BENCHMARKING AUGMENTATION STRATEGIES FOR LLM-BASED SOLID-STATE SYNTHESIS PREDICTION

Thorben Prein[1,3], Elton Pan[2], Anass Al Ammiri[1,3], Alan Albert Piovesana[1,3], Behsad Riemer[1,3], Elsa Olivetti[2], Jennifer L.M. Rupp[1]

[1]TUM, [2]MIT, [3]TUM.ai

ABSTRACT

Identifying synthesis recipes for new inorganic materials remains a major bottleneck in materials discovery. We investigate whether large language models (LLMs) can improve solid-state synthesis prediction through three augmentation strategies: retrieval-augmented generation (RAG) from the literature, the use of domain-specific thermodynamic tools, and multi-step, test-time compute workflows such as debate, self-reflection, and sequential pipelines. When evaluating on 674 literature-derived targets, we find that retrieving relevant synthesis precedents is the most effective strategy, improving top-10 precursor accuracy from 77.0% to 83.5%. Thermodynamic tools also improve performance (80.6%), but provide little additional benefit when RAG is already used (82.9% on Gemini 3 Flash, 77.5% on Gemini 2 Flash). By contrast, test-time compute does not improve performance, and sequential multi-agent workflows often reduce accuracy because errors introduced in earlier stages propagate downstream, causing later steps to mis-rank candidates or overwrite correct answers. Our results show that, for solid-state synthesis prediction, providing models with relevant domain information is more effective than increasing test-time compute through multi-agent deliberation.

1 INTRODUCTION

Discovering new inorganic materials is central to progress in energy, electronics, medicine, and space technologies. Yet synthesis remains a major bottleneck: although computational screening has identified hundreds of thousands of thermodynamically stable compounds Merchant et al. (2023), translating these predictions into real materials still requires identifying suitable precursors and heat-treatment schedules. Because synthesis attempts are costly and often fail, reliable guidance is especially important for autonomous and closed-loop laboratories Butler et al. (2018).

Materials informatics has begun to address this challenge by combining large datasets with predictive models to support experimental design and reduce failure rates Prein et al. (2025a); Noh et al. (2024). Key resources include the Materials Project for thermodynamics-guided screening at scale Jain et al. (2013) and text-mined synthesis datasets extracted from the literature Kim et al. (2017); Kononova et al. (2019). These resources have enabled growing use of machine learning for synthesis planning and precursor prediction Butler et al. (2018); He et al. (2023).

Building on these foundations, large language models (LLMs) have recently been applied to inorganic synthesis planning Miret and Krishnan (2025); Jiang et al. (2025); Hu and Buehler (2023). Prein *et al.* showed that general-purpose LLMs can predict synthesis conditions without task-specific fine-tuning, achieving 53.8% top-1 precursor exact-match accuracy Prein et al. (2025b). In our benchmark, modern Gemini models achieve single-call top-1 accuracies of 60.4% to 62.2% on a 674-item test set, but substantial limitations remain. Inorganic synthesis depends on thermodynamic constraints and phase equilibria that are difficult to infer reliably from text alone Jain et al. (2013), and models may therefore propose chemically plausible yet physically inconsistent precursor sets without explicit grounding Song et al. (2025).

While RAG and tool use have each been studied in isolation, their relative and combined value for solid-state synthesis prediction remains unclear Oche and Biswas (2025); Mostafa et al. (2024); Luo et al. (2025). In this work, we investigate three augmentation strategies: (1) **RAG** over literature recipes, (2) **thermodynamic tools** via the Materials Project API, and (3) **multi-step, test-time compute workflows** including debate, self-reflection, and sequential pipelines.

2 METHODS

2.1 TASK

Given a target formula, the model generates 20 ranked candidate precursor sets and is evaluated on the top K=10. Each candidate is an unordered set of precursors, $\mathcal{P} = \{p_1, \ldots, p_n\}$.

Dataset: We evaluate on 674 literature-derived targets with ground-truth precursor sets Kononova et al. (2019). The remaining train/validation split forms a RAG corpus of 10,764 recipes, with targets partitioned by formula so that no held-out test formula appears in the corpus. During retrieval, we exclude only exact matches to the test target formula; compositionally similar materials (e.g., $Pr_{0.9}MnO_3$ when the target is $PrMnO_3$) may still be returned as precedents. Full prompts and tool specifications are provided in Appendix A.

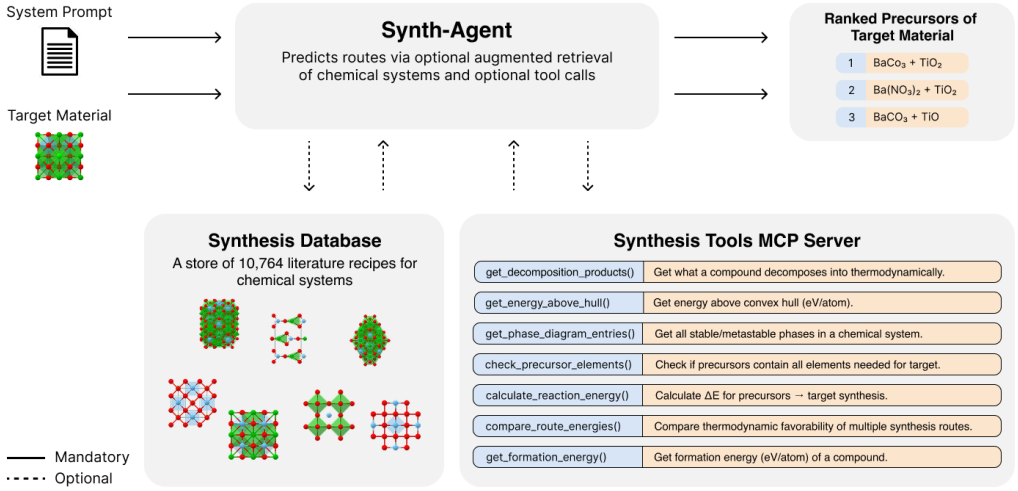

Figure 1: **Multi-tool pipeline.** SYNTHAGENTS takes a system prompt and target material (solid arrows) and outputs ranked precursors. Optional: providing the agent with a Synthesis Database (10,764 recipes) for RAG and Materials Project API for thermodynamic and stoichiometric data.

Metrics: We report precursor exact-match accuracy at $K \in \{1, 3, 5, 10\}$ Prein et al. (2025b):

$$\text{Acc@}K = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\left[\mathcal{P}_{\text{gt}}^{(i)} \in \{\mathcal{P}_1^{(i)}, \ldots, \mathcal{P}_K^{(i)}\}\right] \tag{1}$$

2.2 WORKFLOW VARIANTS

We implemented all workflows directly, without relying on orchestration frameworks such as LangChain, to improve reproducibility. Code and data are available at `github.com/tum-ai/SynthAgent`. We vary two factors: the amount of inference-time computation (a single response versus multi-step workflows with multiple roles or rounds) and the use of external augmentation through retrieval, Materials Project tools, or both (Fig. 1). The baseline is a single-response workflow with no retrieval and no tools: a single role receives the target formula and returns a ranked list of synthesis routes. We then evaluate multi-step workflows including self-reflection, debate, and a sequential pipeline with and without retrieval and Materials Project tools.

2.2.1 TEST-TIME COMPUTE

Self-reflection: A single model instance first proposes a ranked list of routes and then iteratively critiques its own proposal. If the critique identifies issues, the model revises the list, otherwise, the

process stops. We allow up to two critique-revision cycles and terminate early when no major issues are found. This workflow uses no retrieval and no external tools.

Debate: An Advocate proposes routes, and a Skeptic looks for flaws. The Advocate then revises or defends, and this exchange runs for a few rounds. After that, a Judge sees the full exchange and outputs the final ranking. In our setup, the Advocate and Skeptic do not use external tools.

Sequential pipeline: The Chemist proposes routes (and may receive retrieved recipes when retrieval is on), the Critic evaluates it, and the Integrator produces the final list from both. In the base setup neither the Chemist nor the Critic calls external tools. In "sequential with tools", both may call the Materials Project during their steps, and the Integrator only assembles the final list.

Ensemble voting: We generate N independent proposals and perform majority voting on rank-1 precursor sets. The final output is the full proposal whose rank-1 precursors received the most votes, combining multiple diverse model calls into a single answer. No retrieval or tools are used.

2.2.2 Information augmentation

Retrieval-augmented generation (RAG): For each target formula, we retrieve the k=5 most similar synthesis recipes from a disjoint train-set of the data Kononova et al. (2019). Similarity is defined as compositional cosine, in which each formula is represented by a vector of elemental stoichiometric fractions (normalized to sum to 1), for target t and candidate c

$$S(t, c) = \frac{\mathbf{v}_t \cdot \mathbf{v}_c}{\|\mathbf{v}_t\|\|\mathbf{v}_c\|} \tag{2}$$

with $\mathbf{v}_t$, $\mathbf{v}_c$ the respective vectors. This captures both elemental overlap and stoichiometric proportions; for example, $Ba_{0.6}Sr_{0.4}TiO_3$ is closer to $Ba_{0.5}Sr_{0.5}TiO_3$ than to $BaTiO_3$. The retrieved recipes are inserted into the prompt as in-context examples.

Domain-specific tools: When tools are enabled, the pipeline can query the Materials Project Jain et al. (2013) for thermodynamic and stoichiometric data. These tools provide the formation energy per atom (negative values indicate stable compounds), energy above the convex hull (zero for stable compounds and positive for metastable ones), and phase-diagram entries for a chemical system defined by its constituent elements. They can also retrieve the decomposition products of a compound and check whether a proposed precursor set contains all elements required for the target. The tools used in our experiments are listed in Appendix A.

3 Results and Discussion

Table 1 summarizes precursor accuracy across workflows on the full test set (n=674). We organize results into two categories: (i) *test-time compute* methods that add multi-step reasoning or multi-agent interaction, and (ii) *information augmentation* methods that provide domain knowledge via retrieval or thermodynamic tools from the Materials Project. Cross-category comparisons are qualitative only.

3.1 Test-time compute does not improve model performance

Table 1 shows that increasing test-time compute through additional steps or agents does not reliably improve synthesis prediction and often hurts performance. Compared with the Gemini 2 Flash single-call baseline of 60.4% Acc@1, self-reflection falls to 58.5%, while the sequential pipeline drops sharply to 26.4%. These degradations are more pronounced on Gemini 2; Gemini 3 Flash exhibits contrasting model-specific patterns, with sequential recovering to 49.9% Acc@1 but self-reflection falling even further to 46.7%. This behavior is consistent with the "Degeneration-of-Thought" (DoT) phenomenon Liang et al. (2024): once the model commits to an initial solution, iterative self-critique can reinforce that trajectory rather than uncover better alternatives (see Appendix B). The sequential pipeline also performs poorly at Acc@10 on Gemini 2, reaching only 46.7%, suggesting that its errors are not limited to ranking and often prevent the correct precursor set from appearing anywhere in the top 10. Debate matches the baseline at 60.4% Acc@1 on Gemini 2 and does not improve on it. Ensemble voting gives only marginal gains: 61.1% on Gemini 2 and 61.9% on Gemini 3, unlikely to justify the additional cost. These results contrast with prior work reporting gains from multi-agent debate on reasoning and factuality Du et al. (2023), but are consistent with analyses

Table 1: Results for all workflows on the full test set (n=674). Color indicates performance (low to high). Values in parentheses indicate absolute change relative to the single-call baseline for the same model. Best values within each model block are shown in **bold**. Across-model comparisons are qualitative only.

Method	@1	@3	@5	@10
Gemini 3 Flash				
Baseline	62.2	72.0	73.9	77.0
+ Tools	61.0 (-1.2)	71.4 (-0.6)	74.8 (+0.9)	80.6 (+3.6)
+ RAG	**64.7** (+2.5)	**77.8** (+5.8)	**80.9** (+7.0)	**83.5** (+6.5)
+ RAG + Tools	**64.7** (+2.5)	76.3 (+4.3)	80.0 (+6.1)	82.9 (+5.9)
Sequential	49.9 (-12.3)	59.2 (-12.8)	64.7 (-9.2)	73.3 (-3.7)
Self-Reflect	46.7 (-15.5)	60.2 (-11.8)	64.5 (-9.4)	69.4 (-7.6)
Ensemble Vote (3V)	61.9 (-0.3)	72.1 (+0.1)	74.2 (+0.3)	76.3 (-0.7)
Gemini 2 Flash				
Baseline	60.4	66.2	68.8	71.2
+ Tools	57.4 (-3.0)	59.9 (-6.3)	64.5 (-4.3)	69.4 (-1.8)
+ RAG	62.9 (+2.5)	71.5 (+5.3)	73.7 (+4.9)	77.0 (+5.8)
+ RAG + Tools	**63.1** (+2.7)	69.4 (+3.2)	73.2 (+4.4)	**77.5** (+6.3)
Sequential	26.4 (-34.0)	30.3 (-35.9)	33.4 (-35.4)	46.7 (-24.5)
Self-Reflect	58.5 (-1.9)	65.9 (-0.3)	69.7 (+0.9)	72.6 (+1.4)
Debate	60.4 (+0.0)	66.9 (+0.7)	69.6 (+0.8)	72.4 (+1.2)
Ensemble Vote (3V)	60.4 (+0.0)	**67.7** (+1.5)	**69.9** (+1.1)	71.7 (+0.5)
Ensemble Vote (5V)	61.1 (+0.7)	65.9 (-0.3)	69.2 (+0.4)	70.8 (-0.4)

showing that debate can reduce accuracy when models converge on flawed reasoning instead of challenging it Wynn et al. (2025).

3.2 Domain-specific information improves model performance

In contrast to test-time compute, adding domain-specific information improves performance. Retrieval over literature recipes gives the largest gains, raising top-1 accuracy from 62.2% to 64.7% and top-10 accuracy from 77.0% to 83.5%. Mean reciprocal rank also improves, and the gains are statistically significant (Table 3).

Thermodynamic tools are helpful, but their effect is smaller and less consistent. Used on their own, they do not improve top-1 accuracy over the baseline, which falls slightly from 62.2% to 61.0%, but they do improve top-10 accuracy from 77.0% to 80.6%. This suggests that the tools help widen the set of plausible candidates without consistently identifying the best route. Combining retrieval with all tools performs comparably to RAG alone on Gemini 3 Flash (64.7% top-1, 82.9% top-10), but does not improve on it: RAG alone scores higher at @3, @5, and @10. On the 200-sample subset (Figure 2, bottom; Table 4), the formation energy tool provides modest gains, whereas phase diagram entries, decomposition products, element balance, and reaction energy are neutral or slightly harmful.

These patterns suggest that precursor prediction is largely driven by precedent. Many precursor choices follow recurring literature conventions, while the underlying reaction mechanisms are often not accessible from the prompt alone. When the model has to extrapolate beyond known examples, it can produce plausible but weakly grounded explanations that do not improve precursor selection. Retrieval helps because it provides nearby successful recipes that can be adapted directly. By contrast, tools provide indirect constraints, such as energetic preferences, that rarely determine which commercially used precursor set should be chosen and still require additional assumptions to turn into a concrete synthesis decision.

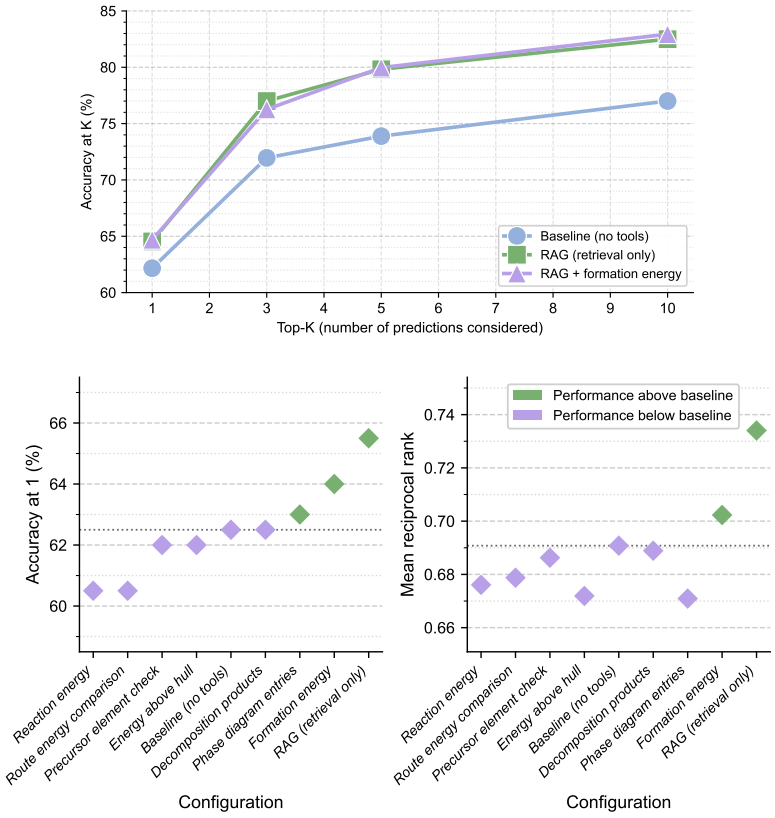

Figure 2: Top: Accuracy at top-1, 3, 5, and 10 for baseline (blue), retrieval-augmented generation (green), and RAG with formation energy tool (purple) on 674 test materials. Both retrieval-based methods reach about 83% at top-10. Bottom: Accuracy at 1 and mean reciprocal rank by individual tool configuration on 200 materials. Retrieval alone performs best; among tools, the formation energy tool provides the most benefit. Other tools are neutral or slightly harmful.

Appendix B illustrates three recurring ways in which retrieval improves correctness: resolving ambiguity between closely related precursor variants (for example, favoring MnO_2 over Mn_2O_3 for $PrMnO_3$); identifying the correct chemical family or anion chemistry; and recovering common dopant-source conventions such as the use of $AgNO_3$ in Ag-doped perovskites. These are discrete, convention-heavy choices that thermodynamic scores do not directly disambiguate, which helps explain why retrieval produces larger and more reliable gains than tools.

Our evaluation is limited to thermodynamics and literature precedent, kinetics are not captured by either the tools or the retrieval similarity measure. We treat literature-reported precursor sets as ground truth, while recognizing that other valid routes may exist. We evaluate only precursor sets: temperatures may be predicted, but they are not scored. Full prompts and tool specifications are provided in the Appendix, so the reported gains are not driven by undocumented implementation details.

4 Conclusion

We investigated augmentation strategies for LLM-based solid-state synthesis prediction and found that RAG is more effective than the test-time compute workflows we evaluated. On Gemini 3 Flash, RAG improves top-10 accuracy by 6.5 percentage points over the single-call baseline, from 77.0% to 83.5%, while combining RAG with tools yields 82.9%. In contrast, sequential pipelines, debate, and self-reflection do not consistently improve performance, and sequential workflows can substantially degrade accuracy. These results suggest that precursor prediction is driven more by access to relevant

synthesis precedents than by additional test-time compute or thermodynamic reasoning, and future work should prioritize improved retrieval and retrieval-based models over more complex agentic workflows. That said, our findings are tied to the model family and capability level evaluated. Stronger reasoning models may respond differently to multi-step workflows, and test-time compute methods could become more effective as base capabilities increase. We also evaluate only precursor sets, not temperatures or other synthesis parameters, and treat literature recipes as ground truth, which may miss valid alternatives.

REFERENCES

Amil Merchant, Simon Batzner, Samuel S Schoenholz, Muratahan Aykol, Gowoon Cheon, and Ekin Dogus Cubuk. Scaling deep learning for materials discovery. *Nature*, 624(7990):80–85, 2023.

Keith T Butler, Daniel W Davies, Hugh Cartwright, Olexandr Isayev, and Aron Walsh. Machine learning for molecular and materials science. *Nature*, 559(7715):547–555, 2018.

Thorben Prein, Elton Pan, Sami Haddouti, Marco Lorenz, Janik Jehkul, Tymoteusz Wilk, Cansu Moran, Menelaos Panagiotis Fotiadis, Artur P Toshev, Elsa Olivetti, et al. Retro-rank-in: A ranking-based approach for inorganic materials synthesis planning. *arXiv preprint arXiv:2502.04289*, 2025a.

Heewoong Noh, Namkyeong Lee, Gyoung S Na, and Chanyoung Park. Retrieval-retro: Retrieval-based inorganic retrosynthesis with expert knowledge. *Advances in Neural Information Processing Systems*, 37:25375–25400, 2024.

Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, et al. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL materials*, 1(1), 2013.

Edward Kim, Kevin Huang, Adam Saunders, Andrew McCallum, Gerbrand Ceder, and Elsa Olivetti. Materials synthesis insights from scientific literature via text extraction and machine learning. *Chemistry of Materials*, 29(21):9436–9444, 2017.

Olga Kononova, Haoyan Huo, Tanjin He, Ziqin Rong, Tiago Botari, Wenhao Sun, Vahe Tshitoyan, and Gerbrand Ceder. Text-mined dataset of inorganic materials synthesis recipes. *Scientific data*, 6 (1):203, 2019.

Tanjin He, Haoyan Huo, Christopher J Bartel, Zheren Wang, Kevin Cruse, and Gerbrand Ceder. Precursor recommendation for inorganic synthesis by machine learning materials similarity from scientific literature. *Science advances*, 9(23):eadg8180, 2023.

Santiago Miret and NM Anoop Krishnan. Enabling large language models for real-world materials discovery. *Nature Machine Intelligence*, 7(7):991–998, 2025.

Xue Jiang, Weiren Wang, Shaohan Tian, Hao Wang, Turab Lookman, and Yanjing Su. Applications of natural language processing and large language models in materials discovery. *npj Computational Materials*, 11(1):79, 2025.

Yiwen Hu and Markus J Buehler. Deep language models for interpretative and predictive materials science. *APL Machine Learning*, 1(1), 2023.

Thorben Prein, Elton Pan, Janik Jehkul, Steffen Weinmann, Elsa Olivetti, and Jennifer LM Rupp. Language models enable data-augmented synthesis planning for inorganic materials. *ACS Applied Materials & Interfaces*, 17(51):69221–69233, 2025b.

Zhilong Song, Shuaihua Lu, Minggang Ju, Qionghua Zhou, and Jinlan Wang. Accurate prediction of synthesizability and precursors of 3d crystal structures via large language models. *Nature Communications*, 16(1):6530, 2025.

Agada Joseph Oche and Arpan Biswas. Role of large language models and retrieval-augmented generation for accelerating crystalline material discovery: A systematic review. *arXiv preprint arXiv:2508.06691*, 2025.

Radeen Mostafa, Mirza Nihal Baig, Mashaekh Tausif Ehsan, and Jakir Hasan. G-rag: Knowledge expansion in material science. *arXiv preprint arXiv:2411.14592*, 2024.

Ziyi Luo, Jian Xu, Qingbo Yan, and Cheng-Lin Liu. A retrieval-augmented multimodal framework for scientific reasoning in materials science. In *Chinese Conference on Image and Graphics Technologies*, pages 289–303. Springer, 2025.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 17889–17904, 2024.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first International Conference on Machine Learning*, 2023.

Andrea Wynn, Harsh Satija, and Gillian Hadfield. Talk isn't always cheap: Understanding failure modes in multi-agent debate. *arXiv preprint arXiv:2509.05396*, 2025.

A PROMPTS AND TOOL SPECIFICATIONS

All workflows share the agent system prompts below; retrieval inserts top-k recipes into the chemist prompt. Workflow prompts use placeholders (`{target_formula}`, `{proposal}`, etc.) filled at runtime. When tools are enabled in single-call experiments, agents call the Materials Project functions listed in §A.3.

A.1 AGENT SYSTEM PROMPTS

Synthesis Chemist - System Prompt

You are an expert solid-state chemist specializing in ceramic and inorganic material synthesis. Given a target material formula, propose **20 different synthesis routes**, ranked by likelihood of success. Each route must have a **UNIQUE precursor set** (different combinations of starting materials).
For each route, specify:
1. **Precursors:** Starting materials (oxides, carbonates, nitrates, hydroxides, etc.)
2. **Calcination Temperature:** Temperature (°C) for initial calcination/decomposition
3. **Sintering Temperature:** Temperature (°C) for final sintering/densification
Consider various precursor options: Oxides (e.g., TiO_2, Fe_2O_3, Al_2O_3), Carbonates (e.g., $BaCO_3$, $CaCO_3$, Na_2CO_3), Nitrates (e.g., $AgNO_3$, $Ba(NO_3)_2$), Hydroxides (e.g., NaOH, $Ca(OH)_2$), and other salts.
Respond with JSON (exactly 20 routes):
{routes: [{rank, precursors, calcination_temp, sintering_temp, reasoning}, ...]}
IMPORTANT: All 20 precursor sets must be UNIQUE. Each route gets exactly ONE calcination and ONE sintering temperature. Rank 1 = most likely to succeed.

Thermodynamics Critic - System Prompt

You are a thermodynamics and materials processing expert reviewing synthesis route proposals. You will receive 20 proposed synthesis routes. For each route, evaluate:
1. **Precursor Validity:** Are these reasonable precursor choices?
2. **Decomposition Check:** Will precursors decompose at proposed calcination temperature?
   Carbonates: 400-900°C (depends on cation)
   Nitrates: 200-600°C
   Hydroxides: 300-500°C
3. **Sintering Feasibility:** Is the sintering temperature appropriate?
   Oxides: 1000-1500°C
   Perovskites: 1100-1400°C
   Spinels: 1200-1500°C
4. **Element Balance:** Do precursors contain all required elements?
Respond with JSON:
{evaluations: [{rank, valid, issues, corrected_calcination_temp, corrected_sintering_temp}, ...], reranking_suggestion: [...]}

Synthesis Integrator - System Prompt

You are the final decision-maker synthesizing input from a Chemist and a Critic.
Your task:
1. Review the 20 proposed routes and the critic's evaluations
2. Apply temperature corrections where suggested
3. Re-rank based on the critic's feedback
4. Ensure element balance for each route
5. Remove any invalid routes and ensure exactly 20 unique routes remain
You MUST respond with ONLY a JSON object (no markdown, no explanation).
RULES:
- Exactly 20 routes, ranked 1-20 (1 = best)
- All precursor sets must be UNIQUE
- precursors: alphabetically sorted chemical formulas
- calcination_temp: integer in °C
- sintering_temp: integer in °C

A.2 WORKFLOW PROMPTS

Placeholders are shown in `{braces}`. Each box represents one LLM call; dependencies indicated between boxes.

A.2.1 SINGLE-CALL BASELINE

User Prompt

```
You are an expert solid-state synthesis chemist.
Target material to synthesize: {target_formula}
[When RAG is enabled, retrieved examples are inserted here; see §A.4.]
Propose 20 different solid-state synthesis routes for this material, ranked by likelihood
of success. Each route must have a UNIQUE precursor set.
Consider: common precursor types (carbonates, oxides, nitrates, hydroxides, acetates),
appropriate calcination temperatures for precursor decomposition, appropriate sintering
temperatures for the target phase.
Respond with JSON only:
{routes: [{rank, precursors, calcination_temp, sintering_temp}, ...]}
```

A.2.2 SELF-REFLECTION

Three sequential calls: Generate → Critique → Refine (up to 2 cycles; early stop if critique finds no major issues).

Step 1 - Initial Generation

```
Target material to synthesize: {target_formula}
[Retrieved examples if RAG is enabled.]
Propose 20 different solid-state synthesis routes, ranked by likelihood of success. Each
route must have a UNIQUE precursor set.
Respond with JSON:
{routes: [{rank, precursors, calcination_temp, sintering_temp, reasoning}, ...]}
```

Step 2 - Self-Critique

```
You proposed these synthesis routes for {target_formula}:
{proposal}
Now critically review your OWN proposal:
1. Are any precursors chemically unreasonable?
2. Are temperatures appropriate for the decomposition/reactions?
3. Is element balance correct for each route?
4. Are there better alternatives you missed?
List specific issues and improvements. If no issues, say "No major issues found."
```

Step 3 - Refinement

```
Target: {target_formula}
Your previous proposal:
{proposal}
Your self-critique identified these issues:
{critique}
Now produce a REVISED set of 20 routes addressing these issues.
Respond with JSON only (no explanation): {routes: [...]}
```

A.2.3 DEBATE

Four sequential calls: Advocate → Skeptic → Advocate → Judge (2 rounds of exchange, then judgment).

Step 1 - Advocate Proposal

```
Target material to synthesize: {target_formula}
[Retrieved examples if RAG is enabled.]
You are the ADVOCATE. Propose 20 synthesis routes and be prepared to DEFEND them. A
Skeptic will try to find flaws. Make your proposal robust.
Respond with JSON:
{routes: [{rank, precursors, calcination_temp, sintering_temp, reasoning}, ...]}
```

Step 2 - Skeptic Attack

```
Target: {target_formula}
You are the SKEPTIC. Your job is to find FATAL FLAWS in the Advocate's proposal.
Current proposal:
{proposal}
Previous debate:
{debate_history}
Be adversarial! Look for:
- Wrong stoichiometry (precursors don't balance to target)
- Impossible temperatures (too low for reaction, too high for stability)
- Missing elements or wrong phases
- Unstable intermediates
- Better alternatives the Advocate missed
List your strongest attacks. Be specific and cite chemistry principles.
```

Step 3 - Advocate Defense

```
Target: {target_formula}
You are the ADVOCATE. The Skeptic has attacked your proposal:
{attack}
Previous debate:
{debate_history}
DEFEND your routes or REVISE them if the Skeptic has valid points. Do NOT simply agree,
push back where you're confident.
Respond with your updated (or defended) proposal as JSON: {routes: [...]}
```

Step 4 - Judge

```
Target: {target_formula}
You are the JUDGE. Review the full debate between Advocate and Skeptic:
{debate_history}
Your task:
1. Evaluate which routes survived the debate
2. Consider both Advocate's reasoning and Skeptic's valid criticisms
3. Produce a FINAL ranked list of 20 routes
Respond with JSON only:
{routes: [{rank, precursors, calcination_temp, sintering_temp}, ...]}
```

A.2.4 SEQUENTIAL PIPELINE

Three sequential calls: Chemist → Critic → Integrator. All three agents use the system prompts from §A.1. No tools are used in this workflow.

Step 1 - Chemist Proposal

```
Target material to synthesize: {target_formula}
```
[Retrieved examples if RAG is enabled.]
```
Please propose 20 solid-state synthesis routes for this material, ranked by likelihood of
success. Each route must have a UNIQUE precursor set.
Respond with JSON only:
{routes: [{rank, precursors, calcination_temp, sintering_temp}, ...]}
```

Step 2 - Critic Review

```
Target material: {target_formula}
The Synthesis Chemist has proposed the following routes:
{proposal}
Please review this proposal and provide your critique.
```

Step 3 - Integrator Synthesis

```
Target material: {target_formula}
=== CHEMIST'S PROPOSAL ===
{proposal}
=== CRITIC'S REVIEW ===
{critique}
Please synthesize these inputs and produce the final ranked list of 20 synthesis routes.
Respond with JSON only:
{routes: [{rank, precursors, calcination_temp, sintering_temp}, ...]}
```

A.3 TOOL SPECIFICATIONS

Table 2 lists the Materials Project tools available when tool access is enabled in single-call experiments (§2). All tools query the Materials Project API Jain et al. (2013). The sequential pipeline, self-reflection, and debate workflows do not use tools.

Table 2: Materials Project tool specifications. Tools are used in single-call experiments with tool access enabled.

Tool	Input	Output
`get_formation_energy`	`formula` (str)	Formation energy in eV / atom. Negative = stable.
`get_energy_above_hull`	`formula` (str)	Energy above convex hull in eV / atom. 0 = stable; >0 = metastable.
`get_phase_diagram_entries`	`elements` (str, e.g. `Ba-Ti-O`)	Stable and near-hull phases in the chemical system.
`get_decomposition_products`	`formula` (str)	Thermodynamically predicted decomposition products.
`check_precursor_elements`	`target`, `precursors` (both str)	Boolean: whether precursors contain all elements required by the target.

A.4 RETRIEVAL FORMAT

When RAG is enabled, each retrieved recipe is inserted into the prompt in the following format:

Retrieved Example (template)

```
Example {i} (similarity: {score})
Target: {retrieved_formula}
Precursors: [{precursor_1}, {precursor_2}, ...]
Calcination temperature: {calc_temp} °C
Sintering temperature: {sinter_temp} °C
```

We exclude only exact target-formula matches from the retrieval corpus. Compositionally similar materials (e.g., $Pr_{0.9}MnO_3$ when the target is $PrMnO_3$) may therefore appear as retrieved examples; see §2 for details on the similarity metric.

Table 3: Confirmatory ablation results with Holm-corrected McNemar tests (Gemini 3 Flash, $n = 674$). Values are from an independent confirmatory run and may differ slightly from Table 1 because of LLM non-determinism.

Comparison	Metric	Baseline	Variant	Δ	Holm p
RAG vs Baseline	Acc@1	62.2%	64.5%	+2.3%	0.023
	Acc@3	72.0%	77.0%	+5.0%	<0.001
	Acc@5	73.9%	79.8%	+5.9%	<0.001
	Acc@10	77.0%	82.5%	+5.5%	<0.001
RAG + form vs RAG	Acc@1	64.5%	64.7%	+0.2%	>0.500
	Acc@3	77.0%	76.3%	−0.7%	0.442
	Acc@5	79.8%	80.0%	+0.2%	>0.500
	Acc@10	82.5%	82.9%	+0.4%	>0.500

B ILLUSTRATIVE EXAMPLES

We present transcript excerpts showing how different workflows succeed or fail. All examples use real experiment logs from the test set.

B.1 SEQUENTIAL PIPELINE: CASCADING MIS-RANKING

The sequential workflow passes proposals through three agents: Chemist → Critic → Integrator. The example below shows how a correct proposal gets buried by downstream reranking.

Target: $LaYZr_2O_7$ **Ground truth:** La_2O_3 + Y_2O_3 + ZrO_2

Table 4: Tool ablation on 200-sample subset. The `compare_route_energies` and `calculate_reaction_energy` rows are exploratory; main experiments use the five tools listed in Appendix A. Abbreviations: *form* = `get_formation_energy`, *phase* = `get_phase_diagram_entries`. SC-n labels denote single-call configurations.

Configuration	Acc@1	Acc@3	Acc@5	Acc@10	Δ@1	Effect
Baseline (SC-0)	62.5%	74.0%	75.5%	80.5%	—	Reference
RAG (SC-1)	**65.5%**	**79.5%**	**83.5%**	**86.5%**	**+3.0%**	**Best**
get_formation_energy	64.0%	74.5%	76.5%	83.5%	+1.5%	Helpful
get_decomposition_products	62.5%	74.0%	76.5%	81.0%	+0.0%	Neutral
get_phase_diagram_entries	63.0%	69.0%	72.0%	76.5%	+0.5%	Neutral
check_precursor_elements	62.0%	73.0%	77.0%	79.5%	−0.5%	Neutral
get_energy_above_hull	62.0%	70.0%	73.0%	78.5%	−0.5%	Neutral
compare_route_energies	60.5%	73.0%	77.0%	80.5%	−2.0%	Harmful
calculate_reaction_energy	60.5%	72.5%	74.0%	79.5%	−2.0%	Harmful
RAG + form (SC-3)	65.0%	76.0%	81.0%	85.0%	+2.5%	≈ RAG
RAG + form + phase (SC-4)	65.5%	76.5%	80.0%	83.0%	+3.0%	≈ RAG

Effect label reflects the overall pattern across @1–@10, not @1 alone.

Table 5: Retrieval-Augmented Generation depth ablation: effect of number of retrieved neighbors (k). Results averaged across two random seeds (100 samples each, n=200 total). Gemini 3 Flash.

k value	Acc@1	Acc@3	Acc@5	Acc@10
$k = 1$	63.5%	76.5%	78.0%	80.5%
$k = 3$	61.0%	73.5%	77.5%	80.5%
$k = 5$ (default)	**65.5%**	**79.5%**	**83.5%**	**86.5%**
$k = 10$	60.0%	75.5%	78.0%	82.5%
$k = 15$	60.5%	73.0%	75.0%	79.0%

Table 6: Retrieval-Augmented Generation similarity metric ablation: effect of composition vs. Jaccard vs. random retrieval (k=5 fixed). Results averaged across two random seeds (100 samples each, n=200 total). Gemini 3 Flash.

Similarity Metric	Acc@1	Acc@3	Acc@5	Acc@10
Compositional (default)	**65.5%**	79.5%	**83.5%**	**86.5%**
Jaccard	65.0%	**80.0%**	82.5%	86.0%
Random retrieval	62.5%	76.0%	79.5%	81.5%

Synthesis Chemist

Rank 1: $La_2O_3 + Y_2O_3 + ZrO_2$ ✓ Correct

"Standard solid-state reaction using stable oxide precursors. High temperatures needed for diffusion."

↓

Thermodynamics Critic

All 10 routes validated as chemically viable (`valid: true`).

Reranking suggestion: `[6, 8, 9, 2, 3, 4, 5, 1, 7, 10]`

Moves correct oxide route from position 1 to position 8.

↓

Synthesis Integrator

Final Rank 1: $La(NO_3)_3 + Y(NO_3)_3 + ZrO_2$ × Wrong
Final Rank 8: $La_2O_3 + Y_2O_3 + ZrO_2$ ✓ Correct (buried)

Result: Correct answer degrades from rank 1 to rank 8. Baseline, self-reflection, and debate all preserve rank 1 for this material.

B.2 Self-Reflection: Over-Correction

Self-reflection degrades 74 materials that baseline predicts correctly. The model's self-critique promotes "sophisticated" wet-chemistry routes over correct simple routes.

Target: $SmFeO_3$ **Ground truth:** $Fe_2O_3 + Sm_2O_3$

Baseline ✓
Rank 1: $Fe_2O_3 + Sm_2O_3$
"Standard solid-state reaction."

Self-Critique (verbatim excerpt)

"Reviewing my proposed synthesis routes for SmFeO3...
Missed Alternatives and Improvements:
• **Pechini Method:** *I missed a route based on the Pechini method using citric acid and ethylene glycol. This would provide excellent mixing at the atomic level.*
• **Citrate Gel Method:** *Similar to Pechini, but using only citric acid to form a gel.*
Specific Improvements:
• **Re-rank based on Pechini/Citrate Gel:** *These methods generally offer better homogeneity and lower sintering temperatures, so they should be ranked higher than some of the simple oxide mixing routes."*

Self-Reflection (Final Output) ×
Final Rank 1: $Sm(NO_3)_3 + Fe(NO_3)_3$ + Citric Acid + Ethylene Glycol
Final Rank 7: $Fe_2O_3 + Sm_2O_3$ (correct, buried)

Pattern: The model "improves" by promoting the Pechini sol-gel method, which is chemically reasonable but not matching the literature ground truth. This illustrates Degeneration-of-Thought (Liang et al., 2024): once the model establishes confidence in its critique, it fails to generate alternatives that would preserve the correct answer.

B.3 Debate: Skeptic-Induced Substitution

Debate degrades 55 materials. The Skeptic challenges precursor choices, and the Advocate substitutes with alternatives that don't match ground truth.

Target: $Ba_2Ca_0 \cdot 8\,Lu_0 \cdot 8\,Cu_3O_7$ **Ground truth:** $BaCO_3 + CaCO_3 + Lu_2O_3 + CuO$

Baseline ✓
Rank 1: $BaCO_3 + CaCO_3 + Lu_2O_3 + CuO$

Debate Round 2: Skeptic Challenge

"CaCO3 requires higher decomposition temperatures than CaO. Using CaO directly would reduce processing time and energy consumption. Consider substituting CaCO3 → CaO for improved efficiency."

Debate (Final Output) ×

Final Rank 1: $BaCO_3$ + **CaO** + Lu_2O_3 + CuO

Single precursor substitution: $CaCO_3 \rightarrow CaO$

Pattern: The Skeptic's efficiency argument is chemically valid: CaO does react more readily than $CaCO_3$. However, the literature ground truth uses $CaCO_3$, not CaO. The debate format encourages such "improvements" that deviate from actual practice.

B.4 RAG: Resolving Precursor Ambiguity

RAG improves 26 materials from baseline failure to hit@1. The key mechanism is the following: retrieved recipes resolve ambiguity about which precursor variant to use. Only exact formula matches are excluded, so similar compositions (e.g. $Pr_{0.9}MnO_3$) can have similarity 1.00.

Target: $PrMnO_3$ **Ground truth:** MnO_2 + Pr_6O_{11}

Baseline ×

Rank 1: Mn_2O_3 + Pr_6O_{11} (wrong Mn precursor)

Rank 2: MnO_2 + Pr_6O_{11} (correct)

Retrieved Context

Example 1 (similarity: 1.00):
Target: $Pr_{0.9}MnO_3$ - Precursors: [**MnO_2**, Pr_6O_{11}], Sintering: 1110°C

Example 2 (similarity: 1.00):
Target: $Ca_{0.2}Pr_{0.8}MnO_3$ - Precursors: [**MnO_2**, Pr_2O_3, $CaCO_3$], Sintering: 1300°C

RAG ✓

Rank 1: MnO_2 + Pr_6O_{11}

"Most standard solid-state route using stable, commercially available oxides."

Mechanism: Multiple Mn oxides are chemically plausible (MnO_2, Mn_2O_3, Mn_3O_4). Retrieved recipes from similar Pr manganites consistently use MnO_2, resolving the ambiguity.

B.5 RAG: Material Class Identification

Some failures occur because the model misidentifies the material class entirely.

Target: $Na_{2.32}Co_{1.84}S_3O_{12}$ **Ground truth:** $CoSO_4$ + Na_2SO_4

Baseline ×

Rank 1: Co_3O_4 + Na_2CO_3 (oxide precursors)

Model treats this as an oxide, missing that the formula contains sulfur.

RAG ✓

Rank 1: $CoSO_4$ + Na_2SO_4 (sulfate precursors)

Retrieved examples correctly identify this as a sulfate compound.

Mechanism: Baseline fails to recognize this is a sulfate. RAG retrieves similar sulfate syntheses showing sulfate precursors are needed.

B.6 TOOLS: FORMATION ENERGY VERIFICATION

When tools help, they typically verify precursor stability through `get_formation_energy` queries.

Target: $Na_3Bi(AsO_4)_2$ **Ground truth:** $NH_4H_2AsO_4 + Bi_2O_3 + Na_2CO_3$

Baseline ×
Rank 1: $As_2O_5 + Bi_2O_3 + Na_2CO_3$
Rank 2: $NH_4H_2AsO_4 + Bi_2O_3 + Na_2CO_3$ (correct)

Tool Calls

```
get_formation_energy('Na3Bi(AsO4)2') → {found: false}
get_formation_energy('Na2CO3')       → {-2.07 eV/atom}
get_formation_energy('Bi2O3')        → {-1.64 eV/atom}
get_formation_energy('As2O5')        → {-1.56 eV/atom}
```

Tools (`get_formation_energy`) ✓
Rank 1: $NH_4H_2AsO_4 + Bi_2O_3 + Na_2CO_3$
"Ammonium arsenate decomposes to highly reactive As_2O_5 in situ."

Mechanism: After confirming precursor stability, the model reasons that $NH_4H_2AsO_4$ provides more reactive arsenic than direct As_2O_5 oxide via in-situ decomposition.

B.7 TOOLS: FAILURE MODE (OVER-COMPLICATION)

Tools can hurt when they lead the model to over-complicate simple synthesis routes.

Target: $GaAg_9Se_{5.99}I_{0.01}$ **Ground truth:** Ag + AgI + Ga + Se

Baseline ✓
Rank 1: Ag + AgI + Ga + Se
"Direct elemental synthesis with AgI as dopant source."

Tool Calls

```
get_decomposition_products('AgNO3')    → {found: true}
get_decomposition_products('Ga(NO3)3') → {found: false}
```

Tools (`get_decomposition_products`) ×
Rank 1: $Ag_2Se + AgI + Ga_2Se_3$ (binary selenides)
Rank 2: Ag + AgI + Ga + Se (correct, demoted)

Mechanism: The decomposition tool returned NULL for $Ga(NO_3)_3$, making the model uncertain about salt routes. It favored binary selenides over elements, but the ground truth requires elemental precursors for stoichiometric control.

B.8 SUMMARY OF FAILURE AND SUCCESS PATTERNS

Table 7: Systematic patterns across transcript analysis for the Gemini 2 runs. ↓ indicates degradation (baseline correct → method wrong); ↑ indicates improvement (baseline wrong → method correct).

Workflow	Count	Dominant Pattern
Sequential	255 ↓	Critic reranks correct routes lower; Integrator adopts bad rankings without verification
Self-Reflect	74 ↓	Self-critique promotes wet chemistry (Pechini, sol-gel) over correct oxide routes
Debate	55 ↓	Single-precursor substitutions from Skeptic challenges (e.g., $CaCO_3$→CaO)
RAG	26 ↑	Resolves precursor variant ambiguity; identifies material classes

B.9 ESTIMATED INFERENCE COST PER WORKFLOW

Table 8: Estimated inference cost per workflow (n=674). Token counts are approximate per-sample averages. Multi-step workflows accumulate tokens as each step receives prior outputs as context. Prices used: Gemini 2 Flash \$0.10 / \$0.40 and Gemini 3 Flash \$0.50 / \$3.00 per 1M input / output tokens (via OpenRouter).

	LLM Calls	Input Tokens	Output Tokens	Total Cost (\$, n=674)		Relative
Method	per sample	per sample	per sample	**G2 Flash**	**G3 Flash**	**to Baseline**
Baseline	1	0.8k	2.5k	\$0.73	\$5.32	1.0×
+ RAG	1	2.0k	2.5k	\$0.81	\$5.73	1.1×
+ Tools	1	1.2k	3.0k	\$0.89	\$6.47	1.2×
+ RAG + Tools	1	2.4k	3.0k	\$0.97	\$6.87	1.3×
Self-Reflect	1–3*	8k	5.5k	\$2.02	\$13.82	2.6×
Sequential	3	9k	6.5k	\$2.36	\$16.18	3.0×
Ensemble (3V)	3	2.4k	7.5k	\$2.18	\$15.97	3.0×
Debate	4–8†	15–30k	8.5–17k	\$3.3–6.6	\$22–44	4.2–8.4×

*Self-Reflect generates once, then iterates critique–refine cycles up to 2 times. Early stops if no major issues.
†Debate scales with number of rounds (1–3): 1 proposal + (n attacks + n defenses) + 1 judge.