PNAS RESEARCH ARTICLE | APPLIED PHYSICAL SCIENCES OPEN ACCESS

Chemputer and chemputation—A universal chemical compound synthesis machine

Leroy Cronin[a,1], Sebastian Pagel[a], and Abhishek Sharma[a]

Edited by Klavs F. Jensen, Massachusetts Institute of Technology, Cambridge, MA; received May 5, 2025; accepted February 26, 2026

Chemputation treats chemical synthesis as the execution of reaction code on programmable hardware. We show that a Chemputer, equipped with an extensible set of reagents, catalysts, and process conditions, together with a compiler that maps reaction and hardware graphs, is universal. This means it can produce any stable, isolable molecule in finite time and detectable quantity, provided real-time error correction maintains sufficient step fidelity relative to the number of steps in the synthesis. We formalize this into a Chemical Synthesis Turing Machine (CSTM), which defines chemical execution through a unified description of reagents, process variables, and catalysts. The framework introduces the Universal Chemputation Principle and a dynamic error-correction scheme that enables fault-tolerant synthesis. Linking this framework to assembly theory strengthens the definition of a molecule by demanding practical synthesizability and error correction becomes a prerequisite for universality. We demonstrate the abstraction is universal with more than 100 χDL programs executed on modular Chemputers, from single-step reactions to multistep syntheses. In each case, the number of unit operations scales linearly with synthetic depth. These results establish programmable chemical synthesis, chemputation, as a subset of general computation where χDL programs are compiled to hardware, executed with closed-loop control, and yield verifiable molecular outputs. This formalization enables shareable chemical code, interoperable hardware, and a machine-verifiable, executable foundation for a searchable and formally provable map of chemical space.

chemical synthesis | computer science | universality | chemputation | chemputer

Significance

Chemical synthesis is still performed today much like bespoke craftsmanship where each target molecule demands specialized equipment, ad hoc protocols, and labor intensive trial and error. We demonstrate that this is not a fundamental limitation of chemistry by formalizing a Chemical Synthesis Turing Machine (CSTM). With a practical modular Chemputer we prove that a single, reconfigurable hardware graph equipped with real time error correction can, in principle, construct any stable, isolable molecule in analytically detectable quantities. We explore this universality with assembly theory, establishing a quantitative bridge between a molecule's intrinsic information content and the resources required for its synthesis. This means that Chemputation paves the way for chemical programs to be formally specified and verified.

Turing completeness is a concept from theoretical computer science that defines the ability of a computational system to perform any computation that can be done by a Turing machine (1–3). For a system to be Turing-complete, it must have the capability to simulate a Turing machine. This means it can execute any algorithm, given sufficient time and memory, and solve any problem that is computationally solvable. Turing completeness is a foundational concept in understanding the limits of what can be computed. In essence, if a programming language or computational system is Turing complete, it can, in theory, perform any computation that a computer can, assuming no constraints on resources like time and memory. Expanding this concept to the realm of chemistry involves envisioning chemical systems that can perform operations that are in a way analogous to a Turing machine (4). Here, we explore this idea where chemical reactions are used to undergo programmable transformations in a device we call a Chemputer (5–9). The Chemputer is designed to automate and control chemical reactions with high precision (10). It uses a combination of hardware and software to carry out complex sequences of chemical processes (11). By programming these sequences, the Chemputer can perform tasks that require conditional logic, loops, and the manipulation of data—key components of Turing completeness.

Practically speaking, there are now many notable examples of chemistry automation with a wide range of chemical reactions, and therefore, represent hardware-specific chemical processes. For instance, the synthesis of sequence-defined biopolymers whose syntheses already follow deterministic, stepwise logic with some feedback control. A great example is modern solid-phase peptide synthesis (12, 13), which is routinely used to construct complex peptides using protected amino-acid cartridges and inline deprotection checks, while on-solid-phase cleavage is triggered only once conversion has occurred. The same approach governs automated oligonucleotide production (14). Perhaps the most ubiquitous reaction to be encoded for small molecules has been the amide-bond formation (15), and this is now followed by Suzuki–Miyaura (16), Buchwald–Hartwig (17), and Sonogashira (18) reactions. Other examples include microfluidic systems for droplet-based chemistry (19), DNA encoded libraries (20), and other combinatorial chemistry approaches (21). In addition, recently there has been an explosion of work using flow systems for

Author affiliations: [a]School of Chemistry, Advanced Research Centre, University of Glasgow, Glasgow G11 6EW, United Kingdom

Author contributions: L.C. designed research; L.C. and S.P. performed research; L.C. and A.S. contributed new reagents/analytic tools; L.C., S.P., and A.S. analyzed data; and L.C. wrote the paper.

Competing interest statement: LC is the founder and a shareholder in Chemify Ltd, https://www.chemify.io/.

This article is a PNAS Direct Submission.

[1]To whom correspondence may be addressed. Email: lee.cronin@glasgow.ac.uk.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2511080123/-/DCSupplemental.

Published April 7, 2026.

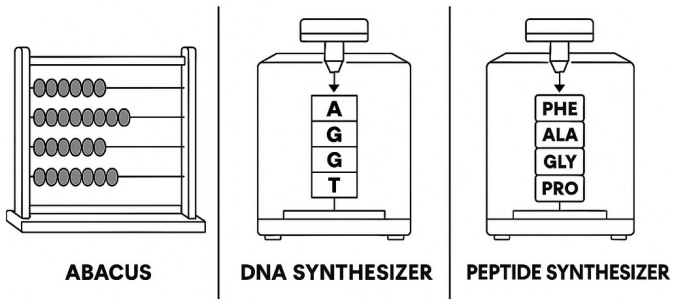

Fig. 1. The abacus, DNA, and peptide synthesizer are both examples of limited programmable machines. The abacus for numerical calculations and the DNA and peptide synthesizer for making sequences of DNA and peptides respectively.

chemical synthesis (18, 22) as well as a vast number of digital chemistry or self-driving lab endeavors (23–26) many of which are aiming for digital synthesis (27–30), real-time monitoring and optimization (31–34), mobility (35, 36), autonomy (37, 38). There are clear advantages for investment in such systems for the control of highly exothermic reactions, process optimization, and exploring new inline sensors. However, in all these examples, they are all encoded from the hardware up, meaning that a fully abstract approach to programming the chemistry has not been possible. These challenges are important because while there are use cases for all these approaches, the real challenge is to find a universal route to in principle encode all of chemistry at an abstract level so that it can then be run on chemically agnostic devices. Only then will the promise of digital chemistry become a reality.

Here, we present a universal approach to programming and executing chemical operations that will lay the foundations for chemputation, unifying the programming of chemistry across all of chemical synthesis, design, discovery, and automation. By having a universal, "Turing complete" standard, it will be possible to embrace the many different approaches into a single programmable paradigm. This will allow aspects of provability, interoperability, defining an entirely new ecosystem. With a universal system, teams will be able to develop a common programming standard, develop interoperable hardware modules, produce systems that are able to reproduce each other's results, build a repository of both negative and positive reaction data, and, finally, publish executable chemical code. We defined a Chemputer as a system that can execute a standard chemical code to make a wide range of different molecules and the process of running the machine with the code to get the chemical outputs as chemputation.

Foundation of the Chemputer

Since the late-1960s progress in modular robotics, low-cost multimodal sensing, and data-driven route design has propelled the field of chemical synthesis from task-specific robots—peptide assemblers, DNA synthesizers, high-throughput flow loops—toward fully programmable chemical platforms. Yet, as described above most current systems remain constrained to a narrow reaction manifold and typically lack the capacity to rewire their own hardware topology in response to a new synthesis plan. To appreciate the qualitative leap from single-use automation to a universal chemical programming language (39), it helps to recall the gulf between an abacus and a programmable computer; see Fig. 1. The abacus performs fixed arithmetic by sliding beads along rails—it is fast and reliable, but it cannot be coaxed into factoring integers, searching a database, or rendering graphics without physically altering its structure. Its function set is frozen in hardware. By contrast, a computer's power stems from a universal instruction set. Here, the processes of load, add, branch, and store are integral. These few op-codes, scripted in software, can realize any computable routine, limited only by time and memory. In contrast to the single use chemical synthesis machines, similar to the abacus, where the reaction process is mostly hard coded, the Chemputer concept generalizes these efforts as it treats synthetic chemistry itself as a form of computation.

This is because the reagents are data structures, reaction conditions are control flow, and hardware modules act as the physical instantiation of op-codes. By introducing a chempiler, a compiler that converts abstract reaction graphs into executable hardware graphs, the Chemputer aims not merely to accelerate known protocols but to search, optimize, and execute entirely new routes on demand, in the same way a universal Turing machine can execute any algorithm expressible in its instruction set.

In this paper, we formalize chemputation by i) defining an abstract extended chemical state machine that captures reagents, catalysts, and process conditions as state variables and show how resources on a graph can be used to instantiate a wide range of chemical processes; ii) proving a Universal Synthesis Theorem that any stable, isolable molecule can be reached through a finite sequence of such state transitions; and iii) embedding dynamic error-correction routines that guarantee robustness in the face of real-world deviations. This foundation lays the groundwork for a future in which chemical manufacture is as programmable and portable as software is today, and it also provides a connection between assembly theory (40) and chemputation. This is because the assembly index is a measure of molecular complexity (41), or the minimum number of constraints required to construct the molecule by considering bonds as building blocks. This work also demonstrates how the concept of the assembly index and the copy number play a profound role in understanding what is synthetically accessible and can be detected using analytical chemistry techniques (42, 43).

The Concept

The concept of a Chemputer as a universal chemical synthesis machine posits that it can instantiate any feasible chemical synthesis (Fig. 2). We outline the proof for the universality of the Chemputer, demonstrating that it can synthesize any target compound within the chemical space defined by the provided parameters. To prove the universality of the Chemputer, we need to demonstrate that it can conduct any feasible chemical synthesis at the most abstract level. This involves showing that the transformation function δ can be used to map all chemical reactions possible under the defined

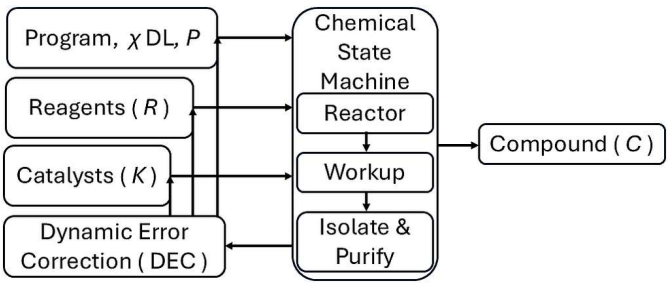

Fig. 2. A schematic of one possible CSTM. The inputs are the Reagents (*R*), and the chemical program or χDL file (8) contains details of the process conditions (*P*) which include any catalysts (*K*), and code to run the hardware. The output is the pure target compounds (*C*) from the chemical state machine which includes a reactor, workup, isolate, and purify system. DEC is invoked at each state transition of the CSTM, acting during reaction, work-up, and isolation steps to validate intermediates against expected outcomes and to adapt process variables prior to progression.

Table 1. Example unit operations from chemistry expressed in the primitives

Unit operations	Primitive sequence	Notes
Liquid–liquid extraction	AM → AE → SM	Add, mix, separate
Drying	AE→SM	Heat, remove water
Crystallization	AE→SE→SM	Heat, cool, filter
Distillation	AE→ SM → SE → AM	Heat, remove vapor, cool, collect
Hot reaction; Cold reaction	AM→AE; AM→SE	Add matter then heat or cool
Sublimation	SM→AE→SE→AM	Add vacuum, heat, cool window, collect from window

reagents, process conditions, and catalysts (it has been suggested that catalysts might themselves be viewed as a type of constructor) (44, 45). Furthermore, we incorporate the mechanisms of dynamic error correction (DEC) (46, 47) during synthesis (22, 48) and the use of universally configurable hardware to support complex chemical processes through a chempiling function. This allows the abstraction to be implemented at the module and device level for chemical synthesis. DEC operates at defined checkpoints within the chemical state machine, specifically during reaction execution, work-up, and isolation/purification. At each of these stages, inline analytical feedback is used to compare the observed state of the system against the expected intermediate or product state encoded in the χDL program. Deviations trigger corrective actions such as parameter adjustment, repetition of unit operations, or rollback to a validated intermediate before the synthesis proceeds to the next state.

The Turing Machine Abstraction of the Chemputer

To build the abstraction of the Chemputer, we need to set up the abstraction of the Chemical Synthesis Turing Machine (CSTM). This comprises an infinite tape where each space on the tape is a vessel, and each vessel can be one of three types of being either empty, filled, or active. The vessels can be subjected to the four primitives of Add Matter [AM], Subtract Matter [SM], Add Energy [AE], or Subtract Energy [SE] using the head. From these four primitives, any unit operation can be emulated over the entirety of chemical synthesis; see Table 1.

To fully express the machinery for chemical synthesis, we define the CSTM, as the 8-tuple:

$$c = \left\langle Q, \Sigma_R, \Sigma_P, \Gamma,\ b, \delta, q_0, H \right\rangle, \qquad [1]$$

where

Q is a finite set of states and $q_o \in Q$ is the initial state;
Σ_R is a finite reagent alphabet;
Σ_P is a process alphabet;
$\Gamma = (\Sigma_R \times \Sigma_P) \cup \{b\}$ is the tape alphabet whose blank symbol is b;
δ: $Q \times \Gamma \rightarrow Q \times \Gamma \times \{Left, Right, N\}$ where the transition function, realized physically by the primitives AM, SM, AE, SE;
$H = \{q_{out}, q_{uout}, q_{nout}, q_{fail}\} \subseteq Q$ is halting set with,

- q_{out} is a successful termination identical to a previously characterized chemputation;
- q_{uout} halting after a theoretically predicted but as yet unoptimized outcome;
- q_{nout} marking the discovery of a genuinely novel transformation;
- q_{fail} indicating a chemically unrecoverable termination.

In every halting case the final tape encodes a complete laboratory trace. For q_{uout} and q_{nout} an external optimization module (DEC) may launch a nondeterministic exploration of $(\Sigma_R \times \Sigma_P)$ to improve yield or generate new reaction rules before the exact trace is committed to the rule database; see Fig. 3. When either a q_{uout} or q_{nout} is repeated it becomes q_{out}. In practical synthetic execution, DEC is engaged before any halting condition is accepted, ensuring that a state is only classified as q_{out}, q_{uout}, or q_{nout} after analytical validation of composition, yield, or selectivity at that stage.

The implementation of the CSTM is designed such that it preserves mass balance see Supplementary Data such that the result of the transformations does not lose matter and hence stoichiometry. When matter is subtracted, the contents are removed to another vessel and if waste, to the waste vessel. Overall, for the synthesis of a compound from the set of all possible compounds, we can show that $\forall$ c $\in$ C, $\exists$ CSTM program X_c such that out(X_c) = c $\wedge$ X_c halts. So, it is now straightforward to construct an archetypal CSTM schema that can be used to react reagents A and B together. By setting the tape of the machine to be equivalent to a series of chemical vessels, it is possible to build the machine that can use the head to address the locations on the tape to conduct one of the four-unit operations; see supplementary information. Here, the cells can be considered to be infinite as long as material can be removed from the cells and the cells instantiated ready for further operations; see Movie S1.

Definitions

Chemputer and Chemputation: A system that runs the code to do the chemistry is the Chemputer and the process of running the code is chemputation.

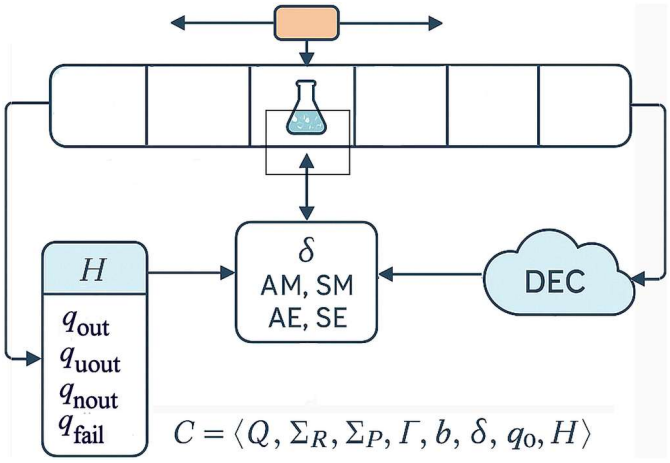

Fig. 3. A schematic depicting the abstraction of the Universal CSTM. This machine is controlled by a finite state controller and the head addresses the cells on the tape by either adding matter, subtracting matter, adding energy, or subtracting energy. The system halts when one of four conditions is satisfied. DEC acts between state transitions and prior to halting, using analytical feedback to either accept the current tape state, initiate corrective actions, or launch local re-optimization before committing the outcome to the rule database.

Reagent Space (R): A finite working set drawn from the space of all possible chemical reagents, including all chemical elements and basic compounds.

Process Conditions (P): A set of environmental parameters (e.g., temperature, pressure, solvent/gas conditions, energy input type) that influence the outcome of reactions. This includes Catalysts (K): A set of substances that alter the reaction pathways or rates without being consumed in the process.

Target Compounds (C): The set of desired products or output compounds, typically molecules. Such molecules can be defined as an electrically neutral entity consisting of more than one atom. In addition, the molecule must correspond to a depression on the potential energy surface that is deep enough to confine at least one vibrational state. Finally, the molecule should be accessible by chemputation, with a sufficiently large amount of the molecule synthetically accessible to be detected.

Universally Configurable Hardware (U): A hardware platform that can be dynamically reconfigured to execute various chemical synthesis processes. In the Chemputer, the system is constrained by a finite number of reagent input vessels, reaction vessels, and product output vessels, V_R, V_P, VO respectively; however, the tape is conceptually infinite since the vessels can be instantiated serially by reusing vessels and off-loading intermediates. This means that any chemical synthesis is realizable if it can be completed within these finite resources. The configuration is represented as a graph $G = (V,E)$, where: V is a set of nodes representing hardware components (e.g., reactors, mixers, sensors) and E is a set of edges representing connections between components, defining the flow of matter, energy, and information; see Fig. 4.

This graph has the basic hardware resources for chemical reactions, work-up, isolation, and purification and the modules shown are practical implementations of the four chemputation primitives: add matter, subtract matter, add energy, subtract energy. As such a system such as that shown in Fig. 4 is capable of a wide range of chemputations. Of course there will always be practical limitations but the system shown above is not process or chemical reaction limited.

DEC: A mechanism embedded within each step of the synthesis process, enabling real-time detection and correction of errors, ensuring the accuracy of each transformation including the discovery of new transformations, before proceeding to the next step.

Chempiling Function (χ): The process of translating a synthesis pathway σ into a corresponding hardware configuration $G(U)$ that can execute the synthesis process.

Definition of a Molecule, the Assembly Index, and the Role of Error Correction

Chemical synthesis is inherently prone to error, and as molecular complexity increases, achieving error-free assembly becomes exponentially more difficult. To formalize this constraint, we expand the traditional definition of a molecule by incorporating practical synthetic accessibility.

Conventionally, a molecule is defined (49) as a finite set of nuclei and electrons occupying a stable local minimum on the Born–Oppenheimer potential energy surface. Here, we refine this by requiring that the molecule must also be realizable: it must be possible to produce enough perfect copies to detect the molecule experimentally, despite finite synthesis resources and inevitable errors.

Specifically, a molecule must satisfy the condition:

$$N \geq N_{min} = \frac{\varphi}{\prod_{k=1}^{a_i} (1 - \varepsilon_k)}, \quad [2]$$

where N_{min} is the minimum number of copies that must be synthesized and N is the number of perfect copies, φ is the minimum number of perfect copies required for reliable detection (typically 10^6–10^8), a_i is the assembly index (41, 42) the minimal number of logical steps needed to construct the molecule, ε_k is the error probability at each assembly step.

$N_{perfect}$ is the number of molecules produced given the intrinsic error rate for each step and is equal to $\prod_{k=1}^{a_i} (1 - \varepsilon_k)$.

For simplicity, assuming a constant error rate ε across all steps, this expression reduces to:

$$N \geq N_{min} = \frac{\varphi}{(1 - \varepsilon)^{a_i}}. \quad [3]$$

Thus, as the assembly index a_i increases, or as the per-step fidelity declines, the number of molecules required to ensure detection grows exponentially. To provide intuition: each synthetic operation is like placing a brick when building a fragile structure. If each brick has a small chance of being misplaced, the probability of completing the structure without error declines rapidly as the number of bricks increases.

Axioms and Lemmas

For the CSTM to operate universally we need to introduce five axioms (A) and three lemmas (L): A1: Conservation of matter; A2: Finite reaction time; A3: Stability of elements found in R under

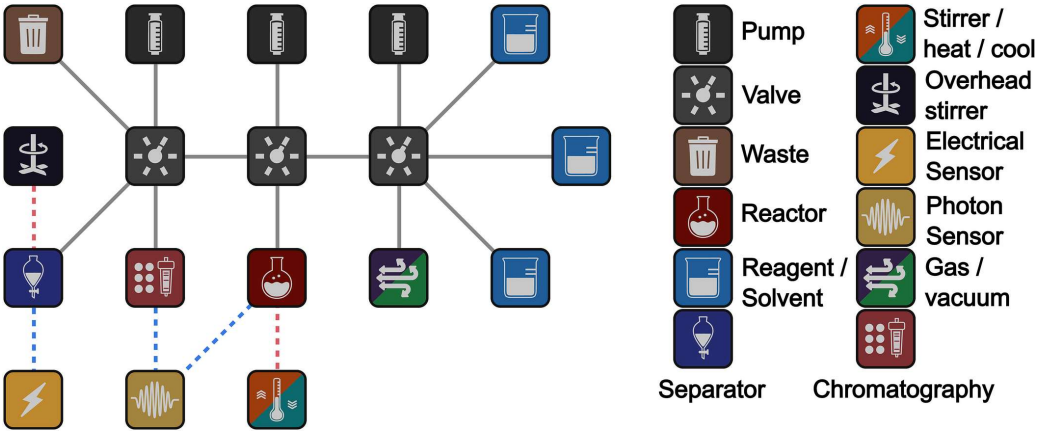

Fig. 4. A general-purpose graph for the Chemputer that can be practically implemented in the laboratory. The modules are shown on the right. The reagents, pumps, and valves are needed with the reactor to set up the reaction. The stirrer/heater/cooler is used to control the reaction. The reaction work-up uses a separator and a conductivity sensor. The combination of spectroscopic devices (photon sensor) on the reactor and chromatography output can allow real time feedback to help optimize both the reaction yield, purity, and selectivity.

standard conditions; A4: Every compound has a shortest path, defined by the assembly index, a_i, which is defined as the shortest path to assemble the compound from fundamental building blocks, allowing only binary combination of parts, and allowing reuse of parts; A5: Detectability constraint; $c \in C$, a synthesis is considered realizable only if the expected flawless-copy count satisfies $N \geq N_{min}$. L1: For any $c \in C$ there exists a finite sequence of transformations σ and finite copy number, N such that σ executed N times satisfies A5. Proof: By the definition of C and finite reaction time axiom with a sequence of transformations σ; L2: For any $c \in C$ there exists a shortest path to construct c on a graph that only uses the building blocks found within c, allowing recursion. Proof: By the definition of C; L3: For any transformation function $t \in \delta$ can be decomposed into a finite sequence of elementary reactions. Proof: By the nature of chemical reactions and the conservation of matter.

Assumptions and Formalization

1. Existence of a Universal Set-Up: This demonstrates that the Chemputer can implement any feasible chemical synthesis, showing that the function δ is sufficiently general to account for all chemical reactions possible under the reagents given, process conditions, and catalysts.
2. Construction of Synthesis Pathway: For each target compound c, a sequence σ of transformations from initial reagents R_0 to c can be constructed. This construction must account for all intermediate transformations and ensure that σ is valid under P, and K.
3. Verification of Stability: This verifies that for the resulting compound c, the stability condition $S(c)$ is satisfied.
4. Dynamic Error Detection and Correction: The Chemputer can detect errors in real-time during each step of the synthesis by continuously monitoring the reaction progress and comparing the actual outcome with the expected result. Upon detecting an error during any synthesis step, the Chemputer applies corrective steps immediately, either reverting to a previous state or adjusting the process to ensure the synthesis remains on track.
5. Universality and Completeness: This proves that for any $c \in C$, there exists a pathway σ and a stable outcome, demonstrating the universality of the Chemputer as a synthesis device, including error detection and correction at each synthesis step.

A molecule $c \in C$ is considered realizable in the Chemputer if

$$S(c)\text{: Stability } c \in C \text{ such that } c \text{ is isolable and stable.} \quad [4]$$

σ exist such that $\sigma\ (R_0, \ldots, R_n) = c$ and

σ produces at least $N_{\min}$ perfect copies as defined in Eq. **2**.

The stability condition $S(c)$ ensures that the resulting compound c is stable and can be isolated, specifically $S(c)$ must hold true for the synthesis to be considered successful. However, the synthesis may or may not utilize unstable reaction intermediates that could be isolated for some period of time. For the target to be produced there is a transformation function (δ) done by the CSTM where

$$\delta \text{ maps } Q \times \Gamma \text{ to } Q \times \Gamma \times \{\text{Left, Right}, N\}\, C. \quad [5]$$

The transformation function δ from the CSTM defines the emergent property we conventionally call the reaction rule which is the resultant outcome when reagents R are added under the process conditions P, in the presence of catalysts K to give the output compounds C. The transformation function can be used to predict how the reagent graphs R can be transformed into the product graphs C as graph transformations between the reagents R. To get to the product we need to achieve the construction of synthesis pathways (σ) so for any target compound $c \in C$, we construct a pathway σ such that

$$\sigma\text{: } (R0, \ldots, Rn, \varepsilon_k) \rightarrow N_C \geq \frac{\varphi}{\prod_{k=1}^{a_i}(1-\varepsilon_k)}. \quad [6]$$

A synthesis pathway σ is a sequence of transformations leading from an initial set of reagents R_0 through intermediate sets $R_0, \ldots, R_n$ to the final product c. The Chemputer is said to be universal if, for any target compound c in the set of desired compounds C, there exists a sequence of transformations σ that leads from an initial set of reagents R_0 to c.

The Universal Chemputation Principle (UCP)

Every stable, isolable molecule $c \in C$ that satisfies the abundance condition is realizable by a finite chemputation program executed on universally reconfigurable hardware where

$$\forall c \in C,\ \exists\, R0 \subseteq R,\ P,\ K \text{ such that } \sigma(R0, \ldots, Rn) = c \text{ and } N_c \geq N_{min}(c). \quad [7]$$

σ is the synthesis pathway constructed accessed using transformation function in the CSTM δ

N_c is the number of defect-free copies of molecule c produced,

N_{min} (c) is the minimum number of such copies required for the molecule to be confirmed real by detection.

This asserts that for every target compound c in C, there exists a set of initial reagents $R_0 \subseteq R$, a set of process conditions P such that a synthesis pathway σ exists, leading from R_0 to c. This requires a dynamic error detection and correction system where

$$\text{DEC}: c' \rightarrow \text{Corrected State.} \quad [8]$$

DEC is applied at each step in the synthesis process. For each transformation, if the outcome c' deviates from the expected intermediate or final product c_n, the error detection function DEC flags the deviation. The error correction function DEC is then applied to revert to a prior valid state or adjust the process dynamically to ensure that the synthesis remains accurate. If the compound c, is novel, then the error correction will be used to maximize the yield of the hitherto unknown compound and output a new set of rules.

The Chempiling Function (χ)

$$\chi : \sigma \rightarrow G(U). \quad [9]$$

χ maps the synthesis pathway σ into a hardware configuration $G(U)$ that can execute the synthesis process. For simple compounds (e.g., elements or basic molecules), the Chemputer can directly synthesize them from their constituent elements or simpler precursors. If an error occurs during the synthesis of these simple compounds, it is detected and corrected dynamically before proceeding. By assuming the chemputer can synthesize all compounds of complexity k (meaning requiring k steps), with DEC applied at each step. For a compound of complexity $k + 1$, there exists a precursor compound requiring k steps and a transformation function δ that can transform this precursor into the target compound under appropriate P, and K in the presence of the reagents R; see Fig. 5. DEC can be used to optimize the output δ of the transformation function as it maps onto the synthesis pathway and chempiles onto the hardware. This ensures that each intermediate step is accurate. Therefore, by extension, the chemputer can synthesize all compounds up to any finite complexity, as long as there is enough compound synthesized to allow analytical detection of the molecule present in c.

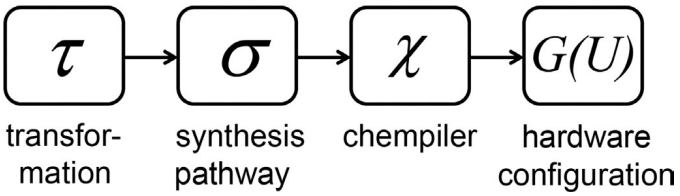

Fig. 5. Outline of the process of chemputation from the CSTM to the synthesis pathway which then leads to chempilation on the available hardware.

The concept of DEC represents a cornerstone of chemputation, critical to ensuring the robustness, reliability, and accuracy of fully automated chemical synthesis (46). The foundation of DEC will lie in its ability to integrate continuous real-time analytical feedback throughout every stage of the chemical reaction, automatically identifying deviations from expected outcomes and implementing corrective measures to maintain the integrity and desired selectivity of the synthetic process. At its core, DEC will operate through a closed-loop feedback mechanism involving continuous data collection, analysis, and adaptive intervention. This cycle begins with the initial calibration of the chemputer's embedded sensors, such as spectroscopic (IR, Raman, UV-Vis, NMR), chromatographic (LC, GC, HPLC), mass spectrometric, and various physical (temperature, pressure, viscosity, conductivity) sensors. Once calibrated, these sensors will continuously acquire real-time data, can be logged and compared against the predefined reaction profiles embedded within the chemputer's internal reference database. As the reaction proceeds, the system will be designed such that embedded software algorithms actively monitor this sensor data, detecting deviations from expected trajectories through threshold-based or machine-learning-driven anomaly detection methods. Upon identifying a deviation, the system will classify the errors according to severity: minor deviations, indicating slight reductions in yield or selectivity; intermediate deviations, reflecting incomplete reactions or minor impurity formation; and major deviations, such as unexpected side reactions or critical intermediate failures.

Each class of error will prompt the chemputer to autonomously execute adaptive corrections tailored specifically to the severity and nature of the observed deviation. For minor deviations, subtle adjustments may be made to reaction parameters, such as incremental temperature changes or adjustments in stirring rates. Intermediate deviations will have to trigger more substantial corrections, such as additional reagent or catalyst doses, extended reaction durations, or moderate temperature adjustments. Major deviations will require more extensive interventions, such as reverting the reaction mixture to a stable precursor state and recalculating alternative reaction conditions before restarting the process. Sensor integration will be central to the effectiveness of DEC. For example spectroscopic sensors, including infrared (IR), Raman, and ultraviolet-visible (UV-Vis), could give detailed, real-time molecular information, confirming the formation of desired intermediates or products and quickly highlighting anomalies. Inline chromatographic and mass spectrometric analyses will offer precise compositional insights, rapidly verifying purity and yield at critical reaction stages. Physical sensors—such as temperature probes, pressure sensors, and pH meters—will continuously ensure that reaction environments remain within the defined optimal conditions.

It is envisaged that DEC will be further enhanced by the integration of machine-learning algorithms trained on extensive historical reaction data. Machine learning allows the chemputer to anticipate and predict potential deviations based on patterns from previous experiments, enabling proactive rather than merely reactive corrections. Moreover, these algorithms will facilitate adaptive parameter tuning, automatically refining reaction conditions over time to progressively improve yields and purity. Additionally, when significant deviations occur, machine learning algorithms may intelligently recalibrate or even redesign reaction pathways in real-time, exploring alternative synthesis routes dynamically. Thus, the careful management of error propagation will be particularly critical in multistep syntheses. DEC effectively mitigates cumulative errors by embedding validation checkpoints after each step, systematically ensuring the quality and completeness of intermediates before progressing. Recursive correction loops further address error propagation by revisiting and correcting previous steps if downstream deviations are detected. This continual optimization of reaction parameters at each stage greatly enhances overall synthetic reliability, dramatically reducing cumulative error effects.

When it comes to exploring synthesis complexity, the assembly index a_i captures the minimum causal information required to assemble a molecule. It is intrinsic to the molecule's molecular structure, independent of any synthetic route. Conceptually, a_i measures how "hard" a molecule is to build in the best possible scenario: allowing for recursive reuse of parts, not just sequential chemical steps. The theoretical bounds relate the number of bonds B in a molecular graph $MG = (V,E)$ to the assembly index where the minimum is $\log_2 B$ and the maximum is $B - 1$. Thus, molecules with a large number of bonds typically have higher a_i, but highly symmetric structures can have surprisingly low assembly indices due to recursive assembly. Importantly, as the assembly index increases, the synthetic error must be controlled more tightly. Otherwise, the number of correctly assembled copies declines sharply, imposing a practical limit on the size, complexity, and detectability of molecules. This effect is illustrated in Fig. 6, where increasing assembly index, coupled with modest per-step error rates, leads to rapid depletion of the perfect copy population. For example, with a 5% error rate per assembly step, after only 20 steps, fewer than 40% of the molecules are flawless.

Given the inevitability of synthesis errors, DEC is a foundational principle of chemputation. This is because DEC integrates real-time monitoring and adaptive interventions at each synthesis step. Sensors embedded within the chemputer (e.g., spectroscopy, chromatography, temperature, and pH probes) continuously assess reaction progress. Deviations from expected outcomes are detected early and corrected—by adjusting reaction parameters, extending reaction time, or reverting to stable intermediates. Minor deviations may trigger fine-tuning (e.g., temperature adjustments), while major deviations may require reverting to a prior synthetic state. Machine learning algorithms further enhance DEC by predicting likely failure points based on accumulated reaction data. This closed-loop correction minimizes cumulative errors, allowing the successful synthesis of molecules with high assembly indices that would otherwise be inaccessible. It is important that we distinguish assembly index from synthetic steps.

A key conceptual challenge is understanding that the assembly index a_i is not equivalent to the number of synthetic steps in a reaction sequence. The assembly index captures the minimal number of nonredundant construction operations required to specify a molecular graph, allowing recursive reuse of substructures. In contrast, traditional synthesis plans enumerate every experimental transformation, including redundant or linear steps, and are deeply dependent on available reagents, human planning, and practical constraints.

A lab synthesis might be "short" because it uses complex building blocks, while the true assembly index—if those blocks must themselves be constructed—is much higher. This difference is not a flaw but a feature: it reveals that assembly index is the only

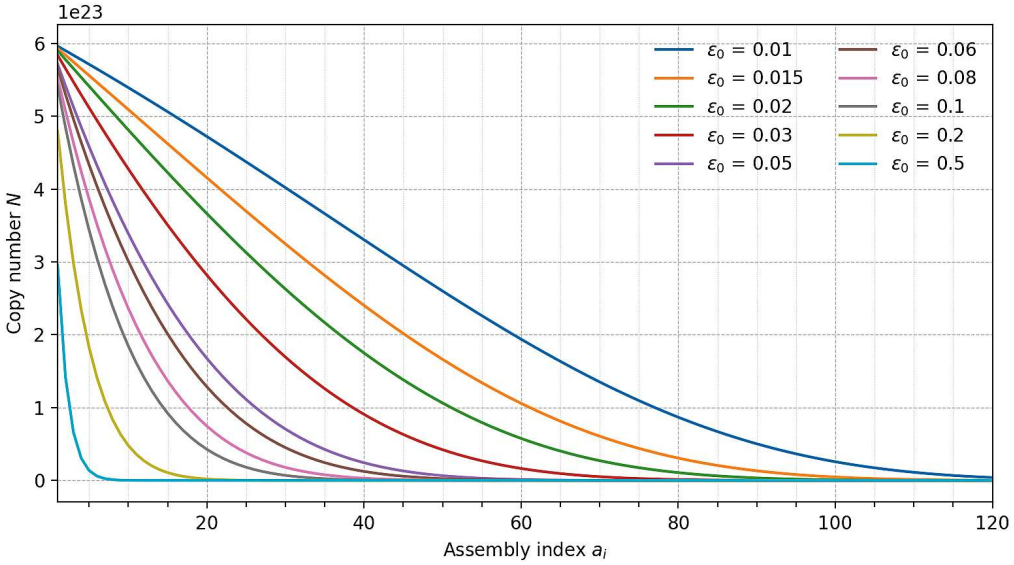

Fig. 6. The mean number of flawless copies N remaining after each assembly step (a_i = 1 − 120) is plotted for ten baseline per-step error probabilities (ε_0 = 0.01, 0.015, 0.02, 0.03, 0.05, 0.06, 0.08, 0.10, 0.20, 0.50; colored lines, legend right). At a_i = 1 the copy number reflects survival after the first assembly step from an initial population of Avogadro's number of copies ($N_0 = 6.022 \times 10^{23}$). The baseline error rate rises exponentially with assembly index (k = 0.02), and a normally distributed systematic term E_k (μ = 0, σ = 0.005) is added to every trajectory. For each ε_0 value, 5,000 Monte-Carlo trajectories were generated, and the solid lines show the arithmetic mean across simulations.

objective, intrinsic measure of molecular complexity. It is independent of lab strategy, human intuition, or access to reagents. In this way, assembly theory serves as an intrinsic molecular complexity, providing a universal lower bound on the information and causation needed to construct any molecule. Synthetic step count, by contrast, is contingent and observer-dependent.

A useful analogy comes from computer science. The assembly index is like the minimal program that generates an output—it is compact, elegant, and often recursive. A lab synthesis, on the other hand, is like the execution trace of that program on real hardware: long, verbose, and constrained by memory, I/O, or user habits. We would not confuse a full execution trace for the "complexity" of an algorithm—similarly, we should not confuse step count with true molecular complexity. While synthetic step count remains a valuable practical metric, it cannot serve as an absolute basis for defining molecular complexity. Only the assembly index reflects the irreducible causal structure required to construct a molecule. This makes it foundational for universal chemputation, where the synthesis must be encoded, predicted, and executed independently of human bias or experimental convenience.

Examples

To explore examples of real-world synthesis done using our Chemical Description Language, χDL, we took 117 different synthesis routes we have run on our automated synthesis platforms to analyze here in terms of our CSTM abstraction (5, 7, 8). We show that the steps in synthetic procedures can be classified into a finite set of categories, similar to the finite instruction set in a Turing Machine. The underlying hardware of the Chemputer allows to be dynamically reconfigured to explore a combinatorial set of potential procedures on demand. The reactions were selected based on popularity in medicinal chemistry, process chemistry, and total syntheses (50–53). The resulting set of validated χDLs covers a substantial range of common reactions and constitutes an entry point to the automation of the entire organic synthesis "toolbox." Also, all the procedures cover highly diverse chemistry showing that one unified hardware and software of the Chemputer can work; see Fig. 7.

The exploration of these previously performed reactions on well-established robotic systems, highlights how this approach can be used to unify and also standardize the use of diverse robotic architectures for chemical automation. While synthesis for all molecules can be expressed using a universal set of hardware components (see above), each synthesis(-step) may employ a unique combination of resources. This is exemplified with three χDL steps in the synthesis of 2-Methyl-4-(4-nitrophenyl)but-3-yn-2-ol (Fig. 7 *A* and *B*). Additionally, as illustrated in Fig. 7*C* using the example of atropine synthesis, the total number of required χDL steps increases approximately linearly with the number of reaction steps (20, 34, 47). Each synthesis step can be fully captured using our CSTM abstraction (compare *SI Appendix*, Fig. S3 for a worked example). This is further demonstrated in Fig. 7*D* with two single-step syntheses—2-(3,4-dimethoxyphenyl)-1*H*-indole and 2-Methyl-4-(4-nitrophenyl)but-3-yn-2-ol which require 18 and 13 χDL/CSTM abstraction steps, respectively. The analysis of 117 physically executed synthesis routes, covering a broad and representative range of chemical transformations, confirms the practical utility and generalizability of our abstraction for driving automated synthesis platforms and self-driving laboratories (Fig. 7*E*). These findings also highlight that even complex, multistep syntheses can be expressed through a linearly scaling number of unit operations as defined in section 3 (see *SI Appendix*, Table S1 for classification of χDL steps). Importantly, this analysis covers a range of synthetic classes, such as protecting- and functional-group manipulations, (un)catalyzed C–C bond formations, heteroatom alkylation and arylation, and ring and heterocycle formation. Reactions were performed under varying conditions, with experiment times from approximately 5.5 h to over 62 h and reaction temperatures ranging from –45 °C to 140 °C. Additionally, the synthesis of gold nanoparticles (54, 55), POM-MOFs (56), and MOFs (55) has been demonstrated in previous works highlighting the universality of this approach.

Practical Limitations

Implementing the concept of chemputation in practice presents a series of significant challenges that extend beyond this robust theoretical framework. One of the foremost challenges lies in the

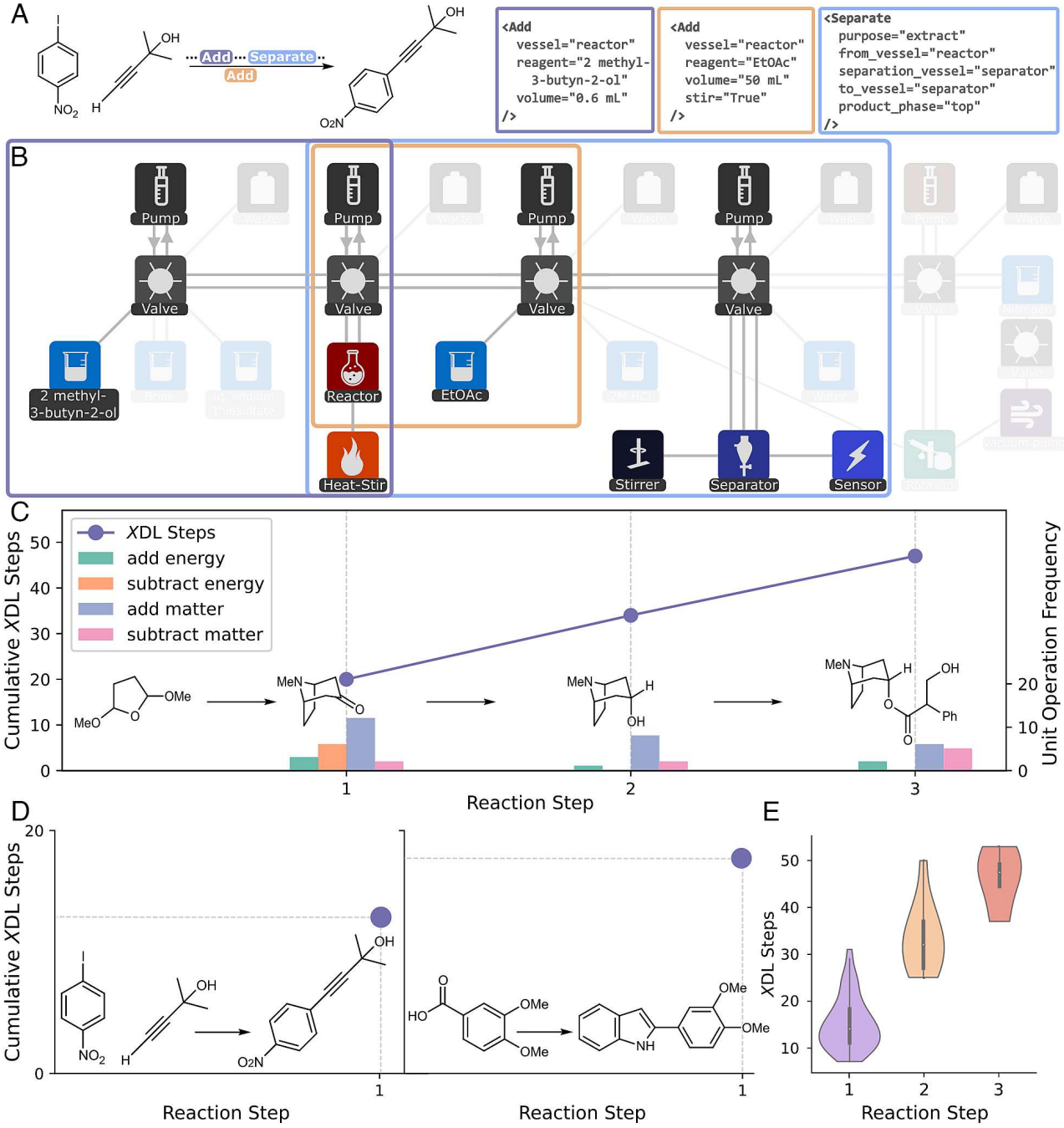

Fig. 7. Graphs showing the scaling behaviour of χDL steps, according to our CSTM abstraction, as a function of reaction steps. (*A*) Reaction schema of 2-Methyl-4-(4-nitrophenyl)but-3-yn-2-ol and three exemplary χDL steps from the respective synthesis. (*B*) Abstract chemputer graph showing the actively addressed component for three χDL steps, as indicated in (*A*). (*C*) Scaling behaviour of the cumulative number of χDL steps for the three-step synthesis of Atropine. The distribution of unit operations per reaction step is shown on the secondary y-axis. (*D*) Number of χDL steps for the two single-step reactions of 2-Methyl-4-(4-nitrophenyl)but-3-yn-2-ol, and 2-(3,4-dimethoxyphenyl)-1*H*-indole. (*E*) Distribution of the cumulative number of χDL steps for single-step, two-step, and three-step reactions, respectively.

complexity and scalability of the chemputer's hardware. The concept of universally configurable hardware, which is central to the chemputer's ability to synthesize any chemical compound, demands a highly versatile and flexible system with a range of different modules for operations such as filtration and extraction. Designing hardware that can seamlessly switch between different configurations for a wide variety of chemical processes is an intricate task. Each module within the system must handle diverse reaction types, process conditions, and scales of operation while maintaining precision and reliability. Moreover, there is an inherent tension between the need for miniaturization, which allows for precision, and the requirement for scalability to manage larger volumes or more complex reactions. Achieving both in a single system, particularly one that remains flexible and configurable, is a significant engineering challenge. Furthermore, the integration of this hardware with the software responsible for the chempiling function—mapping synthesis pathways to specific hardware configurations—adds another layer of complexity. This software must dynamically adjust the hardware setup in real-time, requiring a level of synchronization and control that is difficult to achieve.

One critical point introduced in this work is the expansion of the definition of a molecule by explicitly linking its synthetic accessibility (57–59) to the intrinsic complexity of its molecular graph, quantified through the concept of the assembly index (40, 42, 43). Traditionally, chemists have viewed synthetic complexity through heuristic measures, such as the number of reaction steps or yield, without a

fundamental underlying theoretical framework. While mapping synthetic complexity onto an abstract concept like the assembly index might initially seem nonintuitive, doing so provides a rigorous theoretical grounding. Specifically, the assembly index measures the minimal number of causal steps or constraints required to produce a molecule, giving a precise quantification of molecular complexity directly related to synthesis. By establishing a clear link between molecular complexity and synthesis, this approach effectively sets an absolute lower bound on the number of parameters needed to orchestrate and control the synthetic process. Thus, rather than relying purely on empirical knowledge or reaction heuristics, chemists can now use the assembly index to systematically define, compare, and predict the synthetic accessibility and complexity of molecules. This theoretical advance not only facilitates better predictive models for chemical synthesis but also deepens the fundamental understanding of molecular construction, bridging the gap between theoretical complexity measures and practical chemical synthesis.

Another critical challenge is the implementation of DEC within the chemputer, which is essential for ensuring the accuracy and reliability of chemical syntheses. The system must be capable of real-time monitoring and adjustment, continuously tracking the progress of each reaction, detecting any deviations from the expected pathway, and applying corrective measures immediately. Recently we have shown that systems with inline sensors and dynamic χDL can be used for both error correction and optimization (46). To achieve this demanded advanced sensing technologies and real-time data processing capabilities that can operate effectively across a broad range of reaction conditions. However, in multistep syntheses, errors can propagate through the system, compounding and becoming more difficult to correct as the process continues. Developing mechanisms that can effectively manage and contain such errors, ensuring the robustness and redundancy of the system, is crucial. Achieving this balance between robustness, cost, space, and energy efficiency poses a significant challenge. The theoretical framework also assumes a comprehensive understanding of the chemical space and the ability to encode all possible reactions into the chemputer. However, the reality of chemical synthesis is more complex. Our current knowledge of chemical reactions is not exhaustive, particularly in the fields of complex organic and biological chemistry, where many reactions remain poorly understood or unpredictable. This limitation restricts the chemputer's ability to reliably handle all potential syntheses. Moreover, as complex molecules are synthesized, emergent properties may arise that are not predicted by existing models, leading to unexpected reactions or products. The chemputer must be designed to manage and correct such deviations, even in the face of novel or poorly understood chemistry. Developing algorithms and hardware that can adapt to new chemical data in real-time is a significant hurdle that must be overcome.

Conclusions

Since the chemputer, as seen as the CSTM, can implement any transformation function δ and can control all relevant process conditions, it can instantiate any chemical synthesis process. The inclusion of dynamic error detection and correction at each step ensures the reliability and accuracy of the synthesis. Additionally, the use of universally configurable hardware and the chempiling function allows the chemputer to dynamically adapt its configuration for various synthesis pathways. Thus, the chemputer is universal for chemical synthesis, capable of generating any compound $c \in C$ given the appropriate initial conditions, transformations, and error correction mechanisms. The formalization above establishes the concept of a chemputer as a universal chemical synthesis machine. The CSTM transformation function δ, synthesis pathways σ, stability conditions S, DEC, chempiling function χ, and configurable hardware U together define a universal model capable of synthesizing any target compound within the chemical space defined by R, P. The work presented here establishes the chemputer as a universal chemical synthesis machine, demonstrating its capability to synthesize any target compound within a defined chemical space by the process of chemputation. In addition, the architecture and the use of χDL has been achieved on at least 12 different platforms with many different types of robotic modules across our (6, 11, 54, 56, 60–63) and collaborative research teams (28, 55, 64, 65). By formalizing the key components, such as the transformation function δ, synthesis pathways σ, stability conditions S, DEC, and the chempiling function χ, we have constructed a robust theoretical framework that underpins this universality. The integration of universally configurable hardware further enhances the chemputer's adaptability, allowing it to dynamically reconfigure and execute a wide array of chemical processes with precision. This is universal considering finite constraints on the reaction hardware, reagents, reaction steps, reaction time, and the ability to produce the target molecule in a detectable amount.

Materials and Methods

The exploration of the CSTM is outlined in the Mathematica notebook which describes the ontology for a general chemical reaction and the implementation on a 1D state including the exploration of the rules using a color tape showing the transitions which are also plotted as transition state diagrams (*SI Appendix*). The exploration of the error rate propagation was done using a Monte-Carlo approach generating a set of trajectories for the error in each step showing an exponential growth in error as a function of the number of steps and exploring the impact on the copy number of the molecules produced. Using the CSTM model, we analyzed experimental robotic chemistry data from the chemputer to quantify the scaling of the χDL steps across the real world chemical synthesis done.

Data, Materials, and Software Availability. Analytical data (66) and protocols (67) including data from ref. 8 have been deposited in databases available online (66, 67); Movie S1 shows the CSTM. All other data are included in the manuscript and/or supporting information.

ACKNOWLEDGMENTS. We would like to thank David Deutsch, Muffy Calder, Sara Walker, S. Hessam Mehr, Keith Patarroyo, Emma Clarke, Dario Caramelli, David Cleevely, and Edward Lee for feedback and Niclas Grocholski for help with figure 4. We acknowledge financial support from the John Templeton Foundation (grants 61184 and 62231), Sloan Foundation, Schmidt Futures, NIH, Google, and EPSRC (grant nos. EP/L023652/1, EP/R01308X/1, EP/S019472/1, and EP/P00153X/1 and ERC (project 670467 SMART-POM).

1. A. Church, A note on the entscheidungsproblem. *J. Symb. Log.* **1**, 40–41 (1936).
2. A. M. Turing, On computable numbers, with an application to the Entscheidungsproblem. *Proc. Lond. Math. Soc.* **s2-42**, 230–265 (1937).
3. A. M. Turing, Computing machinery and intelligence. *Mind* **59**, 433–460 (1950).
4. P. J. Kitson, S. Glatzel, L. Cronin, The digital code driven autonomous synthesis of ibuprofen automated in a 3D-printer-based robot. *Beilstein J. Org. Chem.* **12**, 2776–2783 (2016).
5. S. Steiner *et al.*, Organic synthesis in a modular robotic system driven by a chemical programming language. *Science* **363**, eaav2211 (2019).
6. J. S. Manzano *et al.*, An autonomous portable platform for universal chemical synthesis. *Nat. Chem.* **14**, 1311–1318 (2022).
7. S. H. M. Mehr, M. Craven, A. I. Leonov, G. Keenan, L. Cronin, A universal system for digitization and automatic execution of the chemical synthesis literature. *Science* **370**, 101–108 (2020).
8. S. Rohrbach *et al.*, Digitization and validation of a chemical synthesis literature database in the ChemPU. *Science* **377**, 172–180 (2022).
9. P. S. Gromski, J. M. Granda, L. Cronin, Universal chemical synthesis and discovery with 'The Chemputer'. *Trends Chem.* **2**, 4–12 (2020).

10. L. Wilbraham, S. H. M. Mehr, L. Cronin, Digitizing chemistry using the chemical processing unit: From synthesis to discovery. *Acc. Chem. Res.* **54**, 253–262 (2021).
11. R. Rauschen, M. Guy, J. E. Hein, L. Cronin, Universal chemical programming language for robotic synthesis repeatability. *Nat. Synth.* **3**, 488–496 (2024).
12. R. B. Merrifield, J. M. Stewart, Automated peptide synthesis. *Nature* **207**, 522–523 (1965).
13. A. Isidro-Llobet, M. Alvarez, F. Albericio, Amino acid-protecting groups. *Chem. Rev.* **109**, 2455–2504 (2009).
14. A. Hoose, R. Vellacott, M. Storch, P. S. Freemont, M. G. Ryadnov, DNA synthesis technologies to close the gene writing gap. *Nat. Rev. Chem.* **7**, 144–161 (2023).
15. T. Jiang *et al.*, Automated and parallel amide synthesis. *Eur. J. Org. Chem.* **28**, e202401258 (2025).
16. B. J. Reizman, Y.-M. Wang, S. L. Buchwald, K. F. Jensen, Suzuki-miyaura cross-coupling optimization enabled by automated feedback. *React. Chem. Eng.* **1**, 658–666 (2016).
17. B. Wathen, E. Lanehart, L. A. Woodis, A. J. Rojas, Buchwald-hartwig amination, high-throughput experimentation, and process chemistry: An introduction via undergraduate laboratory experimentation. *J. Chem. Educ.* **98**, 996–1000 (2021).
18. A. Sugimoto, T. Fukuyama, Md. T. Rahman, I. Ryu, An automated-flow microreactor system for quick optimization and production: Application of 10- and 100-gram order productions of a matrix metalloproteinase inhibitor using a Sonogashira coupling reaction. *Tetrahedron Lett.* **50**, 6364–6367 (2009).
19. A. C. Sun *et al.*, A droplet microfluidic platform for high-throughput photochemical reaction discovery. *Nat. Commun.* **11**, 6202 (2020).
20. A. L. Satz *et al.*, DNA-encoded chemical libraries. *Nat. Rev. Methods Primers* **2**, 1–17 (2022).
21. G. Schneider, Automating drug discovery. *Nat. Rev. Drug Discov.* **17**, 97–113 (2018).
22. J. Britton, T. F. Jamison, The assembly and use of continuous flow systems for chemical synthesis. *Nat. Protoc.* **12**, 2423–2446 (2017).
23. R. P. Wierenga, S. M. Golas, W. Ho, C. W. Coley, K. M. Esvelt, Pylabrobot: An open-source, hardware-agnostic interface for liquid-handling robots and accessories. *Device* **1**, 100111 (2023).
24. M. Golyadkin, V. Pozdnyakov, L. Zhukov, I. Makarov, Sensorscan: Self-supervised learning and deep clustering for fault diagnosis in chemical processes. *Artif. Intell.* **324**, 104012 (2023).
25. T. Hardwick, N. Ahmed, Digitising chemical synthesis in automated and robotic flow. *Chem. Sci.* **11**, 11973–11988 (2020).
26. G. Tom *et al.*, Self-driving laboratories for chemistry and materials science. *Chem. Rev.* **124**, 9633–9732 (2024).
27. C. W. Coley *et al.*, A robotic platform for flow synthesis of organic compounds informed by AI planning. *Science* **365**, eaax1566 (2019).
28. F. Strieth-Kalthoff *et al.*, Delocalized, asynchronous, closed-loop discovery of organic laser emitters. *Science* **384**, eadk9227 (2024).
29. N. Collins *et al.*, Fully automated chemical synthesis: Toward the universal synthesizer. *Org. Process Res. Dev.* **24**, 2064–2077 (2020).
30. Y. Shen *et al.*, Automation and computer-assisted planning for chemical synthesis. *Nat. Rev. Methods Primers* **1**, 23 (2021).
31. E. Fournier *et al.*, Combining real-time monitoring using raman spectroscopy, rotating-bed reactors, and green solvents to improve sustainability in solid-phase peptide synthesis. *ACS Sustain. Chem. Eng.* **12**, 14629–14637 (2024).
32. D. E. Fitzpatrick, C. Battilocchio, S. V. Ley, A novel internet-based reaction monitoring, control and autonomous self-optimization platform for chemical synthesis. *Org. Process Res. Dev.* **20**, 386–394 (2016).
33. P. Sagmeister *et al.*, Advanced real-time process analytics for multistep synthesis in continuous flow. *Angew. Chem. Int. Ed. Engl.* **60**, 8139–8148 (2021).
34. M. Abolhasani, E. Kumacheva, The rise of self-driving labs in chemical and materials sciences. *Nat. Synth.* **2**, 483–492 (2023).
35. B. Burger *et al.*, A mobile robotic chemist. *Nature* **583**, 237–241 (2020).
36. T. Dai *et al.*, Autonomous mobile robots for exploratory synthetic chemistry. *Nature* **635**, 890–897 (2024).
37. D. A. Boiko, R. MacKnight, B. Kline, G. Gomes, Autonomous chemical research with large language models. *Nature* **624**, 570–578 (2023).
38. J. Liu, J. E. Hein, Automation, analytics and artificial intelligence for chemical synthesis. *Nat. Synth.* **2**, 464–466 (2023).
39. A.-M. Anhel, L. Alejaldre, Á. Goñi-Moreno, The laboratory automation protocol (LAP) format and repository: A platform for enhancing workflow efficiency in synthetic biology. *ACS Synth. Biol.* **12**, 3514–3520 (2023).
40. A. Sharma *et al.*, Assembly theory explains and quantifies selection and evolution. *Nature* **622**, 321–328 (2023).
41. S. M. Marshall, D. G. Moore, A. R. G. Murray, S. I. Walker, L. Cronin, Formalising the pathways to life using assembly spaces. *Entropy* **24**, 884 (2022).
42. S. M. Marshall *et al.*, Identifying molecules as biosignatures with assembly theory and mass spectrometry. *Nat. Commun.* **12**, 3033 (2021).
43. M. Jirasek *et al.*, Investigating and quantifying molecular complexity using assembly theory and spectroscopy. *ACS Cent. Sci.* **10**, 1054–1064 (2024).
44. D. Deutsch, Constructor theory. *Synthese* **190**, 4331–4359 (2013).
45. H. M. Al-Hashimi, Turing, von Neumann, and the computational architecture of biological machines. *Proc. Natl. Acad. Sci. U.S.A.* **120**, e2220022120 (2023).
46. A. I. Leonov *et al.*, An integrated self-optimizing programmable chemical synthesis and reaction engine. *Nat. Commun.* **15**, 1240 (2024).
47. J. M. Granda, L. Donina, V. Dragone, D.-L. Long, L. Cronin, Controlling an organic synthesis robot with machine learning to search for new reactivity. *Nature* **559**, 377–381 (2018).
48. B. J. Reizman, K. F. Jensen, Feedback in flow for accelerated reaction development. *Acc. Chem. Res.* **49**, 1786–1796 (2016).
49. Chemistry (IUPAC), T. I. U. of P. and A. IUPAC—Molecule (M04002) (2025). https://doi.org/10.1351/goldbook.M04002. Accessed 1 June 2025.
50. J. S. Carey, D. Laffan, C. Thomson, M. T. Williams, Analysis of the reactions used for the preparation of drug candidate molecules. *Org. Biomol. Chem.* **4**, 2337 (2006).
51. N. Schneider, D. M. Lowe, R. A. Sayle, M. A. Tarselli, G. A. Landrum, Big data from pharmaceutical patents: A computational analysis of medicinal chemists' bread and butter. *J. Med. Chem.* **59**, 4385–4402 (2016).
52. N. I. Vasilevich, R. V. Kombarov, D. V. Genis, M. A. Kirpichenok, Lessons from natural products chemistry can offer novel approaches for synthetic chemistry in drug discovery: Miniperspective. *J. Med. Chem.* **55**, 7003–7009 (2012).
53. S. D. Roughley, A. M. Jordan, The medicinal chemist's toolbox: An analysis of reactions used in the pursuit of drug candidates. *J. Med. Chem.* **54**, 3451–3479 (2011).
54. Y. Jiang *et al.*, An artificial intelligence enabled chemical synthesis robot for exploration and optimization of nanomaterials. *Sci. Adv.* **8**, eabo2626 (2022).
55. L. Huang *et al.*, Natural-language-interfaced robotic synthesis for AI-copilot-assisted exploration of inorganic materials. *J. Am. Chem. Soc.* **147**, 23014–23025 (2025).
56. D. He *et al.*, Algorithm-driven robotic discovery of polyoxometalate-scaffolding metal–organic frameworks. *J. Am. Chem. Soc.* **146**, 28952–28960 (2024).
57. C. W. Coley, L. Rogers, W. H. Green, K. F. Jensen, SCScore: Synthetic complexity learned from a reaction corpus. *J. Chem. Inf. Model.* **58**, 252–261 (2018).
58. A. Thakkar, V. Chadimová, E. Jannik Bjerrum, O. Engkvist, J.-L. Reymond, Retrosynthetic accessibility score (RAscore) – Rapid machine learned synthesizability classification from AI driven retrosynthetic planning. *Chem. Sci.* **12**, 3339–3349 (2021).
59. H. Kim, K. Lee, C. Kim, J. Lim, W. Y. Kim, DFRscore: Deep learning-based scoring of synthetic complexity with drug-focused retrosynthetic analysis for high-throughput virtual screening. *J. Chem. Inf. Model.* **64**, 2432–2444 (2024).
60. S. Pagel, M. Jirasek, L. Cronin, Validation of the scientific literature via chemputation augmented by large language models. arXiv [Preprint] (2024). https://doi.org/10.48550/arXiv.2410.06384 (Accessed 27 June 2025).
61. K. Laws *et al.*, An autonomous electrochemical discovery robot that utilises probabilistic algorithms: Probing the redox behaviour of inorganic materials. *ChemElectroChem* **11**, e202300532 (2024).
62. N. L. Bell *et al.*, Autonomous execution of highly reactive chemical transformations in the Schlenkputer. *Nat. Chem. Eng.* **1**, 180–189 (2024).
63. M. Šiaučiulis, C. M. Knittl-Frank, S. H. Mehr, E. Clarke, L. Cronin, Reaction blueprints and logical control flow for parallelized chiral synthesis in the chemputer. *Nat. Commun.* **15**, 10261 (2024).
64. N. Yoshikawa *et al.*, Large language models for chemistry robotics. *Auton. Robot.* **47**, 1057–1086 (2023).
65. K. Darvish *et al.*, ORGANA: A robotic assistant for automated chemistry experimentation and characterization. *Matter* **8**, 101897 (2025).
66. L. Cronin *et al.*, Code and Data for Digitization and validation of a chemical synthesis literature database in the ChemPU. Zenodo. https://zenodo.org/records/6534009. Deposited 16 March 2026.
67. L. Cronin, S. Pagel, A. Sharma, Chemputation. Zenodo. https://zenodo.org/records/19046308. Deposited 9 January 2026.