RECIPE: Learning to Rank Complete Precursor Sets for Inorganic Retrosynthesis

Jing Gao[1,2] **Kaipeng Zeng**[1,3] **Fuyuan Xia**[1] **Jian Ma**[1,2]
Yufeng Li[1,2] **Qifeng Li**[1,2] **Lin Yao**[1,2] **Junchi Yan**[1*]

[1]Shanghai Jiao Tong University, Shanghai, China
[2]Zhongguancun Academy, Beijing, China
[3]Shanghai Innovation Institute, Shanghai, China

gaojing1@sjtu.edu.cn yanjunchi@sjtu.edu.cn

Abstract

Single-step inorganic retrosynthesis is evaluated by whether a complete precursor set is recovered, yet many data-driven systems first rank individual precursors and then rely on threshold or count heuristics to assemble sets. This mismatch is especially severe when the true precursor-set size is unknown: individually plausible precursors can combine into incomplete, over-complete, or otherwise wrong near-miss sets. We reformalize the task as variable-size set-level ranking and introduce **RECIPE**–**R**eranking **E**xact **C**andidate **I**norganic **P**recursor **E**ntries–a target-conditioned framework that separates precursor recall from final set-level scoring. The *Precursor Candidate Generator* learns formula-level compatibility to build a high-recall precursor pool, while the *Complete-Set Reranker* compares variable-size candidate precursor sets directly. This design targets a central failure mode under this protocol: a model may retrieve the right individual precursors but fail to rank the right combination first. On the Retrieval-Retro year-split benchmark, the generator improves Combo@20 from 69.00 to 72.75. Compared with Retrieval-Retro, the reranker with full-list inference improves Combo@1 by **11.53 points** to 71.93, Combo@20 by **24.50 points** to 93.50, and Combo MRR by **15.48 points** to 78.77. A preliminary 20-case out-of-distribution evaluation shows the same direction of improvement. These results suggest that set-level prioritization remains important after strong precursor recall, and that optimizing ranked complete precursor sets can reduce the inspection burden.

1 Introduction

Computational materials discovery can now propose vast numbers of promising inorganic crystals, from high-throughput screening pipelines to autonomous laboratories and large generative models [16, 23, 31]. This progress sharpens a familiar bottleneck in the design–make–test loop: many models can suggest what material may be worth making, but a proposed composition is actionable only when paired with a plausible synthetic route to the target phase. In inorganic synthesis, that route often begins with a small set of precursors selected from a large historical vocabulary.

Single-step inorganic retrosynthesis asks for such a set: given a target composition, predict the precursors used to synthesize it. Directly importing organic retrosynthesis methods [30] is difficult. Inorganic reactions are less standardized, often lack transferable reaction centers, depend strongly on stoichiometry, and are affected by solid-state interfaces, volatility, thermodynamics, and kinetics [20, 14, 24]. Existing data-driven systems therefore tend to exploit composition similarity, element-level

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

templates, or retrieved reference materials to score likely precursors [10, 7, 12, 18]. These approaches have advanced precursor recommendation, but their learning objectives still largely operate before the final set-level decision.

The central mismatch is that optimizing precursor scores is not the same as optimizing the object chemists actually need. Benchmarks and chemists judge whether the *complete precursor set* is correct, and the set size is usually unknown before synthesis. A precursor classifier can assign high probability to several mutually incompatible alternatives: precursors from similar target compounds, literature variants that merge near-duplicate formulas, or individually plausible sources that cannot all belong to the same set. Thus precursor-level ranking is a useful recall surrogate, but it is a relaxation of the stricter decision problem. When the precursor-set size is unknown, thresholding at $p > 0.5$, enforcing a predicted count, or multiplying independent precursor scores must introduce a separate assembly heuristic; this structurally exposes the system to incomplete, over-complete, or chemically plausible but wrong near-miss sets. These errors motivate directly comparing complete variable-size sets rather than relying only on independent precursor scores.

We therefore formulate single-step inorganic retrosynthesis as **variable-size set-level ranking**: given a target, the model should return an ordered list of complete precursor sets of different sizes, with the exact literature set ranked as early as possible. Figure 1 summarizes this task-level mismatch: the desired object is a ranked list of complete precursor sets, while precursor-level systems optimize a looser intermediate object and must later assemble and order many near-miss sets. We introduce RECIPE (**R**eranking **E**xact **C**andidate **I**norganic **P**recursor **E**ntries), a two-stage reranking framework that first ranks individual precursor candidates and then reranks complete precursor sets drawn from that pool. The *Precursor Candidate Generator* is a recall module: its job is to avoid omitting correct precursors from the candidate pool, not to optimize the final set ordering. The *Complete-Set Reranker* is the complementary set-ranking module: it does not generate new candidates, but scores enumerated variable-size precursor sets using candidate-set descriptors, precursor-score residuals, and set-completeness supervision. Figure 2 shows the resulting architecture.

This formulation leads to three practical consequences. First, a precursor-level model remains valuable, but its role is candidate generation: it should be judged by whether it avoids missing the required precursors. Second, the final decision should be a set-level ranking problem: the reranker should choose the most correct complete set from the candidate list using evidence that is only meaningful at the set level, such as element coverage, unwanted elements, co-occurrence, and precursor priors. Third, evaluation should reward the ordering of exact complete sets, not only precursor recall: we use Combo MRR as a single Combo Accuracy summary, defined as the reciprocal rank of the exact ground-truth precursor set under the paper's percentage-scale MRR convention.

Our experiments support the reframing. On the Retrieval-Retro benchmark, the Precursor Candidate Generator already surpasses the previous strongest baseline, improving Top-1 precursor coverage from 67.12 to 69.76 and Combo@20 from 69.00 to 72.75. More importantly, the Complete-Set Reranker improves over Retrieval-Retro by 11.53 points on Combo@1, 24.50 points on Combo@20, and 15.48 points on Combo MRR, reaching 71.93, 93.50, and 78.77, respectively. A preliminary OOD evaluation on 20 diverse compounds shows the same direction of improvement, but we do not present it as broad evidence of out-of-distribution generalization.

Our contributions are:

- We formulate single-step inorganic retrosynthesis as variable-size ranking of complete precursor sets and separate learning into precursor recall and complete-set ranking tasks.
- We develop a composition-only *Precursor Candidate Generator* that learns formula-level compatibility to build high-recall precursor pools.
- We introduce a *Complete-Set Reranker* that scores enumerated sets using set descriptors, precursor-score residuals, and listwise supervision.
- We evaluate RECIPE on Retro, Ceder, and a preliminary OOD study, with cross-generator, simple-scorer, descriptor, and pruning controls to examine the sources of improvement.

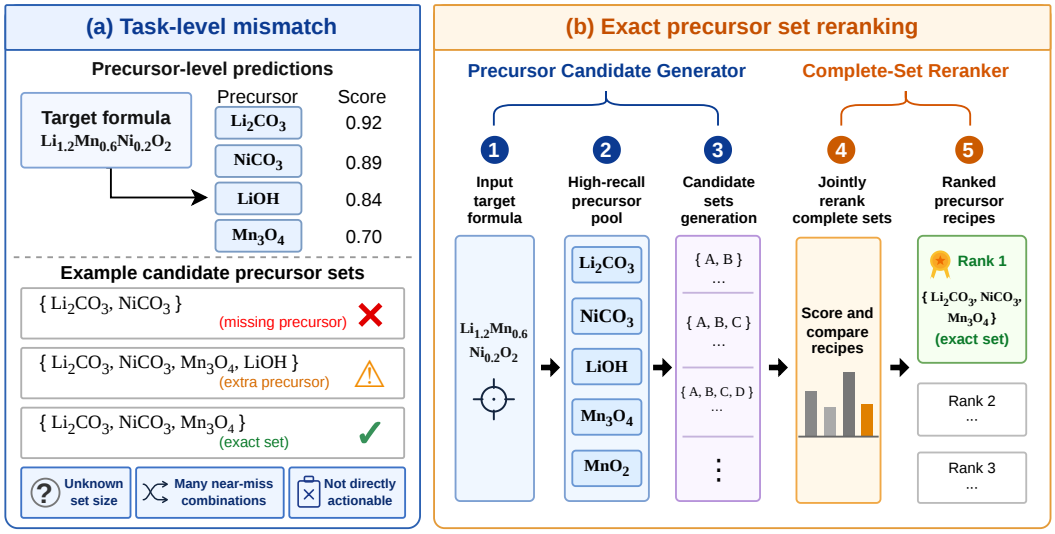


Figure 1: Precursor predictions require unknown-size set assembly. RECIPE directly ranks complete sets, distinguishing exact recipes from incomplete, over-complete, and substituted alternatives.

2 Related Work

Inorganic Synthesis Data and Literature Mining. Large text-mined corpora made data-driven inorganic synthesis planning possible. The Ceder synthesis dataset extracts targets, precursors, and synthesis operations from the literature [13]; follow-up work mined procedure types, material roles, solution-based syntheses, and action graphs [9, 8, 28, 27]. These resources supervise precursor recommendation, but their complete-set records contrast with models that score precursors individually.

Precursor Recommendation and Inorganic Retrosynthesis. Early neural synthesis planners learned from literature recipes to recommend actions and precursors [10]. SynthesisSimilarity recommends precursor sets by analogy to historically synthesized materials [7]. ElementwiseRetro formulates inorganic retrosynthesis through element-level templates and ranks candidate sets by the joint probability of their element-wise precursor choices [12]. Retrieval-Retro combines reference-material retrieval, implicit precursor extraction from retrieved materials, and reaction-energy-informed scoring, establishing the strongest recent benchmark for inorganic precursor prediction [18]. Retro-Rank-In is closest in spirit because it also frames inorganic retrosynthesis as ranking precursor sets, but its ranker is trained on target–precursor pairs and its final set ranking still assumes independent precursor probabilities under precursor-set size and elemental-completeness constraints [19]. In contrast, RECIPE keeps the closed-vocabulary Retrieval-Retro protocol, constructs variable-size candidate sets, and learns a target-conditioned comparator over each list of complete sets directly. Thus the ranked object is the complete candidate set, not an individual precursor, retrieved precedent, or product of independent precursor scores.

Composition-Only Material Representations. Many material models operate directly on composition because crystal structures or reaction conditions are often unavailable at screening time. Matminer featurizes formulas using elemental attributes [29]; Roost learns stoichiometry-aware composition representations without crystal structures [5]; CrabNet applies attention over composition tokens [26]. Our candidate generator follows this composition-only setting, but the final model differs from property predictors and precursor classifiers by ranking structured precursor sets.

Learning-to-Rank Candidate Lists. Learning-to-rank methods distinguish pointwise, pairwise, and listwise objectives. Listwise approaches treat an entire object list as the training instance and define losses on the predicted list rather than on isolated items or pairs [1, 3]. Differentiable sorting work further connects ranking surrogates to top-ranked metrics [6, 22]. Our contribution is not a new

generic ranking loss; we identify the ranked objects in inorganic retrosynthesis as target-specific lists of complete precursor sets.

Thermodynamic and Closed-Loop Synthesis Planning. Thermodynamic pathfinding models solid-state synthesis as search over reaction networks or phase diagrams [15, 17, 4]. Autonomous laboratories integrate computational priors, recipe models, robotics, and characterization feedback [23]. Complementing synthesis planning, SafeLab benchmarks safety during robotic laboratory execution, distinguishing task completion from safe execution [2]. Recent LLM-based systems broaden the interface for synthesizability and precursor prediction [21, 11]. These directions complement our focus on single-step precursor-set prediction and Combo Accuracy; thermodynamic feasibility, reaction conditions, and multi-step planning remain future work.

3 Problem Setup and Metrics

Task. Let x denote a target inorganic formula and let $\mathcal{P}$ be a closed precursor vocabulary. A precursor set is an unordered set $S = \{p_1, \ldots, p_m\} \subseteq \mathcal{P}$ with variable size m. The single-step retrosynthesis task is to return a ranked list of precursor sets

$$\mathcal{R}(x) = (S_1, S_2, \ldots) \tag{1}$$

such that the ground-truth set $S^\star$ appears as early as possible. The closed-vocabulary assumption follows prior benchmark protocols: evaluation targets are restricted to reactions whose precursors appear in the training vocabulary, but the target formulas in test splits may be temporally held out.

Generator–Reranker Formulation. We explicitly separate the task into two decisions with different objectives. The *Precursor Candidate Generator* learns a precursor-level score $q(p \mid x)$ and uses it to build a high-recall pool $\mathcal{C}(x) \subset \mathcal{P}$ retained after ranking, compatibility screening, and capping. The *Complete-Set Reranker* constructs a set space $\Omega(x) = \{S : S \subseteq \mathcal{C}(x), |S| \in \mathcal{M}\}$ and learns a target-conditioned set score $s(x, S)$. Here $\mathcal{M} = \{2, 3, 4, 5\}$ is the set of allowed set sizes. The truth $S^\star$ need not lie in $\Omega(x)$: a required precursor may be absent from the retained pool. The final prediction is not a thresholded precursor list, but the ordering induced by $s(x, S)$ over $\Omega(x)$. This distinction is central: the generator should maximize the chance that all true precursors are available, while the reranker should decide which complete set a chemist should inspect first.

Variable-Size Candidate Sets. The model is not given $|S^\star|$. This is the practical setting: before synthesis, the user typically does not know whether a target should be made from two, three, four, or more precursors. Precursor-level systems therefore face an awkward intermediate problem. If they threshold sigmoid probabilities, the selected count can be wrong; if they assume an oracle count, the evaluation no longer reflects actual use; if they enumerate many counts without a learned set score, the number of plausible near-miss combinations grows quickly. Our reranker addresses this predicament by constructing candidate sets of multiple sizes and ranking them jointly. Missing precursors and extra precursors are both penalized because the exact set, not only precursor coverage, is the evaluated object.

Set-List Learning Objective. The set view changes the ambiguity from independent precursor labels to competing complete hypotheses for the same target. Many near-miss sets can differ from $S^\star$ by only one member, such as a missing source, an extra source, or a plausible substitution. The generator is therefore judged by recall and precursor-level ordering, while the reranker is judged by Combo Accuracy and Combo MRR: given a high-recall pool, it must rank the correct complete set ahead of these alternatives. This separation avoids conflating candidate recall with final set quality.

Precursor Coverage and Combo Accuracy. We report the precursor-coverage and Combo@k metrics used by prior precursor-prediction work, with formal definitions in Section A.2. In the main text, we use them to separate two questions: whether the generator recalls all required precursors, and whether the final ranked list places the exact complete set near the top. Precursor coverage checks whether all true precursors appear among the top $\max(k, |S^\star|)$ predictions. The true set size is used only to compute this metric, not provided to the model.

Mean Reciprocal Rank (MRR) and Combo MRR. Mean Reciprocal Rank (MRR) measures how early a relevant object appears in a ranked list, giving a single-number summary of ranking quality across cutoffs. We use precursor-level MRR only as a generator diagnostic over individual precursors, and Combo MRR as the corresponding exact-set metric over ranked complete precursor sets. For target x_i, r_i is one plus the number of candidate sets scoring strictly above $S_i^\star$; tied scores share a rank. Absent true sets have infinite rank and receive reciprocal rank zero:

$$\text{ComboMRR} = \frac{100\%}{N} \sum_{i=1}^{N} \begin{cases} 1/r_i, & S_i^\star \in \mathcal{R}(x_i), \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Unlike Combo@k, Combo MRR is sensitive to the exact rank: moving the true set from rank 20 to rank 2 improves the score even though Combo@20 is unchanged. It can therefore capture ordering improvements within a fixed cutoff window, such as better ranks inside the Top-10–Top-20 range when coverage or Combo@k accuracy does not change.

Precursor Mean Reciprocal Rank (Prec. MRR) averages reciprocal ranks over all true target–precursor pairs, whereas Combo MRR averages over target-level exact sets. MRR complements selected Top-k cutoffs; it is not claimed to be universally better than accuracy.

4 Method

4.1 Precursor Candidate Generator

The Precursor Candidate Generator encodes each formula as element identities and stoichiometric fractions. Target and precursor roles use separate two-layer, four-head Transformer encoders [25], with 96-dimensional states and a learned CLS token. Their formula representations are projected to 192 dimensions. The pair scorer combines a learned-scaled dot product, an MLP on the concatenated target and precursor embeddings, and a precursor-specific bias:

$$z_p = g_\theta\left(h_\theta^{\text{tar}}(x), h_\theta^{\text{pre}}(p)\right), \qquad q_p = \sigma(z_p). \tag{3}$$

Multi-label binary cross entropy trains precursor compatibility over $\mathcal{P}$. Stage 1 is trained separately and remains fixed during Stage-2 training. Implementation and training settings appear in Section B.1.

4.2 Candidate Construction and Set Descriptors

We take the Top-30 precursor scores, apply a fixed compatibility screen, and retain at most 15 precursors. The screen removes precursors introducing non-volatile elements absent from the target; the coarse volatile-element convention is given in Section B.1. We exhaustively enumerate sizes 2–5, with at most $\sum_{m=2}^{5} \binom{15}{m} = 4{,}928$ sets. These sizes cover all labeled sets in Retro and Ceder. There is no additional inference candidate-set budget. Ablations of the pool cap, compatibility filter, and enumerated size range, including ranking performance and computational cost, are reported in Section D.1 and Table 13.

A descriptor vector $\phi(x, S) \in \mathbb{R}^{374}$ supplies evidence about each complete set. Its six groups describe generator probabilities, target-element coverage, unwanted elements, co-occurrence and precursor priors, stoichiometry, and source mapping (Table 10). Retro and Ceder share this schema, with corpus-count priors recomputed from each training split (definitions in Section B.3).

4.3 Complete-Set Reranker

The Complete-Set Reranker first converts each complete candidate set into a target-conditioned *set token*. For a candidate set S_i, the token combines projected candidate-set descriptors, target-formula context, and set size. A generator-score residual preserves the precursor probabilities in the final score. Implementation and configuration details are provided in Sections B.1 and B.2.

Given the candidate list for the same target, a Transformer list encoder maps $[t_1, \ldots, t_K]$ to contextualized set tokens $[\tilde{t}_1, \ldots, \tilde{t}_K]$. A shared MLP head and the generator-score residual produce $s_\theta(x, S_i)$, allowing each complete set to be scored relative to the other candidates.

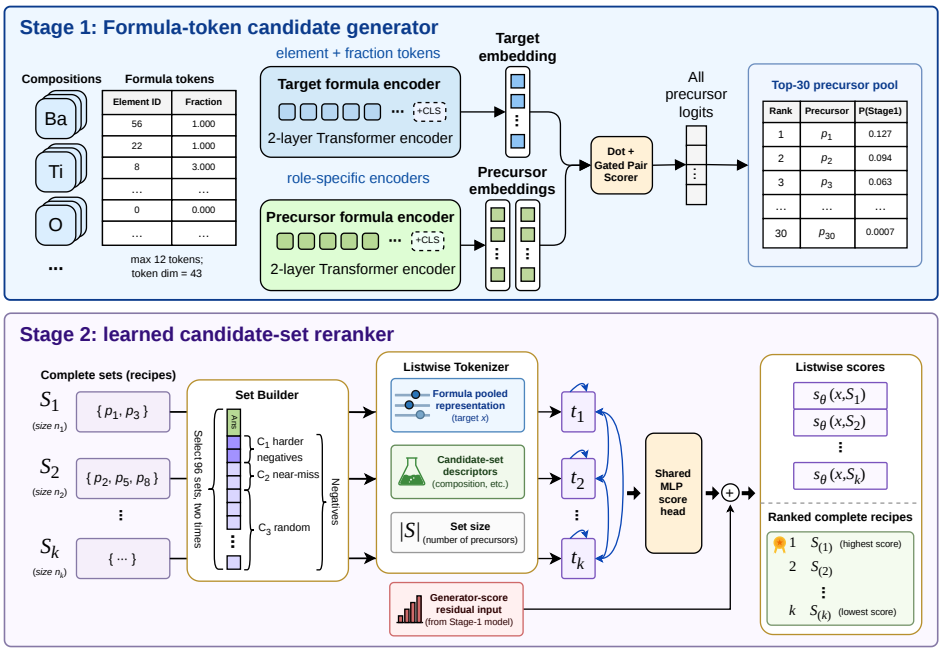


Figure 2: Two-stage RECIPE architecture: formula encoders generate precursor probabilities; candidate construction and listwise scoring rank complete sets. The sampling branch applies only during training; inference scores all enumerated candidates.

4.4 Listwise Set Ranking Loss

The Complete-Set Reranker is trained with a *Listwise Set Ranking Loss* tailored to the chemical set decision. The learning instance is not a precursor label or an independent candidate pair, but a target-specific list of complete sets containing the literature precursor set and chemically plausible competitors. For each target, the training list is arranged as

$$[S^\star, S_1^-, S_2^-, \ldots, S_K^-], \tag{4}$$

where the first set is the ground-truth precursor set and the remaining sets are negatives sampled around the candidate pool produced by the Precursor Candidate Generator. The sampled list has a fixed maximum length, which prevents large precursor pools from forcing full enumeration during training and exposes the scorer to diverse set sizes across updates. Negatives combine high-probability hard sets, near-miss single-substitution sets, and random top-ranked sets, so the scorer learns to compare complete hypotheses rather than memorize one set size. The model outputs scores

$$\mathbf{s} = [s_\theta(x, S^\star), s_\theta(x, S_1^-), \ldots, s_\theta(x, S_K^-)]. \tag{5}$$

The listwise term asks the exact set to win against all alternatives for the same target:

$$\mathcal{L}_{\text{rank}} = -\sum_{j=0}^{K} y_j \log \frac{\exp(s_j/\tau)}{\sum_{\ell=0}^{K} \exp(s_\ell/\tau)}, \tag{6}$$

with label smoothing. To make the loss sensitive to near-miss chemical sets, we add a strongest-negative separation term:

$$\mathcal{L}_{\text{margin}} = \max\left(0, \gamma + \max_j s_\theta(x, S_j^-) - s_\theta(x, S^\star)\right), \tag{7}$$

which penalizes the most competitive incomplete, over-complete, or substituted set in the list. Finally, an element-coverage auxiliary term is applied to the positive list item and asks the auxiliary head to recover the target's constituent elements:

$$\mathcal{L}_{\text{cov}} = \text{BCE}(c_\theta(x, S^\star), e(x)), \tag{8}$$

where $e(x)$ is the binary target-element presence vector. The resulting chemistry-aware set-ranking objective is

$$\mathcal{L}_{\text{set-rank}} = \mathcal{L}_{\text{rank}} + \lambda_{\text{margin}}\mathcal{L}_{\text{margin}} + \lambda_{\text{cov}}\mathcal{L}_{\text{cov}}. \tag{9}$$

The weights λ_{margin} and λ_{cov} balance the auxiliary terms against the listwise ranking loss. Hyperparameter settings are given in Section B.1.

Inference. At inference time, the generator produces precursor probabilities, candidate construction enumerates the allowed complete sets, and the reranker scores every surviving set with full-list attention. Both Retro and Ceder have at most 4,928 candidates per target, fitting within the configured list capacity of 8,192. The softmax loss is a training surrogate: inference sorts all candidates by their raw scalar scores, without interpreting them as probabilities. This global ordering handles variable-size prediction. The chunked-inference comparison is reported in Section D.2.

5 Experiments

Dataset and Protocol. There are two closely related datasets in this literature. The Ceder synthesis dataset is the larger text-mined inorganic synthesis corpus [13]. Retrieval-Retro builds a cleaned benchmark from this corpus for precursor prediction, with standardized precursor vocabularies and chronological train/validation/test splits [18]. Our main results use this Retrieval-Retro benchmark, which we refer to as Retro in tables and figures. The split contains 24,034 train, 1,842 validation, and 2,558 test reactions: training and validation reactions are dated no later than 2017, while test reactions are dated 2018 or later, matching the forward-looking evaluation emphasized by Retrieval-Retro. Following the benchmark, validation and test reactions are restricted to precursors present in the training vocabulary. Unless otherwise noted, candidate sets of multiple sizes are ranked jointly. Seed and uncertainty details are provided in Section B.4.

Metrics. We report Top-k precursor coverage, Combo@k exact-set recovery, and Combo MRR, a percentage-scale reciprocal-rank summary for the ground-truth set. Formal definitions, including Prec. MRR, are in Section A.2.

Baselines. We compare against composition-only models (Matminer [29], Roost [5], CrabNet [26], and an MLP [10]), prior inorganic retrosynthesis methods reported in the Retrieval-Retro benchmark (SynthesisSimilarity [7] and ElemwiseRetro [12]), and Retrieval-Retro itself [18]. The composition-only baselines test whether formula representations alone can rank precursors well. SynthesisSimilarity and ElemwiseRetro are included for their reported Combo metrics on Retro, while Retrieval-Retro is the previous state of the art under the shared benchmark and serves as the main prior baseline.

For precursor-ranking baselines, Combo metrics use the standard precursor-list conversion protocol from prior work: predict the number of precursors from sigmoid probabilities above 0.5, rank combinations by the product of member probabilities, and assign zero when the predicted number differs from the ground-truth set size. This evaluates how prior precursor-ranking systems behave without an oracle set size. RECIPE Reranker rows instead evaluate the proposed complete-set ranking formulation by scoring variable-size candidate sets directly; the RECIPE Generator + metal-filter control in Table 12 applies label-free compatibility pruning directly to generator-threshold predictions to test whether pruning alone explains the gain.

5.1 Precursor Candidate-Pool Quality

Tables 1 and 2 evaluate precursor candidate-pool quality on Retro and Ceder. Compared with Retrieval-Retro, the Precursor Candidate Generator improves Top-20 precursor coverage from 92.96 to 95.93 on Retro and from 90.73 to 94.54 on Ceder. Top-1 coverage also increases from 67.12 to 69.76 and from 65.24 to 67.17, respectively. Prec. MRR improves on both datasets, from 52.49 to 53.32 on Retro and from 52.01 to 52.80 on Ceder. These consistent gains show that the generator provides stronger precursor rankings and higher-recall candidate pools across both datasets.

The stronger generator also improves Combo@20 under standard precursor-list conversion, from 69.00 to 72.75 on Retro and from 67.25 to 70.73 on Ceder. Recovering and prioritizing complete sets from these precursor candidates is the next task, evaluated below.

Table 1: Retro benchmark. Top-k precursor coverage measures recovery of all true precursors; Combo@k measures exact-set recovery among the top k candidate sets. Rows marked * are copied from Retrieval-Retro [18]; C@20 and Combo MRR were not reported for these rows. The reranker uses full-list inference; uncertainties are sample SD over three checkpoints. Final-row gains over Retrieval-Retro use Generator precursor metrics and Reranker Combo metrics (percentage points).

Method	Top-k precursor coverage (%)						Combo Accuracy (%)					
	Top-1 ↑	Top-3 ↑	Top-5 ↑	Top-10 ↑	Top-20 ↑	Prec. MRR ↑	C@1 ↑	C@3 ↑	C@5 ↑	C@10 ↑	C@20 ↑	Combo MRR ↑
Matminer [29]	33.58	35.54	49.57	68.92	82.06	47.23	21.31	24.86	26.35	28.30	30.10	23.75
Roost [5]	44.53	45.82	56.41	70.37	81.94	49.01	33.15	36.28	36.79	38.23	39.21	35.01
MLP [10]	51.45	53.01	63.21	75.72	84.25	51.39	37.61	40.34	41.44	42.77	44.10	39.31
SynthesisSimilarity* [7]	–	–	–	–	–	–	45.03	48.02	49.11	51.09	–	–
ElemwiseRetro* [12]	–	–	–	–	–	–	53.45	57.07	58.19	60.84	–	–
CrabNet [26]	57.47	57.58	62.39	78.54	85.89	51.07	56.10	56.14	56.18	56.76	59.58	56.53
Retrieval-Retro [18]	67.12	68.18	77.01	87.06	92.96	52.49	60.40	65.21	66.22	67.63	69.00	63.29
RECIPE Generator	**69.76** (±1.24)	**70.94** (±1.12)	**81.08** (±0.91)	**92.04** (±0.18)	**95.93** (±0.18)	**53.32** (±0.05)	62.72 (±0.68)	67.06 (±0.45)	68.53 (±0.08)	70.69 (±0.37)	72.75 (±0.28)	65.57 (±0.45)
RECIPE Reranker	–	–	–	–	–	–	**71.93** (±0.63)	**84.25** (±0.14)	**87.91** (±0.06)	**91.22** (±0.33)	**93.50** (±0.26)	**78.77** (±0.34)
Ours vs. Retrieval-Retro	+2.64	+2.76	+4.07	+4.98	+2.97	+0.83	+11.53	+19.04	+21.69	+23.59	+24.50	+15.48

Table 2: Ceder benchmark under the same evaluation protocol as Table 1. Top-k precursor coverage measures whether all ground-truth precursors appear in the individual precursor ranked list. Combo@k measures whether the exact ground-truth precursor set appears among ranked candidate sets. The reranker uses full-list inference; uncertainties are sample SD over three checkpoints. The final row reports percentage-point gains over Retrieval-Retro, using the Generator for precursor-coverage metrics and the Complete-Set Reranker for Combo metrics.

Method	Top-k precursor coverage (%)						Combo Accuracy (%)					
	Top-1 ↑	Top-3 ↑	Top-5 ↑	Top-10 ↑	Top-20 ↑	Prec. MRR ↑	C@1 ↑	C@3 ↑	C@5 ↑	C@10 ↑	C@20 ↑	Combo MRR ↑
Matminer [29]	35.58	37.59	50.41	68.58	78.94	47.17	23.86	27.54	28.77	30.33	32.00	26.22
Roost [5]	51.87	53.14	62.27	76.11	82.89	49.41	38.89	41.21	41.92	43.01	45.09	40.51
MLP [10]	51.30	52.62	63.02	75.12	82.28	50.89	40.08	43.73	44.38	45.88	47.10	42.28
CrabNet [26]	61.86	63.02	69.70	79.31	85.38	51.18	54.06	56.95	58.11	59.75	61.38	56.12
Retrieval-Retro [18]	65.24	66.33	75.84	86.20	90.73	52.01	57.87	62.54	63.60	65.85	67.25	60.68
RECIPE Generator	**67.17** (±0.68)	**68.60** (±0.77)	**78.66** (±0.79)	**90.80** (±0.12)	**94.54** (±0.26)	**52.80** (±0.13)	60.17 (±0.69)	65.08 (±1.07)	66.53 (±1.24)	68.70 (±1.63)	70.73 (±1.63)	63.30 (±0.86)
RECIPE Reranker	–	–	–	–	–	–	**64.21** (±3.73)	**76.07** (±5.82)	**80.90** (±6.49)	**86.15** (±4.95)	**89.60** (±2.83)	**71.63** (±4.45)
Ours vs. Retrieval-Retro	+1.93	+2.27	+2.82	+4.60	+3.81	+0.79	+6.34	+13.53	+17.30	+20.30	+22.35	+10.95

5.2 Complete-Set Reranking Performance

Table 1 reports the main Combo Accuracy gain, with a visual summary in Section C.3 and Figure 3. Compared with Retrieval-Retro, the Complete-Set Reranker improves Combo@1 by 11.53 points to 71.93, Combo@20 by 24.50 points to 93.50, and Combo MRR by 15.48 points to 78.77. Compared with the RECIPE Generator row in the same table, the reranker improves these metrics by 9.21, 20.75, and 13.20 points, respectively. The improvement extends beyond the first prediction to the ordering of complete candidate sets.

Table 3: Generator and scorer controls on Retro. Three-seed means with sample SD beneath.

Scorer	C@1 ↑	C@5 ↑	C@20 ↑	Combo MRR ↑
(a) Retrieval-Retro generator				
Native decoding	60.40	66.22	69.00	63.29
RECIPE Reranker Frozen, chunk-512	68.97 (±0.02)	84.34 (±1.06)	89.30 (±1.68)	75.67 (±0.61)
RECIPE Reranker Retrained, chunk-512	68.27 (±0.59)	81.95 (±3.04)	87.56 (±3.53)	74.45 (±1.55)
(b) RECIPE generator				
Logistic regression	70.81 (±0.02)	**88.04** (±0.00)	93.43 (±0.00)	78.39 (±0.02)
XGBoost	70.00 (±0.14)	87.07 (±0.38)	92.74 (±0.08)	77.52 (±0.08)
MLP	70.63 (±0.27)	87.29 (±0.28)	92.92 (±0.08)	77.98 (±0.27)
RECIPE Reranker Chunk-512	71.70 (±0.10)	84.52 (±1.48)	89.82 (±1.74)	77.43 (±0.69)
RECIPE Reranker Full-list	**71.93** (±0.63)	87.91 (±0.06)	**93.50** (±0.26)	**78.77** (±0.34)

The same trend holds on Ceder (Table 2): the Complete-Set Reranker reaches 64.21 Combo@1, 89.60 Combo@20, and 71.63 Combo MRR, improving over Retrieval-Retro by 6.34, 22.35, and 10.95 points, respectively.

The final Retro candidate space contains the exact set for 2,449/2,558 targets (95.74%). Conditional C@1 is 75.13 when the truth is available; candidate misses account for 15.2% of Top-1 failures and ranking errors for 84.8% (Table 9). Thus candidate recall and complete-set ranking remain distinct objectives, even with a strong generator.

Table 3 shows that our frozen reranker also improves Retrieval-Retro, raising C@1/C@20 from 60.40/69.00 to 68.97/89.30. With matched chunk-512 inference, using the RECIPE generator further raises C@1 to 71.70, supporting the complementary roles of precursor-pool quality and complete-set reranking, with reranking gains that transfer across precursor generators.

With the RECIPE generator and our six descriptor groups, simple scorers—logistic regression, XGBoost and an MLP—also achieve strong C@1 of 70.00–70.81. Their performance supports the complete-set task design and shared representation across different scorers (data organization in Section C.1). We next examine the reranker components.

5.3 Reranker Ablations

Table 4 compares the full reranker with variants that remove the generator-probability residual, disable the coverage auxiliary loss, or remove the set descriptors. Removing descriptors reduces C@1, C@20 and Combo MRR to 40.96, 56.50 and 45.10, respectively. These descriptors expose signals beyond individual precursor scores, including target-element coverage, unwanted elements and stoichiometric compatibility. Removing the probability residual or disabling the coverage auxiliary loss has a small effect on mean C@1 relative to seed variability. The results support RECIPE's complete-set representation and ranking design as a whole, rather than attributing its performance to any single auxiliary component.

Table 4: Retro reranker ablations with mixture pooling off and full-list inference. Mean with sample SD beneath (three seeds).

Variant	C@1 ↑	C@5 ↑	C@20 ↑	Combo MRR ↑
RECIPE	71.93 (±0.63)	87.91 (±0.06)	93.50 (±0.26)	78.77 (±0.34)
w/o prob. res.	71.67 (±0.94)	87.66 (±0.12)	93.52 (±0.28)	78.58 (±0.61)
cov. loss off	71.88 (±0.90)	87.79 (±0.30)	93.56 (±0.16)	78.78 (±0.55)
w/o set desc.	40.96 (±18.52)	49.02 (±23.24)	56.50 (±26.74)	45.10 (±20.45)
Generator + metal filter	63.46 (±0.63)	69.46 (±0.16)	73.89 (±0.04)	66.38 (±0.45)

The generator-plus-filter control reaches only 63.46 C@1, 73.89 C@20 and 66.38 Combo MRR, while removing filtering from the learned reranker retains 71.34 C@1 and 88.06 C@20. Thus compatibility pruning alone does not explain the reranking gain. Same-enumeration scoring and candidate-construction controls are provided in Sections C.2 and D.1.

5.4 Preliminary OOD Evaluation

Table 5: Expanded 20-case OOD evaluation. RECIPE ranks variable-size candidate sets jointly; values are mean ± population standard deviation across three independent seeds.

Method	C@1 ↑	C@3 ↑	C@5 ↑	C@10 ↑	C@20 ↑	Combo MRR ↑	Fail >20 ↓
Retrieval-Retro	35.00	35.00	35.00	40.00	40.00	35.82	12
RECIPE	**55.00** (±0.00)	**63.33** (±2.36)	**68.33** (±2.36)	**80.00** (±8.16)	**85.00** (±12.25)	**61.36** (±1.64)	3.00 (±2.45)

Beyond the main benchmark, we run a preliminary 20-compound out-of-distribution evaluation on expert-annotated, literature-grounded targets selected to span diverse inorganic families, with literature-reported precursor sets used for exact-set matching. The Complete-Set Reranker ranks variable-size candidate sets jointly without the true set size. Table 5 shows that Retrieval-Retro reaches only 40.00 Combo@20 and 35.82 Combo MRR, with 12 of 20 targets still missing after the first 20 ranked sets. In contrast, RECIPE continues to improve as the inspection cutoff grows, rising from 55.00 Combo@1 to 85.00 Combo@20 and reaching 61.36 Combo MRR across three seeds. This evaluation excludes ambiguous formula-normalization cases and is not intended as broad evidence of out-of-distribution generalization; inspected sets are listed in Section E.1.

6 Conclusion

RECIPE reframes single-step inorganic retrosynthesis around the object that practitioners ultimately inspect: the complete precursor set. Rather than thresholding independently scored precursors, the framework separates high-recall precursor-pool construction from target-conditioned reranking of complete candidate precursor sets. This lets the model compare incomplete, over-complete, and chemically plausible near-miss sets directly under the same target. Empirically, the same pattern

appears on both Retro and Ceder. The generator raises precursor coverage and exact-set metrics over the strongest prior baseline, but the remaining errors show that recall-oriented precursor ranking is still only a relaxation of the complete-set decision. The Complete-Set Reranker then converts the high-recall pool into substantially stronger front-rank and deeper Combo retrieval by comparing variable-size candidate sets directly. The broader message is that synthesis support should prioritize complete candidate precursor sets, not only long lists of isolated precursors. This two-stage learning-task design provides a basis for exploring simpler and more effective complete-set scorers as larger and more diverse synthesis datasets become available.

Limitations. The current system is limited to single-step, closed-vocabulary precursor-set prediction and remains dependent on the generator's candidate pool. Full limitations are discussed in Section F.1.

Broader impacts. RECIPE is intended as decision support for expert chemists and for future automated experiment-design loops in self-driving or autonomous laboratories, where better precursor-set ranking may reduce synthesis trial-and-error. Incorrect or historically biased precursor suggestions still require expert review, safety checks, and downstream validation before any robotic or human experiment is executed. Additional discussion appears in Section F.2.

Acknowledgments and Disclosures

This work was partly supported by the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (Grant No. JYB2025XDXM411) and the Zhongguancun Academy (Grant No. 02012413).

References

[1] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *International Conference on Machine Learning*, pages 89–96, 2005.

[2] F. Bai, Y. Li, R. Wu, P. Wang, Y. Wang, B. H. Zhu, Y. Wang, T. Chou, J. Gao, R. Zhu, Y. Wen, Y. Yang, and Y. Chen. SafeLab: An interactive high-fidelity benchmark for embodied safety in scientific robotics. In *Forty-third International Conference on Machine Learning*, 2026.

[3] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: From pairwise approach to listwise approach. In *International Conference on Machine Learning*, pages 129–136, 2007.

[4] J. Chen, S. R. Cross, L. J. Miara, J.-J. Cho, Y. Wang, and W. Sun. Navigating phase diagram complexity to guide robotic inorganic materials synthesis. *Nature Synthesis*, 3(5):606–614, 2024.

[5] R. E. A. Goodall and A. A. Lee. Predicting materials properties without crystal structure: deep representation learning from stoichiometry. *Nature Communications*, 11(1):6280, 2020.

[6] A. Grover, E. Wang, A. Zweig, and S. Ermon. Stochastic optimization of sorting networks via continuous relaxations. In *International Conference on Learning Representations*, 2019.

[7] T. He, H. Huo, C. J. Bartel, Z. Wang, K. Cruse, and G. Ceder. Precursor recommendation for inorganic synthesis by machine learning materials similarity from scientific literature. *Science Advances*, 9(23):eadg8180, 2023.

[8] T. He, W. Sun, H. Huo, O. Kononova, Z. Rong, V. Tshitoyan, T. Botari, and G. Ceder. Similarity of precursors in solid-state synthesis as text-mined from scientific literature. *Chemistry of Materials*, 32(18):7861–7873, 2020.

[9] H. Huo, Z. Rong, O. Kononova, W. Sun, T. Botari, T. He, V. Tshitoyan, and G. Ceder. Semi-supervised machine-learning classification of materials synthesis procedures. *npj Computational Materials*, 5(1):62, 2019.

[10] E. Kim, Z. Jensen, A. van Grootel, K. Huang, M. Staib, S. Mysore, H.-S. Chang, E. Strubell, A. McCallum, S. Jegelka, et al. Inorganic materials synthesis planning with literature-trained neural networks. *Journal of Chemical Information and Modeling*, 60(3):1194–1201, 2020.

[11] S. Kim, Y. Jung, and J. Schrier. Large language models for inorganic synthesis predictions. *Journal of the American Chemical Society*, 146(29):19654–19659, 2024.

[12] S. Kim, J. Noh, G. H. Gu, S. Chen, and Y. Jung. Element-wise formulation of inorganic retrosynthesis. In *AI for Accelerated Materials Design NeurIPS Workshop*, 2022.

[13] O. Kononova, H. Huo, T. He, Z. Rong, T. Botari, W. Sun, V. Tshitoyan, and G. Ceder. Text-mined dataset of inorganic materials synthesis recipes. *Scientific Data*, 6(1):203, 2019.

[14] Z. Lan, Z. Zeng, B. Hong, Z. Liu, and F. Ma. RCsearcher: Reaction center identification in retrosynthesis via deep Q-learning. *Pattern Recognition*, 150:110318, 2024.

[15] M. J. McDermott, S. S. Dwaraknath, and K. A. Persson. A graph-based network for predicting chemical reaction pathways in solid-state materials synthesis. *Nature Communications*, 12(1):3097, 2021.

[16] A. Merchant, S. Batzner, S. S. Schoenholz, M. Aykol, G. Cheon, and E. D. Cubuk. Scaling deep learning for materials discovery. *Nature*, 624(7990):80–85, 2023.

[17] A. Miura, C. J. Bartel, Y. Goto, Y. Mizuguchi, C. Moriyoshi, Y. Kuroiwa, Y. Wang, T. Yaguchi, M. Shirai, M. Nagao, N. C. Rosero-Navarro, K. Tadanaga, G. Ceder, and W. Sun. Observing and modeling the sequential pairwise reactions that drive solid-state ceramic synthesis. *Advanced Materials*, 33(24):2100312, 2021.

[18] H. Noh, N. Lee, G. S. Na, and C. Park. Retrieval-retro: Retrieval-based inorganic retrosynthesis with expert knowledge. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[19] T. Prein, E. Pan, S. Haddouti, M. Lorenz, J. Jehkul, T. Wilk, C. Moran, M. P. Fotiadis, A. P. Toshev, E. Olivetti, and J. L. M. Rupp. Retro-rank-in: A ranking-based approach for inorganic materials synthesis planning, 2025.

[20] L. E. Ratcliff, S. Mohr, G. Huhs, T. Deutsch, M. Masella, and L. Genovese. Challenges in large scale quantum mechanical calculations. *WIREs Computational Molecular Science*, 7(1):e1290, 2017.

[21] Z. Song, S. Lu, M. Ju, Q. Zhou, and J. Wang. Accurate prediction of synthesizability and precursors of 3D crystal structures via large language models. *Nature Communications*, 16(1):6530, 2025.

[22] R. Swezey, A. Grover, B. Charron, and S. Ermon. PiRank: Scalable learning to rank via differentiable sorting. In *Advances in Neural Information Processing Systems*, volume 34, pages 21644–21654, 2021.

[23] N. J. Szymanski, B. Rendy, Y. Fei, R. E. Kumar, T. He, D. Milsted, M. J. McDermott, M. Gallant, E. D. Cubuk, A. Merchant, H. Kim, A. Jain, C. J. Bartel, K. Persson, Y. Zeng, and G. Ceder. An autonomous laboratory for the accelerated synthesis of novel materials. *Nature*, 624(7990):86–91, 2023.

[24] U. V. Ucak, I. Ashyrmamatov, J. Ko, and J. Lee. Retrosynthetic reaction pathway prediction through neural machine translation of atomic environments. *Nature Communications*, 13(1):1186, 2022.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[26] A. Y.-T. Wang, S. K. Kauwe, R. J. Murdock, and T. D. Sparks. Compositionally restricted attention-based network for materials property predictions. *npj Computational Materials*, 7(1):77, 2021.

[27] Z. Wang, K. Cruse, Y. Fei, A. Chia, Y. Zeng, H. Huo, T. He, B. Deng, O. Kononova, and G. Ceder. ULSA: Unified language of synthesis actions for the representation of inorganic synthesis protocols. *Digital Discovery*, 1(2):313–324, 2022.

[28] Z. Wang, O. Kononova, K. Cruse, T. He, H. Huo, Y. Fei, Y. Zeng, Y. Sun, Z. Cai, W. Sun, and G. Ceder. Dataset of solution-based inorganic materials synthesis procedures extracted from the scientific literature. *Scientific Data*, 9(1):231, 2022.

[29] L. Ward, A. Dunn, A. Faghaninia, N. E. R. Zimmermann, S. Bajaj, Q. Wang, J. Montoya, J. Chen, K. Bystrom, M. Dylla, et al. Matminer: An open source toolkit for materials data mining. *Computational Materials Science*, 152:60–69, 2018.

[30] K. Zeng, B. Yang, X. Zhao, Y. Zhang, F. Nie, X. Yang, Y. Jin, and Y. Xu. UAlign: pushing the limit of template-free retrosynthesis prediction with unsupervised SMILES alignment. *Journal of Cheminformatics*, 16(1):80, 2024.

[31] C. Zeni, R. Pinsler, D. Zügner, A. Fowler, M. Horton, X. Fu, Z. Wang, A. Shysheya, J. Crabbé, S. Ueda, R. Sordillo, L. Sun, J. Smith, B. Nguyen, H. Schulz, S. Lewis, C.-W. Huang, Z. Lu, Y. Zhou, H. Yang, H. Hao, J. Li, C. Yang, W. Li, R. Tomioka, and T. Xie. MatterGen: a generative model for inorganic materials design. *Nature*, 639(8055):624–632, 2025.

A Datasets and Evaluation

A.1 Dataset Statistics

The Retrieval-Retro and Ceder corpora are dominated by multimetal targets and multi-precursor sets, making precursor-set prediction a genuine Combo Accuracy problem rather than a single-precursor classification problem. Tables 6 and 7 report precursor vocabulary size and ground-truth precursor-set size. The separate metal-count table describes target-formula composition only and is not used as the definition of $|S^\star|$.

Table 6: Dataset summary statistics. Retro denotes the Retrieval-Retro benchmark used for the main results.

Dataset	Train	Val.	Test	Total	Unique targets	Precursor vocab.	Mean $\lvert S^\star\rvert$
Retro	24,034	1,842	2,558	28,434	28,221	798	3.30
Ceder	44,736	2,254	2,934	49,924	33,046	1,037	2.99

Table 7: Ground-truth precursor-set size distribution over each full dataset.

$\lvert S^\star\rvert$	Retro	Ceder
2	4,864 (17.1%)	16,320 (32.7%)
3	12,504 (44.0%)	20,369 (40.8%)
4	8,809 (31.0%)	10,729 (21.5%)
5	2,257 (7.9%)	2,506 (5.0%)

Table 8: Distribution of metal element count in target compounds.

Metal count	Retrieval-Retro	Ceder
0	180 (0.6%)	263 (0.5%)
1	1,994 (7.0%)	4,218 (8.4%)
2	7,463 (26.2%)	18,714 (37.5%)
3	10,838 (38.1%)	17,090 (34.2%)
4	6,527 (23.0%)	8,047 (16.1%)
5	1,432 (5.0%)	1,592 (3.2%)
Mean	2.91	2.67

A.2 Metric Details

Let $\pi_i(p)$ be the one-based rank of precursor p for target i. The benchmark precursor-coverage diagnostic is

$$\text{Coverage@}k = \frac{100}{N}\sum_i \mathbf{1}\left[\max_{p\in S_i^\star} \pi_i(p) \le \max(k, |S_i^\star|)\right].$$

In particular, Top-1 coverage permits $|S_i^\star|$ inspected precursors and uses the label size only for scoring this diagnostic. Prec. MRR is $100(\sum_i |S_i^\star|)^{-1}\sum_i \sum_{p\in S_i^\star} 1/\pi_i(p)$. Combo metrics instead use the exact-set rank defined in Section 3, with all candidate misses scored zero. An exact match requires equality of unordered precursor-ID sets after dataset normalization; plausible alternative recipes are not counted as correct without a matching label.

A.3 Candidate Availability and Error Decomposition

Availability and error attribution. Let $a_i = \mathbf{1}[S_i^\star \in \Omega(x_i)]$ and $A = N^{-1}\sum_i a_i$. Writing Top-1 accuracy as a fraction $C_1 = \text{C@1}/100$, its conditional counterpart is $C_{1|A} = \sum_i \mathbf{1}[r_i = 1]/\sum_i a_i$. Whenever $A > 0$,

$$C_1 = AC_{1|A}, \qquad 1 - C_1 = \underbrace{1 - A}_{\text{candidate misses}} + \underbrace{A - C_1}_{\text{available but not ranked first}}. \tag{10}$$

The two error terms use the full test denominator and are mutually exclusive. Availability is an upper bound on exact-set recovery within the constructed space, whereas conditional accuracy evaluates ordering after the truth is present. Reporting both avoids crediting a ranker for excluding difficult targets. This is a diagnostic decomposition, not a causal attribution of gains to individual modules.

Table 9: Exact-set availability and conditional C@1 on Retro with fixed construction. Retrained denotes a RECIPE reranker trained on fixed Retrieval-Retro outputs. Conditional accuracy uses each generator's available subset, not a common intersection. Values are mean ± sample SD across three reranker seeds.

Scorer	Available	Avail.%	C@1 (all)	C@1 (avail.)
Retrieval-Retro generator				
RECIPE Reranker Retrained, chunk-512	2,399/2,558	93.78	68.27 ± 0.59	72.79 ± 0.63
RECIPE generator				
RECIPE Reranker Chunk-512	2,449/2,558	95.74	71.70 ± 0.10	74.89 ± 0.11
RECIPE Reranker Full-list (final)	2,449/2,558	95.74	71.93 ± 0.63	75.13 ± 0.66

B Implementation and Reproducibility

B.1 Implementation Details

Stage 1 uses separate target and precursor formula encoders with two layers, four heads, 96-dimensional hidden states and 192-dimensional projected outputs. Formula inputs use 32-dimensional element embeddings and stoichiometric fractions, with learned CLS pooling and no physical/valence or charge features. Stage 2 uses a separate 64-dimensional formula encoder, four target-conditioned slots, a 128-dimensional descriptor projection, a 16-dimensional size embedding, and a two-layer, four-head list Transformer with 192-dimensional states. The generator is frozen during Stage-2 training. The selected Retro configuration disables charge features, slot-to-candidate cross-attention and mixture pooling.

Training lists contain 96 sets with two augmentations. We use temperature 2, margin 0.3, margin weight 0.08, coverage weight 0.03 and label smoothing 0.03. Epochs and batch sizes are summarized in Table 11; the seed protocol is given in Section B.4.

Coarse elemental conventions. The compatibility screen excludes the implementation's designated volatile elements $\{\mathrm{H}, \mathrm{C}, \mathrm{N}, \mathrm{O}, \mathrm{F}, \mathrm{Cl}, \mathrm{S}, \mathrm{Br}, \mathrm{I}\}$ before comparing precursor and target element sets. This is a fixed screening heuristic, not a thermodynamic or safety model. Descriptor key-element masks instead omit only $\{\mathrm{H}, \mathrm{C}, \mathrm{N}\}$. The distinction is relevant when interpreting coverage versus unwanted-element features.

Ceder configuration. Ceder uses the same descriptor schema with a dataset-specific feature subset and optional precursor-formula pooling enabled. Detailed settings are provided in the code.

B.2 Implementation Overview

Each set token combines target-conditioned representations, projected descriptors and set size, with optional pooling of candidate precursor-formula representations. A list Transformer and shared scoring head produce raw set scores, including the learned generator-score residual. Stage 2 has its own formula encoder. The auxiliary head predicts target-element presence from the target-conditioned slot states before the list Transformer. The selected configurations are specified in Section B.1.

B.3 Candidate-Set Descriptor Groups

The six groups in Table 10 partition the 374 descriptor positions. The element masks each have 118 dimensions. Formula-derived fields are shared across vocabularies. Only the pair-frequency, anion-type usage and individual-precursor frequency fields use training-corpus counts.

Table 10: The fixed descriptor schema. The coverage group contains two masks, not all three.

Group	Dim.	Features
Generator probabilities	5	Geometric mean, maximum, minimum, probability gap, and normalized entropy.
Coverage / target overlap	240	Covered-target-element and target-element masks (118 dimensions each); key-element coverage, non-volatile coverage, complete-coverage/no-unwanted indicator, and uncovered non-volatile fraction (4 scalars).
Unwanted elements	119	Unwanted-element mask (118 dimensions) and count (1 scalar).
Co-occurrence / priors	4	Mean pair log-frequency, formula-derived decomposition-type fraction, smoothed training anion-type usage, and mean precursor log-frequency.
Stoichiometry	3	Candidate-set size divided by eight, stoichiometric cosine, and candidate-set-size/target-element-count agreement.
Mapping style	3	Source-count entropy per target non-volatile element, multi-source fraction, and pairwise non-volatile-element Jaccard overlap.

The size/count agreement feature is $1/(1 + ||S| - |E_{\mathrm{nv}}(x)||)$, where S is the candidate precursor set and $E_{\mathrm{nv}}(x)$ is the set of non-volatile element types in the target formula under the implementation's convention. It uses neither the ground-truth precursor set nor its size. The chemistry-prior features are zero when the target has no such elements.

B.4 Reproducibility and Assets

The implementation is organized around the Precursor Candidate Generator and the Complete-Set Reranker. Both modules read the benchmark split files, cache parsed formula tensors, and write prediction files for downstream aggregation. Throughout the paper, seeds 1, 2 and 3 denote 20260503, 20260504 and 20260505, respectively. Generator results vary all three Stage-1 seeds. Reranker results vary all three Stage-2 seeds with Stage-1 outputs fixed at seed 2; their uncertainty therefore measures Stage-2 variation with a fixed generator. Benchmark results report mean and sample standard deviation, while the OOD evaluation reports mean and population standard deviation over three runs. Code repository: `https://github.com/Xuwznln/RECIPE`. Model training and runtime profiling use NVIDIA A100 GPUs. Table 11 summarizes the main run settings.

For the construction profiles in Table 13, standard settings use seed 1. The size-2–8 stress timing combines construction from seed 3 with mean scoring over all three checkpoints; accuracy always uses all three.

Table 11: Training settings for the two learned modules and the OOD evaluation setup. Epochs and batch size refer to training; the OOD row involves evaluation only.

Component	Dataset	Seeds	Epochs	Batch size
Candidate Generator	Retro/Ceder	3	40	128
Complete-Set Reranker	Retro/Ceder	3	3	16
OOD evaluation	20 cases	3	–	–

The Retro benchmark follows the cleaned chronological split released by Retrieval-Retro [18]. The Ceder benchmark is derived from the public text-mined inorganic synthesis corpus [13]. The expanded OOD evaluation is documented in Section E.1; each expert-annotated example records the target formula, material family, literature-reported precursor set, and source DOI used for exact-set matching.

C Additional Comparisons

C.1 Data Organization for Baseline Rerankers

Each training row represents one candidate precursor set for a target using the same 374 descriptors and fixed Stage-1 probabilities. For each of 23,577 training targets whose exact set is available, we include that set as the positive and up to 95 uniformly sampled negative sets, yielding 2,227,992 rows. Positive and negative rows receive equal total weight within each target. At evaluation, all scorers rank the same exhaustively enumerated candidate sets, and metrics use the full test denominator.

C.2 Same-Enumeration Controls

The generator-plus-filter control discussed in Section 5 applies the pruning rule directly to generator-threshold predictions, testing whether label-free pruning alone explains the reranker gain. Table 12 adds stricter scorer-only controls: after using the same generator pool, compatibility pruning, pool cap, and candidate-set enumeration as the reranker, each independent scorer ranks the exact same candidate-set list using only member generator probabilities and, for the final independent row, a validation-derived set-size prior. The compact diagnostic artifact stores each target's candidate count, ground-truth index, true set size, availability, and ranks under the independent scorers; the full candidate-list export is treated as a regeneration artifact rather than the default supplemental object.

Table 12: Same-enumeration scorer controls on Retro. All independent scorer rows use the same enumerated candidate-set list as the reranker and are single diagnostic runs on that exported list. Reranker metrics are the submitted chunk-512 three-seed means with sample standard deviations. Availability is measured on the diagnostic enumeration and is not a ranking metric.

Scorer / diagnostic	C@1 ↑	C@5 ↑	C@20 ↑	Combo MRR ↑	Set availability ↑
Product / sum log generator probabilities	11.65	43.20	75.33	23.24	95.74
Mean log generator probability	11.65	57.43	83.35	32.43	95.74
Validation-size-prior log product	39.87	65.60	76.11	47.63	95.74
Learned Complete-Set Reranker	**71.70** (±0.10)	**84.52** (±1.48)	**89.82** (±1.74)	**77.43** (±0.69)	95.74 –

C.3 Additional Performance Results

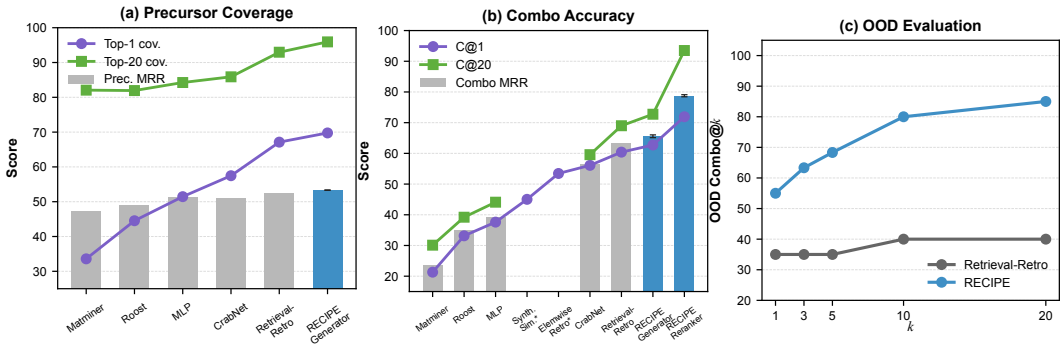


Figure 3: Retro and OOD summary. (a) Precursor Top-1/Top-20 and Prec. MRR. (b) Exact-set C@1/C@20 and Combo MRR; * rows show reported C@1 only. (c) OOD Combo@k; blue denotes RECIPE.

The Retro and Ceder benchmark plots use the final full-list results, emphasizing how the generator, reranker, ablations, and OOD evaluation differ across metrics.

C.3.1 Ceder Benchmark Curves

Figure 4 summarizes the Ceder benchmark results in Table 2.

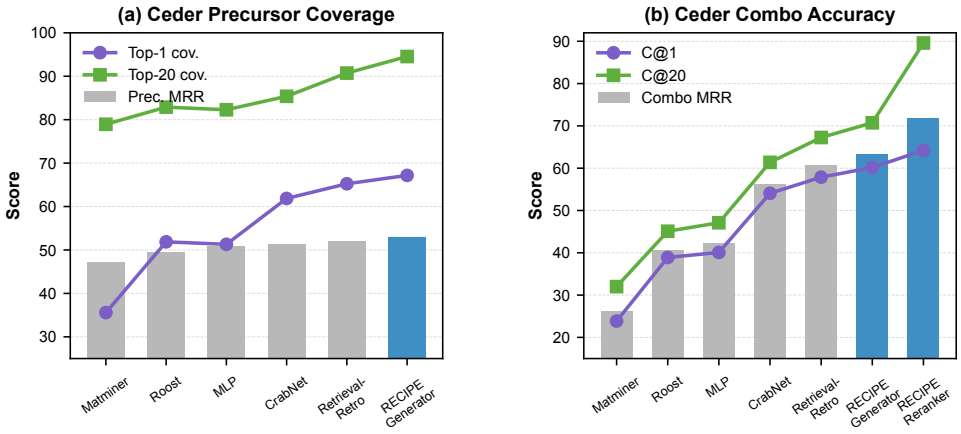


Figure 4: Ceder benchmark results corresponding to Table 2. Left: Top-1/Top-20 precursor-coverage curves with Prec. MRR bars. Right: exact-set C@1/C@20 curves with Combo MRR bars.

D Scalability and Inference

D.1 Candidate Construction and Scalability

Table 13: Candidate-construction sensitivity (cap / filter / size range). Retro results use fixed checkpoints with chunk-512 inference. All timings exclude Stage 1. Accuracy uses three checkpoints; timing seeds follow Section B.4. Array memory is maximum target-local storage, not process peak.

Cap / filter / sizes	C@1	C@3	C@5	C@10	C@20	Combo MRR
10 / on / 2–5	71.63 ± 0.77	84.02 ± 0.49	86.68 ± 0.54	88.90 ± 1.14	89.85 ± 1.43	78.08 ± 0.37
12 / on / 2–5	71.33 ± 0.53	83.39 ± 0.65	85.85 ± 0.77	88.74 ± 1.19	90.17 ± 1.32	77.65 ± 0.09
15 / on / 2–5	71.70 ± 0.10	82.04 ± 1.40	84.53 ± 1.48	87.89 ± 1.52	89.82 ± 1.74	77.43 ± 0.69
15 / off / 2–5	71.34 ± 0.40	80.92 ± 1.24	83.35 ± 1.33	86.20 ± 1.96	88.06 ± 2.23	76.61 ± 0.62
15 / on / 2–4	65.31 ± 0.54	75.02 ± 0.75	77.47 ± 1.02	80.32 ± 1.06	82.19 ± 1.28	70.66 ± 0.16
15 / on / 2–8	71.68 ± 0.11	82.00 ± 1.43	84.38 ± 1.64	87.69 ± 1.76	89.68 ± 1.86	77.36 ± 0.74

Setting	Availability (%)	Sets: mean/median/max	s/target	MiB
10 / on / 2–5	92.42	542/627/627	0.019	1.3
12 / on / 2–5	94.33	1,230/1,573/1,573	0.038	3.2
15 / on / 2–5	95.74	3,170/4,928/4,928	0.085	10.1
15 / off / 2–5	94.68	4,928/4,928/4,928	0.129	10.1
15 / on / 2–4	86.71	1,289/1,925/1,925	0.037	3.9
15 / on / 2–8	95.74	13,721/22,803/22,803	0.266	46.5

We retain the Top-30 shortlist and vary one subsequent construction setting at a time without retraining, using the same frozen rerankers and full test denominator (Table 13). Median equals maximum under cap 15 because more than half the targets reach the cap, which deterministically yields 4,928 size-2–5 subsets. Turning filtering off changes which top-ranked precursors fill the cap; it can reduce truth availability even though more sets are enumerated. Truncating to sizes 2–4 also excludes true size-five sets.

Choice of pool cap, filtering, and enumeration range. Cap 15 gives 95.74% exact-set availability, compared with 92.42% and 94.33% at caps 10 and 12. It has the highest mean C@1 among these caps (71.70 versus 71.63 and 71.33), while smaller caps are cheaper and have slightly higher mean C@20. Thus cap 15 favors candidate recall rather than dominating every accuracy–cost trade-off. At the same cap, removing the compatibility filter changes C@1/C@20 from 71.70/89.82 to 71.34/88.06 and increases mean candidates from 3,170 to 4,928; filtering is a modest refinement, not the main

source of the reranking gain. Exhaustive sizes 2–5 cover all observed ground-truth precursor-set sizes. Restricting enumeration to sizes 2–4 reduces availability to 86.71% and C@1/C@20 to 65.31/82.19; extending to 2–8 gives 71.68/89.68 at 0.266 rather than 0.085 seconds per target. These controls support the selected range's coverage–cost trade-off. They do not establish an optimal pre-filter shortlist size: Top-30 is held fixed in this sweep.

The size-2–8 setting is a computational stress test, not an asserted upper bound on recipe size or an evaluation on larger labeled recipes. All current labels contain 2–5 precursors. Target elemental diversity in Materials Project is not a bound on the number of required precursors, so it is not used to justify such a bound.

Profiles use an A100 with 24 CPU construction workers and exclude Stage 1. Build-plus-score time is amortized over 2,558 targets. Target-local array memory is the size of one target's candidate arrays, not total host memory or peak GPU memory. The measured batch implementation materializes the entire test cache: peak host RSS is about 18.9 GiB for the default run and 80.2 GiB for the size-2–8 stress run. Streaming target construction could reduce this footprint, but that implementation is not measured here. CUDA allocation is reported separately below.

D.2 Chunk Context and Full-List Inference

Chunked inference applies attention within each chunk and globally ranks all candidates by their raw scores. Full-list inference lets every candidate attend to the complete list. We compare these settings using the same three checkpoints and fixed candidate sets, without retraining. Tables 14 and 15 report the Retro and Ceder results.

Table 14: Retro inference-context sweep with the same three checkpoints, generator probabilities and candidate sets. Upper panel: mean and sample SD of canonical-order evaluations. Lower panel: mean scoring latency over three A100 profiles and all targets; maximum CUDA allocated memory (not reserved memory), excluding host arrays. Chunk 8,192 is full-list inference. This profiling rerun gives chunk-512 C@5/C@10 of 84.53/87.89, compared with the archived 84.52/87.91.

Chunk	C@1	C@3	C@5	C@10	C@20	Combo MRR
96	70.25 ± 0.68	78.29 ± 1.83	80.48 ± 2.01	83.70 ± 3.00	85.76 ± 3.34	74.99 ± 1.29
256	71.28 ± 0.18	80.51 ± 1.54	83.33 ± 2.14	86.50 ± 2.22	88.31 ± 2.22	76.61 ± 0.88
512	71.70 ± 0.10	82.04 ± 1.40	84.53 ± 1.48	87.89 ± 1.52	89.82 ± 1.74	77.43 ± 0.69
1,024	71.80 ± 0.28	83.46 ± 0.80	86.30 ± 0.92	89.63 ± 0.93	91.13 ± 1.20	78.13 ± 0.26
8,192	71.93 ± 0.63	84.25 ± 0.14	87.91 ± 0.06	91.22 ± 0.33	93.50 ± 0.26	78.77 ± 0.34

Chunk	Attention context	ms/target	GPU memory (MiB)
96	Local	251.4	46
256	Local	99.7	52
512	Local	51.3	62
1,024	Local	28.3	80
8,192	Full list	9.9	220

Table 15: Ceder inference comparison with fixed checkpoints and candidate construction. Mean $\pm$ sample SD over three checkpoints; full test denominator of 2,934 targets.

Context	C@1	C@3	C@5	C@10	C@20	Combo MRR
Chunk 512	63.38 ± 1.50	73.82 ± 4.11	78.02 ± 4.82	83.00 ± 4.74	86.31 ± 3.65	70.03 ± 2.63
Full list	64.21 ± 3.73	76.07 ± 5.82	80.90 ± 6.49	86.15 ± 4.95	89.60 ± 2.83	71.63 ± 4.45

E OOD Evaluation

E.1 OOD Evaluation Details

The expanded OOD evaluation contains 20 literature-grounded target compounds after removing examples that overlap with benchmark splits or have ambiguous formula normalization. The cases were selected to cover sulfides, nitrides, borates, phosphates, arsenates, germanates, perovskites, and doped oxides, and were not used for model selection. Table 17 lists the target, material family, ground-truth precursor set, and DOI of the source publication.

E.2 OOD Seed-Level Results

Table 16 uses the seed numbering in Section B.4 and reports the seed-level results behind the mean OOD row in Table 5.

Table 16: Seed-level RECIPE results on the expanded 20-case OOD evaluation.

Seed	C@1	C@3	C@5	C@10	C@20	Combo MRR	Fail >20
1	55.00	65.00	70.00	90.00	100.00	63.09	0
2	55.00	65.00	65.00	80.00	85.00	61.82	3
3	55.00	60.00	70.00	70.00	70.00	59.16	6

E.3 Literature-Grounded Cases

Table 17: Out-of-distribution examples used in Table 5. The source DOI identifies the publication from which the target compound and ground-truth precursor set were recorded.

#	Target	Family	Ground-truth precursors	Source DOI
1	$Na_2CdGe_2S_6$	Quaternary sulfide	Na + CdS + Ge + S	10.1039/c7dt00087a
2	$Na_2CdGe_2Se_6$	Quaternary selenide	Na + CdSe + Ge + Se	10.1039/c7dt00087a
3	$CaMg_2AlN_3$	Quaternary nitride	Mg_3N_2 + Ca_3N_2 + AlN	10.1039/c4dt03283g
4	$(SiO_2)_{50}(Na_2O)_{25}(B_2O_3)_{25}$	Borosilicate glass	SiO_2 + Na_2CO_3 + H_3BO_3	10.1039/c7ra04062h
5	$Pb_2Zn_3Te_2P_2O_{14}$	Tellurite-phosphate	PbO + TeO_2 + ZnO + $NH_4H_2PO_4$	10.1039/c6dt00058d
6	$Ba_2Zn_2TeP_2O_{11}$	Tellurite-phosphate	$BaCO_3$ + TeO_2 + ZnO + $NH_4H_2PO_4$	10.1039/c6dt00058d
7	Ba_2GeS_4	Barium germanium sulfide	BaS + GeS_2	10.1039/c5ra00264h
8	Mg_2SnS_4	Magnesium tin sulfide	MgS + SnS_2	10.1039/c5ra00264h
9	$Pb_2Mg(PO_4)_2$	Lead magnesium phosphate	PbO + MgO + $NH_4H_2PO_4$	10.1039/c7dt03055j
10	$NaCa_4B_3O_9$	Alkaline-earth borate	Na_2CO_3 + $CaCO_3$ + H_3BO_3	10.1039/c5cp04420k
11	$CoAs_2O_4$	Cobalt arsenate	$Co(OH)_2$ + As_2O_3	10.1039/c4ra16122j
12	$NiAs_2O_4$	Nickel arsenate	$Ni(NO_3)_2$ + As_2O_3	10.1039/c4ra16122j
13	$Ho_2Ge_2O_7$	Rare-earth germanate	GeO_2 + Ho_2O_3	10.1039/c7dt03370b
14	$LiSrTb_2(BO_3)_3$	Rare-earth borate	Tb_4O_7 + $SrCO_3$ + H_3BO_3 + Li_2CO_3	10.1039/c5nj01913c
15	$BaGa_4O_7$	Barium gallate	$BaCO_3$ + Ga_2O_3	10.1039/c5ce01101a
16	$CsPrNb_2O_7$	Cesium praseodymium niobate	Pr_6O_{11} + Nb_2O_5 + Cs_2CO_3	10.1039/c4dt03882g
17	$Cs_2B_7O_9(OH)_5$	Cesium borate hydroxide	Cs_2CO_3 + H_3BO_3	10.1039/c7ce00450h
18	Ca_2AlTaO_6	Double perovskite	Ta_2O_5 + Al_2O_3 + $CaCO_3$	10.1016/j.jallcom.2015.01.310
19	$Li_3Ti_{1.25}O_4$	Spinel titanate	TiO_2 + Li_2CO_3	10.14233/ajchem.2013.12934B
20	$Ca_{4.23}Sr_{4.5}Eu_{0.27}Sc(PO_4)_7$	Eu-doped phosphate	$SrCO_3$ + Eu_2O_3 + $CaCO_3$ + $NH_4H_2PO_4$ + Sc_2O_3	10.1039/C8TC01825A

F Limitations and Broader Impacts

F.1 Limitations

RECIPE is a single-step, closed-vocabulary retrosynthesis system. If the generator fails to include a required precursor in the candidate pool, the reranker cannot recover the correct precursor set. The model also predicts precursor sets rather than full executable syntheses: reaction conditions, byproducts, thermodynamic stability, kinetic accessibility, intermediate phases, and multi-step routes remain outside the current formulation. The preliminary OOD evaluation is not a complete measure of generalization across inorganic chemistry. In practice, ranked precursor sets may reduce synthesis trial-and-error, but incorrect suggestions remain possible and should be treated as decision support for expert inspection and downstream validation rather than as automatically verified synthesis routes.

Available synthesis datasets remain limited in scale and diversity. High-quality extraction from the literature could expand these datasets and support the development of simpler and more effective complete-set ranking methods.

F.2 Broader Impacts

Better Combo Accuracy for precursor sets can help reduce trial-and-error in inorganic materials synthesis and make computationally proposed materials more actionable. One natural use case is automated experiment-plan generation: a materials discovery system may propose a target composition, rank plausible precursor sets, and pass a small number of candidates to a human chemist or to a self-driving laboratory for condition selection, robotic execution, and characterization feedback [23]. In that setting, variable-size set-level ranking can serve as a filtering and prioritization layer, potentially reducing wasted reagents, instrument time, and failed synthesis attempts.

These benefits should not be read as a license for unsupervised experimental execution. The work predicts precursor sets, not full validated recipes. It does not determine safe reaction conditions, byproducts, gas evolution, incompatibilities, thermodynamic feasibility, kinetic accessibility, or hazards associated with scale-up. Incorrect precursor suggestions could waste experimental resources, bias automated laboratories toward historically common routes, or cause a closed-loop planner to repeatedly explore a narrow region of chemistry. Deployment in automated or autonomous laboratories should therefore include uncertainty estimates, expert review, inventory and safety constraints, thermodynamic and condition checks, and stopping rules that prevent low-confidence predictions from being translated directly into robotic experiments.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the paper's main contributions, including the variable-size set-level ranking formulation, the RECIPE framework with a Precursor Candidate Generator and Complete-Set Reranker, and the claimed empirical gains on Retrieval-Retro and Ceder. These claims are supported by the problem setup, method, experimental results, and ablations in Sections 3 to 5, while the 20-case OOD result is explicitly scoped as a preliminary evaluation.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 includes a concise **Limitations** paragraph covering the single-step, closed-vocabulary scope and dependence on the generator candidate pool. Section F.1 provides the detailed limitations discussion in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper is an application-oriented empirical methods paper and does not present theoretical results, theorems, or proofs.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper gives the benchmark splits, metrics, candidate construction, model components, losses, multi-seed protocol, and run settings in Sections B.4 and 3 to 5. The supplemental package contains scripts for training, evaluation, aggregation, and figure generation.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code repository: https://github.com/Xuwznln/RECIPE.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The main text and appendix specify the benchmark protocol, variable-size candidate-set evaluation, model objectives, multi-seed protocol, and recorded candidate-pool and loss settings; see Sections B.4 and 3 to 5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Main benchmark results are run with multiple independent seeds and report means with sample standard deviations; the text states that the error bars are sample standard deviations over seeds. The OOD evaluation reports mean and population standard deviation across three independent seeds.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section B.4 and Tables 11, 13 and 14 report run settings, CPU workers, and inference time and memory measured on NVIDIA A100 GPUs. Complete historical training and hyperparameter-search compute totals are not available in this review draft.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses literature-derived materials synthesis benchmarks and does not involve human subjects, personal data, surveillance, or deployed decision systems. The proposed method is presented as decision support for expert inspection, not as an autonomous prescription system.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Section 6 includes a concise **Broader impacts** paragraph on decision support, automated experiment-plan generation, self-driving laboratories, reduced synthesis trial-and-error, and the need for expert validation. Section F.2 provides the detailed discussion of positive and negative impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a high-risk generative model, scraped personal-content dataset, or asset with obvious dual-use misuse risk; it uses materials synthesis benchmarks and prepares research code for evaluation.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper credits the public benchmark datasets and baseline sources, including Retrieval-Retro and the Ceder synthesis corpus, and the supplemental manifest records the source links and license/terms information used for these public research assets.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We introduce an additional expert-annotated 20-case OOD evaluation set. Section E.1 documents the target formulas, material families, literature-reported precursor sets, and source DOIs.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing, user studies, or human-subject experiments.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects, user studies, crowdsourcing, or participant data.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not part of the proposed method, data generation, experiment execution, or evaluation pipeline. They were used only for wording polishing and formatting assistance, which does not affect the core methodology, scientific rigor, or originality of the research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.