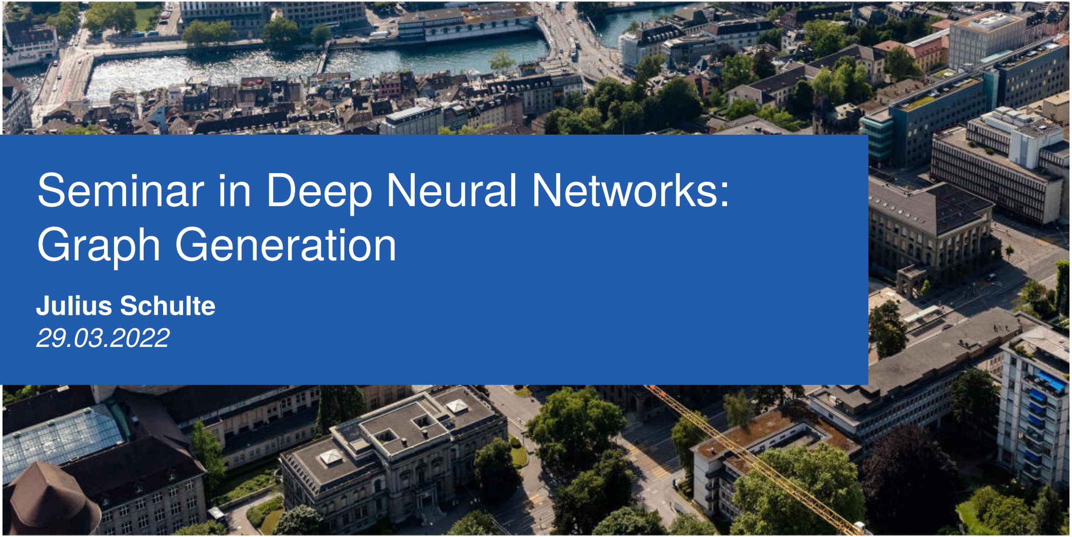


Seminar in Deep Neural Networks: Graph Generation

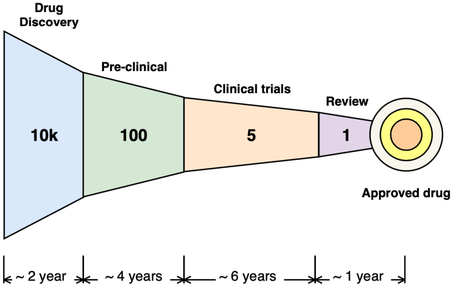
Julius Schulte

29.03.2022

Outline

1. Motivation & Background
2. GraphVAE
3. 3D Generative Model
4. Taking Stock
5. Sources

1. Motivation

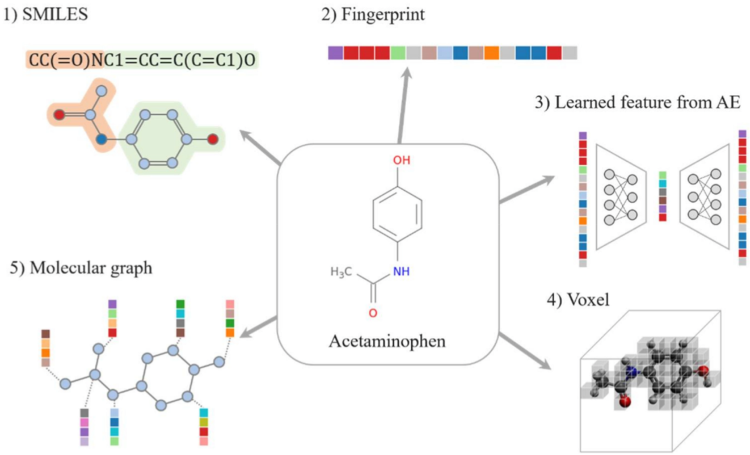


ML helps Drug Design Process

- Molecular Property Prediction
- Molecule Generation + Optimisation (our focus!).

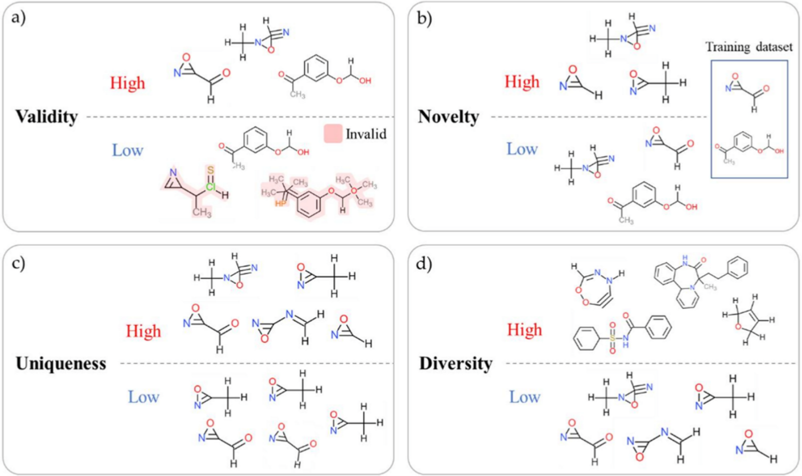
[De Cao 2022, INSILICO]

1.2 Molecule Representations



[Kim et al. 2021; David et al. 2020]

1.3 Molecule Evaluation



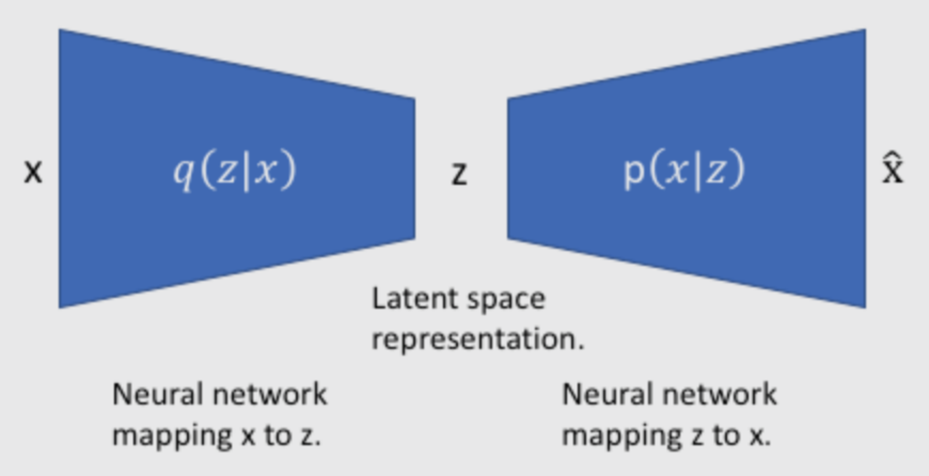
[Kim et al. 2021]

1.4 Graph Generation - Approaches

Paper	Authors	Year	Data	Classification
GraphVAE	Simonovsky et al.	2018	QM9, ZINC (2D)	VAE, A, ST
MolGAN	De Cao, Kipf	2018	QM9 (2D)	GAN (+), A, ST
GCPN	You et al.	2019	ZINC (2D)	RL, S, ST
liGAN	Masuda et al.	2020	CrossDocked (3D)	Conv+GAN, A, LI
3D Generative Model	Luo et al.	2021	CrossDocked (3D)	Autoregressive, S, LI

Generation: All at once → A, Sequentially → S
Target: Structure based → ST, Ligand based → LI

2. Variational Autoencoder

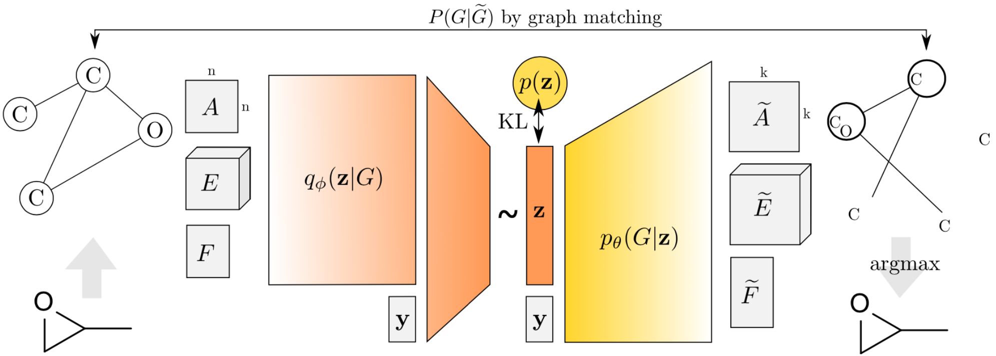

x
$q(z|x)$
z
$p(x|z)$
$\hat{x}$
Latent space representation.
Neural network mapping x to z.
Neural network mapping z to x.

[Image Credit: Jeremy Jordan ↗]

2.1 Challenges

- Graph size is dynamic
- Number of predicted nodes $\neq$ number of nodes ground truth → how to calculate the loss?
- Node ordering (graphs are isomorphic)

2.2 GraphVAE



[Simonovsky and Komodakis 2018]

2.3 Loss Function

$$\mathcal{L}(\phi, \theta; G) = \mathbb{E}_{q_\phi(Z|G)}[-\log p_\theta(G|z)] + KL[q_\phi(z|G)||p(z)] \tag{1}$$

with

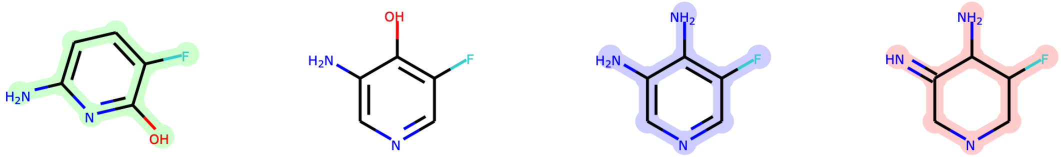
$$-\log p(G|z) = -\lambda_A \log p(A'|z) - \lambda_F \log p(F|z) - \lambda_E \log p(E|z) \tag{2}$$

[Simonovsky and Komodakis 2018]

2.4 Challenges Addressed

Challenge	GraphVAE Approach
Dynamic Graph Size	Fixed max graph size (k)
Number of predicted nodes $\neq$ number of nodes ground truth	Max Pooling Matching - polynomial:(
Node Ordering	Max Pooling Matching

2.5 Results I



- On average 50% of generated molecules are chemically valid.
- With larger embedding size the percentage of unique samples increases but accuracy decreases.
- About 60% of generated molecules are out of the data-set, i.e. the network has never seen them during training.

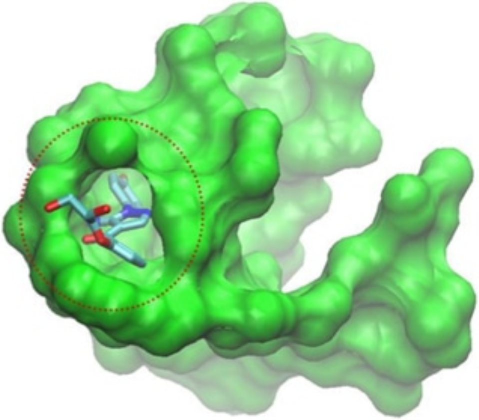
[Simonovsky and Komodakis 2018]

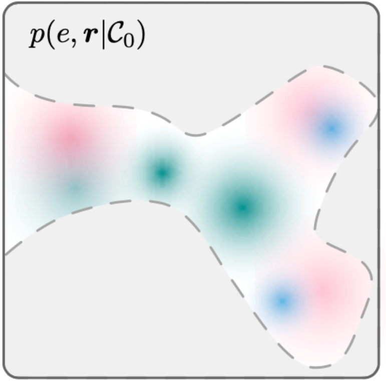
2.6 GraphVAE - Taking Stock

- Easy to train (especially in comparison to GANs)
- Graph matching is computationally expensive and thus scalability limited
- 2D input and output as well as scalability might lessen relevance

→ *How can we use these molecules?*

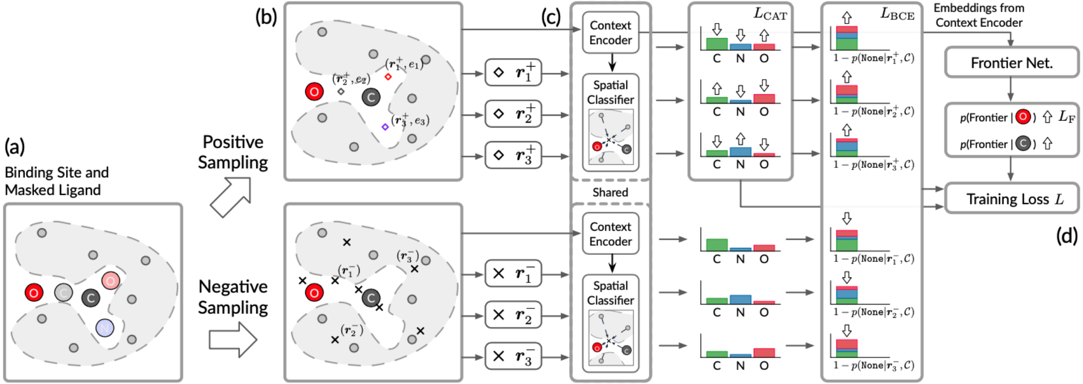
3. Ligand Identification





[Image Credit: Creative Proteomics ↗,Luo et al. 2021]

3.1 Overview



[Luo et al. 2021]

3.2 A 3D Generative Model for Structure-Based Drug Design

Goal: Generate of a "set of atoms that is able to form a valid drug-like molecule fitting to a specific binding site".

We define the **binding site** as $C = (a_i, r_i)_{i=1}^{N_b}$, where N_b is the number of atoms in the binding site, a_i is the i-th atom's attributes and r_i is its 3D coordinate.

1. Context Encoder
2. Spatial Classifier
3. Sampling

[Luo et al. 2021]

3.3 Context Encoder

The context encoder aims to create a representation that is both context aware and invariant to rotations as well as translations.

Input + Output

The **input** is a k-nearest neighbour graph (inter-atomic distances) denoted as $G = \langle C, A \rangle$.

The **output** are structure-aware node embeddings.

[Luo et al. 2021]

3.4 Spatial Classifier

The spatial classifier aims to predict the type of atom that occupies the position r, taking the context around r into account.

Input + Output

The **input** is a query position $r \in \mathbb{R}^3$. together with the atom embeddings from the context encoder:

$$v = \sum_{j \in N_k(r)} W_0 w_{\text{aggr}}(||r - r_j||) \odot W_1 h_j^{(L)} \tag{3}$$

$$c = \text{MLP}(v) \tag{4}$$

$$p(e|r, C) = \frac{\exp(C[e])}{1 + \sum_{e' \in \mathcal{E}} \exp\left(c[e']\right)} = \text{MLP}(v) \tag{5}$$

The **output** is given the position r and the binding site C the probability that we observe a certain type of atom e.

[Schütt et al. 2017; Luo et al. 2021]

3.5 Auto-Regressive Sampling

Sampling Procedure

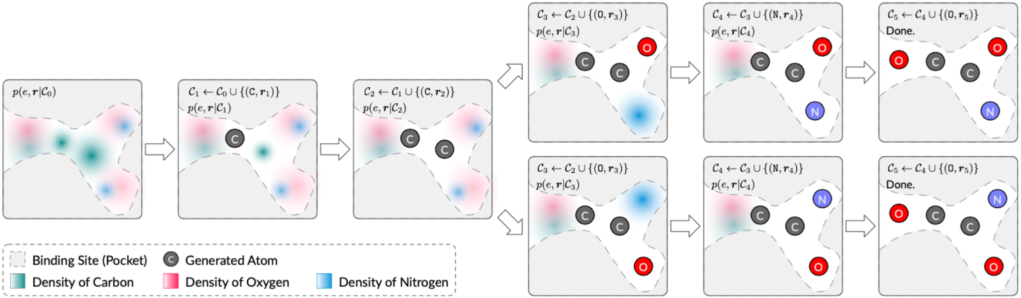
1. A joint distribution of an atom of a certain type e and the position r given the binding site C is defined (MCMC).

2. At each step one atom is sampled from the joint distribution taking into account the t atoms that have been sampled beforehand. This process is repeated until all sampled atoms are "non-frontier" (i.e. no space available anymore) and a binding molecule is obtained.

3. In the end, OpenBabel ↗ is used to generate final structures with bonds.

[Luo et al. 2021]

3.6 Auto-Regressive Sampling


$p(e, \boldsymbol{r}|\mathcal{C}_0)$
$\mathcal{C}_1 \leftarrow \mathcal{C}_0 \cup \{(\texttt{C}, \boldsymbol{r}_1)\}$
$p(e, \boldsymbol{r}|\mathcal{C}_1)$
$\mathcal{C}_2 \leftarrow \mathcal{C}_1 \cup \{(\texttt{C}, \boldsymbol{r}_2)\}$
$p(e, \boldsymbol{r}|\mathcal{C}_2)$
$\mathcal{C}_3 \leftarrow \mathcal{C}_2 \cup \{(\texttt{O}, \boldsymbol{r}_3)\}$
$p(e, \boldsymbol{r}|\mathcal{C}_3)$
$\mathcal{C}_4 \leftarrow \mathcal{C}_3 \cup \{(\texttt{N}, \boldsymbol{r}_4)\}$
$p(e, \boldsymbol{r}|\mathcal{C}_4)$
$\mathcal{C}_5 \leftarrow \mathcal{C}_4 \cup \{(\texttt{O}, \boldsymbol{r}_5)\}$
Done.
$\mathcal{C}_3 \leftarrow \mathcal{C}_2 \cup \{(\texttt{O}, \boldsymbol{r}_3)\}$
$p(e, \boldsymbol{r}|\mathcal{C}_3)$
$\mathcal{C}_4 \leftarrow \mathcal{C}_3 \cup \{(\texttt{N}, \boldsymbol{r}_4)\}$
$p(e, \boldsymbol{r}|\mathcal{C}_4)$
$\mathcal{C}_5 \leftarrow \mathcal{C}_4 \cup \{(\texttt{O}, \boldsymbol{r}_5)\}$
Done.
C
O
N
Binding Site (Pocket)
Generated Atom
Density of Carbon
Density of Oxygen
Density of Nitrogen

[Luo et al. 2021]

3.7 Training I

Procedure

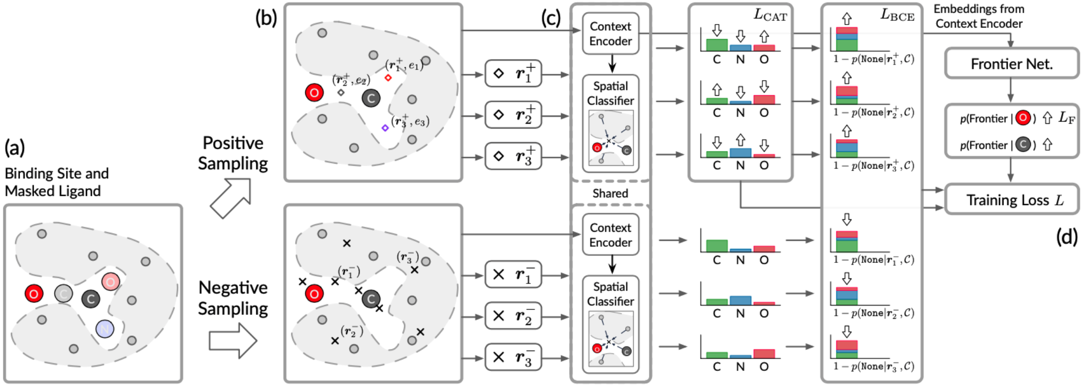
Mask random portion of target molecules during training and train with three loss functions:

1. L_{BCE} enables us to optimize for the prediction of a position that actually contains an atom.
2. L_{CAT} helps us to predict the chemical element of the atom.
3. L_F is necessary to make the sampling process stop. Note that F is the frontier network.

$$L = L_{\text{BCE}} + L_{CAT} + L_F \tag{6}$$

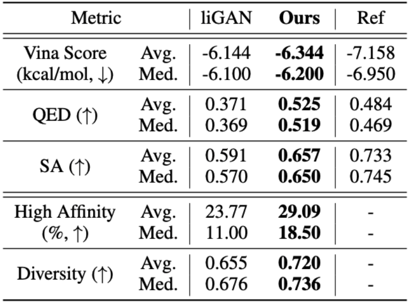
[Luo et al. 2021]

3.8 Training II

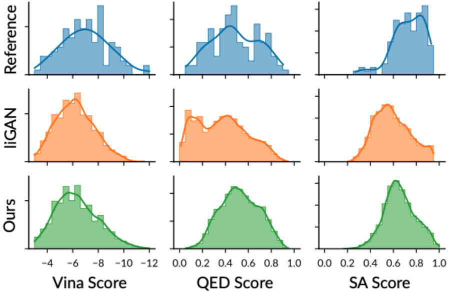


[Luo et al. 2021]

3.9 Results



Metric		liGAN	Ours	Ref
Vina Score (kcal/mol, ↓)	Avg.	-6.144	**-6.344**	-7.158
	Med.	-6.100	**-6.200**	-6.950
QED (↑)	Avg.	0.371	**0.525**	0.484
	Med.	0.369	**0.519**	0.469
SA (↑)	Avg.	0.591	**0.657**	0.733
	Med.	0.570	**0.650**	0.745
High Affinity (%, ↑)	Avg.	23.77	**29.09**	-
	Med.	11.00	**18.50**	-
Diversity (↑)	Avg.	0.655	**0.720**	-
	Med.	0.676	**0.736**	-



[Luo et al. 2021]

4. A 3D Generative Model - Taking Stock

Positives

- High relevance and applicability
- Innovative use of existing approaches
- Recent paper with corresponding limitations in benchmarking

To remark

- Gaps persist in the paper's explanation of the normalization constant, encoder method used etc.
- Limited reproducibility despite code published on Github
- Dependency on outside software (see OpenBabel)

5. Conclusion

- Generated molecules mostly small (<50 nodes).
- There is no clear winner in terms of architecture for graph generation.
- Exploration into 3D molecule representations and molecules present avenues for future research.

[Information Credit: Petar Veličković]

DISCUSSION

Sources I

[1] Laurianne David et al. “Molecular Representations in AI-driven Drug Discovery: A Review and Practical Guide”. In: *Journal of Cheminformatics* 12.1 (Sept. 17, 2020), p. 56. ISSN: 1758-2946. DOI: 10.1186/s13321-020-00460-5. URL: https://doi.org/10.1186/s13321-020-00460-5 (visited on 03/02/2022).

[2] Nicola De Cao. *Publications*. Nicola De Cao. URL: https://nicola-decao.github.io/publications/ (visited on 03/16/2022).

[3] Nicola De Cao and Thomas Kipf. “MolGAN: An Implicit Generative Model for Small Molecular Graphs”. 2018. arXiv: 1805.11973.

[4] Jintae Kim et al. “Comprehensive Survey of Recent Drug Discovery Using Deep Learning”. In: *International Journal of Molecular Sciences* 22.18 (2021). ISSN: 1422-0067. DOI: 10.3390/ijms22189983. URL: https://www.mdpi.com/1422-0067/22/18/9983.

Sources II

[5] Renjie Liao et al. *Efficient Graph Generation with Graph Recurrent Attention Networks*. 2020. arXiv: 1910.00760 [cs.LG].

[6] Shitong Luo et al. "A 3D Generative Model for Structure-Based Drug Design". In: *Advances in Neural Information Processing Systems* 34 (2021).

[7] Matthew Ragoza, Tomohide Masuda, and David Ryan Koes. "Generating 3D Molecules Conditional on Receptor Binding Sites with Deep Generative Models". In: *Chemical Science* 13.9 (2022), pp. 2701–2713.

[8] Roozbeh Razavi-Far et al. *Generative Adversarial Learning: Architectures and Applications*. 2022.

[9] Kristof Schütt et al. "Schnet: A Continuous-Filter Convolutional Neural Network for Modeling Quantum Interactions". In: *Advances in neural information processing systems* 30 (2017).

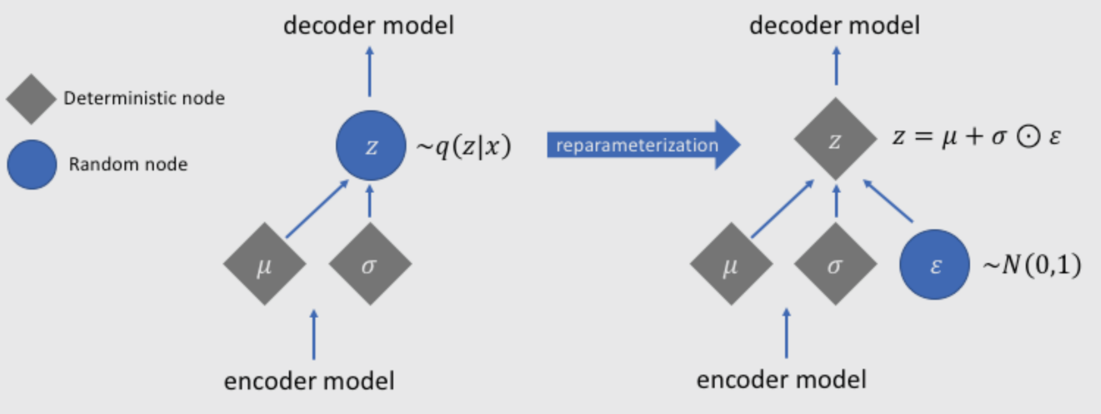
Sources III

[10] Martin Simonovsky and Nikos Komodakis. “GraphVAE: Towards Generation of Small Graphs Using Variational Autoencoders”. Feb. 9, 2018. arXiv: 1802.03480 [cs] ↗. URL: http://arxiv.org/abs/1802.03480 (visited on 03/06/2022).

[11] Jiaxuan You, Rex Ying, and Jure Leskovec. *Design Space for Graph Neural Networks*. 2021. arXiv: 2011.08843 [cs.LG] ↗.

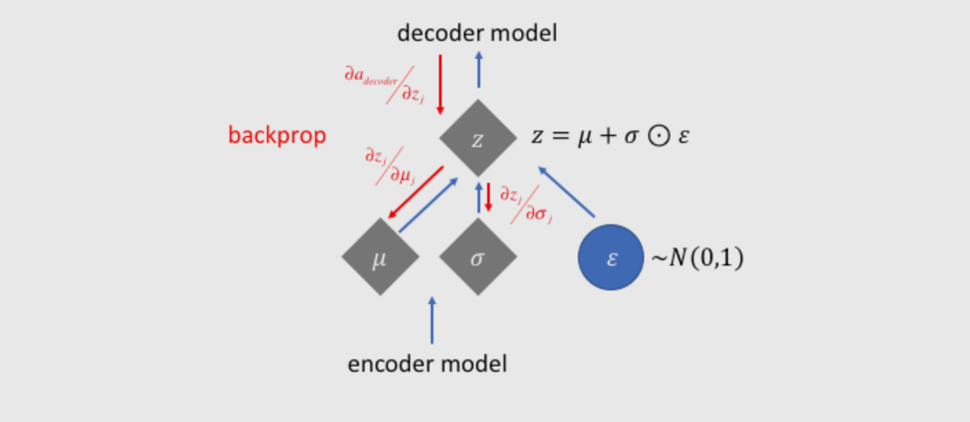
BACKUP

Reparametrization Trick I



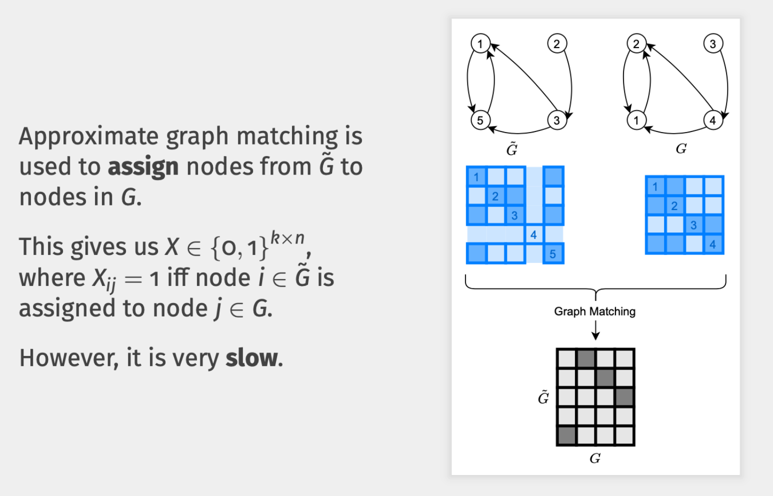
[Image Credit: Jeremy Jordan ↗]

Reparametrization Trick II


decoder model
$\partial a_{decoder} / \partial z_j$
backprop
z
$z = \mu + \sigma \odot \varepsilon$
$\partial z_j / \partial \mu_j$
$\partial z_j / \partial \sigma_j$
μ
σ
ε
$\sim N(0,1)$
encoder model

[Image Credit: Jeremy Jordan ↗]

Graph Matching



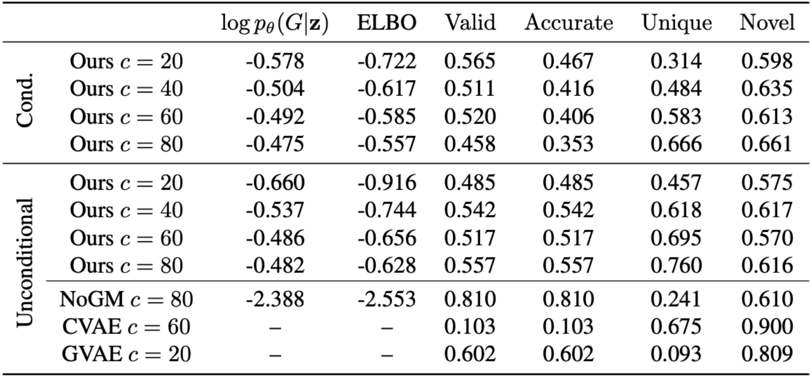
Approximate graph matching is used to **assign** nodes from $\tilde{G}$ to nodes in G.

This gives us $X \in \{0,1\}^{k \times n}$, where $X_{ij} = 1$ iff node $i \in \tilde{G}$ is assigned to node $j \in G$.

However, it is very **slow**.

[Image Credit: Harish Rajagopal, SiDNN 2021]

GraphVAE - Results II



		$\log p_\theta(G\|\mathbf{z})$	ELBO	Valid	Accurate	Unique	Novel
Cond.	Ours $c = 20$	-0.578	-0.722	0.565	0.467	0.314	0.598
	Ours $c = 40$	-0.504	-0.617	0.511	0.416	0.484	0.635
	Ours $c = 60$	-0.492	-0.585	0.520	0.406	0.583	0.613
	Ours $c = 80$	-0.475	-0.557	0.458	0.353	0.666	0.661
Unconditional	Ours $c = 20$	-0.660	-0.916	0.485	0.485	0.457	0.575
	Ours $c = 40$	-0.537	-0.744	0.542	0.542	0.618	0.617
	Ours $c = 60$	-0.486	-0.656	0.517	0.517	0.695	0.570
	Ours $c = 80$	-0.482	-0.628	0.557	0.557	0.760	0.616
	NoGM $c = 80$	-2.388	-2.553	0.810	0.810	0.241	0.610
	CVAE $c = 60$	–	–	0.103	0.103	0.675	0.900
	GVAE $c = 20$	–	–	0.602	0.602	0.093	0.809

[Simonovsky and Komodakis 2018]

Loss Functions Detail

Procedure

Mask random portion of target molecules during training and train with three loss functions:

$$L_{\text{BCE}} = -\mathbb{E}_{r_+}[\log(1 - p(\text{Nothing}|r, C))] - \mathbb{E}_{r\sim p_-}[\log p((\text{Nothing}|r, C)] \tag{7}$$

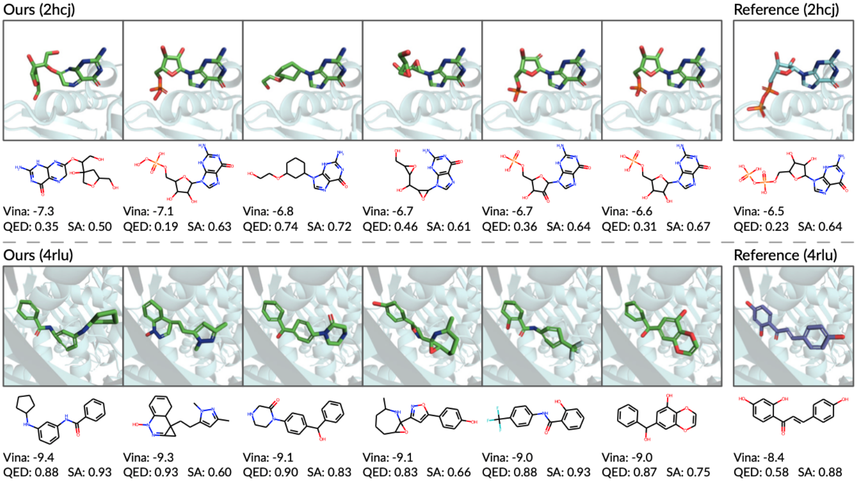
$$L_{CAT} = -\mathbb{E}_{(e,r)\sim p_+}[\log p((e|r, C)] \tag{8}$$

$$L_F = \sum_{i\in\mathcal{F}\subseteq C} \log \sigma(F(h_i)) + \sum_{i\notin\mathcal{F}\subseteq C} \log(1 - \sigma(F(h_i))) \tag{9}$$

$$L = L_{\text{BCE}} + L_{CAT} + L_F \tag{10}$$

[Luo et al. 2021]

Results II


Ours (2hcj)
Reference (2hcj)
Vina: -7.3 QED: 0.35 SA: 0.50
Vina: -7.1 QED: 0.19 SA: 0.63
Vina: -6.8 QED: 0.74 SA: 0.72
Vina: -6.7 QED: 0.46 SA: 0.61
Vina: -6.7 QED: 0.36 SA: 0.64
Vina: -6.6 QED: 0.31 SA: 0.67
Vina: -6.5 QED: 0.23 SA: 0.64
Ours (4rlu)
Reference (4rlu)
Vina: -9.4 QED: 0.88 SA: 0.93
Vina: -9.3 QED: 0.93 SA: 0.60
Vina: -9.1 QED: 0.90 SA: 0.83
Vina: -9.1 QED: 0.83 SA: 0.66
Vina: -9.0 QED: 0.88 SA: 0.93
Vina: -9.0 QED: 0.87 SA: 0.75
Vina: -8.4 QED: 0.58 SA: 0.88

[Luo et al. 2021]