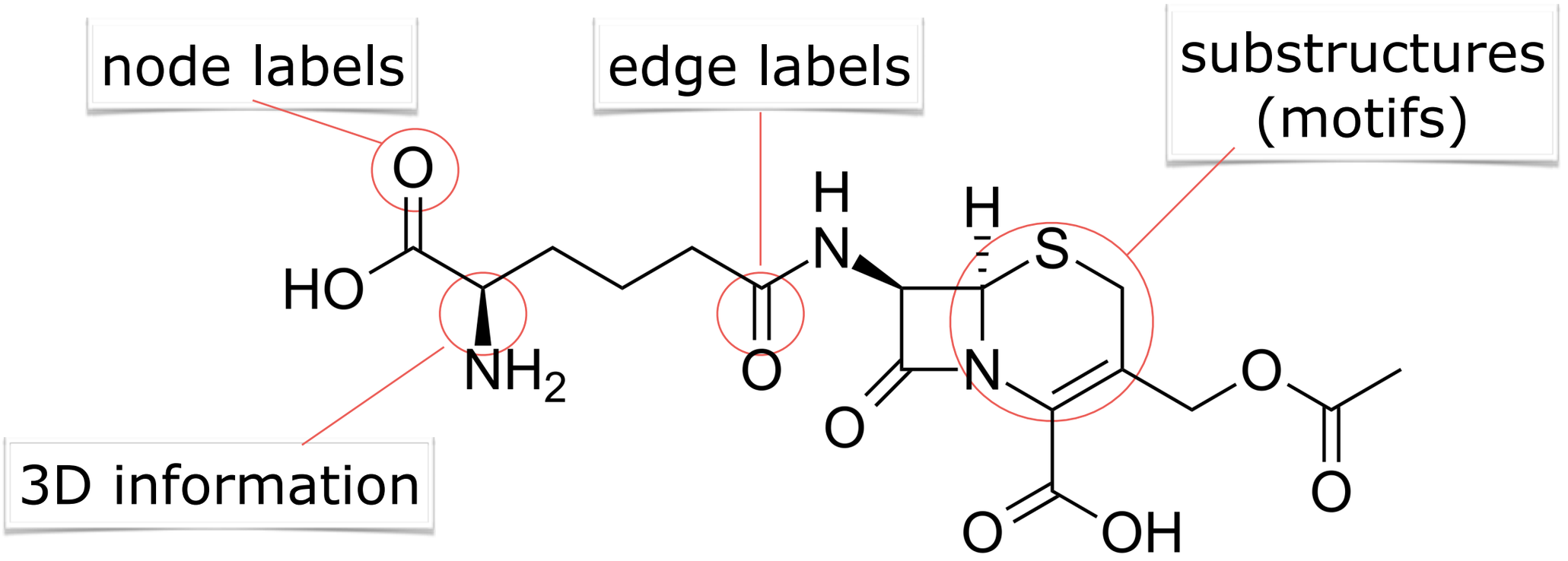
# Representation and Generation of Molecular Graphs

**Tommi Jaakkola**
**MIT**

**in collaboration with**
**Wengong Jin, Vikas Garg, Regina Barzilay**

# Molecules = richly annotated graphs

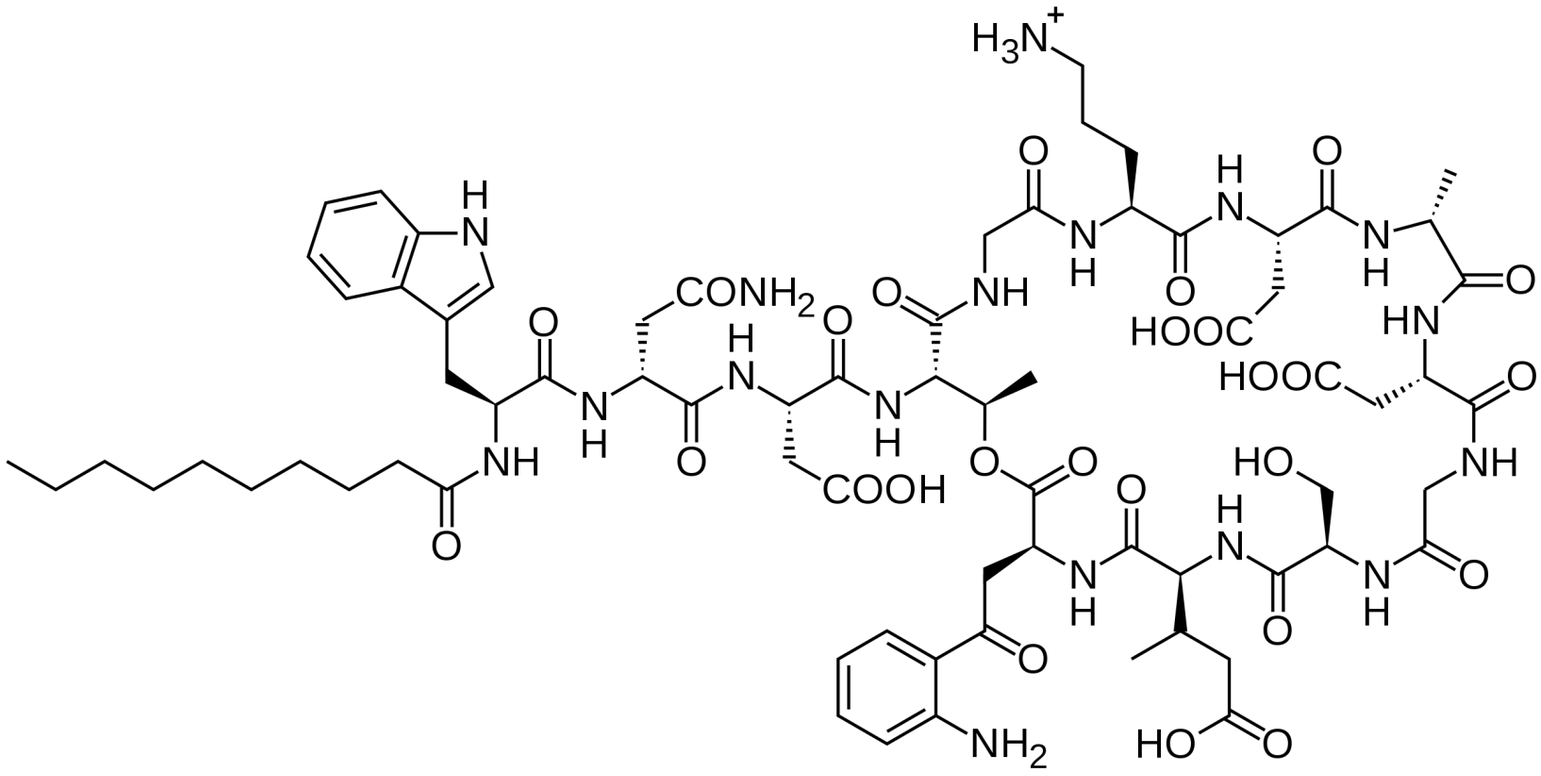
- E.g., antibiotic (cephalosporin)




- Together, the features give rise to various molecular properties (e.g., solubility, toxicity, etc)

# Why interesting for ML?

- **Rich, complex objects:** molecules are complicated structures, properties may depend on intricate features
- **Data:** big and small, heterogenous
- **Estimation/inferential challenges:** many high-impact but non-trivial tasks such as prediction of chemical properties, molecular optimization, etc.




(Daptomycin antibiotic)

# Our motivation: this talk

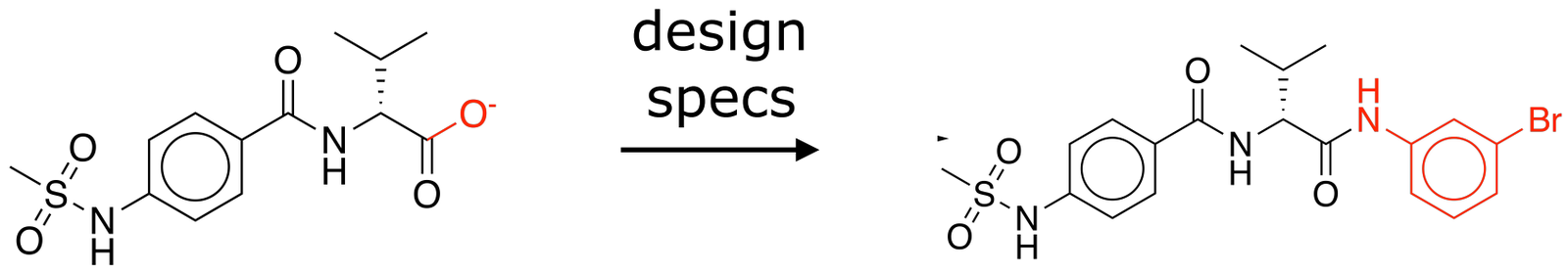
- **Deeper into molecules**
  - molecular property prediction (e.g., toxicity, bioactivity)
  - (multi-criteria) optimization (e.g., potency, toxicity)

# Our motivation: broader context

- **MIT Consortium (https://mlpds.mit.edu/)**
  - 14 major pharmaceutical companies
  - chemistry/chemical engineering (Jensen, Green, Jamison)
  - computer science (Barzilay, Jaakkola)
- **Deeper into known chemistry**
  - extract chemical knowledge from journals, notebooks
- **Deeper into molecules**
  - molecular property prediction (e.g., toxicity, bioactivity)
  - (multi-criteria) optimization (e.g., potency, toxicity)
- **Deeper into reactions**
  - forward synthesis prediction (major products of reactions)
  - forward synthesis optimization (conditions, reagents, etc.)
- **Deeper into making things**
  - retrosynthetic planning (efficient/inexpensive routes)

# Automating Drug design

- **Our problem:** how to programmatically modify pre-cursor molecules to have better properties




- **Key challenges:**

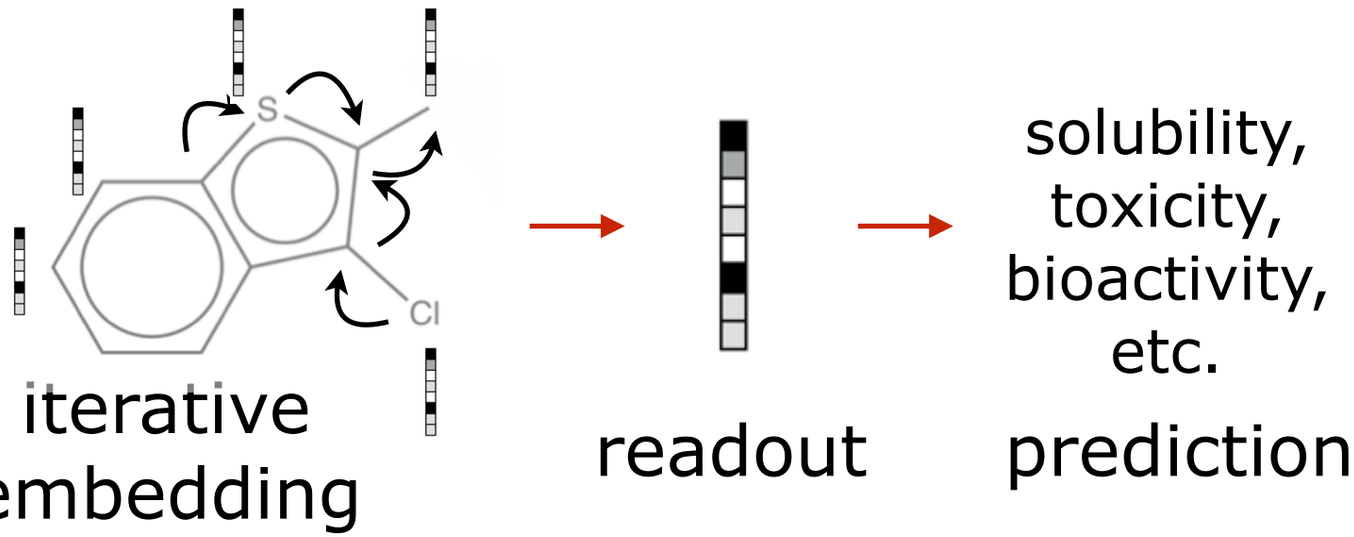
1. **representation and prediction:** learn to predict molecular properties

2. **generation and optimization:** realize target molecules with better properties programmatically

3. **understanding:** uncover principles (or diagnose errors) underlying complex predictions

# Graph neural networks (GNNs)

- GNNs are parameterized message passing algorithms operating on molecular graphs, and result in atom, bond, and graph embeddings, tailored for the end task

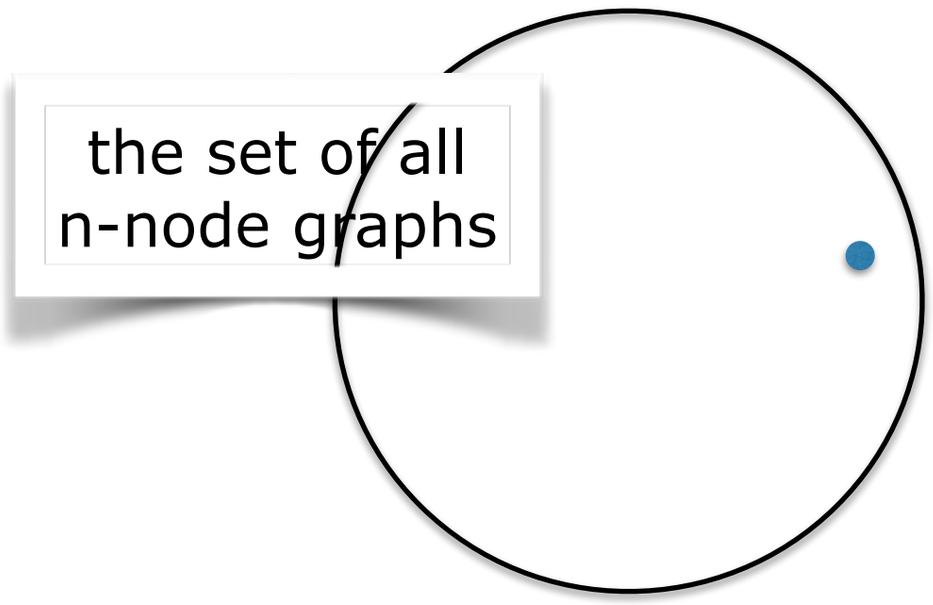


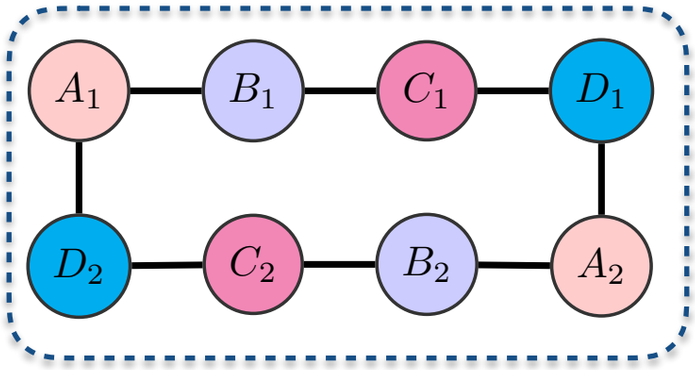

$$h_u^{t+1} = \text{UPDATE}\left(h_u^t, x_u, \text{AGGREGATE}\left(\{h_v^t, x_{uv}\}_{v \in N(u)}\right)\right)$$

$$h_G^T = \text{READOUT}(\{h_u^T\})$$

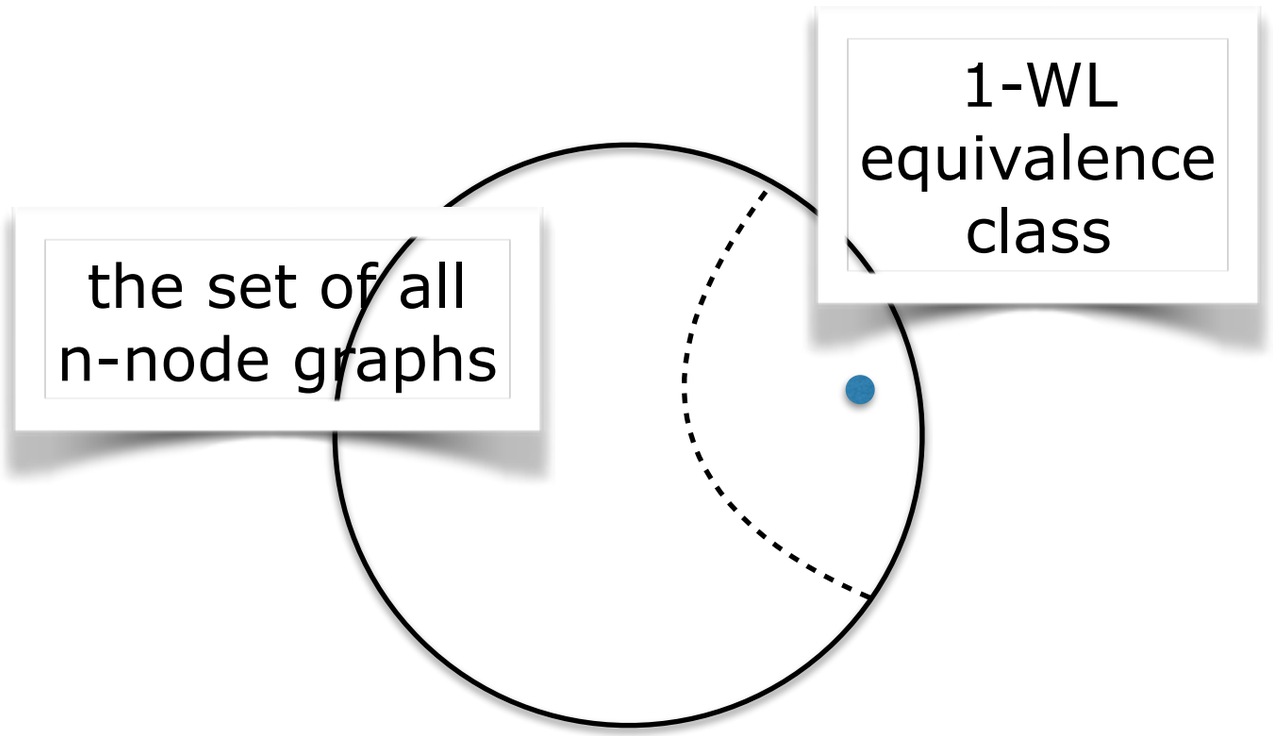
- Many recent results about representational power (Xu et al. 2019, Sato et al. 2019, Maron et al., 2019, …)

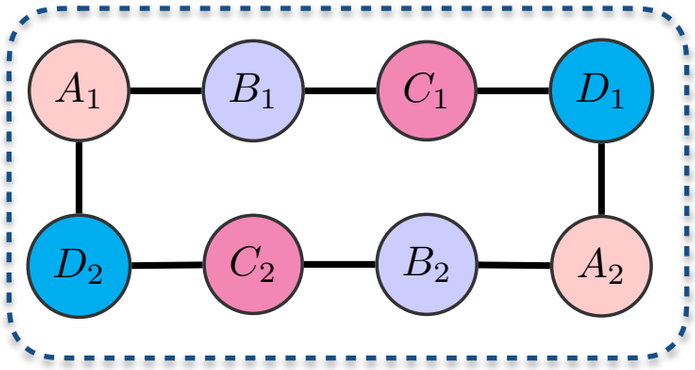
# Representational power of "GNNs"



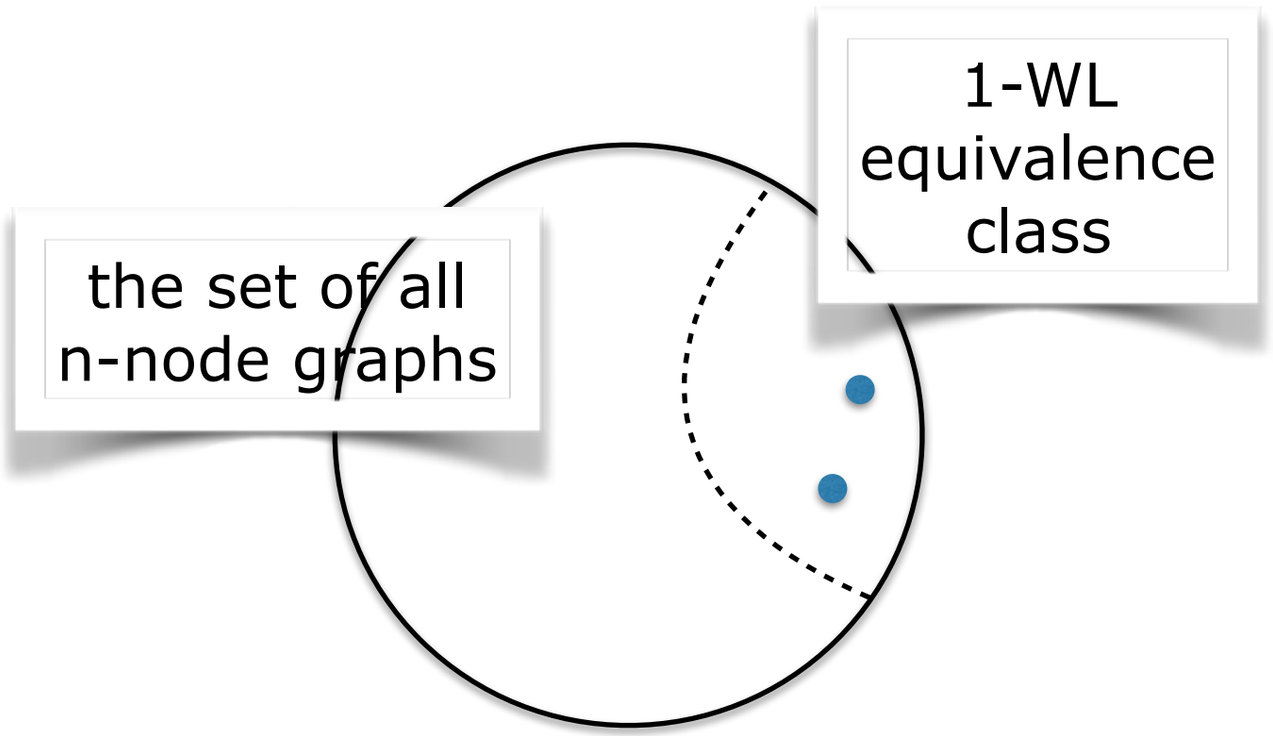



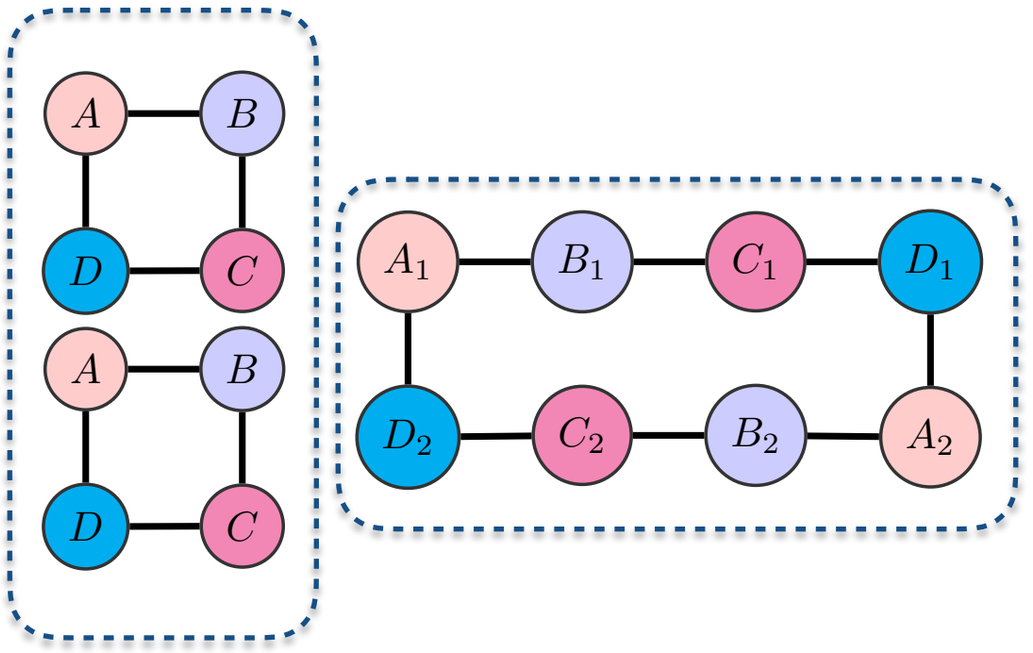
# Representational power of “GNNs”



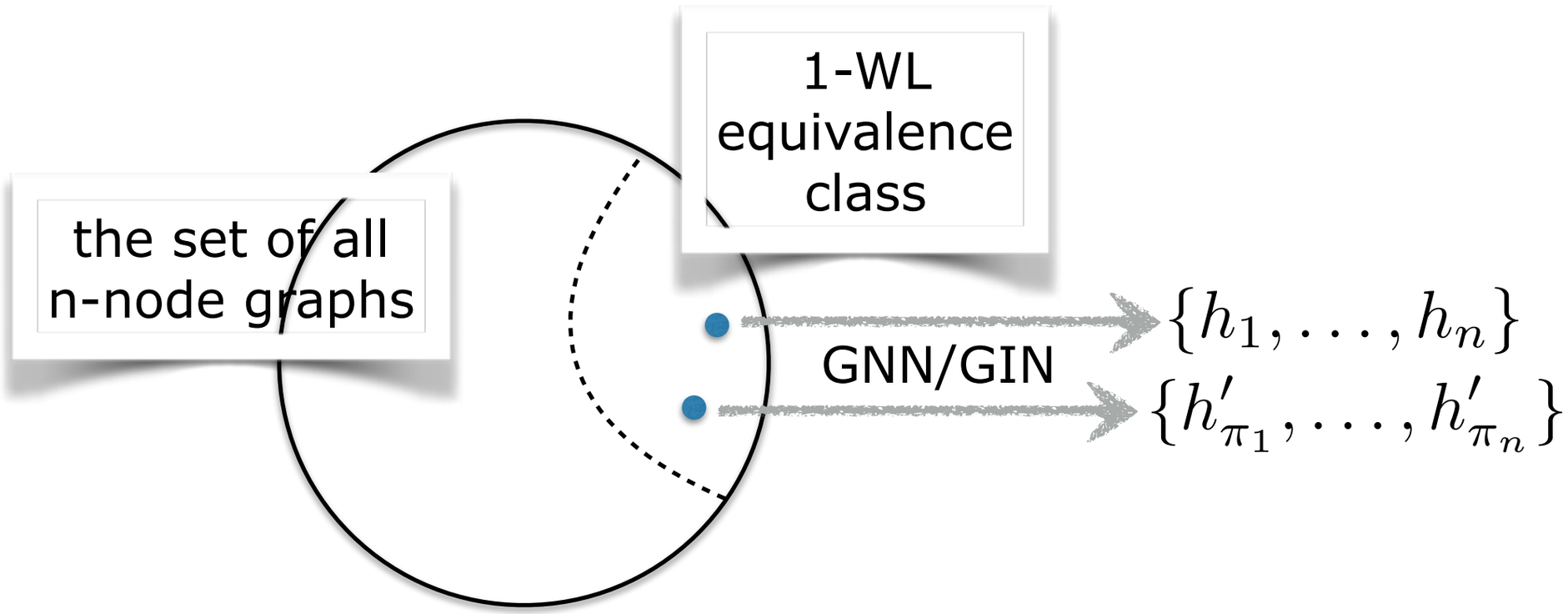



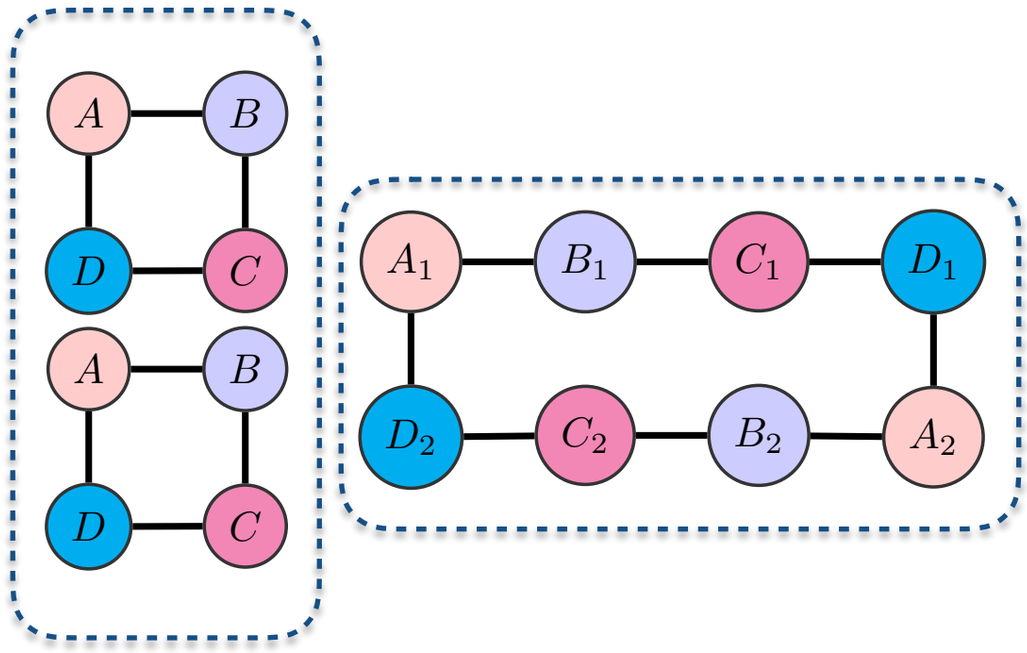
# Representational power of “GNNs”



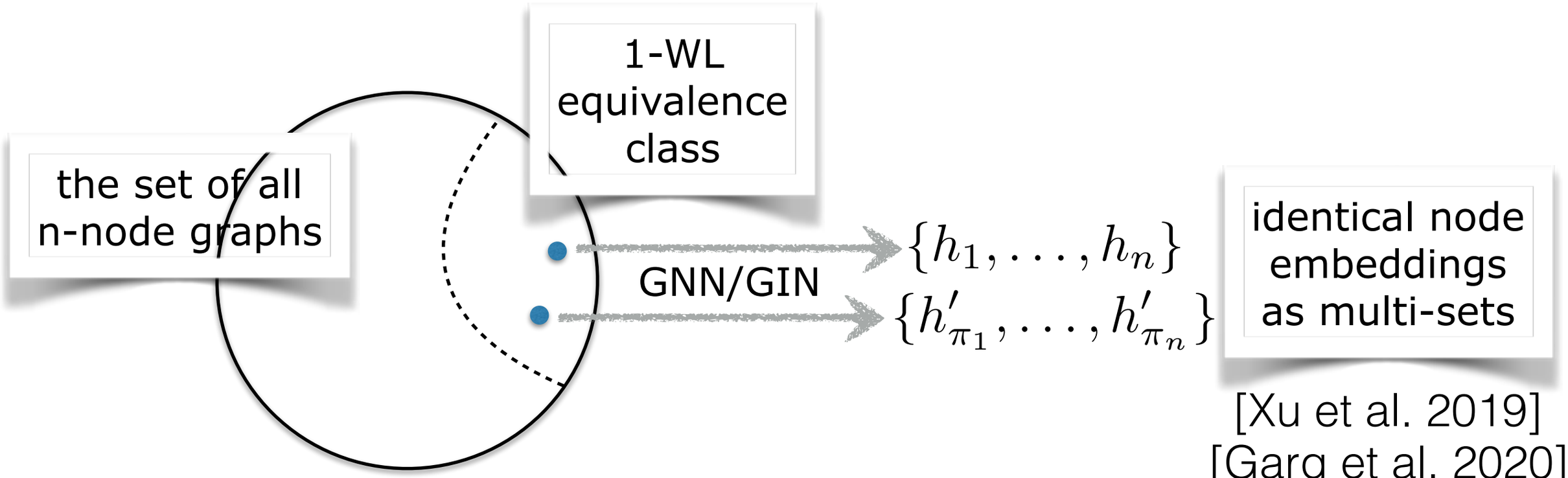



# Representational power of "GNNs"

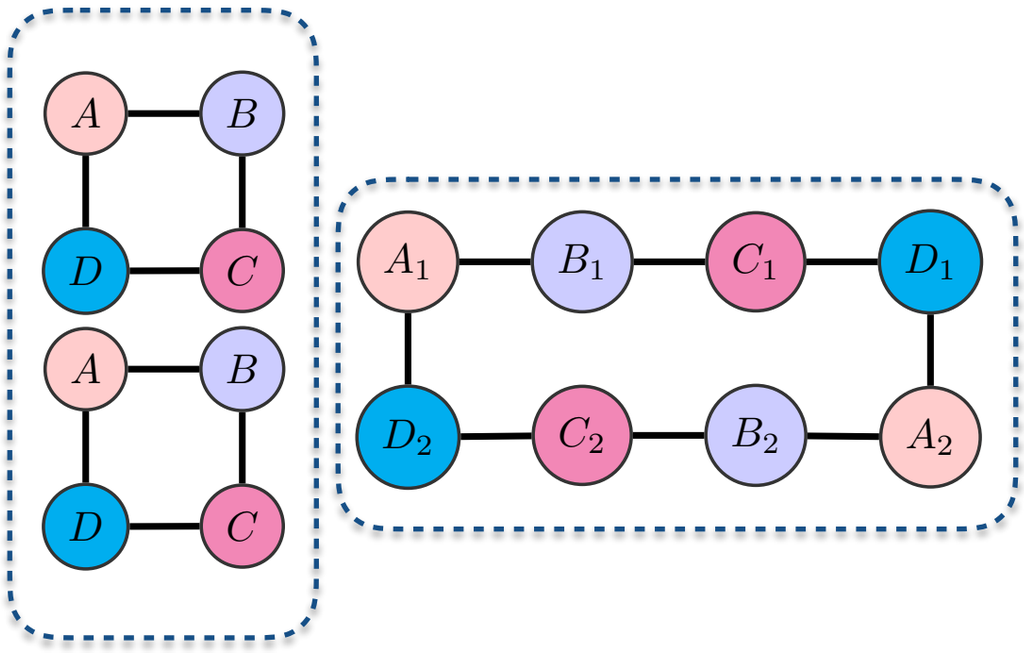


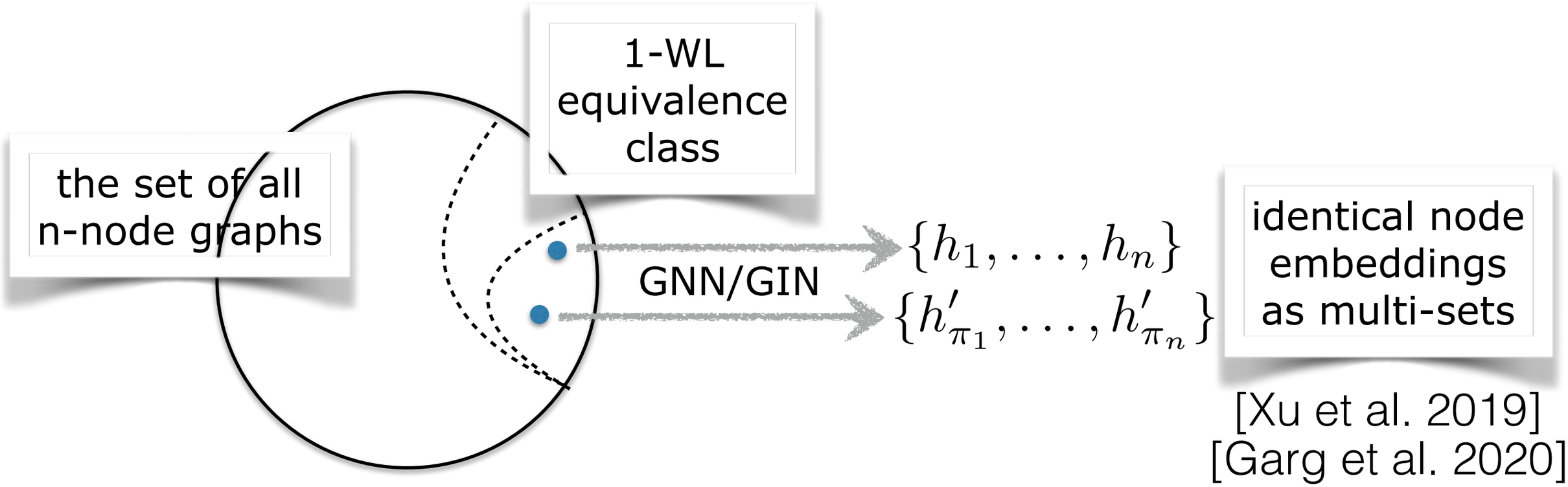
# Representational power of "GNNs"




[Xu et al. 2019]
[Garg et al. 2020]



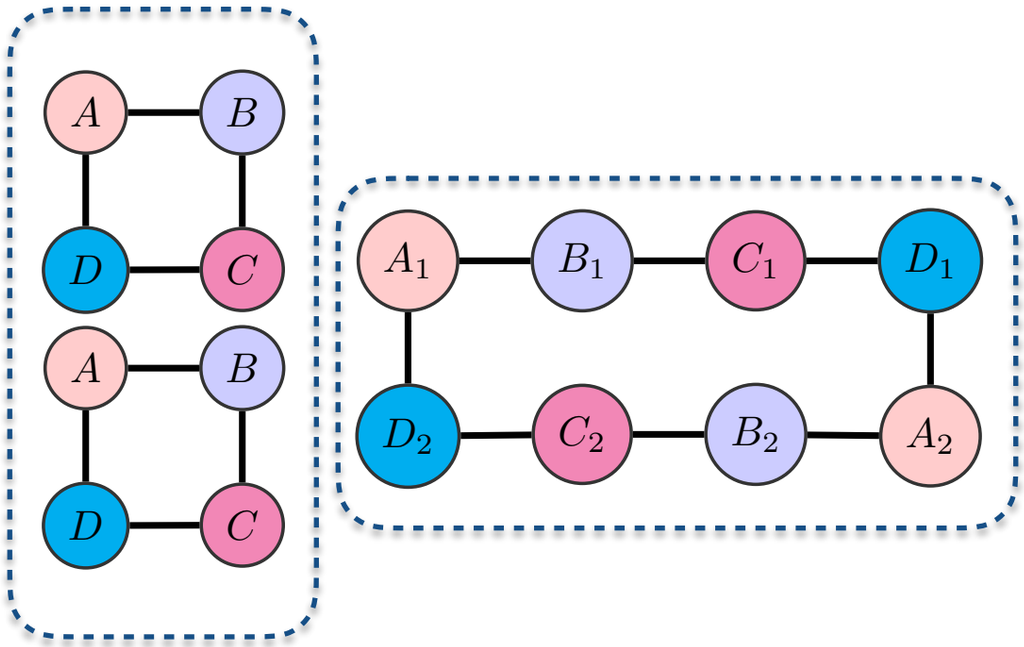
# Representational power of "GNNs"




[Xu et al. 2019]
[Garg et al. 2020]

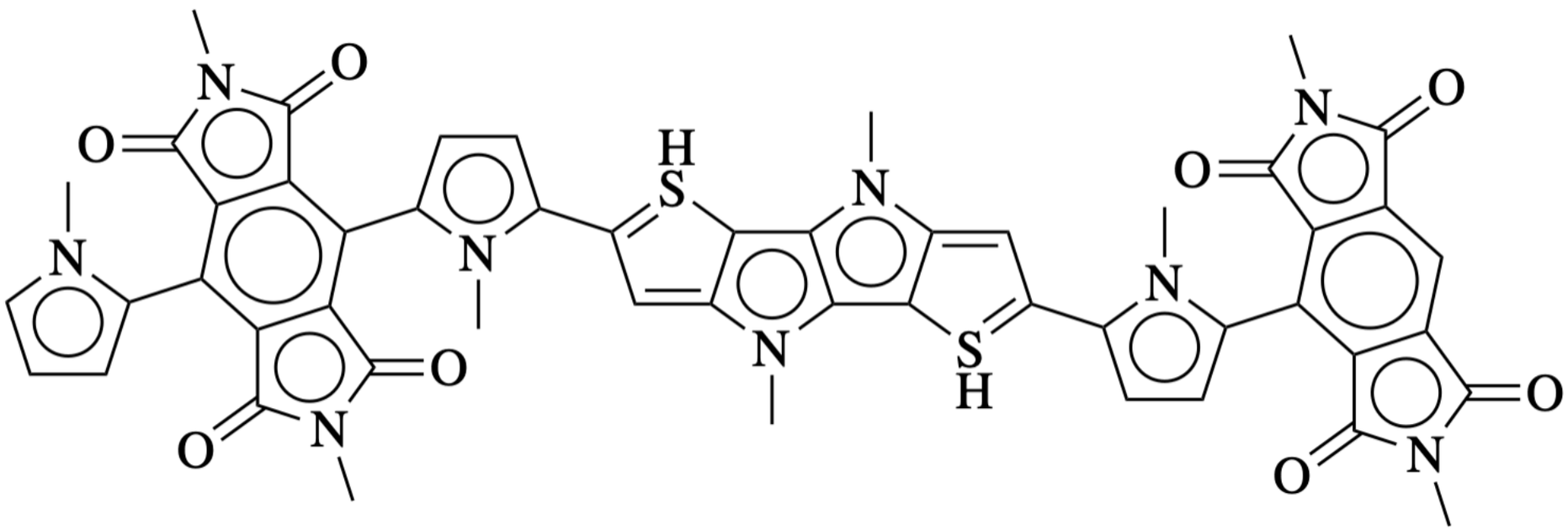
- Indistinguishable graph features
  - shortest/largest cycle, radius,
  - presence of conjoint cycle,
  - number of cycles, c-clique,
  - etc.

[Garg et al. 2020]



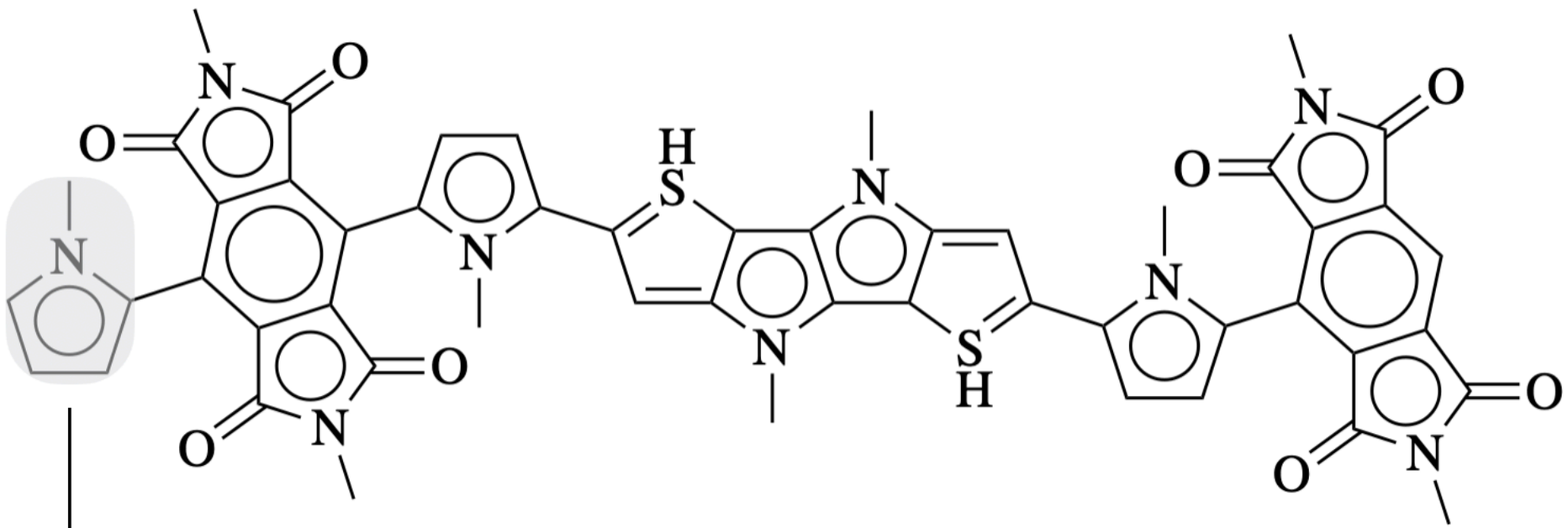
# Structural motifs: polymer

Molecular graph



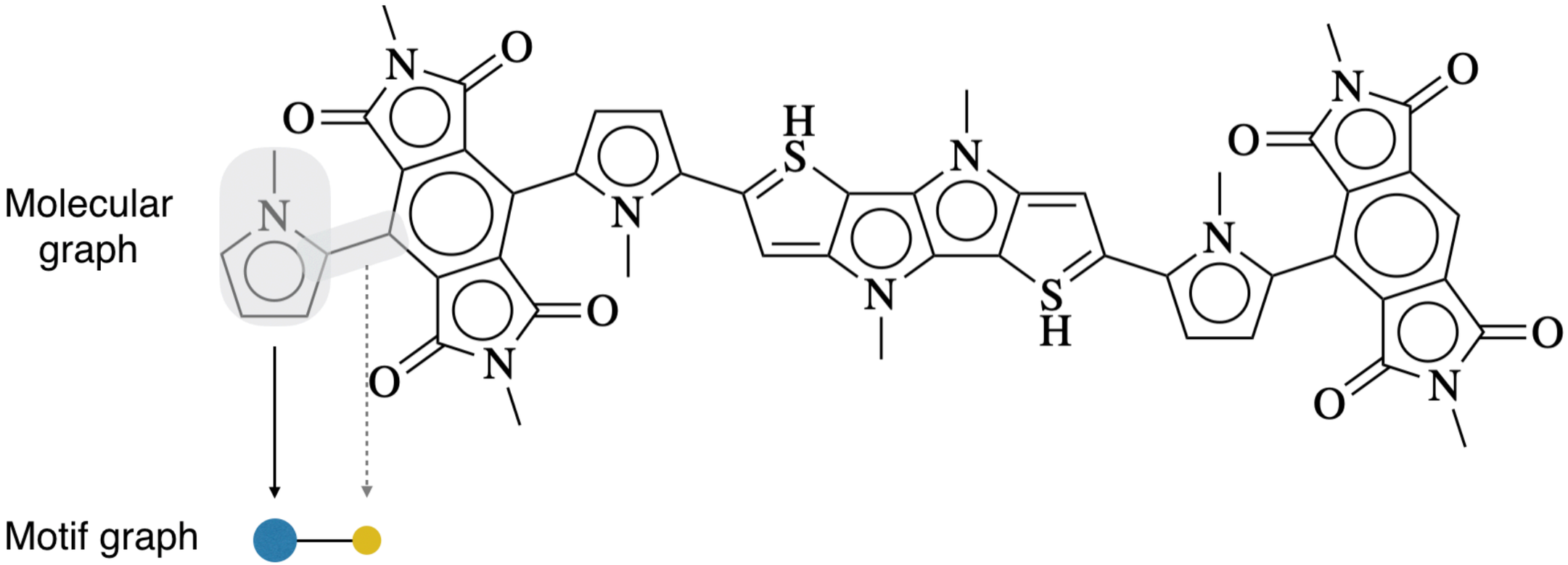
# Structural motifs: polymer

Molecular graph



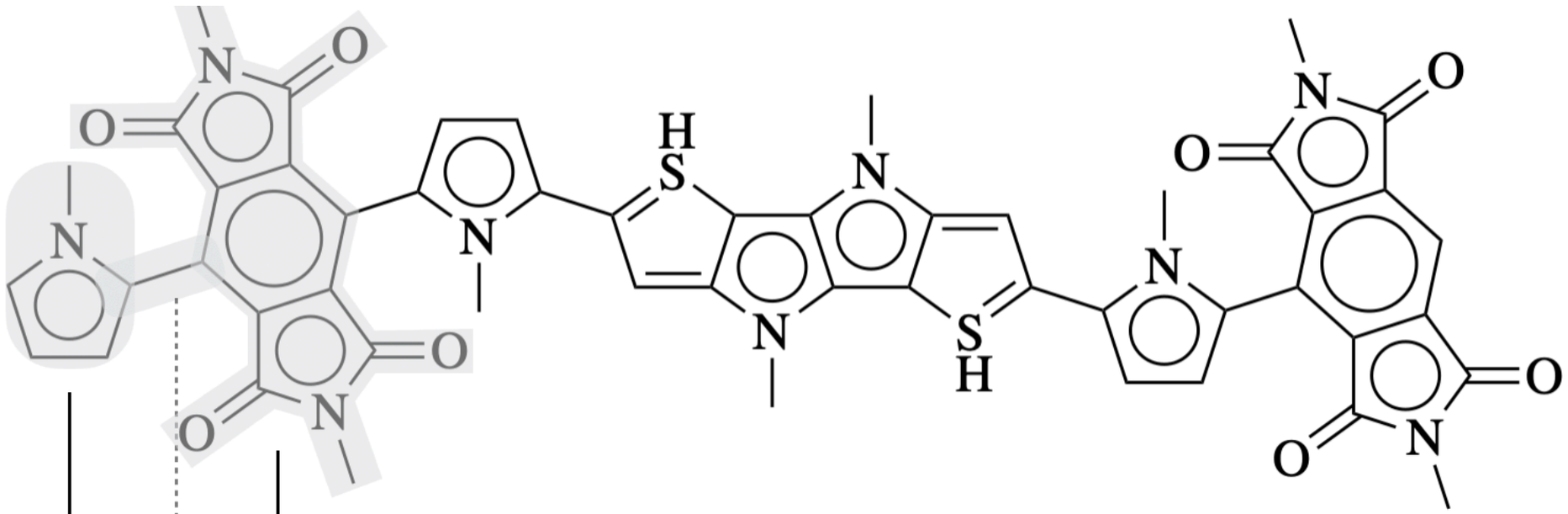
Motif graph

# Structural motifs: polymer



# Structural motifs: polymer

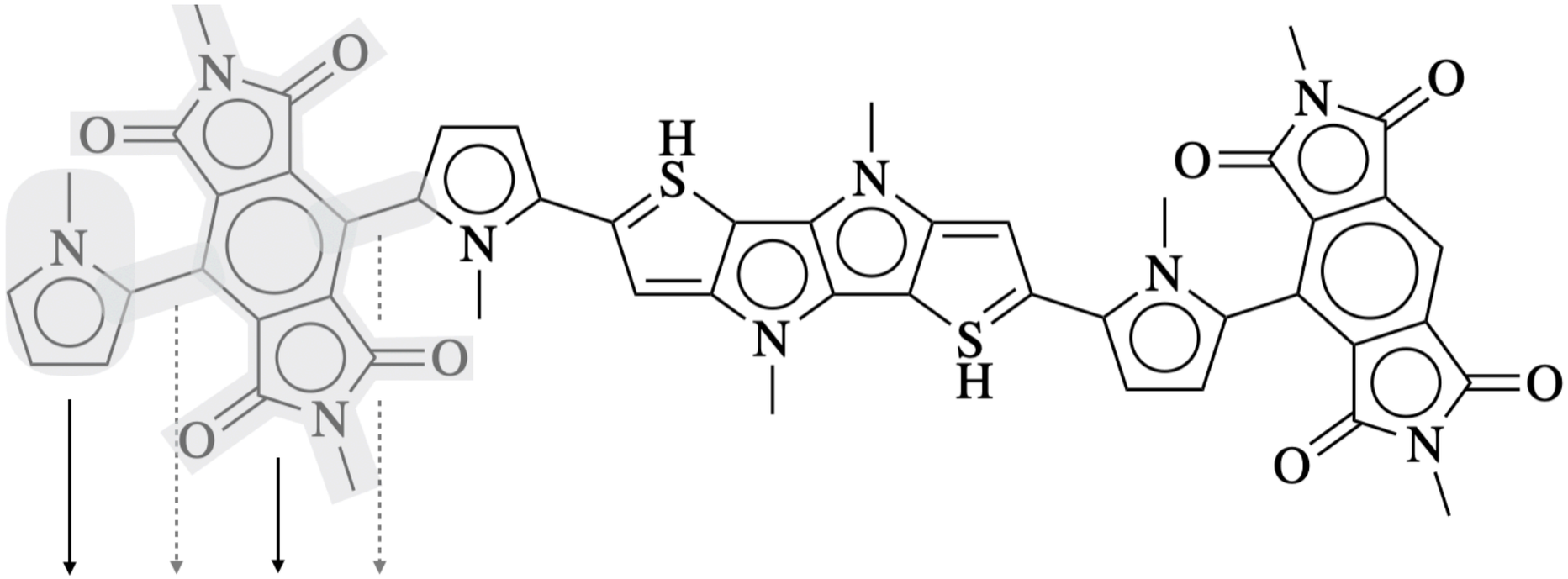
Molecular graph




Motif graph

# Structural motifs: polymer

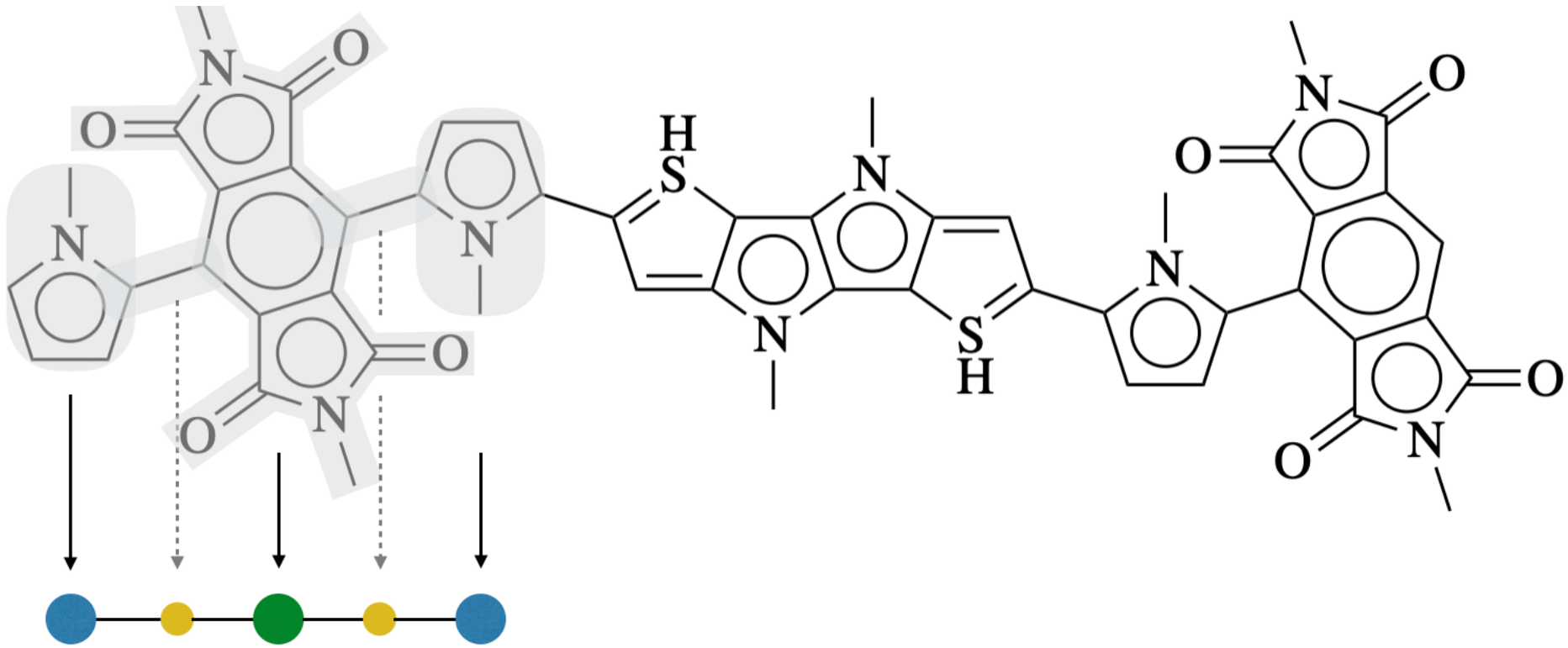
Molecular graph



Motif graph

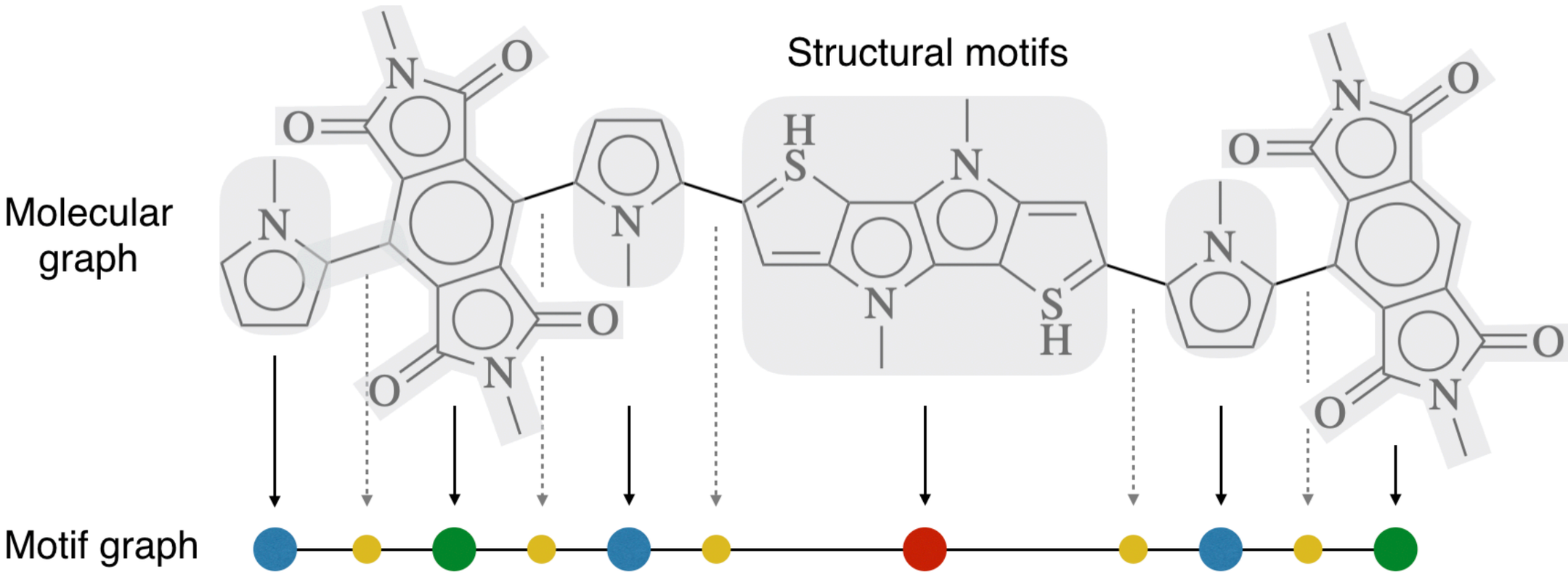
# Structural motifs: polymer

Molecular graph

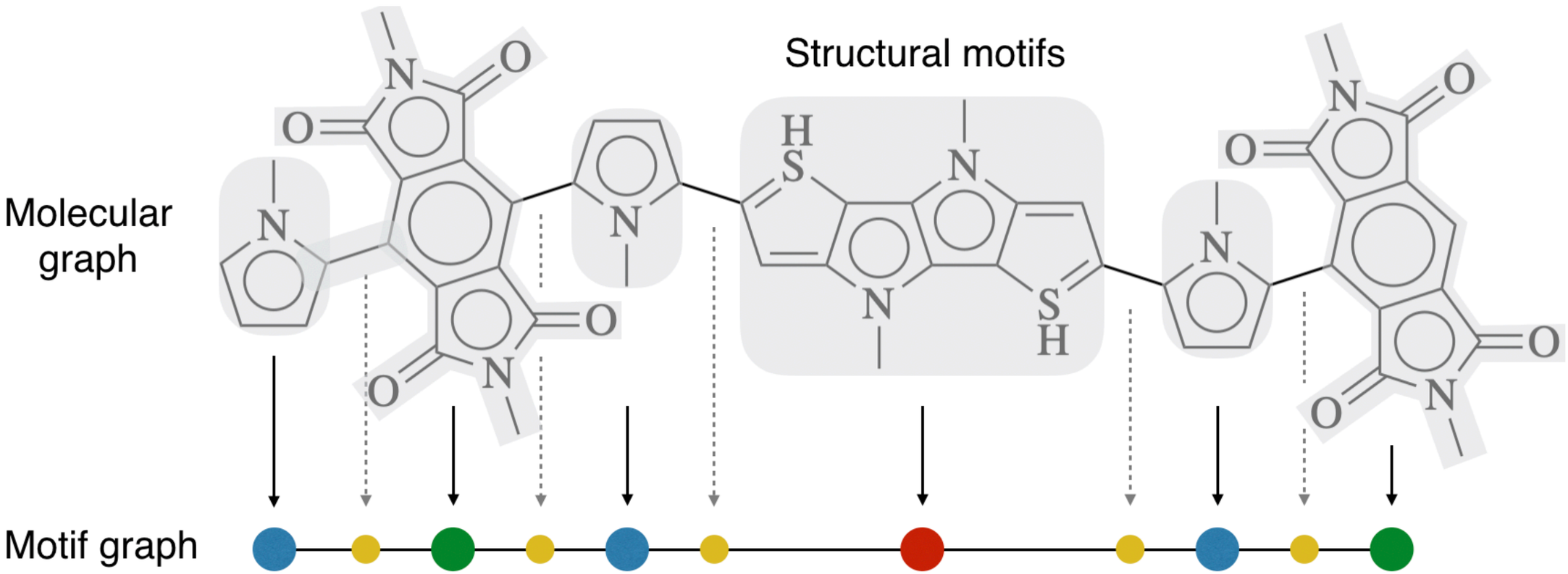


Motif graph

# Structural motifs: polymer

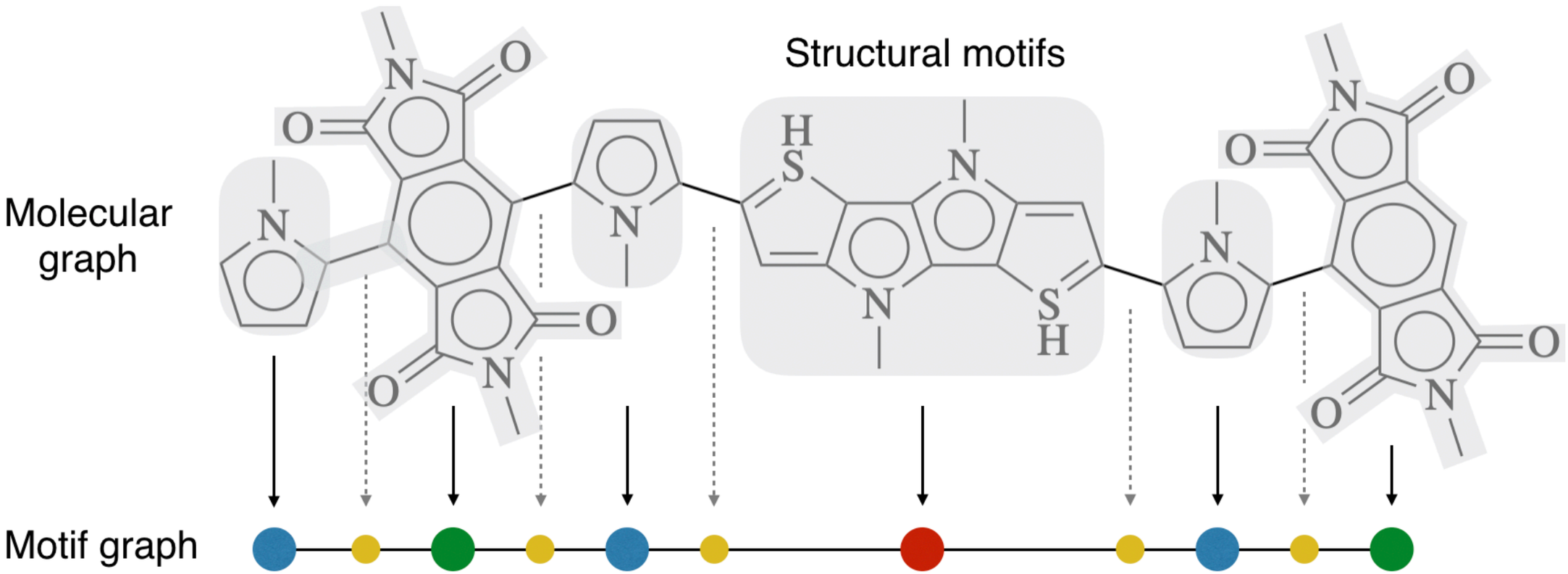


# Structural motifs: polymer



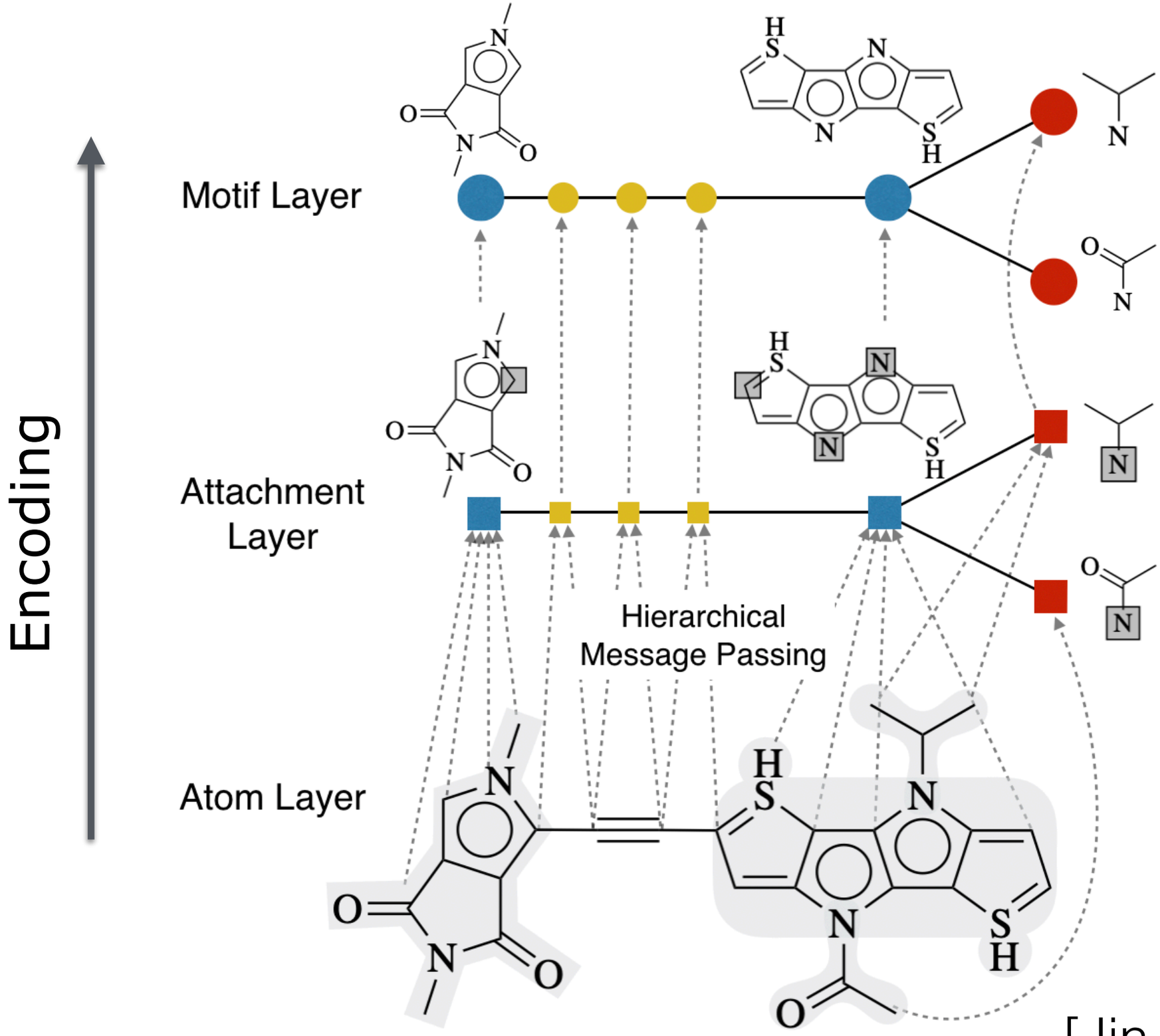

- This is a simple, two-level hierarchy; motif graph does not encode how the substructures are attached to each other

# Structural motifs: polymer




- This is a simple, two-level hierarchy; motif graph does not encode how the substructures are attached to each other
- We extend this to a three-level hierarchical representation for each molecule

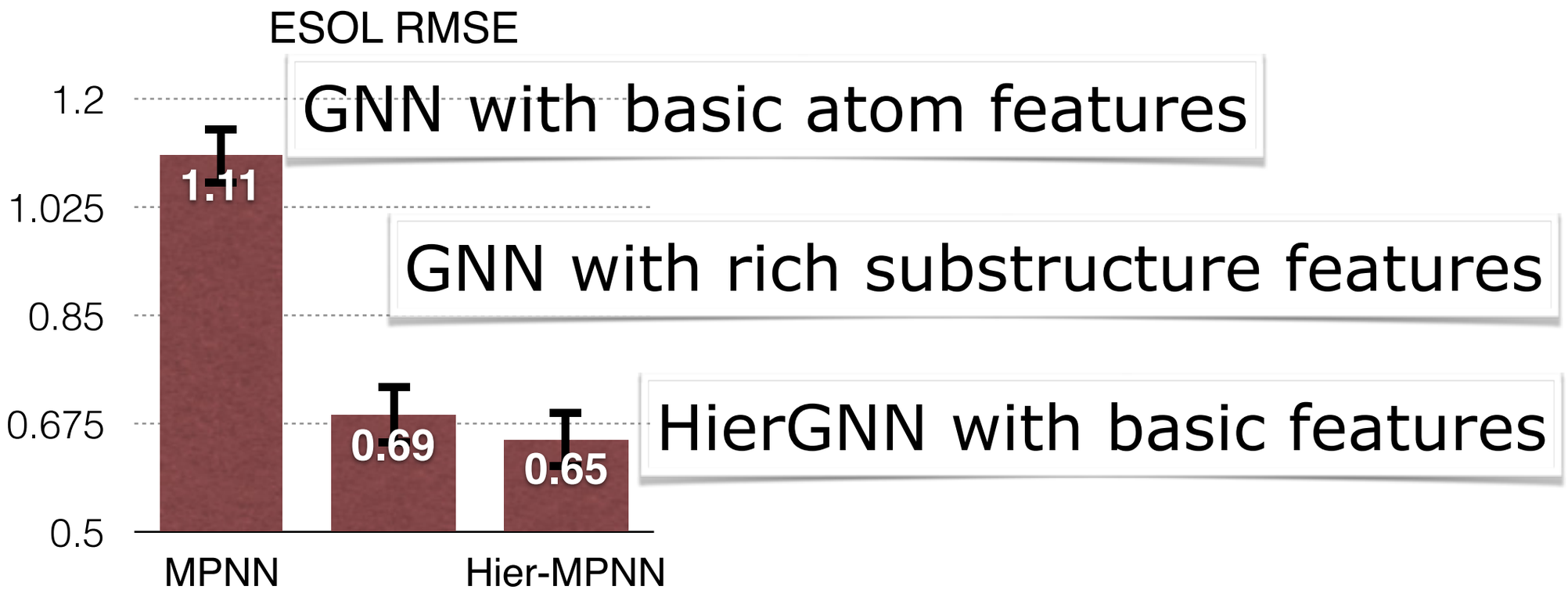
# Fine-to-coarse graph encoding




[Jin et al. 2020]

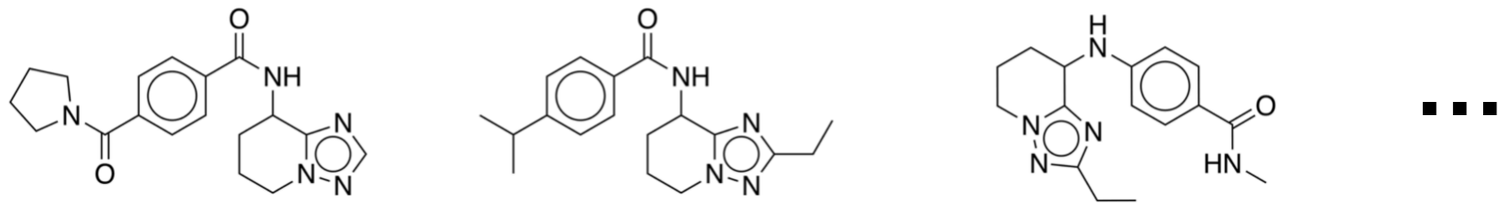
# Is hierarchy helpful?

- A simple example on solubility; ESOL dataset (averaged over 5 folds)



# New Antibiotic Discovery

- If we can accurately predict molecular properties, we can screen (select and repurpose) molecules from a large candidate set

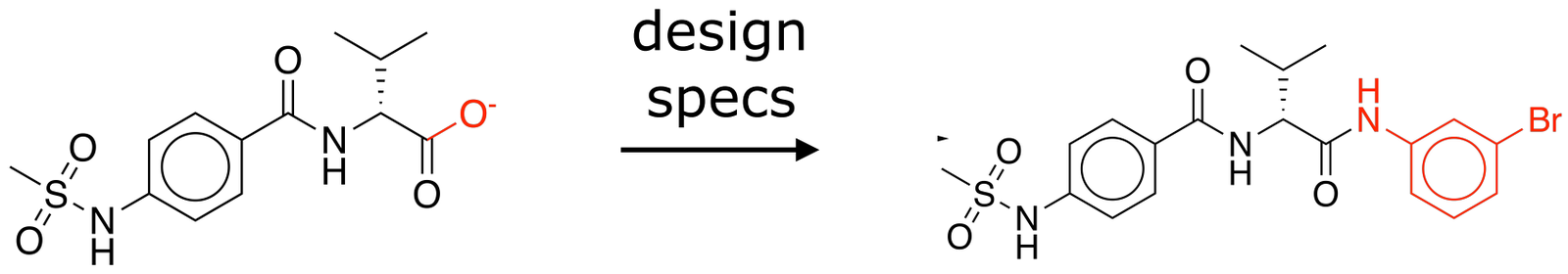



...

- Antibiotic Discovery [Stokes et al., 2020]
  - Trained a model to predict the inhibition against E. Coli
  - Data: ~2000 measured compounds from Broad Institute
  - Screened ~100 million possible compounds
  - Tested 15 highest scoring molecules in the lab
  - 7 of them validated to be inhibitive in-vitro

# Automating Drug design

- **Our problem:** how to programmatically modify pre-cursor molecules to have better properties




- **Key challenges:**

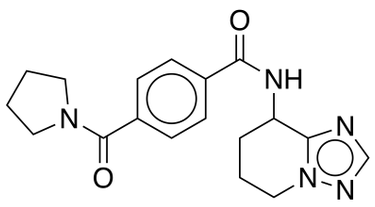
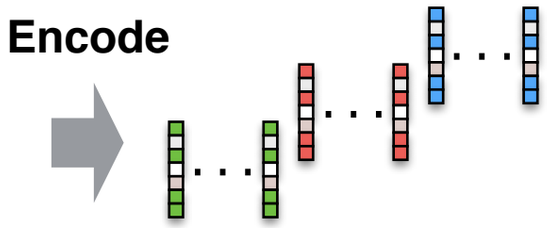
1. **representation and prediction:** learn to predict molecular properties

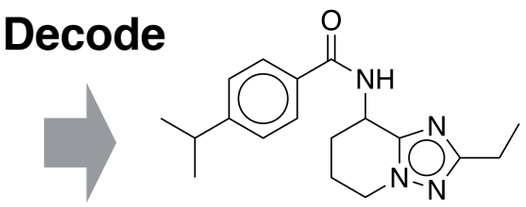
2. **generation and optimization:** realize target molecules with better properties programmatically

3. **understanding:** uncover principles (or diagnose errors) underlying complex predictions

# Optimization as graph translation

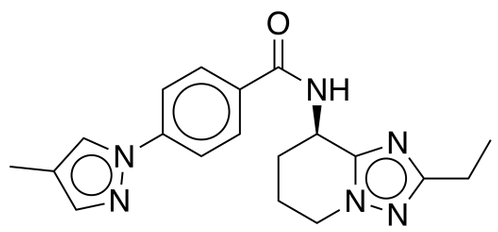
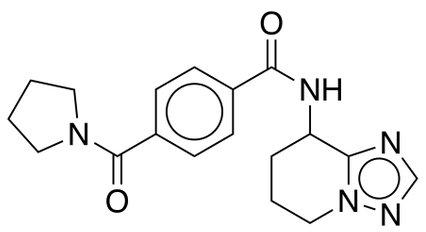
- **Goal:** learn to turn precursor molecules into molecules that satisfy given design specification(s)

Source $X$  



$Y$ Target

- The training set consists of (source, target) molecular pairs

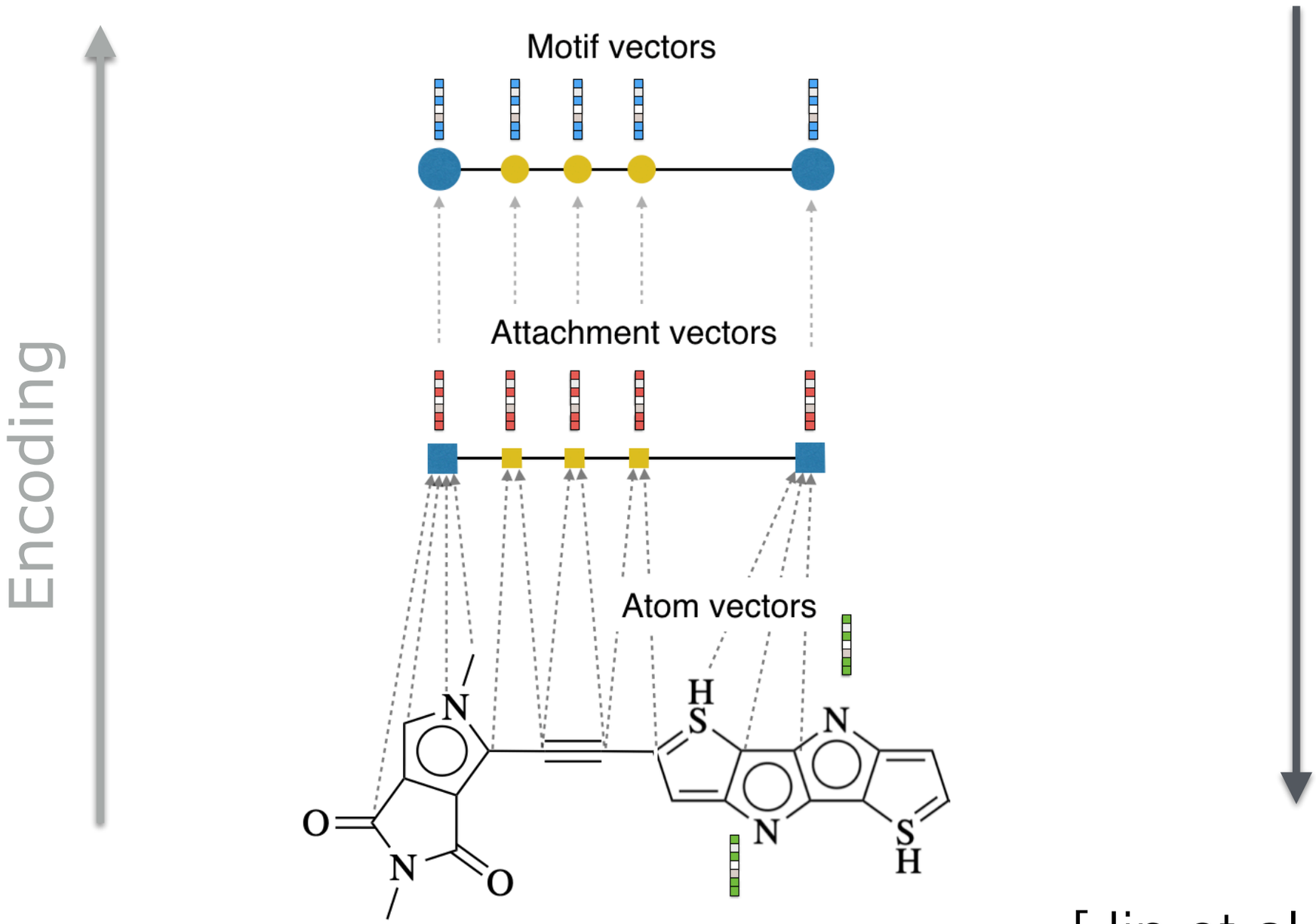
Source   Target

... ...

- **Key challenge:** molecule generation

# Coarse-to-fine graph generation

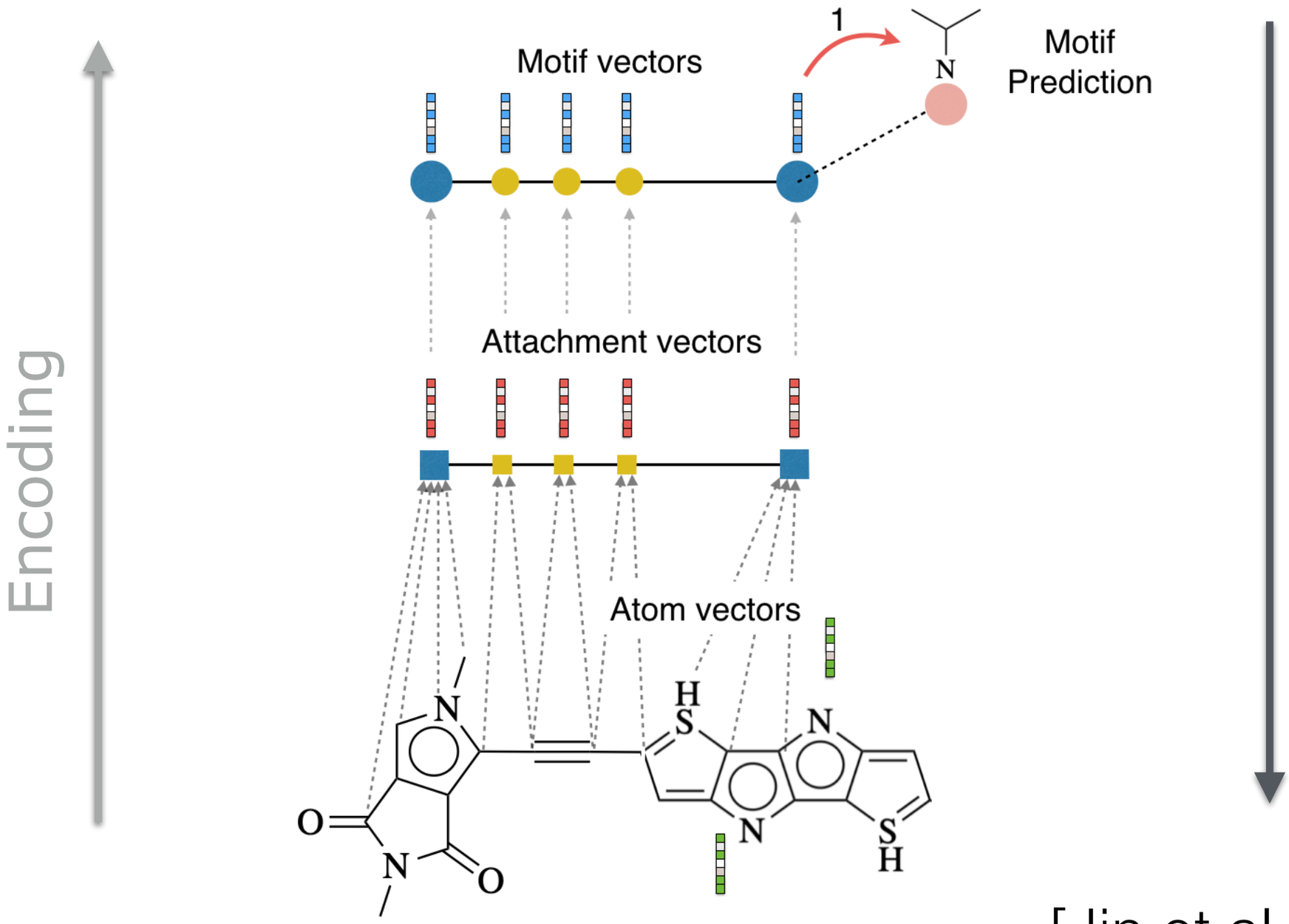
- We realize graphs auto-regressively, in a course to fine manner, one substructure at a time




[Jin et al. 2020]

# Coarse-to-fine graph generation

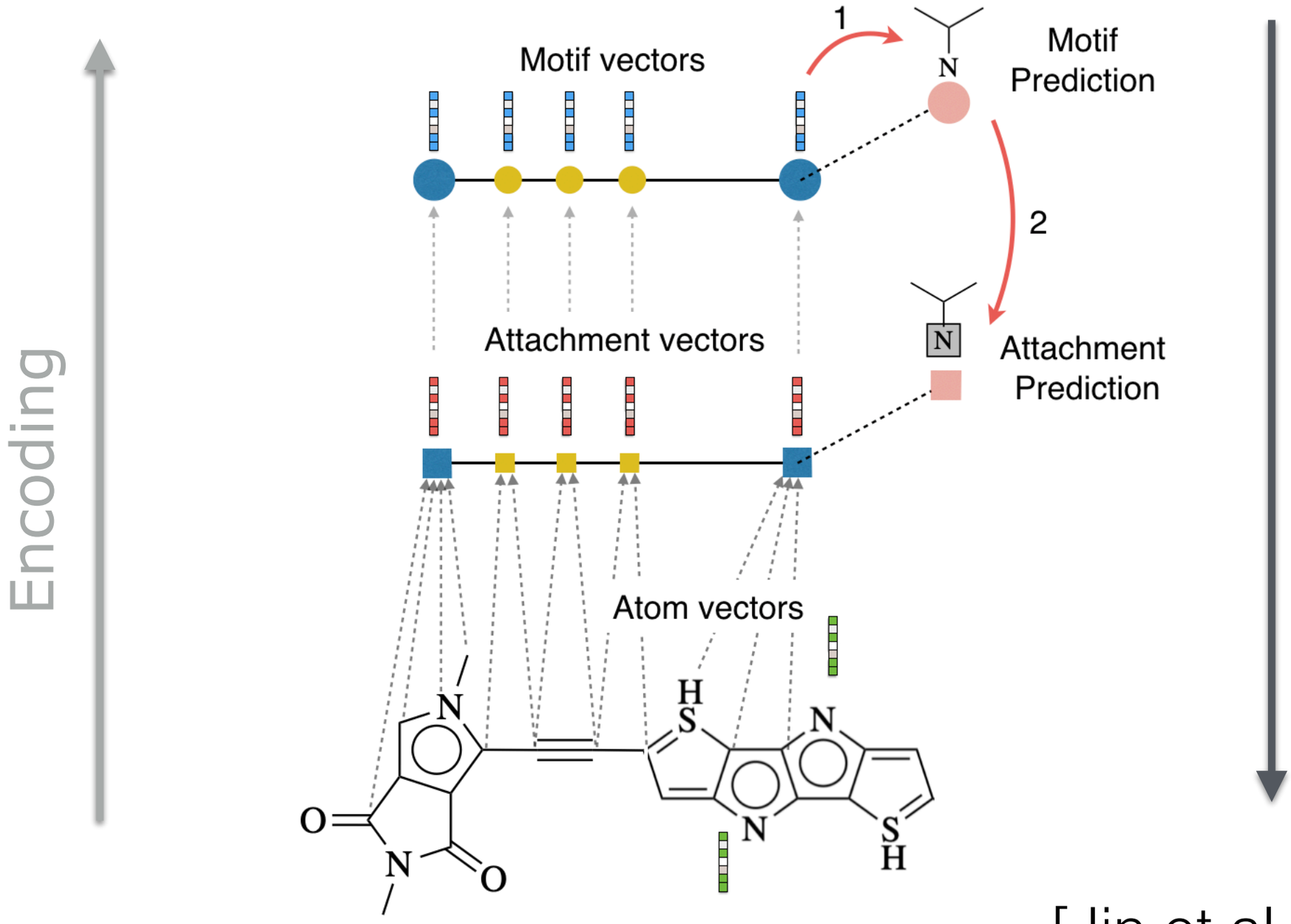
- We realize graphs auto-regressively, in a course to fine manner, one substructure at a time




[Jin et al. 2020]

# Coarse-to-fine graph generation

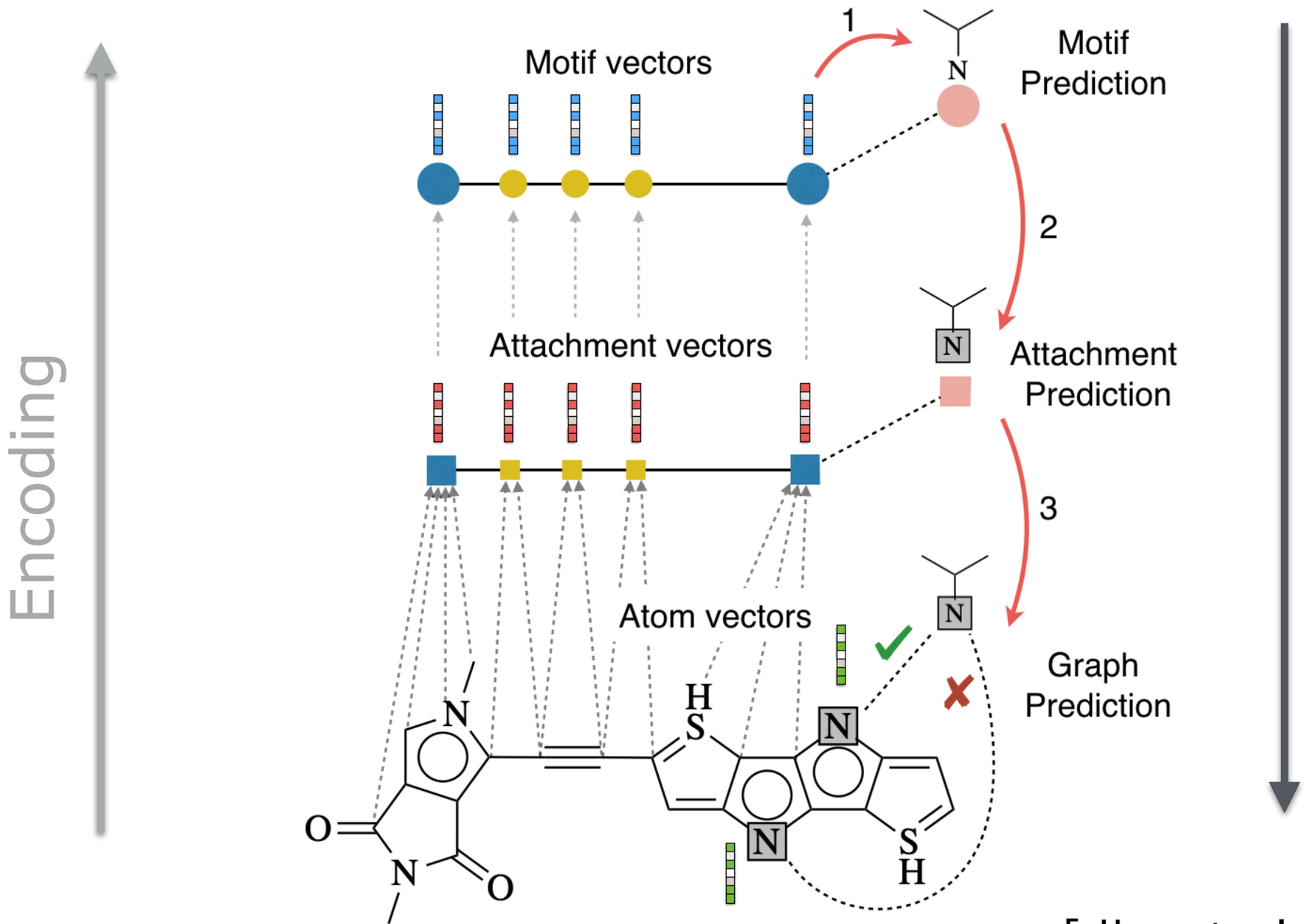
- We realize graphs auto-regressively, in a course to fine manner, one substructure at a time




[Jin et al. 2020]

# Coarse-to-fine graph generation

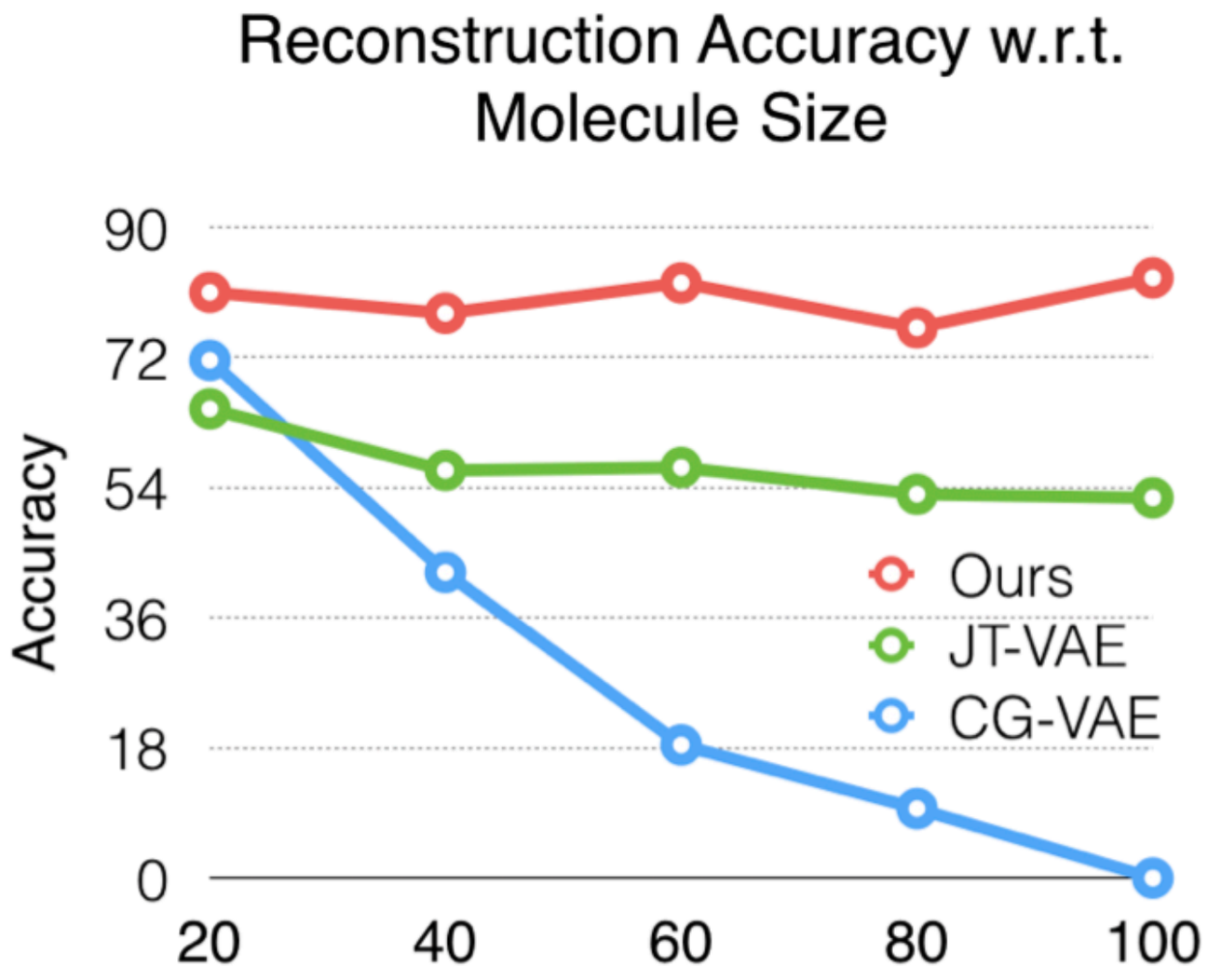
- We realize graphs auto-regressively, in a course to fine manner, one substructure at a time




[Jin et al. 2020]

# Does the hierarchy help?

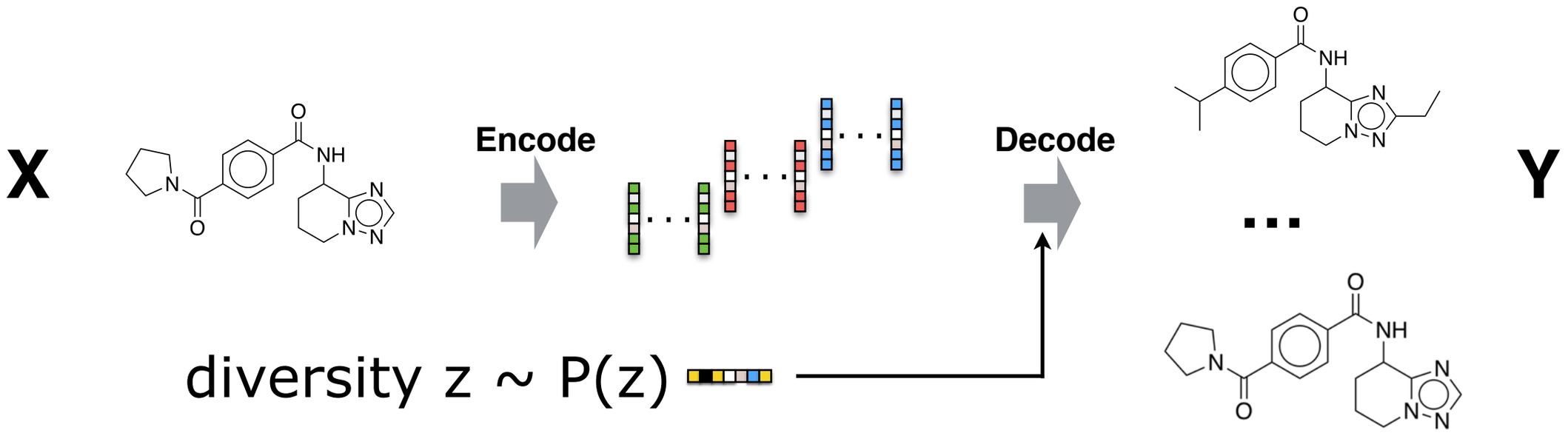
- Example with polymers: 86K (76K+5K+5K)




[Jin et al. 2020]

# graph generation: diversity

- **Goal:** learn to turn precursor molecules into molecules that satisfy given design specification(s)

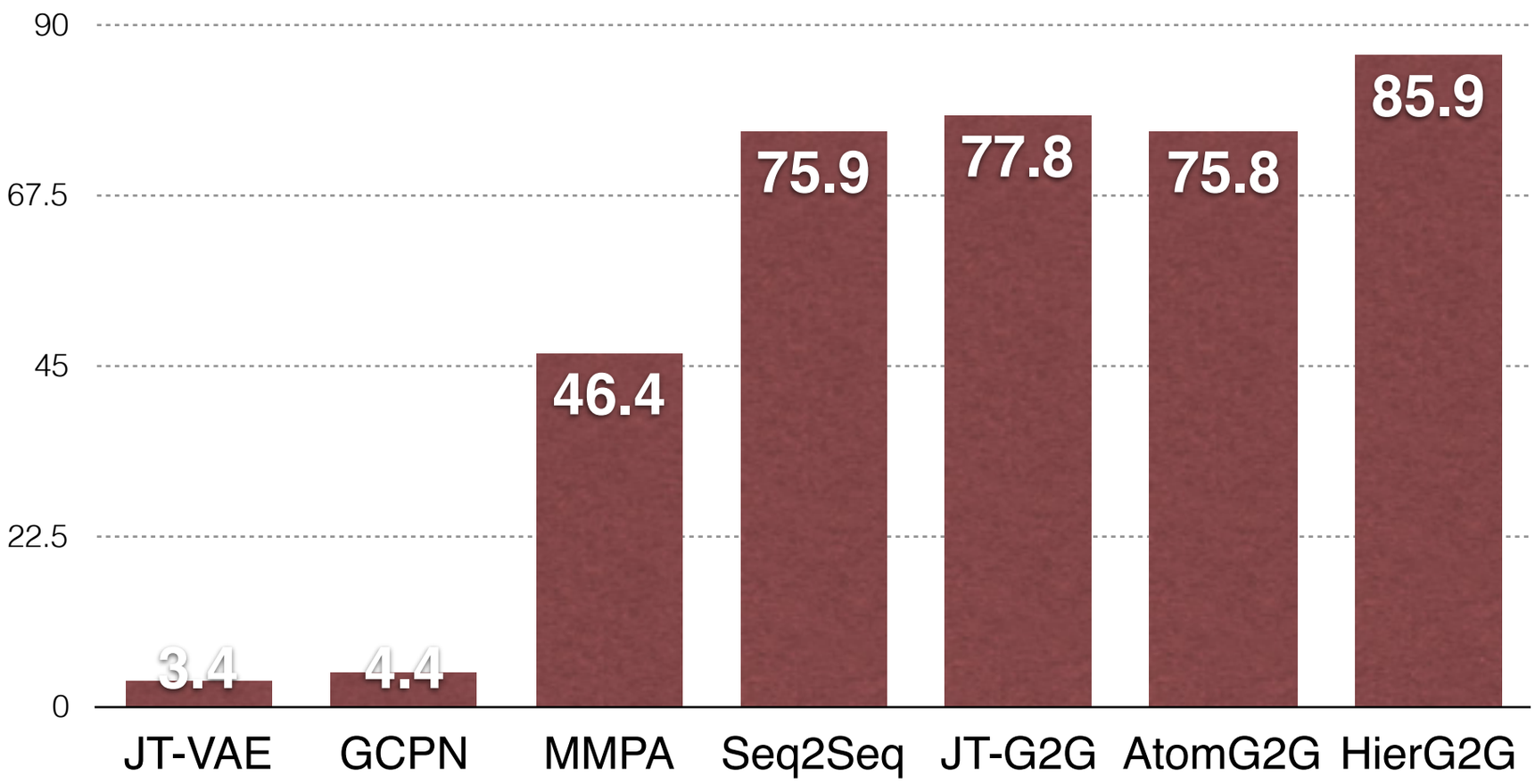



- We'd like to generate a diverse set of candidate molecules that satisfy the criteria

[Jin et al. 2019,2020]

# Example results

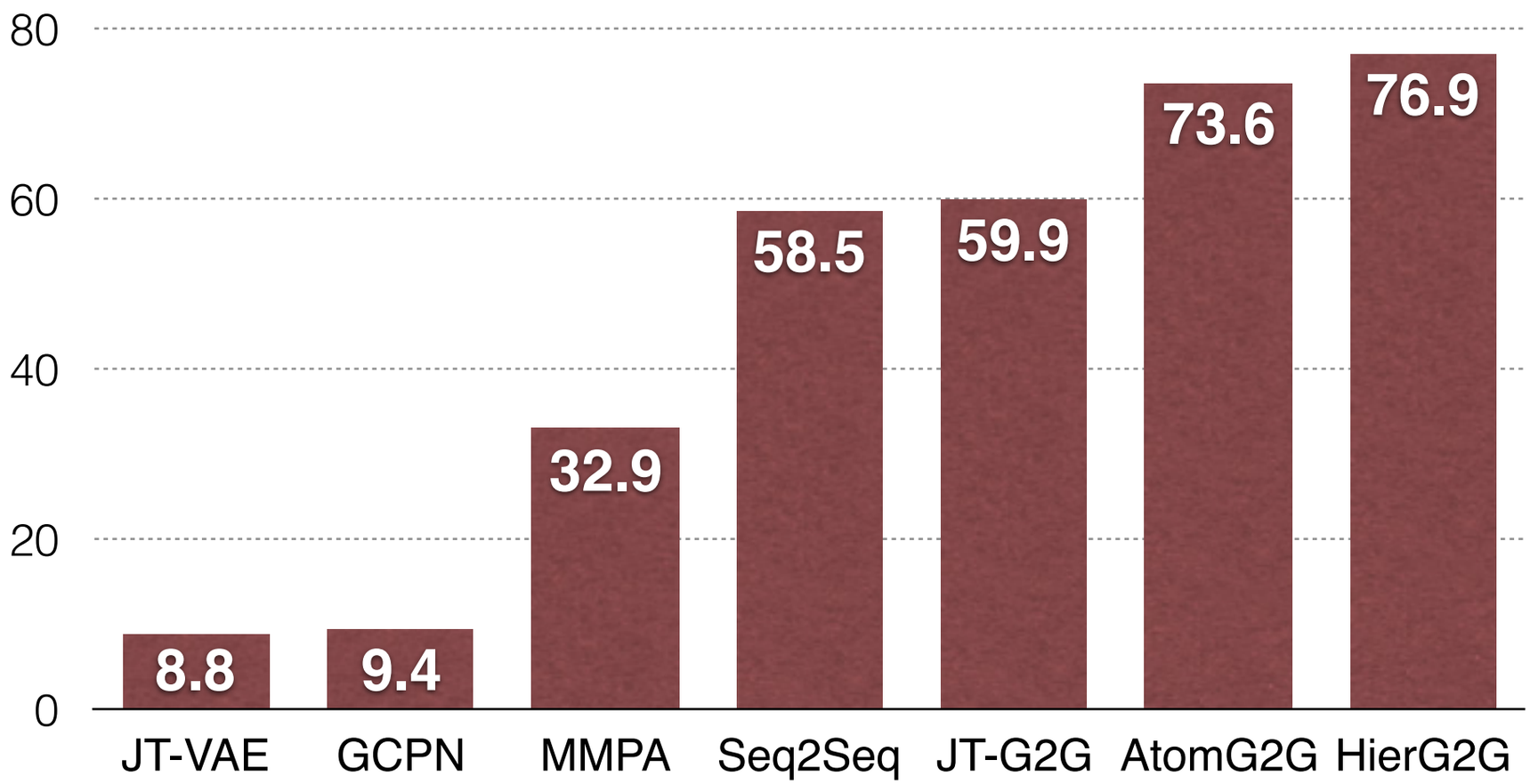
- Single property optimization: DRD2 success % (from inactive to active)




[Jin et al. 2020]

# Example results

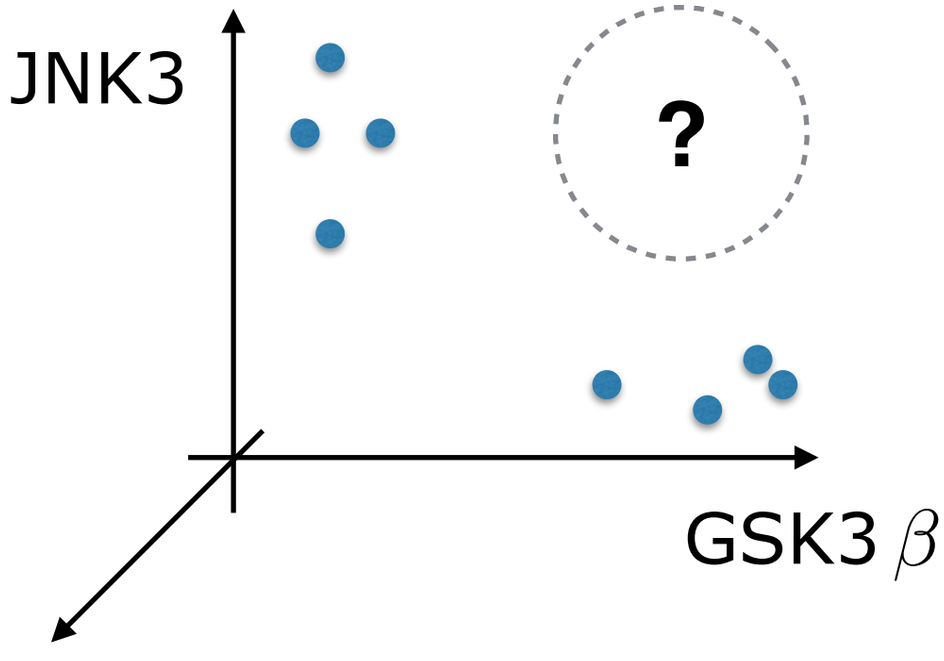
- Single property optimization: QED success % (QED > 0.9)




[Jin et al. 2020]

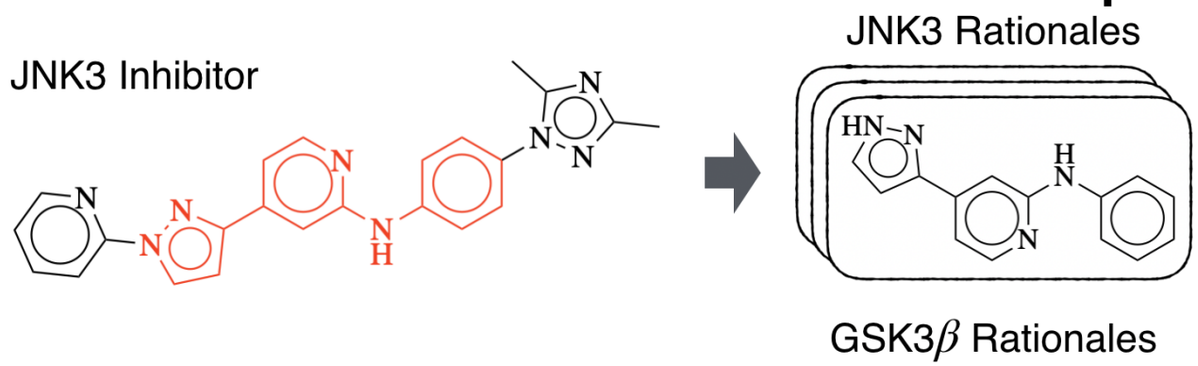
# Inverse design challenge

- **Many** examples of molecules with a particular property
- **Few** instances of molecules that satisfy multiple (esp. new) property combinations
- **Challenge:** How do we realize a diverse distribution of molecules that satisfy all the criteria without any examples of such molecules?



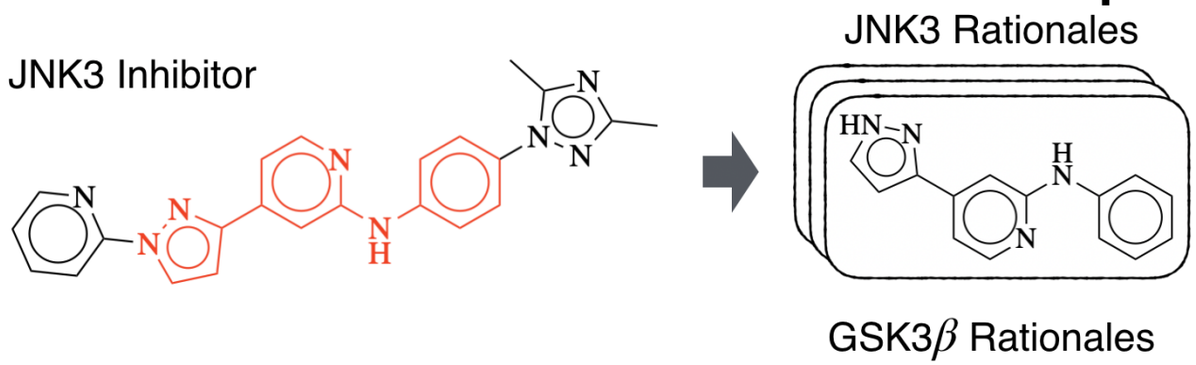
# Our strategy

- **Step 1: rationale extraction**
  - carve out candidate substructures — rationales — likely responsible for each molecular property

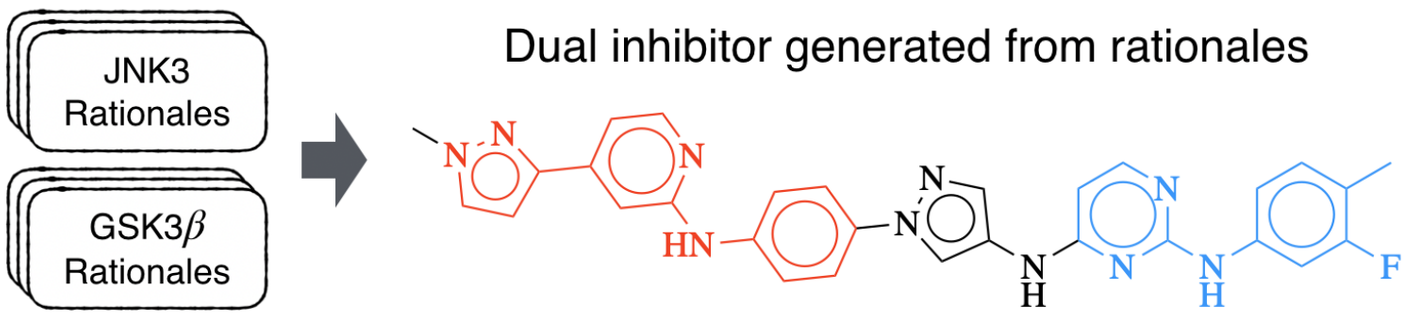


# Our strategy

- **Step 1: rationale extraction**
  - carve out candidate substructures — rationales — likely responsible for each molecular property



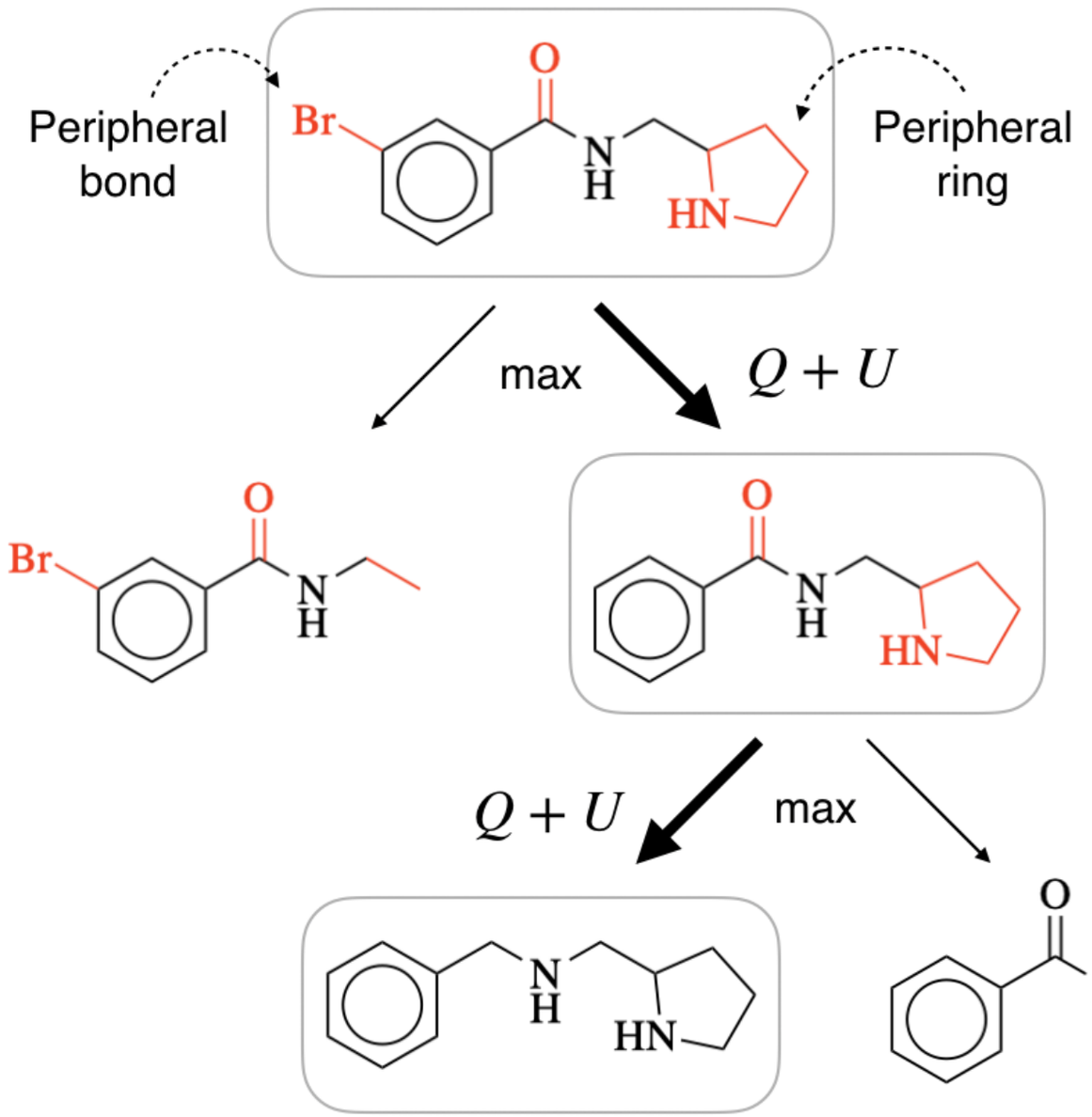

- **Step 2: multi-rationale assembly**
  - learn to assemble pieces together into a complete molecule that satisfies all the properties




[Jin et al. 2020]

# Step 1: rationale extraction

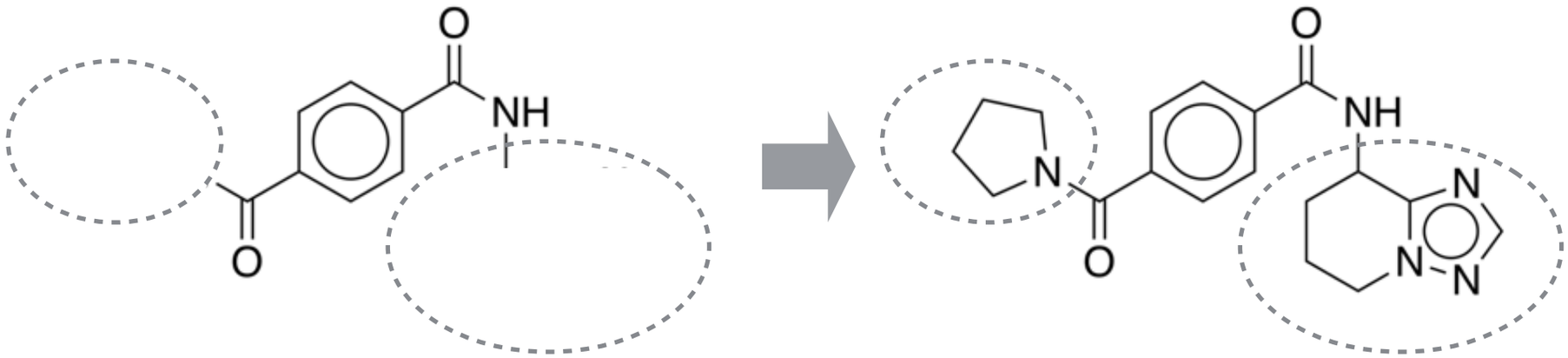
- We can use Monte Carlo Tree Search to remove all parts of the molecule not relevant for the property (according to a given property predictor)




[Jin et al. 2020]

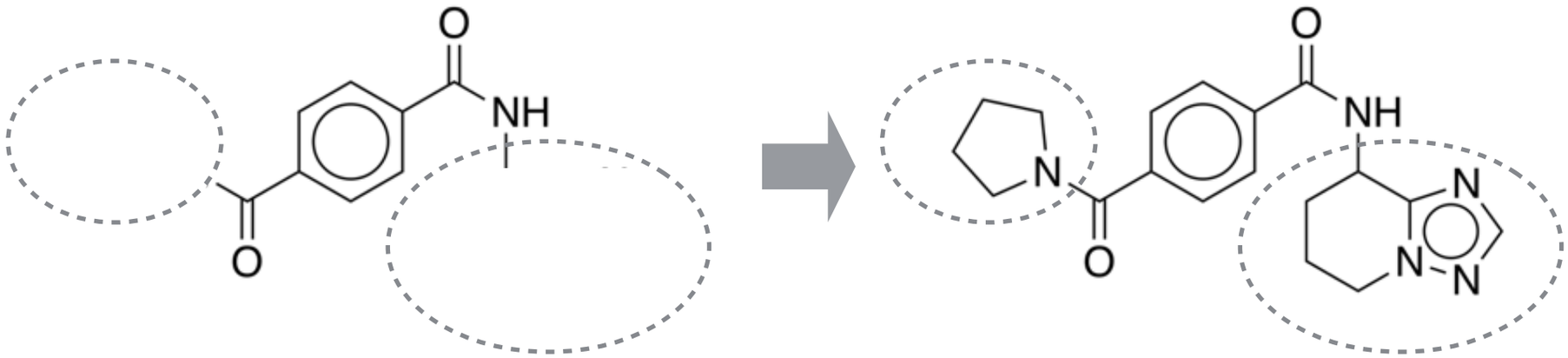
# Step 2: multi-rationale assembly

- **Pre-train assembler:** learn a graph completion model P(G|S) that can expand any substructure S to a complete molecule G



# Step 2: multi-rationale assembly

- **Pre-train assembler:** learn a graph completion model P(G|S) that can expand any substructure S to a complete molecule G




- **Fine tune multi-rationale completions:** we use RL to optimize sample completions towards satisfying all the properties

$$S_{1:k} \sim P(S_{1:k})$$ sample a candidate rationale set

$$G \sim P(G|S_{1:k})$$ sample graph completion

$$\text{reward} = \begin{cases} 1, & \text{if } G \text{ has all the properties} \\ 0. & \text{otherwise} \end{cases}$$ evaluate

[Jin et al. 2020]

# Example results

- Example comparison of RL based multi-criteria optimization methods

| Method | DRD2 + GSK3$\beta$ + JNK3 | | |
|---|---|---|---|
| | Success | Novelty | Diversity |
| GVAE + RL | 0% | 0% | 0.0 |
| GCPN | 0% | 0% | 0.0 |
| REINVENT | 48.3% | **100%** | 0.166 |
| Ours | **86.2%** | **100%** | **0.726** |

[Jin et al. 2020]

# Summary

- Molecules embody many of the key challenges in prediction/generation/manipulation of complex objects

- While molecular design methods are rapidly becoming viable tools for drug discovery, many challenges remain:
  - generalizing predictions to unexplored chemical spaces
  - incorporating 3D features, physical constraints
  - explainability
  - etc.