Database-Guided Program Synthesis of Chemical Reaction Networks

Where Reaction-Database Knowledge is Most Effective in Reducing Search

Tymon Jastrzemski[1]

Supervisor(s): Sebastijan Dumančić[1], Reuben Gardos Reid[1]

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

Name of the student: Tymon Jastrzemski
Final project course: CSE3000 Research Project
Thesis committee: Sebastijan Dumančić, Reuben Gardos Reid, Jana M. Weber

Abstract

Recovering a chemical reaction network (CRN) from concentration data can be framed as program synthesis, but the search scales poorly: reaching the ground-truth network requires enumerating a very large number of candidates. Hard constraints on the synthesizer's grammar prune candidates by chemical rules such as atom valence and mass balance, yet among the valid candidates a uniform search has no sense of which reactions are plausible. We ask where in a top-down CRN synthesizer knowledge from a reaction database most reduces the candidates explored before the target is found? We compare three integration points across five benchmarks, using the USPTO-50K and Rhea databases: (1) a database-derived building-block vocabulary, (2) a probabilistic context-free grammar (PCFG) over the grammar rules, and (3) an output reranker. The building-block vocabulary is the most positive result: it decides whether the target network is recovered at all, and a shuffle control attributes this to the database's frequency content rather than vocabulary size. Each corpus unlocks only the chemistry it covers, and rank-normalised merging recovers targets that a naive union dilutes. The PCFG adds ordering gain, while the reranker, acting only on the finished list, helps under corpus match and hurts under mismatch. Matching a corpus to the target chemistry, not enlarging it, is what turns a reaction database into useful search guidance.

1 Introduction

We describe chemical systems by how the concentration of their species rise and fall. A chemical reaction network (CRN) is a mathematical model defined by a set of species, reactions and rate constants. A CRN fully determines how a chemical system evolves over time, and recovering this network from data has important applications in pharmaceutical development, atmospheric modelling, and combustion engineering [1]. The hard problem is the reverse direction, from observed concentration trajectories, trying to recover: the species, reactions, and rate constants that explain it. In practice this is largely done by hand; it is labour-intensive and relies on expert intuition and knowledge [2]. An automated synthesizer that proposes mechanism directly from data would change how chemists generate and test hypotheses.

CRN discovery has been framed as a program-synthesis problem, with a grammar that defines the space of valid networks and a search mechanism that enumerates candidates. Wijers [3] builds on this idea: layered context-free grammars over molecules, reactions and networks describe the candidate space and a top-down enumerative search explores it, simulating each candidate and scoring against the observed trajectory. The synthesizer scales poorly; reaching the target requires enumerating a very large number of candidates.

The bottleneck is the number of candidates the search must examine before it reaches the target network. Hard syntactic constraints, rules enforcing chemical validity such as atom valence and reaction mass balance, prune invalid candidate networks (Section 2). Constraints alone cannot express the fact that, among the valid candidates, some species and reactions are more likely to occur in chemical reactions than others. A uniform grammar treats every production rule as equally likely, so the search does not prioritise plausible mechanisms over implausible ones. Knowledge of which reactions actually occur is precisely what a reaction database records. Program synthesis literature proposes a probabilistic context-free grammar (PCFG) as a way to bias the search using a database. [4].

A PCFG, however, is only one of several places database knowledge can help. Database knowledge can shape (i) the **terminal vocabulary** of building-block molecules, which species can appear in a candidate at all; (ii) the grammar's **production-rule weights**, the order in which candidates are visited; and (iii) the **scoring of finished candidates**, the order of the completed candidate list.

To what extent, and at which point in a top-down CRN program-synthesis pipeline, does knowledge from a reaction database reduce the number of candidates explored before the target network is found?

We decompose this into three sub-questions:

1. **Q1 (Integration point)** At which integration point does database knowledge most reduce candidates-to-target: the production-rule probabilities of a PCFG, the building-block vocabulary, or output reranking?
2. **Q2 (Content vs. size)** Is any reduction driven by the database's frequency **content**, or merely by a larger or different vocabulary?
3. **Q3 (Corpus mismatch)** How does mismatch between the corpus and the target chemistry affect both the reduction of the search space and whether the search can still reach the target reaction?

We evaluate on Wijers' benchmarks, adding a Diels–Alder reaction as a corpus mismatch case. The primary metric is candidates-to-target: the number of candidates the search explores before it reaches the target network. Database knowledge comes from two corpora, USPTO-50K [5] (largely organic patent chemistry) and Rhea [6] (inorganic and small-molecule reactions). The contributions are:

1. **A database-derived building-block vocabulary for an enumerative CRN synthesizer, with evidence that recovering the target network is determined by the frequency content of the corpus used, not vocabulary size**

2. **A PCFG over the molecule grammar and reaction stoichiometry, using a most-likely-first iterator, giving a reduction in candidates-to-target**
3. **A cross-database study of corpus mismatch, showing that each corpus unlocks targets it covers and that merging corpora can hurt rather than help+ A reranker that transfers similarity-based reranking from retrosynthesis to the synthesizer's output, improving rank under corpus match and hurting it under mismatch**

The remainder of the paper is organised as follows. Section 2 reviews enumerative program synthesis, PCFGs, the Herb.jl framework and Wijers' staged synthesizer, chemical reaction networks, and reaction databases and fingerprints. Section 3 describes the three integration points (the building-block vocabulary, the PCFG and its most-likely-first enumeration, and the reranker) and the experimental design. Section 4 reports the building-block, cross-database, shuffle-control, and PCFG results across the benchmarks. Section 5 discusses reproducibility and the ethical implications. Section 6 answers the research question and its sub-questions and lists recommendations, and Section 7 concludes.

2 Background and Related Work

This section introduces the concepts used throughout the paper: enumerative program synthesis, PCFGs as search priors, the Herb.jl framework and the staged synthesizer Wijers built on top of it, chemical reaction networks, and the reaction databases and fingerprints that supply the prior.

2.1 Program synthesis and enumerative search

Program synthesis is the task of automatically producing a program in a domain-specific language (DSL) that satisfies a given specification in this work a concentration trajectory. Wijers' synthesizer [3] uses top-down enumerative search: starting from the grammar's start symbol, it repeatedly expands non-terminals and prunes partial derivations using syntactic constraints.

For example, given measured concentration curves of hydrogen, oxygen and water, the synthesizer enumerates networks, simulates each one, and returns 2 H_2 + O_2 -> 2 H_2O because its simulated trajectory reproduced the data. The specification is the observed trajectory, the program is the reaction network, and the DSL is the grammar of chemical reactions.

The common idea this work builds on is the search prior. Uniform-cost enumeration treats every production rule as equally likely. A non-uniform prior reorders enumeration so that more plausible derivations are visited first; in a setting where the target program appears once and the search is exhaustive, this directly reduces the candidates-to-target metric, number of candidates explored before reaching the target network.

Taking methane combustion as an example, carbon, hydrogen and oxygen combine into many chemically valid molecules and reactions, only a few of which occur in combustion in nature. Uniform enumeration visits them in an order determined by the grammar; a prior estimated from a corpus of combustion reactions pushes it toward the front and lowers the amount of candidates explored until reaching the target network.

2.2 Probabilistic context-free grammars as priors

A probabilistic context-free grammar (PCFG) attaches a probability $p(r)$ to each production rule r. The probability of a derivation is the product of the probabilities of the rules it uses, and the corresponding cost used by the search is $-\log p$. Enumerating derivations in order of increasing cost is equivalent to most-likely-first search, the pattern used by every PCFG-guided synthesizer cited below.

Multiple works frame the design choices made in Section 3. Multiple PCFG-guided synthesizers used the rule-weight prior: Menon et al.'s PCFG-with-features [4], neural and PHOG-conditioned weights [7], [8]. Most relevant here, HYSYNTH [9] fits a PCFG to LLM samples and uses it to guide a bottom-up enumerative synthesizer. The standard estimation recipe we adopt in Section 3 (count rule occurrences in a corpus, Laplace-smooth, keep zero-support rules off the cost floor, and inline deterministic rules) is the one HYSYNTH uses, applied here at the level of reactions and networks.

The closest chemistry-adjacent precedent is Brence et al. [10], who use a PCFG to encode a parsimony prior on symbolic equation discovery. This work extends this from equations to chemical reactions. Where we apply the recipe (Section 3) we weight the molecule-grammar rules and the reaction grammar's stoichiometry, leaving the network grammar uniform and keeping estimation to the levels at which the corpus supplies reliable counts.

Estimating a PCFG from a corpus has pitfalls: (a) maximum-likelihood weights collapse to zero for unseen rules, requiring smoothing and a probability floor; (b) some rules are deterministic ($p = 1$) and should be inlined rather than scored; (c) atom-mapping (reactant to product atom correspondence) noise in the source corpus propagates into rule frequencies; and (d) any corpus has a coverage bias, USPTO-50K is dominated by organic chemistry, we complement it with Rhea for inorganic reactions. There are reactions not covered be either like the Diels-Alder cycloaddition.

2.3 Herb.jl and the staged CRN synthesizer

Herb.jl is a Julia framework for grammar-based program synthesis. In Herb.jl a grammar holds the production rules; a candidate program is a tree of nodes, there is one node per rule; a program enumarator is a search strategy that traverses these trees. A grammar can attach a probability to each rule; this is where a PCFG's weights are stored, and

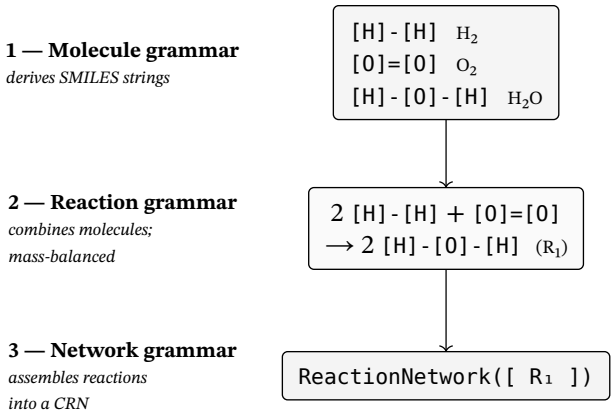

Figure 1: The synthesizer's staged grammar on the water benchmark ($2H_2 + O_2 \rightarrow 2H_2O$). **Layer 1** derives **SMILES strings**, text encodings of molecules, atom by atom, **Layer 2** combines them into a mass-balanced reaction. **Layer 3** assembles reactions into a network. Each layer's input is output of the previous.

the framework's most likely first enumerator expands them in decreasing order of probability. [11]

Wijers' synthesizer [3] composes three layered grammars (Figure 1). The **molecule grammar** derives SMILES strings (a text encoding of molecular structure) constrained by atom valence and ring-bond pairing. The **reaction grammar** derives reactions over molecules, with mass balance enforced and reactant/product order canonicalised. The **network grammar** derives sets of reactions.

2.4 Chemical reaction networks

Formally [12], a chemical reaction network is a pair (Λ, R) where Λ is a finite set of species and R is a set of reactions, each a triple (reactants, products, rate constant).The synthesizer uses the ODE semantics via Catalyst.jl [13] when scoring candidates against the observed trajectory. Rate constants are fitted during scoring.

Two facts about chemistry shape the design of this synthesizer. First, atom valence and mass balance are required: any candidate that violates them is not valid chemically, and the prototype enforces them as hard constraints. Second, **plausibility** matters. It is much harder to encode as a hard constraint, because it is a property defined relative to a class of similar reactions and enviroment the raction takes place in. This is precisely the gap a soft, corpus-derived prior is meant to fill: the PCFG re-weights the candidate by how often similar reactions appear in the database.

2.5 Reaction databases and reaction fingerprints

The vocabulary, the PCFG, and the reranker all depend on a reaction corpus. **USPTO-50K** is the 50,000-reaction, role-labelled subset [5] of the reactions text-mined from US patents by Lowe [14]; each is atom-mapped (each product atom is annotated with the reactant atom it originates from) and encoded as an integer map on the SMILES. Because these labels are assigned automatically they are not always correct. We complement it with **Rhea** [6], an expert-curated database of biochemical and small-molecule reactions, which covers inorganic and gas-phase chemistry that USPTO does not.

Corpus coverage bias is the main external risk this work must confront: USPTO-50K is heavily skewed toward organic chemistry, while two of the four benchmarks (water formation, methane combustion) are inorganic. In USPTO-50K the species CH_4, O_2 and CO_2 never appear and water ranks only 123rd by frequency and in Section 4 we note which benchmarks are corpus-matched and which are not.

A reaction fingerprint is a vector encoding of a reaction usable for similarity search. Extended-Connectivity Fingerprint (ECFP)-difference fingerprints are the historical baseline [15]; learned reaction fingerprints such as rxnfp [16] obtain strong classification scores but require training. The **differential reaction fingerprint** (DRFP) [17] is hash-based, requires no atom mapping and no training, and reaches 93–99% of rxnfp's performance on reaction classification (benchmarked on USPTO dataset). We use it for the output reranker of with radius $r = 2$, dimensionality $d = 2048$, mean Tanimoto similarity ($|a \cap b| \ / \ |a \cup b|$) to the top $k = 5$ retrieved reactions, and it feeds neither the PCFG nor the vocabulary.

3 Method

This section describes the three places where reaction-database knowledge is injected into Wijers' top-down synthesizer, and the experimental design. We build on the existing prototype [3] without changing its grammars, constraints, scoring function, or simulation back-end; every mechanism below is an addition on top.

3.1 Overview

Database knowledge enters the pipeline at three integration points (Figure 2), ordered by how early they act on the search. (1) The **building-block vocabulary** changes the terminal molecules a candidate may contain. (2) A **PCFG** over the grammar rules sets the cost of each production

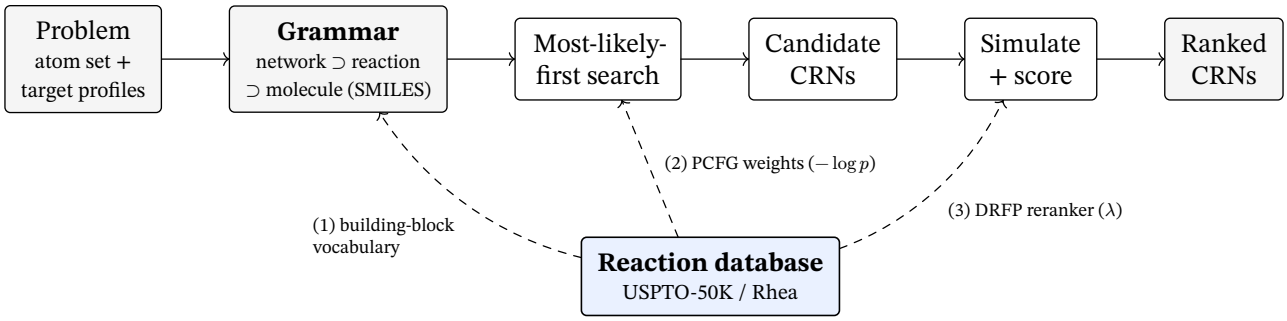

Figure 2: The CRN synthesizer and the three points where reaction database knowledge is injected. The grammar is staged: a **molecule** grammar builds SMILES species, a **reaction** grammar combines species into balanced reactions, and a **network** grammar combines reactions into a CRN. Hard chemical constraints (valence, mass balance) prune the grammar; the database enters only as soft guidance at (1) the molecule-level building-block vocabulary, (2) the PCFG that orders enumeration, and (3) the DRFP reranker acting on final list.

and therefore the order in which reachable candidates are visited. (3) An output **reranker** reorders the finished candidate list using reaction-similarity to the corpus. The first acts on the molecule stage, the second on the enumeration order across all stages, and the third only after enumeration.

3.2 Building-block vocabulary

In the reaction grammar, every reactant and product is drawn from a fixed library of terminal molecules. Whatever molecules the molecule search emits first become the building blocks for the entire downstream reaction and network-level search. The vocabulary has to have all needed molecules: if a species the target needs is not in the pool, no amount of search can recover the target.

We compare three vocabularies that differ in where the terminal molecule comes from. The **atom-level** baseline is Wijers' original molecule grammar, which derives SMILES from atomic terminals. The **oracle** vocabulary is the exact set of species appearing in the benchmark, an upper bound on what a perfect prior could supply. The **database-derived** vocabulary is the top-N molecules by frequency in a corpus. To build it we parse the corpus with RDKit, strip atom-map numbers, canonicalise each molecule to a single SMILES, count how many reactions each canonical molecule appears in, take the N most frequent, and deduplicate by canonical SMILES; the identical pipeline runs on USPTO-50K [5] and on Rhea [6]. The resulting building blocks replace the atom-level terminals, with a fallback mechanism that defaults to atom-level deriviations when not every molecule is found in the vocabulary.

3.3 PCFG estimation

A PCFG attaches a probability to each production rule. The search uses cost $-\log p$, so most-likely-first enumeration visits high-probability derivations first. We estimate these probabilities from a corpus. The molecule grammar's rules and the reaction grammar's molecule list rule, which controls reaction stoichiometry, i.e. how many molecules participate, are weighted from corpus counts. Every network-grammar rule and the top-level Reaction rules are left uniform; weighting them is left to future work.

We weight three groups of rules, estimated from a corpus. A reaction-stoichiometry prior weights the reaction grammar's molecule list rule by how many molecules appear on a reaction side. A molecular-formula atom-bag prior weights the molecule grammar by how often a molecule of a certain atomic composition appears in the corpus, ignoring how the atoms are bonded. A per-block prior weights the building-block library itself. The estimation recipe is the standard corpus-estimated-PCFG one (Section 2): rule occurrences are counted, smoothed with Laplace $\alpha = 1$, given a probability floor of $\log(\mathrm{eps})$ so that unseen rules never produce a $-\infty$ cost that would corrupt the queue. The surface-statistics object is cached so that estimation runs once per corpus.

3.4 Most-likely-first enumeration in Herb.jl

Herb.jl already exposes probabilistic grammars and ships a most-likely-first iterator, which expands derivations in increasing $-\log p$ order [11]. We use it writing the estimated log-probabilities onto the grammars. Using MLFS only changes search strategy, given infinite compute we will get to the same networks.

3.5 DRFP-Tanimoto reranker

At the third integration point we rerank the finished candidate list by similarity to the corpus. Each candidate reaction is encoded with a differential reaction fingerprint (DRFP, radius $r = 2$, dimensionality $d = 2048$) [17]; its database-similarity score is the mean Tanimoto similarity to the $k = 5$ nearest corpus reactions. The reranker is integrated into the synthesizer's scoring function as an extra term weighted by λ, so that $\lambda = 0$ recovers Wijers' baseline. This transfers similarity-based reranking from retrosynthesis [18], [19] to a program-synthesis output. It acts only on the final enumerated list, it does not affect which networks appear.

4 Experimental Setup and Results

4.1 Benchmarks

We evaluate on the four benchmarks from Wijers [3] and add a Diels–Alder cycloaddition as a deliberate corpus-mismatch. Table 1 summarises the five and their coverage by each corpus. Water formation and methane combustion are inorganic; their species are essentially absent from USPTO but well represented in Rhea. Ethylene glycol formation and esterification are organic and matched to USPTO. Ethylene glycol, however, is infeasible to synthesize on 24 GB of ram so it is omitted from results tables. Esterification is the only two-reaction network. Diels–Alder sits outside both corpora by design to check what happens when corpus does not cover the target chemistry.

Table 1: Benchmarks and coverage of their species between both corpora. †Ethylene glycol is memory-infeasible to synthesize on our hardware (Section 6), so it yields no candidates-to-target on any configuration and is excluded from the measured results.

Benchmark	Target reaction	Class	USPTO	Rhea
Water	$2\,H_2 + O_2 \rightarrow 2\,H_2O$	inorganic	mismatch	match
Methane	$CH_4 + 2\,O_2 \rightarrow CO_2 + 2\,H_2O$	inorg. combustion	mismatch	match
Ethylene glycol†	$C_2H_4O + H_2O \rightarrow C_2H_6O_2$	organic	partial	mismatch
Esterification	2-reaction Fischer network	organic	match	mismatch
Diels–Alder	$C_4H_6 + C_2H_4 \rightarrow C_6H_{10}$	organic cycloaddition	mismatch	mismatch

4.2 Experimental design

Q1 (Integration point). At which integration point does database knowledge most reduce candidates-to-target: a PCFG's production-rule probabilities, the building-block vocabulary, or output reranking? Answered in Section 4.3, Section 4.5 and Section 4.6.

Q2 (Content vs. size). Is any reduction driven by the database's frequency content, or merely by a larger or different vocabulary? Answered in Section 4.4.

Q3 (Corpus mismatch). How does mismatch between the corpus and the target chemistry affect both the reduction of the search space and whether the search can still reach the target reaction? Answered in Section 4.3 and Section 4.4.

The primary metric is candidates-to-target, the index at which the target network first appears in the enumeration. The enumeration order is fixed, so every run should yield the same index. The comparison combines multiple experiments across different configurations each marked in its caption. Each design is described in full where its results are reported.

4.3 Building blocks

A database-derived building-block vocabulary changes which networks are reachable at all, and this is the largest effect we observe. Table 2 reports candidates-to-target for each benchmark under four vocabulary sources, holding other variables fixed.

Table 2: **Database content dictates what is reachable.** Candidates-to-target by building-block vocabulary source (PCFG-guided, `ContainsMolecules` on; the per-block weighting only orders the molecules and does not change what is reachable); "–" = not reached within budget

Benchmark	atom	USPTO	Rhea	∪ sum	∪ rank-norm
Water	6	6	6	5	5
Methane	–	–	**1**	**1**	**1**
Esterification	–	1077	–	–	**421**
Diels–Alder	–	–	–	–	–

Esterification is not reachable at the atom level, but with a USPTO building-block vocabulary at a pool of twenty molecules the target appears at enumeration index 1077 (scored rank 1). This is the first configuration in which raw database content makes an atom-level-unsolvable target reachable. A smaller pool of six fails, because formic acid (corpus count ≈ 28) never enters the top-N pool, so both the corpus frequency ranking that selects the pool and a large-enough pool are necessary. Methane combustion is reachable at the atom level under uniform BFS (index 6), but among the database vocabularies only Rhea reaches it: a Rhea building-block vocabulary makes it the very first candidate (index 1, robust across pool sizes from six to twenty, rising only to index 5 at thirty), while USPTO, lacking O_2 and CO_2, never does. Each corpus unlocks the chemistry it covers, while Diels–Alder, whose species are in neither corpus, is reached by none.

A union of USPTO and Rhea inherits Rhea's combustion reach (methane index 1) but still fails esterification, even though the combined run collected more candidates (1534) than the 1077 a single USPTO vocabulary needed. The cause is the raw count scale: Rhea's small-molecule counts (water ≈ 16 034) are much higher than USPTO's organic precursors (formic acid ≈ 28), so the union's top-N is dominated by Rhea and the esterification species are crowded out of the pool. Rank-normalising each corpus to a common percentile scale before taking the per-molecule maximum removes the dilution: the rank-normalised union reaches both targets from a single vocabulary (methane index 1, esterification index 421).

The **oracle** vocabulary, which supplies exactly the target's species as pure building blocks, sets the upper bound on what any vocabulary could provide. The pool then collapses to three to six molecules and the target appears at the top of the enumeration: index 1 for water, methane and Diels–Alder, and index 6 for the two-reaction esterification network (Table 3). The database-derived vocabularies remain far from this ceiling, esterification reaching index 1077 via USPTO blocks against an oracle index of 6, so a corpus-frequency vocabulary leaves headroom over a perfectly targeted one. And Diels–Alder, which no corpus

reaches (Table 2), is recovered at index 1 under the oracle, confirming that its failure is a corpus-coverage gap.

Table 3: **Oracle (best-case)** The exact target species are supplied as pure building blocks, giving a pool of three to six molecules; the target then appears at the top of the enumeration (index 1, or index 6 for the two-reaction esterification network), an upper bound on what any database-derived vocabulary could achieve. Indices identical under different search methods.

Benchmark	Pool size	Candidates-to-target
Water	3	**1**
Methane	4	**1**
Diels–Alder	3	**1**
Esterification	6	6

Finding the target is a property of vocabulary, not of how its molecules are weighted. Table 4 reports the same vocabularies with `ContainsMolecules` off, where the constraint no longer masks the effect: the Rhea vocabulary still reaches methane (idx 77) by supplying O_2 and CO_2 as blocks, while USPTO, which lacks them, never does. Consolidating every corpus for methane (Table 5), the **building-block vocabulary** reaches the target wherever the corpus supplies its species, whereas the **atom-bag prior**, working on the atom-level grammar with no injected blocks, fails on both single corpora even though Rhea contains O_2 abundantly. USPTO never reaches methane under any setting, a orpus mismatch; the rank-normalised union is the only corpus that reaches it across the board.

Table 4: **Building-block vocabulary across corpora, `ContainsMolecules` off.** Unlike the atom-bag prior, the database vocabulary reaches methane via Rhea (77) by supplying O_2/CO_2 as usable blocks; the rank-normalised union keeps both targets reachable (methane 106). The per-block weighting only orders these molecules; it does not change what is reachable.

Benchmark	USPTO	Rhea	∪ raw-sum	∪ rank-norm
Water	37	38	30	23
Methane	–	77	84	106
Diels–Alder	4	4	6	6

Table 5: **Methane across corpus × mechanism × constraint** (candidates-to-target, lower is better). "bag" = the atom-bag **prior** on the atom-level grammar; "block" = the database **building-block vocabulary**. Two invariants stand out: USPTO never reaches methane (unreached in all fours cells), a categorical corpus mismatch independent of mechanism; and the rank-normalised union is the only corpus that reaches methane in every cell, closing the one cell (bag prior, CM off) where plain Rhea fails (221). The vocabulary ("block") reaches methane wherever the corpus supplies O_2/CO_2.

Corpus	bag, CM on	bag, CM off	block, CM on	block, CM off
USPTO	–	–	–	–
Rhea	2	–	1	77
∪ raw-sum	2	–	1	84
∪ rank-norm	**2**	**221**	**1**	**106**

4.4 Database content or vocabulary size

The cross-database results could in principle be an artifact of vocabulary size, a database simply supplying more or different blocks, rather than of its frequency content. We separate the two with a random-vocabulary shuffle control that holds the block set, the pool size and the search budget fixed and changes only the mapping from molecules to their corpus counts (Table 6). If reachability survived the shuffle it would be a size effect. It does not. For methane with the Rhea vocabulary the real per-block ordering reaches the target at index 1, while all three shuffles (seeds 11, 22, 33) fail.

Table 6: **It is the database's frequency content, not vocabulary size.** The random-vocab control holds the block set, pool size and budget fixed and only permutes the molecule↔count mapping. Under real per-block ordering the pool contains every species the target needs; under any shuffle at least one required species is pushed out, so the target is unreachable **by construction**, a budget-independent result (the pool-composition diagnostic), confirming the causal driver is content.

Benchmark / vocab	Vocabulary ordering	Required species in pool?	Target idx
Methane (Rhea, n=10)	real per-block	O_2 ✓ · H_2O ✓	**1**
	shuffled (seed 11)	O_2 ✗ · H_2O ✓	–
	shuffled (seed 22)	O_2 ✗ · H_2O ✗	–
	shuffled (seed 33)	O_2 ✗ · H_2O ✗	–
Esterification (USPTO, n=20)	real per-block	formic ✓ · water ✓ · methanol ✓	**1077**
	shuffled (seed 11)	formic ✗ · water ✗ · methanol ✓	–
	shuffled (seed 22)	formic ✓ · water ✗ · methanol ✗	–

For esterification the shuffled runs collected fewer candidates than the 1077 the real ordering needed, so a raw candidate count is inconclusive here. Looking at pool composition allows us to check it without reference to the budget: under the real ordering the pool contains every species the target needs (O_2 and water for methane; formic acid, water and methanol for esterification); under every shuffle at least one required species is pushed out, so the target is unreachable by construction. The driver of reachability is therefore the database's frequency content, which species the corpus makes frequent enough to enter the pool, not the size or presence of the vocabulary.

4.5 PCFG-guided search

In this benchmark we disable ContainsMolecules constraint, to isolate the PCFG's ordering effect and to test hoe a corpus-learned prior performs against hard constraints. With ContainsMolecules off, the reaction-stoichiometry prior reduces candidates-to-target by 2.11× on water (40 → 19) and 1.63× on methane (363 → 223), as Table 7 shows. The constraint and the prior are complementary.

Table 7: Candidates-to-target under the 2×2 ablation (PCFG × `ContainsMolecules`); lower is better. The PCFG columns use the **reaction-stoichiometry (surface) prior with the molecular-formula atom-bag prior off**. "Reduction (−CM)" is BFS −CM / PCFG −CM, the PCFG's ordering gain with the constraint off; the Diels–Alder ratio is corpus-agnostic and not directly comparable to water/methane. The finer-grained atom-bag prior is the negative result reported in Table 8.

Benchmark	BFS +CM	PCFG +CM	BFS −CM	PCFG −CM	Reduction (−CM)
Water	6	5	40	19	2.11×
Methane	6	8	363	223	1.63×
Diels–Alder	1	1	507	2	253.50×

Pushing the PCFG to a finer grain gives negative results (Table 8). A molecular-formula ("atom-bag") prior over the molecule grammar erases water's gain (warse than baseline) and makes methane unreachable under. Easy explenation of corpus mismatch is not true, re-estimating the same atom-bag prior on Rhea, where O_2 is abundant (6590 occurrences, 3.25% of molecules), reproduces the failure, with water at idx 38 and methane still unreachable. An α-sweep over both corpora showed that less importance this prior gets we start to recover results (Figure 3): the USPTO curve buries water at idx 37–40 across small-to-moderate

α and relaxes to 18 only as $\alpha \to \infty$, and the Rhea curve is flatter still, holding at idx 38 from $\alpha = 0.01$ through $\alpha = 30$ despite Rhea's abundant O_2. All formula priors converge to the surface prior's idx 18–19 at large α, so the surface prior is simply the formula prior smoothed to uniform on the molecule grammar. Finer-grained formula weighting injects noise where the corpus signal is unreliable. In the same α-sweep on methane (Figure 3, b) both single-corpus atom-bag priors push methane out of the top-1500 entirely until α grows large, and only the rank normalised union finds it across the whole range.

Table 8: **Atom-bag (molecular-formula) prior across corpora, `ContainsMolecules` off.** The prior fails on BOTH USPTO and Rhea (water 37/38, methane unreached on both) even though Rhea contains O_2 (6590 occ, 3.25%). Rank-normalised union is the only configuration that recovers methane (221) and improves water (14 < the surface prior's 19).

Benchmark	USPTO	Rhea	∪ raw-sum	∪ rank-norm
Water	37	38	–	**14**
Methane	–	–	–	**221**
Diels–Alder	2	4	–	4

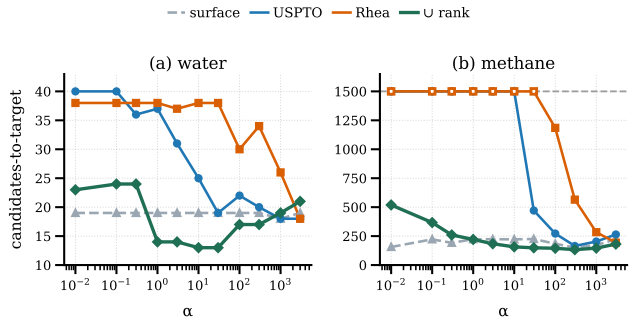

Figure 3: **α-sweep of the formula (molecule-grammar) priors** (`ContainsMolecules` off; y = candidates-to-target, lower is better; surface prior shown for reference; open markers = target not in the top-1500 budget). **(a) Water**: both single-corpus atom-bag priors bury water (idx 37–40) and go back to 18 only as $\alpha \to \infty$; the Rhea curve is flatter still (38) despite abundant O_2, so the hurt is prior formula, not coverage. **(b) Methane**: both single-corpus priors lose methane until α is large (USPTO from $\alpha \geq 30$, Rhea from $\alpha \geq 100$). Rank normalised prior recovers it but still worse than baseline. In both, all formula priors converge to the surface prior at large α.

The Diels–Alder entry in Table 7 (253.50×) must be read with care. It is the reaction-stoichiometry prior acting on a network with greater number of molecules, it still cannot recover the baseline rank 1; We do not treat it as comparable to the water and methane reductions.

4.6 DRFP reranker

The DRFP-Tanimoto reranker adds a database-similarity term, $\lambda \cdot (1 - \text{network\_similarity})$, to Wijers' scoring function, so $\lambda = 0$ recovers the baseline exactly and larger λ pulls candidates resembling known reactions toward the top. It helps most when the corpus matches the target chemistry (Table 9), the same results we observe in the building-block contribution and, acting only on the finished candidate list, it is the weakest of the three integration points.

Table 9: **DRFP reranker, target rank** (lower is better; bold = best per row). Rhea and the rank-normalised union outrank USPTO; USPTO hurts water where Rhea recovers it. †Diels–Alder and esterification are reported from the building-block pipeline under USPTO. We can see esterification imprpoving under match and rank-norm like water and methane while Rhea hurts it. with no headroom under the isolation, so its mismatch hurt (13 → 15) appears only in the pipeline.

Benchmark	No reranker	USPTO	Rhea	∪ rank-norm
Water	3	5	**1**	**1**
Methane	11	9	**7**	**7**
Diels–Alder†	13	15	15	15
Esterification†	12	**10**	14	**10**

On water, whose formation has essentially no organic analog in USPTO, the USPTO reranker hurts, pushing the target from rank 3 to 5. Matching the corpus reverses this, a Rhea reranker recovering water to rank 1 (Table 9). The hurt is therefore corpus mismatch, not a flaw in the reranker. We can observe similar trend in Esterification. Diels–Alder, whose chemistry appears in neither corpus, shows the same mismatch hurt, the USPTO and Rhea reranker demoting it from rank 13 to 15, the cleanest mismatch case since no corpus covers the target at all.

The choice of fingerprint backend matters. DRFP is a reaction-level fingerprint, whereas ECFP4 is molecule-level and has to be aggregated across a reaction as a sum or difference of its molecule fingerprints. A reaction-level fingerprint carries the reaction similarity signal the reranker needs, which a sum or difference of molecule fingerprints does not; this is why we adopt DRFP.

4.7 Discussion of results

Taken together the three mechanisms answer the reseacrh questions and set up the discussion in Section 6. The vocabulary is the only result that determines if the target is found; it alone decides reachability, and the shuffle control attributes that to database content rather than size. The PCFG adds a smaller ordering gain; output reranking only reorders what the search already produced and helps only under matched corpus match. All three are governed by the same corpus-to-chemistry matching.

5 Responsible Research

This section discusses the reproducibility of the methods and reflects on the ethics of guiding a chemical synthesizer with a corpus of known coverage bias.

5.1 Reproducibility

All code extends Wijers' CRNSynthesizer and is publicly available [20] at https://github.com/TymonJ/CRNSynthesizerDB, on a single branch, `CRNSynthesizerdb`, which holds the PCFG estimation, the building-block vocabulary, and the output reranker. Database knowledge comes from two corpora, USPTO-50K and Rhea. Because the enumeration order is fixed, every configuration reproduces its exact candidates-to-target index, and the only stochastic component, the ODE simulation used for scoring, changes a candidate's score but never its position in the enumeration.

5.2 Data licensing and provenance

USPTO-50K [5] is text-mined from public-domain US patents by Lowe [14] and is distributed under CC0; Rhea is released under CC-BY 4.0 [6]. None of the data involves personally identifiable information or human subjects.

5.3 Corpus bias

The prior this work injects is estimated from a reaction corpus, so the corpus's bias becomes the synthesizer's bias: USPTO-50K is dominated by organic patent chemistry, and the PCFG and reranker accordingly treat organic-looking reactions as more plausible. The system here is a research prototype evaluated on five benchmarks, not a tool a chemist would yet rely on, so the stakes are low. Were a synthesizer of this kind used to rank mechanism hypotheses, a "plausibility" score learned from a skewed corpus would reflect what the corpus has seen rather than what is chemically plausible, quietly down-ranking under-represented chemistry such as inorganic or gas-phase reactions. Our results show this in miniature: the USPTO reranker demotes the correct inorganic water network as λ grows, and a database-derived vocabulary can omit a required species. The mitigation, and this paper's empirical recommendation, is to report the corpus alongside any such score and to match it to the target chemistry rather than assume it is universal.

6 Discussion

6.1 Answer to the research question

On the first subquestion, where database knowledge helps most, corpus derived bulding block vocabulary looks to help the most by unlocking a new target, but its big caveat is that it relies on precise database match. It showed to demonstrate reachability: esterification is reachable only through USPTO blocks and methane combustion only through Rhea. The PCFG delivers smaller gain (2.11× on water, 1.63× on methane) but less corpus dependant and the output reranker only reorders an already-produced list. On the second subquestion, the effect is the database's frequency content, not the size of the vocabulary: the random-vocabulary shuffle shows that match to corpus and mining all required molecules is needed. On the third subquestion, corpus mismatch and reachibility of target reactions, reachability follows corpus-to-chemistry matching for the building blocks contribution. The two ordering mechanisms (PCFG and reranker) cannot make a target unreachable by construction, because they only permute an otherwise exhaustive enumeration; a database-derived vocabulary can, because it can omit a species the target needs.

6.2 Comparison to related approaches

The Reaction Mechanism Generator builds kinetic mechanisms by rate-based flux expansion [21]; its prior is mechanistic where ours is statistical, but both confront the same combinatorial growth of candidate networks. Cardelli et al. synthesize CRNs from temporal-logic specifications via SMT-modulo-ODE [12], a different input modality, the same search bottleneck. Grammar-based molecular VAEs place a (P)CFG at the **molecule** level [22], whereas we estimate weights for reactions and the vocabulary; the grammar-ontology transformer of Mann and Venkatasubramanian [23] is closer in spirit but is not enumerative. Finally, nested-evolution CRN design [24] is an alternative CRN-discovery paradigm.

6.3 Limitations

Several limitations constrain these results. The most important is statistical: the benchmark set is small, so every claim rests on a handful of targets and the reductions we report should be read as existence proofs rather than something that can be generalized. The corpus coverage bias is the main external factor: USPTO is organic-dominated and our inorganic benchmarks depend entirely on Rhea. The PCFG leaves the network grammar uniform, a deliberate scope choice. The combined-corpus result is validated only for two corpora and only with rank-normalised merging; whether it scales to three or more corpora, is untested.

Two further caveats concern how the results should be read. candidates-to-target, is an enumeration index taken before scoring; once a target is reachable its scored rank can be markedly better (esterification: enumeration index 1077, scored rank 1), so the index reflects search depth rather than final ranking quality. Main limitation for the synthesizer is hardware namely memory. Ethylene glycol is infeasible on any configuration and is excluded from the measured results (Section 4), and esterification is reached only with the building-block vocabulary (Table 2). These are limits of a single laptop, not of the approach. Finally, the large Diels–Alder ratio is not comparable to the water and methane reductions.

7 Conclusion and Future Work

We studied where, in a top-down CRN program-synthesis pipeline, knowledge from a reaction database most reduces the number of candidates explored before the ground-truth network is found. Across four benchmarks including a deliberate corpus-mismatch boundary. A database-derived building-block vocabulary is the clearest result; it determines which networks are reachable at all, and a random-vocabulary shuffle control shows that the effect comes from the database's frequency content rather than from the size of the vocabulary. A PCFG over the molecule grammar and reaction stoichiometry adds a reduction in candidates-to-target (2.11× on water, 1.63× on methane). A finer-grained molecular-formula prior hurts. Output reranking only reorders the finished candidate list and helps only when the corpus matches the target chemistry. The take-away is that where database knowledge enters, corpus coverage matters as much as how much of it there is. Matching a corpus to the target chemistry, rather than simply enlarging it, is what turns a reaction database into useful search guidance.

7.1 Future work

The most immediate extension of this work is evaluating across more benchmarks. With only four benchmarks, every claim here is an existence proof not generalized claim; extending the benchmark set to a larger and more chemically diverse collection of target networks would let the reductions we report be estimated with more confidence, and would test whether the corpus-matching effect generalizes beyond the cases studied.

Three further directions follow the integration points themselves. First, the PCFG can be pushed beyond the molecule grammar and reaction stoichiometry to the reaction- and network-grammar rules currently left uniform, given a corpus that supplies reliable network level counts. Second, the corpus can be broadened or better matched to the target chemistry, for example by adding the Open Reaction Database [25] or a domain-specific corpus, since we found coverage and matching to be decisive. Third, the rank-normalised corpus merge can be scaled from two corpora to many. Finally, the synthesizer is currently memory-bound: reducing the memory footprint, or candidates it enumerates would bring larger targets such as ethylene glycol within reach.

Bibliography

[1] M. Wen, E. W. C. Spotte-Smith, S. M. Blau, M. J. McDermott, A. S. Krishnapriyan, and K. A. Persson, "Chemical reaction networks and opportunities for machine learning," *Nature Computational Science*, vol. 3, no. 1, pp. 12–24, 2023, doi: 10.1038/s43588-022-00369-z.

[2] J. P. Unsleber and M. Reiher, "The Exploration of Chemical Reaction Networks," *Annual Review of Physical Chemistry*, vol. 71, pp. 121–142, 2020, doi: 10.1146/annurev-physchem-071119-040123.

[3] R. Wijers, "Automated Discovery of Chemical Reaction Networks using Program Synthesis," Master's thesis, 2025.

[4] A. K. Menon, O. Tamuz, S. Gulwani, B. Lampson, and A. Kalai, "A Machine Learning Framework for Programming by Example," in *International Conference on Machine Learning (ICML)*, 2013, pp. 187–195.

[5] N. Schneider, N. Stiefl, and G. A. Landrum, "What's What: The (Nearly) Definitive Guide to Reaction Role Assignment," *Journal of Chemical Information and Modeling*, vol. 56, no. 12, pp. 2336–2346, 2016, doi: 10.1021/acs.jcim.6b00564.

[6] P. Bansal *et al.*, "Rhea, the reaction knowledgebase in 2022," *Nucleic Acids Research*, vol. 50, no. D1, pp. D693–D700, 2022, doi: 10.1093/nar/gkab1016.

[7] M. Balog, A. L. Gaunt, M. Brockschmidt, S. Nowozin, and D. Tarlow, "DeepCoder: Learning to Write Programs," in *International Conference on Learning Representations (ICLR)*, 2017.

[8] W. Lee, K. Heo, R. Alur, and M. Naik, "Accelerating Search-Based Program Synthesis Using Learned Probabilistic Models," in *ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, 2018, pp. 436–449. doi: 10.1145/3192366.3192410.

[9] S. Barke, E. Anaya Gonzalez, S. R. Kasibatla, T. Berg-Kirkpatrick, and N. Polikarpova, "HYSYNTH: Context-Free LLM Approximation for Guiding Program Synthesis," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[10] J. Brence, L. Todorovski, and S. Džeroski, "Probabilistic Grammars for Equation Discovery," *Knowledge-Based Systems*, vol. 224, p. 107077, 2021, doi: 10.1016/j.knosys.2021.107077.

[11] T. Hinnerichs *et al.*, "Herb.jl: A Unifying Program Synthesis Library." 2025.

[12] L. Cardelli *et al.*, "Syntax-Guided Optimal Synthesis for Chemical Reaction Networks," in *Computer Aided Verification (CAV), Part II*, in LNCS, vol. 10427. Springer, 2017, pp. 375–395. doi: 10.1007/978-3-319-63390-9_20.

[13] T. E. Loman *et al.*, "Catalyst: Fast and flexible modeling of reaction networks," *PLOS Computational Biology*, vol. 19, no. 10, p. e1011530, 2023, doi: 10.1371/journal.pcbi.1011530.

[14] D. M. Lowe, "Extraction of Chemical Structures and Reactions from the Literature," Doctoral dissertation, 2012. doi: 10.17863/CAM.16293.

[15] N. Schneider, D. M. Lowe, R. A. Sayle, and G. A. Landrum, "Development of a Novel Fingerprint for Chemical Reactions and Its Application to Large-Scale Reaction Classification and Similarity," *Journal of Chemical Information and Modeling*, vol. 55, no. 1, pp. 39–53, 2015, doi: 10.1021/ci5006614.

[16] P. Schwaller *et al.*, "Mapping the Space of Chemical Reactions Using Attention-Based Neural Networks," *Nature Machine Intelligence*, vol. 3, pp. 144–152, 2021, doi: 10.1038/s42256-020-00284-w.

[17] D. Probst, P. Schwaller, and J.-L. Reymond, "Reaction classification and yield prediction using the differential reaction fingerprint DRFP," *Digital Discovery*, vol. 1, pp. 91–97, 2022, doi: 10.1039/D1DD00006C.

[18] C. W. Coley, L. Rogers, W. H. Green, and K. F. Jensen, "Computer-Assisted Retrosynthesis Based on Molecular Similarity," *ACS Central Science*, vol. 3, no. 12, pp. 1237–1245, 2017, doi: 10.1021/acscentsci.7b00355.

[19] M. H. Lin, Z. Tu, and C. W. Coley, "Improving the Performance of Models for One-Step Retrosynthesis through Re-Ranking," *Journal of Cheminformatics*, vol. 14, p. 15, 2022, doi: 10.1186/s13321-022-00594-8.

[20] T. Jastrzemski, "CRNSynthesizerDB: Database-Guided Program Synthesis of Chemical Reaction Networks." [Online]. Available: https://github.com/TymonJ/CRNSynthesizerDB

[21] C. W. Gao, J. W. Allen, W. H. Green, and R. H. West, "Reaction Mechanism Generator: Automatic construction of chemical kinetic mechanisms," *Computer Physics Communications*, vol. 203, pp. 212–225, 2016, doi: 10.1016/j.cpc.2016.02.013.

[22] M. J. Kusner, B. Paige, and J. M. Hernández-Lobato, "Grammar Variational Autoencoder," in *International Conference on Machine Learning (ICML)*, 2017, pp. 1945–1954.

[23] V. Mann and V. Venkatasubramanian, "Predicting Chemical Reaction Outcomes: A Grammar Ontology-Based Transformer Framework," *AIChE Journal*, vol. 67, no. 3, p. e17190, 2021, doi: 10.1002/aic.17190.

[24] É. Degrand, M. Hemery, and F. Fages, "On Chemical Reaction Network Design by a Nested Evolution Algorithm," in *Computational Methods in Systems Biology (CMSB)*, Springer, 2019, pp. 78–95. doi: 10.1007/978-3-030-31304-3_5.

[25] S. M. Kearnes *et al.*, "The Open Reaction Database," *Journal of the American Chemical Society*, vol. 143, no. 45, pp. 18820–18826, 2021, doi: 10.1021/jacs.1c09820.