TOWARDS LITERATURE-GROUNDED AGENTIC CO-CRYSTAL SYNTHESIS DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

Co-crystallization has become an important approach for modulating the physicochemical properties of drug molecules, yet experimental development remains challenging. While recent machine-learning methods have advanced co-crystal screening, the downstream problem of synthesis design remains largely unresolved. We present COSYN[1], a literature-grounded LLM-based agentic framework for supporting co-crystal synthesis design that combines (i) hierarchical extraction of method-specific synthesis information, (ii) a large-scale machine-readable database of 20,783 co-crystal synthesis records across multiple methods, and (iii) method-specific starting-condition prediction for new molecular pairs. We systematically evaluate each component using manually annotated datasets and prospectively validate recommended conditions, achieving phase-pure co-crystals for three of four held-out pairs. At the system level, tool-augmented orchestration improves the average score on our targeted domain benchmark from 1.88 for an LLM without tool use to 3.58 on a 1–5 scale. These results demonstrate that structured, literature-derived experimental evidence can support co-crystal synthesis decision-making beyond pair-level screening.

1 INTRODUCTION

Pharmaceutical co-crystals have emerged as an important solid-form strategy for improving the developability of active pharmaceutical ingredients, with reported benefits for solubility, dissolution, stability, and bioavailability (Yadav et al., 2025; Marcos Valdez et al., 2026). A pharmaceutical co-crystal comprises an active pharmaceutical ingredient (API) and a pharmaceutically acceptable co-former that co-crystallize through non-covalent interactions (Bolla et al., 2022). Despite their broad potential, co-crystal development remains experimentally challenging, requiring iterative screening and empirical optimization across heterogeneous synthesis conditions (Karimi-Jafari et al., 2018; Zhang et al., 2025a).

Artificial intelligence has advanced the early stages of co-crystal design, from machine-learning models for coformer screening and co-crystal formation prediction (Wicker et al., 2017; Devogelaer et al., 2020) to recent approaches using richer molecular representations and thermodynamic information (Guo et al., 2024; Birolo et al., 2025; Song et al., 2025b). However, these advances largely address whether a molecular pair may form a co-crystal, while downstream synthesis design remains unresolved. Since co-crystal preparation spans solution-based and solid-state approaches, synthesis design must account for method-dependent constraints and parameter spaces. This requires coordinated selection of stoichiometry, component and solvent quantities, processing time and temperature, and experimental setup (Karimi-Jafari et al., 2018; Guo et al., 2021).

A major obstacle to systematic progress is that co-crystal synthesis knowledge remains scattered across the literature as heterogeneous free-text procedures rather than standardized, machine-readable records, hindering cross-study comparison, identification of method-specific regularities, and construction of datasets for predictive modeling (Ai et al., 2024). Chemical information extraction has progressed from systems such as OSCAR4 and ChemDataExtractor (Jessop et al., 2011; Swain & Cole, 2016) to recent general-purpose methods (Fan et al., 2024; Dagdelen et al., 2024) and domain-specific approaches for organic synthesis (Vaucher et al., 2021; Ai et al., 2024), MOF

[1]Code and data are available at `https://anonymous.4open.science/r/CoSyn/`.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

chemistry (Shi et al., 2024; Zheng et al., 2023), and nanomaterials (Odobesku et al., 2025). Recent studies nevertheless emphasize that reliable extraction in specialized domains depends on domain-specific schemas, normalization, and validation (Schilling-Wilhelmi et al., 2025; Vepreva et al., 2025; Chong & Colindres, 2026). For co-crystals, where synthesis routes expose distinct parameter spaces, this motivates a standardized, method-aware extraction schema for converting synthesis narratives into structured records (Karimi-Jafari et al., 2018; Yadav et al., 2025).

Large language models (LLMs) are increasingly used as general-purpose components of scientific AI systems, supporting tasks that range from information extraction to reasoning and tool-mediated workflows (Zhang et al., 2025b; M. Bran et al., 2024; Ruan et al., 2024). In chemistry, these capabilities have been applied to extraction, prediction, and agentic tool use (Jablonka et al., 2024; Ramos et al., 2025), making them well suited to co-crystal synthesis, where literature knowledge must be structured, retrieved, and translated into method-specific synthesis recommendations. However, these capabilities have not been integrated into a unified literature-grounded framework.

In this work, we take a step toward literature-grounded agentic co-crystal synthesis design with CoSyn, an LLM-based framework that transforms synthesis literature into structured experimental records and uses them for condition recommendation and decision support. Our contributions are:

1. We introduce a method-aware taxonomy and hierarchical extraction pipeline for structuring co-crystal synthesis procedures and their method-specific parameters.
2. We construct CoSyn-DB, a large-scale machine-readable database of 20,783 literature-derived records linking API–coformer pairs to method-specific synthesis conditions.
3. We formulate and evaluate literature-grounded starting-condition recommendation for new API–coformer pairs as a downstream task beyond pair-level co-crystal screening.
4. We integrate literature analysis, database retrieval, and condition prediction into an agentic decision-support system for co-crystal synthesis.

2 Related Work

AI for Co-Crystal Design. Computational co-crystal design has primarily addressed pair formation and coformer prioritization (von Essen & Luedeker, 2023; Zhang et al., 2025a). Early descriptor-based models (Wicker et al., 2017; Wang et al., 2020) were followed by network-based prediction (Devogelaer et al., 2019) and learned molecular representations, including CCGNet, CCPT, and DeepCocrystal (Jiang et al., 2021; Guo et al., 2024; Birolo et al., 2025). More recent work incorporates solvent selection and thermodynamic information (Song et al., 2025b; Zeng et al., 2025), while GEMCODE explores property-driven generative design (Gubina et al., 2024). Building on this progress, CoSyn moves downstream from pair selection to literature-grounded recommendation of method-specific synthesis conditions.

Synthesis Condition Prediction. Synthesis-condition prediction requires joint selection of categorical and numerical variables, including solvents, reagents, temperature, and time (Ball et al., 2025). Prior approaches use hierarchical models (Gao et al., 2018), staged prediction and ranking (Chen & Li, 2024), or template- and cluster-based formulations for coherent condition sets (Wang et al., 2025). LLM-based methods further support joint prediction of correlated conditions in low-data settings, including BatGPT-Chem (Yang et al., 2025), Chemma-RC (Zhang et al., 2024), and in-context prediction for inorganic synthesis (Prein et al., 2025). CoSyn applies this paradigm to co-crystals by retrieving route-specific literature precedents and predicting method-specific starting conditions for API–coformer pairs.

Automated Extraction of Scientific Data. Chemical literature extraction has progressed from general text-mining tools such as OSCAR4, ChemicalTagger, and ChemDataExtractor (Jessop et al., 2011; Hawizy et al., 2011; Swain & Cole, 2016) to document-level and LLM-based systems for text, tables, and figures (Dagdelen et al., 2024; Fan et al., 2024). Synthesis-oriented work includes procedure parsing and reaction extraction (Vaucher et al., 2021; Kearnes et al., 2021; Ai et al., 2024), together with domain-specific pipelines for MOFs and nanomaterials (Zheng et al., 2023; Odobesku et al., 2025). Recent reviews and benchmarks emphasize that reliable specialized extraction also depends on domain schemas, normalization, provenance, and validation (Schilling-Wilhelmi et al.,

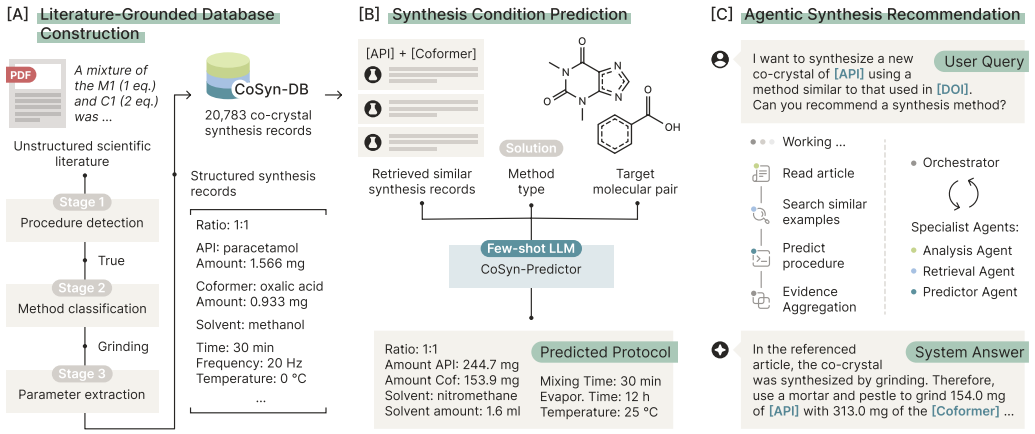

Figure 1: Overview of CoSyn. [A] Automated extraction builds CoSyn-DB from co-crystal articles via procedure detection, method classification, and parameter extraction. [B] Condition prediction retrieves similar API–coformer records and uses few-shot LLM prompting to generate method-specific protocols.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

2025; Vepreva et al., 2025; Chong & Colindres, 2026). Accordingly, we formulate co-crystal synthesis extraction as a method-aware task for constructing structured experimental records.

Agentic Systems and Evaluation. Beyond extraction and prediction, LLM agents increasingly support multi-step scientific workflows, from tool-augmented systems such as Coscientist and ChemCrow (Boiko et al., 2023; M. Bran et al., 2024) to role-specialized multi-agent systems such as ChemAgent and LLM-RDF (Song et al., 2025a; Ruan et al., 2024). Evaluation remains difficult because scientific tasks are open-ended and tool contributions task-dependent (Gu et al., 2024; Yu et al., 2025). Existing benchmarks target specific scientific workflows (Chen et al., 2025) or broad chemistry tasks (Mirza et al., 2025), but not the coupled analysis, retrieval, and condition prediction required here. We therefore adopt an agentic architecture for co-crystal synthesis decision support and pair it with a targeted domain benchmark.

3 Dataset Construction

3.1 Source Corpus

We constructed 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 associated with relevant CSD entries, we obtained 4,287 unique source DOIs after deduplication (Figure 2).

Automated full-text retrieval yielded 3,997 PDFs, which were converted to machine-readable text and used as input to the extraction pipeline. Retrieval failures were due to access restrictions, missing full-text links, publisher-side failures, or conversion errors. Corpus-level statistics, including open-access status and publication-year distributions, are provided in Appendix A.1.

3.2 Method Taxonomy

Co-crystal synthesis methods differ in experimental structure and parameter spaces, making a uniform schema unsuitable. We therefore defined a method-aware taxonomy of 12 synthesis types, covering solution-based, solid-state, and specialized routes (Karagianni et al., 2018; Buddhadev & Garala, 2021; Douroumis et al., 2017; Pawar et al., 2021). For each method, we prepared two artifacts: (i) a natural-language description of the key physicochemical steps and (ii) a representative procedure template with explicitly marked parameter slots.

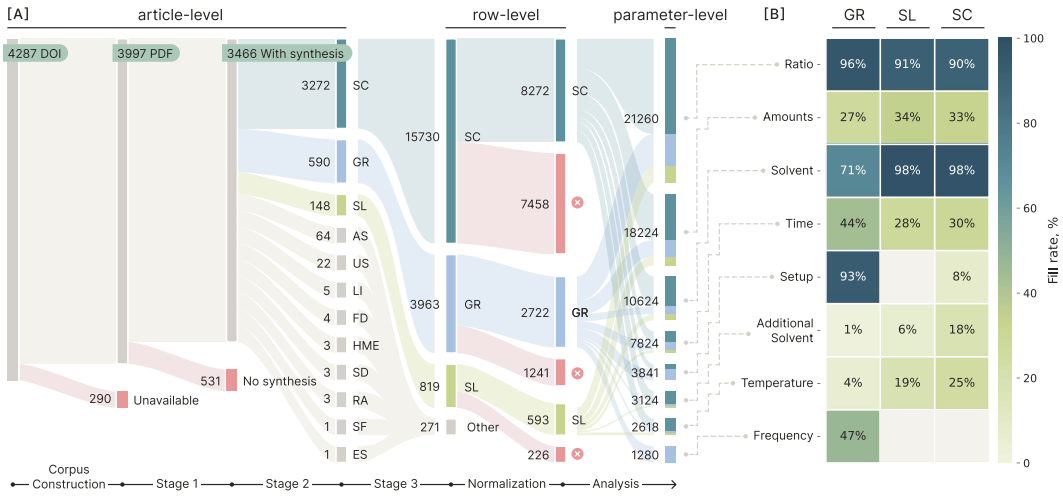

Figure 2: CoSyn-DB construction pipeline and parameter coverage. [A] Data flow from 4,287 CSD-derived DOIs to normalized synthesis records and parameter families. [B] Parameter-family fill rates for the downstream datasets after quality filtering. Method abbreviations and parameter-family definitions are provided in Appendix A.2 and Appendix A.14.

The taxonomy is operational rather than exhaustive: rare or non-standard routes may be absent, while procedures sharing experimental logic and parameter spaces are grouped within a single category, with intra-category variation captured through optional fields. The full taxonomy, parameter sets, and templates are provided in Appendix A.2.

3.3 Synthesis Extraction

Hierarchical pipeline. A naive one-step extraction procedure is unsuitable here because (i) not every article contains an explicit synthesis procedure, as it may be omitted or replaced by a reference to earlier work, and (ii) the parameter schema is method-dependent, while the method is not known in advance. A single prompt covering detection, method inference, and full parameter extraction would force the model to decide among many irrelevant fields, increasing the risk of hallucination (Ji et al., 2023). We therefore formulate extraction as a three-stage hierarchical task.

Stage 1: Procedure detection. Given the full article text, the model decides 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 A.2), isolating method identification so that downstream prompts use only the relevant schema.

Stage 3: Parameter extraction. For each detected method, the model extracts experimental parameters into a structured synthesis record using the corresponding method-specific schema.

Strategy development. Three hand-annotated development sets guided strategy refinement and backbone selection: 100 articles for Stage 1 procedure detection (92 positive, 8 negative), 109 for Stage 2 classification across all 12 methods (54 single-method, 55 multi-method), and 55 liquid-assisted grinding (LAG) articles with 295 records over 25 parameters for Stage 3. LAG was chosen as a well-supported method with a rich schema. Because these sets informed both prompt design and model selection, all resulting metrics are treated as development estimates (see Appendix A.3).

Using `GPT-4o-mini`, we selected span-based procedure detection for Stage 1 and template-conditioned taxonomy classification for Stage 2. For Stage 3, we refined the extraction prompt with expert feedback on the LAG set and adapted it to each method-specific schema. Prompt variants and comparison results are provided in Appendices A.4 and A.5.

Table 1: Backbone selection results on development data for procedure detection (Stage 1), 12-route method classification (Stage 2), and LAG parameter extraction (Stage 3).

	Stage 1		Stage 2		Stage 3	
Model	**F1**	**Accuracy**	**F1**	**Accuracy**	**F1**	**Accuracy**
GPT-4o-mini	0.88 ±0.03	0.80 ±0.04	0.64 ±0.05	0.61 ±0.05	0.54 ±0.03	0.45 ±0.03
Gemma-3-27B	0.92 ±0.02	0.85 ±0.04	0.44 ±0.05	0.47 ±0.05	0.56 ±0.03	0.47 ±0.04
Llama-3.3-70B	0.73 ±0.04	0.61 ±0.05	0.19 ±0.04	0.14 ±0.03	0.61 ±0.03	0.53 ±0.04
Gemini-2.5-Flash	**0.97 ±0.01**	**0.95 ±0.02**	**0.81 ±0.04**	**0.82 ±0.04**	**0.63 ±0.03**	**0.56 ±0.04**

Table 2: Held-out extraction performance of `Gemini-2.5-Flash` on independent articles excluded from strategy development, covering grinding, slurry, and solution crystallization.

	Task	Precision	Recall	F1	Accuracy
Stage 1	Procedure detection	0.92 ±0.04	1.00 ±0.00	0.96 ±0.02	0.93 ±0.03
Stage 2	Method classification	0.97 ±0.02	0.91 ±0.03	0.94 ±0.02	0.92 ±0.02
Stage 3	Parameter extraction	0.72 ±0.02	0.79 ±0.02	0.72 ±0.02	0.65 ±0.02

With the prompting configuration fixed, we compared four LLM backbones (Table 1). `Gemini-2.5-Flash` achieved the highest F1 and accuracy across all stages and was selected for full-corpus extraction. `Gemini-3.1-Pro` gave only modest held-out gains at substantially higher inference cost (see Appendix A.7). Full backbone and per-parameter Stage 3 results are reported in Appendices A.8 and A.9. Stage 1–2 errors mainly involved omissions under non-standard terminology or comparative descriptions, whereas Stage 3 discrepancies more often involved record alignment, nearby procedural details, and strict multiset scoring (see Appendix A.10).

Held-out evaluation. After fixing the extraction strategy and LLM backbone, we evaluated the finalized pipeline on 60 previously unseen articles: 15 for each dominant synthesis route (grinding, slurry crystallization, and solution crystallization) and 15 without an explicit co-crystal synthesis procedure. The 45 synthesis-containing articles comprise 207 manually annotated records.

Table 2 shows that Stage 1 retained perfect recall, detecting all synthesis-containing articles. For Stage 2, development performance on the matched three-route scope (F1 = 0.91, accuracy = 0.94) closely matched held-out results. Stage 3 extended from LAG to all three dominant routes, although differing parameter compositions preclude direct comparison. At the record level, one-to-one matching yielded an identification F1 of 0.726, with median consistency of 0.833 and 25.3% exact matches (Appendix A.11). We further examined Stage 3 on one annotated article for each remaining route as qualitative case studies (Appendix A.12). Together, these results support the finalized pipeline as a structured evidence base for downstream condition recommendation.

3.4 CoSyn-DB Dataset

Applied to the source corpus, the finalized pipeline identified 3,466 articles (86.7%) with co-crystal synthesis procedures and yielded 20,783 unique records across 12 methods. Unlike existing machine-readable resources focused mainly on pair-level screening or selected synthesis variables, CoSyn-DB links molecular pairs to record-specific conditions across multiple routes (see Appendix A.13). Solution crystallization (15,730 records), grinding (3,963), and slurry (819) account for 98.7% of entries and support downstream condition prediction, while the remaining routes are retained but too sparse for reliable modeling. Figure 2A summarizes the data flow.

Records were normalized by resolving API–coformer identities, harmonising categorical fields, and standardizing numerical units. Coverage remained uneven, with stoichiometric ratio and solvent identity reported more often than amounts, times, and temperatures (Figure 2B), reflecting heterogeneous literature reporting. Parameter families and normalization are detailed in Appendices A.14 and A.15.

Table 3: Grinding synthesis-condition prediction with $k = 3$ retrieved examples. Classical baselines and few-shot LLM predictors are compared using weighted F1, with solvent and ratio reported separately from categorical, numerical, and overall aggregates. Dashes denote metrics that are not defined for models that predict only numerical parameters.

Model	Solvent	Ratio	Categorical	Numerical	Overall
Frequency prior	0.17 ±0.00	0.87 ±0.00	0.26 ±0.00	0.38 ±0.00	0.35 ±0.00
RF	0.32 ±0.05	0.87 ±0.00	**0.54** ±0.00	0.43 ±0.01	**0.49** ±0.00
RF-Regression	–	0.87 ±0.00	–	0.33 ±0.00	–
RF-Chain	–	0.87 ±0.01	–	0.32 ±0.00	–
CB-Regression	–	0.87 ±0.00	–	0.34 ±0.01	–
GPT-4o-mini	0.36 ±0.01	0.85 ±0.00	0.48 ±0.00	0.45 ±0.01	0.46 ±0.00
GPT-5.4	0.41 ±0.00	**0.88** ±0.00	0.47 ±0.00	**0.49** ±0.00	0.48 ±0.00
Gemini-2.5-Flash	0.41 ±0.01	0.85 ±0.00	0.49 ±0.00	0.48 ±0.00	**0.49** ±0.00
Gemini-3.1-Pro	**0.44** ±0.01	**0.88** ±0.01	0.50 ±0.00	0.47 ±0.01	**0.49** ±0.00

4 Synthesis Condition Prediction

We next use CoSyn-DB for condition recommendation on new co-crystal pairs. Given an API–coformer pair and target synthesis method, the task is to predict a coherent set of method-specific categorical and numerical conditions.

4.1 Experimental Setup

Data preparation. We restrict CoSyn-DB to solution crystallization, grinding, and slurry crystallization, yielding 11,587 records after normalization and quality filtering. Records were split by connected components over shared DOIs and API–coformer pairs to prevent literature- and pair-level leakage, producing a balanced held-out set of 300 records (100 per method). Missing values were imputed only in training records used for retrieval, while original test records were retained for evaluation. Data partitioning and imputation are detailed in Appendices B.1 and B.2.

Prediction models. Classical baselines use concatenated API–coformer Morgan fingerprints computed with RDKit (Landrum et al., 2025) (BSD-3-Clause). We evaluate a frequency prior and multi-output Random Forest classifier for categorical fields, and Random Forest regression, regressor chains, and CatBoost (Prokhorenkova et al., 2018) (Apache-2.0) for numerical fields. We compare these baselines with four LLMs evaluated in a few-shot setting: `GPT-4o-mini`, `GPT-5.4`, `Gemini-2.5-Flash`, and `Gemini-3.1-Pro`. For each test pair, the three most similar training pairs by molecular fingerprint similarity are used as in-context examples. The choice of k=3, retrieval procedure, and prediction prompts are detailed in Appendices B.6, B.4, and B.5.

Evaluation. The main cross-model comparison focuses on grinding, which combines categorical choices with numerical process parameters. We evaluate synthesis fields observed in at least 20% of held-out records, excluding auxiliary unit columns. Categorical fields use exact-match accuracy and weighted F1 after normalization, with solvent scored by family because of its high cardinality (Appendix B.7). Numerical fields use tolerance bins (Appendix B.8). We report weighted Categorical, Numerical, and Overall averages, with Solvent and Ratio shown separately to represent high- and low-entropy fields.

4.2 Prediction Results

Table 3 reports the main cross-model comparison. The frequency prior performs strongly on Ratio, reflecting common equimolar stoichiometries, but poorly on chemistry-dependent fields such as Solvent. All LLMs improve Solvent F1 from 0.17 to 0.36–0.44, indicating the value of query-specific retrieved precedents (Appendix B.9). Fingerprint-based ML baselines improve categorical prediction but underperform LLMs on numerical fields, suggesting that retrieved literature examples better capture procedural variation than molecular fingerprints alone.

Table 4: Prospective validation on four held-out API–coformer pairs under slurry crystallization. ● denotes phase-pure co-crystal formation, ◐ a mixed product containing residual starting material, and ○ no co-crystal formation.

Molecular pair	Caffeine + Oxalic acid	Caffeine + Benzoic acid	Urea + Oxalic acid	Urea + Salicylic acid
CoSyn	●	●	●	○
Frequency prior	◐	○	◐	○

Among LLMs, the Gemini models achieve the strongest aggregate performance, with `Gemini-2.5-Flash` and `Gemini-3.1-Pro` sharing the best Overall F1 and `Gemini-3.1-Pro` leading on Solvent. `GPT-5.4` approaches their aggregate performance, while `GPT-4o-mini` is weaker. Because `Gemini-2.5-Flash` matches `Gemini-3.1-Pro` overall at substantially lower inference cost, we use it as the Predictor agent in Section 5 and report its per-parameter results in Appendix B.10.

Because prediction depends on automatically extracted and partially completed literature records, we additionally examine sensitivity to these upstream steps. Appendix B.3 compares the standard pipeline with an observed-only setting, while Appendix B.11 examines how extraction errors propagate through retrieval and condition prediction.

4.3 Prospective Experimental Validation

To test whether the recommendations provide experimentally actionable starting conditions, we prospectively evaluated four literature-confirmed API–coformer pairs after removing all pair-specific records from CoSyn-DB. We fixed the synthesis method to slurry crystallization and compared CoSyn with the pair-agnostic frequency prior. Because prior-selected ethyl acetate was unavailable, it was replaced with 1,2-dimethoxyethane in all baseline experiments.

As summarized in Table 4, CoSyn yielded phase-pure co-crystals for three of four pairs, whereas the frequency prior produced two mixed products and two failures. For the fourth pair, CoSyn led to complete dissolution and the prior to phase separation. This small, single-method study provides initial prospective evidence for experimentally actionable starting conditions, not comprehensive validation. Full conditions and observations are reported in Appendix B.12.

5 Agentic Co-Crystal Design

The components introduced above, article extraction, CoSyn-DB retrieval, and few-shot prediction, address complementary sub-problems often required jointly in practical co-crystal synthesis design. We therefore integrate them through an agentic interface and compare coordinated tool use against non-agentic LLM baselines.

5.1 Experimental Setup

Benchmark. We introduce CoSyn-Bench, a compact benchmark of 30 expert-curated synthesis-design questions. It 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 scoring metadata specifying the available information, task constraints, and expected answer components. Representative items are provided in Appendix C.4.

System configurations. We compare two non-agentic baselines with two agentic architectures. The LLM-only baseline receives only the user question, while the Context LLM additionally receives task-relevant static evidence, such as extracted article text or database-derived information. The **Single-Agent** (SA) uses one controller to plan, invoke CoSyn tools, and produce the final response. The **Multi-Agent** (MA) uses an Orchestrator that delegates article analysis, database retrieval, and condition prediction to specialist agents before aggregating their outputs. To isolate individual capabilities, we additionally ablate SA by removing Retrieval, Prediction, or Article Analysis

Table 5: CoSyn-Bench evaluation of non-agentic baselines, ablated agents, and full agentic systems. Scores are reported by capability and question complexity on a 1–5 scale. Pass Rate is the fraction of questions with an overall score of at least 4.

	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 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
SA 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
SA 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
SA 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 (SA)	**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 (MA)	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

one at a time and replacing the removed capability with the corresponding static context available to the Context LLM. All agent controllers use `GPT-4o-mini`. Architecture, tool, and prompt details are provided in Appendices C.2, C.1, and C.3.

Evaluation. Responses are scored by `GPT-5.4` using the question, gold answer, benchmark metadata, execution context, and candidate response. The judge assigns scores from 1 to 5 for Task completion, Correctness, Constraint adherence, and Groundedness, with the question-level overall score defined as their mean. We report scores by capability and question complexity together with Pass Rate, defined as the fraction of questions with an overall score of at least 4. The judge prompt is provided in Appendix C.5.

5.2 Agentic Results

Table 5 compares the non-agentic baselines, single-capability ablations, and full agentic architectures, highlighting the contribution of coordinated tool use.

System comparison. The non-agentic baselines achieve substantially lower overall scores: LLM-only reaches 1.88, while adding static task-relevant context increases the score only to 2.01. In contrast, SA and MA reach comparable overall scores of 3.58 and 3.56, respectively. Their capability profiles differ: SA achieves the highest Retrieval and Analysis scores, while MA achieves the highest Prediction score, Pass Rate, and multi-capability score. Thus, explicit decomposition does not provide a consistent aggregate advantage, but changes how performance is distributed across task types. MA also uses fewer tokens while requiring more calls and wall-clock time than SA (Appendix C.6). Independent expert scoring of 60 SA and MA responses showed that overall LLM–expert agreement ($\rho = 0.768$) was comparable to inter-expert agreement ($\rho = 0.778$), with similar patterns for both architectures (Appendix C.7).

Capability ablations. Removing Retrieval, Prediction, and Article Analysis from SA reduces the corresponding capability scores by 39%, 27%, and 34%, respectively. The smaller drop for Article Analysis is consistent with the ablated system still receiving article text as static context and therefore retaining access to some directly recoverable information. Together, the ablations indicate that the benchmark gains depend on interactive use of the domain-specific capabilities rather than static evidence alone.

6 Discussion

Taken together, our results support extending co-crystal AI beyond pair-level screening toward literature-grounded synthesis design. CoSyn takes a step in this direction by converting heterogeneous synthesis literature into structured experimental evidence and using it for method-specific starting-condition recommendation and agentic decision support.

The database-construction results show that the published co-crystal literature contains substantial procedural knowledge, but using it for data-driven modeling requires method-aware structuring and normalization. CoSyn-DB captures this knowledge in 20,783 record-specific synthesis procedures, revealing both the richness and imbalance of the evidence: 98.7% of records belong to three dominant methods, while ratios and solvents are reported far more consistently than amounts, times, and temperatures. Thus, CoSyn-DB provides both a structured evidence base and a practical boundary for downstream co-crystal synthesis modeling.

The prediction results indicate that retrieved procedural precedents can inform method-specific starting conditions for new molecular pairs. Performance remains parameter-dependent, with stoichiometric ratios benefiting from strong domain regularities and solvents and sparse numerical quantities presenting greater difficulty. Imputation improves the completeness of retrieved evidence but has mixed parameter-level effects, while extraction errors can substantially change retrieved precedents without producing a consistent aggregate loss on the held-out grinding set. Prospective experiments further support the actionability of these recommendations, yielding phase-pure co-crystals for three of four tested pairs under slurry crystallization. Together, these findings support CoSyn as a literature-grounded decision-support system for starting-condition recommendation and subsequent expert refinement.

The system-level evaluation shows that static context alone provides little benefit, whereas tool-augmented agents improve performance by combining retrieval, article analysis, and condition prediction. Ablations show that removing individual capabilities selectively weakens the corresponding task types. Single-Agent and Multi-Agent systems perform similarly overall but differ in profile: the former is faster and simpler, while the latter is more context-efficient and stronger on prediction-heavy and multi-capability questions. Overall, reliable orchestration of complementary evidence sources matters more than the specific agent architecture.

Limitations. The current study has several limitations. (i) CoSyn inherits biases and uneven reporting from the published literature, including sparse coverage of unsuccessful experiments and quantitative conditions. (ii) The operational taxonomy provides broad extraction coverage, but available data support quantitative condition prediction primarily for the three dominant synthesis methods. (iii) Extraction and condition prediction remain imperfect, particularly for high-cardinality and sparsely reported parameters, allowing uncertainty to propagate through downstream use. (iv) Recommendations are literature-grounded and conditioned on a specified synthesis method, constraining them to literature-supported regions of the condition space. (v) Prospective and system-level evaluation remain limited in scale.

Future directions. Future work should strengthen CoSyn along four directions. First, the evidence base should expand to additional literature, unsuccessful experiments, and rarer synthesis routes. Second, method-specific schemas could be complemented by a cross-method representation of shared synthesis stages and operations, better capturing mixed and non-standard procedures. Third, prediction should be treated as a multi-solution problem, since one API–coformer pair may admit multiple valid methods and condition sets. Finally, literature-grounded recommendations should incorporate physicochemical models, uncertainty estimation, and prospective feedback to prioritize well-supported conditions for experimental validation.

7 Conclusion

We introduced CoSyn, a literature-grounded LLM-based framework that frames co-crystal synthesis design as a downstream task beyond pair-level screening. Its central contribution is to make dispersed synthesis knowledge usable for this task by converting heterogeneous literature procedures into method-specific experimental records in CoSyn-DB. These records support starting-condition recommendation for new API–coformer pairs, with prospective experiments providing initial evidence of experimental actionability. At the system level, coordinating retrieval, article analysis, and condition prediction further improves synthesis assistance over an LLM without tool use. The results show how structured experimental literature can serve as an evidence base for agentic decision support in co-crystal synthesis, laying the groundwork for broader research on systems that connect scientific knowledge with experimental decision making.

AI USE STATEMENT

We used generative AI tools during manuscript preparation to assist with drafting, restructuring, and refining parts of the text, including improvements to clarity, concision, and organization. AI-assisted suggestions were used iteratively and were reviewed and edited by the authors, who retained control over the scientific content and final wording. This use was limited to manuscript preparation and is distinct from the LLMs evaluated as components of CoSyn, which are described in the corresponding methodological sections.

REPRODUCIBILITY STATEMENT

Code and data supporting this work are available at `https://anonymous.4open.science/r/CoSyn/`, with methodological, implementation, data-processing, and evaluation details documented in the paper and supplementary material. Reproduction of the full data-construction pipeline additionally requires access to the original data sources. The Cambridge Structural Database (CSD) requires a CCDC licence, while subscription-based articles require publisher or institutional access and cannot be redistributed. Experiments relying on proprietary LLM APIs may also be affected by model availability and provider-side updates.

REFERENCES

Qianxiang Ai, Fanwang Meng, Jiale Shi, Brenden Pelkie, and Connor W Coley. Extracting structured data from organic synthesis procedures using a fine-tuned large language model. *Digital discovery*, 3(9):1822–1831, 2024.

Matthew Ball, Dragos Horvath, Thierry Kogej, Mikhail Kabeshov, and Alexandre Varnek. Predicting reaction conditions: a data-driven perspective. *Chemical Science*, 16(38):17523–17541, 2025.

Rebecca Birolo, Rıza Özçelik, Andrea Aramini, Roberto Gobetto, Michele R Chierotti, and Francesca Grisoni. Deep supramolecular language processing for co-crystal prediction. *Angewandte Chemie International Edition*, 64(29):e202507835, 2025.

Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023.

Geetha Bolla, Bipul Sarma, and Ashwini K Nangia. Crystal engineering of pharmaceutical cocrystals in the discovery and development of improved drugs. *Chemical reviews*, 122(13):11514–11603, 2022.

Sheetal S Buddhadev and Kevin C Garala. Pharmaceutical cocrystals—a review. In *Proceedings*, volume 62, pp. 14. MDPI, 2021.

Harrison Chase. LangChain, October 2022. URL `https://github.com/langchain-ai/langchain`.

Lung-Yi Chen and Yi-Pei Li. Enhancing chemical synthesis: a two-stage deep neural network for predicting feasible reaction conditions. *Journal of Cheminformatics*, 16(1):11, 2024.

Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, et al. Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *International Conference on Learning Representations*, volume 2025, pp. 96934–96990, 2025.

Curtis Chong and Jorge Colindres. Litxbench: A benchmark for extracting experiments from scientific literature. *arXiv preprint arXiv:2604.07649*, 2026.

John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S Rosen, Gerbrand Ceder, Kristin A Persson, and Anubhav Jain. Structured information extraction from scientific text with large language models. *Nature communications*, 15(1):1418, 2024.

Jan-Joris Devogelaer, Sander JT Brugman, Hugo Meekes, Paul Tinnemans, Elias Vlieg, and René de Gelder. Cocrystal design by network-based link prediction. *CrystEngComm*, 21(44):6875–6885, 2019.

Jan-Joris Devogelaer, Hugo Meekes, Paul Tinnemans, Elias Vlieg, and Rene De Gelder. Co-crystal prediction by artificial neural networks. *Angewandte Chemie International Edition*, 59 (48):21711–21718, 2020.

Dennis Douroumis, Steven A Ross, and Ali Nokhodchi. Advanced methodologies for cocrystal synthesis. *Advanced drug delivery reviews*, 117:178–195, 2017.

Vincent Fan, Yujie Qian, Alex Wang, Amber Wang, Connor W Coley, and Regina Barzilay. Openchemie: an information extraction toolkit for chemistry literature. *Journal of Chemical Information and Modeling*, 64(14):5521–5534, 2024.

Hanyu Gao, Thomas J Struble, Connor W Coley, Yuran Wang, William H Green, and Klavs F Jensen. Using machine learning to predict suitable conditions for organic reactions. *ACS central science*, 4(11):1465, 2018.

Jan Roland Gröls and Bernardo Castro-Dominguez. Mechanochemical co-crystallization: Insights and predictions. *Computers & Chemical Engineering*, 153:107416, 2021.

Colin R Groom, Ian J Bruno, Matthew P Lightfoot, and Suzanna C Ward. The cambridge structural database. *Structural Science*, 72(2):171–179, 2016.

Ken Gu, Ruoxi Shang, Ruien Jiang, Keying Kuang, Richard-John Lin, Donghe Lyu, Yue Mao, Youran Pan, Teng Wu, Jiaqian Yu, et al. Blade: Benchmarking language model agents for data-driven science. In *Findings of the association for computational linguistics: EMNLP 2024*, pp. 13936–13971, 2024.

Nina Gubina, Andrei Dmitrenko, Gleb Solovev, Lyubov Yamshchikova, Oleg Petrov, Ivan Lebedev, Nikita Serov, Grigorii Kirgizov, Nikolay Nikitin, and Vladimir Vinogradov. Hybrid generative ai for de novo design of co-crystals with enhanced tabletability. *Advances in Neural Information Processing Systems*, 37:84606–84644, 2024.

Jiali Guo, Songran Yang, Chenghui Wang, Jing Liu, Yanzhi Guo, Zongwei Yang, Xueyan Zhao, and Xuemei Pu. Cocrystal prediction tool (ccpt): a web server for deep learning-assisted cocrystal screening and density evaluation. *Crystal Growth & Design*, 24(20):8407–8414, 2024.

Minshan Guo, Xiaojie Sun, Jiahui Chen, and Ting Cai. Pharmaceutical cocrystals: A review of preparations, physicochemical properties and applications. *Acta Pharmaceutica Sinica B*, 11(8): 2537–2564, 2021.

Lezan Hawizy, David M Jessop, Nico Adams, and Peter Murray-Rust. Chemicaltagger: A tool for semantic text-mining in chemistry. *Journal of cheminformatics*, 3(1):17, 2011.

Kevin Maik Jablonka, Philippe Schwaller, Andres Ortega-Guerrero, and Berend Smit. Leveraging large language models for predictive chemistry. *Nature Machine Intelligence*, 6(2):161–169, 2024.

David M Jessop, Sam E Adams, Egon L Willighagen, Lezan Hawizy, and Peter Murray-Rust. Oscar4: a flexible architecture for chemical text-mining. *Journal of cheminformatics*, 3(1):41, 2011.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38, 2023.

Yuanyuan Jiang, Zongwei Yang, Jiali Guo, Hongzhen Li, Yijing Liu, Yanzhi Guo, Menglong Li, and Xuemei Pu. Coupling complementary strategy to flexible graph neural network for quick discovery of coformer in diverse co-crystal materials. *Nature Communications*, 12(1):5950, 2021.

Anna Karagianni, Maria Malamatari, and Kyriakos Kachrimanis. Pharmaceutical cocrystals: New solid phase modification approaches for the formulation of apis. *Pharmaceutics*, 10(1):18, 2018.

Maryam Karimi-Jafari, Luis Padrela, Gavin M Walker, and Denise M Croker. Creating cocrystals: A review of pharmaceutical cocrystal preparation routes and applications. *Crystal Growth & Design*, 18(10):6370–6387, 2018.

Steven M Kearnes, Michael R Maser, Michael Wleklinski, Anton Kast, Abigail G Doyle, Spencer D Dreher, Joel M Hawkins, Klavs F Jensen, and Connor W Coley. The open reaction database. *Journal of the American Chemical Society*, 143(45):18820–18826, 2021.

Greg Landrum, Paolo Tosco, Brian Kelley, Ricardo Rodriguez, David Cosgrove, Riccardo Vianello, sriniker, Peter Gedeck, Gareth Jones, NadineSchneider, Eisuke Kawashima, Dan Nealschneider, Andrew Dalke, Matt Swain, Brian Cole, Samo Turk, Aleksandr Savelev, tadhurst cdd, Alain Vaucher, Maciej Wójcikowski, Ichiru Take, Rachel Walker, Vincent F. Scalfani, Hussein Faara, Kazuya Ujihara, Daniel Probst, Juuso Lehtivarjo, guillaume godin, Axel Pahl, and Jeremy Monat. rdkit/rdkit: 2025_03_1 (q1 2025) release, March 2025. URL `https://doi.org/10.5281/zenodo.15115844`.

Huiyu Liu, Muyao Yu, Zikai Xie, Xiangyu Zhang, Yiru Sun, Yuyang Ling, Zezhao Xu, Zidi Wang, Na Yu, Yue-Biao Zhang, et al. Data-driven discovery of polar organic cocrystals: integration of machine learning and automated screening. *Journal of the American Chemical Society*, 147(49): 45523, 2025.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature machine intelligence*, 6(5):525–535, 2024.

Marina Monserrat Marcos Valdez, Norma Rebeca Sperandeo, Maria Soledad Bueno, and Claudia Garnero. Pharmaceutical cocrystals in drug-delivery technologies: advances from rational design to therapeutic applications. *Pharmaceutics*, 18(1):128, 2026.

Adrian Mirza, Nawaf Alampara, Sreekanth Kunchapu, Martiño Ríos-García, Benedict Emoekabu, Aswanth Krishnan, Tanya Gupta, Mara Schilling-Wilhelmi, Macjonathan Okereke, Anagha Aneesh, et al. A framework for evaluating the chemical knowledge and reasoning abilities of large language models against the expertise of chemists. *Nature Chemistry*, 17(7):1027–1034, 2025.

R Odobesku, K Romanova, S Mirzaeva, O Zagorulko, R Sim, R Khakimullin, J Razlivina, A Dmitrenko, and V Vinogradov. Agent-based multimodal information extraction for nanomaterials. *npj Computational Materials*, 11(1):194, 2025.

Nitin Pawar, Anindita Saha, Neelesh Nandan, and Jose V Parambil. Solution cocrystallization: A scalable approach for cocrystal production. *Crystals*, 11(3):303, 2021.

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

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

Mayk Caldas Ramos, Christopher J Collison, and Andrew D White. A review of large language models and autonomous agents in chemistry. *Chemical science*, 16(6):2514–2572, 2025.

Yixiang Ruan, Chenyin Lu, Ning Xu, Yuchen He, Yixin Chen, Jian Zhang, Jun Xuan, Jianzhang Pan, Qun Fang, Hanyu Gao, et al. An automatic end-to-end chemical synthesis development platform powered by large language models. *Nature communications*, 15(1):10160, 2024.

Mara Schilling-Wilhelmi, Martiño Ríos-García, Sherjeel Shabih, María Victoria Gil, Santiago Miret, Christoph T Koch, José A Márquez, and Kevin Maik Jablonka. From text to insight: large language models for chemical data extraction. *Chemical Society Reviews*, 54(3):1125–1150, 2025.

Lei Shi, Zhimeng Liu, Yi Yang, Weize Wu, Yuyang Zhang, Hongbo Zhang, Jing Lin, Siyu Wu, Zihan Chen, Ruiming Li, et al. Llm-based mofs synthesis condition extraction using few-shot demonstrations. *arXiv preprint arXiv:2408.04665*, 2024.

Tao Song, Man Luo, Xiaolong Zhang, Linjiang Chen, Yan Huang, Jiaqi Cao, Qing Zhu, Daobin Liu, Baicheng Zhang, Gang Zou, 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.

Yutong Song, Yewei Ding, Junyi Su, Jian Li, and Yuanhui Ji. Unlocking the potential of machine learning in co-crystal prediction by a novel approach integrating molecular thermodynamics. *Angewandte Chemie International Edition*, 64(21):e202502410, 2025b.

Matthew C Swain and Jacqueline M Cole. Chemdataextractor: a toolkit for automated extraction of chemical information from the scientific literature. *Journal of chemical information and modeling*, 56(10):1894–1904, 2016.

Alain C Vaucher, Philippe Schwaller, Joppe Geluykens, Vishnu H Nair, Anna Iuliano, and Teodoro Laino. Inferring experimental procedures from text-based representations of chemical reactions. *Nature communications*, 12(1):2573, 2021.

Anastasia Vepreva, Julia Razlivina, Mariia Eremeyeva, Nina Gubina, Anastasia Orlova, Aleksei Dmitrenko, Kapranova Xenia, Susan Jyakhwo, Nikita Vasilev, Arsen Sarkisyan, et al. Chemx: A collection of chemistry datasets for benchmarking automated information extraction. *Advances in Neural Information Processing Systems*, 38, 2025.

Carolina von Essen and David Luedeker. In silico co-crystal design: Assessment of the latest advances. *Drug Discovery Today*, 28(11):103763, 2023.

Aikaterini Vriza, Ioana Sovago, Daniel Widdowson, Vitaliy Kurlin, Peter A Wood, and Matthew S Dyer. Molecular set transformer: attending to the co-crystals in the cambridge structural database. *Digital Discovery*, 1(6):834–850, 2022.

Dingyan Wang, Zeen Yang, Bingqing Zhu, Xuefeng Mei, and Xiaomin Luo. Machine-learning-guided cocrystal prediction based on large data base. *Crystal Growth & Design*, 20(10):6610–6621, 2020.

Zihan Wang, Kangjie Lin, Jianfeng Pei, and Luhua Lai. Reacon: a template-and cluster-based framework for reaction condition prediction. *Chemical Science*, 16(2):854–866, 2025.

Jerome GP Wicker, Lorraine M Crowley, Oliver Robshaw, Edmund J Little, Stephen P Stokes, Richard I Cooper, and Simon E Lawrence. Will they co-crystallize? *CrystEngComm*, 19(36): 5336–5340, 2017.

Vaishali Yadav, Rohitash Kumar, Meghana Sharma, Abhimanew Dhir, and Vimal K Bhardwaj. Pharmaceutical cocrystals: An overview of synthesis, characterization, and applications. *Journal of Molecular Structure*, pp. 143682, 2025.

Yifei Yang, Runhan Shi, Zuchao Li, Shu Jiang, Bao-Liang Lu, Qibin Zhao, Yang Yang, and Hai Zhao. Batgpt-chem: A foundation large model for chemical engineering. *Research*, 8:0827, 2025.

Botao Yu, Frazier N Baker, Ziru Chen, Garrett Herb, Boyu Gou, Daniel Adu-Ampratwum, Xia Ning, and Huan Sun. 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, 2025.

Qi Zeng, Yiqun Zhang, Yuting Peng, Qun Zeng, Guangxu Sun, Minshan Guo, and Ting Cai. Interpretable machine learning for solvent prediction and mechanistic insights in multi-component crystal screening. *Chemical Engineering Journal*, pp. 169397, 2025.

Chang Zhang, Li Zhang, Dezhi Yang, Henry HY Tong, Yang Lu, and Zhengzheng Zhou. From traditional screening to machine learning facilitated development of pharmaceutical cocrystals. *Chinese Chemical Letters*, pp. 111828, 2025a.

Yanbo Zhang, Sumeer A Khan, Adnan Mahmud, Huck Yang, Alexander Lavin, Michael Levin, Jeremy Frey, Jared Dunnmon, James Evans, Alan Bundy, et al. Exploring the role of large language models in the scientific method: from hypothesis to discovery. *npj Artificial Intelligence*, 1 (1):14, 2025b.

Yu Zhang, Ruijie Yu, Kaipeng Zeng, Ding Li, Feng Zhu, Xiaokang Yang, Yaohui Jin, and Yanyan Xu. Text-augmented multimodal llms for chemical reaction condition recommendation. *arXiv preprint arXiv:2407.15141*, 2024.

Zhiling Zheng, Oufan Zhang, Christian Borgs, Jennifer T Chayes, and Omar M Yaghi. Chatgpt chemistry assistant for text mining and the prediction of mof synthesis. *Journal of the American Chemical Society*, 145(32):18048, 2023.

A Dataset Construction

A.1 Source Corpus Statistics

Figure 3 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.

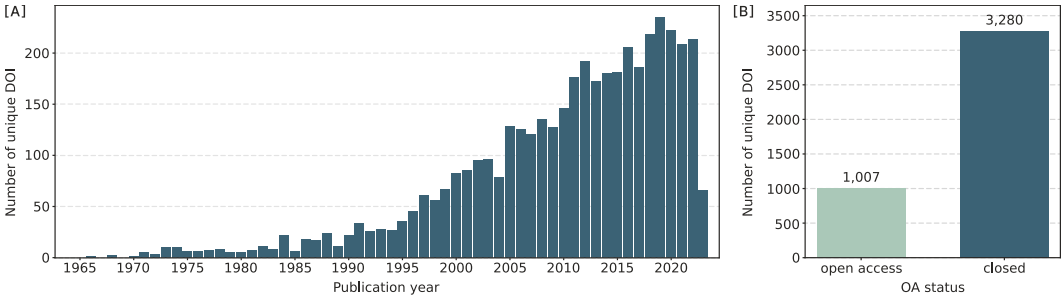

Figure 3: Source corpus statistics. [A] Publication-year distribution of the collected CSD-derived source DOIs. [B] Open-access status of the source corpus, summarizing the accessibility of articles used for full-text retrieval and extraction.

A.2 Structured representation of co-crystal synthesis methods

A.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

Continued on next page

Method	Description	Key parameters	Template
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
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

Continued on next page

Method	Description	Key parameters	Template
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
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

A.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].
```

A.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:

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

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:

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

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.

A.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.

A.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]
```

A.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 A.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 types 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
```

A.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 the 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.
```

A.5 Prompting Strategy Optimisation Results

Prompt variants and strategy descriptions are provided in Appendix A.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 A.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

A.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 4 summarizes the per-method results in terms of F1-score and accuracy.

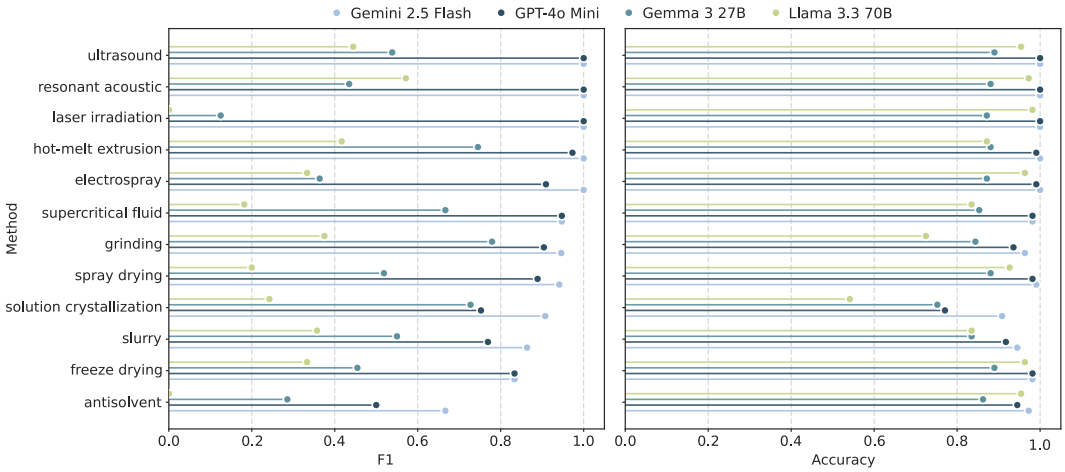

Figure 4: Per-method Stage 2 classification performance for the 12 co-crystal synthesis method types. Results are reported under a one-vs-rest evaluation setup using F1-score and accuracy across the evaluated LLMs.

A.7 Gemini-3.1-Pro Extraction Comparison

To assess whether extraction performance is limited by the selected backbone, we reran the finalized pipeline on the independent held-out set with `Gemini-3.1-Pro`, changing only the LLM backbone. All prompts, method-specific schemas, and evaluation procedures were kept fixed. Table 8 compares the results with the `Gemini-2.5-Flash` backbone used for corpus-scale extraction.

Table 8: Held-out extraction performance with `Gemini-2.5-Flash` and `Gemini-3.1-Pro` under the same extraction pipeline, prompts, schemas, and evaluation procedure. Δ denotes the difference between models.

	Gemini-2.5-Flash		**Gemini-3.1-Pro**		Δ	
	F1	**Accuracy**	**F1**	**Accuracy**	**F1**	**Accuracy**
Stage 1	0.96 $\pm$0.02	0.93 $\pm$0.03	0.99 $\pm$0.01	0.98 $\pm$0.02	+0.03	+0.05
Stage 2	0.94 $\pm$0.02	0.92 $\pm$0.02	0.97 $\pm$0.01	0.96 $\pm$0.02	+0.03	+0.04
Stage 3	0.72 $\pm$0.02	0.65 $\pm$0.02	0.75 $\pm$0.03	0.70 $\pm$0.04	+0.03	+0.05

`Gemini-3.1-Pro` improved both F1 and accuracy across all three stages, while the magnitude of the gains remained moderate. Its Stage 3 inference cost was approximately 15–17 times higher than that of `Gemini-2.5-Flash`. We therefore retain `Gemini-2.5-Flash` for corpus-scale extraction and treat the Pro comparison as a stronger-backbone diagnostic.

A.8 Full Extraction Backbone Comparison

Table 9 compares four LLM backbones across all three stages of the extraction pipeline using the corresponding development sets. Stage 1 primarily separated models by recall: `GPT-4o-mini` and `Llama-3.3-70B` achieved perfect precision but missed substantially more valid procedures, whereas `Gemini-2.5-Flash` maintained both high precision and recall (F1 = 0.97). Stage 2 was the most model-sensitive, with F1 ranging from 0.19 to 0.81, indicating substantial differences in how effectively models match procedure text against template-based method definitions. Stage 3 results were more closely clustered (F1 = 0.54–0.63), with many mismatches arising from the strict multiset-based evaluation rather than entirely incorrect extractions.

Because the Stage 1–2 prompting strategies were selected using `GPT-4o-mini`, this procedure may introduce a small optimistic bias for that model. Nevertheless, `Gemini-2.5-Flash` achieved the highest F1 and accuracy across all three stages and was therefore selected for full-corpus extraction. Per-parameter precision and recall for Stage 3 are reported separately in Appendix A.9.

Table 9: Full extraction-backbone comparison on the development sets. Precision, recall, F1, and accuracy are reported for procedure detection (Stage 1), method classification (Stage 2), and parameter extraction (Stage 3). **Bold** indicates the best mean within each stage, and † 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

A.9 Per-parameter extraction results

Figure 5 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.

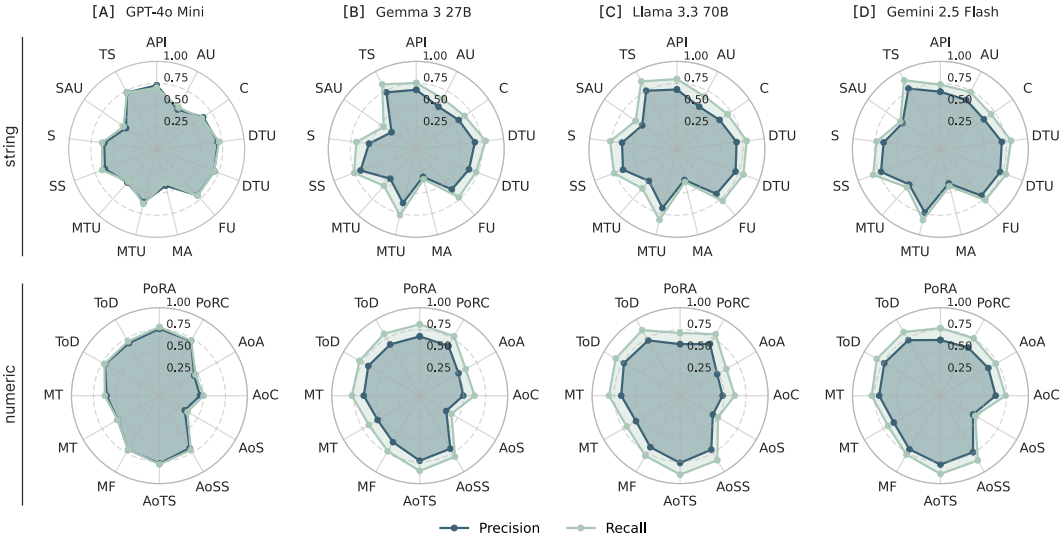

Figure 5: Per-parameter Stage 3 extraction performance on the LAG validation set. Precision and recall are shown separately for categorical and numerical parameters for [A] `GPT-4o-mini`, [B] `Gemma-3-27B`, [C] `Llama-3.3-70B`, and [D] `Gemini-2.5-Flash`.

A.10 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 10). 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 10: 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.

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.

A.11 RECORD-LEVEL DATABASE QUALITY

Field-level extraction metrics assess individual parameters but do not capture whether extracted information is assembled into coherent synthesis records. We therefore evaluated record-level quality on the independent held-out set described in Section 3.3. Predicted and gold records were matched one-to-one within the same article and synthesis method. Matched records were then compared across the active semantic blocks of the corresponding method-specific schema, including component identities, stoichiometry, quantities, solvents, and method-specific conditions. We report record identification and consistency among successfully matched records.

Table 11: Record-level extraction quality on the independent held-out set. Record identification measures recovery of synthesis records, while matched-record consistency measures agreement across active semantic blocks. Intervals are 95% CIs, except for median consistency, for which the IQR is reported.

Metric	**Estimate**	**Interval**
Record identification		
Gold records identified	158/207 (76.3%)	[64.1, 88.0]
Predictions matched to gold	158/228 (69.3%)	[57.7, 81.6]
Identification F1	0.726	[0.620, 0.824]
Matched-record consistency		
Exact match	40/158 (25.3%)	[14.3, 37.8]
Consistency ≥ 0.80	90/158 (57.0%)	[41.8, 72.1]
Consistency ≥ 0.70	126/158 (79.7%)	[68.9, 89.7]
Median consistency	0.833	[0.714, 0.972]

The pipeline identified 158 of 207 gold records while producing 228 predicted records, corresponding to an identification F1 of 0.726. Among matched records, the median consistency was 0.833, with 57.0% reaching at least 0.80 consistency and 79.7% reaching at least 0.70. Exact agreement across the full record was obtained for 25.3% of matched records. These results complement field-level extraction metrics by quantifying how well extracted parameters remain associated within individual synthesis records.

A.12 RARE-ROUTE EXTRACTION CASE STUDIES

The three dominant synthesis routes account for most records in COSYN-DB, while the remaining nine routes are too sparse to support reliable train–test evaluation. To examine whether the corresponding method-specific schemas can nevertheless be applied in practice, we manually annotated one article for each rare route and evaluated Stage 3 extraction using the same parameter-level evaluator. The schemas contain 22–32 parameters and the nine case studies comprise 36 gold and 39 predicted records.

Performance varied across routes, with the case studies primarily serving to verify that the route-specific schemas can represent and recover synthesis information outside the three dominant methods. The observed variation also reflects route-specific extraction challenges, including record granularity, component assignment, shared experimental conditions, and preservation of method-specific parameter combinations.

A.13 COMPARISON WITH EXISTING CO-CRYSTAL RESOURCES

Most existing machine-readable co-crystal resources are organized around molecular pairs and primarily support co-crystal screening rather than protocol-level synthesis analysis. More recent datasets increasingly include solvent identity or selected experimental variables, but these are typically restricted to a particular synthesis setting. In contrast, COSYN-DB links each molecular pair to record-specific synthesis conditions using method-aware schemas that capture stoichiometry, component and solvent quantities, apparatus, time, temperature, processing conditions, and post-processing when reported.

Table 12: Stage 3 extraction results for the nine rare synthesis routes. Each route is represented by one manually annotated article, so the results are descriptive case studies rather than per-route performance estimates.

Method	Gold	Predicted	Fields	Precision	Recall	F1	Accuracy
Supercritical fluid	3	3	26	0.63	0.63	0.63	0.61
Hot-melt extrusion	6	5	32	0.80	0.67	0.73	0.65
Spray drying	3	4	28	0.70	0.93	0.80	0.70
Ultrasound-assisted	1	3	28	0.23	0.68	0.34	0.23
Freeze drying	1	1	22	0.86	0.86	0.86	0.86
Electrospray	8	8	28	0.95	0.95	0.95	0.95
Resonant acoustic mixing	2	2	25	0.84	0.84	0.84	0.84
Antisolvent crystallization	9	9	22	0.82	0.82	0.82	0.82
Laser irradiation	3	4	27	0.50	0.67	0.57	0.50
Macro average				0.70	0.78	0.73	0.68

Table 13: Comparison of CoSyn-DB with existing machine-readable co-crystal resources. "Partial" denotes selected synthesis variables available within a specific experimental setting rather than a general method-aware representation of record-specific synthesis conditions. Sample counts follow the corresponding resources; for CSD, the reported count refers to the full database rather than a co-crystal-specific subset.

Source	No. of samples	Molecular pair	Solvent	Synthesis parameters	Multiple methods
Groom et al. (2016)	1,362,809	✓	–	–	–
Jiang et al. (2021)	7,871	✓	–	–	–
Gröls & Castro-Dominguez (2021)	1,000	✓	–	Partial	–
Vriza et al. (2022)	7,479	✓	–	–	–
Guo et al. (2024)	7,871	✓	–	–	–
Liu et al. (2025)	11,250	✓	✓	Partial	–
Birolo et al. (2025)	6,632	✓	–	–	–
Song et al. (2025b)	1,435	✓	✓	–	–
Zeng et al. (2025)	1,050	✓	✓	Partial	–
Ours	20,783	✓	✓	✓	✓

The comparison highlights a distinction in representation rather than sample count alone. Pair-oriented resources provide the molecular identities required for formation screening, while condition-aware datasets add solvent or selected variables for narrower experimental settings. CoSyn-DB instead represents complete synthesis records across multiple routes, preserving the association between a specific molecular pair, synthesis method, and its reported experimental conditions.

A.14 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 6. The full mapping is shown in Table 14.

A.15 Dataset Normalization

For the three synthesis routes used in downstream condition prediction, extracted records were processed using deterministic, method-specific normalization pipelines. The main steps were:

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

Table 14: 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

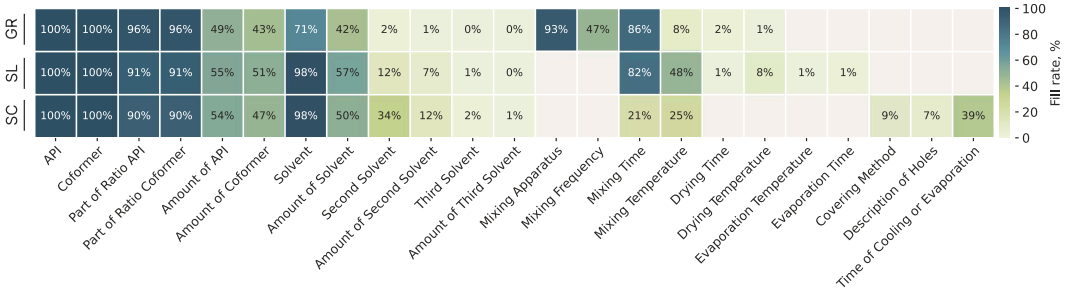

Figure 6: Per-parameter coverage after quality filtering. Fill rates are reported for individual synthesis parameters in the grinding, slurry, and solution crystallization datasets. Unit fields are omitted from the visualization. Structurally absent values are treated as empty, and blank cells denote parameters that are not defined for the corresponding synthesis method.

- **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 component 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. Omitted or chemically implausible ratios were assigned a latent 1:1 value when no evidence for a different stoichiometry was present. This is a domain-informed normalization rule rather than statistical imputation: 1:1 is the dominant API–coformer stoichiometry in the corpus and is frequently left implicit in experimental procedures. 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 then applied, such as assigning no solvent to neat grinding and no mixing frequency to mortar-and-pestle grinding.
- **Missingness markers.** `NOT_DETECTED` denotes a parameter expected for the corresponding procedure but not recovered from the article, whereas `NO_VALUE` denotes structural absence, such as solvent in neat grinding.
- **Quality filtering and deduplication.** Records with fewer than four populated synthesis-parameter fields were removed. Identifiers, compound fields, synthesis type, formula, DOI, and unit columns were excluded from this count. Compatible duplicate records were merged by retaining the most complete representation.

After normalization and quality filtering, the three datasets contained 8,272 solution-crystallization records, 2,722 grinding records, and 593 slurry records, for a total of 11,587 records.

B Synthesis Condition Prediction

B.1 Data Partitioning

For downstream prediction, records were partitioned at the level of connected components rather than individual synthesis records. Two records were linked when they shared either the same DOI or the same canonical API–coformer pair, and each connected component was assigned entirely to one split. This prevents both publication-level and molecular-pair leakage between training and test data.

For each synthesis method, test components were selected greedily until reaching 100 records, followed by a repair step that adjusted assignments without breaking components. Selection favored DOI diversity within the 100-record budget to broaden the range of independent literature sources in the test set. Consequently, test records are distributed across more distinct articles and have fewer records per DOI than the corresponding training data (Table 15). The resulting held-out benchmark contains 300 records, with 100 examples from each of the three synthesis methods.

Table 15: Leakage-safe partitioning statistics for condition prediction. Splits are constructed over connected components of shared DOIs and API–coformer pairs.

	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.2 Missing-Value Imputation

Missing-value imputation was used only to construct complete training records that may subsequently appear as retrieved few-shot examples. Values marked as `NOT_DETECTED` or left empty were treated as unknown and eligible for imputation, whereas structurally absent `NO_VALUE` fields were preserved. All imputation statistics were estimated exclusively from the training split. The 1:1 stoichiometry rule described in Appendix A.15 is part of deterministic normalization and is therefore not included in this statistical imputation procedure.

Pre-imputation test records were retained as evaluation references, so prediction metrics are computed only against originally observed values. Generated protocols are outputs of the predictor rather than records completed by the imputer. Thus, imputed values can affect prediction through the literature examples supplied to the model, but they do not replace held-out labels.

Categorical fields. Missing categorical values were filled through a hierarchy of increasingly broad training-set modes. The cascade first used the joint API–coformer pair, then API alone, then coformer alone, followed by the global training mode and a fixed fallback when necessary.

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.

B.3 Observed-Only Imputation Ablation

To quantify the downstream effect of imputation, we repeated the grinding condition-prediction experiment using training records without statistical imputation. Test references remained unchanged in both settings and contained only originally observed values. We compare this observed-only setting with the standard imputed setting for the frequency prior and `Gemini-2.5-Flash`.

Table 16: Effect of training-record imputation on grinding condition prediction. Values report F1 under the standard imputed and observed-only settings.

Model	Setting	Solvent	Ratio	Categorical	Numerical	Overall
Frequency prior	With imputation	0.17	0.87	0.26	0.38	0.35
Frequency prior	Observed only	0.29	0.87	0.32	0.28	0.29
Gemini-2.5-Flash	With imputation	0.41	0.85	0.49	0.48	0.49
Gemini-2.5-Flash	Observed only	0.36	0.89	0.45	0.45	0.45

Without imputation, Overall F1 decreased from 0.35 to 0.29 for the frequency prior and from 0.49 to 0.45 for `Gemini-2.5-Flash`. Numerical F1 also decreased for both models, while the effects on solvent and ratio prediction were mixed. These results indicate that imputation improves the completeness of the evidence available in retrieved examples without replacing observed evaluation labels or directly filling generated protocols.

B.4 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).

B.5 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]
```

B.6 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.

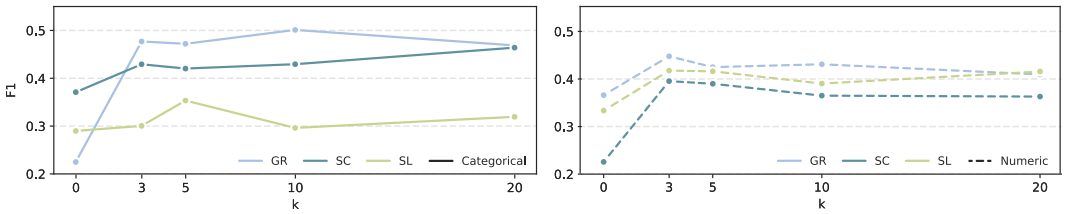

Figure 7: Effect of the number of retrieved in-context examples on synthesis-condition prediction. Weighted F1 score is reported for `GPT-4o-mini` as k varies across grinding, slurry, and solution crystallization, with categorical and numerical aggregates shown separately.

B.7 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 17 lists the families with their descriptions and the number of canonical solvents covered.

B.8 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 17: 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

B.9 Solvent Confusion Analysis

Figure 8 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.

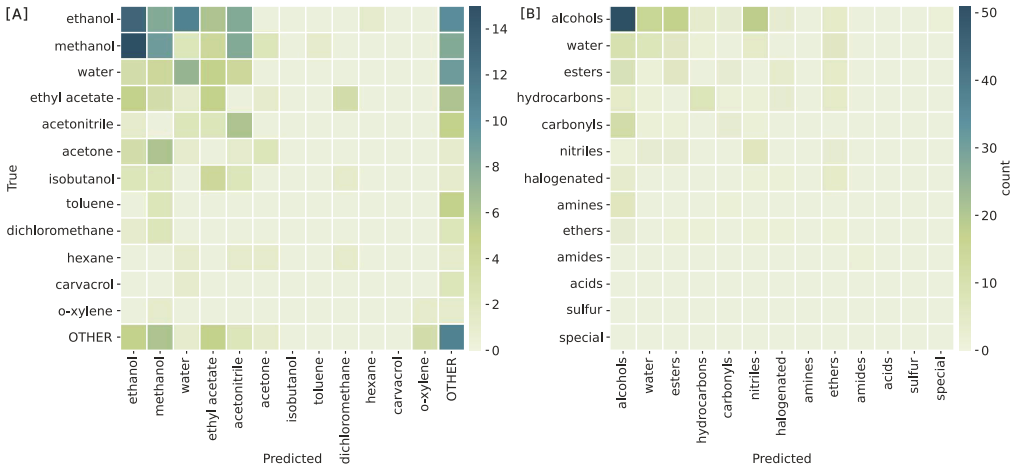

Figure 8: Solvent confusion analysis. [A] Exact-label confusion matrix for the 12 most frequent solvents, with all remaining labels grouped as `OTHER`. [B] Confusion matrix after aggregation into solvent families.

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.

B.10 Per-Method Prediction Results

Tables 18–20 report per-parameter Accuracy and weighted F1 for `Gemini-2.5-Flash` (k=3) across two independent runs. Reported values are mean $\pm$ 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 18: 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 19: 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

Table 20: 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

B.11 Extraction-Error Propagation

Condition prediction in CoSyn relies on automatically extracted literature records, so extraction errors can affect the downstream system in two ways: by changing which literature precedents are retrieved and by altering the parameter content supplied in those precedents. We evaluated both pathways using the held-out grinding set, for which manually annotated and automatically extracted versions of the same synthesis records were available. Both versions were processed using the same normalization, retrieval, and condition-prediction pipeline as in Section 4.

Controlled decomposition. We independently varied the records used for retrieval and the parameter content supplied in the retrieved demonstrations. We denote manually annotated records as Gold (G) and automatically extracted records as Extracted (E), yielding four conditions: GG, EG, GE, and EE. For example, EG retrieves neighbors from the extracted database but supplies the corresponding gold record content to the predictor, while GE retains gold retrieval but replaces the retrieved content with its extracted counterpart. This decomposition requires one-to-one correspondence between gold and extracted records and was therefore evaluated on the largest common eligible subset of 38 queries from 10 DOIs over three independent seeds.

Table 21: Controlled extraction-error propagation on 38 held-out grinding queries. Gold (G) and Extracted (E) denote manually annotated and automatically extracted records, respectively. Record Overlap@3 measures preservation of gold-retrieval top-3 precedents.

Condition	Retrieval	Content	Record Overlap@3	Categorical F1	Numerical F1	Overall F1
GG	Gold	Gold	1.00	0.21	0.36	0.33
EG	Extracted	Gold	0.56	0.20	0.36	0.33
GE	Gold	Extracted	1.00	0.21	0.37	0.33
EE	Extracted	Extracted	0.56	0.21	0.36	0.33

Extraction substantially changed the retrieved evidence: only 56% of the gold top-3 precedents were preserved when retrieval operated over extracted records. In contrast, aggregate prediction performance remained nearly unchanged across the four conditions. Overall F1 was 0.33 in both the fully gold and fully extracted settings, with similarly small variation in categorical and numerical prediction scores.

End-to-end comparison. We next compared the fully gold and fully extracted pipelines on all 85 held-out grinding queries from 15 DOIs. This analysis does not require strict record correspondence

beyond the evaluation query and therefore covers the complete held-out grinding set. Extraction changed approximately half of the retrieved precedents, with a record-level Overlap@3 of 0.50.

Table 22: End-to-end comparison of prediction using manually annotated Gold (GG) and automatically Extracted (EE) records on all 85 held-out grinding queries. Δ is computed as GG$-$EE. The reported intervals are 95% DOI-clustered CIs.

Metric	GG	EE	Δ	Interval
Categorical F1	0.17	0.19	-0.02	$[-0.06, 0.03]$
Numerical F1	0.25	0.26	-0.01	$[-0.04, 0.03]$
Overall F1	0.23	0.24	-0.01	$[-0.03, 0.01]$

Despite the substantial change in retrieved precedents, aggregate prediction performance remained similar between the gold and extracted pipelines, with all confidence intervals spanning zero. Thus, extraction errors clearly propagate into the evidence supplied to the predictor, but we do not observe a systematic degradation in aggregate condition-prediction performance on this held-out grinding set. This result should not be interpreted as implying that extraction errors are inconsequential for individual recommendations. The analysis is restricted to one synthesis route, and broader coverage of Stage 1–2 failures, additional synthesis methods, and parameter-level effects remains future work.

B.12 Prospective Experimental Validation

We prospectively evaluated four literature-confirmed API–coformer pairs after removing all pair-specific records from CoSyn-DB. Slurry crystallization was fixed as the synthesis method, and CoSyn recommendations were generated from the three most similar retained literature examples. We compared these conditions with the pair-agnostic frequency prior.

B.12.1 Frequency prior

The prior selected ethyl acetate, which was unavailable experimentally and was replaced with 1,2-dimethoxyethane (DME) in all baseline experiments.

```
Slurry crystallization was conducted by suspending 1000 mg of API
and 200 mg of coformer in 2 mL of DME. The suspension was stirred
at 25°C for 24 h.  The solid product was separated by filtration,
and residual solvent was evaporated at 25°C for 72 h.  The resulting
product was dried for 24 h at 50°C.
```

B.12.2 CoSyn: Caffeine–oxalic acid

```
Slurry crystallization was conducted by suspending 388.4 mg of
caffeine and 118.1 mg of oxalic acid.  The solids were suspended in
1 mL of acetonitrile and 1 mL of water.  The suspension was stirred
at 25°C for 12 h.  The solid product was separated by filtration,
and residual solvent was evaporated at 25°C for 7 d.  The resulting
co-crystal was dried for 24 h at 45°C.
```

B.12.3 CoSyn: Caffeine–benzoic acid

```
Slurry crystallization was conducted by suspending 388.4 mg of
caffeine and 276.2 mg of benzoic acid.  The solids were suspended in
2 mL of acetonitrile.  The suspension was stirred at 25°C for 12 h.
The solid product was separated by filtration, and residual solvent
was evaporated at 25°C for 72 h.  The resulting co-crystal was dried
for 24 h at 45°C.
```

B.12.4 CoSyn: Urea–oxalic acid

```
Slurry crystallization was conducted by suspending 1000 mg of urea
and 200 mg of oxalic acid.  The solids were suspended in 2 mL of
water.  The suspension was stirred at 25°C for 48 h.  The solid
product was separated by filtration, and residual solvent was
evaporated at 25°C for 72 h.  The resulting co-crystal was dried for
24 h at 45°C.
```

B.12.5 CoSyn: Urea–salicylic acid

```
Slurry crystallization was conducted by suspending 100 mg of urea
and 115.1 mg of salicylic acid.  The solids were suspended in 2
mL of methanol.  The suspension was stirred at 25°C for 24 h.  The
solid product was separated by filtration, and residual solvent was
evaporated at 25°C for 72 h.  The resulting product was dried for 24
h at 50°C.
```

PXRD characterization. PXRD was used to assess phase formation for the three pairs that yielded phase-pure co-crystals under CoSyn conditions (Figure 9). The experimental patterns obtained with CoSyn reproduce the diagnostic reflections of the corresponding reference co-crystals without prominent residual reflections from the starting components. In contrast, the frequency-prior experiments retain clear starting-material signatures, consistent with the mixed or failed outcomes observed experimentally.

Across the four pairs, CoSyn yielded three phase-pure co-crystals, whereas the frequency prior yielded two mixed products and two failures. The prior also produced essentially the same pair-agnostic conditions across all four systems, while CoSyn adapted solvent choice, component loading, and processing time to the molecular pair. The PXRD results support the experimental actionability of these pair-specific recommendations, but also illustrate their current limits: urea–salicylic acid failed under both conditions, with complete dissolution under CoSyn and phase separation under the prior. Together with the small number of systems and the baseline solvent substitution, this motivates treating the experiment as an initial feasibility study rather than a controlled comparison of synthesis performance. The results support using CoSyn to propose literature-grounded starting conditions for subsequent assessment and refinement by an experimental chemist.

C Agentic System Details

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

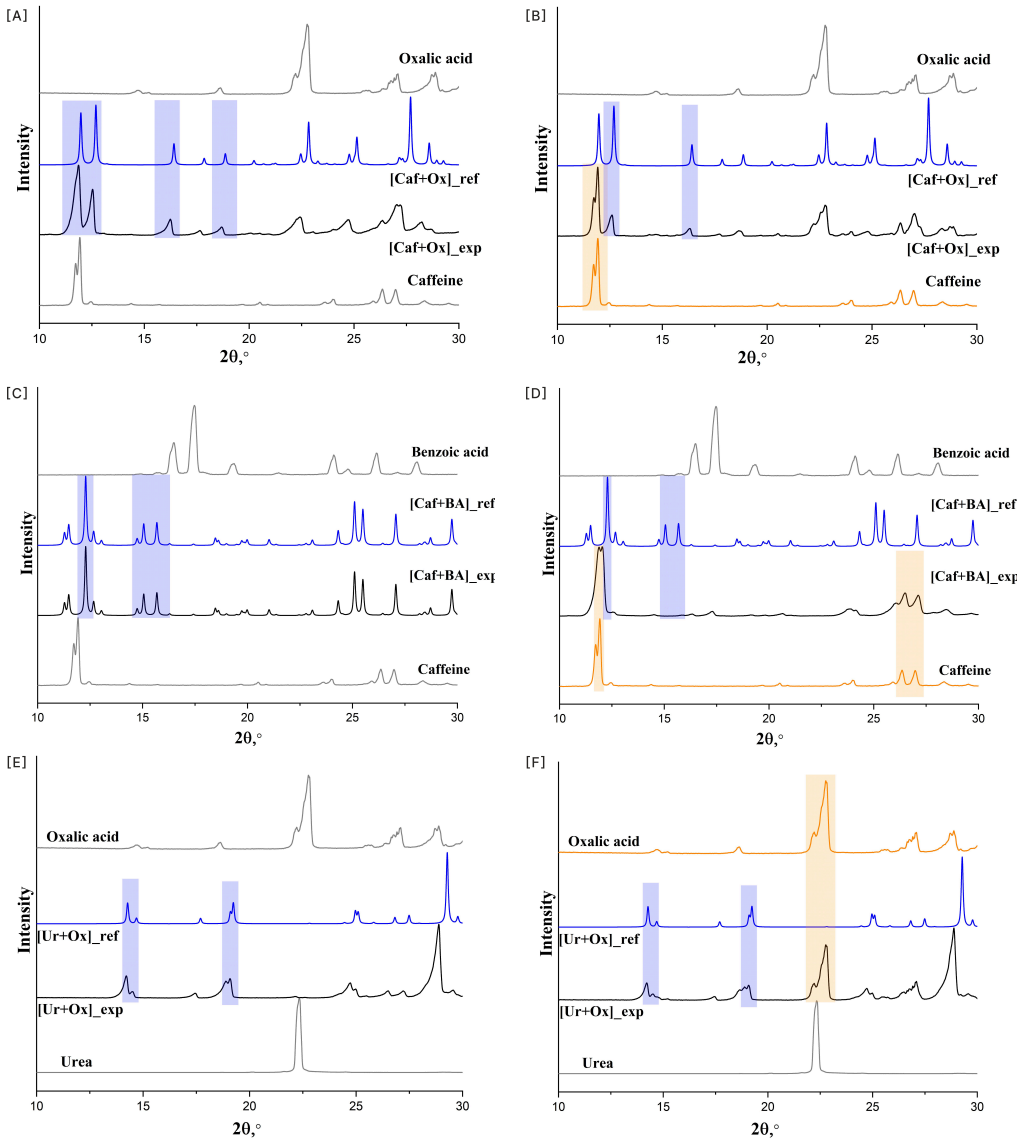

Figure 9: PXRD patterns for prospective slurry-crystallization experiments: caffeine–oxalic acid ([A], [B]), caffeine–benzoic acid ([C], [D]), and urea–oxalic acid ([E], [F]). Left panels show CoSyn conditions and right panels the frequency prior. Experimental products are shown in black, reference co-crystals in blue, and starting materials in gray or orange. Shaded regions highlight diagnostic co-crystal and residual starting-material reflections.

- **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.

C.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.

C.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.

C.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 23.

```
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 23: 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.

C.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 C.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.
```

C.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 24.

Table 24: 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.

C.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.

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 final",
  "final_answer": "final user-facing answer if next_agent is final,
      otherwise empty string",
  "reason": "short routing reason"
}
```

C.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 C.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 C.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 C.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]
```

C.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 solution crystallization method?",
  "gold_answer": "The five most commonly used solvents in solution
      crystallization are methanol, ethanol, water, acetonitrile, and
      acetone.",
  "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-ethoxybenzamide and 2,5-dihydroxybenzoic
       acid. Does the predicted method use the most common mixing
      apparatus for grinding?",
  "gold_answer": "Yes, the predicted method uses the most common mixing
      apparatus for grinding. The predicted synthesis method:
      Mechanochemical synthesis was carried out using a ball mill. The
      mixture was processed at 20.0 Hz for 30.0 min. A blend of 100.0 mg
       of CCOc1ccccc1C(N)=O and 93.3 mg of O=C(O)c1cc(O)ccc1O was
      prepared in a ratio of 1:1. Additionally, 0.1 mL of methanol was
      added. The mixture was then ground. The resulting co-crystal was
```

```
      dried. In the predicted method, a ball mill is used as the mixing
      apparatus. Based on analysis of the database, the ball mill is the
       most commonly used mixing apparatus for grinding.",
  "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."
    ]
  }
}
```


C.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 four 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.

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 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,
  "rationale": "short explanation"
}
```

C.6 Runtime and Resource Usage

Table 25 reports median runtime, call count, and token usage for each evaluated system. Brackets denote the interquartile range.

Table 25: 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]

C.7 Human Evaluation

To independently assess the LLM-based evaluation, two experts scored 60 responses from the reported CoSyn-Bench runs, comprising 30 Single-Agent and 30 Multi-Agent responses. Agreement was measured using Spearman's ρ for the same four evaluation criteria used by the LLM judge: Task completion, Correctness, Constraint adherence, and Groundedness. We compare agreement between `GPT-5.4` and Expert 1 with agreement between the two experts.

Table 26: Agreement between the LLM judge and Expert 1 and between two independent experts, reported as Spearman's ρ with 95% CIs.

Criterion / subset	N	LLM–Expert 1	Expert 1–Expert 2
Task completion	60	0.672 [0.485, 0.814]	0.713 [0.522, 0.854]
Correctness	60	0.779 [0.643, 0.871]	0.646 [0.426, 0.812]
Constraint adherence	60	0.696 [0.521, 0.823]	0.803 [0.681, 0.889]
Groundedness	60	0.686 [0.514, 0.816]	0.762 [0.628, 0.861]
Overall	60	0.768 [0.615, 0.867]	0.778 [0.610, 0.896]
Overall, Single-Agent	30	0.758 [0.491, 0.907]	0.723 [0.422, 0.912]
Overall, Multi-Agent	30	0.785 [0.540, 0.898]	0.838 [0.624, 0.946]

Overall LLM–expert agreement ($\rho = 0.768$) was close to inter-expert agreement ($\rho = 0.778$). The same pattern held when Single-Agent and Multi-Agent responses were considered separately, and criterion-level correlations were positive across all four dimensions. These results support the consistency of the LLM-based scoring used for CoSyn-Bench, while the benchmark itself remains a compact 30-task evaluation rather than a broad measure of general system capability.