# Graph Convolutional Neural Networks for Molecule Generation

Xavier Bresson

School of Computer Science and Engineering
Data Science and AI Research Centre
Nanyang Technological University (NTU), Singapore

Joint work with Thomas Laurent (LMU)

IPAM Workshop
From Passive to Active: Generative and Reinforcement Learning with Physics
September 23rd 2019

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

NATIONAL RESEARCH FOUNDATION
PRIME MINISTER'S OFFICE
SINGAPORE

# Outline

- **Graph ConvNets**
- **Molecule Generation**
- **Conclusion**

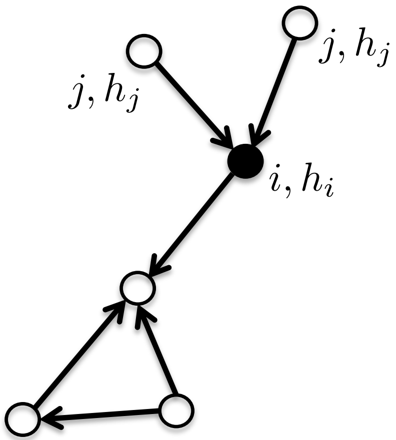
# Outline

- **Graph ConvNets**
- Molecule Generation
- Conclusion

# Graph Neural Networks[1]

- NNs specialized to data on graphs.
- Minimal inner structures to design GNNs :
  - Invariant by vertex re-indexing (no graph matching is required)
  - Locality/local reception field (only neighbors are considered)
  - Weight sharing (convolutional operations)
  - Independence w.r.t. graph size

$$h_i = f_{\text{GNN}}\left(\{h_j : j \to i\}\right)$$




- What instantiation of $f_{\text{GNN}}$ ?

[1] Scarselli, Gori, Tsoi, Hagenbuchner, Monfardini, The Graph Neural Network Model, 2009

# Graph Recurrent Neural Networks

- Graph RNN with Multi-Layer Perceptron (MLP)[1] :

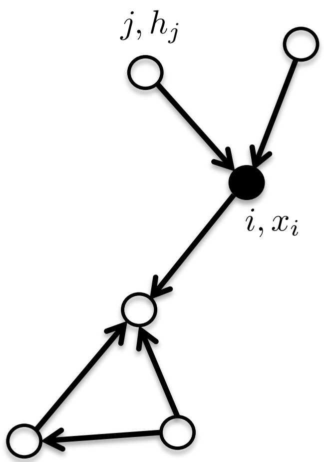
$$h_i = \sum_{j \to i} \mathcal{C}_{\text{G-MLP}}(x_i, h_j) = \sum_{j \to i} A\sigma(B\sigma(Ux_i + Vh_j))$$

- Graph GRU[2,3] (Gated Recurrent Unit) :

$$h_i = \mathcal{C}_{\text{G-GRU}}(x_i, \sum_{j \to i} h_j)$$

Fixed-point iterative scheme needed :

$$\begin{aligned}
\bar{h}_i^t &= \sum_{j \to i} h_j^t, \quad h_i^{t=0} = x_i \\
z_i^{t+1} &= \sigma(U_z h_i^t + V_z \bar{h}_i^t) \\
r_i^{t+1} &= \sigma(U_r h_i^t + V_r \bar{h}_i^t) \\
\tilde{h}_i^{t+1} &= \tanh\big(U_h(h_i^t \odot r_i^{t+1}) + V_h \bar{h}_i^t\big) \\
h_i^{t+1} &= (1 - z_i^{t+1}) \odot h_i^t + z_i^{t+1} \odot \tilde{h}_i^{t+1}
\end{aligned}$$



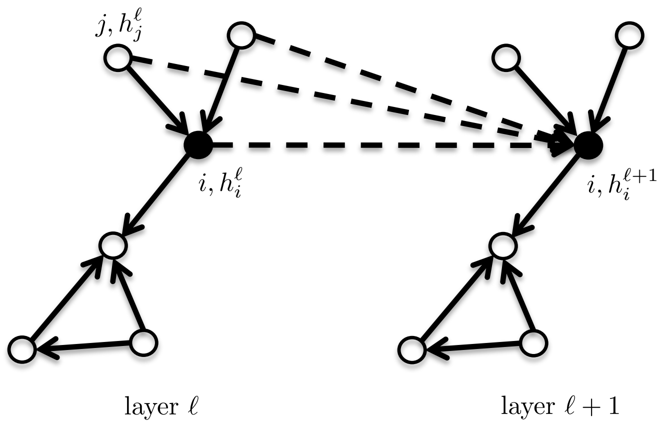


[1] Scarselli, Gori, Tsoi, Hagenbuchner, Monfardini, The Graph Neural Network Model, 2009
[2] Li, Tarlow, Brockschmidt, Zemel, Gated Graph Sequence Neural Networks, 2015
[3] Cho, Merrienboer, Gulcehre, Bahdanau, Bougares, Schwenk, Bengio, Learning Phrase Representations using RNN for Statistical Machine Translation, 2014

# Graph ConvNets

- Graph ConvNets[1,2,3], GCN[4] (with ReLU) and GraphSAGE[5] (with max) :

$$
\begin{aligned}
h_i^{\ell+1} &= \mathcal{C}_{\text{G-VCN}}\Big(h_i^\ell, \sum_{j\to i} h_j^\ell\Big), \quad h_i^{\ell=0} = x_i \\
&= \text{ReLU}\Big(U^\ell h_i^\ell + V^\ell \sum_{j\to i} h_j^\ell\Big), \quad h_i^{\ell=0} = x_i
\end{aligned}
$$





[1] Bruna, Zaremba, Szlam, LeCun, Spectral networks and locally connected networks on graphs, 2013
[2] Defferrard, Bresson, Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, 2016
[3] Sukhbaatar, Szlam, Fergus, Learning Multiagent Communication with Backpropagation, 2016
[4] Kipf, Welling, Semi-Supervised Classification with Graph Convolutional Networks, 2017
[5] Hamilton, Ying, Leskovec, Inductive representation learning on large graphs, 2017

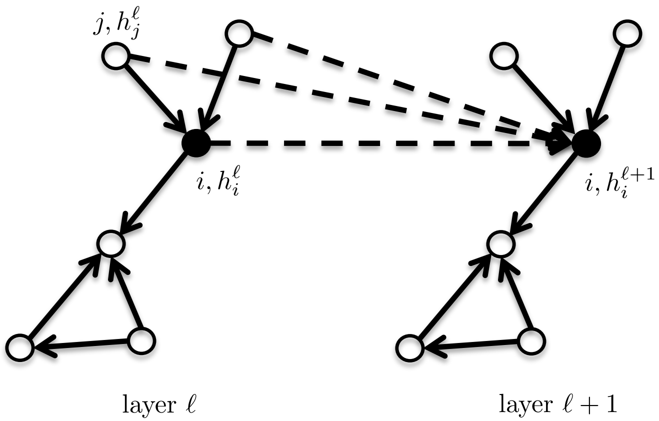
# Gated Graph ConvNets[1]

- Graph ConvNets architecture with edge gating mechanism leveraging[2-4], residuality[5] and batch normalization[6] :

$$h_i^{\ell+1} = h_i^\ell + \mathrm{ReLU}\Big(\mathrm{BN}\Big(U^\ell h_i^\ell + \sum_{j \to i} \eta_{ij}^\ell \odot V^\ell h_j^\ell\Big)\Big)$$

edge gates
(anisotropic property)

$$\eta_{ij} = \sigma\left(A^\ell h_i^\ell + B^\ell h_j^\ell\right)$$




The idea is to design the simplest learnable anisotropic and multiscale diffusion operator on graphs [Perona-Malik'87 inspiration]

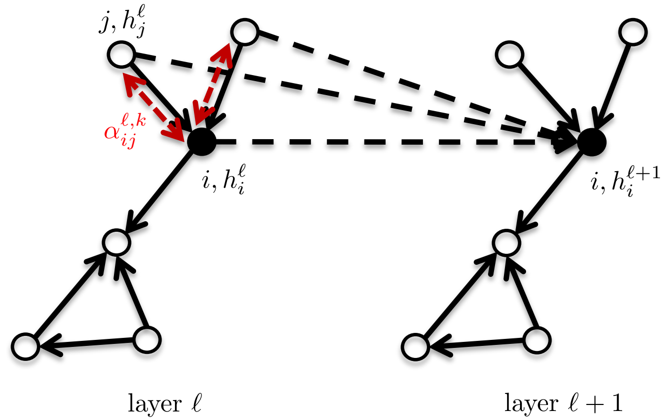

[1] Bresson, Laurent, Residual gated graph convnets, 2017
[2] Sukhbaatar, Szlam, Fergus, Learning Multiagent Communication with Backpropagation, 2016
[3] Hamilton, Ying, Leskovec, Inductive representation learning on large graphs, 2017
[4] Marcheggiani, Titov, Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling, 2017
[5] He, Zhang, Ren, Sun, Deep Residual Learning for Image Recognition, 2016
[6] Ioffe, Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, 2015

# Graph Attention Networks[1]

- Attention mechanism in 1-hop neighborhood :

$$
\begin{aligned}
h_i^{\ell+1} &= \Vert_{k=1}^{K} \, \sigma\Big( \sum_{j \to i} \alpha_{ij}^{\ell,k} \, W^{\ell,k} h_j^{\ell} \Big) \\
\alpha_{ij}^{\ell,k} &= \mathrm{softmax}_{\text{1-hop}}(V^{\ell,k} h_i^{\ell}) \\
&= \frac{e^{V^{\ell,k} h_i^{\ell}}}{\sum_{j \to i} e^{V^{\ell,k} h_j^{\ell}}}
\end{aligned}
$$

**Self-attention**




[1] Velickovic, Cucurull, Casanova, Romero, Lio, Bengio, Graph Attention Networks, 2018

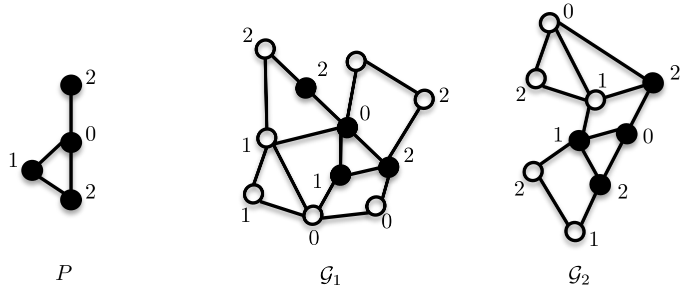
# Graph RNNs vs Graph ConvNets/AttentionNets

- Numerical study to compare both graph architectures[1] on two basic and representative graph problems:
  - Sub-graph matching[2]
  - Semi-supervised classification

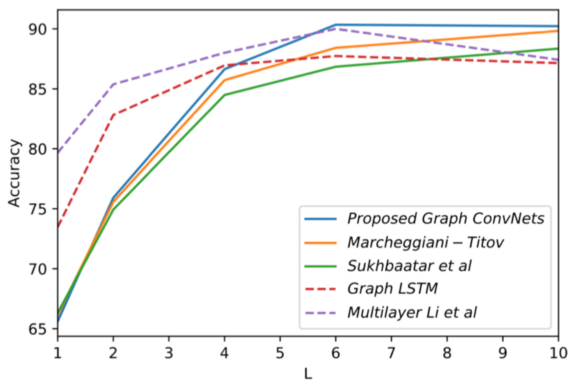
[1] Bresson, Laurent, Residual gated graph convnets, 2017
[2] Scarselli, Gori, Tsoi, Hagenbuchner, Monfardini, The Graph Neural Network Model, 2009

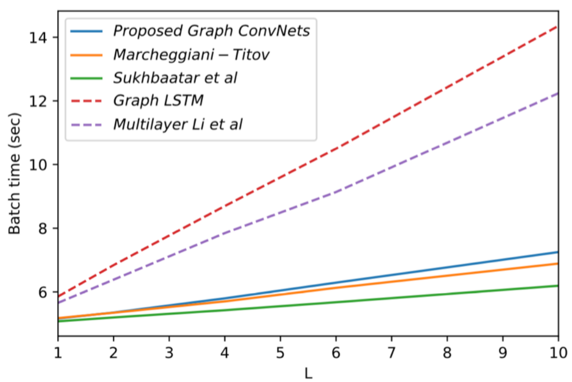
# Numerical Experiments

- Graph learning problem :
  - Pattern matching




- Experimental results:

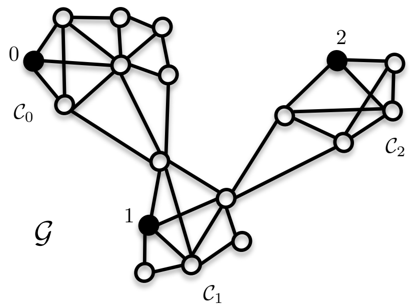


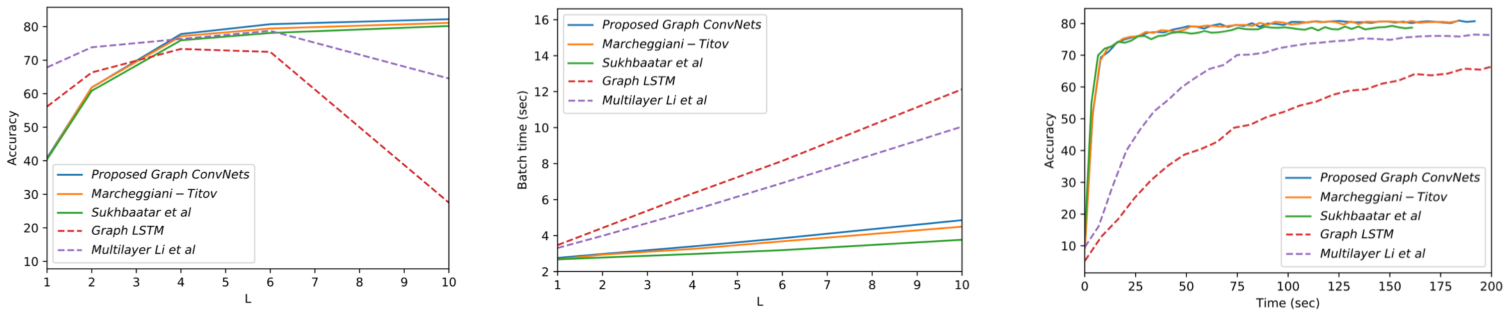

- All graph NNs are upgraded with residuality and batch normalization (offers 10% improvement).

# Numerical Experiments

- Graph learning problem :
  - Semi-supervised clustering

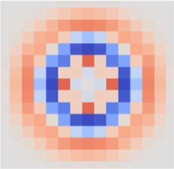



- Experimental results :




- ConvNets architectures that can be deep (by stacking many layers) offer competitive performances for graphs with variable sizes.

# Anisotropy vs Isotropy



- Standard ConvNets produce anisotropic filters because Euclidean grids have directional structure.
- Graph ConvNets compute isotropic filters because there is no notion of directions on arbitrary graphs.
- How to get anisotropy back in GNNs ?
  - Edge gates[1]/attention mechanism[2] information to treat neighbors differently.
  - Differentiate graph edges[3] (e.g. different connections between atoms)

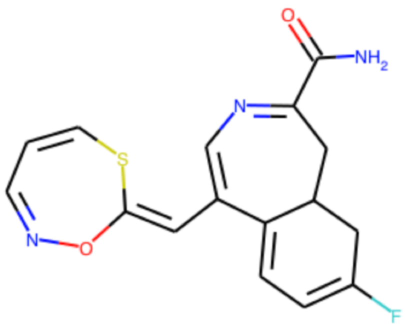
[1] Bresson, Laurent, Residual gated graph convnets, 2017
[2] Velickovic, Cucurull, Casanova, Romero, Lio, Bengio, Graph Attention Networks, 2018
[3] Gilmer, Schoenholz, Riley, Vinyals, Dahl, Neural message passing for quantum chemistry, 2017

# Outline

- Graph ConvNets
- **Molecule Generation**
- Conclusion

# Molecule Generation

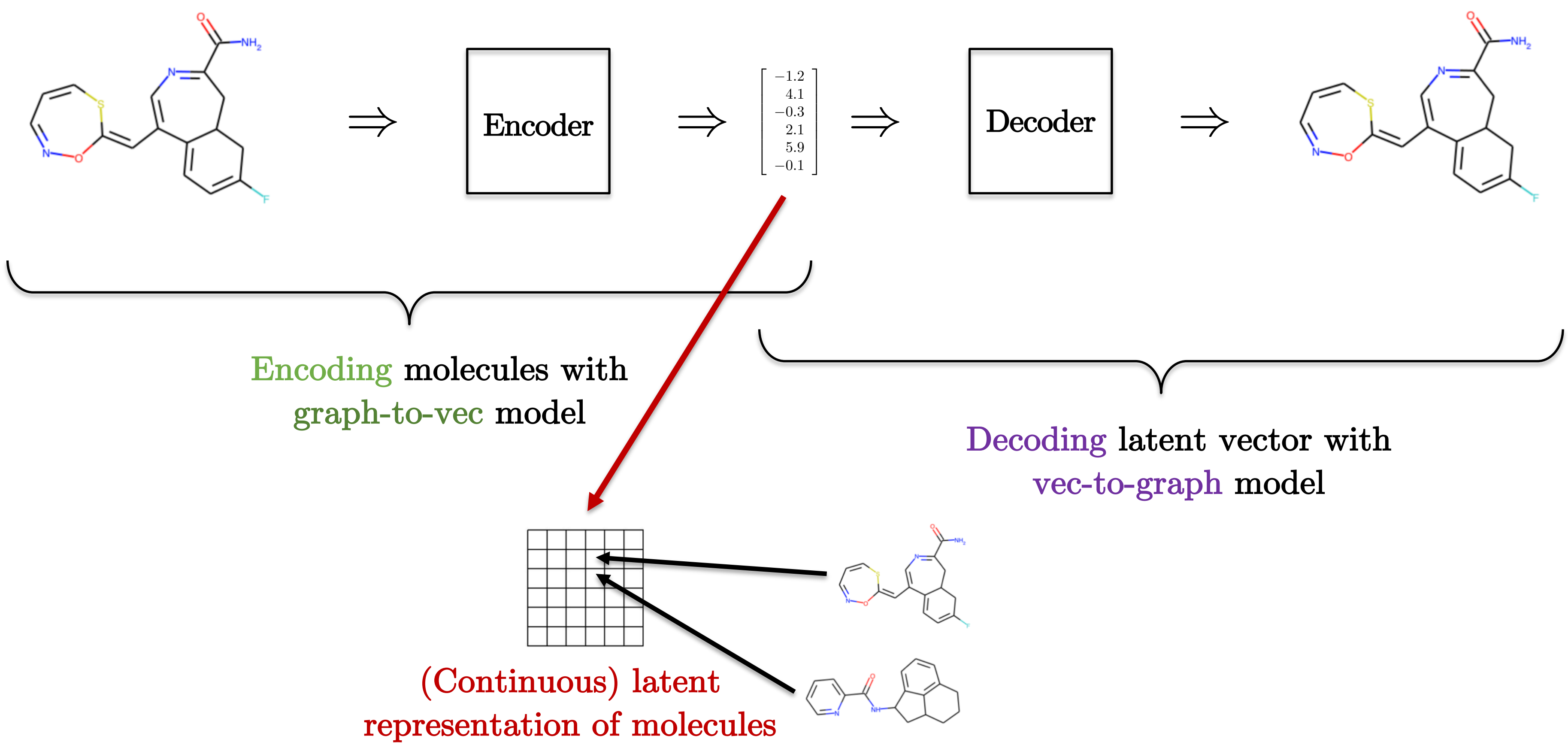
- Goal is to design a neural network that can
  - Auto-encode molecules,
  - Generate novel molecules,
  - Produce molecules with optimized chemical property.




Paper : https://arxiv.org/pdf/1906.03412.pdf

# Graph Auto-Encoder

- Graph-to-Graph Model :



# Graph Encoder

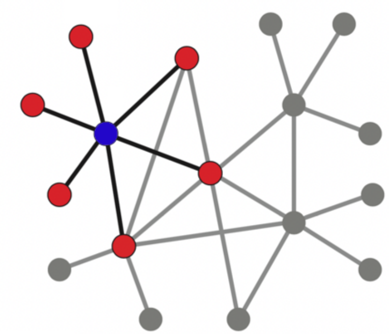
- Graph NNs has been used to encode molecules into a continuous vectorial space.
  - GNNs used for regression s.a. Duvenaud-Gómez-Bombarelli-Aspuru-Guzik-et-al[1], Gilmer-Riley-et.al[2] to predict molecular properties (1-2 orders of magnitude faster than solving Schrodinger equation w/ DFT).

Graph RNNs, Graph ConvNets, Graph Attention Nets

$$h_i = f_{\text{node}}(\{h_j\}, j \in \mathcal{N}(i))$$

$$z = g_{\text{graph}}(\{h_i\}, i \in V)$$

Reduce function : Sum or Mean

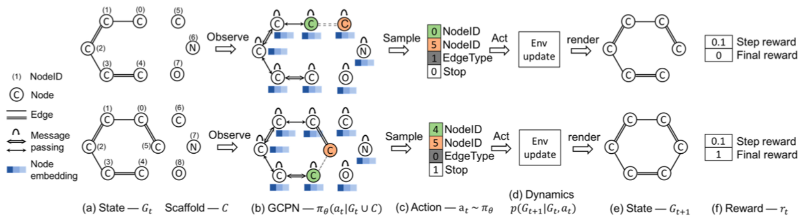


$$\Rightarrow \begin{bmatrix} -1.2 \\ 4.1 \\ -0.3 \\ 2.1 \\ 5.9 \\ -0.1 \end{bmatrix}$$

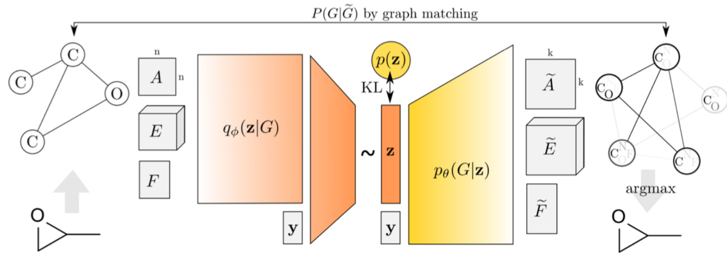
[1] Duvenaud, Maclaurin, Iparraguirre, Bombarell, Hirzel, Aspuru-Guzik, Adams, Convolutional networks on graphs for learning molecular fingerprints, 2015
[2] Gilmer, Schoenholz, Riley, Vinyals, Dahl, Neural message passing for quantum chemistry, 2017

# Decoder & Graph Generation

- Encoding is easy. Decoding is more challenging !
- Two approaches :
  - Auto-regressive models : Sequential generation of molecules (atom-by-atom).
    - Jin-et.al, 2018[1], You-Leskovec-et.al, 2018[2], etc
  - One-shot models : Generation of all atoms and bonds in a single pass.
    - Simonovsky, Komodakis, 2018[3], De Cao, Kipf, 2018[4], etc




You-Leskovec-et.al, 2018

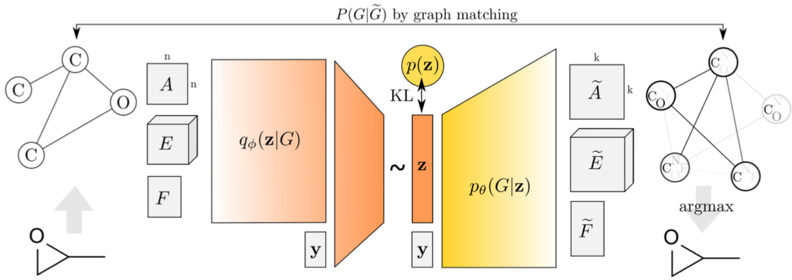



Simonovsky, Komodakis, 2018


[1] Jin, Barzilay, Jaakkola, Junction Tree Variational Autoencoder for Molecular Graph Generation, 2018
[2] You, Liu, Ying, Pande, Leskovec, Graph convolutional policy network for goal-directed molecular graph generation, 2018
[2] Simonovsky, Komodakis, GraphVAE: Towards generation of small graphs using variational autoencoders, 2018
[4] De Cao, Kipf, MolGAN: An implicit generative model for small molecular graphs, 2018

# One-Shot Decoder

- A challenge with one-shot decoder is to generate molecules of different sizes.
  - It is hard to generate simultaneously :
    - The number of atoms,
    - The atoms,
    - The bond structures between the atoms.
  - Authors[1,2] generated molecules with a fixed size (the size of the largest molecule).




[1] Simonovsky, Komodakis, GraphVAE: Towards generation of small graphs using variational autoencoders, 2018
[2] De Cao, Kipf, MolGAN: An implicit generative model for small molecular graphs, 2018

# Our Decoder

- We propose to disentangle these 3 problems :

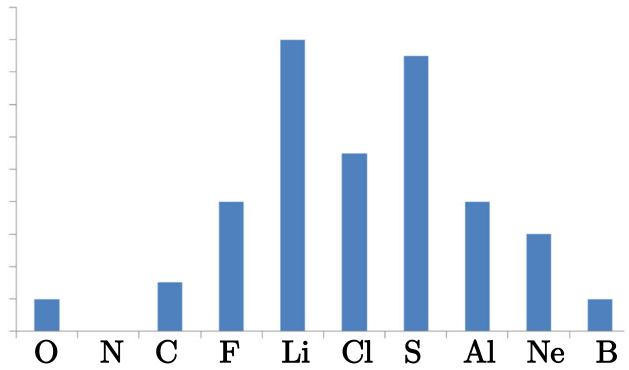
$$\begin{bmatrix}-1.2\\4.1\\-0.3\\2.1\\5.9\\-0.1\end{bmatrix}$$

Latent representation of the molecule

Atom generation

Molecule formula

Ex: $OC_2F_4Li_7Cl_3S_8Al_2NeB$

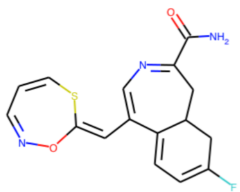



Bag-of-atoms

Sum of histogram is the number of atoms in a molecule

Bond generation

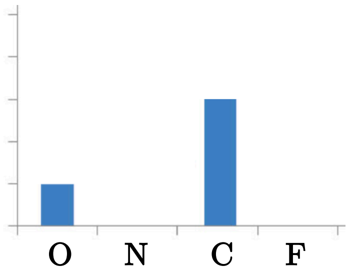
Bond structure w/ molecule formula



# Atom Decoder

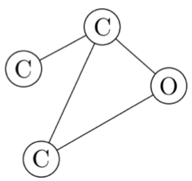
- We decode the latent representation of the molecule with a Multi-Layer Perceptron (MLP) to produce the histogram over the atoms in a molecule :

$$\begin{bmatrix} -1.2 \\ 4.1 \\ -0.3 \\ 2.1 \\ 5.9 \\ -0.1 \end{bmatrix} \xrightarrow{\text{MLP}}$$




Latent representation of the molecule

Bag-of-atoms



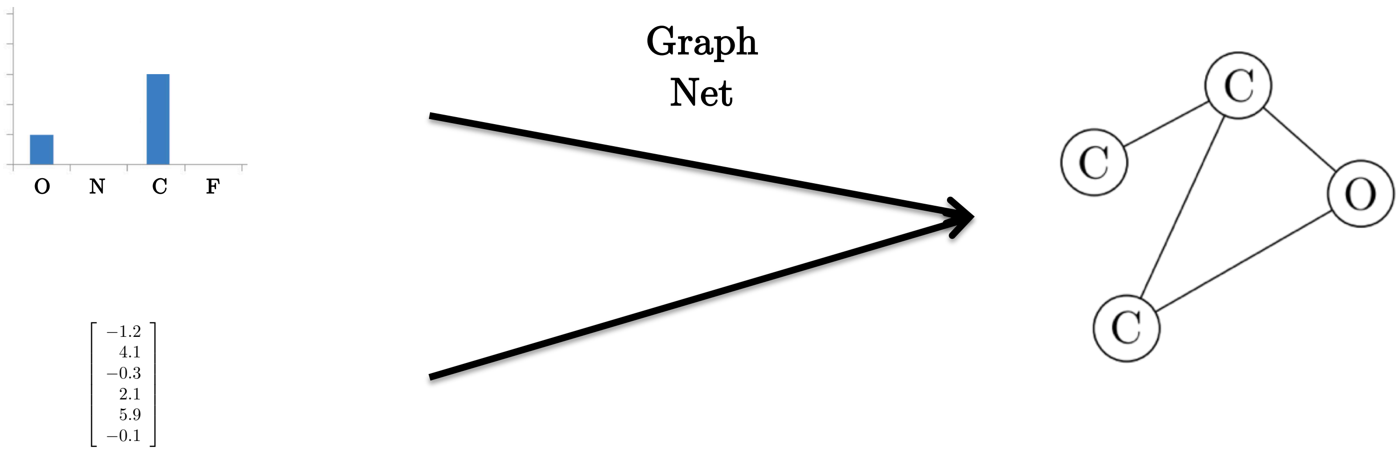

$OC_3$

# Bond Decoder

- The “IKEA” model :
  - The bag-of-atoms indicates what atoms are in the molecule (IKEA pieces),
  - The atoms are assembled with a graph NN (IKEA assembly instructions).

Bag-of-atoms
$OC_3$

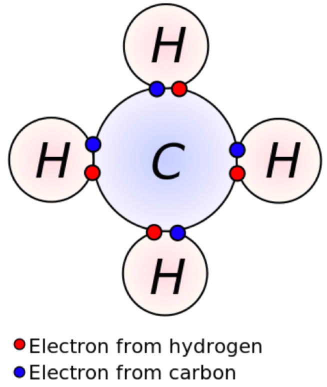
Latent
representation
of the molecule



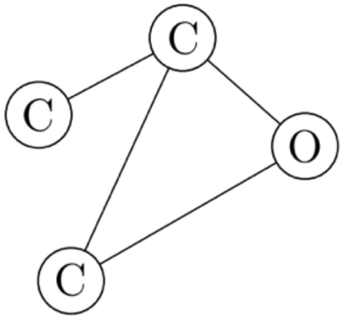

Decoded
molecule

# Beam Search

- The one-shot model may not produce a chemically valid molecule.
  - Violation of atom valency (maximum number of electrons in the outer shell of the atom that can participate of a chemical bond).

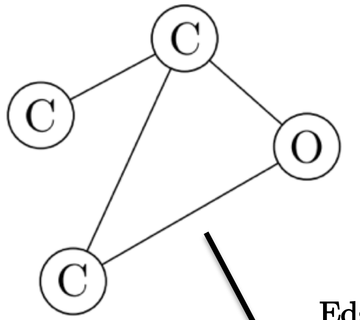



- We use beam search to produce valid molecules.

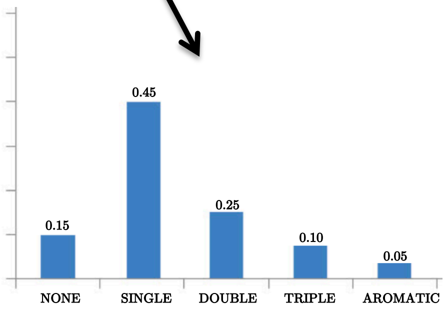


# Beam Search

- Beam search :
  - Start with a random edge.
    - Select the next edges that
      - have the largest probability (or Bernouilli sampling),
      - are connected to selected edges,
      - do not violate valency.
  - Repeat for a number of different initializations.
  - Select the molecule that maximizes
    - The product of edge probabilities or,
    - The chemical property to be optimized s.a. druglikeness (QED), constrained solubility (logP), etc.

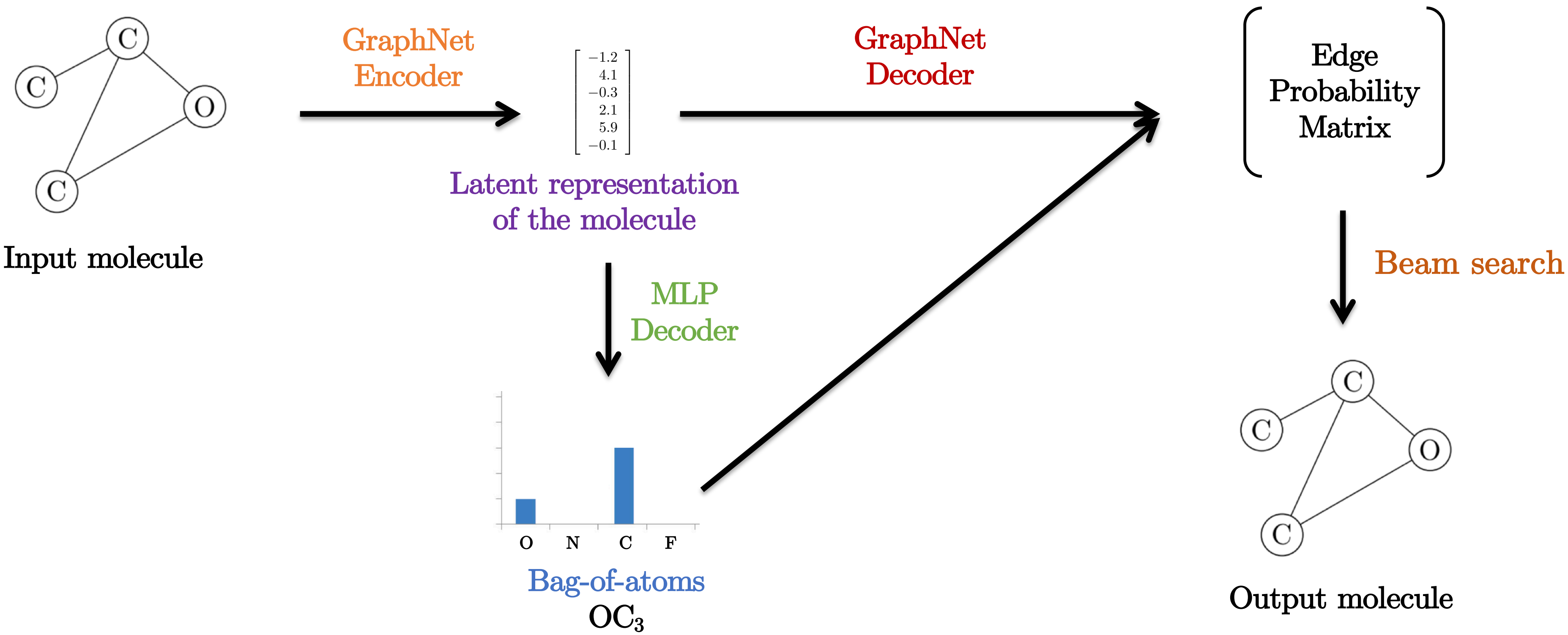



Edge probability map given by the output of the graph NN.



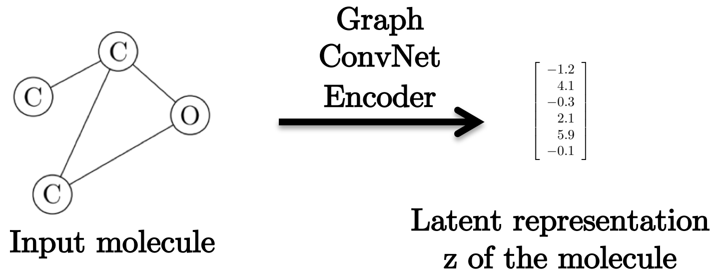
# Summary

- Molecule auto-encoder system :



# Encoder Description

- We use graph ConvNet[1] :




Node and edge representations

$$\begin{cases} h_i^{\ell+1} = h_i^\ell + \mathrm{ReLU}\Big(\mathrm{BN}\Big(W_1^\ell h_i^\ell + \sum_{j\sim i} \eta_{ij}^\ell \odot W_2^\ell h_j^\ell\Big)\Big) \\ e_{ij}^{\ell+1} = e_{ij}^\ell + \mathrm{ReLU}\Big(\mathrm{BN}\Big(V_1^\ell e_{ij}^\ell + V_2^\ell h_i^\ell + V_3^\ell h_j^\ell\Big)\Big) \end{cases}$$

**with** $\eta_{ij}^\ell = \dfrac{\sigma(e_{ij}^\ell)}{\sum_{j'\sim i}\sigma(e_{ij'}^\ell) + \varepsilon}$

Dense attention

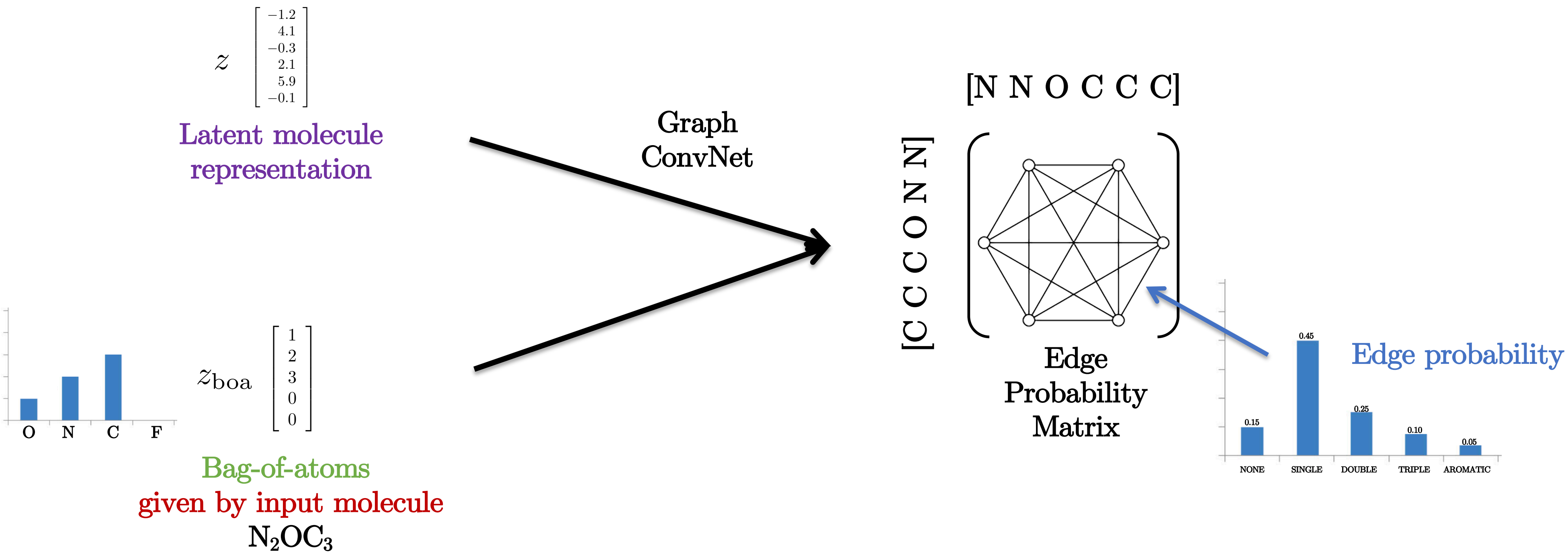
Graph representation

$$z = \sum_{i,j=1}^{N} \sigma\left(Ae_{ij}^L + Bh_i^L + Ch_j^L\right) \odot We_{ij}^L$$

[1] Bresson, Laurent, Residual gated graph convnets, 2017

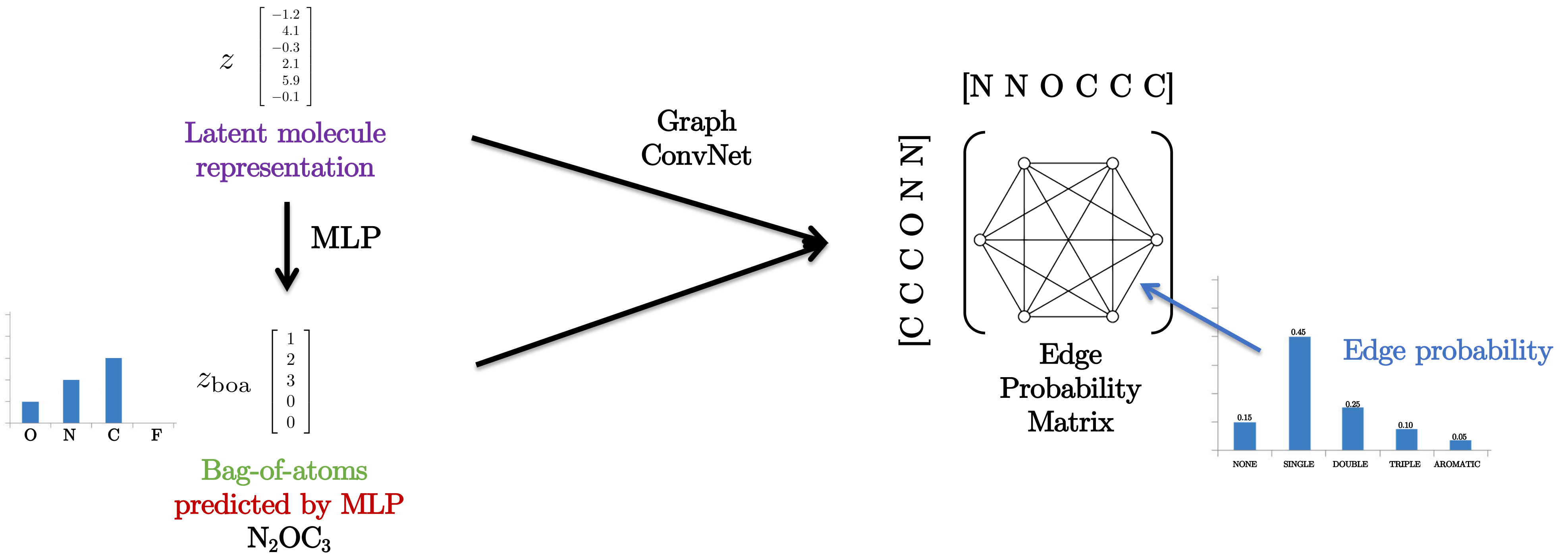
# Bond Decoding during Training

- **Given the latent encoding $z$ of the molecule and the bag-of-atoms $z_{boa}$ , we use a graph ConvNet to decode the bonds between the atoms :**



# Bond Decoding at Test Time

- The bag-of-atoms of the input molecule is predicted by a MLP :



# Breaking symmetry

- The bond decoder starts with a fully connected graph with the atom type $z_{\text{ato}}$ on each node.

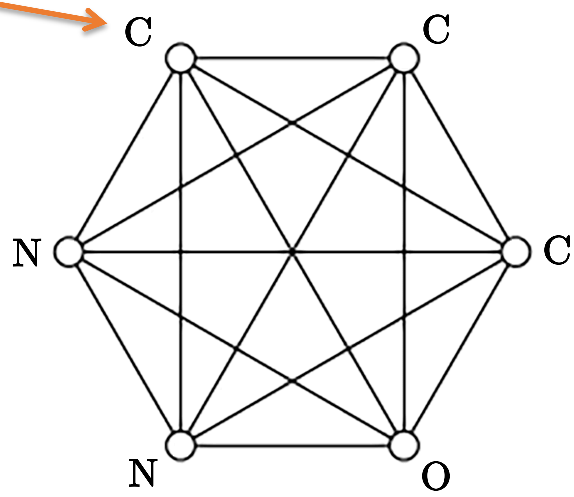
- This is not enough for the graph NN to be able to differentiate the 3 atoms of Carbon and the 2 atoms of Nitrogen !

  - We break the symmetry by introducing positional features $z_{\text{pos}}$ , which will differentiate several atoms of the same type.

  - We concatenate this positional feature with the atom type $z_{\text{ato}}$ to form the input node feature of the decoder.

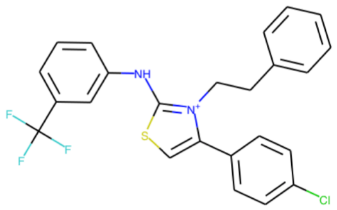
$$h_i^{\ell=0} = \left[ \begin{array}{c} z_{\text{ato}\,i} \\ z_{\text{pos}\,i} \end{array} \right]$$

$$h_i^{\ell=0} = \left[ \; z_{\text{ato}\,i} \; \right]$$



# Positional Features

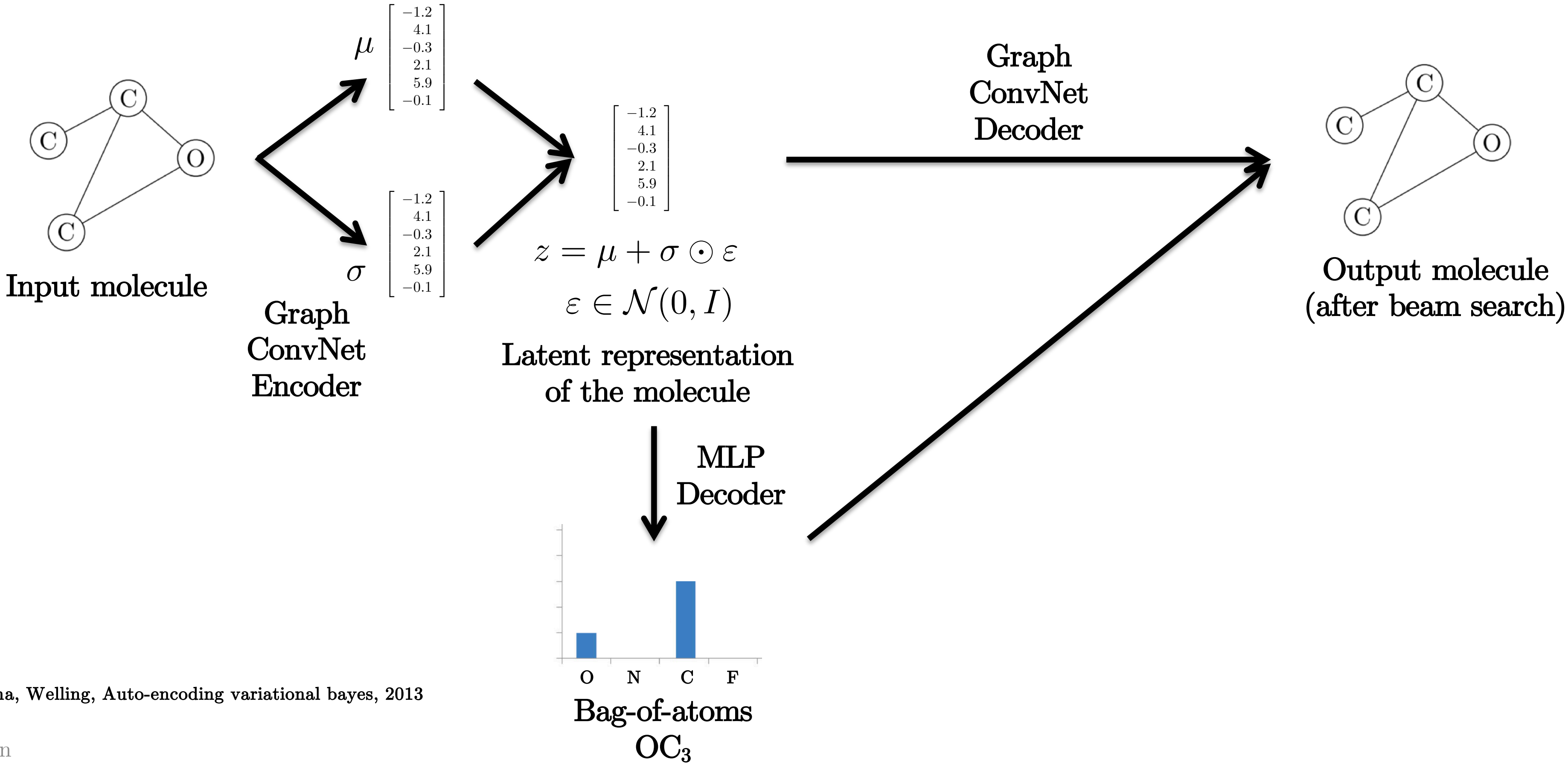
- We need to order the atoms.
- We use the SMILE representation of molecules to order the atoms.
  - A SMILE is a sequence of string characters that encodes atoms and bonds of a molecule.

SMILE (  ) = FC(F)(F)C1=CC=CC(NC2=[N+](CCC3=CC=CC=C3)...
...C(C3=CC=C(Cl)C=C3)=CS2)=C1

$(\mathrm{F},1)$ $(\mathrm{C},1)$ $(\mathrm{F},2)$ $(\mathrm{F},3)$ $(\mathrm{C},2)\ldots$

# Variational Auto-Encoder

- Finally, we use a VAE formulation[1] to improve molecule generation “by filling the latent space” :




[1] Kingma, Welling, Auto-encoding variational bayes, 2013

# Loss

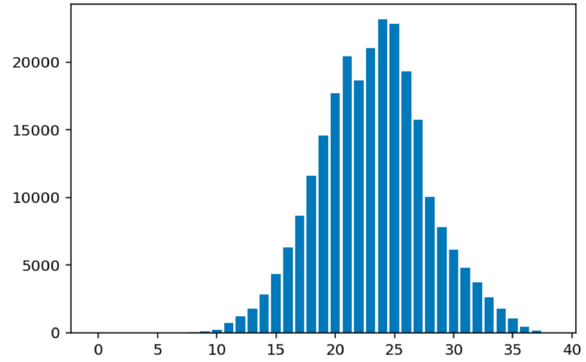
- Final loss is composed of
  - Cross-entropy loss for edge probability,
  - Cross-entropy loss for bag-of-atoms probability,
  - Kullback–Leibler divergence for the VAE Gaussian distribution.

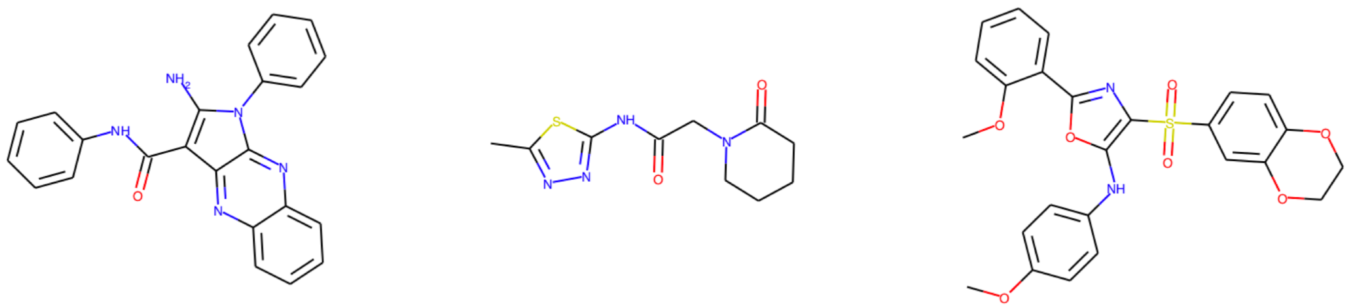
$$L = \lambda_e \sum_e \hat{p}_e \log p_e + \lambda_a \sum_a \hat{p}_a \log p_a - \frac{\lambda_{vae}}{2} \sum_k (1 + \log \sigma_k^2 - \mu_k^2 - \sigma_k^2)$$

- No matching process necessary between input and output molecules because the same atom ordering is used (with the SMILE representation).

# Dataset

- ZINC :
  - 250k drug like molecules,
  - Up to 38 heavy atoms (excluded Hydrogen).



# Training

- Mini-batch of 50 molecules
- Learning rate is decreased by 1.25 after each epoch if training loss does not decrease by 1%.
- Learning stops when LR is less than $10^{-6}$.
- Training takes 28 hours on a single Nvidia 1080Ti GPU.

# Numerical Experiments

- Molecule reconstruction
  - How many molecules are correctly decoded?
- Molecule novelty
  - Beyond memorization – how many molecules sampled from the learned distribution are not in the training set?
- Molecule optimization
  - How much property improvement can we obtain when optimizing in the latent space?
  - The chemical property is here the constrained solubility of molecules.

# Main Baseline Techniques

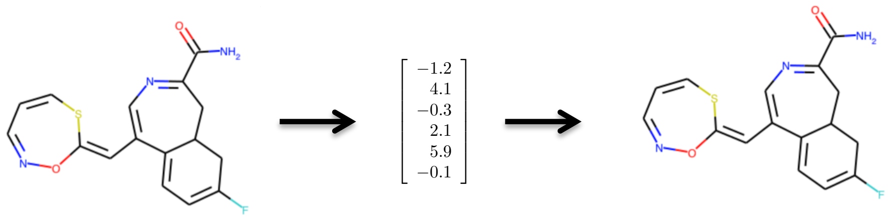
- VAE + SL + AR :
  - JT-VAE : Jin, Barzilay, Jaakkola, Junction Tree Variational Autoencoder for Molecular Graph Generation, 2018

- GAN + RL + AR :
  - GCPN : You, Liu, Ying, Pande, Leskovec, Graph convolutional policy network for goal-directed molecular graph generation, 2018

# Molecule Reconstruction

| Method | Reconstruction | Validity |
|---|---|---|
| CVAE [Gomez-Bombarelli et al., 2016] | 44.6% | 0.7% |
| GVAE [Kusner et al., 2017] | 53.7% | 7.2% |
| SD-VAE [Dai et al, 2018] | 76.2% | 43.5% |
| GraphVAE [Simonovsky, Komodakis, 2018] | - | 13.5% |
| JT-VAE (SL) [Jin et al, 2018] | 76.7% | **100.0%** |
| GCPN (GAN+RL) [You et al, 2018] | - | - |
| OURS (VAE+SL) | **90.5%** | **100.0%** |

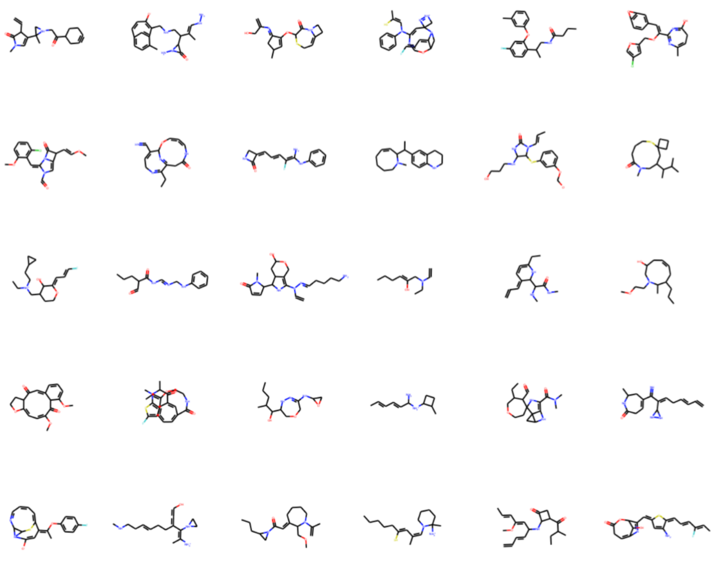
Table 1: Percentage of successful reconstruction of 250k ZINC molecules.



# Molecule Novelty

| Method | Novelty | Uniqueness |
|---|---|---|
| JT-VAE (SL) [Jin et al, 2018] | **100.0%** | **100.0%** |
| GCPN (GAN+RL) [You et al, 2018] | - | - |
| OURS (VAE+SL) | **100.0%** | **100.0%** |

Table 2: Sample 5000 molecules from learned prior distribution.



# Molecule Optimization #1

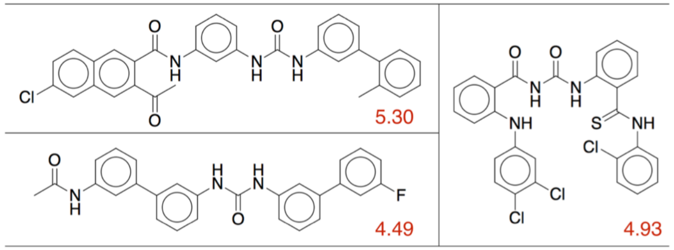
- Molecule optimization :
  - Goal is to maximize the constrained solubility of the training molecules.
  - Optimization is done by gradient ascent in the latent space of molecules.
  - Following JT-VAE, we report the top 3 optimized molecules :

| Method | 1st | 2nd | 3rd | Mean |
|---|---|---|---|---|
| ZINC | 4.52 | 4.30 | 4.23 | 4.35 |
| CVAE, Gómez-Bombarelli et al. [2018] | 1.98 | 1.42 | 1.19 | 1.53 |
| GVAE, Kusner et al. [2017] | 2.94 | 2.89 | 2.80 | 2.87 |
| SD-VAE, Dai et al. [2018] | 4.04 | 3.50 | 2.96 | 3.50 |
| JT-VAE, Jin et al. [2018] | 5.30 | 4.93 | 4.49 | 4.90 |
| OURS (VAE+SL) | 5.24 | 5.10 | 5.06 | 5.14 |
| GCPN (GAN+RL), You et al. [2018] | **7.98** | **7.85** | **7.80** | **7.88** |

# Molecule Optimization #1

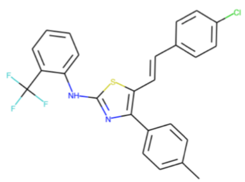
- Top 3 optimized molecules :

JT-VAE (VAE+SL)

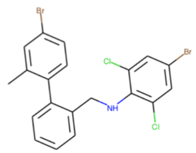



Mean is 4.90

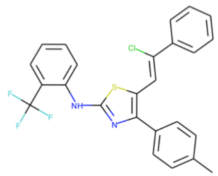
OURS (VAE+SL)



5.24



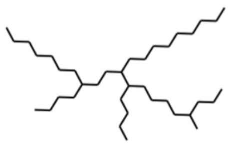
5.14



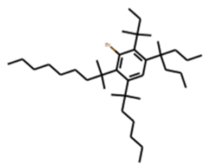
5.06

Mean is 5.14

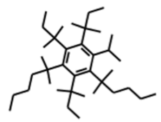
GCPN (GAN+RL)



7.98



7.48



7.12

Mean is 7.52

# Molecule Optimization #2

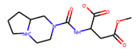
- Constrained optimization :
  - Goal is to maximize the constrained solubility of the 800 test molecules with the lowest value.
  - The optimization of the chemical property is constrained by the similarity between the original molecule and the new generated molecule.
  - Following JT-VAE, we report property improvements w.r.t. molecule similarity $\delta$ :

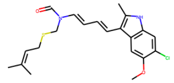
| | JT-VAE [Jin et al, 2018] (SL) | | | GCPN [You et al, 2018] (GAN+RL) | | | OURS (VAE+SL) | | |
|---|---|---|---|---|---|---|---|---|---|
| $\delta$ | Improvement | Similarity | Success | Improvement | Similarity | Success | Improvement | Similarity | Success |
| 0.0 | $1.91 \pm 2.04$ | $0.28 \pm 0.15$ | 97.5% | $4.20 \pm 1.28$ | $\mathbf{0.32 \pm 0.12}$ | **100.0%** | $\mathbf{5.24 \pm 1.55}$ | $0.18 \pm 0.12$ | **100.0%** |
| 0.2 | $1.68 \pm 1.85$ | $0.33 \pm 0.13$ | 97.1% | $4.12 \pm 1.19$ | $\mathbf{0.34 \pm 0.11}$ | **100.0%** | $\mathbf{4.29 \pm 1.57}$ | $0.31 \pm 0.12$ | 98.6% |
| 0.4 | $0.84 \pm 1.45$ | $\mathbf{0.51 \pm 0.10}$ | 83.6% | $2.49 \pm 1.30$ | $0.47 \pm 0.08$ | **100.0%** | $\mathbf{3.05 \pm 1.46}$ | $\mathbf{0.51 \pm 0.10}$ | 84.0% |
| 0.6 | $0.21 \pm 0.71$ | $\mathbf{0.69 \pm 0.06}$ | 46.4% | $0.79 \pm 0.63$ | $0.68 \pm 0.08$ | **100.0%** | $\mathbf{2.46 \pm 1.27}$ | $0.67 \pm 0.05$ | 40.1% |

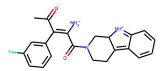
Table 7: Molecule optimization results.

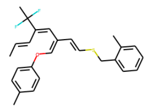
# Molecule Optimization #2

Molecule
similarity 0.0

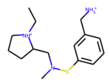

4: -8.38

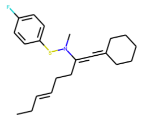

4: 2.19

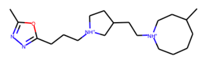

77: -5.81

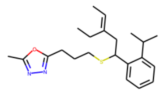

77: 4.75

Molecule
similarity 0.2

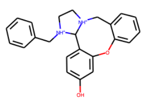

143: -5.01

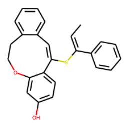

143: 3.54

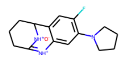

136: -5.06

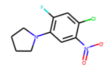

136: 3.10

Molecule
similarity 0.4

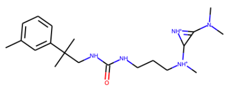

604: -2.94

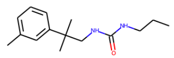

604: 3.39

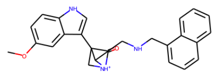

103: -5.40

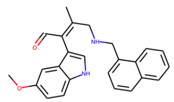

103: 0.88

Molecule
similarity 0.6


89: -5.64


89: 0.94


782: -2.57


782: 2.44

# Outline

- Graph ConvNets
- Molecule Generation
- **Conclusion**

# Conclusion

- We propose a simple and efficient VAE for atoms and bonds decoding.
- We report highest VAE accuracy on ZINC dataset for
  - Molecule reconstruction,
  - Molecule optimization of constrained solubility property.
- Comparing VAE+SL vs GAN+RL :
  - GAN+RL generates better molecules (outside the training statistics),
  - VAE+SL generates better optimized molecules similar to original ones,
  - GAN+RL generates optimized molecules with 100% success.

# Conclusion

- An alternative to auto-regressive graph NN methods.
  - We have solved the molecule generation task with :
    - Single-shot reconstruction + beam search
  - Simple and fast solution (GPU parallelizable)

- Next steps :
  - SL to RL : Learn molecules beyond training statistics.
  - Large molecules by hierarchical representation (represent molecules of N atoms with log(N) layers with graph coarsening)
  - Collaboration with domain experts to solve chemical tasks !

Questions?