Supplementary materials for MarkushGrapher: Joint Visual and Textual Recognition of Markush Structures

Lucas Morin[1, 2] Valéry Weber[1] Ahmed Nassar[1] Gerhard Ingmar Meijer[1]
Luc Van Gool [2, 3] Yawei Li [2] Peter Staar[1]
[1]IBM Research [2]ETH Zurich [3]INSAIT
{lum, vwe, ahn, inm, taa}@zurich.ibm.com {yawei.li, vangool}@vision.ee.ethz.ch

Here, we provide additional details and visualizations regarding Markush structures, the benchmark datasets, the MarkushGrapher analysis, and the synthetic training set, introduced in Section 1, Section 2, Section 3 and Section 4, respectively.

1. Markush Structure

Figure 1 illustrates the different components of a Markush structure. A Markush structure contains two main components: a visual definition (referred to as Markush structure backbone) and a textual definition. The Markush structure backbone represents the core of the chemical structure template. It can identified with a Chemaxon Extended SMILES (CXSMILES) [1] string. For the Markush structure in Figure 1, the CXSMILES is:

CXSMILES

[H]C1=C([*])C([*])=C([*])C=C1N(C)C(=O)C1=CC=CC(=C1)S(=O)(=O)NC1CCCC1.CCO.*[*].*[*] |$;;;X;;X;;G1;;;;;;;;;;;;;;;;;;;;;;;;;;G2;;G4$,m:29:24.25.26.27.28,m:32:14.19.15.18.17.16,m:34:24.25.26.27.28,Sg:n:28:w:ht,Sg:n:30: :ht|

The CXSMILES is composed of two sections. The first section holds the SMILES (in black) that identifies the atoms, the bonds and the connectivity of the structure. The second section is an extension table. It contains the variable groups (in red), the position variation indicators (sections starting with 'm', in blue) and the frequency variation indicators (sections starting with 'Sg', in green). The numbers written in the position variation and frequency variation indicators correspond to the index of the atoms in the SMILES. More details can be found in the CXSMILES documentation [1]. As shown in Figure 1, the textual definition of the Markush structure defines the possible substituents for the different variable groups and frequency variation labels depicted in the Markush structure backbone.

2. Benchmark Datasets

2.1. Document Selection

To build M2S, we manually sample documents published by the US Patent and Trademark Office (USPTO), European Patent Office (EPO) and World Intellectual Property Organization (WIPO). The selected patents are published between 1999 and 2023.

To build USPTO-Markush, we sample images published by the USPTO between 2010 and 2016.

2.2. Visual Examples

Figure 2 illustrates some images randomly sampled from MarkushGrapher-Synthetic, M2S and USPTO-Markush.

2.3. Statistics

Table 1 shows some statistics on MarkushGrapher-Synthetic, M2S and USPTO-Markush benchmarks. We observe that the three benchmarks contain a large fraction of Markush structures having R-groups. USPTO-Markush contains about twice as much images with 'm' and 'Sg' sections than M2S. Given that MolScribe is unable to predict 'm' and 'Sg' sections, this clarifies why MolScribe [9] performs worse on USPTO-Markush than on M2S (see Table 1 of the main paper). Besides, the ablation study shown in Table 4 of the main paper demonstrates that adding the atom indices in the CXSMILES improves MarkushGrapher performance on USPTO-Markush substantially more than on M2S. It suggests that the atom indexing is particularly useful for predicting the 'm' and 'Sg' sections. Table 1 also reports the mean number of atoms per sample, reflecting the average size of Markush structure backbones. It is similar for all three benchmarks. Additionally, Table 1 reports the mean number of variable groups, frequency variation labels, and substituents. These metrics are correlated with the average length of textual definitions of Markush structures. These definitions are on average longer for MarkushGrapher-Synthetic compared to M2S.

3. MarkushGrapher Detailed Analysis

3.1. Impact of Input Modalities

Here, we analyze how each input modality contributes to the MarkushGrapher predictions.

Figure 3 illustrates MarkushGrapher predictions after selectively removing components from an example input. We remove, one at a time, the Markush structure textual definition from the image, the Markush structure backbone from the image, all OCR cells, OCR cells from the Markush structure textual definition, and OCR cells from the Markush structure backbone. Inputs 2, 4 and 5 in Figure 3 suggest that MarkushGrapher predicts the substituents tables using only OCR cells from the textual definition. The textual definition image appears unnecessary. According to input 4 in Figure 3, MarkushGrapher appears to utilize the image of the Markush structure backbone to predict an initial structure. This first prediction represents the shape of the structure, *i.e.* if all atom types and Markush structure features were ignored, the prediction would correct. Some common atoms such as oxygen or nitrogen are occasionally added to this initial structure, while the model does not have access to the OCR text provided during training. At this stage, most Markush structure features are ignored (R-groups and 'm' sections) or incomplete ('Sg' sections). Input 6 in Figure 3 indicates that MarkushGrapher leverages OCR cells of the textual definition to know which variable group and frequency variation indicator need to be added

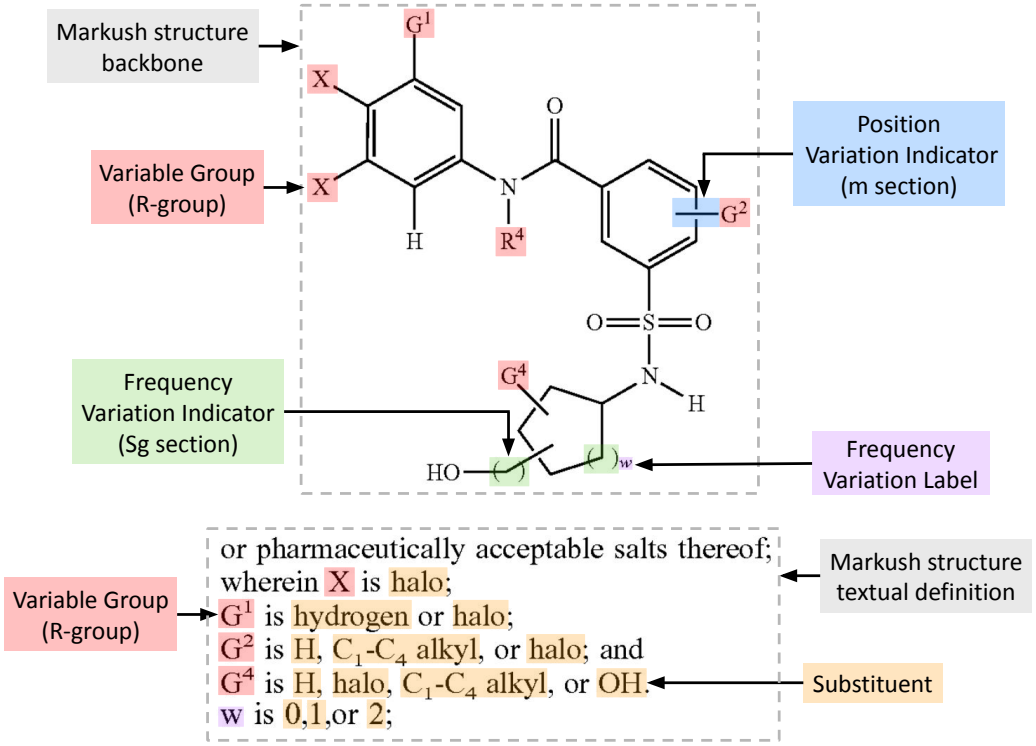
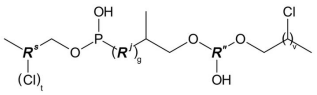


Figure 1. **Markush structure components.**. Illustration of the two mains components of a Markush structure: the backbone and the textual definition. The backbone depicts the core of the chemical structure template: atoms, bonds, connectivity, variable groups (red), frequency variation indicators (green), and position variation indicators (blue). The textual definition lists substituents (orange) that can replace their respective variable groups and frequency variation labels in the backbone.

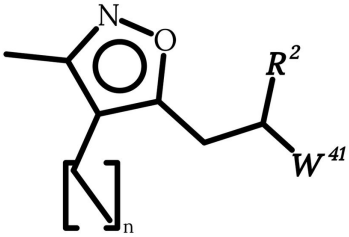
Table 1. **Benchmarks statistics.** Comparison of the number of samples, the proportion of images containing each Markush structure features (R-group, 'm' section, 'Sg' section), the number of atoms, the number of variable groups and frequency variation labels, and the number of substituents for the different benchmarks.

Benchmarks	Number of samples	Proportion of CXSMILES with at least one:			Mean number of atoms	Mean number of variable groups and frequency variation label	Mean number of substituents
		R-group	'm' section	'Sg' section			
MarkushGrapher-Synthetic	**1000**	0.95	0.54	0.39	**23**	3.9	**11**
M2S	103	**0.97**	0.30	0.25	19	2.4	9.2
USPTO-Markush	74	0.91	**0.74**	**0.42**	20	**4.7**	-

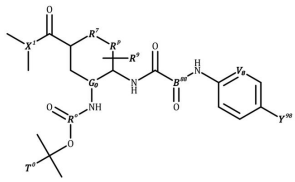
MarkushGrapher-Synthetic



wherein:
v is an integer of 0, 18 or 20;
R" is CH, and N;
t is 20 to 5;
Rj is N-alkyl;
Rs represents CH;
G is 19 to 20.

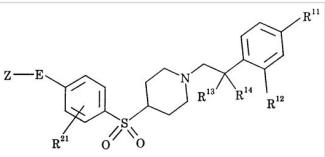


in which R2, or W41 are C1-C6 chloroalkyl and n is a number from 8-14.

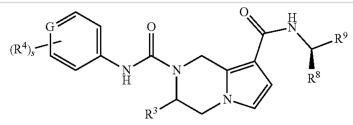


In the provided Formulas 0-5 to 4-5, B88 is individually chosen from silicon, germanium, or carbon, and R9 is a lower alkoxy group with carbon chain length of 1 to 6. Ro is either N,N,N'-Carbamimidothioate ester or N,N-Hydrazone (ketone). Y98 is either n-C4H9 or H37C18, and T0 represents an alkyl group. Rp is selected from amine (NH) or an ester bond. X1 represents a methyl group (CH), and R7 is independently chosen from a sulfonyl group (–SO2) or a simple bond. G0 is chosen from nitrogen (N), carbon (C), or methane (CH), and V8 represents a 6-membered heteroaryl ring containing 1 or 2 nitrogen atoms.

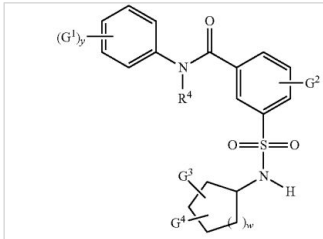
M2S



wherein Z and E are as defined in claim 1; and
R^{11}, R^{12}, R^{13}, R^{14} and R^{21} are as defined in claim 3.



wherein:
G is C or N;
R^8 is methyl; and
R^9 is methyl or ethyl, wherein the methyl or ethyl is optionally further substituted by one or more halogen.



or pharmaceutically acceptable salts thereof;

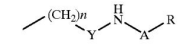
wherein G^1 is independently selected at each occurrence from CH_3, OCH_3, halo, CF_3, CCl_3, CH_2Cl, CCl_2H, CF_2H, CH_2F, and CF_3;

G^2 is H, C_1-C_4 alkyl, or halo;

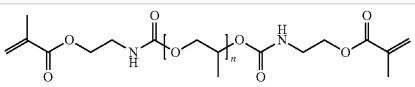
G^3 is OH, CH_2OH, or CH_2CH_2OH;

G^4 is H, OH, halo, C_1-C_6 alkyl, C_1-C_6 alkyl-OH, —C_1-C_6 chloroalkyl, —C_1-C_6 dichloroalkyl, —C_1-C_6 trichloroalkyl, —C_1-C_6 fluoroalkyl, —C_1-C_6 difluoroalkyl, —C_1-C_6 trifluoroalkyl, or phenyl, wherein the phenyl group is optionally independently substituted with 1-5 substituents selected from halo, —C_1-C_6 alkyl, and —C_1-C_6 alkoxy; and

y is 1, 2, or 3.

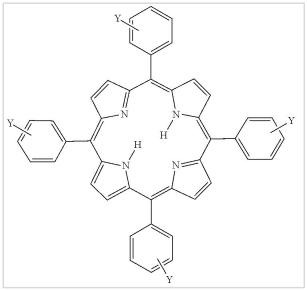


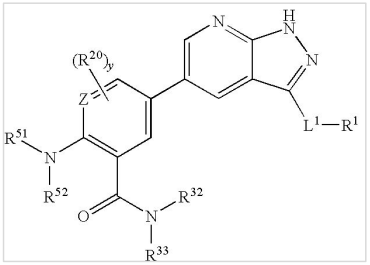
in which
n represents 1, 2, 3, 4, 5, 6, 7, 8 or 9
Y, A represent C═O or —SO_2, and
R represents an optionally substituted alkyl or aryl radical,
and at least one substituent selected from the group consisting of

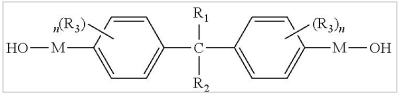


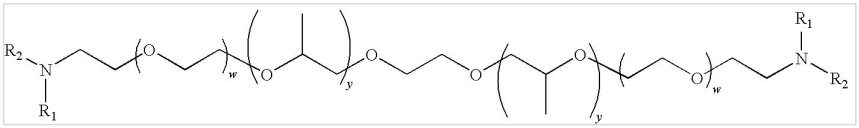
wherein n is 5 to 70.

USPTO-Markush









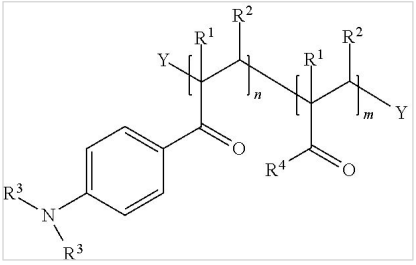


Figure 2. **Benchmarks example images.** Example images randomly selected in MarkushGrapher-Synthetic, M2S, and USPTO-Markush.

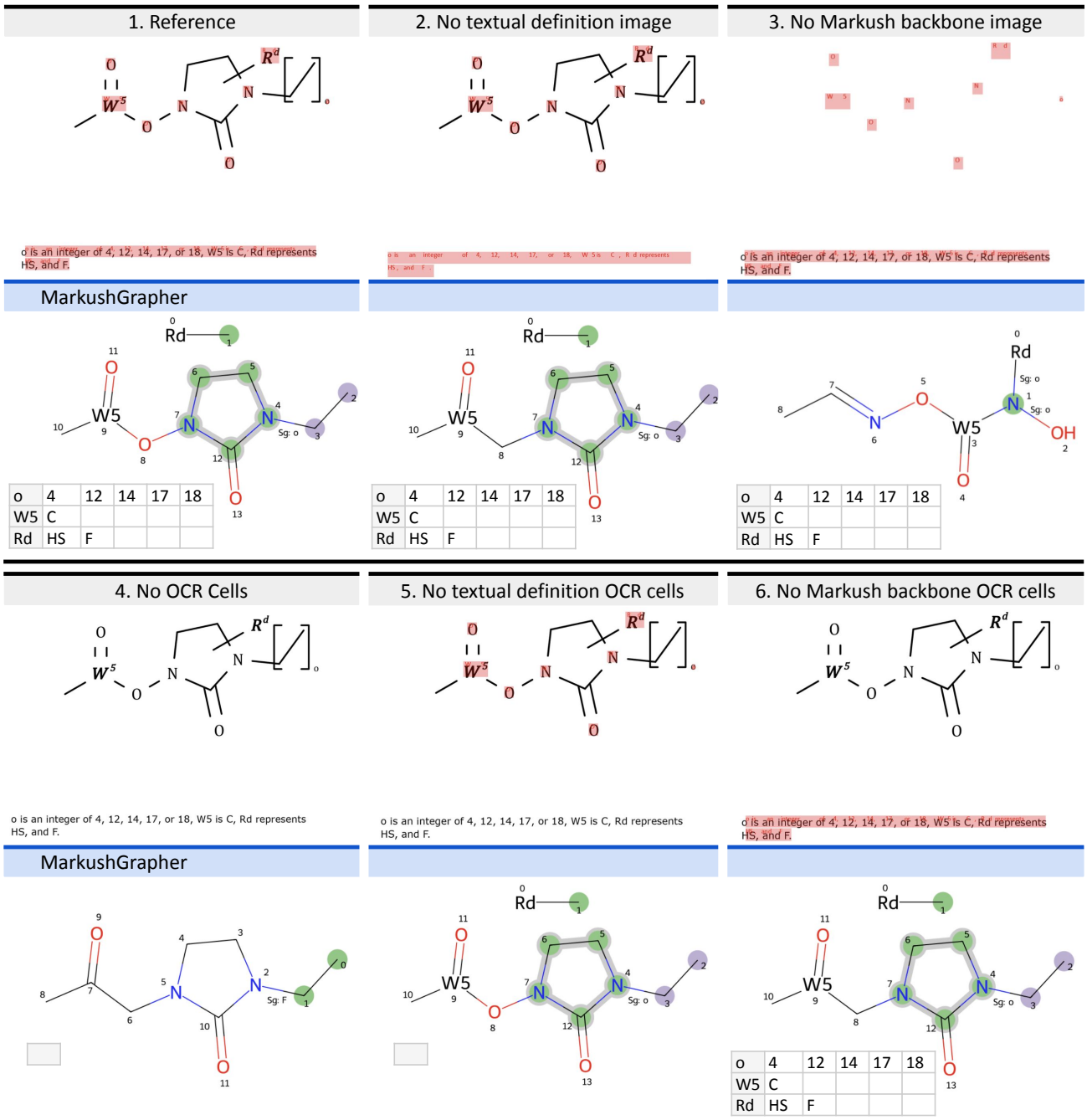


Figure 3. **Modalities removal.** MarkushGrapher predictions after selectively removing components from a reference (input 1). The input OCR boxes are highlighted in red, with the corresponding OCR text written in red as well. In input 2, the Markush structure textual definition is removed from the image. In input 3, the Markush structure backbone is removed from the image. In input 4, all OCR cells are removed. In input 5, the OCR cells in the Markush structure textual definition are removed. In input 6, the OCR cells in the Markush structure backbone are removed.

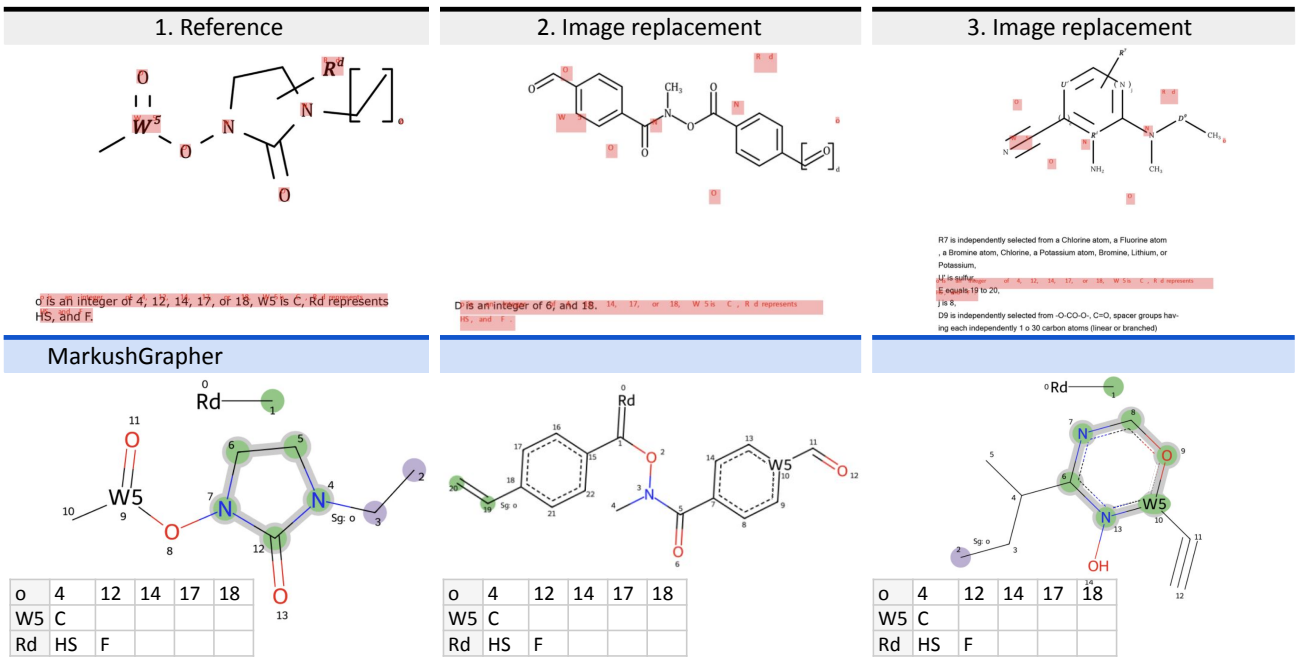


Figure 4. **Image modality replacement.**. Examples of MarkushGrapher predictions after replacing input image 1 with either image input 2 or image input 3. The input OCR boxes are highlighted in red, with the corresponding OCR text written in red as well. The OCR cells remain unchanged.

to the initial structure. Futhermore, input 3 in Figure 3 is an indication that if no input backbone image is provided, MarkushGrapher attempts to infer a compact structure that connects the OCR cells. In this case, said structure respects the valence constraints of variable groups given by the textual definition.

Figure 4 shows examples of MarkushGrapher predictions after replacing the input image with alternative images while keeping the OCR cells unchanged. The inputs 2 and 3 in Figure 4 seem to confirm that MarkushGrapher uses the image of the Markush structure backbone to predict the shape of the structure. The model appears to detect that some regions of this initial structure need to be replaced using the content of OCR cells. Most of the time, this replacement is then done using the closest OCR cell in the image. For example in input 2, the variable group 'W5' is placed at the closest location where it has 4 connections, thus correctly respecting the valence constraints given by the Markush structure textual definition.

3.2. Qualitative Evaluation

Qualitative examples. Figure 5 provides examples of MarkushGrapher predictions on real-world data. Notably, MarkushGrapher correctly predicts long tables (see input 1 in Figure 5). MarkushGrapher can also correctly recognize multi-modal Markush structures with drawing styles from the different patent offices. (In Figure 5, input 1 is published by EPO, input 2 is published by WIPO, and input 3 is published by USPTO.) MarkushGrapher also handle Markush structures backbones having a large number of variable groups (see input 9 in Figure 5), a large number of position variation indicators on the same cycle (see input 4 in Figure 5), a large number of frequency variation indicators (see input 5 in Figure 5), and function groups connected to cycles (see input 6 in Figure 5). Additionally, the model recognizes Markush backbones even when atom indices are displayed (see input 4 in Figure 5).

Failure cases. Figure 6 illustrates examples of failure cases of MarkushGrapher on the M2S benchmark. Input 1 in Figure 6 shows an inversion between a solid wedge bond and a double bond. Input 2 in Figure 6 shows an incorrect variable group label. While the OCR text correctly contains 'R2a', MarkushGrapher only predicts 'R2'. It is probably due to the OCR cells augmentations used during training (see Section 4) and the abundance of 'R2' as a variable group label. Input 3 in Figure 6 highlights an incorrect prediction for a frequency variation indicator. In this example, the bounding box of the frequency variation indicator label is near to a carbon atom, and then incorrectly associated with this atom. Similary, input 4 in Figure 6 presents an incorrect prediction for a position variation indicator. Here, a complex substructure is attached to a cycle. The model struggles with this because, during training, it only encounters R-groups and functional groups connected to cycles. The input 5 in Figure 6 shows an incorrect abbreviation prediction. The input image contains the abbreviation 'OG' (oxygen atom con-

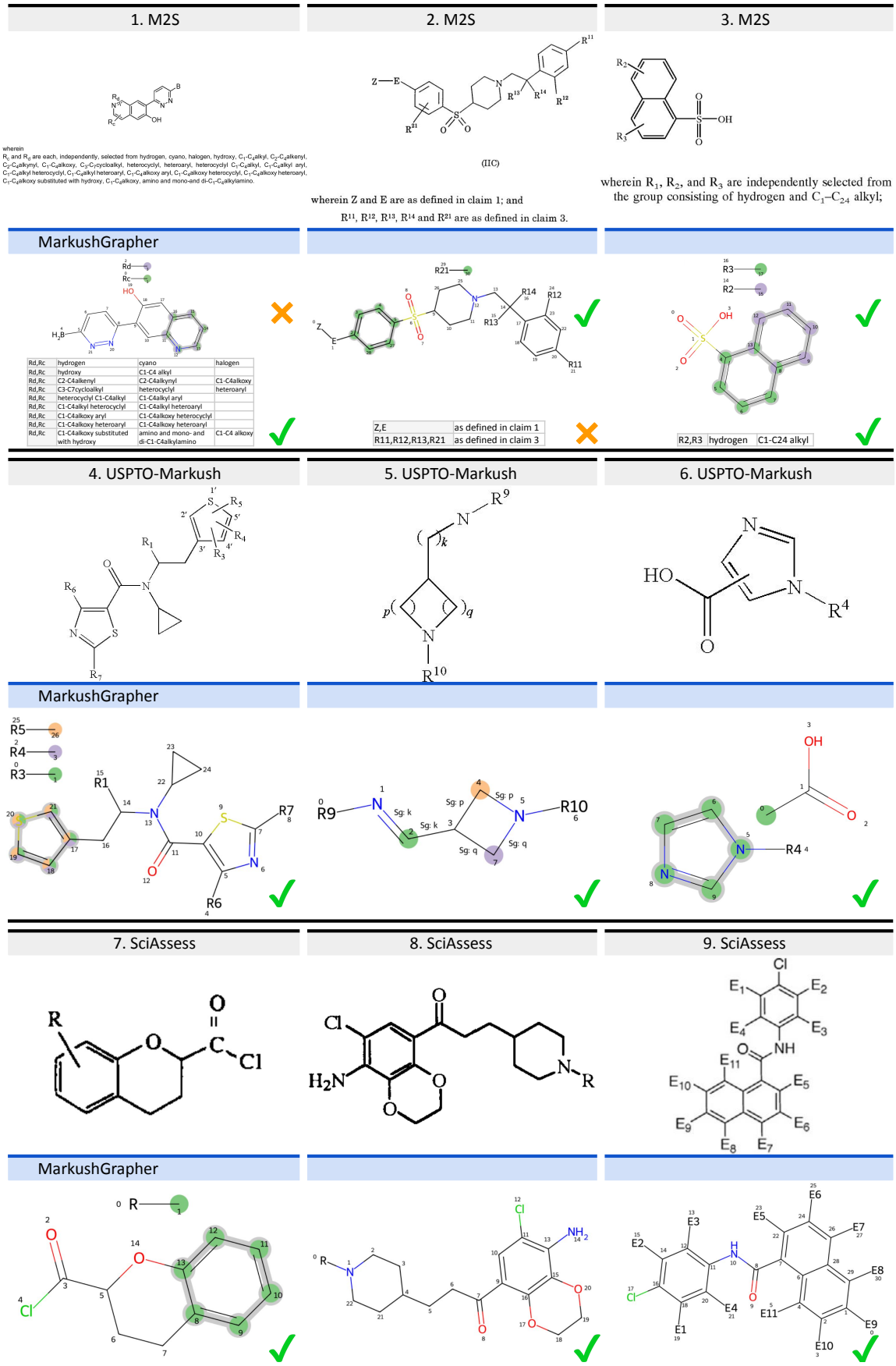


Figure 5. **Qualitative evaluation.** Examples of MarkushGrapher predictions are shown on real-world data: M2S (inputs 1, 2 and 3), USPTO-Markush (inputs 4, 5 and 6), SciAssess (inputs 7, 8 and 9).

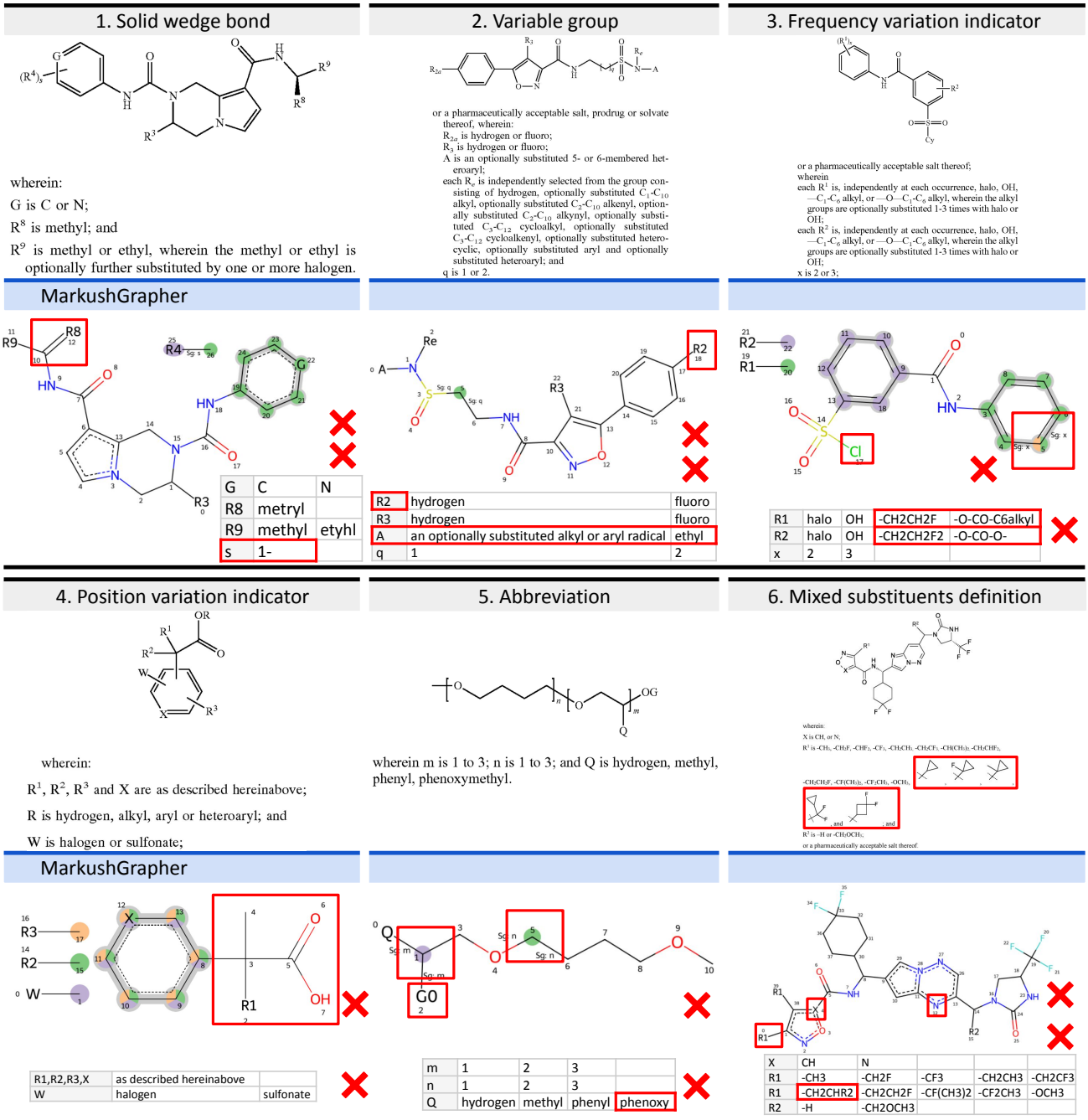


Figure 6. **Failure cases.** Example of failure cases of MarkushGrapher on real-world data from M2S. The errors are highlighted with red boxes. Typical failure cases include solid wedge bond (input 1), incorrect variable group label (input 2), incorrect frequency variation indicator (inputs 3 and 4), incorrect position variation indicator (input 4), unsupported abbreviation (input 5) and unsupported substituents definition (input 6).

nected to variable group G) but MarkushGrapher predicts a variable group 'G0', as it does not currently support abbreviations. On the same image, the frequency variation indicators represented with brackets are also only partially predicted. Input 6 in Figure 6 shows a substituent definition that combines text and interleaved chemical structure drawings. This challenging setup is currently not supported by MarkushGrapher. Additionally, we observe that the predicted substituent table occasionally contain additional labels, which are not in the input (see input 1 in Figure 6), as well as missing labels (see input 4 in Figure 6).

Besides, It is worth noting that MarkushGrapher is trained to predict 'm' sections which connect to all atoms in a cycle. Strictly speaking, this could be seen as incorrect, as some connections violate valence constraints. However, the MarkushGrapher output contains all information needed to reconstruct only the valid connections.

3.3. Model robustness

Despite being trained on synthetic data, MarkushGrapher generalizes well to real-world datasets like M2S and USPTO-M. To further validate this, we tested the model on augmented versions of these benchmarks, simulating low-quality inputs such as scanned documents, using the same augmentations applied during training (see examples of augmentations in Figure 7). Table 2 shows that MarkushGrapher maintains strong performance in these challenging scenarios.

Table 2. **MarkushGrapher robustness.** Exact match accuracy is reported on augmented (A) versions of the real-world benchmark.

Method	M2S-100-A		USPTO-Markush-A
	CXSMILES	Table	CXSMILES
MarkushGrapher	31	28	32

3.4. Inference on real-data

MarkushGrapher currently relies on ground-truth OCR cells as input. To enhance usability, OCR cells should be obtained through an OCR model. One possible approach is to use the abbreviation recognition approach introduced in MolGrapher [8]. In this method, text cells are extracted from chemical images using rule-based processing and PaddleOCR [4]. Candidate text positions are identified by filtering connected components based on size, followed by character detection and recognition with PaddleOCR. Post-processing then corrects common chemical symbol inver-

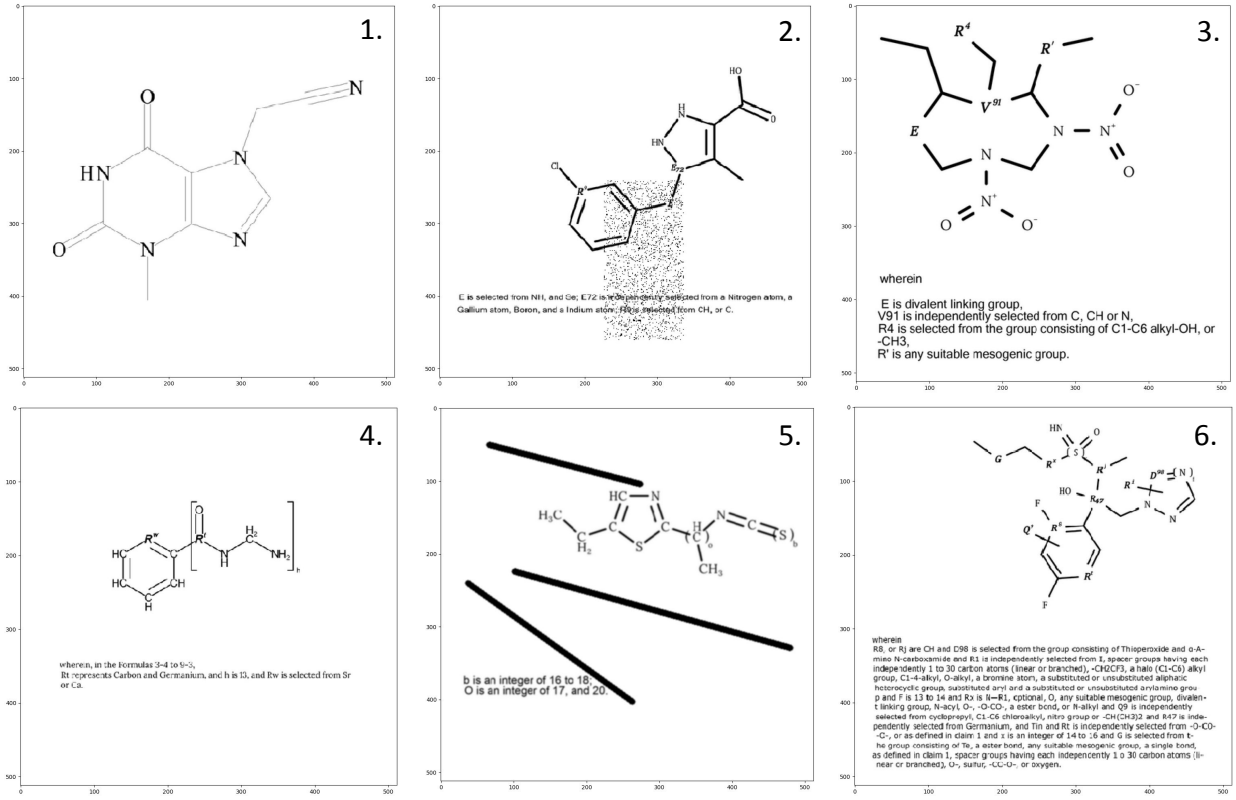


Figure 7. **Examples of training images.** Randomly selected training images augmented by applying shifting (used in examples 1 to 6), scaling (used in examples 1 to 6), downscaling (used in examples 1 to 6), gaussian blur (used in examples 1 to 6), adding random pepper patches (used in example 2) and random lines (used in example 5).

sions. Using this approach, we annotated 100 images from the standard USPTO benchmark. Manual inspection showed that 95% of predicted text cells were correct. To further improve OCR quality, training a dedicated OCR model for Markush structure images would be necessary.

4. Synthetic Training Set Details

4.1. Visualization and Image Augmentation

Figure 7 shows randomly selected training images. A small portion of training images are standard chemical structure images (see example 1 in Figure 7). Some training samples contain short (see example 2 in Figure 7) or long (see example 6 in Figure 7) textual definitions. Training images are generated using synthetic CXSMILES. To create them, we use SMILES from the PubChem [6] database and augment them using the RDKit [7] library. Based on predefined probabilities, we randomly:

- Replace atom labels by variable groups (except for atoms with charges),
- Add parentheses on atoms,
- Add brackets on pairs of atoms (except for atoms in rings),
- Connect [R-*] fragments to atoms in rings,
- Connect [R-*] fragments to rings,
- Connect functional groups to rings.

RDKit's sanitization ensures that the generated structures are chemically valid. Then, images are augmented by applying shifting, scaling, downscaling, gaussian blur, adding random pepper patches and random lines. The generated structures are chemically correct but can be probably unlikely, due to the automatic generation of substituent definitions. For example, the image 6 in Figure 7 gives for the variable group 'R47' the possible substituent 'Germanium'. It would be chemically unlikely given to the rest of the molecule.

4.2. Textual Definition Augmentation

Markush structure textual definition are generated using manually-created templates. A fraction of these definitions is then paraphrased with Mistral-7B-Instruct-v0.3 [5], using the prompt:

Prompt

I want you to augment a text description. Paraphrase it without changing its semantic meaning, but only its formulation. Do not add or remove any information. Use the writing style of patents in the chemistry domain. To help you preserving the semantic meaning of the description, a dictionary is also provided. Its keys and values should not be modified in the augmented text description. Directly answer with one augmented text description, and nothing else. Do not give any dictionary output. Text description (to be paraphrased): **Description**. Dictionary input (for context only): **Substituent table**.

The variables 'description' and 'substituent_table' are replaced for each textual definition to be augmented. For example, if the initial description is:

Description

in which M31 or Rj are lower alkyl, an alkenyl, CH(CH3)2, and heteroarylcarbonyl, and Ry is selected from the group consisting of N-alkyl, oxygen, a hydrocarbon group or optional, and Rp is selected from CH, C or N, and R4 is selected from a hydrocarbon group and N-aryl, and T9 represents C2-C6-alkenylcarbamoyl, and M4 represents a Nitrogen atom, and E is selected from the group consisting of NH, sulfur, hydrocarbon group or -O-CO-, and W1 is a Beryllium atom.

And the substituent table is:

Substituent table

```
{
  'M31': [
    'lower alkyl', 'an alkenyl', 'CH(CH3)2',
    'heteroarylcarbonyl'
  ],
  'Rj': [
    'lower alkyl', 'an alkenyl', 'CH(CH3)2',
    'heteroarylcarbonyl'
  ],
  'Ry': [
    'N-alkyl', 'oxygen', 'a hydrocarbon group',
    'optional'
  ],
  'Rp': [
    'CH', 'C', 'N'
  ],
  'R4': [
    'a hydrocarbon group', 'N-aryl'
  ],
  'T9': [
    'C2-C6-alkenylcarbamoyl'
  ],
  'M4': [
    'a Nitrogen atom'
  ],
```

```
  'E': [
    'NH', 'sulfur', 'hydrocarbon group', '-O-CO-'
  ],
  'W1': [
    'a Beryllium atom'
  ]
}
```

The resulting paraphrased description is:

Paraphrased description

In said composition, M31 or Rj may encompass lower alkyl, alkenyl, CH(CH3)2, and heteroarylcarbonyl, while Ry may be chosen from the group consisting of N-alkyl, oxygen, a hydrocarbon group, or an optional group. Rp may be selected from CH, C, or N, and R4 may be chosen from a hydrocarbon group or N-aryl. T9 represents C2-C6-alkenylcarbamoyl, and M4 represents a Nitrogen atom. E may be selected from the group consisting of NH, sulfur, a hydrocarbon group, or -O-CO-, and W1 denotes a Beryllium atom.

4.3. OCR Cells Augmentation

Using our synthetic generation pipeline, we can generate multi-modal Markush structure images along with OCR cells of all text in the image. During training, we apply augmentations to the OCR cells by shifting the position of OCR boxes, and modifying the OCR text by simulating OCR errors. These text augmentations include character substitution, character insertion, character deletion, characters transposition and case alteration.

5. Limitations and Future Works

Currently, we made the choice to not handle abbreviations in MarkushGrapher. As future work, we aim to train an OCR dedicated to the detection and recognition of text in multi-modal Markush structures images. We plan to apply MarkushGrapher at scale to build a large scale database of Markush structures and make it searchable by extending Markush structures encoding techniques [2, 3].

References

[1] ChemAxon Extended SMILES (CXSMILES) Documentation. https://docs.chemaxon.com/display/docs/formats_chemaxon-extended-smiles-and-smarts-cxsmiles-and-cxsmarts. Accessed: January 2025. 1

[2] John M. Barnard, Michael F. Lynch, and Stephen M. Welford. Computer storage and retrieval of generic chemical structures in patents. 2. GENSAL, a formal language for the description of generic chemical structures. *Journal of Chemical Information and Computer Sciences*, 21(3):151–161, 1981. 10

[3] David A. Cosgrove, Keith M. Green, Andrew G. Leach, Andrew Poirrette, and Jon Winter. A System for Encoding and Searching Markush Structures. *Journal of Chemical Information and Modeling*, 52(8):1936–1947, 2012. 10

[4] Yuning Du, Chenxia Li, Ruoyu Guo, Xiaoting Yin, Weiwei Liu, Jun Zhou, Yifan Bai, Zilin Yu, Yehua Yang, Qingqing Dang, and Haoshuang Wang. PP-OCR: A Practical Ultra Lightweight OCR System. *CoRR*, abs/2009.09941, 2020. 8

[5] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of Experts. *arXiv preprint arXiv:2401.04088*, 2024. 9

[6] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, Leonid Zaslavsky, Jian Zhang, and Evan E Bolton. PubChem 2019 update: improved access to chemical data. *Nucleic Acids Research*, 47(D1):D1102–D1109, 2018. 9

[7] Greg Landrum. RDKit: Open-Source Cheminformatics Software. http://www.rdkit.org/. Accessed: January 2025. 9

[8] 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 (ICCV)*, pages 19552–19561, 2023. 8

[9] 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. 1