

# MolRecBench-Wild: A Real-World Benchmark for Optical Chemical Structure Recognition

## Supplementary Material

### 1. Case Study

To better analyze the performance of different methods under various evaluation metrics, we conducted a visual analysis of the results from the best-performing method among the five approaches.

#### 1.1. SMILES Qualitative results

As shown in 1, we selected two samples for demonstration. In the first sample, although the final displayed SMILES are not absolutely identical, DECIMERv2.2, GTR-Mol-VLM, InternVL3.5, and Gemini 2.5 Pro all predicted the correct results while preserving the sequence numbers for different R values. On the contrary, MathPix not only represents all R groups in a unified format but also omits the prediction of P(phosphorus) atoms. In the second sample, although the diagram depicts both solid and hollow wedge bonds, this molecule is actually achiral. GTR-Mol-VLM, InternVL3.5, and Gemini 2.5 Pro effectively filtered out these distractions and provided accurate predictions. However, DECIMERv2.2 misidentified the functional group 'Ph' as '[Pb]' and introduced unnecessary chiral information. Additionally, MathPix also produced incorrect predictions.

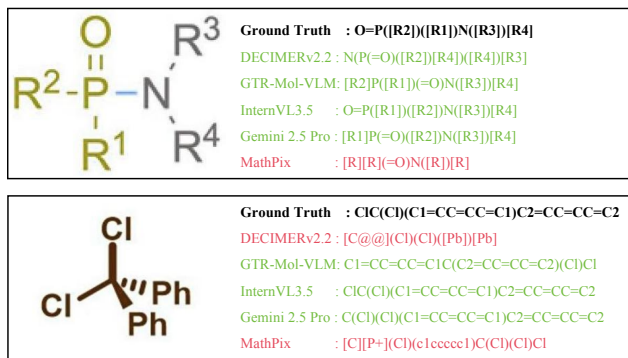


Figure 1. Comparison of SMILES results produced by different models.

#### 1.2. Simplified Graph Qualitative results

As shown in Fig. 2, we present the prediction results of the three methods in Simplified Graph. It is evident that for simple molecular results, all three methods produce correct predictions. However, for slightly more complex molecular structures, the specially trained GTR-Mol-VLM yields the most accurate predictions. Gemini 2.5 Pro incorrectly predicts all double and single bonds in the benzene ring

as aromatic bonds—though chemically correct, this is erroneous in graph-based evaluations. Finally, GPT-5 exhibits the greatest discrepancy in its predictions.

#### 1.3. Graph Qualitative results

As shown in Fig.3, we present the prediction results of GPT-5 and Gemini 2.5 Pro in Graph. Graph is the most challenging evaluation protocol in MolRecBench-Wild. Although GPT-5 and Gemini 2.5 Pro demonstrate strong instruction adherence when converting circular shapes, they exhibit poor predictive capabilities for diverse chemical bond types.

### 2. MOSAIC

The visual difficulty types are reported in Tables 1 and 2, and the chemical difficulty types are reported in Tables 3 and 4. The visual-difficulty examples demonstrate that our dataset encompasses a wide range of appearance-level perturbations, substantially increasing the challenge of accurately recognizing molecular diagrams. The chemical-difficulty examples further reveal that the dataset includes numerous uncommon or non-standard structural expression styles, which can markedly hinder a model’s ability to interpret molecules. Collectively, these examples underscore that our benchmark provides substantially broader coverage of visual interference, symbolic variability, and long-tail structural edge cases than existing OCSR test sets, offering a more rigorous and fine-grained evaluation of model robustness in realistic chemical-document scenarios.

### 3. Statistics

For each sample, we annotated the types of visual and chemical difficulty labels and counted the number of combinations of different difficulty labels that appeared. As shown in Fig. 4, each circle represents the number of samples that exhibit both types of difficulties simultaneously. This distribution offers a more fine-grained assessment of model robustness under varying levels of visual and chemical complexity.

### 4. Prompt

We designed specific prompt templates for each prediction task, as detailed below:

- **SMILES Prediction:** The corresponding prompt template is shown in Figure 5.
- **Simplified Graph Prediction:** The corresponding prompt template is shown in Figure 6.

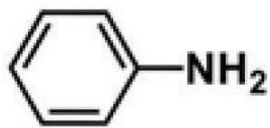
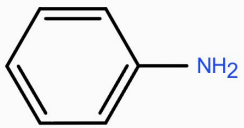
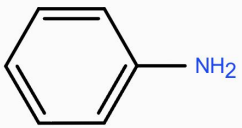
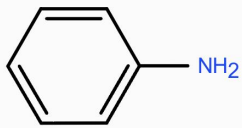
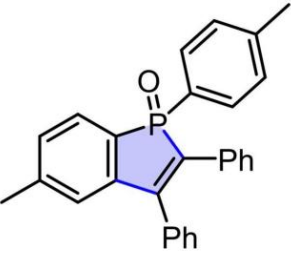
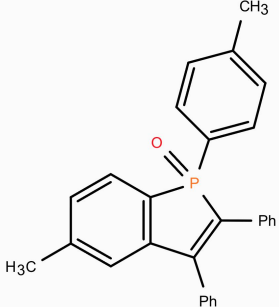
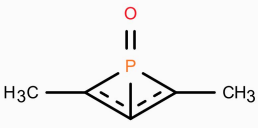
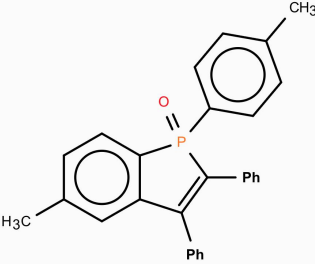
Input	GTR-Mol-VLM	GPT-5	Gemini 2.5 Pro
			
			

Figure 2. Comparison of Simplified Graph results produced by different models.

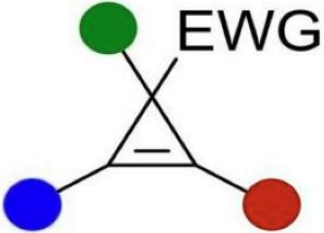
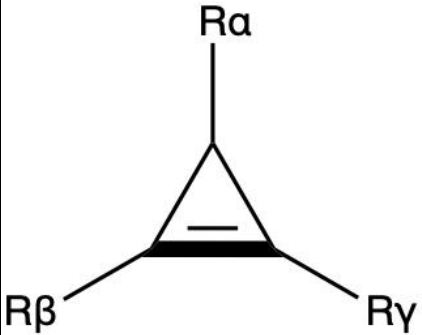
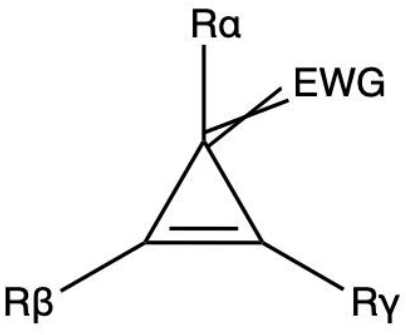
Input	GPT-5	Gemini 2.5 Pro
		

Figure 3. Comparison of Graph results produced by different models.

- **Full Graph Prediction:** The prompt template for this task, shown in Figure 9, combines two visual prompts (Figure 7 and Figure 8).

## 5. Annotation Details

We recruited a total of 47 professional chemical annotators to participate in our chemical reaction annotation task. All annotators possess backgrounds in chemistry-related disciplines, including 9 master's degree candidates and 31 bachelor's degree holders. Their academic specialties cover multiple chemistry-related fields such as Chemistry, Applied Chemistry, Materials Chemistry, Chemical Engineering and Technology, and Organic Chemistry. Prior to formally commencing the annotation work, all annotators passed rigorous chemical knowledge assessments and comprehensive annotation protocol training. This ensures their solid foundation in chemistry and consistent adherence to annotation stan-

dards, thereby guaranteeing the professionalism and reliability of the annotated data.

## 6. Source Journals of MolRecBench-Wild

The journals selected for data collection and the number of molecular images obtained from each journal are shown in Tab.5.

Table 1. Example of visual dimension hard cases, ID 1–12. Each column group lists the hard case ID, type name, and a representative molecule image.

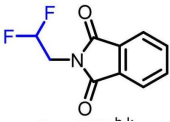
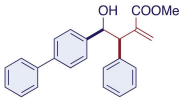
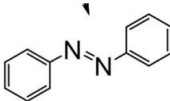
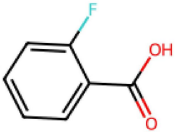
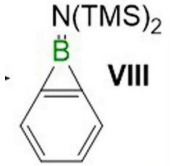
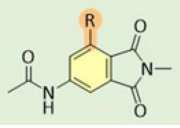
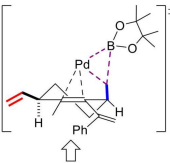
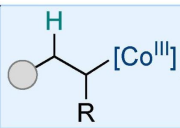
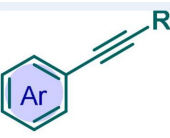
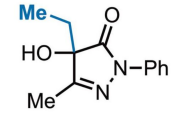
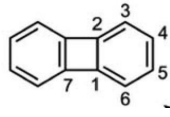
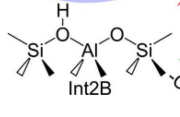
ID	Type	Image	ID	Type	Image
1	Decorated Text		2	Decorated Bond	
3	Polluted Boundary		4	Blurry Image	
5	Additional Arrow, Box, Text		6	Colored Areas or Image Background	
7	Bond Crossing		8	R Represented by Pattern	
9	Colored Ar		10	Short Bond	
11	Numbered Atom		12	Incomplete Molecule	

Table 2. Example of visual dimension hard cases, ID 13–18.

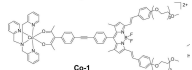
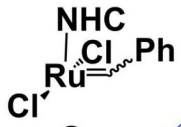
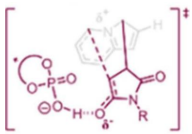

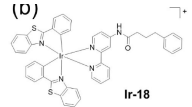
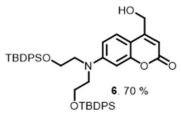
ID	Type	Image	ID	Type	Image
13	Large Molecule		14	Large Font	
15	Long Bond		16	Thick Bond	
17	Thin Bond		18	Long Functional Group Name	

Table 3. Example of chemical dimension hard cases, ID 1–12.

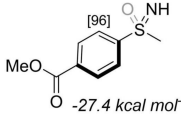
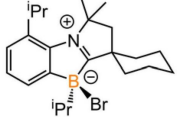
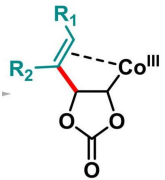
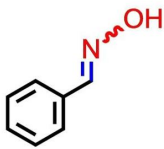
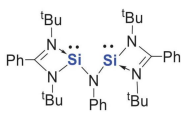
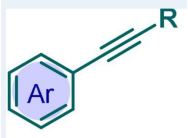
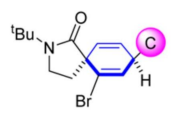
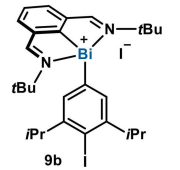
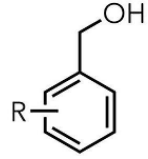
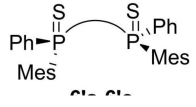
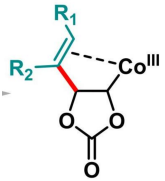
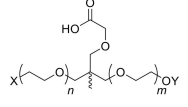
ID	Type	Image	ID	Type	Image
1	Equal-width Chiral Bond		2	Charge Symbol	
3	Dashed Bond		4	Wavy Bond	
5	Lone Pair		6	Triple Bond	
7	Hash Bond		8	Ionic Bond	
9	R on Ar with uncertain position		10	Abbreviated structure	
11	Valence Symbol		12	Polymer	

Table 4. Example of chemical dimension hard cases, ID 13–19.

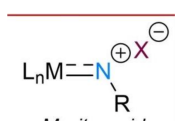
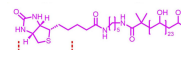
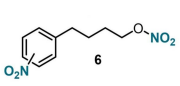
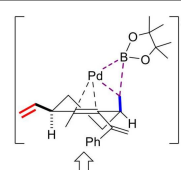
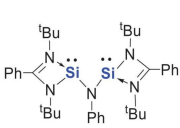
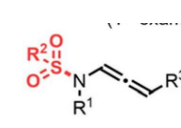
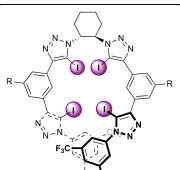
ID	Type	Image	ID	Type	Image
13	Aromatic Bond		14	Multi-group	
15	Atom on Ar with uncertain position		16	Transition State	
17	Coordination Bond		18	Consecutive Double Bond	
19	Double Dashed Bond				

Table 5. Journal Sources and the Number of Molecules Extracted from each.

ID	Journal Name	# Molecules
1	Angewandte Chemie International Edition	1490
2	Accounts of Chemical Research	164
3	ACS Catalysis	586
4	ACS Central Science	462
5	Journal of the American Chemical Society	1036
6	Nature Chemistry	17
7	Chemical Science	927
8	Nature Catalysis	83
9	Green Chemistry	264

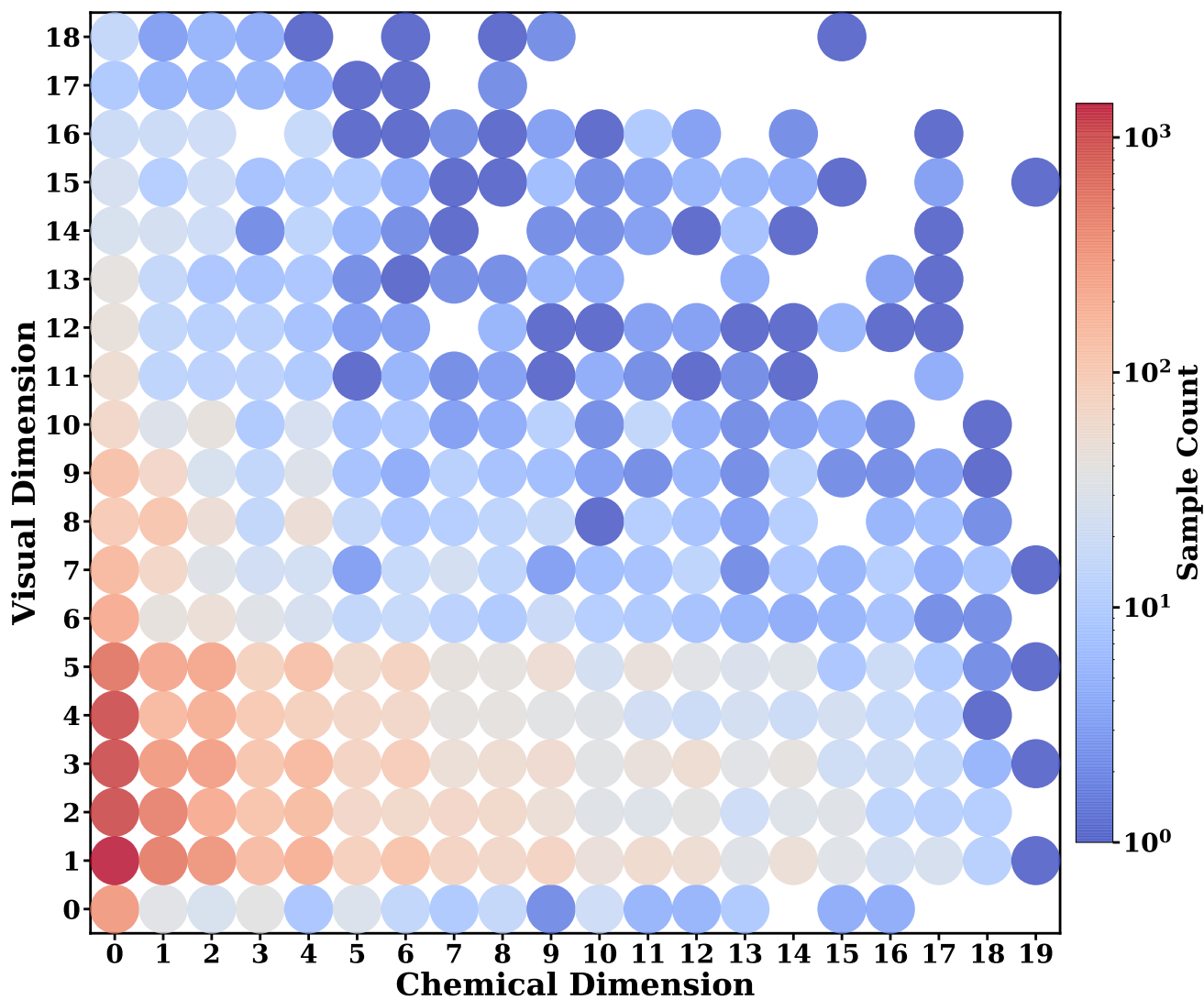


Figure 4. Joint heatmap showing the co-occurrence counts between each visual difficulty dimension and each chemical difficulty dimension in our benchmark. The horizontal axis corresponds to the indices of the chemical difficulty labels, and the vertical axis corresponds to the indices of the visual difficulty labels. The color intensity encodes the number of samples for each pair: warmer tones (reds) indicate higher counts, while cooler tones (blues) indicate lower counts.

### Prompt template for the SMILES

You are an expert chemist viewing a diagram of a chemical molecular structure. Your task is to generate the corresponding SMILES string for the molecule(s) shown in the image.

Follow these rules strictly: Follow these rules strictly:

- **Output Format:** Present the result in a single JSON format. The JSON object should contain one key, `smiles`.
- **Abbreviations:** If there are chemical formula abbreviations or groups written together (e.g., MeO, Et, Ph, Boc), treat them as a single unit and enclose them in square brackets. For example, MeO becomes `[MeO]`.
- **Stereochemistry:** If stereochemistry is indicated with wedges or dashes, use `@` or `@@` to represent the chiral center. For double bonds with defined geometry, use `/` and `\` to specify the E/Z (cis/trans) isomerism.
- **Charges and Isotopes:** Represent atomic charges (e.g., `[O-]`, `[NH4+]`) and isotopes (e.g., `[2H]`, `[14C]`) exactly as shown.
- **Error Handling:** If the image is unreadable or does not contain a chemical structure, the value for the "smiles" key should be null and you must add an error key, like: `{"smiles": null, "error": "Unable to recognize a chemical structure."}`

Here is an example of a successful conversion:

**User:** *<shot of an image showing 2-methoxypyridine-4-carbonitrile with stereochemistry>*

**Assistant:**

```
{  
  "smiles": "[MeO]c1nc(C#N)cc(C)c1"  
}
```

Figure 5. Prompt template for the SMILES.

### Prompt template for the simplified graph

You are an expert model specializing in the analysis of chemical diagrams. Your sole task is to view the provided image and convert the molecular structure into a graph representation, consisting of atoms and bonds.

Present the results as a single, complete JSON object.

Follow these instructions meticulously:

- **JSON Structure:** The final JSON object must contain exactly two top-level keys:
  - "atoms": A list of objects, where each object represents one atom.
  - "bonds": A list of objects, where each object represents one bond.
- **Atom Attributes:** Each object in the "atoms" list must contain three keys:
  - "id": A unique integer identifier for the atom, starting from 0.
  - "atom": The atom's symbol (e.g., C, O, N). If an abbreviation or group is shown (e.g., Ph, Boc), treat it as a single unit and represent it as a string (e.g., [Ph]).
  - "point\_2d": The [x, y] coordinates of the atom in the image.
- **Bond Attributes:** Each object in the "bonds" list must contain three keys:
  - "atom1" and "atom2": The integer ids of the two atoms connected by the bond.
  - "bond\_type": The type of the bond, which must be one of the following strings: `single`, `double`, `triple`, `aromatic`, `solid wedge`, or `dashed wedge`. For solid wedge and dashed wedge bonds, the direction is from atom1 to atom2. A solid wedge means atom2 is pointing out of the plane, while a dashed wedge means atom2 is pointing into the plane.
- **Strict Formatting:** You must output only the single JSON object and nothing else. Do not include any additional text, explanations, or formatting like markdown code fences.

Here is an example of a successful conversion:

**User:** <shot of a chemical molecular formula image>

**Assistant:**

```
{
  "atoms": [
    {"id": 0, "atom": "C", "point_2d": [150, 200]},
    {"id": 1, "atom": "O", "point_2d": [250, 200]},
    {"id": 2, "atom": "N", "point_2d": [150, 100]},
    {"id": 3, "atom": "[Ph]", "point_2d": [50, 100]}
  ],
  "bonds": [
    {"atom1": 0, "atom2": 1, "bond_type": "double"},
    {"atom1": 0, "atom2": 2, "bond_type": "single"},
    {"atom1": 2, "atom2": 3, "bond_type": "single"}
  ]
}
```

Figure 6. Prompt template for the simplified graph.



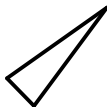



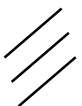
















bond type	visual example	bond type	visual example	bond type	visual example
single		single or double		hollow wedge	
double		single or aromatic		bold double	
triple		double or aromatic		dashed double	
aromatic		dative		dashed triple	
solid wedge		hydrogen		dashed dative	
dashed wedge		wavy		attachment point	
any		double either		triple with single dash	
bold		dashed bold			

Figure 7. Visual appearance of different bonds.

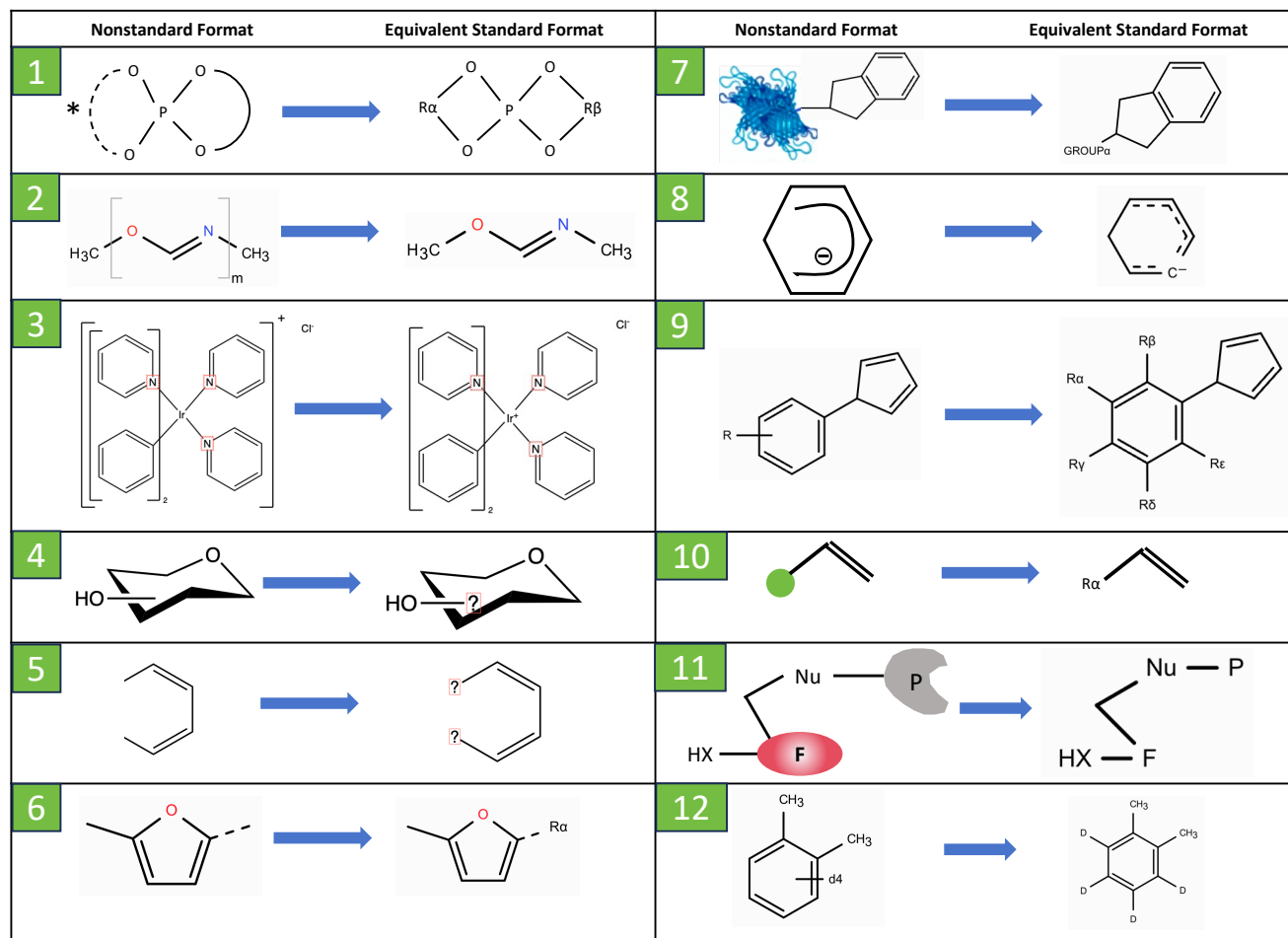


Figure 8. Different cases and their corresponding standardized forms.

### Prompt template for the graph

You are an expert model specializing in the analysis of chemical diagrams. Your sole task is to view the third (last) provided image and convert the molecular structure into a graph representation, consisting of atoms and bonds. Before you begin analyzing the target molecular structure, please carefully examine the first image provided (Figure 7), which serves as a visual reference example. This example demonstrates the different types of chemical bonds and their visual representations. Use this visual guide to accurately identify and classify bond types in the structure you will analyze.

Present the results as a single, complete JSON object.

Follow these instructions meticulously:

- **JSON Structure:** The final JSON object must contain exactly two top-level keys:
  - "atoms": A list of objects, where each object represents one atom.
  - "bonds": A list of objects, where each object represents one bond.
  - "brackets": A list of objects, where each object represents one bracket.
- **Atom Attributes:** Each object in the "atoms" list must contain the following keys:
  - "id": A unique integer identifier for the atom, starting from 0.
  - "atom": The atom's symbol (e.g., C, O, N). If an abbreviation or group is shown (e.g., Ph, Boc), treat it as a single unit and represent it as a string (e.g., [Ph]).
  - "point\_2d": The [x, y] coordinates of the atom in the image.
  - "charge": (Optional) The formal charge of the atom. An integer (e.g., 1, -1). Only include this key if a charge is explicitly shown.
  - "isotope": (Optional) The mass number of an isotope. An integer (e.g., 14 for <sup>14</sup>C). Only include this key if an isotope is explicitly shown.
  - "valence": (Optional) The explicit valence state of the atom. An integer. Only include this key if the valence is explicitly specified (which is rare).
  - "radical": (Optional) The radical state of the atom. Use 1 for a doublet (single radical electron), 2 for a singlet (carbene/nitrene), or 3 for a triplet. Only include this key if a radical is explicitly shown.
- **Bond Attributes:** Each object in the "bonds" list must contain three keys:
  - "atom1" and "atom2": The integer ids of the two atoms connected by the bond.
  - "bond\_type": The type of the bond. The value must be one of the following strings. Refer to the visual example image in (Figure 7) to understand how each bond type appears:
    - \* "single": A single bond (single line)
    - \* "double": A double bond (double line)
    - \* "triple": A triple bond (triple line)
    - \* "aromatic": Aromatic bond
    - \* "solid\_wedge": Solid wedge bond (thick line indicating atom pointing out of plane)
    - \* "dashed\_wedge": Dashed wedge bond (dashed line indicating atom pointing into plane)
    - \* "hollow\_wedge": Hollow wedge bond (open triangle indicating atom pointing out of plane)
    - \* "wavy": Wavy bond (indicating unknown or unspecified stereochemistry)
    - \* "any": Any bond type (often depicted as a simple, non-stereochemical dashed line, typically for query structures)
    - \* "bold": Bold bond (thick line, often indicating bonds in the plane of the page)
    - \* "dashed\_bold": Dashed bold bond
    - \* "dashed\_double": Dashed double bond
    - \* "dashed\_triple": Dashed triple bond
    - \* "single\_or\_double": single or double bond represented by triple dashed line
    - \* "bold\_double": Bold double bond
    - \* "double\_either": Double bond with unknown E/Z configuration
    - \* "single\_or\_aromatic": Single or aromatic bond (ambiguous)
    - \* "double\_or\_aromatic": Double or aromatic bond (ambiguous)

### Prompt template for the graph

- \* "dative" - dative bond
  - \* "dashed dative" - Dashed dative bond
  - \* "hydrogen" - Hydrogen bond
  - \* "attachment point" - Attachment point (often used in query structures)
  - \* "triple with single dash" - Triple bond with a single dash
- **Bracket Attributes:** Each object in the "brackets" list must contain two keys:
    - "atoms": A list containing the integer ids of all the atoms within a bracket.
    - "mark": A number, character or string written in the lower right corner of a bracket indicates the number of times the atoms within the brackets are repeated.
  - **Strict Formatting:** You must output only the single JSON object and nothing else. Do not include any additional text, explanations, or formatting like markdown code fences.

Example of the required output format:

```
{
  "atoms": [
    {"id": 0, "atom": "C", "point_2d": [151, 202]},
    {"id": 1, "atom": "O", "point_2d": [255, 221], "charge": -1},
    {"id": 2, "atom": "N", "point_2d": [132, 434]},
    {"id": 3, "atom": "[Ph]", "point_2d": [59, 100]},
    {"id": 4, "atom": "C", "point_2d": [276, 348], "isotope": 14}
  ],
  "bonds": [
    {"atom1": 0, "atom2": 1, "bond_type": "double"},
    {"atom1": 0, "atom2": 2, "bond_type": "single"},
    {"atom1": 2, "atom2": 3, "bond_type": "wavy"},
    {"atom1": 0, "atom2": 4, "bond_type": "solid_wedge"}
  ],
  "brackets": [
    {"atoms": [0, 1, 2], "mark": "3"},
    {"atoms": [6, 9, 10], "mark": "n"},
    {"atoms": [3, 5, 13], "mark": "n=1, 2"},
  ]
}
```

#### Important Notes:

- The first image you see is the visual example that illustrates different bond types and their visual representations. Study it carefully before analyzing the target structure.
- The third (last) image contains the chemical structure you need to analyze and convert to the graph representation.
- If you encounter non-standard styles (one or more) in the test image, please convert them to standardized chemical molecular formula styles for parsing according to the transformation format shown in the second image (Figure 8). This includes 12 common transformation cases.
- Case 1: When an asterisk (\*) or a curved arc extends from an atom in the molecular structure, it indicates an undefined attachment point. Such cases should be standardized as an R-group variable, denoted as  $R\alpha$ ,  $R\beta$ , etc., connected to the corresponding atom by a single bond.
- Case 2: If the molecular structure contains a bracket with a subscript (e.g.,  $m$ ) indicating that the atoms inside the bracket are repeated  $m$  times, you should first predict the structure as if there were no bracket. In addition, extract the information from the bracket. For example, if the atoms inside the bracket are O, C, and N, and their indices are 1, 2, and 3, then output: "brackets": [{"atoms": [1, 2, 3], "mark": "m"}].
- Case 3: For a metal complex, if the charge is placed outside the square brackets [], this formal charge should be assigned to the central metal atom. For example, in Case 3, the + charge outside the brackets should be marked as a +1 charge on the central Ir atom.
- Case 4: When an HO- group is connected to a ring via a single bond with an undefined attachment point, the other end of the bond should be represented as a wildcard ?.
- Case 5: If a bond is truncated at the edge of an image, its other end should be represented as a wildcard atom ?.

### Prompt template for the graph

- Case 6: A dashed line (representing any bond type) extending from the main structure indicates a bond to an undefined group. The other end of this bond should be treated as an R-group variable, denoted as  $R\alpha$ .
- Case 7: When a molecule is connected to a complex graphical structure, that structure should be represented as  $GROUP\alpha$ . If multiple distinct groups exist, they should be sequentially labeled  $GROUP\beta$ ,  $GROUP\gamma$ , etc.
- Case 8: When a negative charge is drawn in the center of a ring (as in a cyclopentadienyl anion), it can be assigned to any atom on the ring. In addition, incomplete arcs within a ring should be interpreted as aromatic bonds.
- Case 9: When a Markush structure (e.g., R) is connected to a ring via a bond with an undefined attachment point, an R group should be added to every carbon atom on the ring that still has an attached hydrogen. These groups should be labeled sequentially as  $R\alpha$ ,  $R\beta$ ,  $R\gamma$ , etc., using unique Greek letter subscripts.
- Case 10: In generic structures or reaction schemes, simple geometric shapes (e.g., circles, ovals, stars, squares) can represent Markush structures. They should be represented as  $R\alpha$ ,  $R\beta$ ,  $R\gamma$ , etc. All instances of the same geometric shape should share the same label, while different shapes should be assigned unique labels.
- Case 11: The model must ignore non-chemical information, such as background colors, highlights, or decorative overlays on atoms and bonds. It should only recognize the underlying chemical entity (e.g., the F atom, not the red oval on top of it).
- Case 12: A label like  $d_n$  or  $d_n$  indicates that n hydrogen atoms on an aromatic ring have been replaced by deuterium (D) atoms. These deuterium (D) atoms should be explicitly represented in the structure.

Figure 9. Illustration of the prompt template for the graph generation task. This template adopts a few-shot, in-context learning structure. It is composed of two visual exemplars, namely a bond example (Figure 7) and a case example (Figure 8), followed by the query, which is the chemical formula of the target molecule. The entire sequence is then fed into the model as a single prompt.