VICOR: RELIABLE MOLECULAR STRUCTURE EXTRACTION VIA SPATIALLY ALIGNED VERIFICATION AND EXECUTABLE REVISION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reliable optical chemical structure recognition (OCSR) is essential for building high-quality chemical data from scientific literature, yet even small recognition errors can propagate into chemical databases and downstream models. In practice, recognized structures often require manual inspection and correction before use, making large-scale data curation costly and difficult to scale. We therefore study *Selective Structure Recognition (SSR)*, a post-recognition setting that automatically produces reliable structured outputs while rejecting unresolved cases. Selection-only approaches can improve reliability by rejection, but cannot create additional correct outputs beyond those produced by the base recognizer. We propose **ViCoR**, a *repair-before-rejection* framework for iterative **V**er**I**fi**C**ati**O**n and **R**evision. Its key idea is to make observation–prediction correspondence explicit: coordinate-preserving rendering establishes spatial correspondence between the source image and predicted structure, while index anchoring maps localized visual discrepancies to executable graph edits without full-structure regeneration. A shared VLM is progressively trained from verification to revision. On two real-world OCSR benchmarks, ViCoR improves overall accuracy from 73.53% to 88.26% and from 61.83% to 84.32%, while achieving over 97% accepted accuracy at 85–89% coverage. The resulting molecular data further improve reaction-extraction F1 by 15.5 and literature-sourced reaction prediction accuracy by 7.7 and 5.8 points, demonstrating the value of automated reliability control for scientific data curation and downstream chemical learning. Codes to be released.

1 INTRODUCTION

Extracting molecular structures from chemical literature and patents is fundamental to building machine-readable chemical knowledge bases (Fang et al., 2025; Fan et al., 2024). Because molecular structures are predominantly communicated as two-dimensional depictions, optical chemical structure recognition (OCSR) is used to convert these images into structured representations such as SMILES (Weininger, 1988) or molecular graphs (Qian et al., 2023). Recent OCSR systems have achieved strong recognition performance, enabling increasingly large-scale extraction from scientific documents (Morin et al., 2024). However, recognition errors remain inevitable in real-world deployment at scale, and even a single incorrect atom or bond can change the identity of the entire molecule. Such errors often require costly and time-consuming human correction before extracted structures can be reliably used in chemical databases or downstream model training (Tan et al., 2026; Nippa et al., 2024). This raises a practical question beyond improving the recognizer itself: *can we automatically turn imperfect OCSR predictions into reliable molecular data?*

A natural post-recognition solution is to identify unreliable predictions and reject them before downstream use. However, because selection can only retain or discard predictions already produced by recognizer, improving reliability comes at the cost of reduced coverage. This limitation is particularly wasteful for OCSR, where many recognition errors are localized to only a few atoms or bonds while most of the structure is already correct. We therefore study *Selective Structure Recognition (SSR)*, a post-recognition setting that seeks reliable structured outputs while allowing unresolved cases to be rejected. Rather than relying on rejection alone, we pursue *repair-before-rejection* principle: repair incorrect predictions when possible, and reject them only when they remain unresolved.

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.

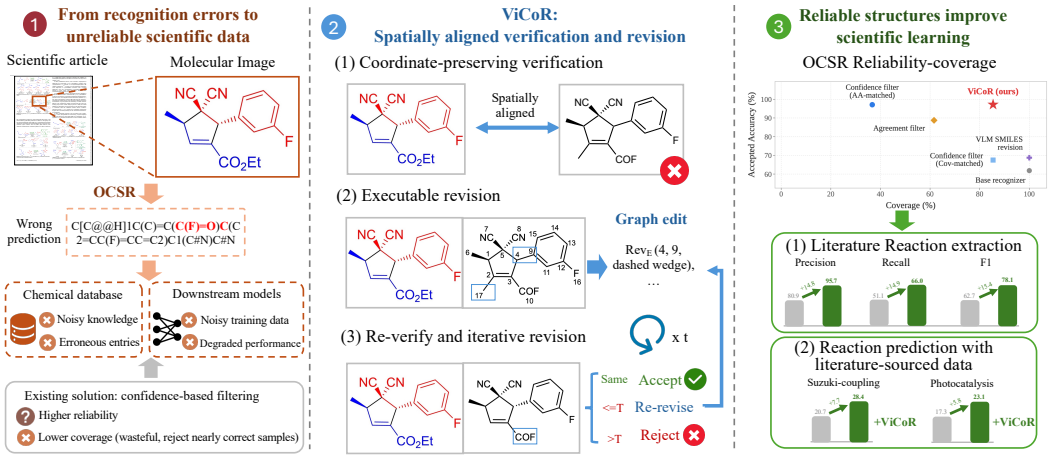


Figure 1: **From unreliable OCSR outputs to reliable scientific data with ViCoR.** Recognition errors can corrupt chemical data, while filtering improves reliability at the cost of coverage. ViCoR uses spatially aligned verification and executable revision to recover correct structures, improving both the reliability–coverage trade-off and downstream chemical learning.

The key challenge is not merely determining whether a structured prediction is incorrect, but grounding the error in the structure. This requires comparing the prediction with the source image, localizing the visual inconsistency, and associating it with the corresponding structured entity. However, the two representations are not naturally aligned: SMILES does not preserve the spatial layout of the molecule, while a molecular graph does not explicitly associate its atoms and bonds with visual regions. Reliable verification and localized correction therefore share a central challenge: establishing explicit correspondence between the source observation and the structured prediction.

To address these challenges, we introduce **ViCoR**, a post-recognition framework that makes observation–prediction correspondence explicit. ViCoR re-renders the current prediction using its recognized spatial layout, turning cross-representation alignment into direct spatial comparison with the source image. For revision, ViCoR augments the rendered prediction with atom indices, providing explicit anchors that associate localized visual discrepancies with specific graph entities and executable graph edits. Together, these designs establish a grounded correction path from *visual inconsistency* to *graph entity* to *structural edit*, allowing incorrect regions to be revised while preserving the remaining structure. The revised prediction is then iteratively re-rendered and re-verified, enabling multiple localized errors to be resolved before unresolved cases are rejected. Verification and revision share this aligned visual interface and are implemented by a single VLM, progressively trained on 1.3M synthetic samples.

We evaluate ViCoR on two newly curated in-the-wild benchmarks and eight standard OCSR benchmarks. On real-world benchmarks, ViCoR raises overall accuracy from 73.53% to 88.26% and from 61.83% to 84.32%, while retaining over 97% accepted accuracy at 85–89% coverage. These gains translate to downstream chemical tasks, improving reaction-extraction F1 by 15.45 and literature-sourced reaction-prediction exact match by 7.7/5.8 points on Suzuki coupling/photocatalysis.

In summary, our main contributions are highlighted as follows:

- We formulate *Selective Structure Recognition (SSR)* as a post-recognition problem for reliable structure prediction, and show that selection-only methods cannot produce more correct data than base recognizer already predicts, motivating *repair-before-rejection* strategy.
- We introduce **ViCoR**, which repurposes spatial information from the initial structured prediction to establish explicit correspondence with the source image. Coordinate-preserving rendering enables direct spatial comparison, while index anchoring maps localized discrepancies to executable graph edits, forming an iterative verification–revision process.
- Extensive experiments across ten OCSR benchmarks show that ViCoR improves both recognition accuracy and the reliability–coverage trade-off. The recovered molecular structures further improve reaction extraction and reaction prediction from literature-sourced data, demonstrating their value for constructing reliable scientific datasets.

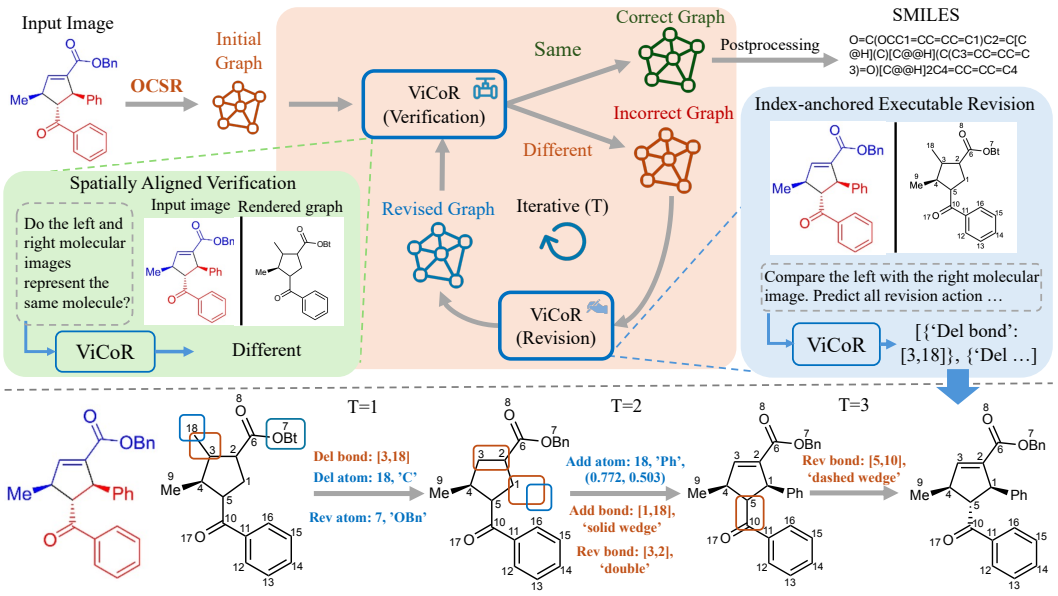


Figure 2: **Overview of ViCoR.** Starting from an OCSR-predicted initial graph, ViCoR iteratively verifies and revises the prediction against the source image. Coordinate-preserving rendering establishes spatial correspondence for verification, while atom indexing maps localized visual discrepancies to executable graph edits for revision. Each revised graph is re-rendered and re-verified until it is accepted or reaches the maximum iteration T.

2 Related Work

Optical Chemical Structure Recognition (OCSR). OCSR converts molecular images into structured representations. Existing methods range from rule-based pipelines (Peryea et al., 2019; Filippov et al., 2009) to end-to-end SMILES generation (Fang et al., 2025; Rajan et al., 2020) and graph-based recognition (Qian et al., 2023; Morin et al., 2023; Chen et al., 2024). VLMs have also been explored for broader chemical figure understanding and reaction extraction (Chen et al., 2025a; Song et al., 2025; Chen et al., 2025b; Leong et al., 2025). Despite strong recognition performance, the remaining structural errors often require manual inspection and correction before extracted molecules can be reliably used in chemical databases or downstream learning. Our work targets this complementary post-recognition stage, aiming to automatically produce reliable structured outputs from imperfect OCSR predictions while rejecting unresolved cases.

Selective Prediction and Post-Recognition Refinement. Selective prediction improves reliability by rejecting uncertain outputs, leading to the classical risk–coverage trade-off (El-Yaniv et al., 2010; Geifman & El-Yaniv, 2017; 2019). Recent multimodal methods reduce unnecessary abstention by gathering additional visual evidence, but still leave the underlying structured prediction unchanged (Srinivasan et al., 2024). A complementary line of work refines model outputs through critique, verification, or iterative feedback (Madaan et al., 2023; Gou et al., 2024), primarily for free-form generation. Image-to-structure correction poses an additional challenge: a localized visual discrepancy must be associated with the corresponding structured entity and translated into a valid edit. ViCoR makes this correspondence explicit through coordinate-preserving alignment and index-anchored executable graph edits, while rejecting cases that remain unresolved.

3 Method

We first formulate *Selective Structure Recognition (SSR)* as a post-recognition problem that seeks reliable structured outputs from the predictions of an existing recognizer. To move beyond selection-only rejection, we propose **ViCoR**, which follows a *repair-before-rejection* principle and iteratively verifies and revises incorrect predictions before rejecting unresolved cases. For OCSR, ViCoR establishes explicit spatial correspondence through coordinate-preserving rendering and uses index anchoring to map localized visual discrepancies to executable graph edits. Finally, we progressively train a shared VLM from verification to revision to support both operations in the same framework.

3.1 TASK DEFINITION: SELECTIVE STRUCTURE RECOGNITION

We study *Selective Structure Recognition (SSR)*, a post-recognition setting that operates on structured predictions produced by an existing recognizer. Given an initial prediction together with its source observation, SSR seeks to produce a reliable structured output for downstream use, while rejecting cases that cannot be resolved reliably.

In this work, we instantiate SSR for optical chemical structure recognition (OCSR) (Qian et al., 2023). Given a set of molecular images $\{I_k\}_{k=1}^{N}$, an base OCSR recognizer $\mathcal{A}$ first the predicts initial molecular structure $Y_k^{(0)} \in \mathcal{Y}$ for each image:

$$Y_k^{(0)} = \mathcal{A}(I_k), \tag{1}$$

A post-recognition system $\mathcal{F}$ then processes each image–prediction pair and returns a final candidate together with an acceptance decision:

$$(\widetilde{Y}_k, a_k) = \mathcal{F}(I_k, Y_k^{(0)}), \qquad \widetilde{Y}_k \in \mathcal{Y}, \quad a_k \in \{0, 1\}, \tag{2}$$

where $a_k = 1$ denotes acceptance and $a_k = 0$ denotes rejection. Let $Y_k^\star \in \mathcal{Y}$ denote the ground-truth structure. We evaluate $\mathcal{F}$ using three complementary metrics:

$$\mathrm{OA} = \frac{1}{N}\sum_{k=1}^{N} \mathbb{I}[\widetilde{Y}_k = Y_k^\star], \quad \mathrm{AA} = \frac{\sum_{k=1}^{N} a_k \mathbb{I}[\widetilde{Y}_k = Y_k^\star]}{\sum_{k=1}^{N} a_k}, \quad \mathrm{Cov} = \frac{1}{N}\sum_{k=1}^{N} a_k. \tag{3}$$

Overall Accuracy (OA) measures the accuracy of the final structured outputs over the full input set, while Accepted Accuracy (AA) measures the reliability of accepted predictions and Coverage (Cov) measures the fraction of inputs accepted. Since the preferred reliability–coverage trade-off depends on downstream requirements, SSR does not assume a universally optimal operating point. We say that $\mathcal{F}_a$ *Pareto-dominates* $\mathcal{F}_b$, denoted by $\mathcal{F}_a \succ \mathcal{F}_b$, if

$$\mathrm{AA}(\mathcal{F}_a) \geq \mathrm{AA}(\mathcal{F}_b), \qquad \mathrm{Cov}(\mathcal{F}_a) \geq \mathrm{Cov}(\mathcal{F}_b), \tag{4}$$

with at least one inequality being strict. SSR therefore seeks a better reliability–coverage trade-off, balancing high reliability with broad coverage.

3.2 ITERATIVE VERIFICATION AND REVISION

A natural approach to SSR is *selection-only post-recognition*: accept the judged-reliable predictions and reject the rest without changing their structured outputs.

Selection-only ceiling. Consider a selection-only system $\mathcal{F}_{\mathrm{sel}}$, for which $\widetilde{Y}_k = Y_k^{(0)}$ for every sample. Since selection cannot turn an incorrect prediction into a correct one,

$$\begin{aligned} &\mathrm{AA}\cdot\mathrm{Cov} = \frac{1}{N}\sum_{k=1}^{N} a_k\, \mathbb{I}\Big[Y_k^{(0)} = Y_k^\star\Big] \leq \mathrm{OA}^{(0)}, \\ &\mathrm{AA} \geq \tau \Longrightarrow \mathrm{Cov} \leq \min\left(1, \frac{\mathrm{OA}^{(0)}}{\tau}\right). \end{aligned} \tag{5}$$

Thus, with the base accuracy $\mathrm{OA}^{(0)}$ fixed, the maximum attainable coverage at any target reliability τ is also bounded. Selection can only move along this reliability–coverage trade-off. Improving the attainable coverage at a given reliability therefore requires increasing the overall number of correct predictions. This motivates repairing incorrect predictions before rejecting unresolved cases.

Repair Before Rejection. OCSR provides an opportunity to move beyond the selection-only ceiling. Many recognition errors are localized to a small number of atoms or bonds, while most of the predicted structure remains correct (Chen et al., 2024). Rather than directly rejecting such predictions, ViCoR follows a *repair-before-rejection* principle: it first attempts to revise correctable errors and rejects a prediction only if it remains unresolved. ViCoR realizes this principle through verification and revision. Given an input image I and the current prediction $Y^{(t)}$, the verifier $\mathcal{V}$ determines whether the prediction is consistent with the source observation:

$$\mathcal{V}(I, Y^{(t)}) \in \{0, 1\}. \tag{6}$$

If verified, $Y^{(t)}$ is accepted. Otherwise, the reviser $\mathcal{R}$ predicts a structured update

$$\Delta^{(t)} = \mathcal{R}(I, Y^{(t)}), \qquad Y^{(t+1)} = Y^{(t)} \oplus \Delta^{(t)}, \tag{7}$$

where $\oplus$ applies the predicted update to the current structure.

Iterative Verification and Revision. Since challenging samples often contain multiple localized errors, a single-trial revision is usually insufficient to perfectly repair the entire structure. ViCoR therefore re-verifies each revised prediction and performs another revision only if inconsistencies remain. As shown in Alg. 1, verification and revision alternate from $Y^{(0)}$ until the prediction is verified or the maximum of T revisions is reached. Verification therefore serves both as the acceptance gate and an adaptive stopping criterion: verified predictions are accepted immediately, while unresolved ones proceed to further revision. Predictions that remain unverified after T revisions are rejected.

Algorithm 1: Iterative verification and revision.

Input: $I, Y^{(0)}, T$
for $t = 0, \ldots, T$ **do**
  if $\mathcal{V}(I, Y^{(t)}) = 1$ **then**
    return $(Y^{(t)}, 1)$
  end if
  if $t < T$ **then**
    $Y^{(t+1)} \leftarrow Y^{(t)} \oplus \mathcal{R}(I, Y^{(t)})$
  end if
end for
return $(Y^{(T)}, 0)$

3.3 Spatially Aligned Verification and Index-Anchored Revision

Verification and revision require comparing a structured prediction with its source image, yet the two are represented in different spaces. Typical OCSR outputs such as SMILES discard spatial layout, while molecular graphs encode structure without explicitly associating atoms and bonds with visual regions. ViCoR closes this representation gap in image space: coordinate-preserving rendering establishes spatial correspondence for verification, while atom indexing makes the aligned graph entities directly addressable for revision.

Spatially Aligned Visual Verification. At each iteration, we represent the current structured prediction $Y^{(t)}$ as a molecular graph

$$G = (V, E, X), \tag{8}$$

where $V = \{v_i\}_{i=1}^{n}$ denotes the atoms, $E \subseteq V \times V$ denotes the bonds, and $X = \{(x_i, y_i)\}_{i=1}^{n}$ contains the corresponding two-dimensional atom locations provided by the graph-based OCSR recognizer. Each atom v_i has a label $\ell_i \in \mathcal{L}$, and each edge $(v_i, v_j) \in E$ has a bond type $b_{ij} \in \mathcal{B}$.

ViCoR renders the predicted graph using its recognized coordinates:

$$\widetilde{I} = \rho(G; \mathbf{X}), \tag{9}$$

where ρ denotes coordinate-preserving molecular rendering. Unlike canonical-layout rendering, which may rearrange the molecular structure, coordinate-preserving rendering keeps corresponding atoms and bonds at approximately matching locations in the source and rendered images.

We then place the source image and rendered prediction side by side and use the resulting interface for verification:

$$P(I, G) = I \,\|\, \widetilde{I}, \qquad \mathcal{V}(P(I, G)) \in \{0, 1\}, \tag{10}$$

where $\|$ denotes side-by-side presentation. This keeps both views separately visible while placing spatially corresponding regions in a shared visual context, enabling direct spatial comparison between the source image and the predicted structure.

Index-Anchored Executable Revision. Verification only needs to determine whether an inconsistency exists, whereas revision must additionally identify the graph entity associated with it. We therefore overlay a unique index on each predicted atom and construct

$$\widetilde{I}_{\text{idx}} = \rho_{\text{idx}}(G; \mathbf{X}), \qquad P_{\text{idx}}(I, G) = I \,\|\, \widetilde{I}_{\text{idx}}. \tag{11}$$

The atom indices make graph entities explicitly addressable, allowing a localized visual discrepancy to be mapped to a specific atom or bond. Rather than regenerating the complete molecular structure, the reviser takes the indexed interface as input and predicts a sequence of executable graph edits:

$$\mathcal{R}(I, G) = \mathcal{R}(P_{\text{idx}}(I, G)) = \Delta = (\delta_1, \ldots, \delta_K). \tag{12}$$

We define six edit types covering node- and edge-level changes:

$$\Omega_{\text{node}} = \{\text{Add}_V(i, \ell, (x, y)), \text{Del}_V(i), \text{Rev}_V(i, \ell')\}, \tag{13}$$

$$\Omega_{\text{edge}} = \{\text{Add}_E(i, j, b), \text{Del}_E(i, j), \text{Rev}_E(i, j, b')\}. \tag{14}$$

Here, i and j denote atom indices, $\ell, \ell' \in \mathcal{L}$ denote atom labels, $b, b' \in \mathcal{B}$ denote bond types, and $(x, y) \in \mathbb{R}^2$ denotes the normalized location of an inserted atom. The update operator $\oplus$ applies the predicted edits sequentially and skips any edit invalidated by earlier graph modifications. Further details are provided in the Appendix.

3.4 Verification-to-Revision Progressive Training

Verification and revision are built on the same spatially aligned comparison between the source image and the predicted structure. The main difference is that verification only determines whether the two are consistent, whereas revision further identifies the inconsistency and translates it into a structural edit. Based on the shared visual basis, we propose to train a single VLM $\mathcal{M}$, with task-specific prompts controlling the two behaviors:

$$\mathcal{V}(I, G) = \mathcal{M}(P(I, G), p_{\text{ver}}), \qquad \mathcal{R}(I, G) = \mathcal{M}(P_{\text{idx}}(I, G), p_{\text{rev}}). \tag{15}$$

More importantly, this asymmetry motivates our *verification-to-revision progressive training*: we first train the model to recognize structural discrepancies through verification, and then transfer this learned correspondence to the more fine-grained revision task.

Stage 1: Verification-Oriented Pretraining. We first train $\mathcal{M}$ on the spatially aligned interface $P(I, G)$ to distinguish structurally consistent predictions from inconsistent ones. By varying drawing styles and image noise during training, the model learns to ignore appearance differences unrelated to molecular structure and focus on structure-relevant discrepancies. Meanwhile, verification encourages robust spatial correspondence between the paired images, providing a foundation for subsequent localized revision.

Stage 2: Revision-Oriented Fine-Tuning. Starting from the verification-pretrained model, we further train $\mathcal{M}$ on the indexed interface $P_{\text{idx}}(I, G)$ to generate executable graph edits. This extends the learned correspondence from deciding *whether* an inconsistency exists to identifying *where* it occurs and *how* the corresponding graph entity should be revised.

4 Experiment

Training Data Synthesis. We construct 1.3M synthetic training samples from PubChem molecules (Kim et al., 2016), including 800K for verification-oriented pretraining and 500K for revision-oriented fine-tuning. For each sample, the source image is generated with Indigo using randomized drawing styles and image noise, while the candidate graph is rendered by RDKit (Landrum, 2016) under the same coordinate frame with stereochemistry preserved. To better mimic realistic OCSR errors, we also include incorrect predictions produced by MolScribe on synthesized molecular images. Further details are provided in the Appendix.

Evaluation Datasets. We evaluate ViCoR on eight standard OCSR benchmarks, including six realistic datasets (CLEF (Piroi et al., 2010), UOB (Sadawi et al., 2012), JPO (Fujiyoshi et al., 2011), USPTO (Filippov et al., 2009), Staker (Staker et al., 2019), and ACS (Qian et al., 2023)) and two synthetic datasets (Indigo and ChemDraw (Qian et al., 2023)). To evaluate OCSR under recent journal-domain conditions, we curate two in-the-wild benchmarks, OCIE and JIE. They contain low-resolution journal crops with diverse drawing styles, color, visual distractors, and naturally co-occurring challenges such as R-groups, complex abbreviations, and disconnected structures. OCIE contains 1,141 molecular images cropped from the OpenChemIE test set (Fan et al., 2024), while JIE contains 1,779 molecules cropped from 243 recent journal figures (e.g., *JACS*, *Nature*).

Metrics. We evaluate SSR using the three metrics defined in Eq. 3: Overall Accuracy (OA), Accepted Accuracy (AA), and Coverage (Cov). OA measures exact-match accuracy over the full test set, while AA and Cov measure the reliability and coverage of accepted predictions, respectively.

Implementation Details. We use Qwen2.5-VL-3B (Bai et al., 2025) as the shared VLM and fine-tune it with LoRA (Hu et al., 2022). The source and rendered images are horizontally concatenated

Table 1: Selective Structure Recognition on real-world OCSR benchmarks. OA/AA denote exact-match accuracy over all/accepted predictions, and Cov denotes acceptance coverage. Agreement filter denotes that only consistent results are accepted. VLM SMILES revision used GPT-5.6-sol.

Category	Method	OCIE			JIE		
		OA↑	AA↑	Cov↑	OA↑	AA↑	Cov↑
VLM (e2e)	Qwen2.5-VL-3B (Bai et al., 2025)	0.54	0.54	100.0	0.90	0.90	100.0
	GPT-4o (OpenAI, 2026)	13.91	13.91	100.0	23.58	23.58	100.0
	GPT-5.6-sol (OpenAI, 2026)	49.04	49.04	100.0	51.95	51.95	100.0
Expert	OSRA (Filippov et al., 2009)	36.17	36.17	100.0	19.88	19.88	100.0
	DECIMER (Rajan et al., 2020)	35.07	35.07	100.0	16.42	16.42	100.0
	MolGrapher (Morin et al., 2023)	41.20	41.20	100.0	25.40	25.40	100.0
	MolNexTR (Chen et al., 2024)	73.24	73.24	100.0	61.62	61.62	100.0
	MolScribe (Qian et al., 2023)	73.53	73.53	100.0	61.83	61.83	100.0
SSR (with MolScribe)	Confidence filter (AA-matched)	73.53	97.50	52.3	61.83	97.08	36.7
	Confidence filter (Cov-matched)	73.53	80.12	89.0	61.83	67.48	85.4
	Agreement filter with MolNexTR	73.53	91.83	74.3	61.83	88.74	61.6
	VLM SMILES revision	78.91	78.91	100.0	68.74	68.74	100.0
Ours	MolGrapher + ViCoR	76.84 +35.64	95.68 +54.48	78.2	66.95 +41.55	94.81 +69.41	69.8
	MolNexTR + ViCoR	88.08 +14.84	97.44 +24.20	88.9	**84.32** +22.70	97.10 +35.48	85.3
	MolScribe + ViCoR	**88.26** +14.73	**97.54** +24.01	89.0	**84.32** +22.49	**97.17** +35.34	85.4

Table 2: Exact-match accuracy (OA) on eight standard OCSR benchmarks. Δ denotes the gain of ViCoR over its base recognizer, MolScribe.

Category	Method	Synthetic		Realistic					
		Indigo	ChemDraw	CLEF	UOB	JPO	USPTO	Staker	ACS
Rule-based	MolVec (Peryea et al., 2019)	95.4	87.9	82.8	80.6	67.8	88.4	0.8	47.4
	OSRA (Filippov et al., 2009)	95.0	87.3	84.6	78.5	55.3	87.4	0.0	55.3
End-to-end	DECIMER (Rajan et al., 2020)	69.6	86.1	62.7	88.2	55.2	41.1	40.8	46.5
	MolParser (Fang et al., 2025)	—	—	91.0	91.6	75.6	93.0	—	—
	MolSight (Zhang et al., 2026)	—	—	85.5	87.4	57.6	92.0	—	—
Graph-based	MolGrapher (Morin et al., 2023)	—	—	90.5	—	67.5	91.5	—	—
	MolNexTR (Chen et al., 2024)	97.8	95.1	90.4	88.5	82.1	93.8	88.3	81.9
	MolScribe (Qian et al., 2023)	97.5	93.8	88.9	87.4	76.2	93.1	86.9	71.9
Ours	**MolScribe + ViCoR**	**98.5**	**96.4**	**93.8**	**92.8**	**86.5**	**94.7**	**90.7**	**89.7**
	Δ *vs. MolScribe*	+1.0	+2.6	+4.9	+5.4	+10.3	+1.6	+3.8	+17.8

as a single side-by-side input. Normalized insertion coordinates are quantized into 1,000 location tokens, and predicted graph edits are executed sequentially by $\oplus$. We train with batch size 64 using AdamW with weight decay 0.1 for 2 epochs of stage 1 pretraining followed by 1 epoch of stage 2 fine-tuning. We use cosine learning-rate decay with a peak rate of 2×10^{-5} and 100 warm-up steps. All experiments are conducted on 8 NVIDIA RTX 3090 GPUs.

4.1 Selective Structure Recognition for Reliable OCSR

Table 1 compares ViCoR with end-to-end VLMs (e.g., GPT-5.6-sol), expert OCSR models (e.g., DECIMER), and baseline SSR strategies (e.g., confidence filtering) on OCIE and JIE. For VLM SMILES revision, GPT-5.6-sol directly revises the MolScribe prediction conditioned on the source image. ViCoR consistently improves both OA and AA across different graph-based recognizers.

Observation 1: ViCoR improves recognition beyond filtering. With MolScribe as the base recognizer, ViCoR raises OA from 73.53 to 88.26 on OCIE and from 61.83 to 84.32 on JIE. In contrast, selection-only methods leave OA unchanged because they do not modify the initial predictions. Direct VLM SMILES revision improves OA to 78.91 and 68.74, but remains well below ViCoR, highlighting the difficulty of revising structures directly across image and symbolic representations. By establishing spatial correspondence and executable graph edits, ViCoR can instead revise localized recognition errors while preserving the remaining structure. Table 2 further shows consistent OA improvements on eight standard OCSR benchmarks, such as JPO (+10.3) and ACS (+17.8).

Table 3: Chemical reaction extraction performance using OpenChemIE tool.

Methods	Precision	Recall	F1
Raw OCSR	80.85	51.14	62.65
confidence filter	80.70	36.54	50.31
Agreement filter	85.10	44.70	58.61
VLM SMILES rev.	85.60	57.80	69.01
ViCoR (Verify only)	92.30	48.90	63.93
ViCoR (Rev. only)	90.85	62.80	74.26
ViCoR	**95.72**	**65.96**	**78.10**

Table 4: Forward reaction prediction performance using literature-sourced training data from different SSR strategies.

Methods	Suzuki Coupling			Photocatalysis		
	EXACT↑	RDK FTS↑	Morgan↑	EXACT↑	RDK FTS↑	Morgan↑
Raw OCSR	0.207	0.445	0.484	0.173	0.471	0.480
Confidence filtering	0.212	0.440	0.486	0.189	0.486	0.492
Agreement filtering	0.218	0.448	0.490	0.195	0.489	0.496
VLM SMILES rev.	0.234	0.462	0.496	0.198	0.496	0.499
ViCoR (Verify only)	0.220	0.444	0.491	0.201	0.494	0.502
ViCoR (Rev. only)	0.265	0.487	0.523	0.211	0.508	0.509
ViCoR	**0.284**	**0.503**	**0.536**	**0.231**	**0.521**	**0.526**

Table 5: Ablation of the spatially aligned visual comparison. *Spat. Align* denotes coordinate-preserving rendering, *SBS* side-by-side presentation, and *Index* explicit atom indexing.

Interface	Design			OCIE			JIE		
	Spat. Align	SBS	Index	OA↑	AA↑	Cov↑	OA↑	AA↑	Cov↑
Base recognizer	–	–	–	73.53	73.53	100.0	61.83	61.83	100.0
Source-prediction interfaces									
Image & SMILES	✗	✗	✗	74.26	82.70	68.8	62.91	77.80	59.6
Image & graph	✗	✗	✓	76.48	86.90	73.4	65.74	81.90	65.1
Rendered interfaces									
Overlay	✓	✗	✓	83.64	93.60	84.2	78.42	91.40	78.3
Side-by-side, default layout	✗	✓	✓	79.71	89.70	78.1	72.88	86.20	71.5
Side-by-side, coord.-preserving	✓	✓	✗	82.85	96.84	84.7	77.81	96.35	80.1
ViCoR (ours)	✓	✓	✓	**88.26**	**97.54**	**89.0**	**84.32**	**97.17**	**85.4**

Observation 2: ViCoR achieves high reliability at broad coverage. ViCoR achieves 97.54%/97.17% AA at 89.0%/85.4% coverage on OCIE/JIE, yielding 86.9%/83.0% of all predictions that are both accepted and correct, already exceeding the base recognizer's OA. Compared with confidence filtering, ViCoR achieves higher AA at matched coverage and higher coverage at matched AA. Agreement filtering also reaches high AA but with substantially lower coverage, since disagreements can only be rejected rather than revised. Overall, ViCoR combines reliable acceptance with broad coverage, which further benefits downstream chemical tasks in Sec. 4.2.

4.2 Real-world Downstream Utility of Reliable Structures

Chemical Reaction Extraction from Literature. Automated reaction extraction is a complex, multi-stage pipeline (e.g., template and product recognition, and R-group substitution), highly sensitive to single errors. We integrate ViCoR into OpenChemIE (Fan et al., 2024) and evaluate reaction-level extraction performance in Tab. 3. Confidence filtering improves reliability by rejecting uncertain structures, but substantially reduces recall and overall F1. VLM SMILES revision provides a moderate improvement over raw OCSR outputs. In contrast, ViCoR substantially improves both precision and F1, showing that reliable structure recognition directly benefits downstream extraction.

Reaction Prediction with Literature-Extracted Data. To further study whether SSR improved downstream model learning, we collect 2,100 photocatalysis and 430 Suzuki-coupling figures from chemical journals and use OpenChemIE to extract reaction data under different post-recognition strategies. The resulting datasets are then used to fine-tune Qwen3-8B (Yang et al., 2025) for forward reaction prediction. As shown in Tab. 4, confidence filtering provides only modest gains because cleaner data comes at the cost of reduced training coverage, while VLM SMILES revision recovers part of the corrupted data. ViCoR achieves the best performance across both reaction domains, showing that improved structure reliability and coverage translate into better downstream learning.

4.3 Ablation and Design Analysis

Spatially Aligned Visual Comparison. Table 5 studies how the prediction interface affects verification and revision. Directly comparing the source image with symbolic representations such as SMILES or graph performs poorly because the model must infer the correspondence between visual regions and symbolic entities. Rendering the prediction reduces this representation gap, but canonical-layout rendering still changes the spatial arrangement of atoms and bonds. In contrast, coordinate-preserving rendering substantially improves both OA and AA by explicitly aligning cor-

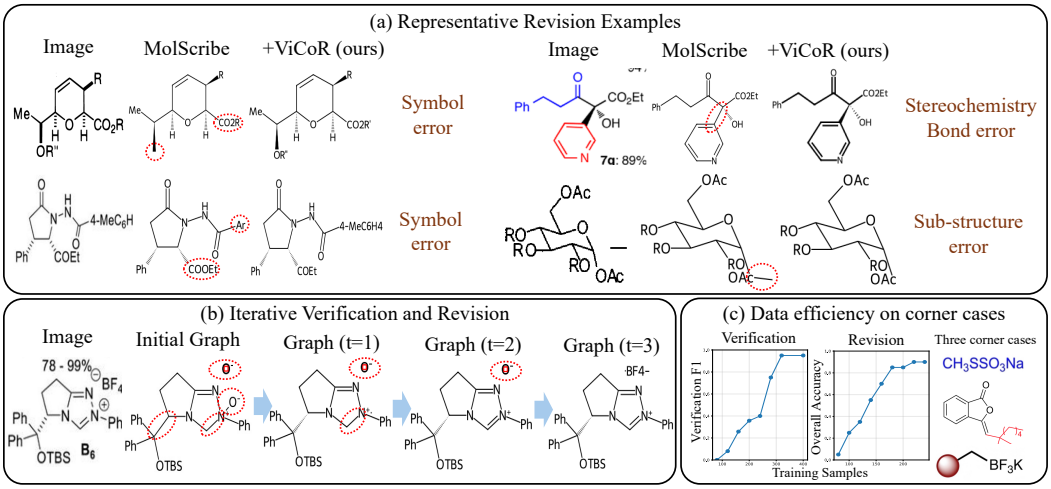


Figure 3: Qualitative results of ViCoR. (a) Representative Revision Examples. (b) ViCoR improves recognition through iterative verification and revision. (c) Data efficiency on corner cases.

Table 6: Ablation on iterative revision.

Models	Thpt.(img/s)	OCIE(OA)	JIE(OA)
Baseline	1.61	73.53	61.83
$T = 1$	1.33	80.82	72.45
$T = 2$	1.21	<u>86.45</u>	<u>81.36</u>
$T = 3$	1.18	**88.26**	**84.32**

Table 7: Ablation on training strategy.

Training	OCIE			JIE		
	OA↑	AA↑	Cov↑	OA↑	AA↑	Cov↑
Separate	87.81	97.50	88.55	84.04	97.14	85.11
Mixed	87.45	96.85	88.67	83.85	96.95	84.70
Progressive	**88.26**	**97.54**	**89.04**	**84.32**	**97.17**	**85.44**

responding regions. Side-by-side presentation provides a further gain over image overlay, while atom indices further improve performance by making graph entities directly addressable for revision. Overall, these results validate spatial alignment and index anchoring as the key interfaces.

Iterative Revision and Computational Cost. Tab. 6 evaluates different revision budgets. Increasing the number of iterations consistently improves OA, showing that single revision may leave residual structural errors. Re-verifying each intermediate prediction allows ViCoR to identify and revise these remaining errors, with only a modest reduction in throughput. Fig. 3(b) shows an example.

Effect of Progressive Training. Tab. 7 shows that progressive training consistently outperforms separate and mixed training. Verification-oriented pretraining provides a stronger initialization for subsequent graph revision by learning robust visual correspondence.

4.4 Further Analysis

Data Efficiency on Corner Cases. We evaluate how efficiently ViCoR adapts to three challenging corner cases: single-string labels, Markush repetition, and special R-group symbols (Fig. 3(c)). With only 300 targeted synthetic samples, ViCoR already achieves strong verification and revision performance. This shows that rare failure modes can be learned with little task-specific data.

Transfer to Another Image-to-Structure Task. We transfer ViCoR to hand-drawn BPMN process-graph recognition (Schäfer et al., 2022). With Faster R-CNN (Ren et al., 2016) and YOLO11m (Khanam et al., 2024), ViCoR improves OA from 41.6% to 62.1% and from 50.4% to 68.8%, respectively, while retaining over 95% AA, showing applicability beyond OCSR. Details in Appendix.

5 Conclusion

We introduce **ViCoR**, a post-recognition framework for reliable OCSR that follows a *repair-before-rejection* principle. By combining spatially aligned verification, index-anchored executable revision, and verification-to-revision progressive training, ViCoR improves both recognition accuracy and the reliability–coverage trade-off, while providing higher-quality molecular data for downstream reaction extraction and prediction. ViCoR still has limitations on highly complex molecular depictions, especially 3D structures that are difficult to interpret from 2D images, and its iterative inference introduces additional computational cost. Future work will improve recognition and revision for complex 3D molecules and explore more efficient inference for large-scale scientific data curation.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing and improving the clarity and presentation of the manuscript. We have not used generative AI tools to formulate research hypotheses, design the methodology or experiments, or interpret experimental results, and the remaining required disclosure tasks are not applicable to this work. We have reviewed all AI-assisted edits and revised them where necessary to ensure accuracy and consistency with the authors' intended meaning. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

We anticipate that ViCoR can facilitate the reliable extraction and digitization of molecular structures from scientific literature, thereby supporting chemical data curation and downstream cheminformatics research. At the same time, incorrectly recognized or revised structures may propagate errors into chemical databases or downstream models if used without appropriate validation. ViCoR is intended to improve the reliability of automated structure recognition rather than replace expert verification in safety-critical applications. Our study does not involve human subjects or personal data. We encourage responsible use of the released models, code, and benchmarks in accordance with applicable licenses and research ethics guidelines.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our research, we provide the implementation and experimental details of ViCoR in Section 4.1 and the Appendix, including model configurations, training settings, dataset construction, and evaluation protocols. To facilitate future research and technical transparency, we will publicly release our source code, model configurations, trained model weights, and curated benchmarks in accordance with relevant licenses. These resources will enable researchers to reproduce our results and build upon ViCoR.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Yufan Chen, Ching Ting Leung, Yong Huang, Jianwei Sun, Hao Chen, and Hanyu Gao. Molnextr: a generalized deep learning model for molecular image recognition. *Journal of Cheminformatics*, 16(1):141, 2024.

Yufan Chen, Ching Ting Leung, Jianwei Sun, Yong Huang, Linyan Li, Hao Chen, and Hanyu Gao. Towards large-scale chemical reaction image parsing via a multimodal large language model. *arXiv preprint arXiv:2503.08156*, 2025a.

Yufan Chen, Ching Ting Leung, Bowen Yu, Jianwei Sun, Yong Huang, Linyan Li, Hao Chen, and Hanyu Gao. A multi-agent system enables versatile information extraction from the chemical literature. *arXiv preprint arXiv:2507.20230*, 2025b.

Ran El-Yaniv et al. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(5), 2010.

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

Xi Fang, Jiankun Wang, Xiaochen Cai, Shangqian Chen, Shuwen Yang, Haoyi Tao, Nan Wang, Lin Yao, Linfeng Zhang, and Guolin Ke. Molparser: End-to-end visual recognition of molecule structures in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 24528–24538, 2025.

Igor V Filippov, Marc C Nicklaus, and Marc C Nicklaus. Optical structure recognition software to recover chemical information: Osra, an open source solution, 2009.

Akio Fujiyoshi, Koji Nakagawa, and Masakazu Suzuki. Robust method of segmentation and recognition of chemical structure images in cheminfty. In *Pre-proceedings of the 9th IAPR international workshop on graphics recognition, GREC*, volume 1, 2011.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. *Advances in neural information processing systems*, 30, 2017.

Yonatan Geifman and Ran El-Yaniv. Selectivenet: A deep neural network with an integrated reject option. In *International conference on machine learning*, pp. 2151–2159. PMLR, 2019.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yujiu Yang, Nan Duan, Weizhu Chen, et al. Critic: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*, volume 2024, pp. 57734–57811, 2024.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Rahima Khanam, Muhammad Hussain, and Muhammad Hussain. Yolov11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725*, 2024.

Sunghwan Kim, Paul A Thiessen, Evan E Bolton, Jie Chen, Gang Fu, Asta Gindulyte, Lianyi Han, Jane He, Siqian He, Benjamin A Shoemaker, et al. Pubchem substance and compound databases. *Nucleic acids research*, 44(D1):D1202–D1213, 2016.

Greg Landrum. Rdkit: Open-source cheminformatics software. `https://www.rdkit.org`, 2016. Accessed 2016.

Shi Xuan Leong, Sergio Pablo-García, Brandon Wong, and Alán Aspuru-Guzik. Mermaid: Universal multimodal mining of chemical reactions from pdfs using vision-language models. 2025.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594, 2023.

Lucas Morin, Martin Danelljan, Maria Isabel Agea, Ahmed Nassar, Valery Weber, Ingmar Meijer, Peter Staar, and Fisher Yu. Molgrapher: graph-based visual recognition of chemical structures. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Lucas Morin, Valéry Weber, Gerhard Ingmar Meijer, Fisher Yu, and Peter WJ Staar. Patcid: an open-access dataset of chemical structures in patent documents. *Nature Communications*, 15(1): 6532, 2024.

David F Nippa, Kenneth Atz, Remo Hohler, Alex T Müller, Andreas Marx, Christian Bartelmus, Georg Wuitschik, Irene Marzuoli, Vera Jost, Jens Wolfard, et al. Enabling late-stage drug diversification by high-throughput experimentation with geometric deep learning. *Nature Chemistry*, 16 (2):239–248, 2024.

OpenAI. Openai api. `https://openai.com/api/`, 2026. Accessed: 2026-05-03.

Tyler Peryea, Daniel Katzel, Tongan Zhao, Noel Southall, and Dac-Trung Nguyen. Molvec: Open source library for chemical structure recognition. In *Abstracts of papers of the American Chemical Society*. Amer Chemical Soc 1155 16TH ST, NW, WASHINGTON, DC 20036 USA, 2019.

Florina Piroi, Mihai Lupu, Allan Hanbury, Alan P Sexton, Walid Magdy, and Igor V Filippov. Clef-ip 2010: Retrieval experiments in the intellectual property domain. In *CLEF (notebook papers/labs/workshops)*, 2010.

Yujie Qian, Jiang Guo, Zhengkai Tu, Zhening Li, Connor W Coley, and Regina Barzilay. Molscribe: robust molecular structure recognition with image-to-graph generation. *Journal of chemical information and modeling*, 63(7):1925–1934, 2023.

Kohulan Rajan, Achim Zielesny, and Christoph Steinbeck. Decimer: towards deep learning for chemical image recognition. *Journal of Cheminformatics*, 12(1):65, 2020.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

Noureddin M Sadawi, Alan P Sexton, and Volker Sorge. Chemical structure recognition: a rule-based approach. In *Document recognition and retrieval XIX*, pp. 101–109. SPIE, 2012.

Bernhard Schäfer, Han Van Der Aa, Henrik Leopold, and Heiner Stuckenschmidt. Sketch2process: End-to-end bpmn sketch recognition based on neural networks. *IEEE Transactions on Software Engineering*, 49(4):2621–2641, 2022.

Jiahe Song, Chuang Wang, Bowen Jiang, Yinfan Wang, Hao Zheng, Xingjian Wei, Chengjin Liu, Junyuan Gao, Yubin Wang, Lijun Wu, et al. Rxncaption: Reformulating reaction diagram parsing as visual prompt guided captioning. *arXiv preprint arXiv:2511.02384*, 2025.

Tejas Srinivasan, Jack Hessel, Tanmay Gupta, Bill Yuchen Lin, Yejin Choi, Jesse Thomason, and Khyathi Chandu. Selective "selective prediction": Reducing unnecessary abstention in vision-language reasoning. In *ACL 2024 Finding*, pp. 12935–12948, 2024.

Joshua Staker, Kyle Marshall, Robert Abel, and Carolyn M McQuaw. Molecular structure extraction from documents using deep learning. *Journal of chemical information and modeling*, 2019.

Hao-Dong Tan, Ben Gao, Huaihai Huang, Tingjun Xu, Yao Li, Jiajia Dong, and Xiao-Song Xue. Data-driven, mechanistically guided prediction of yield and chemoselectivity in sufex reactions. *Journal of the American Chemical Society*, 148(23):24138–24150, 2026.

David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 1988.

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.

Wenrui Zhang, Xinggang Wang, Bin Feng, and Wenyu Liu. Molsight: Optical chemical structure recognition with smiles pretraining, multi-granularity learning and reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

A Training Data Synthesis

We present the training data synthesis procedure for ViCoR. The pipeline includes a set of common data augmentations applied to both verification and revision tasks, along with task-specific synthesis methods designed to support their respective learning goals. Finally, we introduce the data synthesis for corner cases.

A.1 Data Augmentation

To improve the diversity and robustness of the synthesized molecular images, we apply a combination of rendering-level, image-level, and molecular-level augmentations during data synthesis.

Rendering-level Augmention. Molecular depictions in real-world chemical literature exhibit substantial stylistic variation. Following (Chen et al., 2024), we introduce randomized rendering variations using Indigo to simulate this diversity. Specifically, we vary bond thicknesses, line widths, font families and sizes, inter-line spacing for double and triple bonds, label rendering modes, and the visibility of implicit hydrogens.

Image-level Augmentation. After rendering the image, we apply additional image-level augmentations to further diversify the appearance of the synthesized images. The applied transformations include: (1) random Gaussian blur, (2) random grayscale conversion, (3) random white padding, and (4) random insertion of background circles for individual atoms.

Molecular-level Augmentation. Although the molecular SMILES sourced from PubChem (Kim et al., 2016) provide substantial structural diversity, they lack sufficient coverage of R-group notation and commonly used abbreviations (e.g., 4-MeC_6H_4) frequently observed in chemical literature. To address this limitation, we construct curated lists of R-group symbols and 66 commonly used complex abbreviations (e.g., O-*t*-Bu, 4-CNC_6H_4). After Indigo parses a SMILES string into a molecular graph, we randomly replace selected terminal nodes with R-group symbols or abbreviations. The modified graph is then rendered into an image by Indigo. This augmentation introduces realistic symbolic variations into the training distribution, improving the model's robustness to R-group and abbreviation usage in real-world molecular images.

A.2 Verification Training Data Synthesis

To construct training data for the verification (800k samples), we describe the image rendering and concatenation procedure and the strategy used to generate negative samples. Positive samples are obtained straightforwardly by using the same molecular graph for both image renderings.

Image Rendering and Concatenation. As illustrated in Fig. 2, given a predicted molecular graph, we render a corresponding molecular image using its atomic coordinates and bond connectivity, ensuring strict spatial alignment with the original input image. The original image and the rendered image are then concatenated into a single composite image, separated by a clear black divider to visually distinguish the two views. During verification, ViCoR takes the concatenated image together with a verification prompt as input.

Negative Sample Construction. Rendering an image from the ground-truth molecular graph yields only positive verification samples. To obtain informative negative examples, we employ two complementary strategies.

(1) Failure Cases from an Expert OCSR Model. To capture realistic error patterns produced by OCSR systems, we incorporate incorrect predictions generated by MolScribe (Qian et al., 2023) as negative samples. Specifically, we synthesize 1M molecular images from PubChem SMILES using Indigo and apply MolScribe to predict their molecular graphs. The predicted graphs are then matched against the ground-truth graphs provided by Indigo using a graph isomorphism algorithm. Samples with mismatched graphs (approximately 140k) are labeled as negative examples, rendered into images, and concatenated with the corresponding original images.

(2) Random Graph Perturbation. Starting from the ground-truth (GT) molecular graph corresponding to the original image, we introduce controlled random perturbations to synthesize challenging negative samples. These perturbations are systematically applied at both the atom and bond levels—specifically through the addition, substitution, and deletion of atoms and bonds—to comprehensively cover the six standard types of graph edit operations. To faithfully simulate common OCSR failures, we distribute these perturbations across three typical error categories in our training set: atom symbol errors (33%), bond type errors (33%), and sub-structure errors (34%). Furthermore, to ensure varying levels of difficulty, each perturbed molecule is injected with a random number of 1 to 6 errors.

A.3 Revision Training Data Synthesis

To construct training data for the revision task (500k samples), we describe the image rendering and concatenation procedure and the generation of revision supervision.

Image Rendering and Concatenation. The rendering process largely follows that used for the verification task. To enable location-precise graph revision, we additionally annotate each atom in the rendered image with a unique index. This is implemented by enabling the *addAtomIndices* option when rendering molecular graphs with RDKit.

Revision Format Identification. Unlike the verification task which relies on holistic negative samples, the revision task requires fine-grained, localized structural supervision. To construct training samples for this generative task, we apply a controlled sequence of graph edit operations to corrupt the ground-truth (GT) molecular graph. The supervision target for the VLM is then formulated as the sequence of corrective edits Δ required to restore the corrupted graph back to its original GT.

To seamlessly integrate the mathematical edit space Ω (defined in Eq. 13& 14) with the autoregressive nature of VLMs, we serialize the sequence of corrective operators into a structured, JSON-like text format. Each edit instruction $\delta_k \in \Delta$ is represented as a dictionary mapping the operation type to its specific parameters. Explicitly, the six types of graph edits are formatted as follows:

- **Node Addition** (Add_V): {'Add atom': [i, ℓ_i, [x_i, y_i]]}, where i is the new node index, ℓ_i is the atomic symbol, and $[x_i, y_i]$ are the normalized 2D coordinates .
- **Node Deletion** (Del_V): {'Del atom': [i, ℓ_i]}.
- **Node Revision** (Rev_V): {'Rev atom': [i, ℓ'_i]}, denoting the correction of the atom at index i to the target symbol ℓ'_i.
- **Edge Addition** (Add_E): {'Add bond': [[i, j], $e_{i,j}$]}, specifying the source(i) and target(j) node indices, and the discrete bond type (e.g., *solid wedge*, *single*).
- **Edge Deletion** (Del_E): {'Del bond': [[i, j], $e_{i,j}$]}.
- **Edge Revision** (Rev_E): {'Rev bond': [[i, j], $e'_{i,j}$]}.

During training, an entire revision target sequence is concatenated into a single list. For instance, a complex revision sample involving multiple topological corrections is supervised using the format:

```
[
  {'Add atom': [3, 'C', [<0.312>, <0.578>]]},
  {'Add bond': [[3, 4], 'solid wedge']},
  {'Del atom': [9, 'N']},
  {'Del bond': [[1, 2], 'single']},
  {'Rev atom': [6, 'H']},
  {'Rev bond': [[4, 5], 'double']}
]
```

Consistent with the VLM's tokenizer design mentioned in the main text, the continuous spatial coordinates (e.g., `0.312`, `0.572`) in the `Add atom` instruction are dynamically mapped to their corresponding quantized location tokens (e.g., `<0.312>`) during the text generation process. This structured prompt formulation effectively bridges the gap between abstract graph topology and language modeling. Fig. 5 illustrates an example of the input and output of the VLM.

Training Sample Construction. To construct a robust training set that faithfully simulates realistic OCSR failures, we systematically control both the distribution and the topological coherence of the injected perturbations. While atom symbol errors (33%) and bond type errors (33%) typically involve isolated revisions (Rev_V or Rev_E), the sub-structure errors (34%) are designed as edits.

Unlike naive random modifications, these composite edits consist of coordinated operations to simulate complex structural hallucinations while maintaining valid graph topologies. For instance, the addition of a hallucinated node is strictly coupled with the addition of corresponding connecting edges ($\text{Add}_V \rightarrow \text{Add}_E$) to prevent disconnected components. Conversely, the deletion of a node inherently triggers a cascading deletion of all its incident edges ($\text{Del}_V \rightarrow \text{Del}_E$) to avoid invalid dangling bonds. We further extend this to motif-level perturbations, such as replacing or hallucinating entire functional groups, which require a synchronized sequence of multiple additions/deletions.

To ensure varying levels of structural degradation, each ground-truth molecule is injected with 1 to 6 such logical errors. Due to the coordinated nature of composite edits, a single logical sub-structure error may expand into multiple discrete edit instructions in the target sequence Δ. This targeted corruption strategy forces the VLM to go beyond trivial one-to-one symbol corrections, teaching it to perform multi-step, chemically plausible topological restorations.

A.4 Data Synthesis for Corner Cases

Since Indigo does not support image synthesis for certain corner cases, we adopt a post-processing–based synthesis strategy combined with human filtering to construct such samples. As in Fig. 4, we insert repeated brackets at the target atom locations using their coordinates in the synthesized images. The resulting samples are then manually filtered to remove visually implausible instances, yielding the final corner-case training data. For corner cases involving special R-group symbols(Fig. 3), we assign a dedicated label `r` to distinguish them from standard R-group symbols.

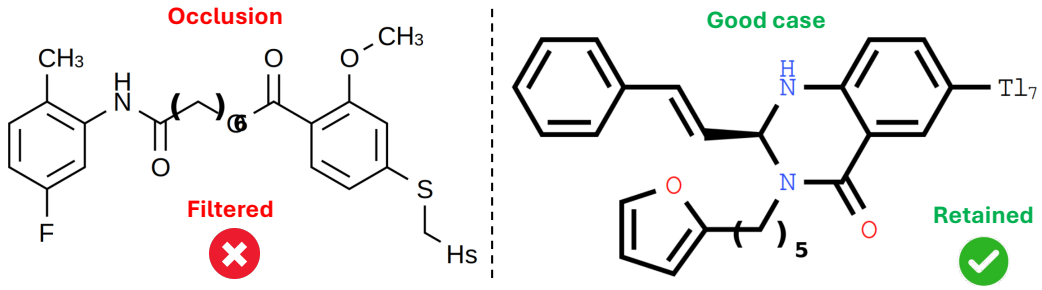


Figure 4: Examples of human filtered and retained corner case training samples.

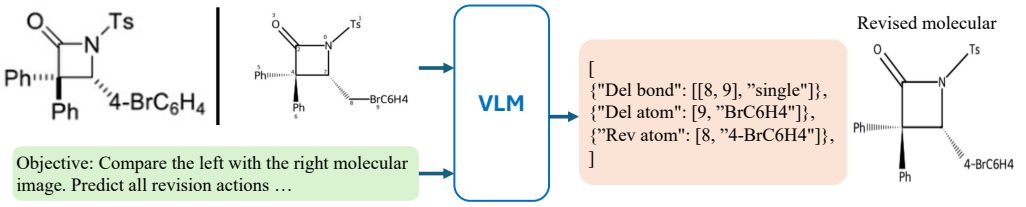


Figure 5: An example of the input and output of the ViCoR for revision.

B PROMPT FOR VICOR.

We employ task-specific prompts for the ViCoR to support graph verification and revision, summarized in Tab. 8. It is worth noting that our VLM is task-specifically trained rather than direct prompt-engineering on an existing pre-trained VLM.

Table 8: Prompts used for verification and revision in ViCoR

Task	**Prompt**
Verification	Do the left and right molecular images represent the same molecule? Ignore the drawing style and other noises. Answer only `'Yes.'` or `'No.'`.
Revision	Objective: Compare the left with the right molecular image. Predict all revision actions to reconstruct the exact graph shown in the reference image. Instructions: 1. Check for missing, redundant, or incorrect atoms and bonds. 2. Match indices to specific visual regions strictly using spatial proximity. 3. Formulate the corrections as a sequence of discrete graph edit operations. Output Format: Return ONLY a JSON list of dictionaries containing the required edit instructions. You must use the following templates: • {"Add atom": [<index>, "<symbol>", ["<x>", "<y>"]]} • {"Del atom": [<index>, "<symbol>"]} • {"Rev atom": [<index>, "<target_symbol>"]} • {"Add bond": [[<src>, <tgt>], "<bond_type>"]} • {"Del bond": [[<src>, <tgt>], "<bond_type>"]} • {"Rev bond": [[<src>, <tgt>], "<target_bond_type>"]} Example: [{"Add atom": [3, "C", ["<0.314>", "<0.576>"]]}, {"Add bond": [[3, 4], "solid wedge"]}]

C Details on Index-anchored Executable Revision

As introduced in the main text, the Vision-Language Model (VLM) predicts a sequence of graph edit operations $\Delta = (\delta_1, \delta_2, \ldots, \delta_K)$ in a single forward pass. Because these instructions are generated autoregressively based on the initial input state, applying them simultaneously to the parsed graph can lead to logical collisions and topological invalidity. To manage overlapping edits and ensure structural robustness, the graph update operator $\oplus$ processes the sequence strictly in order, computing intermediate graph states $G_k = G_{k-1} \oplus \delta_k$ (where G_0 is the initial parsed graph).

During this sequential execution, the $\oplus$ operator employs a dynamic validity enforcement mechanism to resolve conflicts through two core strategies:

Absorbing Implicit Topological Changes. Certain graph edits inherently trigger cascading effects that alter the topology beyond the explicit instruction. The $\oplus$ operator automatically absorbs these implicit changes to maintain chemical validity. The most prominent case is **cascading deletion**: when a node deletion instruction $\text{Del}_V(i)$ is executed, the operator not only removes atom i, but also automatically identifies and removes all incident edges $E(i, j)$ connected to atom i. This prevents the emergence of invalid "dangling bonds" (edges without endpoints) in the intermediate graph G_k.

Bypassing Obsolete Instructions. Because the target graph continuously evolves during the sequential update, some subsequent instructions δ_k may become invalid or obsolete relative to the current state G_{k-1}. Instead of raising execution errors, the $\oplus$ operator evaluates the pre-conditions of each δ_k and safely bypasses (ignores) it if conflicts are detected. Typical scenarios include:

- **Target Missing (Invalid Edge Operations):** If an instruction attempts to add, delete, or revise an edge between nodes i and j (e.g., $\text{Add}_E(i \to j)$), but either node i or j has already been removed by a preceding Del_V operation, the operator bypasses this edge instruction.
- **Redundant Operations:** If an instruction requests to delete an atom/bond that has already been implicitly deleted (via cascading deletion), or requests to revise a target to its current state, the operation is skipped.
- **Index Collisions:** If a node addition $\text{Add}_V(i)$ attempts to use an index i that already exists in G_{k-1}, the operator dynamically assigns the next available unique index and updates subsequent edges pointing to this new node.

By coupling sequential execution with these conflict resolution protocols, the $\oplus$ operator guarantees that the final reconstructed graph G_K is strictly valid in terms of graph topology, fully decoding the VLM's predictions into chemically sound molecular structures.

D Test Datasets

To compare with prior state-of-the-art methods, we evaluate molecular structure recognition on six public realistic benchmarks: CLEF (Piroi et al., 2010), UOB (Sadawi et al., 2012), JPO (Fujiyoshi et al., 2011), USPTO (Filippov et al., 2009), Staker (Staker et al., 2019), and ACS (Qian et al., 2023), as well as two synthetic datasets (Indigo and ChemDraw (Qian et al., 2023)). However, these classical OCSR benchmarks contain limited coverage of R-group symbols, complex abbreviations, and dissociative noise, which are frequently encountered in real-world chemical literature.

To better assess ViCoR under practical conditions, we curate two in-the-wild datasets, OCIE and JIE. OCIE consists of 1,141 molecular images cropped from the chemical information extraction test set of OpenChemIE (Fan et al., 2024). JIE contains 1,779 molecular images extracted from 243 figures collected from chemical literature sources such as *JACS* and *Nature*. All dataset statistics are summarized in Tab. 10.

Complementarity to Existing Real-World Benchmarks. OCIE and JIE are designed to complement existing real-world OCSR benchmarks. In particular, IP5-M focuses on Markush structures and R-groups in patent images, while USPTO-10K-Abb targets abbreviated superatoms in the patent domain. OCIE and JIE instead provide an updated journal-domain evaluation with the following complementary properties:

Table 9: Comparison of OCIE and JIE with existing real-world OCSR benchmarks. OCIE and JIE complement patent-domain benchmarks with recent journal figures containing smaller image crops, frequent color usage, and naturally co-occurring recognition challenges.

Dataset	Source domain	Primary focus	Images	Median size	Color (%)
IP5-M	Patents	Markush / R-groups	878	1024×1024	0.2
USPTO-10K-Abb	Patents	Abbreviated superatoms	10,000	648×394	0.0
OCIE	Journal figures (2022–2023)	Mixed OCSR factors	1,141	220×155	72.0
JIE	Journal figures (2022–2024)	Mixed OCSR factors	1,779	233×158	19.6

- **Journal-domain visual distribution.** Journal substrate-scope and reaction figures exhibit heterogeneous layouts, line widths, backgrounds, drawing conventions, and color usage that differ from predominantly monochrome patent drawings.
- **Low-resolution crops with natural interference.** OCIE and JIE consist of natural image crops with median resolutions of approximately 220×155 and 233×158 pixels, respectively. Since individual molecules are cropped from complete journal figures, the images may retain neighboring structure fragments, labels, or other residual figure content, introducing realistic visual distractors.
- **Naturally co-occurring recognition challenges.** Rather than isolating a single difficulty, OCIE and JIE preserve the combinations naturally found in journal figures. R-groups, abbreviations, stereochemistry, formal charges, and disconnected structures may co-occur with color, low resolution, heterogeneous drawing styles, and crop interference.

Together, these properties provide a complementary test setting for evaluating OCSR systems under realistic and visually heterogeneous journal conditions.

Dataset Construction. To construct the OCIE and JIE datasets, we compile a diverse corpus of scientific articles in PDF format from prominent chemistry and multidisciplinary journals (e.g., *JACS*, *Nature*). Figures are systematically extracted from these PDFs using VisualHeist, followed by the manual selection of reaction scope diagrams. We subsequently apply MolDetect (Fan et al., 2024) to localize individual molecular bounding boxes. These boxes are then cropped to generate the single-molecule images that comprise our final datasets, ensuring a comprehensive representation of realistic molecular depictions. For ground-truth annotation, we employ a semi-automatic expert curation pipeline. Initially, MolScribe is utilized to generate a candidate molecular graph for each image. These preliminary graphs are then rigorously reviewed and manually corrected by three independent annotators with advanced chemistry backgrounds. To guarantee high-quality and consistent annotations, each sample undergoes independent verification, and any inter-annotator disagreements are resolved through consensus discussions.

Table 10: Summary of the test datasets

Dataset	Type	Total Images	Abbreviations
Indigo (Qian et al., 2023)	Synthetic	5,719	×
ChemDraw (Qian et al., 2023)	Synthetic	5,719	×
CLEF(Piroi et al., 2010)	Real	992	✓
UOB (Sadawi et al., 2012)	Real	5,740	✓
JPO (Fujiyoshi et al., 2011)	Real	450	×
USPTO (Filippov et al., 2009)	Real	5,719	✓
Staker (Staker et al., 2019)	Real	50,000	✓
ACS (Qian et al., 2023)	Real	331	✓
OCIE	Real	1141	✓
JIE	Real	1779	✓

E More Experiments and Ablation Study

We provide additional analyses of the computational cost and design choices of ViCoR that complement the ablations in the main text. We first quantify the accuracy–efficiency trade-off introduced by iterative verification and revision, and then study robustness to rendering styles and the effect of the VLM backbone and model scale.

Computational Cost and Iteration Budget. We first compare the end-to-end computational cost of ViCoR with representative OCSR systems under the same local inference environment. All measurements are obtained on a single NVIDIA RTX 3090 and include the complete inference pipeline. For ViCoR, this includes the base recognizer, prediction rendering, VLM verification and revision, and graph-update operations. All reported accuracy differences are absolute percentage-point changes.

Table 11: End-to-end accuracy and computational cost on a single NVIDIA RTX 3090. Relative latency is normalized by MolScribe.

Method	Throughput ↑ (img/s)	Latency ↓ (s/img)	Rel. latency vs. MolScribe	OCIE		JIE	
				OA ↑	Δ vs. MolScribe	OA ↑	Δ vs. MolScribe
OSRA	6.64	0.151	0.24×	36.17	−37.36	19.88	−41.95
DECIMER	0.28	3.571	5.75×	35.07	−38.46	16.42	−45.41
MolGrapher	0.42	2.381	3.83×	41.20	−32.33	25.40	−36.43
MolNexTR	1.42	0.704	1.13×	73.24	−0.29	61.62	−0.21
MolScribe	1.61	0.621	1.00×	73.53	–	61.83	–
MolScribe + ViCoR	1.18	0.847	1.36×	**88.26**	**+14.73**	**84.32**	**+22.49**

As shown in Tab. 11, ViCoR introduces a moderate latency increase over MolScribe while providing substantially higher strict recognition accuracy. The default $T = 3$ configuration increases latency by $1.36\times$ relative to MolScribe, but improves OA by 14.73 and 22.49 points on OCIE and JIE, respectively. ViCoR is also considerably faster than DECIMER and MolGrapher while achieving much higher accuracy. OSRA provides the highest throughput, but at a substantial loss in recognition quality. These results indicate that ViCoR trades a moderate amount of additional computation for a large improvement in reliable whole-structure recognition.

We further isolate the additional cost introduced by iterative verification and revision by varying the maximum number of iterations T.

Table 12: Accuracy–cost trade-off under different ViCoR iteration budgets. Relative latency is normalized by the MolScribe base recognizer.

Method	Throughput ↑ (img/s)	Latency ↓ (s/img)	Rel. latency vs. base	OCIE		JIE	
				OA ↑	Δ vs. base	OA ↑	Δ vs. base
MolScribe base	1.61	0.621	1.00×	73.53	–	61.83	–
ViCoR ($T = 1$)	1.33	0.752	1.21×	80.82	+7.29	72.45	+10.62
ViCoR ($T = 2$)	1.21	0.826	1.33×	86.45	+12.92	81.36	+19.53
ViCoR ($T = 3$)	1.18	0.847	1.36×	**88.26**	**+14.73**	**84.32**	**+22.49**

Tab. 12 shows a clear accuracy–cost trade-off. Increasing the iteration budget consistently improves OA, while the incremental latency becomes progressively smaller. Notably, $T = 1$ already recovers substantial accuracy at only $1.21\times$ the base latency, whereas the default $T = 3$ setting provides the strongest recognition performance at $1.36\times$ latency. The iteration budget can therefore be selected according to the deployment requirement: fewer iterations favor high-throughput extraction, while additional iterations are beneficial when strict structural correctness is prioritized.

Robustness to Rendering Style. ViCoR compares a real molecular image against a rendered prediction. A natural concern is therefore whether the verifier learns renderer-specific appearance rather than structural correspondence. We test this by comparing training pairs generated with the same renderer (RDKit → RDKit) against cross-renderer training (Indigo → RDKit).

Table 13: Effect of rendering diversity during verification training (%).

Training comparison	OCIE OA ↑	JIE OA ↑
RDKit → RDKit	85.42	80.76
Indigo → RDKit	**88.26**	**84.32**

Cross-renderer training improves OA by 2.84 and 3.56 points on OCIE and JIE, respectively. This suggests that exposing the verifier to depiction shifts during training reduces reliance on renderer-specific visual cues and encourages comparison based on the underlying molecular structure. We therefore use cross-renderer training in the default ViCoR configuration.

Effect of VLM Backbone and Scale. We next examine whether the performance of ViCoR primarily arises from using a larger general-purpose VLM or from task-specific verification–revision training. We compare two general-purpose GPT models with trained Qwen2.5-VL models at 3B and 7B parameter scales.

Table 14: Effect of the VLM backbone and model scale on strict recognition accuracy (%).

VLM	OCIE OA ↑	JIE OA ↑
ViCoR(GPT-4o) + MolScribe	76.60	73.97
ViCoR(GPT-5.6-sol) + MolScribe	77.83	75.77
ViCoR(Trained Qwen2.5-VL-3B) + MolScribe	88.26	84.32
ViCoR(Trained Qwen2.5-VL-7B) + MolScribe	**88.48**	**84.53**

As shown in Tab. 14, increasing the trained Qwen2.5-VL backbone from 3B to 7B parameters yields only marginal gains of 0.22 and 0.21 points on OCIE and JIE. In contrast, both trained models substantially outperform the general-purpose GPT baselines. This indicates that, within the tested scale range, ViCoR benefits more from task-specific verification–revision training than from simply increasing VLM parameter count. We therefore use Qwen2.5-VL-3B as the default verifier/reviser to balance accuracy and computational efficiency.

F Visualization.

Fig. 6 illustrates more examples of molecular structure recognition results by MolScribe and MolScribe + ViCoR.

G Downstream 1: Chemical Reaction Extraction

An important application of ViCoR is chemical reaction extraction from scientific literature. Chemical reaction extraction aims to extract structured reactions from complex reaction scope figures, which typically requires a multi-stage pipeline including **reaction template recognition**, **recognition of multiple representative product molecules**, and **R-group substitution between products and reaction templates**. This process is highly error-sensitive, as failures at any stage can propagate and result in invalid or incorrect reactions, degrading the performance of the downstream tasks trained on the extracted reactions. An example of chemical reaction extraction is shown in Fig. 7.

H Downstream 2: Reaction Prediction with Literature-Extracted Data

We further evaluate whether improvements in selective structure recognition translate into better downstream model learning by constructing reaction-prediction datasets directly from figures in the chemical literature. All molecular and reaction images in this experiment are cropped from original journal articles.

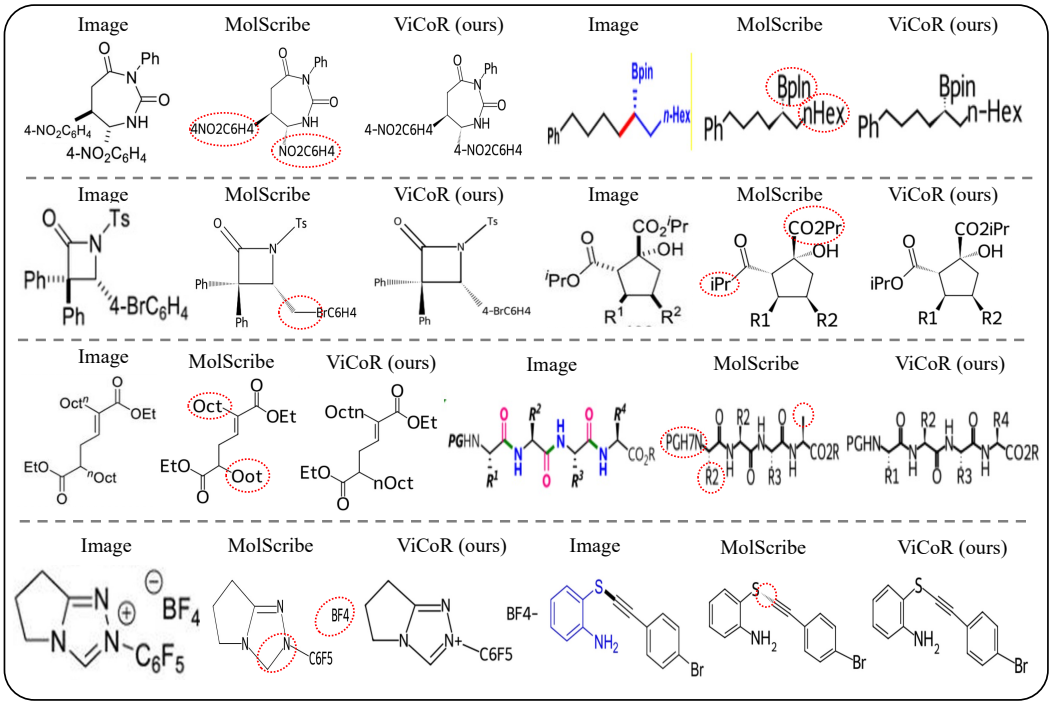


Figure 6: Examples of comparison between baseline MolScribe and MolScribe + ViCoR.

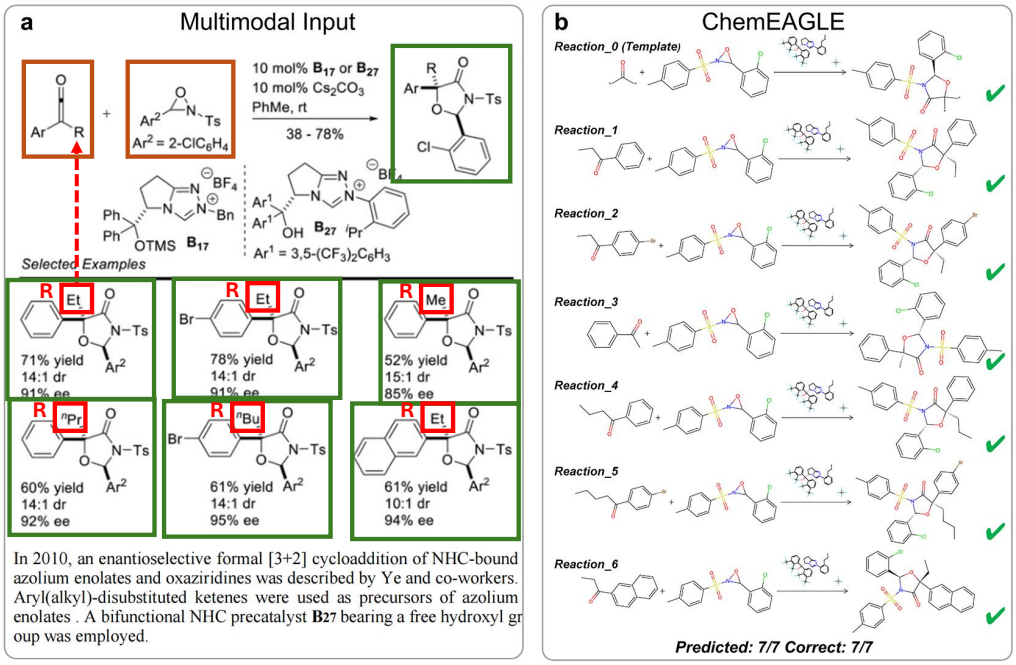


Figure 7: Example of chemical information extraction (Left: input reaction scope image. Right: expected extracted 7 reactions from the image). This figure comes from (Chen et al., 2025b).

Literature Data Collection. We collect reaction figures from papers published in *The Journal of Organic Chemistry*, *Organic Letters*, and *Journal of the American Chemical Society (JACS)* between 2000 and 2025. The resulting corpus contains 2,100 figures related to photocatalytic reactions and 430 figures related to Suzuki coupling. To avoid leakage across visually or chemically related figures from the same publication, we split the corpus into training and test sets at the article level using an 80/20 split.

Reaction Extraction and SSR Processing. We apply OpenChemIE (Fan et al., 2024) to each literature figure to recover structured reaction records. OpenChemIE first detects molecules and reaction arrows and establishes their spatial relationships, after which its OCSR module converts each localized molecular image into a molecular structure. The recognized structures are then assembled according to the detected reaction layout to obtain complete reactant–product pairs. Using raw OpenChemIE extraction, the training split contains 25,659 photocatalytic reactions and 4,233 Suzuki-coupling reactions.

Expert-Corrected Evaluation Sets. To ensure that downstream evaluation is independent of OCSR errors, we construct fixed reference test sets from the held-out literature figures. Three domain experts manually inspect and correct the automatically extracted reactions, including molecular structures, reactant–product assignments, and reaction boundaries when necessary. After correction, the test sets contain 4,291 photocatalytic reactions and 668 Suzuki-coupling reactions. The same expert-corrected test sets are used to evaluate models trained under all extraction strategies.

Downstream Reaction Prediction. For each training corpus, we fine-tune the same Qwen3-8B model (Yang et al., 2025) for forward reaction prediction, where the input consists of the extracted reactants and the target is the corresponding product. All model architecture, optimization settings, and downstream evaluation procedures are kept identical across conditions; the only difference is the post-recognition strategy used to construct the training reactions. We evaluate prediction quality using exact-match accuracy together with molecular similarity metrics based on RDKit and Morgan fingerprints. This experiment isolates the effect of extraction quality on downstream learning. Because a single OCSR error can alter molecular connectivity, atom identity, or bond order, erroneous structures may convert an otherwise valid literature reaction into an incorrect training pair. The results in Tab. 4 show that improving both the reliability and coverage of extracted structures leads to consistently better reaction-prediction models.

I Transfer to BPMN Process-Graph Recognition

To examine whether ViCoR is specific to molecular diagrams, we transfer its index-anchored render–verify–revise interface to hand-drawn BPMN process-graph recognition. The key idea remains unchanged: a base parser first predicts a structured graph, which is rendered into a spatially aligned and explicitly indexed visual reference; the VLM then verifies the prediction and produces localized, executable node/edge edits rather than regenerating the entire graph.

Dataset and task. We use hdBPMN v1.0.0 from Sketch2Process (Schäfer et al., 2022) and follow its official writer-disjoint split, which contains 704 images in total. We consider structure-only image-to-graph recognition with eight BPMN node classes and directed sequence-flow edges, excluding text, OCR, and other document metadata. Faster R-CNN (Ren et al., 2016) and YOLO11m (Khanam et al., 2024) serve as two representative base parsers.

Training and evaluation protocol. For each base parser, correction candidates for training are generated through three-fold writer-disjoint out-of-fold inference, preventing the verifier from seeing predictions produced on images used to train the corresponding parser. Decision thresholds are selected exclusively on the validation split, and the test set is evaluated once. We report the same reliability-oriented metrics used in the main experiments. In particular, OA requires strict whole-graph exact match after Hungarian node matching, such that all evaluated node labels and directed edges must be correct simultaneously.

Results. As shown in Tab. 15, ViCoR substantially improves strict whole-graph recognition with both base parsers. OA increases from 41.6% to 62.1% with Faster R-CNN and from 50.4% to 68.8% with YOLO11m, corresponding to gains of 20.5 and 18.4 percentage points, respectively. At the same time, ViCoR achieves high accepted accuracy of 95.5% and 96.3%, while retaining 57.8% and 64.8% coverage. The consistent improvements across two different parsers indicate that the

Table 15: Transfer of ViCoR to hand-drawn BPMN process-graph recognition (%). Results are reported on the official writer-disjoint split of hdBPMN v1.0.0.

Method	**OA** ↑	**AA** ↑	**Cov** ↑
Faster R-CNN base	41.60	41.60	100.00
YOLO11m base	50.40	50.40	100.00
Faster R-CNN + ViCoR	62.10	95.50	57.80
YOLO11m + ViCoR	68.80	96.30	64.80

verification–revision interface is not tied to a particular detection architecture. Since OA requires the entire predicted graph to match exactly, these gains reflect graph-level structural correction rather than improvements to isolated node detections. Overall, the results suggest that spatially aligned verification and index-anchored revision can transfer beyond OCSR to other image-to-structure tasks with explicit node–edge structure.

J Model Details

ViCoR adopts Qwen2.5-VL-3B (Bai et al., 2025) as the backbone VLM. We also summarize the architecture used in MolScribe. The detailed configurations of image encoders and language/decoder modules are reported in Tab. 16.

Table 16: Model architecture details.

	Qwen2.5-VL-3B	**MolScribe**
Image Encoder		
Architecture	Vision Transformer	Swin Transformer (Swin-B)
Layers	32	36
Hidden Size	1280	–
Attention Heads	16	–
Patch Size	14	–
Parameters	–	88M
Pre-training	–	ImageNet-22K
Input Resolution	–	384×384
Language / Decoder Module		
Architecture	Transformer LLM	Transformer Decoder
Layers	36	6
Hidden Size	2048	256
Attention / KV Heads	2 (KV heads)	8
Head Size	128	–
Positional Encoding	Rotary (LLM)	Sinusoidal
Dropout	–	0.1
Prediction Head		
Bond Predictor	–	2-layer FFN (ReLU)