ChemVD: Constructing Open-world Reaction Variable Spaces via Hierarchical Contrastive Alignment

Anonymous authors
Paper under double-blind review

Abstract

Reaction condition recommendation should construct compact, high-value variable subspaces from expanding candidate sets for more efficient experimental search. However, existing methods trained to recover historical conditions or rank predefined candidates struggle to generalize to unseen reagents under sparse comparative supervision, or to effectively model joint reaction–category–reagent dependencies. To address these limitations, we propose Chemical Variable-space Design (ChemVD), a hierarchical open-candidate retrieval framework for compact screening-space construction. ChemVD jointly conditions reagent retrieval on reactions and categories in a shared embedding space. Multi-positive contrastive learning learns from multiple annotated alternatives, while yield-aware supervision and chemical constraints guide rankings toward productive reaction conditions. Experiments show that ChemVD generalizes effectively to reagents unseen during training. On HTE-OOD, with a 20% screening budget, ChemVD achieves a mean achievable maximum yield of 68.78% using an 8B-parameter backbone, outperforming a 1.6T-parameter LLM baseline by 20.44 percentage points. Beyond reaction conditions, ChemVD provides a framework for turning large candidate libraries into compact search subspaces across diverse hierarchical recommendation tasks.

1 Introduction

The choice of reaction conditions directly affects the feasibility, efficiency, and selectivity of a synthetic transformation (Taylor et al., 2023; Sin et al., 2025). Experimental researchers must identify functional categories as variable dimensions and then select reagent subsets within each category (Torres et al., 2022; Han et al., 2025) to construct a combinatorial space under limited experimental budget. Similar role-conditioned selection occurs in agentic tools retrieval Zheng et al. (2024), AutoML pipeline construction He et al. (2021), and other recommendation systems. We therefore reformulate the problem as hierarchical candidate retrieval over open-candidate libraries for compact subspace construction. Given a task query x, a system identifies relevant categories C_x and selects candidate subsets $S_m \subseteq \mathcal{G}_m$ within each category. For candidate-space construction, the Cartesian product $\xi_x = \prod_{c_m \in C_x} S_m$ defines the downstream search space. Given observed data $\mathcal{D}$, the objective is to preserve high-utility configurations under budget B, while admitting multiple plausible candidates, including unseen entities. With downstream utility U, we write:

$$\underset{\{S_m\}}{\text{maximize}}\ \mathbb{E}\big[\max_{\mathbf{g}\in\xi_x} U(x,\mathbf{g}) \mid \mathcal{D}\big] \qquad \text{s.t.} \quad |\xi_x| \le B.$$

Existing reaction condition prediction methods, however, are not well aligned with this objective. Most data-driven approaches learn conditional probabilities $p_\theta(g|x, c_m)$ in historical records through classification Wang et al. (2025), autoregressive prediction Wang et al. (2023); Qian et al. (2023), or prioritize alternatives using candidate-ranking scores $s_\theta(x, c_m, g)$ (Wu et al., 2025; Yan et al., 2025b). Two challenges remain in using these predictions for screening-space construction. First, the prediction space is typically defined by condition entities observed during training, making it difficult to evaluate previously unseen reagents. Second, reaction databases cover only the selected, tested, and reported conditions while unrecorded candidates may still yield productive reactions (Boser et al., 2026; Ball et al., 2025). Supervision derived solely from historical observations cannot

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.

reliably distinguish between candidates that are chemically unsuitable or those that are merely unrecorded. Therefore, models may reflect historical selection and co-occurrence patterns rather than learn transferable reaction–reagent compatibility for practical variable-space design.

Addressing these limitations requires more than allowing unseen reagents into the candidate library. Candidate suitability depends jointly on the task and functional role (Taylor et al., 2023; Ai et al., 2024; Leonov et al., 2024; Torres et al., 2022), motivating explicit hierarchical modeling. This hierarchy provides intermediate semantics beyond candidate identity: the same candidate can have different suitability under different task and role contexts. A natural approach is to independently learn task–category, category–candidate, and task–candidate correspondences through pairwise alignment (Liu et al., 2023b; Girdhar et al., 2023) (Fig.1a). However, such pairwise objectives do not ensure a coherent joint conditional dependency. A candidate's similarity to a category or its match to a task may overlook its different suitability under different joint contexts. The required modeling is therefore an explicit representation of the hierarchical conditional relationship (Fig.1b). A reaction–category query may also have multiple acceptable reagents. Training should therefore accommodate all annotated alternatives, while observed yields provide additional guidance about utility value.

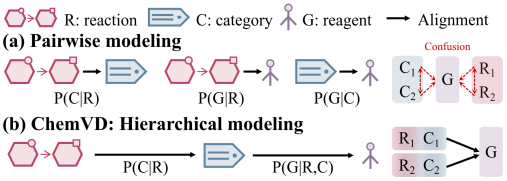

Figure 1: Pairwise and hierarchical modeling of reaction–condition relationships.

Based on this insight, we propose Chemical Variable-space Design (ChemVD), a structure-aware hierarchical cross-modal contrastive learning framework. ChemVD separately encodes reactions, functional categories, and candidate reagents into a shared embedding space for open-space retrieval. The task-aligned hierarchical framework explicitly models the joint dependency between functional categories and specific reagents conditioned on the reaction. It first retrieves the functional categories for the target reaction and then ranks candidate reagents with queries conditioned jointly on the category and reaction context. We train this retrieval framework using multi-positive contrastive learning, yield-aware supervision and chemically informed constraints that optimize candidates ranking in large unseen candidate pools. The resulting cartesian-product construction of the compact variable subspaces retains high utility combinations under low-budget settings. Our main contributions are as follows:

1. **We reformulate reaction condition recommendation as hierarchical open-candidate retrieval for compact subspace construction.** This formulation aligns recommendation with budget-constrained experimental screening by selecting category-wise candidate subsets from open-sapce reagent libraries, aiming to retain multiple promising combinations for evaluation.
2. **We introduce a hierarchical open-candidate retrieval framework.** ChemVD hierarchically aligns reaction representations, condition-category semantics, and reagent structures in a unified retrieval space. Theoretical analysis shows that the hierarchical framework consistently converges toward the theoretical optimum in controlled experiments. This advantage translates into better candidate rankings in both in-distribution (ID) and out-of-distribution (OOD) settings, together with a 58.1% improvement over the Pairwise baseline in category prediction.
3. **We optimize unseen candidate rankings using multi-positive contrastive learning and yield-aware supervision.** Multi-positive contrastive learning accommodates multiple annotated alternatives for each query, while yield-aware supervision uses experimental outcomes to emphasize high-utility candidates. On the HTE-OOD benchmark, ChemVD with an 8B-parameter backbone achieves a 20.44 percentage points superiority in the maximum yield over a 1.6T-parameter LLM at a 20% experimental budget.

2 Related Work

Data-driven reaction condition prediction commonly follows classification, autoregressive paradigms. Classification models map reaction representations to predefined condition labels using multi-task, multi-label, or structure-aware objectives (Gao et al., 2018; Maser et al., 2021; Wang et al., 2025; Yan et al., 2025a). Autoregressive models instead generate serialized condition components, as exemplified by Parrot and the patent-augmented TextReact (Wang et al., 2023; Qian

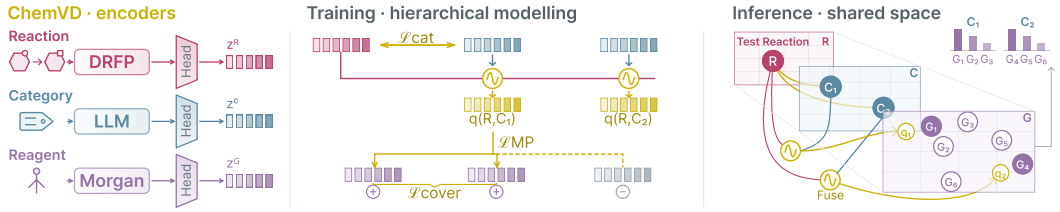

Figure 2: ChemVD for open-candidate variable space design. ChemVD represents reactions, categories and reagents with three encoders. During training, ChemVD learns reaction–category alignment and category-conditioned reagent retrieval. Hierarchical modeling retrieves categories and then ranking reagents with reaction–category query at inference time.

et al., 2023). For large-language-models-based methods, Chemma-RC integrates multiple modalities, while chemical agents incorporate retrieval and tools (Zhang et al., 2024; Chen et al., 2023; M. Bran et al., 2024; Boiko et al., 2023). However, extending these methods to expanding candidate libraries faces limitations from fixed output vocabularies, inconsistent candidate scoring, or the context length and inference cost of explicitly listing candidates. These limitations motivate shared embedding retrieval, which uses dot products with pre-encoded candidates to support library expansion and scalable search without listing the full pool in a generation prompt.

Learning an effective retrieval space requires objectives that align embedding similarity with candidate suitability. Contrastive learning proves effective in aligning multi-modal representation learning for an informative embedding space. Prior work has aligned molecular structures with text, reactions with patents, and reaction graphs with catalysts (Liu et al., 2023a; Qian et al., 2023; Shi et al., 2026). Supervised contrastive learning also accommodates utility labels into candidate preferences (Zhang et al., 2024). Beyond the training objective, retrieval also depends on how queries and candidates are related through intermediate categories. Yet existing reaction–reagent methods generally model a single functional role. Pooling different roles ignores role-dependent reagent applicability, whereas separate models restrict knowledge transfer across roles. ChemVD instead formulates condition recommendation as budget-constrained retrieval and ranking over an open-candidate pool. It pre-encodes candidates and introduces functional categories to jointly model reaction–category–reagent compatibility.

3 Method

3.1 Task Formulation

We formulate reaction condition recommendation as budget-constrained variable-space design over an open candidate set. Each reaction is represented by a reaction simplified molecular input line entry system (SMILES) string $x_i = (\mathcal{R}_i, \mathcal{P}_i)$, where $\mathcal{R}_i$ and $\mathcal{P}_i$ denote its reactants and products. The objective is to rank candidate reagents and construct a high-value condition subspace under a finite experimental budget. Here, the budget specifies the proportion of reagents retained in each condition category. The candidate space is organized hierarchically. Let $C = \{c_1, \ldots, c_M\}$ denote the set of M condition categories. For category c_m, its candidate reagent set is $\mathcal{G}_m = \{g_{m,1}, \ldots, g_{m,N_m}\}$.

A reaction may require multiple condition categories, and multiple reagents may be suitable within each category. The supervision for sample i is represented by positive categories $C_i^+ \subseteq C$, positive reagents $\mathcal{G}_{i,m}^+ \subseteq \mathcal{G}_m$, and optionally explicit negative reagents $\mathcal{G}_{i,m}^-$ for each $c_m \in C_i^+$. During training, the category-specific candidate pool is formed from training samples as $\mathcal{G}_m = \bigcup_{i:c_m \in C_i^+} (\mathcal{G}_{i,m}^+ \cup \mathcal{G}_{i,m}^-)$. At inference time, $\mathcal{G}_m$ can be dynamically replaced or extended by an external candidate library, where any unseen molecule with a valid structure can be evaluated without retraining. The model learns the hierarchical mapping

$$x_i \longrightarrow \widehat{C}_i \longrightarrow \{\widehat{\mathcal{G}}_{i,m} : c_m \in \widehat{C}_i\}, \tag{1}$$

3.2 Hierarchical Contrastive Alignment Overview

The formulation above motivates three design requirements: unseen candidates must remain scorable, conditions must be retrieved hierarchically, and multiple valid answers must be preserved during training. To address these requirements, ChemVD employs three entity-specific representation encoding, two-stage hierarchical retrieval, and multi-positive contrastive learning as summarized in Fig. 2.

Three representation branches. ChemVD first encodes reactions, categories and reagents through three separate branches. During training, it aligns reactions with their relevant categories and constructs category-conditioned queries to learn from valid reagents. Specifically, the reaction branch uses a difference reaction fingerprint (DRFP) (Probst et al., 2022) for its emphasis on the structural transformations between reactants and products. The category branch uses a frozen LLaMA 3.1-8B language model (Grattafiori et al., 2024) to incorporate semantic information about category inclusion and exclusion, thereby delineating category boundaries more explicitly. The reagent branch uses a Morgan fingerprint (Rogers & Hahn, 2010) to represent the molecular structure of each reagent. Frozen feature extractors and trainable projection heads map all features to a shared d-dimensional unit hypersphere. All representations are L_2-normalized.

Stage 1: Reaction-to-Category Retrieval. Given a reaction x_i, the reaction embedding first retrieves the required functional categories by computing $s^c_{i,m} = (\mathbf{z}^r_i)^\top \mathbf{z}^c_m$. This stage identifies the functional roles relevant to the reaction without directly selecting individual reagents. At inference, categories are selected by top-K_c ranking or a score threshold τ_c.

Stage 2: Category-Conditioned Reagent Retrieval. For each selected category c_m, ChemVD constructs a category-conditioned query by jointly encoding the reaction and category representations, and then ranks its constituent reagents:

$$\mathbf{q}_{i,m} = f_{\text{fus}}([\mathbf{z}^r_i; \mathbf{z}^c_m; \mathbf{z}^r_i \odot \mathbf{z}^c_m]), \qquad s^g_{i,m,n} = \mathbf{q}^\top_{i,m} \mathbf{z}^g_{m,n}, \tag{2}$$

where the operators $[\cdot;\cdot]$ and $\odot$ denote concatenation and element-wise multiplication. The top-K_g reagents are returned for each selected category. Details are provided in Appendix A.2.1.

The resulting query therefore represents reagent suitability in the context of both the reaction and its functional category, rather than through a direct reaction–reagent match. At inference time, $\mathcal{G}_m$ can be dynamically replaced or extended by an external candidate library, allowing previously unseen reagents with valid structures to be scored without retraining. ChemVD first identifies the required categories and then ranks candidate reagents within each selected category during inference.

3.3 Multi-Positive Learning with Yield-Aware Supervision

3.3.1 Multi-Positive Contrastive Objective

In real-world, one reaction may be associated with multiple observed categories and one reaction-category query may likewise have multiple observed suitable reagents. A standard single-positive contrastive objective would treat alternative valid conditions as mutually exclusive. Therefore, we adopt a multi-positive contrastive loss. For query i, let $\mathcal{A}_i$ be the candidate set, $\mathcal{P}_i \subseteq \mathcal{A}_i$ the positive set, s_{ij} the matching score, and τ the temperature. The per-query loss is

$$\ell^{\text{MP}}_i = -\log \frac{\sum_{j\in\mathcal{P}_i} \exp(s_{ij}/\tau)}{\sum_{k\in\mathcal{A}_i} \exp(s_{ik}/\tau)}. \tag{3}$$

The aggregate loss over all valid queries in a batch is denoted as $\mathcal{L}_{\text{MP}}$. Let w^+_{ij} denote the weight of positive sample j. In the implementation, the sum of the positive weights in a row, $\omega_i = \sum_{j\in\mathcal{P}_i} w^+_{ij}$, is used as the query weight, giving

$$\mathcal{L}_{\text{MP}} = \frac{\sum_{i\in\mathcal{I}_{\text{valid}}} \omega_i \ell^{\text{MP}}_i}{\sum_{i\in\mathcal{I}_{\text{valid}}} \omega_i}, \tag{4}$$

where $\mathcal{I}_{\text{valid}}$ denotes the set of queries containing at least one valid positive sample. We apply the general multi-positive objective at two levels:

$$\mathcal{L}_{\text{cat}} = \mathcal{L}_{\text{MP}}(\{s^c_{i,m}\}, C^+_i, C; \tau_{\text{cat}}), \qquad \mathcal{L}^{\text{MP}}_{\text{reag}} = \mathcal{L}_{\text{MP}}(\{s^g_{i,m,n}\}, \mathcal{G}^+_{i,m}, \widetilde{\mathcal{G}}^{\mathcal{B}}_m; \tau_{\text{reag}}). \tag{5}$$

For the reagent-level loss, each positive reaction–category pair (i, m) is contrasted only against candidates from its corresponding category, with $\widetilde{\mathcal{G}}_m^{\mathcal{B}} \subseteq \mathcal{G}_m$ comprising observed positives and additional sampled contrasts.

3.3.2 Positive-Reagent Coverage

Multi-positive loss aggregates the scores of all positive samples, but a small number of high-scoring positives may therefore mask low-scoring positives in the same set. To provide direct supervision for every annotated positive reagent, we introduce an additional positive-reagent coverage loss:

$$\ell_{i,m,n}^{\text{cover}} = -\log \frac{\exp(s_{i,m,n}^{g}/\tau_{\text{reag}})}{\sum_{g_{m,k} \in \widetilde{\mathcal{G}}_m^{\mathcal{B}}} \exp(s_{i,m,k}^{g}/\tau_{\text{reag}})}, \qquad g_{m,n} \in \mathcal{G}_{i,m}^{+}. \tag{6}$$

The weighted average over all such instances in a batch is denoted as $\mathcal{L}_{\text{cover}}$. Similar to $\mathcal{L}_{\text{MP}}$, the weighted coverage loss is

$$\mathcal{L}_{\text{cover}} = \frac{\sum_{(i,m)} \sum_{n: g_{m,n} \in \mathcal{G}_{i,m}^{+}} w_{i,m,n}^{+} \ell_{i,m,n}^{\text{cover}}}{\sum_{(i,m)} \sum_{n: g_{m,n} \in \mathcal{G}_{i,m}^{+}} w_{i,m,n}^{+}}. \tag{7}$$

Therefore, the overall reagent-level objective is defined as $\mathcal{L}_{\text{reag}} = \mathcal{L}_{\text{reag}}^{\text{MP}} + \gamma \mathcal{L}_{\text{cover}}$. The two objectives are complementary rather than redundant. The multi-positive loss provides set-level supervision by treating all annotated reagents as jointly valid positives, whereas the coverage loss imposes an instance-level constraint that prevents a few dominant positives from obscuring other valid reagents. Using the coverage loss alone would require each positive to compete independently against the candidate set and would not explicitly preserve the joint multi-positive structure.

3.3.3 Yield-aware Positive Weighting

In experimental screening, observed positive conditions are not equally informative, ranging from low- to high-yield outcomes. We therefore use normalized yield $y_i \in [0, 1]$ as a graded reliability signal to emphasize high-yield conditions and downweight unpromising ones. Let y_{low} and y_{high} denote the low- and high-yield thresholds, respectively, and let $w_y^{\max}$ denote the maximum positive-sample weight. We define the yield weight as $w_i^y = 1$ if y_i is unknown, 0 if $y_i < y_{\text{low}}$, $w_y^{\max}$ if $y_i \geq y_{\text{high}}$, and $1 + \frac{y_i - y_{\text{low}}}{y_{\text{high}} - y_{\text{low}}}(w_y^{\max} - 1)$ for $y_{\text{low}} \leq y_i < y_{\text{high}}$. Low-yield records therefore do not enter the category-level or reagent-level multi-positive contrastive objectives as strong positives, whereas medium- and high-yield conditions receive progressively greater weight as yield increases. The positive category and reagent weights are $w_{i,m}^{+} = w_m^c w_i^y$ and $w_{i,m,n}^{+} = w_i^y$. The threshold values and maximum weight are reported in Appendix A.4.3.

3.4 Chemically Informed Training Constraints

Multi-positive and yield-aware training address label plurality and positive quality, but not biases inherent in sparse reaction records. These biases reflect experimental selection and publication processes rather than exhaustive chemical validity. ChemVD incorporates several chemically informed training regularizers. Specifically, we (i) treat low-yield observations as weak negative evidence ($\mathcal{L}_{\text{low}}$), (ii) use dataset provided explicit negative pairs ($\mathcal{L}_{\text{neg}}$), (iii) reweight category-level supervision to mitigate distribution imbalance of functional-category (w_m^c), and (iv) select high-frequency reagents within each category as hard negatives ($\text{TopFreq}_{K_{\text{hard}}}$) to migrate preference on historically prevalent reagents. These priors affect only the training objective and are independent of the hierarchical retrieval architecture. Details are provided in Appendix A.2.2.

The complete training objective is $\mathcal{L} = \lambda_{\text{cat}}\mathcal{L}_{\text{cat}} + \lambda_{\text{reag}}\mathcal{L}_{\text{reag}} + \lambda_{\text{low}}\mathcal{L}_{\text{low}} + \lambda_{\text{neg}}\mathcal{L}_{\text{neg}}$. Together, these components optimize hierarchical retrieval while accounting for the multi-solution and incompletely observed nature of reaction conditions.

Table 1: Reagent retrieval performance on Reagent-Split. Fixed-vocabulary models cannot rank unseen reagents and are marked as N/A.

Model	Seen reagents (N = 30,851)								Unseen reagents (N = 1,546)							
	Ranking quality			wAPC@K (%)					Ranking quality			wAPC@K (%)				
	wMRR ↑	wMPR ↓	wAPP ↑	@1 ↑	@3 ↑	@5 ↑	@10 ↑	@20 ↑	wMRR ↑	wMPR ↓	wAPP ↑	@1 ↑	@3 ↑	@5 ↑	@10 ↑	@20 ↑
ReaCon	$17.09_{\pm 0.04}$	$39.0698_{\pm 0.0016}$	$88.10_{\pm 0.00}$	$8.97_{\pm 0.06}$	$12.95_{\pm 0.04}$	$24.23_{\pm 0.00}$	$31.10_{\pm 0.00}$	$38.08_{\pm 0.00}$								
Label-mix	$50.86_{\pm 0.62}$	$16.8131_{\pm 0.0622}$	$91.16_{\pm 0.14}$	$27.66_{\pm 0.11}$	$45.37_{\pm 1.66}$	$55.03_{\pm 1.47}$	$64.50_{\pm 0.91}$	$75.00_{\pm 0.32}$								
Parrot	$53.99_{\pm 0.53}$	$9.2390_{\pm 0.1841}$	$98.81_{\pm 0.00}$	$35.35_{\pm 0.36}$	$55.21_{\pm 0.61}$	$65.58_{\pm 0.87}$	$80.35_{\pm 0.42}$	$90.93_{\pm 0.28}$				N/A				
Text-React	$59.40_{\pm 0.91}$	$8.3370_{\pm 0.1735}$	$98.92_{\pm 0.02}$	$40.81_{\pm 0.91}$	$59.52_{\pm 1.20}$	$69.28_{\pm 1.04}$	$82.53_{\pm 0.61}$	$91.27_{\pm 0.32}$								
Train-frequency baseline	$55.11_{\pm 0.00}$	$9.6464_{\pm 0.0000}$	$94.63_{\pm 0.00}$	$30.03_{\pm 0.00}$	$53.23_{\pm 0.00}$	$64.03_{\pm 0.00}$	$80.16_{\pm 0.00}$	$89.28_{\pm 0.00}$								
BCE	$2.77_{\pm 0.69}$	$143.3613_{\pm 6.7252}$	$66.14_{\pm 37.94}$	$0.02_{\pm 0.01}$	$2.63_{\pm 4.12}$	$5.15_{\pm 4.22}$	$8.31_{\pm 0.27}$	$9.39_{\pm 0.37}$	$3.09_{\pm 0.49}$	$148.0613_{\pm 4.2588}$	$65.96_{\pm 41.64}$	$0.27_{\pm 0.08}$	$2.34_{\pm 2.82}$	$4.39_{\pm 2.87}$	$7.10_{\pm 0.48}$	$9.17_{\pm 0.49}$
Pairwise	$41.56_{\pm 0.45}$	$13.5354_{\pm 0.2975}$	$97.46_{\pm 0.14}$	$18.42_{\pm 0.49}$	$45.08_{\pm 0.18}$	$58.03_{\pm 0.91}$	$70.61_{\pm 0.99}$	$81.87_{\pm 1.06}$	$30.19_{\pm 0.50}$	$53.3645_{\pm 0.7330}$	$91.94_{\pm 0.62}$	$11.21_{\pm 0.41}$	$31.49_{\pm 0.22}$	$39.72_{\pm 0.60}$	$48.98_{\pm 1.05}$	$59.68_{\pm 0.92}$
ChemVD	$\mathbf{65.09}_{\pm 0.16}$	$\mathbf{7.7210}_{\pm 0.0702}$	$\mathbf{99.40}_{\pm 0.07}$	$\mathbf{41.37}_{\pm 0.08}$	$\mathbf{63.63}_{\pm 0.13}$	$\mathbf{73.53}_{\pm 0.30}$	$\mathbf{85.01}_{\pm 0.36}$	$\mathbf{91.69}_{\pm 0.23}$	$\mathbf{48.80}_{\pm 0.43}$	$\mathbf{31.8061}_{\pm 1.1246}$	$\mathbf{96.08}_{\pm 0.47}$	$\mathbf{27.52}_{\pm 0.43}$	$\mathbf{45.78}_{\pm 0.77}$	$\mathbf{53.18}_{\pm 0.83}$	$\mathbf{65.00}_{\pm 0.31}$	$\mathbf{73.16}_{\pm 0.60}$

4 Experiments

4.1 Candidate Retrieval and Variable-Space Design in an Open Reagent Space

Practical variable-space design requires a model to retrieve compatible reagents, generalize beyond the training vocabulary, and exclude chemically infeasible candidates under a limited screening budget. We therefore designed two complementary benchmarks that differ in evaluation unit and data sources. Reagent-Split was curated from ChemRANG (Cai et al., 2026) using only valid reactions, reagents with functional roles, and percentage yields from the same data source as the training set; it contains 9,690 Reagent-Seen and 402 Reagent-Unseen examples, supporting reagent-level retrieval evaluation. HTE-OOD builds on complete high-throughput measurements across the expert-defined Suzuki design space reported by Perera et al. (2018), supporting evaluation of designed subspaces. Its data source is independent of the training set, allowing assessment of extrapolation to an external experimental space. For each of the 15 subsets, we add 24 mechanistically infeasible catalysts absent from the training set. This expansion increases the candidate space from 384 to 9,600 conditions. Appendix A.3 details the construction and statistics. All experiments use three random seeds, with means and standard deviations reported.

4.1.1 Performance in an Open Candidate Space

We evaluated reagent-level generalization on Reagent-Seen where all reagents occurred during training, and Reagent-Unseen where each reaction contained at least one unseen reagent. Table 1 reports the resulting retrieval performance across both subsets. Definitions of the evaluation metrics are provided in Appendix A.4.2. On Reagent-Seen, the closed-set classification models remained strongly influenced by the training distribution. ReaCon and Label-mix achieved wMRR of 17.09 and 50.86, respectively, both below the training-frequency baseline of 55.11. Consistent with our analysis in Appendix A.1.2, the closed-set objective fits the historical label distribution rather than transferable reaction–reagent compatibility.

Autoregressive generation reduced this dependence on the training distribution. However, the strongest autoregressive baseline, Text-React, achieved a wMRR of 59.40, only 4.29 points above the frequency baseline. ChemVD instead achieved the highest wMRR (65.09), lowest wMPR (7.7210), highest wAPP (99.40%), and highest wAPC@K at every cutoff.

The limitation of fixed-vocabulary models became especially explicit on the Reagent-Unseen subset. Fixed-vocabulary models cannot rank ground-truth candidates that carry no trained output label. ChemVD decoupled the inference-time candidates from the training label set and achieved an wMRR of 48.80. However, candidate-space openness alone did not fully explain this result. BCE and Pairwise also supported arbitrary inference-time candidates but performed substantially worse: BCE achieved wMRR values of 2.77 and 3.09 on Reagent-Seen and Reagent-Unseen, whereas Pairwise achieved 41.56 and 30.19. These comparisons reveal distinct limitations in the baselines. BCE lacks the relative candidate discrimination provided by contrastive learning. Pairwise lacks hierarchical reaction–category–reagent modeling. ChemVD addresses both limitations and generalizes effectively beyond the training reagent vocabulary.

These retrieval metrics do not establish whether better rankings yield useful experimental designs. We therefore evaluated whether ChemVD could identify high-yield conditions under a limited screening budget.

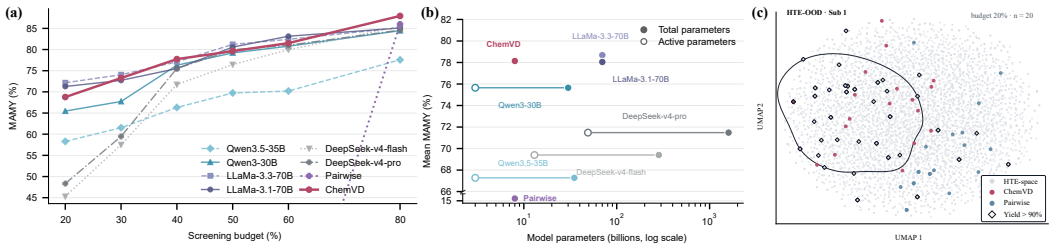

Figure 3: HTE-OOD subspace-design performance. (a) MAMY across screening budgets on 15 reaction subsets. (b) Average MAMY versus model size. (c) UMAP visualization of the conditions selected by ChemVD and Pairwise at a 20% budget.

4.1.2 High-Throughput Experimental Subspace Design

We next simulated variable-space screening across the 15 HTE-OOD subsets. For each subset and screening budget, we selected the top-ranked reagent candidates in each category and enumerated their combinations. We measured subspace quality using mean achievable maximum yield (MAMY). The budget specifies the retained proportion of reagents per category detailed in Appendix A.4.2.

As illustrated in Figure 3(a) and (b), ChemVD exhibits exceptional parameter efficiency. Across budgets, 8B-parameter ChemVD attained an average MAMY of 78.14%. It matched both 70B Llama baselines (78.68% and 78.04%) and surpassed the 1.6T DeepSeek-v4-pro by 6.7 points (71.48%). ChemVD remained competitive at a 20% budget (68.78%) and attained the highest MAMY of all methods at an 80% budget (87.93%). These results highlight that ChemVD can obtain screening utility comparable to, or even exceeding, massive frontier general-purpose LLMs without relying on brute-force parameter scaling. Detailed results are available in Appendix A.5.3.

Pairwise achieved an average MAMY of only 15.27%, compared with 78.14% for ChemVD. This indicates that ranking models can retrieve plausible individual reagents, whereas hierarchical design is critical for constructing a compact subspace of compatible and high-yield combinations. Consistent with this, uniform manifold approximation and projection (UMAP) embeddings at a 20% budget showed ChemVD selections concentrating in high-yield regions, whereas Pairwise covered fewer productive combinations as shown in Figure 3(c). Visualization of all UMAP analysis is detailed in Appendix A.5.2. This motivates the next question: whether hierarchical dependency modeling explains the difference between Pairwise and ChemVD.

4.2 Hierarchical Decomposition Improves Candidate-Ranking Quality

The preceding results exposed a consistent gap between Pairwise and ChemVD in both reagent retrieval and experimental-subspace design. To enable a direct architectural comparison, we removed all chemically informed constraints from ChemVD, retaining only the contrastive objective and yield-aware positive weighting. This simplified variant is referred as Hierarchical. We hypothesized that hierarchical architecture captures higher-order compatibility patterns that are lost when the three associations are optimized independently. We tested this hypothesis through theoretical analysis, a controlled experiment, and two evaluations in the real chemical space.

4.2.1 Distribution-Level Experiment

To isolate differences in representing higher-order dependencies, we designed a minimal experiment independent of chemical features and candidate-space openness. Appendix A.1.5 shows that Pairwise incurs an irreducible error when the target distribution contains a ternary dependency, whereas the Hierarchical objective can represent that dependency. We tested this using Independent and XOR distributions over three binary variables, R, C, and G. In both distributions, R and C were independent Bernoulli variables with parameter 0.5. The variable G was also sampled independently in Independent but defined as $G = R \oplus C$ in XOR. These distributions shown in Figure 4(a) have identical pairwise marginals but different ternary dependencies.

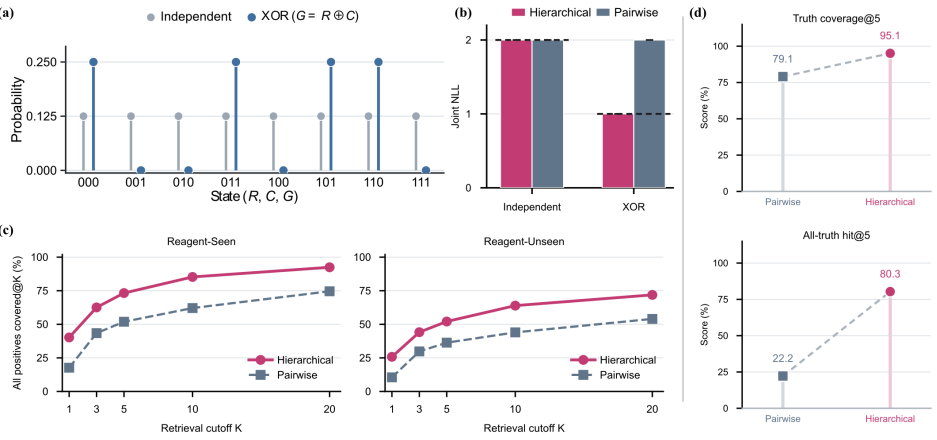

Figure 4: Pairwise versus hierarchical modeling. (a) Independent and XOR distributions with identical pairwise marginals. (b) Performance on the controlled distributions. (c) Reagent retrieval on Reagent-Split. (d) Category retrieval with limited training-set overlap.

On Independent, both architectures attained a joint negative log-likelihood (Joint NLL) of 2 bits (Figure 4(b)). Thus, hierarchical modeling conferred no advantage when the target contained no dependency beyond pairwise statistics.

The two architectures diverged sharply when evaluated on XOR. As illustrated in Figure 4(b), Hierarchical reached the theoretical optimum of 1 bit, whereas Pairwise remained at 2 bits Joint NLL.. The observed 1-bit gap matched the structural error derived in Appendix A.1.5. Despite identical pairwise marginals, the independent scores could not recover the XOR dependency, whereas hierarchical conditioning recovered it exactly. The controlled comparison therefore isolates higher-order dependency modeling as an architectural advantage rather than a consequence of chemical representations or candidate-space access.

4.2.2 Reagent-Level Evaluation

We next tested whether the controlled result transferred to a real reagent space. Pairwise ranked candidates by averaging its reaction–reagent and category–reagent scores. Hierarchical instead used the reagent distribution conditioned jointly on the reaction and category. In Figure 4(c) Hierarchical achieved higher wAPC@K at every cutoff on both Reagent-Seen and Reagent-Unseen. The consistent improvement across in-distribution and out-of-distribution subsets shows that introducing the category as an explicit intermediate variable for hierarchical modeling enhances category-conditioned reagent retrieval. This allowing relevant reagents to be prioritized within the appropriate functional context, including reagents absent from training.

4.2.3 Category-Level Evaluation

Finally, we tested whether hierarchical training also improved the upstream reaction-to-category decision. Both models ranked categories using only $q(c \mid r)$, without performing reagent-level inference. Their difference therefore reflects how the respective objectives shaped category representations rather than any downstream reagent score.

To limit training-set overlap, we evaluated 117 reactions whose maximum chemical reaction fingerprints RXNFP cosine similarity to any training reaction was below 0.9 (detailed in Appendix A.3). Evaluation metrics definition are provided in Appendix A.4.2. Hierarchical increased Truth coverage@5 from 79.1% to 95.1% and All-truth hit@5 from 22.2% to 80.3% (Figure 4d). The gain at this upstream stage shows that hierarchical training improved category identification, not only the final reranking of reagents.

The distribution-, category- and reagent-level results show that hierarchical modeling captures higher-order dependencies and translates this representational capacity into improved reagent ranking and category identification for variable-space design.

Table 2: Ablation of chemically informed constraints and contrastive-objective components on Reagent-Split and HTE-OOD. Panel I ablates the chemically informed constraints, and Panel II evaluates the multi-positive and positive-reagent coverage losses. Subscripts denote standard deviations across 3 runs. Positive Δ values indicate degradation relative to ChemVD.

Model variant	Reagent-Split Overall						HTE-OOD			
	wMRR ↑	Δ_{wMRR} ↓	wMPR ↓	Δ_{wMPR} ↓	wAPP ↑	Δ_{wAPP} ↓	MAMY@20% ↑	$\Delta_{20\%}$ ↓	MAMY@30% ↑	$\Delta_{30\%}$ ↓
ChemVD	$64.34_{\pm 0.17}$	-	$8.862_{\pm 0.040}$	-	$\mathbf{99.27}_{\pm 0.08}$	-	$\mathbf{68.78}_{\pm 1.93}$	-	$\mathbf{73.26}_{\pm 1.51}$	-
Panel I: Chemical-constraint ablation										
w/o (L_{neg})	$64.80_{\pm 0.05}$	-0.46	$8.764_{\pm 0.089}$	-0.098	$98.17_{\pm 0.10}$	1.10	$25.97_{\pm 15.76}$	42.81	$37.54_{\pm 21.10}$	35.72
w/o ($\text{TopFreq}_{K_{\text{hard}}}$)	$64.91_{\pm 0.19}$	-0.57	$8.048_{\pm 0.103}$	-0.814	$98.97_{\pm 0.04}$	0.30	$43.78_{\pm 8.28}$	25.00	$56.05_{\pm 13.24}$	17.21
w/o (w_m^c)	$64.79_{\pm 0.15}$	-0.45	$8.743_{\pm 0.077}$	-0.119	$99.20_{\pm 0.08}$	0.07	$35.24_{\pm 19.65}$	33.54	$49.31_{\pm 18.31}$	23.95
w/o (L_{low})	$64.70_{\pm 0.38}$	-0.36	$8.724_{\pm 0.192}$	-0.138	$98.88_{\pm 0.13}$	0.39	$35.03_{\pm 18.78}$	33.75	$53.33_{\pm 21.05}$	19.93
w/o (w_{yield})	$64.07_{\pm 0.36}$	0.27	$8.844_{\pm 0.140}$	-0.018	$99.14_{\pm 0.09}$	0.13	$38.96_{\pm 15.88}$	29.82	$52.70_{\pm 16.10}$	20.56
Panel II: Contrastive-loss ablation										
w/o $\mathcal{L}_{\text{MP}}$	$64.38_{\pm 0.23}$	-0.04	$8.680_{\pm 0.164}$	-0.182	$99.36_{\pm 0.01}$	-0.09	$35.09_{\pm 19.69}$	33.69	$54.56_{\pm 15.61}$	18.70
w/o $\mathcal{L}_{\text{cover}}$	$64.50_{\pm 0.10}$	-0.16	$10.212_{\pm 0.066}$	1.350	$99.40_{\pm 0.02}$	-0.13	$22.01_{\pm 6.07}$	46.77	$37.39_{\pm 1.02}$	35.87

4.3 Ablation Study of Chemical Constraints

To analysize how the chemically informed constraints primarily influence ID retrieval and robustness to OOD screening, we first ablated each constraint while retaining the full contrastive objective (Panel I of Table 2). On Reagent-Split, the constraints incurred only a small and mixed trade-off, with wMRR changing from 64.07 to 64.91 across ablations, compared with 64.34 for ChemVD. In contrast, their contribution was pronounced on HTE-OOD: removing individual constraints reduced MAMY by 25.00–42.81 percentage points at a 20% screening budget, with similarly large reductions at 30%. Together, these results indicate that the constraints may slightly compromise ID retrieval performance but substantially enhance generalization to OOD spaces.

We next ablated the two complementary components of the contrastive objective (Panel II) to determine whether each provided distinct supervision. Removing the multi-positive loss ($\mathcal{L}_{\text{MP}}$) had little effect on aggregate retrieval but reduced MAMY by 33.69 and 18.70 points at the 20% and 30% budgets. Removing positive-reagent coverage ($\mathcal{L}_{\text{cover}}$) caused larger reductions of 46.77 and 35.87 points and increased wMPR from 8.862 to 10.212. When used alone, $\mathcal{L}_{\text{MP}}$ preserves the joint validity of alternative reagents but permits a few high-scoring positives to dominate the set-level objective, leaving other valid alternatives poorly ranked. $\mathcal{L}_{\text{cover}}$ corrects this weakness by supervising every annotated positive and preventing less dominant alternatives from being excluded from the compact candidate subspace. However, $\mathcal{L}_{\text{cover}}$ alone treats each positive independently and therefore fails to encode that multiple reagents can constitute a jointly valid and interchangeable solution set. $\mathcal{L}_{\text{MP}}$ supplies this missing set-level structure by optimizing their collective probability mass. Their combination therefore preserves both the joint structure and individual coverage of valid reagents.

5 Conclusion

Reaction condition recommendation should not merely reproduce a historical condition, but construct a compact, high-value variable space for experimentation from an evolving candidate pool. From this perspective, we formulated the task as open-candidate retrieval under a limited experimental budget. ChemVD models reaction–category–reagent dependencies hierarchically and incorporates chemical priors into training, achieving effective ranking of both seen and unseen reagents and identifying high-yield subspaces at low screening budgets. Nevertheless, ChemVD is limited by the pretrained encoder and has yet to be validated prospectively in wet-lab experiments. Future work should therefore investigate domain-specific pretraining and task-specific post-training to improve chemical representations. Further validation should assess whether the retrieved subspaces increase screening efficiency in real-world experimental settings. Finally, ChemVD could serve as a variable-space construction module within automated experimentation workflows. Integration with Bayesian optimization could enable closed-loop condition recommendation and iterative refinement.

AI USE STATEMENT

In this work, we used generative AI tools for providing critical ingredients for proving mathematical claims, implementing methods, assisting in the writing of proofs, and assisting with translation. We have not used generative AI tools for generating synthetic data sets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, cleaning and reformatting datasets, supporting qualitative and thematic data analysis, or interpreting results. Additionally, we used generative AI tools for sourcing/searching for information and editing the paper to improve readability. We have reviewed the AI-assisted work. Specifically, we manually verified the LLM-generated mathematical ingredients and proof-writing assistance for correctness and logical soundness, and checked the LLM-assisted translations for accuracy against the original text. In addition, the LLM-generated code used to implement methods was tested and verified for correctness by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

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.

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

Florian Boser, Jan C Spies, and Frank Glorius. Yield prediction of organic reactions in biased data sets via positive-unlabeled learning. *Journal of the American Chemical Society*, 148(14):15066, 2026.

Pengxiang Cai, Zihao Ye, Shuzhou Sun, Longhan Zhang, Zhehong Ai, and Yuqiang Li. Chemrang: Building ai-ready datasets via mechanism-informed curation to learn reaction feasibility boundaries. *Chinese Chemical Letters*, pp. 113222, 2026.

Kexin Chen, Jiamin Lu, Junyou Li, Xiaoran Yang, Yuyang Du, Kunyi Wang, Qiannuan Shi, Jiahui Yu, Lanqing Li, Jiezhong Qiu, et al. Chemist-x: large language model-empowered agent for reaction condition recommendation in chemical synthesis. *arXiv preprint arXiv:2311.10776*, 2023.

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):1465–1476, 2018.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind one embedding space to bind them all. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15180–15190. IEEE, 2023.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Dong Han, Zhehong Ai, Pengxiang Cai, Shanya Lu, Jianpeng Chen, Zihao Ye, Shuzhou Sun, Ben Gao, Lingli Ge, Weida Wang, et al. Chembomas: Accelerated bo in chemistry with llm-enhanced multi-agent system. *arXiv preprint arXiv:2509.08736*, 2025.

Xin He, Kaiyong Zhao, and Xiaowen Chu. Automl: A survey of the state-of-the-art. *Knowledge-based systems*, 212:106622, 2021.

Artem I Leonov, Alexander JS Hammer, Slawomir Lach, S Hessam M Mehr, Dario Caramelli, Davide Angelone, Aamir Khan, Steven O'Sullivan, Matthew Craven, Liam Wilbraham, et al. An integrated self-optimizing programmable chemical synthesis and reaction engine. *Nature Communications*, 15(1):1240, 2024.

Shengchao Liu, Weili Nie, Chengpeng Wang, Jiarui Lu, Zhuoran Qiao, Ling Liu, Jian Tang, Chaowei Xiao, and Animashree Anandkumar. Multi-modal molecule structure–text model for text-based retrieval and editing. *Nature Machine Intelligence*, 5(12):1447–1457, 2023a.

Shengchao Liu, Weili Nie, Chengpeng Wang, Jiarui Lu, Zhuoran Qiao, Ling Liu, Jian Tang, Chaowei Xiao, and Animashree Anandkumar. Multi-modal molecule structure–text model for text-based retrieval and editing. *Nature Machine Intelligence*, 5(12):1447–1457, 2023b.

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.

Michael R Maser, Alexander Y Cui, Serim Ryou, Travis J DeLano, Yisong Yue, and Sarah E Reisman. Multilabel classification models for the prediction of cross-coupling reaction conditions. *Journal of Chemical Information and Modeling*, 61(1):156–166, 2021.

Damith Perera, Joseph W Tucker, Shalini Brahmbhatt, Christopher J Helal, Ashley Chong, William Farrell, Paul Richardson, and Neal W Sach. A platform for automated nanomole-scale reaction screening and micromole-scale synthesis in flow. *Science*, 359(6374):429–434, 2018.

Daniel Probst, Philippe Schwaller, and Jean-Louis Reymond. Reaction classification and yield prediction using the differential reaction fingerprint drfp. *Digital discovery*, 1(2):91–97, 2022.

Yujie Qian, Zhening Li, Zhengkai Tu, Connor Coley, and Regina Barzilay. Predictive chemistry augmented with text retrieval. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12731–12745, 2023.

David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010.

Hua Shi, Yuzhe Wang, Shouzhen Song, Kefei Li, Wenjia Gao, Shuan Liu, Jianbo Qiao, and Leyi Wei. Catacon: a contrastive graph representation learning framework for catalyst prediction. *Journal of Cheminformatics*, 18:92, 2026.

Joshua W Sin, Siu Lun Chau, Ryan P Burwood, Kurt Püntener, Raphael Bigler, and Philippe Schwaller. Highly parallel optimisation of chemical reactions through automation and machine intelligence. *Nature Communications*, 16(1):6464, 2025.

Connor J Taylor, Alexander Pomberger, Kobi C Felton, Rachel Grainger, Magda Barecka, Thomas W Chamberlain, Richard A Bourne, Christopher N Johnson, and Alexei A Lapkin. A brief introduction to chemical reaction optimization. *Chemical Reviews*, 123(6):3089, 2023.

Jose Antonio Garrido Torres, Sii Hong Lau, Pranay Anchuri, Jason M Stevens, Jose E Tabora, Jun Li, Alina Borovika, Ryan P Adams, and Abigail G Doyle. A multi-objective active learning platform and web app for reaction optimization. *Journal of the American Chemical Society*, 144 (43):19999–20007, 2022.

Xiaorui Wang, Chang-Yu Hsieh, Xiaodan Yin, Jike Wang, Yuquan Li, Yafeng Deng, Dejun Jiang, Zhenxing Wu, Hongyan Du, Hongming Chen, et al. Generic interpretable reaction condition predictions with open reaction condition datasets and unsupervised learning of reaction center. *Research*, 6:0231, 2023.

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.

Jialu Wu, Yiheng Zhu, Xiaorui Wang, Yitong Li, Mingze Yin, Tianyue Wang, Yuqiang Han, Yu Kang, Yafeng Deng, Jian Wu, et al. Hiclr: Knowledge-induced hierarchical contrastive learning with retrosynthesis prediction yields a reaction foundation model. *JACS Au*, 5(7):3140, 2025.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Xin Yan, Haowen Zhong, and Xiaoxue Wang. Robust chemical reaction condition recommendations via label mix strategy. *Journal of Chemical Information and Modeling*, 65(23):12775–12785, 2025a.

Xin Yan, Haowen Zhong, and Xiaoxue Wang. Robust chemical reaction condition recommendations via label mix strategy. *Journal of Chemical Information and Modeling*, 65(23):12775–12785, 2025b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

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.

Yuanhang Zheng, Peng Li, Wei Liu, Yang Liu, Jian Luan, and Bin Wang. Toolrerank: Adaptive and hierarchy-aware reranking for tool retrieval. In *Proceedings of the 2024 joint international conference on computational linguistics, language resources and evaluation (LREC-COLING 2024)*, pp. 16263–16273, 2024.

A Appendix

A.1 Theoretical Analysis of Candidate-Space Recommendation and Hierarchical Modeling

A.1.1 Notation and Objective for Candidate-Space Optimization

Let x_i denote the i-th reaction, c_m the m-th reagent category, and $g_{m,n}$ the n-th candidate reagent in category c_m. Let $Y_i(g)$ denote the experimental yield obtained by applying reagent g to reaction x_i, ξ_i the candidate space selected for reaction x_i, and B the maximum number of experiments that can be performed.

We assess the quality of a candidate space from two complementary perspectives: the best achievable yield and the probability of reaching a target yield:

$$\mathcal{J}_1(\xi_i) = \mathbb{E}\left[\max_{g\in\xi_i} Y_i(g)\right], \tag{8}$$

$$\mathcal{J}_2(\xi_i) = \Pr\left(\max_{g\in\xi_i} Y_i(g) \geq Y_{\text{tar}}\right) \tag{9}$$

where Y_{tar} is a prespecified target yield. Because these quantities may have different units or numerical ranges, we define the weighted utility

$$\mathcal{J}(\xi_i) = \lambda_1\widetilde{\mathcal{J}}_1(\xi_i) + \lambda_2\mathcal{J}_2(\xi_i), \qquad \lambda_1, \lambda_2 \geq 0, \tag{10}$$

where $\widetilde{\mathcal{J}}_1$ is the normalized expected maximum yield. The ideal candidate-space optimization problem is therefore

$$\xi_i^* \in \arg\max_{\xi_i\subseteq G_{\text{all}}} \mathcal{J}(\xi_i), \qquad \text{s.t.} \quad |\xi_i| \leq B, \tag{11}$$

where G_{all} is the set of all available reagents. This objective emphasizes that a recommendation model should not merely reproduce historical selections. Its ultimate purpose is to increase the probability of discovering effective reagents under a finite experimental budget.

A.1.2 Limitations of Supervised Classification

Cross-entropy learns historical selection probabilities

For reaction x_i and reagent category c_m, define a binary label indicating whether reagent $g_{m,n}$ appears in the historical record:

$$y_{i,m,n} = \begin{cases} 1, & \text{if } g_{m,n} \text{ was selected and recorded for reaction } x_i, \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

Let $\widehat{y}_{i,m,n} \in (0,1)$ be the model prediction. The binary cross-entropy loss for one example is:

$$\ell_{i,m,n} = -y_{i,m,n} \log \widehat{y}_{i,m,n} - (1 - y_{i,m,n}) \log(1 - \widehat{y}_{i,m,n}). \tag{13}$$

Conditioned on $(x_i, c_m, g_{m,n})$, let $p_{i,m,n} = \Pr(y_{i,m,n} = 1 \mid x_i, c_m, g_{m,n})$. The conditional expected loss is:

$$\begin{aligned} \mathbb{E}[\ell_{i,m,n} \mid x_i, c_m, g_{m,n}] = & - p_{i,m,n} \log \widehat{y}_{i,m,n} \\ & - (1 - p_{i,m,n}) \log(1 - \widehat{y}_{i,m,n}) \\ = & H(p_{i,m,n}) + D_{\mathrm{KL}}\left[\mathrm{Bern}(p_{i,m,n}) \parallel \mathrm{Bern}(\widehat{y}_{i,m,n})\right]. \end{aligned} \tag{14}$$

Because $D_{\mathrm{KL}} \geq 0$, the expected loss is minimized if and only if $\widehat{y}_{i,m,n} = p_{i,m,n}$. With sufficient data and model capacity, a cross-entropy classifier therefore estimates the conditional probability that a reagent was selected and recorded historically, given the reaction and reagent.

Historical recording probability is not experimental efficacy

Historical labels arise from at least two processes: an experimenter selects a reagent, and the corresponding result is retained and recorded. Define $p_{\mathrm{sel}}(g \mid x, c) = \Pr(S = 1 \mid x, c, g)$ and $p_{\mathrm{rec}}(g \mid x, c, Y) = \Pr(R = 1 \mid S = 1, x, c, g, Y)$, where S and R indicate selection and recording, respectively. Marginalizing over the potential experimental outcome Y gives

$$\begin{aligned} &\Pr(S = 1, R = 1 \mid x, c, g) \\ &\quad = p_{\mathrm{sel}}(g \mid x, c) \int p_{\mathrm{rec}}(g \mid x, c, y) p(y \mid x, c, g, S = 1) \, \mathrm{d}y. \end{aligned} \tag{15}$$

This probability depends on experimenter preferences, reagent availability, cost, and the recording mechanism. The quantity relevant to recommendation is instead the conditional efficacy distribution $p(Y \mid x, c, g)$.

Without additional assumptions about the selection and recording mechanisms, historical recording probabilities do not identify $p(Y \mid x, c, g)$. In particular, a reagent occurring more frequently in historical data does not necessarily produce a higher yield for the current reaction.

Selecting candidates only by their marginal historical frequencies $p_{m,n}$ implicitly assumes exchange monotonicity. Specifically, if $p_{m,n_1} > p_{m,n_2}$, then

$$\mathcal{J}\left[(\xi \setminus \{g_{m,n_2}\}) \cup \{g_{m,n_1}\}\right] > \mathcal{J}\left[(\xi \setminus \{g_{m,n_1}\}) \cup \{g_{m,n_2}\}\right]. \tag{16}$$

No universal monotonic relationship exists between historical frequency and experimental efficacy. Consequently, a candidate space constructed solely from historical occurrence frequencies is not guaranteed to maximize $\mathcal{J}(\xi_i)$.

Irreducible regret under a closed candidate set

Let the set of reagents observed during training be $G_{\text{train}} = \bigcup_{i,m} \left(G^{+}_{i,m} \cup G^{-}_{i,m} \right)$, where $G^{+}_{i,m}$ and $G^{-}_{i,m}$ are the observed positive and negative reagent sets. A standard closed-set classifier defines predictions only over G_{train} and satisfies $\sum_{g \in G_{\text{train}}} \widehat{p}(g \mid x_i, c_m) = 1$.

For an unseen reagent $g' \in G_{\text{all}} \setminus G_{\text{train}}$, the classifier has no corresponding output class and cannot assign a meaningful score. Extending the distribution by convention gives $\widehat{p}(g' \mid x_i, c_m) = 0$.

If an unseen reagent g' has a higher yield than every reagent in the training candidate set, the closed-set decision incurs

$$\text{Regret}_i = \max_{g \in G_{\text{all}}} Y_i(g) - \max_{g \in G_{\text{train}}} Y_i(g) > 0 \tag{17}$$

Improving classification accuracy within the closed candidate set cannot eliminate this gap. Evaluating reagents absent from training therefore requires comparable reagent representations and an open-set scoring mechanism.

A.1.3 From Classification to Reranking

The preceding analysis reveals a mismatch between historical selection labels and the experimental utility of a candidate space. To make model outputs directly support reagent prioritization, we reformulate closed-set classification,

$$(x_i, c_m) \longmapsto \widehat{p}(g_{m,n} \mid x_i, c_m), \tag{18}$$

as triplet-conditioned reranking:

$$(x_i, c_m, g_{m,n}) \longmapsto s(x_i, c_m, g_{m,n}) \in \mathbb{R}. \tag{19}$$

Ideally, the score should preserve the ordering induced by experimental efficacy. For any candidate reagents g_1 and g_2,

$$Y_i(g_1) > Y_i(g_2) \quad \Longrightarrow \quad s(x_i, c_m, g_1) > s(x_i, c_m, g_2). \tag{20}$$

This requirement aligns reranking with $\mathcal{J}$. Reformulating classification as reranking does not itself remove historical selection bias. Its direct advantage is that g becomes an input rather than a fixed class index, allowing the model to score any unseen reagent with an available representation.

A.1.4 Why a Category Encoder Helps: An Upper Bound on Ranking Errors

Let the scores of a positive example $i \in \mathcal{P}$ and a negative example $j \in \mathcal{N}$ be $s_i = \mu_i + \varepsilon_i$ and $s_j = \mu_j + \varepsilon_j$, where $\mu_i = \mathbb{E}[s_i]$ and $\mu_j = \mathbb{E}[s_j]$. Assume independent Gaussian noise $\varepsilon_i, \varepsilon_j \sim \mathcal{N}(0, \sigma^2)$.

Define the expected positive-negative margin as $\Delta_{ij} = \mu_i - \mu_j > 0$. The probability that the negative example outranks the positive example is

$$\Pr(s_j \geq s_i) = \Pr(\varepsilon_j - \varepsilon_i \geq \Delta_{ij}) = \overline{\Phi}\left(\frac{\Delta_{ij}}{\sqrt{2}\sigma} \right), \tag{21}$$

where $\overline{\Phi}(t) = 1 - \Phi(t)$ is the upper-tail probability of a standard normal variable. The Gaussian tail bound gives $\Pr(s_j \geq s_i) \leq \frac{1}{2} \exp\left(-\frac{\Delta_{ij}^2}{4\sigma^2} \right)$.

Define the event that at least one positive-negative inversion occurs as $\mathcal{E} = \bigcup_{i \in \mathcal{P}} \bigcup_{j \in \mathcal{N}} \{s_j \geq s_i\}$. By the union bound

$$\Pr(\mathcal{E}) \le \frac{1}{2} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} \exp\left(-\frac{\Delta_{ij}^2}{4\sigma^2}\right). \tag{22}$$

If $\Delta_{\min} = \min_{i \in \mathcal{P}, j \in \mathcal{N}} \Delta_{ij} > 0$, then

$$\Pr(\mathcal{E}) \le \frac{|\mathcal{P}|\,|\mathcal{N}|}{2} \exp\left(-\frac{\Delta_{\min}^2}{4\sigma^2}\right). \tag{23}$$

After introducing category c, comparisons can be restricted to the within-category negative set $\mathcal{N}_c \subseteq \mathcal{N}$. The corresponding bound is

$$\Pr(\mathcal{E}_c) \le \frac{1}{2} \sum_{i \in \mathcal{P}_c} \sum_{j \in \mathcal{N}_c} \exp\left(-\frac{\Delta_{ij}^2}{4\sigma^2}\right). \tag{24}$$

Provided that restricting the candidate set does not reduce score margins or increase noise variance, $\mathcal{N}_c \subseteq \mathcal{N}$ reduces the number of error events in the union bound. A category encoder can therefore constrain the comparison set and reduce irrelevant cross-category ranking errors.

A.1.5 Hierarchical versus Pairwise Trimodal Architectures

A coarse-grained cross-entropy interpretation of MP-InfoNCE

For anchor u_i, let $\mathcal{A}_i$ be the contrastive set and $\mathcal{P}_i \subseteq \mathcal{A}_i$ the positive set. MP-InfoNCE assigns candidate v_j the normalized probability

$$q_{ij} = \frac{\exp(s_{ij}/\tau)}{\sum_{k \in \mathcal{A}_i} \exp(s_{ik}/\tau)} \tag{25}$$

where s_{ij} is a similarity score and $\tau > 0$ is the temperature. The multi-positive loss is

$$\ell_i^{\mathrm{MP}} = -\log \sum_{j \in \mathcal{P}_i} q_{ij}. \tag{26}$$

Let $Z_i = 1$ denote the event that a sample drawn from q_i belongs to the positive set. Then

$$\Pr_{q_i}(Z_i = 1 \mid u_i) = \sum_{j \in \mathcal{P}_i} q_{ij}, \tag{27}$$

and hence $\ell_i^{\mathrm{MP}} = -\log \Pr_{q_i}(Z_i = 1 \mid u_i)$.

MP-InfoNCE is therefore exactly the negative log-likelihood of the set-level event that a candidate belongs to the positive set. It shares the log-likelihood form of cross-entropy but does not distinguish members within the positive set, making it a set-level, coarse-grained cross-entropy objective.

The information-theoretic analysis below uses negative log-likelihood as a proxy. It characterizes the set-level optimization direction of MP-InfoNCE without claiming numerical equivalence under finite batches, specific negative-sampling schemes, or temperature scaling.

Hierarchical modeling

For the category-reagent joint distribution conditioned on reaction $X = x$, the chain rule gives $p(c, g \mid x) = p(c \mid x)p(g \mid x, c)$. A hierarchical model approximates these terms using $q_1(c \mid x)$ and $q_2(g \mid x, c)$, defining $q_{\mathrm{hier}}(c, g \mid x) = q_1(c \mid x)q_2(g \mid x, c)$. Its expected negative log-likelihood is:

$$\begin{aligned}\mathcal{L}_{\text{hier}} &= \mathbb{E}_{p(x,c,g)}[-\log q_1(c \mid x) - \log q_2(g \mid x, c)] \\ &= H_p(C, G \mid X) + \mathbb{E}_{p(x)} D_{\text{KL}}[p(c, g \mid x) \,\|\, q_{\text{hier}}(c, g \mid x)] \,.\end{aligned} \tag{28}$$

With sufficient model capacity, global optimization, and adequate distributional coverage:

$$\begin{aligned}q_{\text{hier}}(c, g \mid x) &= p(c, g \mid x) \\ \mathcal{L}^*_{\text{hier}} &= H_p(C, G \mid X) = H_p(C \mid X) + H_p(G \mid X, C)\end{aligned} \tag{29}$$

Thus, the hierarchical factorization is an exact chain-rule decomposition of the joint conditional distribution and introduces no additional independence assumption by itself.

Pairwise marginals cannot recover all higher-order information

Consider a pairwise trimodal architecture that separately aligns (X, C), (X, G), and (C, G). The following counterexample shows that identical pairwise marginals can correspond to different joint distributions.

Let $X, C \overset{\text{i.i.d.}}{\sim} \text{Bernoulli}(1/2)$ and consider two cases:

- **Case A (mutual independence):** $G \sim \text{Bernoulli}(1/2)$, and X, C, and G are mutually independent.
- **Case B (exclusive-or relation):** $G = X \oplus C$, where $\oplus$ denotes exclusive OR.

In both cases, (X, C), (X, G), and (C, G) are uniform distributions over two independent Bernoulli variables. The three pairwise marginals therefore cannot distinguish Case A from Case B.

However, Case A satisfies $p(g \mid x, c) = p(g) = \frac{1}{2}$, whereas Case B satisfies $p(g \mid x, c) = \mathbb{I}\{g = x \oplus c\}$.

Consequently, $H(G \mid X, C) = 1$ bit in Case A but $H(G \mid X, C) = 0$ in Case B. The three pairwise marginals are therefore insufficient to identify $p(x, c, g)$ or $p(c, g \mid x)$. Any prediction rule depending only on these pairwise marginals cannot recover the correct conditional distribution in both cases. This conclusion is independent of parameterization and reflects the non-identifiability of higher-order structure from pairwise marginal information alone.

Pairwise modeling

A pairwise model may express the triplet relation as a product of three conditional terms $r(c, g \mid x) \propto q(c \mid x)q(g \mid x)q(g \mid c)$. To obtain a valid conditional distribution, it must introduce $Z(x) = \sum_{c,g} q(c \mid x)q(g \mid x)q(g \mid c)$, which yields $r(c, g \mid x) = \frac{q(c|x)q(g|x)q(g|c)}{Z(x)}$. This product form represents only conditional distributions composed of pairwise potentials. It does not generally contain arbitrary higher-order X-C-G interactions. The exclusive-or counterexample above is a canonical instance of such an interaction.

From an information-theoretic perspective,

$$\begin{aligned}H(G \mid C) - H(G \mid X, C) &= I(G; X \mid C) \geq 0 \\ H(G \mid X) - H(G \mid X, C) &= I(G; C \mid X) \geq 0.\end{aligned} \tag{30}$$

When $I(G; X \mid C) > 0$ or $I(G; C \mid X) > 0$, learning only $q(g \mid c)$ or $q(g \mid x)$ discards predictive information supplied by the other conditioning variable. The hierarchical term $q(g \mid x, c)$ directly models this conditional dependence.

Thus, when the data contain nonzero higher-order interactions and the pairwise trimodal architecture has no additional mechanism to recover them, the hierarchical architecture can have greater representational capacity. This statement does not imply that every hierarchical model must outperform every trimodal model with finite data. It establishes only that the chain-rule factorization $q(c \mid x)q(g \mid x, c)$

matches the target joint distribution, whereas pairwise-only schemes may be non-identifiable under higher-order dependence.

A.2 Methodological Details

A.2.1 Hierarchical Representation and Retrieval

Modality-Specific Encoders

For each modality $u \in \{r, c, g\}$, a fixed base encoder extracts a feature vector $\mathbf{h}^u$, which is mapped into the shared d-dimensional space by a trainable linear projection $\mathbf{z}^u = \mathrm{Norm}_2(\mathbf{W}_u\mathbf{h}^u + \mathbf{b}_u)$. The base encoders remain frozen, and only the modality-specific projection parameters $\{\mathbf{W}_u, \mathbf{b}_u\}$ are optimized. Implementation details are provided in Appendix A.4.3.

Reaction-to-Category Retrieval

The model first computes the similarity between reaction x_i and every condition category $s^c_{i,m} = \cos(\mathbf{z}^r_i, \mathbf{z}^c_m) = (\mathbf{z}^r_i)^\top \mathbf{z}^c_m$. This stage identifies the functional types of conditions required by the reaction without directly distinguishing among individual reagents. At inference time, the categories are retained according to a selection criterion that is either the K_c highest-scoring categories or a fixed threshold τ_c on the predicted scores.

Category-Conditioned Reagent Retrieval

To explicitly differentiate the suitability of the same molecular structure across distinct functional roles, a category-conditioned query is constructed for each candidate category. Specifically, the reaction representation, category representation, and their element-wise product are concatenated and passed through a fusion network, followed by normalization:

$$\mathbf{q}_{i,m} = \mathrm{Norm}_2\left(\mathbf{W}_{f,2}\,\mathrm{GELU}\left(\mathbf{W}_{f,1}[\mathbf{z}^r_i; \mathbf{z}^c_m; \mathbf{z}^r_i \odot \mathbf{z}^c_m] + \mathbf{b}_{f,1}\right) + \mathbf{b}_{f,2}\right). \tag{31}$$

For each candidate reagent $g_{m,n} \in \mathcal{G}_m$, ChemVD computes $s^g_{i,m,n} = \cos(\mathbf{q}_{i,m}, \mathbf{z}^g_{m,n})$ and returns the K_g highest-scoring candidates as $\widehat{\mathcal{G}}_{i,m}$.

A.2.2 Chemically Informed Training Constraints

We first define a unified condition-level score $S(C, \mathcal{G})$ over active categories C and reagents $\mathcal{G}$ via mean hierarchical pooling:

$$S(C, \mathcal{G}) = \frac{1}{|C|}\sum_{c_m \in C} s^c_{i,m} + \frac{1}{|\mathcal{G}|}\sum_{g_{m,n} \in \mathcal{G}} s^g_{i,m,n}. \tag{32}$$

Low-Yield Suppression. A low yield does not necessarily imply that a condition is chemically invalid. Nevertheless, a low-yield condition should not receive the same matching score as a high-quality positive condition. We therefore treat observed conditions with $y_i < y_{\text{low}}$ as weak negatives. For a low-yield condition l, its condition-level score $S(C_l, \mathcal{G}_l)$ is the sum of the mean scores of its constituent categories and reagents. The corresponding one-sided penalty is

$$\mathcal{L}_{\text{low}} = \frac{1}{|\mathcal{Q}_{\text{low}}|}\sum_{l \in \mathcal{Q}_{\text{low}}} \max\left(0,\ S(C_l, \mathcal{G}_l) - \mu_{\text{low}}\right), \tag{33}$$

where μ_{low} is the penalty threshold and $\mathcal{Q}_{\text{low}}$ is the set of low-yield conditions in the mini-batch. This objective suppresses only excessively high scores for low-yield conditions and does not impose the same degree of negative supervision as that used for explicitly incorrect conditions.

Explicit-Negative Scoring. In addition to sampled within-category contrast candidates, the data may provide explicit negative conditions $\mathcal{Q}_{\text{neg}}$ paired with a positive condition. For the positive condition

of sample i, we first calculates the positive score $S(C_i^+, \mathcal{G}_i^+)$ and the negative score $S(C_{i,q}^-, \mathcal{G}_{i,q}^-)$ of the q-th explicit negative condition. If either set is empty, its corresponding term is defined as zero. We use a pairwise margin ranking loss for explicit negative conditions:

$$\mathcal{L}_{\text{neg}} = \frac{1}{|\mathcal{Q}_{\text{neg}}|} \sum_{(i,q)\in\mathcal{Q}_{\text{neg}}} \max\left(0,\, \mu_{\text{neg}} - (S(C_i^+, \mathcal{G}_i^+) - S(C_{i,q}^-, \mathcal{G}_{i,q}^-))\right) \tag{34}$$

where μ_{neg} is the prescribed margin. This constraint directly requires the mean matching score of the complete positive condition to exceed that of its paired negative condition by at least the specified margin.

Category-Balanced Weighting. Condition categories occur with markedly different frequencies in the training set. Let n_m denote the number of training samples containing category c_m. To prevent common functional categories from dominating cross-modal alignment, ChemVD assigns each category an effective-number weight $\widetilde{w}_m^c = \frac{1-\beta_c}{1-\beta_c^{n_m}}$. The raw weights are normalized by their mean across all categories and clipped from above: $w_m^c = \min(\frac{\widetilde{w}_m^c}{M^{-1}\sum_{j=1}^{M}\widetilde{w}_j^c}, w_{\max})$. The settings of β_c and $w_{\max}$ are provided in Appendix A.4.3. The resulting w_m^c enters the positive category weight $w_{i,m}^+ = w_m^c w_i^y$ defined above. Thus, category reweighting changes the contribution of category-level positive supervision without altering the candidate space or the inference-time scoring function.

Within-Category Contrast Sampling. For each positive category c_m represented in mini-batch $\mathcal{B}$, we first define the set of within-batch positive reagents and the corresponding implicit contrast pool as $\mathcal{G}_m^{+,\mathcal{B}} = \bigcup_{i\in\mathcal{B}} \mathcal{G}_{i,m}^+$ and $\mathcal{N}_m^{\text{imp},\mathcal{B}} = \mathcal{G}_m \setminus \mathcal{G}_m^{+,\mathcal{B}}$. Here, "implicit" means that a reagent belongs to the registered candidate set for category c_m but is not annotated as positive for that category in the current mini-batch. It does not imply that the reagent is chemically invalid for the queried reaction. We sample from this pool according to

$$\widetilde{\mathcal{N}}_m^{\text{imp},\mathcal{B}} = \begin{cases} \mathcal{N}_m^{\text{imp},\mathcal{B}}, & \left|\mathcal{N}_m^{\text{imp},\mathcal{B}}\right| \le K_{\text{neg}} \\ \text{TopFreq}_{K_{\text{hard}}}\left(\mathcal{N}_m^{\text{imp},\mathcal{B}}\right) & \\ \quad \cup\, \text{Rand}_{K_{\text{rand}}}\left(\mathcal{N}_m^{\text{imp},\mathcal{B}} \setminus \text{TopFreq}_{K_{\text{hard}}}\left(\mathcal{N}_m^{\text{imp},\mathcal{B}}\right)\right), & \left|\mathcal{N}_m^{\text{imp},\mathcal{B}}\right| > K_{\text{neg}} \end{cases} \tag{35}$$

where K_{neg} is a hyperparameter specifying the maximum number of negative reagents sampled for each positive category. $K_{\text{hard}} = \text{round}(\rho K_{\text{neg}})$ and $K_{\text{rand}} = K_{\text{neg}} - K_{\text{hard}}$. The category-specific training candidate set is then $\widetilde{\mathcal{G}}_m^{\mathcal{B}} = \mathcal{G}_m^{+,\mathcal{B}} \cup \widetilde{\mathcal{N}}_m^{\text{imp},\mathcal{B}}$. TopFreq selects reagents according to their occurrence frequencies in the training records of the same category, providing frequently observed within-category alternatives for finer discrimination. Rand denotes uniform random sampling from the remainder of the pool and preserves candidate diversity. The sampling ratio and candidate-pool size are provided in Appendix A.4.3.

A.3 Datasets and Data Splits

This section describes the primary dataset curated from ChemRANG and three derived evaluation settings: (1) Reagent-Split, stratified by reagent visibility during training; (2) the closed catalyst-space evaluations based on high-throughput experimentation (HTE), HTE-OOD; and (3) Category-Level Eval, comprising test examples with low reaction similarity to the training set.

A.3.1 Reaction and Reagent Standardization

For each raw record, we first extract the reactant and product SMILES strings, independently deduplicate and sort them, and then construct a standardized reaction SMILES string. Records with missing reactants, missing products, or an invalid reaction representation are excluded from training. Condition entities are described by an IUPAC name and a SMILES string. When a SMILES string is available, it serves as the primary unique identifier of the reagent; otherwise, the normalized IUPAC name is used.

Entities without a valid name or molecular structure are discarded. Category names are lowercased, standardized by replacing spaces with underscores, processed according to predefined category-merging rules, and mapped to a fixed category table. In addition to its name, each category

stores inclusion criteria and exclusion boundaries to reduce ambiguity between semantically similar categories.

A.3.2 Construction of the Training, Validation, and Test Sets

Data standardization and supervision aggregation

We cleaned and repartitioned the raw Suzuki-RANG data using a unified pipeline. Reactants and products were converted to canonical SMILES and sorted separately to construct the normalized reaction identifier

$$k_{\text{rxn}} = \text{sort}(R_{\text{canonical}}) >> \text{sort}(P_{\text{canonical}}), \tag{36}$$

where R and P denote the reactant and product sets. Records with missing valid reactants or products, or structures that could not be parsed, were excluded. Condition-category names and reagent structures were also normalized. Records sharing the same normalized reaction identifier and condition combination were merged into one example,

$$g = (k_{\text{rxn}}, s_{\text{condition}}), \tag{37}$$

where $s_{\text{condition}}$ is the condition signature obtained by sorting condition categories and reagents. When an example contained repeated experiments, the median valid yield was used as the supervision target to reduce sensitivity to individual measurement fluctuations.

Reaction-cluster-based splitting

To reduce information leakage caused by structurally similar reactions crossing split boundaries, we used reaction clusters rather than individual records as the allocation unit. Records with identical normalized reaction identifiers were first assigned to the same exact-reaction group. We then represented each reaction with a 2,048-bit RDKit structural reaction fingerprint and further grouped similar reactions using Tanimoto similarity, product scaffold, and a coarse reaction template. A new reaction i was assigned to the cluster represented by reaction j if any of the following criteria was satisfied:

1. The reaction fingerprints were highly similar: $T(i, j) \geq 0.85$.
2. The product scaffolds matched and fingerprint similarity was at least 0.70: $\text{scaffold}(i) = \text{scaffold}(j)$ and $T(i, j) \geq 0.70$.
3. The coarse reaction templates matched and fingerprint similarity was at least 0.75: $\text{template}(i) = \text{template}(j)$ and $T(i, j) \geq 0.75$.

The target Train:Validation:Test ratio was 8:1:1. Each complete reaction cluster was assigned to exactly one split. Allocation jointly considered the number of raw records, yield distribution, condition-category distribution, and frequent-reagent distribution. Thus, 80:10:10 was a global optimization target rather than a strict record-level ratio. The results is shown in Table 3.

To prevent external HTE reactions from leaking into training or model selection, we compared the primary dataset against the 15 HTE reactions. Any cluster exactly matching an HTE reaction, or having reaction-fingerprint Tanimoto similarity of at least 0.85 to one, was forced into Test.

Table 3: Sizes of the training, validation, and test sets.

Split	Number of examples
Train	98,032
Validation	19,163
Test	19,348

A.3.3 Reagent-Split: Stratifying Test Examples by Training Visibility

Definitions of seen and unseen

Reagent-Split is a post hoc stratification of the fixed Test set and does not alter the Train, Validation, or Test boundaries. It separately evaluates retrieval of within-category reagents observed during training and extrapolation to examples containing at least one within-category unseen reagent.

We constructed a category-specific reagent registry from Train using

$$(c, r) = (\text{condition category}, \text{canonical reagent structure}) \tag{38}$$

as the unique key. The same reagent structure appearing under different condition categories was treated as a different entry. The resulting registry contained 8,279 category-reagent entries.

Let R_g^+ be the set of all positive category-structure pairs in test example g, and let $n_{\text{train}}(c, r)$ be the training frequency of an entry. We define

$$\text{Seen}(g) \iff \forall (c, r) \in R_g^+, \quad n_{\text{train}}(c, r) > 0, \tag{39}$$

$$\text{Unseen}(g) \iff \exists (c, r) \in R_g^+, \quad n_{\text{train}}(c, r) = 0. \tag{40}$$

Thus, every positive reagent in a Seen example must have appeared under the same category in Train. An example is Unseen if it contains at least one positive reagent absent from that category during training. Because the definition operates at the example level, an Unseen example may also contain other seen reagents.

Category-specific candidate pools

Seen and Unseen queries within a condition category use the same candidate pool. Each pool contains all positive and labeled-negative reagents appearing under that category in Test. Sharing candidate pools prevents pool size or composition from directly revealing the Seen/Unseen label.

Overall size

As shown in Table 4, Reagent-Split contained 18,975 examples, of which 94.20% were Seen and 5.80% were Unseen. Because the strata were highly imbalanced, we report them separately rather than replacing them with a pooled metric. Only examples with yield measurements were used for the yield-weighted evaluation metrics.

Table 4: Sizes of the Reagent-Split strata.

Stratum	Number of examples	Examples with yields
Seen	17,875	9,690
Unseen	1,100	402
Total	18,975	10,092

A.3.4 HTE Closed Catalyst-Space Evaluation Sets

Original HTE condition space

The HTE data originated from a Suzuki high-throughput reaction dataset Perera et al. (2018). The original data fixed the product while varying substrates. We instead fixed the substrates and obtained

15 reaction subsets defining the original HTE condition spaces. Within each subset, reactants and products remained fixed while Catalyst, Ligand, Base, and Solvent varied. Each subset contained 1 Catalyst, 12 Ligands, 8 Bases, and 4 Solvents, giving $1 \times 12 \times 8 \times 4 = 384$ measured examples. Across 15 subsets, the dataset contained 5,760 measured examples.

Synthetic negative catalysts and labels

HTE-OOD was constructed by expanding the Catalyst candidate space. Each added synthetic negative catalyst was combined factorially with all original Ligand, Base, and Solvent levels. Combinations containing a synthetic negative catalyst were treated as explicit negatives and assigned a pseudo-yield of zero. Combinations containing the original Catalyst retained their measured yields.

HTE-OOD added 24 synthetic negative catalysts representing acetate, chloride, bromide, fluoride, sulfate, tetrafluoroborate, and hexafluorophosphate salts. Each reaction subset therefore contained 25 Catalysts and $25 \times 12 \times 8 \times 4 = 9{,}600$ examples. Of these, 384 were measured examples and 9,216 were synthetic negatives. The 15 subsets contained 144,000 examples in total. The statics is shown in Table 5.

Table 5: Sizes of the HTE-raw and HTE-OOD evaluation sets.

Dataset	**Per subset**			**Total**
	Measured	Synthetic negatives	Total	(15 subsets)
HTE-raw	384	0	384	5,760
HTE-OOD	384	9,216	9,600	144,000

Catalyst-negative candidates

Table 6 lists the 24 synthetic negative catalysts added to construct HTE-OOD. Each candidate was combined factorially with all original ligand, base, and solvent levels and assigned a pseudo-yield of zero, as described above.

A.3.5 Category-Level Eval: A Low-Similarity Test Subset

Construction

Category-Level Eval was derived from Test without changing the primary splits. For each Train and Test example, we constructed a reaction string using only positive reactants and products, excluding condition reagents such as Catalyst, Ligand, Base, and Solvent. We then encoded each reaction using RXNFP. For test example t, proximity to Train was defined as its maximum cosine similarity to any training example:

$$s(t) = \max_{r \in \mathcal{D}_{\text{train}}} \cos(\text{RXNFP}(t), \text{RXNFP}(r)) \,. \tag{41}$$

Using a threshold of 0.9, we stratified Test as

$$\mathcal{D}_{\text{low}} = \{t : s(t) < 0.9\}, \qquad \mathcal{D}_{\text{high}} = \{t : s(t) \geq 0.9\}. \tag{42}$$

We used $\mathcal{D}_{\text{low}}$ as Category-Level Eval to assess condition-category prediction when no highly similar training reaction was available.

Table 6: Catalyst-negative candidates used to expand the HTE-OOD catalyst space.

No.	Catalyst name	SMILES
1	lithium acetate	CC(=O)[O-].[Li+]
2	sodium acetate	CC(=O)[O-].[Na+]
3	potassium acetate	CC(=O)[O-].[K+]
4	magnesium acetate	CC(=O)[O-].CC(=O)[O-].[Mg+2]
5	calcium acetate	CC(=O)[O-].CC(=O)[O-].[Ca+2]
6	strontium acetate	CC(=O)[O-].CC(=O)[O-].[Sr+2]
7	barium acetate	CC(=O)[O-].CC(=O)[O-].[Ba+2]
8	aluminium acetate	CC(=O)[O-].CC(=O)[O-].[Al+3]
9	sodium chloride	[Na+].[Cl-]
10	potassium chloride	[K+].[Cl-]
11	lithium chloride	[Li+].[Cl-]
12	magnesium chloride	[Mg+2].[Cl-].[Cl-]
13	calcium chloride	[Ca+2].[Cl-].[Cl-]
14	sodium bromide	[Na+].[Br-]
15	potassium bromide	[K+].[Br-]
16	lithium bromide	[Li+].[Br-]
17	sodium fluoride	[Na+].[F-]
18	potassium fluoride	[K+].[F-]
19	sodium sulfate	[Na+].[Na+].[O-]S(=O)(=O)[O-]
20	potassium sulfate	[K+].[K+].[O-]S(=O)(=O)[O-]
21	sodium tetrafluoroborate	[Na+].F[B-](F)(F)F
22	potassium tetrafluoroborate	[K+].F[B-](F)(F)F
23	sodium hexafluorophosphate	[Na+].F[P-](F)(F)(F)(F)F
24	potassium hexafluorophosphate	[K+].F[P-](F)(F)(F)(F)F

Size and similarity distribution

Valid RXNFP representations were obtained for all 98,032 Train examples and 19,348 Test examples. Category-Level Eval contained 117 low-similarity examples, accounting for 0.6047% of Test. The remaining 19,231 examples had maximum training similarity of at least 0.9. The complete similarity distribution results is shown in Table 7.

Table 7: Distribution of maximum RXNFP cosine similarity to the training set.

Similarity interval	Count	Fraction	Mean	Median	Min.	Max.
$[-1.0, 0.7)$	0	0 %	—	—	—	—
$[0.7, 0.8)$	7	0.0362%	0.776 431	0.784 667	0.750 968	0.787 393
$[0.8, 0.9)$	110	0.5685%	0.874 669	0.880 131	0.808 278	0.899 833
$[0.9, 1.0]$	19 231	99.3953%	0.987 062	0.994 876	0.900 150	0.999 999

The primary split and this similarity stratification used different reaction representations and similarity measures. The former used RDKit structural reaction fingerprints and Tanimoto similarity to construct indivisible reaction clusters. The latter used RXNFP and cosine similarity to stratify the fixed Test set by difficulty after splitting.

A.4 Experimental Setup

A.4.1 Baselines

Closed-set supervised models for chemistry

- **ReaCon.** (Wang et al., 2025) ReaCon is a closed-set reaction-condition recommendation method based on reaction graphs. It encodes reactant and product structures with a message-passing neural network (MPNN) and trains separate multilabel classifiers for reagent categories such as catalysts, ligands, bases, and solvents. During evaluation, we rank candidates within each category by their classification scores. Because its output space is fixed by the training vocabulary, ReaCon measures the recommendation ability of a conventional graph-classification architecture within a known reagent space.

- **Label-mix.** (Yan et al., 2025b) Label-mix introduces label mixing and self-knowledge distillation into multilabel reaction-condition prediction to improve robustness to label noise and long-tailed conditions. An MPNN encodes molecular graphs, while teacher-student training combines observed labels with teacher-generated soft labels. Co-occurrence constraints and a ranking loss model relations among reagents. At inference time, the model scores candidates from a fixed category-specific vocabulary.

- **Parrot.** (Wang et al., 2023) Parrot formulates reaction-condition recommendation as sequence-to-sequence generation. A BERT-based reaction encoder represents reaction SMILES, and a cross-attentive Transformer decoder autoregressively generates condition labels in a predefined category order. Each prediction is conditioned on previously generated conditions, allowing dependencies among reaction conditions to be modeled explicitly. We use the conditional probabilities at the relevant decoding positions to rank candidates within each category.

- **TextReact.** (Qian et al., 2023) The original TextReact method augments reaction-condition prediction with patent text retrieved for the target reaction. To ensure matched input information, we use its autoregressive variant without external retrieval. SciBERT encodes reaction SMILES, and a cross-attentive Transformer decoder generates reaction conditions in a fixed category order. We use the conditional probabilities at the corresponding generation positions as candidate-ranking scores.

General-purpose language models

- **Qwen3-30B (Yang et al., 2025) and Llama-3.1-70B (Grattafiori et al., 2024).** These models provide general-purpose language-model baselines from a technological period similar to that of our category encoder, but with larger parameter counts. Our category encoder uses Llama-3.1-8B, whereas these baselines contain approximately 30 and 70 billion parameters. Both models were deployed locally and ranked candidates using only their internalized chemical knowledge, without task-specific fine-tuning.
- **Qwen3.5-35B and Llama-3.3-70B.** These newer general-purpose models allow us to examine potential gains from advances in pretraining data, training strategies, and base reasoning capabilities. They were also deployed locally and received only the reaction and candidate-reagent information, without task-specific fine-tuning.
- **DeepSeek-V4-Flash and DeepSeek-V4-Pro Xu et al. (2026).** We additionally evaluated two DeepSeek models through an API as general-purpose language-model baselines at the hundred-billion-parameter scale. They were not given experimental yields and were not trained further on our data.

API LLM baseline

For the API-based LLM baseline, we required a complete within-category ranking of every supplied candidate. The system prompt was as follows:

```
You are a reaction-condition expert. Rank every supplied candidate
within each condition category from most to least promising for the
    supplied reaction.

Your response is consumed by a strict JSON parser. Follow all rules
    below:
1. Return exactly one JSON object and nothing else (no Markdown,
    explanation, or comments).
2. Use exactly this shape: {"rankings":{"CATEGORY":["CANDIDATE_ID",
    "..."]}}.
3. Copy category names and candidate IDs verbatim from output_contract.
4. Include every supplied category exactly once and no additional
    categories.
5. For each category, include every allowed ID exactly once, with no
    omissions,
   duplicates, names, SMILES, scores, or additional fields.
6. Order each ID list from most to least promising.
7. Before responding, silently verify that every list has the required
    count and is an
   exact permutation of allowed_ids_by_category.
```

The API implementation serializes the following object as compact JSON in the user message:

```
{
  "task": "Rank all candidates in every category from most to least
    promising.",
  "input": {
    "reaction_smiles": "<REACTION_SMILES>",
    "reaction_type": "<REACTION_TYPE>",
    "candidates_by_category": {
      "<CATEGORY>": [
        {"id": "<CANDIDATE_ID>", "name": "<CANDIDATE_NAME>", "smiles":
    "<CANDIDATE_SMILES>"}
      ]
    }
  },
  "output_contract": {
    "response_type": "json_object_only",
```

```
    "exact_top_level_keys": ["rankings"],
    "category_order": ["<CATEGORY>", "..."],
    "allowed_ids_by_category": {"<CATEGORY>": ["<CANDIDATE_ID>",
    "..."]},
    "required_count_by_category": {"<CATEGORY>": "<COUNT>"},
    "rules": [
      "Use exactly the category keys listed in category_order.",
      "Each ranking must be an exact permutation of that category's
    allowed IDs.",
      "Use IDs only; do not output candidate names, SMILES, scores, or
    explanations."
    ]
  }
}
```

The expected response format was

```
{"rankings":{"Catalyst":["c0001","c0002"],"Ligand":["c0002","c0001"],"
    Base":["c0001","c0002"],"Solvent":["c0002","c0001"]}}
```

The response had to be a complete permutation of all allowed IDs. Invalid responses were retried for up to three validation attempts, with the validation error and the same output contract supplied to the model.

Table 8: Settings for the API LLM baseline in the catalyst-variable evaluation.

Item	Setting
Models	`deepseek-v4-flash`; `deepseek-v4-pro`
Temperature	0.1
Maximum output tokens	16,384
Calls	One valid ranking per seed for each model and HTE subset, yielding three rankings in total; each ranking was reused across evaluation budgets.

Local LLM baseline

For locally deployed LLMs, the system prompt was

```
You are a chemistry reaction-condition ranking model. Follow the
    requested JSON schema exactly.
```

The user-message template was

```
Rank the candidate reaction conditions for the reaction below from most
     to least promising for high yield. Rank every candidate exactly
    once within its existing category. Do not invent candidates or move
    them between categories. Return only a JSON object mapping each
    category to an ordered list of candidate IDs.

Reaction SMILES: <REACTION_SMILES>
Candidates: <JSON_OBJECT_OF_CATEGORY_TO_[ID,NAME,SMILES]_RECORDS>
```

The expected response format was

```
{"Catalyst":["c0001","c0002"],"Ligand":["l0001","l0002"],"Base":["b0001
    ","b0002"],"Solvent":["s0001","s0002"]}
```

The tokenizer chat template was applied with `enable_thinking=False`. The parser extracted the final JSON object, discarded invalid or duplicate IDs, and appended omitted candidates in their original input order.

Table 9: Settings for the local LLM baseline in the catalyst-variable evaluation.

Item	Setting
Models	`Llama-3.1-70B-Instruct`; `Llama-3.3-70B-Instruct`; `Qwen3-30B-A3B-Instruct-2507`; `Qwen3.5-35B-A3B`
Precision	BF16
Sampling	Enabled
Temperature	0.1
Top-p	0.9
Maximum new tokens	1,024
Calls	One valid ranking per seed for each model and HTE subset, yielding three rankings in total; each ranking was reused across evaluation budgets.

Learning-objective baselines

Both baselines retain the reaction, category, and reagent encoders of ChemVD and independently encode candidate reagents, thereby supporting open-candidate inference.

- **BCE.** This baseline retains the category-conditioned query but replaces the reagent-level contrastive objective with binary cross-entropy. Selected reaction–category–reagent tuples receive a label of 1, whereas dataset-provided explicit negative tuples receive a label of 0. The selection probability for reagent n in category m for reaction i is

$$p_{i,m,n} = \sigma\left(\mathbf{w}^\top [\mathbf{q}_{i,m}; \mathbf{z}^g_{m,n}] + b\right), \tag{43}$$

  where $[\cdot\,;\cdot]$ denotes concatenation, and $\mathbf{w}$ and b are learned classifier parameters. Candidate reagents within each category are ranked by $p_{i,m,n}$.

- **Pairwise.** This baseline isolates the contribution of hierarchical dependency modeling by replacing the category-conditioned reaction query with three independently learned pairwise alignments. These alignments connect reaction–category, reaction–reagent, and category–reagent representations, with similarity scores

$$s^{rc}_{i,m} = (\mathbf{z}^r_i)^\top \mathbf{z}^c_m, \qquad s^{rg}_{i,m,n} = (\mathbf{z}^r_i)^\top \mathbf{z}^g_{m,n}, \qquad s^{cg}_{i,m,n} = (\mathbf{z}^c_m)^\top \mathbf{z}^g_{m,n}. \tag{44}$$

  Each alignment is optimized independently using the same multi-positive contrastive objective as ChemVD. During inference, categories are ranked by $s^{rc}_{i,m}$. Reagents within category c_m are ranked by the mean reaction–reagent and category–reagent similarity:

$$s^{\text{Pairwise}}_{i,m,n} = \frac{1}{2}\left(s^{rg}_{i,m,n} + s^{cg}_{i,m,n}\right). \tag{45}$$

  Unlike ChemVD, this score combines two separately learned associations and therefore does not explicitly model reagent suitability jointly conditioned on the reaction and functional category.

Statistical baseline

- **Train-frequency.** This nonparametric baseline ignores reaction structure. It ranks each reagent by its positive-example frequency within the corresponding category in Train, with deterministic identifier-based tie breaking. Every test reaction within a category therefore shares the same ranking. This baseline measures performance attributable only to reagent popularity and cannot meaningfully rank reagents absent from Train.

A.4.2 Evaluation Tasks and Metrics

We evaluated models at three levels: reagent recommendation for individual examples, reagent-category recommendation, and HTE subspace selection. The first two tasks assess recovery of observed reagent categories and specific reagents from a reaction. The third assesses whether high-yield conditions remain accessible under a finite experimental budget.

Reagent recommendation for individual examples

The input consists of reaction x_i and a candidate reagent pool. The model ranks candidates by reaction-reagent compatibility and receives neither ground-truth reagent labels nor experimental yields at evaluation time.

We evaluated seen and unseen examples separately. Target reagents in Seen examples occurred during training, whereas at least one target reagent in an Unseen example did not. Let the test set contain N reaction examples, with normalized yield $y_i \in [0, 1]$. Let G_i^+ be the set of ground-truth reagents across observed categories and $r_i^{\min}$ the best rank assigned to any ground-truth reagent. All ranking metrics are yield-weighted, assigning greater weight to higher-yield reactions.

- **Weighted mean reciprocal rank (wMRR, ↑):**

$$\text{wMRR} = \frac{\sum_{i=1}^{N} y_i / r_i^{\min}}{\sum_{i=1}^{N} y_i}. \tag{46}$$

- **Weighted mean positive rank (wMPR, ↓):** Let r_{ig} be the rank of ground-truth reagent $g \in G_i^+$.

$$\text{wMPR} = \frac{\sum_{i=1}^{N} y_i \sum_{g \in G_i^+} r_{ig}}{\sum_{i=1}^{N} y_i |G_i^+|}. \tag{47}$$

Lower wMPR indicates that multiple ground-truth reagents for a reaction are collectively ranked closer to the top.

- **Weighted all-positive coverage at K (wAPC@K, ↑):** For $K \in \{1, 3, 5, 10, 20\}$ and the predicted Top-K set $\widehat{G}_i^K$,

$$\text{wAPC@}K = \frac{\sum_{i=1}^{N} y_i \, \mathbb{I}(G_i^+ \subseteq \widehat{G}_i^K)}{\sum_{i=1}^{N} y_i}. \tag{48}$$

This is an all-or-nothing example-level criterion: an example is counted only when all ground-truth reagents appear in the Top-K set.

- **Weighted all-positive preference (wAPP, ↑):** Let G_i^- be the explicit negative set and $\mathcal{I}_{\text{neg}}$ the test examples containing valid explicit negatives.

$$\text{wAPP} = \frac{\sum_{i \in \mathcal{I}_{\text{neg}}} y_i \, \mathbb{I}\left(\max_{g \in G_i^+} r_{ig} < \min_{g \in G_i^-} r_{ig}\right)}{\sum_{i \in \mathcal{I}_{\text{neg}}} y_i}. \tag{49}$$

Category recommendation

This task takes reaction x_i as input. The model ranks all categories in C using category-matching scores $s_{i,m}^c$ and returns the five highest-scoring categories. The task evaluates category identification only, not within-category reagent ranking.

Ground-truth categories are defined by the reaction family rather than inferred individually from the condition fields of each record. In the fixed-label evaluation protocol, every Suzuki reaction has

$$C_i^+ = \{\text{Catalyst}, \text{Ligand}, \text{Base}, \text{Solvent}\}, \tag{50}$$

whereas every Buchwald reaction has

$$C_i^+ = \{\text{Ligand}, \text{Additive}, \text{Base}, \text{Solvent}\}. \tag{51}$$

Thus, $|C_i^+| = 4$ for every evaluated reaction. Let $\widehat{C}_i^5$ the predicted Top-5 set.

- **Truth Coverage@5 (↑):** the micro-averaged recall over ground-truth categories,

$$\text{Coverage@5} = \frac{\sum_{i=1}^{N} |C_i^+ \cap \widehat{C}_i^5|}{\sum_{i=1}^{N} |C_i^+|}. \tag{52}$$

- **All-Truth Hit Rate@5 (↑):** the fraction of examples for which Top-5 predictions contain every ground-truth category,

$$\text{AllHit@5} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left(C_i^+ \subseteq \widehat{C}_i^5\right). \tag{53}$$

- **Precision@5 (↑):** the proportion of Top-5 predictions corresponding to ground-truth categories,

$$\text{Precision@5} = \frac{\sum_{i=1}^{N} |C_i^+ \cap \widehat{C}_i^5|}{\sum_{i=1}^{N} |\widehat{C}_i^5|}. \tag{54}$$

Because each example returns five categories, the denominator is generally $5N$.

HTE subspace selection

The input is an HTE dataset and the candidate reagents available under each category. A model ranks candidates within each category and retains the highest-ranked reagents under a specified budget. The Cartesian product of retained reagents defines the recommended reaction-condition subspace. The model does not directly predict combination yields; instead, it compresses the candidate space to preserve high-yield conditions in a smaller screen.

We used reagent budgets of 20%, 30%, 40%, 50%, 60%, and 80%. The budget is the retained fraction within each category, rounded by a common rule while preserving at least one reagent per category. Metrics were first computed within each HTE subset and then macro-averaged across subsets, giving equal weight to subsets of different sizes.

A screening budget $b \in (0, 1]$ specifies the proportion of reagents retained in each category independently. For a category with n reagents, the number retained is $m = \max\big(1, \text{round}(b\,n)\big)$, where the outer max guarantees at least one reagent per category so that a valid subspace always exists. Given a ranking of the reagents in a category, the top m candidates were retained, and all of their combinations were enumerated to form the selected subspace.

At $b = 20\%$, the retained counts were $\text{round}(0.2 \times 25) = 5$ catalysts, $\text{round}(0.2 \times 12) = 2$ ligands, $\text{round}(0.2 \times 8) = 2$ bases, and $\text{round}(0.2 \times 4) = 1$ solvent, yielding a subspace of $5 \times 2 \times 2 \times 1 = 20$ reactions.

The budget applies to each reagent dimension separately rather than to the total number of reactions. Retaining 20% of the reagents in every category therefore does not retain 20% of the 9,600 reactions ($0.2 \times 9{,}600 = 1{,}920$ combinations), but only 20 combinations, approximately 0.21% of the full

space, because the per-category reductions compound multiplicatively across the four categories. We adopt the per-dimension definition because candidate selection is performed within each reagent category before the combinations are enumerated.

Let $\mathcal{S}_l^{(b)}$ be the recommended subspace for subset l under budget b, $\mathcal{X}_l$ the full experimental space, and $y_l(x)$ the normalized yield of condition combination x.

- **Maximum yield (↑):**

$$\mathrm{MaxYield}_l(b) = \max_{x \in \mathcal{S}_l^{(b)}} y_l(x). \tag{55}$$

- **Mean achievable maximum yield (MAMY, ↑):** for L HTE subsets,

$$\mathrm{MAMY}(b) = \frac{1}{L} \sum_{l=1}^{L} \mathrm{MaxYield}_l(b). \tag{56}$$

MAMY measures whether high-yield conditions remain accessible across HTE tasks, especially under small budgets.

A.4.3 Implementation Details

Model configuration

The model encoded reactions with differential reaction fingerprints (DRFP), condition categories with a frozen Llama-3.1-8B (Grattafiori et al., 2024) backbone, and reagents with Morgan fingerprints. All three representations were projected into a shared 256-dimensional space. A single-layer fusion module with hidden dimension 256 generated the category-conditioned query representation. Details of configuration is shown in Table 10.

Training and optimization

We trained the model for 10 epochs using AdamW, cosine learning-rate decay, and a 10% warm-up ratio. The per-device batch size was 80, with three gradient-accumulation steps. Details of hyperparameters is shown in Table 11

A.5 Supplementary Experimental Results

A.5.1 Visualization of Encoder Representations

As illustrated in Figure 5(left), the reaction embeddings formed three largely separated regions. Michael reactions occupied a compact region distinct from the principal Suzuki and indole clusters. Suzuki and indole examples also showed class-specific concentration, although each class contained smaller satellite groups. The category projection shown in Figure 5(right) similarly revealed dense catalyst-, ligand-, and solvent-dominated regions, together with a diffuse mixed region containing examples from all three categories. These patterns indicate that the encoders captured substantial reaction-family and category structure without collapsing each label into a perfectly separated cluster.

A.5.2 Visualization of HTE-OOD Subspace from Two Architectures

The location and density of high-yield conditions varied markedly across the 15 reaction subsets. Several subsets contained compact high-yield regions, most visibly Subsets 1, 7, 8, 10, and 11, whereas others contained few or no conditions above 90%. As shown in Figure 6, in subsets with concentrated high-yield regions, the hierarchical model's selections more often overlapped or bordered these regions, while pairwise selections were frequently displaced toward lower-density areas. This contrast was most apparent in Subsets 7, 8, 10, and 11.

Table 10: Model configuration.

Hyperparameter	**Value**
Reaction encoder	
Representation	DRFP
Fingerprint dimension	2,048
Radius	3
Category encoder	
Backbone	Llama-3.1-8B
Pooling	Last-token pooling
Maximum sequence length	256 tokens
Backbone parameters	Frozen
Reagent encoder	
Representation	Morgan fingerprint
Fingerprint dimension	2,048
Radius	2
Chirality	Included
Shared representation	
Embedding dimension d	256
Projection head	
Hidden dimension / layers	768 / 1
Category-conditioned query fusion	
Hidden dimension / layers	256 / 1

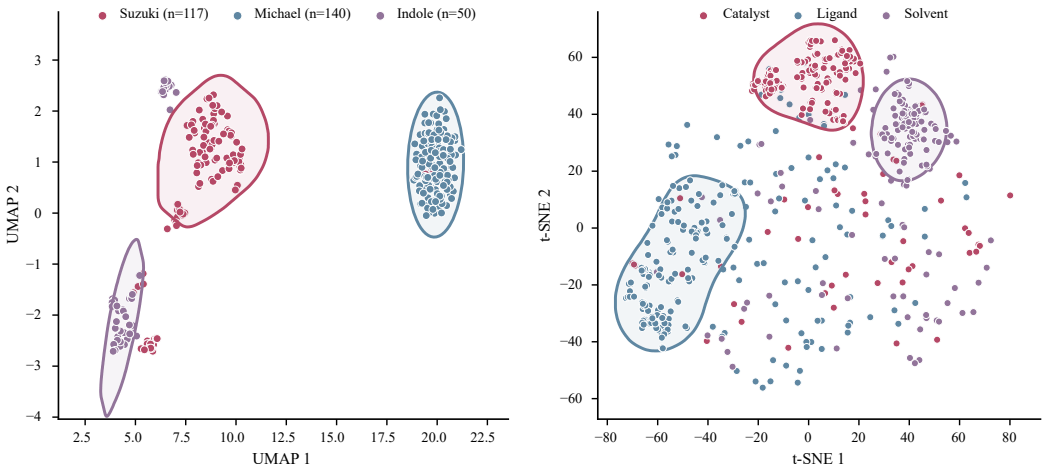

Figure 5: Low-dimensional visualization of learned reaction and condition-category representations. The left panel shows a UMAP projection of reaction representations for Suzuki (n = 117), Michael (n = 140), and indole (n = 50) reactions. The right panel shows a t-SNE projection of category representations for catalysts, ligands, and solvents. Ellipses indicate the principal regions occupied by the corresponding groups and are intended as visual guides.

Table 11: Training and optimization hyperparameters.

Hyperparameter	**Value**
Optimization	
Learning rate	1×10^{-4}
Epochs	10
Per-device batch size	80
Gradient-accumulation steps	3
Weight decay	0.01
Learning-rate scheduler	Cosine decay
Warm-up ratio	0.10
Multi-positive InfoNCE	
Category temperature, τ_{cat}	0.07
Reagent temperature, τ_{reag}	0.07
Category-loss weight, λ_{cat}	0.3
Reagent-loss weight, λ_{reag}	4.0
Positive-reagent coverage	
Loss weight, γ	1.0
Hard-negative sampling	
Maximum candidates/category, K_{neg}	128
Frequency-based sampling ratio, ρ	0.5
Category reweighting	
Effective-number parameter, β_c	0.999
Maximum category weight, $w_{\max}$	10.0
Explicit-negative loss	
Loss weight, λ_{neg}	1.0
Ranking margin, μ_{neg}	0.2
Yield-aware weighting	
Low-yield threshold, y_{low}	0.2
High-yield threshold, y_{high}	0.7
Maximum positive weight, $w_y^{\max}$	2.0
Low-yield loss	
Loss weight, λ_{low}	0.2
Penalty threshold, μ_{low}	0.0

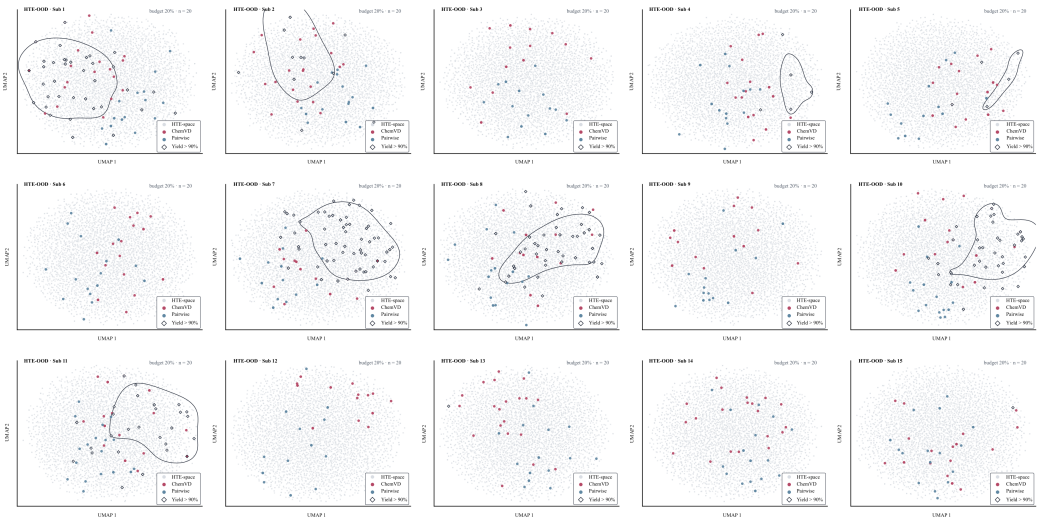

Figure 6: HTE-OOD recommendations from the hierarchical model and the pairwise architecture at a 20% budget. Each panel corresponds to one of the 15 HTE-OOD reaction subsets. Light-gray points denote the full HTE space, red and blue points denote conditions selected by the hierarchical model and pairwise architecture, respectively, and diamonds mark conditions with yield above 90%. Contours highlight visually concentrated high-yield regions where present. Each method selected 20 conditions per subset.

A.5.3 Detailed Results on HTE-OOD

Table 12 provides the full numerical evaluation across all screening budgets and model scales, further corroborating ChemVD's competitive screening efficiency and strong parameter efficiency. In the most demanding low-budget regime (20%), the highest MAMY is achieved by LLaMa-3.3-70B (72.16%), followed by LLaMa-3.1-70B (71.26%) and ChemVD (68.78%). Although ChemVD is not the absolute best at this budget, its 8B model remains within 3.38 percentage points of the strongest 70B dense baseline and substantially outperforms much larger MoE models, such as DeepSeek-v4-pro (48.34%) and DeepSeek-v4-flash (45.30%). At 30%, ChemVD (73.26%) is only 0.79 percentage points behind LLaMa-3.3-70B (74.05%). ChemVD achieves the highest MAMY at 40% (77.74%) and 80% (87.93%); at 80%, despite convergence due to search-space saturation, it still surpasses LLaMa-3.3-70B (85.19%) and DeepSeek-v4-pro (84.58%). These results confirm that domain-tailored condition prioritization provides a favorable parameter–efficiency trade-off over sheer parameter scaling, enabling ChemVD to deliver top-tier subspace design with exceptional parameter and computational efficiency.

Table 12: Comparison of MAMY under different screening budgets.

Model	**#Params (B)**		**MAMY (%)**					
	Total	**Active**	**20%**	**30%**	**40%**	**50%**	**60%**	**80%**
ChemVD	8.0	8.0	$68.78_{\pm 1.93}$	$73.26_{\pm 1.51}$	$77.74_{\pm 2.13}$	$79.62_{\pm 1.52}$	$81.51_{\pm 0.88}$	$87.93_{\pm 1.82}$
Pairwise	8.0	8.0	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$1.60_{\pm 2.77}$	$4.03_{\pm 6.99}$	$85.99_{\pm 1.82}$
Qwen3-30B	30.0	3.0	$65.46_{\pm 2.25}$	$67.72_{\pm 2.23}$	$76.29_{\pm 3.18}$	$79.20_{\pm 0.27}$	$80.79_{\pm 1.14}$	$84.46_{\pm 0.64}$
Qwen3.5-35B	35.0	3.0	$58.29_{\pm 3.11}$	$61.54_{\pm 3.71}$	$66.31_{\pm 0.97}$	$69.74_{\pm 1.02}$	$70.20_{\pm 0.95}$	$77.58_{\pm 3.94}$
LLaMa-3.1-70B	70.0	70.0	$71.26_{\pm 0.77}$	$72.71_{\pm 0.69}$	$75.51_{\pm 0.62}$	$80.57_{\pm 0.62}$	$83.14_{\pm 0.94}$	$85.09_{\pm 0.09}$
LLaMa-3.3-70B	70.0	70.0	$72.16_{\pm 0.00}$	$74.05_{\pm 0.00}$	$77.08_{\pm 0.00}$	$81.22_{\pm 0.00}$	$82.39_{\pm 0.00}$	$85.19_{\pm 0.00}$
DeepSeek-v4-flash	284.0	13.0	$45.30_{\pm 1.67}$	$57.51_{\pm 1.37}$	$71.74_{\pm 1.96}$	$76.44_{\pm 3.94}$	$79.95_{\pm 0.35}$	$85.47_{\pm 0.73}$
DeepSeek-v4-pro	1600.0	49.0	$48.34_{\pm 2.30}$	$59.48_{\pm 2.92}$	$75.52_{\pm 3.07}$	$79.95_{\pm 1.88}$	$81.01_{\pm 1.73}$	$84.58_{\pm 0.37}$

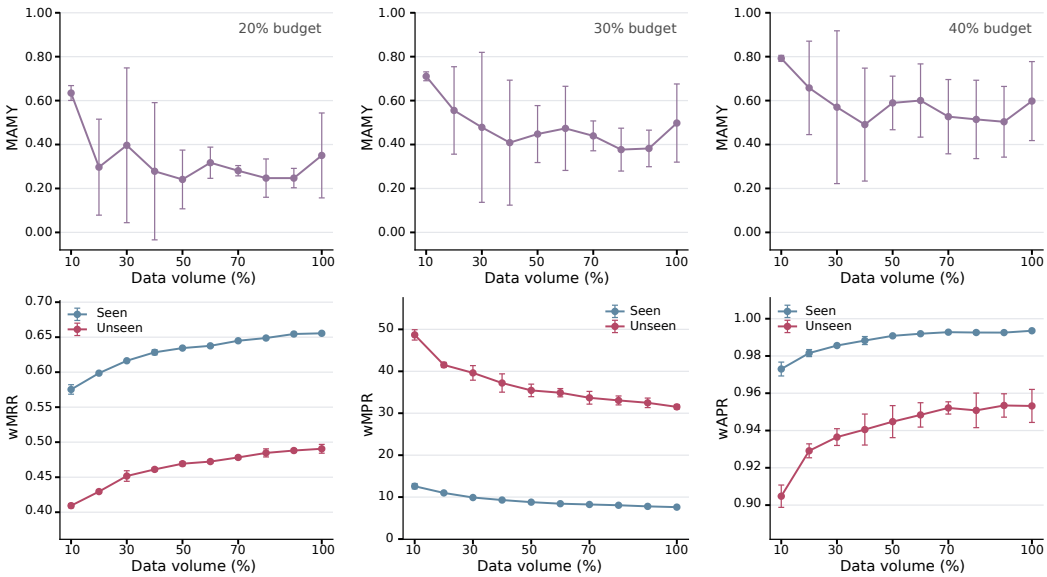

Figure 7: Performance as a function of training-data volume. The top row reports MAMY under 20%, 30%, and 40% HTE budgets. The bottom row reports wMRR, wMPR, and wAPP for Seen and Unseen reagent evaluations. Points show the reported central estimates, and error bars show the variability measure used in the experiment.

A.5.4 Effect of Training-Data Volume

Reagent-ranking performance improved consistently as the training fraction increased. From 10% to 100% of the data, Seen wMRR rose from approximately 0.58 to 0.66, while Unseen wMRR increased from approximately 0.41 to 0.49. Over the same range, wMPR decreased from approximately 12.5 to 7.5 for Seen examples and from approximately 49 to 32 for Unseen examples. The wAPP curves also increased, from approximately 0.97 to 0.99 for Seen examples and from approximately 0.91 to 0.95 for Unseen examples. Improvements were largest at smaller data fractions and diminished as the dataset approached full size, particularly for Seen examples. The persistent Seen-Unseen gap indicates that additional data improved open-set ranking but did not eliminate the difficulty of recommending reagents absent from training.

MAMY did not exhibit the same monotonic trend. At all three budgets, the estimate was highest at the 10% training fraction, dropped sharply at 20%, and then fluctuated as more data were added. Variability was especially large at 20%-40% of the data and narrowed at several larger fractions. Increasing the HTE budget shifted MAMY upward overall, as expected from evaluating a larger recommended subspace, but did not remove the non-monotonic dependence on training volume.

A.5.5 Robustness to Encoder Choices

As shown in Table 13, the model remained broadly robust to different reaction and language encoders when the Morgan reagent representation was retained, achieving consistently high performance on Reagent-Split and competitive results on HTE-OOD. In contrast, replacing Morgan with the ChemT5-based reagent encoder led to a noticeable performance degradation, highlighting the importance of reagent representation. We therefore selected the red-highlighted combination of DRFP, LLaMA-3.1-8B, and Morgan as our final model, as it provided the best overall balance across both benchmarks.

Table 13: Comparison of reaction, language-model, and reagent encoder combinations on Reagent-Split and HTE-OOD.

Encoder type			Reagent-Split Overall			HTE-OOD	
Reaction	**Category**	**Reagent**	**wMRR ↑**	**wMPR ↓**	**wAPP ↑**	**MAMY@0.20 ↑**	**MAMY@0.30 ↑**
DRFP	LLaMA-3.1-8B	Morgan	64.34	8.862	99.27	68.78	73.26
DRFP	LLaMA-3.1-8B	D-MPNN	61.03	11.679	98.40	48.98	64.11
RxnFP	LLaMA-3.1-8B	Morgan	61.07	8.946	99.39	63.67	71.46
ChemT5-Base	LLaMA-3.1-8B	Morgan	60.80	8.966	99.12	48.85	65.06
DRFP	LLaMA-3.1-8B	ChemT5-Base	47.54	19.556	97.47	30.52	39.72
ChemT5-Base	LLaMA-3.1-8B	ChemT5-Base	46.16	21.469	97.31	55.97	61.18
DRFP	ChemT5-Base	Morgan	63.49	9.215	99.17	63.67	69.21
DRFP	Qwen3-8B	Morgan	64.72	8.658	99.15	53.92	70.66
DRFP	Qwen3-Embedding-8B	Morgan	64.63	8.672	99.18	52.08	68.97