Development of Machine Learning Systems for Drug Discovery

Seongok Ryu

KAIST Chemistry, AITRICS

Who am I?

- Seongok Ryu, 류성옥

- Education
  - ✓ 2020.02 ~ : Research Scientist, AITRICS
  - ✓ 2014.02 ~ 2020. 02: Ph.D in Chemistry, KAIST,
  - ✓ 2009.02 ~ 2014. 02: B.S. in Chemistry and Physics, KAIST

https://github.com/SeongokRyu

KAIST CHEMISTRY

Research motivations

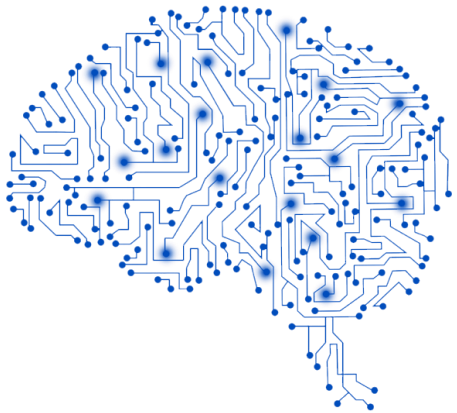
KAIST

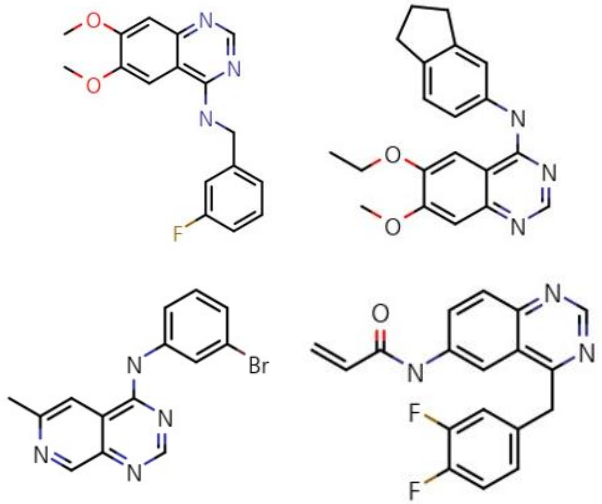
Research motivations

Research goal) Comprehensive platform for generation, prediction, and real-world assay

EGFR active?

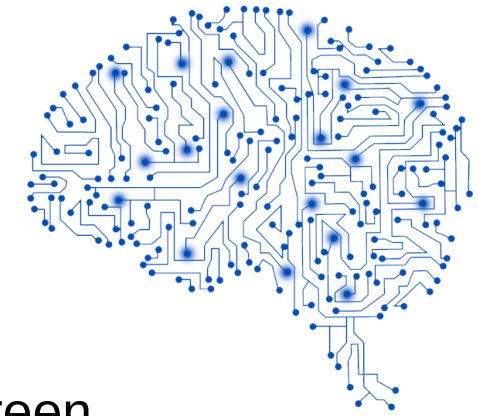
Generation



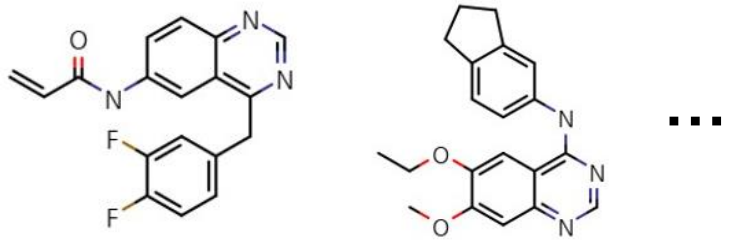


...

Evaluation (prediction models)

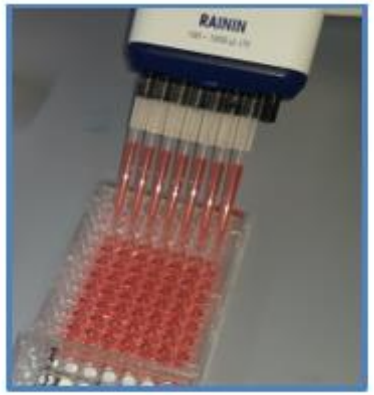


To screen
- High efficacy
- Non-toxic



...

Real-world assays

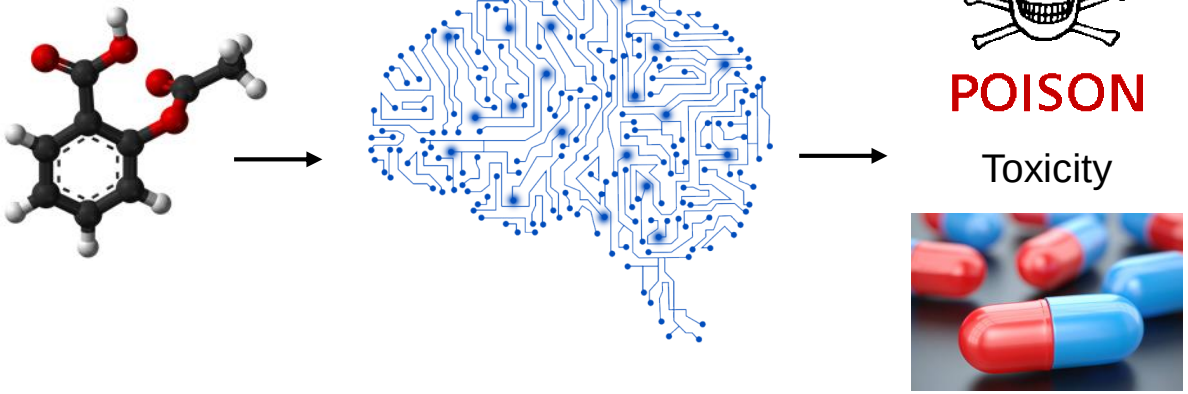


Feedback
- Gathering more data
- Active learning

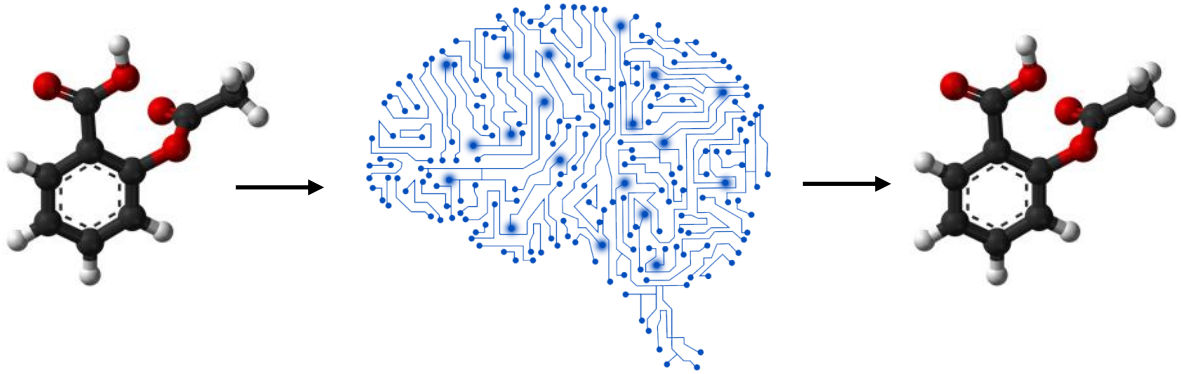
Research motivations

Two major uses of deep neural networks for molecular applications

1. Predicting molecular properties



2. Generating novel compounds



Possible applications)

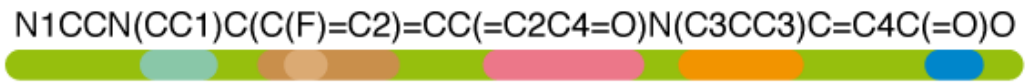
- *In silico* predictions of bio-activity, toxicity and other ADME properties
- *De novo* molecular design of novel compounds with desired properties
- Planning chemical synthesis

Research motivations

Need for well-designed neural networks for dealing with commonly used structure representations

- **Data representations of molecules**

Simplified **M**olecular **I**nput **L**ine **E**ntry **S**ystem



Pros)

- 1-dimensional sequence of characters
  → easy to handle
- CNNs/RNNs well-studied in NLP, speech tasks can be applied

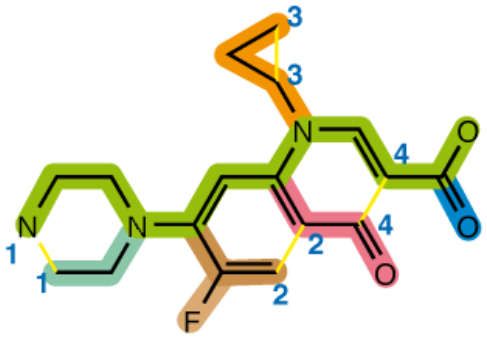
Cons)

- Similar structures may be encoded into quite different SMILES

Cc1cn2c(CN(C)C(=O)c3ccc(F)cc3C)c(C)nc2s1
Cc1cc(F)ccc1C(=O)N(C)Cc1c(C)nc2scc(C)n12

Source: Wengong Jin et. al., ICML 2018

Molecular graph



Source: SMILES, Wikipeda

Pros)

- Can represent geometry of molecular structures
- Can represent useful atomic/bond information with node/edge features

Cons)

- Should satisfy permutation invariance under node ordering
- Decoding process for generative models is difficult.

Research motivations

Data-hungry problems make developing reliable machine learning systems difficult.

- There is no plentiful labeled data for *in silico* drug discovery,
- Because labeling processes demand cost-expensive/laborious/time-consuming experiments.

	VGGNet	DeepVideo	GNMT
Used For	Identifying Image Category	Identifying Video Category	Translation
Input	Image 	Video 	English Text 
Output	1000 Categories	47 Categories	French Text
Parameters	140M	~100M	380M
Data Size	1.2M Images with assigned Category	1.1M Videos with assigned Category	6M Sentence Pairs, 340M Words
Dataset	ILSVRC-2012	Sports-1M	WMT'14

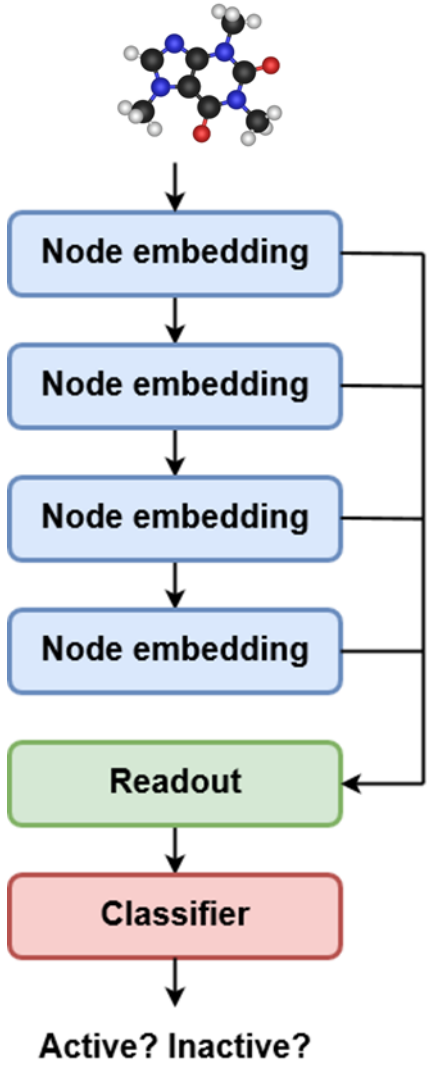
https://medium.com/nanonets/nanonets-how-to-use-deep-learning-when-you-have-limited-data-f68c0b512cab

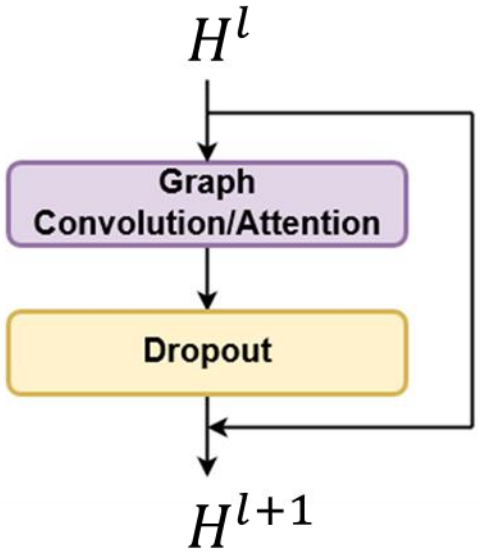
	Bio-assays	BBBP	Tox21
Used for	To predict bio-activity on a certain target	To predict blood-brain barrier permeability	To predict toxicity
Input	SMILES, Graph	SMILES, Graph	SMILES, Graph
Output	Binary label/ Continuous value	Binary label	Binary label
Data Size	**100 ~ 30,000**	**2,000**	**10,000**

Research motivations

My contributions:

1. Comprehensive study on graph convolutional networks for predicting molecular properties

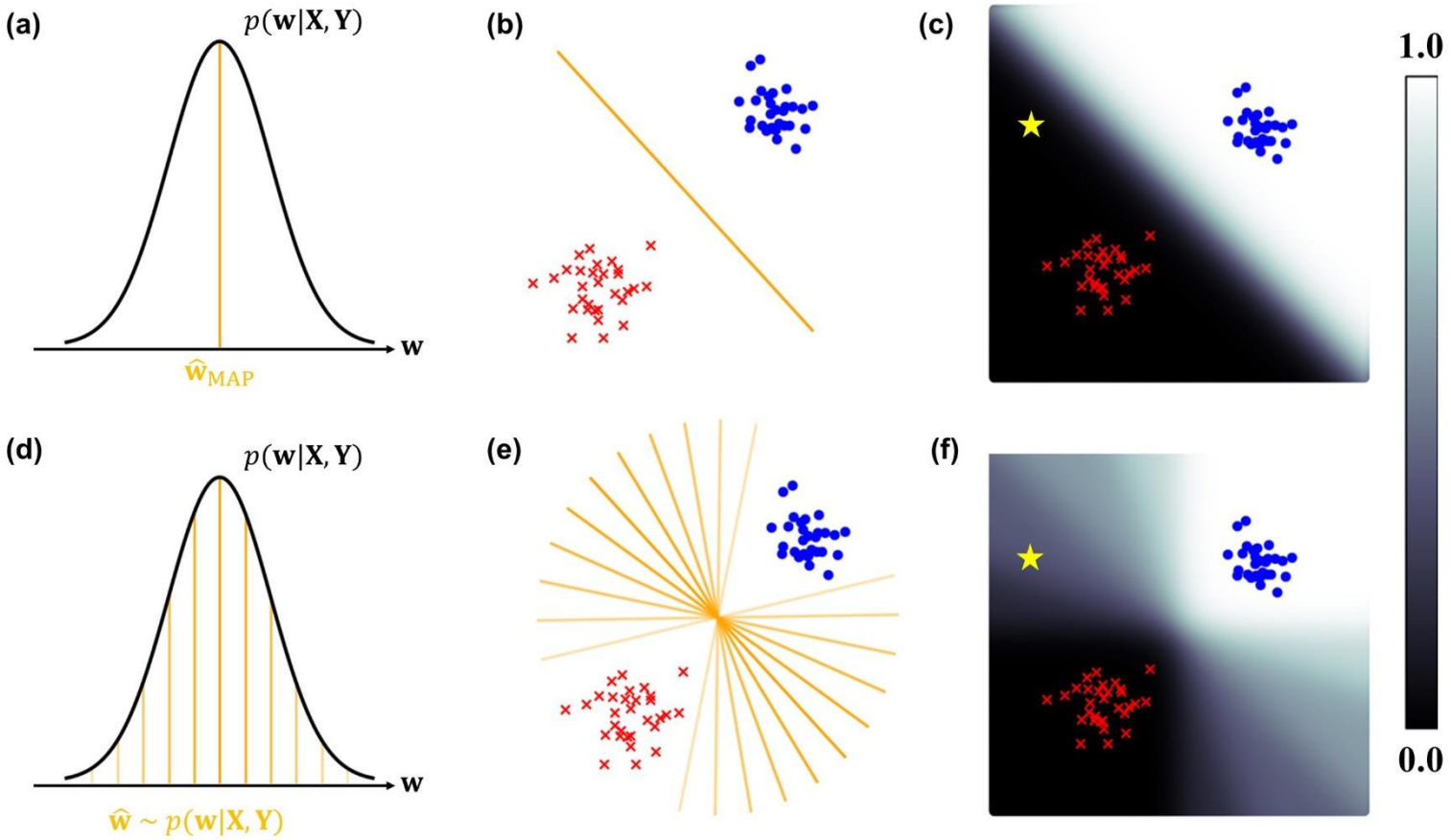


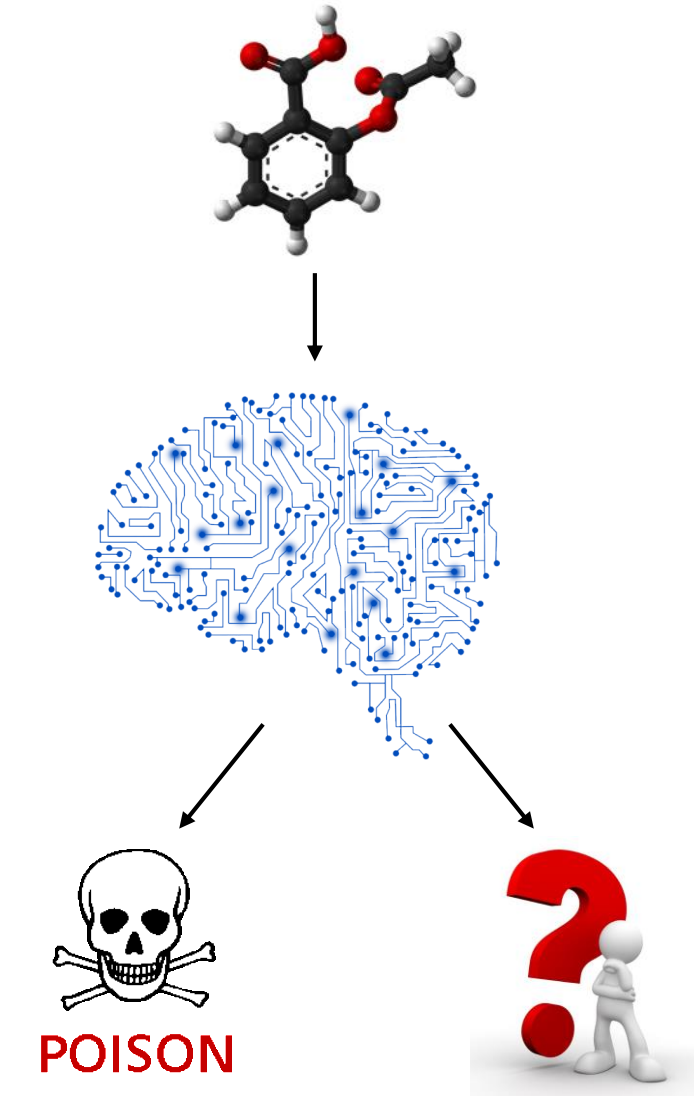


In preparation

Research motivations

My contributions:
2. Bayesian deep learning for drug discovery
3. Reliability of prediction systems





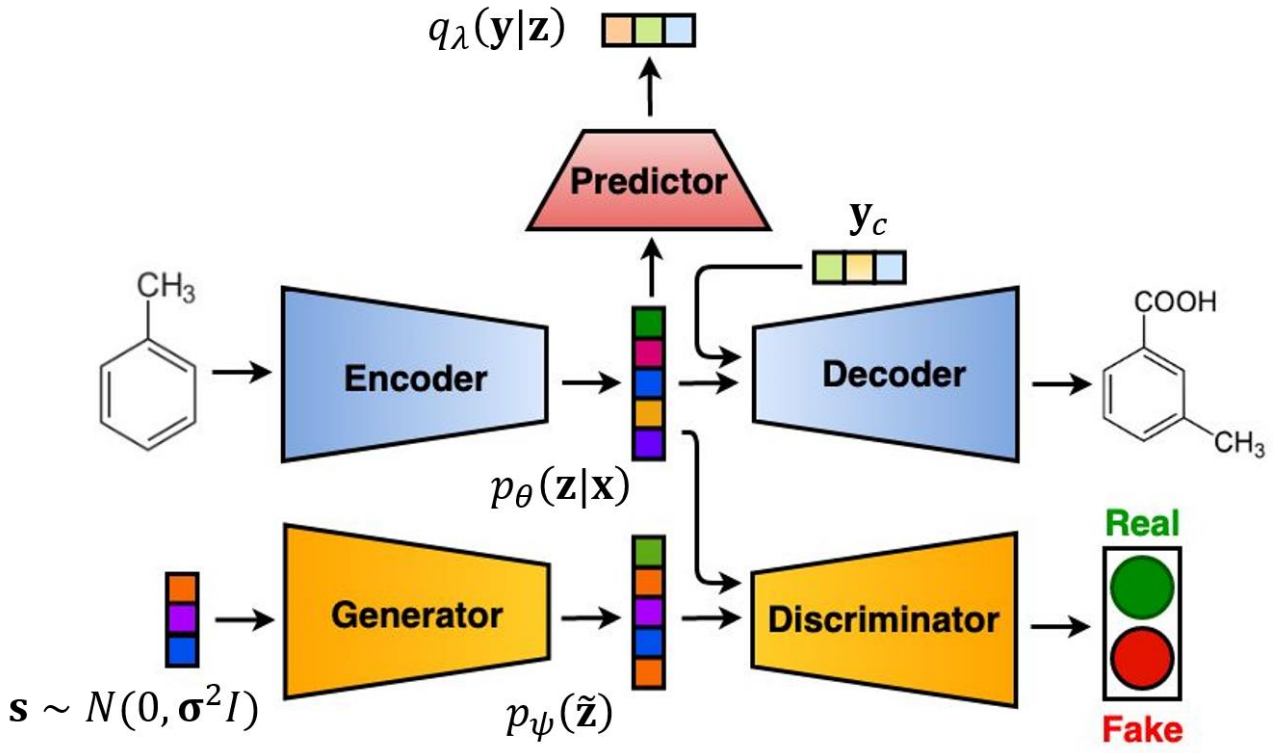
Ryu, Seongok, Yongchan Kwon, and Woo Youn Kim. "A Bayesian graph convolutional network for reliable prediction of molecular properties with uncertainty quantification." *Chemical Science* 10.36 (2019): 8438-8446.

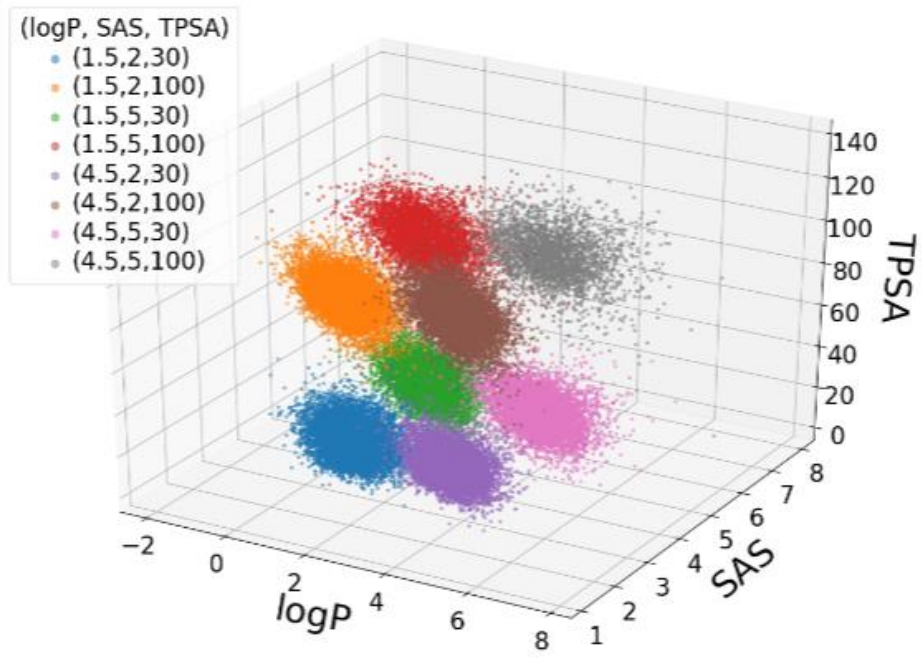
Research motivations

My contributions:

4. Molecular generative models based on adversarially regularized autoencoders (ARAEs)

- Improving molecular generations by overcoming drawbacks in previous models, i.e. VAEs and GANs.





Seung Hwan Hong, Ryu, Seongok, Jaechang Lim, and Woo Youn Kim.
"Molecular generative model based on an adversarially regularized autoencoder", JCIM (2019)

KAIST CHEMISTRY

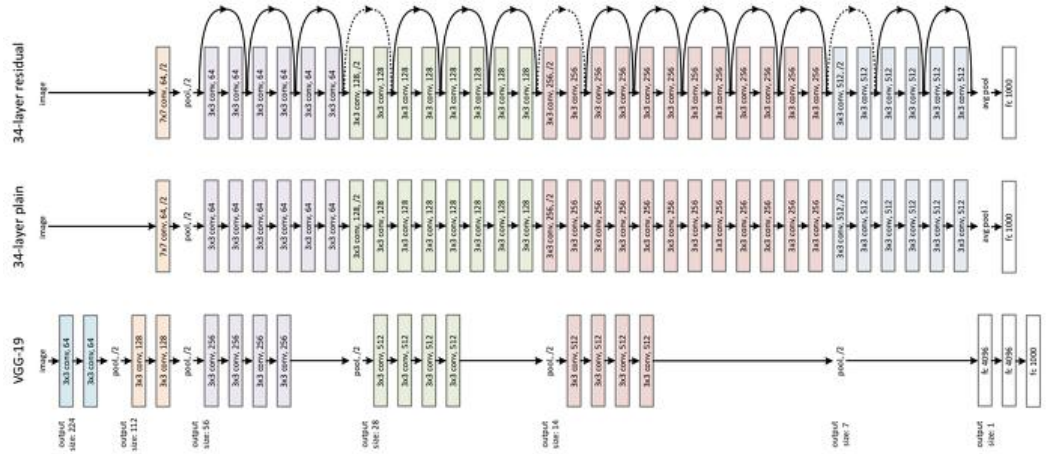
Graph convolutional networks for property prediction systems

KAIST

GCN for property prediction systems

Benchmark models and datasets in computer vision researches

- Models
  - ✓ VGG-16, 19, …
  - ✓ ResNet-32, 44, 110, …
  - ✓ WideResNet-28/10
  - ✓ …
- Datasets
  - ✓ MNIST
  - ✓ CIFAR-10, 100
  - ✓ ImageNet
  - ✓ …



GCN for property prediction systems

There is no shared benchmark models and datasets for chemistry-ML research

- Researchers have done experiments with their own models and hyper-parameter settings.
- Datasets have a few labeled examples, and they are usually imbalanced
- Moreover, different research perform dataset splitting with different random seeds.
  - ✓ Hard to reproduce experimental results
  - ✓ Therefore, proper ablation studies must be performed.

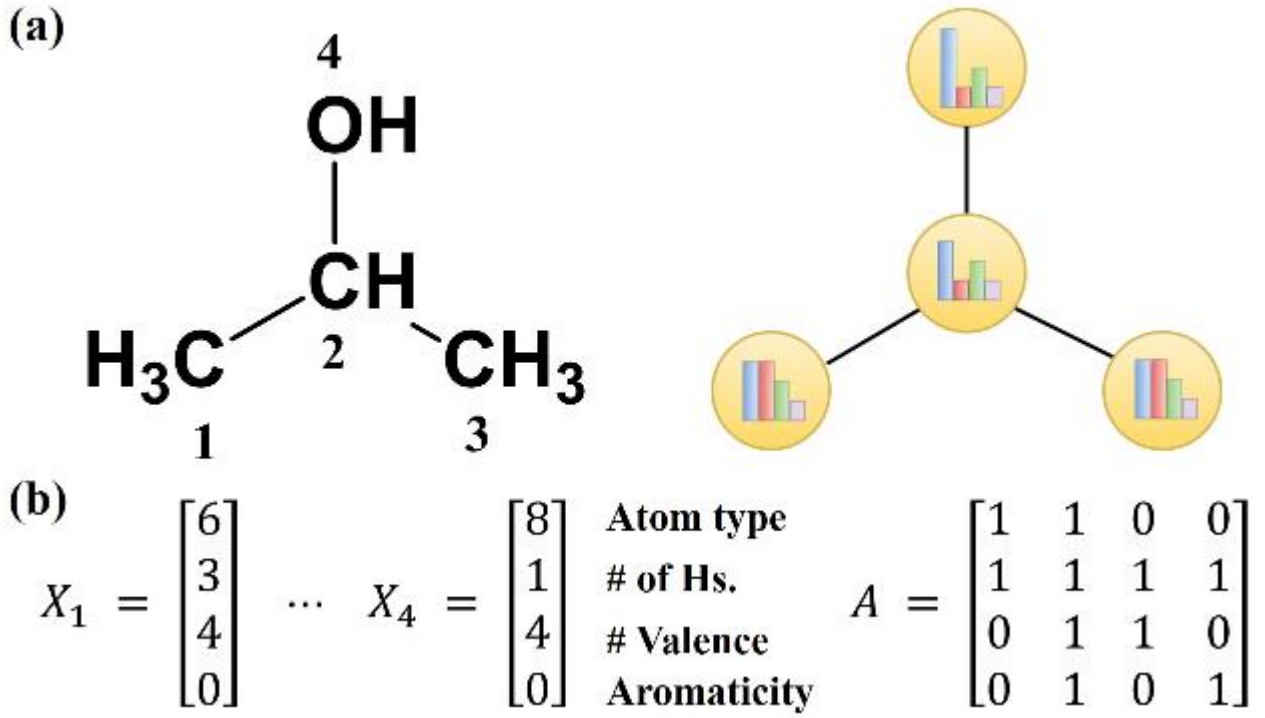
I have developed a framework of property prediction for my research

- https://github.com/SeongokRyu/ACGT (early-version, need refactoring)
- Sharing model architecture
- Sharing random seeds and data splitting
- Pre-/post-processing scripts

GCN for property prediction systems

Preliminary) Molecular Graph

- Undirected graph, $G(X, A)$ where $X = H^{(0)} \in \mathbb{R}^{n \times d^{(0)}}$, $A \in \mathbb{R}^{n \times n}$ and $A_{ij} \in [0,1]$

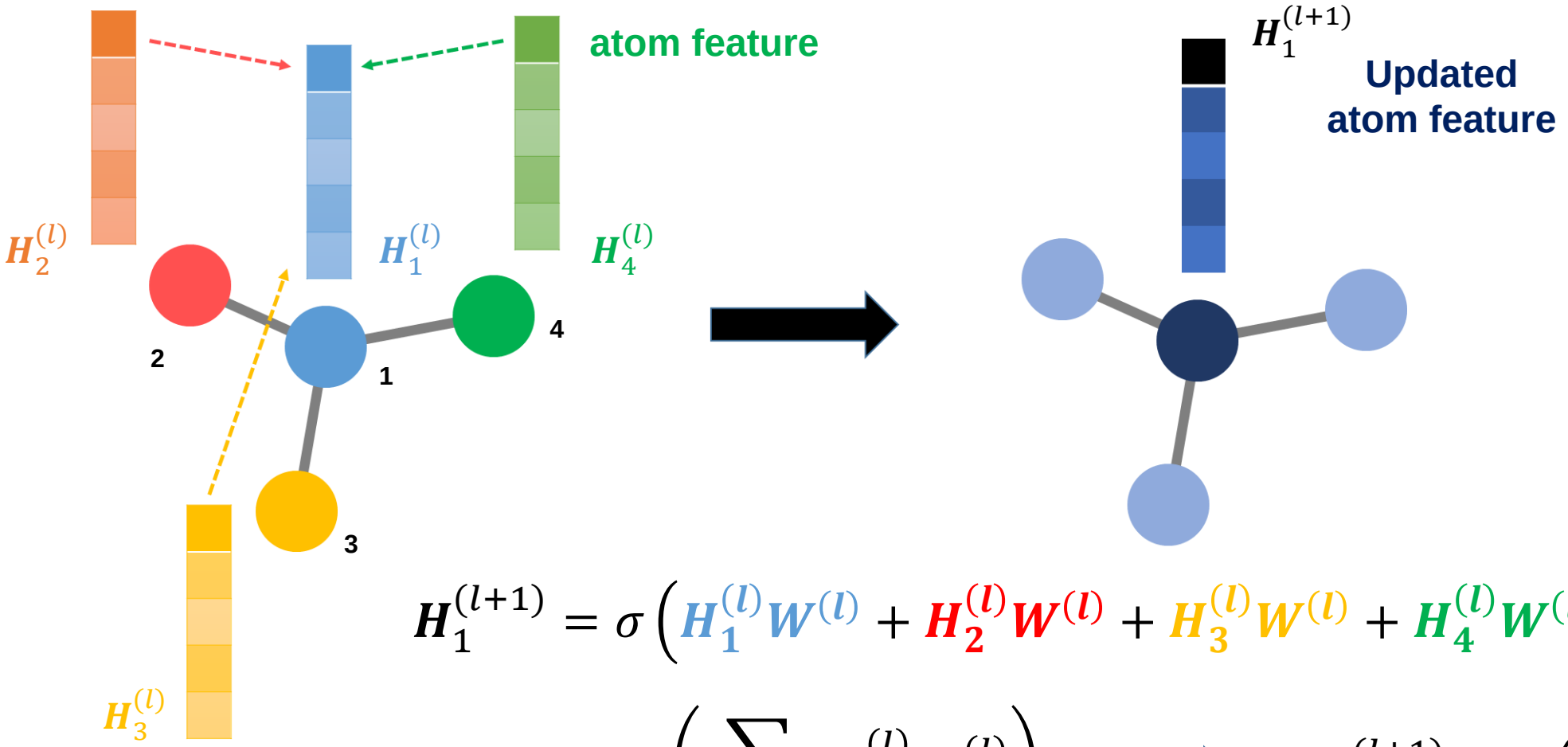


Input molecular graph of 2-propanol

GCN for property prediction systems

Preliminary) Graph convolutional networks

: updating node features at the l-th graph convolution layer



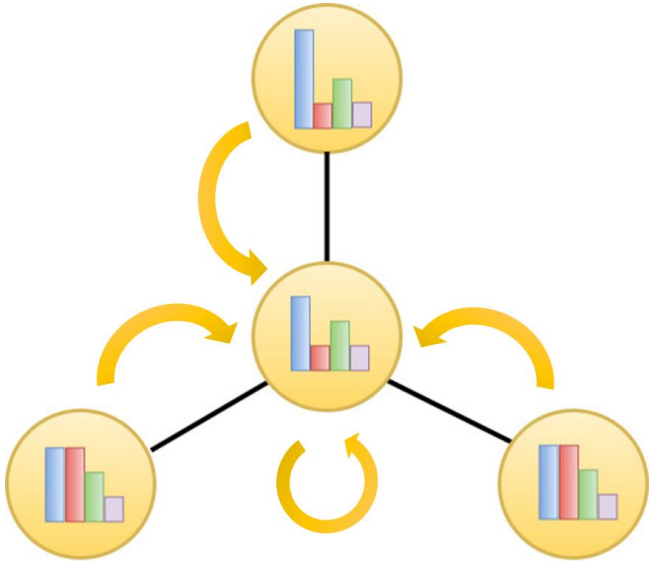
$$H_1^{(l+1)} = \sigma\left(H_1^{(l)}W^{(l)} + H_2^{(l)}W^{(l)} + H_3^{(l)}W^{(l)} + H_4^{(l)}W^{(l)}\right)$$

$$= \sigma\left(\sum_{j \in N(i)} H_j^{(l)}W^{(l)}\right) \quad \Rightarrow \quad H^{(l+1)} = \sigma\left(AH^{(l)}W^{(l)}\right)$$

GCN for property prediction systems

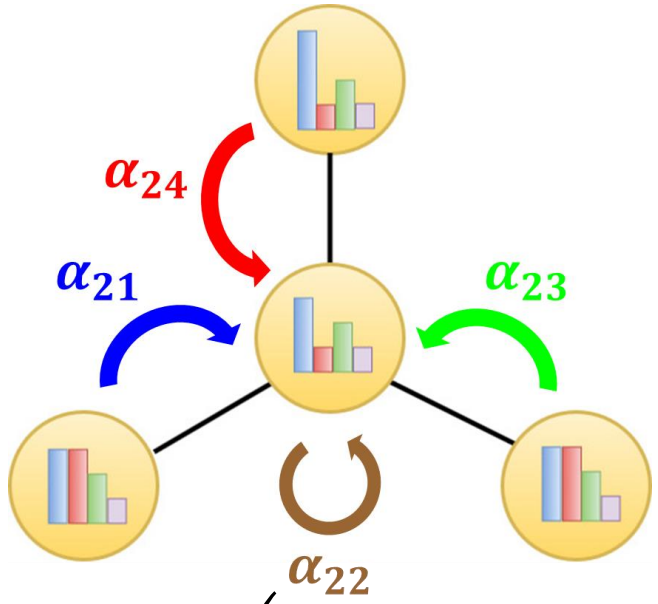
Applying the self-attention in graph convolutions

Vanilla GCN updates node features
with a same importance.



$$H_i^{(l+1)} = \sigma\left(\sum_{j \in N(i)} H_j^{(l)} W^{(l)}\right)$$

Attention mechanism enables GCN to
update nodes **with different importances.**



$$H_i^{(l+1)} = \sigma\left(\sum_{j \in N(i)} \alpha_{ij} H_j^{(l)} W^{(l)}\right)$$

In preparation

GCN for property prediction systems

Applying the self-attention in graph convolutions

$$\alpha_{ij} = f\left(H_i^{(l)}W^{(l)}, H_j^{(l)}W^{(l)}\right)$$

- Graph Attention Networks (Veličković, Petar, et al., 2017)

$$\alpha_{ij} = \tau\left(MLP\left(\left[H_i^{(l)}W^{(l)}, H_j^{(l)}W^{(l)}\right]\right)\right)$$

→ Concat & MLP

- Transformer (Vaswani et al., 2017), Set Transformer (Lee et al., 2019)

$$\alpha_{ij} = \tau\left(\frac{1}{\sqrt{d^{(l)}}}\left(H_i^{(l)}W^{(l)}\right)\left(H_j^{(l)}W^{(l)}\right)^T\right)$$

→ Scaled-dot product

In preparation

GCN for property prediction systems

Readout : aggregating node features to map input graphs to graph features

$$z^{(l)} = \text{Readout}\left(\left\{H_i^{(l)}\right\}\right)$$

- Average pooling

$$z^{(l)} = \frac{1}{N}\sum_{i=1}^{N} H_i^{(l)} W_g^{(l)}, \qquad W_g^{(l)} \in \mathbb{R}^{d^{(l)} \times d_g^{(l)}}$$

- Sum pooling

$$z^{(l)} = \sum_{i=1}^{N} H_i^{(l)} W_g^{(l)}, \qquad W_g^{(l)} \in \mathbb{R}^{d^{(l)} \times d_g^{(l)}}$$

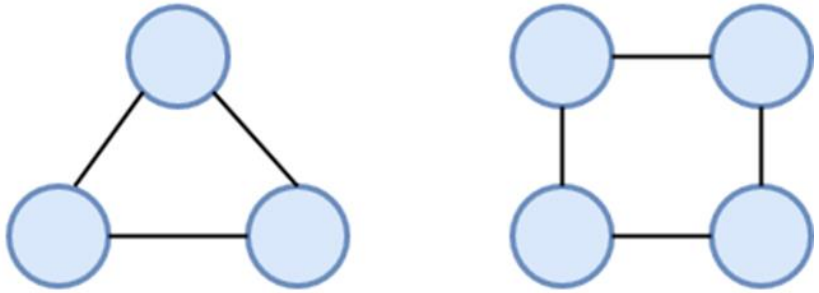
GCN for property prediction systems

Readout : aggregating node features to map input graphs to graph features

$$z^{(l)} = \text{Readout}\left(\left\{H_i^{(l)}\right\}\right)$$

- Average pooling cannot give correct summary statistics of node features

$$z_1 = \frac{1}{3}(h + h + h) = h \qquad z_2 = \frac{1}{4}(h + h + h + h) = h$$



$\mathcal{G}_1$ $\mathcal{G}_2$

- Sum pooling gives correct summary statistics of node features

$$z_1 = (h + h + h) = 3h \qquad z_2 = (h + h + h + h) = 4h$$

In preparation

GCN for property prediction systems

Readout : aggregating node features to map input graphs to graph features

$$z^{(l)} = \text{Readout}\left(\left\{H_i^{(l)}\right\}\right)$$

- Attention pooling can summarize node features with different (learnable) weights

$$z^{(l)} = \sum_{i=1}^{N} \alpha_i^{(l)} H_i^{(l)} W_g^{(l)}, \qquad \alpha_i^{(l)} = N \times \text{softmax}\left(\frac{1}{\sqrt{d_g^{(l)}}}\right)(\mathbf{1})\left(H_i^{(l)} W_g^{(l)}\right)^T, \qquad W_g^{(l)} \in \mathbb{R}^{d^{(l)} \times d_g^{(l)}}$$

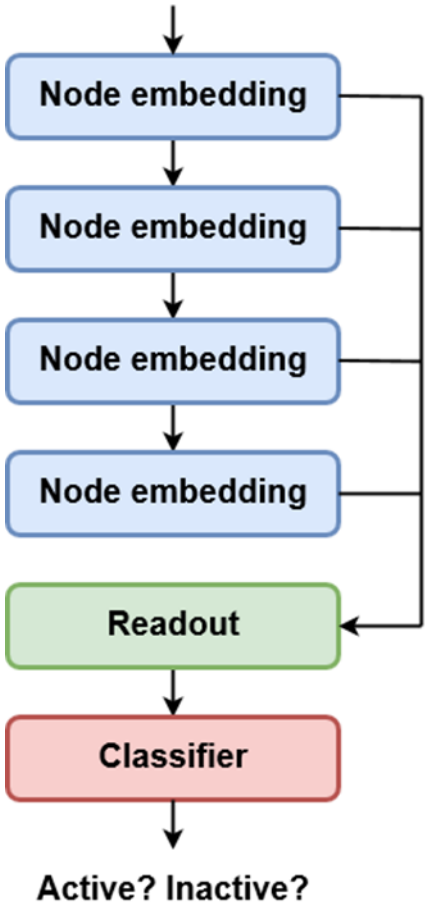
- Concatenation can summarize hierarchical sub-graph structures

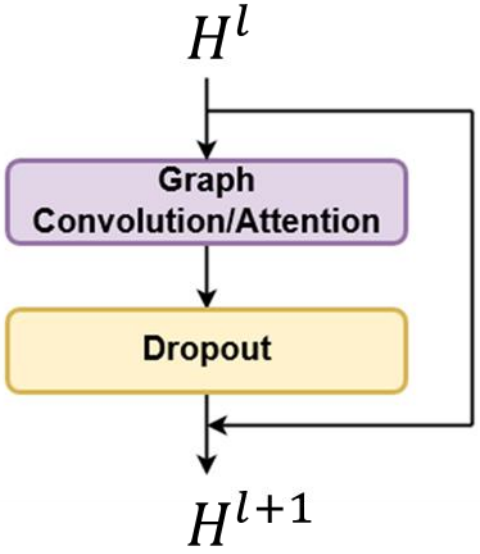
$$z_G = \text{Concat}\left(\left[z^{(1)}, \ldots, z^{(L)}\right]\right)$$

In preparation

GCN for property prediction systems

Readout : aggregating node features to map input graphs to graph features



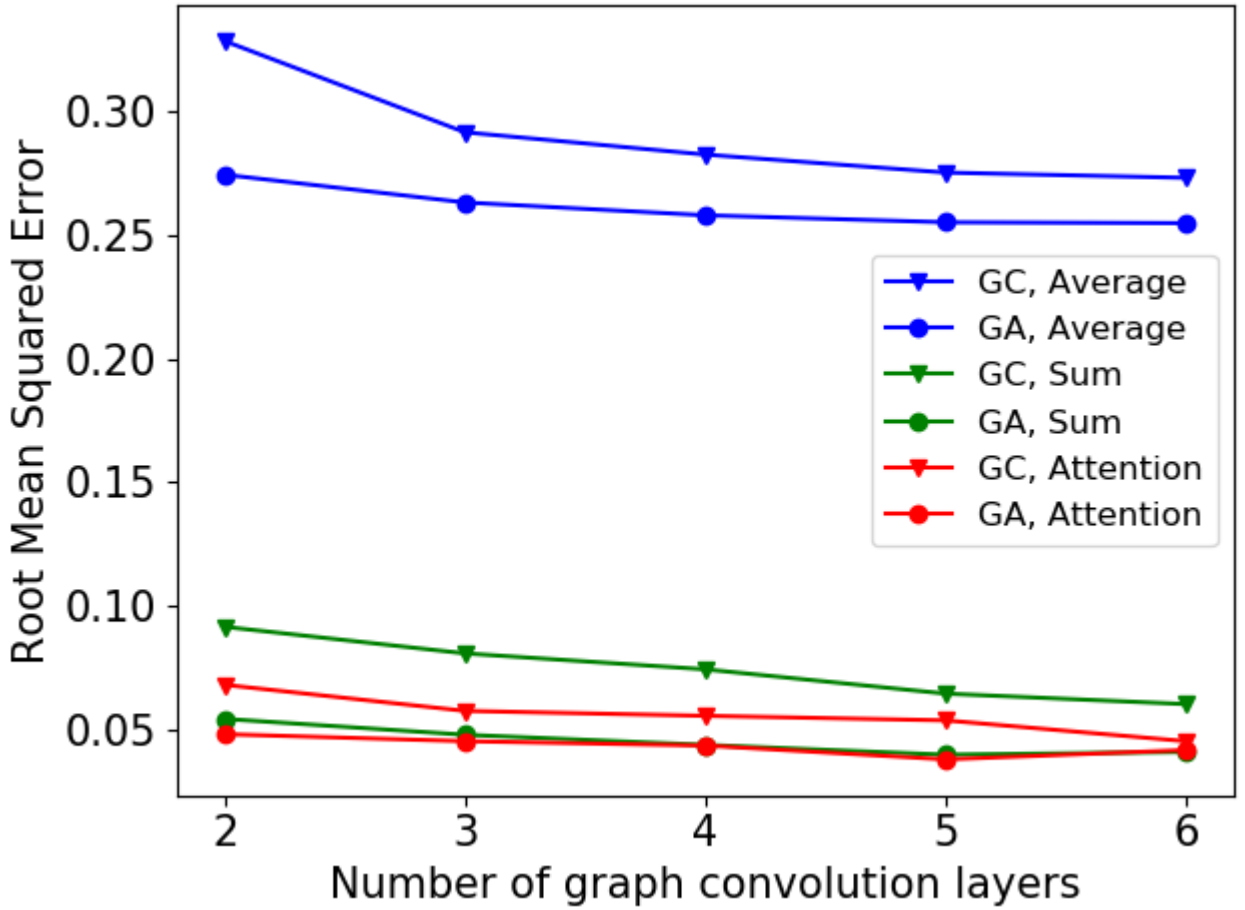


In preparation

GCN for property prediction systems

Result 1) Self-attention for node embedding and readout improves prediction performances

- Toy example: logP prediction
- Augmenting the self-attention for node-embedding and readout improves prediction performances.
- Average pooling significantly underperforms sum and attention pooling.
- Attention pooling outperforms the other readouts.



✓ # training : 80,000
✓ # test : 20,000

In preparation

GCN for property prediction systems

Result 2) Other prediction tasks

- ✓ # training : 80,000
- ✓ # test : 20,000

		LogP		TPSA		SAS	
		Concat	Last	Concat	Last	Concat	Last
GC	Mean	0.283	0.280	6.01	5.76	0.104	0.108
	Sum	0.074	0.083	0.52	0.62	0.068	0.066
	Attention	0.055	0.077	0.42	0.50	0.060	0.063
GA	Mean	0.258	0.261	5.88	0.56	0.091	0.098
	Sum	0.044	0.054	0.53	0.55	0.057	0.057
	Attention	**0.043**	0.055	**0.52**	0.60	**0.053**	0.054

- Augmenting the self-attention for node-embedding and readout improves prediction performances.
- Concatenation of all graph features at $l = 1, \ldots, 4$ layers outperforms using only the last graph feature.

In preparation

GCN for property prediction systems

Result 3) Binary classification tasks

# of samples (5-fold random splitting with 80:20 ratio)

- ✓ BACE: 1,513
- ✓ BBBP: 2,050
- ✓ HIV: 41,127

	Architecture	Accuracy (↑)	AUROC (↑)	F1-score (↑)	ECE (↓)	OCE (↓)
BACE	GCN+Sum	0.809	0.878	0.780	0.102	0.086
	GCN+Attn	**0.822**	**0.897**	**0.802**	**0.091**	**0.065**
	GAT+Sum	0.793	0.879	0.764	0.166	0.139
	GAT+Attn	0.799	0.880	0.781	0.174	0.202
BBBP	GCN+Sum	0.890	0.915	0.929	**0.066**	**0.082**
	GCN+Attn	**0.892**	**0.919**	**0.931**	0.087	0.235
	GAT+Sum	0.864	0.902	0.913	0.120	0.211
	GAT+Attn	0.871	0.898	0.917	0.115	0.169
HIV	GCN+Sum	0.970	0.805	0.392	**0.008**	0.008
	GCN+Attn	**0.971**	**0.816**	**0.438**	0.010	**0.007**
	GAT+Sum	0.965	0.797	0.425	0.030	0.021
	GAT+Attn	0.970	0.812	0.410	0.010	0.008

- Using attention in node update involves over-fitting
- More parameterization discourage reliability (see more details in later)

In preparation

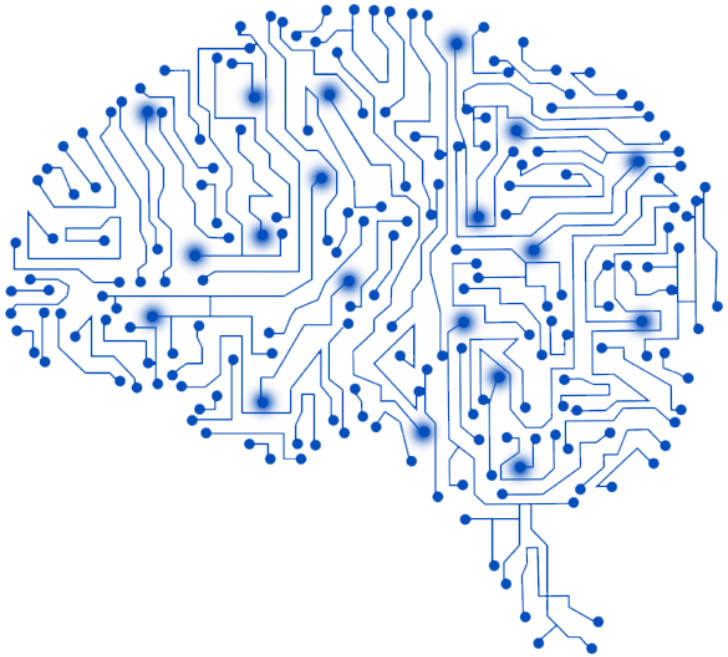
KAIST CHEMISTRY

Bayesian deep learning for drug discovery

KAIST

Bayesian deep learning for drug discovery

Importance of knowing what we do not know



"Is it toxic?"

"Is it EGFR-active?"

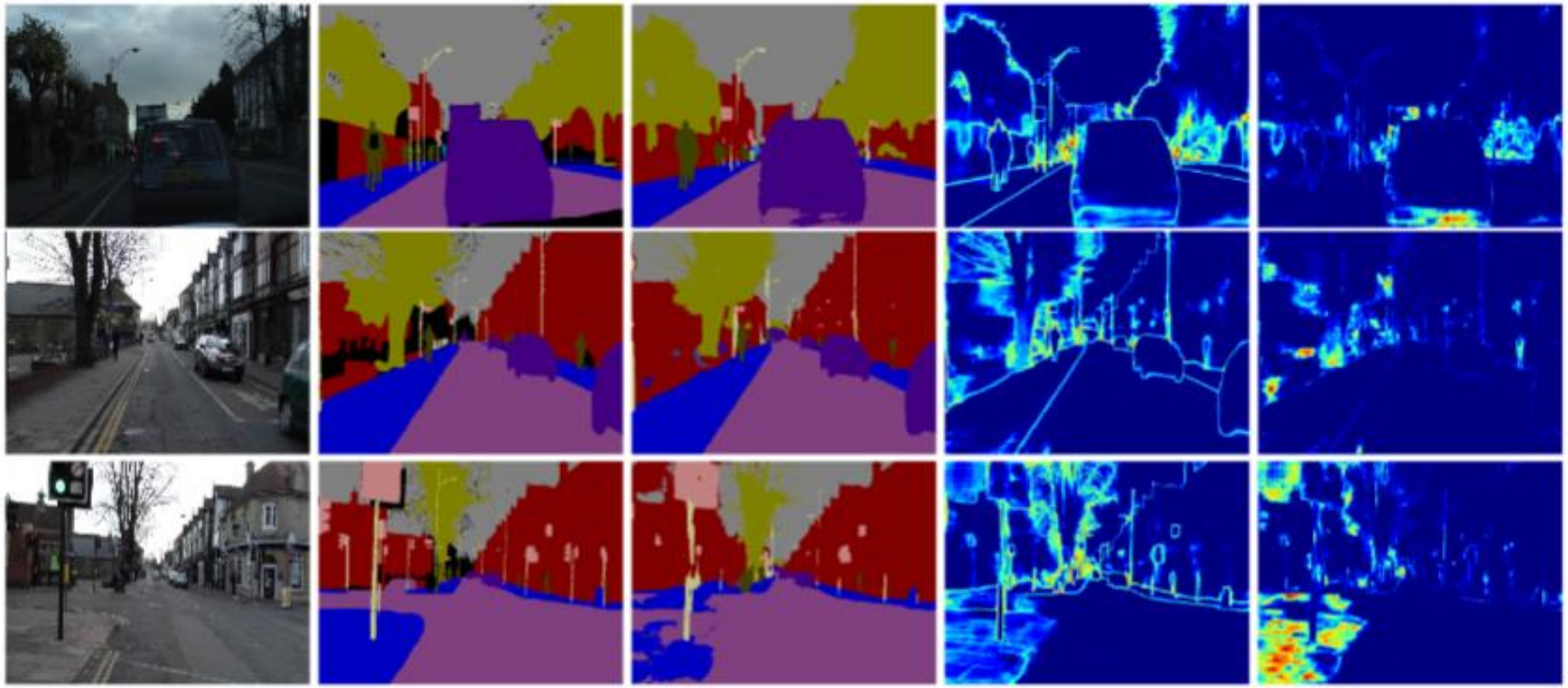
"HOW MUCH CAN WE TRUST?"

→ Measuring predictive uncertainty

→ Measuring reliability of results

Bayesian deep learning for drug discovery

Importance of knowing what we do not know



(a) Input Image (b) Ground Truth (c) Semantic Segmentation (d) Aleatoric Uncertainty (e) Epistemic Uncertainty

Kendall, Alex, and Yarin Gal."What uncertainties do we need in bayesian deep learning for computer vision?." *Advances in neural information processing systems*. 2017.

Bayesian deep learning for drug discovery

Motivation – obstacles in molecular applications of deep learning

1. Acquiring data is very time-consuming, laborious and cost-expensive.
2. Insufficient amount of data
3. Noisy experimental values

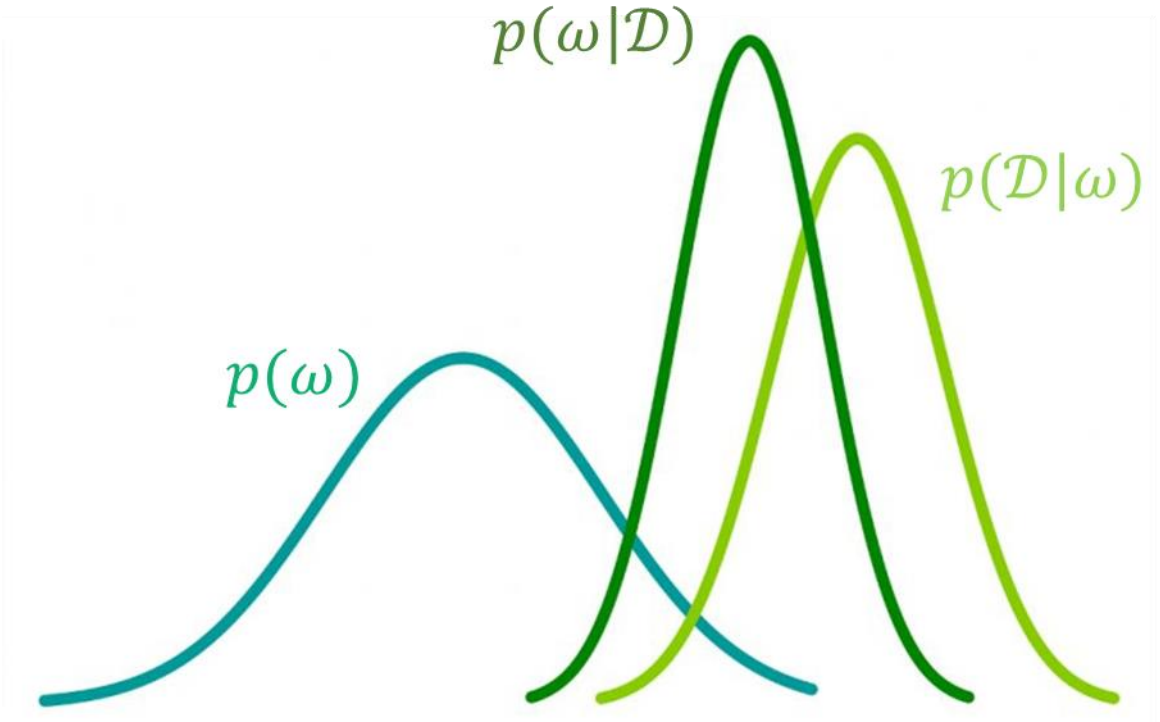
- In such cases, a model trained with insufficient quantity and quality of data can give unreliable predictions.
- Thus, it is necessary to estimate uncertainty in prediction results to utilize machine learning systems for real life applications.
- We can communicate with experimental researchers to determine what compounds should be labeled.

Bayesian deep learning for drug discovery

Preliminary) Bayes' rule

$$p(\omega|\mathcal{D}) = \frac{p(\mathcal{D}|\omega)p(\omega)}{p(\mathcal{D})}$$

- $p(\mathcal{D}|\omega)$: **likelihood function**, the probability of observing data $\mathcal{D}$ given hypothesis ω.
- $p(\omega)$: **prior distribution**, our prior belief on hypothesis
- $p(\omega|D)$: **posterior distribution**, probability of hypothesis parameter ω given observation $\mathcal{D}$.



Bayesian deep learning for drug discovery

Preliminary) Bayesian inference

$$p(y^*|x^*, \mathcal{D}) = \int p(y^*|x^*, \omega)p(\omega|\mathcal{D})d\omega$$

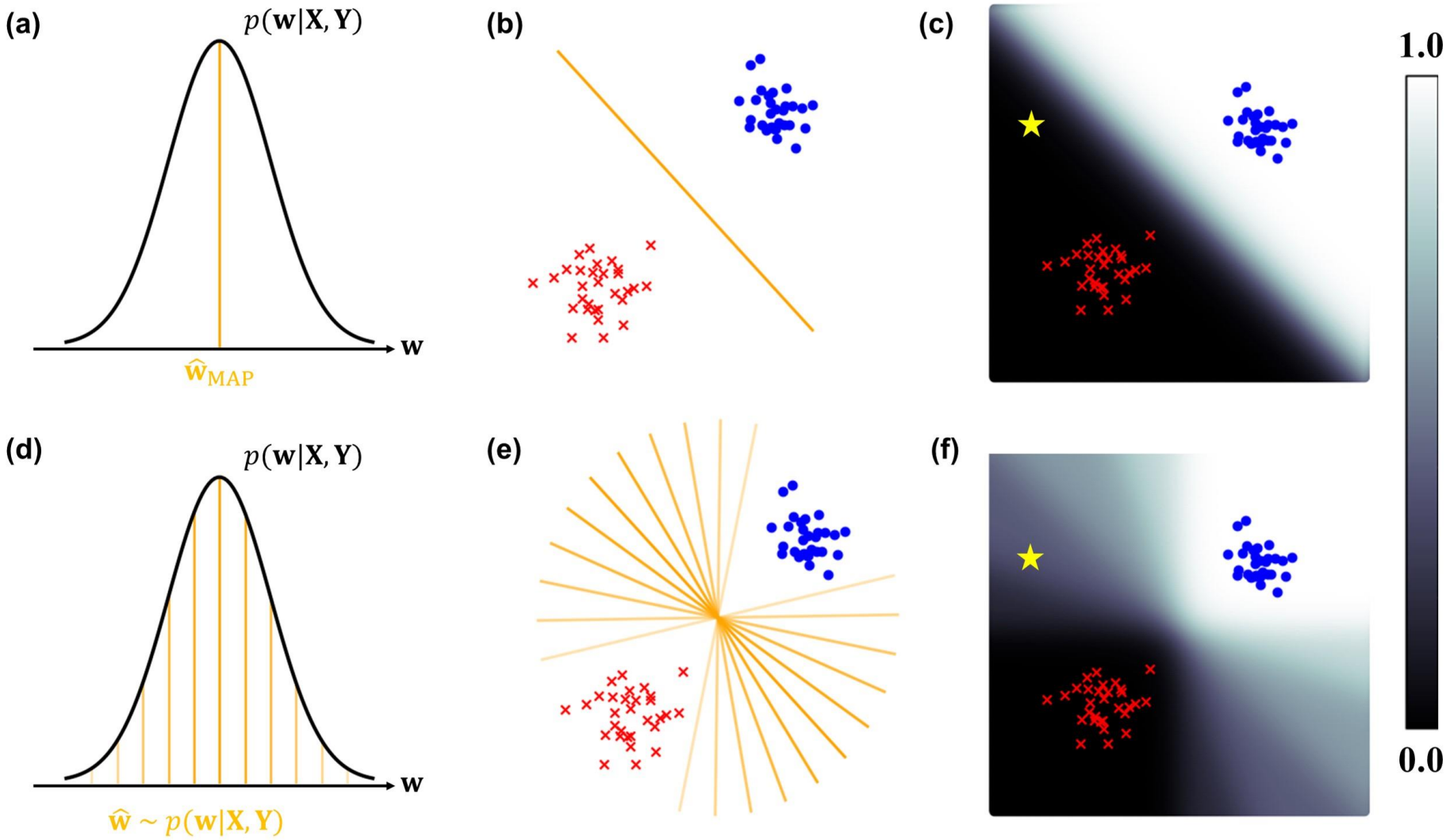
Predictive distribution

- $p(y^*|x^*, \omega)$ is the probability of observing y^* given input x^* and hypothesis ω
- $p(\omega|D)$ is our distribution of hypothesis is obtained by training data $\mathcal{D}$.
- Therefore, by integrating all possible hypothesis, our predictive distribution of output y^* given x^* and training data $\mathcal{D}$ is given by the above equation.
- We can estimate expectation value and its uncertainty as the mean and variance of the predictive distribution.

$$\bar{y}^* = \text{Mean}_{p(y^*|x^*,\mathcal{D})}[y^*] \qquad \text{unc}(y^*)^2 = \text{Var}_{p(y^*|x^*,\mathcal{D})}[y^*]$$

Bayesian deep learning for drug discovery

Preliminary) maximum-a-posteriori (MAP) inference vs Bayesian inference

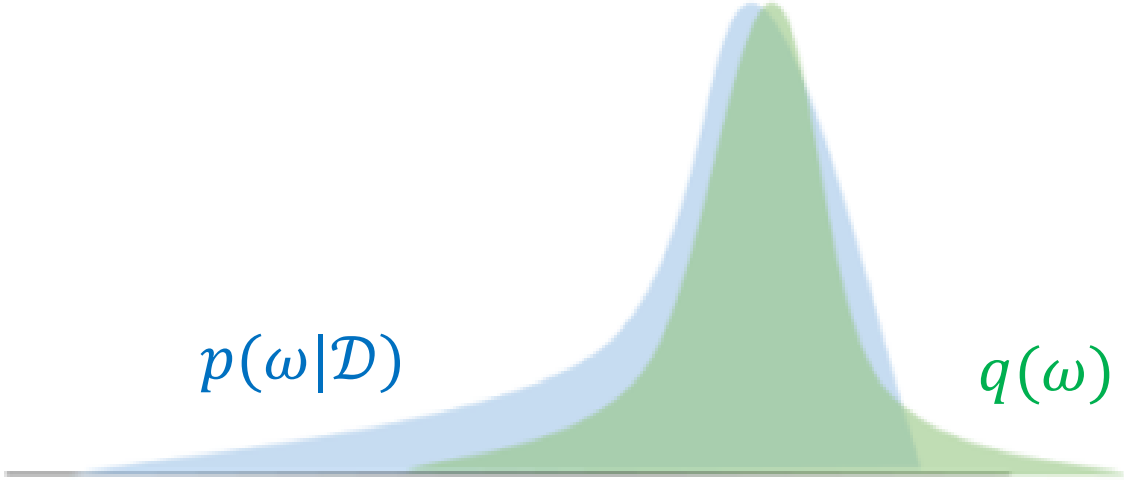


Ryu, Seongok, Yongchan Kwon, and Woo Youn Kim. "A Bayesian graph convolutional network for reliable prediction of molecular properties with uncertainty quantification." *Chemical Science* 10.36 (2019): 8438-8446.

Bayesian deep learning for drug discovery

Preliminary) Variational Inference

$$\min_{q \in \mathcal{Q}} \mathrm{KL}\big(q(\omega) | p(\omega|\mathcal{D})\big)$$



- Minimize the KL-divergence between variational distribution $q(\omega)$ in variational family $\mathcal{Q}$ and true posterior $p(\omega|\mathcal{D})$

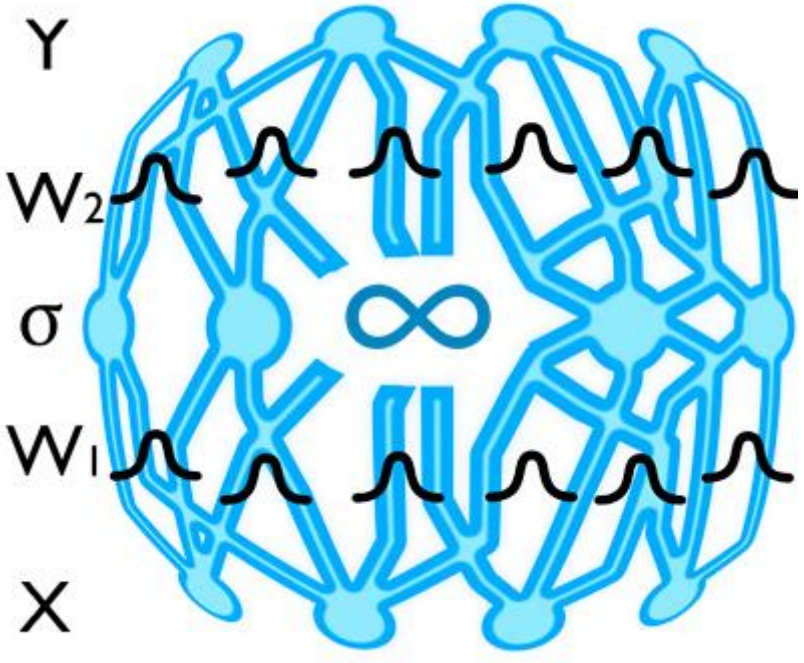
- Choice on variational family $\mathcal{Q}$:
  - ✓ Dropout variational inference (Gal. et. al., 2016)
  - ✓ Gaussian mean field (Hernandez-Lobato and Adams, 2015)
  - ✓ Multiplicative normalizing flow (Luizos and Welling, 2017)
  - ✓ Gaussian mean field with natural gradient (Mandt et al., 2017)
  - ✓ …

Ryu, Seongok, Yongchan Kwon, and Woo Youn Kim. "A Bayesian graph convolutional network for reliable prediction of molecular properties with uncertainty quantification." *Chemical Science* 10.36 (2019): 8438-8446.

Bayesian deep learning for drug discovery

Approximate posterior with dropout variational distribution (MC-Dropout)

- First, minimize $\mathrm{KL}(q_\theta(\omega)||p(\omega|D))$, where $q_\theta(\omega)$ is a dropout variational distribution. (Training time)
- Then, use the approximated posterior to compute the predictive distribution by MC-sampling (Inference time)



http://www.cs.ox.ac.uk/people/yarin.gal/website/blog_3d801aa532c1ce.html

```
outputs = []
for i in range(num_mc_sampling):
    outputs.append(model.predict(inp, use_dropout=True))
predictive_mean = np.mean(outputs, axis=0)
predictive_variance = np.var(outputs, axis=0)
```

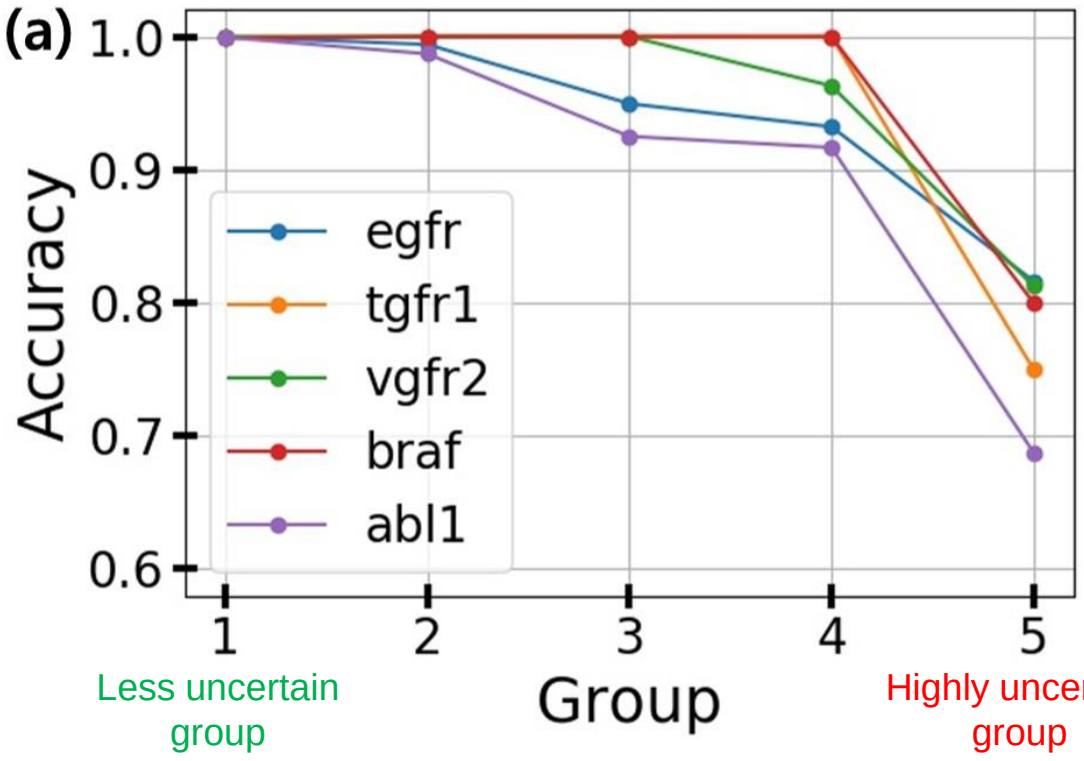

- Gal, Yarin, and Zoubin Ghahramani. "Dropout as a bayesian approximation: Representing model uncertainty in deep learning." *international conference on machine learning*. 2016.
- Gal, Yarin. *Uncertainty in deep learning*. Diss. PhD thesis, University of Cambridge, 2016.
- Kendall, Alex, and Yarin Gal. "What uncertainties do we need in bayesian deep learning for computer vision?." *Advances in neural information processing systems*. 2017.
- Gal, Yarin, Riashat Islam, and Zoubin Ghahramani. "Deep bayesian active learning with image data." *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017.

Bayesian deep learning for drug discovery

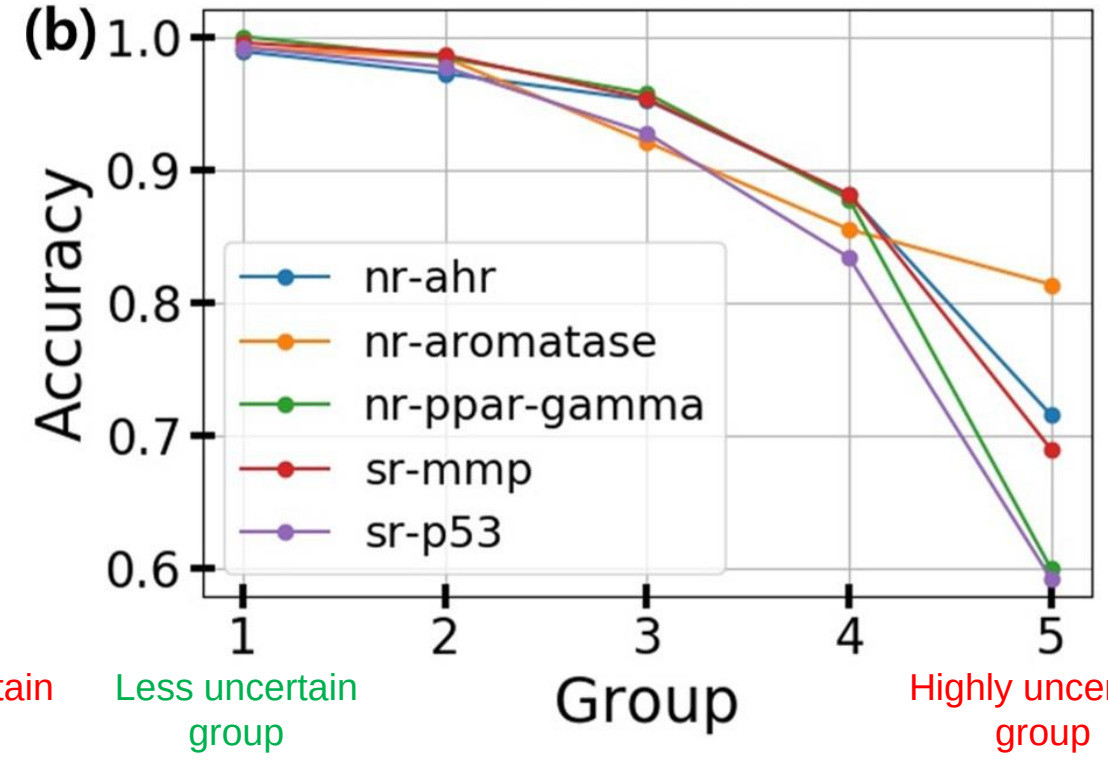
Results 1: Uncertainty in classification tasks

- Molecules in the *i*-th group have uncertainty in range of $[(i-1)\times 0.1, (i+1)\times 0.1)$.
- Below results show that we can screen more accurate predictions based on predictive uncertainty values.

DUD-E dataset: bio-activity classification



Tox21 dataset: toxicity classification



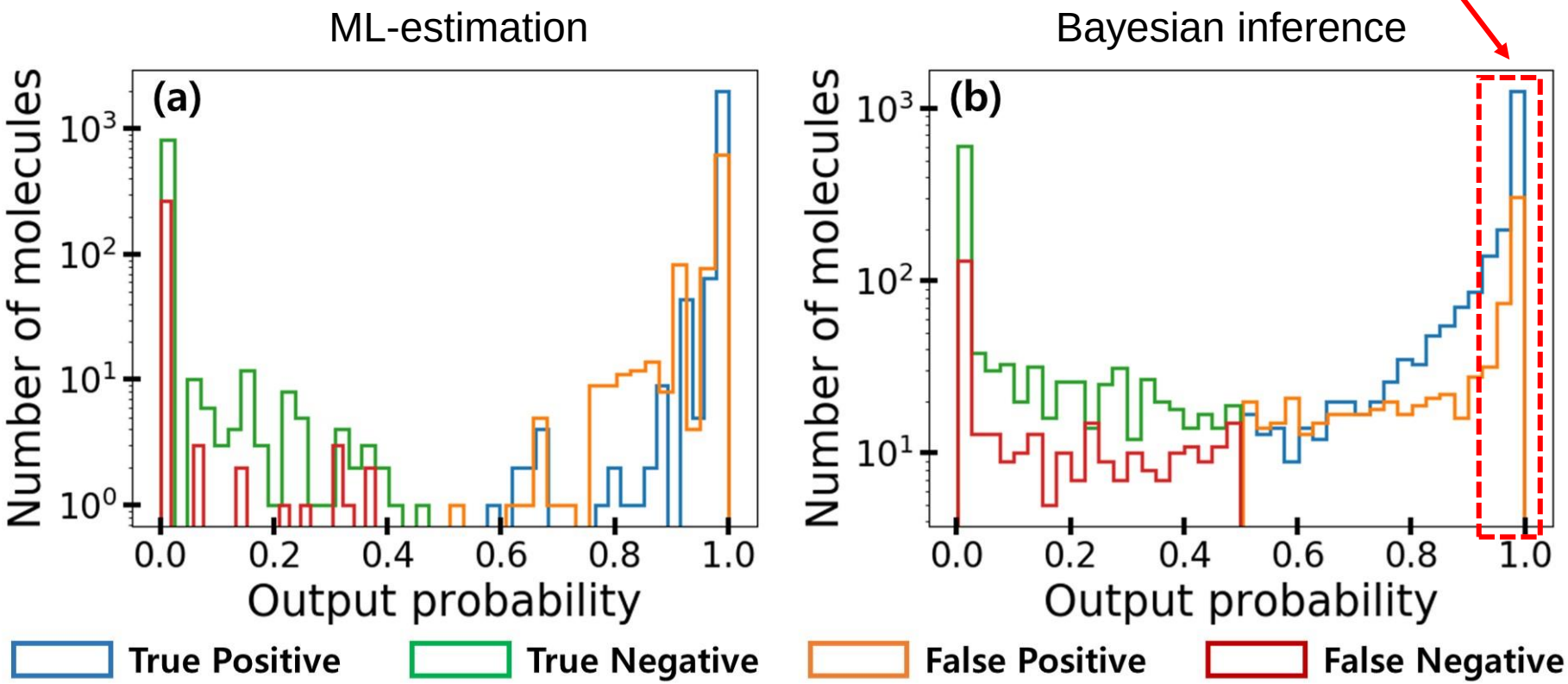
Ryu, Seongok, Yongchan Kwon, and Woo Youn Kim. "A Bayesian graph convolutional network for reliable prediction of molecular properties with uncertainty quantification." *Chemical Science* 10.36 (2019): 8438-8446.

Bayesian deep learning for drug discovery

Results 2: Virtual screening of EGFR inhibitors

- Training : EGFR inhibitory activity data in the **DUD-E dataset**.
- Screening : compounds in the **ChEMBL dataset**

Interested in predictions
with high probability (confidence)



Ryu, Seongok, Yongchan Kwon, and Woo Youn Kim.
"A Bayesian graph convolutional network for reliable prediction of molecular properties with uncertainty quantification." *Chemical Science* 10.36 (2019): 8438-8446.

Bayesian deep learning for drug discovery

Results 2: Virtual screening of EGFR inhibitors

- The Bayesian model outperforms the non-Bayesian models.

	ML-estimation	MAP-inference	Bayesian inference
Accuracy	0.728	0.739	**0.752**
AUROC	0.756	0.781	**0.785**
Precision	0.714	0.68	**0.746**
Recall	0.886	**0.939**	0.868
F1-score	0.791	0.789	**0.803**

- The Bayesian model is more effective for virtual screening of EGFR-inhibitors than the non-Bayesian model.

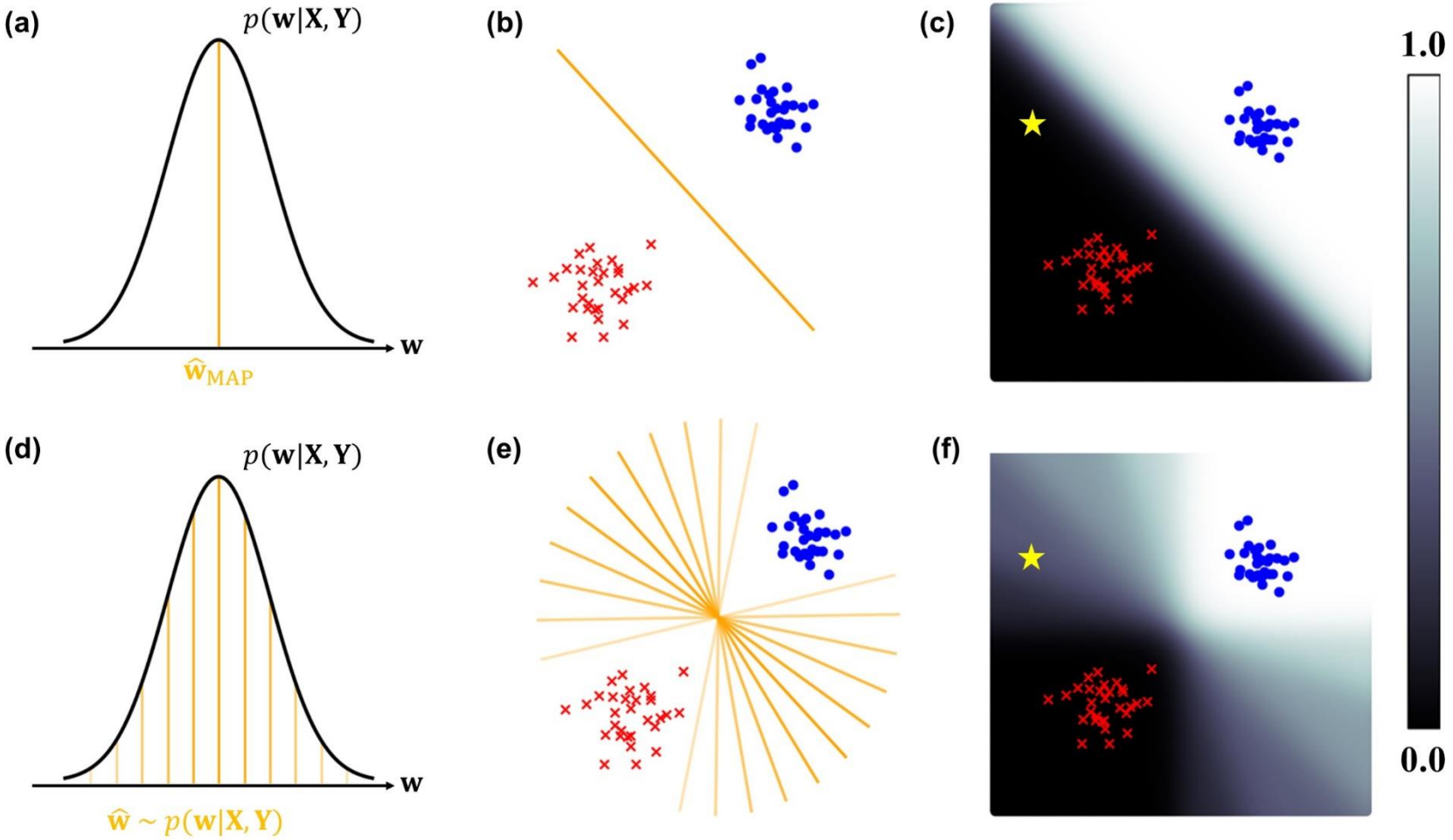
Top N	ML-estimation	MAP-inference	Bayesian inference
100	29	57	**69**
200	67	130	**140**
300	139	202	**214**
500	277	346	**368**

Ryu, Seongok, Yongchan Kwon, and Woo Youn Kim.
"A Bayesian graph convolutional network for reliable prediction of molecular properties with uncertainty quantification." *Chemical Science* 10.36 (2019): 8438-8446.

Bayesian deep learning for drug discovery

Results 3: Active learning on HIV dataset

- Please remind this illustration

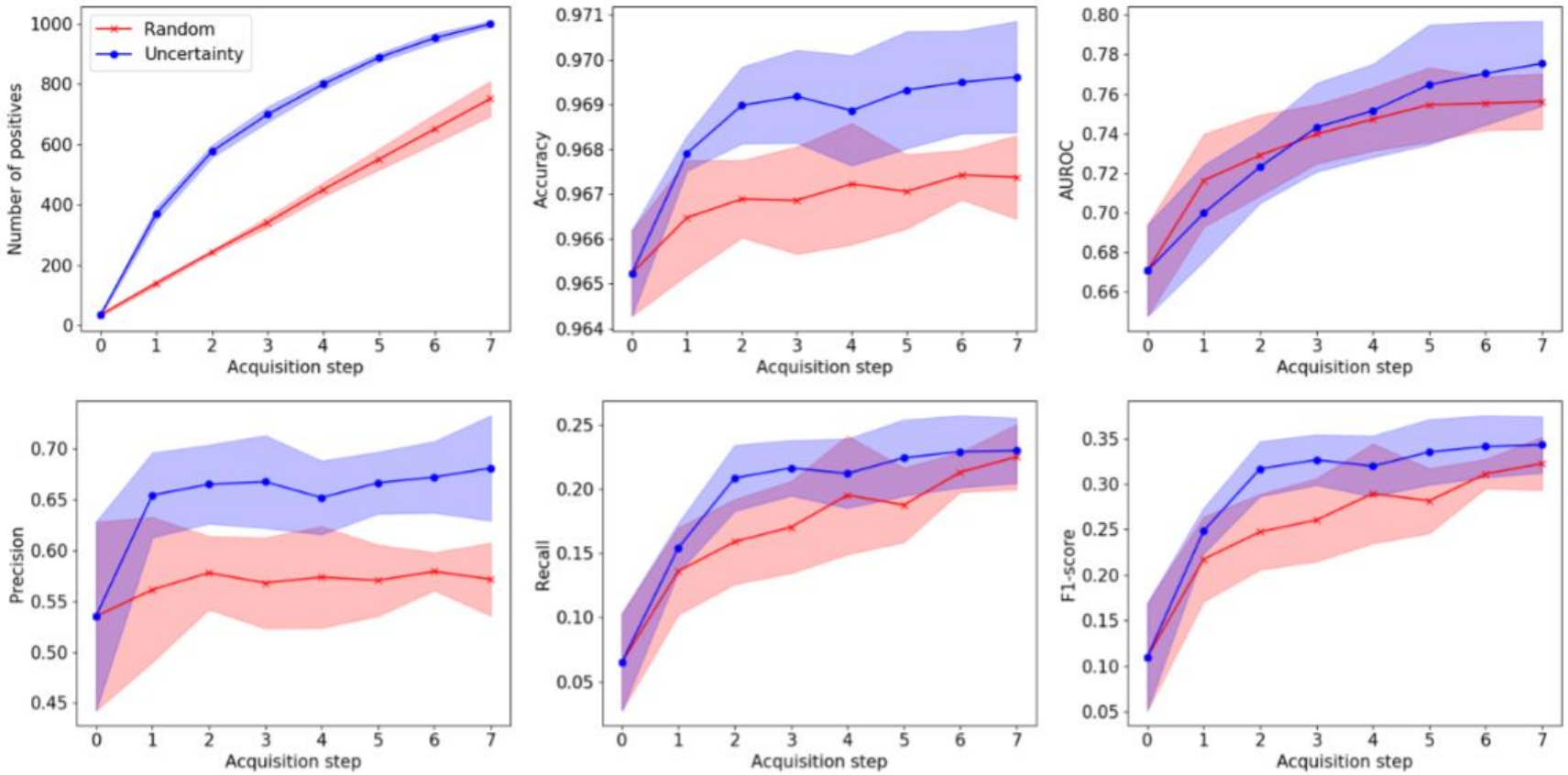


Ryu, Seongok, Yongchan Kwon, and Woo Youn Kim. "A Bayesian graph convolutional network for reliable prediction of molecular properties with uncertainty quantification." *Chemical Science* 10.36 (2019): 8438-8446.

Bayesian deep learning for drug discovery

Results 3: Active learning on HIV dataset

- Uncertainty-based acquisition improves the classification model more than random acquisition.
- For HIV-dataset, the ratio of positive (negative) samples is 3% (97%).
- Sampling positive molecules from the pool is more expected.



Unpublished results

KAIST CHEMISTRY

Reliability of prediction systems

KAIST

Reliability of prediction systems

Modern neural networks are usually over-confident

	LeNet-5	ResNet-110
# parameters	6×10^4	$\mathbf{1.7 \times 10^6}$
Classification error (%)	44.9%	30.6%
ECE (%)	4.85%	16.53%

- Better architecture with more parameters

  → better classification accuracy

  → worse calibration error (reliability)

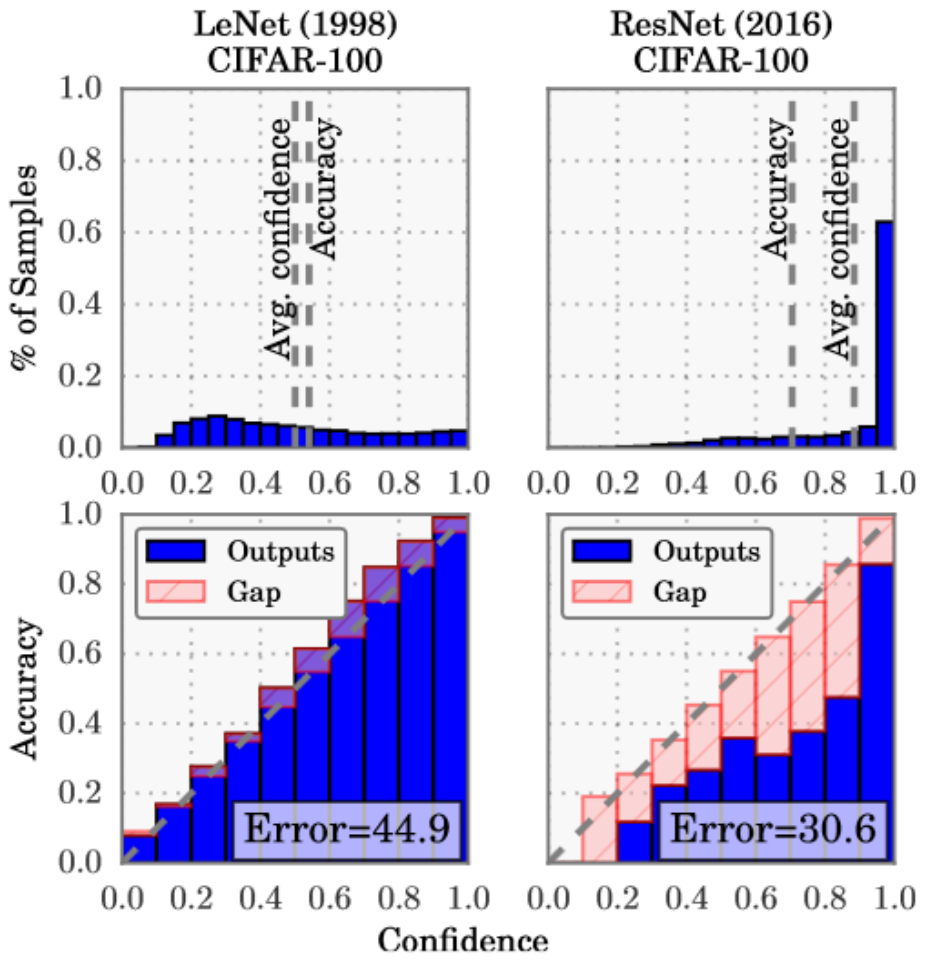


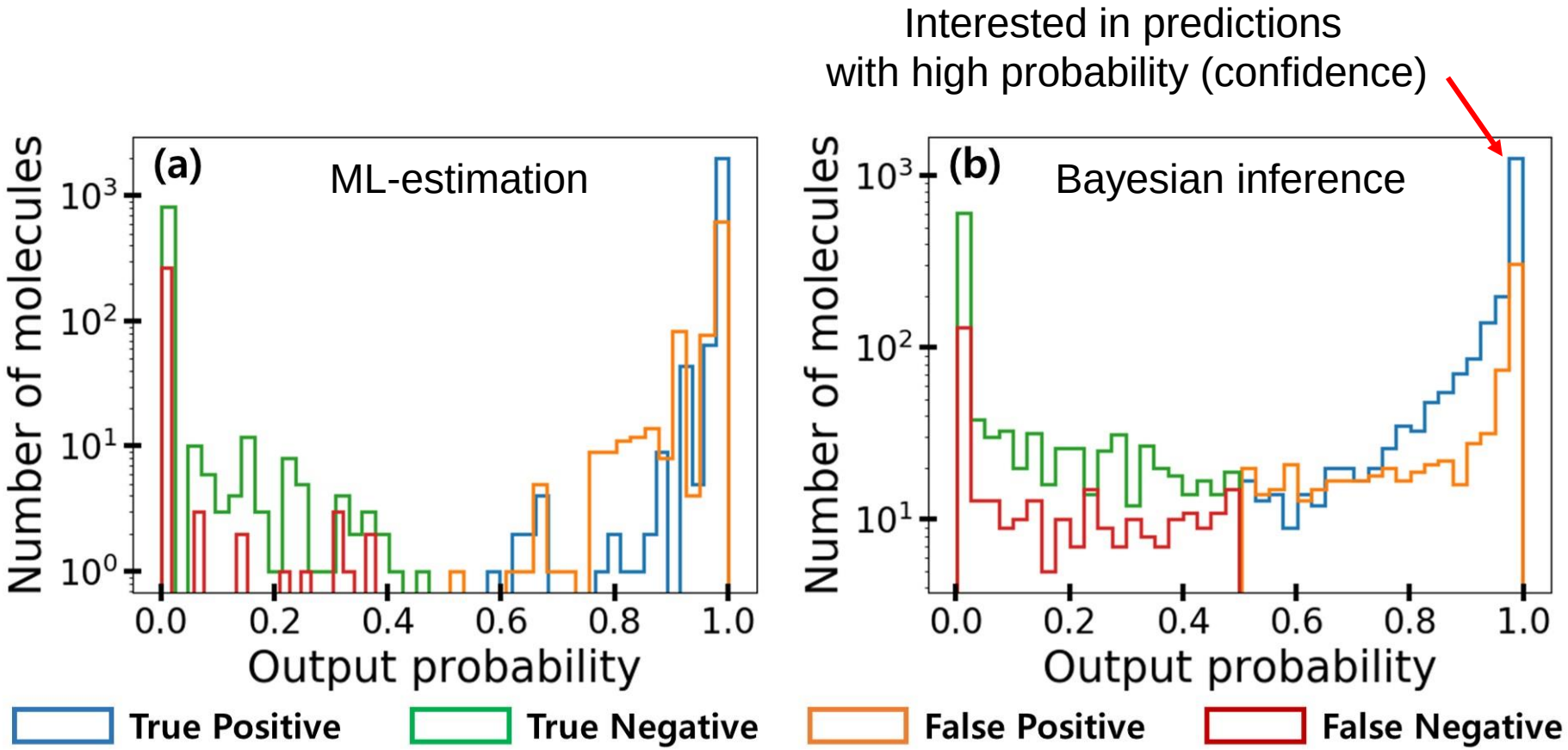
Figure 1. Confidence histograms (top) and reliability diagrams (bottom) for a 5-layer LeNet (left) and a 110-layer ResNet (right) on CIFAR-100. Refer to the text below for detailed illustration.

Guo, Chuan, et al. "On calibration of modern neural networks." *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017.

Reliability of prediction systems

Virtual screening scenario

- We are interesting in top-100 compounds, which is expected to be a true positive, in a entire predictions.
- For that purpose, higher output probability (confidence) results must show higher accuracy.



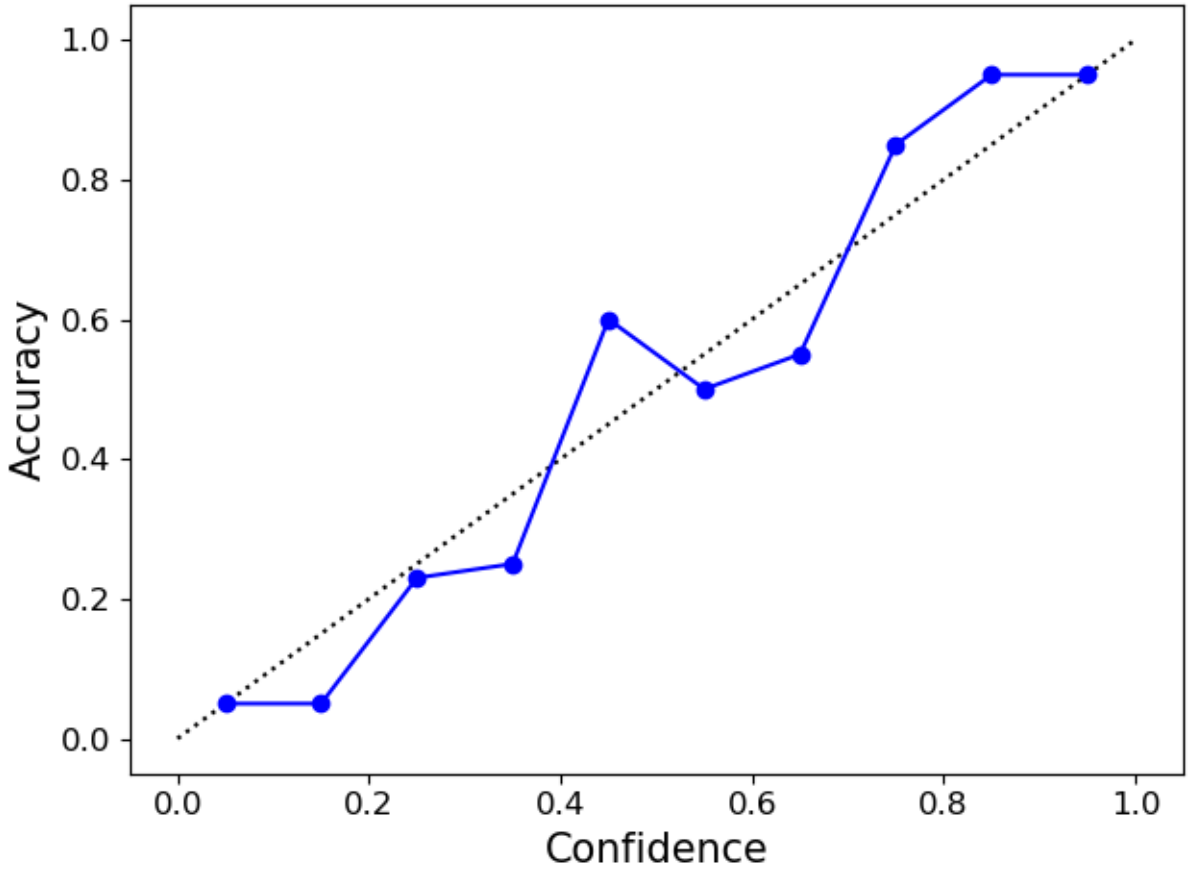
In preparation

Reliability of prediction systems

Calibration curve

- B_m : the set of indices of samples whose prediction confidence $\hat{p}_i$ falls into the interval $I_m = \left(\frac{m-1}{M}, \frac{m}{M}\right]$.
- $\text{conf}(B_m) = \frac{1}{|B_m|}\sum_{i \in B_m} \hat{p}_i$
- $\text{acc}(B_m) = \frac{1}{|B_m|}\sum_{i \in B_m} \mathbf{1}(\hat{y}_i = y_i)$
- Black dotted line: prefect calibration

  ex) The output result 0.6 will be correct with 60% probability.



Guo, Chuan, et al. "On calibration of modern neural networks."
Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017.

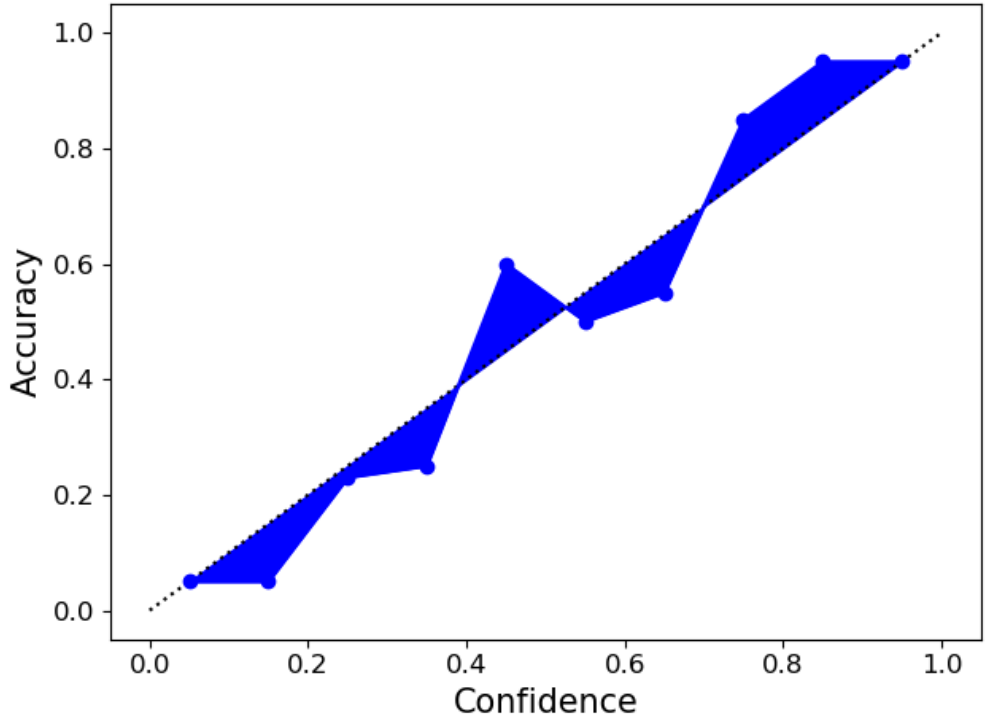
Reliability of prediction systems

Expected Calibration Error (OCE)

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} |\mathrm{acc}(B_m) - \mathrm{conf}(B_m)|$$

Over-confidence Error (OCE)

$$\mathrm{OCE} = \sum_{m=1}^{M} \frac{|B_m|}{N} [\mathrm{conf}(B_m) \times \max(\mathrm{conf}(B_m) - \mathrm{acc}(B_m), 0)]$$



Guo, Chuan, et al. "On calibration of modern neural networks." *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017.

Reliability of prediction systems

Binary classification tasks

# of samples (5-fold random splitting with 80:20 ratio)

- ✓ BACE: 1,513
- ✓ BBBP: 2,050
- ✓ HIV: 41,127

	Architecture	Accuracy	AUROC	F1-score	ECE (%)	OCE (%)
BACE	GCN+Sum	0.809	0.878	0.780	10.19	8.59
	GCN+Attn	**0.822**	**0.897**	**0.802**	**9.09**	**6.53**
	GAT+Sum	0.793	0.879	0.764	16.61	13.87
	GAT+Attn	0.799	0.880	0.781	17.41	20.25
BBBP	GCN+Sum	0.890	0.915	0.929	**6.59**	**8.21**
	GCN+Attn	**0.892**	**0.919**	**0.931**	8.68	23.47
	GAT+Sum	0.864	0.902	0.913	12.03	21.09
	GAT+Attn	0.871	0.898	0.917	11.46	16.94
HIV	GCN+Sum	0.970	0.805	0.392	**0.83**	0.84
	GCN+Attn	**0.971**	**0.816**	**0.438**	0.97	**0.72**
	GAT+Sum	0.965	0.797	0.425	3.04	2.09
	GAT+Attn	0.970	0.812	0.410	0.99	0.82

- Need regularizations to improve reliability.
- Use ‘GCN+Attn’ for a baseline.

In preparation

Reliability of prediction systems

Regularization methods

- Standard dropout (DO)
- MC-dropout (MC-DO)
- Label Smoothing (LS)

$$y_c^{LS} = y_c \times (1 - \alpha) + \frac{\alpha}{C}$$

For example, with label smoothing amount $\alpha = 0.1$, [0.0, 1.0] becomes [0.05, 0.95]

- Entropy Regularization (ERL)

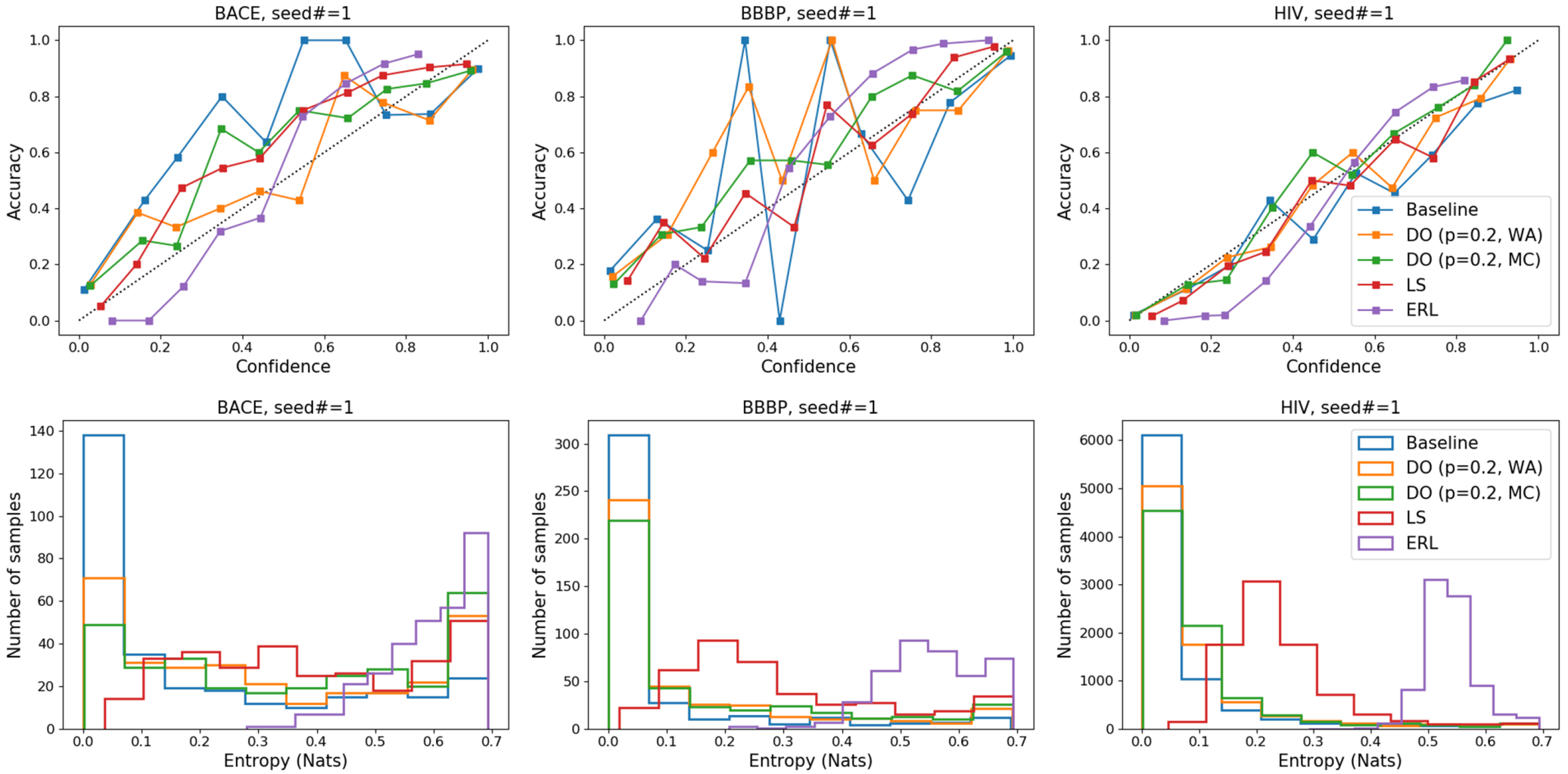
$$\mathcal{L}_{total} = \mathcal{L}_{BCE} - \beta \mathrm{H}[p_\theta(\mathbf{y}|\mathbf{x})]$$

where $\mathrm{H}[p_\theta(\mathbf{y}|\mathbf{x})] = -\sum_c^C p_\theta(\mathbf{y}|\mathbf{x}) \log p_\theta(\mathbf{y}|\mathbf{x})$.

In preparation

Reliability of prediction systems

Experimental results



In preparation

Reliability of prediction systems

Experimental results

	BACE		BBBP		HIV	
	ECE (%)	OCE (%)	ECE (%)	OCE (%)	ECE (%)	OCE (%)
Baseline	9.09	6.53	8.68	23.47	0.97	0.72
DO ($p = 0.2$)	8.73	11.27	7.36	13.46	0.61	0.74
MC-DO ($p = 0.2$)	7.33	6.91	5.77	7.59	**0.42**	**0.50**
LS ($\alpha = 0.1$)	**5.81**	**4.57**	**4.26**	**5.61**	4.86	1.03
ERL ($\beta = 1.0$)	12.88	9.29	16.47	8.54	20.62	6.54

- MC-dropout and label smoothing shows good calibration results.
- In contrast to previous works in computer vision, entropy regularization does not work well.

  → may be the result of cost-penalty

In preparation

Reliability of prediction systems

Learning with focal loss

Lin, Tsung-Yi, et al. "Focal loss for dense object detection." *Proceedings of the IEEE international conference on computer vision*. 2017.

- Focal loss is widely used for learning with imbalanced data.

$$\mathcal{L}_{\mathrm{BCE}}(y,\hat{y}) = -y\log\hat{y} - (1-y)\log(1-\hat{y})$$

$$\mathcal{L}_{\mathrm{FL}}(y,\hat{y};\gamma) = -y(1-\hat{y})^{\gamma}\log\hat{y} - (1-y)\hat{y}^{\gamma}\log(1-\hat{y})$$

- We also can be interpret focal loss as asymmetric entropy regularization.

  By using the relation $(1-\hat{y})^{\gamma} \approx 1-\gamma\hat{y}$

$$\mathcal{L}_{\mathrm{FL}}(y,\hat{y};\gamma)$$

$$= -y(1-\hat{y})^{\gamma}\log\hat{y} \approx -y\log\hat{y} - (1-y)\log(1-\hat{y}) - \gamma\{-y\hat{y}\log\hat{y} - (1-y)(1-\hat{y})\log(1-\hat{y})\}$$

$$= \mathcal{L}_{BCE}(y,\hat{y}) - \gamma H_{asym}(y,\hat{y})$$

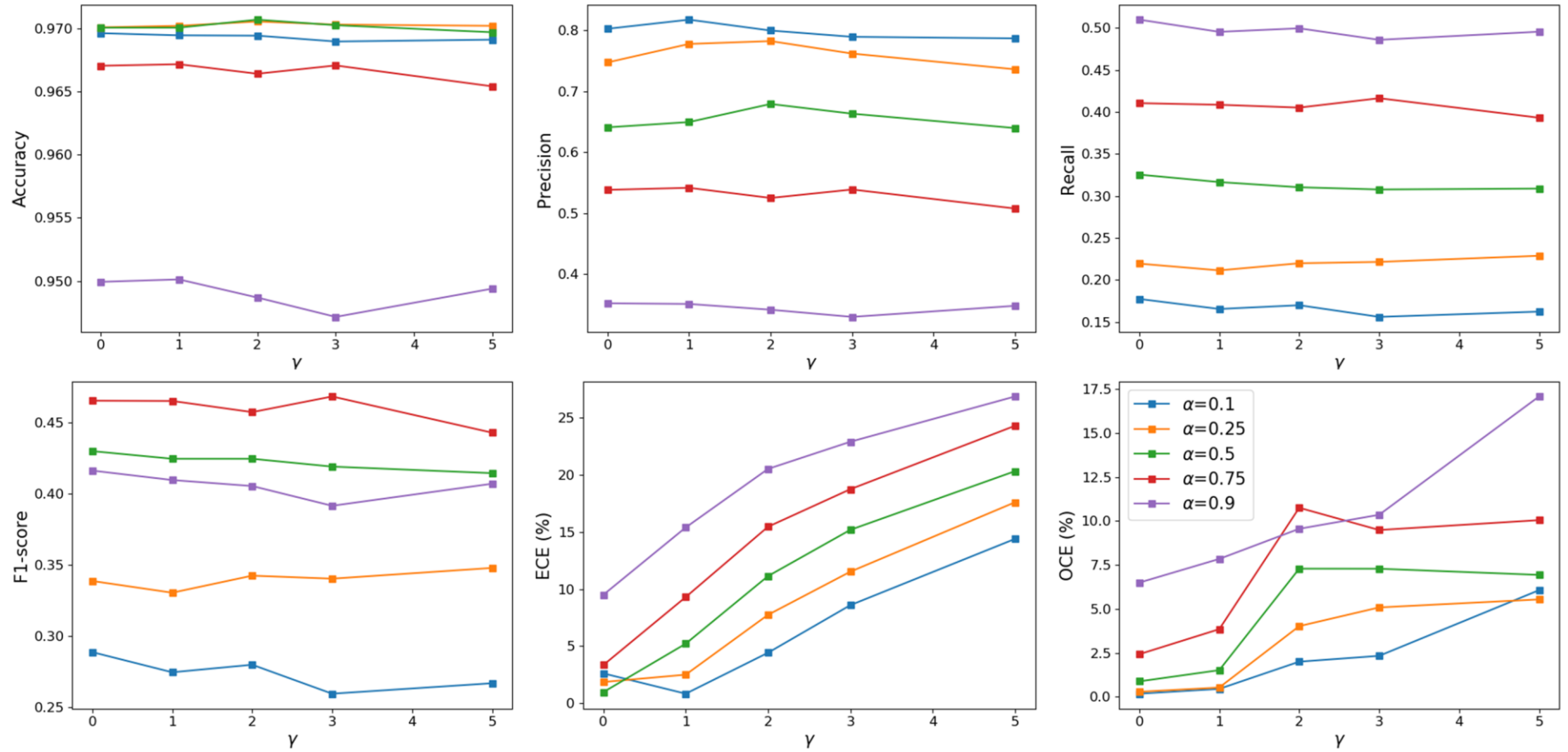
- It penalize easy-to-detect samples, i.e. high-confident predictions, more.

In preparation

Reliability of prediction systems

Experimental results

$$\mathcal{L}_{\mathrm{WFL}}(y, \hat{y}; \alpha, \gamma) = -\alpha y (1-\hat{y})^{\gamma} \log \hat{y} \; - (1-\alpha)(1-y)\hat{y}^{\gamma} \log(1-\hat{y})$$



- Using focal loss can improve precision, recall and f1-score.
- However, it degrades reliability of predictions (low ECE and OCE) → may be the result of cost-penalty.

In preparation

KAIST CHEMISTRY

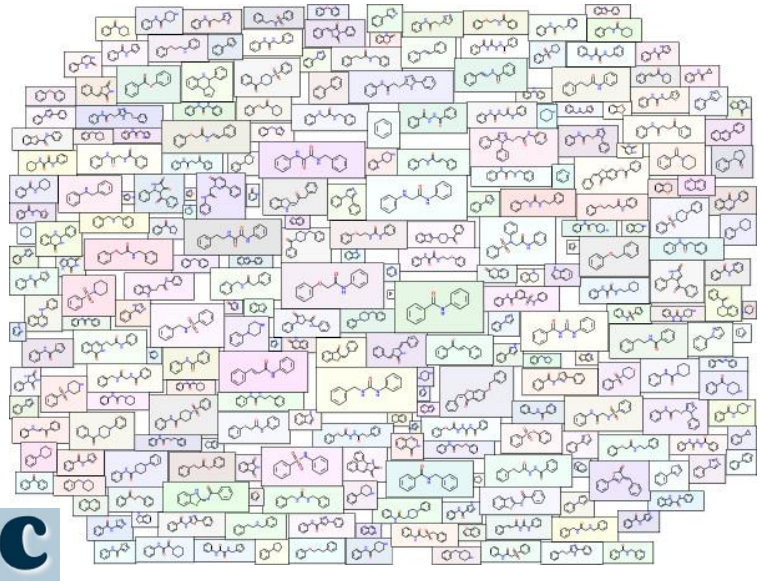
Molecular generative model based on ARAE

KAIST

Molecular generative model based on ARAE

Motivation

10^{10} molecules in the ZINC dataset



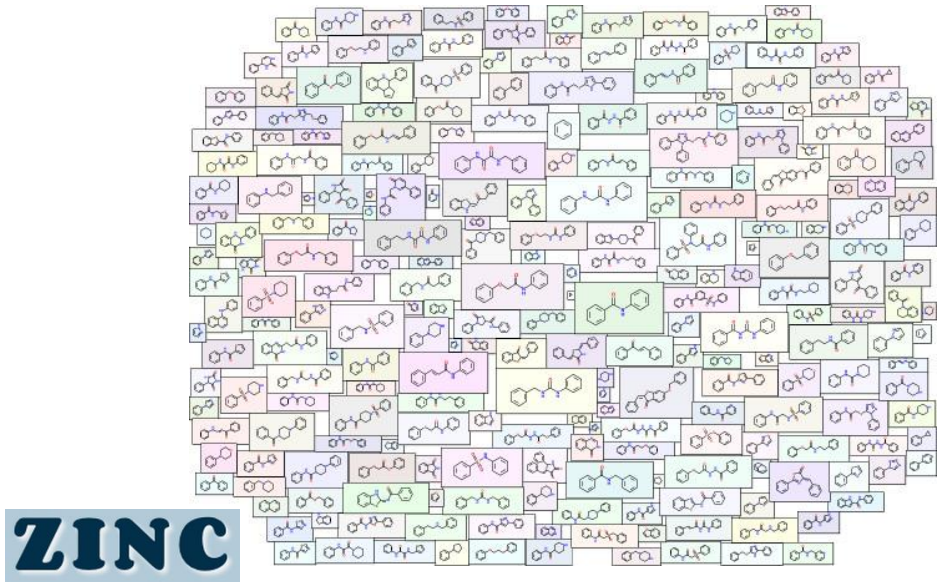
Screening candidates from a library is …

- Like **“finding a needle in a haystack”**

Molecular generative model based on ARAE

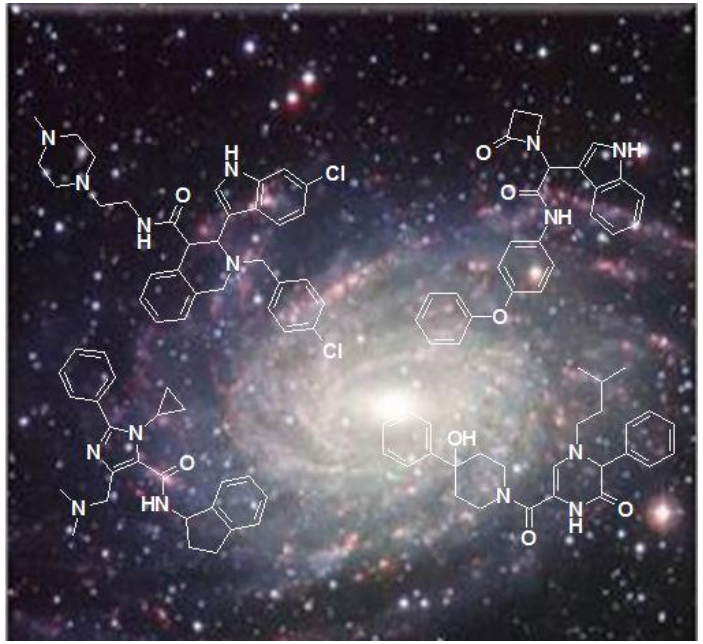
Motivation

10^{10} molecules in the ZINC dataset



≪

10^{60} molecules in the chemical space

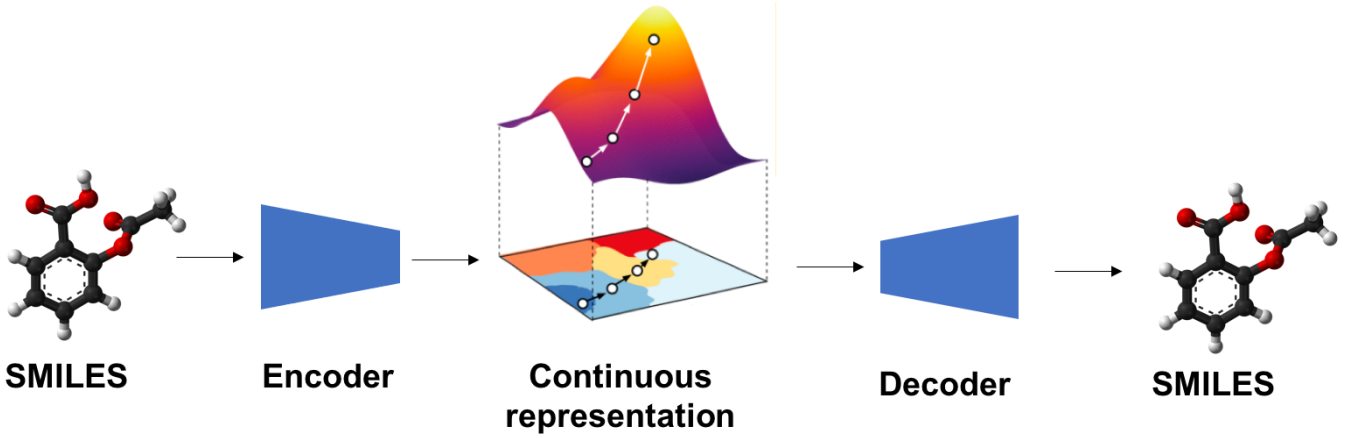


Screening candidates from a library has a few drawbacks…

- Like “finding a needle in a haystack”
- **Cannot design novel molecules which are not included in the library**

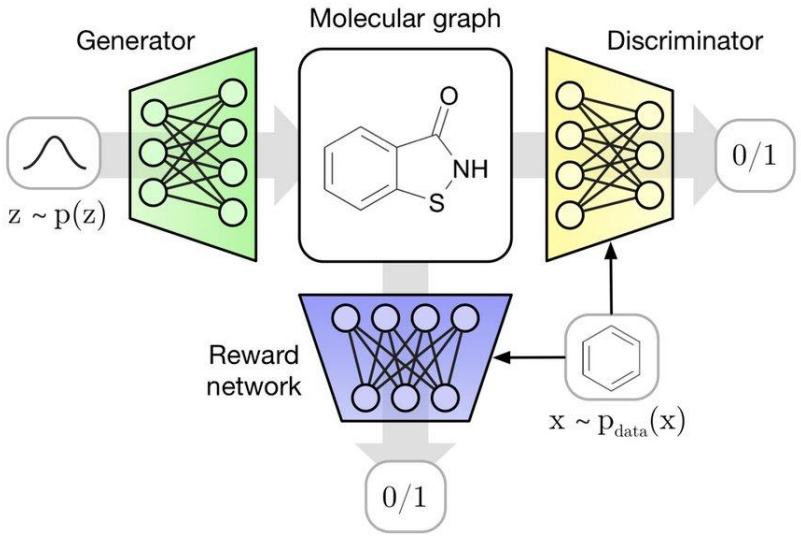
Molecular generative model based on ARAE

We need generative models for *de novo* molecular designs



Based on "variational autoencoder (VAE)"

Gómez-Bombarelli, Rafael, et al. "Automatic chemical design using a data-driven continuous representation of molecules." *ACS central science* 4.2 (2018): 268-276.



Based on "generative adversarial network (GAN)"

De Cao, Nicola, and Thomas Kipf. "MolGAN: An implicit generative model for small molecular graphs." *arXiv preprint arXiv:1805.11973* (2018).

Molecular generative model based on ARAE

Preliminary) VAE and ChemicalVAE

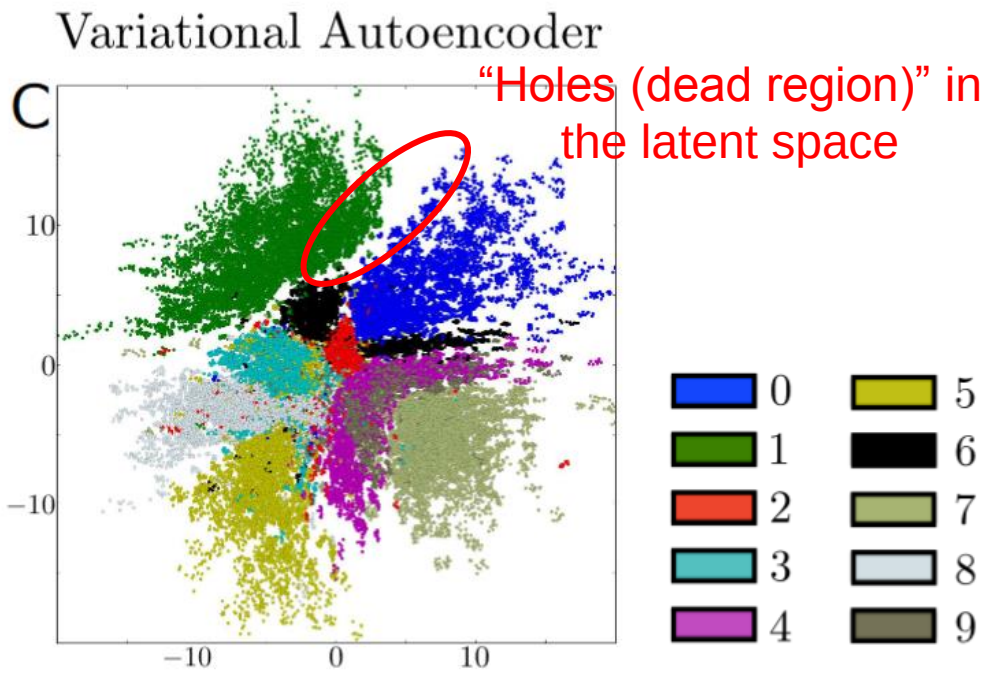
$$\min_{\theta,\phi} \mathbb{E}_{\mathbf{x}\sim p_r(\mathbf{x})}\left[\mathcal{L}_{rec}(\theta,\phi) + \mathrm{KL}\left(q_\theta(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\right)\right]$$

intractable prior

Drawbacks of VAE

- Variational inference of the posterior distribution $q_\phi(\mathbf{z}|\mathbf{x})$ sometimes does not work well.

  → Simple Gaussian prior $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2 I)$ hurts the quality of approximate posterior



"Holes (dead region)" in the latent space

Makhzani, Alireza, et al. "Adversarial autoencoders." *arXiv preprint arXiv:1511.05644* (2015).

Blurred images are generated with VAE

Molecular generative model based on ARAE

Preliminary) VAE and ChemicalVAE

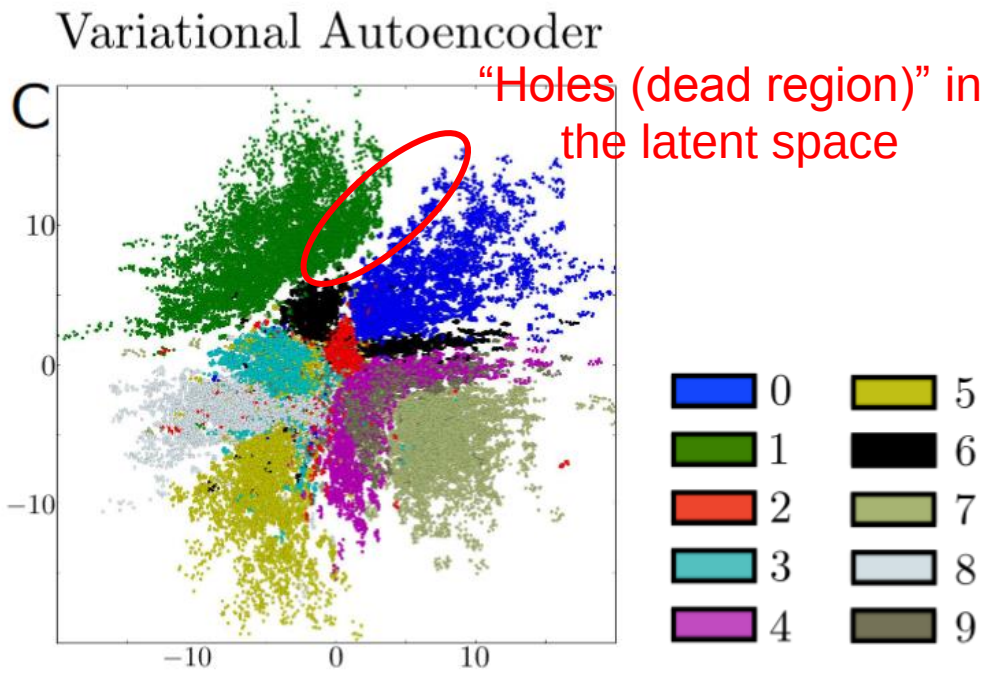
$$\min_{\theta,\phi} \mathbb{E}_{\mathbf{x}\sim p_r(\mathbf{x})}\left[\mathcal{L}_{rec}(\theta,\phi) + \mathrm{KL}\left(q_\theta(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\right)\right]$$

intractable prior

Drawbacks of VAE

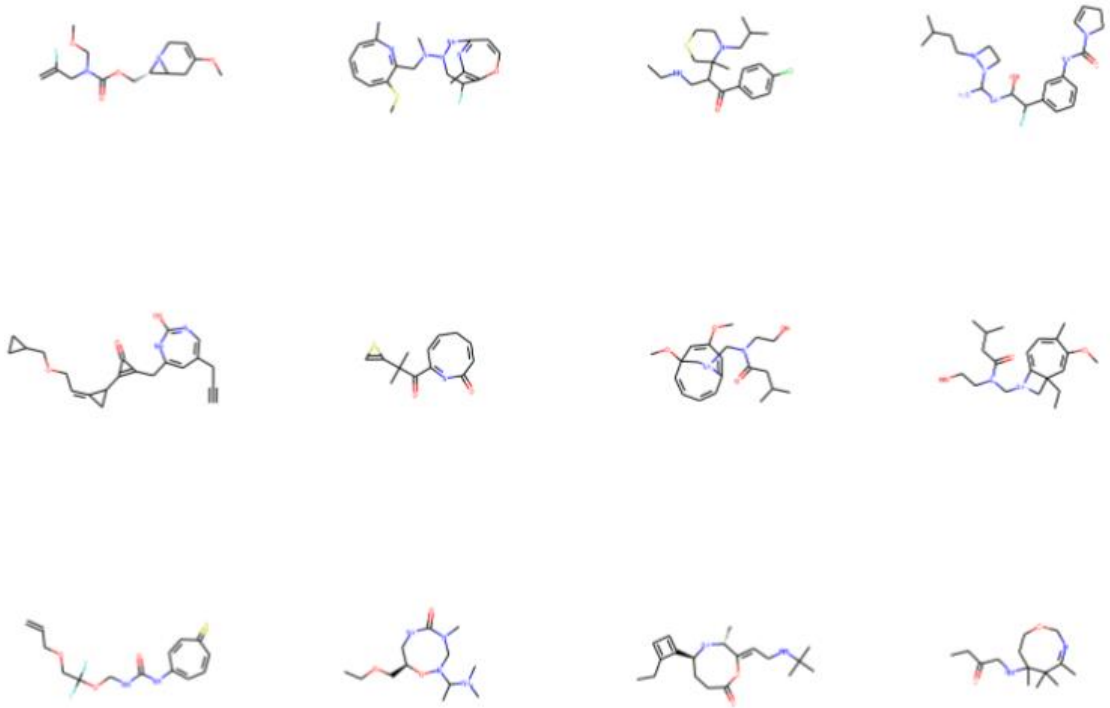
- Variational inference of the posterior distribution $q_\phi(\mathbf{z}|\mathbf{x})$ sometimes does not work well.

  → Simple Gaussian prior $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2 I)$ hurts the quality of approximate posterior



"Holes (dead region)" in the latent space

Makhzani, Alireza, et al. "Adversarial autoencoders." *arXiv preprint arXiv:1511.05644* (2015).



ChemicalVAE generates non-druglike molecules

Gómez-Bombarelli, Rafael, et al. "Automatic chemical design using a data-driven continuous representation of molecules." *ACS central science* 4.2 (2018): 268-276.

Molecular generative model based on ARAE

Preliminary) Probability measures

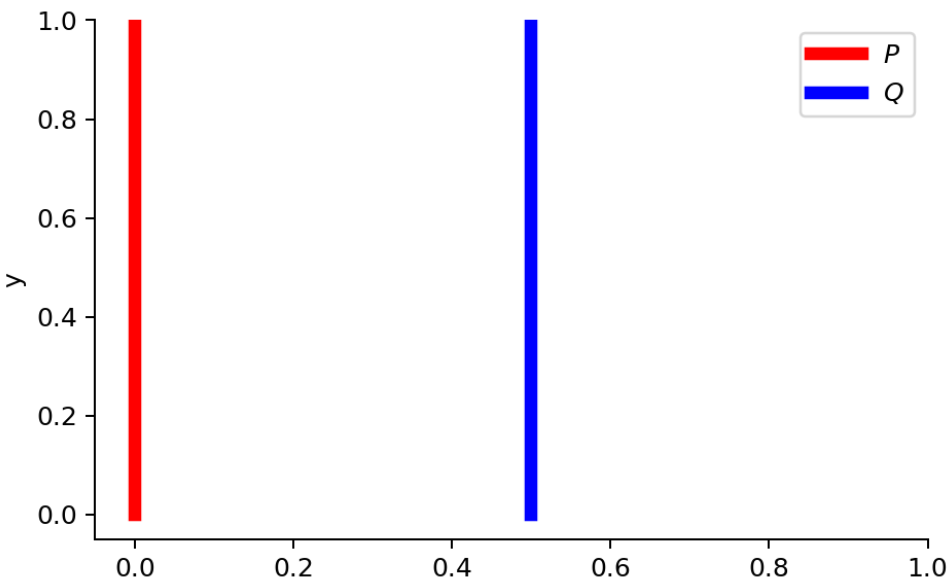
Let us consider very simple yet intuitive example: two probability distributions with disjoint support

$$\mathrm{KL}(P||Q) = \sum_{x=0, y\sim U(0,1)} 1 \cdot \log \frac{1}{0} = +\infty$$

$$\mathrm{JS}(P||Q) = \frac{1}{2}\left(\sum_{x=0, y\sim U(0,1)} 1 \cdot \log \frac{1}{1/2} + 1 \cdot \log \frac{1}{1/2} \right) = \log 2$$

$$\forall (x, y) \in P, x = 0 \text{ and } y \sim U(0,1)$$

$$\forall (x, y) \in Q, x = \theta, s.t. 0 \leq \theta \leq 1, \text{and } y \sim U(0,1)$$



- The derivative of KL- and JS- divergences w.r.t θ is none or zero.

  → We cannot apply gradient-based optimizations.

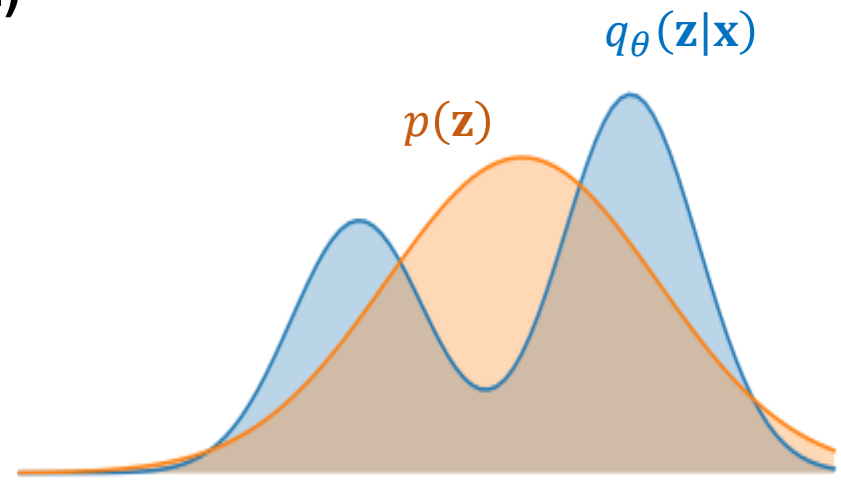
$\mathrm{W}(P, Q) = |\theta|$: Wasserstein distance

- Using Wasserstein distance gives correct gradient and enables us to correctly estimate data distributions.

Arjovsky, Martin, Soumith Chintala, and Léon Bottou. "Wasserstein gan." *arXiv preprint arXiv:1701.07875* (2017).

Molecular generative model based on ARAE

Our approach: "Adversarially Regularized Autoencoder (ARAE)"



- VAE

$$\min_{\theta,\phi} \mathbb{E}_{\mathbf{x}\sim p_r(\mathbf{x})}\left[\mathcal{L}_{rec}(\theta,\phi) + \mathrm{KL}\left(q_\theta(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\right)\right]$$

  - ✓ Using KL-divergence with too simple prior can hurts the quality of approximate posterior

- ARAE

$$\min_{\theta,\phi,\psi} \mathbb{E}_{\mathbf{x}\sim p_r(\mathbf{x})}\left[\mathcal{L}_{rec}(\theta,\phi) + \mathrm{W}\left(q_\theta(\mathbf{z}|\mathbf{x}), p_\psi(\tilde{\mathbf{z}})\right)\right]$$

$$\mathrm{W}\left(q_\theta(\mathbf{z}|\mathbf{x}), p_\psi(\tilde{\mathbf{z}})\right) = \sup_{\mathrm{Lip}(\omega)\leq 1} \mathbb{E}_{\mathbf{z}\sim q_\theta(\mathbf{z}|\mathbf{x})}[f_\omega(\mathbf{z})] - \mathbb{E}_{\tilde{\mathbf{z}}\sim p_\psi(\tilde{\mathbf{z}})}[f_\omega(\tilde{\mathbf{z}})]$$

  - ✓ Using Wasserstein-distance to measure the distance between the posterior and the (generated) prior.
  - ✓ The generator network maps simple Gaussian distribution $p(s) = \mathcal{N}(s, I^2)$ to the more flexible prior distribution $p_\psi(\tilde{\mathbf{z}})$, increasing the expressive power on the posterior.

Zhao, Jake, et al. "Adversarially regularized autoencoders." *arXiv preprint arXiv:1706.04223* (2017).

Molecular generative model based on ARAE

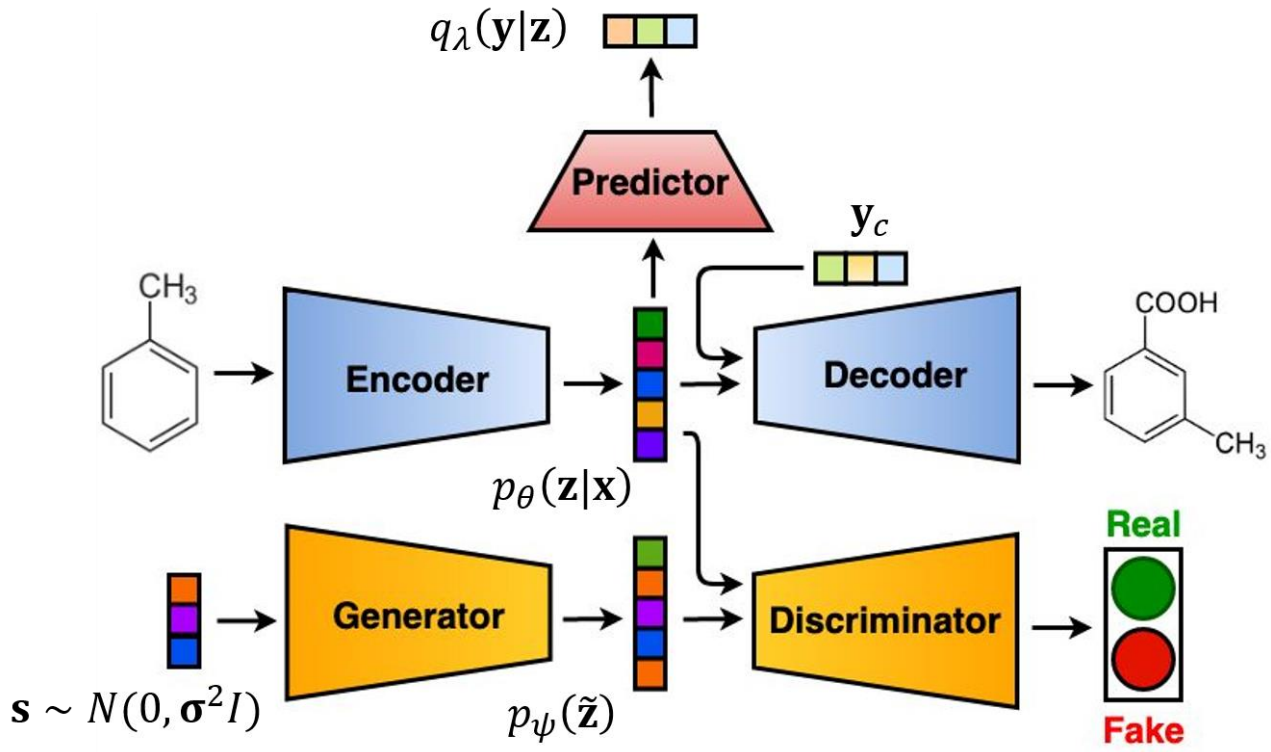
Our approach: "Adversarially Regularized Autoencoder (ARAE)"

- ARAE with property disentanglement (conditional ARAE)

$$\min_{\theta,\phi} \mathbb{E}_{\mathbf{x}\sim p_r(\mathbf{x})}\left[\mathcal{L}_{rec}(\theta,\phi) + \mathrm{W}\left(q_\theta(\mathbf{z}|\mathbf{x}), p_\psi(\tilde{\mathbf{z}})\right) + \mathrm{VMI}(z, y; \theta, \lambda)\right]$$

- ✓ $\mathrm{VMI}(\boldsymbol{z}, \boldsymbol{y}; \theta, \lambda) = \mathbb{E}_{\mathbf{z}\sim p_\theta(\mathbf{z}|\mathbf{x})}\left[\mathbb{E}_{\mathbf{y}\sim p(\mathbf{y}|\mathbf{x})}[\log q_\lambda(\mathbf{y}|\mathbf{z})]\right]$

  : information amount of $\mathbf{y}$ in $\mathbf{z}$
- ✓ To alleviate the original property information in $\mathbf{z}$
- ✓ In the decoding (generation) phase, we add a vector of target property $\mathbf{y}_c$ to generate molecules having desired property.



Seung Hwan Hong, Ryu, Seongok, Jaechang Lim, and Woo Youn Kim.
"Molecular generative model based on an adversarially regularized autoencoder", JCIM (2019)

Molecular generative model based on ARAE

Results 1: Performance on the QM9 dataset (small molecule dataset)

Method	Validity (A)	Uniqueness (B)	Novelty (C)	Novel/Sample (A×B×C)
ChemicalVAE	0.103	0.675	0.900	0.063
GrammarVAE	0.602	0.093	0.809	0.045
GraphVAE	0.557	0.670	0.616	0.261
GraphVAE/imp	0.562	0.520	0.758	0.179
GraphVAE NoGM	0.810	0.241	0.610	0.129
MolGAN	**0.981**	0.104	**0.942**	0.096
ARAE	0.862	**0.935**	0.371	**0.299**
ARAE (ZINC)	0.903	1.000	1.000	0.903

- Validity : the ratio of the number of valid molecules to the number of generated samples
- Uniqueness : the ratio of the number of unrepeated molecules to the number of valid molecules
- Novelty : the ratio of the number of molecules which are not included in the training set to the number of unique molecules

✓ We can mitigate limitations in previous models, low-validity and low-uniqueness problems.

Seung Hwan Hong, Ryu, Seongok, Jaechang Lim, and Woo Youn Kim.
"Molecular generative model based on an adversarially regularized autoencoder", JCIM (2019)

Molecular generative model based on ARAE

Results 2: Conditional generation of molecules with CARAE

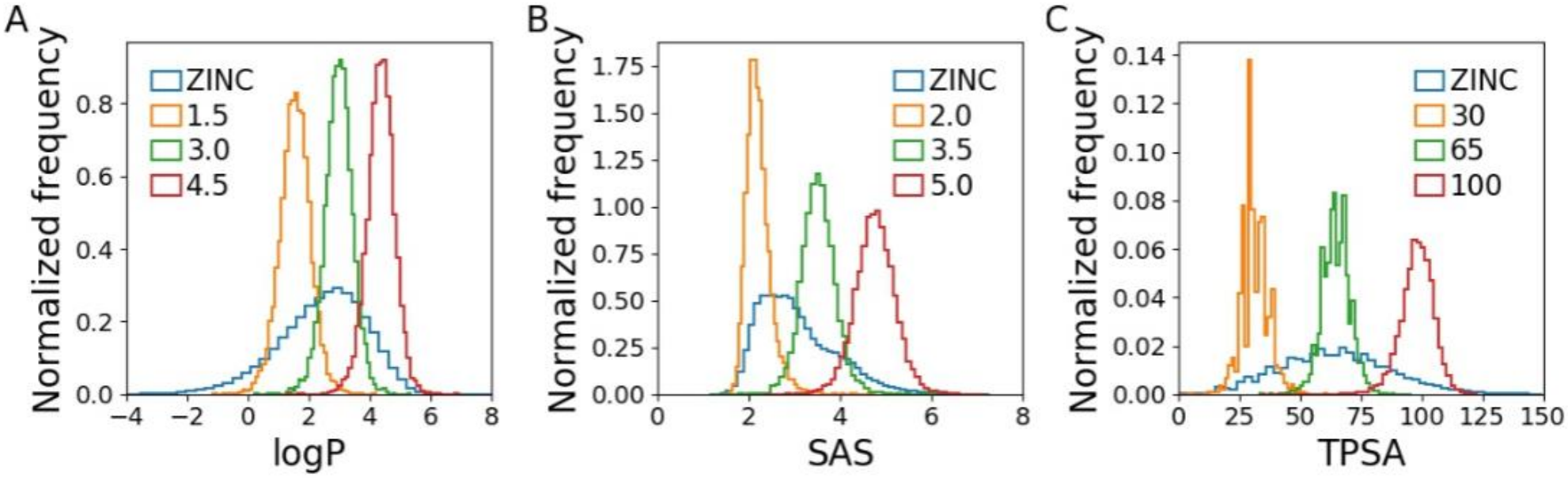


Figure 4: Distributions of molecular property - (a) logP, (b) SAS and (c) TPSA - when molecules are generated by specifying a desired property. Note that the curves labeled with ZINC denote the distribution of each molecular property in the ZINC dataset.

Seung Hwan Hong, Ryu, Seongok, Jaechang Lim, and Woo Youn Kim.
"Molecular generative model based on an adversarially regularized autoencoder", JCIM (2019)

Molecular generative model based on ARAE

Results 2: Conditional generation of molecules with CARAE

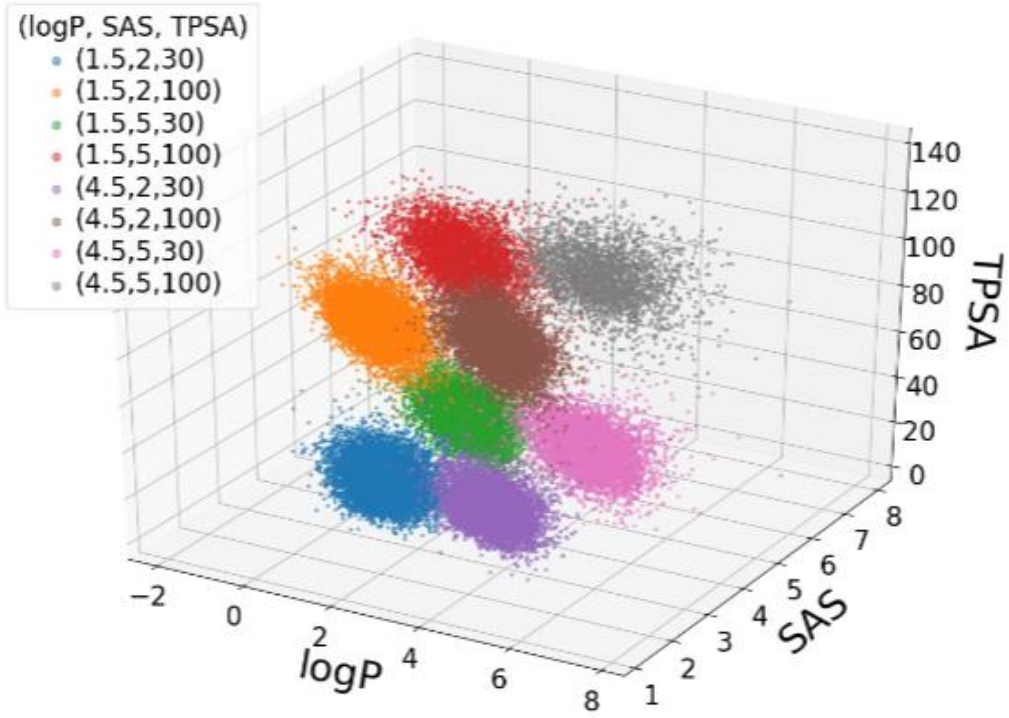


Figure 5: Joint distribution of the logP, SAS and TPSA values of molecules generated with the simultaneous control of the three target properties denoted in the legend.

Seung Hwan Hong, Ryu, Seongok, Jaechang Lim, and Woo Youn Kim.
"Molecular generative model based on an adversarially regularized autoencoder", JCIM (2019)

Molecular generative model based on ARAE

Results 3: Design EGFR-inhibitors (for lung cancer medications)

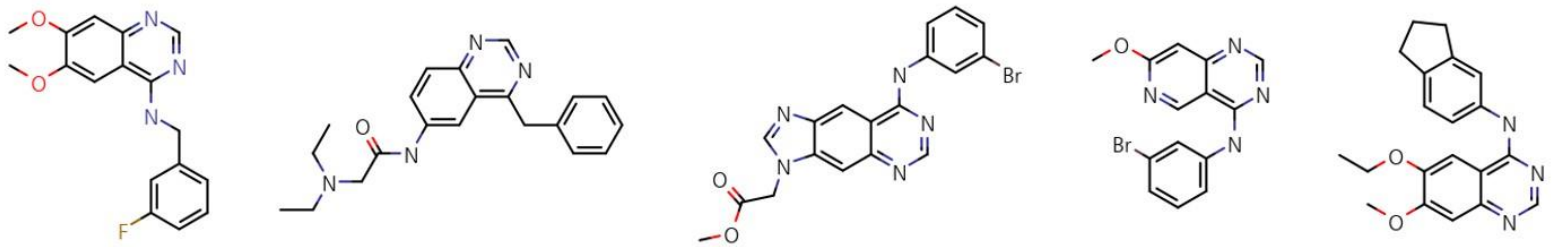
- Scenario A : to generate compounds satisfying EGFR-active

  → 537 molecules out of 931 (57.7%) were predicted as active by the Bayesian-GCN.

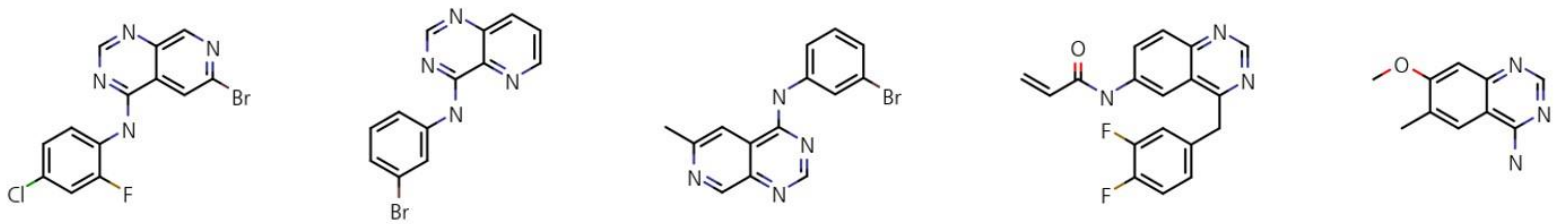
- Scenario B : to generate compounds satisfying EGFR-active, logP=2.5, SAS=1.5, TPSA=60 (Lipinski's rule)

  → 502 molecules out of 1067 (47.1%) are predicted as active by the Bayesian-GCN.

A



B



Seung Hwan Hong, Ryu, Seongok, Jaechang Lim, and Woo Youn Kim.
"Molecular generative model based on an adversarially regularized autoencoder", JCIM (2019)

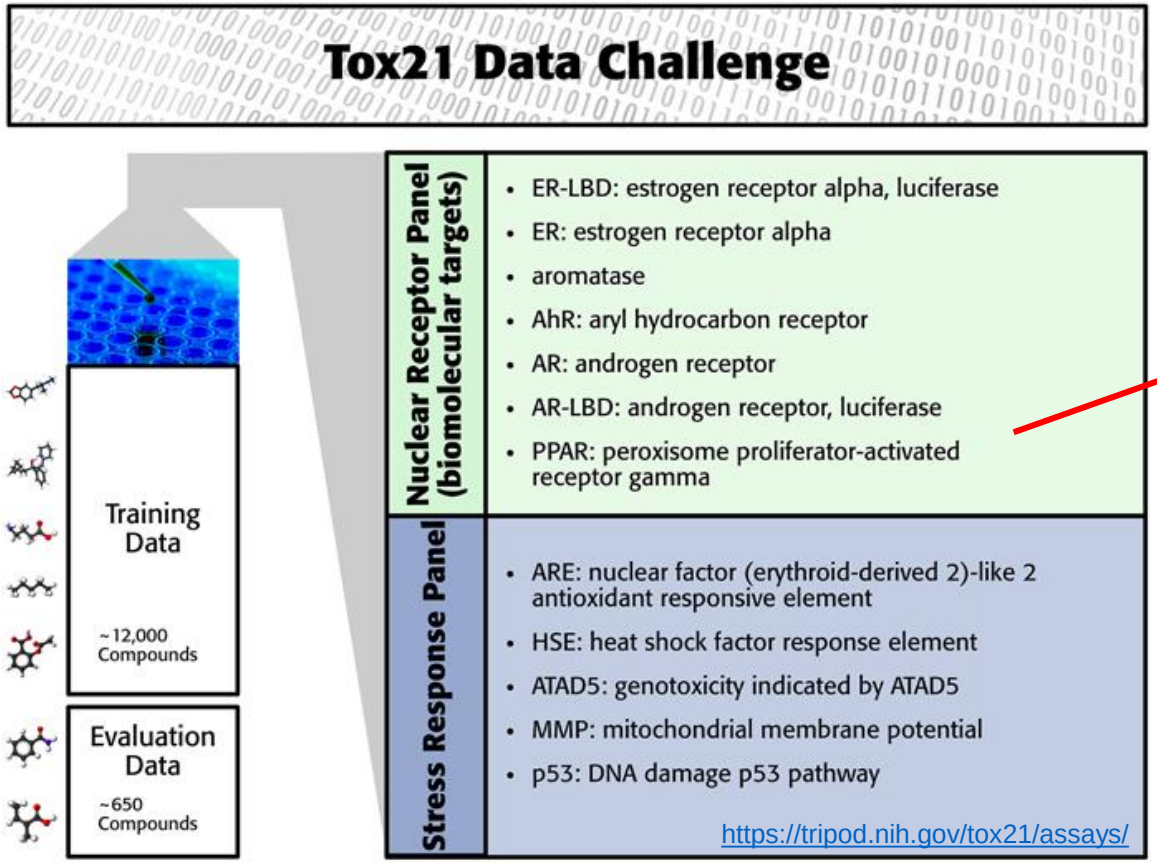
KAIST CHEMISTRY

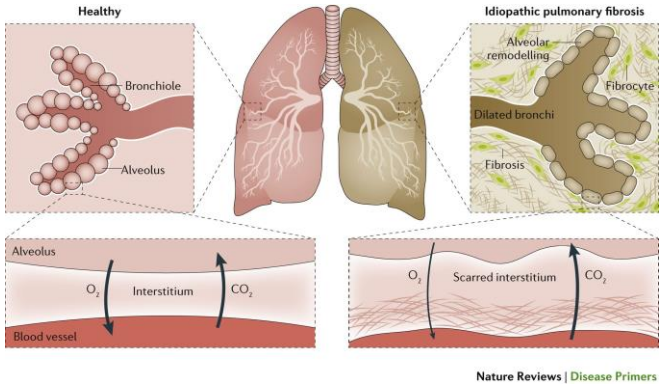
Future directions

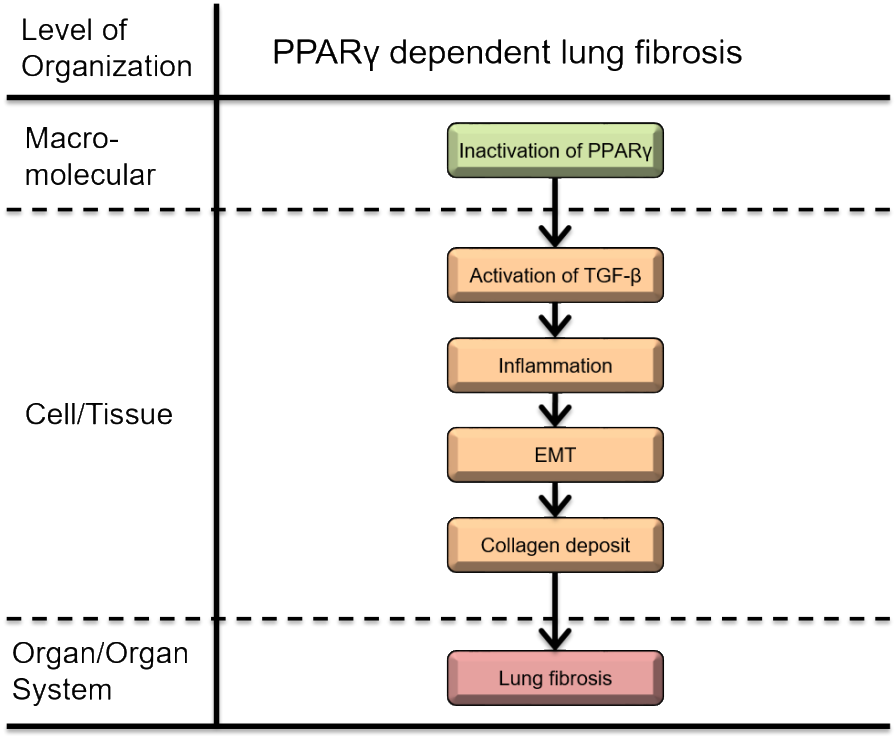
KAIST

Future directions

Real-world applicable *in silico* models to replace in vitro/in vivo assays



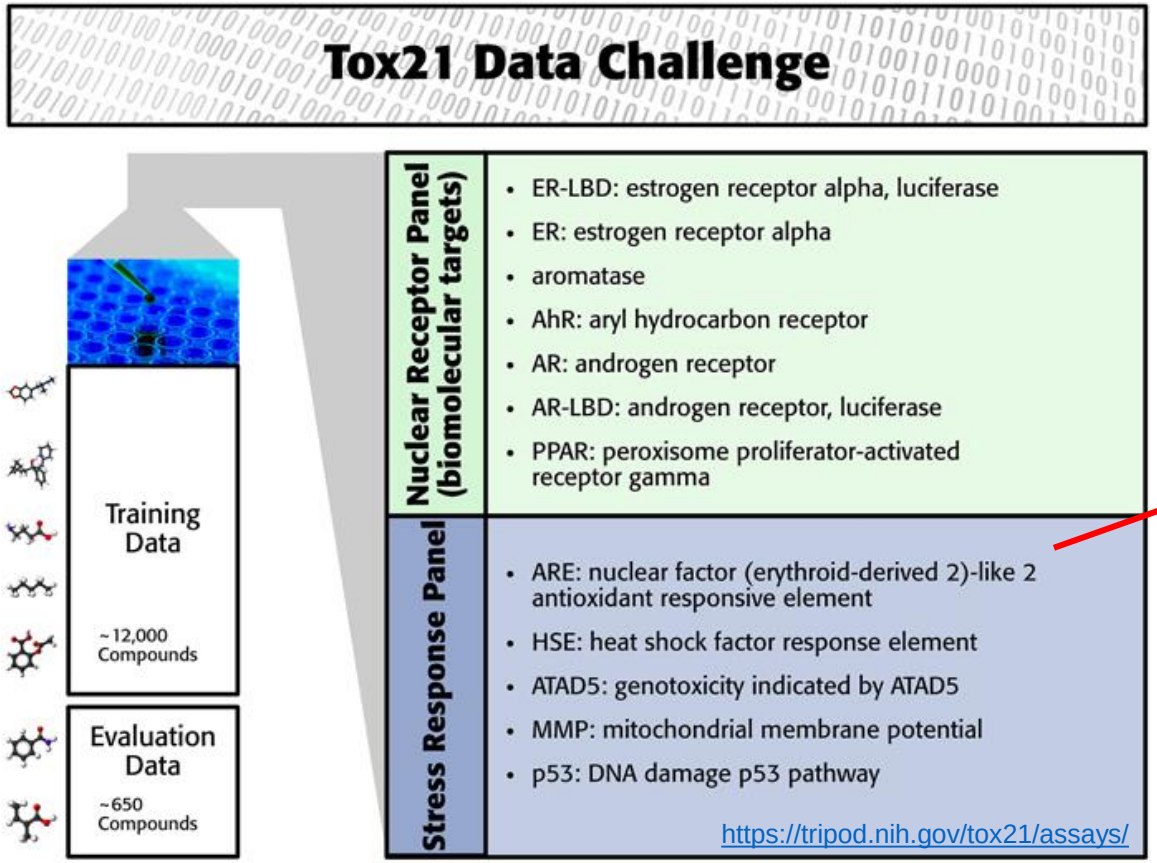




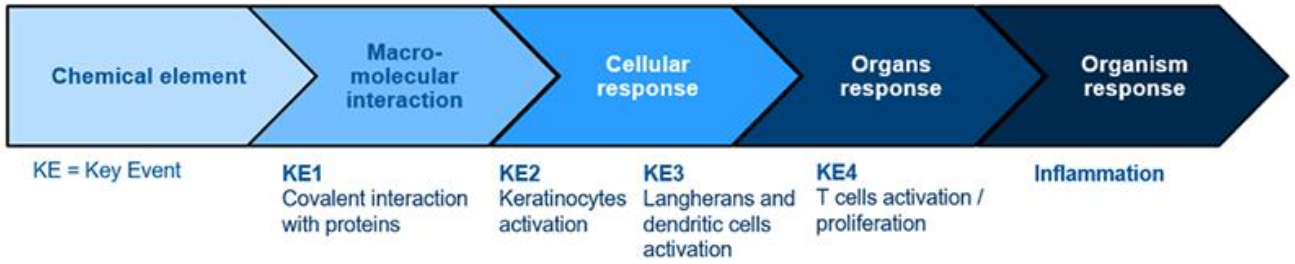
- NR-ppar-gamma is related to lung fibrosis.

Future directions

Real-world applicable *in silico* models to replace in vitro/in vivo assays



- SR-ARE is related to skin-sensitisation.



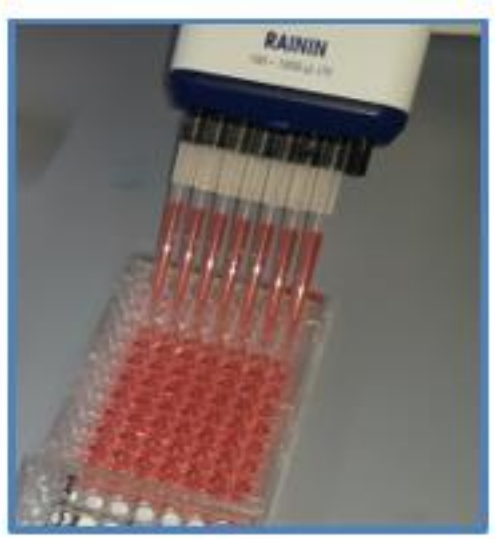
Future directions

Real-world applicable *in silico* models to replace in vitro/in vivo assays

In vivo

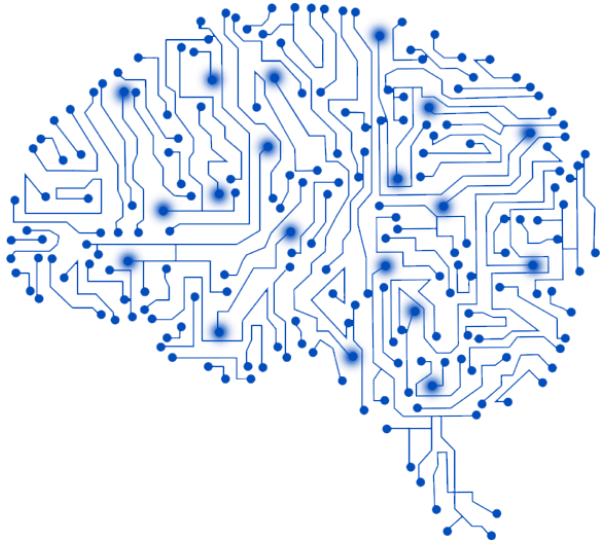
Alternative

In vitro



Alternative

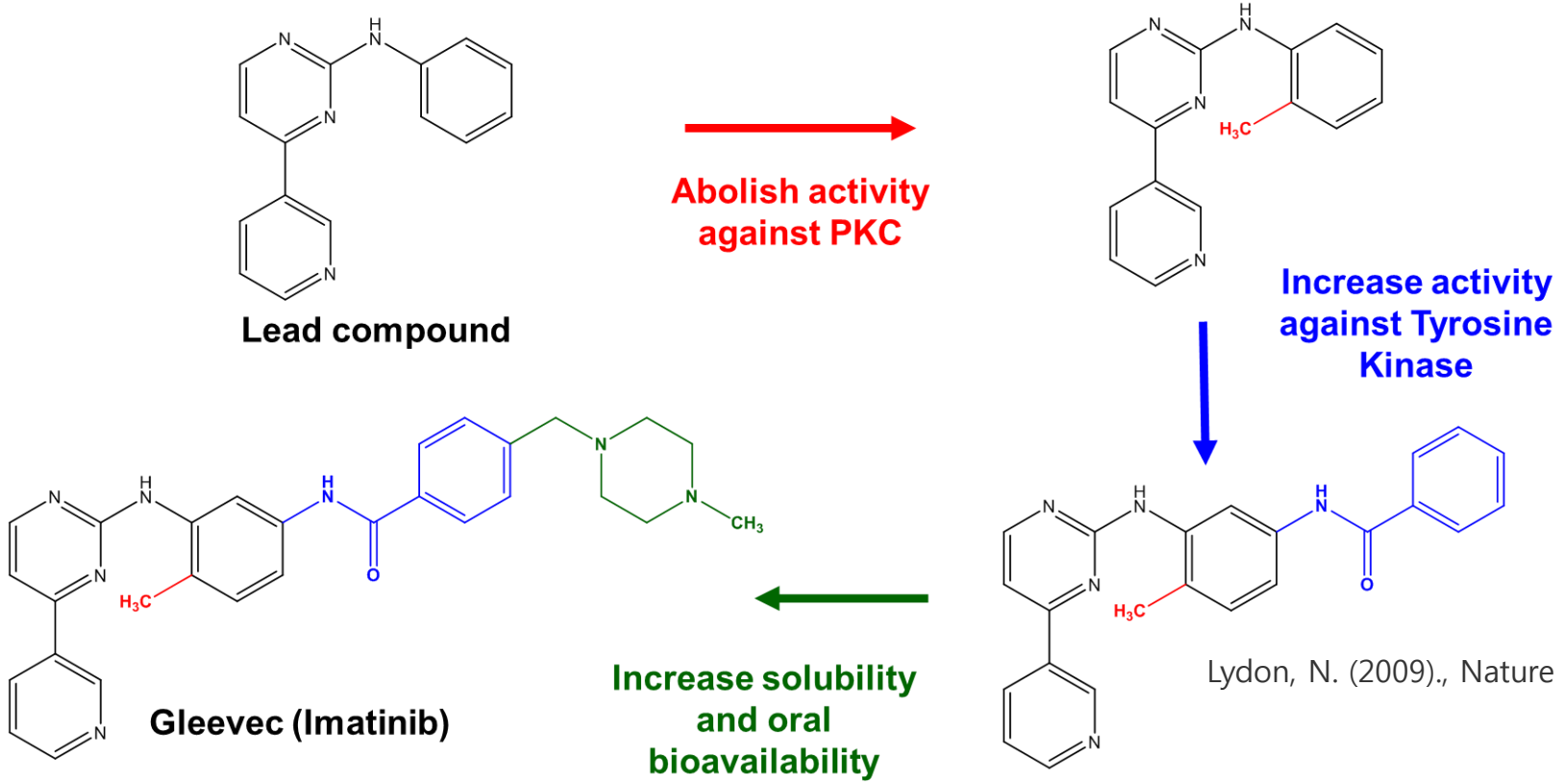
In silico



- In 2013 ~ 2015, 25 ~ 30% of clinical trials were failed due to toxicity (ADME/T) issues.
- Toward cheap and fast assay platforms for developing cosmetics and pharmaceutics.
- Uncertainty quantification, active learning and some other skills (e.g. prediction calibrations) would be useful.

Future directions

Lead optimization: changing hit molecules to more drug-like ones



Lydon, N. (2009)., Nature medicine, 15(10), 1153-1157

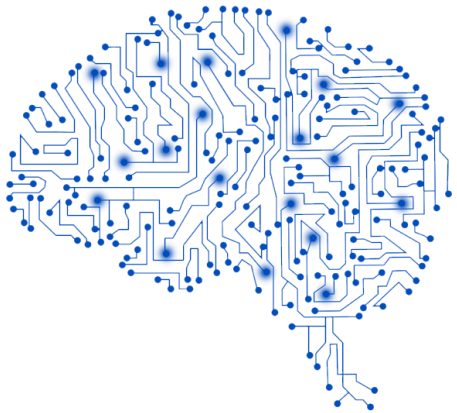
- "Scaffold-based molecular design using graph generative model." *arXiv:1905.13639* (2019).
- "Mol-CycleGAN: a generative model for molecular optimization." *Journal of Cheminformatics* 12.1 (2020)
- "DeepScaffold: a comprehensive tool for scaffold-based de novo drug discovery using deep learning." *Journal of Chemical Information and Modeling* (2019).

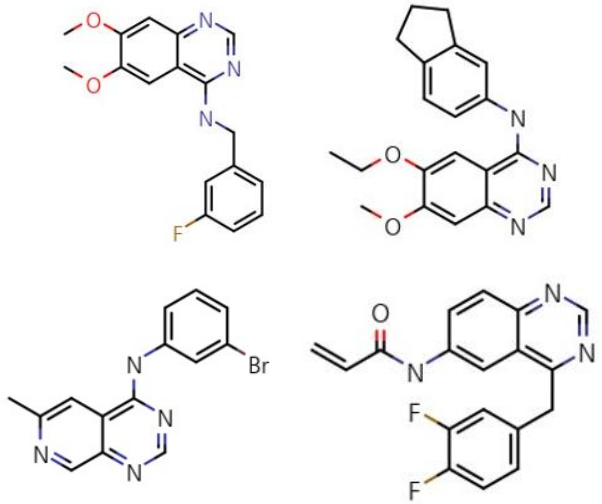
Future directions

Comprehensive platform: generation, prediction and real-world assay

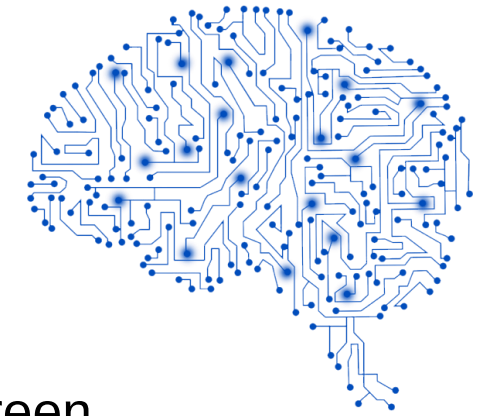
EGFR active

Generation

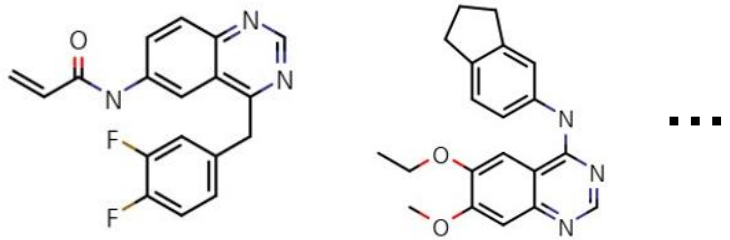




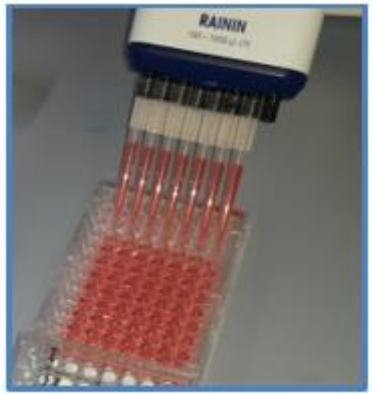
Evaluation (prediction models)



To screen
- High efficacy
- Non-toxic



Real-world assays



Feedback
- Gathering more data
- Active learning

KAIST CHEMISTRY

Thank you for your attention!

KAIST