Supplementary Material

A. Tree Decomposition

Algorithm 2 presents our tree decomposition of molecules. V_1 and V_2 contain non-ring bonds and simple rings respectively. Simple rings are extracted via RDKit's `GetSymmSSSR` function. We then merge rings that share three or more atoms as they form bridged compounds. We note that the junction tree of a molecule is not unique when its cluster graph contains cycles. This introduces additional uncertainty for our probabilistic modeling. To reduce such variation, for any of the three (or more) intersecting bonds, we add their intersecting atom as a cluster and remove the cycle connecting them in the cluster graph. Finally, we construct a junction tree as the maximum spanning tree of a cluster graph $(\mathcal{V}, \mathcal{E})$. Note that we assign an large weight over edges involving clusters in V_0 to ensure no edges in any cycles will be selected into the junction tree.

Algorithm 2 Tree decomposition of molecule $G = (V, E)$

$V_1 \leftarrow$ the set of bonds $(u, v) \in E$ that do not belong to any rings.
$V_2 \leftarrow$ the set of simple rings of G.
for r_1, r_2 **in** V_2 **do**
  Merge rings r_1, r_2 into one ring if they share more than two atoms (bridged rings).
end for
$V_0 \leftarrow$ atoms being the intersection of three or more clusters in $V_1 \cup V_2$.
$\mathcal{V} \leftarrow V_0 \cup V_1 \cup V_2$
$\mathcal{E} \leftarrow \{(i, j, c) \in \mathcal{V} \times \mathcal{V} \times \mathbb{R} \mid |i \cap j| > 0\}$. Set $c = \infty$ if $i \in V_0$ or $j \in V_0$, and $c = 1$ otherwise.
Return The maximum spanning tree over cluster graph $(\mathcal{V}, \mathcal{E})$.

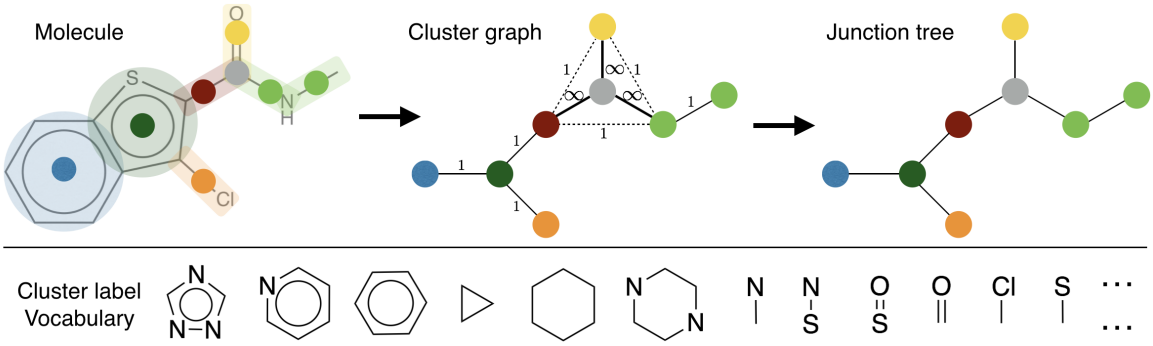

Figure 9. Illustration of tree decomposition and sample of cluster label vocabulary.

B. Stereochemistry

Though usually presented as two-dimensional graphs, molecules are three-dimensional objects, i.e. molecules are defined not only by its atom types and bond connections, but also the spatial configuration between atoms (chiral atoms and cis-trans isomerism). *Stereoisomers* are molecules that have the same 2D structure, but differ in the 3D orientations of their atoms in space. We note that stereochemical feasibility could not be simply encoded as context free or attribute grammars.

Empirically, we found it more efficient to predict the stereochemical configuration separately from the molecule generation. Specifically, the JT-VAE first generates the 2D structure of a molecule m, following the same procedure described in section 2. Then we generate all its stereoisomers $\mathcal{S}_m$ using RDKit's `EnumerateStereoisomers` function, which identifies atoms that could be chiral. For each isomer $m' \in \mathcal{S}_m$, we encode its graph representation $\mathbf{h}_{m'}$ with the graph encoder and compute their cosine similarity $f^s(m') = \cos(\mathbf{h}_{m'}, \mathbf{z}_m)$ (note that $\mathbf{z}_m$ is stochastic). We reconstruct the

final 3D structure by picking the stereoisomer $\widehat{m} = \arg\max_{m'} f^s(m')$. Since on average only few atoms could have stereochemical variations, this post ranking process is very efficient. Combining this with tree and graph generation, the molecule reconstruction loss $\mathcal{L}$ becomes

$$\mathcal{L} = \mathcal{L}_c + \mathcal{L}_g + \mathcal{L}_s; \qquad \mathcal{L}_s = f^s(m) - \log \sum_{m' \in \mathcal{S}_m} \exp(f^s(m')) \tag{17}$$

C. Training Details

By applying tree decomposition over 240K molecules in ZINC dataset, we collected our vocabulary set $\mathcal{X}$ of size $|\mathcal{X}| = 780$. The hidden state dimension is 450 for all modules in JT-VAE and the latent bottleneck dimension is 56. For the graph encoder, the initial atom features include its atom type, degree, its formal charge and its chiral configuration. Bond feature is a concatenation of its bond type, whether the bond is in a ring, and its cis-trans configuration. For our tree encoder, we represent each cluster with a neural embedding vector, similar to word embedding for words. The tree and graph decoder use the same feature setting as encoders. The graph encoder and decoder runs three iterations of neural message passing. For fair comparison to SMILES based method, we minimized feature engineering. We use PyTorch to implement all neural components and RDKit to process molecules.

D. More Experimental Results

Sampled Molecules Note that a degenerate model could also achieve 100% prior validity by keep generating simple structures like chains. To prove that our model does not converge to such trivial solutions, we randomly sample and plot 250 molecules from prior distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. As shown in Figure 10, our sampled molecules present rich variety and structural complexity. This demonstrates the soundness of the prior validity improvement of our model.

Neighborhood Visualization Given a molecule, we follow Kusner et al. (2017) to construct a grid visualization of its neighborhood. Specifically, we encode a molecule into the latent space and generate two random orthogonal unit vectors as two axis of a grid. Moving in combinations of these directions yields a set of latent vectors and we decode them into corresponding molecules. In Figure 11 and 12, we visualize the local neighborhood of two molecules presented in Dai et al. (2018). Figure 11 visualizes the same molecule in Figure 6, but with wider neighborhood ranges.

Bayesian Optimization We directly used open sourced implementation in Kusner et al. (2017) for Bayesian optimization (BO). Specifically, we train a sparse Gaussian process with 500 inducing points to predict properties of molecules. Five iterations of batch BO with expected improvement heuristic is used to propose new latent vectors. In each iteration, 50 latent vectors are proposed, from which molecules are decoded and added to the training set for next iteration. We perform 10 independent runs and aggregate results. In Figure 13, we present the top 50 molecules found among 10 runs using JT-VAE. Following Kusner et al.'s implementation, the scores reported are **normalized** to zero mean and unit variance by the mean and variance computed from training set.

Constrained Optimization For this task, a property predictor F is trained jointly with VAE to predict $y(m) = logP(m) - SA(m)$ from the latent embedding of m. F is a feed-forward network with one hidden layer of dimension 450 followed by $\tanh$ activation. To optimize a molecule m, we start with its mean encoding $\mathbf{z}_m^0 = \boldsymbol{\mu}_m$ and apply 80 gradient ascent steps: $\mathbf{z}_m^t = \mathbf{z}_m^{t-1} + \alpha \frac{\partial y}{\partial z}$ with $\alpha = 2.0$. 80 molecules are decoded from latent vectors $\{\mathbf{z}_m^i\}$ and their property is calculated. Molecular similarity $sim(m, m')$ is calculated via Morgan fingerprint of radius 2 with Tanimoto similarity. For each molecule m, we report the best modified molecule m' with $sim(m, m') > \delta$ for some threshold δ. In Figure 14, we present three groups of modification examples with $\delta = 0.2, 0.4, 0.6$. For each group, we present top three pairs that leads to best improvement $y(m') - y(m)$ as well as one pair decreased property ($y(m') < y(m)$). This is caused by inaccurate property prediction. From Figure 14, we can see that tighter similarity constraint forces the model to preserve the original structure.

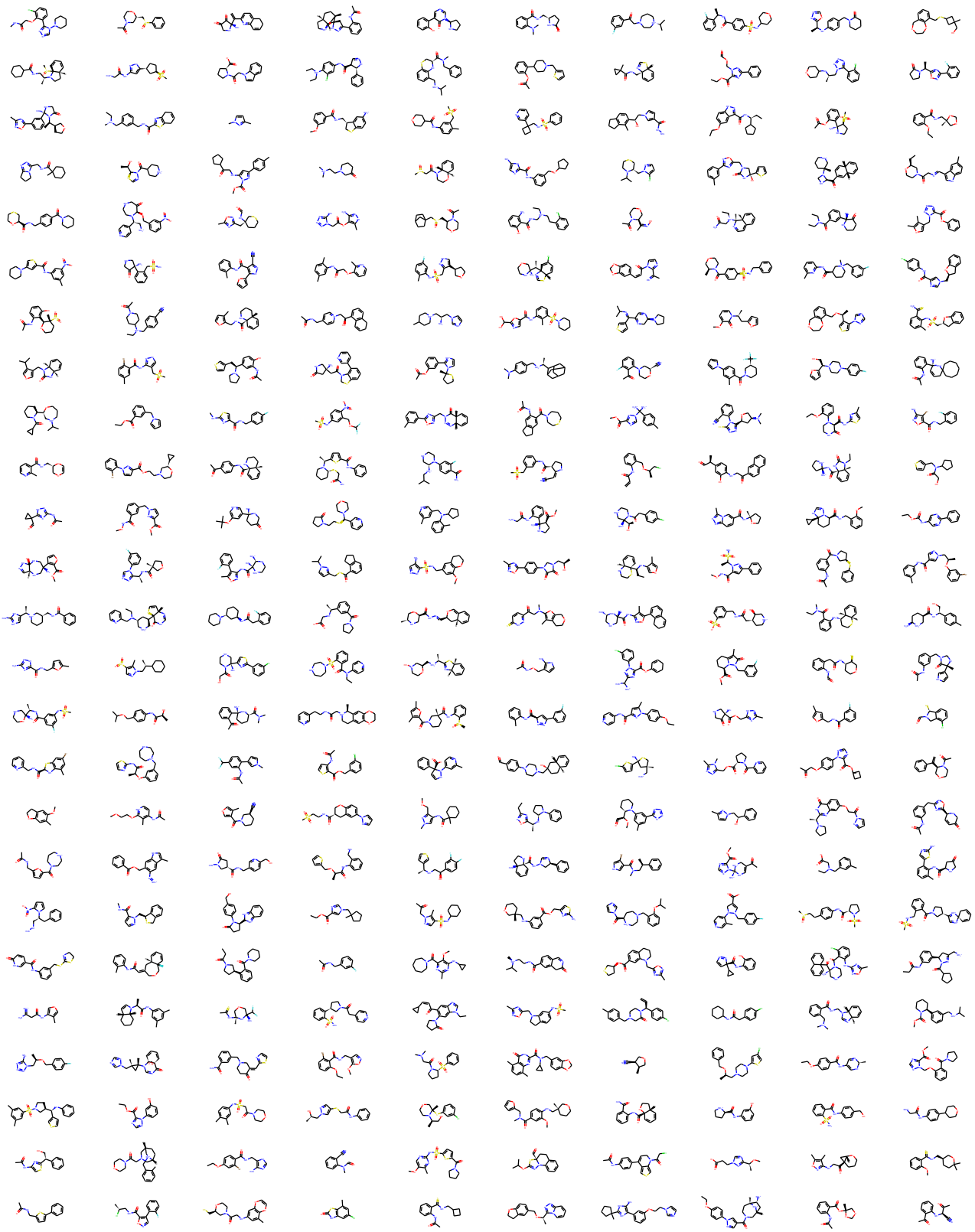

Figure 10. 250 molecules sampled from prior distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

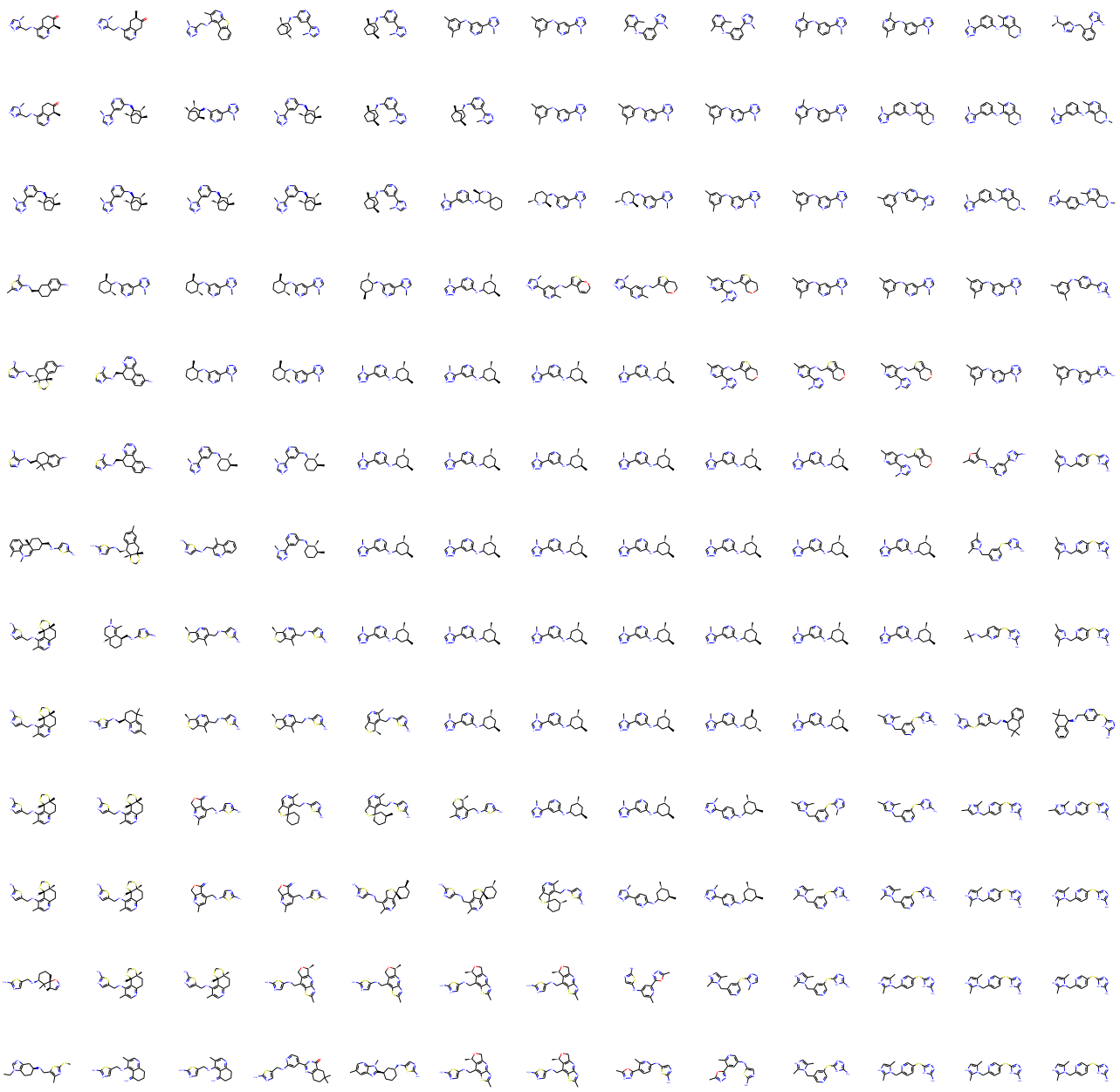

Figure 11. Neighborhood visualization of molecule C[C@H]1CC(Nc2cncc(-c3nncn3C)c2)C[C@H](C)C1.

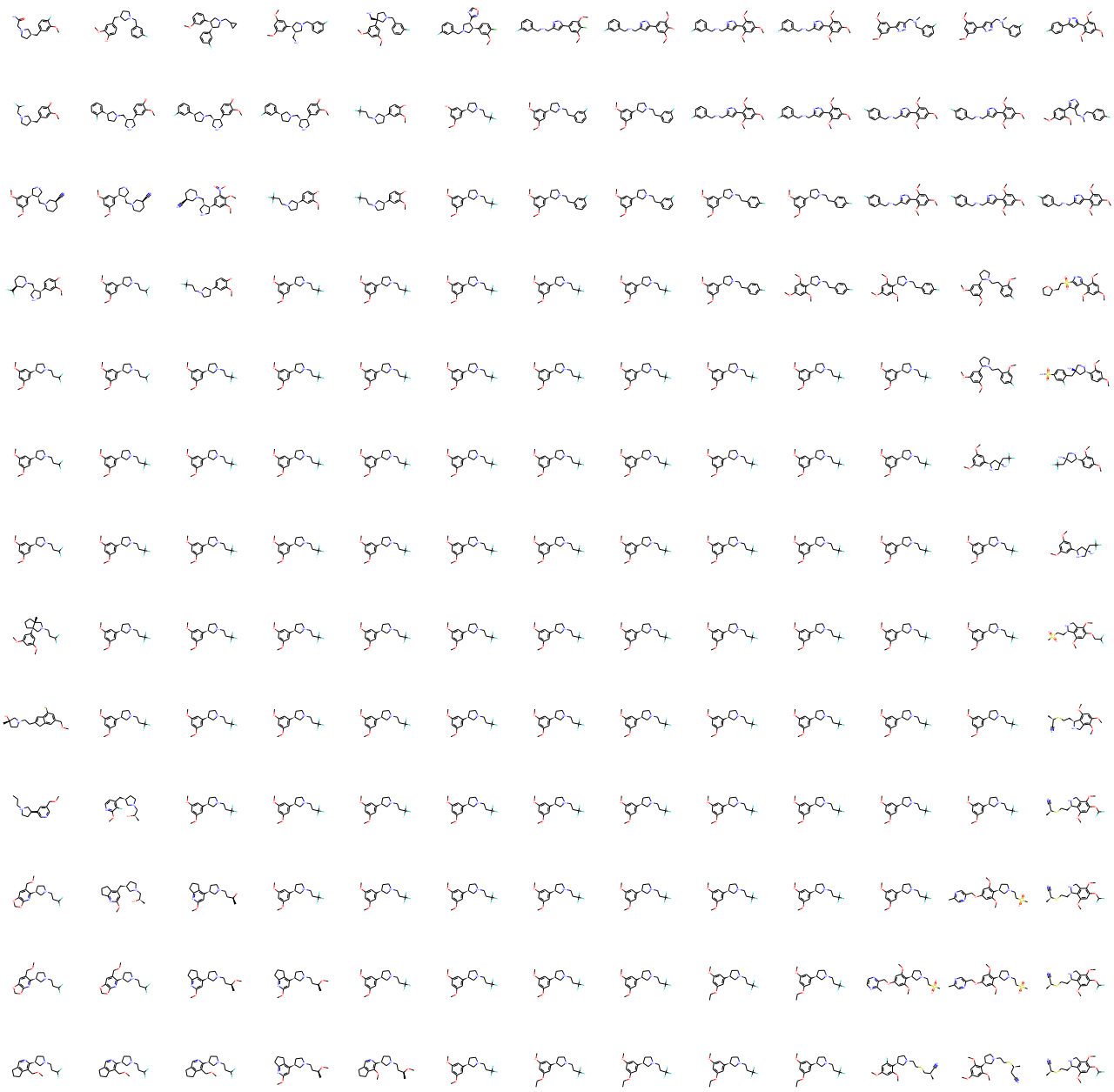

Figure 12. Neighborhood visualization of molecule COc1cc(OC)cc([C@H]2CC[NH+](CCC(F)(F)F)C2)c1.

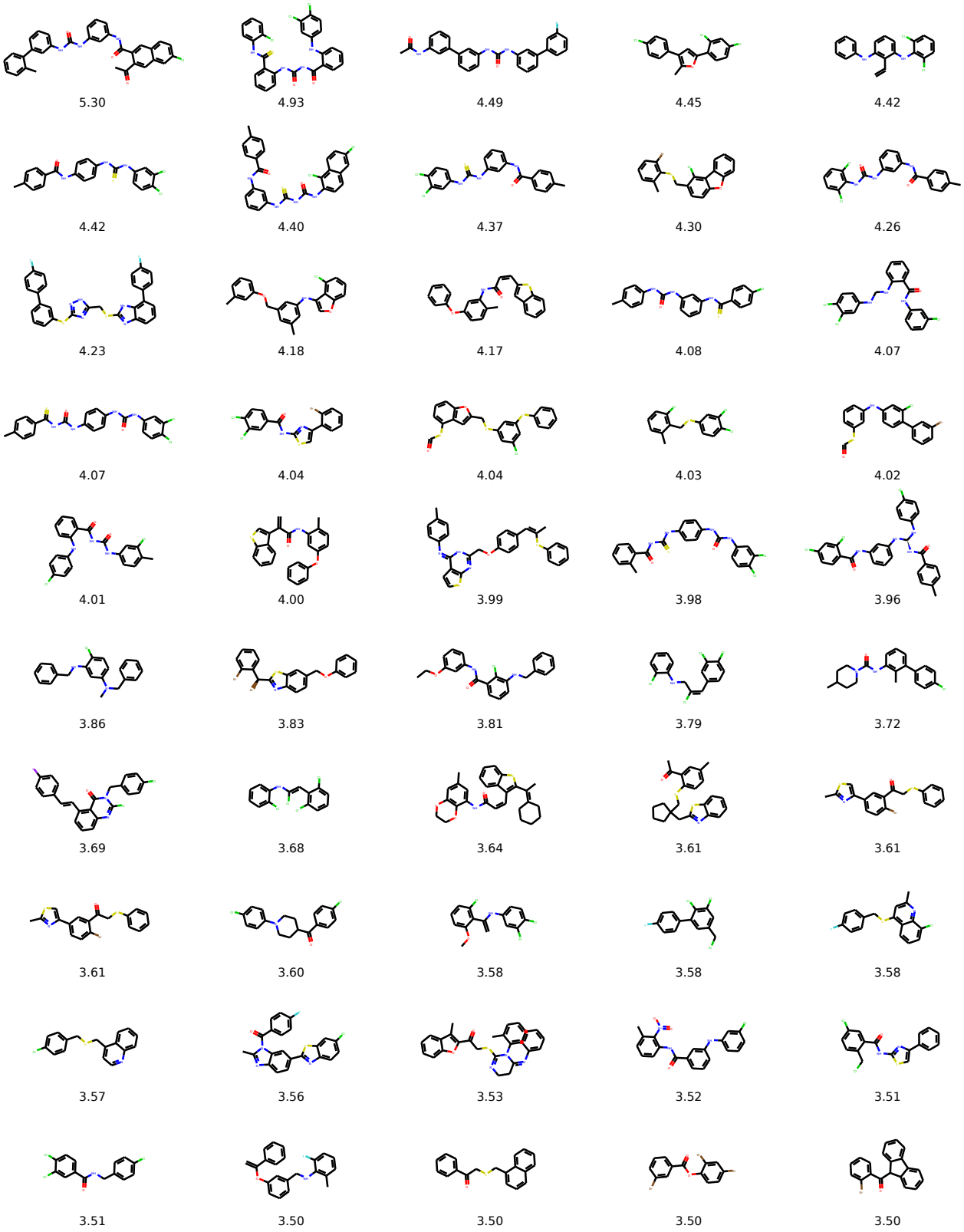

Figure 13. Top 50 molecules found by Bayesian optimization using JT-VAE.

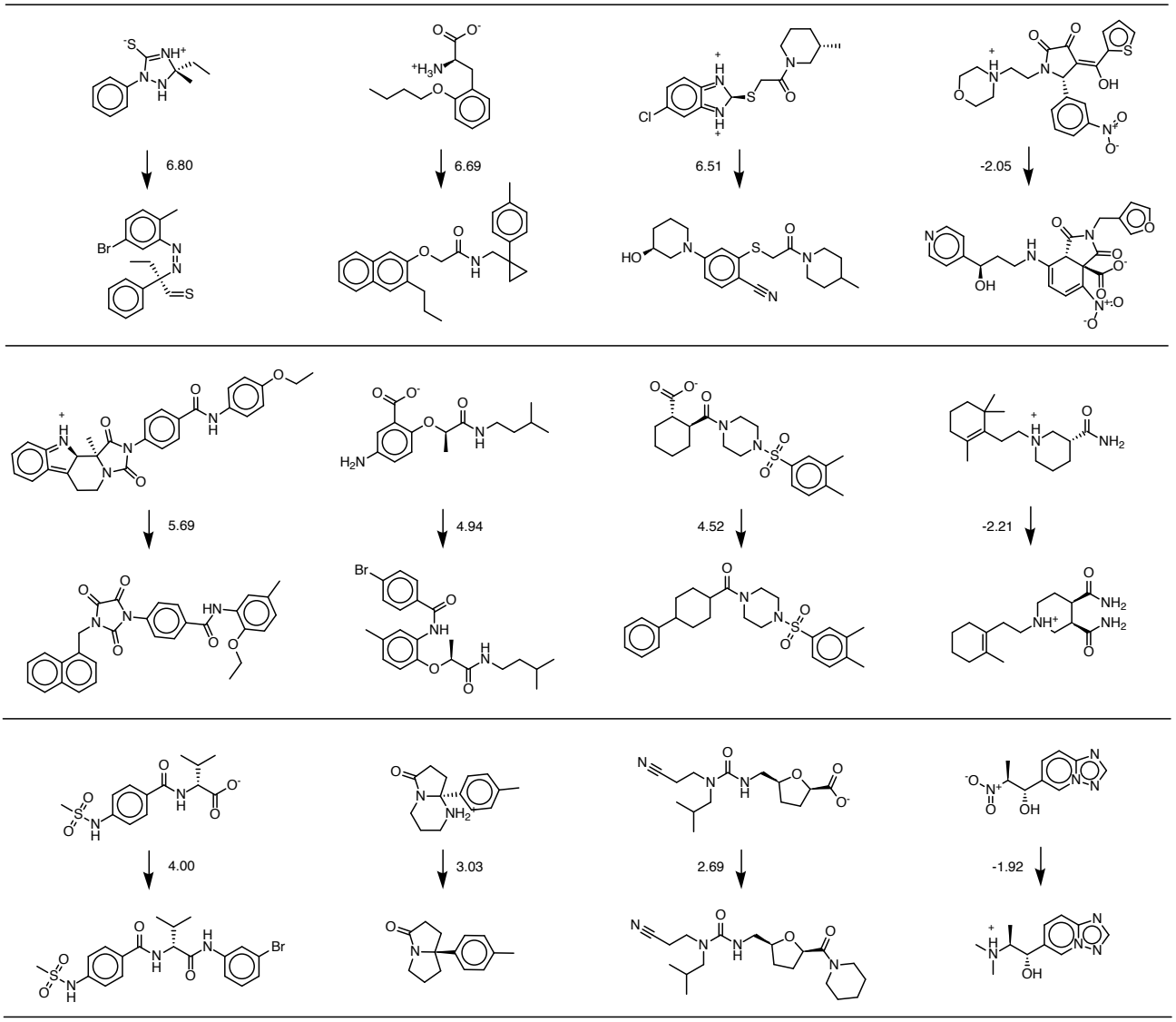

Figure 14. Row 1-3: Molecule modification results with similarity constraint $sim(m, m') \geq 0.2, 0.4, 0.6$. For each group, we plot the top three pairs that leads to actual property improvement, and one pair with decreased property. We can see that tighter similarity constraint forces the model to preserve the original structure.