Molecular Hypergraph Grammar with Its Application to Molecular Optimization

Hiroshi Kajino

MIT-IBM Watson AI Lab
IBM Research - Tokyo
We wish to learn a generative model of a molecule.

- **Generative model of a molecule** $p(G \mid z)$ [Gómez-Bombarelli+, 16]
  - **Input**: Latent vector $z \in \mathbb{R}^D \sim \mathcal{N}(0, I_D)$
  - **Output**: Molecular graph $G$ (graph w/ node labels)

Continuous optimization problem $\iff$ Molecular optimization problem

**Background**

Continuous latent sp. $\quad p(G \mid z) \quad$ Discrete molecular graph sp.
Molecular graph generation is a non-trivial task

- **Two technical challenges**

  1. No consensus on a generative model of a graph
     - LSTM, GRU for path and tree
     - Ring is non-trivial
  
  2. Hard constraints such as valence conditions
     - Degree of each node (=atom) is specified by its label
     - e.g., carbon has degree 4, oxygen has degree 2.
Most of the existing work employs a text representation of a molecule, called SMILES.

- **Pros:** a standard sequential model can generate SMILES

![Diagram](image_url)
Statistical model has to learn a rule-based grammar

- **Use text representation "SMILES"** [Gómez-Bombarelli+, 16]
  - **Pros:** a standard sequential model can generate SMILES
  - **Cons:**
    - NN has to learn SMILES’ grammar
    - No guarantee on valence conditions

Existing work 1/3

CC(C)(O)C#Cc1ccc(C[NH2+][C@H]2 CCCN(c3nc4ccccccc4s3)c2)s1
SMILES’ grammar does not prescribe valence conditions

- **Use text representation "SMILES"** [Gómez-Bombarelli+, 16]
  - **Pros:** a standard sequential model can generate SMILES
  - **Cons:**
    - NN has to learn SMILES’ grammar
    - No guarantee on valence conditions

This atom has a valence of 5. That of carbon must be 4.
JT-VAE achieves 100% validity for the first time, but requires multiple NNs.

- Generate a molecule by assembling subgraphs [Jin+, 18]

Enumerate all possible combinations, and predict the best using NN 😱

Image from [Jin+, 18] © 2019 IBM Corporation
Existing work 3/3

RL-based method discovers better molecules than VAE-based methods, but with enormous cost

- **Molecular optimization using RL** [You+, 18]

  - **Idea:** Molecular generation as MDP
    - State: Molecular graph
    - Action: Modify the graph
    - Reward: Target property

  - **Pros:** Better optimization capability than VAE-based methods
  
  - **Cons:** Requires a number of target property evaluations
    - # of evals $> 10^9$
    - Infeasible to work with the first principle calculation / wet-lab experiments
No existing work satisfies all of the three properties

<table>
<thead>
<tr>
<th></th>
<th>VAE-based</th>
<th>RL-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMILES Validity</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Graph</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Easy-to-generate ✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sample complexity ✔</td>
<td>✔</td>
<td></td>
</tr>
</tbody>
</table>
Our contribution is to facilitate graph-based generation with help of "graph grammar".

<table>
<thead>
<tr>
<th></th>
<th>VAE-based</th>
<th>RL-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMILES</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Validity</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Easy-to-generate</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Sample complexity</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>
Our work

We develop a graph grammar tailored for molecular generation

- **Idea**
  
  Use a context-free graph grammar for graph generation
  
  - Graph generation boils down to *tree generation*

- **Requirements:**
  
  - Always satisfy valence conditions
  - Context-freeness
  - Inference algorithm from data, not hand-written rules

Our main contribution

Hyperedge replacement grammar
A hyperedge replacement grammar can be constructed from data via tree decomposition

Existing work on graph grammar

[Aguiñaga+, 16]

Set of general hypergraphs

Tree decomposition

Hyperedge Replacement Grammar

* HRG is a context-free grammar generating hypergraphs
Each restriction is our contribution in the literature of graph grammar.

**Existing work on graph grammar**  
[Aguiñaga+, 16]

- Set of general hypergraphs
  - Tree decomposition
    - Hyperedge Replacement Grammar

**Our method**

- Set of molecular hypergraphs
  - Irredundant tree decomposition
    - Molecular Hypergraph Grammar

* HRG is a context-free grammar generating hypergraphs
Each restriction is our contribution in the literature of graph grammar.

**Existing work on graph grammar**

[Aguiñaga+ 16]

- Set of general hypergraphs
  - Tree decomposition
    - Hyperedge Replacement Grammar

**Our method**

- Set of molecular hypergraphs
  - Irredundant tree decomposition
    - Molecular Hypergraph Grammar

* HRG is a context-free grammar generating hypergraphs
Hypergraph is a generalization of a graph

- **Hypergraph** $\mathcal{H} = (V, E)$ consists of...
  - Node $v \in V$
  - Hyperedge $e \in E \subseteq 2^{|V|}$: Connect an arbitrary number of nodes
    
    cf, An edge in a graph connects exactly two nodes
We represent a molecule using a hypergraph, not a graph. This helps to satisfy the valence conditions.

- **Molecular hypergraph models...**
  - Atom = hyperedge
  - bond = node

- **Molecular graph models...**
  - Atom = node
  - Bond = Edge
These requirements guarantee transformation between graph & hypergraph

- **Molecular hypergraph**
  
  1. Each node has degree 2 (=2-regular)
  2. Label on a hyperedge determines # of nodes it has (= valence)
Our work

Each restriction is our contribution in the literature of graph grammar

Existing work on graph grammar

[Aguiñaga+, 16]

- Set of general hypergraphs
- Tree decomposition
- Hyperedge Replacement Grammar

Our method

- Set of molecular hypergraphs
- Irredundant tree decomposition
- Molecular Hypergraph Grammar

* HRG is a context-free grammar generating hypergraphs
Hyperedge replacement grammar & Molecular hypergraph grammar

HRG generates a hypergraph by repeatedly replacing non-terminal hyperedges with hypergraphs

- Hyperedge replacement grammar (HRG) $G = (N, T, S, P)$
  - $N$: set of non-terminals
  - $T$: set of terminals
  - $S$: starting symbol
  - $P$: set of production rules

A rule replaces a non-terminal hyperedge with a hypergraph

Replace l.h.s. with r.h.s.
MHG is defined as a subclass of HRG

- **Molecular Hypergraph Grammar (MHG)**
  - **Definition:** HRG that generates molecular hypergraphs only
  - **Counterexamples:**

```
valence 😞

2-regularity 😞
```
Hyperedge replacement grammar & Molecular hypergraph grammar

Start from starting symbol $S$

Production rules $P$

- $S \rightarrow N \ N$
- $N \rightarrow H \ C$

© 2019 IBM Corporation
Hyperedge replacement grammar & Molecular hypergraph grammar

The left rule is applicable
We obtain a hypergraph with three non-terminals.
Apply the right rule to one of the non-terminals
Two non-terminals remain

Production rules $P$
Repeat the procedure until there is no non-terminal

Production rules $P$
Repeat the procedure until there is no non-terminal
Repeat the procedure until there is no non-terminal

Production rules $P$

[Diagram showing hyperedge replacement grammar & molecular hypergraph grammar]
Graph generation halts when there is no non-terminal,
Hyperedge replacement grammar & Molecular hypergraph grammar

MHG is defined as a subclass of HRG

- **Molecular Hypergraph Grammar (MHG)**
  - **Definition:** HRG that generates molecular hypergraphs only
  - **Counterexamples:**

  - **Valence 🙁**
  - **2-regularity 😞**

  ![Diagram](image)

  This can be avoided by learning HRG from data [Aguiñaga+, 16]
MHG is defined as a subclass of HRG

- **Molecular Hypergraph Grammar (MHG)**
  - **Definition:** HRG that generates molecular hypergraphs only
  - **Counterexamples:**

  - **Valence 😞**
  - **2-regularity 😞**

  This can be avoided by learning HRG from data [Aguiñaga+, 16]

  Use an irredundant tree decomposition (our contribution)
Our work

Each restriction is our contribution in the literature of graph grammar.

Existing work on graph grammar

[Aguiñaga+, 16]

Set of general hypergraphs

Tree decomposition

Hyperedge Replacement Grammar

Our method

Set of molecular hypergraphs

Irredundant tree decomposition

Molecular Hypergraph Grammar

* HRG is a context-free grammar generating hypergraphs
Tree decomposition & Irredundant tree decomposition

Tree decomposition discovers a tree-like structure in a graph

- **Tree decomposition**
  - All of the nodes and edges must be included in the tree
  - For each node, the tree nodes that contain it must be connected

* Digits represent the node correspondence
Tree decomposition and (a syntax tree of) HRG are equivalent

**Relationship between tree decomposition and HRG**

1. Connecting hypergraphs in tree recovers the original hypergraph
2. Connection ⇔ Hyperedge replacement
Tree decomposition and Irredundant tree decomposition

Tree decomposition and (a syntax tree of) HRG are equivalent

- Relationship between tree decomposition and HRG
  1. Connecting hypergraphs in tree recovers the original hypergraph
  2. Connection $\Leftrightarrow$ Hyperedge replacement
Tree decomposition & Irredundant tree decomposition

Tree decomposition and (a syntax tree of) HRG are equivalent

- **Relationship between tree decomposition and HRG**
  1. Connecting hypergraphs in tree recovers the original hypergraph
  2. Connection ⇔ Hyperedge replacement

![Diagram showing the relationship between tree decomposition and HRG](image-url)
Tree decomposition and (a syntax tree of) HRG are equivalent

- **Relationship between tree decomposition and HRG**
  1. Connecting hypergraphs in tree recovers the original hypergraph
  2. Connection $\Leftrightarrow$ Hyperedge replacement

`Production rule`
Tree decomposition & Irredundant tree decomposition

Extract production rules from a tree decomposition; then HRG with the rules can reconstruct the original hypergraph

- **HRG inference algorithm** [Aguiñaga+, 16]
  
  - **Input:** Set of hypergraphs
  
  - **Output:** HRG w/ the following properties:
    
    - All of the input hypergraphs are in the language 😊
    
    - Guarantee the valence conditions 😞
    
    - No guarantee on 2-regularity 😭

1. Compute tree decompositions of input hypergraphs
2. Extract production rules
3. Compose HRG by taking their union
Extract production rules from a tree decomposition; then HRG with the rules can reconstruct the original hypergraph

- **HRG inference algorithm** [Aguiñaga+, 16]
  - **Input:** Set of hypergraphs
  - **Output:** HRG w/ the following properties:
    - All of the input hypergraphs are in the language 😊
    - Guarantee the valence conditions 😐
    - No guarantee on 2-regularity 😞

This cannot be transformed into a molecular graph
Irredundant tree decomposition is a key to guarantee 2-regularity

- Irredundant tree decomposition
  - The connected subgraph induced by a node must be a path
  - Any tree decomposition can be made irredundant in poly-time
Our work

Each restriction is our contribution in the literature of graph grammar

Existing work on graph grammar
[Aguiñaga+, 16]

- Set of general hypergraphs
  - Tree decomposition
    - Hyperedge Replacement Grammar
- Restrict

Our method

- Set of molecular hypergraphs
  - Irredundant tree decomposition
  - Molecular Hypergraph Grammar
- Restrict

* HRG is a context-free grammar generating hypergraphs
Molecular hypergraph is used to satisfy the valence conditions, and irredundant tree decomposition guarantees 2-regularity.

**MHG Inference algorithm**

- **Input:** Set of molecular graphs
- **Output:** MHG w/ the following properties:
  - All of the input hypergraphs are in the language
  - Guarantee the valence conditions
  - Guarantee 2-regularity

Thanks to HRG

Our contribution

1. Convert molecular graphs into molecular hypergraphs
2. Compute tree decompositions of molecular hypergraphs
3. Convert each tree decomposition to be irredundant
4. Extract production rules
5. Compose MHG by taking their union
Application to Molecular Optimization
Application to Molecular Optimization

We obtain (Enc, Dec) between molecule and latent vector by combining MHG and RNN-VAE

- **MHG-VAE: (Enc, Dec) between molecule & latent vector**

![Diagram showing MHG-VAE encoder with molecular graph, hypergraph, and parse tree connections leading to latent vector output.](image-url)
First, we learn (Enc, Dec) between a molecule and its vector representation using MHG-VAE

- **Global molecular optimization**
  - **Find**: Molecule that maximizes the target
  - **Method**: VAE+BO
    1. Obtain MHG from the input molecules
    2. Train RNN-VAE on syntax trees
    3. Obtain vector representations \( \{z_n \in \mathbb{R}^D \}_{n=1}^N \)
      Some of which have target values \( \{y_n \in \mathbb{R} \} \)
    4. BO gives us candidates \( \{z_m \in \mathbb{R}^D \}_{m=1}^M \) that may maximize the target
    5. Decode them to obtain molecules \( \{G_m \}_{m=1}^M \)
Global molecular optimization

- **Find:** Molecule that maximizes the target
- **Method:** VAE+BO

1. Obtain MHG from the input molecules
2. Train RNN-VAE on syntax trees
3. Obtain vector representations \( \{z_n \in \mathbb{R}^D\}_{n=1}^N \)
   
   Some of which have target values \( \{y_n \in \mathbb{R}\} \)
4. BO gives us candidates \( \{z_m \in \mathbb{R}^D\}_{m=1}^M \) that may maximize the target
5. Decode them to obtain molecules \( \{G_m\}_{m=1}^M \)

Application to Molecular Optimization

Image from [Gómez-Bombarelli+, 16]
Empirical studies

We wish to understand the dependency of the performance on # of evaluations of function, \( f : \text{Molecule} \leftrightarrow \text{Property} \)

- **Purpose of our empirical studies**

  Validate the following questions:

  1. MHG improves the performance compared to JT-VAE?
  2. If # of evaluations is **not limited**, RL will be better than VAE
  3. If # of evaluations is **limited**, VAE will be better than RL

\[
f(m) = \log P(m) - \text{SA}(m) - \text{cycle}(m)
\]

- Penalty to a ring larger than six
- Synthetic accessibility score
- Water solubility
Empirical studies

MHG achieves higher performance than the other VAE-based methods 😊

- Purpose of our empirical studies

  Validate the following questions:

  1. MHG improves the performance compared to VAE-based ones?
  2. If # of evaluations is not limited, RL will be better than VAE
  3. If # of evaluations is limited, VAE will be better than RL

<table>
<thead>
<tr>
<th>Method</th>
<th>% Reconst.</th>
<th>Valid prior</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>50th</th>
<th>Top 50 Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVAE</td>
<td>44.6%</td>
<td>0.7%</td>
<td>1.98</td>
<td>1.42</td>
<td>1.19</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>GVAE</td>
<td>53.7%</td>
<td>7.2%</td>
<td>2.94</td>
<td>2.89</td>
<td>2.80</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>SD-VAE</td>
<td>76.2%</td>
<td>43.5%</td>
<td>4.04</td>
<td>3.50</td>
<td>2.96</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>JT-VAE</td>
<td>76.7%</td>
<td>100%</td>
<td>5.30</td>
<td>4.93</td>
<td>4.49</td>
<td>3.48</td>
<td>3.93</td>
</tr>
<tr>
<td>GCPN</td>
<td>–</td>
<td>–</td>
<td>7.98</td>
<td>7.85</td>
<td>7.80</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Ours</td>
<td>94.8%</td>
<td>100%</td>
<td>5.56</td>
<td>5.40</td>
<td>5.34</td>
<td>4.12</td>
<td>4.49</td>
</tr>
</tbody>
</table>

JT-VAE by Jin et al., ICML ’18 (VAE+BO)  
GCPN by You et al., NeurlPS ‘18 (RL)
Empirical studies

But RL-based method is better than ours when # of evaluations is not limited 😞

- Purpose of our empirical studies

  Validate the following questions:

  1. MHG improves the performance compared to VAE-based ones?
  2. If # of evaluations is not limited, RL will be better than VAE
  3. If # of evaluations is limited, VAE will be better than RL

<table>
<thead>
<tr>
<th>Method</th>
<th>% Reconst.</th>
<th>Valid prior</th>
<th>Unlimited oracle case</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>1st</td>
</tr>
<tr>
<td>CVAE</td>
<td>44.6%</td>
<td>0.7%</td>
<td>1.98</td>
</tr>
<tr>
<td>GVAE</td>
<td>53.7%</td>
<td>7.2%</td>
<td>2.94</td>
</tr>
<tr>
<td>SD-VAE</td>
<td>76.2%</td>
<td>43.5%</td>
<td>4.04</td>
</tr>
<tr>
<td>JT-VAE</td>
<td>76.7%</td>
<td>100%</td>
<td>5.30</td>
</tr>
<tr>
<td>GCPN</td>
<td>–</td>
<td>–</td>
<td>7.98</td>
</tr>
<tr>
<td>Ours</td>
<td>94.8%</td>
<td>100%</td>
<td>5.56</td>
</tr>
</tbody>
</table>

JT-VAE by Jin et al., ICML ’18 (VAE+BO)
GCPN by You et al., NeurIPS ‘18 (RL)
Ours is better than the RL-based method when # of evaluations is limited to 500 😁

**Purpose of our empirical studies**

Validate the following questions:

1. MHG improves the performance compared to VAE-based ones?
2. If # of evaluations is **not limited**, RL will be better than VAE
3. If # of evaluations is **limited**, VAE will be better than RL

<table>
<thead>
<tr>
<th>Method</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>50th</th>
<th>Top 50 Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>JT-VAE</td>
<td>1.69</td>
<td>1.68</td>
<td>1.60</td>
<td>-9.93</td>
<td>-1.33</td>
</tr>
<tr>
<td>GCPN</td>
<td>2.77</td>
<td>2.73</td>
<td>2.34</td>
<td>0.91</td>
<td>1.36</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td><strong>5.24</strong></td>
<td><strong>5.06</strong></td>
<td><strong>4.91</strong></td>
<td><strong>4.25</strong></td>
<td><strong>4.53</strong></td>
</tr>
</tbody>
</table>

JT-VAE by Jin et al., ICML '18 (VAE+BO)
GCPN by You et al., NeurIPS '18 (RL)
We develop a graph-grammar based molecular representation, MHG, as well as its combination with VAE.

- We develop a molecular hypergraph grammar (MHG)
  - Molecules generated by MHG always satisfy the valence conditions
    - Hypergraph representation
    - Irredundant tree decomposition
  - Inference algorithm from data
    - No need to write rules by hand
    - The inferred MHG can describe all of the input data

- We apply MHG to molecular optimization
  - MHG-VAE > JT-VAE
  - MHG-VAE > GCPN when # of evals is limited

