From Literature to Experimental Conditions: Large Language Models for Co-Crystal Synthesis Design

Anonymous Authors[1]

Abstract

Co-crystallization enables modulation of drug physicochemical properties, yet practical synthesis design remains largely empirical. Existing AI methods mainly address molecular pair selection, leaving synthesis-condition design underexplored because the required experimental knowledge is scattered across scientific articles. We present CoSyn, a framework that uses large language models to convert dispersed procedure descriptions into machine-readable synthesis records, enabling experimental condition recommendations for new molecular pairs and tool-augmented assistance. Using manually annotated datasets, we evaluate both individual components and system-level performance. Our results highlight the potential of a modular LLM-based approach to organize literature-derived evidence and support co-crystal synthesis design beyond pair-level screening.

1. Introduction

Pharmaceutical co-crystals offer a solid-form route for modulating drug properties, including solubility, stability, pharmacokinetic behavior, and mechanical properties (Yadav et al., 2025; Marcos Valdez et al., 2026). They combine an active pharmaceutical ingredient (API) with an additional pharmaceutically acceptable molecule (coformer) in a single crystalline phase, stabilized by non-covalent interactions (Bolla et al., 2022). Despite their broad potential, co-crystal design remains a challenging experimental task that often relies on empirical screening across heterogeneous synthesis conditions (Karimi-Jafari et al., 2018b; Zhang et al., 2025a).

Artificial intelligence has advanced the early stages of co-crystal design, including machine-learning models for coformer screening and co-crystal formability prediction (Wicker et al., 2017; Mswahili et al., 2021; Devogelaer et al., 2019; 2020; Jiang et al., 2021), as well as recent deep learning, thermodynamics-informed, and generative systems (Guo et al., 2024; Birolo et al., 2025; Song et al., 2025b; Gubina et al., 2024). However, these methods primarily operate at the level of molecular-pair selection, rather than practical synthesis planning, whereas synthesis-condition design remains comparatively underexplored. A key reason is that synthesis planning depends on method-specific experimental precedents that are scattered across scientific articles as heterogeneous synthesis descriptions rather than collected in standardized datasets.

Automated extraction of chemical information from literature is a long-standing goal of chemical informatics, from general-purpose systems such as OSCAR4 and ChemDataExtractor (Jessop et al., 2011; Swain & Cole, 2016) to recent extraction frameworks (Fan et al., 2024; Dagdelen et al., 2024) and domain-specific pipelines (Vaucher et al., 2021; Shi et al., 2024). However, recent reviews and benchmarks show that reliable scientific extraction depends on domain-specific schemas, normalization, and validation rather than generic text-mining alone (Schilling-Wilhelmi et al., 2025; Vepreva et al., 2025; Chong & Colindres, 2026). This is particularly important for co-crystal synthesis, where different preparation methods require different experimental fields and constraints.

Large language models (LLMs) offer a flexible mechanism for connecting unstructured scientific text with downstream scientific decision-making (Zhang et al., 2025b). In chemistry and materials science, they have been used for information extraction (Schilling-Wilhelmi et al., 2025), molecular and materials design (Ramos et al., 2025), predictive modeling (Jablonka et al., 2024), and tool-augmented reasoning over scientific workflows (Bran et al., 2024; Ruan et al., 2024; Song et al., 2025a). These capabilities are well aligned with co-crystal synthesis design, where useful AI support requires both recovering experimental knowledge from articles and reusing that knowledge to recommend conditions for new molecular pairs. Yet the use of LLMs for method-aware co-crystal synthesis extraction, condition prediction, and tool-augmented assistance remains largely unexplored.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country.**AUTHORERR: Missing \icmlcorrespondingauthor.**

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

In this work, we present CoSyn, a modular LLM-based framework for co-crystal synthesis design from literature-derived experimental evidence. CoSyn extracts method-aware synthesis records from articles, uses them to recommend conditions for new molecular pairs, and composes these capabilities within an agentic assistant. Our contributions are as follows:

- We develop a hierarchical LLM pipeline for detecting co-crystal synthesis procedures, classifying synthesis methods, and extracting method-specific experimental fields from full-text articles.
- We construct CoSyn-DB, a machine-readable database of co-crystal synthesis records containing 20,783 method-specific entries.
- We evaluate machine-learning baselines and few-shot LLM prediction of method-specific synthesis conditions, extending co-crystal AI beyond pair-level screening.
- We integrate these components into an agentic assistant and evaluate system-level behavior on CoSyn-Bench, a domain-specific benchmark of expert-curated questions.

Code and data used in this study are available at https://anonymous.4open.science/r/CoSyn/

2. Related Work

AI for Co-Crystal and Synthesis-Condition Design Pharmaceutical co-crystal development remains highly empirical, depending on coformer choice, preparation route, and process parameters that differ across solution-based and solid-state methods (Karimi-Jafari et al., 2018a; Bolla et al., 2022). AI methods for co-crystal design have therefore focused mainly on upstream molecular-pair selection, including descriptor-based classifiers trained on experimental data (Wicker et al., 2017; Wang et al., 2020), network-based link prediction (Devogelaer et al., 2019), graph neural networks such as CCGNet (Jiang et al., 2021), deployable platforms such as CCPT (Guo et al., 2024), and language-model-based systems such as DeepCocrystal (Birolo et al., 2025). Recent work extends this line to coformer–solvent screening (Song et al., 2025b; Zeng et al., 2025) and property-driven generative design (Gubina et al., 2024). Overall, co-crystal AI has advanced molecular-pair prioritization, but has not yet addressed synthesis design at the level of routes and method-specific experimental conditions.

Synthesis-condition design is a heterogeneous structured-output problem over categorical and numerical experimental variables (Ball et al., 2025). Prior work has addressed this structure with hierarchical networks (Gao et al., 2018), two-stage prediction and ranking (Chen & Li, 2024), and template- or cluster-based condition representations (Wang et al., 2025). LLMs provide a flexible alternative by generating correlated structured outputs, as shown for low-data reaction-condition prediction (Yang et al., 2025), multimodal recommendation (Zhang et al., 2024), and inorganic synthesis planning (Prein et al., 2025). CoSyn brings this idea to method-aware co-crystal synthesis, where condition fields vary by preparation route and must be grounded in literature-derived precedents.

Automated Extraction of Scientific Data Chemical information extraction from literature has progressed from foundational text-mining systems such as OSCAR4, ChemicalTagger, and ChemDataExtractor (Jessop et al., 2011; Hawizy et al., 2011; Swain & Cole, 2016) to document-level and LLM-based extraction across text, tables, and figures (Dagdelen et al., 2024; Fan et al., 2024). Domain-specific systems have further targeted synthesis procedures, reaction data, MOF chemistry, and nanomaterials (Vaucher et al., 2021; Kearnes et al., 2021; Ai et al., 2024; Zheng et al., 2023; Odobesku et al., 2025). Recent reviews and benchmarks show that robust extraction in specialized scientific domains depends not only on stronger models, but also on domain-specific schemas, normalization, provenance, and validation workflows (Schilling-Wilhelmi et al., 2025; Vepreva et al., 2025; Chong & Colindres, 2026). We therefore formulate co-crystal synthesis extraction as a method-aware record construction task rather than a generic text-mining problem.

Agentic Systems and Evaluation LLMs are increasingly used as agents for scientific workflows, coordinating retrieval, tool use, experiment planning, and result interpretation. Chemistry-oriented systems range from tool-augmented agents such as Coscientist and ChemCrow (Boiko et al., 2023; Bran et al., 2024) to role-specialized multi-agent architectures such as ChemAgents and LLM-RDF (Song et al., 2025a; Ruan et al., 2024). Evaluating such systems is difficult because scientific tasks are open-ended, allow multiple valid solution paths, and depend strongly on which tools are available (Gu et al., 2024; Yu et al., 2025). Existing benchmarks typically target either general scientific workflows (Chen et al., 2025) or broad chemistry task collections (Mirza et al., 2025), but do not capture the coupled retrieval, extraction, and prediction demands of co-crystal synthesis design. We therefore pair our agentic architecture with a targeted benchmark reflecting these domain-specific requirements.

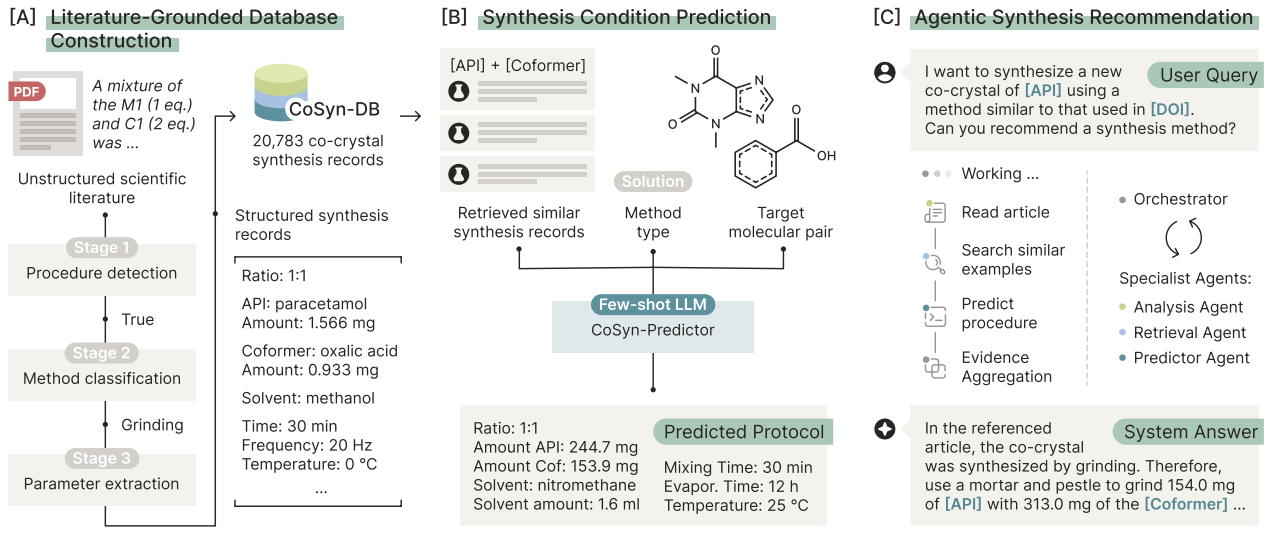

Figure 1. Overview of CoSyn. [A] Automated extraction builds CoSyn-DB from co-crystal articles via procedure detection, method classification, and parameter extraction. [B] Condition prediction retrieves similar API–coformer records and uses few-shot LLM prompting to generate method-specific protocols. [C] Agentic recommendation combines article analysis, database retrieval, and prediction for grounded synthesis guidance.

3. Dataset Construction

3.1. Source Corpus

We built the source corpus from the Cambridge Structural Database (CSD), a curated repository of experimentally determined co-crystal structures (Groom et al., 2016). Using publication DOIs linked to relevant CSD entries, we obtained 4,287 unique source DOIs after deduplication (Figure 2).

Automated full-text retrieval produced 3,997 PDFs (not redistributed), which were converted to machine-readable text for the extraction pipeline. Retrieval failures were due to access restrictions, missing full-text links, publisher-side failures, or conversion errors. Corpus statistics, including open-access status and publication years, are reported in Appendix B.1.

3.2. Method Taxonomy

Co-crystal synthesis methods vary widely in experimental structure and reported parameters, making a single schema unsuitable. We therefore defined a method-aware taxonomy of 12 synthesis types, covering solution-based routes such as solution, slurry, and antisolvent crystallization, solid-state routes such as grinding and hot-melt extrusion, and specialized methods (Karagianni et al., 2018; Buddhadev & Garala, 2020; Douroumis et al., 2017; Pawar et al., 2021). For each method, we prepared (i) a natural-language description of key physicochemical steps and (ii) a representative procedure template with marked parameter slots.

The taxonomy is operational rather than exhaustive: rare routes may be absent, and procedures with similar logic and parameter spaces are merged, with variation captured through optional fields. The full taxonomy, parameters, and templates are provided in Appendix B.2.

3.3. Hierarchical Extraction Pipeline

A one-step extraction procedure is unsuitable because (i) not every article contains an explicit synthesis procedure, which may be omitted or cited from earlier work, and (ii) the parameter schema depends on the method, which is unknown in advance. A single prompt for detection, method inference, and parameter extraction would force the model to consider many irrelevant fields, increasing hallucination risk (Ji et al., 2023). We therefore use a three-stage hierarchical task.

Stage 1: Procedure detection. Given the full article text, the model determines whether an explicit synthesis procedure is present.

Stage 2: Method classification. The model assigns one or more labels from the 12 predefined method types (Section B.2), so downstream prompts use only the relevant schema.

Stage 3: Parameter extraction. For each detected method, a method-specific prompt extracts experimental parame-

Table 1. Stage 1 prompting strategy comparison on GPT-4o-mini, evaluated on a pilot subset of 12 positive and 8 negative articles.

Strategy	Precision	Recall	F1	Accuracy
S1-A	0.62 ±0.11	1.00 ±0.00	0.76 ±0.08	0.62 ±0.11
S1-B	1.00 ±0.00	0.92 ±0.08	0.96 ±0.04	0.95 ±0.05
S1-C	0.72 ±0.11	1.00 ±0.00	0.84 ±0.07	0.76 ±0.09

Table 2. Stage 2 prompting strategy comparison on GPT-4o-mini, evaluated on the full Stage 2 validation set.

Strategy	Precision	Recall	F1	Accuracy
S2-A	0.69 ±0.06	0.53 ±0.05	0.58 ±0.05	0.53 ±0.05
S2-B	0.70 ±0.05	0.56 ±0.05	0.59 ±0.05	0.56 ±0.05
S2-C	0.75 ±0.05	0.61 ±0.05	0.64 ±0.05	0.61 ±0.05

ters into structured output. Prompts were developed with domain experts and specify field names and examples for consistency.

All prompts are provided in Appendix B.4, with the grinding prompt shown as a representative Stage 3 example.

Validation datasets. We constructed three hand-annotated validation datasets, one per stage: (i) 100 articles for Stage 1 (92 positive, 8 negative), (ii) 109 articles for Stage 2 covering all 12 method types (54 single-method, 55 multi-method), and (iii) a LAG-specific Stage 3 set of 55 articles with 295 records over 25 parameters. The same datasets are used for prompting strategy comparison in Section 3.4 and cross-model evaluation in Section 3.5. Metric definitions are given in Appendix B.3.

3.4. Prompting Strategy Optimisation

Setup. We used `GPT-4o-mini` to develop prompts for all stages, using the validation datasets from Section 3.3. For Stages 1 and 2, we compared three predefined strategies. For Stage 3, we refined one prompt through expert-in-the-loop iteration on the LAG validation set, a representative high-support method with a rich schema.

Strategies. The three Stage 1 strategies differ in the required **response format**:

- **S1-A:** binary classification of synthesis presence.
- **S1-B:** full text span of the synthesis method.
- **S1-C:** same as S1-B, separated per molecular pair and technique.

The three Stage 2 strategies differ in the **method-type context** provided alongside the procedure:

- **S2-A:** predefined list of method names.
- **S2-B:** method names with natural-language descriptions from Table 6.
- **S2-C:** method names with procedural templates from Appendix B.2.2.

Results. Table 1 shows that S1-A and S1-C achieved perfect recall but lower precision, indicating over-detection in the binary formulation and spurious extractions from pair-and-technique decomposition. S1-B gave the best F1 and accuracy and was selected for Stage 1.

The Stage 2 results in Table 2 show a consistent gradient: names alone performed worst, descriptions improved modestly, and templates achieved the best scores. Templates likely help because related methods are easier to distinguish by procedural surface form than by abstract definitions. We therefore used S2-C for Stage 2 in the final pipeline. Per-method accuracy and F1 are reported in Appendix B.6.

3.5. Evaluation of Extraction Pipeline

Setup. Using the selected strategies (S1-B and S2-C), we evaluated four LLMs on the validation datasets from Section 3.3: `GPT-4o-mini`, `Gemma-3-27B`, `Llama-3.3-70B`, and `Gemini-2.5-Flash`. Metric definitions are given in Appendix B.3.

Results. Table 3 reports performance across the three stages. `Gemini-2.5-Flash` achieved the highest F1 and accuracy at every stage and was selected for full-corpus extraction. Stage 1 mainly separated models by recall: `GPT-4o-mini` and `Llama-3.3-70B` reached perfect precision but missed many valid procedures, whereas `Gemini-2.5-Flash` preserved both precision and recall (F1 = 0.97). Stage 2 was most model-sensitive, with F1 from 0.19 to 0.81. Stage 3 scores were tighter (overall F1 0.54–0.63) and conservative because many mismatches arise from the strict multiset metric. Per-parameter results are reported in Appendix B.7. Although strategy selection on `GPT-4o-mini` may slightly favor that model, `Gemini-2.5-Flash` remains consistently superior.

Error analysis. Manual annotation of `Gemini-2.5-Flash` outputs identified dominant failure modes at each stage (Table 8). Stage 1 and Stage 2 were dominated by omissions caused by non-standard terminology and methods reported only in supplementary or comparative experiments. Stage 3 was dominated by artefacts from strict multiset comparison. The main genuine Stage 3 error was fabrication of details apparently lifted from nearby text but assigned to the wrong record. A full breakdown is provided in Appendix B.8.

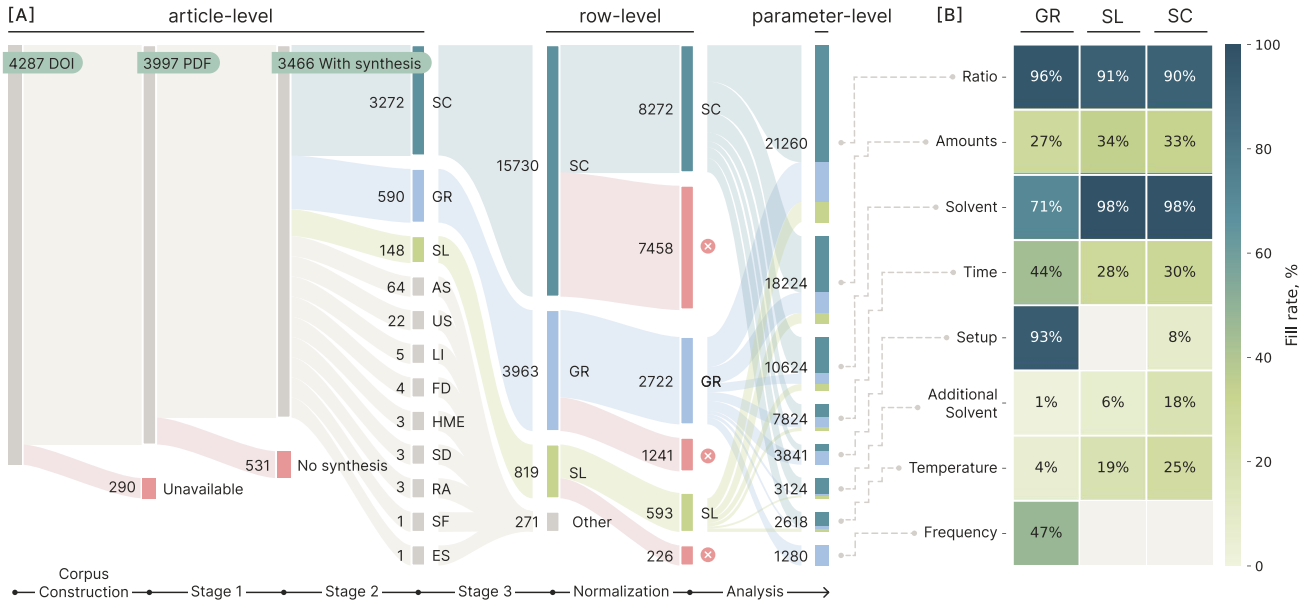

Figure 2. CoSyn-DB construction pipeline and parameter coverage. Method abbreviations: AS, antisolvent; ES, electrospray; FD, freeze drying; GR, grinding; HME, hot-melt extrusion; LI, laser irradiation; RA, resonant acoustic; SL, slurry; SC, solution crystallization; SD, spray drying; SF, supercritical fluid; US, ultrasound. [A] Sankey diagram showing the flow from 4,287 source DOIs to normalised synthesis records and parameter families. At the parameter level, the diagram reports filled values within each family (Appendix B.9). [B] Parameter family fill rates for the three downstream datasets after quality filtering, where fill rate is the proportion of non-empty values per family.

3.6. CoSyn-DB Dataset

Database extraction. Applying the pipeline to the source corpus identified 3,466 articles (86.7%) with at least one co-crystal synthesis procedure and yielded 20,783 unique synthesis records. Three methods account for 98.7% of all entries: solution crystallization (15,730 records), grinding (3,963), and slurry (819). We retained these for downstream experiments and treated the remaining nine categories as too sparse for reliable modeling. The full data flow is shown in Figure 2A.

Post-processing. Extracted records were converted into modeling-ready datasets by normalising method-specific fields, handling missing values, and applying leakage-safe splits. Normalization resolved API–coformer pairs, harmonised categorical fields, converted values to common units, and removed invalid, duplicate, or low-information entries, retaining 11,587 records across the three downstream methods. Parameter coverage remained uneven: ratios and solvents were frequent, while amounts, times, and temperatures were sparse (Figure 2B). Missing values were imputed only when expected but absent from text; structurally absent values were left empty. Pre-imputation test snapshots were retained so metrics in Section 4 use only observed values. Records were split by connected components over shared record identifiers and API–coformer pairs, yielding a balanced held-out test set of 300 examples. Details are in Appendix B.10.

4. Synthesis Condition Prediction

We next use CoSyn-DB to test whether literature-derived synthesis records support condition recommendation for new co-crystal pairs. Given an API–coformer pair and target method, the task is to predict coherent method-specific categorical and numerical parameters, such as solvent, apparatus, time, temperature, and reagent amounts.

4.1. Experimental Setup

Prediction models. Classical baselines use concatenated API–coformer Morgan fingerprints as input (computed with RDKit (Landrum), BSD-3-Clause). We evaluate a frequency prior, a multi-output Random Forest classifier (RF, BSD-3-Clause), and continuous-output regressors for numerical fields: independent Random Forest regressors (RF-Regression), regressor chains (RF-Chain), and CatBoost regression (CB-Regression (Prokhorenkova et al., 2018), Apache-2.0). We compare them with four few-shot LLM predictors: `GPT-4o-mini`, `GPT-5.4`, `Gemini-2.5-Flash`, and `Gemini-3.1-Pro`. For

Table 3. Performance of four LLMs across the three extraction stages. † marks the model selected for full-corpus extraction.

Model	Precision	Recall	F1	Accuracy
Stage 1				
GPT-4o-mini	1.00 ±0.00	0.78 ±0.04	0.88 ±0.03	0.80 ±0.04
Gemma-3-27B	0.91 ±0.03	0.92 ±0.03	0.92 ±0.02	0.85 ±0.04
Llama-3.3-70B	1.00 ±0.00	0.57 ±0.05	0.73 ±0.04	0.61 ±0.05
Gemini-2.5-Flash†	0.98 ±0.02	0.97 ±0.02	0.97 ±0.01	0.95 ±0.02
Stage 2				
GPT-4o-mini	0.75 ±0.05	0.61 ±0.05	0.64 ±0.05	0.61 ±0.05
Gemma-3-27B	0.58 ±0.08	0.47 ±0.05	0.44 ±0.05	0.47 ±0.05
Llama-3.3-70B	0.41 ±0.08	0.14 ±0.03	0.19 ±0.04	0.14 ±0.03
Gemini-2.5-Flash†	0.86 ±0.04	0.82 ±0.04	0.81 ±0.04	0.82 ±0.04
Stage 3				
GPT-4o-mini	0.61 ±0.03	0.63 ±0.03	0.54 ±0.03	0.45 ±0.03
Gemma-3-27B	0.59 ±0.04	0.70 ±0.03	0.56 ±0.03	0.47 ±0.04
Llama-3.3-70B	0.62 ±0.03	0.73 ±0.03	0.61 ±0.03	0.53 ±0.04
Gemini-2.5-Flash†	0.65 ±0.04	0.74 ±0.03	0.63 ±0.03	0.56 ±0.04

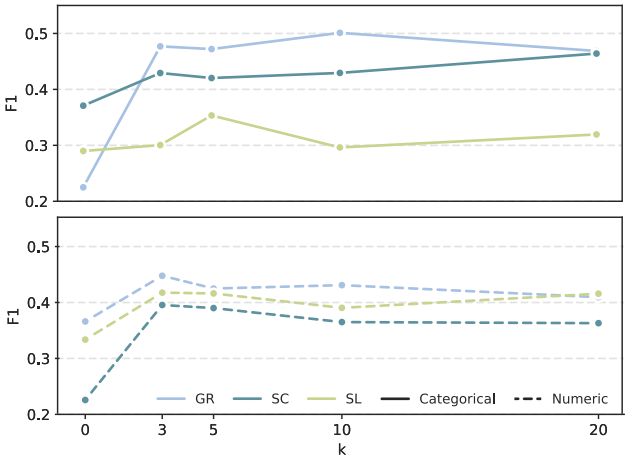

Figure 3. Effect of the number of in-context examples k on weighted F1, evaluated on `GPT-4o-mini` across the three methods: GR, grinding; SL, slurry; SC, solution crystallization.

each test pair, the k most similar training pairs are retrieved by molecular fingerprint similarity and used as in-context examples. We use k=3, selected by the ablation in Figure 3 and detailed in Appendix C.3. Retrieval and prompts are described in Appendices C.1 and C.2.

Evaluation. For the main comparison, we focus on grinding because it is compact but method-rich, with optional solvents, apparatus choices, and numerical process parameters. We evaluate synthesis-parameter fields, exclude auxiliary unit columns, and retain parameters observed in at least 20% of held-out records. Categorical fields use exact-match accuracy and weighted F1 after normalization, except Solvent, which is scored by solvent family due to high cardinality (Appendix C.4). Numerical fields use tolerance bins (Appendix C.5). We report weighted Categorical, Numerical, and Overall averages, with Solvent and Ratio shown separately.

4.2. Prediction Results

Table 4 reports the main comparison for synthesis-condition prediction. The frequency prior performs strongly on Ratio, reflecting common equimolar stoichiometries, but weakly on chemistry-dependent fields such as Solvent. All LLMs improve Solvent F1 from 0.17 to 0.36–0.44, showing the value of pair-specific retrieved precedents (Appendix C.6). Fingerprint-based ML baselines improve categorical prediction but underperform LLMs on numerical fields, suggesting that retrieved literature examples provide a stronger procedural prior than fingerprints alone.

Among LLMs, the Gemini models perform best overall. `Gemini-2.5-Flash` and `Gemini-3.1-Pro` tie for best Overall F1, while `Gemini-3.1-Pro` leads on Solvent. The GPT models trail, with `GPT-5.4` close on aggregate metrics and `GPT-4o-mini` weakest. Since `Gemini-2.5-Flash` matches `Gemini-3.1-Pro` on aggregate metrics at much lower inference cost, we adopt it as the Predictor agent in Section 5 and report per-parameter results in Appendix C.7.

5. Agentic Co-Crystal Design

The components above, article extraction, CoSyn-DB retrieval, and few-shot prediction, solve complementary sub-problems, whereas practical co-crystal synthesis design often requires their joint use. We therefore expose all CoSyn components through an agentic interface and evaluate whether this improves synthesis assistance over non-agentic LLM baselines.

5.1. Agentic System Design

We compare two non-agentic baselines with two agentic architectures. The LLM-only baseline receives only the user question, while the Context LLM also receives task-relevant static context, such as extracted article text or database-derived evidence. This tests whether static evidence alone is sufficient without iterative tool use or task decomposition.

The agentic systems share the same CoSyn toolset but differ in control structure. The **Single-Agent** uses one LLM controller to plan, select tools, and write the final response. The **Multi-Agent** separates these roles: an Orchestrator decomposes the query and routes subtasks to specialist agents for article analysis, database search, and condition prediction, whose reports are aggregated into the final answer. Implementation details are provided in Appendix D.2.

Table 4. Synthesis parameter prediction on the Grinding test set, comparing classical baselines and few-shot LLMs (k=3). Categorical, Numerical, and Overall are weighted averages over field groups; Solvent and Ratio are individual fields. †Solvent uses exact-match on canonical names, while the Categorical aggregate uses solvent-family labels.

	Solvent†		Ratio		Categorical		Numerical		Overall	
Model	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy
Baseline	0.17 ±0.00	0.34 ±0.00	0.87 ±0.00	0.92 ±0.00	0.26 ±0.00	0.43 ±0.00	0.38 ±0.00	0.46 ±0.00	0.35 ±0.00	0.45 ±0.00
RF	0.32 ±0.05	0.37 ±0.01	0.87 ±0.00	0.92 ±0.00	0.54 ±0.00	0.55 ±0.00	0.43 ±0.01	0.46 ±0.01	0.49 ±0.00	0.51 ±0.00
RF-Regression	–	–	0.87 ±0.00	0.91 ±0.00	–	–	0.33 ±0.00	0.37 ±0.00	–	–
RF-Chain	–	–	0.87 ±0.01	0.90 ±0.01	–	–	0.32 ±0.00	0.35 ±0.00	–	–
CB-Regression	–	–	0.87 ±0.00	0.92 ±0.00	–	–	0.34 ±0.01	0.36 ±0.01	–	–
GPT-4o-mini	0.36 ±0.01	0.38 ±0.01	0.85 ±0.00	0.84 ±0.00	0.48 ±0.00	0.49 ±0.00	0.45 ±0.01	0.45 ±0.00	0.46 ±0.00	0.46 ±0.00
GPT-5.4	0.41 ±0.00	0.40 ±0.00	0.88 ±0.00	0.91 ±0.00	0.47 ±0.00	0.46 ±0.00	0.49 ±0.00	0.49 ±0.00	0.48 ±0.00	0.48 ±0.00
Gemini-2.5-Flash	0.41 ±0.01	0.41 ±0.01	0.85 ±0.00	0.86 ±0.00	0.49 ±0.00	0.49 ±0.00	0.48 ±0.00	0.48 ±0.00	0.49 ±0.00	0.48 ±0.00
Gemini-3.1-Pro	0.44 ±0.01	0.44 ±0.01	0.88 ±0.01	0.91 ±0.01	0.50 ±0.00	0.50 ±0.00	0.47 ±0.01	0.48 ±0.01	0.49 ±0.00	0.49 ±0.00

Table 5. Ablation study and comparison of baseline, single-agent, and multi-agent systems on CoSyn-Bench. Capability, complexity, and overall scores are reported on a 1–5 scale; Pass Rate is reported on a 0–1 scale.

	Capability			Complexity		Overall	
System	Retrieval	Prediction	Analysis	Single	Multi	Mean	Pass Rate
LLM-Only	1.71 ±0.08	2.22 ±0.05	1.72 ±0.05	1.94 ±0.03	1.76 ±0.02	1.88 ±0.01	0.00 ±0.00
Context-Augmented LLM	1.66 ±0.06	2.05 ±0.00	1.96 ±0.01	2.26 ±0.10	1.51 ±0.05	2.01 ±0.05	0.07 ±0.00
Single-Agent w/o Retrieval	2.20 ±0.09	3.17 ±0.14	3.09 ±0.15	3.21 ±0.09	2.45 ±0.11	2.96 ±0.09	0.32 ±0.02
Single-Agent w/o Prediction	3.48 ±0.22	2.19 ±0.18	3.29 ±0.04	3.36 ±0.10	2.79 ±0.12	3.17 ±0.02	0.35 ±0.07
Single-Agent w/o Analysis	3.01 ±0.29	3.03 ±0.11	2.40 ±0.02	3.37 ±0.10	2.23 ±0.18	2.99 ±0.12	0.35 ±0.02
Single-Agent	3.61 ±0.08	3.02 ±0.22	3.66 ±0.25	3.74 ±0.40	3.25 ±0.04	3.58 ±0.25	0.47 ±0.09
Multi-Agent	3.47 ±0.09	3.73 ±0.10	3.41 ±0.34	3.67 ±0.22	3.35 ±0.11	3.56 ±0.18	0.48 ±0.16

To isolate each capability, we ablate the Single-Agent by removing Retrieval, Prediction, or Article Analysis one at a time. Each removed capability is replaced with the same static context used by the Context LLM, keeping information access comparable while testing interactive tool use. Tool details and prompts are summarized in Appendices D.1 and D.3. All controllers use `GPT-4o-mini`.

5.2. Evaluation of the Agentic System

We introduce CoSyn-Bench, a compact set of 30 expert-curated, high-information questions testing whether a system can use CoSyn capabilities in realistic synthesis-design scenarios. Each question is a constrained, multi-step synthesis-design task rather than an automatically generated or atomic fact-retrieval item. The benchmark contains 20 single-capability questions requiring retrieval, article analysis, or prediction, and 10 multi-capability questions combining several capabilities. Each item includes a gold answer and structured metadata specifying available information, constraints, and expected answer components. Representative items are shown in Appendix D.4.

Answers are scored by `GPT-5.4` using the question, gold answer, benchmark metadata, execution context, and candidate response. The judge assigns integer scores from 1 to 5 for Task completion, Correctness, Constraint adherence, Groundedness, and Answer quality. We report scores by capability and complexity, plus Pass Rate, defined as the fraction of questions scoring ≥ 4 on every criterion. The judge prompt is summarized in Appendix D.5.

System comparison. Table 5 summarizes the benchmark results. The non-agentic baselines perform poorly: LLM-Only reaches 1.88 overall, and static task-relevant context improves this only to 2.01. Both remain far below agentic systems, showing that evidence alone is insufficient without coordinated tool use. Single-Agent and Multi-Agent achieve similar overall scores, 3.58 and 3.56, but differ in strengths. Single-Agent performs best on Retrieval and Analysis, suggesting one controller works well for compact workflows. Multi-Agent performs best on Prediction, Pass Rate, and multi-capability questions, indicating benefits for prediction-heavy and multi-source tasks. Efficiency measurements show that Multi-Agent requires more calls and wall-clock time but fewer tokens, making it more context-efficient on multi-step queries (Appendix D.6).

Capability ablations. Ablations confirm that each capability supports its task type. Removing Retrieval, Prediction, and Article Analysis reduces the corresponding scores by 39%, 27%, and 34%, respectively. The Article Analysis drop is smaller because the ablated agent still receives article text as static context. Overall, these losses show that performance depends on interactive domain-specific tool use rather than generic LLM reasoning alone.

6. Discussion

Together, our results show that co-crystal AI should address protocol-level synthesis decisions as well as molecular pair compatibility. CoSyn addresses this by converting heterogeneous literature procedures into normalized synthesis records and using them for condition recommendation and agentic assistance.

The database-construction results show that published co-crystal literature contains substantial procedural knowledge, but it becomes useful for modeling only after conversion into method-aware, normalized records. The taxonomy and hierarchical extraction pipeline map heterogeneous procedures to method-specific schemas. At scale, CoSyn extracted 20,783 records, revealing both richness and imbalance: 98.7% of records come from three dominant methods, while rarer routes remain underrepresented. Reporting is also uneven, with ratios and solvents more consistent than amounts, times, and temperatures. Thus, CoSyn-DB defines both the opportunity and evidence boundary for data-driven co-crystal synthesis modeling.

The prediction results show that locally relevant procedural analogues can support method-specific condition recommendation, with a few retrieved examples outperforming broader context. Performance is strongest for well-reported parameters, especially ratios, although ratio prediction partly reflects a strong domain prior and the convention that omitted equimolar ratios imply 1:1 mixtures. Solvent identity and absolute quantities remain harder because they depend more on experiment-specific conditions than stable pair-level patterns. We therefore treat predicted conditions as evidence-grounded starting points for expert review rather than optimized protocols.

The system-level evaluation shows that useful synthesis assistance requires more than static context: systems must retrieve experiments, analyse procedures, predict conditions, and combine results into grounded recommendations. In CoSyn-Bench, static context improves only marginally over LLM-only, while tool-augmented agents improve substantially. Ablations show that removing each tool group selectively weakens the corresponding tasks. Single-Agent is faster and simpler, whereas Multi-Agent is slower but more context-efficient and stronger on prediction-heavy and multi-capability questions. Overall, the key requirement is reliable orchestration of complementary evidence sources, not a specific architecture.

Limitations. The study is limited by (i) biases and uneven reporting in the literature, including limited unsuccessful screens and sparse quantitative parameters; (ii) an operational, non-exhaustive taxonomy that may miss rare routes; (iii) manually annotated validation sets, with detailed Stage 3 validation focused on LAG; (iv) downstream prediction focused on the three dominant methods; (v) moderate predictive accuracy, especially for high-cardinality and experiment-specific parameters; and (vi) no prospective wet-lab validation, with system-level evaluation performed on compact, LLM-judged CoSyn-Bench.

Future directions. Future work should strengthen CoSyn in three directions. First, the evidence base should expand with more literature, negative or unsuccessful experiments where available, and better coverage of rare routes. Second, prediction should be formulated as a multi-solution problem, since one API–coformer pair may have several valid methods and condition sets. Third, recommendations should be combined with physicochemical models, uncertainty estimates, and prospective experimental feedback to distinguish strong from weak evidence and prioritize wet-lab validation.

7. Conclusion

We introduced CoSyn, a literature-grounded LLM-based framework extending co-crystal AI from pair-level screening to protocol-level synthesis design. By combining method-aware extraction with normalization into CoSyn-DB, CoSyn converts heterogeneous full-text procedures into 20,783 structured synthesis records and exposes empirical regularities across dominant synthesis routes. Retrieved few-shot precedents support method-specific condition recommendation, while tool-augmented agents combine retrieval, article analysis, and prediction for grounded synthesis assistance. Across component- and system-level evaluations, CoSyn improves over non-agentic LLM baselines, increasing the CoSyn-Bench score from 1.88 to 3.58. These results position CoSyn as a step toward evidence-grounded co-crystal synthesis recommendation, while highlighting the need for broader validation, uncertainty-aware prediction, and prospective experimental testing.

References

Ai, Q., Meng, Y., et al. Extracting structured data from organic synthesis procedures using a fine-tuned large language model. *Digital Discovery*, 3:1822–1831, 2024. doi: 10.1039/D4DD00091A.

Ball, M., Horvath, D., Kogej, T., Kabeshov, M., and Varnek, A. Predicting reaction conditions: a data-driven perspective. *Chemical Science*, 16:17523–17541, 2025. doi: 10.1039/D5SC03045E.

Birolo, R., Özçelik, R., Aramini, A., Gobetto, R., Chierotti, M. R., and Grisoni, F. Deep supramolecular language processing for co-crystal prediction. *Angewandte Chemie International Edition*, 64:e202507835, 2025. doi: 10.1002/anie.202507835.

Boiko, D. A., MacKnight, R., Kline, B., and Gomes, G. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023. doi: 10.1038/s41586-023-06792-0.

Bolla, G., Sarma, B., Nangia, A. K., et al. Crystal engineering of pharmaceutical cocrystals in the discovery and development of improved drugs. *Chemical Reviews*, 122(13):11514–11603, 2022. doi: 10.1021/acs.chemrev.1c00987.

Bran, A. M., Cox, S., Schilter, O., Baldassari, C., White, A. D., and Schwaller, P. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6:525–535, 2024. doi: 10.1038/s42256-024-00832-8.

Buddhadev, S. S. and Garala, K. C. Pharmaceutical cocrystals—a review. *Proceedings*, 62(1):14, 2020. doi: 10.3390/proceedings2020062014.

Chase, H. Langchain. https://github.com/langchain-ai/langchain, 2022. Released: 2022-10-17.

Chen, L.-Y. and Li, Y.-P. Enhancing chemical synthesis: a two-stage deep neural network for predicting feasible reaction conditions. *Journal of Cheminformatics*, 16(1):11, 2024. doi: 10.1186/s13321-024-00805-4.

Chen, Z., Chen, S., Ning, Y., Zhang, Q., Wang, B., Yu, B., Li, Y., Liao, Z., Wei, C., Lu, Z., et al. Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=6z4YKr0GK6.

Chong, C. and Colindres, J. Litxbench: A benchmark for extracting experiments from scientific literature. *arXiv*, 2026. URL https://arxiv.org/abs/2604.07649.

Dagdelen, J., Dunn, A., Lee, S., Walker, N., Rosen, A. S., Ceder, G., Persson, K. A., and Jain, A. Structured information extraction from scientific text with large language models. *Nature Communications*, 15:1418, 2024. doi: 10.1038/s41467-024-45563-x.

Devogelaer, J.-J., Meekes, H., Tinnemans, P., Vlieg, E., and de Gelder, R. Co-crystal prediction by artificial neural networks. *Angewandte Chemie International Edition*, 59(48):21711–21718, 2020. doi: 10.1002/anie.202009467.

Devogelaer, J.-J. J., Brugman, S. J. T., Meekes, H., Tinnemans, P., Vlieg, E., and de Gelder, R. Cocrystal design by network-based link prediction. *CrystEngComm*, 21:6875–6885, 2019. doi: 10.1039/C9CE01110B.

Douroumis, D., Ross, S. A., and Nokhodchi, A. Advanced methodologies for cocrystal synthesis. *Advanced Drug Delivery Reviews*, 117:178–195, 2017. doi: 10.1016/j.addr.2017.07.008.

Fan, V., Qian, Y., Wang, A., Wang, A., Coley, C. W., and Barzilay, R. Openchemie: An information extraction toolkit for chemistry literature. *Journal of Chemical Information and Modeling*, 64(14):5576–5589, 2024. doi: 10.1021/acs.jcim.4c00572.

Gao, H., Struble, T. J., Coley, C. W., Wang, Y., Green, W. H., and Jensen, K. F. Using machine learning to predict suitable conditions for organic reactions. *ACS Central Science*, 4(11):1465–1476, 2018. doi: 10.1021/acscentsci.8b00357.

Groom, C. R., Bruno, I. J., Lightfoot, M. P., and Ward, S. C. The cambridge structural database. *Acta Crystallographica Section B: Structural Science, Crystal Engineering and Materials*, 72(2):171–179, 2016. doi: 10.1107/S2052520616003954.

Gu, K., Shang, R., Jiang, R., Kuang, K., Lin, R.-J., Lyu, D., Mao, Y., Pan, Y., Wu, T., Yu, J., Zhang, Y., Zhang, T. M., Zhu, L., Merrill, M. A., Heer, J., and Althoff, T. Blade: Benchmarking language model agents for data-driven science. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 13936–13971, 2024. doi: 10.18653/v1/2024.findings-emnlp.815. URL https://aclanthology.org/2024.findings-emnlp.815/.

Gubina, N., Dmitrenko, A., Solovev, G., Yamshchikova, L., Petrov, O., Lebedev, I., Serov, N., Kirgizov, G., Nikitin, N., and Vinogradov, V. Hybrid generative ai for de novo design of co-crystals with enhanced tabletability. In *Advances in Neural Information Processing Systems*, 2024.

Guo, J., Yang, S., Wang, C., Liu, J., Guo, Y., Yang, Z., Zhao, X., and Pu, X. Cocrystal prediction tool (CCPT): A web server for deep learning-assisted cocrystal screening and density evaluation. *Crystal Growth & Design*, 24(20):8407–8414, 2024. doi: 10.1021/acs.cgd.4c00915.

Hawizy, L., Jessop, D. M., Adams, N., and Murray-Rust, P. Chemicaltagger: A tool for semantic text-mining in

chemistry. *Journal of Cheminformatics*, 3(1):17, 2011. doi: 10.1186/1758-2946-3-17.

Jablonka, K. M., Schwaller, P., Ortega-Guerrero, A., et al. Leveraging large language models for predictive chemistry. *Nature Machine Intelligence*, 6:161–169, 2024. doi: 10.1038/s42256-023-00788-1.

Jessop, D. M., Adams, S. E., Willighagen, E. L., Hawizy, L., and Murray-Rust, P. Oscar4: A flexible architecture for chemical text-mining. *Journal of Cheminformatics*, 3 (1):41, 2011. doi: 10.1186/1758-2946-3-41.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023. doi: 10.1145/3571730.

Jiang, Y., Yang, Z., Guo, J., Li, H., Liu, Y., Guo, Y., Li, M., and Pu, X. Coupling complementary strategy to flexible graph neural network for quick discovery of coformer in diverse co-crystal materials. *Nature Communications*, 12: 5950, 2021. doi: 10.1038/s41467-021-26226-7.

Karagianni, A., Malamatari, M., and Kachrimanis, K. Pharmaceutical cocrystals: New solid phase modification approaches for the formulation of apis. *Pharmaceutics*, 10 (1):18, 2018. doi: 10.3390/pharmaceutics10010018.

Karimi-Jafari, M., Padrela, L., Walker, G. M., and Croker, D. M. Creating cocrystals: A review of pharmaceutical cocrystal preparation routes and applications. *Crystal Growth & Design*, 18(10):6370–6387, 2018a. doi: 10.1021/acs.cgd.8b00933.

Karimi-Jafari, M., Padrela, L., Walker, G. M., and Croker, D. M. A review of pharmaceutical cocrystal preparation routes and applications. *Crystal Growth & Design*, 18 (10):6370–6387, 2018b. doi: 10.1021/acs.cgd.8b00933.

Kearnes, S. M., Maser, M. R., Wleklinski, M., Kast, A., Doyle, A. G., Dreher, S. D., Hawkins, J. M., Jensen, K. F., and Coley, C. W. The open reaction database. *Journal of the American Chemical Society*, 143(45):18820–18826, 2021. doi: 10.1021/jacs.1c09820.

Landrum, G. Rdkit: Open-source cheminformatics. http://www.rdkit.org.

Marcos Valdez, M. M., Sperandeo, N. R., Bueno, M. S., and Garnero, C. Pharmaceutical cocrystals in drug-delivery technologies: Advances from rational design to therapeutic applications. *Pharmaceutics*, 18(1):128, 2026. doi: 10.3390/pharmaceutics18010128.

Mirza, A. et al. A framework for evaluating the chemical knowledge and reasoning abilities of large language models against the expertise of chemists. *Nature Chemistry*, 17:1027–1034, 2025. doi: 10.1038/s41557-025-01815-x.

Mswahili, M. E., Lee, M.-J., Martin, G. L., Kim, J., Kim, P., Choi, G. J., and Jeong, Y.-S. Cocrystal prediction using machine learning models and descriptors. *Applied Sciences*, 11(3):1323, 2021. doi: 10.3390/app11031323.

Odobesku, R., Romanova, K., Mirzaeva, S., Zagorulko, O., Sim, R., Khakimullin, R., Razlivina, J., Dmitrenko, A., and Vinogradov, V. Agent-based multimodal information extraction for nanomaterials. *npj Computational Materials*, 11:194, 2025. doi: 10.1038/s41524-025-01674-7.

Pawar, N., Saha, A., Nandan, N., and Parambil, J. V. Solution cocrystallization: A scalable approach for cocrystal production. *Crystals*, 11(3):303, 2021. doi: 10.3390/cryst11030303.

Prein, T., Pan, E., Jehkul, J., Weinmann, S., Olivetti, E. A., and Rupp, J. L. M. Language models enable data-augmented synthesis planning for inorganic materials. *CoRR*, abs/2506.12557, 2025. doi: 10.48550/arXiv.2506.12557.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

Ramos, M. C. et al. A review of large language models and autonomous agents in chemistry. *Chemical Science*, 16: 2514–2572, 2025. doi: 10.1039/D4SC03921A.

Ruan, Y., Lu, C., Xu, N., et al. An automatic end-to-end chemical synthesis development platform powered by large language models. *Nature Communications*, 15: 10160, 2024. doi: 10.1038/s41467-024-54457-x.

Schilling-Wilhelmi, M., Ríos-García, M., et al. From text to insight: large language models for chemical data extraction. *Chemical Society Reviews*, 54:1125–1150, 2025. doi: 10.1039/D4CS00913D.

Shi, L., Liu, Z., Yang, Y., Wu, W., Zhang, Y., Zhang, H., Lin, J., Wu, S., Chen, Z., Li, R., Wang, N., Liu, Z., Tan, H., Gao, H., Zhang, Y., and Wang, G. LLM-based MOFs synthesis condition extraction using few-shot demonstrations. *arXiv preprint arXiv:2408.04665*, 2024. doi: 10.48550/arXiv.2408.04665. URL https://arxiv.org/abs/2408.04665.

Song, T., Luo, M., Zhang, X., et al. A multiagent-driven robotic ai chemist enabling autonomous chemical research on demand. *Journal of the American Chemical Society*, 147(15):12534–12545, 2025a. doi: 10.1021/jacs.4c17738.

Song, Y., Ding, Y., Su, J., Li, J., and Ji, Y. Unlocking the potential of machine learning in co-crystal prediction by a novel approach integrating molecular thermodynamics. *Angewandte Chemie International Edition*, 64: e202502410, 2025b. doi: 10.1002/anie.202502410.

Swain, M. C. and Cole, J. M. Chemdataextractor: A toolkit for automated extraction of chemical information from the scientific literature. *Journal of Chemical Information and Modeling*, 56(10):1894–1904, 2016. doi: 10.1021/acs.jcim.6b00207.

Vaucher, A. C., Schwaller, P., Geluykens, J., Nair, V. H., Iuliano, A., and Laino, T. Inferring experimental procedures from text-based representations of chemical reactions. *Nature Communications*, 12:2573, 2021. doi: 10.1038/s41467-021-22951-1.

Vepreva, A., Razlivina, J., Eremeyeva, M., Gubina, N., Orlova, A., Dmitrenko, A., Xenia, K., Jyakhwo, S., Vasilev, N. A., Sarkisyan, A., Chernyshov, I. Y., Vinogradov, V., and Dmitrenko, A. Chemx: A collection of chemistry datasets for benchmarking automated information extraction. In *NeurIPS 2025 Datasets and Benchmarks Track*, 2025. URL https://openreview.net/forum?id=PU4AcbqKp4.

Wang, D., Yang, Z., Zhu, B., Mei, X., and Luo, X. Machine-learning-guided cocrystal prediction based on large data base. *Crystal Growth & Design*, 20(10):6610–6621, 2020. doi: 10.1021/acs.cgd.0c00767.

Wang, Z., Lin, K., Pei, J., and Lai, L. Reacon: a template- and cluster-based framework for reaction condition prediction. *Chemical Science*, 16:854–866, 2025. doi: 10.1039/D4SC05946H.

Wicker, J. G. P., Crowley, L. M., Robshaw, O., Little, E. J., Stokes, S. P., Cooper, R. I., and Lawrence, S. E. Will they co-crystallize? *CrystEngComm*, 19(36):5336–5340, 2017. doi: 10.1039/C7CE00587C.

Yadav, V., Kumar, R., Sharma, M., et al. Pharmaceutical cocrystals: An overview of synthesis, characterization, and applications. *Journal of Molecular Structure*, 1349: 143682, 2025. doi: 10.1016/j.molstruc.2025.143682.

Yang, Y., Shi, R., Li, Z., Jiang, S., Lu, B.-L., Yang, Y., and Zhao, H. Batgpt-chem: A foundation large model for chemical engineering. *Research*, 8:0827, 2025. doi: 10.34133/research.0827.

Yu, B., Baker, F. N., Chen, Z., Herb, G., Gou, B., Adu-Ampratwum, D., Ning, X., and Sun, H. Tooling or not tooling? the impact of tools on language agents for chemistry problem solving. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 7635–7655. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-naacl.424. URL https://aclanthology.org/2025.findings-naacl.424/.

Zeng, Q., Zhang, Y., Peng, Y., Zeng, Q., Sun, G., Guo, M., and Cai, T. Interpretable machine learning for solvent prediction and mechanistic insights in multi-component crystal screening. *Chemical Engineering Journal*, 524: 169397, 2025. doi: 10.1016/j.cej.2025.169397.

Zhang, C., Zhang, L., Yang, D., Tong, H. H. Y., Lu, Y., and Zhou, Z. From traditional screening to machine learning facilitated development of pharmaceutical cocrystals. *Chinese Chemical Letters*, pp. 111828, 2025a. doi: 10.1016/j.cclet.2025.111828.

Zhang, Y., Yu, R., Zeng, K., Li, D., Zhu, F., Yang, X., Jin, Y., and Xu, Y. Text-augmented multimodal llms for chemical reaction condition recommendation. *CoRR*, abs/2407.15141, 2024. doi: 10.48550/arXiv.2407.15141.

Zhang, Y., Khan, S. A., Mahmud, A., Yang, H., Lavin, A., Levin, M., Frey, J., Dunnmon, J., Evans, J., Bundy, A., Džeroski, S., Tegnér, J., and Zenil, H. Exploring the role of large language models in the scientific method: from hypothesis to discovery. *npj Artificial Intelligence*, 1(1), 2025b. doi: 10.1038/s44387-025-00019-5.

Zheng, Z., Zhang, O., Borgs, C., Chayes, J. T., and Yaghi, O. M. Chatgpt chemistry assistant for text mining and the prediction of mof synthesis. *Journal of the American Chemical Society*, 145(32):18048–18062, 2023. doi: 10.1021/jacs.3c05819.

A. Impact Statement

COSYN can support co-crystal synthesis research by converting dispersed literature procedures into reusable, structured evidence for synthesis-design decisions. Potential benefits include faster literature analysis, more systematic comparison of synthesis conditions, improved reuse of prior experimental knowledge, and evidence-grounded suggestions for initial protocols. The system may be especially useful for prioritizing candidate synthesis conditions before laboratory testing and for making literature-derived assumptions more explicit.

At the same time, COSYN inherits biases and gaps from published reports, including uneven parameter reporting, over-representation of successful experiments, and limited coverage of unsuccessful screens or rare synthesis routes. Its predictions should therefore be treated as expert-reviewed starting points rather than optimized or validated protocols. Any wet-lab use requires standard chemical safety assessment, human oversight, appropriate institutional procedures, and prospective experimental validation before conclusions are drawn from recommended conditions.

B. Dataset Construction

B.1. Source Corpus Statistics

Figure 4 summarizes the composition of the collected source corpus. We report the distribution of retrieved articles by publication year and open-access status to characterize the temporal coverage and accessibility of the literature used for extraction.

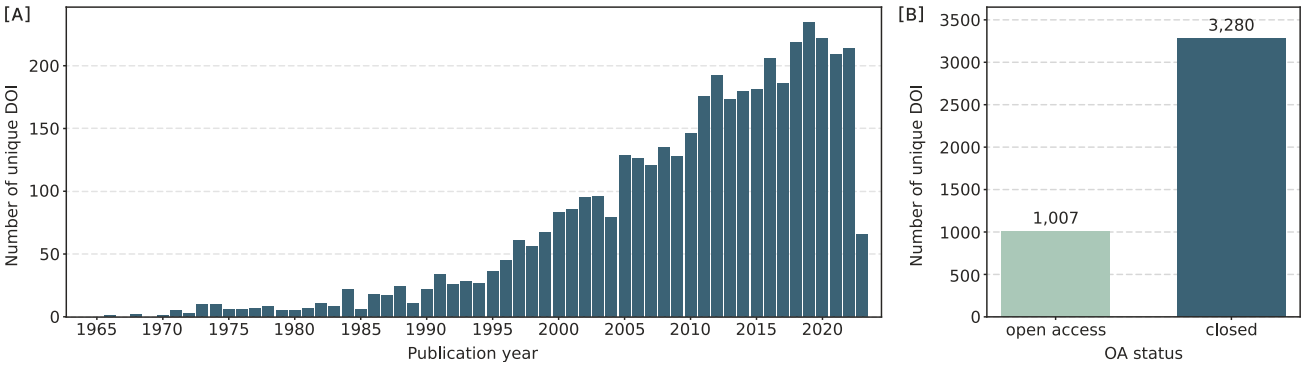

Figure 4. Source corpus statistics. [A] Publication-year distribution of the collected CSD-derived source DOIs. [B] Open-access status of the source corpus, summarizing the accessibility of articles used for full-text retrieval and extraction.

B.2. Structured representation of co-crystal synthesis methods

B.2.1. CLASSIFICATION OF CO-CRYSTAL SYNTHESIS METHODS

Co-crystal synthesis procedures differ in the physical mechanism used to induce co-crystallization, the required experimental setup, and the process variables reported in experimental sections. To account for this variability, we defined 12 co-crystal synthesis method categories. Table 6 summarizes the corresponding method descriptions, generalized parameter groups, and links to the procedural templates. The method descriptions were used in strategy S2-B to support method-aware extraction.

Table 6. Operational taxonomy for co-crystal synthesis extraction. Method categories are defined by abbreviations, method descriptions, generalized parameter groups, and procedural templates used for method-aware extraction.

Method	Description	Key parameters	Template
Solution crystallization (SC)	Solution crystallization is a widely used method for co-crystal synthesis that involves dissolving the individual coformers in a suitable solvent or solvent mixture. Co-crystallization is then induced by allowing the solvent to evaporate (slow solvent removal to achieve supersaturation), cooling the solution (temperature reduction to decrease solubility), or introducing several immiscible phases (crystal formation at liquid–liquid or liquid–gas boundaries).	component composition / solvent system / stirring conditions / evaporation or cooling conditions / filtration and drying	T1
Grinding (GR)	Grinding is a mechanochemical method for co-crystal synthesis that entails mixing solid coformers together through manual (mortar and pestle) or mechanical (ball milling) means to induce co-crystallization. This approach can be performed as neat grinding, using only the solid constituents, or as LAG, where a minimal amount of solvent is added to promote better molecular contact.	component composition / optional liquid additive / grinding apparatus / milling conditions / drying	T2
Slurry (SL)	Slurry is a solution-mediated method for co-crystal synthesis that entails suspending solid coformers in a minimal amount of solvent to induce co-crystallization. This approach involves allowing the coformers to interact in the solvent over a period (at constant temperature or with temperature variation), followed by separation and drying of the solid product. The approach relies on the dynamic equilibrium between dissolved and solid phases to achieve desired crystal properties.	component composition / solvent volume / stirring conditions / filtration and solvent evaporation / drying	T3
Anti-solvent (AS)	The anti-solvent method is a solution-based technique for co-crystal synthesis that involves the addition of a second solvent, the "anti-solvent," to a solution containing the mixture of coformers. The anti-solvent is chosen for its ability to reduce the solubility of the cocrystal, thereby inducing its precipitation from the solution. This method often requires specialized equipment in which a separate tank for anti-solvent and peristaltic pump will be used.	component composition / anti-solvent system / addition setup / addition and mixing conditions / filtration and drying	T4
Hot-melt extrusion (HME)	Hot-melt extrusion is a continuous, solvent-free method for co-crystal synthesis that involves feeding a mixture of solid coformers into a heated extruder, where they are subjected to mixing and conveying by rotating screws. The extruder is typically divided into several zones, each with independently controlled temperatures, facilitating melting of at least one component and promoting molecular contact and subsequent co-crystallization as the melt cools.	component composition / optional additional substance / extruder configuration / feeding and temperature profile / post-processing	T5

Continued on next page

Method	Description	Key parameters	Template
Freeze drying (FD)	Freeze drying (or lyophilization) is a solvent-removal method for co-crystal synthesis that involves dissolving coformers in a solvent, freezing the solution, and then removing the solvent through sublimation under vacuum to obtain the co-crystal. This technique can be performed using special equipment capable of maintaining low pressure to create amorphous or nanostructured products.	component composition / solvent / freeze-dryer setup / freezing and lyophilization conditions / product collection	T6
Spray drying (SD)	Spray drying is a continuous method for co-crystal synthesis that transforms liquids into dry powders through atomization into a hot gas stream, followed by rapid solvent evaporation and particle collection. During this process, the cocrystals nucleate and grow within highly supersaturated regions of the coformer substance due to rapid solvent evaporation and solidification of the generated droplets.	component composition / solvent system / spray-dryer setup / atomization and drying conditions / powder collection	T7
Ultrasound (US)	Ultrasound (or sonocrystallization) is a solution-based co-crystallization method that involves the application of high-frequency sound waves to induce co-crystallization. This approach is typically performed by subjecting a supersaturated solution of the coformers to ultrasonic pulses, which can promote nucleation and crystal growth.	component composition / solvent system / ultrasonic setup / sonication conditions / filtration and drying	T8
Supercritical fluid (SF)	Supercritical fluid is a method for synthesizing co-crystals where solid coformers are exposed to a supercritical fluid, typically supercritical carbon dioxide ($scCO_2$), which acts as a solvent or anti-solvent to facilitate co-crystal formation. The coformers may partially dissolve or become highly dispersed within the supercritical fluid, and the mixture is stirred for a defined period to allow co-crystallization to occur.	component composition / supercritical fluid or co-solvent / reactor setup / pressure–temperature–flow conditions / depressurization and drying	T9
Electrospray (ES)	Electrospray is a solution-based method for co-crystal synthesis that involves atomizing a solution containing the co-crystal components into a fine mist of charged droplets using a high electric field. As the solvent evaporates from these droplets, the supersaturation of the components increases, leading to the rapid formation of cocrystals.	component composition / solvent system / syringe and electrospray setup / flow–voltage–distance conditions / collection and drying	T10
Resonant acoustic (RA)	Resonant acoustic co-crystal synthesis is a mechanochemical method for co-crystal formation that involves intense mixing of solid coformers (sometimes with a small amount of solvent), through high-frequency and high-intensity vibratory motion generated by a resonant acoustic mixer. This technique leverages powerful resonant frequencies to promote contact and induce co-crystallization.	component composition / optional liquid additive / resonant acoustic mixer / acceleration–frequency–time conditions / drying	T11

Continued on next page

Method	Description	Key parameters	Template
Laser irradiation (LI)	Laser irradiation is a co-crystallization method that entails exposing a physical mixture of solid coformers to a focused laser beam to induce co-crystallization. This approach can be performed directly on dry powder blends, where the laser's energy promotes the necessary molecular mobility for co-crystal formation.	component composition / optional additive / substrate and laser setup / irradiation conditions / collection and storage	T12

B.2.2. METHOD-SPECIFIC PROCEDURAL TEMPLATES

The templates below define normalized procedural patterns for the 12 co-crystal synthesis method categories listed in Table 6. These templates were used in strategy S2-C to represent synthesis procedures in a method-specific format. They were also used during Stage 3 to construct extraction prompts and to guide the prediction of method-specific experimental parameters.

Each template contains slots marked by [...]. These slots correspond to the main entities typically reported in experimental procedures, including component composition, solvent system, apparatus, operating conditions, and other. Optional fields are explicitly marked only for the solution crystallization, grinding, and slurry templates using [...], in all other templates, bracketed fields denote the target extraction slots without an additional optionality annotation. For all templates except solution crystallization, grinding, and slurry, separate slots for the second and third solvents were omitted for readability.

T1. SOLUTION CRYSTALLIZATION

```
Solution crystallization was performed using [Amount of API] [Amount Unit of API] of
[API] and [Amount of Coformer] [Amount Unit of Coformer] of [Coformer] in a ratio
of [Part of Ratio API]:[Part of Ratio Coformer]. The components were dissolved
in [Amount of Solvent] [Solvent Amount Unit] of [Solvent], [Amount of Second
Solvent] [Second Solvent Amount Unit] of [Second Solvent], and [Amount of Third
Solvent] [Third Solvent Amount Unit] of [Third Solvent]. The solution was stirred at
[Mixing Temperature] [Mixing Temperature Unit] for [Mixing Time] [Mixing Time Unit].
The resulting solution was filtered and covered with [Covering Method] featuring
[Description of Holes]. The solution was then left at room temperature for [Time
of Cooling or Evaporation] [Time of Cooling or Evaporation Unit]. The resulting solid
was collected by filtration and dried to obtain the co-crystal.
```

T2. GRINDING

```
Mechanochemical synthesis was carried out using a [Mixing Apparatus]. The mixture
was processed at [Mixing Frequency] [Frequency Unit] for [Mixing Time] [Mixing Time
Unit] at [Mixing Temperature] [Mixing Temperature Unit]. A blend of [Amount of
API] [Amount Unit of API] of [API] and [Amount of Coformer] [Amount Unit of Coformer]
of [Coformer] was prepared in a ratio of [Part of Ratio API]:[Part of Ratio Coformer].
Additionally, [Amount of Solvent] [Solvent Amount Unit] of [Solvent], [Amount of
Second Solvent] [Second Solvent Amount Unit] of [Second Solvent], and [Amount of Third
Solvent] [Third Solvent Amount Unit] of [Third Solvent] were added. The mixture was
then ground. The resulting co-crystal was dried for [Drying Time] [Drying Time Unit]
at [Drying Temperature] [Drying Temperature Unit].
```

T3. Slurry

Slurry crystallization was conducted by suspending [Amount of API] [Amount Unit of API] of [API] and [Amount of Coformer] [Amount Unit of Coformer] of [Coformer] in a ratio of [Part of Ratio API]:[Part of Ratio Coformer]. The solids were suspended in [Amount of Solvent] [Solvent Amount Unit] of [Solvent], [Amount of Second Solvent] [Second Solvent Amount Unit] of [Second Solvent], and [Amount of Third Solvent] [Third Solvent Amount Unit] of [Third Solvent]. The suspension was stirred at [Mixing Temperature] [Mixing Temperature Unit] for [Mixing Time] [Mixing Time Unit]. The solid product was separated by filtration, and residual solvent was evaporated at [Evaporation Temperature] [Evaporation Temperature Unit] for [Evaporation Time] [Evaporation Time Unit]. The resulting co-crystal was dried for [Drying Time] [Drying Time Unit] at [Drying Temperature] [Drying Temperature Unit].

T4. Anti-solvent

Anti-solvent crystallization was performed using [Amount of API] [Amount Unit of API] of [API] and [Amount of Coformer] [Amount Unit of Coformer] of [Coformer] in a ratio of [Part of Ratio API]:[Part of Ratio Coformer]. The components were dissolved in [Amount of Solvent] [Solvent Amount Unit] of [Solvent]. An anti-solvent stream containing [Amount of Anti-solvent] [Anti-solvent Amount Unit] of [Anti-solvent] was added to the solution at [Speed of Addition] [Speed of Addition Unit] using a peristaltic pump, while mixing at [Mixing Frequency] [Frequency Unit] and [Mixing Temperature] [Mixing Temperature Unit]. The resulting suspension/solution was filtered and dried at [Drying Temperature] [Drying Temperature Unit].

T5. Hot-melt extrusion

Hot-melt extrusion was performed using a [Type of Extruder] with a screw diameter of [Screw Diameter] [Screw Diameter Unit] and an L/D ratio of [L/D Ratio]. [Amount of API] [Amount Unit of API] of [API] and [Amount of Coformer] [Amount Unit of Coformer] of [Coformer] were weighed in a ratio of [Part of Ratio API]:[Part of Ratio Coformer]. [Amount of Additional Substance] [Amount of Additional Substance Unit] of [Additional Substance] was added, and the components were pre-mixed using a [Mixing Apparatus] for [Mixing Time] [Mixing Time Unit]. The batch size was [Batch Size] [Batch Size Unit]. The blend was fed into the extruder at [Feed Rate] [Feed Rate Unit]. The extruder comprised [Amount of Types of Zones] zones ([List of Types of Zones]) with temperatures set to [List of Temperature in Zones] [Temperature in Zones Unit]. The screw configuration ([List of Screw Type]) and screw speed ([Screw Speed] [Screw Speed Unit]) were selected according to the zone design. Extrusion was carried out using a [Type of Die] with a diameter of [Die Diameter] [Die Diameter Unit]. The resulting co-crystal was pulverized using an agate mortar and pestle.

T6. Freeze drying

Freeze-drying co-crystal synthesis was performed using a [Freeze-Dryer Apparatus] at [Freeze-Drying Temperature] [Freeze-Drying Temperature Unit] and under [Freeze-Drying Pressure] [Freeze-Drying Pressure Unit]. A mixture containing [Amount of API] [Amount Unit of API] of [API] and [Amount of Coformer] [Amount Unit of Coformer] of [Coformer] was prepared in a ratio of [Part of Ratio API]:[Part of Ratio Coformer]. The components were dissolved in [Amount of Solvent] [Solvent Amount Unit] of [Solvent]. The solution was then frozen at [Cooling Temperature] [Cooling Temperature Unit] and lyophilized to obtain the co-crystal.

T7. Spray drying

```
Spray-drying was performed using [Amount of API] [Amount Unit of API] of [API] and
[Amount of Coformer] [Amount Unit of Coformer] of [Coformer] in a ratio of [Part of
Ratio API]:[Part of Ratio Coformer]. The components were dissolved in [Amount of
Solvent] [Solvent Amount Unit] of [Solvent]. The resulting solution was spray-dried
using a [Spray Dryer Model] with [Drying Gas] as the drying gas. Processing
conditions were: air flow [Air Flow Speed] [Air Flow Unit], inlet temperature [Inlet
Temperature] [Inlet Temperature Unit], aspiration rate [Aspiration Rate]% ([Aspiration
Flow Rate] [Flow Rate Unit]), and feed rate [Feed Rate] [Feed Rate Unit]. The outlet
temperature was [Outlet Temperature] [Outlet Temperature Unit].
```

T8. Ultrasound

```
Ultrasound-assisted synthesis was performed using [Amount of API] [Amount Unit of
API] of [API] and [Amount of Coformer] [Amount Unit of Coformer] of [Coformer] in a
ratio of [Part of Ratio API]:[Part of Ratio Coformer]. The components were mixed with
[Amount of Solvent] [Solvent Amount Unit] of [Solvent]. The mixture was sonicated for
[Sonication Time] [Sonication Time Unit] using an [Ultrasonic Apparatus] operated
at [Mixing Frequency] [Mixing Frequency Unit] and [Mixing Temperature] [Mixing
Temperature Unit], with a power output of [Power] [Power Unit]. The resulting solid
was isolated by filtration and dried for [Drying Time] [Drying Time Unit] at [Drying
Temperature] [Drying Temperature Unit].
```

T9. Supercritical fluid

```
Supercritical fluid synthesis was performed using [Amount of API] [Amount Unit of
API] of [API] and [Amount of Coformer] [Amount Unit of Coformer] of [Coformer]
in a ratio of [Part of Ratio API]:[Part of Ratio Coformer]. The components were
mixed with [Amount of Solvent] [Solvent Amount Unit] of [Solvent]. The mixture was
processed in a [Supercritical Apparatus] at [Pressure] [Pressure Unit] and [Mixing
Temperature] [Mixing Temperature Unit], with a flow rate of [Flow Rate] [Flow Rate
Unit]. The resulting co-crystal was dried for [Drying Time] [Drying Time Unit] at
[Drying Temperature] [Drying Temperature Unit].
```

T10. Electrospray

```
Electrospraying synthesis was conducted using [Amount of API] [Amount Unit of API]
of [API] and [Amount of Coformer] [Amount Unit of Coformer] of [Coformer] in a ratio
of [Part of Ratio API]:[Part of Ratio Coformer]. The components were dissolved in
[Amount of Solvent] [Solvent Amount Unit] of [Solvent] and loaded into a [Syringe
Volume] [Syringe Volume Unit] syringe. The prepared solution was electrosprayed using
an [Electrospraying Apparatus]. The flow rate, voltage, and needle tip-to-collector
distance were set to [Flow Rate] [Flow Rate Unit], [Voltage] [Voltage Unit], and [Tip
to Collector Distance] [Tip to Collector Distance Unit], respectively. The deposited
co-crystal was collected on the collector and dried for [Drying Time] [Drying Time
Unit] at [Drying Temperature] [Drying Temperature Unit].
```

T11. Resonant acoustic

```
Resonant acoustic synthesis was performed using [Amount of API] [Amount Unit of API]
of [API] and [Amount of Coformer] [Amount Unit of Coformer] of [Coformer] in a ratio
of [Part of Ratio API]:[Part of Ratio Coformer]. The components were combined with
[Amount of Solvent] [Solvent Amount Unit] of [Solvent]. The reaction mixture was
processed for [Reaction Time] [Reaction Time Unit] at [Reaction Times of Gravity
```

```
Acceleration] [Reaction Times of Gravity Acceleration Unit] ×g and a frequency of
[Reaction Frequency] [Reaction Frequency Unit]. The resulting co-crystal was dried
for [Drying Time] [Drying Time Unit] at [Drying Temperature] [Drying Temperature
Unit].
```

T12. LASER IRRADIATION

```
Laser irradiation synthesis was performed using [Amount of API] [Amount Unit of
API] of [API] and [Amount of Coformer] [Amount Unit of Coformer] of [Coformer] in a
ratio of [Part of Ratio API]:[Part of Ratio Coformer]. The components were mixed
with [Amount of Solvent] [Solvent Amount Unit] of [Solvent]. The mixture was spread
as a thin layer on [Foil Material] foil. Irradiation was performed using a [Laser
Apparatus] with a focal length of [Focal Length] [Focal Length Unit], operated at
[Power] [Power Unit] and a raster speed of [Raster Speed] [Raster Speed Unit]. The
resulting co-crystal was collected from the foil and dried for [Drying Time] [Drying
Time Unit] at [Drying Temperature] [Drying Temperature Unit].
```

B.3. Extraction metrics

We evaluated the extraction pipeline separately for the three stages, since each stage produces a different type of output: procedure presence, method labels, and structured parameter values. Unless stated otherwise, precision, recall, F1-score, and accuracy were computed as:

$$\text{Precision} = \frac{TP}{TP + FP}, \qquad \text{Recall} = \frac{TP}{TP + FN},$$

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \qquad \text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}.$$

Stage 1: Procedure detection. This stage was evaluated as binary classification at the DOI level. Each article was labeled as either containing or not containing an explicit co-crystal synthesis procedure. Empty or missing-value responses were treated as negative predictions; all other responses were treated as positive.

Stage 2: Method classification. This stage was evaluated by comparing the predicted set of synthesis methods with the gold method set for each DOI. Aggregate scores were computed over complete method-set labels using weighted precision, recall, and F1-score. We also computed per-method one-vs-rest scores for detailed analysis.

Stage 3: Parameter extraction. This stage was evaluated at the level of synthesis records and extracted parameters. For each record and parameter, predicted and gold values were compared as multisets, preserving duplicate values. The error counts were defined as follows:

- **True positives (TP):** values that appear in both the predicted and gold multisets.
- **False positives (FP):** predicted values that are absent from the gold multiset or appear too many times.
- **False negatives (FN):** gold values that are missing from the prediction or appear too few times.

Precision, recall, and F1-score follow the definitions above, while Stage 3 accuracy is defined as:

$$\text{Accuracy}_{\text{S3}} = \frac{TP}{TP + FP + FN}.$$

Thus, Stage 3 accuracy measures value-level overlap rather than standard classification accuracy with true negatives. Final Stage 3 scores were obtained by averaging parameter-level scores across all evaluated parameters.

B.4. Extraction Prompting Strategies

The prompts presented below were used in the hierarchical extraction pipeline for co-crystal synthesis data. Each stage addresses a separate decision: (i) whether an article contains an extractable synthesis procedure, (ii) which synthesis method or methods are described, and (iii) which method-specific parameters should be extracted. This staged design reduces the number of irrelevant fields considered at each step and allows later prompts to operate on progressively more specific information.

B.4.1. Stage 1: Procedure detection

The Stage 1 prompt was used to determine whether a full-text article contains an explicit co-crystal synthesis procedure. During prompting-strategy optimisation, we tested three [Response formats]:

- **S1-A:** If the article describes such techniques, respond with 1. If the article does not mention any specific techniques for co-crystal synthesis, respond with 0. Answer only 1 or 0, do not add anything else to your answer.
- **S1-B:** If the article contains a complete method for synthesizing a co-crystal with the name of the method and/or parameters, write the entire text that relates to the co-crystal synthesis method. If the article does not describe a complete method for the synthesis of a co-crystal, specify `None` in the answer.
- **S1-C:** If the article contains a complete co-crystal synthesis method with the name of the method and/or parameters, write their text separately for each molecular pair and the type of technique separately. If the article does not describe the complete method of crystal synthesis, write `None`.

The full Stage 1 prompt shown below contains the placeholder [Response format], which was instantiated according to the corresponding strategy.

```
You are a chemistry assistant specializing in co-crystals.
Your task is to analyze scientific articles and determine whether the article
describes a complete and explicit synthesis method specifically for a co-crystal (not
for a host molecule, not for analysis, and not just molecular interactions).
Inclusion criteria:
 1. The article must describe a specific method for synthesizing a co-crystal, such
    as grinding, contact formation, hot-melt extrusion, evaporation, cooling, slurry,
    antisolvent, spray drying, reactive co-crystallization, ultrasound, freeze drying,
    hydrothermal, supercritical fluid, microfluidic, electrospray, resonant acoustic,
    or laser irradiation.
 2. The article must also provide at least one key synthesis parameter, such as
    temperature, time, solvent, or the ratio of starting materials.
 3. Additionally, look for keywords indicating co-crystal formation, such as
    co-assembled, molar ratio N:N, two-component molecular complex, or inclusion
    complex.
Do not consider the following as synthesis methods:
 1. Synthesis of a host molecule (this is **not** a co-crystal synthesis method).
    If the article describes the synthesis of a single molecular entity (not the
    combination of two or more molecules into a co-crystal), it must be ignored.
 2. Mentions of co-crystal synthesis without method details.  If the article
    states that a co-crystal was synthesized but does not provide a method name or
    parameters, it must be ignored.
 3. Analysis methods, such as Powder X-ray diffraction (PXRD), Fourier-transform
    infrared spectroscopy (FTIR), Differential scanning calorimetry (DSC), Nuclear
    magnetic resonance (NMR), or Thermogravimetric analysis (TGA).
 4. Descriptions of molecular interactions, e.g., hydrogen bonds or halogen bonds,
    must be ignored unless they are directly linked to the synthesis method.
```

```
5. Formation of another polymorphic form: Formation of another polymorphic form of a
   crystal is not co-crystallization.

                                    [Response format]
```

B.4.2. Stage 2: Method classification

The Stage 2 prompt was used to identify the synthesis method types described in the extracted procedure text. This classification determines which method-specific parameter extraction prompt is applied at Stage 3. During prompting-strategy optimisation, we tested three variants of [Method context]:

- **S2-A:** The prompt included only the predefined list of synthesis method names.
- **S2-B:** The prompt included the predefined list of synthesis method names together with concise natural-language descriptions of the methods from Table 6.
- **S2-C:** The prompt included the predefined list of synthesis method names together with representative procedural templates from Appendix B.2.2.

The full Stage 2 prompt shown below contains the placeholder [Method context], which was instantiated according to the corresponding strategy.

```
You are a domain-specific chemical information extraction assistant. You specialize
in the chemistry of cocrystals and their synthesis. Your area of expertise includes
the analysis of types of co-crystal synthesis methods (not for a host molecule, not
for analysis, and not just molecular interactions).

Your task is to find the text of the synthesis technique for each co-crystal presented
in the article and identify the type of synthesis techniques that were used and
response should contain only the names of the methods from the List of techniques
in exact accordance with their spelling (no comments, no additional text).

List of techniques: grinding, hot-melt extrusion, solution crystallization,
antisolvent, slurry, freeze drying, spray drying, ultrasound, supercritical fluid,
electrospray, resonant acoustic, laser irradiation.

                                    [Method context]

Extraction rules:

1. Do not write the methodology again in the response if it already exists.

2. You must use technique names only from the List of techniques; do not name the
   techniques otherwise.

3. If the text is not in English, first translate it to English before analysis.

4. If the synthesis method does not correspond to any technique in the List of
   techniques, the answer must be strictly OTHER. Do not invent or suggest any
   additional technique name.

5. If several techniques were used, list them separated by commas.

6. It is very critical that you analyze each synthesis experiment separately to
   determine its type, without referring to other experiments in the paper.

7. Grinding includes both types of mechanochemical co-crystal synthesis: neat
   grinding and LAG. You do not need to specify the type of grinding; just write
   grinding.

8. Grinding is considered only if: (i) it is the primary synthesis method, e.g., the
   co-crystal is obtained by mechanical grinding; and (ii) it is not merely used for
   pre-mixing.
```

```
 9. The solution crystallization method considers three main processes:  evaporation,
    cooling, and interface formation.  If any of these processes are mentioned in the
    text, classify the method as solution crystallization.

10. If the text describes a multi-step process, select only the method that directly
    led to the formation of the co-crystal.  Preliminary steps are not taken into
    account.

11. If a method is explicitly named in the text, choose it, even if there are hints at
    other techniques.

12. A response cannot contain OTHER together with other methods.  If there is at least
    one matching method from the list, OTHER is not used.

13. Specify each type of technique only once.

Output examples:

    Output 1:  grinding

    Output 2:  hot-melt extrusion, solution crystallization
```

B.4.3. Stage 3: Method-specific parameter extraction

After method classification, parameters were extracted using a method-specific schema so that each prompt targeted only the fields relevant to the identified synthesis route. The example below shows the Stage 3 prompt used for grinding procedures.

```
You are a domain-specific chemical information extraction assistant.  You specialize
in the chemistry of cocrystals and their synthesis.  Your area of expertise includes
the analysis of co-crystal synthesis parameters.

Your task is to extract every mention of co-crystal synthesis parameters for grinding
synthesis methodology from a scientific article and output a response with formatting
as in the example (no comments, no additional text).

Fields for each object:

  - Type of Synthesis (string):  one of ``liquid-assisted grinding'', ``neat
    grinding'', or ``solvent-drop grinding''.  The key step in synthesis that leads
    to the formation of a cocrystal.

  - Formula (string):  name of co-crystal as cited in the text, e.g., ``CAR-HCT'',
    ``DMZ-SAC''.

  - API (string):  full-name of the active pharmaceutical ingredient, e.g.,
    ``caffeine'', ``betulin''.

  - Coformer (string):  full-name of used coformer, e.g., ``citric acid'', ``adipic
    acid''.

  - Part of Ratio API (numeric):  part of API in API-coformer ratio.

  - Part of Ratio Coformer (numeric):  part of Coformer in API-coformer ratio.

  - Amount of API (numeric):  mass or amount of API.

  - Amount of Coformer (numeric):  mass or amount of Coformer.

  - Amount Unit (string):  unit of measurement, e.g., ``mmol'', ``mg''.

  - Solvent (string):  full-name of used Solvent, e.g., ``water'', ``methanol''.

  - Amount of Solvent (numeric):  amount showing how much of the Solvent has been
    used.
```

- Solvent Amount Unit (string): Solvent unit of measurement, e.g., ``mL'', ``µL''.
- Second Solvent (string): full-name of used Second Solvent, e.g., ``ethanol'', ``dioxane''.
- Amount of Second Solvent (numeric): amount showing how much of the Second Solvent has been used.
- Third Solvent (string): full-name of used Third Solvent, e.g., ``methanol'', ``ethanol''.
- Amount of Third Solvent (numeric): amount showing how much of the Third Solvent has been used.
- Mixing Apparatus (string): Mixing Apparatus that has been used for grinding, e.g., ``ball mill'', ``mortar and pestle''.
- Mixing Frequency (numeric): frequency of the mixing process if that's what the device implies.
- Frequency Unit (string): Mixing Frequency unit of measurement, e.g., ``Hz'', ``rpm''.
- Mixing Time (numeric): duration of Mixing process.
- Mixing Time Unit (string): Mixing Time unit of measurement, e.g., ``minute''.
- Mixing Temperature (numeric): Temperature of Mixing.
- Mixing Temperature Unit (string): Mixing Temperature unit of measurement, e.g., ``°C''.
- Time of Drying (numeric): duration of the Drying process of the co-crystal after grinding.
- Drying Time Unit (string): Time of Drying unit of measurement, e.g., ``hour''.
- Temperature of Drying (numeric): Temperature of Drying process of the co-crystal after grinding.
- Drying Temperature Unit (string): Temperature of Drying unit of measurement, e.g., ``°C''.

Extraction rules:

1. It is crucial for you to accurately and comprehensively extract each synthesis experiment separately, without referring to other experiments in the article.
2. Do **not** filter, group, summarize, or deduplicate. Include repeated mentions and duplicates if they occur in different contexts.
3. Keep numerical values in the units reported in the article.
4. In the fields Part of Ratio API and Part of Ratio Coformer, write exactly the values reported in the article. Do not convert them to fractions of a unit.
5. If you cannot find a required field for an object, re-check the context; if it is still absent, set that field's value to ``NOT_DETECTED''.
6. For missing numeric fields, use the string ``NOT_DETECTED''.
7. When writing a parameter value, provide only one value per field.
8. The example below shows only two extracted samples; however, your output should contain **all** mentions of grinding synthesis methodology present in the article.

9. Depending on the type of synthesis, some parameters may be irrelevant. For ``neat grinding'', set solvent-related and drying-related fields to ``NOT_DETECTED'' if they are not reported, including Solvent, Amount of Solvent, Solvent Amount Unit, Second Solvent, Amount of Second Solvent, Third Solvent, Amount of Third Solvent, Time of Drying, Drying Time Unit, Temperature of Drying, and Drying Temperature Unit.

Output example:


```
[
    {
        "Type␣of␣Synthesis": "liquid-assisted␣grinding",
        "Formula": "ASP-CIT",
        "API": "aspirin",
        "Coformer": "citric␣acid",
        "Part␣of␣Ratio␣API": 4,
        "Part␣of␣Ratio␣Coformer": 1,
        "Amount␣of␣API": 1.53,
        "Amount␣of␣Coformer": 0.3,
        "Amount␣Unit": "g",
        "Solvent": "methanol",
        "Amount␣of␣Solvent": 0.1,
        "Solvent␣Amount␣Unit": "ml",
        "Second␣Solvent": "acetone",
        "Amount␣of␣Second␣Solvent": 0.4,
        "Third␣Solvent": "ethyl␣acetate",
        "Amount␣of␣Third␣Solvent": 0.5,
        "Mixing␣Apparatus": "planetary␣mill",
        "Mixing␣Frequency": 30,
        "Frequency␣Unit": "hz",
        "Mixing␣Time": 25,
        "Mixing␣Time␣Unit": "min",
        "Mixing␣Temperature": 80,
        "Mixing␣Temperature␣Unit": "C",
        "Time␣of␣Drying": 8,
        "Drying␣Time␣Unit": "h",
        "Temperature␣of␣Drying": 40,
        "Drying␣Temperature␣Unit": "C"
    },
    {
        "Type␣of␣Synthesis": "neat␣grinding",
        "Formula": "ACP-MAL",
        "API": "acetaminophen",
        "Coformer": "malic␣acid",
        "Part␣of␣Ratio␣API": 1,
        "Part␣of␣Ratio␣Coformer": 1,
        "Amount␣of␣API": 1.2,
        "Amount␣of␣Coformer": 0.4,
        "Amount␣Unit": "mg",
        "Solvent": "NOT_DETECTED",
        "Amount␣of␣Solvent": "NOT_DETECTED",
        "Solvent␣Amount␣Unit": "NOT_DETECTED",
        "Second␣Solvent": "NOT_DETECTED",
        "Amount␣of␣Second␣Solvent": "NOT_DETECTED",
        "Third␣Solvent": "NOT_DETECTED",
        "Amount␣of␣Third␣Solvent": "NOT_DETECTED",
        "Mixing␣Apparatus": "MM400,␣Retsch,␣Germany",
        "Mixing␣Frequency": 250,
        "Frequency␣Unit": "rpm",
        "Mixing␣Time": 1,
        "Mixing␣Time␣Unit": "h",
        "Mixing␣Temperature": 60,
        "Mixing␣Temperature␣Unit": "C",
```

```
        "Time_of_Drying": 12,
        "Drying_Time_Unit": "h",
        "Temperature_of_Drying": 40,
        "Drying_Temperature_Unit": "C"
    }
]

Final output constraints:  Generate the output as pure JSON only.
Do not include:

  - code block markers,

  - explanatory text,

  - any other non-JSON content.
```

B.5. Prompting Strategy Optimisation Results

Prompt variants and strategy descriptions are provided in Appendix B.4. We used `GPT-4o-mini` to develop prompts for all three extraction stages, evaluated on the validation datasets introduced in Section 3.3. Table 7 shows that, for Stage 1, both S1-A and S1-C achieved perfect recall but at a clear cost to precision. The binary formulation was prone to over-detection, while the additional pair-and-technique decomposition in S1-C introduced spurious extractions. S1-B provided the best balance, with the highest F1 and accuracy, and was therefore selected for Stage 1.

For Stage 2, performance improved as the method context became more procedural: names alone yielded the weakest performance, natural-language descriptions gave a modest improvement, and procedural templates achieved the best scores across all four metrics. We therefore used S2-C for Stage 2 in the final extraction pipeline. Per-method accuracy and F1 across all 12 method types are reported in Appendix B.6.

Table 7. Prompting-strategy selection for hierarchical extraction. `GPT-4o-mini` is evaluated on Stage 1 procedure-detection and Stage 2 method-classification validation sets. **Bold** indicates the best mean within each stage.

	Strategy	Precision	Recall	F1	Accuracy
Stage 1	S1-A	0.62 ±0.11	**1.00** ±0.00	0.76 ±0.08	0.62 ±0.11
	S1-B	**1.00** ±0.00	0.92 ±0.08	**0.96** ±0.04	**0.95** ±0.05
	S1-C	0.72 ±0.11	**1.00** ±0.00	0.84 ±0.07	0.76 ±0.09
Stage 2	S2-A	0.69 ±0.06	0.53 ±0.05	0.58 ±0.05	0.53 ±0.05
	S2-B	0.70 ±0.05	0.56 ±0.05	0.59 ±0.05	0.56 ±0.05
	S2-C	**0.75** ±0.05	**0.61** ±0.05	**0.64** ±0.05	**0.61** ±0.05

B.6. Per-method classification results

To complement the aggregate Stage 2 comparison, we also examined method-classification performance separately for each of the 12 synthesis types using a one-vs-rest evaluation setup. This analysis reveals substantial variation across methods, reflecting differences in how explicitly individual techniques are described and how strongly they overlap with other procedure types. In particular, methods with more distinctive procedural signatures are classified more reliably, whereas closely related or less frequently represented methods are more challenging. Figure 5 summarizes the per-method results in terms of F1-score and accuracy.

B.7. Per-parameter extraction results

Figure 6 reports per-parameter precision and recall at Stage 3 for all four evaluated models on the LAG validation set. Categorical and numerical parameters are shown separately. Each chart has one axis per parameter, with the outer boundary corresponding to a perfect score of 1.0.

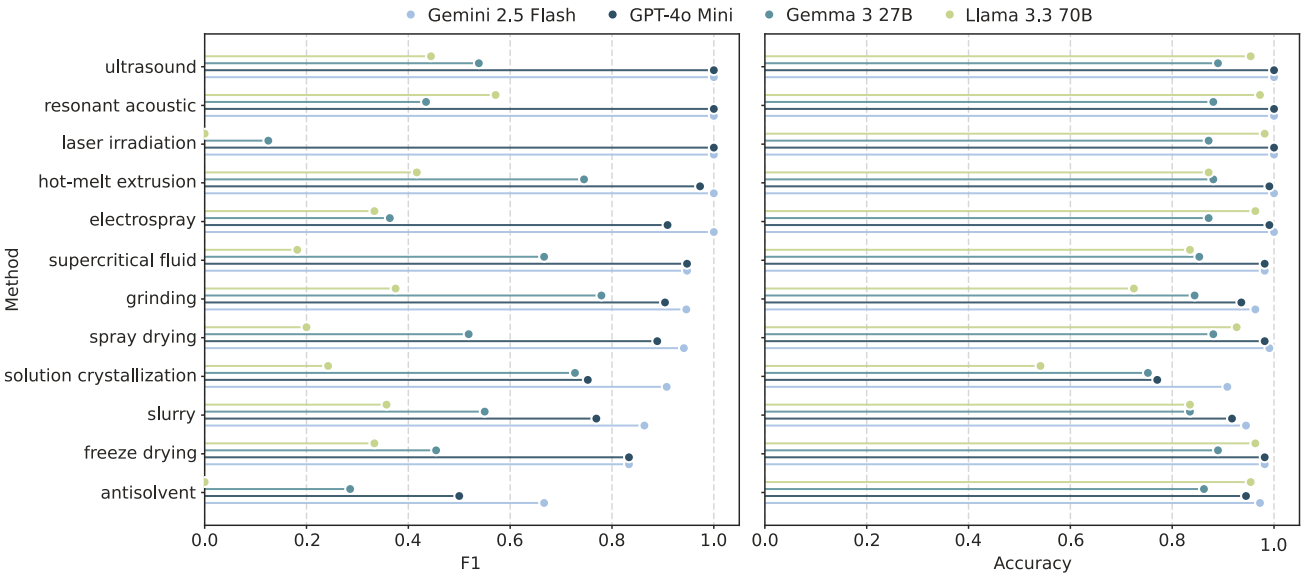

Figure 5. Per-method Stage 2 classification performance for the 12 co-crystal synthesis method types. Results are reported under a one-vs-rest evaluation setup using F1-score and accuracy across the evaluated LLMs.

B.8. Error analysis

We manually annotated the validation outputs of `Gemini-2.5-Flash` to identify dominant error types and their primary causes across the three extraction stages (Table 8). Categories are standardized across stages: Omission denotes information present in the article but missed by the model, Fabrication denotes unsupported model output, Substitution denotes extraction of the correct target with incorrect content, and Evaluation artifact denotes a mismatch caused by the scoring procedure rather than a substantive extraction error. Total counts are 5 errors at Stage 1, 20 at Stage 2, and 898 at Stage 3.

Table 8. Error analysis of `Gemini-2.5-Flash` extraction outputs. Dominant error types, stage-level frequencies (% stage), primary causes, and cause-level frequencies (% type) are reported across the three extraction stages.

	Error type	% stage	Primary cause	% type
Stage 1	Omission	60.0	The compound is described using alternative terminology, e.g., solvate or polymorph, rather than explicitly as a co-crystal.	66.7
	Fabrication	40.0	The article reports a crystal structure without an explicit synthesis procedure, yet the model returns a fabricated procedure text.	100.0
Stage 2	Omission	55.0	The method appears only in an additional, comparative, or control experiment rather than in the main synthesis section.	72.7
	Substitution	25.0	The case lies near the boundary between two method types, e.g., solution crystallization and slurry.	100.0
	Fabrication	20.0	The model infers a method from a brief textual mention or contextual cue, although that method was not actually used.	50.0
Stage 3	Evaluation artifact	67.6	Strict multiset comparison treats formatting and placeholder variants as mismatches even when the substantive value is correct.	68.4
	Fabrication	13.0	The model produces a parameter value that is not supported by the article text, often a procedural detail belonging to a different record.	81.2
	Omission	11.6	The model fails to extract a parameter value that is present in the article text.	67.3
	Substitution	7.8	The model extracts the correct field but with incorrect content, e.g., wrong solvent name or quantity.	80.0

Stage 1. Omissions (60%) are caused by non-standard terminology: the target material is described as a solvate, polymorph, or molecular complex rather than as a co-crystal, and the article is treated as out of scope. Fabrications (40%) recover procedure-shaped text from articles that report only a crystal structure.

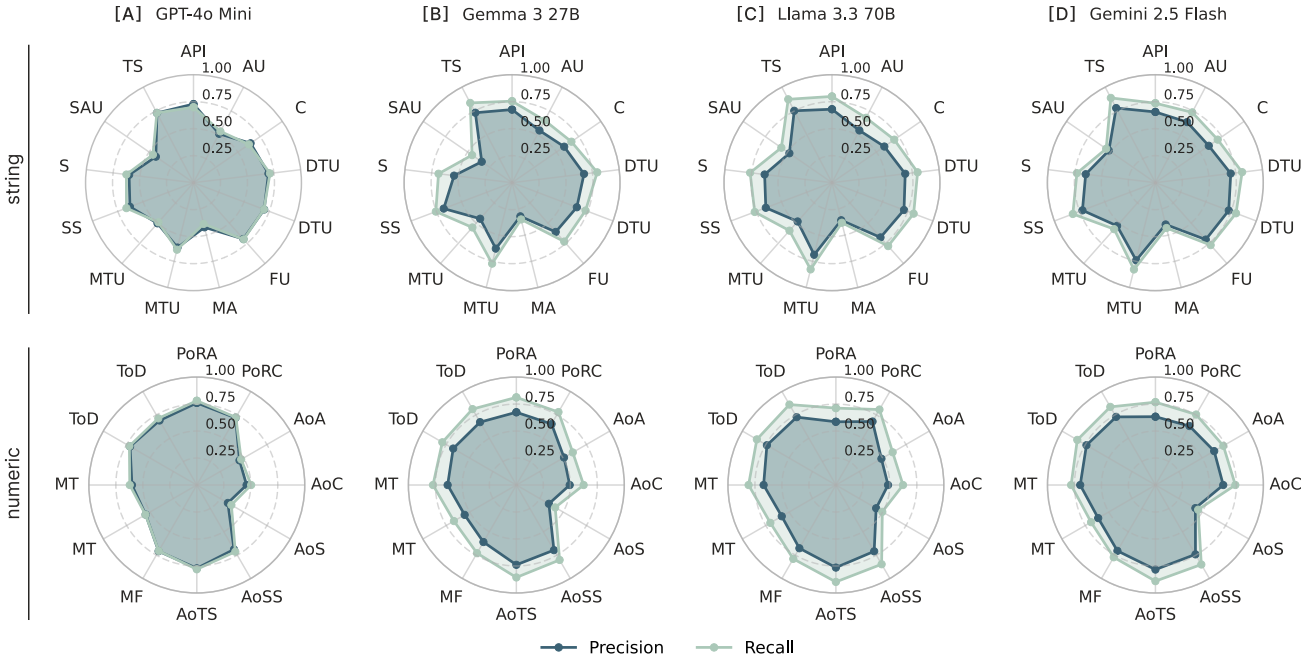

Figure 6. Per-parameter Stage 3 extraction performance on the LAG validation set. Precision and recall are shown separately for categorical and numerical parameters for [A] `GPT-4o-mini`, [B] `Gemma-3-27B`, [C] `Llama-3.3-70B`, and [D] `Gemini-2.5-Flash`.

Stage 2. Omissions (55%) are caused by methods reported only in additional, comparative, or control experiments rather than in the main synthesis section. Substitutions (25%) arise exclusively from boundary cases between closely related methods. Fabrications (20%) infer a method from a brief contextual mention rather than from an actual experimental description.

Stage 3. The majority of Stage 3 errors (67.6%) are evaluation artifacts from strict multiset-based comparison, where the substantive value is correct but formatting or placeholder variants are counted as mismatches. Among genuine errors, fabrications (13.0%) are the most informative: the model returns values that exist elsewhere in the article, often procedural details such as mixing time, frequency, or apparatus, but assigns them to the wrong record. Omissions (11.6%) are mostly complete misses of present fields, and substitutions (7.8%) involve the correct field with wrong content, most often solvent identity or quantity.

B.9. Parameter Families

For interpretability, individual parameters were grouped into eight families covering related physicochemical or procedural roles. Figure 2B reports aggregated statistics at the family level, while the per-parameter view is given in Figure 7. The full mapping is shown in Table 9.

B.10. Dataset Post-processing

This section details the deterministic post-processing steps used to convert extracted synthesis records into modeling-ready datasets. We describe the normalization pipeline, leakage-safe data partitioning, and imputation procedure used to generate prediction templates while preserving non-imputed test references for evaluation. Stoichiometric ratios are an exception: omitted or implausible ratios were treated during normalization as latent 1:1 values when no evidence for a different stoichiometry was present, reflecting the domain convention described below.

B.10.1. Normalization Pipeline

After extraction, records were normalized with deterministic method-specific pipelines before downstream modeling. The main steps were:

Table 9. Parameter-family definitions for coverage analysis. Extraction fields are grouped by their experimental role in the synthesis protocol.

Family	Count	Parameters
Ratio	2	Part of Ratio API / Part of Ratio Coformer
Amounts	5	Amount of API / Amount of Coformer / Amount of Solvent / Amount of Second and Third Solvent
Solvent	1	Solvent
Additional solvent	2	Second Solvent / Third Solvent
Time	4	Mixing Time / Drying Time / Evaporation Time / Time of Cooling or Evaporation
Temperature	3	Mixing Temperature / Drying Temperature / Evaporation Temperature
Frequency	1	Mixing Frequency
Setup	3	Mixing Apparatus / Covering Method / Description of Holes

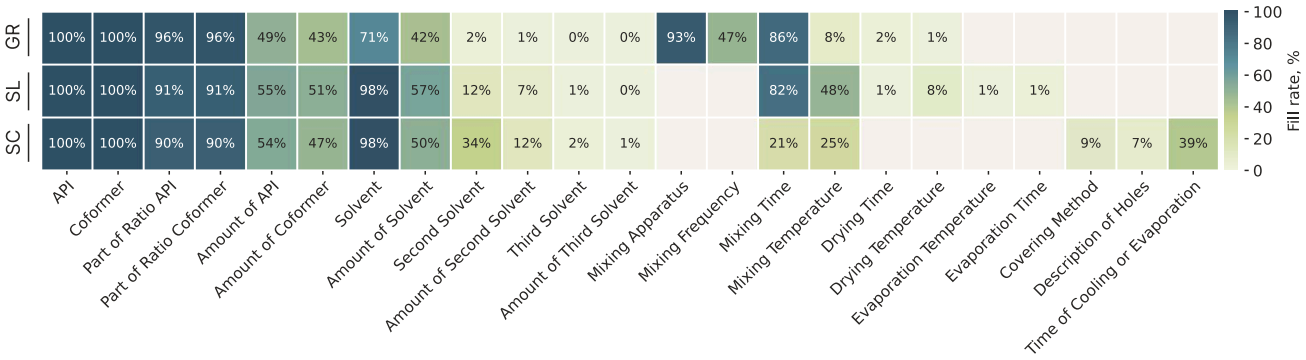

Figure 7. Per-parameter coverage after quality filtering. Fill rates are reported for individual synthesis parameters in the grinding, slurry, and solution crystallization datasets. Unit fields are omitted from the visualization. Structurally absent values are treated as empty, and blank cells denote parameters that are not defined for the corresponding synthesis method.

- **Schema harmonization.** Method-specific extraction fields were mapped to a common tabular schema for grinding, slurry crystallization, and solution crystallization.

- **Compound resolution and filtering.** Records were retained only when both API and coformer were present and resolvable to canonical SMILES. RDKit-based checks removed invalid molecules, identical API–coformer pairs, multi-fragment or charged species, molecules without carbon atoms, and molecules containing unsupported elements.

- **Numerical normalization.** Numerical parameters were parsed and converted to common units, including mg for compound amounts, mL for solvent volumes, min for times, °C for temperatures, Hz for frequencies, and W for powers. Stoichiometric ratios were reduced to integer API:coformer ratios. Missing or chemically implausible stoichiometric ratios were normalized to 1:1. This reflects a domain-informed assumption rather than a purely statistical imputation: 1:1 is by far the most common reported API/coformer stoichiometry in the corpus, and in many experimental procedures this ratio is omitted because equimolar mixing is treated as implicit. We therefore interpret such cases as latent 1:1 ratios unless the procedure provides evidence for a different stoichiometry. Physically implausible values and extreme numerical tails were removed.

- **Categorical normalization.** Solvent names and procedural categories were mapped to controlled vocabularies after text normalization. Method-specific consistency rules were applied, such as assigning no solvent to neat grinding and no mixing frequency to mortar-and-pestle grinding.

- **Missingness markers.** `NOT_DETECTED` denotes information expected but not extracted from the article, whereas `NO_VALUE` denotes structural absence of a parameter, such as no solvent in neat grinding.

- **Quality filtering and deduplication.** Records with fewer than four filled synthesis-parameter fields were removed, excluding identifiers, compound fields, synthesis type, formula, DOI, and unit columns from this count. Compatible duplicate rows were merged by retaining the most complete record.

After normalization and quality filtering, the retained datasets contained 8,272 solution-crystallisation records, 2,722 grinding records, and 593 slurry records.

B.10.2. Data Partitioning Details

Splits were constructed at the level of connected components rather than individual records. Two records were linked if they shared either the same DOI or the same canonical API–coformer pair, and entire components were assigned to a single split. This guarantees that no publication or molecular pair appears in both train and test sets. For each method, test components were selected greedily until reaching 100 records, followed by a repair step that adjusted assignments without breaking components.

Test components were chosen to maximise DOI diversity within the 100-record budget, since a more varied test set yields more representative metrics. As a side effect, the mean number of records per DOI is lower in test than in train (Table 10), which should be kept in mind when interpreting per-record metrics.

Table 10. Leakage-safe partitioning statistics for downstream prediction. For each method-specific dataset, train and test splits are reported by the number of DOIs, synthesis records, and records per DOI.

	Train			Test		
Dataset	**DOIs**	**Records**	**Rec./DOI**	**DOIs**	**Records**	**Rec./DOI**
Solution crystallization	2,814	8,172	2.90	52	100	1.92
Grinding	434	2,622	6.04	51	100	1.96
Slurry	77	493	6.40	45	100	2.22

B.10.3. Imputation Details

Imputation produced fully populated tables from which coherent textual procedure descriptions were generated as prediction templates, rather than ground-truth values. Two parallel artifacts were therefore maintained: imputed tables for template generation, and pre-imputation test reference snapshots for honest metric computation. All statistics (unit defaults, modes, group-wise modes, fallbacks) were estimated exclusively on the training split and applied independently to train and test. Values flagged as not detected and empty strings were treated as unknowns eligible for filling, whereas structurally absent values were preserved and never replaced by a statistical mode. Per-method deterministic rules assigned the structural absence marker to fields that should not exist in a given procedure.

Categorical fields. Filling proceeded as a cascade from most to least specific group, with the global training mode and a fixed default as final fallbacks. Groups were defined by chemical context: joint API and coformer pair, then API alone, then coformer alone.

Numerical fields. Domain rules were applied first, such as recovering a missing component mass when the other component mass, molecular weights, and stoichiometric ratio were available. Remaining gaps were filled by group-wise modes, with groups chosen per parameter family, for example:

- amounts: API, coformer, and ratio;
- solvent volume: chemical pair and solvent;
- mixing parameters: chemical pair and apparatus;
- temperature and time: chemical pair and solvent.

Global training modes were used as a final fallback. For columns populated in at most 1% of the training rows, a fixed default was assigned instead, preventing reconstruction from one or two incidental observations.

C. Synthesis Condition Prediction

C.1. Pair-level similarity

Few-shot examples were retrieved from the training set within the same synthesis method type. For a query co-crystal pair (a, b) and a candidate pair (c, d), where each element denotes one molecular component, pair-level similarity was computed symmetrically to respect the unordered nature of co-crystals:

$$S\big((a,b),(c,d)\big) = \max\big(\tfrac{1}{2}(T(a,c)+T(b,d)), \tfrac{1}{2}(T(a,d)+T(b,c))\big)\,,$$

where T is the Tanimoto similarity between Morgan fingerprints (radius 2, 2048 bits).

C.2. Prediction prompts

We used a unified prompt for synthesis parameter prediction across all experiments. In the few-shot setting ($k \in \{1, 3, 5, 10, 20\}$), the model receives retrieved examples of the same synthesis method type, a query API–coformer pair, and a synthesis template with missing parameters.

For zero-shot experiments, we used the same prompt but removed the `Examples` section and the corresponding instructions related to using examples. All other instructions and formatting constraints remained unchanged. The full few-shot prediction prompt is shown below.

```
You will be given numbered examples of co-crystal synthesis methods of the SAME method
type, a query API--coformer pair, and a method template.

Your task is to fill every bracketed placeholder in the template to produce one
best-supported synthesis method for the query pair.

Use the examples as strong evidence for structure, slot usage, value ranges, units,
formatting, and typical procedures.  The goal is to predict synthesis parameters
that are appropriate for this specific query API--coformer pair, not to copy one
example mechanically.  Similarity to an example does not guarantee that every value
in that example is suitable for the query pair.  Use general chemistry and synthesis
knowledge only when the examples are sparse, inconsistent, incomplete, clearly not
transferable for a specific slot, or when it is needed to reject an impossible or
clearly unsuitable value.  Do not use general chemistry to override a clear local
pattern from the most chemically and procedurally similar examples, especially for
exact numeric values.

Rules:

  - Work slot by slot, not example by example.

  - Prefer values supported by the most chemically and procedurally similar examples
    for that specific slot.

  - Prefer the nearest local pattern, but do not copy one whole example blindly.

  - Do not average across all examples when a smaller set of closer examples gives a
    clearer signal.

  - Treat categorical value choice, numeric value choice, NO_VALUE choice, and
    formatting choice as separate decisions.

  - Replace [API] with the query API SMILES exactly.

  - Replace [Coformer] with the query coformer SMILES exactly.

  - Replace every other [placeholder] with exactly one concrete value or NO_VALUE, and
    leave no placeholder unreplaced.

  - Use NO_VALUE only when the field is truly not applicable, not when uncertain or
    simply unsupported by some examples.

  - Presence of an optional field requires positive support.  Do not add a solvent,
    second solvent, additive, apparatus, mixing frequency, temperature, time, or
    amount just because such fields appear often overall.
```

```
  - If a quantity is NO_VALUE, its unit must also be NO_VALUE.

  - If a solvent, second solvent, additive, or apparatus is NO_VALUE, directly
    dependent amount and unit fields must also be NO_VALUE unless the template clearly
    separates them.

  - Keep dependent fields mutually consistent.

Rules for sensitive numeric slots:

  - Choose ratio slots from ratio evidence, not from mass evidence.  Do not derive
    [Part of Ratio API] or [Part of Ratio Coformer] from [Amount of API] and [Amount
    of Coformer].

  - Choose amount slots from amount patterns, not from ratio slots.  Do not force
    masses to numerically match stoichiometric parts.

  - Keep numbers as plain numbers, units only in unit placeholders, and ratio parts as
    separate numbers unless the template explicitly requires another format.

  - Follow the local numeric scale, precision, and granularity of the nearest relevant
    examples.

  - Do not drift toward convenient anchor values such as round or highly frequent
    numbers unless they are strongly supported by the nearest examples.

  - Keep time, temperature, frequency, amount, and ratio fields separate.  Do not
    substitute one for another or infer one from another unless the examples clearly
    establish that relation.

  - Do not invent unusual or highly specific numeric values without strong local
    support.

Rules for solvent-like and optional slots:

  - Solvent identity is highly local.  Do not choose a solvent because it is common
    overall or chemically plausible in general.

  - Choose the solvent best supported by the nearest relevant examples for this query
    pair.

  - Second solvent or anti-solvent fields require positive local evidence of a real
    two-solvent pattern.  Do not default them to a common solvent, and do not default
    them to NO_VALUE when the nearest relevant examples show a real second-solvent
    pattern.

  - For optional solvent in grinding-like procedures, use a solvent only when the
    nearest examples support liquid assistance or there is a clear reason the query
    pair likely requires it; otherwise use NO_VALUE and keep dependent solvent
    amount/unit fields consistent.

  - For mixing frequency, apparatus, and similar optional operation fields, prefer
    NO_VALUE unless the nearest examples give positive support that the field is
    genuinely used for this kind of query/template.

Method-type-specific priorities:

  - If method type is grinding:  be conservative about adding solvent and mixing
    frequency.  Do not infer liquid-assisted grinding unless the nearest grinding
    examples support it or there is a clear reason the query pair likely needs it.

  - If method type is slurry:  do not infer stoichiometric ratio from mg
    values.  Treat solvent identity, mixing time, and mixing temperature as local
    slurry-pattern choices.  Do not collapse to a globally common solvent.
```

```
  - If method type is solution_crystallization:  solvent and second solvent are
    highly local and should usually be chosen from the nearest solution examples,
    not from global defaults such as common alcohols.  Distinguish mixing time from
    cooling/evaporation time; do not substitute one for the other.

Formatting rules:

  - Use example unit and style conventions whenever possible.

  - If a slot behaves like a closed or near-closed vocabulary in the examples, use the
    canonical spelling/style used in the examples.

  - Keep the template text otherwise identical:  same wording, order, and punctuation.
    Only replace bracketed placeholders.

  - Every placeholder must be filled exactly once.

  - Output exactly one paragraph and nothing except:

    [BEGIN_METHOD]

    <filled template>

    [END_METHOD]

Before finalizing internally, verify:

  - The method is intended for the specific query API--coformer pair, not just for a
    similar example pair.

  - [API] and [Coformer] were copied exactly from the query.

  - Every placeholder was replaced exactly once.

  - NO_VALUE is used only for true non-applicability.

  - Dependent fields are consistent with NO_VALUE.

  - Ratios were chosen from ratio evidence, not from masses.

  - Solvent and second-solvent choices follow the nearest relevant local examples
    rather than global common values.

  - Numeric values stay anchored to local example scale rather than chemistry-driven
    invention.

Method type:  [Method type]

Examples:  [Examples]

Query:
API SMILES: [API SMILES]
Coformer SMILES: [Coformer SMILES]

Template to fill:  [Template]
```

C.3. In-Context Example Ablation

We evaluated the effect of the number of retrieved in-context examples using `GPT-4o-mini`. The largest gain across all three methods came from moving from zero-shot prediction to k=3. Beyond this point, categorical scores improved only marginally, while numerical scores peaked at k=3 and then declined, likely because additional examples diluted the signal from the closest precedent. Since numerical fields dominate the schema, we used k=3 in all subsequent few-shot prediction experiments. The full ablation curves are shown in Figure 8.

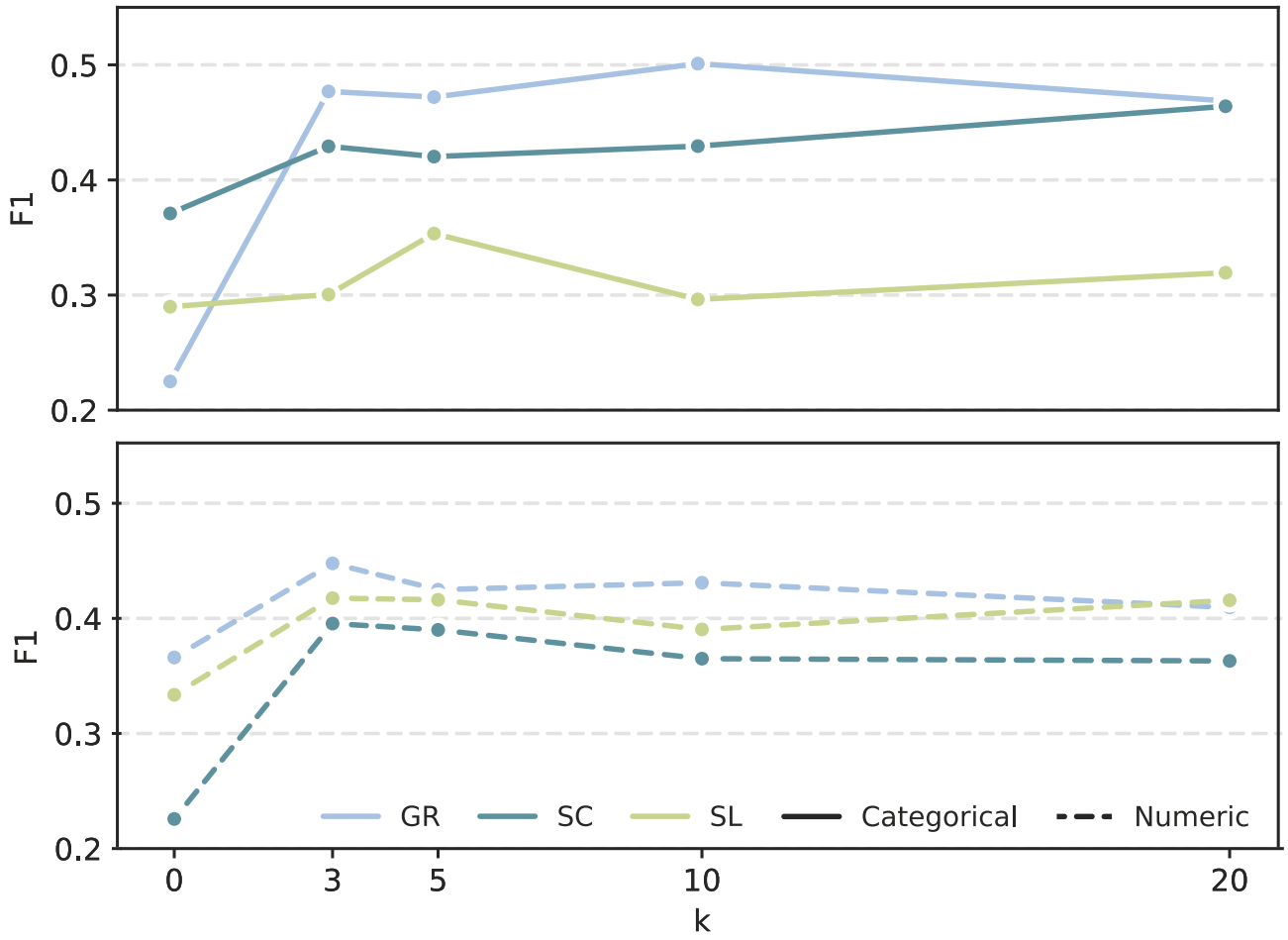

Figure 8. Effect of the number of retrieved in-context examples on synthesis-condition prediction. Weighted F1 Score is reported for `GPT-4o-mini` as k varies across grinding, slurry, and solution crystallization, with categorical and numerical aggregates shown separately.

C.4. Solvent Families

A curated synonym dictionary normalizes raw solvent labels to 475 canonical names. Since exact-match accuracy over 475 classes is dominated by long-tail confusions between solvents of similar chemical nature, we additionally group canonical solvents into 13 chemical families. Table 11 lists the families with their descriptions and the number of canonical solvents covered.

C.5. Numerical tolerance bins

For numerical-parameter evaluation, we additionally used tolerance bins to count near-correct predictions as correct when they fall into the same experimentally meaningful range. The bin definitions were generated from the imputed training data only, using the train split to avoid using test-set information.

For each method, bins were fitted for the numerical columns when the corresponding column was present in the method table. Ratio parameters were treated globally by pooling values across all methods. If a parameter had at most n_{bins} unique values, bins were created from sorted unique values using midpoint boundaries. Otherwise, bins were created using quantile binning and duplicate boundaries removed. All bin boundaries were rounded to three significant figures.

During tolerance-based scoring, a value is assigned to a bin if it satisfies $lo \leq x < hi$; the final bin is open-ended above. A prediction is counted as tolerance-correct when the gold and predicted values fall into the same bin.

Table 11. Solvent families used for categorical aggregation, sorted by the number of canonical solvent names covered.

Family	Num.	Description
Alcohols	97	Protic alcohols, polyols, and phenols
Amines	80	Amines and N-heteroaromatics, basic and coordinating
Hydrocarbons	58	Aliphatic and aromatic hydrocarbons, low polarity
Esters	50	Esters, carbonates, and lactones
Halogenated	48	Halogenated aliphatics and aromatics, polarizable
Carbonyls	44	Ketones and aldehydes, polar aprotic
Ethers	30	Ethers, cyclic ethers, and glymes
Acids	25	Acidic and protic media
Amides	17	Amides, ureas, and lactams, strong solvators
Special	14	Oxidants, gases, and reactive or unusual cases
Nitriles	8	Nitriles, polar aprotic
Sulfur	3	Sulfur-containing solvents such as DMSO and sulfolane
Water	1	Aqueous media, very polar with strong H-bonding

C.6. Solvent Confusion Analysis

Figure 9 provides a more detailed view of solvent-prediction errors. The exact-solvent confusion matrix shows that errors are concentrated among high-frequency solvents and the long-tail `OTHER` class, reflecting the high cardinality and local nature of solvent choice.

The family-level confusion matrix indicates that many mistakes remain chemically structured: predictions often collapse toward broad, frequently observed families, especially alcohols, rather than arbitrary solvent classes. These results explain why solvent prediction remains challenging despite the use of retrieved few-shot precedents, and support reporting both exact-solvent and solvent-family metrics.

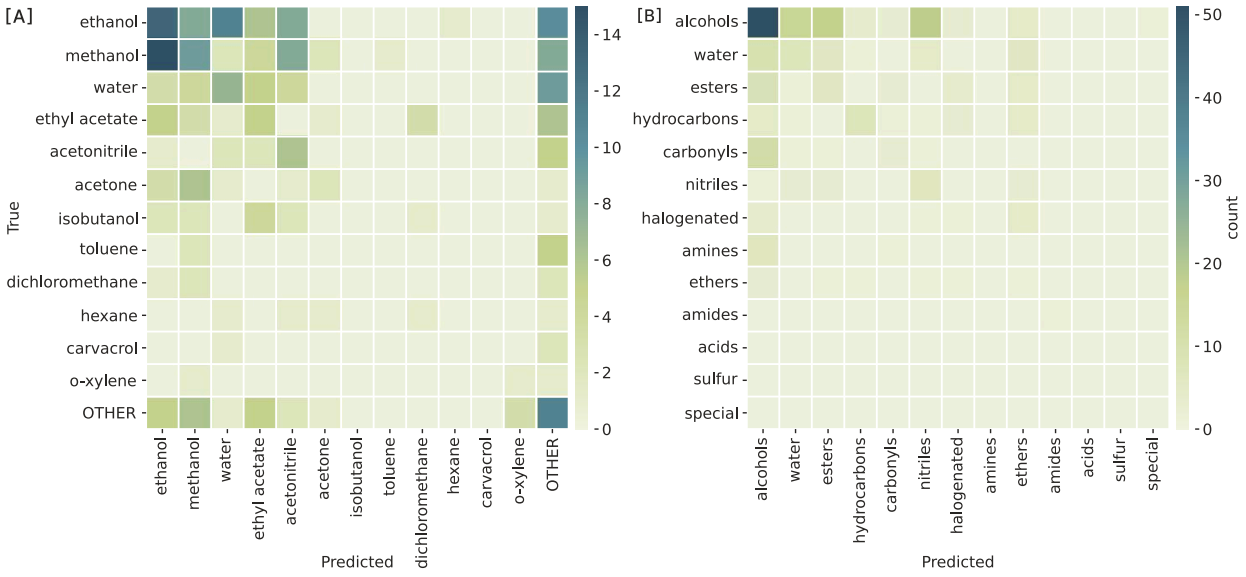

Figure 9. Solvent confusion analysis. [A] Exact-label confusion matrix for the 12 most frequent solvents, with all remaining labels grouped as `OTHER`. [B] Confusion matrix after aggregation into solvent families.

C.7. Per-Method Prediction Results

Tables 12, 13, and 14 report per-parameter Accuracy and weighted F1 for `Gemini-2.5-Flash` (k=3) across two independent runs. Reported values are mean ± standard deviation. Fill Rate denotes the proportion of test records containing a valid ground-truth value for a given parameter.

Overall, the results show that the model is consistently more reliable for predicting stoichiometric ratios (Part of Ratio

API/Coformer), where performance remains high across all synthesis methods (F1 ¿ 0.77). In contrast, absolute quantity parameters (e.g., Amount of API, Amount of Solvent) remain difficult, typically yielding F1 ¡ 0.30, likely because these values depend on experiment-specific conditions and are less standardized across publications.

Categorical solvent prediction remains particularly challenging for Slurry and Solution Crystallization. One likely reason is that the model does not have explicit knowledge of the full solvent space and tends to over-rely on solvents observed in the retrieved in-context examples. As a result, predictions become biased toward a narrow subset of frequently occurring solvents, reducing generalization when the correct solvent category is rare or absent from the provided examples.

The low standard deviations across runs indicate that the model behavior is stable, suggesting that these limitations are systematic rather than caused by sampling variability.

Table 12. Per-parameter grinding prediction results for `Gemini-2.5-Flash`. Fill rate, weighted F1, and accuracy are reported for each evaluated categorical and numerical parameter using $k = 3$ retrieved examples.

Type	Parameter	Fill %	F1	Accuracy
Categorical	Solvent	95	0.34 ±0.00	0.35 ±0.00
	Mixing Apparatus	89	0.57 ±0.00	0.57 ±0.00
Numerical	Part of Ratio API	100	0.84 ±0.00	0.82 ±0.01
	Part of Ratio Coformer	100	0.86 ±0.00	0.90 ±0.00
	Amount of API	47	0.20 ±0.00	0.21 ±0.00
	Amount of Coformer	48	0.29 ±0.00	0.27 ±0.00
	Amount of Solvent	63	0.42 ±0.01	0.39 ±0.01
	Mixing Frequency	71	0.46 ±0.01	0.42 ±0.01
	Mixing Time	88	0.31 ±0.00	0.34 ±0.00
Categorical avg.		92.0	0.46 ±0.00	0.46 ±0.01
Numerical avg.		73.9	0.48 ±0.01	0.48 ±0.01
Overall		78.0	**0.48** ±0.01	**0.47** ±0.01

Table 13. Per-parameter solution crystallization prediction results for `Gemini-2.5-Flash`. Fill rate, weighted F1, and accuracy are reported for each evaluated categorical and numerical parameter using $k = 3$ retrieved examples.

Type	Parameter	Fill %	F1	Accuracy
Categorical	Solvent	98	0.17 ±0.01	0.17 ±0.01
	Second Solvent	100	0.53 ±0.00	0.57 ±0.01
Numerical	Part of Ratio API	100	0.82 ±0.00	0.84 ±0.01
	Part of Ratio Coformer	100	0.77 ±0.00	0.76 ±0.01
	Amount of API	61	0.12 ±0.00	0.11 ±0.00
	Amount of Coformer	61	0.19 ±0.01	0.17 ±0.01
	Amount of Solvent	57	0.20 ±0.00	0.21 ±0.00
	Time of Cooling or Evaporation	44	0.25 ±0.01	0.24 ±0.02
	Mixing Temperature	32	0.46 ±0.00	0.47 ±0.00
	Mixing Time	27	0.22 ±0.00	0.26 ±0.00
Categorical avg.		99.0	0.35 ±0.01	0.37 ±0.01
Numerical avg.		60.3	0.38 ±0.01	0.38 ±0.01
Overall		68.0	**0.37** ±0.01	**0.38** ±0.01

D. Agentic System Details

D.1. Domain Tool Groups

The agentic system uses four tool groups, which are shared by the Single-Agent architecture and distributed across specialists in the Multi-Agent architecture.

- **Common Tools** normalize chemical inputs, resolve molecule names, canonicalize structures, identify molecular analogues, and convert structured synthesis parameters into method text.

Table 14. Per-parameter slurry prediction results for `Gemini-2.5-Flash`. Fill rate, weighted F1, and accuracy are reported for each evaluated categorical and numerical parameter using $k = 3$ retrieved examples.

Type	Parameter	Fill %	F1	Accuracy
Categorical	Solvent	100	0.15 ±0.00	0.16 ±0.00
Numerical	Part of Ratio API	100	0.80 ±0.00	0.84 ±0.01
	Part of Ratio Coformer	100	0.88 ±0.00	0.85 ±0.00
	Amount of API	58	0.22 ±0.00	0.21 ±0.00
	Amount of Coformer	59	0.19 ±0.00	0.24 ±0.00
	Amount of Solvent	59	0.16 ±0.00	0.19 ±0.00
	Mixing Time	83	0.25 ±0.01	0.27 ±0.02
	Mixing Temperature	34	0.48 ±0.01	0.41 ±0.00
Categorical avg.		100.0	0.15 ±0.00	0.16 ±0.00
Numerical avg.		70.4	0.43 ±0.01	0.43 ±0.01
Overall		74.1	**0.39** ±0.01	**0.39** ±0.01

- **Database Tools** query CoSyn-DB for experimental precedents, exact or similar API–coformer pairs, parameter-constrained records, component ratios, and common synthesis conditions.

- **Article-analysis Tools** load scientific articles from DOI, PDF, or TXT inputs, detect whether a synthesis procedure is present, classify the synthesis method, and extract method-specific parameters.

- **Prediction Tools** generate candidate synthesis procedures for new molecular pairs, select related examples, extract parameters from generated text, and validate explicit user constraints.

D.2. Multi-Agent System Implementation

The agentic CoSyn systems were implemented as LangGraph/LangChain (MIT) (Chase, 2022) workflows with tool-calling through an OpenAI-compatible Chat API. In the Single-Agent setting, one LLM controller has access to the full CoSyn tool set and iteratively decides whether to call a tool or produce a final answer.

In the Multi-Agent setting, an orchestrator routes the user request to specialist agents with role-specific tool access. The article agent handles DOI/PDF processing and synthesis extraction, the database agent queries and aggregates records from CoSyn-DB, and the prediction agent retrieves few-shot examples and generates synthesis protocols. Tool outputs are stored as structured artifacts in the graph state, including article-analysis results, database hits, retrieved examples, prediction outputs, and constraint checks.

All architectures use the same underlying domain tools for molecule normalization, database search, literature extraction, few-shot retrieval, protocol generation, and parameter validation. To keep execution bounded and comparable across systems, each run uses fixed limits on LLM calls, tool calls, specialist handoffs, and graph recursion depth.

D.3. Agentic System Prompts

This section summarizes the system prompts used for the non-agentic baselines, the Single-Agent system, its ablated variants, and the Multi-Agent system. To avoid repeating identical grounding and abstention rules across prompts, we report the prompts as compact templates with explicitly marked variable parts. The prompts differ primarily in the evidence available to the model, the allowed tool groups, and the routing structure.

D.3.1. Baseline Prompts

We evaluated two non-agentic baselines. The LLM-Only baseline receives only the user question and must answer from the model's internal knowledge. The Context-Augmented LLM additionally receives task-relevant static context, such as article text or database-derived evidence, but it cannot iteratively call tools or decompose the task through tool use. The corresponding prompt-field instantiations are summarized in Table 15.

```
You are [Baseline role] helping with questions about co-crystal synthesis.
Your job is to answer the user's request using [Available evidence].
Core behavior:
  - Determine which parts of the request can be answered from the available evidence.
  - Answer as fully as possible while clearly marking uncertainty.
  - Do not claim that an article was analyzed, a database was searched, or a
    prediction was validated unless such evidence is explicitly available.
  - Clearly distinguish evidence-based conclusions from general-knowledge reasoning.
Evidence policy:
  - [Evidence-use instruction]
  - If evidence is insufficient for part of the request, state the limitation and
    provide a cautious best-effort answer when appropriate.
  - Do not invent article-specific facts, database-wide statistics, retrieved
    examples, or validated synthesis predictions.
Final answer policy:
  - Provide a concise user-facing answer.
  - For yes/no questions, begin the relevant answer section with an explicit yes or no
    when possible.
  - If the answer is not evidence-backed, make this clear in the wording.
```

Table 15. Non-agentic baseline prompt configurations, contrasting evidence availability and grounding constraints between the LLM-only and context-augmented settings.

Prompt field	**LLM-Only**	**Context-Augmented LLM**
Baseline role	An expert assistant for co-crystal synthesis.	A context-grounded assistant for co-crystal synthesis.
Available evidence	Only the user question and the model's own chemistry knowledge.	The user question and static task-relevant context provided with the query.
Evidence-use instruction	Answer from general chemistry knowledge, but do not present the answer as article- or database-supported.	Reason directly from the provided context, but do not infer beyond what the context supports.

D.3.2. Single-Agent Prompt

The Single-Agent system uses one controller to decompose the request, select tools, and produce the final answer. It has access to the four domain tool groups defined in Appendix D.1. The detailed tool-selection rules are omitted for brevity and represented as a placeholder in the template below.

```
You are a single-agent assistant helping with questions about co-crystal synthesis.
Your job is to answer the user's request by using the available tools and returning a
grounded final answer.  You may call tools in any order and as many times as needed,
within the step limit.
Core behavior:
  - First determine which subtasks are required by the request.
  - Classify each subtask as article analysis, database retrieval, synthesis
```

```
    prediction, or background answering.

  - Keep track of which subtasks are completed, pending, unsupported, or abstained.

  - Do not finalize until every required tool-backed subtask has been completed or
    explicitly marked impossible.

  - If the request combines multiple subtasks, resolve each one separately and then
    merge the results into a single final answer.

  - Avoid redundant tool calls unless new information, a refined constraint, or a
    failed prior attempt justifies them.

                                    [Tool policy]
Grounding policy:

  - Use tool outputs as the primary evidence source.

  - Report article facts, database examples, retrieved conditions, and
    prediction-derived details only when they are supported by tool outputs.

  - Do not invent article contents, database matches, retrieved examples, or validated
    predictions.

  - Clearly distinguish tool-grounded conclusions from general-knowledge reasoning.

Failure and abstention behavior:

  - If input is insufficient for a tool, ask for the missing field when appropriate or
    abstain for that subtask.

  - If one part of a multi-part request fails, continue solving the remaining parts
    whenever possible.

  - Do not turn a partial failure into a full failure if other required subtasks can
    still be completed.

  - Clearly mark unsupported or abstained parts in the final response.

Final answer policy:

  - Provide a concise user-facing answer.

  - For yes/no questions, begin the relevant answer section with an explicit yes or
    no.

  - If the user asked for a synthesis method, procedure, or protocol, present it as
    readable method text rather than only as raw parameters unless structured fields
    were explicitly requested.

  - Keep the final answer grounded, complete, and aligned with the resolved subtasks.
```

D.3.3. Ablation Prompts

The ablation prompts use the same Single-Agent controller and grounding rules, but remove one tool group at a time. The removed tool group is replaced with the same kind of static context used in the Context-Augmented LLM baseline when such context is available. The resulting prompt-level changes are summarized in Table 16.

D.3.4. Orchestrator Prompt

The orchestrator controls the hierarchical Multi-Agent workflow. It does not call domain tools directly. Instead, it decomposes the user request, selects the appropriate specialist, and writes the final answer after the required specialist reports are available.

Table 16. Single-agent ablation configurations, with one CoSyn capability disabled in each setting and replaced by comparable static context when available.

System	Removed capability	Prompt-level change
SA w/o Retrieval	Database tools	The agent cannot query CoSyn-DB or make database-wide claims. It can use static context when available.
SA w/o Prediction	Prediction tools	The agent cannot use the developed prediction module for synthesis-procedure generation or validation. It can still suggest procedures from retrieved evidence or general reasoning when supported.
SA w/o Analysis	Article-analysis tools	The agent cannot use the developed article-analysis pipeline. It can still reason from static article context when available.

The coordination guardrails require the orchestrator to preserve the user's original constraints, treat multiple constraints as conjunctive, and use only grounded specialist outputs in the final answer. Failed, partial, empty, or unsupported specialist results must not be presented as completed evidence. If a requested item cannot be fully supported, the answer must state the unsupported part rather than fill it with non-matching results.

```
You are the orchestrator for a hierarchical assistant that helps with questions about
co-crystal synthesis.
You do not call domain tools yourself.  Your job is to decompose the user request,
choose the next specialist, and produce the final answer after specialists have
supplied enough evidence.
Available specialists:  article_agent, db_agent, prediction_agent.
Routing policy:
  - Use article_agent for questions about a specific article, PDF, TXT file, DOI,
    extracted synthesis parameters, method presence, method type, or article evidence.
  - Use db_agent for questions asking to find database examples, database conditions,
    common solvents/apparatus/times, exact API--coformer pairs, ratios, or similar
    precedents.
  - Use prediction_agent for questions asking to predict, propose, generate, or
    validate a new synthesis method.
  - If multiple capabilities are needed, call one specialist at a time and then route
    to the next missing specialist.
  - Do not call a specialist again unless the previous output was insufficient,
    failed, or a refined follow-up is needed.
  - Finalize only when every required part is answered or explicitly
    impossible/abstained.
                                [Coordination guardrails]
Return only valid JSON with this schema:
{
  "next_agent": "article_agent" | "db_agent" | "prediction_agent" | "final",
  "specialist_task": "short␣concrete␣instruction␣for␣the␣selected␣specialist,␣or␣empty␣string␣for␣fina
  "final_answer": "final␣user-facing␣answer␣if␣next_agent␣is␣final,␣otherwise␣empty␣string",
  "reason": "short␣routing␣reason"
}
```

D.3.5. Shared Specialist Prompt Skeleton

The Multi-Agent system uses three specialist roles: `article_agent`, `db_agent`, and `prediction_agent`. Their specialist-specific policies follow the tool grouping defined in Section D.1.

The Multi-Agent system uses three specialist roles: article_agent, db_agent, and prediction_agent. Their specialist-specific policies are derived from the logical tool grouping defined in Section D.1.

[Specialist-specific policy]

Constraint policy:

- Preserve the original user request and all constraints from the orchestrator task.
- Use only tools from your assigned capability group, as defined in Section D.1.
- Follow the specialist-specific workflow for the assigned subtask.
- Do not solve unrelated subtasks assigned to other specialists.
- If the assigned subtask cannot be completed with the available tools, explicitly mark it as unsupported or abstained.

Return a concise report containing:

- completed steps;
- extracted, retrieved, or generated facts relevant to the user request;
- evidence used, such as source path, DOI, database filters, retrieved examples, generated method text, or validation result;
- unsupported or abstained parts, if any.

Assigned specialist subtask:

[Assigned subtask]

D.4. CoSyn-Bench Examples

Each CoSyn-Bench item consists of a user question, gold answer, required capabilities, and scoring metadata. The first example illustrates a single-capability retrieval task, where the answer can be obtained from CoSyn-DB alone.


```
{
  "type": "single_capability",
  "capabilities": ["Retrieval"],
  "question": "Which five solvents are most commonly used in the synthesis of co-crystals by the soluti
  "gold_answer": "The five most commonly used solvents in solution crystallization are methanol, ethanc
  "metadata": {
    "input_artifacts": {
      "method_family": "solution crystallization",
      "target": "most common solvents",
      "k": 5
    },
    "hard_constraints": [
      "return exactly five solvents",
      "count only solution-crystallization records",
      "rank solvents by frequency"
    ],
    "required_items": [
      "five most common solvents used in solution crystallization"
    ]
  }
}
```


The second example combines prediction and retrieval. The system must first generate a grinding-based synthesis method and then compare the predicted apparatus with the most common apparatus for grinding in CoSyn-DB.

```
{
  "type": "multi_capability",
  "capabilities": ["Prediction", "Retrieval"],
  "question": "Predict the method for synthesising a co-crystal by grinding, consisting of 2-ethoxybenz
  "gold_answer": "Yes, the predicted method uses the most common mixing apparatus for grinding. The pre
  "metadata": {
    "input_artifacts": {
      "molecular_pair": [
        "2-ethoxybenzamide",
        "2,5-dihydroxybenzoic acid"
      ],
      "method_family": "grinding",
      "db_stat": "most common mixing apparatus for grinding"
    },
    "hard_constraints": [
      "Predict a grinding-based method.",
      "Determine the most common mixing apparatus for grinding.",
      "State whether the predicted method uses that apparatus."
    ],
    "required_items": [
      "Propose the target synthesis procedure.",
      "Identify the mixing apparatus used in the prediction.",
      "Determine the most common apparatus for grinding.",
      "Compare them and answer yes or no."
    ]
  }
}
```


D.5. LLM-as-a-Judge Prompt

We used an LLM-as-a-judge evaluator to score candidate answers against the user question, the gold reference answer, benchmark metadata, and execution context. The judge was instructed to prioritize the gold reference and tool-derived execution context over its own background knowledge, and to penalize unsupported article, database, or prediction claims.

```
You are an expert evaluator for a benchmark on co-crystal synthesis agents.

Evaluate the candidate answer against the user question and the gold reference.
Use the benchmark input artifacts to understand the intended task setup.  Use the
execution input context as the concise record of evidence and tool-derived context
available to the agent.

When scoring groundedness, check whether the candidate answer is supported by
the execution input context and does not invent unsupported article, database,
or prediction details.  Prefer the gold reference and the explicitly provided
requirements over your own background knowledge.

You must score the answer on five criteria:

 1. Task completion.  Measures whether the candidate answer fully addresses the user
    request, including all required subtasks, examples, comparisons, extractions,
    predictions, or checks.

 2. Correctness.  Measures whether the answer is factually and semantically consistent
    with the gold reference and available context.  Minor wording differences are
    acceptable if the meaning is correct.  For prediction tasks, the gold answer
    should be treated as a reference solution rather than the only valid solution.

 3. Constraint adherence.  Measures whether the answer follows explicit requirements
    in the question, such as synthesis method type, number of examples, solvent,
    temperature range, apparatus, ratio, API, coformer, yes/no decision, or method
    family.

 4. Groundedness.  Measures whether the answer is supported by the execution input
    context, including article evidence, database evidence, retrieved examples,
```

```
    extracted parameters, or prediction context.  Unsupported details and hallucinated
    evidence should be penalized.

 5. Answer quality.  Measures whether the answer is clear, well-structured, readable,
    concise, complete, and directly usable.

Special-case scoring rules:

  - If a required PDF or article could not be downloaded or accessed, do not penalize
    the candidate for failing to extract information from that unavailable source.
    A clear abstention or limitation statement is acceptable, but invented article
    contents must still be penalized.

  - For prediction tasks, treat the gold answer as a reference solution.  Chemically
    plausible and partially correct predictions should receive meaningful partial
    credit even when they do not match every parameter.

  - Placeholders such as NO_VALUE or NOT_DETECTED are acceptable for missing,
    unavailable, or unsupported parameters.  Penalize them only when the evidence
    clearly supports a concrete value.

  - Additional synthesis parameters beyond the gold reference are acceptable if they
    do not contradict the reference or execution context.

  - For yes/no questions, reward answers that state the decision explicitly.  If the
    answer is substantively correct but indirect, reduce task completion or answer
    quality slightly.

Scoring rubric:

  - 5 = fully satisfies the criterion.

  - 4 = mostly satisfies it, with only minor issues.

  - 3 = partially satisfies it, with meaningful correct content but important
    omissions, mismatches, or mistakes.

  - 2 = limited correctness or coverage, but still contains some non-trivial correct
    content.

  - 1 = completely fails the criterion, is almost entirely incorrect, or is unusable.

Assign only integer scores from 1 to 5 for every criterion.  Do not use fractional
values.  Do not assign 0 under any circumstances.

Return only valid JSON with this schema:

{
  "task_completion": 1-5,
  "correctness": 1-5,
  "constraint_adherence": 1-5,
  "groundedness": 1-5,
  "answer_quality": 1-5,
  "rationale": "short␣explanation"
}
```

D.6. Runtime and Resource Usage

Table 17 reports median runtime, call count, and token usage for each evaluated system. Brackets denote the interquartile range.

Table 17. Runtime and resource usage on CoSyn-Bench, reported as medians with interquartile ranges for each evaluated system.

System	Time (s)	Calls	Tokens
LLM-Only	4.04 [2.31, 5.17]	1.0 [1.0, 1.0]	779 [680, 901]
Context-Augmented LLM	16.52 [11.63, 26.89]	4.0 [2.0, 6.0]	21,590 [12,498, 41,202]
SA w/o Retrieval	14.61 [10.72, 24.49]	4.0 [3.0, 5.0]	11,902 [7,382, 18,100]
SA w/o Prediction	13.00 [9.22, 18.67]	4.0 [3.0, 6.0]	18,616 [14,032, 30,944]
SA w/o Article Analysis	12.59 [7.29, 15.67]	3.0 [2.0, 4.0]	20,298 [14,126, 29,479]
Single-Agent (SA)	17.87 [11.42, 25.34]	3.5 [3.0, 4.25]	57,610 [22,246, 105,856]
Multi-Agent (MA)	29.24 [21.74, 39.82]	7.0 [5.0, 10.0]	20,130 [14,994, 33,248]