동국대학교
dongguk university

MolGAN

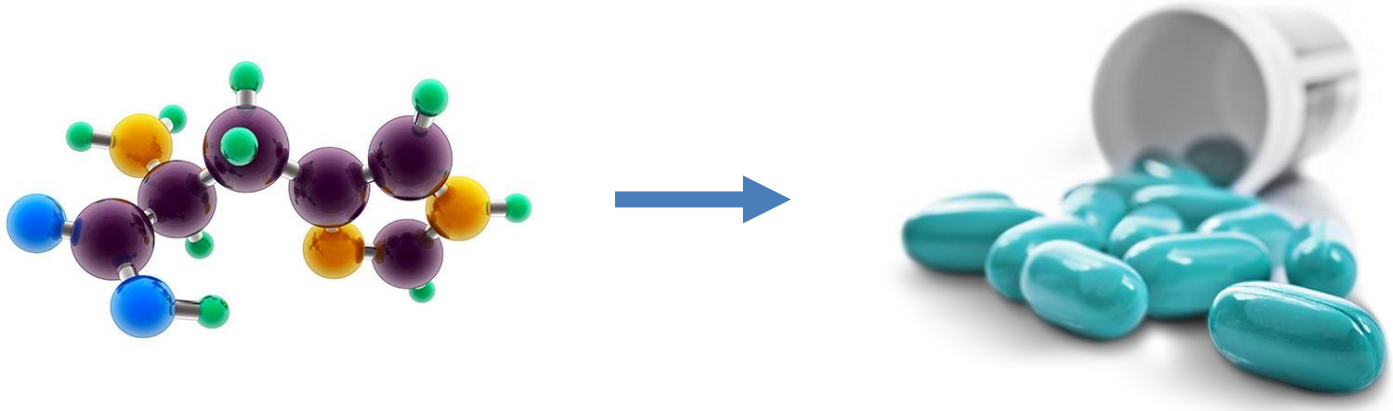
인공지능 연구실

2019.02

장성은

Drug design

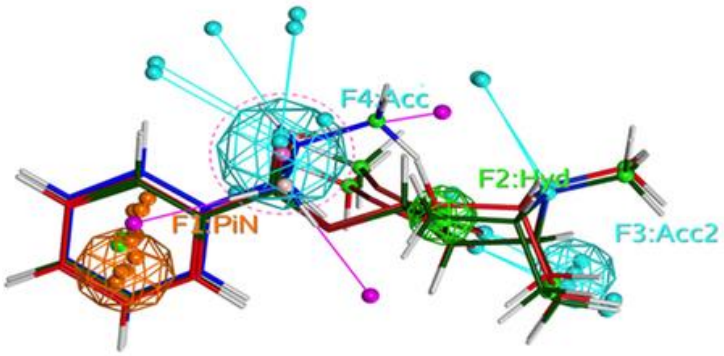
- **Drug design**
  - Inventive process of finding new medications based on the knowledge of a biological target.



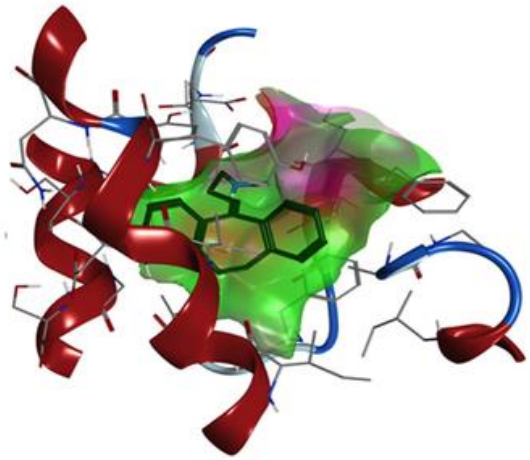
Drug design

- **Computer-aided drug design**

Ligand-Based Drug Design

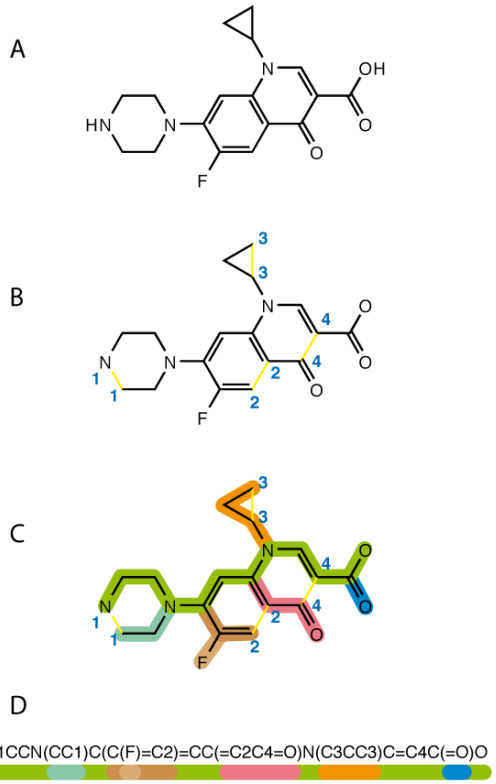


Structure-Based Drug Design



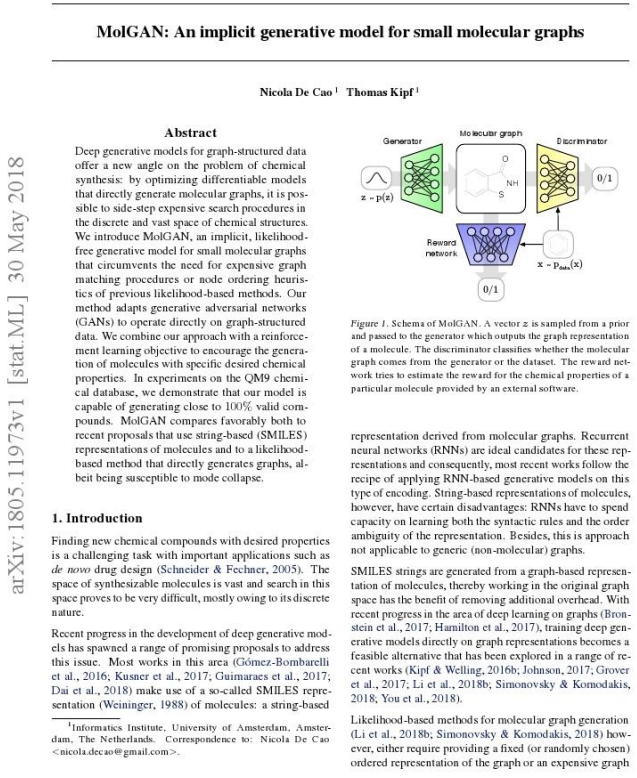
Molecular graphs

- **SMILES (Simplified molecular-input line-entry system)**
  - Representation of molecules
  - initiated by David Weininger at the USEPA Mid-Continent Ecology Division Laboratory in Duluth in the 1980s
  - A string-based representation derived from molecular graphs
  - Recurrent neural networks (RNNs) are ideal candidates for these representations



MolGAN: An implicit generative model for small molecular graphs

- **MolGAN**
  - An implicit, likelihood-free generative model for small molecular graphs
  - Circumvents the need for expensive graph matching procedures or node ordering heuristics of previous likelihood-based methods



MolGAN: An implicit generative model for small molecular graphs

Nicola De Cao [1] Thomas Kipf [1]

Abstract

Deep generative models for graph-structured data offer a new angle on the problem of chemical synthesis: by optimizing differentiable models that directly generate molecular graphs, it is possible to side-step expensive search procedures in the discrete and vast space of chemical structures. We introduce MolGAN, an implicit, likelihood-free generative model for small molecular graphs that circumvents the need for expensive graph matching procedures or node ordering heuristics of previous likelihood-based methods. Our method adapts generative adversarial networks (GANs) to operate directly on graph-structured data. We combine our approach with a reinforcement learning objective to encourage the generation of molecules with specific desired chemical properties. In experiments on the QM9 chemical database, we demonstrate that our model is capable of generating close to 100% valid compounds. MolGAN compares favorably both to recent proposals that use string-based (SMILES) representations of molecules and to a likelihood-based method that directly generates graphs, albeit being susceptible to mode collapse.

1. Introduction

Finding new chemical compounds with desired properties is a challenging task with important applications such as *de novo* drug design (Schneider & Fechner, 2005). The space of synthesizable molecules is vast and search in this space proves to be very difficult, mostly owing to its discrete nature.

Recent progress in the development of deep generative models has spawned a range of promising proposals to address this issue. Most works in this area (Gómez-Bombarelli et al., 2016; Kusner et al., 2017; Guimaraes et al., 2017; Dai et al., 2018) make use of a so-called SMILES representation (Weininger, 1988) of molecules: a string-based representation derived from molecular graphs. Recurrent neural networks (RNNs) are ideal candidates for these representations and consequently, most recent works follow the recipe of applying RNN-based generative models on this type of encoding. String-based representations of molecules, however, have certain disadvantages: RNNs have to spend capacity on learning both the syntactic rules and the order ambiguity of the representation. Besides, this is approach not applicable to generic (non-molecular) graphs.

SMILES strings are generated from a graph-based representation of molecules, thereby working in the original graph space has the benefit of removing additional overhead. With recent progress in the area of deep learning on graphs (Bronstein et al., 2017; Hamilton et al., 2017), training deep generative models directly on graph representations becomes a feasible alternative that has been explored in a range of recent works (Kipf & Welling, 2016b; Johnson, 2017; Grover et al., 2017; Li et al., 2018b; Simonovsky & Komodakis, 2018; You et al., 2018).

Likelihood-based methods for molecular graph generation (Li et al., 2018b; Simonovsky & Komodakis, 2018) however, either require providing a fixed (or randomly chosen) ordered representation of the graph or an expensive graph

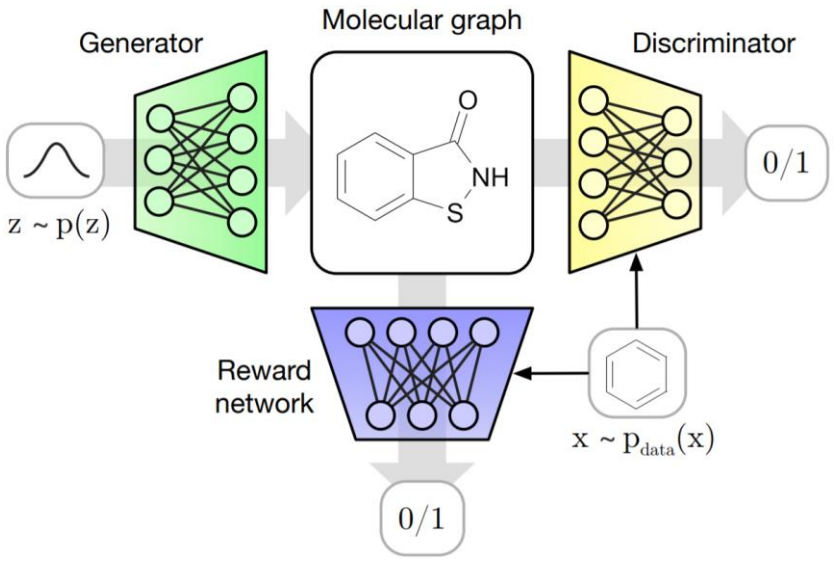
[1]Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands. Correspondence to: Nicola De Cao <nicola.decao@gmail.com>.

Figure 1. Schema of MolGAN. A vector z is sampled from a prior and passed to the generator which outputs the graph representation of a molecule. The discriminator classifies whether the molecular graph comes from the generator or the dataset. The reward network tries to estimate the reward for the chemical properties of a particular molecule provided by an external software.

MolGAN

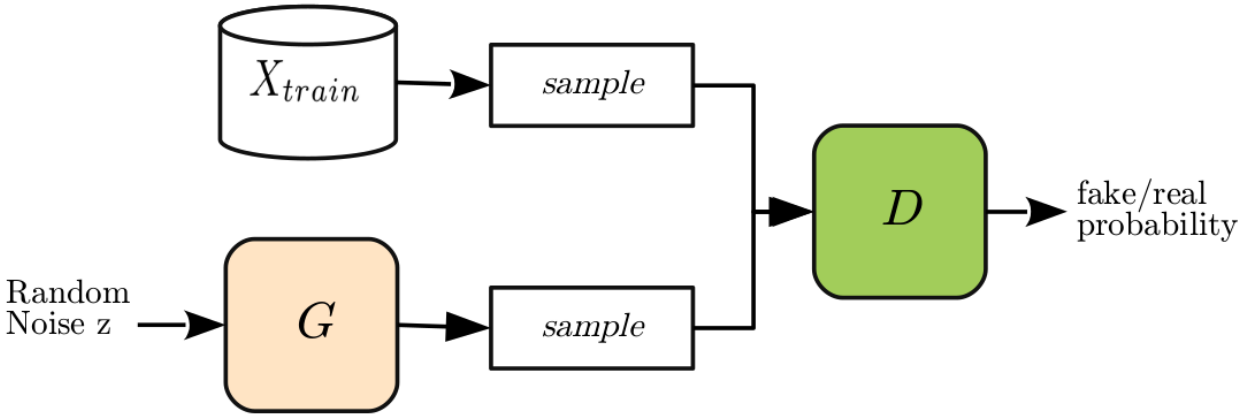
- **Schema of MolGAN**

GAN + Reinforcement learning



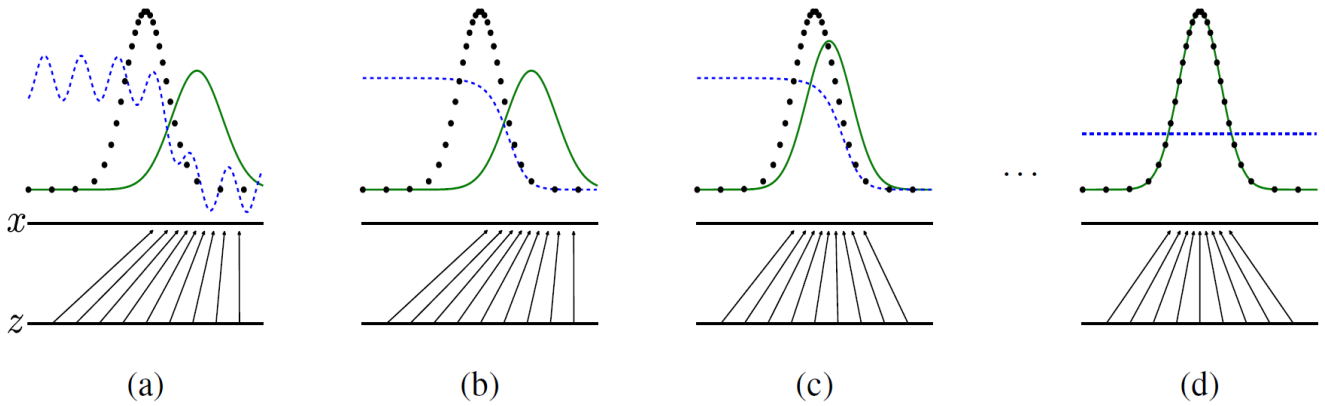
Generative Adversarial Nets

- **Generative Adversarial Nets**



$$D\big(G(z)\big) = D(X) = \frac{1}{2}$$

Generative Adversarial Nets



Black dotted line : Probability distribution of raw data
Green dotted line : Probability distributions of generator
Blue dotted line : Probability distribution of discriminator

- **Generative Adversarial Nets**

$$\min_G \max_D V(D,G)$$

$$= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

Algorithm

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

  for k steps **do**

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
  - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

- **Background - WGAN**
  - Wasserstein distance (Earth move distance)

$$W_p(P_r, P_\theta) = \inf_{\gamma \in \Gamma} E_{(x,y)\sim\gamma(x,y)}(|x-y|^p)$$

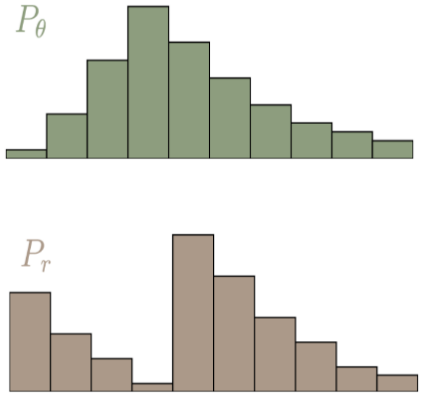


Fig.1: Probability distribution P_r and P_θ, each with ten states

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_\theta}[f(x)]$$

$$|f(x_1) - f(x_2)| \leq |x_1 - x_2|.$$

- **Background - WGAN**
  - WGAN

	Discriminator/Critic	Generator
GAN	$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right]$	$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)$
WGAN	$\nabla_w \frac{1}{m} \sum_{i=1}^{m} \left[f\left(x^{(i)}\right) - f\left(G\left(z^{(i)}\right)\right) \right]$ $w \leftarrow \text{clip}(w, -c, c)$	$\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} f\left(G\left(z^{(i)}\right)\right)$

- **Background - WGAN**
  - Improved WGAN

$$w \leftarrow \text{clip}(w, -c, c)$$

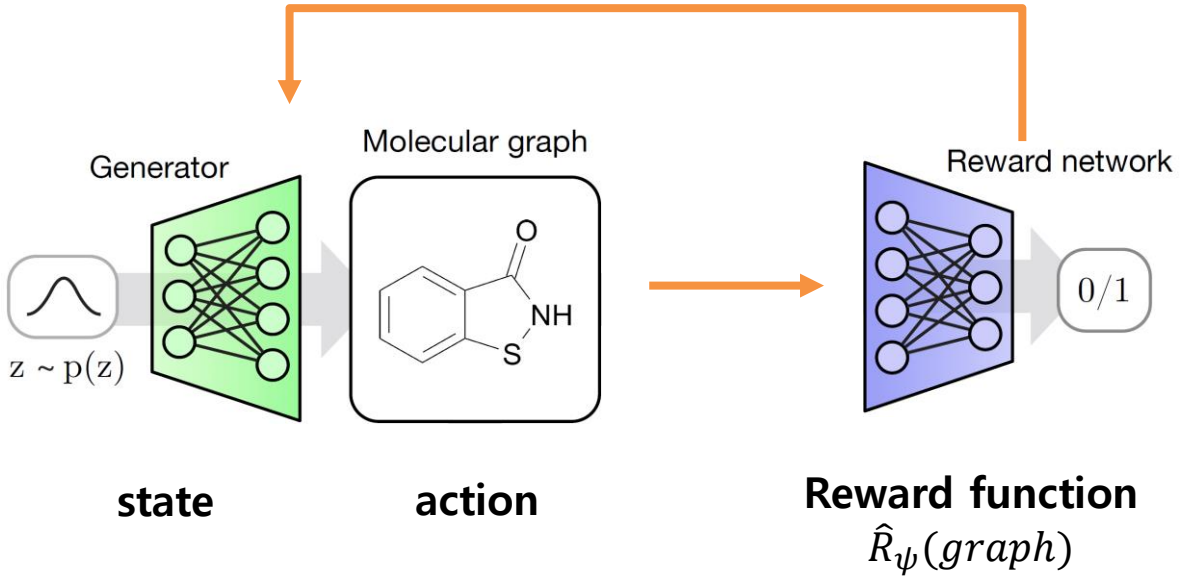
$$\nabla_w \frac{1}{m} \sum_{i=1}^{m} \left[f\left(x^{(i)}\right) - f\left(G\left(z^{(i)}\right)\right) \right]$$

$$L_D^{WGAN-GP} = L_D^{WGAN} + \alpha \left(\|\nabla_{\hat{\boldsymbol{x}}^{(i)}} D_\phi(\hat{\boldsymbol{x}}^{(i)})\| - 1 \right)^2$$

gradient penalty

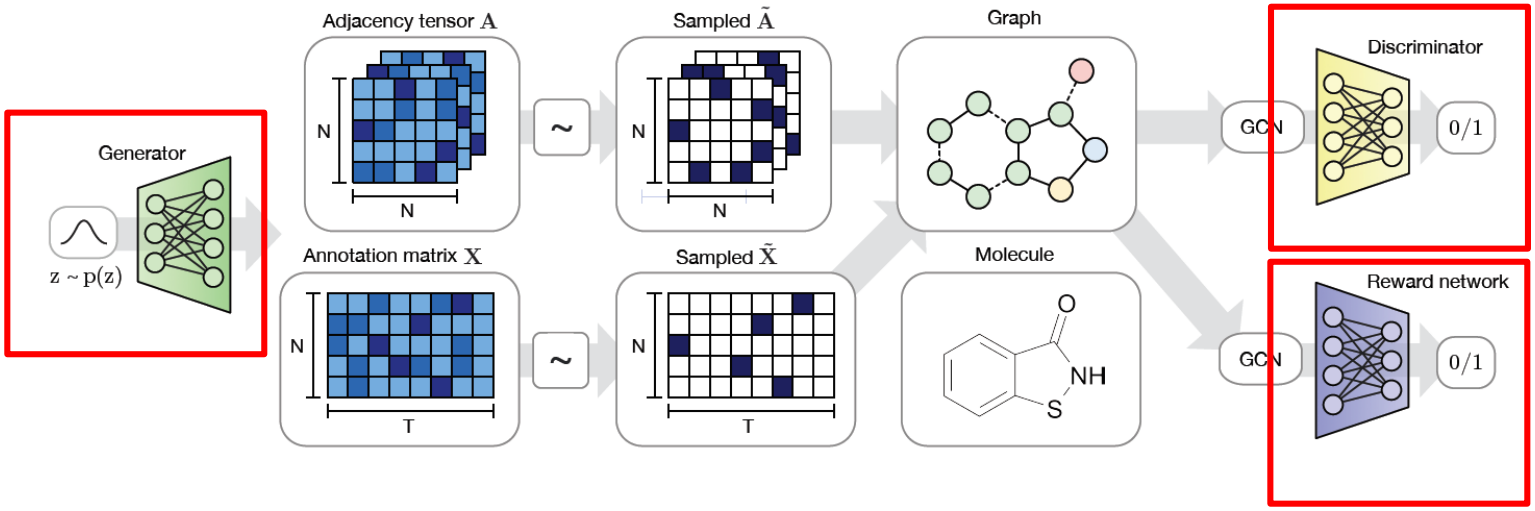
MolGAN

- **Background – Deep deterministic policy gradients**



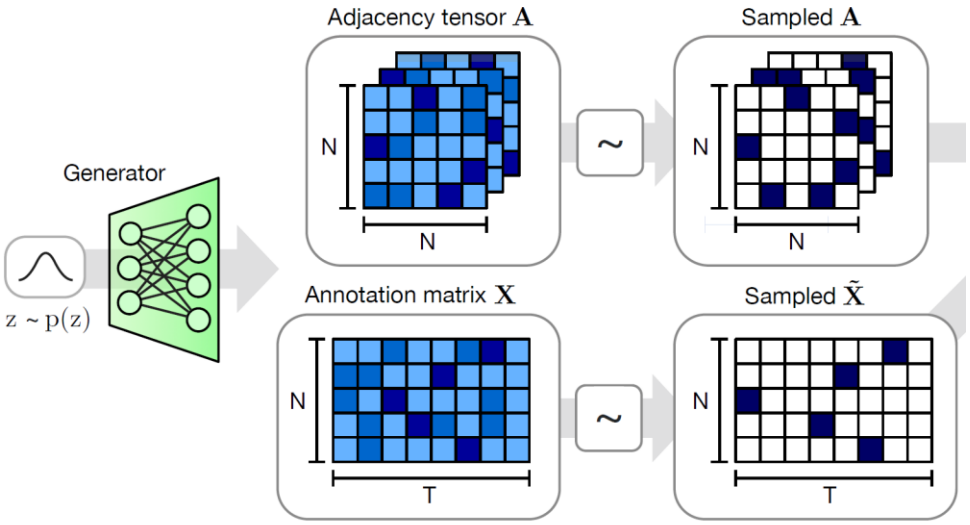
MolGAN

- **Model**

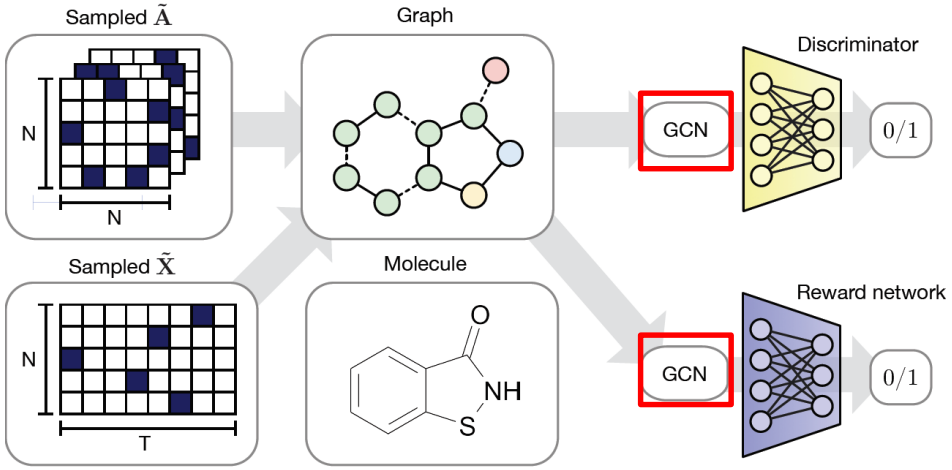


$$L(\theta) = \lambda \cdot L_{WGAN} + (1 - \lambda) \cdot L_{RL}$$

- **Model – Generator**



- **Model - Discriminator and reward network**



- **Model - Discriminator and reward network**
  - Graph convolution

$$\boldsymbol{h}_i^{\prime(\ell+1)} = f_s^{(\ell)}(\boldsymbol{h}_i^{(\ell)}, \boldsymbol{x}_i) + \sum_{j=1}^{N}\sum_{y=1}^{Y} \frac{\tilde{\boldsymbol{A}}_{ijy}}{|\mathcal{N}_i|} f_y^{(\ell)}(\boldsymbol{h}_j^{(\ell)}, \boldsymbol{x}_i) \ ,$$

$$\boldsymbol{h}_i^{(\ell+1)} = \tanh(\boldsymbol{h}_i^{\prime(\ell+1)}) \ ,$$

$h_i^{(l)} = signal\ of\ the\ node\ i\ at\ layer\ l$

$f_s^{(l)} = linear\ transformation\ function\ that\ acts\ as\ a\ self-connection\ between\ layers$

$f_y^{(l)} = edge\ type-specific\ affine\ function\ for\ each\ layer$

$N_i = set\ of\ neighbors\ for\ node\ i$

- **Model - Discriminator and reward network**
  - Node embedding

$$\boldsymbol{h}'_{\mathcal{G}} = \sum_{v \in \mathcal{V}} \sigma(i(\boldsymbol{h}_v^{(L)}, \boldsymbol{x}_v)) \odot \tanh(j(\boldsymbol{h}_v^{(L)}, \boldsymbol{x}_v)) \ ,$$

$$\boldsymbol{h}_{\mathcal{G}} = \tanh \boldsymbol{h}'_{\mathcal{G}} \ ,$$

Molecular graphs

- **Experiments**
  - Studying the effect of the λ parameter to find the best trade-off between the GAN and RL objective

$$L(\theta) = \lambda \cdot L_{WGAN} + (1 - \lambda)$$

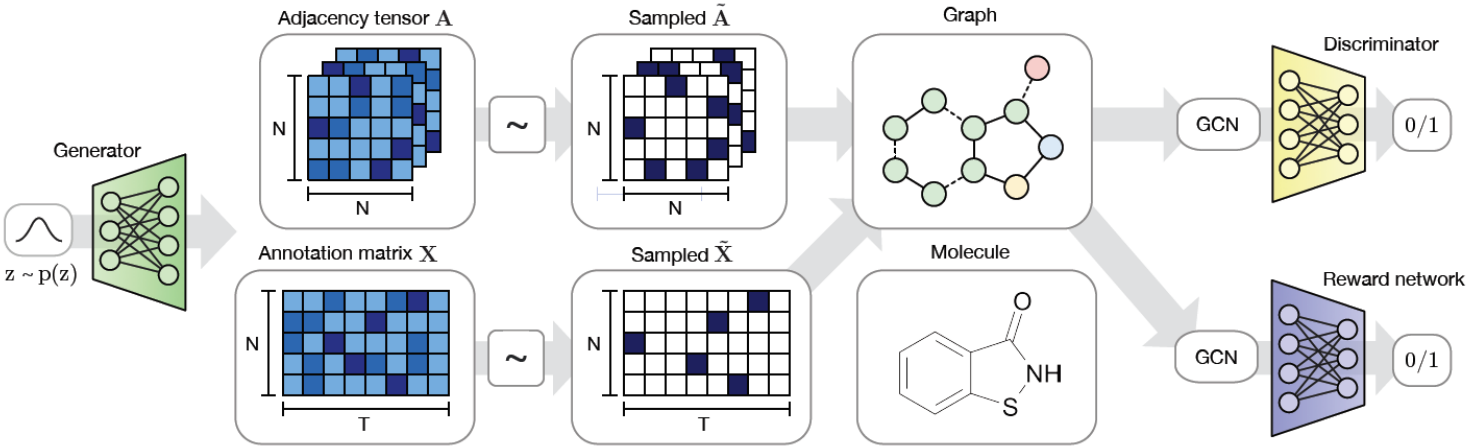
  - Compare MolGAN with ORGAN
    - druglikeness, solubility, synthetizability
  - Compare MolGAN against variational autoencoding methods
    - CharacterVAE, GrammarVAE, GraphVAE

Molecular graphs

- **Experiments – dataset**
  - GDB-17 Dataset
    - dataset composed of 166.4 billion molecules of up to 17 atoms of C, N, O, S, and halogens
    - contains millions of isomers of known drugs
  - QM9 Dataset
    - Subset of GDB-13
    - contains 133,885 organic compounds up to 9 heavy atoms

Molecular graphs

- **Experiments – model**



Molecular graphs

- **Experiments – evaluation measures**
  - **Validity**
    - number of valid molecules / number of all generated molecules
  - **Novelty**
    - set of valid samples that are not in the dataset / total number of valid samples
  - **Uniqueness**
    - number of unique samples and valid samples and it measures the degree of variety during sampling.

- **Experiments – results**
  - Effect of λ

Algorithm	Valid	Unique	Novel	Solubility
$\lambda = 0$ (full RL)	**99.8**	2.3	97.9	**0.86**
$\lambda = 0.01$	98.2	2.2	**98.1**	0.74
$\lambda = 0.05$	92.2	2.7	95.0	0.67
$\lambda = 0.1$	87.3	**3.2**	87.2	0.56
$\lambda = 0.25$	88.2	2.1	88.2	0.65
$\lambda = 0.5$	86.6	2.1	87.5	0.48
$\lambda = 0.75$	89.6	2.8	89.6	0.57
$\lambda = 1$ (no RL)	87.7	2.9	97.7	0.54

Table 1. Comparison of different combinations of RL and GAN objectives on the small 5k dataset after GAN-based pretraining for 150 epochs. All values are reported in percentages except for the solubility score.

Experiments – results

Objectives optimization

Objective	Algorithm	Valid (%)	Unique (%)	Time (h)	Diversity	Druglikeliness	Synthesizability	Solubility
Druglikeliness	ORGAN	88.2	69.4*	9.63*	0.55	0.52	0.32	0.35
	OR(W)GAN	85.0	8.2*	10.06*	0.95	0.60	0.54	0.47
	Naive RL	97.1	54.0*	9.39*	0.80	0.57	0.53	0.50
	MolGAN	**99.9**	2.0	1.66	0.95	**0.61**	0.68	0.52
	MolGAN (QM9)	**100.0**	2.2	4.12	**0.97**	**0.62**	0.59	0.53
Synthesizability	ORGAN	96.5	45.9*	8.66*	0.92	0.51	0.83	0.45
	OR(W)GAN	97.6	30.7*	9.60*	**1.00**	0.20	0.75	0.84
	Naive RL	97.7	13.6*	10.60*	0.96	0.52	0.83	0.46
	MolGAN	**99.4**	2.1	1.04	0.75	0.52	**0.90**	0.67
	MolGAN (QM9)	**100.0**	2.1	2.49	0.95	0.53	**0.95**	0.68
Solubility	ORGAN	94.7	54.3*	8.65*	0.76	0.50	0.63	0.55
	OR(W)GAN	94.1	20.8*	9.21*	0.90	0.42	0.66	0.54
	Naive RL	92.7	100.0*	10.51*	0.75	0.49	0.70	0.78
	MolGAN	**99.8**	2.3	0.58	**0.97**	0.45	0.42	**0.86**
	MolGAN (QM9)	**99.8**	2.0	1.62	**0.99**	0.44	0.22	**0.89**
All/Alternated	ORGAN	96.1	97.2*	10.2*	0.92	**0.52**	0.71	0.53
All/Simultaneously	*MolGAN*	**97.4**	2.4	2.12	0.91	0.47	**0.84**	**0.65**
All/Simultaneously	*MolGAN (QM9)*	**98.0**	2.3	5.83	**0.93**	0.51	**0.82**	**0.69**

Table 2. Gray cells indicate directly optimized objectives. Baseline results are taken from Guimaraes et al. (2017) (Table 1) and * indicates results reproduced by us using the code provided by the authors.

Molecular graphs

- **Experiments – results**
  - VAE Baselines

Algorithm	Valid	Unique	Novel
CharacterVAE	10.3	67.5	90.0
GrammarVAE	60.2	9.3	80.9
GraphVAE	55.7	**76.0**	61.6
GraphVAE/imp	56.2	42.0	75.8
GraphVAE NoGM	81.0	24.1	61.0
MolGAN	**98.1**	10.4	**94.2**

Table 3. Comparison with different algorithms on QM9. Values are reported in percentages. Baseline results are taken from Simonovsky & Komodakis (2018).

Molecular graphs

Conclusion

- Model is capable of generating molecular graphs with both higher validity and novelty than previous comparable VAE-based generative models
- Compared to a recent SMILES-based sequential GAN model for molecular generation, MolGAN can achieve higher chemical property while allowing for at least 5times faster training time

---

- Limitation of our current formulation is their susceptibility to mode collapse

Reference

- Ferreira, Leonardo & Santos, Ricardo & Oliva, Glaucius & Andricopulo, Adriano. (2015). Molecular Docking and Structure-Based Drug Design Strategies. Molecules.
- J Wasko, Michael & Pellegrene, Kendy & Madura, Jeffry & Surratt, Christopher. (2015). A Role for Fragment-Based Drug Design in Developing Novel Lead Compounds for Central Nervous System Targets. Frontiers in neurology.
- Goodfellow, Ian, et al. "Generative adversarial nets." *Advances in neural information processing systems*. 2014.
- De Cao, Nicola, and Thomas Kipf. "MolGAN: An implicit generative model for small molecular graphs." *arXiv preprint arXiv:1805.11973* (2018).
- Arjovsky, Martin, Soumith Chintala, and Léon Bottou. "Wasserstein generative adversarial networks." International Conference on Machine Learning. 2017.
- Gulrajani, Ishaan, et al. "Improved training of wasserstein gans." Advances in Neural Information Processing Systems. 2017.

Thank you !