}$ establishes a Banach–Mazur computable pseudometric on $\Xi_{\mu, N}$ that satisfies

$$0 \leq d_{\Delta}(\underline{u}, \underline{v}) \leq d(\underline{u}, \underline{v}).$$

Specifically, we have $d_{\Delta}(\underline{u}, \underline{v}) = 0$ if and only if the gate families \underline{u} and \underline{v} satisfy (4).

Consider $N \in \mathbb{N}$ once again. The pseudometric d_{Δ} naturally induces the equivalence relation

$$\underline{u} \bar{\Delta} \underline{v} : \Leftrightarrow d_{\Delta}(\underline{u}, \underline{v}) = 0, \quad \underline{u}, \underline{v} \in \Xi_{\mu, N}.$$

The resulting equivalence classes are in one-to-one correspondence with the gate sets (in the above sense) of cardinality $N + 1$ or less, where we tacitly require that every gate set contains the identity matrix. Then, d_{Δ} defines a Banach–Mazur computable metric on the set of all such gate sets. Note that given a different operator space or measure of gate similarity, an analogous construction is feasible. Essentially, d_{Δ} assesses how well we can approximate the members of \underline{u} through the members of \underline{v} and vice versa. From a practical perspective, the two gate families \underline{u} and \underline{v} approximately characterize the same set of atomic physical-layer quantum algorithms whenever $d_{\Delta}(\underline{u}, \underline{v})$ is (sufficiently) small. As indicated before, Section VII-B will briefly continue discussing different measures of gate similarity.

V. COMPUTABILITY ASPECTS OF EXACT CIRCUIT SYNTHESIS

The present section establishes Proposition 1, Theorem 1, and Corollary 1, which constitute our main results on exact circuit synthesis. Our findings are especially relevant to several established results concerning the gate family $\underline{u}_{\text{HT}} \in U_{\mu, 2}(2)$ (cf., Section II), which we will comment on further below.

In the following, consider an arbitrary gate family $\underline{u} \in \Xi_{\mu, L}^{\diamond}$, $L \in \mathbb{N}$, and define

$$\Omega(U|\underline{u}) := \min \{l \in \mathbb{N} : U \in \{\underline{u}(m) : m \in \langle l \rangle\}\}$$

for $U \in G(\underline{u})$. We call $\Omega(U|\underline{u})$ the word length of U with respect to \underline{u} . In other words, $\Omega(U|\underline{u})$ equals the least possible depth a quantum circuit built from members of the family \underline{u} must have to implement U . For notational convenience, we additionally set

$$\Omega(U|\underline{u}) := \infty \quad \text{whenever } U \in \neg G(\underline{u}) := \Xi_{\mu} \setminus G(\underline{u}).$$

Given $U \in \Xi_{\mu}$, estimating or possibly even determining $\Omega(U|\underline{u})$ is a relevant problem in both the broader context of exact circuit synthesis and the specific case of $\underline{u}_{\text{HT}}$. We obtain the following.

Proposition 1: For $\underline{u} \in \Xi_{\mu, L}^{\diamond}$, $L \in \mathbb{N}$, consider any computable sequence $(U_n)_{n \in \mathbb{N}} \subset \Xi_{\mu}$. There exists a (total) μ -recursive function $g : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ that satisfies the following for all $n \in \mathbb{N}$.

- 1) We have $g(n, 0) = 0$. For all $m \in \mathbb{N}$, we moreover have $g(n, m) \leq g(n, m + 1) \leq g(n, m) + 1$.
- 2) If, for $m \in \mathbb{N}$, we have $g(n, m) < \Omega(U_n|\underline{u})$, there exists $l \in \mathbb{N}$, $l > m$, such that we have $g(n, l) = g(n, m) + 1$.
- 3) If, for $m \in \mathbb{N}$, we have $g(n, m) = \Omega(U_n|\underline{u})$, we have $g(n, l) = g(n, m)$ for all $l \in \mathbb{N}$, $l \geq m$.

Consider the sequence $(\Omega(U_n|\underline{u}))_{n \in \mathbb{N}} \subset \mathbb{N}$. By Proposition 1, there exists a computable sequence $(l_{n, m})_{n, m \in \mathbb{N}} \subset \mathbb{N}$ (cf., Section III for the definition of computable sequence of natural numbers) that satisfies the following:

- 1) for all $m \in \mathbb{N}$, the sequence $(l_{n, m})_{n \in \mathbb{N}} \subset \mathbb{N}$ bounds the sequence $(\Omega(U_n|\underline{u}))_{n \in \mathbb{N}}$ from below;

- 2) for $m \rightarrow \infty$, the sequence $(l_{n,m})_{n,m \in \mathbb{N}}$ converges toward the sequence $(\Omega(U_n|\underline{u}))_{n \in \mathbb{N}}$.

If the convergence of $(l_{n,m})_{n,m \in \mathbb{N}}$ toward $(\Omega(U_n|\underline{u}))_{n \in \mathbb{N}}$ is “effective”—i.e., if there exists a matching sequence of converging *upper* bounds—the function

$$\Omega(U|\underline{u}) : \mathbb{N} \rightarrow \mathbb{N}, n \mapsto \Omega(U_n|\underline{u})$$

must be μ -recursive. As we will see in the following, this is not always the case. In fact, for any $L \in \mathbb{N}$ and any gate family $\underline{u} \in \Xi_{\mu,L}^\diamond$, there exists a computable sequence $(U_n)_{n \in \mathbb{N}} \subset G(\underline{u})$ such that *no* μ -recursive upper bound for $(\Omega(U_n|\underline{u}))_{n \in \mathbb{N}}$ can exist.

Theorem 1: For $N \in \mathbb{N}$, let $\underline{u} \in \Xi_{\mu,N}^\diamond$ be any gate family. Then, there exists a computable sequence $(U_n)_{n \in \mathbb{N}} \subset G(\underline{u})$ that satisfies the following.

- 1) We have $U_* = \text{Id}$ for the sequence’s unique accumulation point $U_* \in G(\underline{u})$. With $\mathcal{B} := \{n \in \mathbb{N} : U_n \neq \text{Id}\}$, $-\mathcal{B} := \mathbb{N} \setminus \mathcal{B}$, we have $\mathcal{B} \xrightarrow{\mu} -\mathcal{B}$ and $-\mathcal{B} \xrightarrow{\mu} \mathcal{B}$.
- 2) Let $\bar{\Omega} : \mathbb{N} \rightarrow \mathbb{Q}$ be any function such that for all $n \in \mathbb{N}$, we have $\bar{\Omega}(n) \geq \Omega(U_n|\underline{u})$. Then, $\bar{\Omega}$ is not a (Banach–Mazur) computable function.
- 3) For all $q \in \mathbb{Q}$, $q \geq \Omega(\text{Id}|\underline{u}) = 1$, and with $\mathcal{B}_q := \{n \in \mathbb{N} : \Omega(U_n|\underline{u}) > q\}$, $-\mathcal{B}_q := \mathbb{N} \setminus \mathcal{B}_q$, we have $\mathcal{B}_q \xrightarrow{\mu} -\mathcal{B}_q$ and $-\mathcal{B}_q \xrightarrow{\mu} \mathcal{B}_q$.

Descriptively speaking, Theorem 1 implies—for each universal gate set—the existence of a computable sequence $(U_n)_{n \in \mathbb{N}}$ of unitary matrices that satisfies the following.

- 1) Each of the sequence’s members is the matrix representation of a physical-layer quantum algorithm, i.e., exhibits exact quantum-circuit decompositions.
- 2) With respect to the sequence index, the length of the shortest such decomposition—i.e., the relevant algorithm’s word length—grows faster than any computable function.

We add some comments on the mathematical intuition behind Theorem 1. The elements of $\{U_{n,*}\}_{n \in \mathbb{N}}$ cluster around the “trivial” physical-layer quantum algorithm Id . Despite the close proximity of its elements to Id , however, the word length varies arbitrarily within the set $\{U_{n,*}\}_{n \in \mathbb{N}}$. In a broader context, Theorem 1 is thus an extension of observations previously made in [37]: For every universal set of (computable) quantum gates, there exists a computable sequence of physical-layer quantum algorithms that reflects the word length’s sensitivity to small perturbations.

Observe that by Theorem 1, the equality relation within $G(\underline{u})$ is undecidable. Let $(U_n)_{n \in \mathbb{N}} \subset G(\underline{u})$ be a computable sequence that satisfies Theorem 1. Choose any $V \in G(\underline{u})$ and define the computable sequence $(V_n)_{n \in \mathbb{N}} : V_n = VU_n$. Then, with $\mathcal{B} := \{n \in \mathbb{N} : U_n \neq \text{Id}\}$, $-\mathcal{B} := \mathbb{N} \setminus \mathcal{B}$, we obtain $\mathcal{B} = \{n \in \mathbb{N} : V_n \neq V\}$ and, by Theorem 1, $-\mathcal{B} \xrightarrow{\mu} \mathcal{B}$. Note that this concerns decidability within the domain of matrix representations. Deciding the equality of symbolic quantum

circuits is a more intricate issue that concerns the theory of word problems [43, Ch. 12, p. 418ff] rather than computable analysis, and will likely depend on the relevant gate family’s individual properties.

While the problem of estimating the word length of physical-layer quantum algorithms is relevant on its own, Theorem 1 also provides a direct no-go result for the explicit task of exact circuit synthesis. For $\underline{u} \in \Xi_{\mu,N}$, $N \in \mathbb{N}$, consider $u := \{u_0, \dots, u_N\}$ together with its canonic mapping $\omega \mapsto \underline{u}(\omega)$, $\omega \in u^*$. Moreover, let $(\omega_m)_{m \in \mathbb{N}} \subset u^*$ be any sequence. We call $(\omega_m)_{m \in \mathbb{N}}$ computable if there exists a μ -recursive function $g : \mathbb{N} \rightarrow \mathbb{N}$ such that for all $m \in \mathbb{N}$, $l \in \mathbb{N} \setminus \{0\}$, and $\mathbf{n} = (n_1, \dots, n_l) \in [N]^l$, we have

$$g(m) = \sum_{k=1}^l (n_k + 1)(N + 1)^{k-1} \dots$$

$$\dots \text{ whenever } \omega_m = (u(n_1), \dots, u(n_l)). \quad (5)$$

Essentially, (5) identifies each word in u^* with its position according to the lexicographic ordering, the right-end letter being most significant. For $\omega = (u(n_1), \dots, u(n_l)) \in u^*$, we write $l = |\omega|$. Note that if $(\omega_m)_{m \in \mathbb{N}} \subset u^*$ is computable, there exists a (total) μ -recursive function $h : \mathbb{N} \rightarrow \mathbb{N}$ that satisfies $|\omega_m| = h(m)$ for all $m \in \mathbb{N}$. We have the following.

Corollary 1: For $N \in \mathbb{N}$, let $\underline{u} \in \Xi_{\mu,N}^\diamond$ be any gate family and consider any finite alphabet u such that the mapping $\omega \mapsto \underline{u}(\omega)$ with domain u^* is canonic (in the sense of Section IV). We have the following.

- 1) Let $(\omega_n)_{n \in \mathbb{N}} \subset u^*$ be any computable sequence. Then, the sequence $(U_n)_{n \in \mathbb{N}} : U_n = \underline{u}(\omega_n)$ is computable as well.
- 2) There exists a computable sequence $(U_n)_{n \in \mathbb{N}} \subset G(\underline{u})$ such that no sequence $(\omega_n)_{n \in \mathbb{N}} \subset u^*$ that satisfies $\underline{u}(\omega_n) = U_n$ for all $n \in \mathbb{N}$ is computable.

Corollary 1 is directly applicable to several methods of exact circuit synthesis that concern the special case of $\underline{u}_{\text{HT}}$ and other related gate families [15], [16], [17], [18], [19], [44]. Kliuchnikov et al. [16] established a pivotal insight regarding the monoid $G(\underline{u}_{\text{HT}})$: We have

$$G(\underline{u}_{\text{HT}}) = \left\{ \begin{bmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \end{bmatrix} \in U_\mu(2) : \dots \dots \{z_{11}, z_{12}, z_{21}, z_{22}\} \subset \mathbb{Z}\left[\frac{1}{\sqrt{2}}, j\right] \right\} \quad (6)$$

where $\mathbb{Z}\left[\frac{1}{\sqrt{2}}, j\right]$ is the smallest subset of \mathbb{C} that

- a) contains the numbers $\frac{1}{\sqrt{2}}$ and j ;
- b) is closed with respect to addition and multiplication.

Observe that $\mathbb{Z}\left[\frac{1}{\sqrt{2}}, j\right]$ is a subset of \mathbb{C}_μ . Informally speaking, (6) provides an alternative characterization of $G(\underline{u}_{\text{HT}})$ that is not immediately related to the notion of quantum circuits. Later, [17] and [18] extended (6) to the multiqubit case and specific Clifford-cyclotomic gate families, respectively.

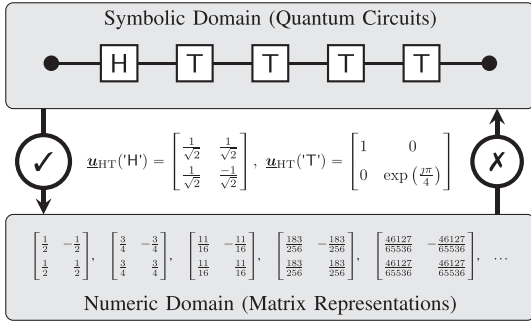


FIGURE 3. Comparison of symbolic quantum circuits and matrix representations using the gate family \underline{u}_{HT} (observe that since the imaginary part of \underline{u}_{HT} (“HTTTT”) vanishes, we omit its display in the figure). Given any symbolic quantum circuit, it is always possible to compute the corresponding matrix representation, i.e., arbitrarily precise rational approximations of the represented unitary transformation. In contrast, even if an exact circuit decomposition of some unitary transformation does exist, it is not possible to compute any such decomposition using only the transformation’s matrix representation.

Notably, the theory behind (6) gives rise to special synthesis functions¹ of the form

$$\text{syn}_{HT} : G(\underline{u}_{HT}) \rightarrow \{\text{‘Id’}, \text{‘H’}, \text{‘T’}\}^*, U \mapsto \text{syn}_{HT}(U)$$

that, for all $U \in G(\underline{u}_{HT})$, satisfy $U = \underline{u}_{HT}(\text{syn}_{HT}(U))$ plus several other desirable analytic properties [15], [16], [17], [18], [44]. To the best of the authors’ knowledge, algorithmically evaluating any of these functions given $U \in G(\underline{u}_{HT})$ —that is, computing $\text{syn}_{HT}(U)$ —presupposes real-RAM computers. As follows from Corollary 1, the methods established in [15], [16], [17], [18], and [44] cannot have an equivalent within the domain of Turing machines. Fig. 3 provides a visual account of Corollary 1 in this context. The rational-valued approximation depicted in the “numeric domain” employs the fixed-point algorithm illustrated in Fig. 2.

VI. COMPUTABILITY ASPECTS OF UNIVERSALITY

Our analysis of approximate circuit synthesis comprises the present Section VI and the subsequent Section VII. Concerning the analysis’s structure, Proposition 2, Theorem 2, and Corollary 2 below are primarily supplementary to our main contributions. Despite their supplementary character, this section’s results provide relevant insights on the topic of universality checking [33], [34], [41]. Notably, universality checking is strongly related to the theory of approximate t-designs [40], [41]. On a more subtle level, approximate t-designs and recent findings on the quantum circuit overhead [42] are in turn related to quantum big-O analysis and its computability aspects. In Section VII-A, we will provide a detailed account of these relationships. Among others, they involve the (Banach–Mazur) computability of η_{Ξ} .

¹ Note that the authors in [15], [16], [17], [18], and [44] referred to these functions as algorithms. Moreover, the authors in [16], [18], and [44] included pseudocode descriptions. To maintain a clear distinction to the terminology established in Section III, we will not employ the term “algorithm” in the context of [15], [16], [17], [18], and [44].

Proposition 2: Consider $M \in \mathbb{N}$ arbitrary. The function $\eta_{\Xi} : \Xi_{\mu,M} \times \mathbb{N} \rightarrow \mathbb{R}_{\mu}$ is Banach–Mazur computable.

Given a computable sequence of gate families, Proposition 2 asserts that assessing the individual gate families’ compiling performance algorithmically is always possible. Note, however, that Proposition 2 concerns arbitrary but finite maximum circuit lengths. In contrast, the compiling performance in relation to an increasing circuit length falls into the domain of big-O analysis. By Corollary 2 below, Proposition 2 does not allow for any direct conclusions in this regard.

Consider $M \in \mathbb{N}$. Universality checking refers to the problem of (algorithmically) verifying whether we have

$$\underline{u} \in \Xi_{\mu,M}^{\diamond} \quad \text{or} \quad \underline{u} \in \Xi_{\mu,M} \setminus \Xi_{\mu,M}^{\diamond} =: \neg \Xi_{\mu,M}^{\diamond}.$$

given $\underline{u} \in \Xi_{\mu,M}$. From the perspective of computable analysis, we have the following.

Theorem 2: Consider $M \in \mathbb{N}$ arbitrary.

- 1) If $\Xi_{\mu,M}^{\diamond}$ is nonempty, we have $\neg \Xi_{\mu,M}^{\diamond} \xrightarrow{\mu} \Xi_{\mu,M}^{\diamond}$.
- 2) If $\Xi_{\mu,M}^{\diamond}$ is nonempty, we have $\Xi_{\mu,M}^{\diamond} \xrightarrow{\mu} \neg \Xi_{\mu,M}^{\diamond}$.

The theory of approximate t-designs provides necessary and sufficient conditions for the universality of gate families. Among others, the authors in [33] and [34] established such conditions and phrased them in terms of decision algorithms. To the best of the authors’ knowledge, these algorithms require the ability to store and manipulate exact real numbers and, accordingly, fall into the domain of BSS theory. By Theorem 2, they cannot carry over to Turing’s framework. In addition, Theorem 2 implies the following.

Corollary 2: For $M \in \mathbb{N}$ such that $\Xi_{\mu,M}^{\diamond}$ is nonempty, the function $\eta_{\Xi}^{\infty} : \Xi_{\mu,M} \rightarrow \mathbb{R}$ is not Banach–Mazur computable.

Quantum big-O analysis fundamentally relies on asymptotic mathematics. The contrast between Propositions 2 and Corollary 2 is a first hint toward the relevant bottleneck: the feasibility of algorithmically assessing a gate families compiling performance in the finite-circuit case does not provide conclusions regarding the asymptotic regime.

VII. COMPUTABILITY ASPECTS OF QUANTUM BIG-O THEORY

The present section establishes Proposition 3 and Theorem 3, which constitute our main results on approximative circuit synthesis. Consider $N \in \mathbb{N}$, a computable growth function $f : \mathbb{N} \rightarrow \mathbb{Q}$, and an arbitrary computable gate family $\underline{u} \in \mathcal{O}_{\Xi,N}(f)$. Let u be any finite alphabet such that the mapping $\omega \mapsto \underline{u}(\omega)$ with domain u^* is canonic (cf., Section IV). Then, for $M \in \mathbb{N}$ arbitrary and any computable sequence $(U_n)_{n \in \mathbb{N}} \subset \Xi_{\mu}$, there exists a computable double sequence $(\omega_{n,m})_{n,m \in \mathbb{N}} \subset u^*$ (cf., Section V) such that we have

$$\|U_n - \underline{u}(\omega_{n,m})\| < 2^{-(m+M)+1} \quad (7)$$

for all $n, m \in \mathbb{N}$. This is because of Lemma 6 and the Banach–Mazur computability of $\|\cdot\|$: Informally speaking, we can search through the set u^* until we find a word whose corresponding element in $G(\underline{u})$ is sufficiently close to the

unitary operator we wish to approximate. Due to [12, Corollary 9], it is not possible to make general statements on other analytic properties of $(\underline{u}(\omega_{n,m}))_{n,m \in \mathbb{N}}$ or the complexity of computing $\omega_{n,m}$ for $n, m \in \mathbb{N}$, which may be arbitrarily high. Especially, it is impossible to guarantee that $(\omega_{n,m})_{n,m \in \mathbb{N}}$ is well behaved in terms of its component's word lengths. For some $n, m, l \in \mathbb{N}$, even if we have $U_n = U_l$, the difference between $|\omega_{n,m}|$ and $|\omega_{l,m}|$ may be arbitrarily large.

In addition to the above, consider $q \in \mathbb{Q}^+$ such that we have $q > \Phi_{f,M}(\underline{u})$, $M \in \mathbb{N}$. In other words, assume q is an admissible big-O coefficient for (\underline{u}, f) , given the offset M . While it is not generally possible to ensure optimal word-lengths, f and q yield some mitigation regarding the behavior of $(\omega_{n,m})_{n,m \in \mathbb{N}}$. Particularly, there exists $(\omega_{n,m})_{n,m \in \mathbb{N}}$ such that besides (7), we also have $|\omega_{n,m}| < qf(m+M) + 1$ for all $n, m \in \mathbb{N}$. Essentially, the triple (M, f, q) provides a stopping criterion for the search through u^* .

Observe that [5] found explicit admissible big-O coefficients for several specific gate families in the linear big-O class. However, the sheer variety of practically relevant gate families, the possible differences in hardware per manufacturer, and the variations even within a specific hardware platform due to adjustable or environmental parameters motivates the search for methods of determining admissible big-O coefficients for arbitrary gate families within a given big-O class. Proposition 3 and Theorem 3 concern the question of whether any such method lies within the capabilities of digital computing. We have the following.

Proposition 3: For $N \in \mathbb{N}$, consider a computable growth function $f : \mathbb{N} \rightarrow \mathbb{Q}$ and any computable sequence $(\underline{u}_n)_{n \in \mathbb{N}} \subset \mathcal{O}_{\Xi,N}(f)$. We have the following.

- 1) For all $M \in \mathbb{N}$, there exists a computable double sequence $(q_{n,m})_{n,m \in \mathbb{N}} \subset \mathbb{Q}$ such that for all $n, m \in \mathbb{N}$, we have $q_{n,m} \leq q_{n,m+1} \leq \sup_{l \in \mathbb{N}} q_{n,l} = \Phi_{f,M}(\underline{u}_n)$.

Given $(\underline{u}_n)_{n \in \mathbb{N}} \subset \mathcal{O}_{\Xi,N}(f)$ as above, assume there exists a computable double sequence $(p_{n,m})_{n,m \in \mathbb{N}} \subset \mathbb{Q}$ such that for all $n, m \in \mathbb{N}$, we have $p_{n,m} \geq p_{n,m+1} \geq \inf_{l \in \mathbb{N}} p_{n,l} = \Phi_{f,M}(\underline{u}_n)$. Due to Proposition 3, $(\Phi_{f,M}(\underline{u}_n))_{n \in \mathbb{N}}$ must then be a computable sequence of computable real numbers. Thus, if such a sequence $(p_{n,m})_{n,m \in \mathbb{N}}$ were to exist for all gate families $(\underline{u}_n)_{n \in \mathbb{N}} \subset \mathcal{O}_{\Xi,N}(f)$, Proposition 3 would imply the Banach-Mazur computability of $\Phi_{f,M} : \mathcal{O}_{\Xi,N}(f) \rightarrow \mathbb{R}^+$. Observe the following.

Theorem 3: For $N, M, L \in \mathbb{N}$, let $f : \mathbb{N} \rightarrow \mathbb{Q}$ be any computable growth function

- a) if $\Xi \equiv U$, such that $\mathcal{O}_{U,M}(f)$ is nonempty;
- b) if $\Xi \equiv SU$, such that $\mathcal{O}_{SU,M}(f)$ is nonempty and, for some $J \in \mathbb{N}$, there exists $\underline{u} \in \text{SU}_{\mu,J}^\circ \setminus \mathcal{O}_{SU,J}(f)$.

For all $\underline{v}_* \in \mathcal{O}_{\Xi,N}(f)$, there exists a computable sequence $(\underline{u}_{n,*})_{n \in \mathbb{N}} \subset \mathcal{O}_{\Xi,N+M}(f)$ that satisfies the following.

- 1) We have $d_\Delta(\underline{v}_*, \underline{u}_*) = 0$ for the sequence's unique accumulation point \underline{u}_* . With $\mathcal{B} := \{n \in \mathbb{N} : \underline{u}_{n,*} \neq \underline{u}_*\}$, $\neg \mathcal{B} := \mathbb{N} \setminus \mathcal{B}$, we have $\mathcal{B} \xrightarrow{\mu} \neg \mathcal{B}$ and $\neg \mathcal{B} \xrightarrow{\mu} \mathcal{B}$.

- 2) Let $\overline{\Phi} : \mathbb{N} \rightarrow \mathbb{Q}$ be any function such that for all $n \in \mathbb{N}$, we have $\overline{\Phi}(n) \geq \Phi_{f,L}(\underline{u}_{n,*})$. Then, $\overline{\Phi}$ is not a (Banach–Mazur) computable function.
- 3) For all $q \in \mathbb{Q}$, $q \geq \Phi_{f,L}(\underline{v}_*)$, and with $\mathcal{B}_{q,L} := \{n \in \mathbb{N} : \Phi_{f,L}(\underline{u}_{n,*}) > q\}$, $\neg \mathcal{B}_{q,L} := \mathbb{N} \setminus \mathcal{B}_{q,L}$, we have $\mathcal{B}_{q,L} \xrightarrow{\mu} \neg \mathcal{B}_{q,L}$ and $\neg \mathcal{B}_{q,L} \xrightarrow{\mu} \mathcal{B}_{q,L}$.

From Section III, recall that any Banach–Mazur computable function $\overline{\Phi} : \mathbb{N} \rightarrow \mathbb{Q}$ equivalently defines a computable sequence $(\overline{\Phi}(n))_{n \in \mathbb{N}} \subset \mathbb{Q}$. Thus, given $\overline{\Phi} : \mathbb{N} \rightarrow \mathbb{Q}$ as in Statement 2 of Theorem 3, $(\overline{\Phi}(n))_{n \in \mathbb{N}}$ cannot be computable. Consequently, provided the relevant conditions are met, the function $\Phi_{f,L} : \mathcal{O}_{\Xi,N+M}(f) \rightarrow \mathbb{R}$ itself cannot be Banach–Mazur computable either.

The mathematical intuition behind Theorem 3 is similar to the one behind Theorem 1. Assume the relevant condition of Theorem 3 satisfied. Then, for each gate family $\underline{v}_* \in \mathcal{O}_{\Xi,N}(f)$, we can find a computable sequence $(\underline{u}_{n,*})_{n \in \mathbb{N}} \subset \mathcal{O}_{\Xi,N+M}(f)$ of gate families such that with respect to d_Δ , the elements of $\{\underline{u}_{n,*}\}_{n \in \mathbb{N}}$ cluster around \underline{v}_* . In other words, both \underline{v}_* and $\underline{u}_{0,*}, \underline{u}_{1,*}, \underline{u}_{2,*}, \dots$ all characterize approximately the same set of atomic physical-layer quantum algorithms. Despite the close proximity of its elements to \underline{v}_* , however, the admissible big-O coefficients vary arbitrarily within the set $\{\underline{u}_{n,*}\}_{n \in \mathbb{N}}$. Thus, the big-O characteristics of \underline{v}_* are sensitive to small perturbations. Heuristic approaches to computing admissible big-O coefficients—including those based on digital implementations of BSS algorithms—must inevitably reflect this sensitivity through numerical instabilities and incorrect results.

From the perspective of computable analysis, Condition a) of Theorem 3 is more or less a mere formality. If the function f grows slowly enough for $\mathcal{O}_{U,M}(f)$ to be empty for all $M \in \mathbb{N}$, the theorem's assertions are trivially true. Otherwise, for $L, K \in \mathbb{N}$ with K sufficiently large and any possible algorithm (cf., Section III)

$$\text{Alg}_\phi : D(\text{Alg}_\phi) \rightarrow \mathbb{R}_\mu \quad \text{with} \quad \mathcal{O}_{U,K}(f) \subseteq D(\text{Alg}_\phi)$$

there must exist an infinite number of gate families $\underline{u} \in \mathcal{O}_{U,K}(f)$ such that we have $\text{Alg}_\phi(\underline{u}) < \phi_{f,L}(\underline{u})$. Unless the relevant growth function trivializes the problem, there definitely cannot exist a general algorithm that reliably determines admissible big-O coefficients for arbitrary gate families $\underline{u} \in \mathcal{O}_{U,K}(f)$. Much in contrast, Condition b) of Theorem 3 is nontrivial and might turn out impossible to satisfy. In this sense, Theorem 3 establishes the uncomputability of $\phi_{f,L}$ unconditionally, but the uncomputability of $\phi_{f,L}$ only conditionally. The distinction of both cases will be relevant in Sections VII-A and VII-B below. Specifically, Section VI-I-A will discuss the satisfiability of Condition b) in the context of [5].

A. DISCUSSION–CONJECTURE OF GENERAL EFFICIENCY AND THE SPECTRAL GAP

Consider $N \in \mathbb{N}$ such that $\text{SU}_{\mu,N}^\circ$ is nonempty and recall $\chi : \mathbb{N} \rightarrow \mathbb{Q}$, $n \mapsto \chi(n) = n$, from Section II. We call $\underline{u} \in$

$\mathcal{O}_{\text{SU},N}(\chi)$ computationally efficient [5].² As indicated in Section I, the question of whether there exists an inefficient (but still universal) computable gate family—formally, whether we have $\mathcal{O}_{\text{SU},N}(\chi) \neq \text{SU}_{\mu,N}^\diamond$ —is nontrivial. By $\mathcal{L}^2_{\text{SU}}$, we denote the Hilbert space of square-integrable functions $f : \text{SU} \rightarrow \mathbb{C}$, where integration is performed with respect to the normalized Haar measure. Furthermore, with some abuse of notation, we denote both the vector and operator norm of $\mathcal{L}^2_{\text{SU}}$ by $\|\cdot\|_{\mathcal{L}}$. With

$$\tilde{\chi} : \mathbb{N} \rightarrow \mathbb{Q}, M \mapsto \tilde{\chi}(M) := M + \kappa$$

(note that we have $\mathcal{O}_{\text{SU},N}(\tilde{\chi}) = \mathcal{O}_{\text{SU},N}(\chi)$, accordingly), Harrow et al. [5] provided a relationship of the form

$$\varphi_{\tilde{\chi},0}(\underline{u}) \leq \frac{\varsigma}{-\log_2 \|\Gamma(\underline{u})\|_{\mathcal{L}}}, \quad \underline{u} \in \mathcal{O}_{\text{SU},N}(\tilde{\chi}) \quad (8)$$

where $\varsigma, \kappa \in \mathbb{Q}$ are constants that do not depend on \underline{u} . Further, $\Gamma(\underline{u}) : \mathcal{L}^2_{\text{SU}} \rightarrow \mathcal{L}^2_{\text{SU}}$ is a special linear operator defined through \underline{u} . For $f \in \mathcal{L}^2_{\text{SU}}, \mathbf{V} \in \text{SU}$, we have

$$\begin{aligned} \Gamma(\underline{u})f(\mathbf{V}) &:= \dots \\ &\dots \left(\frac{1}{N+1} \sum_{n=0}^N f(\underline{u}^{-1}(n)\mathbf{V}) \right) - \int_{\text{SU}} f(\mathbf{U}) d\mathbf{U}. \end{aligned}$$

The informal interpretation of $\Gamma(\underline{u})$ is as follows. The more uniform the components of the tuple $(\underline{u}(n))_{n \in \{0, \dots, N\}}$, distribute within the set SU for $m \in \mathbb{N}$, the closer is $\Gamma^m(\underline{u})$ to the zero operator. Note that in principle, $\Gamma(\underline{u})$ is well defined not only for $\underline{u} \in \mathcal{O}_{\text{SU},N}(\chi)$, but for all $\underline{u} \in \text{SU}_{\mu,N}$. In this case, we call $\gamma(\underline{u}) := 1 - \|\Gamma(\underline{u})\|_{\mathcal{L}} \geq 0$ the spectral gap and obtain the implication

$$\gamma(\underline{u}) > 0 \quad \Rightarrow \quad \underline{u} \in \mathcal{O}_{\text{SU},N}(\chi). \quad (9)$$

Harrow et al. [5] conjectured that (9) is an equivalence (see further below). However, to the best of the authors' knowledge, it remains an open question whether this is truly the case.

Consider $\underline{v} \in \mathcal{O}_{\text{SU},N}(\chi)$. By Proposition 3, there exists a computable sequence $(q_n)_{n \in \mathbb{N}} \subset \mathbb{Q}$ such that we have $\sup_{n \in \mathbb{N}} q_n = \varphi_{\chi,0}(\underline{v})$. Analogously, if we can find a sequence $(f_n)_{n \in \mathbb{N}} \subset \mathcal{L}^2_{\text{SU}}$ of unit vectors such that $\{f_n\}_{n \in \mathbb{N}}$ is dense in the unit sphere of $\mathcal{L}^2_{\text{SU}}$ and, for $\underline{u} \in \text{SU}_{\mu,N}$, there exists a computable sequence $(p_{n,m})_{n,m \in \mathbb{N}} \subset \mathbb{Q}$ satisfying

$$\sup_{m \in \mathbb{N}} p_{n,m} = \|\Gamma(\underline{u})f_n\|_{\mathcal{L}}$$

for all $n \in \mathbb{N}$, we can successively approximate $\|\Gamma(\underline{u})\|_{\mathcal{L}}$ from below; note that if it is possible to equip $\mathcal{L}^2_{\text{SU}}$ with a suitable computability structure, it is also possible to find such sequences $(f_n)_{n \in \mathbb{N}}$ and $(p_{n,m})_{n,m \in \mathbb{N}}$. However, in order to compute $\varphi_{\chi,0}(\underline{v})$ or $\|\Gamma(\underline{u})\|_{\mathcal{L}}$ in finite time up to a prescribed precision, we must also be able to successively

² Note that some publications additionally require $\varphi_{\chi,0}(\underline{u})$ to attain a certain optimal value for \underline{u} to be called efficient, see, e.g., [42]. In the scope of our analysis, this is not the case.

approximate either quantity from above (see Part 7 in Summary 3). Observe the conceptual relationship to information theory. Models of data transmission commonly entail a set of achievable rates, whose supremum equals the associated channel capacity. If it is possible to enumerate the set of achievable rates, we can successively approximate the channel capacity from below. However, in order to compute the channel capacity up to prescribed accuracy, we must also be able to successively approximate the supremum of achievable rates from above. Intuitively speaking, we thus require the “dual” set of “unachievable” rates to be enumerable as well. The literature of information theory refers to this duality as achievability and converse. For a detailed treatment in the context of computability theory, we refer to [45].

As follows from Theorem 3 and (8), it is impossible to compute the functions

$$\varphi_{\chi,0} : \mathcal{O}_{\text{SU},N}(\chi) \rightarrow \mathbb{R}, \quad \|\Gamma(\cdot)\|_{\mathcal{L}} : \text{SU}_{\mu,N} \rightarrow \mathbb{R}$$

(for $N \in \mathbb{N}$ sufficiently large) if there exists at least one inefficient gate family. Particularly, Theorem 3 and (8) conditionally prove the nonexistence of a computable converse characterization of $\varphi_{\chi,0}$ and $\|\Gamma(\cdot)\|_{\mathcal{L}}$ in the information-theoretic sense: if there exists at least one inefficient gate family, there also exists a computable sequence $(\underline{u}_n)_{n \in \mathbb{N}} \subset \mathcal{O}_{\text{SU},N}(\chi)$ such that all sequences $(q_{n,m})_{n,m \in \mathbb{N}} \subset \mathbb{Q}$ that satisfy, respectively, $\inf_{m \in \mathbb{N}} q_{n,m} = \varphi_{\chi,0}(\underline{u}_n)$ or $\inf_{m \in \mathbb{N}} q_{n,m} = \|\Gamma(\underline{u}_n)\|_{\mathcal{L}}$ for all $n \in \mathbb{N}$ are necessarily uncomputable.

Harrow et al. [5] conjectured all universal gate families to be efficient, which has since been referred to as the spectral gap conjecture. Specifically they observe that (9) became an equivalence if $\|\Gamma(\cdot)\|_{\mathcal{L}}$ turned out to be a continuous function. Then, we would also have we have $\text{SU}_{\mu,N}^\diamond = \mathcal{O}_{\text{SU},N}(\chi)$. This observation is in line with our results: if there exists at least one inefficient gate family, $\|\Gamma(\underline{u})\|_{\mathcal{L}}$ must be discontinuous and thus uncomputable. Notably, they furthermore observe that computing $\|\Gamma(\underline{u})\|_{\mathcal{L}}$ given $\underline{u} \in \mathcal{O}_{\text{SU},N}(\chi)$ seems to be difficult, though “likely [being] an important step in determining the prefactor $[\varphi_{\chi,0}(\underline{u})]$, which measures how effective $[\underline{u}]$ would be for compiling.” As our results demonstrate, providing an algorithm that determines a gate family's asymptotic compiling performance is the mathematically harder problem as compared to proving the spectral gap conjecture, at least concerning computable gate families. Researchers have made several advances regarding the conjecture's resolution, which we briefly summarize as follows.

- Universal gate families are efficient if they satisfy some mild technical assumptions and their members' matrix entries are algebraic numbers [31], [32]. Remarkably, the result holds for general compact simple Lie groups [46].
- Quantum big-O analysis is strongly related to the theory of exact and approximate t-designs (see below) and uniformly Haar-random unitary matrices [35], [38], [39], [40], [42], [47]. Specifically, Dujian and Sawicki [39]

conjectured the Haar-uniform probability of a random gate family being efficient to equal one.

- Moreover, the authors in [35], [38], [39], and [42] supported their results through extensive numerical evaluations.

In one way or another, most of the listed advances are related to the theory of approximate t-designs. Notably, the relevant insights exhibit a direct interpretation in the context of computable analysis. For $U \in \text{SU}_{\mu,N}$, consider

$$U^{t,t} := U^{\otimes t} \otimes \overline{U}^{\otimes t}, \quad t \in \mathbb{N}$$

where $\overline{U} \in \text{SU}_{\mu,N}$ denotes the entry wise complex conjugate of U . Given $\underline{u} \in \text{SU}_{\mu,N}$, we define the residual moment operator

$$\mathbf{T}_t(\underline{u}) := \int_{\text{SU}} U^{t,t} dU - \frac{1}{N+1} \sum_{n=0}^N (\underline{u}(n))^{t,t}.$$

As before, the integral is with respect to the Haar measure. Moreover, we employ the term “residual” to indicate that $\mathbf{T}_t(\underline{u})$ is a difference of moment operators in the sense of [38], [39], [40], [42]. By the Peter–Weyl theorem, we have

$$\|\Gamma(\underline{u})\|_{\mathcal{L}} = \sup_{t \in \mathbb{N}} \|\mathbf{T}_t(\underline{u})\| \quad (10)$$

(cf., [39], [40], and [42] for details). Due to the computability of Haar-integration [48] and the standard matrix norm, we can find a computable sequence $(q_{t,m})_{t,m \in \mathbb{N}} \subset \mathbb{Q}$ such that for all $t \in \mathbb{N}$, we have $\sup_{m \in \mathbb{N}} q_{t,m} = \|\mathbf{T}_t(\underline{u})\|$. In information-theoretic terminology, (10) provides a way of implementing the achievability part in computing $\|\Gamma(\underline{u})\|_{\mathcal{L}}$ without having to equip $\mathcal{L}^2_{\text{SU}}$ with a computability structure.

The characterization of $\|\Gamma(\cdot)\|_{\mathcal{L}}$ in terms of residual moment operators also provides explicit asymptotic upper bounds for η_{SU} —there exist constants $\tau, \sigma \in \mathbb{Q}$ such that with $t : \mathbb{N} \rightarrow \mathbb{Q}, K \mapsto t(K) := \frac{\tau}{K}$, and for $\underline{u} \in \text{SU}_{\mu,N}^{\circ}, K \in \mathbb{N}$, arbitrary, we obtain

$$\eta_{\text{SU}}(\underline{u}, k) \leq 2^{-K} \quad \text{for all } k \geq \frac{-\sigma K}{\log \|\mathbf{T}_{t(K)}(\underline{u})\|}$$

(for details, we again refer to [39], [40], [42]). Accordingly, defining $F_{\underline{u}} : \mathbb{N} \rightarrow \mathbb{R}_{\mu}$ through

$$F_{\underline{u}}(K) := \frac{-K}{\log \|\mathbf{T}_{t(K)}(\underline{u})\|}$$

we obtain $\underline{u} \in \mathcal{O}_{\text{SU},N}(F_{\underline{u}})$ and observe that $F_{\underline{u}}$ is Banach–Mazur computable. Moreover, if we have $\underline{u} \in \mathcal{O}_{\text{SU},N}(\chi)$, we also have

$$\frac{-\sigma}{\log \|\mathbf{T}_{t(K)}(\underline{u})\|} \leq \varphi_{\chi,0}(\underline{u})$$

for all $K \in \mathbb{N}$, which is consistent with Proposition 3. From the perspective of mere computability, the existence of $F_{\underline{u}}$ is thus not surprising. Numerically, however, computing $F_{\underline{u}}$ is much easier than computing η_{SU} .

While significantly simplifying the achievability part in computing $\|\Gamma(\cdot)\|_{\mathcal{L}}$, the theory of approximate t-designs has

not provided a corresponding converse so far. As indicated above, the authors in [35], [38], [39], and [42] performed extensive numerical studies of quantities building upon and related to residual moment operators, such as the quantum circuit overhead [42]. Specifically, the authors in [38] and [42] observed the sequence $\|\mathbf{T}_0(\cdot)\|, \|\mathbf{T}_1(\cdot)\|, \|\mathbf{T}_2(\cdot)\|, \dots$ to stabilize swiftly. On the one hand, this serves as an indicator for convergence in practical scenarios. On the other hand, it is well-known in information theory that sequences converging toward asymptotic quantities may stabilize for an arbitrary long time before they perform arbitrarily large “jumps” [49]. Therefore, mathematically rigorous estimates of such quantities additionally require complementary converse characterizations. Concerning $\varphi_{\chi,0}$ and $\|\Gamma(\cdot)\|_{\mathcal{L}}$, the authors are skeptical as to whether the theory of approximate T-designs entails such characterizations. Perhaps, a different mathematical framework will be necessary.

B. DISCUSSION: THE BIG-O THEORY OF SYMMETRIC SUBGROUPS

In practical quantum-computing systems that include a large number of physical qubits, it may not always be intended or even possible to establish universality with respect to SU_{μ} —in other words, universality on the global level—rather than some subgroup $V_{\mu} \subset \text{SU}_{\mu}$ thereof. All real-world quantum gates must respect some form of locality. That is, physical quantum gates act nontrivially on specific subsets of all physical qubits only, and these subsets consist of qubits located in close spatial proximity to each other. Thus, successively “spreading” local transformations throughout the system is the only physically feasible method of implementing computational operations that affect the entire system. The fundamental symmetries intrinsic to almost all real-world physical systems further restrict the set of physically feasible quantum gates and lead to a remarkable phenomenon. Let V_{μ} be any subgroup of SU_{μ} and, for $N \in \mathbb{N}, \underline{v} \in \text{SU}_{\mu,N}$ a gate family. If we have $\text{efcG}(\underline{v}) = V_{\mu}$, we call \underline{v} universal relative to V_{μ} . Then, if V_{μ} is a subgroup that respects some symmetry relation (see below), there may not exist a gate family $\underline{v} \in V_{\mu}$ that is universal relative to V_{μ} and, at the same time, consists of local gates only [36], [50], [51].

The big-O analysis of computable gate families analogously applies to symmetric subgroups of SU_{μ} . Throughout the rest of Section VII-B, consider $N, M \in \mathbb{N}$ such that $\text{SU}_{\mu,N}^{\circ}$ is nonempty, otherwise arbitrary. Theorem 3 establishes a relation between the possible computability of $\varphi_{\chi,M}$ and the conjecture $\mathcal{O}_{\text{SU},N}(\chi) = \text{SU}_{\mu,N}^{\circ}$ [5], which we have extensively discussed in Section VII-A. Notably, if $\varphi_{\chi,M}$ turned out computable, symmetry relations might also determine which subgroups of SU_{μ} exhibit pathological computability behavior. The rest of Section VII-B will address this phenomenon. For $\underline{v} \in \Xi_{\mu,N}, j \in \mathbb{N}$, define

$$\eta_G(\underline{v}, j) := \sup_{V \in \text{efcG}(\underline{v})} \inf_{n \in \{N; j\}} \|V - \underline{v}(n)\|$$

and assume there exists a μ -recursive function $\nu : \mathbb{N} \rightarrow \mathbb{N}$ such that for all $j, J \in \mathbb{N}, j \geq \nu(J)$, we have $\eta_G(\underline{v}, j) \leq \frac{1}{2^J}$.

Upon slight adjustments, the line of reasoning underlying the proof of Proposition 2 (cf., Appendix B2) then provides the following for all $\underline{u} \in \Xi_{\mu,L}, L \in \mathbb{N}$.

- 1) If we have $\text{efcG}(\underline{w}) = \text{efcG}(\underline{v})$, there exists a μ -recursive function $\tilde{v} : \mathbb{N} \rightarrow \mathbb{N}$ such that $(\eta_G(\underline{w}, j))_{j \in \mathbb{N}}$ satisfies $\eta_G(\underline{w}, j) \leq \frac{1}{2^j}$ for all $j, J \in \mathbb{N}, j \geq \tilde{v}(J)$.

In this case, we call $\text{efcG}(\underline{v})$ effectively generated. Precisely, $\text{efcG}(\underline{v})$ is effectively generated if and only if $(\eta_G(\underline{v}, j))_{j \in \mathbb{N}}$ is a computable sequence of computable numbers.

Consider $L, K \in \mathbb{N}$ and $\underline{v} \in \text{SU}_{\mu,N}(K)$. If K is sufficiently large, there exists an isometric embedding of $U_\mu(L)$ in $\text{SU}_\mu(K)$. Specifically, we call $\mathfrak{g} : U_\mu(L) \rightarrow \text{SU}_\mu(K)$ an isometric embedding if, for all $U, V \in U_\mu(L)$, we have

$$\mathfrak{g}(UV) = \mathfrak{g}(U)\mathfrak{g}(V) \quad \|U - V\| = \|\mathfrak{g}(U) - \mathfrak{g}(V)\|.$$

For $\underline{u} \in U_{\mu,N}(L)$, define $\mathfrak{g}(\underline{u}) \in \text{SU}_{\mu,N}(K)$ through component wise application of \mathfrak{g} to \underline{u} . In general, we have

$$\text{efcG}(\mathfrak{g}(\underline{u})) = \{\mathfrak{g}(U) : U \in \text{efcG}(\underline{u})\}.$$

Thus, whenever we have $\underline{v} = \mathfrak{g}(\underline{u})$ for some $\underline{u} \in U_{\mu,N}(L)$, then $\text{efcG}(\underline{v})$ is effectively generated if and only if $\text{efcG}(\underline{u})$ is effectively generated. Lastly, we define

$$\tilde{\varphi}_{f,M}(\underline{v}) := \inf \left\{ q \in \mathbb{Q} : \eta_G(\underline{v}, j) \leq \frac{1}{2^{(J+M)}} \dots \right.$$

... for all $j, J \in \mathbb{N}$ that satisfy $j \geq qf(J+M)$

for computable growth functions $f : \mathbb{N} \rightarrow \mathbb{Q}$, provided the relevant subset of \mathbb{Q} is nonempty. Informally speaking, f must be an asymptotic upper bound of \underline{v} in $\text{efcG}(\underline{v})$ rather than $\text{SU}_\mu(K)$ (cf., Definition 11).

Consider $L, K \in \mathbb{N}$ and $\mathfrak{g} : U_\mu(L) \rightarrow \text{SU}_\mu(K)$ as above. The embedding \mathfrak{g} induces a subgroup of $\text{SU}_\mu(K)$ which we denote (with some abuse of notation) by

$$\mathfrak{g}U_\mu(L) := \{\mathfrak{g}(U) : U \in U_\mu(L)\} \subset \text{SU}_\mu(K).$$

Note that $\mathfrak{g}U_\mu(L)$ is always effectively generated. Next, let $\underline{u} \in U_{\mu,N}(L)$ be arbitrary. As follows from the above, we then have $\eta_G(\mathfrak{g}(\underline{u}), j) = \eta_U(\underline{u}, j)$ for all $j \in \mathbb{N}$. If we additionally have $\underline{u} \in U_{\mu,N}^\circ(L)$ and $f : \mathbb{N} \rightarrow \mathbb{Q}$ is any asymptotic upper bound of \underline{u} in $U_\mu(L)$, we must also have

$$\text{efcG}(\mathfrak{g}(\underline{u})) = \mathfrak{g}U_\mu(L) \quad \tilde{\varphi}_{f,M}(\mathfrak{g}(\underline{u})) = \phi_{f,M}(\underline{u}).$$

Accordingly, the big-O structure and characteristics of $U_\mu(L)$ transfer to the (sub)group $\mathfrak{g}U_\mu(L)$. The uncomputability of $\phi_{f,M}$ (in the sense of Theorem 3) equally applies to $\tilde{\varphi}_{f,M}$ for gate families $\underline{v} \in \{\mathfrak{g}(\underline{u}) : \underline{u} \in \mathcal{O}_{U,N}(f)\}$, regardless of whether $\phi_{\chi,M}$ turns out computable or uncomputable (again in the sense of Theorem 3).

Let $V_\mu \subset \text{SU}_\mu$ be any effectively generated subgroup of SU_μ and $f : \mathbb{N} \rightarrow \mathbb{Q}$ any computable growth function. First, define the set

$$V_{\mu,N}^\circ := \{\underline{v} \in \text{SU}_{\mu,N} : \text{efcG}(\underline{v}) = V_\mu\}.$$

In other words, for $\underline{v} \in \text{SU}_{\mu,N}$, we have $\underline{v} \in V_{\mu,N}^\circ$ if and only if \underline{v} is universal relative to V_μ . Next, denote the set of gate families $\underline{v} \in V_{\mu,N}^\circ$ such that f is an asymptotic upper bound of \underline{v} in V_μ (see above) by $\mathcal{O}_{V,N}(f)$. We subsequently assume $\phi_{\chi,M}$ Banach–Mazur computable, in which case we also have $\mathcal{O}_{\text{SU},N}(\chi) = \text{SU}_{\mu,N}^\circ$. From our preceding observations, we conclude that $\tilde{\varphi}_{f,M} : \mathcal{O}_{V,N}(f) \rightarrow \mathbb{R}$ is computationally pathological nevertheless for “unfortunate” choices of V_μ . Furthermore, there might exist gate families $\underline{v} \in V_{\mu,N}^\circ$ such that we have $\underline{v} \in \mathcal{O}_{V,N}(f)$ but not $\underline{v} \in \mathcal{O}_{V,N}(\chi)$.

Let $V_\mu \subset \text{SU}_\mu(K)$ be an arbitrary subgroup of $\text{SU}_\mu(K)$ and $\mathcal{Z} \subset \mathbb{C}^{K \times K}$ an arbitrary set. We define (again with some abuse of notation) the group of \mathcal{Z} -symmetric unitary matrices

$$\text{SU}_\mu(\mathcal{Z}) := \left\{ U \in \text{SU}_\mu(K) : UZU^\dagger = Z \text{ for all } Z \in \mathcal{Z} \right\}.$$

If we have $V_\mu \subset \text{SU}_\mu(\mathcal{Z})$, we call V_μ a \mathcal{Z} -symmetric subgroup of $\text{SU}_\mu(K)$. Since $\text{SU}_\mu(\mathcal{Z})$ is a group, we have $\text{efcG}(\underline{v}) \subseteq \text{SU}_\mu(\mathcal{Z})$ for all $\underline{v} \in \text{SU}_{\mu,N}(\mathcal{Z})$. However, as indicated above, it may not be possible to find a gate family $\underline{v} \in \text{SU}_{\mu,N}(\mathcal{Z})$ that respects locality and, at the same time, satisfies $\text{efcG}(\underline{v}) = \text{SU}_\mu(\mathcal{Z})$.

Finally, consider $V_\mu \subset \text{SU}_\mu(K)$ as above, and assume there exists another subgroup $W_\mu \subset \text{SU}_\mu(K)$ and an isometric embedding $\mathfrak{g} : U_\mu(1) \rightarrow \text{SU}_\mu(K)$ such that

- a) W_μ is $\mathfrak{g}U_\mu(1)$ -symmetric (note that W_μ is $\mathfrak{g}U_\mu(1)$ -symmetric if and only if $\mathfrak{g}U_\mu(1)$ is W_μ -symmetric);
- b) we have $V_\mu = \{WU : W \in W_\mu, U \in \mathfrak{g}U_\mu(1)\}$.

Let $f : \mathbb{N} \rightarrow \mathbb{Q}$ once more be a computable growth function, and assume $\mathcal{O}_{V,N}(f)$ is nonempty. As follows from Theorem 3 and its proof, the existence of W_μ and \mathfrak{g} as above is a sufficient condition for $\tilde{\varphi}_{f,M} : \mathcal{O}_{V,N}(f) \rightarrow \mathbb{R}$ —in a broader sense, for V_μ as a whole as well—to be computationally ill behaved. Perhaps, if $\phi_{f,M}$ turns out computable, the condition may also be a necessary one. Notably, the latter case would attribute computational ill behavior entirely to the presence of a (sub)system wide phase factor.

While it is impossible to determine a quantum system’s global phase (cf., Section IV), relative phase factors play a crucial role in many applications of quantum computing (see, e.g., [13, Sect. 13, p. 116ff]). Recently, Li et al. [36] provided some novel understanding of locality and symmetry relations in the context of computational efficiency, explicitly discussing the role of relative phase factors. Operationally, the distinction between global and relative phase relates to a fundamental question: *what is a suitable measure for the similarity of quantum gates?* The relevant literature commonly resorts to metrics that emerge from the matrix norm $\|\cdot\| : \mathbb{C}^{L \times L} \rightarrow \mathbb{R}$. Since $\|\cdot\|$ induces a metric that is sensitive to multiplying either of its arguments by phase factors $z \in U(1)$, it is customary to consider the special unitary group. Strictly speaking, however, $\text{SU}(L)$ is closed under multiplication with L th roots of unity and thus not entirely free of phase factors.

In order to remove any phase sensitivity completely, we may define $d_{\mathbb{P}} : \Xi_{\mu} \times \Xi_{\mu} \rightarrow \mathbb{R}_{\mu}$ through

$$d_{\mathbb{P}}(\mathbf{U}, \mathbf{V}) := \min_{|z|=1} \|\mathbf{U} - z\mathbf{V}\|.$$

Then, $d_{\mathbb{P}}$ is a Banach–Mazur computable pseudometric that satisfies $0 \leq d_{\mathbb{P}}(\mathbf{U}, \mathbf{V}) \leq d_{\Xi}(\mathbf{U}, \mathbf{V})$ for all $\mathbf{U}, \mathbf{V} \in \Xi_{\mu}$. The equivalence relation

$$\mathbf{U} \sim_{\mathbb{P}} \mathbf{V} :\Leftrightarrow d_{\mathbb{P}}(\mathbf{U}, \mathbf{V}) = 0, \quad \mathbf{U}, \mathbf{V} \in \Xi_{\mu}$$

then provides the computable projective unitary group $\mathbb{P}\mathbf{U}_{\mu}$. The reader may verify that the SU-variant of Theorem 3 applies analogously: due to the Banach–Mazur computability of $d_{\mathbb{P}}$, all relevant constructions transfer to the $\mathbb{P}\mathbf{U}_{\mu}$ -setting.

In contrast to its special and general counterparts, the projective unitary group suppresses the notion of a global phase entirely. Albeit not being more common than d_{Ξ} , $d_{\mathbb{P}}$ is thus arguably a more suitable measure for gate similarity on the global scale. In any potentially local setting, however, the situation is much different. Given $\mathbf{U}, \mathbf{V} \in \Xi_{\mu}(L)$, $L \in \mathbb{N}$, we henceforth denote

$$\mathbf{U} \oplus \mathbf{V} := \mathbf{U} \otimes |e_1\rangle\langle e_1| + \mathbf{V} \otimes |e_2\rangle\langle e_2| \in \Xi_{\mu}(2L).$$

Observe that we have $\mathbf{U} \oplus \mathbf{V} \in \Xi_{\mu}(2L)$. For $N \in \mathbb{N}$, let $\underline{\mathbf{u}} \in \mathbf{SU}_{\mu,N}^{\circ}(L)$ and $z \in \{\exp(j\pi\theta) : \theta \in \mathbb{R}_{\mu} \setminus \mathbb{Q}\}$ be arbitrary. We define the gate family $\underline{\mathbf{v}} \in \mathbf{SU}_{\mu,N}(2L)$ through

$$\underline{\mathbf{v}}(n) := \begin{cases} \text{Id} \oplus \underline{\mathbf{u}}(n), & \text{if } 0 \leq n \leq N \\ z\text{Id} \oplus \bar{z}\text{Id}, & \text{if } n = N + 1 \end{cases}$$

i.e., $\underline{\mathbf{v}}$ consists of controlled variants of $\underline{\mathbf{u}}(1), \dots, \underline{\mathbf{u}}(N)$ plus a relative phase shift [14, Sect. 4.3, p. 177 ff]. In addition, consider

$$\mathfrak{g}(x) := x\text{Id} \oplus \bar{x}\text{Id}, \quad x \in \mathbf{U}_{\mu}(1).$$

Accordingly, $\mathfrak{g} : \mathbf{U}_{\mu}(1) \rightarrow \mathbf{SU}_{\mu}(2L)$ is an isometric embedding—arguably the simplest of its kind. Finally, define

$$\mathbf{V}_{\mu} := \text{efcG}(\underline{\mathbf{v}}) \quad \mathbf{W}_{\mu} := \{\text{Id} \oplus \mathbf{U} : \mathbf{U} \in \mathbf{SU}_{\mu}(L)\}.$$

Then, the subgroup \mathbf{W}_{μ} is $\mathfrak{g}\mathbf{U}_{\mu}(1)$ -symmetric. Moreover, we obtain $\mathbf{V}_{\mu} = \{\mathbf{W}\mathbf{U} : \mathbf{W} \in \mathbf{W}_{\mu}, \mathbf{U} \in \mathfrak{g}\mathbf{U}_{\mu}(1)\}$. Accordingly, \mathbf{V}_{μ} “inherits” the computational properties of $\mathbf{U}_{\mu}(1)$ in the above sense. Specifically, the same holds true in the $\mathbb{P}\mathbf{U}_{\mu}$ -setting, since we have

$$d_{\mathbb{P}}(x\text{Id} \oplus \bar{x}\mathbf{U}, y\text{Id} \oplus \bar{y}\mathbf{V}) = 0 \Leftrightarrow d_{\mathbf{U}}(\bar{x}^2\mathbf{U}, \bar{y}^2\mathbf{V}) = 0$$

for all $\mathbf{U}, \mathbf{V} \in \mathbf{SU}_{\mu}(L)$, $x, y \in \mathbf{U}_{\mu}(1)$. In this context, local gate families provide the operational interpretation of strong universality: In contrast to computational and $\mathbb{P}\mathbf{U}_{\mu}$ -universality, it allows from controlling relative phases when embedding the relevant quantum gates into larger supersystems. Notably, this is precisely the case for quantum phase estimation (cf., e.g., [13, Sect. 13.5, p. 125ff] or [14, Sect. 5.2, p. 221ff]), which underlies a large fraction of quantum algorithms developed to date.

VIII. CONCLUSION

In previous publications [11], [12], we have started developing an effective formalism of quantum circuits that builds upon the tools of computability theory. The present article extends this formalism by a model and analysis of exact quantum circuit synthesis and quantum big-O theory. Note that we imposed an asymmetric setting, i.e., we did not assume that gate families are closed under inversion. Instead of $\underline{\mathbf{u}} \in \Xi_{\mu,N}$, $N \in \mathbb{N}$, as defined in Section IV, we could have equally considered a symmetric setting, i.e., gate families of the form

$$\begin{aligned} \tilde{\underline{\mathbf{u}}} : \{-N, \dots, N\} &\rightarrow \Xi_{\mu}, \dots \\ \dots n &\mapsto \tilde{\underline{\mathbf{u}}}(n) := \begin{cases} \underline{\mathbf{u}}(n), & \text{if } 0 \leq n \leq N \\ \underline{\mathbf{u}}(-n)^{\dagger}, & \text{otherwise.} \end{cases} \end{aligned}$$

The reader may verify that our methods readily transfer to the symmetric setting, and our results apply analogously.

Quantum physics and theoretical quantum engineering employ a highly abstract language, with constructive aspects—such as computable analysis—almost nonexistent. Nevertheless, the successful real-world implementation of quantum computers, including the automation of tasks, such as quantum compiling and quantum error correction, arguably relies on constructive theories. Nonconstructive mathematics is largely indifferent to discontinuities and infinities. In real-world engineering, these manifest in instabilities and sensitivity toward perturbations. Their intrinsic relations to mathematical topology equip constructive theories with an inherent aspect of continuity, entailing more robust forms of convergence and function behavior. The lack of such theories in quantum engineering is thus in line with the error-proneness and scaling obstacles contemporary quantum computers experience. Hence, the authors believe that parts of the relevant mathematical basis may require a critical review.

On a final remark concerning perturbation sensitivity, observe that [5] hinted toward the ambiguous role of almost-trivial quantum gates—that is, quantum gates that are close to the identity matrix—in the efficiency analysis of gate families and the circuit model of quantum computing in general. On the one hand, such gates are a significant source of instability in compiling and circuit synthesis [37]. On the other hand, they are fundamental to quantum computing from both the practical and the theoretical perspective. For example, the Solovay–Kitaev “algorithm” [4] relies on almost-trivial unitary matrices. Almost-trivial gate families play an equally essential role in our proof of Theorem 3 (cf., Appendix B3), and we will briefly get back to this discussion thereafter.

APPENDIX A MATHEMATICAL AUXILIARIES

In the following, we provide several mathematical auxiliaries necessary for proving this article’s results. We divide the rest of Appendix A into subsections, each of which refers thematically to a section in the article’s main body.

A. SECTION III: COMPUTABLE ANALYSIS

From Section III, recall the idea of μ -recursive functions and Turing machines. In the context of Turing machines, the partiality of μ -recursive functions exhibits a dedicated interpretation. Consider a Turing machine that computes some μ -recursive function

$$g : \mathbb{N} \times \cdots \times \mathbb{N} \supseteq \rightarrow \mathbb{N}$$

and note that Turing machines are state-based automata that execute sequential processing steps. The function's domain $D(g) \subseteq \mathbb{N} \times \cdots \times \mathbb{N}$ corresponds to the set of inputs for which the Turing machine eventually terminates its computation and returns some (well-defined) output. As a consequence of the well-known *halting problem*, $D(g)$ is, in general, not a recursive subset of $\mathbb{N} \times \cdots \times \mathbb{N}$. That is, the *indicator function*

$$\mathbb{1}_{D(g)} : \mathbb{N} \times \cdots \times \mathbb{N} \rightarrow \{0, 1\}, \mathbf{n} \mapsto \begin{cases} 1, & \text{if } \mathbf{n} \in D(g) \\ 0, & \text{otherwise} \end{cases}$$

is not necessarily a μ -recursive function. Particularly, we call a set $\mathcal{A} \subseteq \mathbb{N}$ recursively enumerable if we have $\mathcal{A} = D(g)$ for any μ -recursive function $g : \mathbb{N} \supseteq \rightarrow \mathbb{N}$. Provided \mathcal{A} is nonempty, this is the case if and only if there exists a total μ -recursive function $h : \mathbb{N} \rightarrow \mathbb{N}$ such that we have $\mathcal{A} = \{h(n) : n \in \mathbb{N}\}$, i.e., the function h enumerates the set \mathcal{A} . If \mathcal{A} and $\mathbb{N} \setminus \mathcal{A}$ are both recursively enumerable, we refer to \mathcal{A} as recursive. As indicated above, a set is recursive if and only if its indicator function is μ -recursive.

Let $(\mathcal{A}_n)_{n \in \mathbb{N}}$ be a sequence of recursively enumerable sets. We call $(\mathcal{A}_n)_{n \in \mathbb{N}}$ computable if there exists a μ -recursive function $g : \mathbb{N} \times \mathbb{N} \supseteq \rightarrow \mathbb{N}$ such that for all $n \in \mathbb{N}$, we have

$$\mathcal{A}_n = \{m \in \mathbb{N} : (n, m) \in D(g)\}.$$

Accordingly, the following statements are equivalent.

- The sequence $(\mathcal{A}_n)_{n \in \mathbb{N}}$ is a computable sequence of recursively enumerable sets.
- There exists a μ -recursive function $g : \mathbb{N} \times \mathbb{N} \supseteq \rightarrow \mathbb{N}$ such that for all $n \in \mathbb{N}$, we have $\mathcal{A}_n = \{m \in \mathbb{N} : (n, m) \in D(g)\}$.
- There exists a μ -recursive function $h : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ with domain $D(h) = \{(n, m) \in \mathbb{N} \times \mathbb{N} : \mathcal{A}_n \neq \emptyset\}$ such that for all $n \in \mathbb{N}$, we either have $\mathcal{A}_n = \{h(n, m) : m \in \mathbb{N}\}$ (in which case $\mathcal{A}_n \neq \emptyset$) or $\mathcal{A}_n = \emptyset$.

The reader may find a comprehensive analysis of recursively enumerable sets and their properties in, e.g., [21].

Paired with notion of recursively enumerable sets, the subsequent lemma on the semi decidability of the (strict) inequality relation over \mathbb{R}_μ will be an essential tool in proving this article's theoretical contributions.

Lemma 1: Given $x \in \mathbb{R}_\mu$, let $(y_{n,m})_{n,m \in \mathbb{N}} \subset \mathbb{R}_\mu$ be a computable sequence and consider

$$\mathcal{Y}_> := \{n \in \mathbb{N} : \exists m \in \mathbb{N} \text{ s.t. } y_{n,m} > x\}$$

$$\mathcal{Y}_< := \{n \in \mathbb{N} : \exists m \in \mathbb{N} \text{ s.t. } y_{n,m} < x\}$$

where $\neg \mathcal{Y}_> := \mathbb{N} \setminus \mathcal{Y}_>$ and $\neg \mathcal{Y}_< := \mathbb{N} \setminus \mathcal{Y}_<$.

- 1) There exists a μ -recursive function $g_> : D(g_>) \rightarrow \mathbb{N}$ with $D(g_>) = \mathcal{Y}_>$ such that for all $n \in D(g_>)$, we have $y_{n,g_>(n)} > x$. Accordingly, we have $\mathcal{Y}_> \xrightarrow{\mu} \neg \mathcal{Y}_>$.
- 2) There exists a μ -recursive function $g_< : D(g_<) \rightarrow \mathbb{N}$ with $D(g_<) = \mathcal{Y}_<$ such that for all $n \in D(g_<)$, we have $y_{n,g_<(n)} < x$. Accordingly, we have $\mathcal{Y}_< \xrightarrow{\mu} \neg \mathcal{Y}_<$.

Proof: See [12, Lemma 1, Appendix A]. \square

Building upon Lemma 1 we are now able to prove a most fundamental result³ from computable topology (additional details may be found in [22]). For the remainder of Appendix A1, we will tacitly assume all relevant sets equipped with a formal definition of computable sequences and a Banach–Mazur-computable (pseudo)metric.

Lemma 2: Let \mathcal{A}_μ be effectively closed and $(a_n)_{n \in \mathbb{N}} \subset \mathcal{A}_\mu$ a computable sequence with effective limit $a_* \in \mathcal{A}_\mu \setminus \{a_n\}_{n \in \mathbb{N}}$. We have $\{a_n\}_{n \in \mathbb{N}} \xrightarrow{\mu} \{a_*\}$ but $\{a_*\} \not\xrightarrow{\mu} \{a_n\}_{n \in \mathbb{N}}$.

Before stating its proof, we provide a brief intuitive interpretation of Lemma 2. In (very) informal terms, a_* is the “value” of $(a_n)_{n \in \mathbb{N}}$ at infinity, i.e., at “ $n = \infty$.” For any computable sequence $(\tilde{a}_n)_{n \in \mathbb{N}} \subset (\{a_n\}_{n \in \mathbb{N}} \cup \{a_*\})$, index values $n \in \mathbb{N}$ that satisfy $\tilde{a}_n \in \{a_m\}_{m \in \mathbb{N}}$ thus correspond to computations that, after a finite number of computing steps, reach their terminal value and halt. Analogously, index values $n \in \mathbb{N}$ that satisfy $\tilde{a}_n = a_*$ correspond to computations that require an infinite number of computing steps to reach their terminal value. At its core, Lemma 2 thus follows by reduction of the halting problem.

Proof of Lemma 2: We first prove $\{a_n\}_{n \in \mathbb{N}} \xrightarrow{\mu} \{a_*\}$ (§1). The subsequent proof of $\{a_*\} \not\xrightarrow{\mu} \{a_n\}_{n \in \mathbb{N}}$ consists of two construction steps (Parts 2.1 and 2.2) and a contradiction step (Part 2.3).

Part 1. Let $(\tilde{a}_n)_{n \in \mathbb{N}} \subset (\{a_n\}_{n \in \mathbb{N}} \cup \{a_*\})$ be any computable sequence. Then, defining $(x_n)_{n \in \mathbb{N}} : x_n = \|a_* - \tilde{a}_n\|$, we have

$$x_m > 0 \quad \Leftrightarrow \quad \tilde{a}_m \in \{a_n\}_{n \in \mathbb{N}}$$

for all $m \in \mathbb{N}$. Further, $(x_n)_{n \in \mathbb{N}}$ is a computable sequence of (nonnegative) real numbers.

Next, let $(q_{n,m})_{n,m \in \mathbb{N}} \subset \mathbb{Q}$ be any computable sequence such that for all $n, m \in \mathbb{N}$, we have

$$0 \leq q_{n,m} \leq q_{n,m+1} \leq \sup_{l \in \mathbb{N}} q_{n,l} = x_n.$$

Then, for $n \in \mathbb{N}$ arbitrary, we have $x_n > 0$ if and only if there exists $m \in \mathbb{N}$ such that $q_{n,m} > 0$.

Finally, observe that every computable sequence of rational numbers is also a computable sequence of real numbers. Accordingly, due to Lemma 1, there exists a μ -recursive function $g : \mathbb{N} \supseteq \rightarrow \mathbb{N}$ with domain

$$D(g) = \{n \in \mathbb{N} : \exists m \in \mathbb{N} \text{ s.t. } q_{n,m} > 0\}$$

³ Note that Lemma 2 and [12, Lemma 2, Appendix A] (including their proofs) are identical. Since the proof of Lemma 4 explicitly references the proof of Lemma 2 on several occasions, we repeat the proof of Lemma 2 in the present article for the sake of completeness.

$$= \{m \in \mathbb{N} : \tilde{a}_m \in \{a_n\}_{n \in \mathbb{N}}\}.$$

Since $(\tilde{a}_n)_{n \in \mathbb{N}}$ was arbitrary, we have $\{a_n\}_{n \in \mathbb{N}} \xrightarrow{\mu} \{a_*\}$.

Part 2.1. Let $\mathcal{B} \subset \mathbb{N}$ be a recursively enumerable, nonrecursive set and let $g_{1,\mathcal{B}} : \mathbb{N} \rightarrow \mathcal{B}$ be any total and surjective μ -recursive function. Then, we have $\mathcal{B} = \{g_{1,\mathcal{B}}(n) : n \in \mathbb{N}\}$. We define $g_{2,\mathcal{B}} : \mathbb{N}^2 \rightarrow \mathbb{N}$, $(n, m) \mapsto g_{2,\mathcal{B}}(n, m)$, through

$$g_{2,\mathcal{B}}(n, m) := \begin{cases} 1, & \text{if } n \in \{g_{1,\mathcal{B}}(0), \dots, g_{1,\mathcal{B}}(m)\} \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

Furthermore, we define $h_{\mathcal{B}} : \mathbb{N}^2 \rightarrow \mathbb{N}$, $(n, m) \mapsto h(n, m)$ according to

$$h(n, m) := \sum_{l=0}^m (1 - g_{2,\mathcal{B}}(n, l)). \quad (12)$$

Thus, $g_{2,\mathcal{B}}$ and h are total μ -recursive functions. With $\neg\mathcal{B} := \mathbb{N} \setminus \mathcal{B}$, we arrive at the following case distinction.

- 1) We have $h(n, m) = \min\{m, \min\{l \in \mathbb{N} : g_{2,\mathcal{B}}(n, l) = 1\}\}$ for all $n \in \mathcal{B}, m \in \mathbb{N}$.
- 2) We have $h(n, m) = m$ for all $n \in \neg\mathcal{B}, m \in \mathbb{N}$.

Next, recall the computable sequence $(a_n)_{n \in \mathbb{N}} \subset \mathcal{A}_\mu$ from the lemma's conditions and define

$$(a_{n,m})_{n,m \in \mathbb{N}} : a_{n,m} = a_{h(n,m)} \quad (13)$$

$$(a_{n,*})_{n \in \mathbb{N}} : a_{n,*} = \lim_{m \rightarrow \infty} a_{n,m}. \quad (14)$$

Then, $(a_{n,m})_{n,m \in \mathbb{N}} \subset \mathcal{A}_\mu$ is a computable sequence that converges toward $(a_{n,*})_{n \in \mathbb{N}} \subset \mathcal{A}_\mu$ for $m \rightarrow \infty$ (the limit value of is well defined for all $n \in \mathbb{N}$ by construction). In Part 2.2, we will show that this convergence is effective, implying that $(a_{n,*})_{n \in \mathbb{N}}$ a computable sequence as well.

Part 2.2. Let $\nu : \mathbb{N} \rightarrow \mathbb{N}$ be a μ -recursive modulus of convergence for $(a_n)_{n \in \mathbb{N}}$ and observe the following.

- 1) If $l \in \mathcal{B}$ holds true, we have $a_{l,*} = a_m \in \{a_n\}_{n \in \mathbb{N}}$ with $m = \min\{k \in \mathbb{N} : g_{2,\mathcal{B}}(l, k) = 1\}$.
- 2) If $l \in \neg\mathcal{B}$ holds true, we have $a_{l,*} = a_*$.

Accordingly, $(a_{n,m})_{n,m \in \mathbb{N}}$ satisfies the subsequent case distinction for all $m, M \in \mathbb{N}$, $m \geq \nu(M + 1)$:

- 1) If $n \in \mathcal{B}$ and $m < l$ with $l = \min\{k \in \mathbb{N} : g_{2,\mathcal{B}}(n, k) = 1\}$ hold true, we have

$$d(a_{n,*}, a_{n,m}) \leq \dots$$

$$\dots d(a_*, a_l) + d(a_*, a_m) \leq 2 \cdot 2^{-(M+1)} = 2^{-M}$$

by application of $a_{n,*} = a_l$, $a_{n,m} = a_m$, and the triangle inequality.

- 2) If $n \in \mathcal{B}$ and $m \geq l$ with $l = \min\{k \in \mathbb{N} : g_{2,\mathcal{B}}(n, k) = 1\}$ hold true, we have

$$d(a_{n,*}, a_{n,m}) = d(a_l, a_l) = 0 \leq 2^{-M}.$$

- 3) Last but not least, whenever $n \in \neg\mathcal{B}$ holds true, we have $d(a_{n,*}, a_{n,m}) = d(a_*, a_m) \leq 2^{-(M+1)} \leq 2^{-M}$.

Consequently, $(a_{n,m})_{n,m \in \mathbb{N}}$ converges effectively toward $(a_{n,*})_{n \in \mathbb{N}}$ for $m \rightarrow \infty$. Since \mathcal{A}_μ is effectively closed by requirement, $(a_{n,*})_{n \in \mathbb{N}}$ must be a computable sequence.

Part 2.3. We now prove $\{a_*\} \xrightarrow{\mu} \{a_n\}_{n \in \mathbb{N}}$ by contradiction. To this end, assume $\{a_*\} \xrightarrow{\mu} \{a_n\}_{n \in \mathbb{N}}$ and observe that we have

$$\neg\mathcal{B} = \{m \in \mathbb{N} : a_m \in \{a_*\}\}.$$

Accordingly, there must exist a μ -recursive function $g : \mathbb{N} \supseteq \rightarrow \mathbb{N}$ with $D(g) = \neg\mathcal{B}$. But we chose \mathcal{B} recursively enumerable and nonrecursive, which is absurd. \square

Note that whenever $\mathcal{A}_{1,\mu}$ is effectively dense in $\mathcal{A}_{2,\mu}$, the pathological behavior characterized by Lemma 2 occurs at every point of $\mathcal{A}_{2,\mu}$. Moreover, consider $\mathcal{B}_{1,\mu} \subseteq \mathcal{A}_{1,\mu}$, $\mathcal{B}_{2,\mu} \subseteq \mathcal{A}_{2,\mu}$, and assume $\mathcal{A}_{1,\mu} \cap \mathcal{A}_{2,\mu} = \emptyset$. Then

- 1) if we have $\mathcal{A}_{1,\mu} \xrightarrow{\mu} \mathcal{A}_{2,\mu}$, we also have $\mathcal{B}_{1,\mu} \xrightarrow{\mu} \mathcal{B}_{2,\mu}$;
- 2) if we have $\mathcal{B}_{1,\mu} \xrightarrow{\mu} \mathcal{B}_{2,\mu}$, we also have $\mathcal{A}_{1,\mu} \xrightarrow{\mu} \mathcal{A}_{2,\mu}$.

Accordingly, we obtain the subsequent Lemma 3 as a generalization of Lemma 2.

Lemma 3: Let $\mathcal{A}_{1,\mu} \subset \mathcal{A}_\mu$ and $\mathcal{A}_{2,\mu} \subset \mathcal{A}_\mu$ be nonempty, and let \mathcal{A}_μ be effectively closed. Further, let $\mathcal{A}_{1,\mu}$ be effectively dense in $\mathcal{A}_{2,\mu}$. We have $\mathcal{A}_{2,\mu} \xrightarrow{\mu} \mathcal{A}_{1,\mu}$.

Proof: The claim is an immediate consequence of applying Lemma 2 to Definition 8. \square

To prove this article's main contributions—Theorem 1 in Section V and Theorem 3 in Section VII—we require more intricate reductions of the halting problem. Ultimately, however, the subsequent Lemma 4 builds upon the same principles as Lemma 2. In the following, let \mathcal{A}_μ be effectively closed and consider a function $F : \mathcal{A}_\mu \supseteq \rightarrow \mathbb{R}$. Further, consider a computable sequence $(a_n)_{n \in \mathbb{N}} \subset D(F)$ that converges effectively toward some point $a_* \in D(F) \setminus \{a_n\}_{n \in \mathbb{N}}$. If, for all $n \in \mathbb{N}$, we have $n \leq F(a_n)$, we say that F isolates the sequence $(a_n)_{n \in \mathbb{N}}$ from its limit point a_* .

Lemma 4 (Principle of Isolation): Consider an effectively closed set \mathcal{A}_μ and a function $F : \mathcal{A}_\mu \supseteq \rightarrow \mathbb{R}$ that isolates the (computable) sequence $(a_n)_{n \in \mathbb{N}} \subset D(F)$ from its limit point $a_* \in D(F) \setminus \{a_n\}_{n \in \mathbb{N}}$. Then, there exists a computable sequence $(a_{n,*})_{n \in \mathbb{N}} \subset \mathcal{A}_\mu$ that satisfies the following.

- 1) We have $\{a_{n,*}\}_{n \in \mathbb{N}} \subset (\{a_n\}_{n \in \mathbb{N}} \cup \{a_*\}) \subseteq D(F)$ and the unique accumulation point of $\{a_{n,*}\}_{n \in \mathbb{N}}$ equals a_* . With $\mathcal{B} := \{n \in \mathbb{N} : a_{n,*} \neq a_*\}$, $\neg\mathcal{B} := \mathbb{N} \setminus \mathcal{B}$, we furthermore have $\mathcal{B} \xrightarrow{\mu} \neg\mathcal{B}$ and $\neg\mathcal{B} \xrightarrow{\mu} \mathcal{B}$.
- 2) Let $\bar{f} : \mathbb{N} \rightarrow \mathbb{Q}$ be any function such that for all $n \in \mathbb{N}$, we have $\bar{f}(n) \geq F(a_{n,*})$. Then, \bar{f} is not a (Banach–Mazur) computable function.
- 3) For all $q \in \mathbb{Q}$, $q \geq F(a_*)$, and with $\mathcal{B}_q := \{n \in \mathbb{N} : F(a_{n,*}) > q\}$, $\neg\mathcal{B}_q := \mathbb{N} \setminus \mathcal{B}_q$, we have $\mathcal{B}_q \xrightarrow{\mu} \neg\mathcal{B}_q$ and $\neg\mathcal{B}_q \xrightarrow{\mu} \mathcal{B}_q$.

Proof: First, let $\mathcal{B} \subset \mathbb{N}$ be a recursively enumerable, nonrecursive set and let $g_{1,\mathcal{B}} : \mathbb{N} \rightarrow \mathcal{B}$ be any total and surjective

μ -recursive function. Accordingly, we have $\mathcal{B} = \{g_{1,\mathcal{B}}(n) : n \in \mathbb{N}\}$. We define

- a) the μ -recursive functions $g_{2,\mathcal{B}}, h_{\mathcal{B}} : \mathbb{N}^2 \rightarrow \mathbb{N}$ according to (11) and (12), respectively;
- b) the sequences $(a_{n,m})_{n,m \in \mathbb{N}}, (a_{n,*})_{n \in \mathbb{N}} \subset \mathcal{A}_{\mu}$ according to (13) and (14), respectively.

Accordingly, recall the Proof of Lemma 2 (Parts 2.1, 2.2, and 2.3) and observe the following.

- 1) The sequences $(a_{n,m})_{n,m \in \mathbb{N}}$ and $(a_{n,*})_{n \in \mathbb{N}}$ are both computable in \mathcal{A}_{μ} . Further, we have $\{a_{n,m}\}_{n,m \in \mathbb{N}} = \{a_n\}_{n \in \mathbb{N}} \cup \{a_*\}$.
- 2) The sequence $(a_{n,m})_{n,m \in \mathbb{N}}$ converges effectively toward $(a_{n,*})_{n \in \mathbb{N}}$ for $m \rightarrow \infty$.
- 3) We have $\mathcal{B} = \{m \in \mathbb{N} : a_m \neq a_*\}$ and, with $\neg\mathcal{B} = \mathbb{N} \setminus \mathcal{B}$, $\neg\mathcal{B} = \{m \in \mathbb{N} : a_m = a_*\}$. Consequently, \mathcal{B} also satisfies the definition given in Statement 1.

Subsequently, we prove Statements 1–3, one at a time. For clarity, we label the relevant paragraphs accordingly.

Proof of Statement 1. Observe that we have $\{a_{n,*}\}_{n \in \mathbb{N}} \subseteq (\{a_n\}_{n \in \mathbb{N}} \cup \{a_*\})$, $\mathcal{B} = \{n \in \mathbb{N} : a_{n,*} \neq a\} \xrightarrow{\mu} \neg\mathcal{B}$, and $\neg\mathcal{B} \xrightarrow{\mu} \mathcal{B}$ by construction. Accordingly, it remains to show that a_* equals the unique accumulation point of $\{a_{n,*}\}_{n \in \mathbb{N}}$.

Since a_* is the unique accumulation point of the set $(\{a_n\}_{n \in \mathbb{N}} \cup \{a_*\})$, it is likewise the only possible candidate for an accumulation point of the set $\{a_{n,*}\}_{n \in \mathbb{N}}$. Accordingly, it is sufficient to prove that $\{a_*\}$ is indeed an accumulation point of $\{a_{n,*}\}_{n \in \mathbb{N}}$, i.e., we need to prove that

$$\{a \in \{a_{n,*}\}_{n \in \mathbb{N}} : 0 < d(a_*, a) < \epsilon\} \neq \emptyset \quad (15)$$

holds true for all $\epsilon > 0$. Thus, choose $\epsilon > 0$ arbitrarily and observe the following.

- 1) There exists $n \in \mathbb{N}$ such that for all $m \in \mathbb{N}$, $m \geq n$, we have $0 < d(a_*, a_m) < \epsilon$.
- 2) For all $m \in \mathbb{N}$, there exists $l \in \mathcal{B}$ satisfying $\min\{k \in \mathbb{N} : g_{2,\mathcal{B}}(l, k) = 1\} \geq m$. Otherwise, \mathcal{B} was a recursive set.

Accordingly, there must exist $l \in \mathcal{B}$ such that we have

$$\min\{k \in \mathbb{N} : g_{2,\mathcal{B}}(l, k) = 1\} \geq \min\{n \in \mathbb{N} : \dots \\ \dots d(a_*, a_m) < \epsilon \text{ for all } m \in \mathbb{N}, m \geq n\}.$$

Consequently, by construction of $(a_{n,*})_{n \in \mathbb{N}}$ and with $l \in \mathcal{B}$ as above, we also have $0 < d(a_*, a_{l,*}) < \epsilon$, in turn implying (15). Since we chose $\epsilon > 0$ arbitrarily, the claim follows.

Proof of Statement 2. We prove Statement 2 by contradiction. To this end, assume there exists a computable function $\bar{f} : \mathbb{N} \rightarrow \mathbb{Q}$ such that for all $n \in \mathbb{N}$, we have $\bar{f}(n) \geq F(a_{n,*})$. We define $\tilde{g}_{2,\mathcal{B}} : \mathbb{N} \rightarrow \mathbb{N}$, $(n, m) \mapsto \tilde{g}_{2,\mathcal{B}}(n)$, according to

$$\tilde{g}_{2,\mathcal{B}}(n) := g_{2,\mathcal{B}}(n, \min\{l \in \mathbb{N} : l \geq \bar{f}(n)\}).$$

Then, $\tilde{g}_{2,\mathcal{B}}$ is a total μ -recursive function and satisfies the following case distinction for all $n \in \mathbb{N}$.

- 1) If $n \in \mathcal{B}$, there exists $m = \min\{l \in \mathbb{N} : g_{2,\mathcal{B}}(n, l) = 1\} \in \mathbb{N}$ such that we have $a_{n,*} = a_m$ and

$$\min\{l \in \mathbb{N} : l \geq \bar{f}(n)\} \geq \dots \\ \dots \bar{f}(n) \geq F(a_{n,*}) = F(a_m) \geq m.$$

With $k = \min\{l \in \mathbb{N} : l \geq \bar{f}(n)\}$, we have $g_{2,\mathcal{B}}(n, k) = 1$. From the definition of $\tilde{g}_{2,\mathcal{B}}$, it then follows that we have $\tilde{g}_{2,\mathcal{B}}(n) = 1$.

- 2) If $n \in \neg\mathcal{B}$, we have $g_{2,\mathcal{B}}(n, m) = 0$ for all $m \in \mathbb{N}$. Accordingly, with $k = \min\{l \in \mathbb{N} : l \geq \bar{f}(n)\}$, we also have $g_{2,\mathcal{B}}(n, k) = 0$. From the definition of $\tilde{g}_{2,\mathcal{B}}$, it then follows that we have $\tilde{g}_{2,\mathcal{B}}(n) = 0$.

As follows from the above, we have $\mathbb{1}_{\mathcal{B}}(n) = \tilde{g}_{2,\mathcal{B}}(n)$ for all $n \in \mathbb{N}$, i.e., the indicator function of \mathcal{B} is a total μ -recursive function. Hence, \mathcal{B} must be a recursive set, which contradicts the assumption of \mathcal{B} being nonrecursive.

Proof of Statement 3.1. We first prove $\mathcal{B}_q \xrightarrow{\mu} \neg\tilde{\mathcal{B}}_q$. Since we have $q \leq F(a_n)$ for all $n \in \mathbb{N}$, $n \geq \min\{m \in \mathbb{N} : m \geq q\}$, the set

$$\neg\tilde{\mathcal{B}}_q := \{n \in \mathbb{N} : F(a_n) \leq q\} \quad (16)$$

is finite. Accordingly, $\tilde{\mathcal{B}}_q$ is recursively enumerable. Finally, observe that by construction of $(a_{m,*})_{m \in \mathbb{N}}$, we have

$$\mathcal{B}_q = \{n \in \mathcal{B} : \min\{m \in \mathbb{N} : g_{2,\mathcal{B}}(n, m) = 1\} \in \tilde{\mathcal{B}}_q\}.$$

As both \mathcal{B} and $\tilde{\mathcal{B}}_q$ are recursively enumerable, so must be \mathcal{B}_q .

Proof of Statement 3.2. We now prove $\mathcal{B}_q \xrightarrow{\mu} \neg\mathcal{B}_q$ by contradiction. To this end, assume $\neg\mathcal{B}_q$ recursively enumerable. Consider $\neg\tilde{\mathcal{B}}_q$ as defined in (16) and observe that we have

$$\neg\mathcal{B} = \{n \in \neg\mathcal{B}_q : \dots \\ \dots \{m \in \neg\tilde{\mathcal{B}}_q : g_{2,\mathcal{B}}(n, m) = 1\} = \emptyset\} \quad (17)$$

by construction of $(a_{m,*})_{m \in \mathbb{N}}$. Since $\neg\tilde{\mathcal{B}}_q$ is finite, it is recursively enumerable, and by assumption, so is $\neg\mathcal{B}_q$. Thus, as follows from (17), $\neg\mathcal{B}$ must be recursively enumerable as well. But we chose \mathcal{B} nonrecursive, which is absurd. \square

Finally, we establish special conditions that imply the applicability of Lemma 4. In the subsequent Lemma 5, observe that we do not require $D(F)$ to include the set $\mathcal{A}_{2,\mu}$.

Lemma 5: Let $\mathcal{A}_{1,\mu}$ and $\mathcal{A}_{2,\mu}$ be sets and $F : \mathcal{A}_{1,\mu} \rightarrow \mathbb{Q}$ a function. Furthermore, assume the following.

- a) With respect to the relevant (pseudo)metric, $\mathcal{A}_{1,\mu}$ is effectively dense in $\mathcal{A}_{2,\mu}$.
- b) For every computable sequence $(\tilde{a}_n)_{n \in \mathbb{N}} \subset \mathcal{A}_{1,\mu}$, there exists a computable double sequence $(q_{n,m})_{n,m \in \mathbb{N}} \subset \mathbb{Q}$, such that for all $n, m \in \mathbb{N}$, we have $q_{n,m} \leq q_{n,m+1} \leq \sup_{l \in \mathbb{N}} q_{n,l} = F(\tilde{a}_n)$.
- c) For every computable sequence $(\tilde{a}_n)_{n \in \mathbb{N}} \subset \mathcal{A}_{1,\mu}$ with effective limit in $\mathcal{A}_{2,\mu}$ and all $q \in \mathbb{Q}$, $q > 0$, there exists $m \in \mathbb{N}$ such that we have $F(\tilde{a}_m) > q$.

Then, if $(a_{n,*})_{n \in \mathbb{N}} \subset \mathcal{A}_{2,\mu}$ is any computable sequence with effective limit in $\mathcal{A}_{1,\mu}$, the sequence's limit point

$$\lim_{n \rightarrow \infty} a_{n,*} =: a_* \in \mathcal{A}_{1,\mu}$$

is also the effective limit of some computable sequence $(a_n)_{n \in \mathbb{N}} \subset \mathcal{A}_{1,\mu}$ such that F isolates $(a_n)_{n \in \mathbb{N}}$ from a_* .

Proof of Lemma 5: We subsequently divide the proof into paragraphs. First, we construct an ancillary sequence $(\tilde{a}_{n,m})_{n,m \in \mathbb{N}}$ (Part 1) and an ancillary function h (Part 2). Then, based on $(\tilde{a}_{n,m})_{n,m \in \mathbb{N}}$ and h , we construct the required sequence $(a_n)_{n \in \mathbb{N}}$ (Part 3).

Part 1. Let $(a_{n,*})_{n \in \mathbb{N}} \subset \mathcal{A}_{2,\mu}$ be a computable sequence with effective limit point $\lim_{n \rightarrow \infty} a_{n,*} =: a_* \in \mathcal{A}_{1,\mu}$. Further, let $v : \mathbb{N} \rightarrow \mathbb{N}$ be a μ -recursive modulus of convergence for $(a_{n,*})_{n \in \mathbb{N}}$. We define $(\tilde{a}_{n,*})_{n \in \mathbb{N}} : \tilde{a}_{n,*} = a_{v(n+1),*}$. Then

$$d(a_*, \tilde{a}_{n,*}) < 2^{-(n+1)} \quad (18)$$

holds true for all $n \in \mathbb{N}$. By assumption (a), $\mathcal{A}_{1,\mu}$ is effectively dense in $\mathcal{A}_{2,\mu}$. Thus, there exists a computable sequence $(a_{n,m})_{n,m \in \mathbb{N}} \subset \mathcal{A}_{1,\mu}$ that converges effectively toward $(\tilde{a}_{n,*})_{n \in \mathbb{N}}$ for $m \rightarrow \infty$. Let $\tilde{v} : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ be a μ -recursive modulus of convergence for $(a_{n,m})_{n,m \in \mathbb{N}}$. Similarly to the above, we define $(\tilde{a}_{n,m})_{n,m \in \mathbb{N}} : \tilde{a}_{n,m} = a_{n,v(n+m+1)}$. Then

$$d(\tilde{a}_{n,*}, \tilde{a}_{n,m}) < 2^{-(n+m+1)} \quad (19)$$

holds true for all $n, m \in \mathbb{N}$.

Part 2. By assumption (b), there exists a computable sequence $(q_{n,m,l})_{n,m,l \in \mathbb{N}} \subset \mathbb{Q}$ such that for all $n, m, l \in \mathbb{N}$, we have $q_{n,m,l} \leq q_{n,m,l+1} \leq \sup_{k \in \mathbb{N}} q_{n,m,k} = F(\tilde{a}_{n,m})$. Let $(g_1, g_2) : \mathbb{N} \rightarrow \mathbb{N} \times \mathbb{N}$ be a pair of μ -recursive functions that enumerates the set $\mathbb{N} \times \mathbb{N}$. We define $h : \mathbb{N} \rightarrow \mathbb{N}, n \mapsto h(n)$, according to

$$h(n) := g_1(\min\{m \in \mathbb{N} : q_{n,g_1(m),g_2(m)} \geq n\})$$

where it follows from assumption (c) that the relevant minimum always exists. Thus, h is a total μ -recursive function, and we have

$$n \leq \sup_{m \in \mathbb{N}} q_{n,h(n),m} \leq F(\tilde{a}_{n,h(n)}) \quad (20)$$

for all $n \in \mathbb{N}$ by construction of h .

Part 3. Define $(a_n)_{n \in \mathbb{N}} : a_n = \tilde{a}_{n,h(n)}$. Then, $(a_n)_{n \in \mathbb{N}} \subset \mathcal{A}_{1,\mu}$ is a computable sequence. Further, by application of (20), we have $n \leq F(a_n)$ and, by application of (18), (19), and the triangle inequality

$$\begin{aligned} d(a_*, a_n) &\leq d(a_*, \tilde{a}_{n,*}) + d(\tilde{a}_{n,*}, \tilde{a}_{n,h(n)}) < \dots \\ &\dots 2^{-(n+1)} + 2^{-(n+h(n)+1)} \leq 2 \cdot 2^{-(n+1)} = 2^{-n} \end{aligned}$$

for all $n \in \mathbb{N}$. Hence, $(a_n)_{n \in \mathbb{N}}$ converges effectively toward $a_* \in \mathcal{A}_{1,\mu}$ for $n \rightarrow \infty$ and F isolates $(a_n)_{n \in \mathbb{N}}$ from a_* . \square

B. SECTION IV: COMPUTABLE UNITARY MATRICES AND COMPUTABLE GATE FAMILIES

First, we discuss the relationship between unitary matrices and their eigenvalues in view of computability. For $N \in$

$\mathbb{N} \setminus \{0\}$, consider $(U_m)_{m \in \mathbb{N}} \subset \Xi(N)$ and assume there exist computable sequences $(\theta_{1,m})_{m \in \mathbb{N}}, \dots, (\theta_{N,m})_{m \in \mathbb{N}} \subset \mathbb{R}_\mu$, $(V_m)_{m \in \mathbb{N}} \subset \Xi_\mu(N)$, such that we have

$$U_m = \sum_{n=1}^N \exp(j2\pi\theta_{n,m}) V_m |e_n\rangle \langle e_n| V_m^\dagger$$

for all $m \in \mathbb{N}$. Then, $(U_m)_{m \in \mathbb{N}}$ is a computable sequence of computable unitary matrices. On the other hand, let $U \in \Xi(N)$ be arbitrary. Then, there exist numbers $\theta_1, \dots, \theta_N \in \mathbb{R}_\mu$ and a matrix $V \in \Xi_\mu(N)$ that satisfy

$$U = \sum_{n=1}^N \exp(j2\pi\theta_n) V |e_n\rangle \langle e_n| V^\dagger. \quad (21)$$

Moreover, if $\Xi \equiv \text{SU}$, we can choose $\theta_1, \dots, \theta_N \in \mathbb{R}_\mu$ such that we have $\theta_1 + \dots + \theta_N = 0$ in addition to (21).

Consider $\theta_{1,0}, \dots, \theta_{N,0} \in \mathbb{R}_\mu, V \in \Xi_\mu(N)$, and $N \in \mathbb{N}$. As a special case of the above principle, we may define

$$(U_0)^{\frac{1}{m}} := \sum_{n=1}^N \exp\left(j2\pi \frac{\theta_{n,0}}{m}\right) V |e_n\rangle \langle e_n| V^\dagger, m \in \mathbb{N} \setminus \{0\}.$$

Then, $(U_m)_{m \in \mathbb{N}} : U_m = (U_0)^{\frac{1}{(m+1)}}$ is a computable sequence of computable unitary matrices. As follows from basic trigonometry, there exist $l, k \in \mathbb{N}$ such that we have $\|\text{Id} - U_{km+l}\| < \frac{1}{m}$ for all $m \in \mathbb{N}$. Thus, $(U_m)_{m \in \mathbb{N}}$ converges effectively toward Id for $m \rightarrow \infty$. Given $U_0 \in U_\mu(N)$ and $m \in \mathbb{N}$, the matrix $(U_0)^{\frac{1}{(m+1)}}$, depends on the choice of $\theta_{1,0}, \dots, \theta_{N,0}$. This dependency is irrelevant to us, up to one exception: If $\Xi \equiv \text{SU}$, we must choose $\theta_{1,0}, \dots, \theta_{N,0}$ such that we have $\theta_{1,0} + \dots + \theta_{N,0} = 0$. Henceforth, we will forgo the explicit mention of $\theta_{1,0}, \dots, \theta_{N,0}$ in all cases involving the matrix root $(\cdot)^{\frac{1}{m}}, m \in \mathbb{N} \setminus \{0\}$.

Next, we must fix a computable sequence of computable unitary matrices that provides $\frac{1}{2^M}$ -coverings of Ξ_μ for all $M \in \mathbb{N}$. The proof of Proposition 2, which establishes the Banach–Mazur computability of η_Ξ , will build upon such a sequence. From before, recall that the behavior of $\eta_\Xi(\underline{u}, n)$ for $n \rightarrow \infty$ is known for some specific gate families $\underline{u} \in \Xi_{\mu,N}^\diamond, N \in \mathbb{N}$. The reader may find explicit examples in, e.g., [5]. Particularly, we can find a computable sequence $(U_m)_{m \in \mathbb{N}} \subset \Xi_\mu$ that enumerates $G(\underline{u})$ (in the sense of Section IV) and a (total) recursive function $\check{g} : \mathbb{N} \rightarrow \mathbb{N}$ such that for all $n, M \in \mathbb{N}, n \geq \check{g}(M)$, we have

$$\begin{aligned} \sup_{U \in \Xi_\mu} \inf_{0 \leq m < (N+1)^n - 1} \|U - \check{U}_m\| \dots \\ \dots = \eta_\Xi(\underline{u}, n) \leq 2^{-M}. \end{aligned} \quad (22)$$

As required above, we fix a suitable choice of $\underline{u}, (\check{U}_m)_{m \in \mathbb{N}}$, and $\check{g} : \mathbb{N} \rightarrow \mathbb{N}$ for the rest of this article. Informally speaking, the sequence $(\eta_\Xi(\underline{u}, n))_{n \in \mathbb{N}}$ will act as a “convergence reference” in computing η_Ξ for arbitrary gate families. For ease of notation, we will not distinguish between whether we have $\underline{u} \in U_{\mu,N}^\diamond$ or $\underline{u} \in \text{SU}_{\mu,N}^\diamond$, as one may always infer this detail from the relevant context.

Finally, we introduce the joining of gate families. Let $M, L \in \mathbb{N}$ be arbitrary and consider $\underline{u} \in \mathfrak{E}_{\mu, M}$, $\underline{v} \in \mathfrak{E}_{\mu, L}$. We define $\underline{u} \circ \underline{v} \in \mathfrak{E}_{\mu, M+L}$ —the joint gate family of \underline{u} and \underline{v} —through

$$[\underline{u} \circ \underline{v}](n) := \begin{cases} \underline{u}(n), & \text{if } n \in \{0, \dots, M\} \\ \underline{v}(n - M), & \text{if } n \in \{M + 1, \dots, M + L\}. \end{cases}$$

Note that joining gate families is Banach–Mazur computable: Given computable sequences $(\underline{u}_m)_{m \in \mathbb{N}} \subset \mathfrak{E}_{\mu, M}$ and $(\underline{v}_l)_{l \in \mathbb{N}} \subset \mathfrak{E}_{\mu, L}$, the sequence $(\underline{u}_m \circ \underline{v}_l)_{m, l \in \mathbb{N}} \subset \mathfrak{E}_{\mu, M+L}$ is computable as well.

APPENDIX B FORMAL PROOFS

In the following, we provide a proof for each of this article’s propositions, theorems, and corollaries. We divide the remainder of Appendix B into subsections, one for each section in the article’s main body that establishes a formal result.

A. SECTION V: PROPOSITION 1, THEOREM 1, AND COROLLARY 1

From Section IV, recall that given any gate family $\underline{u} \in \mathfrak{E}_{\mu, N}$, $N \in \mathbb{N}$, there exists a computable sequence $(\mathbf{U}_n)_{n \in \mathbb{N}} \subset \mathfrak{E}_\mu$ that enumerates the monoid $G(\underline{u})$. We start by formally establishing a more general variant of this claim.

Lemma 6: For $M \in \mathbb{N}$ arbitrary, let $(\underline{u}_n)_{n \in \mathbb{N}} \subset \mathfrak{E}_{\mu, M}$ be any computable sequence. There exists a computable sequence $(\mathbf{U}_{n, m})_{n, m \in \mathbb{N}} \subset \mathfrak{E}_\mu$ such that for all $n, l \in \mathbb{N}$, we have

$$\{\mathbf{U}_{n, 0}, \dots, \mathbf{U}_{n, \lfloor (M; l) \rfloor - 1}\} = \{\underline{u}_n(m) : m \in \langle M; l \rangle\}. \quad (23)$$

Proof: We consider $M \in \mathbb{N} \setminus \{0\}$, as the statement is trivial otherwise. Then, the construction of $(\mathbf{U}_{n, m})_{n, m \in \mathbb{N}}$ is conceptually straightforward: given $m \in \mathbb{N}$, compute its base- $(M + 1)$ expansion; then, successively apply the iterative computation of matrix multiplications (see Summary 4, Part 3) to the matrices corresponding to the expansion’s digits.

For notational convenience, consider the μ -recursive functions $\tilde{g}_1, \tilde{g}_2 : \mathbb{N} \rightarrow \mathbb{N}$ defined through

$$m \mapsto \tilde{g}_1(m) := \text{mod}(m, M + 1)$$

$$m \mapsto \tilde{g}_2(m) := \frac{m - \tilde{g}_1(m)}{M + 1}.$$

Let $(\mathbf{V}_{n, m})_{n, m \in \mathbb{N}}$, $(\mathbf{W}_n)_{n \in \mathbb{N}} \subset \mathfrak{E}_\mu$ be computable sequences and $g, g_1 : \mathbb{N} \times \mathbb{N} \rightrightarrows \mathbb{N}$, $g_2 : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$, μ -recursive functions that satisfy

$$(n, l) \in D(g) \cap D(g_1), \mathbf{V}_{g(n, l)} = \underline{u}_n(l), \dots \\ \dots \mathbf{W}_{g_1(n, l)} = \mathbf{V}_{g(n, l)}, \text{ and } \mathbf{W}_{g_2(n, m)} = \mathbf{W}_n \mathbf{W}_m$$

for all $(n, m, l) \in \mathbb{N} \times \mathbb{N} \times [M]$ (see Summary 4, Part 3). We define $h : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ recursively through $h(n, 0) := g_1(n, 0)$ and, for $m \geq 1$

$$h(n, m) := g_2(g_1(n, \tilde{g}_1(m)), h(n, \tilde{g}_2(m))).$$

Accordingly, h is a total μ -recursive function.

Next, let $n \in \mathbb{N}$ and $l \in \mathbb{N} \setminus \{0\}$ be arbitrary. Furthermore, consider $m, m_1, \dots, m_l, k_1, \dots, k_l \in \mathbb{N}$, such that we have

$m = k_1, m_l = k_l \leq M$, and, for $1 \leq j < l$

$$m_j = \tilde{g}_1(k_j) \quad k_{j+1} = \tilde{g}_2(k_j).$$

In other words, the tuple (m_1, \dots, m_l) corresponds to the base- $(M + 1)$ expansion of the number m . Then, we have $h(n, k_j) = g_2(g_1(n, m_j), h(n, k_{j+1}))$ and thus

$$\mathbf{W}_{h(n, m)} = \underline{u}_n(m_1) \cdots \underline{u}_n(m_j) \mathbf{W}_{h(n, k_{j+1})} \\ = \underline{u}_n(m_1) \cdots \underline{u}_n(m_l) \underline{u}_n(0) \\ = \underline{u}_n(m_1) \cdots \underline{u}_n(m_l)$$

for $1 \leq j < l$. Defining $(\mathbf{U}_{n, m})_{n, m \in \mathbb{N}} : \mathbf{U}_{n, m} = \mathbf{W}_{h(n, m)}$, we obtain a computable sequence in \mathfrak{E}_μ that, as we chose n and l arbitrarily, satisfies (23) for all $n \in \mathbb{N}$, $l \in \mathbb{N} \setminus \{0\}$. \square

We are equipped to prove Proposition 1. To this end, formally establishing our initial claim—the enumerability of $G(\underline{u})$ for all $\underline{u} \in \mathfrak{E}_{\mu, N}$, $N \in \mathbb{N}$, in the respective sense—would, in principle, have been sufficient. However, the claim’s more general variant stated in Lemma 6 will be necessary later on to prove Proposition 2.

Proof of Proposition 1: Let $(\mathbf{V}_m)_{m \in \mathbb{N}} \subset \mathfrak{E}_\mu$ be a computable sequence that enumerates the monoid $G(\underline{u})$ in the sense of Lemma 6. We define

$$(x_{n, m})_{n, m \in \mathbb{N}} : x_{n, m} = \min_{0 \leq l \leq \lfloor (L; m) \rfloor - 1} \|\mathbf{U}_n - \mathbf{V}_l\|.$$

As follows from Summary 4, Part 1, and Summary 3, Part 6, $(x_{n, m})_{n, m \in \mathbb{N}}$ is a computable sequence of real numbers. By construction, we have $x_{n, m} \geq 0$ for all $n, m \in \mathbb{N}$, with $x_{n, m} = 0$ if and only if $\Omega(\mathbf{U}_n | \underline{u}) \leq m$.

Let $(q_{n, m, l})_{n, m, l \in \mathbb{N}}$ be a computable sequence of rational numbers that satisfies

$$0 \leq q_{n, m, l} \leq q_{n, m, l+1} \leq x_{n, m} \leq q_{n, m, l} + 2^{-l}$$

for all $n, m, l \in \mathbb{N}$: since $(x_{n, m})_{n, m \in \mathbb{N}}$ is a computable sequence of nonnegative real numbers, an appropriate sequence $(q_{n, m, l})_{n, m, l \in \mathbb{N}}$ always exists (cf., Section III). Then, $(q_{n, m, l})_{n, m, l \in \mathbb{N}}$ satisfies the following for all $n, m \in \mathbb{N}$.

- 1) If we have $m < \Omega(\mathbf{U}_n | \underline{u})$, there exists $l \in \mathbb{N}$ such that for all $k \in \mathbb{N}$, $k \geq l$, we have $q_{n, m, k} > 0$.
- 2) If, on the other hand, we have $m \geq \Omega(\mathbf{U}_n | \underline{u})$, we also have $q_{n, m, l} = 0$ for all $l \in \mathbb{N}$.

We define $g : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ recursively through $g(n, 0) = 0$ and

$$g(n, m + 1) = \begin{cases} g(n, m), & \text{if } q_{n, g(n, m), m} = 0 \\ g(n, m) + 1, & \text{if } q_{n, g(n, m), m} > 0. \end{cases}$$

Observing that g is a (total) μ -recursive function by construction, we conclude the proof. \square

Proving Theorem 1 requires an additional auxiliary result—Lemma 7—which we subsequently establish.

Lemma 7: Consider $\mathbf{U}_0 \in \mathfrak{E}_\mu$, $\underline{u} \in \mathfrak{E}_{\mu, L}^\circ$, $L \in \mathbb{N}$, and define

$$(\mathbf{U}_n)_{n \in \mathbb{N}} : \mathbf{U}_n = (\mathbf{U}_0)^{\frac{1}{(n+1)}}.$$

If there exists $n \in \mathbb{N}$ such that we have $U_n \in G(\underline{u})$, we also have $U_0 \in G(\underline{u})$.

Proof: Per definition, $G(\underline{u})$ is closed with respect to matrix multiplication. Thus, if we have $U_n \in G(\underline{u})$ for any $n \in \mathbb{N}$, we also have $U_0 = (U_n)^{n+1} \in G(\underline{u})$. \square

We now prove Theorem 1 by applying Lemmas 4 and 5. In addition, our proof of Theorem 1 makes use of Proposition 1.

Proof of Theorem 1: Choose any $U_{0,*} \in \neg G(\underline{u})$ and define the sequence

$$(U_{n,*})_{n \in \mathbb{N}} : U_{n,*} = (U_{0,*})^{\frac{1}{(n+1)}}.$$

Then, $(U_{n,*})_{n \in \mathbb{N}}$ is a computable sequence, and, as follows from Lemma 7 by contradiction, we have $U_{n,*} \in \neg G(\underline{u})$ for all $n \in \mathbb{N}$. Furthermore, $(U_{n,*})_{n \in \mathbb{N}}$ converges effectively toward $\text{Id} \in G(\underline{u})$ (cf., Section IV). Accordingly, upon proving Conditions (a)–(c) of Lemma 5 fulfilled for

$$F(\cdot) \equiv \Omega(\cdot|\underline{u}) \quad \mathcal{A}_{1,\mu} \equiv G(\underline{u}) \quad \mathcal{A}_{2,\mu} \equiv \neg G(\underline{u})$$

there exists a computable sequence $(U_n)_{n \in \mathbb{N}} \subset G(\underline{u})$ with effective limit $\text{Id} \in G(\underline{u})$ such that $\Omega(\cdot|\underline{u})$ isolates $(U_n)_{n \in \mathbb{N}}$ from Id . The claim then follows by subsequent application of Lemma 4. Observe the following to conclude the proof.

Condition (a). Let $(V_n)_{n \in \mathbb{N}} \subset G(\underline{u})$ be any sequence that enumerates $G(\underline{u})$ in the sense of Lemma 6. Then, since we have $\underline{u} \in \Xi_{\mu,N}^\diamond$ by assumption, $\{V_n\}_{n \in \mathbb{N}}$ must be dense (in the usual sense of topology) in Ξ_μ . Accordingly, by Summary 4, Part 4, we have

$$\Xi_\mu = \text{efc}\{V_n\}_{n \in \mathbb{N}} = \text{efc}G(\underline{u}).$$

Thus, $G(\underline{u})$ is effectively dense in $\Xi_\mu \setminus G(\underline{u}) = \neg G(\underline{u})$.

Condition (b). Let $(V_n)_{n \in \mathbb{N}} \subset G(\underline{u})$ be any computable sequence. By Proposition 1, there exists a (total) μ -recursive function $g : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ such that for all $n, m \in \mathbb{N}$, we have

$$g(n, m) \leq g(n, m+1) \leq \sup_{l \in \mathbb{N}} g(n, l) = \Omega(V_n|\underline{u}).$$

Defining $(q_{n,m})_{n,m \in \mathbb{N}} : q_{n,m} = g(n, m)$ then yields the required computable sequence.

Condition (c). Let $(V_n)_{n \in \mathbb{N}} \subset G(\underline{u})$ be any computable sequence with effective limit $V_* \in \neg G(\underline{u})$. We prove the condition's fulfillment by contradiction. To this end, consider $q \in \mathbb{Q}$, $q > 0$, to be arbitrary, and assume we have $\Omega(V_n|\underline{u}) \leq q$ for all $n \in \mathbb{N}$. In other words, we have

$$V_n \in \{V \in G(\underline{u}) : \Omega(V|\underline{u}) \leq q\} \quad (24)$$

for all $n \in \mathbb{N}$. But the right-hand side of (24) is a finite set. Thus, we must also have

$$(\lim_{n \rightarrow \infty} V_n) \in \{V \in G(\underline{u}) : \Omega(V|\underline{u}) \leq q\} \subset G(\underline{u}).$$

But we chose $(V_n)_{n \in \mathbb{N}}$ to satisfy $\lim_{n \rightarrow \infty} V_n = V_* \in \neg G(\underline{u})$, which is absurd. \square

It remains to prove Corollary 1. Observe that of its two claims, only Claim 2 is a consequence of Theorem 1. However, the proof of Claim 1 is straightforward, much in contrast to a direct proof of Claim 2. One may thus consider the ‘‘major’’ part of Corollary 1 a consequence of Theorem 1.

Proof of Corollary 1: We prove Claims 1 and 2 one at a time. For the sake of a clear structure, we label the relevant paragraphs accordingly.

Part 1. Conceptually, the proof of Claim 1 is analogous to the proof of Lemma 6. Particularly, we have the following.

- 1) There exists a computable sequence $(V_m)_{m \in \mathbb{N}} \subset G(\underline{u})$ such that for all $M \in \mathbb{N} \setminus \{0\}$, $m \in \mathbb{N}$, $m_1, \dots, m_M \in [N]$, whenever we have

$$m = \sum_{l=1}^M m_l (N+1)^{l-1}$$

we also have $V_m = \underline{u}(m_1) \cdots \underline{u}(m_M)$ and therefore, with $\omega = (u(m_1), \dots, u(m_M))$, $V_m = \underline{u}(\omega)$.

- 2) There exists a (total) μ -recursive function $h : \mathbb{N} \rightarrow \mathbb{N}$ that satisfies

$$h \left(\sum_{l=1}^M (m_l + 1)(N+1)^{l-1} \right) = \sum_{l=1}^M m_l (N+1)^{l-1}$$

for all $M \in \mathbb{N} \setminus \{0\}$, $m_1, \dots, m_M \in [N]$.

Hence, consider any computable sequence $(\omega_n)_{n \in \mathbb{N}} \subset u^*$, and let $g : \mathbb{N} \rightarrow \mathbb{N}$ determine $(\omega_n)_{n \in \mathbb{N}}$ in the sense of Section V [cf., (5)]. As follows from the above, the sequence

$$(U_n)_{n \in \mathbb{N}} : U_n = V_{h(g(n))}$$

is computable and satisfies $U_n = \underline{u}(\omega_n)$ for all $n \in \mathbb{N}$.

Part 2. Consider any computable sequence $(U_n)_{n \in \mathbb{N}} \subset G(\underline{u})$. If there exists a computable sequence $(\omega_n)_{n \in \mathbb{N}} \subset u^*$ that satisfies $\underline{u}(\omega_n) = U_n$ for all $n \in \mathbb{N}$, there also exists a (total) μ -recursive function $h : \mathbb{N} \rightarrow \mathbb{N}$ that satisfies

$$h(n) = |\omega_n| \geq \Omega(U_n|\underline{u})$$

for all $n \in \mathbb{N}$ (cf., Section V). Claim 2 then follows from Theorem 1 by contradiction. \square

B. SECTION VI: PROPOSITION 2, THEOREM 2, AND COROLLARY 2

We start by proving Proposition 2, which does not require any auxiliaries other than those we already established.

Proof of Proposition 2: Choose $(\underline{u}_n)_{n \in \mathbb{N}} \subset \Xi_{\mu,M}$ arbitrarily (but computable). We then need to show that $(\eta_{\Xi}(\underline{u}_n, m))_{n,m \in \mathbb{N}}$ is a computable sequence of real numbers. To this end, let $(U_{n,m})_{n,m \in \mathbb{N}} \subset \Xi_\mu$ satisfy Lemma 6 with respect to $(\underline{u}_n)_{n \in \mathbb{N}}$ in the following.

From Appendix A2, recall the triple $(\check{\underline{u}}, (\check{U}_m)_{m \in \mathbb{N}}, \check{g})$ and consider $n, m, l, L \in \mathbb{N}$, $l \geq \check{g}(L)$, arbitrary but fixed. We have

$$\begin{aligned} & \sup_{U \in \Xi_\mu} \inf_{0 \leq k < (M+1)^{m-1}} \|U - U_{n,k}\| \\ & \geq \sup_{0 \leq j < (|\check{\underline{u}}|+1)^{l-1}} \inf_{0 \leq k < (M+1)^{m-1}} \|\check{U}_j - U_{n,k}\| \\ & = \max_{0 \leq j < (|\check{\underline{u}}|+1)^{l-1}} \min_{0 \leq k < (M+1)^{m-1}} \|\check{U}_j - U_{n,k}\| \\ & =: \tilde{\eta}_{\Xi}(\underline{u}_n, m, l) \end{aligned} \quad (25)$$

since $\{\check{U}_j : 0 \leq j < (\|\check{\underline{u}}\| + 1)^l - 1\}$ is a subset of Ξ_{μ} . Next, consider $\underline{U} \in \Xi_{\mu}$ arbitrary but fixed and define

$$\check{V}_l(\underline{U}) := \arg \min \left\{ \|\underline{U} - \underline{V}\| : \dots \right. \\ \left. \dots \underline{V} \in \left\{ \check{U}_j : 0 \leq j < (\|\check{\underline{u}}\| + 1)^l - 1 \right\} \right\}.$$

Then, as follows from applying the triangular inequality and (22) and with $\tilde{\eta}_{\Xi}(\underline{u}_n, m, l)$ as defined above, we have

$$\inf_{0 \leq k < (M+1)^{m-1}} \|\underline{U} - \underline{U}_{n,k}\| \\ \leq \inf_{0 \leq k < (M+1)^{m-1}} \left(\|\underline{U} - \check{V}_l(\underline{U})\| + \|\check{V}_l(\underline{U}) - \underline{U}_{n,k}\| \right) \\ \leq \inf_{0 \leq k < (M+1)^{m-1}} \|\check{V}_l(\underline{U}) - \underline{U}_{n,k}\| + 2^{-L} \\ \leq \sup_{0 \leq j < (\|\check{\underline{u}}\| + 1)^l - 1} \inf_{0 \leq k < (M+1)^{m-1}} \|\check{U}_j - \underline{U}_{n,k}\| + 2^{-L} \\ = \tilde{\eta}_{\Xi}(\underline{u}_n, m, l) + 2^{-L}. \quad (26)$$

Since we chose \underline{U} arbitrarily and the right-hand side of (26) does not depend on \underline{U} , we obtain

$$\tilde{\eta}_{\Xi}(\underline{u}_n, m, l) + 2^{-L} \geq \eta_{\Xi}(\underline{u}_n, m) \geq \tilde{\eta}_{\Xi}(\underline{u}_n, m, l) \quad (27)$$

from combining (2) with (26) and (2) with (25). In summary, we thus have $|\eta_{\Xi}(\underline{u}_n, m) - \tilde{\eta}_{\Xi}(\underline{u}_n, m, l)| \leq \frac{1}{2^L}$.

From Summary 4, Part 1, and Summary 3, Part 6, we conclude that $(\tilde{\eta}_{\Xi}(\underline{u}_n, m, l))_{n,m,l \in \mathbb{N}}$ is a computable sequence of real numbers. Further, as we had chosen the indices involved in (27) arbitrarily [up to satisfying $l \geq \check{g}(L)$], we conclude that $(\tilde{\eta}_{\Xi}(\underline{u}_n, m, l))_{n,m,l \in \mathbb{N}}$ converges effectively toward $(\eta_{\Xi}(\underline{u}_n, m))_{n,m \in \mathbb{N}}$ for $l \rightarrow \infty$. Thus, as follows from Summary 3, Part 3, $(\eta_{\Xi}(\underline{u}_n, m))_{n,m \in \mathbb{N}}$ must also be a computable sequence of real numbers. Finally, as we had initially chosen $(\underline{u}_n)_{n \in \mathbb{N}} \subset \Xi_{\mu,M}$ arbitrarily as well, the claim follows. \square

Proving Theorem 2 requires two additional auxiliary results—Lemmas 8 and 9—which we subsequently establish. For $M \in \mathbb{N}$, we henceforth denote

$$\underline{\text{Id}}_M : [M] \rightarrow \Xi_{\mu}, m \mapsto \underline{\text{Id}}_M(m) := \text{Id}$$

i.e., the gate family $\underline{\text{Id}}_M \in \Xi_{\mu,M}$ consists of the $(M+1)$ -fold repetition of the identity matrix. We refer to $\underline{\text{Id}}_M$ as the trivial gate family in $\Xi_{\mu,M}$. Moreover, for $m \in \mathbb{N} \setminus \{0\}$, recall the definition of the m th matrix root $(\cdot)^{\frac{1}{m}}$ from Appendix A2 and let $\underline{u} \in \Xi_{\mu,M}$ be arbitrary. We denote the component wise application of $(\cdot)^{\frac{1}{m}}$ to \underline{u} by $\underline{u}^{\frac{1}{m}}$. Finally, given a computable growth function $f : \mathbb{N} \rightarrow \mathbb{Q}$, we denote the set $\Xi_{\mu,M} \setminus \mathcal{O}_{\Xi,M}(f)$ by $\neg\mathcal{O}_{\Xi,M}(f)$.

Lemma 8: For $M \in \mathbb{N}$, let $f : \mathbb{N} \rightarrow \mathbb{Q}$ be any computable growth function such that $\mathcal{O}_{\Xi,M}(f)$ is non-empty. We have the following.

- 1) For all $N \in \mathbb{N}$, the set $\mathcal{O}_{\Xi,N+M}(f)$ is effectively dense in the set $\{\underline{u} \circ \underline{\text{Id}}_M : \underline{u} \in \neg\mathcal{O}_{\Xi,N}(f)\} \subset \neg\mathcal{O}_{\Xi,N+M}(f)$.

Proof: Let $(\underline{u}_{n,*})_{n \in \mathbb{N}} \subset \{\underline{u} \circ \underline{\text{Id}}_M : \underline{u} \in \neg\mathcal{O}_{\Xi,N}(f)\}$ be arbitrary (but computable). Then, we must prove the existence

of a computable sequence $(\underline{u}_{n,m})_{n,m \in \mathbb{N}} \subset \mathcal{O}_{\Xi,N+M}(f)$ that converges effectively to $(\underline{u}_{n,*})_{n \in \mathbb{N}}$ for $m \rightarrow \infty$. To this end, consider $\underline{v} \in \mathcal{O}_{\Xi,M}(f)$ and define $(\underline{v}_m)_{m \in \mathbb{N}} : \underline{v}_m = \underline{v}^{\frac{1}{(m+1)}}$. Then, observe the following.

- 1) We have $\underline{v}_m \in \mathcal{O}_{\Xi,M}(f)$ for all $m \in \mathbb{N}$. Hence, we also have $\underline{u} \circ \underline{v}_m \in \mathcal{O}_{\Xi,N+M}(f)$ for all $m \in \mathbb{N}$, $\underline{u} \in \neg\mathcal{O}_{\Xi,N}(f)$.
- 2) For $m \rightarrow \infty$, the sequence $(\underline{v}_m)_{m \in \mathbb{N}}$ converges effectively toward $\underline{\text{Id}}_M$. Hence, for $m \rightarrow \infty$ and all $\underline{u} \in \neg\mathcal{O}_{\Xi,N}(f)$, the sequence $(\underline{u} \circ \underline{v}_m)_{m \in \mathbb{N}}$ converges effectively to $\underline{u} \circ \underline{\text{Id}}_M$.

Accordingly, consider $(\underline{u}_n)_{n \in \mathbb{N}} \subset \neg\mathcal{O}_{\Xi,N}(f)$ such that for all $n \in \mathbb{N}$, we have $\underline{u}_n \circ \underline{\text{Id}}_M = \underline{u}_{n,*}$, and define

$$(\underline{u}_{n,m})_{n,m \in \mathbb{N}} : \underline{u}_{n,m} = \underline{u}_n \circ \underline{v}_m.$$

Then, $(\underline{u}_{n,m})_{n,m \in \mathbb{N}} \subset \mathcal{O}_{\Xi,N+M}(f)$ is computable and converges effectively to $(\underline{u}_{n,*})_{n \in \mathbb{N}}$ for $m \rightarrow \infty$. \square

To prove the subsequent Lemma 9, we require a special computable sequence of diagonal unitary matrices. For $M \in \mathbb{N}$, let $g_0, g_1, \dots, g_M, h : \mathbb{N} \rightarrow \mathbb{N}$ be μ -recursive functions such that for all $(m_0, m_1, \dots, m_M, m_{M+1}) \in \mathbb{N}^{M+2}$, there exists $l \in \mathbb{N}$ that satisfies

$$m_0 = g_0(l), m_1 = g_1(l), \dots \\ \dots, m_M = g_M(l), m_{M+1} = h(l).$$

In other words, we require that the functions g_0, g_1, \dots, g_M, h enumerate the set \mathbb{N}^{M+2} . A suitable family g_0, g_1, \dots, g_M, h may, for example, be obtained from nesting inverse pairing functions [24, Sect. 1.4, p. 12]. Next, recall the computational basis vectors $|e_1\rangle, \dots, |e_M\rangle$. Consider $\underline{v} \in \mathcal{U}_{\mu,N}^{\diamond}$, $N \in \mathbb{N}$, and let $(\underline{V}_l)_{l \in \mathbb{N}}$ enumerate $\mathcal{G}(\underline{v})$ in the sense of Lemma 6. We define

$$(\check{\mathcal{Q}}_l)_{l \in \mathbb{N}} : \check{\mathcal{Q}}_l = \dots \\ \dots \sum_{k=1}^M \exp\left(\frac{j\pi g_k(l)}{g_0(l) + 1}\right) \underline{V}_{h(l)} |e_k\rangle \langle e_k| \underline{V}_{h(l)}^{\dagger}.$$

Then, $(\check{\mathcal{Q}}_l)_{l \in \mathbb{N}} \subset \mathcal{U}_{\mu}(M)$ is a computable sequence. Moreover, $\{\check{\mathcal{Q}}_l\}_{l \in \mathbb{N}}$ is (topologically) dense in $\mathcal{U}_{\mu}(M)$.

Lemma 9: Consider $N \in \mathbb{N}$ arbitrary. The set $\neg\mathcal{U}_{\mu,N}^{\diamond}$ is effectively dense in the set $\mathcal{U}_{\mu,N}^{\diamond}$.

Proof: In the following, let $(\underline{u}_{n,*})_{n \in \mathbb{N}} \subset \mathcal{U}_{\mu,N}^{\diamond}$ be arbitrary. Analogously to the Proof of Lemma 8, we demonstrate the existence of a computable sequence $(\underline{u}_{n,m})_{n,m \in \mathbb{N}} \subset \neg\mathcal{U}_{\mu,N}^{\diamond}$ that converges effectively to $(\underline{u}_{n,*})_{n \in \mathbb{N}}$ for $m \rightarrow \infty$.

Consider a computable sequence $(\underline{U}_{n,*})_{n \in \mathbb{N}} \subset \mathcal{U}_{\mu}$ and a μ -recursive function $\tilde{g} : \mathbb{N} \times \mathbb{N} \supseteq \rightarrow \mathbb{N}$ that satisfy

$$(n, k) \in D(\tilde{g}) \quad \text{and} \quad \underline{u}_{n,*}(k) = \underline{U}_{\tilde{g}(n,k),*}$$

for all $(n, k) \in \mathbb{N} \times [N]$. Observe that we may choose \tilde{g} such that we have $D(\tilde{g}) = \mathbb{N} \times \mathbb{N}$, in which case

$$(\underline{y}_{n,m,l,k})_{n,m,l,k \in \mathbb{N}} : y_{n,m,l,k} = 2^{-m} - \|\underline{U}_{\tilde{g}(n,k),*} - \check{\mathcal{Q}}_l\|$$

defines a computable sequence of computable real numbers. Accordingly, due to Lemma 1 and $\{\check{Q}_l\}_{l \in \mathbb{N}}$ being topologically dense in U_μ , there exists a total μ -recursive function $h : \mathbb{N}^3 \rightarrow \mathbb{N}$ such that for all $n, m, k \in \mathbb{N}$, we have

$$\|U_{\check{g}(n,k),*} - \check{Q}_{h(n,m,k)}\| < 2^{-m}.$$

Moreover, by construction of $(\check{Q}_l)_{l \in \mathbb{N}}$, we can find $K \in \mathbb{N}$ that satisfies $\check{Q}_K = \text{Id}$. Consider $g : \mathbb{N}^3 \rightarrow \mathbb{N}$,

$$(n, m, k) \mapsto g(n, m, k) := \begin{cases} K, & \text{if } k = 0 \\ h(n, m, k), & \text{otherwise} \end{cases}$$

and define the sequence $(\underline{u}_{n,m})_{n,m \in \mathbb{N}} \subset U_{\mu,N}$ such that for all $(n, m, k) \in \mathbb{N} \times \mathbb{N} \times [N]$, we have

$$\underline{u}_{n,m}(k) = \check{Q}_{g(n,m,k)}.$$

Then, $(\underline{u}_{n,m})_{n,m \in \mathbb{N}}$ is a computable sequence of gate families and converges effectively to $(\underline{u}_{n,*})_{n \in \mathbb{N}}$ for $m \rightarrow \infty$.

It remains to show that we have $\underline{u}_{n,m} \in \neg U_{\mu,N}^\circ$ for all $n, m \in \mathbb{N}$. To this end, consider $n, m \in \mathbb{N}$ arbitrary. By construction, there exist numbers $q_1, \dots, q_N \in \mathbb{Q}$ that satisfy

$$\begin{aligned} \det \underline{u}_{n,m}(1) &= \exp(j2\pi q_1), \dots \\ \dots, \det \underline{u}_{n,m}(N) &= \exp(j2\pi q_N). \end{aligned}$$

Thus, denoting the least common denominator of q_1, \dots, q_N by $q \in \mathbb{Q}$, we have

$$\{\det \mathbf{W} : \mathbf{W} \in G(\underline{u}_{n,m})\} \subseteq \{\exp(j2\pi kq) : k \in \mathbb{N}\}. \quad (28)$$

The right-hand side of (28) is a finite set of roots of unity. Thus, there exists $\mathbf{W} \in U_\mu$ satisfying

$$\lim_{l \rightarrow \infty} |\det \mathbf{W} - \det \mathbf{W}_l| > 0$$

for all sequences $(\mathbf{W}_l)_{l \in \mathbb{N}} \subset G(\underline{u}_{n,m})$. Since $\det : U \rightarrow \mathbb{C}$ is a continuous function, we also have $\lim_{l \rightarrow \infty} \|\mathbf{W} - \mathbf{W}_l\| > 0$ for all such $\mathbf{W} \in U_\mu$ and all $(\mathbf{W}_l)_{l \in \mathbb{N}} \subset G(\underline{u}_{n,m})$. Accordingly, we must have $\underline{u}_{n,m} \in \neg U_{\mu,N}^\circ$. Since we chose $n, m \in \mathbb{N}$ arbitrarily, the claim follows. \square

Finally, we are equipped to establish a formal proof of Section VI's main contribution.

Proof of Theorem 2: We prove Claims 1 and 2 one at a time. For the sake of a clear structure, we label the relevant paragraphs accordingly.

Part 1. Let $\underline{u} \in \Xi_{\mu,M}^\circ$ be arbitrary and observe that there always exists a suitable computable function $f : \mathbb{N} \rightarrow \mathbb{Q}$ such that we have $\underline{u} \in \mathcal{O}_{\Xi,M}(f)$. Specifically, define

$$(y_{n,m})_{n,m \in \mathbb{N}} : y_{n,m} = \eta_{\Xi}(\underline{u}, m) - 2^{-n}.$$

Then, $(y_{n,m})_{n,m \in \mathbb{N}}$ is a computable sequence of computable numbers. As follows from Lemma 1, there exists a μ -recursive function $g : D(g) \rightarrow \mathbb{N}$ such that for all $n \in D(g)$, we have $y_{n,g(n)} < 0$. Since \underline{u} is universal and by construction of $(y_{n,m})_{n,m \in \mathbb{N}}$, we must have $D(g) = \mathbb{N}$. Therefore, defining $f : \mathbb{N} \rightarrow \mathbb{Q}$ through

$$f(N) := \max\{g(0), \dots, g(N)\}$$

yields a computable growth function such that we have $\underline{u} \in \mathcal{O}_{\Xi,M}(f)$. By Lemma 8, $\mathcal{O}_{\Xi,M}(f)$ is effectively dense in $\{\text{Id}_M\}$. Consequently, by Lemma 3, we have $\{\text{Id}_M\} \xrightarrow{\mu} \mathcal{O}_{\Xi,M}(f)$. Since we have $\mathcal{O}_{\Xi,M}(f) \subseteq \Xi_{\mu,M}^\circ$ and $\{\text{Id}_M\} \subseteq \neg \Xi_{\mu,M}^\circ$, the claim follows.

Part 2. By Lemma 9, $\neg U_{\mu,N}^\circ$ is effectively dense in $U_{\mu,N}^\circ$. Thus, by application of Lemma 3, the claim follows. \square

It remains to prove Corollary 2. By means of Lemma 1, it follows directly from Theorem 2.

Proof of Corollary 2: The claim follows from Theorem 2 by contradiction. To this end, consider $(\underline{u}_n)_{n \in \mathbb{N}} \subset \Xi_{\mu,M}$ arbitrary and assume $\eta_{\Xi}^\circ : \Xi_{\mu,M} \rightarrow \mathbb{R}$ Banach–Mazur computable in the following.

Define $(y_{n,m})_{n,m \in \mathbb{N}} : y_{n,m} := \eta_{\Xi}^\circ(\underline{u}_n)$.⁴ By assumption, $(y_{n,m})_{n,m \in \mathbb{N}} \subset \mathbb{R}_\mu$ is a computable sequence. Thus, as follows from Lemma 1, there exists a μ -recursive function $g_{>} : \mathbb{N} \supseteq \rightarrow \mathbb{N}$ such that we have

$$D(g_{>}) = \{n \in \mathbb{N} : \exists m \in \mathbb{N} \text{ s.t. } y_{n,m} > 0\}$$

for the domain of $g_{>}$. By construction, $(y_{n,m})_{n,m \in \mathbb{N}}$ and $g_{>}$ satisfy the following for all $n \in \mathbb{N}$.

- 1) If we have $\underline{u}_n \in \Xi_{\mu,M}^\circ$, we also have $y_{n,m} = 0$ for all $m \in \mathbb{N}$. Accordingly, we must have $n \in \neg D(g_{>})$.
- 2) If we have $\underline{u}_n \in \neg \Xi_{\mu,M}^\circ$, we also have $y_{n,m} > 0$ for all $m \in \mathbb{N}$. Accordingly, we must have $n \in D(g_{>})$.

We obtain $D(g_{>}) = \{n \in \mathbb{N} : \underline{u}_n \in \neg \Xi_{\mu,M}^\circ\}$ in summary, implying that $\{n \in \mathbb{N} : \underline{u}_n \in \neg \Xi_{\mu,M}^\circ\}$ is recursively enumerable. Thus, as we chose $(\underline{u}_n)_{n \in \mathbb{N}}$ arbitrarily, we must have $\neg \Xi_{\mu,M}^\circ \xrightarrow{\mu} \Xi_{\mu,M}^\circ$. But we have $\neg \Xi_{\mu,M}^\circ \not\xrightarrow{\mu} \Xi_{\mu,M}^\circ$ by Theorem 2, which is absurd. \square

C. SECTION VII: PROPOSITION 3 AND THEOREM 3

We start by proving Proposition 3. To this end, we require a special computable sequence of rational numbers. Let $g, h : \mathbb{N} \rightarrow \mathbb{N}$ be μ -recursive functions such that for all $(n, l) \in \mathbb{N} \times \mathbb{N}$, there exists $m \in \mathbb{N}$ that satisfies $g(m) = n$ and $h(m) = l$; see [24, Sect. 1.4, p. 12] for an explicit example of such inverse pairing functions. We define

$$(\check{\tau}_m)_{m \in \mathbb{N}} : \check{\tau}_m = \frac{g(m) + 1}{h(m) + 1}.$$

Accordingly, $(\check{\tau}_m)_{m \in \mathbb{N}} \subset \mathbb{Q}$ is a computable sequence of rational numbers and, for all $p \in \mathbb{Q}^+ := \{q \in \mathbb{Q} : q > 0\}$, there exists $m \in \mathbb{N}$ such that we have $p = \check{\tau}_m$. In other words, the sequence $(\check{\tau}_m)_{m \in \mathbb{N}}$ enumerates the set \mathbb{Q}^+ .

Proof of Proposition 3: Let $M \in \mathbb{N}$ be arbitrary and, with $(\check{\tau}_m)_{m \in \mathbb{N}} \subset \mathbb{Q}$ as above, define

$$\begin{aligned} (y_{n,m,l})_{n,m,l \in \mathbb{N}} : y_{n,m,l} &= \dots \\ \dots \eta_{\Xi}(\underline{u}_n, \min\{k \in \mathbb{N} : k \geq \check{\tau}_m f(l + M)\}) &- 2^{l+M}. \end{aligned}$$

⁴ Observe that $(y_{n,m})_{n,m \in \mathbb{N}}$ is constant in m . Adding a second index is a mere formality to provide consistency with Lemma 1.

Then, $(y_{n,m,l})_{n,m,l \in \mathbb{N}}$ is a computable sequence of real numbers, as follows from f being a computable function, the Banach–Mazur computability of η_{Ξ} , and Summary 3, Part 4. Then, as follows from Lemma 1, there exists a μ -recursive function $g_{>} : D(g_{>}) \rightarrow \mathbb{N}$ satisfying

$$D(g_{>}) = \{(n, m) \in \mathbb{N} \times \mathbb{N} : \exists l \in \mathbb{N}. \text{s.t. } y_{n,m,l} > 0\}.$$

Let $n \in \mathbb{N}$ be arbitrary. By construction of $(y_{n,m,l})_{n,m,l \in \mathbb{N}}$ and $g_{>}$, we have the following for all $m \in \mathbb{N}$.

- 1) Assume $\check{r}_m < \Phi_{f,M}(\underline{u}_n)$. Then, there exists $l \in \mathbb{N}$ such that we have $y_{n,m,l} > 0$, implying $(n, m) \in D(g_{>})$.
- 2) Assume $\check{r}_m > \Phi_{f,M}(\underline{u}_n)$. Then, we have $y_{n,m,l} < 0$ for all $l \in \mathbb{N}$, implying $(n, m) \in \neg D(g_{>})$.

Thus, we obtain $\Phi_{f,M}(\underline{u}_n) = \sup\{\check{r}_m : (n, m) \in D(g_{>})\}$.

Define the sequence $(\mathcal{A}_n)_{n \in \mathbb{N}} : \mathcal{A}_n = \{m \in \mathbb{N} : (n, m) \in D(g_{>})\}$. Then, as follows from the above, we have

$$\Phi_{f,M}(\underline{u}_n) = \sup\{\check{r}_m : m \in \mathcal{A}_n\}$$

for all $n \in \mathbb{N}$. Since $g_{>}$ is μ -recursive, $(\mathcal{A}_n)_{n \in \mathbb{N}}$ is a computable sequence of recursively enumerable sets. Thus, there exists a total μ -recursive function $h : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ such that for all $n \in \mathbb{N}$, we have $\mathcal{A}_n = \{h(n, m) : m \in \mathbb{N}\}$. We define

$$(q_{n,m})_{n,m \in \mathbb{N}} : q_{n,m} = \max\{\check{r}_{h(n,l)} : 0 \leq l \leq m\}.$$

By construction, $(q_{n,m})_{n,m \in \mathbb{N}}$ is a computable sequence of rational numbers. Furthermore, for all $n, m \in \mathbb{N}$, we have $q_{n,m} \leq q_{n,m+1} \leq \sup_{l \in \mathbb{N}} q_{n,l} = \Phi_{f,M}(\underline{u}_n)$, which concludes the proof. \square

Proving Theorem 3 requires two additional auxiliary results—Lemmas 10 and 11—which we subsequently establish. From Appendix B2, recall that

$$-\mathcal{O}_{\text{SU},N}(f) \equiv \Xi_{\mu,N} \setminus \mathcal{O}_{\text{SU},N}(f).$$

Lemma 10: Consider $N, J \in \mathbb{N}$ and any computable growth function $f : \mathbb{N} \rightarrow \mathbb{Q}$. Assume the following.

- a) The set $\text{SU}_{\mu,J}^{\diamond} \setminus \mathcal{O}_{\text{SU},J}(f)$ is nonempty.

Then, $-\mathcal{O}_{\text{SU},N}(f)$ is effectively dense in $\mathcal{O}_{\text{SU},N}(f)$.

Proof: Consider a gate family $\underline{u} \in \text{SU}_{\mu,J}^{\diamond} \setminus \mathcal{O}_{\text{SU},J}(f)$ and choose any computable sequence $(\underline{u}_{n,*})_{n \in \mathbb{N}} \subset \mathcal{O}_{\text{SU},N}(f)$. Observe the following.

- 1) Let $\underline{v} \in \text{SU}_{\mu,N}$ satisfy $\{\underline{v}(1), \dots, \underline{v}(N)\} \subset G(\underline{u})$. We must have $\underline{v} \in \text{SU}_{\mu,N}^{\diamond} \setminus \mathcal{O}_{\text{SU},N}(f)$ in this case, since we would have $\underline{u} \in \mathcal{O}_{\text{SU},J}(f)$ otherwise.

Below, we devise a computable sequence $(\underline{u}_{n,m})_{n,m \in \mathbb{N}} \subset \text{SU}_{\mu,N}$ that satisfies $\{\underline{u}_{n,m}(1), \dots, \underline{u}_{n,m}(N)\} \subset G(\underline{u})$ for all $n, m \in \mathbb{N}$ and converges effectively to $(\underline{u}_{n,*})_{n \in \mathbb{N}}$ for $m \rightarrow \infty$. Since we chose $(\underline{u}_{n,*})_{n \in \mathbb{N}}$ arbitrarily, the lemma follows.

Consider a computable sequence $(\underline{U}_{n,*})_{n \in \mathbb{N}} \subset \text{SU}_{\mu}$ and a μ -recursive function $\tilde{g} : \mathbb{N} \times \mathbb{N} \rightrightarrows \mathbb{N}$ that satisfy

$$(n, k) \in D(\tilde{g}) \quad \underline{u}_{n,*}(k) = \underline{U}_{\tilde{g}(n,k),*}$$

for all $(n, k) \in \mathbb{N} \times [N]$, and let $(\mathbf{V}_l)_{l \in \mathbb{N}} \subset \text{SU}_{\mu}$ enumerate $G(\underline{u})$ in the sense of Lemma 6. We may choose \tilde{g} such that

we have $D(\tilde{g}) = \mathbb{N} \times \mathbb{N}$, in which case

$$(y_{n,m,l,k})_{n,m,l,k \in \mathbb{N}} : y_{n,m,l,k} = 2^{-m} - \|\underline{U}_{\tilde{g}(n,k),*} - \mathbf{V}_l\|$$

defines a computable sequence of computable real numbers. Accordingly, due to Lemma 1 and $G(\underline{u})$ being topologically dense in SU_{μ} , there exists a total μ -recursive function $h : \mathbb{N}^3 \rightarrow \mathbb{N}$ such that for all $n, m, k \in \mathbb{N}$, we have

$$\|\underline{U}_{\tilde{g}(n,k),*} - \mathbf{V}_{h(n,m,k)}\| < 2^{-m}.$$

Moreover, since we have $\text{Id} \in G(\underline{u})$, we can find $K \in \mathbb{N}$ that satisfies $\mathbf{V}_K = \text{Id}$. Finally, consider $g : \mathbb{N}^3 \rightarrow \mathbb{N}$

$$(n, m, k) \mapsto g(n, m, k) := \begin{cases} K, & \text{if } k = 0 \\ h(n, m, k), & \text{otherwise.} \end{cases}$$

We obtain the required computable sequence of gate families by defining $(\underline{u}_{n,m})_{n,m \in \mathbb{N}} \subset \text{SU}_{\mu,N}$ such that we have $\underline{u}_{n,m}(k) = \mathbf{V}_{g(n,m,k)}$ for all $(n, m, k) \in \mathbb{N} \times \mathbb{N} \times [N]$. \square

Lemma 11: For $N \in \mathbb{N}$, consider a computable growth function $f : \mathbb{N} \rightarrow \mathbb{Q}$ and any computable sequence $(\underline{u}_n)_{n \in \mathbb{N}} \subset \mathcal{O}_{\Xi,N}(f)$ with effective limit $\underline{u}_* \in \neg \mathcal{O}_{\Xi,N}(f)$. We have the following.

- 1) For all $M \in \mathbb{N}, q \in \mathbb{Q}, q > 0$, there exists $n \in \mathbb{N}$ such that we have $\Phi_{f,M}(\underline{u}_n) > q$.

Proof: The proof is divided into a construction and a contradiction part, which we label by paragraphs for the sake of a clear structure.

Part 1. Let $M \in \mathbb{N}$ and $q \in \mathbb{Q}, q > 0$, be arbitrary. Further, consider any computable sequence $(\underline{v}_n)_{n \in \mathbb{N}} \subset \Xi_{\mu,N}$ and define

$$(y_{n,m})_{n,m \in \mathbb{N}} : y_{n,m} = \dots \\ \dots \eta_{\Xi}(\underline{v}_n, \min\{l \in \mathbb{N} : l \geq qf(m+M)\}) - 2^{m+M}.$$

Then, $(y_{n,m})_{n,m \in \mathbb{N}}$ is a computable sequence of real numbers, as follows from f being a computable function, the Banach–Mazur computability of η_{Ξ} , and Summary 3, Part 4. Further, as follows from Lemma 1, there exists a μ -recursive function $g_{>} : D(g_{>}) \rightarrow \mathbb{N}$ satisfying

$$D(g_{>}) = \{n \in \mathbb{N} : \exists m \in \mathbb{N}. \text{s.t. } y_{n,m} > 0\}.$$

By construction of $(y_{n,m})_{n,m \in \mathbb{N}}$ and $g_{>}$, we have the following for all $n \in \mathbb{N}$.

- 1) Assume $\underline{v}_n \in \neg \mathcal{O}_{\Xi,N}(f)$. Then, there exists $m \in \mathbb{N}$ such that we have $y_{n,m} > 0$, implying $n \in D(g_{>})$.
- 2) Assume $\underline{v}_n \in \mathcal{O}_{\Xi,N}(f)$ and $\Phi_{f,M}(\underline{v}_n) \leq q$. Then, we have $y_{n,m} \leq 0$ for all $m \in \mathbb{N}$, implying $n \in \neg D(g_{>})$.

In summary, we obtain $\{n \in \mathbb{N} : \underline{v}_n \in \neg \mathcal{O}_{\Xi,N}(f)\} \subseteq D(g_{>})$ and $\{n \in \mathbb{N} : \underline{v}_n \in \mathcal{O}_{\Xi,N}(f), \Phi_{f,M}(\underline{v}_n) \leq q\} \subseteq \neg D(g_{>})$.

Part 2. We now prove the claim by contradiction. To this end, assume there exist $M \in \mathbb{N}, q \in \mathbb{Q}, q > 0$, such that for all $n \in \mathbb{N}$, we have $\Phi_{f,M}(\underline{u}_n) \leq q$. Observe the following.

- 1) Let $(\mathbf{v}_n)_{n \in \mathbb{N}} \subset (\{\mathbf{u}_n\}_{n \in \mathbb{N}} \cup \{\mathbf{u}_*\})$ be an arbitrary computable sequence. Then, as follows from Part 1, there exists a μ -recursive function $g_{>} : \mathbb{N} \supseteq \rightarrow \mathbb{N}$ satisfying

$$\begin{aligned} \{n \in \mathbb{N} : \mathbf{v}_n \in \{\mathbf{u}_*\}\} &\subseteq D(g_{>}) \\ \{n \in \mathbb{N} : \mathbf{v}_n \in \{\mathbf{u}_n\}_{n \in \mathbb{N}}\} &\subseteq \neg D(g_{>}) \end{aligned}$$

and thus $\{n \in \mathbb{N} : \mathbf{v}_n = \mathbf{u}_*\} = D(g_{>})$. Since we chose

$(\mathbf{v}_n)_{n \in \mathbb{N}}$ arbitrarily, we must have $\{\mathbf{u}_*\} \xrightarrow{\mu} \{\mathbf{u}_n\}_{n \in \mathbb{N}}$.

- 2) We have $(\{\mathbf{u}_n\}_{n \in \mathbb{N}} \cup \{\mathbf{u}_*\}) \subset \mathfrak{E}_{\mu, N}$ and $\mathfrak{E}_{\mu, N}$ is effectively closed. Thus, Lemma 2 implies $\{\mathbf{u}_*\} \xrightarrow{\mu} \{\mathbf{u}_n\}_{n \in \mathbb{N}}$.

This is absurd. \square

We now prove Theorem 3 by applying Lemmas 4 and 5. To this end, recall the definition of Id_M from Appendix B2. Moreover, note that the use of Proposition 3 in our proof Theorem 3 is structurally analogous to the use of Proposition 1 in our proof of Theorem 1.

Proof of Theorem 3: Observe that by Lemma 9 (if $\mathfrak{E} = \text{U}$) and Lemma 10 (if $\mathfrak{E} = \text{SU}$), there exists a computable sequence $(\mathbf{v}_n)_{n \in \mathbb{N}} \subset \neg \mathcal{O}_{\mathfrak{E}, N}(f)$ such that

$$\mathbf{v}_* \circ \text{Id}_M = \mathbf{u}_* \in \mathcal{O}_{\mathfrak{E}, N+M}(f)$$

is the effective limit of the sequence $(\mathbf{v}_n \circ \text{Id}_M)_{n \in \mathbb{N}} \subset \{\mathbf{v} \circ \text{Id}_M : \mathbf{v} \in \neg \mathcal{O}_{\mathfrak{E}, N}(f)\}$. Accordingly, upon proving Conditions (a)–(c) of Lemma 5 fulfilled for

$$\begin{aligned} F &\equiv \Phi_{f, L} \quad \mathcal{A}_{1, \mu} \equiv \mathcal{O}_{\mathfrak{E}, N+M}(f) \\ \mathcal{A}_{2, \mu} &\equiv \{\mathbf{v} \circ \text{Id}_M : \mathbf{v} \in \neg \mathcal{O}_{\mathfrak{E}, N}(f)\} \end{aligned}$$

there exists a computable sequence $(\mathbf{u}_n)_{n \in \mathbb{N}} \subset \mathcal{O}_{\mathfrak{E}, N+M}(f)$ with effective limit \mathbf{u}_* such that $\Phi_{f, L}$ isolates $(\mathbf{u}_n)_{n \in \mathbb{N}}$ from \mathbf{u}_* . The claim then follows by subsequent application of Lemma 4. Observe the following to conclude the proof.

Condition (a). By assumption, $\mathcal{O}_{\mathfrak{E}, M}(f)$ is nonempty. Thus, as follows from Lemma 8, $\mathcal{O}_{\mathfrak{E}, N+M}(f)$ is effectively dense in $\{\mathbf{v} \circ \text{Id}_M : \mathbf{v} \in \neg \mathcal{O}_{\mathfrak{E}, N}(f)\}$.

Condition (b). Let $(\tilde{\mathbf{u}}_n)_{n \in \mathbb{N}} \subset \mathcal{O}_{\mathfrak{E}, N+M}(f)$ be any computable sequence. By Proposition 3, there exists a computable sequence $(q_{n, m})_{n, m \in \mathbb{N}} \subset \mathbb{Q}$ such that for all $n, m \in \mathbb{N}$, we have $q_{n, m} \leq q_{n, m+1} \leq \sup_{l \in \mathbb{N}} q_{n, l} = \Phi_{f, L}(\tilde{\mathbf{u}}_n)$.

Condition (c). Let $(\tilde{\mathbf{u}}_n)_{n \in \mathbb{N}} \subset \mathcal{O}_{\mathfrak{E}, N+M}(f)$ be any computable sequence with effective limit $\tilde{\mathbf{u}}_* \in \neg \mathcal{O}_{\mathfrak{E}, N+M}$ and consider $q \in \mathbb{Q}$, $q > 0$, arbitrary. Then, by Lemma 11, there exists $m \in \mathbb{N}$ such that we have $\Phi_{f, L}(\mathbf{u}_m) > q$. \square

Subsequently, we provide another more technical interpretation of Theorem 3. To this end, observe that the proof of Theorem 3 crucially depends on the following.

- Given $N, M \in \mathbb{N}$ and any two gate families $\mathbf{u} \in \mathfrak{E}_{\mu, N}$, $\mathbf{v} \in \mathfrak{E}_{\mu, M}$, we may form the joint gate family $\mathbf{u} \circ \mathbf{v}$.
- Given $N \in \mathbb{N}$ and any gate family $\mathbf{u} \in \mathfrak{E}_{\mu, N}$, there is no a priori restriction on how close (with respect to d_{Δ}) \mathbf{u} is to the trivial gate family Id_N .

Joining gate families is, albeit rarely discussed explicitly, a pivotal concept in the gate-circuit formalism. The trivial gate family Id_N and, more precisely, the identity matrix Id play an equally fundamental yet simple role. Specifically, they formalize the ability to not execute any operation at all. The big-O characteristics of gate families are robust toward the joining operation, in the sense that the big-O characteristics of a joint gate family are always at least as good as the individual big-O characteristics of the joined gate families. Ultimately, Theorem 3 is thus a consequence of the discontinuity that occurs in the big-O characteristics of a joint gate family when its asymptotically dominant component gradually approaches the trivial gate family. As indicated in Section VII-A, Harrow et al. [5] already pointed out the dilemma regarding almost-trivial quantum gates. On the one hand, these gates are of high practical and theoretical relevance. On the other hand, they are ill-behaved in terms of compiling performance.

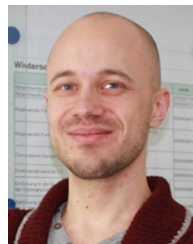
ACKNOWLEDGMENT

The authors thank Zoe Garcia del Toro from the LIP6 Quantum Information Team for her support in the literature review and in the editing of earlier versions of the article.

REFERENCES

- [1] R. P. Feynman, “Quantum mechanical computers,” *Opt. News*, vol. 11, no. 2, pp. 11–20, Feb. 1985, doi: [10.1364/ON.11.000011](https://doi.org/10.1364/ON.11.000011).
- [2] R. P. Feynman, “Quantum mechanical computers,” *Found. Phys.*, vol. 16, no. 6, pp. 507–531, Jun. 1986, doi: [10.1007/BF01886518](https://doi.org/10.1007/BF01886518).
- [3] A. Y. Kitaev, “Quantum computations: Algorithms and error correction,” *Russian Math. Surveys*, vol. 52, no. 6, Dec. 1997, Art. no. 1191, doi: [10.1070/RM1997v052n06ABEH002155](https://doi.org/10.1070/RM1997v052n06ABEH002155).
- [4] C. M. Dawson and M. A. Nielsen, “The Solovay-Kitaev algorithm,” *Quantum Inf. Computation*, vol. 6, no. 1, pp. 81–95, Jan. 2006, doi: [10.26421/QIC6.1-6](https://doi.org/10.26421/QIC6.1-6).
- [5] A. W. Harrow, B. Recht, and I. L. Chuang, “Efficient discrete approximations of quantum gates,” *J. Math. Phys.*, vol. 43, no. 9, pp. 4445–4451, 2002, doi: [10.1063/1.1495899](https://doi.org/10.1063/1.1495899).
- [6] J. Kusyk, S. M. Saeed, and M. U. Uyar, “Survey on quantum circuit compilation for noisy intermediate-scale quantum computers: Artificial intelligence to heuristics,” *IEEE Trans. Quantum Eng.*, vol. 2, 2021, Art. no. 2501616, doi: [10.1109/TQE.2021.3068355](https://doi.org/10.1109/TQE.2021.3068355).
- [7] L. Blum, M. Shub, and S. Smale, “On a theory of computation over the real numbers; NP completeness, recursive functions and universal machines,” in *Proc. 29th Annu. Symp. Found. Comput. Sci.*, 1988, pp. 387–397, doi: [10.1109/SFCS.1988.21955](https://doi.org/10.1109/SFCS.1988.21955).
- [8] L. Blum, “Computing over the reals: Where turing meets Newton,” *Notices Amer. Math. Soc.*, vol. 51, no. 9, pp. 1024–1034, 2004. [Online]. Available: <https://www.ams.org/journals/notices/200409/fea-blum.pdf?adat=September%202004&trk=200409fea-blum&cat=none&type=.pdf>
- [9] P. Papayannopoulos, “Unrealistic models for realistic computations: How idealisations help represent mathematical structures and found scientific computing,” *Synthese*, vol. 199, no. 1, pp. 249–283, 2021, doi: [10.1007/s11229-020-02654-8](https://doi.org/10.1007/s11229-020-02654-8).
- [10] F. P. Preparata and M. I. Shamos, *Computational Geometry* (Monographs in Computer Science Series), 1st ed. New York, NY, USA: Springer, 1985, doi: [10.1007/978-1-4612-1098-6_1](https://doi.org/10.1007/978-1-4612-1098-6_1).

- [11] H. Boche, Y. N. Böck, Z. G. del Toro, and F. H. Fitzek, "Feynman meets turing: The uncomputability of quantum gate-circuit emulation and concatenation," *IEEE Trans. Comput.*, vol. 74, no. 3, pp. 1053–1065, Mar. 2025, doi: [10.1109/TC.2024.3506861](https://doi.org/10.1109/TC.2024.3506861).
- [12] Y. N. Böck, H. Boche, Z. G. del Toro, and F. H. Fitzek, "Feynman meets turing: Computability aspects of quantum compiling revisited," *IEEE Trans. Comput.*, early access, Oct. 28, 2025, doi: [10.1109/TC.2025.3626469](https://doi.org/10.1109/TC.2025.3626469).
- [13] A. Y. Kitaev, A. Shen, and M. N. Vyalıy, *Classical and Quantum Computation*. Providence, RI, USA: Amer. Math. Soc., 2002, doi: [10.1090/gsm/047](https://doi.org/10.1090/gsm/047).
- [14] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge, U.K.: Cambridge Univ. Press, 2010, doi: [10.1017/CBO9780511976667](https://doi.org/10.1017/CBO9780511976667).
- [15] A. Bocharov and K. M. Svore, "Resource-optimal single-qubit quantum circuits," *Phys. Rev. Lett.*, vol. 109, no. 19, 2012, Art. no. 190501, doi: [10.1103/PhysRevLett.109.190501](https://doi.org/10.1103/PhysRevLett.109.190501).
- [16] V. Kliuchnikov, D. Maslov, and M. Mosca, "Fast and efficient exact synthesis of single-qubit unitaries generated by Clifford and T gates," *Quantum Inf. Computation*, vol. 13, no. 7–8, pp. 607–630, 2013, doi: [10.26421/QIC13.7-8-4](https://doi.org/10.26421/QIC13.7-8-4).
- [17] B. Giles and P. Selinger, "Exact synthesis of multiqubit Clifford + T circuits," *Phys. Rev. A—At., Mol., Opt. Phys.*, vol. 87, no. 3, 2013, Art. no. 032332, doi: [10.1103/PhysRevA.87.032332](https://doi.org/10.1103/PhysRevA.87.032332).
- [18] S. Forest, D. Gosset, V. Kliuchnikov, and D. McKinnon, "Exact synthesis of single-qubit unitaries over Clifford-cyclotomic gate sets," *J. Math. Phys.*, vol. 56, no. 8, 2015, Art. no. 082201, doi: [10.1063/1.4927100](https://doi.org/10.1063/1.4927100).
- [19] M. Amy, O. Bennett-Gibbs, and N. J. Ross, "Symbolic synthesis of Clifford circuits and beyond," *EPTCS*, vol. 394, pp. 343–362, 2023, doi: [10.4204/EPTCS.394.17](https://doi.org/10.4204/EPTCS.394.17).
- [20] A. Robert, P. K. Barkoutsos, S. Woerner, and I. Tavernelli, "Resource-efficient quantum algorithm for protein folding," *npj Quantum Inf.*, vol. 7, no. 1, 2021, Art. no. 38, doi: [10.1038/s41534-021-00368-4](https://doi.org/10.1038/s41534-021-00368-4).
- [21] R. I. Soare, *Recursively Enumerable Sets and Degrees* (Perspectives in Mathematical Logic Series). Berlin, Germany: Springer, 1987, doi: [10.1007/978-3-662-02460-7](https://doi.org/10.1007/978-3-662-02460-7).
- [22] M. B. Pour-El and J. I. Richards, *Computability in Analysis and Physics* (Perspectives in Logic Series). Cambridge, U.K.: Cambridge Univ. Press, 1989, doi: [10.1007/978-3-662-21717-7](https://doi.org/10.1007/978-3-662-21717-7).
- [23] A. Bauer, "The realizability approach to computable analysis and topology," Sch. Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-CS-00-164, 2000. [Online]. Available: <https://www.csd.cmu.edu/sites/default/files/phd-thesis/CMU-CS-00-164.pdf>
- [24] K. Weihrauch, *Computable Analysis* (Texts in Theoretical Computer Science. An EATCS Series). Berlin, Germany: Springer, 2000, doi: [10.1007/978-3-642-56999-9](https://doi.org/10.1007/978-3-642-56999-9).
- [25] V. Brattka and P. Hertling, *Handbook of Computability and Complexity in Analysis*. Berlin, Germany: Springer, 2021, doi: [10.1007/978-3-030-59234-9](https://doi.org/10.1007/978-3-030-59234-9).
- [26] S. C. Kleene, "General recursive functions of natural numbers," *Mathematische Annalen*, vol. 112, no. 1, pp. 727–742, 1936, doi: [10.1007/BF01565439](https://doi.org/10.1007/BF01565439).
- [27] A. M. Turing, "On computable numbers, with an application to the Entscheidungsproblem," *Proc. London Math. Soc.*, vol. s2-42, no. 1, pp. 230–265, Nov. 1936, doi: [10.1112/plms/s2-42.1.230](https://doi.org/10.1112/plms/s2-42.1.230).
- [28] A. M. Turing, "On computable numbers, with an application to the Entscheidungsproblem. A correction," *Proc. London Math. Soc.*, vol. s2-43, no. 1, pp. 544–546, Jan. 1937, doi: [10.1112/plms/s2-43.6.544](https://doi.org/10.1112/plms/s2-43.6.544).
- [29] A. M. Turing, "Computability and λ -definability," *J. Symbolic Log.*, vol. 2, no. 4, pp. 153–163, Dec. 1937, doi: [10.2307/2268280](https://doi.org/10.2307/2268280).
- [30] T. Heath, *A History of Greek Mathematics*, vol. 2. Oxford, U.K.: Oxford Univ. Press, 1921. [Online]. Available: <https://www.cambridge.org/us/universitypress/subjects/mathematics/historical-mathematical-texts/history-greek-mathematics-volume-2?format=PB&isbn=9781108063074>
- [31] J. Bourgain and A. Gamburd, "On the spectral gap for finitely-generated subgroups of $SU(2)$," *Inventiones Mathematicae*, vol. 171, no. 1, pp. 83–121, 2008, doi: [10.1007/s00222-007-0072-z](https://doi.org/10.1007/s00222-007-0072-z).
- [32] J. Bourgain and A. Gamburd, "A spectral gap theorem in $SU(d)$," *J. Eur. Math. Soc.*, vol. 14, no. 5, pp. 1455–1511, 2012, doi: [10.4171/jems/337](https://doi.org/10.4171/jems/337).
- [33] A. Sawicki and K. Karnas, "Criteria for universality of quantum gates," *Phys. Rev. A*, vol. 95, no. 6, 2017, Art. no. 062303, doi: [10.1103/PhysRevA.95.062303](https://doi.org/10.1103/PhysRevA.95.062303).
- [34] A. Sawicki and K. Karnas, "Universality of single-qudit gates," *Annales Henri Poincaré*, vol. 18, no. 11, pp. 3515–3552, 2017, doi: [10.1007/s00023-017-0604-z](https://doi.org/10.1007/s00023-017-0604-z).
- [35] O. Słowik and A. Sawicki, "Calculable lower bounds on the efficiency of universal sets of quantum gates," *J. Phys. A: Math. Theor.*, vol. 56, no. 11, 2023, Art. no. 115304, doi: [10.1088/1751-8121/acbd24](https://doi.org/10.1088/1751-8121/acbd24).
- [36] Z. Li, H. Zheng, J. Liu, L. Jiang, and Z.-W. Liu, "Designs from local random quantum circuits with $SU(d)$ symmetry," *PRX Quantum*, vol. 5, no. 4, 2024, Art. no. 040349, doi: [10.1103/PRXQuantum.5.040349](https://doi.org/10.1103/PRXQuantum.5.040349).
- [37] V. B. Bulchandani and S. Sondhi, "How smooth is quantum complexity?," *J. High Energy Phys.*, vol. 2021, no. 10, 2021, Art. no. 230, doi: [10.1007/JHEP10\(2021\)230](https://doi.org/10.1007/JHEP10(2021)230).
- [38] P. Dulian and A. Sawicki, "Matrix concentration inequalities and efficiency of random universal sets of quantum gates," *Quantum*, vol. 7, 2023, Art. no. 983, doi: [10.22331/q-2023-04-20-983](https://doi.org/10.22331/q-2023-04-20-983).
- [39] P. Dulian and A. Sawicki, "A random matrix model for random approximate t-designs," *IEEE Trans. Inf. Theory*, vol. 70, no. 4, pp. 2637–2654, Apr. 2024, doi: [10.1109/TIT.2024.3367787](https://doi.org/10.1109/TIT.2024.3367787).
- [40] M. Oszmaniec, A. Sawicki, and M. Horodecki, "Epsilon-nets, unitary designs, and random quantum circuits," *IEEE Trans. Inf. Theory*, vol. 68, no. 2, pp. 989–1015, Feb. 2022, doi: [10.1109/TIT.2021.3128110](https://doi.org/10.1109/TIT.2021.3128110).
- [41] A. Sawicki, L. Mattioli, and Z. Zimborás, "Universality verification for a set of quantum gates," *Phys. Rev. A*, vol. 105, no. 5, 2022, Art. no. 052602, doi: [10.1103/PhysRevA.105.052602](https://doi.org/10.1103/PhysRevA.105.052602).
- [42] O. Słowik, P. Dulian, and A. Sawicki, "Quantum circuit overhead," 2025, *arXiv:2505.00683*, doi: [10.48550/arXiv.2505.00683](https://doi.org/10.48550/arXiv.2505.00683).
- [43] J. J. Rotman, *An Introduction to the Theory of Groups*, 4th ed. New York, NY, USA: Springer, 1995, doi: [10.1007/978-1-4612-4176-1](https://doi.org/10.1007/978-1-4612-4176-1).
- [44] M. Amy, D. Maslov, M. Mosca, and M. Roetteler, "A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 32, no. 6, pp. 818–830, Jun. 2013, doi: [10.1109/TCAD.2013.2244643](https://doi.org/10.1109/TCAD.2013.2244643).
- [45] C. D. H. Boche and Y. Böck, "On effective convergence in Fekete's lemma and related combinatorial problems in information theory," in *Information Theory and Related Fields: Festschrift in Memory of Ning Cai* (Lecture Notes in Computer Science Series), vol. 14620. Cham, Switzerland: Springer, Apr. 2025, pp. 286–313, doi: [10.1007/978-3-031-82014-4_12](https://doi.org/10.1007/978-3-031-82014-4_12).
- [46] Y. Benoist and N. de Saxcé, "A spectral gap theorem in simple Lie groups," *Inventiones Mathematicae*, vol. 205, no. 2, pp. 337–361, 2016, doi: [10.1007/s00222-015-0636-2](https://doi.org/10.1007/s00222-015-0636-2).
- [47] P. P. Varjú, "Random walks in compact groups," *Documenta Mathematica*, vol. 18, pp. 1137–1175, 2013, doi: [10.4171/dm/423](https://doi.org/10.4171/dm/423).
- [48] A. Pauly, D. Seon, and M. Ziegler, "Computing haar measures," in *Proc. 28th EACSL Annu. Conf. Comput. Sci. Log.*, 2020, pp. 34:1–34:17, doi: [10.4230/LIPIcs.CSL.2020.34](https://doi.org/10.4230/LIPIcs.CSL.2020.34).
- [49] N. Alon and E. Lubetzky, "The Shannon capacity of a graph and the independence numbers of its powers," *IEEE Trans. Inf. Theory*, vol. 52, no. 5, pp. 2172–2176, May 2006, doi: [10.1109/TIT.2006.872856](https://doi.org/10.1109/TIT.2006.872856).
- [50] I. Marvian, "Restrictions on realizable unitary operations imposed by symmetry and locality," *Nature Phys.*, vol. 18, no. 3, pp. 283–289, 2022, doi: [10.1038/s41567-021-01464-0](https://doi.org/10.1038/s41567-021-01464-0).
- [51] I. Marvian, "Theory of quantum circuits with Abelian symmetries," *Phys. Rev. Res.*, vol. 6, no. 4, 2024, Art. no. 043292, doi: [10.1103/PhysRevResearch.6.043292](https://doi.org/10.1103/PhysRevResearch.6.043292).



Yannik N. Boeck received the B.Sc. and M.Sc. degrees in electronics engineering and information technology in 2016 and 2019, respectively, from Technische Universität München (TUM), Munich, Germany, where he is currently working toward the Ph.D. degree in electronics engineering and information technology.

Since 2019, he has been a Member of the research and teaching staff with the Chair of Theoretical Information Technology, TUM. His research interests include quantum information

theory, discrete mathematics, and applications of computability theory in engineering.



Holger Boche (Fellow, IEEE) received the Dipl.-Ing. in electrical engineering, Dipl. Math. in mathematics, and the Dr.-Ing. degree in electrical engineering from Technische Universität Dresden, Dresden, Germany, in 1990, 1992, and 1994, respectively, and the Dr. rer. nat. degree in pure mathematics from Technische Universität Berlin, Berlin, Germany, in 1998.

He is currently a Full Professor with the Chair of Theoretical Information Technology, Technische Universität München, Munich, Germany,

which he joined in October 2010. Since 2021, he and Frank H.P. Fitzek have been jointly heading the BMBF research hub 6G-life. He was a member with the German Academy of Sciences (Leopoldina) in 2008 and the Berlin Brandenburg Academy of Sciences and Humanities in 2009.

Dr. Boche was the recipient of the “Innovation Award” from the Vodafone Foundation in June 2006 and the Gottfried Wilhelm Leibniz Prize from the German Research Foundation in 2008.



Frank H.P. Fitzek (Fellow, IEEE) received the diploma in electrical engineering from RWTH Aachen University, Aachen, Germany, in 1997, and the Ph.D. degree in electrical engineering from the Technical University Berlin, Berlin, Germany, in 2002.

In 2002, he was with the University of Ferrara, Ferrara, Italy. In 2003, he was a Professor with Aalborg University, Aalborg, Denmark. He is currently a Professor and Chair with the “Deutsche Telekom Chair of Communication

Networks,” TU Dresden, Dresden, Germany, and leads at the forefront of telecommunications research in Germany. As the spokesperson for the DFG Cluster of Excellence CeTI and the 6G-life hub, his contributions have significantly shaped the field of communication networks. His research interests include 5G/6G communication networks, in-network computing, network coding, compressed sensing, post-Shannon theory, quantum and molecular communication, and immersive human-machine interaction in virtual environments.