Atom-Level Optical Chemical Structure Recognition with Limited Supervision

Supplementary Material

7. Source code and datasets

7.1. Source code

The source code and basic instructions are available on https://github.com/molden/atomlenz

7.2. Datasets

Several datasets were used in this work and all are available.

7.2.1 Hand-drawn images dataset

The dataset introduced by Brinkhaus et al. [4], which consists of hand-drawn chemical depictions matched with their corresponding SMILES representations, is partitioned into 4,070 samples for training and validation purposes, along with an additional 1,018 samples for testing. These sets are referred to as the *hand-drawn training set* and the *hand-drawn test set* and available here: https://dx.doi.org/10.6084/m9.figshare.24599412

The *hand-drawn training set* was then relabeled using EditKT* to annotate corresponding bounding box labels for 1417 images (see Experiments Section 5.3). The format of the bounding box labels are further explained in Section 7.2.2. The dataset is available here: https://dx.doi.org/10.6084/m9.figshare.24599172

To streamline the process, the *hand-drawn training set* is offered in different formats together with instructions to assist in training other baseline models (see Experiments Section 5.3). When applicable, localization annotations are also included. The different datasets are available here:

- DECIMER format: https://dx.doi.org/10.6084/m9.figshare.24591252
- Img2Mol format https://dx.doi.org/10.6084/m9.figshare.24591381
- MolScribe format https://dx.doi.org/10.6084/m9.figshare.24591300
- ChemGrapher format https://dx.doi.org/10.6084/m9.figshare.24591495

7.2.2 Synthetically generated dataset

For the pretraining of the object detection models of AtomLenz, we generate images synthetically using RdKit [1] and Indigo [22] paired with bounding boxes delineating all objects within, including atoms, bonds, charges, and stereocenters, similarly to what is used in other chemical structure recognition tools [19, 23]. Specifically, we collect approximately 214,000 chemical compounds in SMILES format from the ChEMBL [8] database. To enhance the method's resilience to stylistic variations, we introduce variability in elements such as fonts, font sizes, line widths, and the spacing between multiple bonds during image generation. Dataset is available in 2 parts:

- **part 1** atom and bond entity annotated images: https://zenodo.org/records/10185264
- **part 2** charge and stereocenter entity annotated images: https://zenodo.org/records/10200185

Example label file:

```
label,xmin,ymin,xmax,ymax
0,267,522,286,541
2,317,489,336,508
0,313,429,332,448
0,363,396,382,415
2,360,337,379,356
0,306,310,325,329
2,256,343,275,362
0,370,516,389,535
0,374,576,393,595
0,428,603,447,622
```

```
2,478,570,497,589
0,474,510,493,529
2,524,477,543,496
0,578,504,597,523
0,628,471,647,490
0,682,498,701,517
0,732,465,751,484
0,728,405,747,424
0,675,378,694,397
6,431,663,450,682
3,581,564,600,583
6,671,318,690,337
0,260,403,279,422
0,421,483,440,502
0,625,411,644,430
```

Above an example csv label file is illustrated of the bounding box labels for one image. There are several fields in the csv file:

- **label** field will annotate for every bounding box in the image what class the atom,bond,charge or stereo center the entity belongs to. For atom-type entities these are the different possible labels:

```
        {0: ’C’, 1: ’H’, 2: ’N’, 3: ’O’, 4: ’S’, 5: ’F’, 6: ’Cl’,
    7: ’Br’, 8: ’I’, 9: ’Se’, 10: ’P’, 11: ’B’, 12: ’Si’,
    13: ’⋆’, 14:’Te’, 15:’Sn’, 16: ’As’, 17:’Al’, 18:’Ge’,
    19:’D’, 20:’T’}
```

  For bond-type entities the different possible labels are:

```
        {1: ’single’, 2: ’double’, 3: ’triple’,
        4: ’aromatic’, 5: ’wedged’, 6: ’dashed’}
```

  For charge-type entities the different possible labels are:

```
        {0: 0, 1: +1, 2: -1, 3: +2, 4: -2, 5: +3, 6: +4, 7: +5, 8: +6}
```

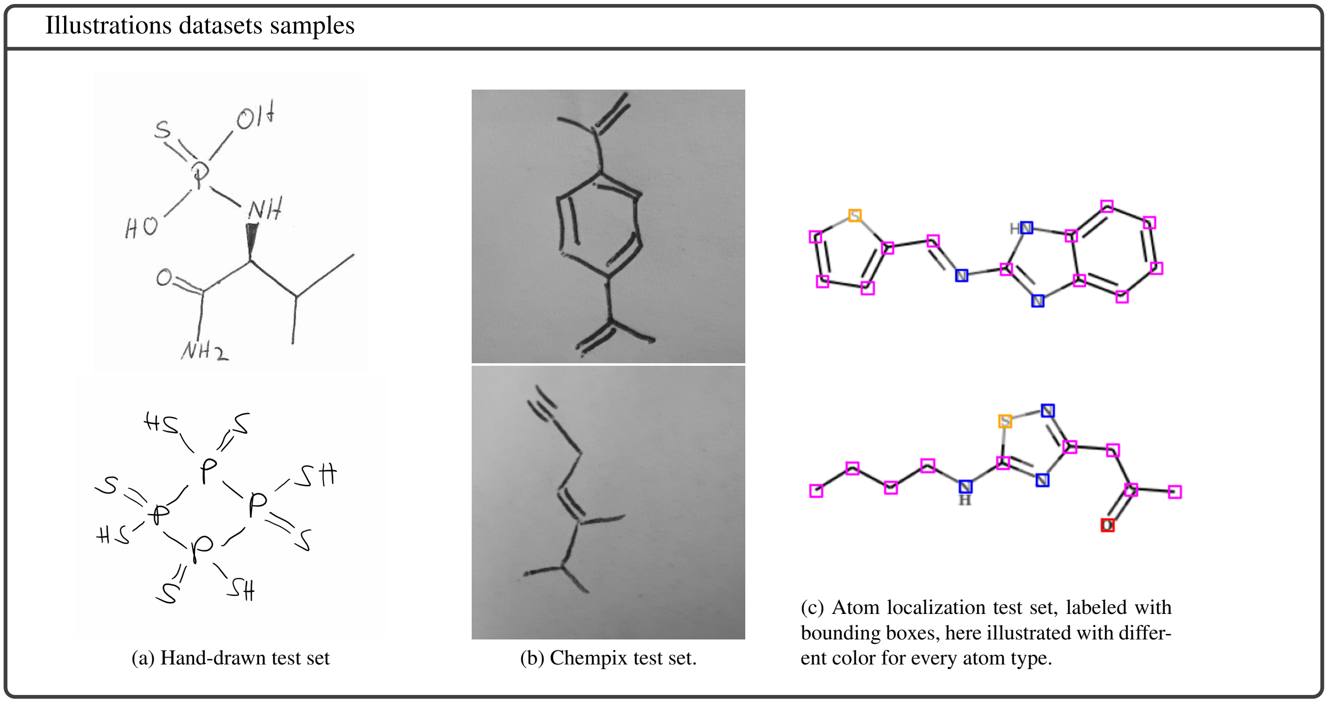
  Finally for stereocenters entities:

```
        {0:’stereocenter’}
```

- **xmin,ymin:** coordinates of top left corner of the bounding box.
- **xmax,ymax:** coordinates of the bottom right corner of the bounding box.

Examples of samples from the synthetically generated training set are illustrated in Figure 10 together with the drawn bounding box labels for the different object types. Also some extra examples of samples of all test sets described in Section 4 are illustrated in Figure 5.

Illustrations datasets samples

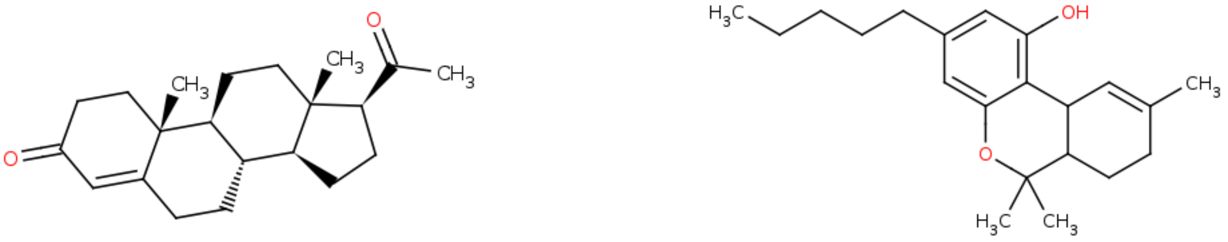


(a) Hand-drawn test set

(b) Chempix test set.

(c) Atom localization test set, labeled with bounding boxes, here illustrated with different color for every atom type.

Figure 5. Different example samples for the different datasets used in experiments.



(a) progesterone (chemical formula $C_{21}H_{30}O_2$).

(b) Δ9-tetrahydrocannabinol (chemical formula $C_{21}H_{30}O_2$).

Figure 6. Constitutional isomerism between two unrelated compounds.

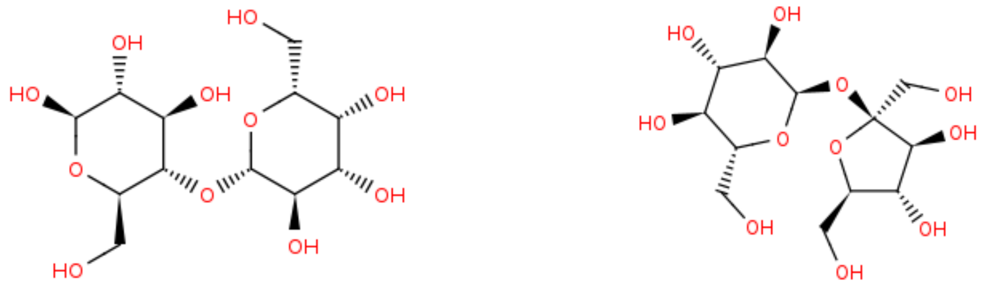
8. Illustrations of isomers

In all imperfect representation levels there are compounds that cannot be distinguished. These undistinguishable groups correspond to the concept of isomerism in chemistry. In the level of molecular formula where only the count of different atoms are given, these equivalent compounds called constitutional isomers. Compound graphs with identical adjacency matrix but different spatial organization are called stereoisomers. In the following we give examples to help clarify these concepts.

8.1. Constitutional isomerism

Constitutional isomerism is a quite simple concept. It is clear that if we specify the number of atoms for all type, multiple possible compound graphs can be built. There are valence constraints of course, for example a C_nH_{2n+2} compound cannot contain double or triple bonds. However, there is still a large variety of graphs that can be realized.

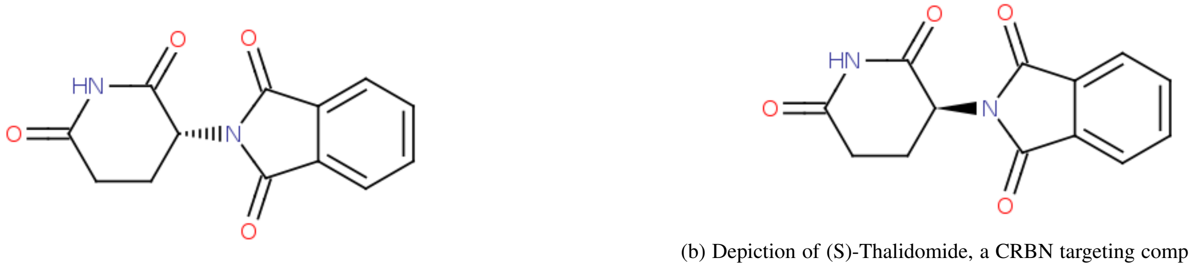
The case of sucrose and lactose (see Figure 7) is easy to follow as the galactose and fructose unit only different in the position of the ring closure. A more accidental case is progesterone and THC depicted on Figure 6. This starkly illustrates the pitfalls of using the chemical formula as a representation. The effects of these two compounds are clearly unrelated.



(a) Depiction of lactose (chemical formula $C_{12}H_{22}O_{11}$).

(b) Depiction of sucrose (chemical formula $C_{12}H_{22}O_{11}$).

Figure 7. Constitutional isomerism between two compounds from the same family (disacharides).



(a) Depiction of (R)-Thalidomide, a compound with sedative effect.

(b) Depiction of (S)-Thalidomide, a CRBN targeting compound with teratogenic effect

Figure 8. Illustration of stereochemisty and its depiction: wedge bonds.

8.2. Stereoisomerism

If we only take into account the atom and bond adjacency relations we have some relevant degree of freedom undescribed. An often used example is our hands. While all bones have the same adjacency in both of our hands, we cannot rotate the two such that they are identical: they are mirror images.

Note two important details. Firstly, we do not care about exact positions of atoms in 3D space when the molecule is flexible, similarly as we do not distinguish a hand with closed or opened fingers, but differentiating between the left and right hand. Secondly, the spatial organization has nothing to do with the placement of the atoms on the 2D depiction plane, these positions are arbitrary.

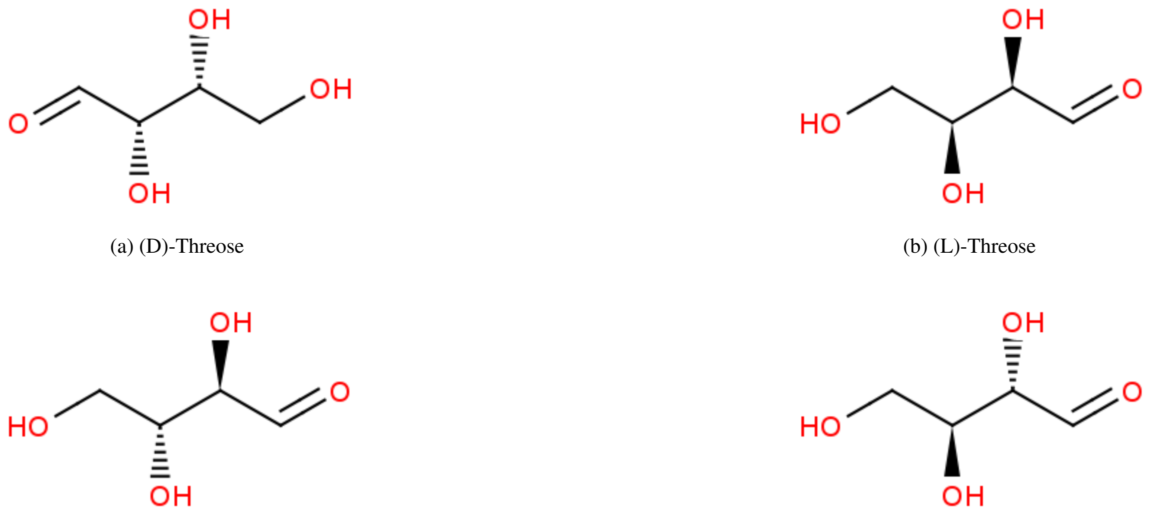
To enhance our representation, new labels need to be introduced: wedge bonds and/or stereocenters. For example see Figure 8. The filled wedge bond indicates that the atom or group at the thick end pointing out of the plane of the drawing, while the dashed wedge bond indicates that the group is under that plane. The depicted compounds are mirror images of each other, however, the difference in the biological effect can be dramatic (in the case of thalidomide the picture is more complicated, as the two form can interconvert in the body, but for didactic pourposes let us assume this is not the case). Note that if the left ring would be symmetric, for example by connecting the nitrogen to the neighboring carbon, the two compound would be identical. A simple 180 degree rotation around the long axis of the compound would show this. Stereoisomerism necessitates the presence of an atom lacking symmetric surroundings. This unique atom, such as the carbon at the wedge bond in this scenario, is referred to as a stereo center.

Stereoisomers are not always mirror images of each other. If there are n stereocenters in a molecule (see Figure 9) there are 2^n stereoisomers, forming pairs of mirror images (called enantiomers). The non-mirror image pairs are called diastereomers.

9. Details of graph algorithm subroutines

This section aims to provide in-depth insights into the subroutines utilized within the molecular graph constructor as introduced in Algorithm 1.

- The first subroutine used in the molecular graph constructor is $\mathrm{filterAtoms}(\mathbf{O}^a)$. This subroutine is implemented inside the function `iou_filter_bboxes` in the file `utils_graph.py`.



(a) (D)-Threose

(b) (L)-Threose

(c) (D)-Erythrose

(d) (L)-Erythrose

Figure 9. Illustration of stereoisomeric relations: enantiomers and diastereomers. All four compound differs only in the orientation of the two OH grops. While the compounds on the right can be transformed to the compound on the left by mirroring (they are enantiomers in pairs) the compounds in top of each other cannot (they are diastereomers)

The function goes over all overlapping bounding boxes of atoms with IoU higher than 0.5. For every group of overlapping bounding boxes the function will keep the bounding box with the highest score.

- $\text{checkCharges}(\mathbf{O}^c, \mathbf{o}^a)$ is responsible for determining which atom objects should carry a charge and is implemented in `predict_smiles.py` from line `95` until `99`:

```
95 charge_atoms = np.ones(len(filtered_bboxes))
96 for index,box_atom in enumerate(filtered_bboxes):
97    for box_charge,label_charge in zip(filtered_ch_boxes,filtered_ch_labels):
98        if bb_box_intersects(box_atom,box_charge) == 1:
99            charge_atoms[index]=label_charge
```

- $\text{checkStereoChem}(\mathbf{O}^s, \mathbf{o}^a_c)$ is applied to identify atoms functioning as stereocenters and is implemented in `predict_smiles.py` from line `141` until `151`:

```
141 stereo_bonds = np.where(mol_graph>4, True, False)
142 if np.any(stereo_bonds):
143       stereo_boxes = stereo_preds[image_idx]['boxes'][0]
144       stereo_labels= stereo_preds[image_idx]['preds'][0]
145       for stereo_box in stereo_boxes:
146         result=[]
147         for atom_box in filtered_bboxes:
148            result.append(bb_box_intersects(atom_box,stereo_box))
149         indices = [i for i, x in enumerate(result) if x == 1]
150         if len(indices) == 1:
151            stereo_atoms[indices[0]]=1
```

- $\text{checkEdge}(V, \mathbf{o}^b)$ evaluates which vertices (atoms) overlap with the bonds and is implmented in `predict_smiles.py` from line `109` until `118`:

```
109 result = []
```

```
limit = 0

while result.count(1) < 2 and limit < 80:
     result=[]
     bigger_bond_box = [bond_box[0]-limit,
     bond_box[1]-limit,bond_box[2]+limit,bond_box[3]+limit]
     for atom_box in filtered_bboxes:
        result.append(bb_box_intersects(atom_box,bigger_bond_box))
    limit+=5
indices = [i for i, x in enumerate(result) if x == 1]
```

- filterCands(candAtoms) will select the two most probable atoms to form a bond when more than 2 candidate atoms appear. This step is implemented in `dist_filter_bboxes(cand_bboxes)` in file `utils.graph.py`.
- Finally the validation step is performed by performing several iterations this code:

  ```
  mol =  Chem.MolFromMolFile('molfile',sanitize=False)
  problematic = 0
  try:
     problems = Chem.DetectChemistryProblems(mol)
     if len(problems) > 0:
        mol = solve_mol_problems(mol,problems)
  ```

  Where `solve_mol_problems` is implemented in file `utils_graph.py`.

10. Illustrations of types of atom-level entities

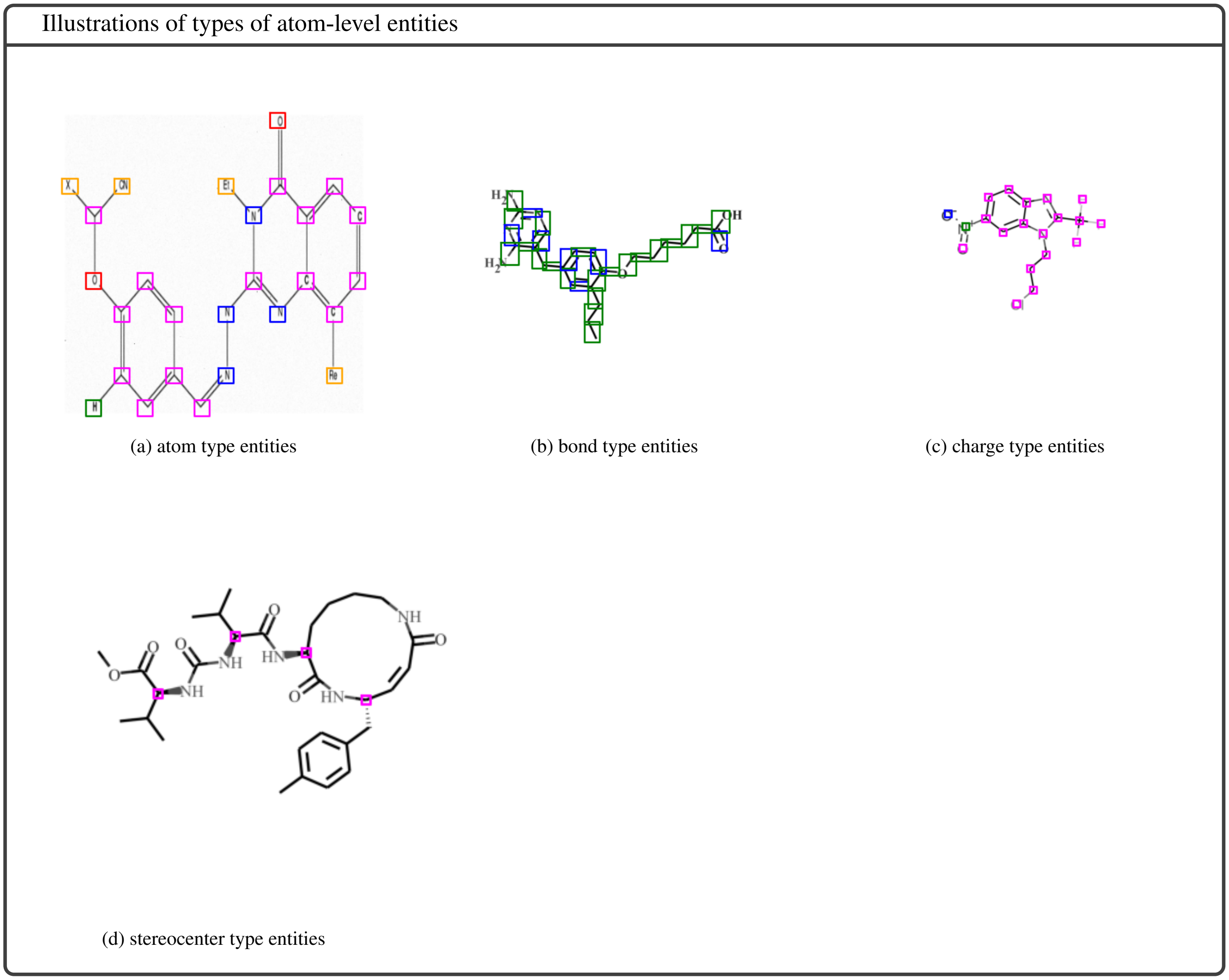
Examples of samples from the synthetically generated training set are illustrated in Figure 10 together with the drawn bounding box labels for the different atom-level entity types: atoms, bonds, charges and stereocenters.

11. All results

In our experiments we assess the molecular structure prediction performance using accuracy and Tanimoto similarity, a widely used metric for quantifying molecular similarity, to assess the resemblance between the model's predictions and the actual molecular graphs. Tanimoto similarity values range from 0 to 1, with higher values indicating greater similarity. A Tanimoto similarity of 1 indicates that the structural descriptors are identical or that they are matching 'on-bits' in a binary fingerprint. The binary fingerprint employed to measure the Tanimoto similarity is the Extended-connectivity fingerprint [28] with radius 3 (ECFP6) and fingerprint length of 2048. Crafted with precision to capture essential molecular features relevant to molecular activity, ECFPs (Extended-Connectivity Fingerprints) [28] are generated through a customized adaptation of the Morgan [18] algorithm. This involves systematically traversing each atom in the molecule to extract all possible paths within a specified radius. Following this, every unique path undergoes hashing into a numerical value within a predetermined bit range. It is worth noting that the encoded fragment size expands proportionally with an increased radius.

Our Tables 4 and 5 report both the accuracy, computed by counting the instances where the predicted structures have identical structural ECFP6 descriptors (denoted by a Tanimoto similarity of 1) and the average Tanimoto similarity. As an additional metric, we include the accuracy when assessing whether the predicted resulting SMILES exactly match the true SMILES.

Lastly, we conduct supplementary experiments utilizing ChemExpert on both the Chempix and hand-drawn test sets, while altering the sequence of chemical structure tools. In both datasets, we note that the combined utilization of AtomLenz+EditKT* and DECIMER fine-tuned within ChemExpert yields the best performance. Nevertheless, the arrangement of tools within ChemExpert slightly alters the performance, depending on the test set and the specific performance metric, as demonstrated in Table 5.



Illustrations of types of atom-level entities

(a) atom type entities

(b) bond type entities

(c) charge type entities

(d) stereocenter type entities

Figure 10. Different illustrations of types of atom-level entities

Method	Acc. (exact match)	Acc.($T = 1$)	$\overline{T}$
DECIMER (v2.2.0) [24]	0.281	0.295	0.451
DECIMER fine-tuned(v2.2.0) [25]	0.567	0.622	0.727
Img2Mol [5]	0.047	0.084	0.275
MolScribe [23]	0.094	0.102	0.288
ChemGrapher [19]	0.002	0.002	0.065
OSRA [7]	0.006	0.006	0.065
AtomLenz	0.008	0.009	0.087
AtomLenz+EditKT*	0.279	0.338	0.484
ChemExpert(AtomLenz+EditKT*,[25])	0.416	0.417	0.572
ChemExpert([25],[24])	0.571	0.626	0738
ChemExpert([25],AtomLenz+EditKT*)	**0.579**	**0.635**	**0.749**

Table 4. Benchmark results on target domain (hand-drawn images test set). Both the accuracy, computed by counting the instances where the predicted structures have identical structural ECFP6 descriptors (denoted by a Tanimoto (T) similarity of 1) and the average Tanimoto similarity ($\overline{T}$) are reported. As an additional metric, we include the accuracy when assessing whether the predicted resulting SMILES exactly match the true SMILES.

Method	Acc. (exact match)	Acc.($T = 1$)	$\overline{T}$
DECIMER (v2.2.0) [24]	0.036	0.05	0.1
DECIMER fine-tuned (v2.2.0) [25]	0.482	0.508	0.643
Img2Mol [5]	0.015	0.015	0.084
MolScribe [23]	0.228	0.269	0.417
ChemGrapher [19]	0.151	0.187	0.286
OSRA[7]	0.044	0.047	0.071
AtomLenz	0.026	0.054	0.064
AtomLenz+EditKT*	0.4	0.484	0.605
ChemExpert(AtomLenz+EditKT*,[23])	0.412	0.5	0.619
ChemExpert(AtomLenz+EditKT*,[25])	0.441	**0.529**	0.65
ChemExpert([25],AtomLenz+EditKT*)	**0.487**	0.518	**0.655**

Table 5. Benchmark results on out of domain ChemPix test set. Both the accuracy, computed by counting the instances where the predicted structures have identical structural ECFP6 descriptors (denoted by a Tanimoto (T) similarity of 1) and the average Tanimoto similarity ($\overline{T}$) are reported. As an additional metric, we include the accuracy when assessing whether the predicted resulting SMILES exactly match the true SMILES.

Method	Acc.($T = 1$) (test set)	$\overline{T}$ (test set)	Acc.($T = 1$) (train set)	$\overline{T}$ (train set)
DECIMER (v2.2.0) [24]	0.001	0.039	0.099	0.142
Img2Mol [5]	0.0	0.0867	0.237	0.388
MolScribe [23]	0.013	0.0865	0.234	0.275
ChemGrapher [19]	0.004	0.067	0.007	0.073
AtomLenz	**0.338**	**0.484**	**0.383**	**0.522**

Table 6. All methods are retrained from scratch on same training dataset (4070 samples of hand-drawn images) to asses data efficiency. Benchmark results on both hand-drawn images train and test set. Both the accuracy, computed by counting the instances where the predicted structures have identical structural ECFP6 descriptors (denoted by a Tanimoto (T) similarity of 1) and the average Tanimoto similarity ($\overline{T}$) are reported.

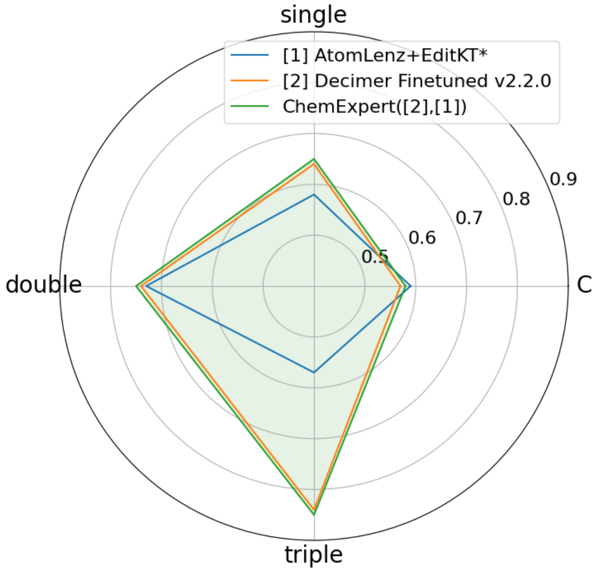


Figure 11. Count accuracies per type over images if type is present in image for ChemPix test set. We observe errors of 'AtomLenz+EditKT*' and 'DECIMER fine-tuned' tend to occur on different samples. Combining both approaches in ChemExpert improves performance.