Junction Tree Variational Autoencoder for Molecular Graph Generation

Wengong Jin, Regina Barzilay and Tommi Jaakkola

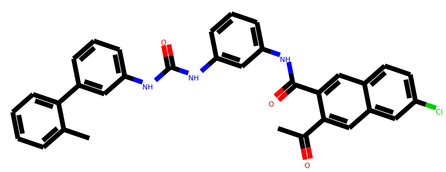
Presentation adapted from slides by: Wengong Jin
Presenter: Yevgeny Tkach

2019 Spring @
https://qdata.github.io/deep2Read/

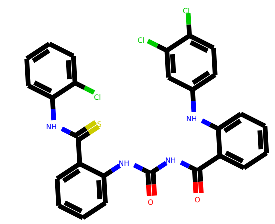
Executive Summary

- Molecule generation using VAE. Encoding and decoding is based on spacial graph message passing algorithm.
- Instead of generating the molecule node by node which can be looked at as “character level” generation, this work builds higher level vocabulary based on tree decomposition of the molecule graph.
- Using proper “words/parts of speech” helps to make sure that the final molecule is valid.

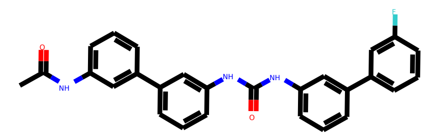
Drug Discovery



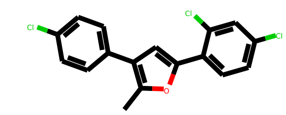
5.30



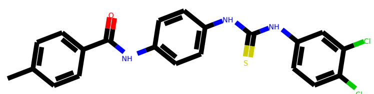
4.93



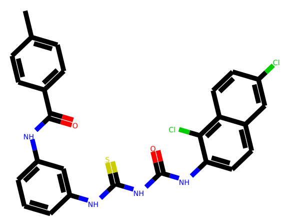
4.49



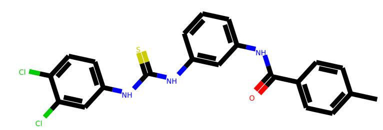
4.45



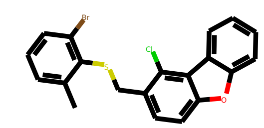
4.42



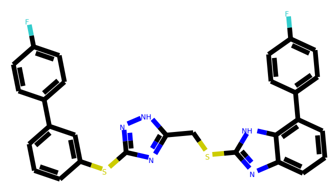
4.40



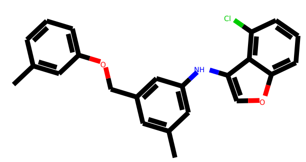
4.37



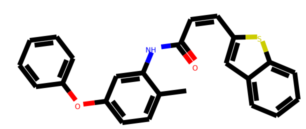
4.30



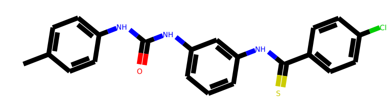
4.23



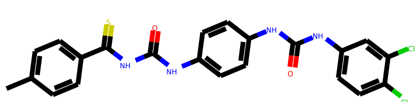
4.18



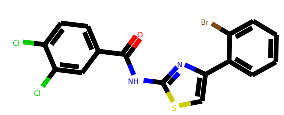
4.17



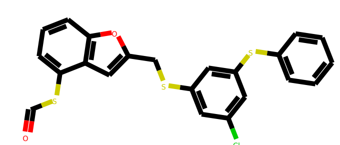
4.08



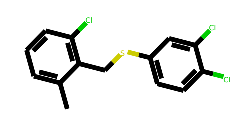
4.07



4.04



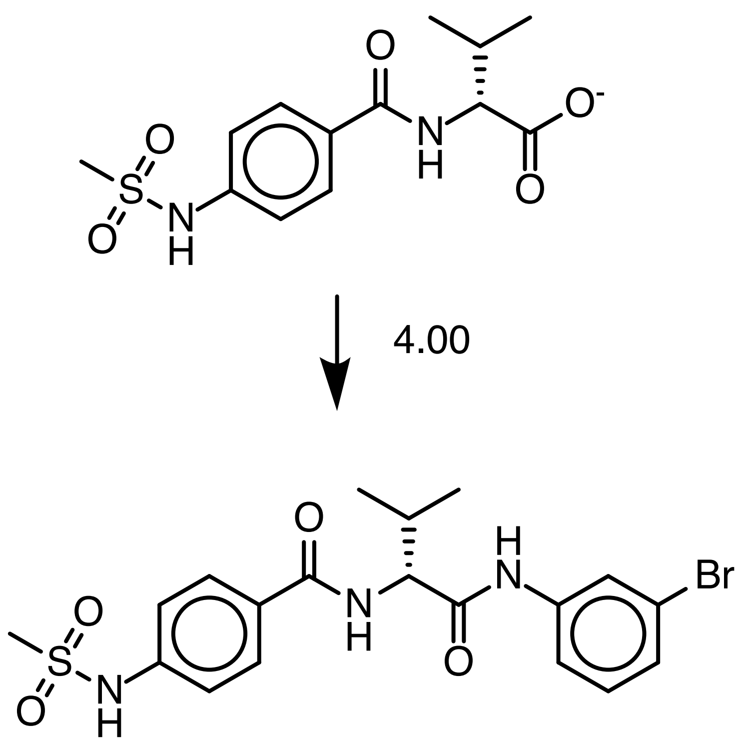
4.04

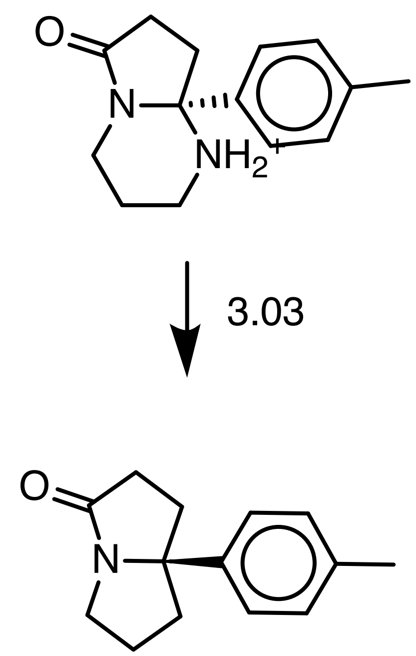


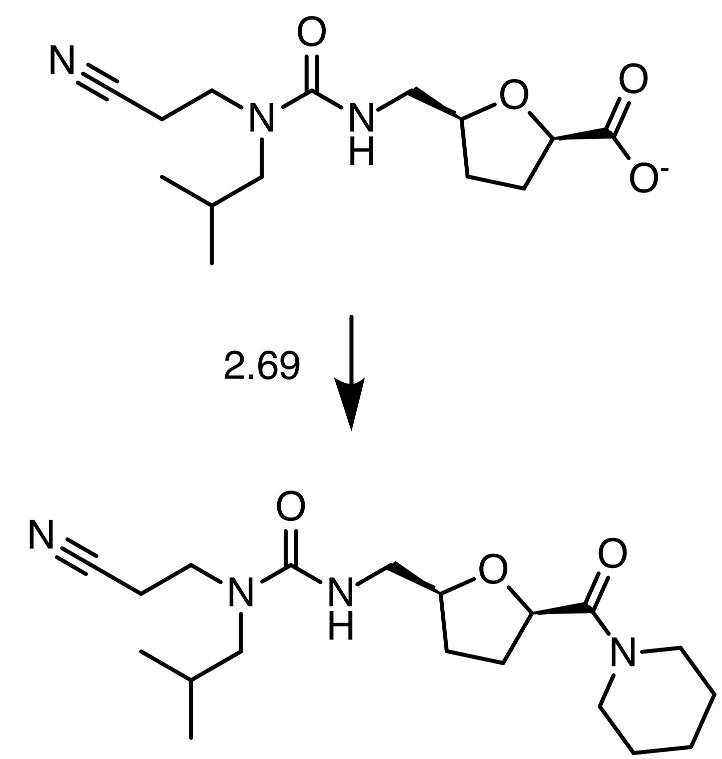
4.03

Generate molecules with high potency

Drug Discovery

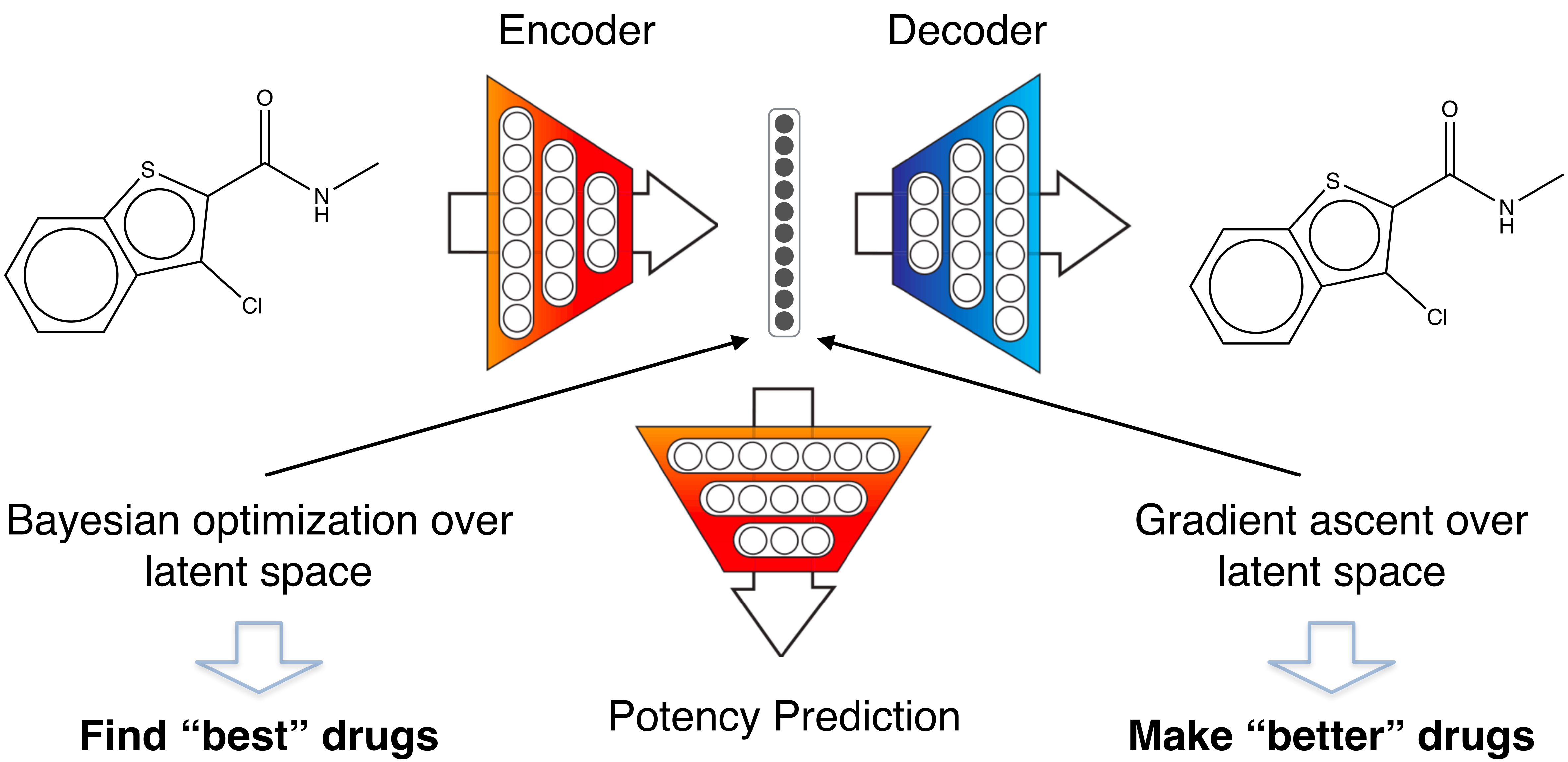






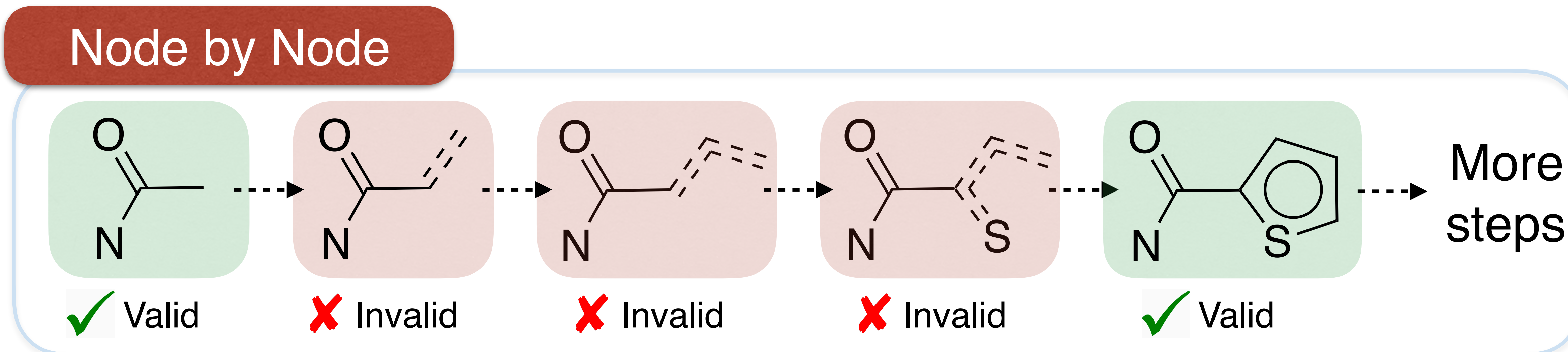
Modify molecules to increase potency

Molecular Variational Autoencoder



[1] Gomez-Bombarelli et al.,Automatic chemical design using a data-driven continuous representation of molecules, 2016

How to generate graphs?

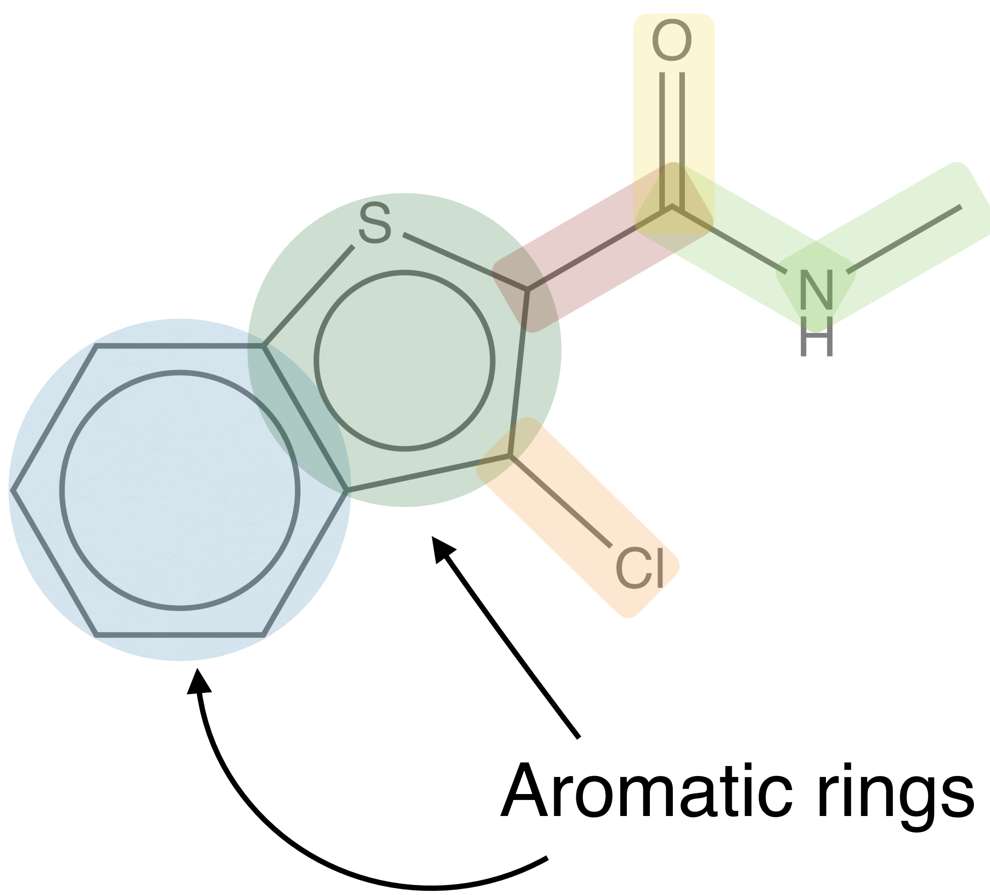


- Not every graphs is chemically valid
- Invalid intermediate states ⟶ hard to validate
- Very long intermediate steps ⟶ difficult to train (Li et al., 2018)

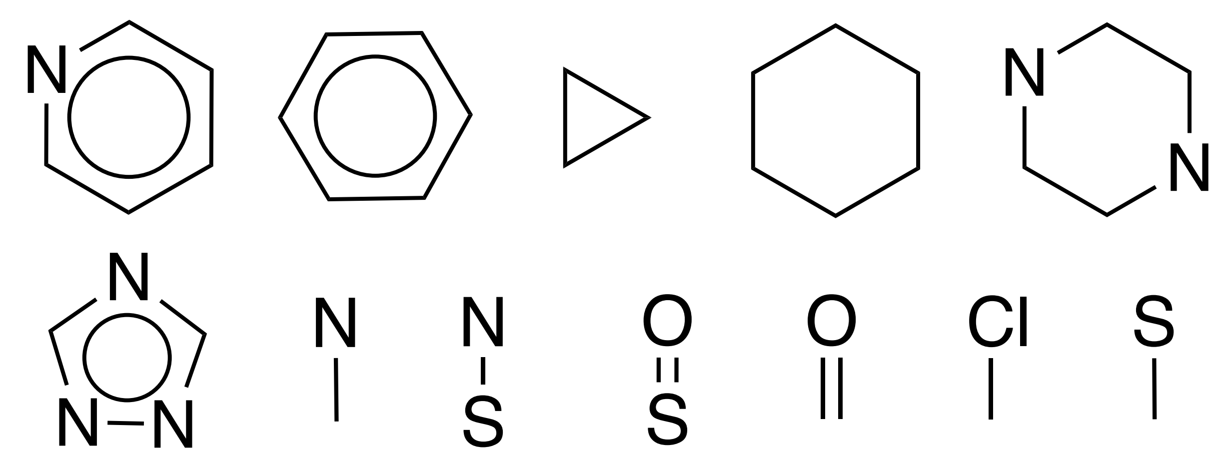
---

[2] Li et al., Learning Deep Generative Models of Graphs, 2018

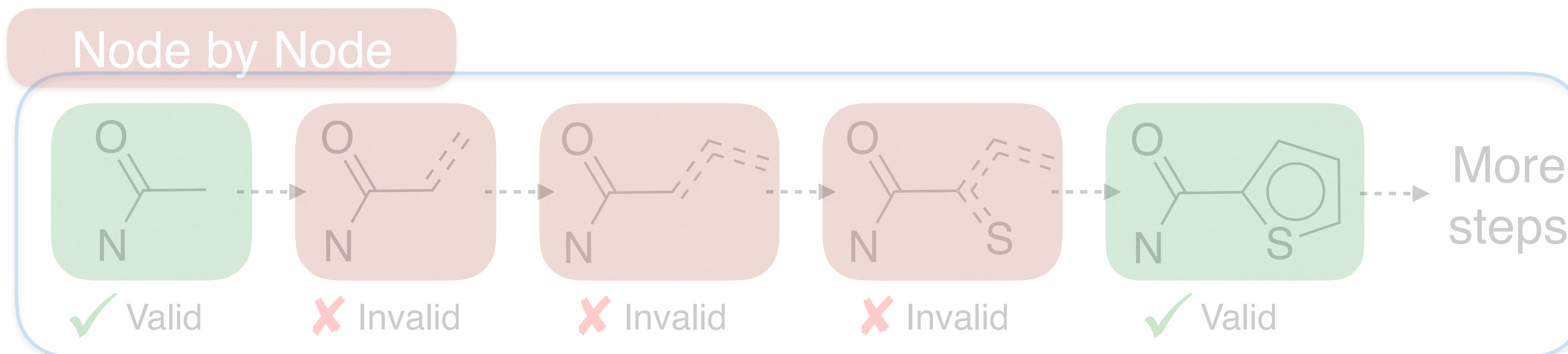
Functional Group

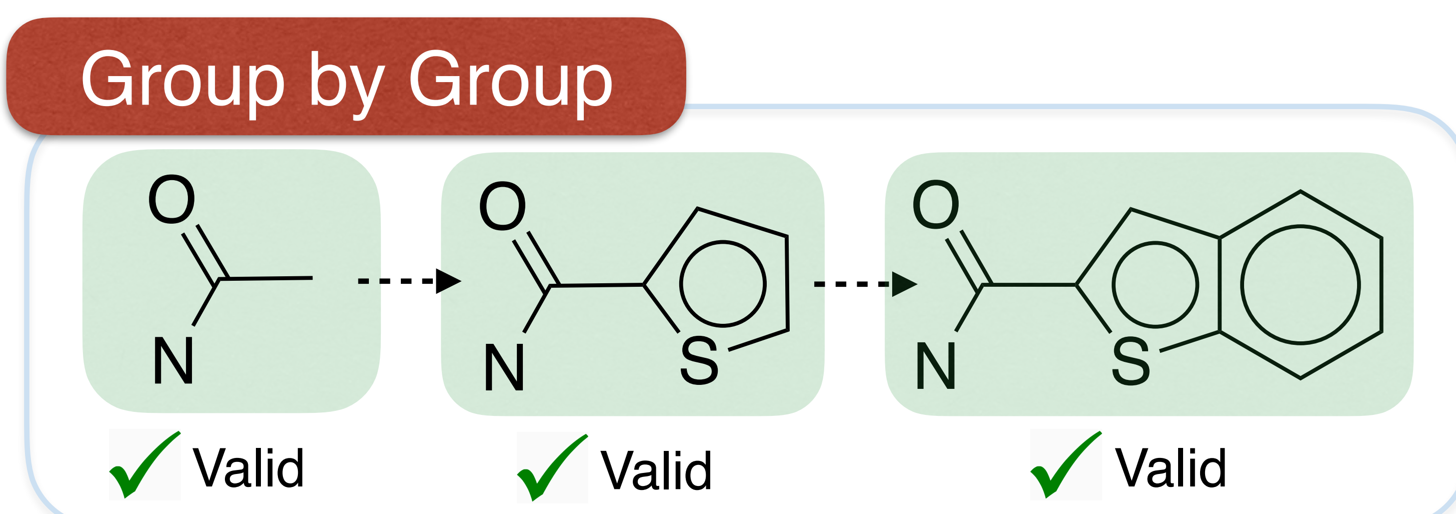


Functional Groups



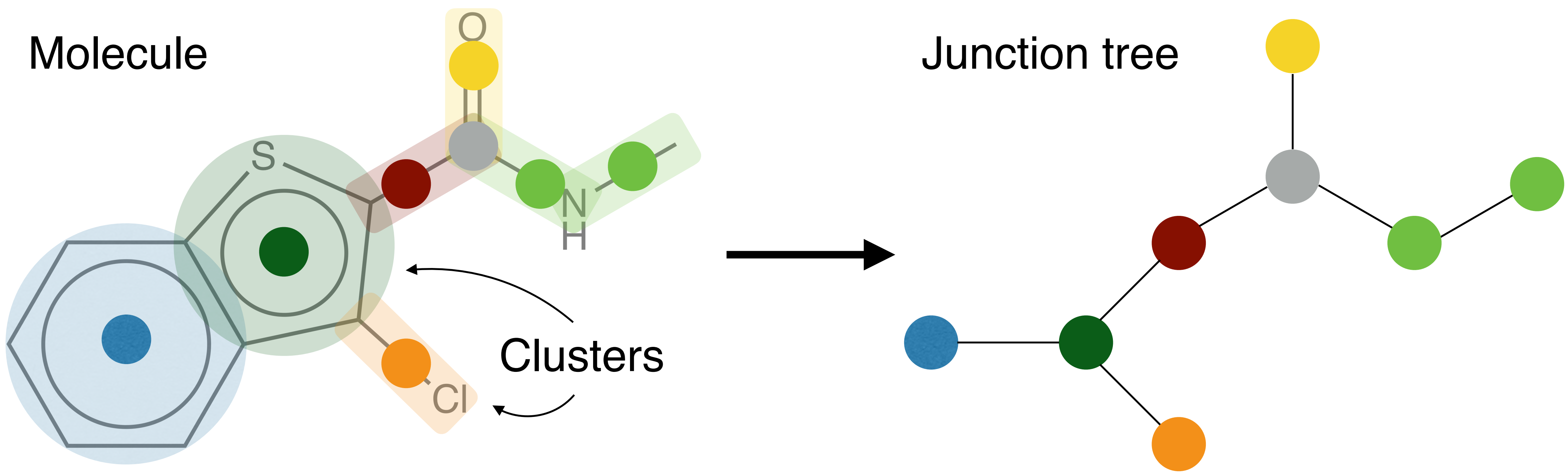
How to generate graphs?



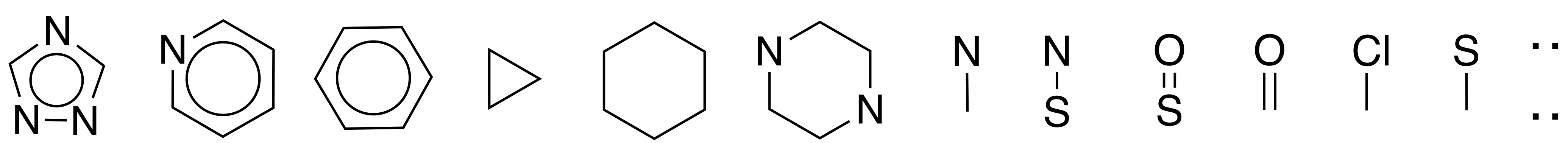


- Shorter action sequence
- Easy to check validity

Tree Decomposition

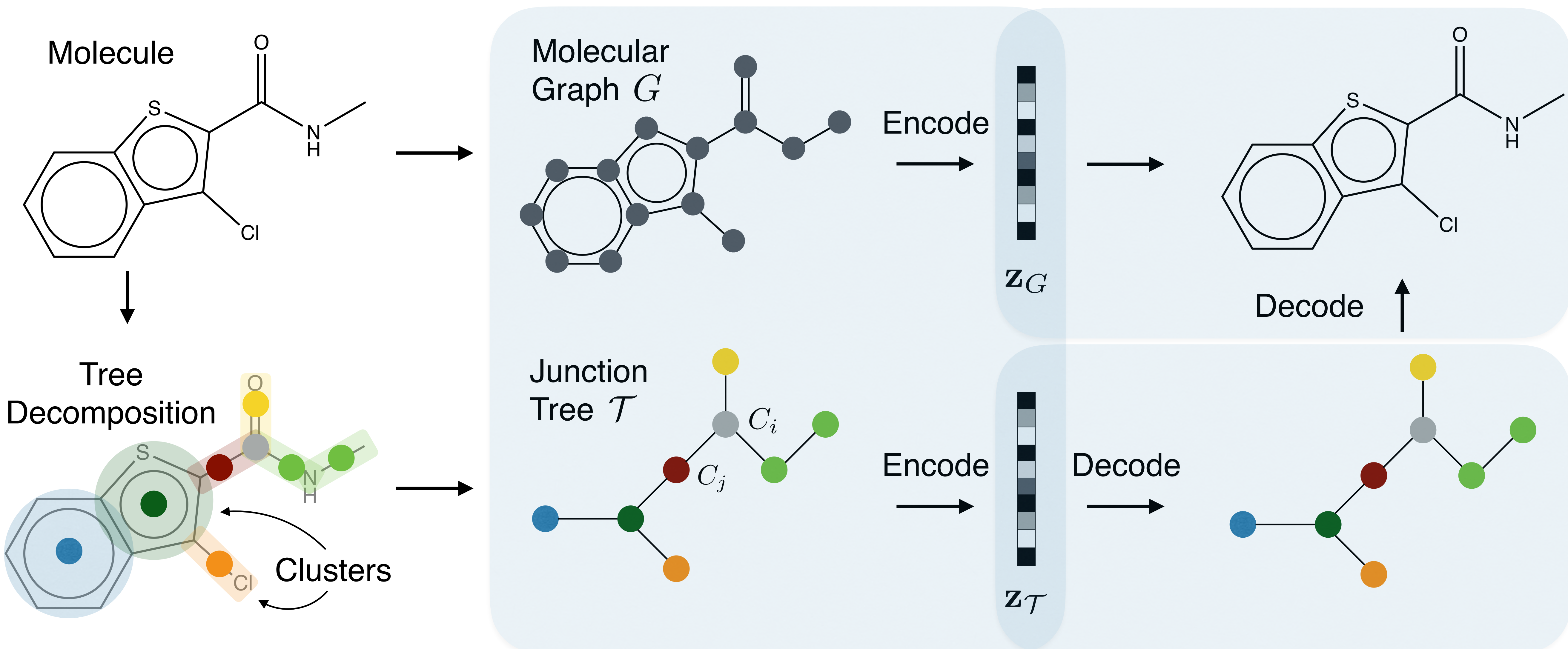


Cluster label
Vocabulary

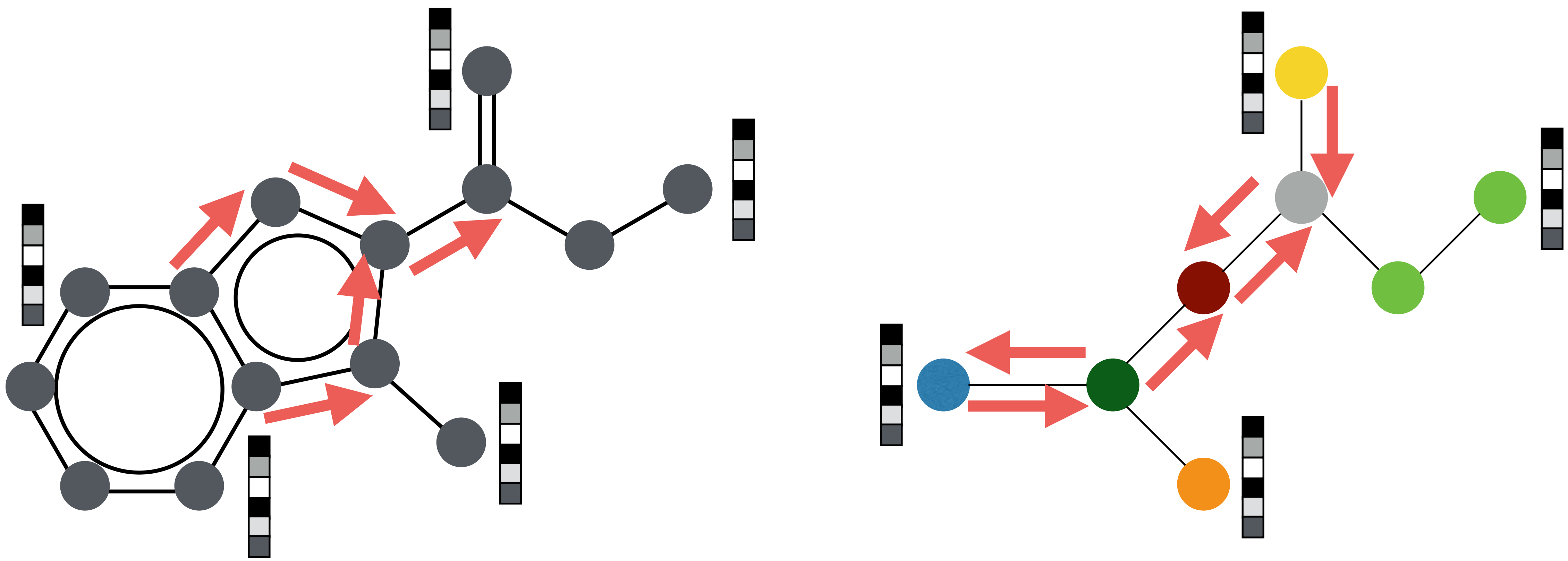


- Generate junction tree ⟶ Generate graph group by group
- Vocabulary size: less than 800 given 250K molecules

Our Approach

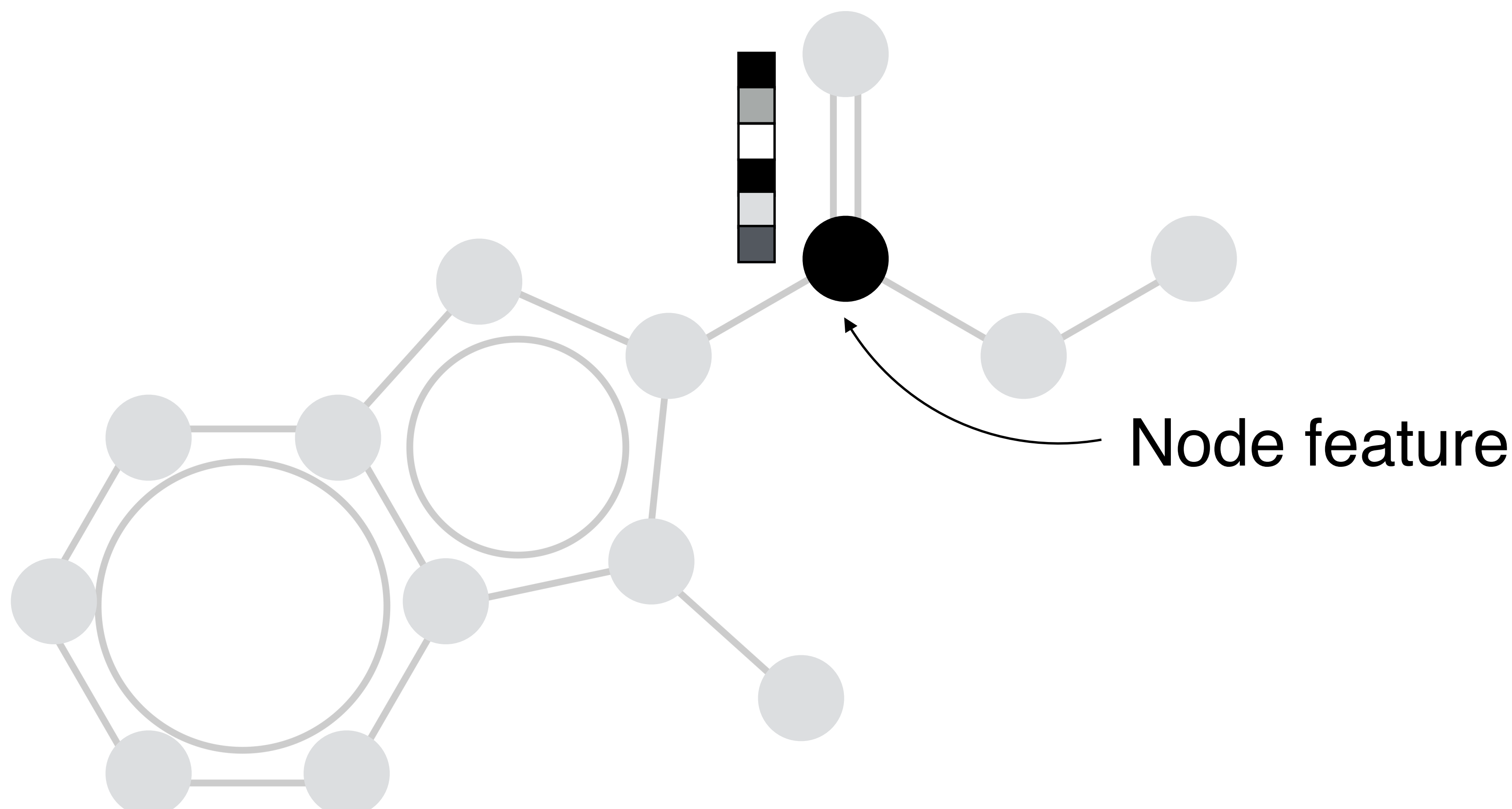


Graph & Tree Encoder



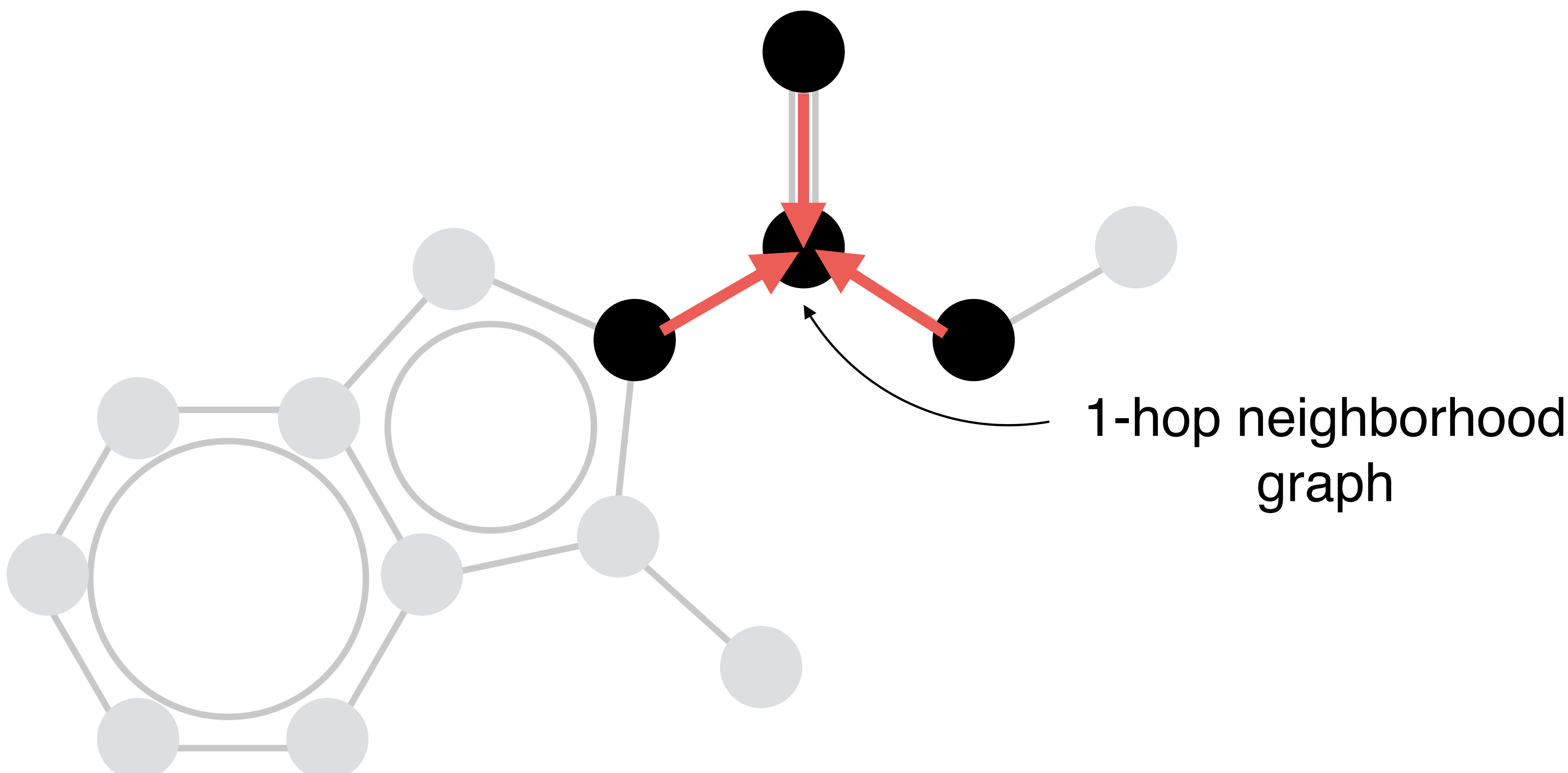
Neural Message Passing Network (MPN)

Graph Encoding



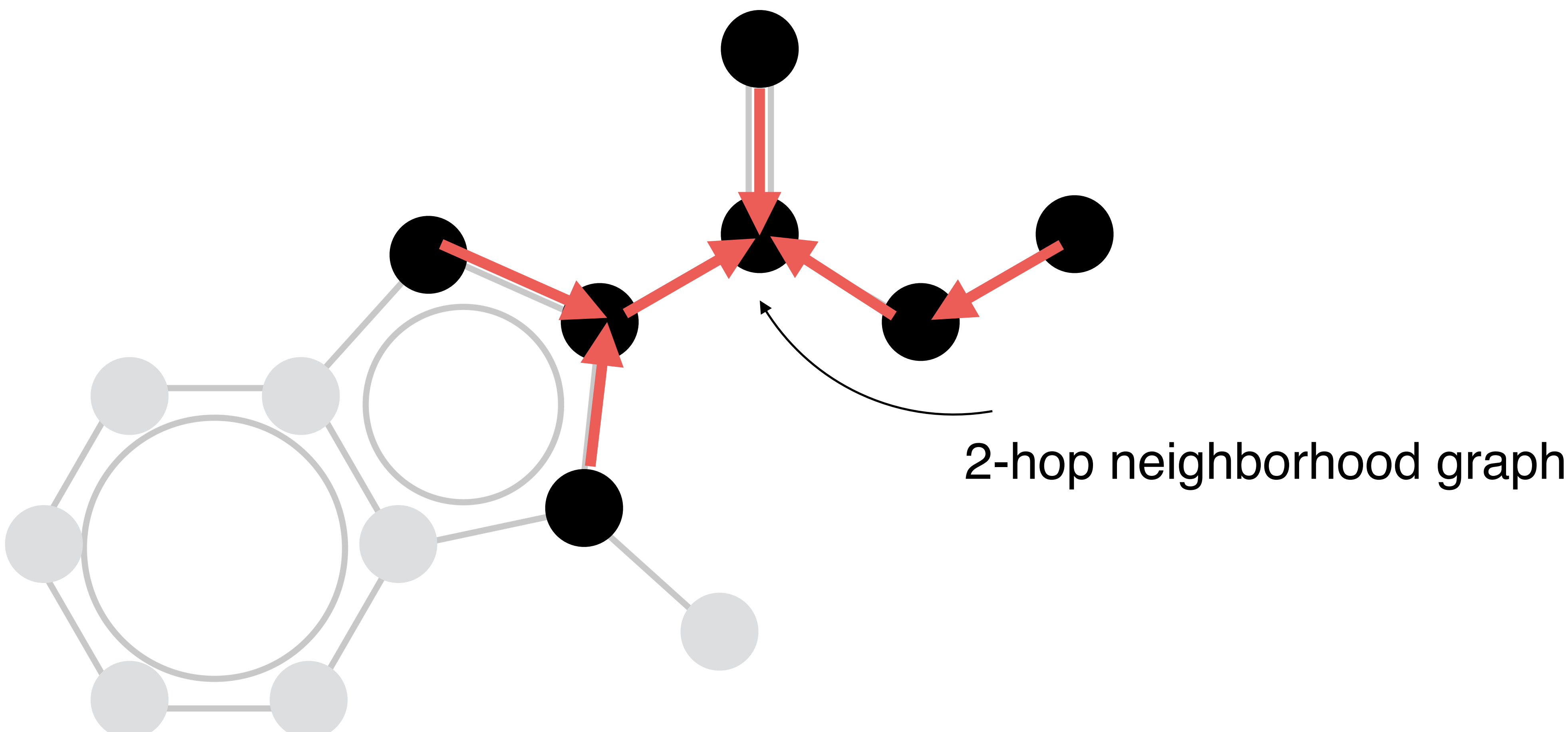
[3] Dai et al., Discriminative embeddings of latent variable models for structured data, 2016

Graph Encoding



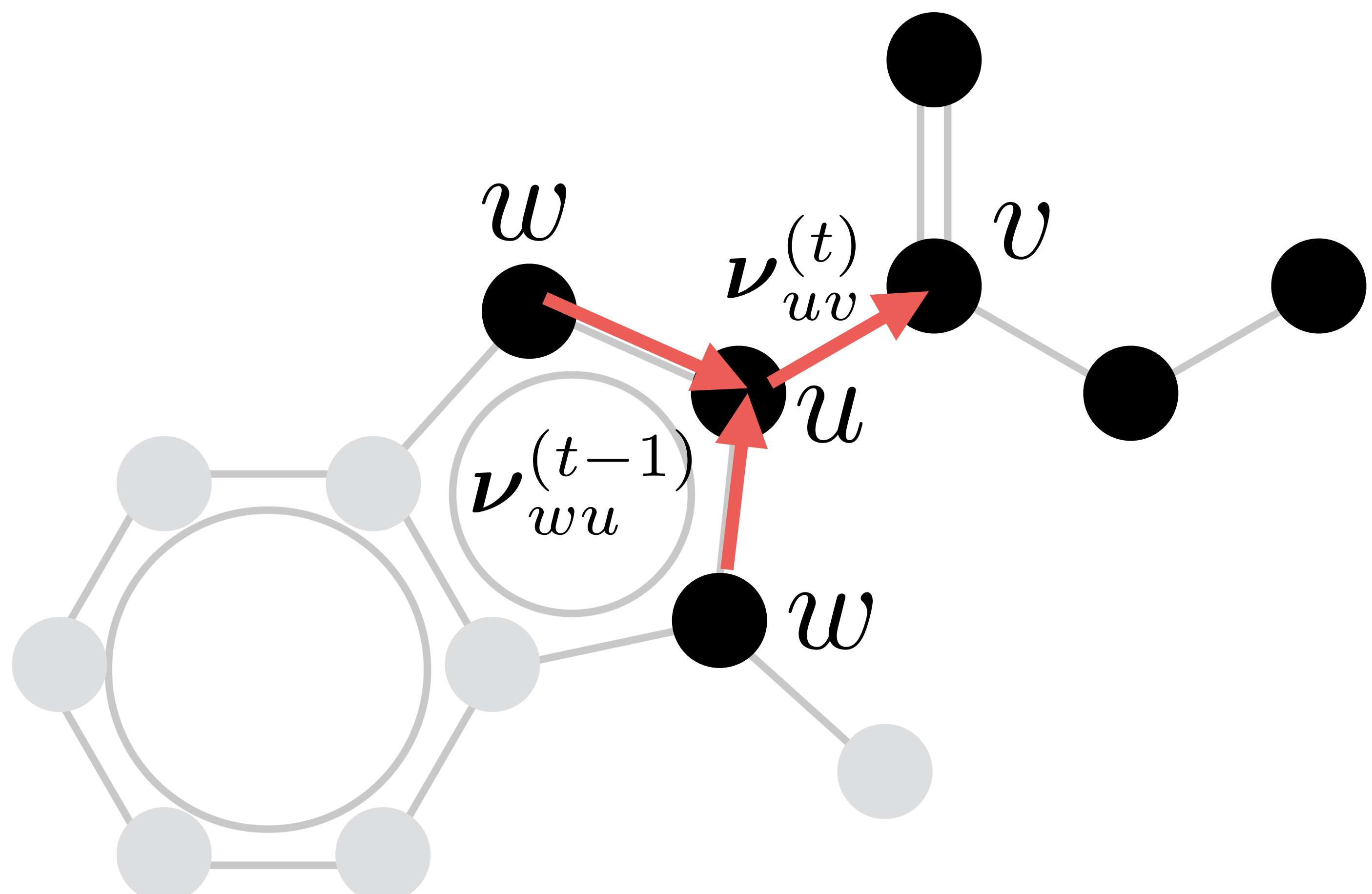
[3] Dai et al., Discriminative embeddings of latent variable models for structured data, 2016

Graph Encoding



[3] Dai et al., Discriminative embeddings of latent variable models for structured data, 2016

Graph Encoding

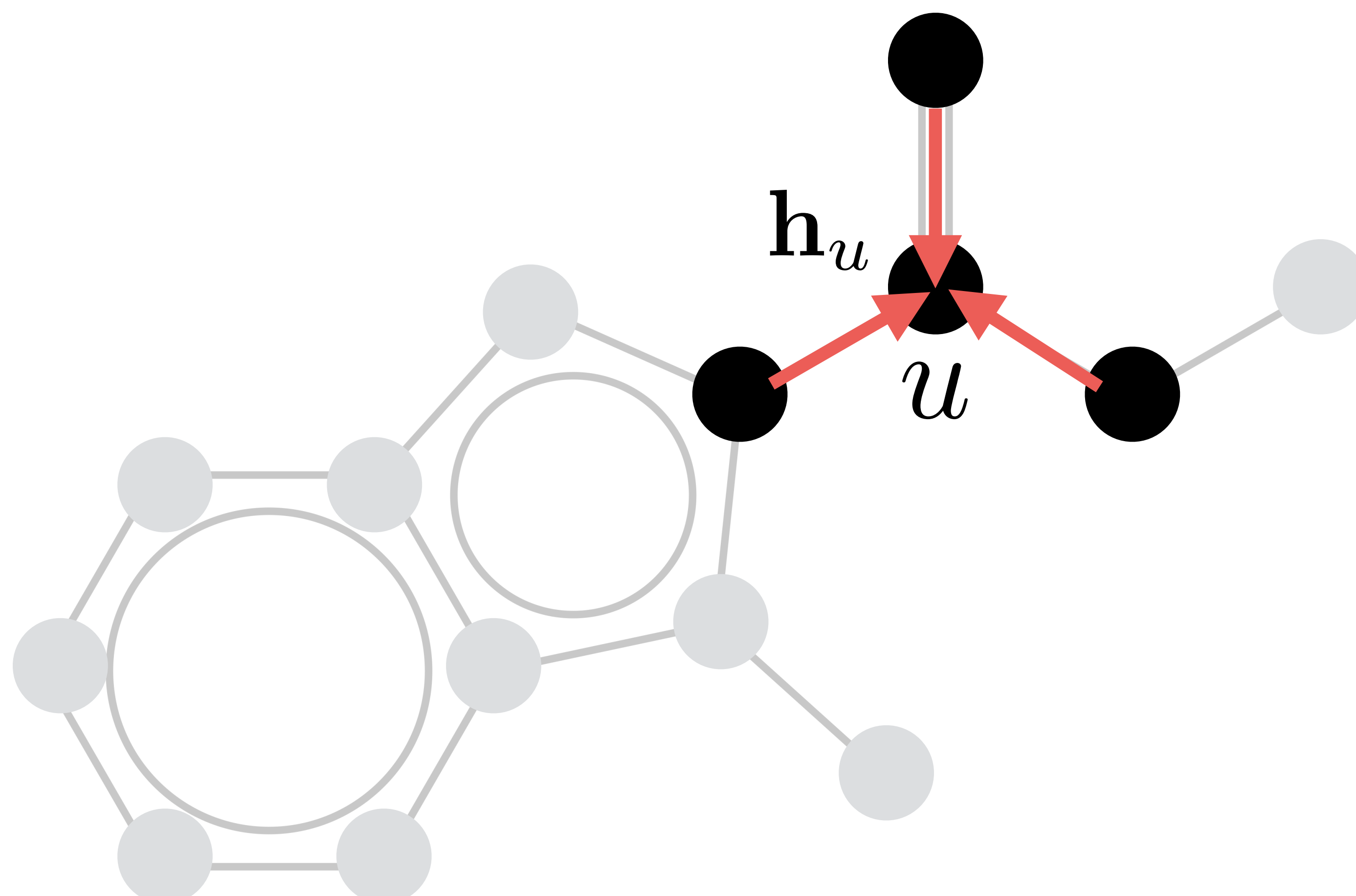


$$\boldsymbol{\nu}_{uv}^{(t)} = \tau(\mathbf{W}_1^g \mathbf{x}_u + \mathbf{W}_2^g \mathbf{x}_{uv} + \mathbf{W}_3^g \sum_{\boxed{w \in N(u) \setminus v}} \boldsymbol{\nu}_{wu}^{(t-1)})$$

Messages Node feature Edge feature

[3] Dai et al., Discriminative embeddings of latent variable models for structured data, 2016

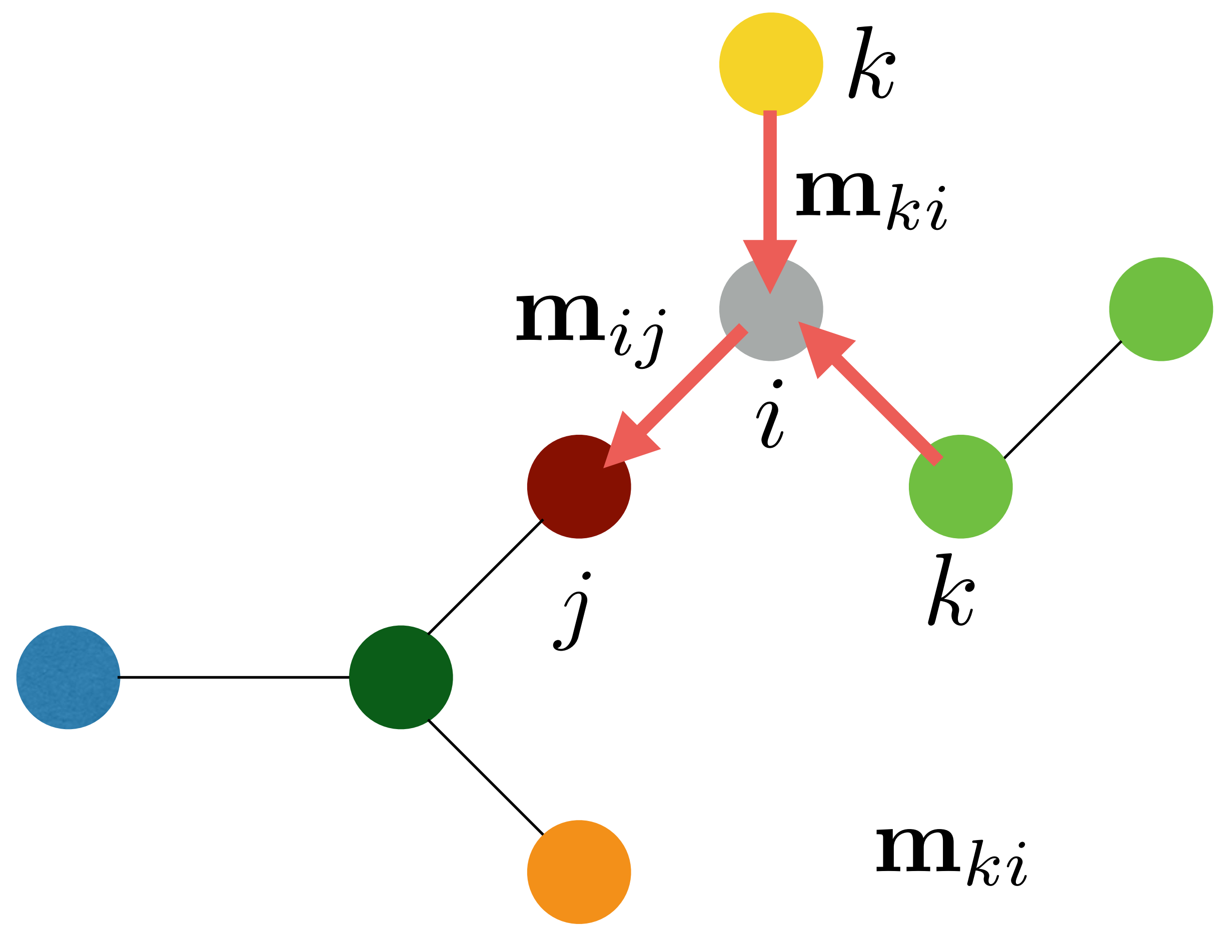
Graph Encoding



$$\mathbf{h}_u = \tau(\mathbf{U}_1^g \mathbf{x}_u + \sum\nolimits_{v \in N(u)} \mathbf{U}_2^g \boldsymbol{\nu}_{vu}^{(T)})$$

[3] Dai et al., Discriminative embeddings of latent variable models for structured data, 2016

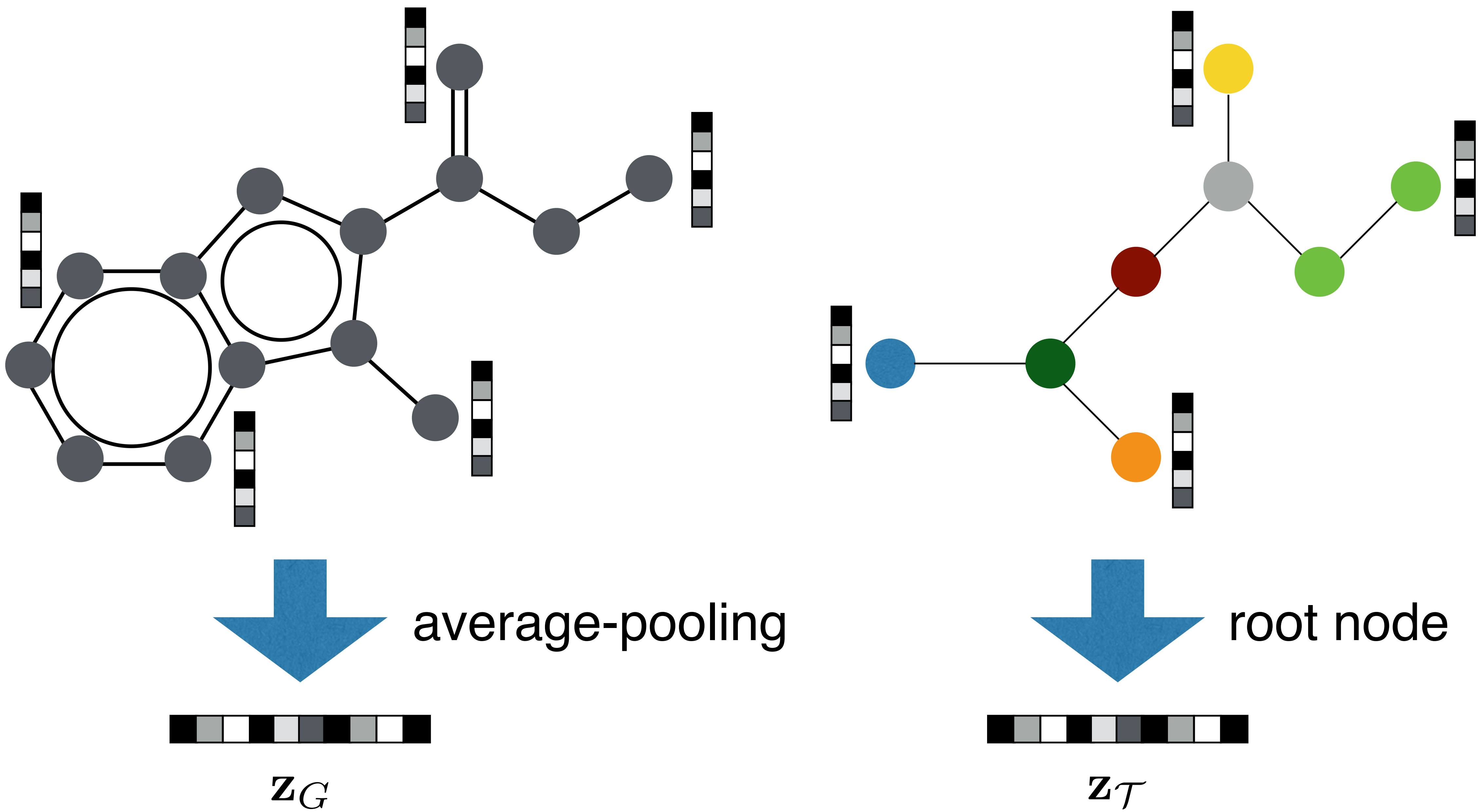
Tree Encoding



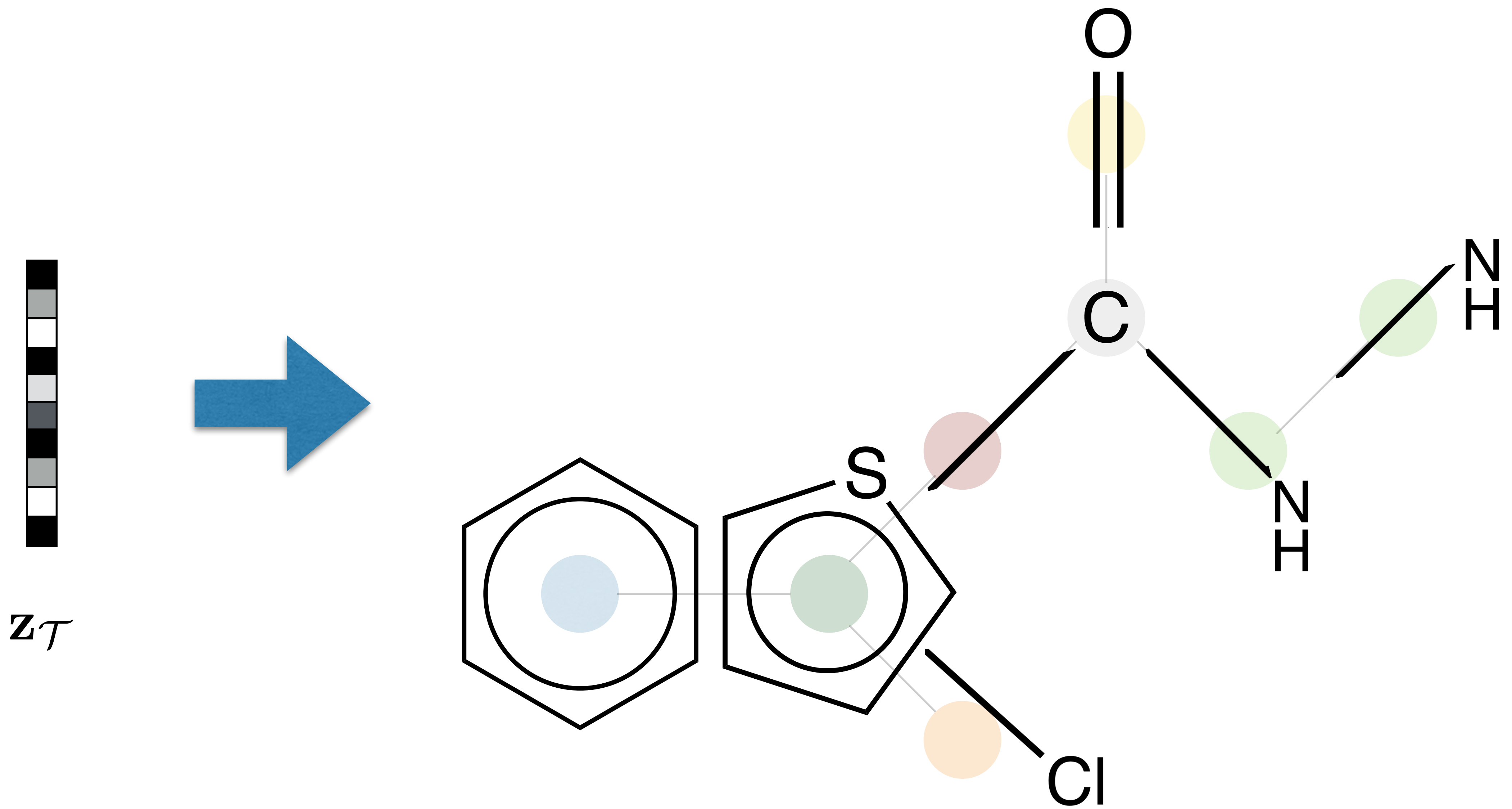
$$\mathbf{m}_{ij} = \mathrm{GRU}(\mathbf{x}_i, \{\mathbf{m}_{ki}\}_{k \in N(i) \setminus j})$$

To capture long range interactions

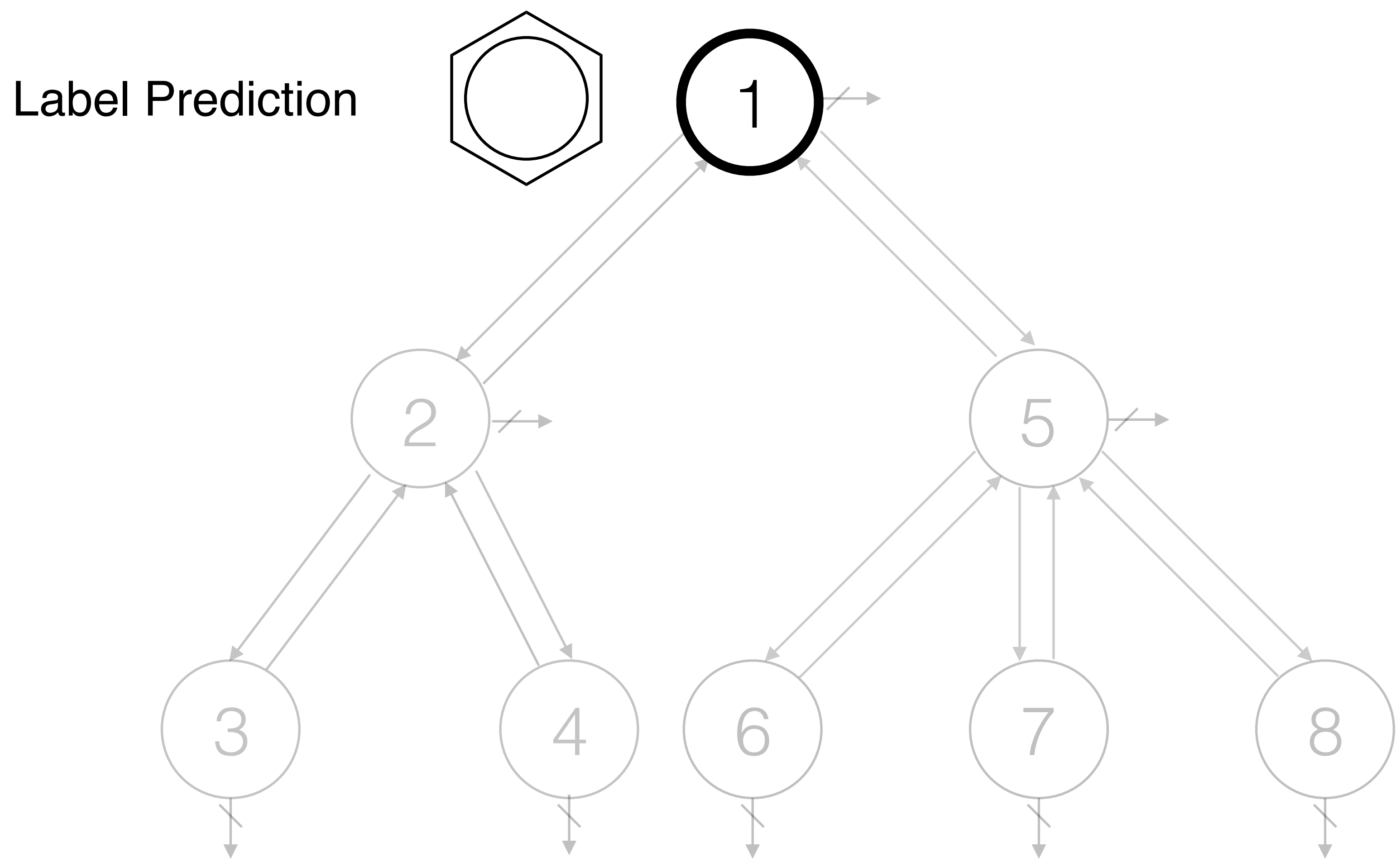
Graph & Tree Encoder



Tree Decoder

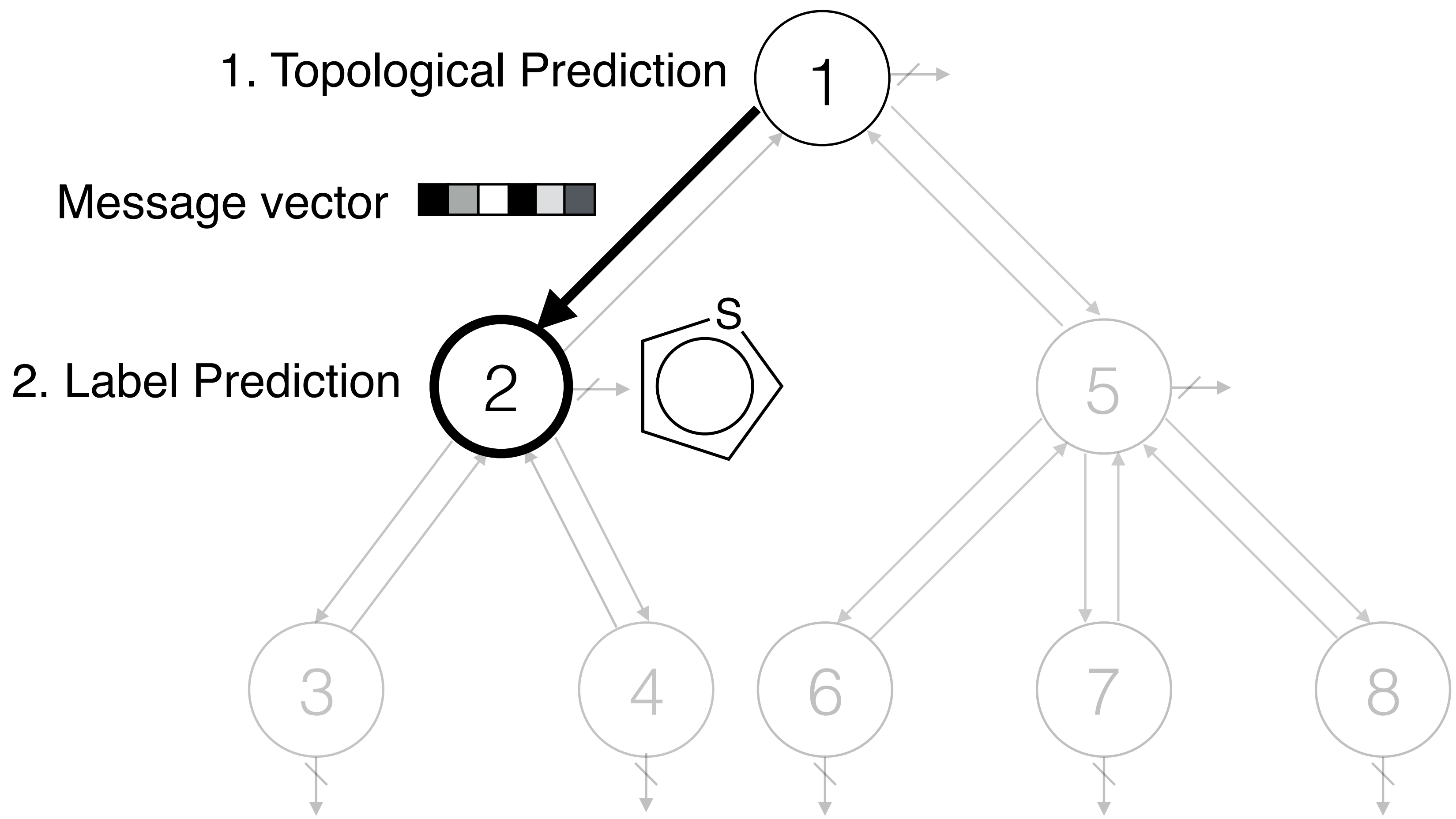


Tree Decoder



[4] Alvarez-Melis & Jaakkola, Tree-structured decoding with doubly-recurrent neural networks

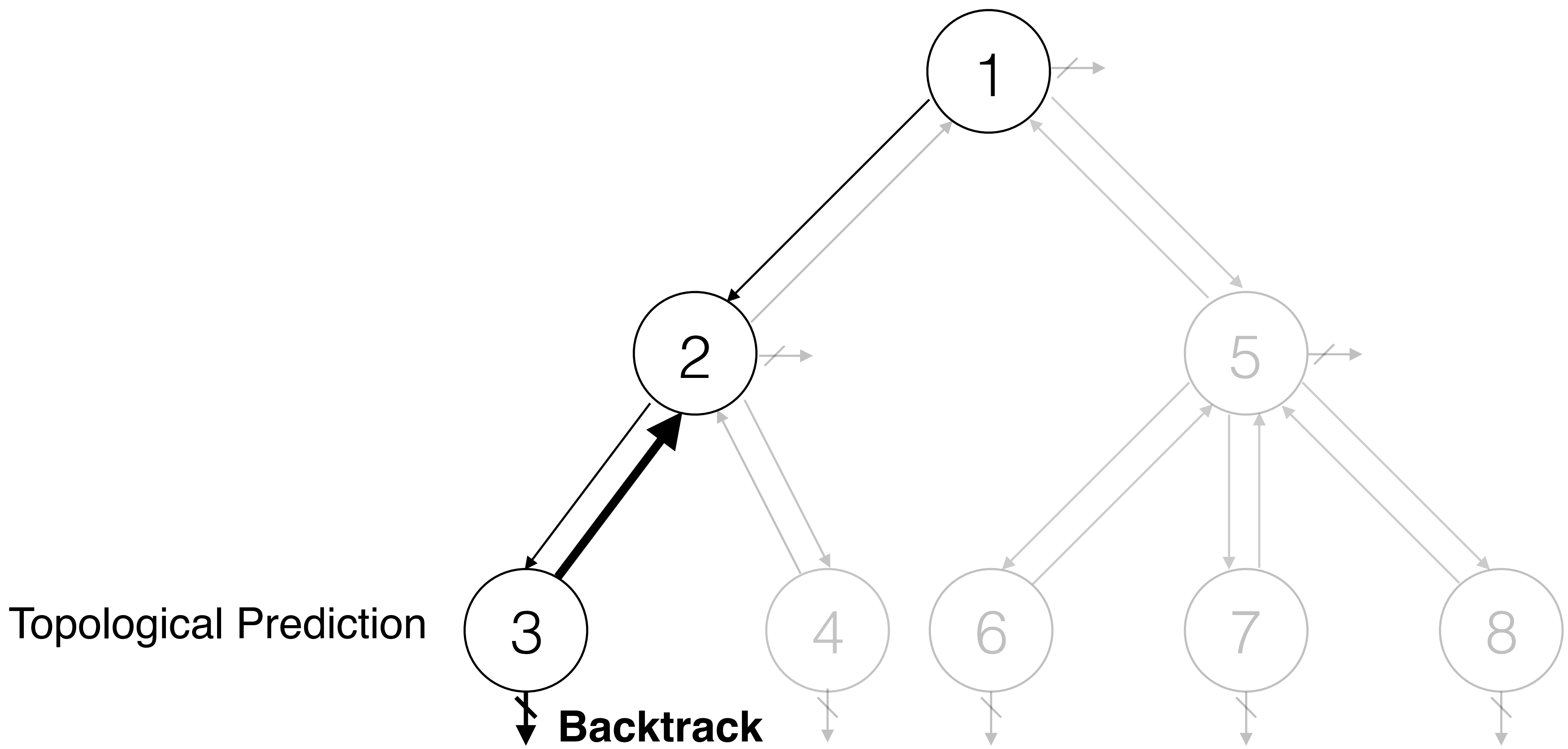
Tree Decoder



Topological Prediction: Whether to expand a child or backtrack?

Label Prediction: What is the label of a node?

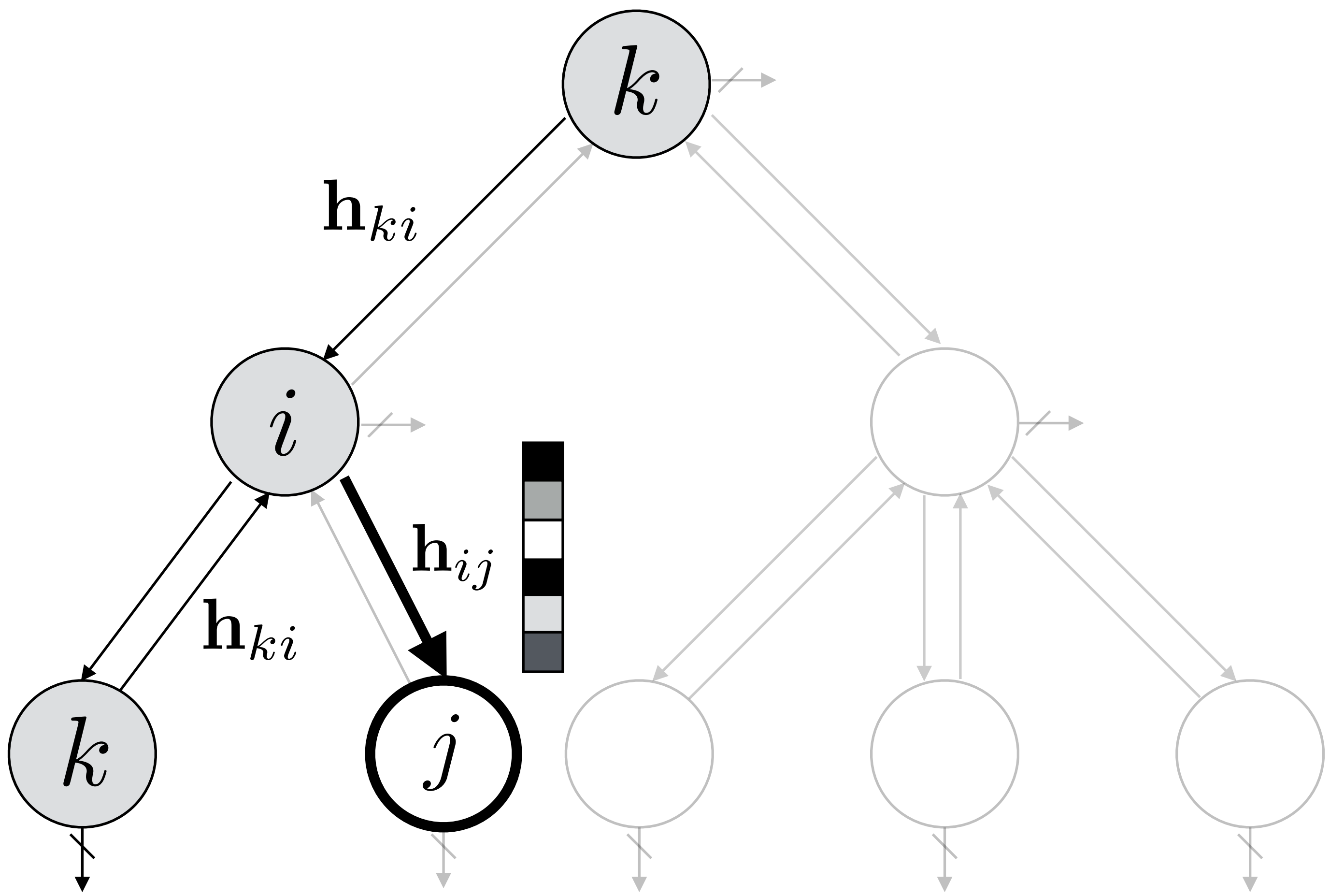
Tree Decoder

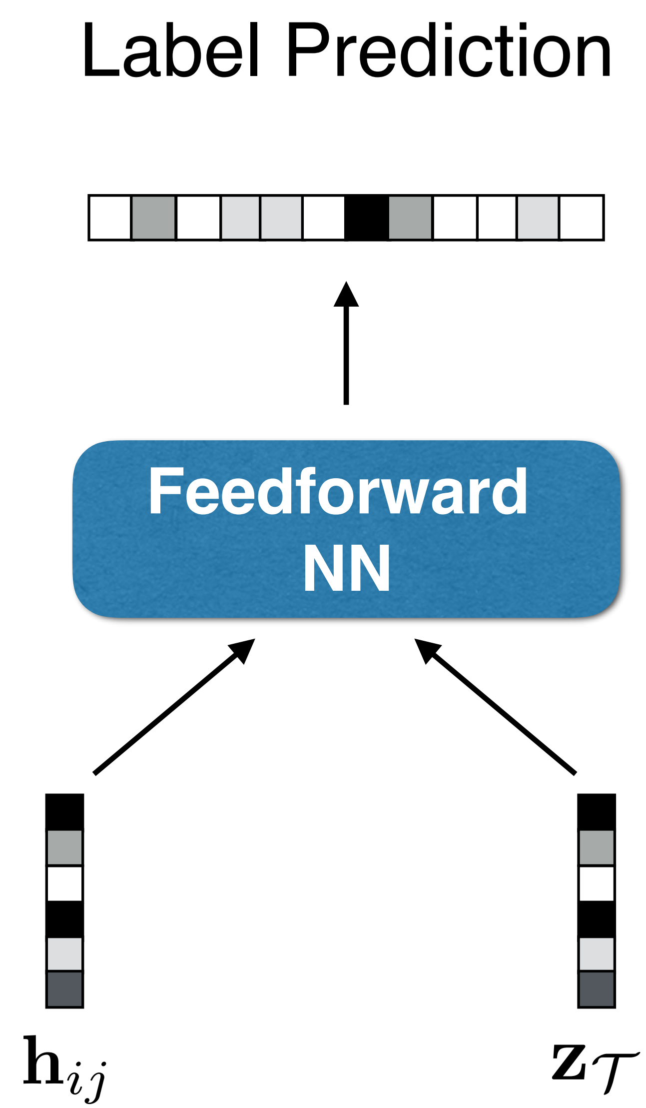


Topological Prediction: Whether to expand a node or backtrack?

Label Prediction: What is the label of a node?

Tree Decoder





$$\mathbf{h}_{ij} = \mathrm{GRU}(\mathbf{x}_i, \{\mathbf{h}_{ki}\}_{k \in N_t(i) \setminus j})$$

Encodes the entire subtree of current state

Tree Decoder

Algorithm 1 Tree decoding at sampling time

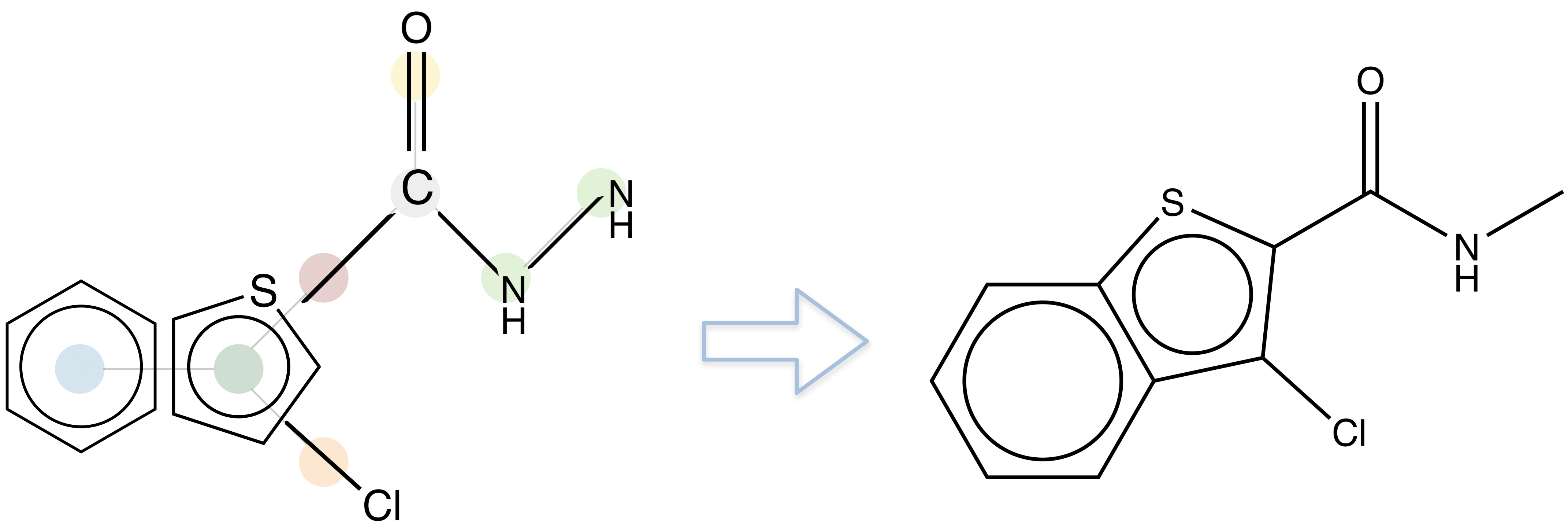
```
Require: Latent representation z_T
Initialize: Tree T̂ ← ∅
function SampleTree(i, t)
    Set X_i ← all cluster labels that are chemically compatible with node i and its current neighbors.
    Set d_t ← expand with probability p_t.            ▷ Eq.(11)
    if d_t = expand and X_i ≠ ∅ then
        Create a node j and add it to tree T̂.
        Sample the label of node j from X_i            ▷. Eq.(12)
        SampleTree(j, t + 1)
    end if
end function
```

$$p_t = \sigma(\mathbf{u}^d \cdot \tau(\mathbf{W}_1^d \mathbf{x}_{i_t} + \mathbf{W}_2^d \mathbf{z}_{\mathcal{T}} + \mathbf{W}_3^d \sum_{(k,i_t)\in\tilde{\mathcal{E}}_t} \mathbf{h}_{k,i_t}) \quad (11)$$

$$\mathbf{q}_j = \mathrm{softmax}(\mathbf{U}^l \tau(\mathbf{W}_1^l \mathbf{z}_{\mathcal{T}} + \mathbf{W}_2^l \mathbf{h}_{ij})) \quad (12)$$

$$\mathcal{L}_c(\mathcal{T}) = \sum_t \mathcal{L}^d(p_t, \hat{p}_t) + \sum_j \mathcal{L}^l(\mathbf{q}_j, \hat{\mathbf{q}}_j) \quad (13)$$

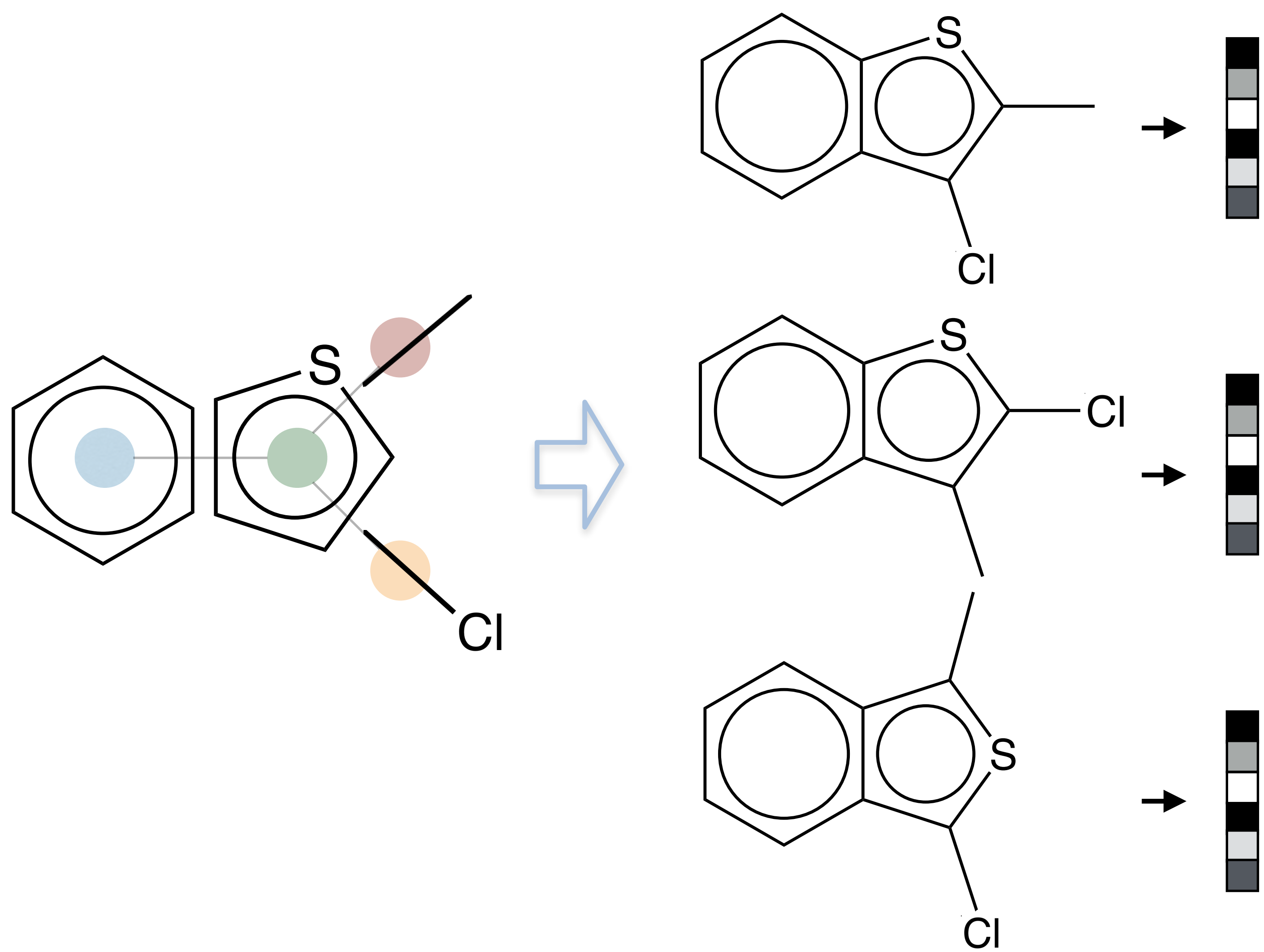
Graph Decoder



Predicted Junction Tree

Molecular Graph

Graph Decoder



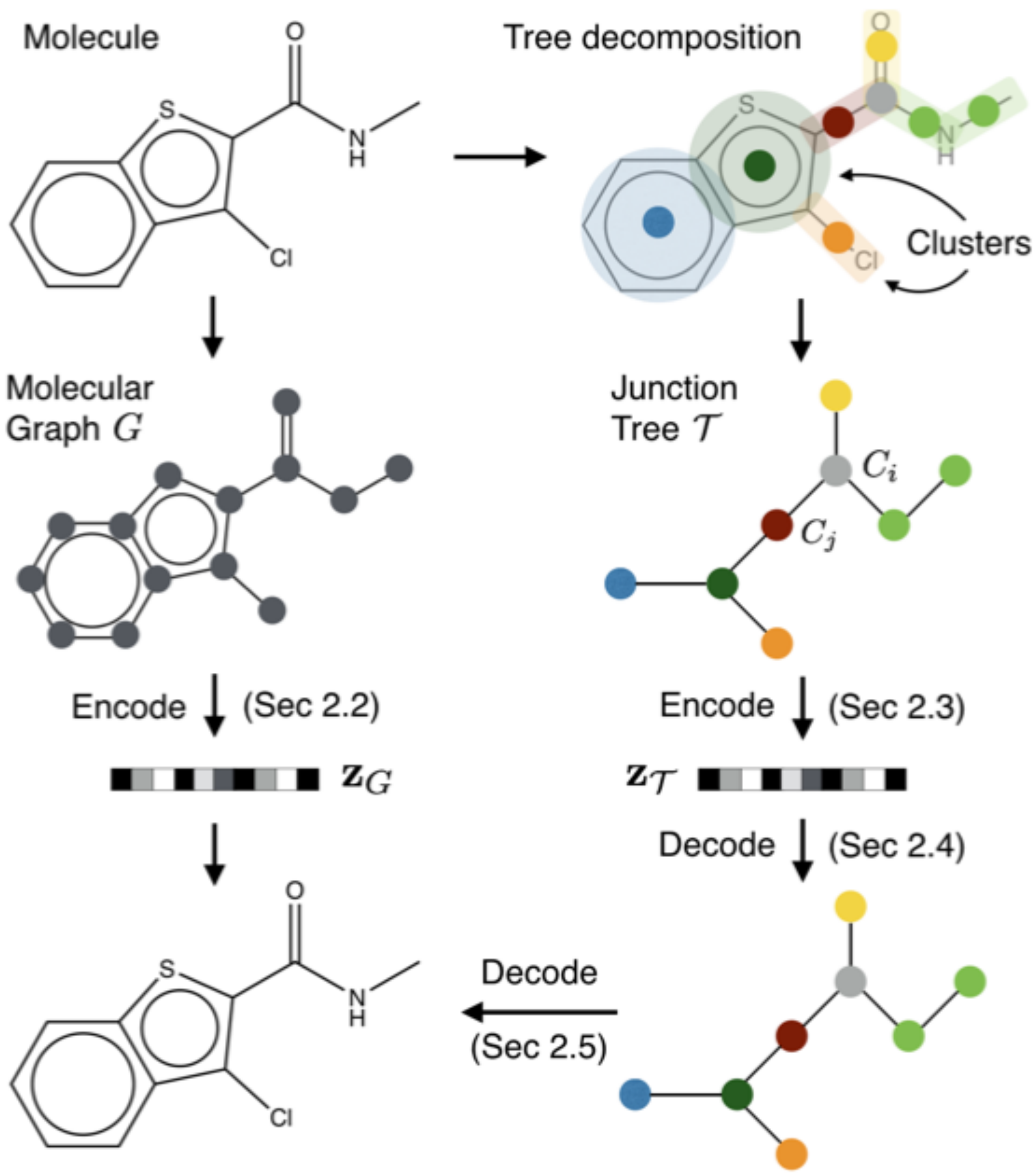
Enumerated subgraphs G_i

Graph encoder

1. Enumerate how clusters are merged together
2. Encode each candidate graph by graph encoder
3. Score each candidate: $f_i^a(G_i) = \mathbf{h}_{G_i} \cdot \mathbf{z}_G$

$$\mathcal{L}_g(G) = \sum_i \left[f^a(G_i) - \log \sum_{G'_i \in \mathcal{G}_i} \exp(f^a(G'_i)) \right] \quad (16)$$

Training? VAE?



- The KL divergence part on the latent space is not discussed in the paper.
- z_G is only used for generated subgraphs ranking so not clear how it falls in the VAE paradigm.
- From the code, training is with KL annealing following "Generating Sentences from a continuous space" paper by Bowman et al.

Experiments

- **Data**: 250K compounds from ZINC dataset
- **Molecule Generation**: How many molecules are valid when sampled from Gaussian prior?
- **Molecule Optimization**
    - **Global**: Find the best molecule in the entire latent space.
    - **Local**: Modify a molecule to increase its potency

Baselines

SMILES string based:

1. Grammar VAE (GVAE) (Kusner et al., 2017);

2. Syntax-directed VAE (SD-VAE) (Dai et al., 2018)

Graph based:

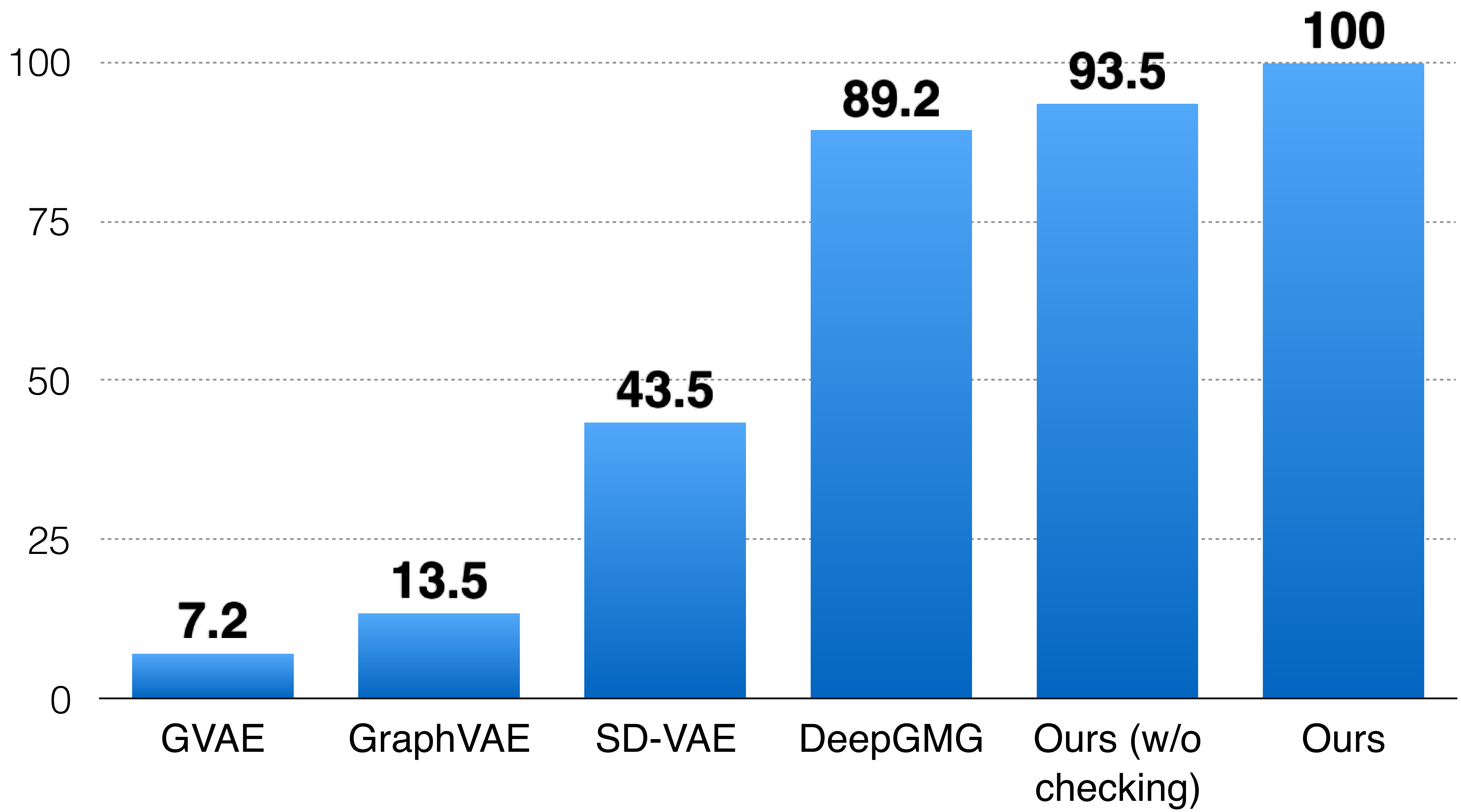
1. Graph VAE (Simonovsky & Komodakis, 2018)

2. DeepGMG (Li et al., 2018)

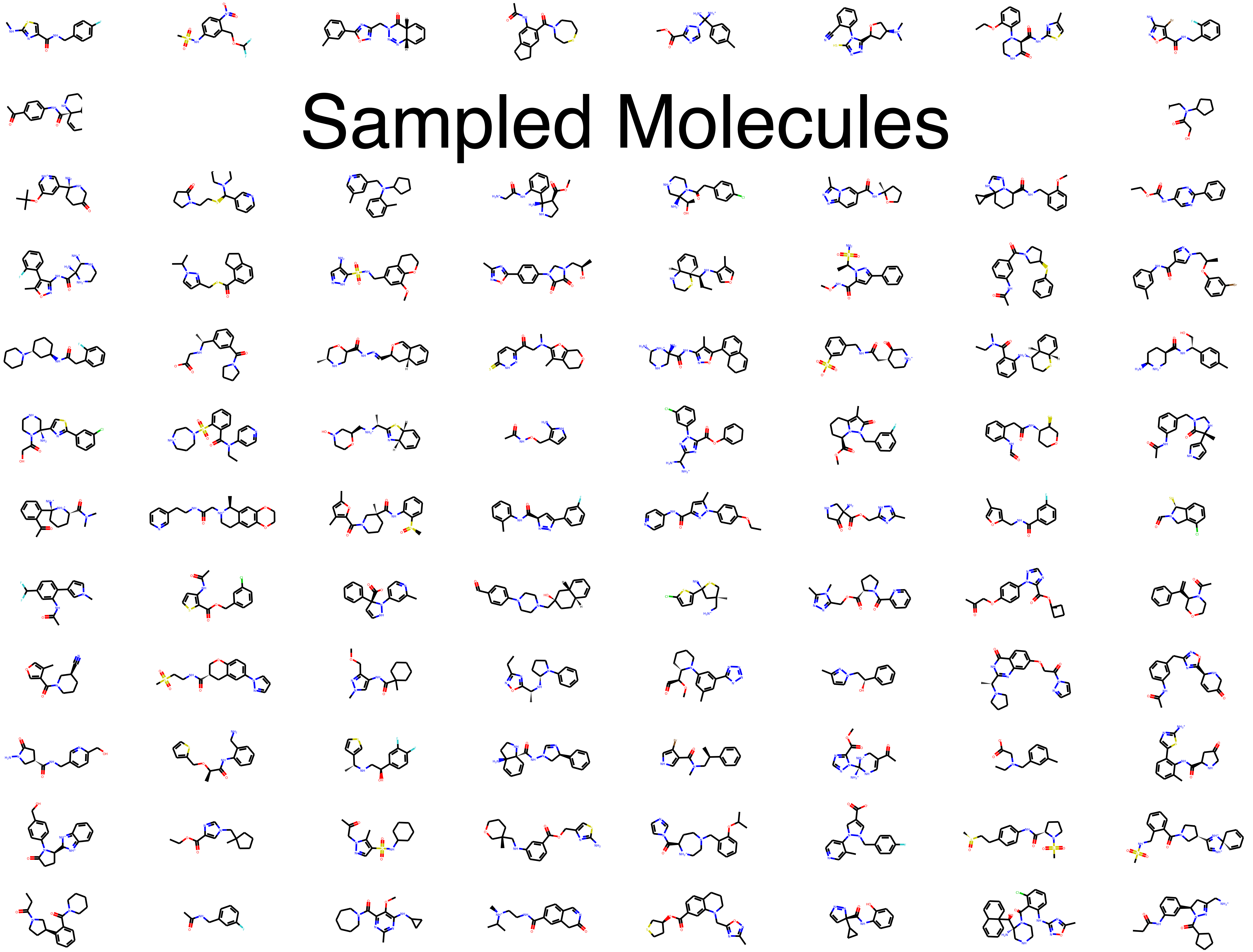
---

[2] Li et al., Learning Deep Generative Models of Graphs, 2018
[5] Kusner et al., Grammar Variational Autoencoder, 2017
[6] Dai et al., Syntax-directed Variational Autoencoder for structured data, 2018
[7] Simonovsky & Komodakis, GraphVAE: Towards generation of small graphs using variational autoencoders

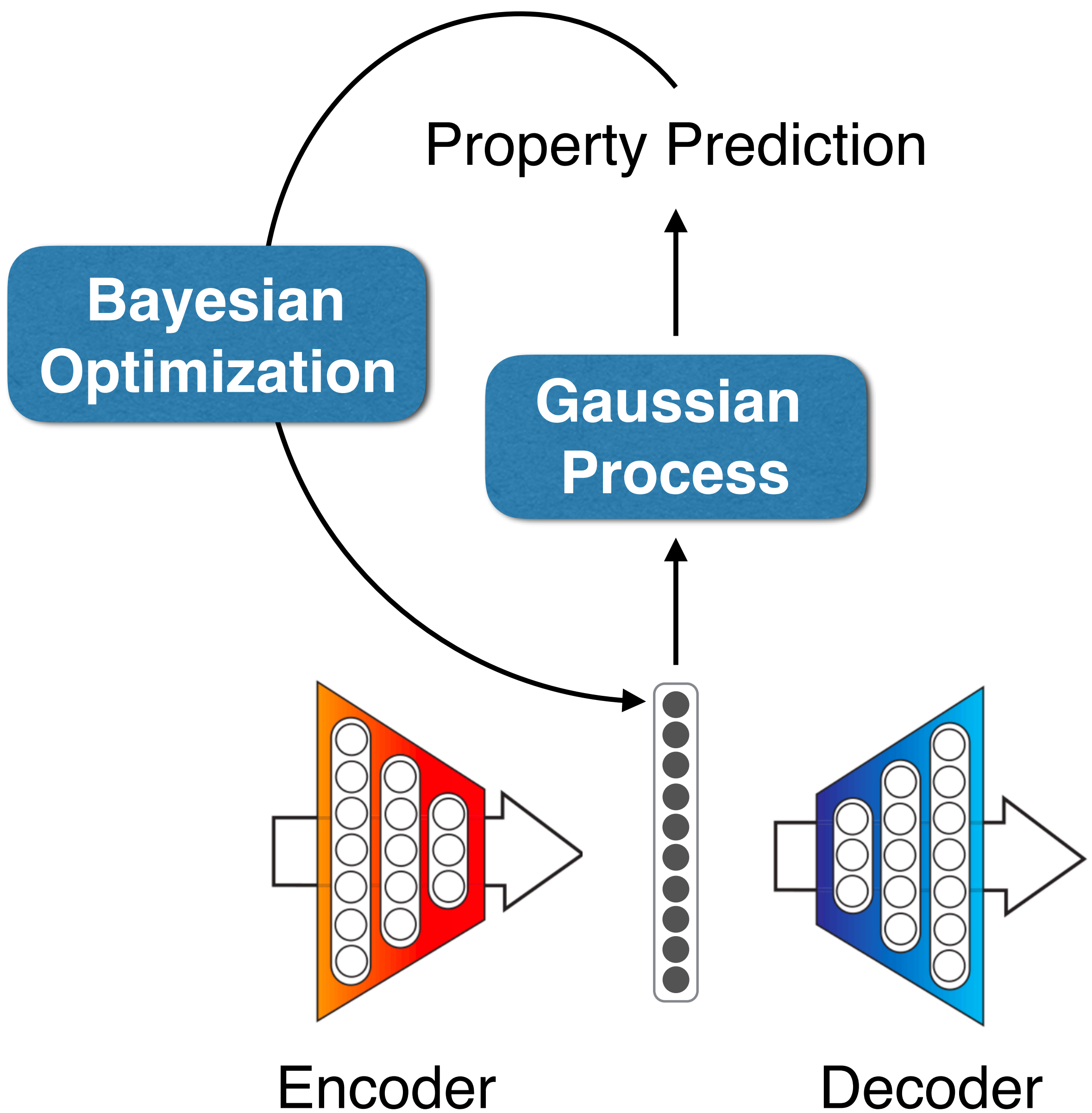
Molecule Generation (Validity)

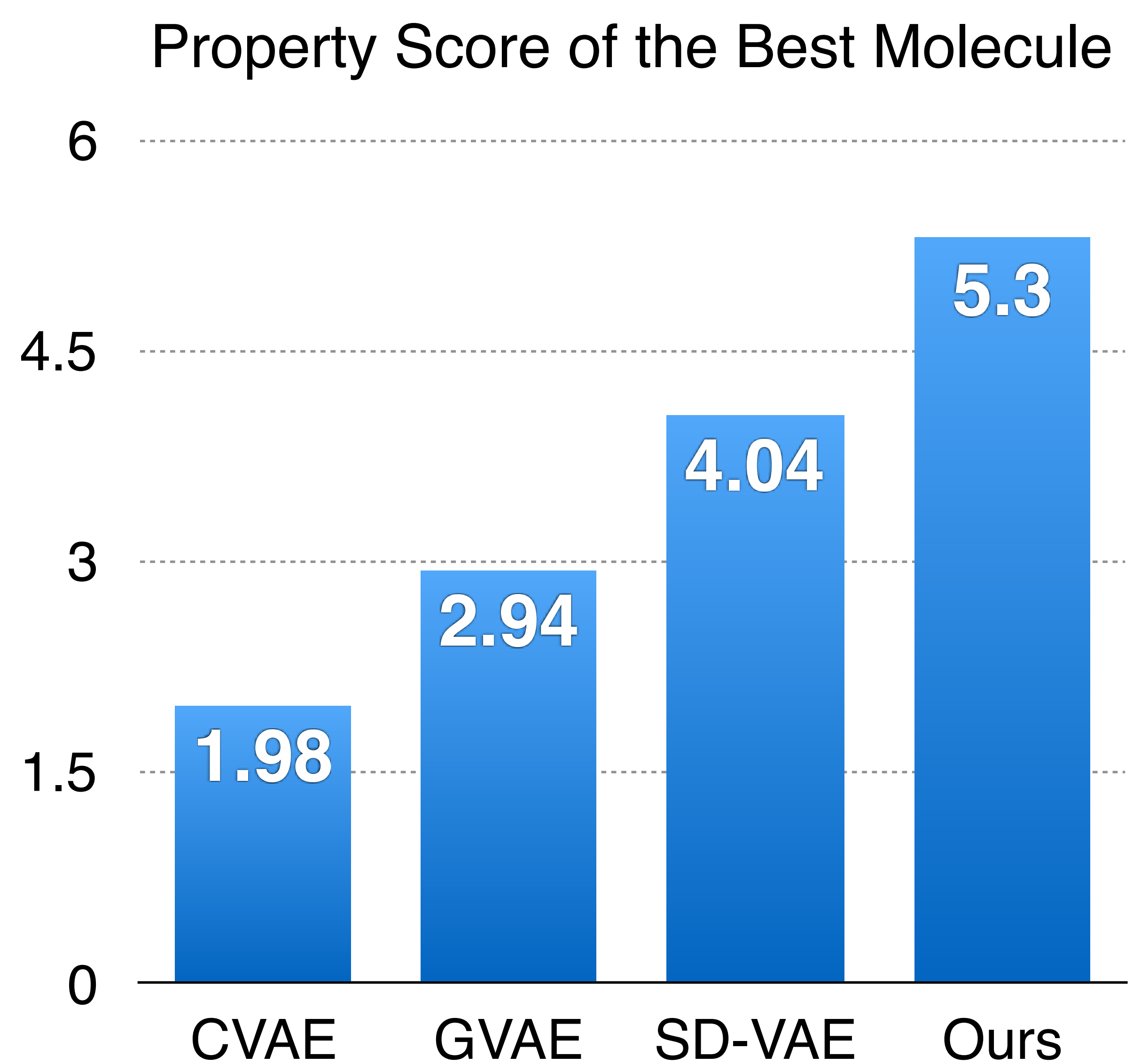


Sampled Molecules



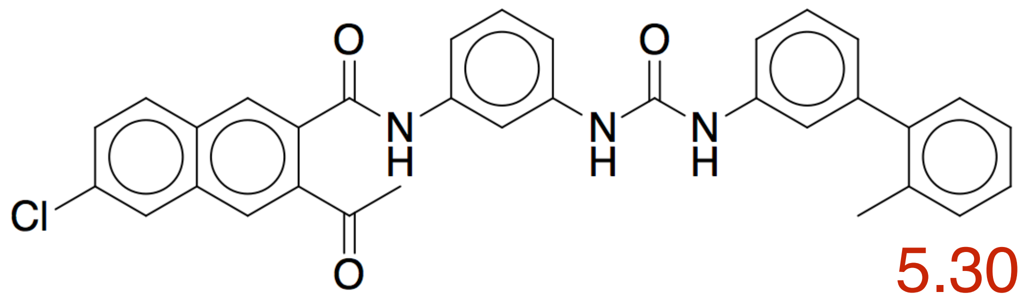
Molecule Optimization (Global)



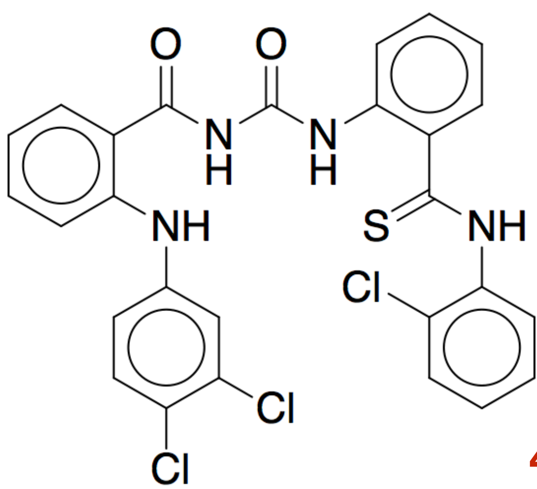


Property: Solubility + Ease of Synthesis

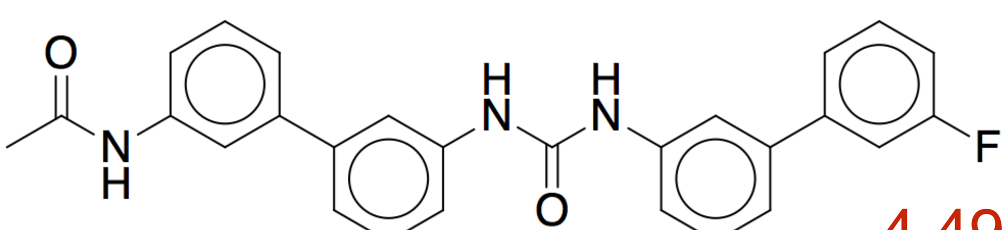
Molecule Optimization (Global)



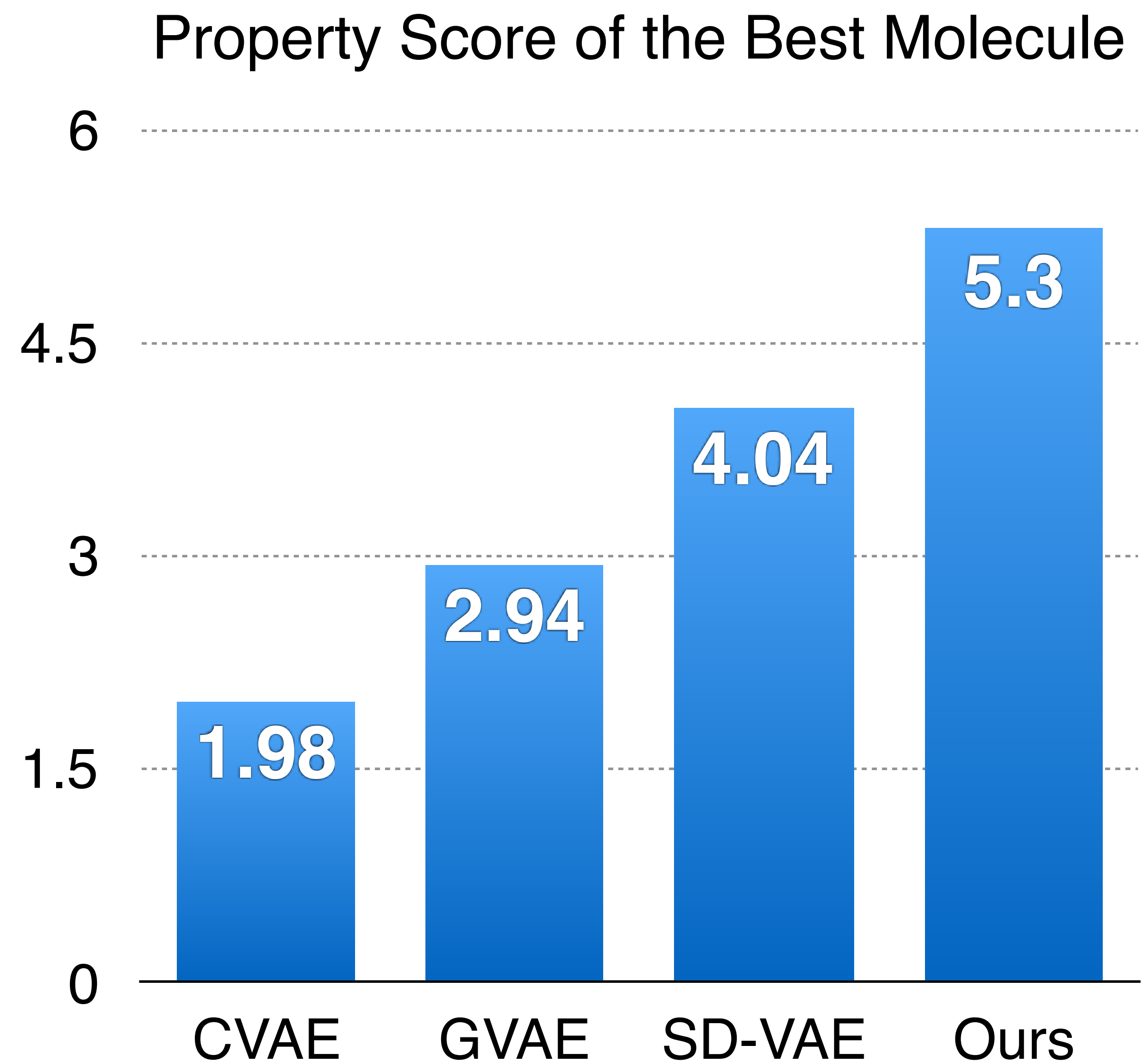
5.30



4.93

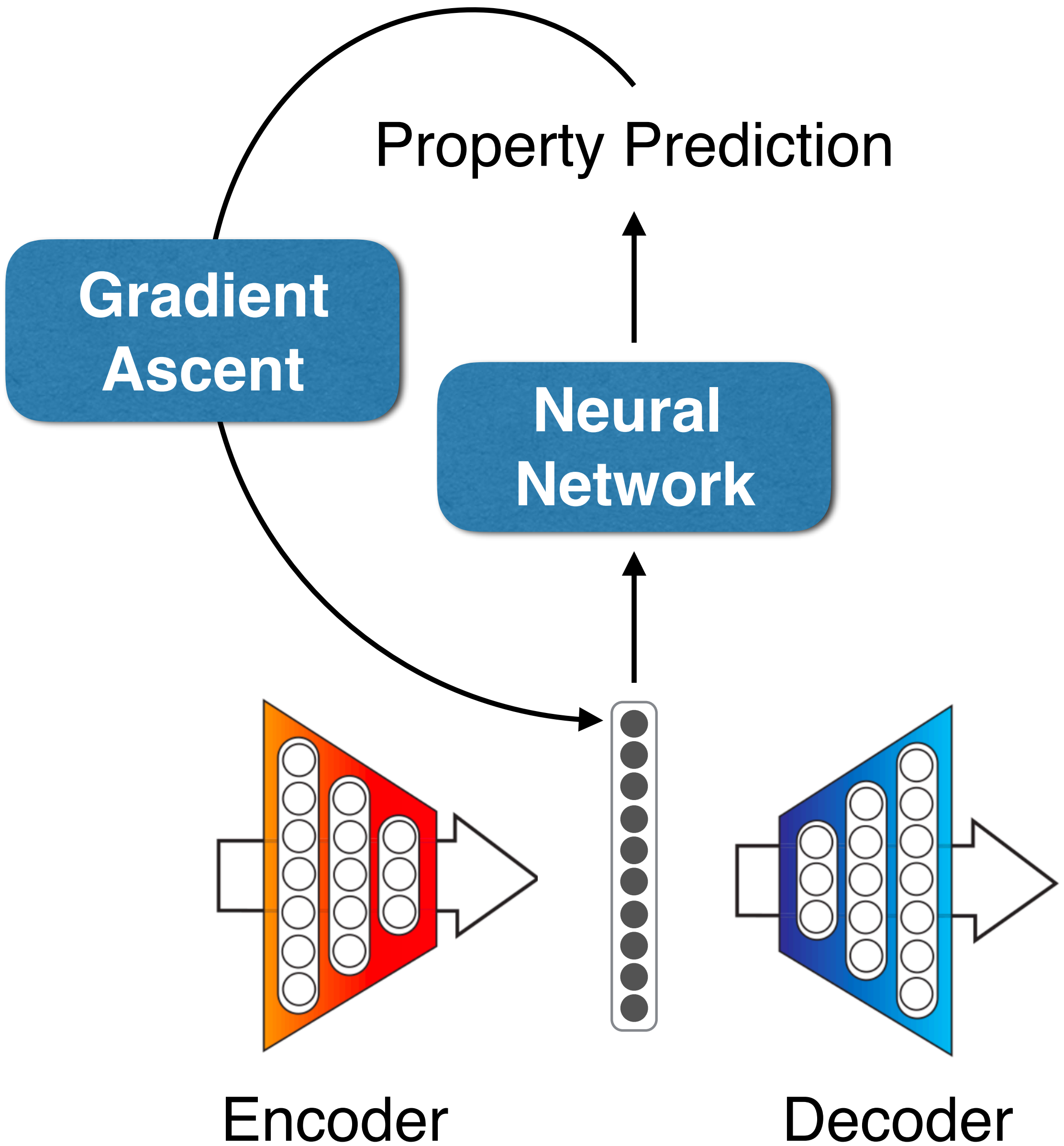


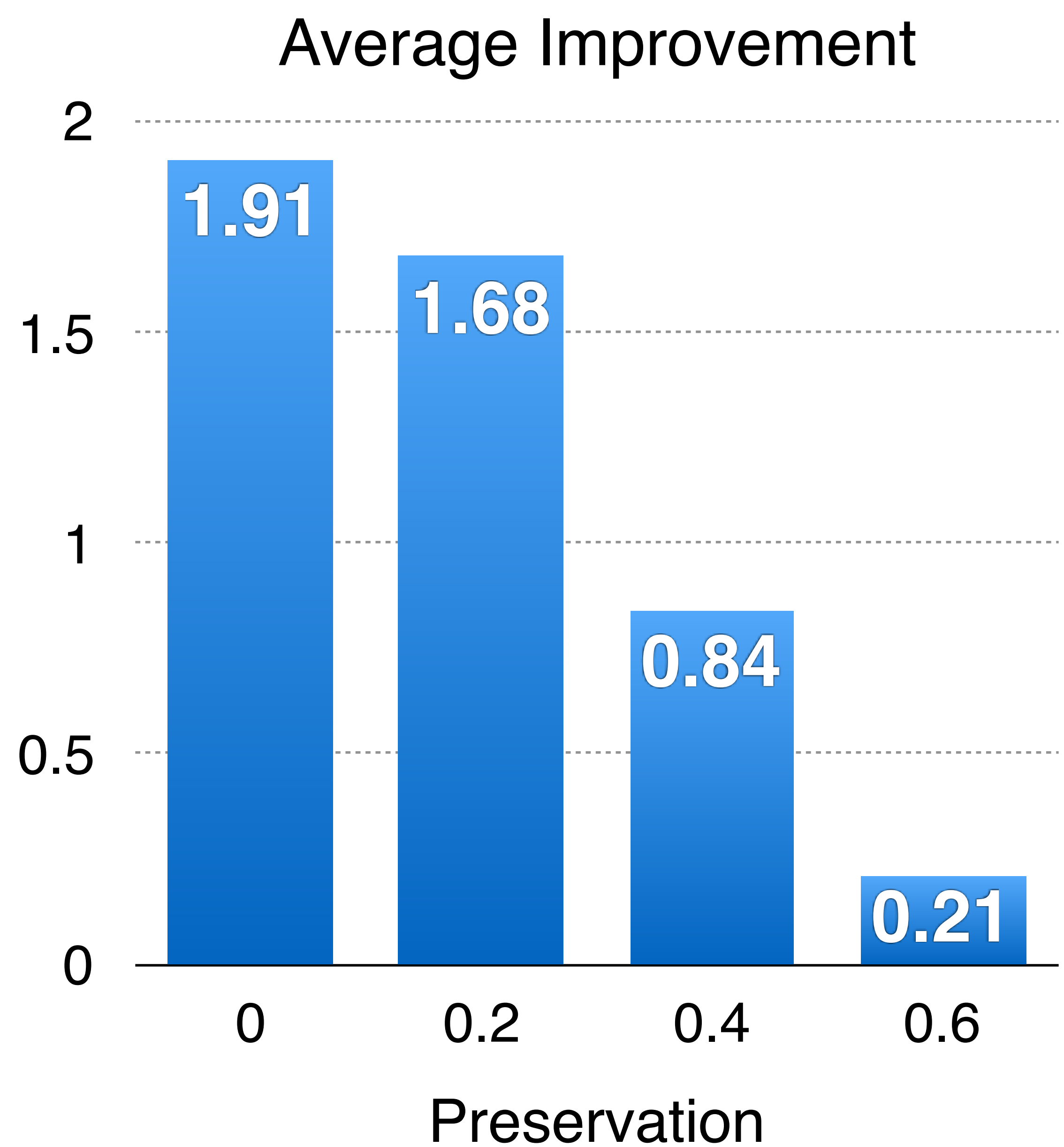
4.49



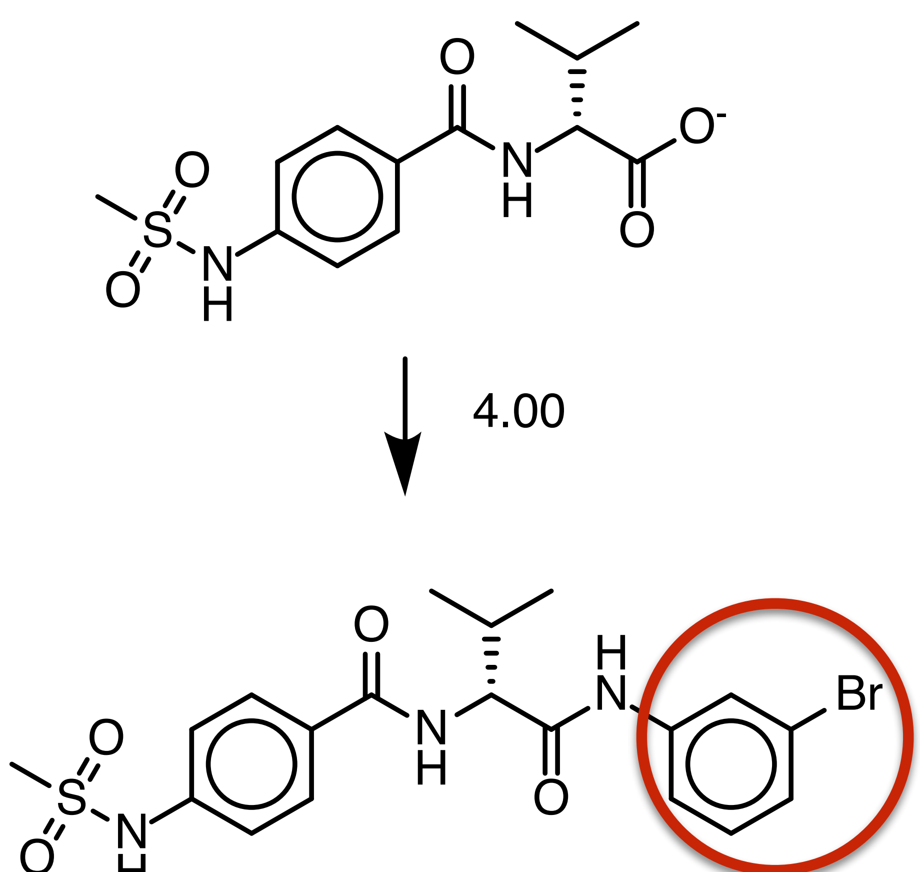
Property: Solubility + Ease of Synthesis

Molecule Optimization (Local)

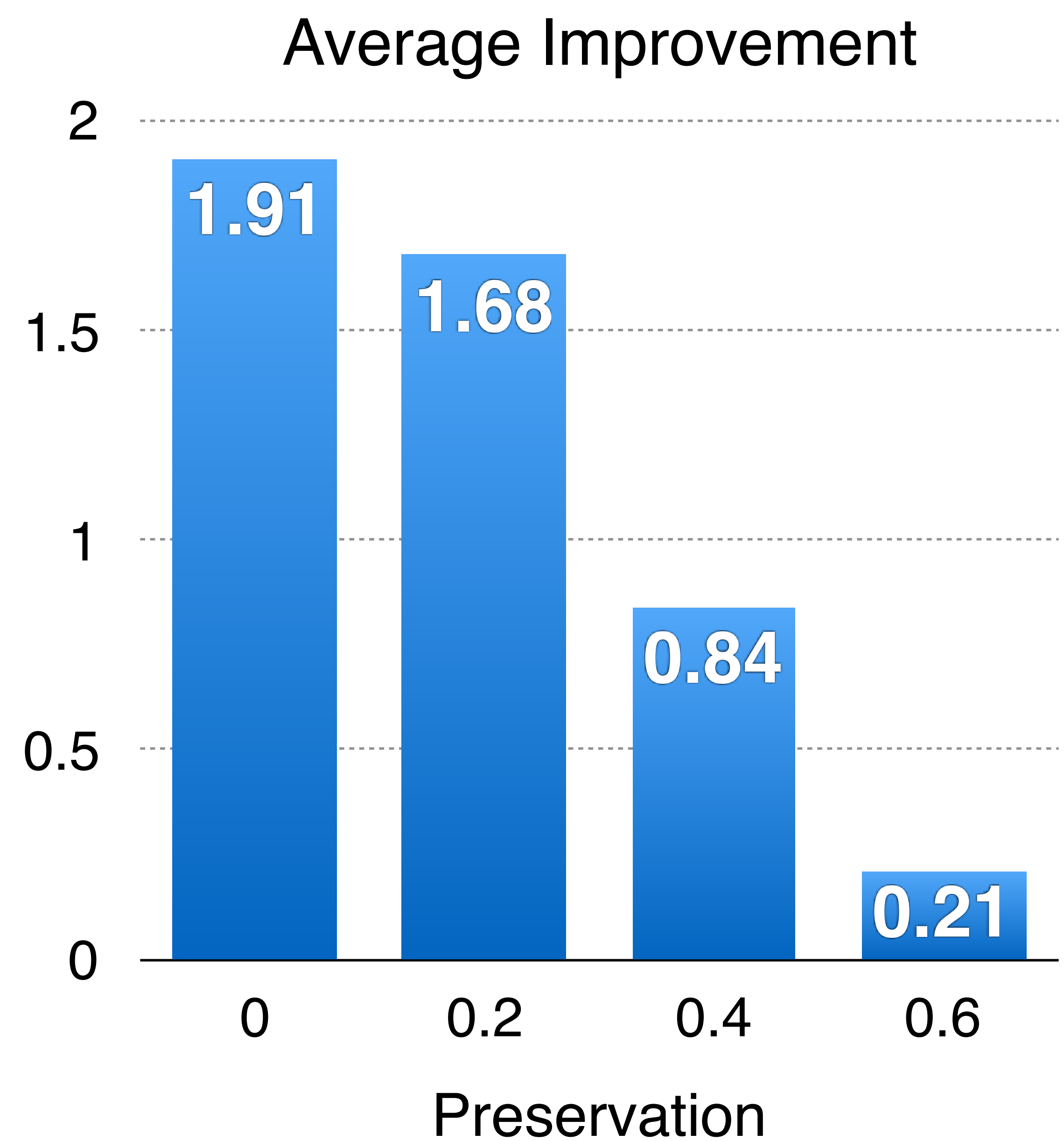




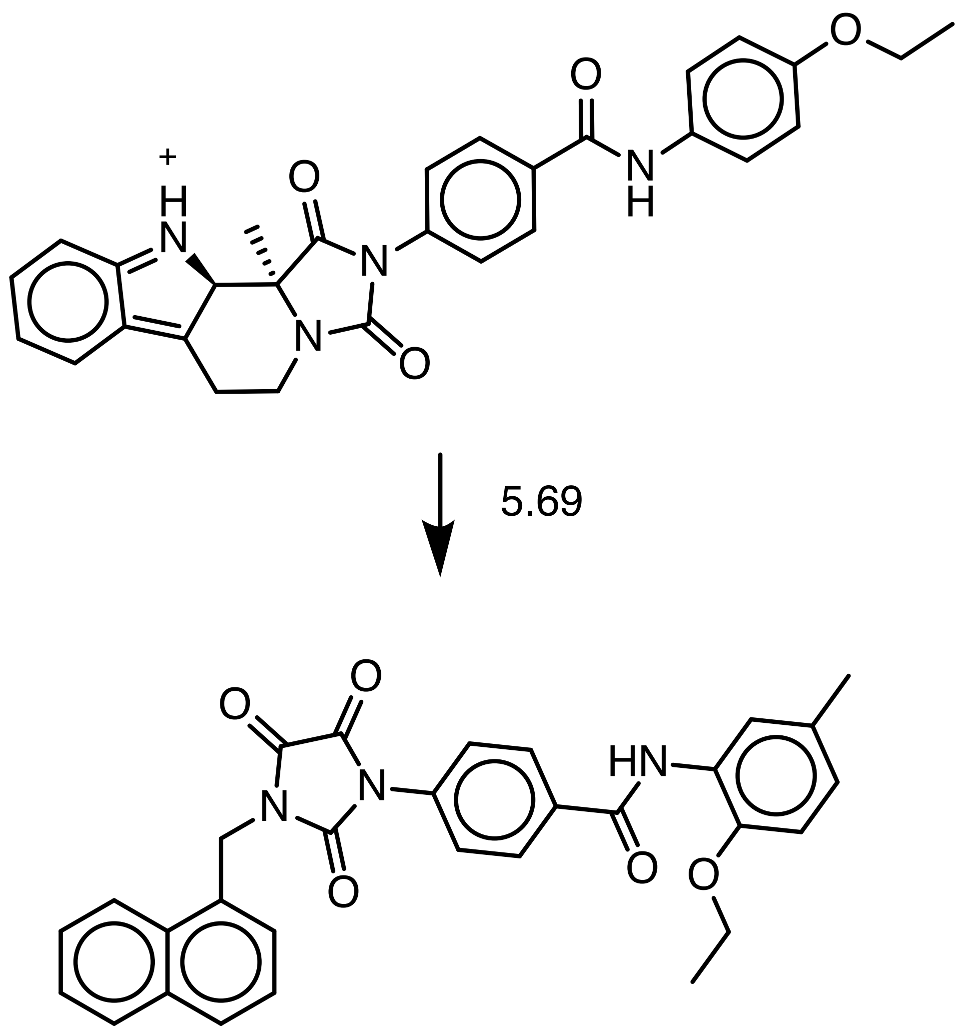
Molecule Optimization (Local)



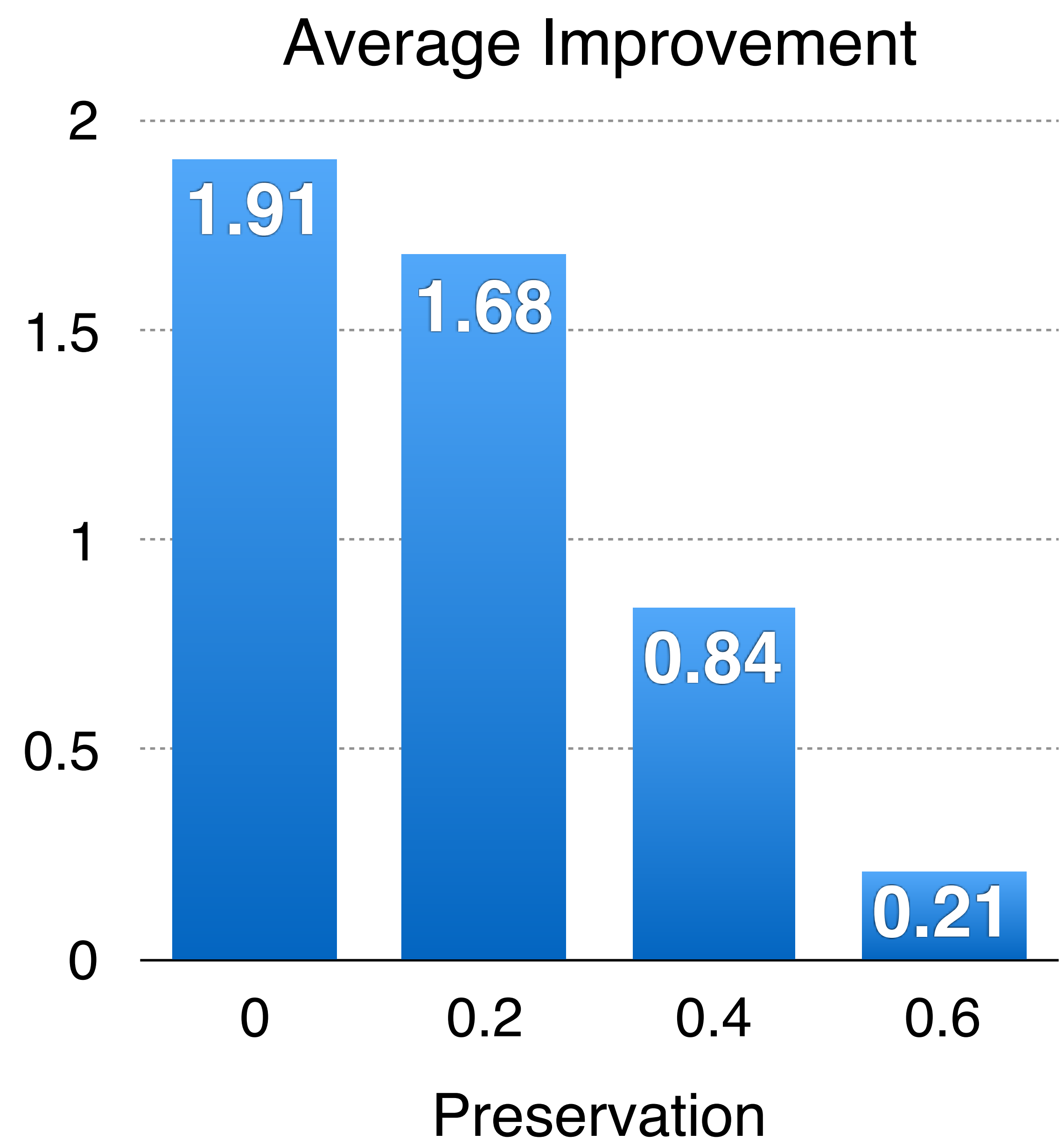
Preservation ≈ 0.6



Molecule Optimization (Local)



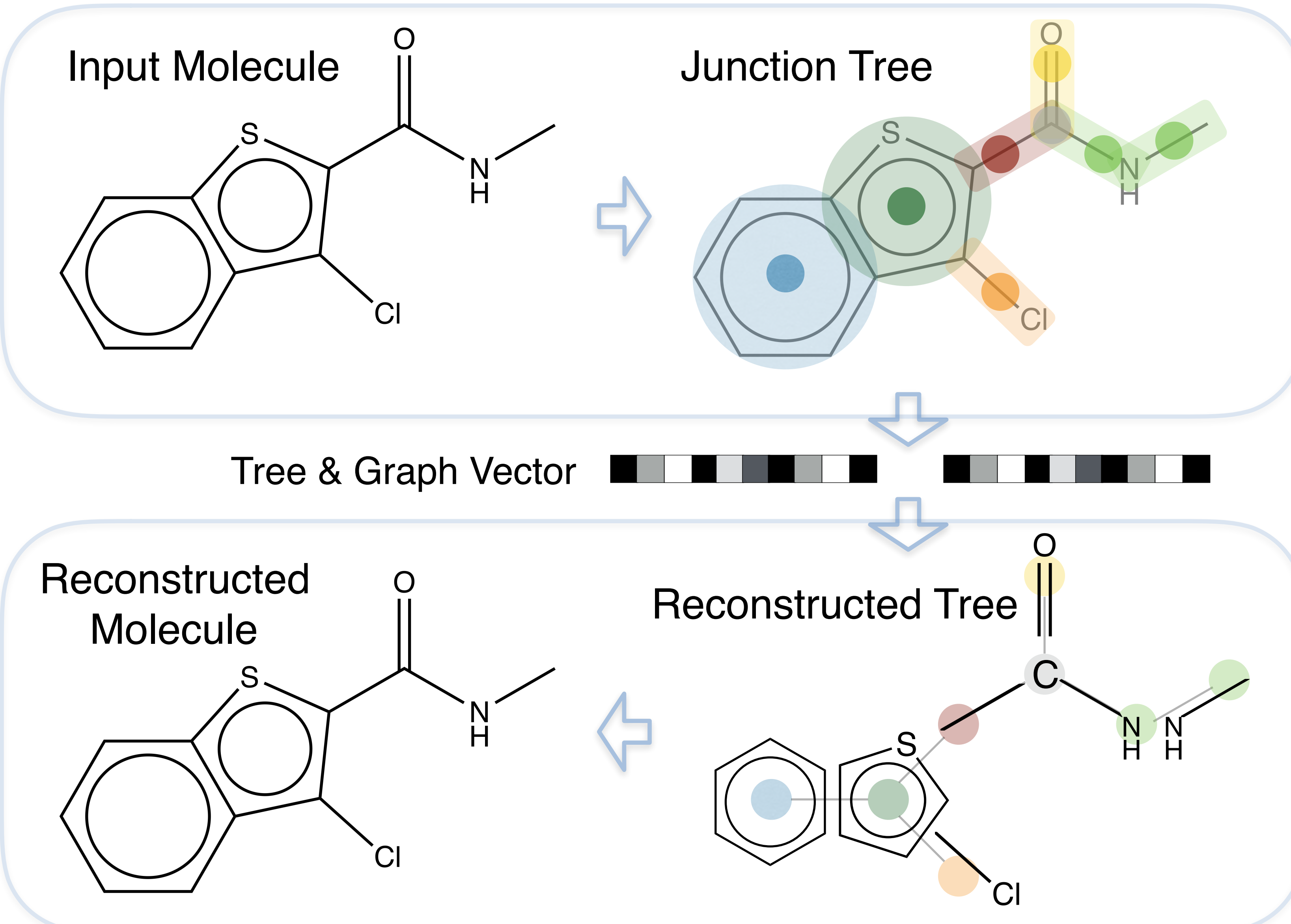
Preservation ≈ 0.4



Discussion

- “word level” prediction can offer significant improvement by shortening the decision process.
- Latent space optimization is an interesting and powerful technique.
- “Teacher forcing” introduces data bias which can be reduced via RL techniques and the GAN complete graph valuation approach.
- Similar to SMILES this paper samples a random order in the graph tree structure when: using an arbitrary minimal spanning tree, choosing an arbitrary node to be the root of the tree, choosing a random ordering of the children of each tree node.

Thanks



Original code is available at: https://github.com/wengong-jin/icml18-jtnn