Week 07

KAIST CHEMISTRY

Molecular graphs & Graph Neural Network (GNN)

Prof. Ph.D. Woo Youn Kim

Chemistry, KAIST

KAIST

Goals

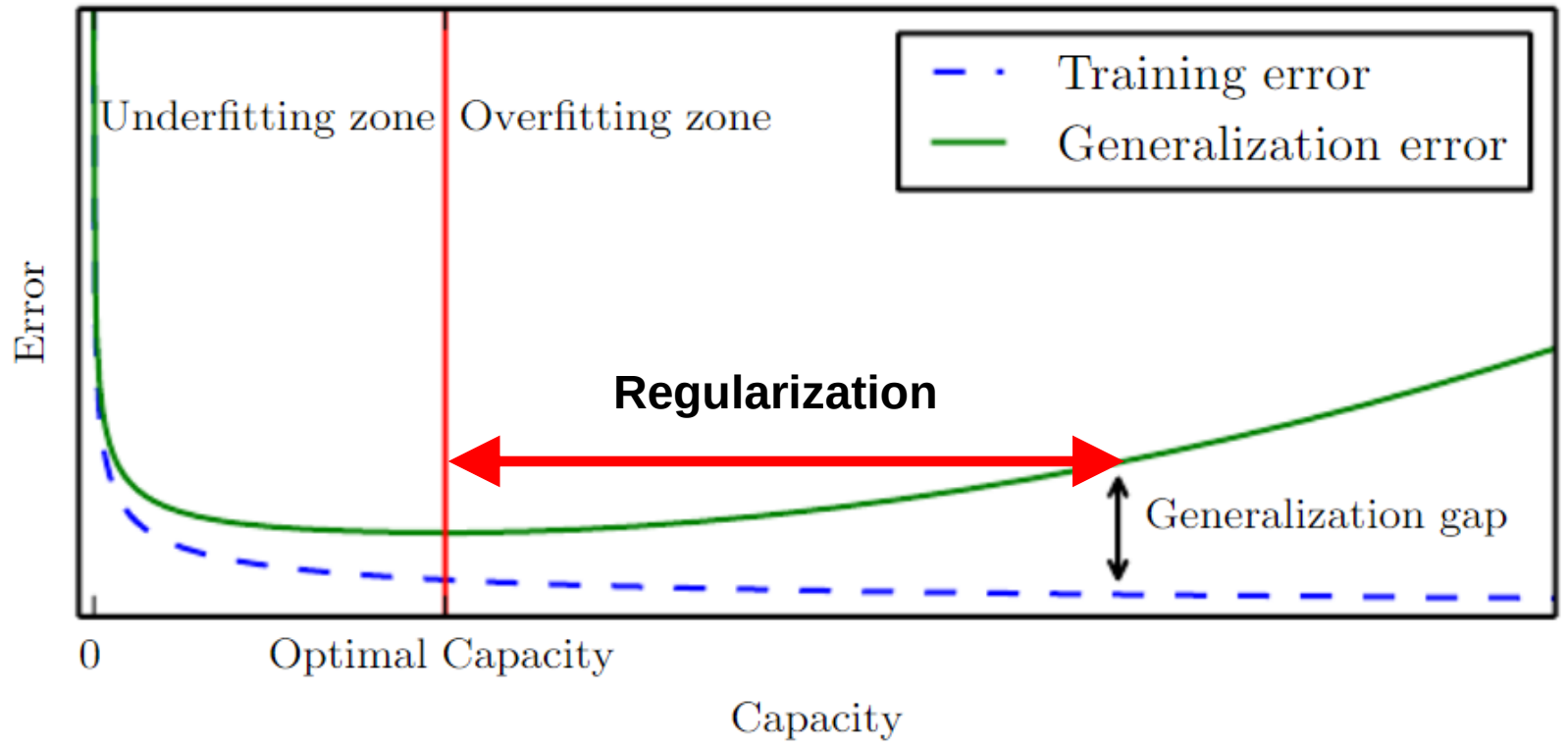
6주	주제	Convolutional Neural Network (CNN) & SMILES
	목표	Understanding CNN and molecular representation with SMILES
	내용	Convolution, receptive field, stride, pooling Supervised learning of Log P and TPSA
7주	주제	Molecular graphs & Graph Neural Network (GNN)
	목표	Understanding GCN and molecular representation with graphs
	내용	Molecular graph representation, graph convolutional network Supervised learning of logP and TPSA
8주	주제	Recurrent Neural Network (RNN)
	목표	Understanding RNN and molecular representation with smiles
	내용	RNN, LSTM, GRU Feature extraction of molecules using RNN

Contents

- Molecular representation with graph
- Weight sharing
- Graph convolution network (GCN)
- Technical issues – inception, skip-connection and attention
- GNN in chemistry
- Practice session: supervised learning of log P and TPSA:

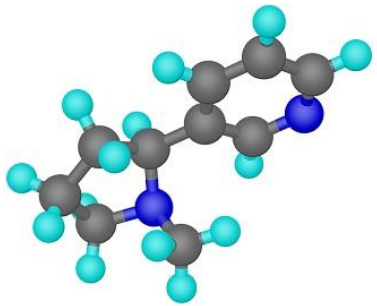
Review

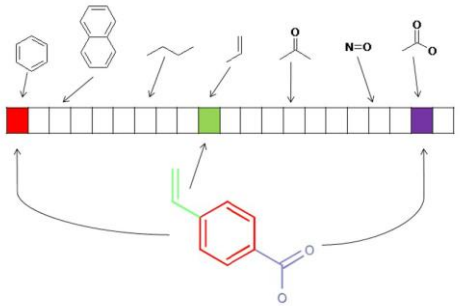
- The main challenge is to find an **optimal capacity** of model for a given task.
- **Regularization** is any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error.



CNN, RNN, GCN

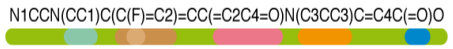
Review





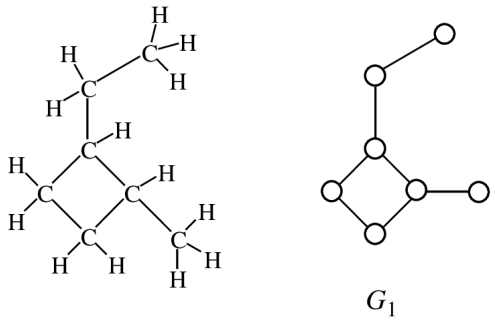
Molecular fingerprint

Multi-layer perceptron (MLP)



SMILES

Convolutional neural network (CNN)



G_1

Molecular graph

Graph neural network (GNN)

Source

Data Science Training

Standard Techniques and Deep Learning

November 2017

Overview

In 2012, the Harvard Business School called data scientist the sexiest job of the 21st century, and in 2015 McKinsey projected that "by 2018, the U.S. alone may face a 50-60 percent gap between supply and requisite demand of deep analytic talent". Why has data scientist become a most in-demand job ?
Today massive amounts of data are available in all areas of science, government and industry. Exploited sensibly, these raw data can significantly improve the efficiency of research, services and industries in as many fields as healthcare, engineering, finance, telecommunications or urban developments just to name a few.
How are powerful companies like Google, Facebook, IBM or Apple using data analysis techniques ? What are the most important algorithms to know and how to apply them to your projects ?

Lecture 3 - Graph Science
The slides are available here.
Data and coding examples (Python 3) are available here.

Lecture 16 - Deep Learning on Graphs
The slides are available here.
Data and coding examples (Python 3) are available here.

Data Science Training : Standard Techniques and Deep Learning
http://data-science-training-xb.com/

KAIST

Reference paper

MoleculeNet: a benchmark for molecular machine learning†

Zhenqin Wu, ‡[a] Bharath Ramsundar,‡[b] Evan N. Feinberg,§[c] Joseph Gomes, §[a] Caleb Geniesse,[c] Aneesh S. Pappu,[b] Karl Leswing[d] and Vijay Pande*[a]

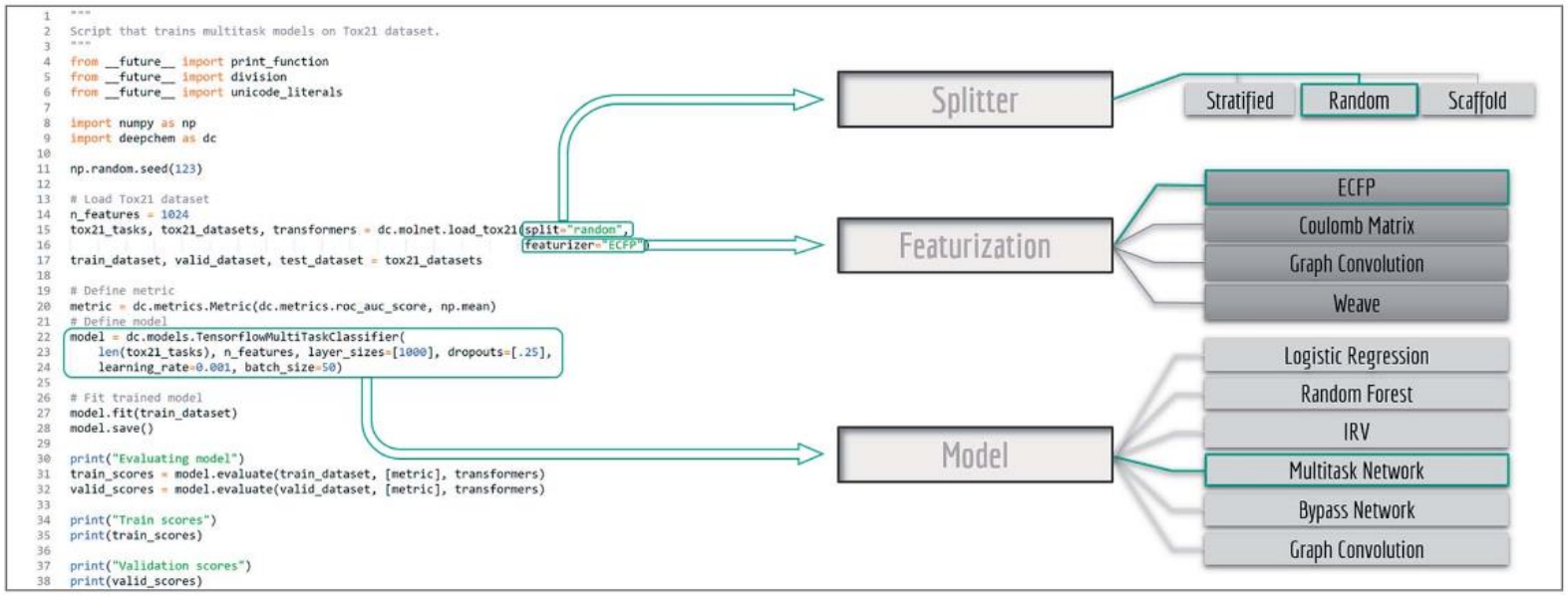


Fig. 1 Example code for benchmark evaluation with DeepChem, multiple methods are provided for data splitting, featurization and learning.

Wu, Zhenqin, et al.
"MoleculeNet: a benchmark for molecular machine learning."
Chemical science 9.2 (2018): 513-530.

Reference paper

Molecular machine learning has been maturing rapidly over the last few years. Improved methods and the presence of larger datasets have enabled machine learning algorithms to make increasingly accurate predictions about molecular properties. However, algorithmic progress has been limited due to the lack of a standard benchmark to compare the efficacy of proposed methods; most new algorithms are benchmarked on different datasets making it challenging to gauge the quality of proposed methods. This work introduces MoleculeNet, a large scale benchmark for molecular machine learning. MoleculeNet curates multiple public datasets, establishes metrics for evaluation, and offers high quality open-source implementations of multiple previously proposed molecular featurization and learning algorithms (released as part of the **DeepChem open source library**). MoleculeNet benchmarks demonstrate that learnable representations are powerful tools for molecular machine learning and broadly offer the best performance. However, this result comes with caveats. Learnable representations still struggle to deal with complex tasks under data scarcity and highly imbalanced classification. For quantum mechanical and biophysical datasets, **the use of physics-aware featurizations can be more important than choice of particular learning algorithm.**

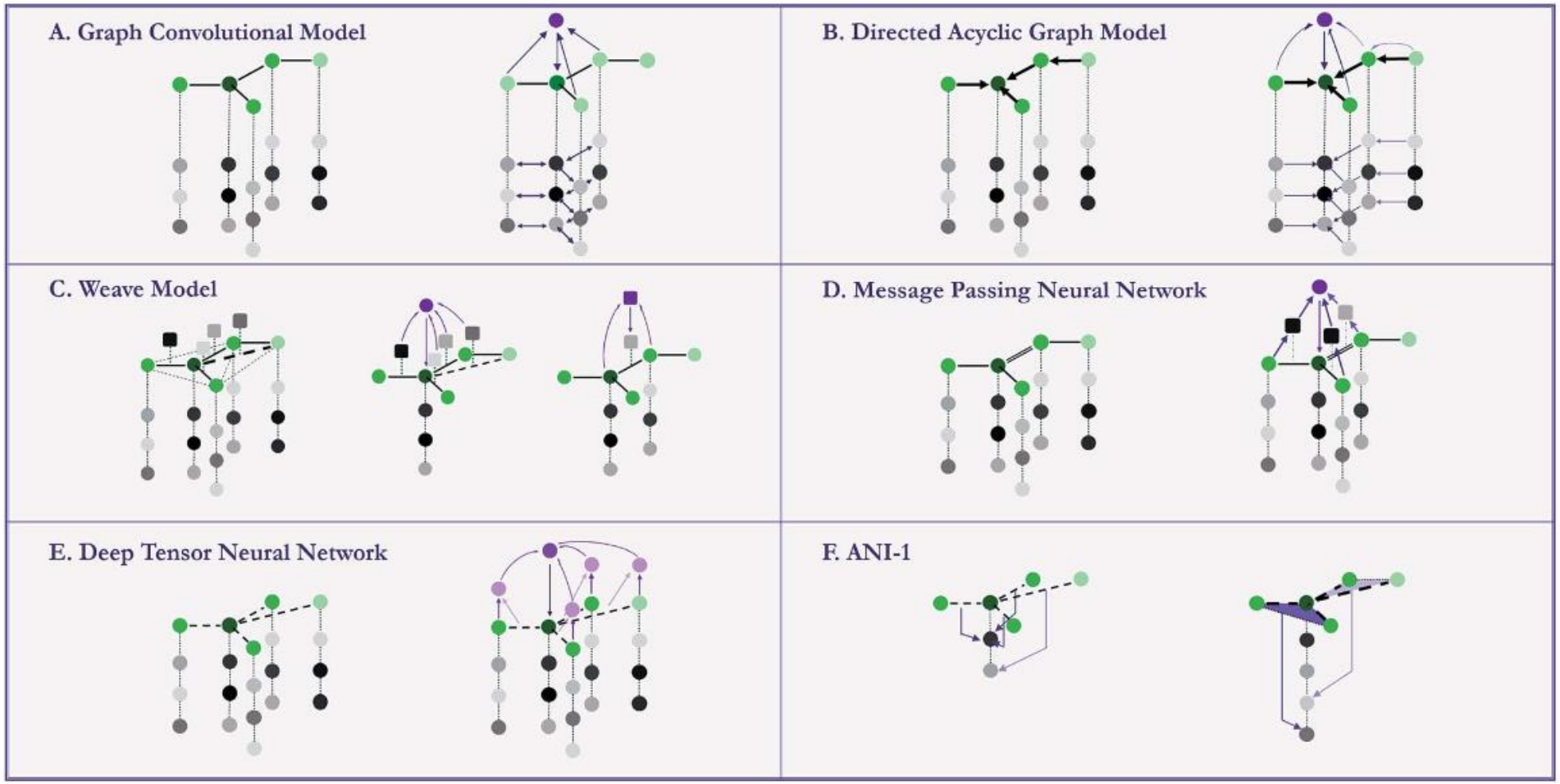
Let machines to featurize molecular data and all we need is graph!

Wu, Zhenqin, et al.
"MoleculeNet: a benchmark for molecular machine learning."
Chemical science 9.2 (2018): 513-530.

Reference paper

DeepChem supports various machine learning tools for successful molecular applications.

https://github.com/deepchem/deepchem

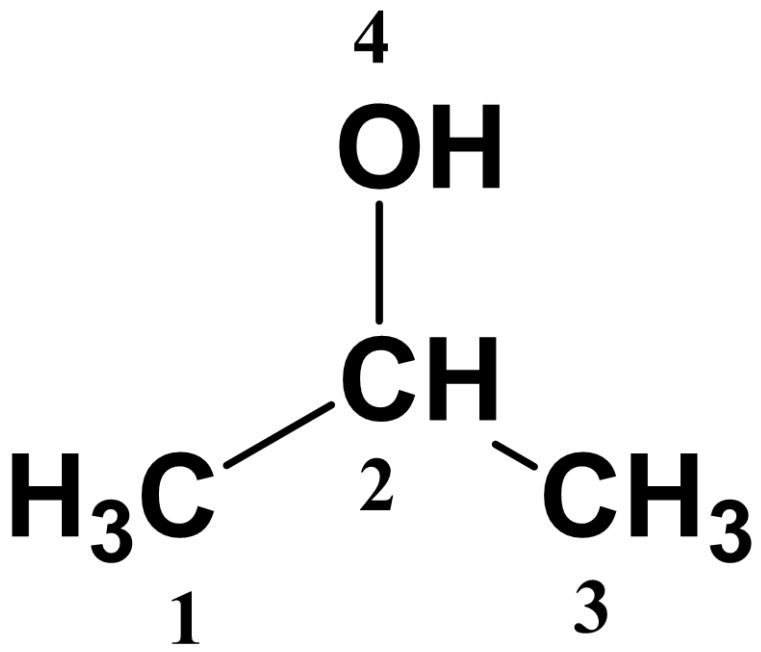


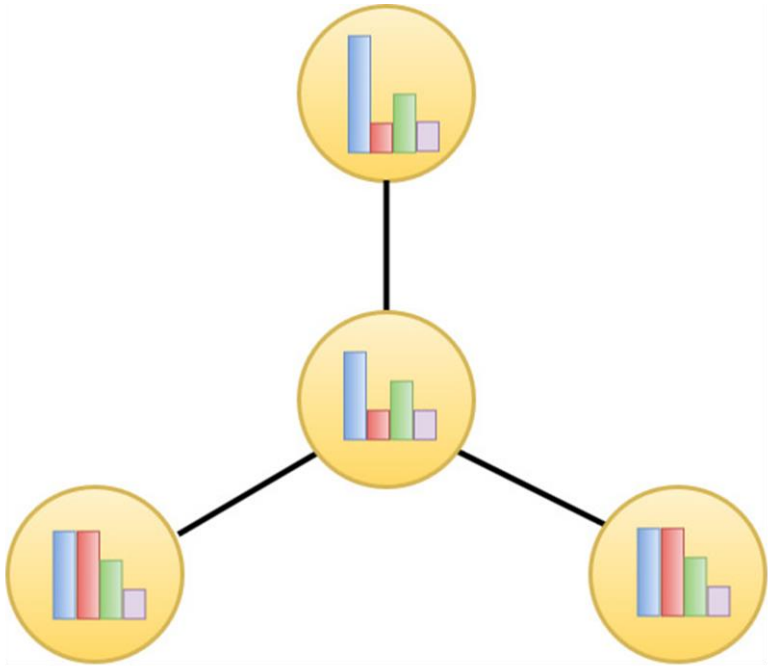
Wu, Zhenqin, et al.
"MoleculeNet: a benchmark for molecular machine learning."
Chemical science 9.2 (2018): 513-530.

KAIST

Molecular representation

Molecular graphs





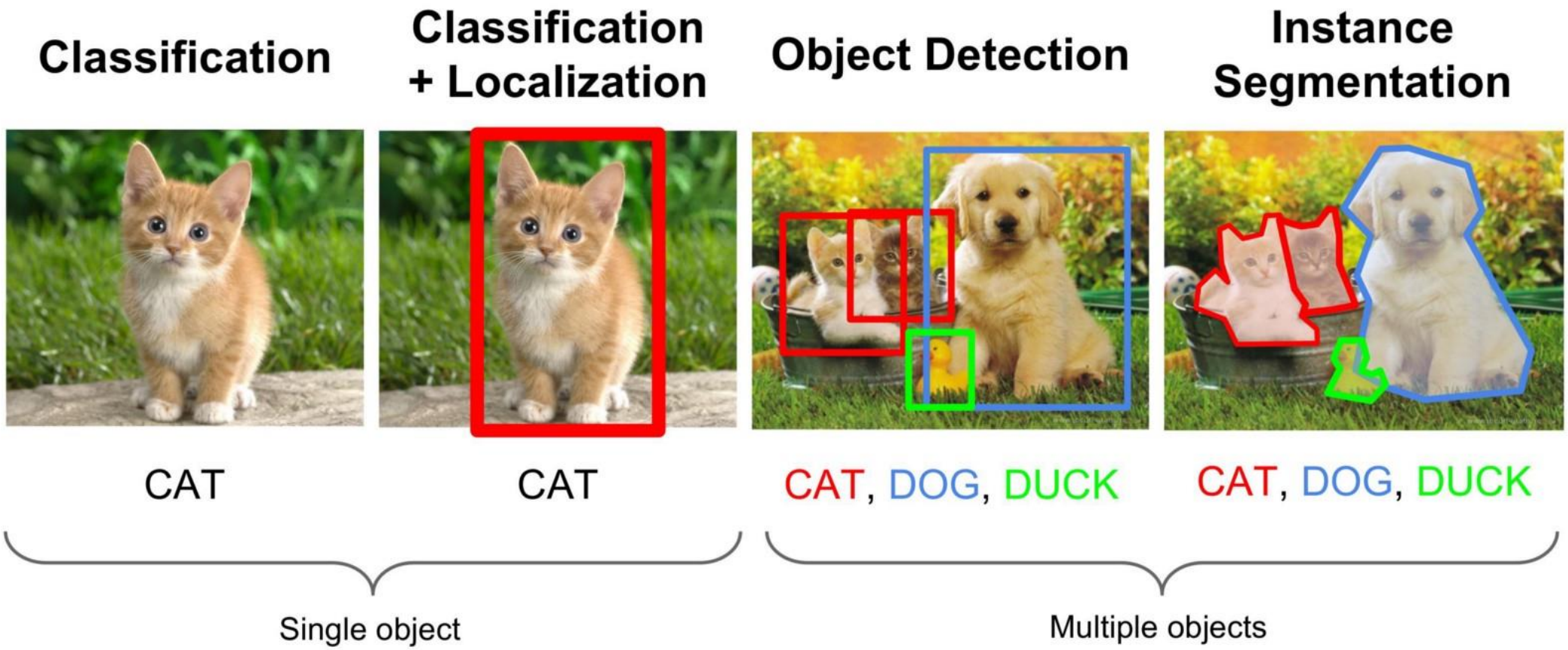
$$X_1 = \begin{bmatrix} 6 \\ 3 \\ 4 \\ 0 \end{bmatrix} \cdots X_4 = \begin{bmatrix} 8 \\ 1 \\ 4 \\ 0 \end{bmatrix} \begin{matrix} \text{Atom type} \\ \text{\# of Hs.} \\ \text{\# Valence} \\ \text{Aromaticity} \end{matrix} \quad A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

KAIST

Introduction to GNN

Limitations of CNN

Even though CNNs have shown great successes in various field, particularly vision recognition,



https://medium.com/comet-app/review-of-deep-learning-algorithms-for-object-detection-c1f3d437b852

Limitations of CNN

Even though CNNs have shown great successes in various field, particularly vision recognition,

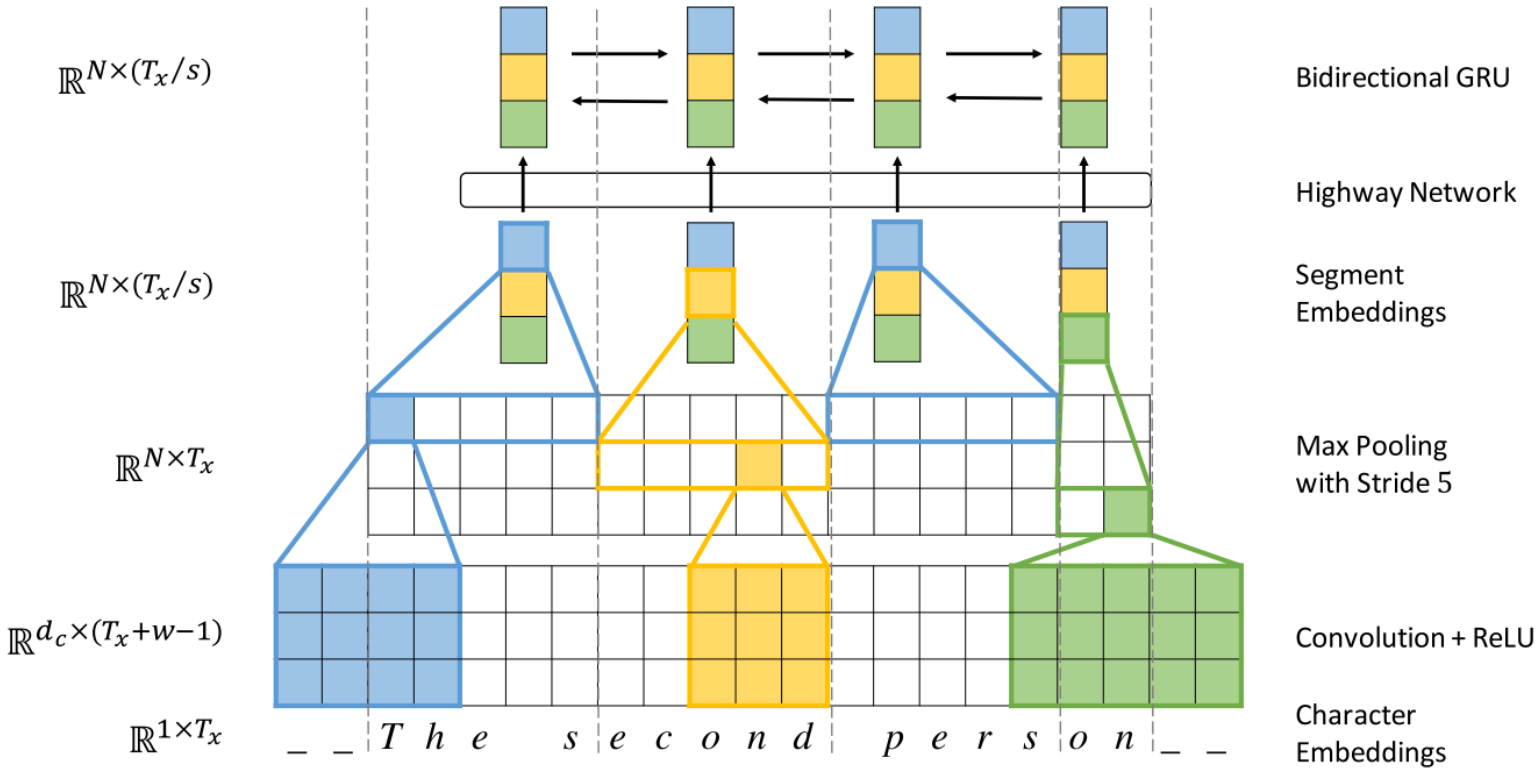


Figure 1: Encoder architecture schematics. Underscore denotes padding. A dotted vertical line delimits each segment.

Lee, Jason, Kyunghyun Cho, and Thomas Hofmann. "Fully character-level neural machine translation without explicit segmentation." *arXiv preprint arXiv:1610.03017* (2016).

Limitations of CNN

We can apply CNNs only for regularized data structure, for example…

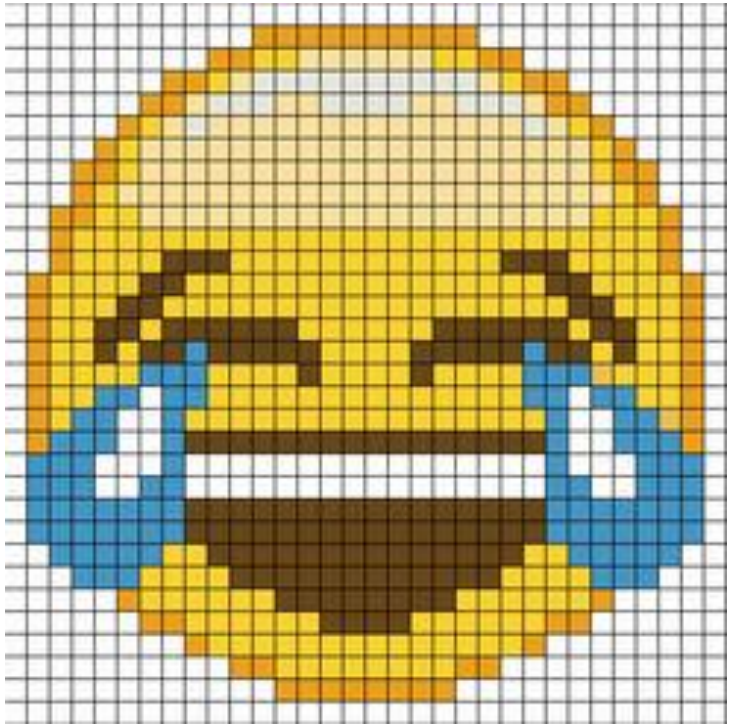
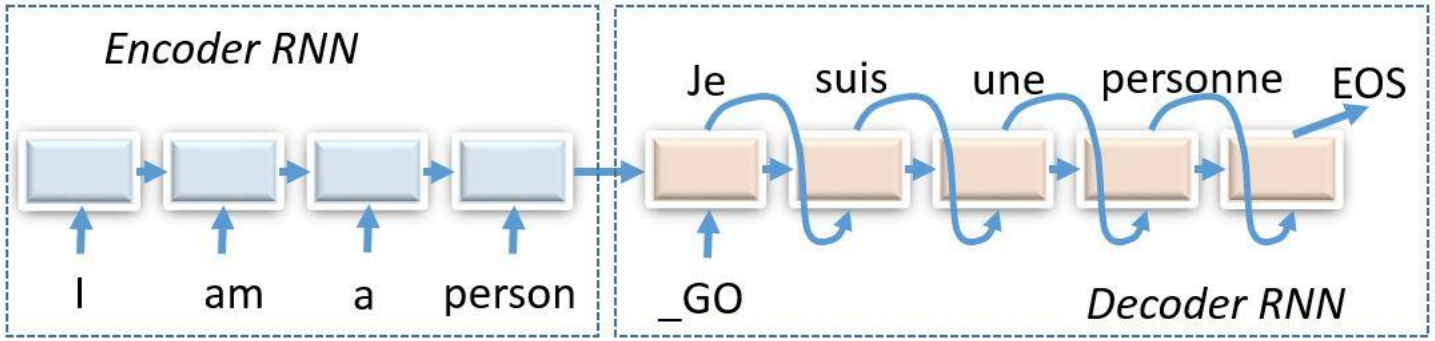


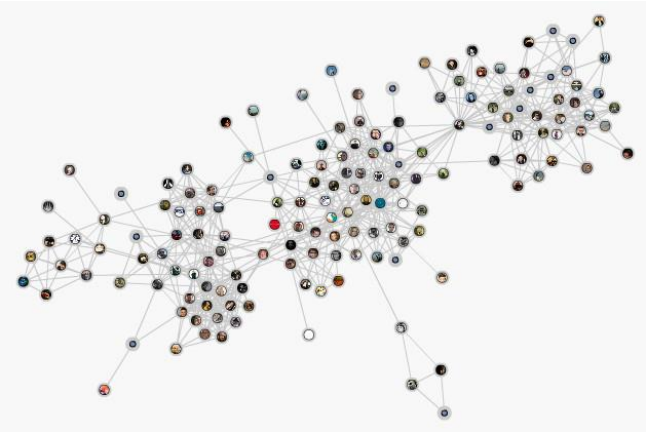
Image – values on pixels (grids)



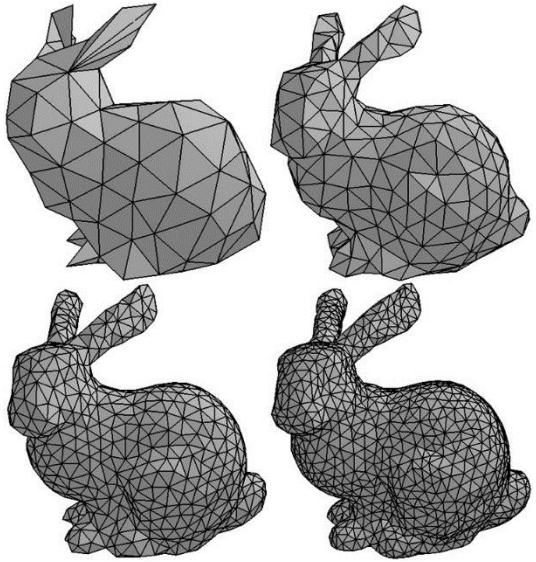
Sentence – sequential structure

Representations in graph

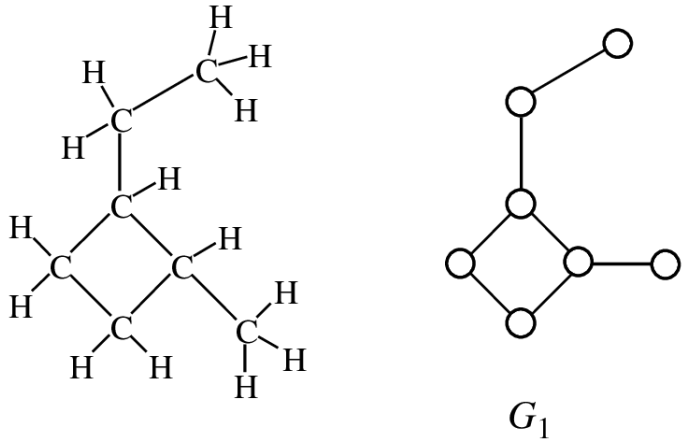
However, there are lots of irregular data structures



Social Graph
(Facebook, Wikipedia)



3D Mesh



Molecular Graph

All you need is GRAPH!

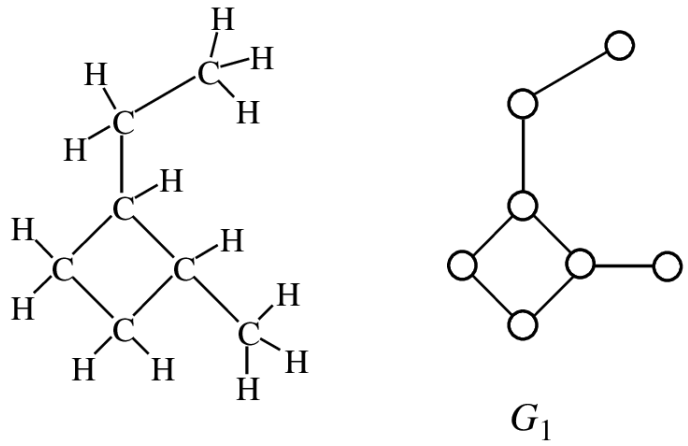
Representations in graph

Graph structure represents two information – **node features** and connectivity between nodes.

$$\boldsymbol{Graph = G(X, A)}$$

$\boldsymbol{X}$: Node, Vertex

- Individual person in a social network
- Atoms in a molecule



Represent elements of a system

Representations in graph

Donald Trump

- 72 years old
- Male
- American
- President, business man
- …

Ariana Grande

- 25 years old
- Female
- American
- Singer
- …

Vladimir Putin

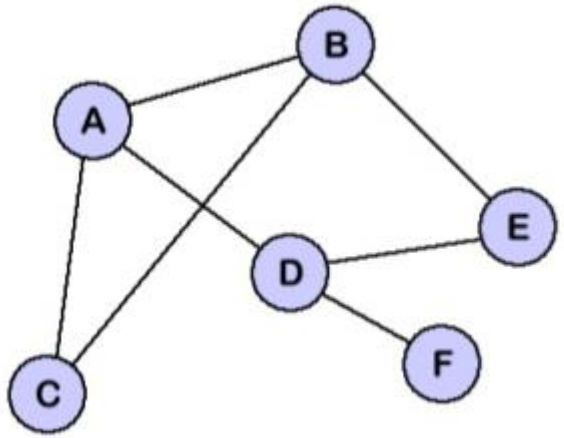
- 65 years old
- Male
- Russian
- President
- …

KAIST

Representations in graph

Graph structure represents two information – node features and **connectivity between nodes**.

$$\boldsymbol{Graph = G(X, A)}$$



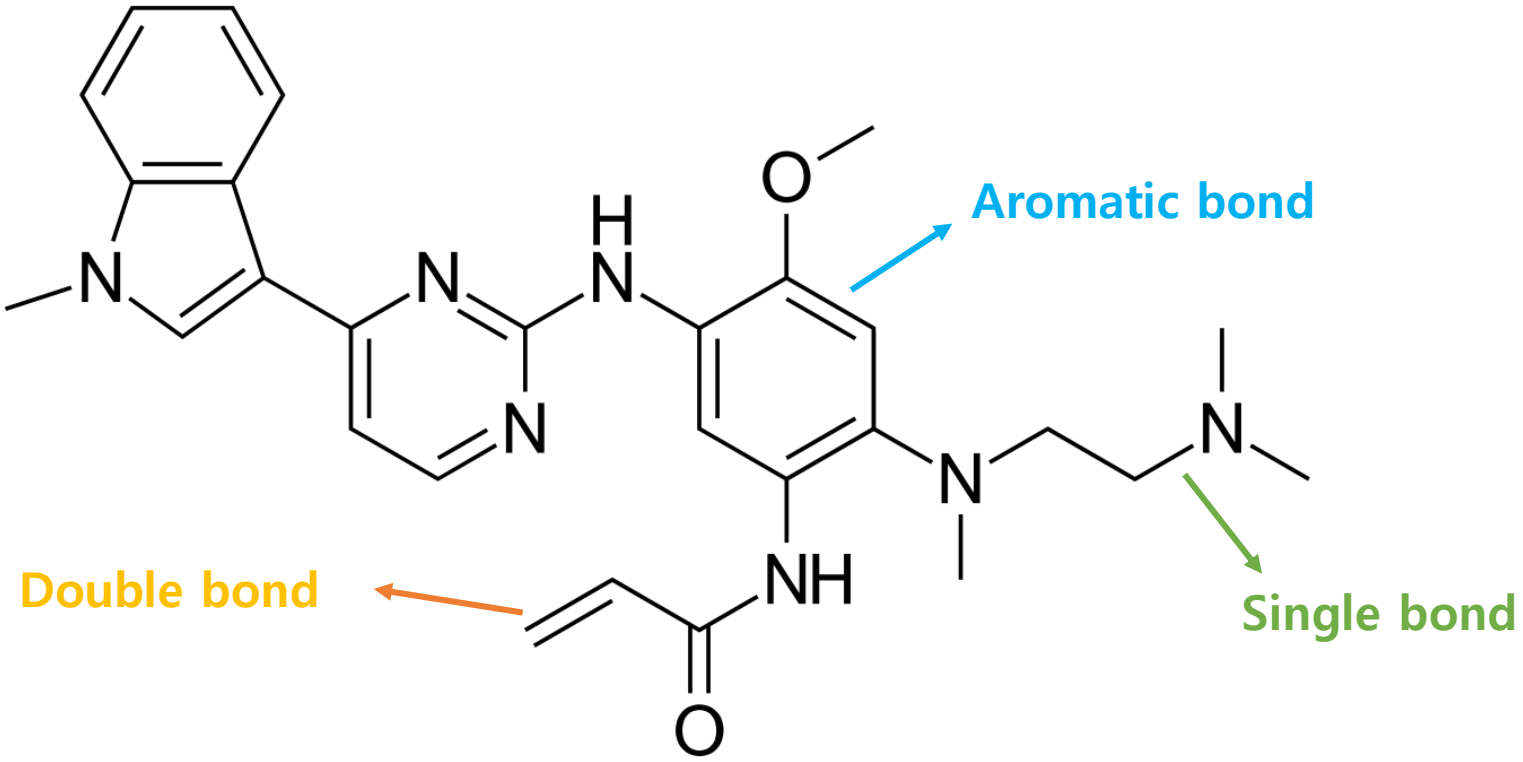
$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$\boldsymbol{A}$: Adjacency matrix

- Edges of a graph
- Connectivity, Relationship

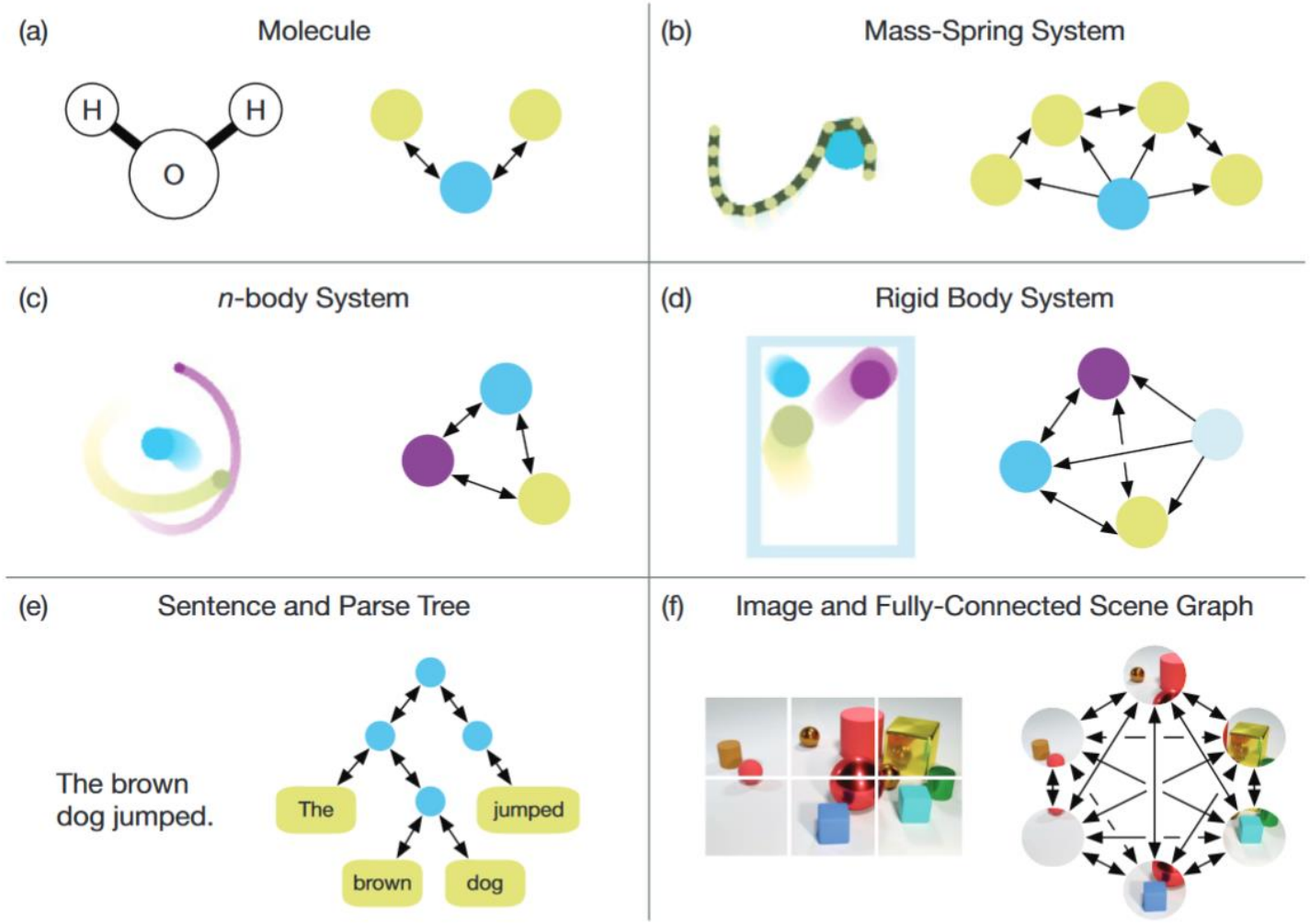
Represent relationship or interaction between elements of the system

Representations in graph



Representations in graph

GNN can facilitate many problems including chemistry, physics and relational data structure to be solved.

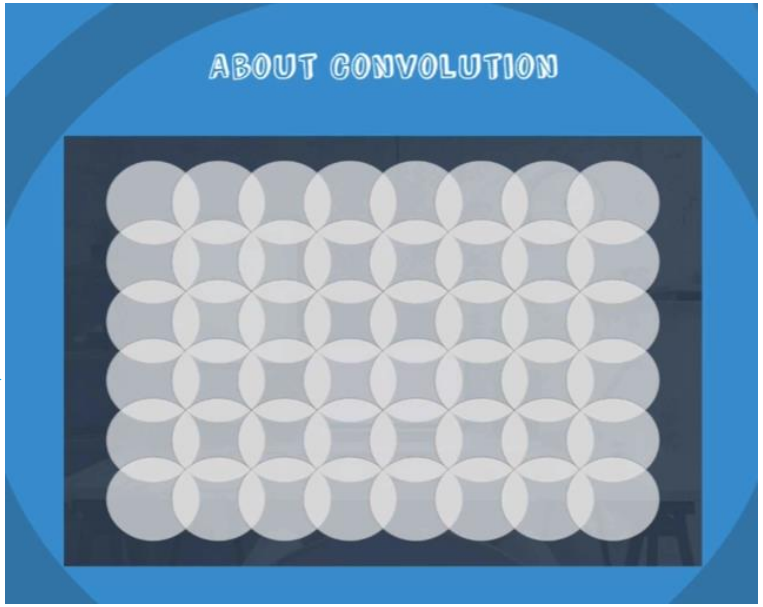


Battaglia, Peter W., et al. "Relational inductive biases, deep learning, and graph networks." *arXiv preprint arXiv:1806.01261* (2018).

Weight sharing

Weight sharing in neural networks

Weight sharing with a **receptive field** and **convolving** it over inputs

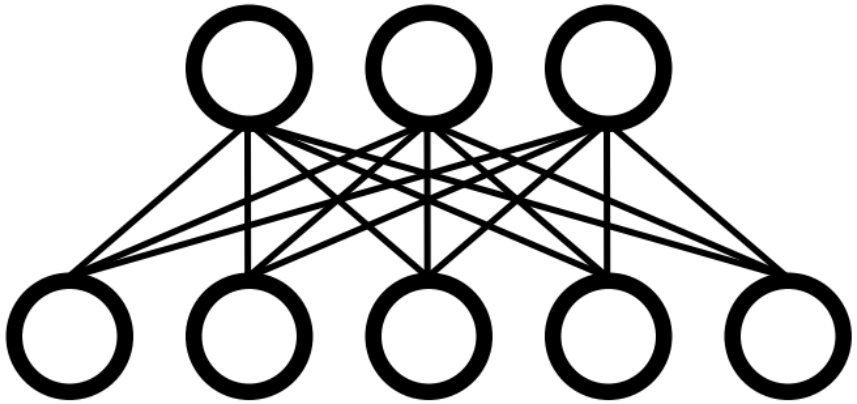


- **Each receptive field (filter) shines on specific area of picture and convolves over inputs**

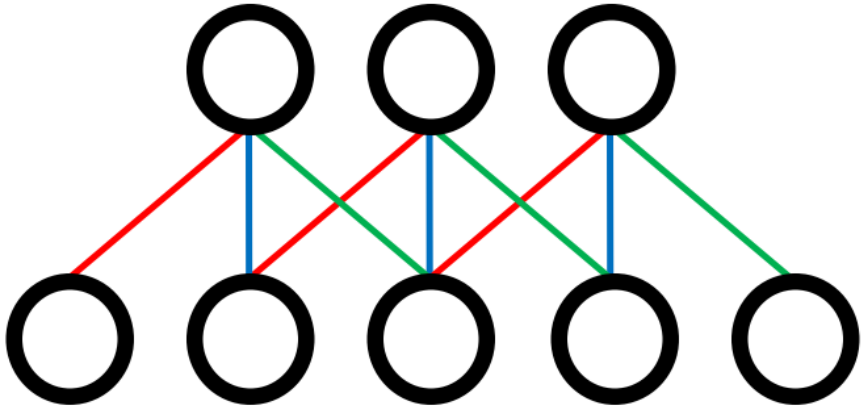
https://www.youtube.com/watch?v=JiN9p5vWHDY&index=8&list=PLjJh1vlSEYgvGod9wWiydumYl8hOXixNu

Weight sharing in neural networks

Weight sharing with a **receptive field** and **convolving** it over inputs



DNN (fully connected)

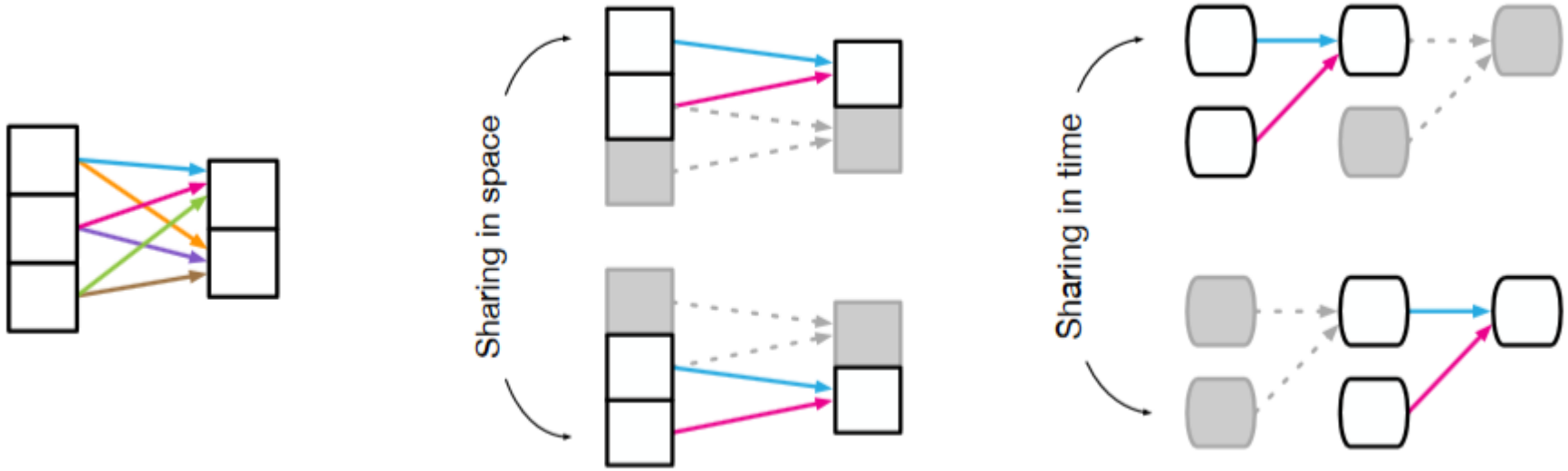


CNN (weight sharing and convolving)

- Reduce the number of parameters (less overfitting)
- Learn local features
- Translation invariance

KAIST

Weight sharing in neural networks



(a) Fully connected (b) Convolutional (c) Recurrent

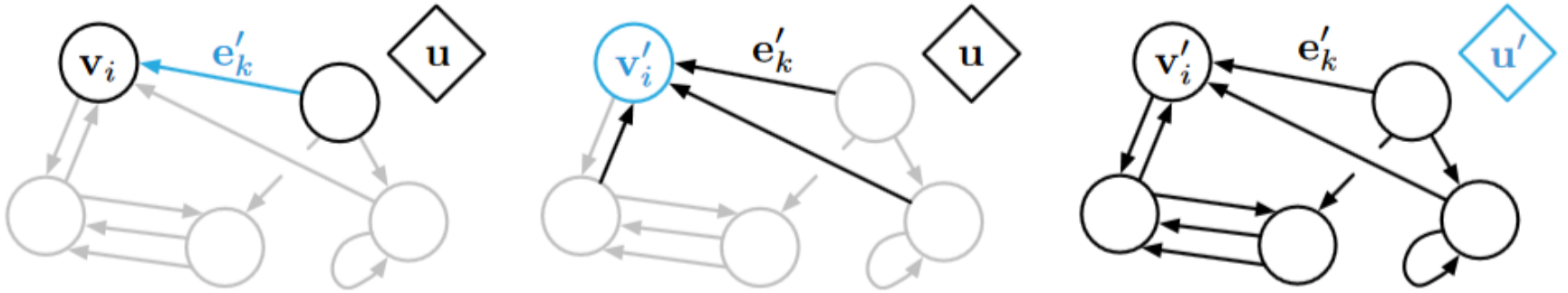
Designing architectures properly to **share weight parameters** enables us to process data with **low computational cost**.

→ Capturing key features of data

→ Avoiding over-fitting

Battaglia, Peter W., et al. "Relational inductive biases, deep learning, and graph networks." *arXiv preprint arXiv:1806.01261* (2018).

Weight parameters in GNN



(a) Edge update (b) Node update (c) Global update

Figure 3: Updates in a GN block. Blue indicates the element that is being updated, and black indicates other elements which are involved in the update (note that the pre-update value of the blue element is also used in the update). See Equation 1 for details on the notation.

- **Edge update** : relationship or interactions, sometimes called as ‘message passing’

  ex) the forces of spring

- **Node update** : aggregates the edge updates and used in the node update

  ex) the forces acting on the ball

- **Global update** : an update for the global attribute

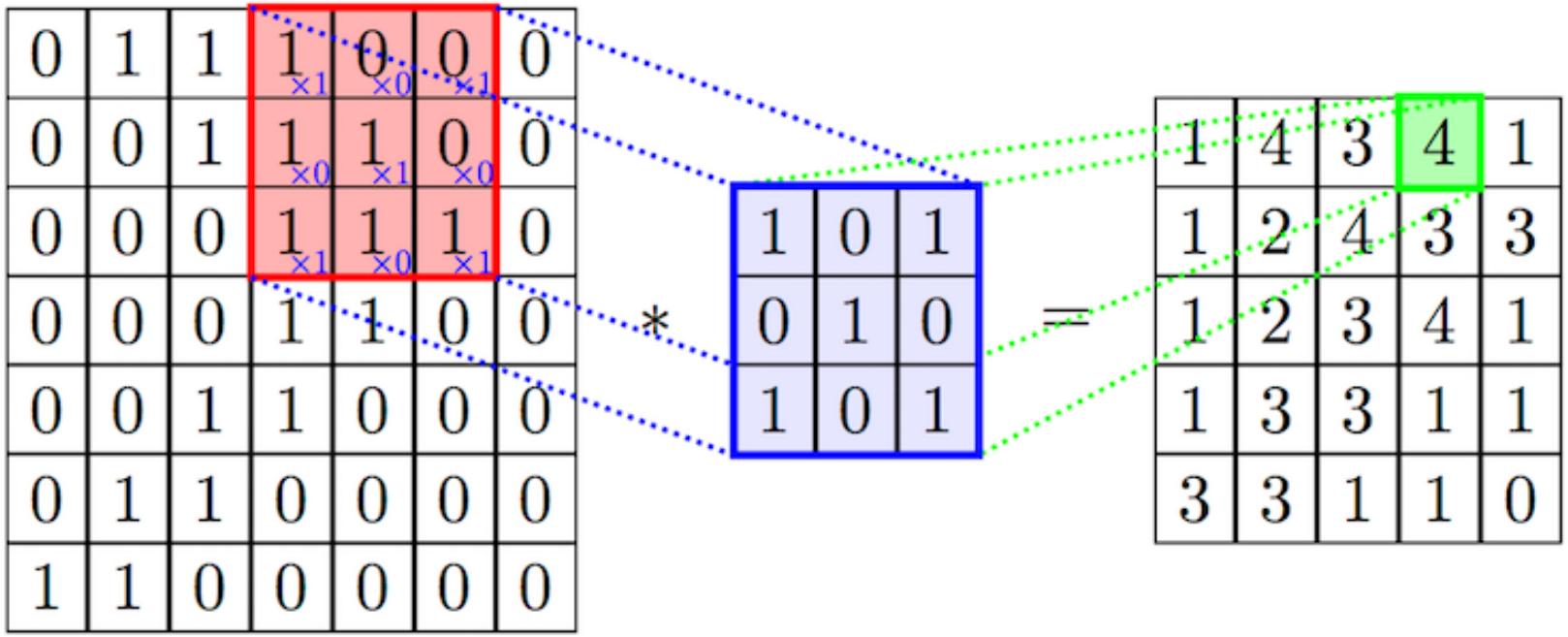
  ex) the net forces and total energy of the physical system

KAIST

Battaglia, Peter W., et al. "Relational inductive biases, deep learning, and graph networks." *arXiv preprint arXiv:1806.01261* (2018).

Graph Convolution Network (GCN)

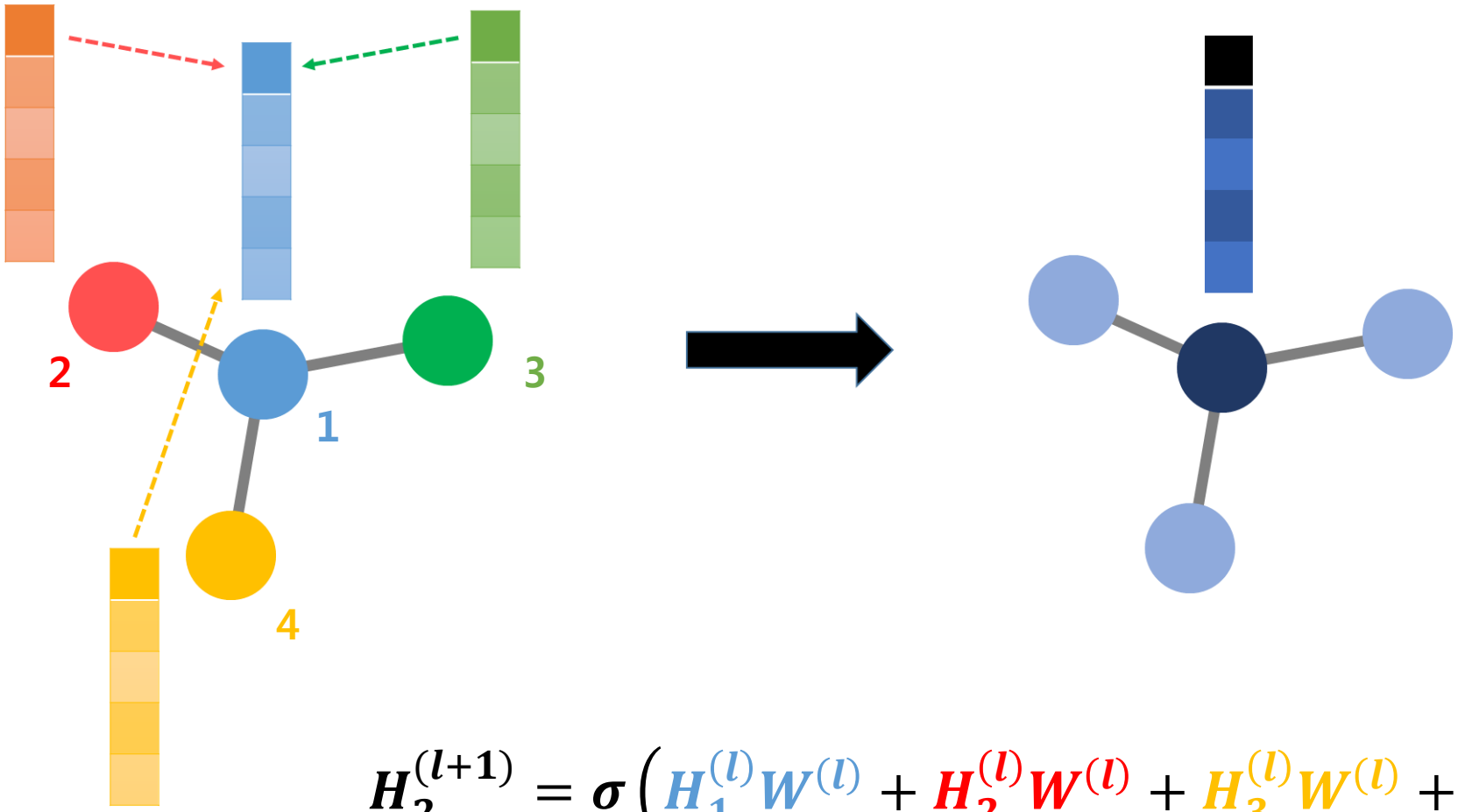
Update hidden states in CNN



$$X_i^{(l+1)} = \sigma\left(\sum_{j\in[i-k,i+k]} W_j^{(l)} X_j^{(l)} + b^{(l)}\right)$$

learnable parameters are shared

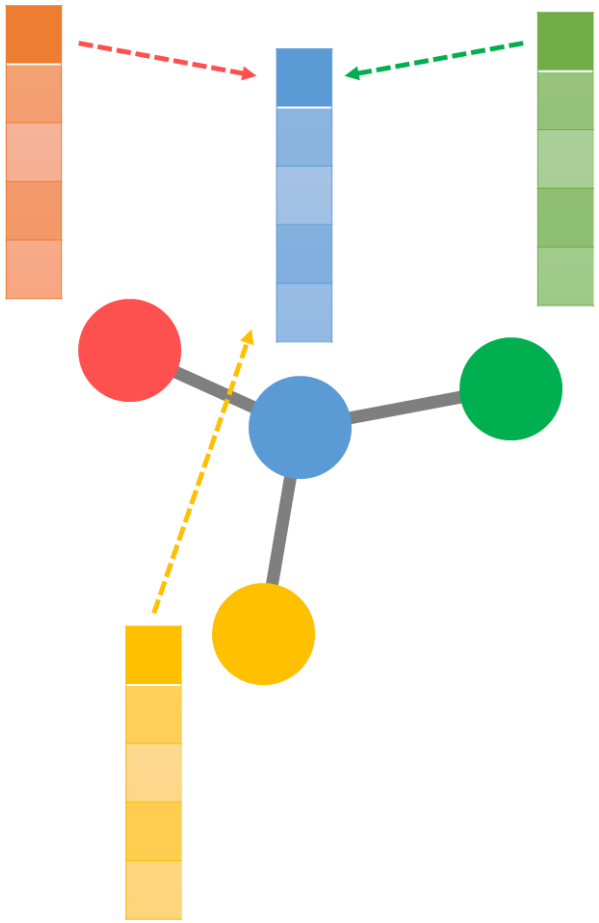
Update hidden states in GCN

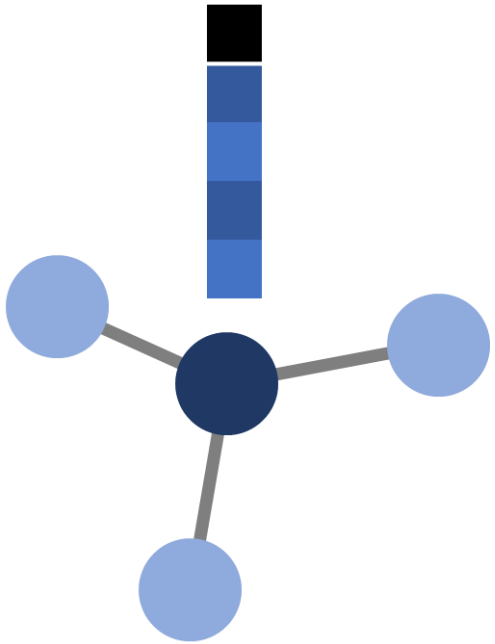


$$H_2^{(l+1)} = \sigma\left(H_1^{(l)}W^{(l)} + H_2^{(l)}W^{(l)} + H_3^{(l)}W^{(l)} + H_4^{(l)}W^{(l)} + b^{(l)}\right)$$

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

Update hidden states in GCN





$$H^{(l+1)} = \sigma\left(AH^{(l)}W^{(l)} + b^{(l)}\right)$$

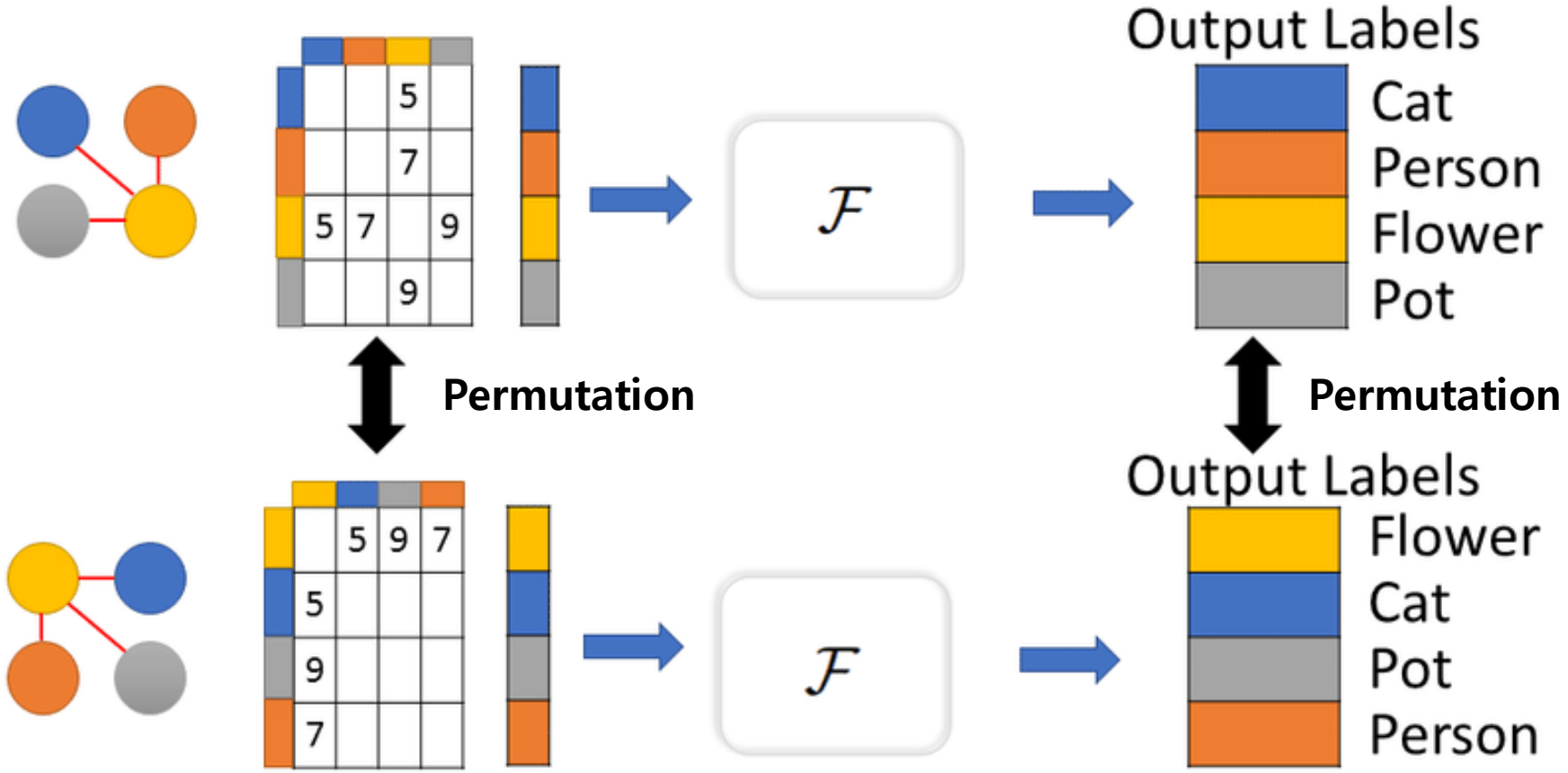
learnable parameters are shared

Sharing weights for all nodes in graph,
but nodes are differently updated by reflecting individual node features, $H_j^{(l)}$

KAIST

Readout

Readout makes graph features a permutation invariance



Mapping Images to Scene Graphs with Permutation-Invariant Structured Prediction - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/Graph-permutation-invariance-and-structured-prediction-A-graph-labeling-function-F-is_fig1_323217335 [accessed 8 Sep, 2018]

Readout

Readout makes graph features a permutation invariance

- **Graph feature**

$$z_G = f\left(\left\{H_i^{(L)}\right\}\right)$$

1) Node-wise summation

$$z_G = \tau\left(\sum_{i \in G} MLP\left(H_i^{(L)}\right)\right)$$

Summation over all existing nodes in the graph satisfies the permutation invariance

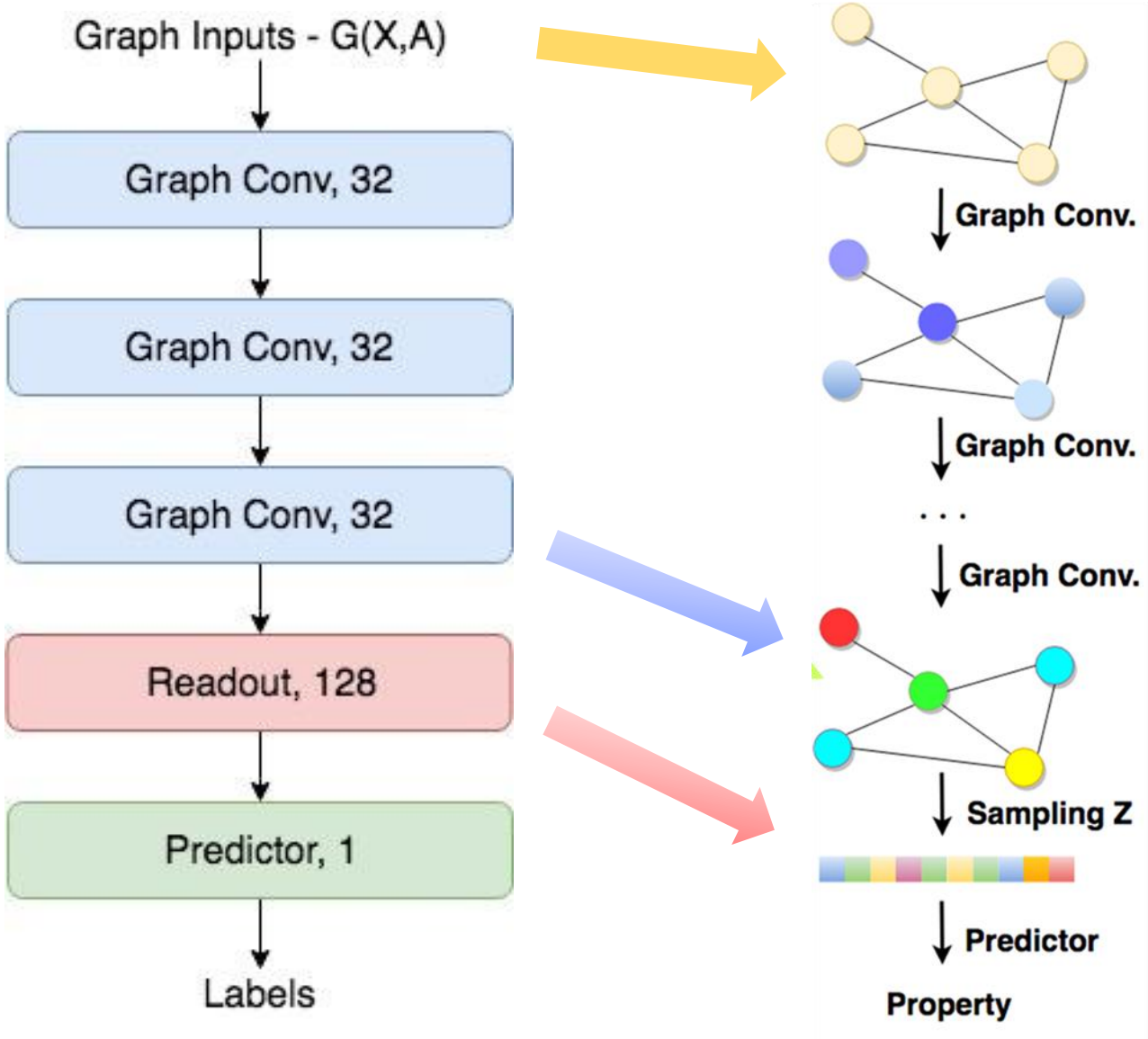
1) Graph gathering

$$z_G = \tau\left(\sum_{i \in G} \sigma\left(MLP_1\left(H_i^{(L)}, H_i^{(0)}\right)\right) \odot MLP_2\left(H_i^{(L)}\right)\right)$$

- τ : ReLU activation
- σ : sigmoid activation

Gilmer, Justin, et al. "Neural message passing for quantum chemistry." *arXiv preprint arXiv:1704.01212* (2017).

Overall architecture of GCN



Input node features, $\left\{H_i^{(0)}\right\}$

Raw node information

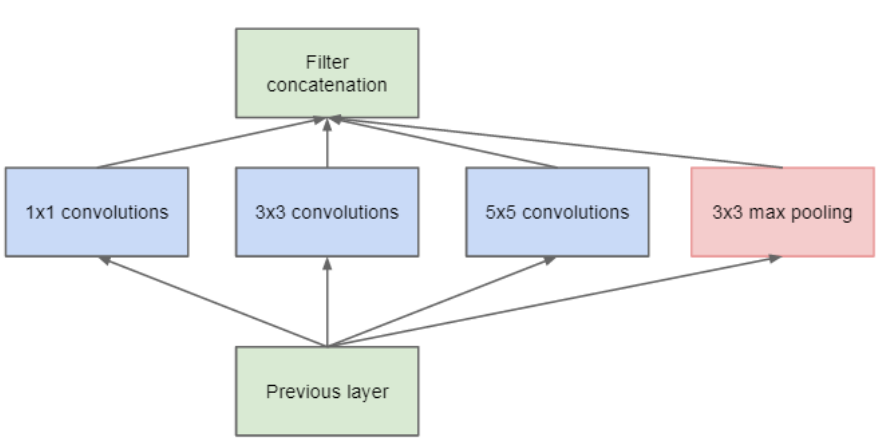
Final node states, $\left\{H_i^{(L)}\right\}$

Graph features, Z

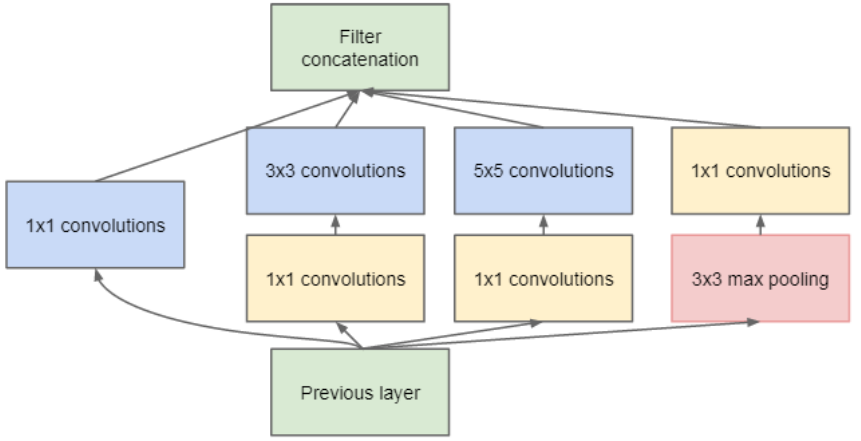
Technical issues

Going wider: inception
Going deeper: skip connection
Going smarter : attention

Inception



(a) Inception module, naïve version



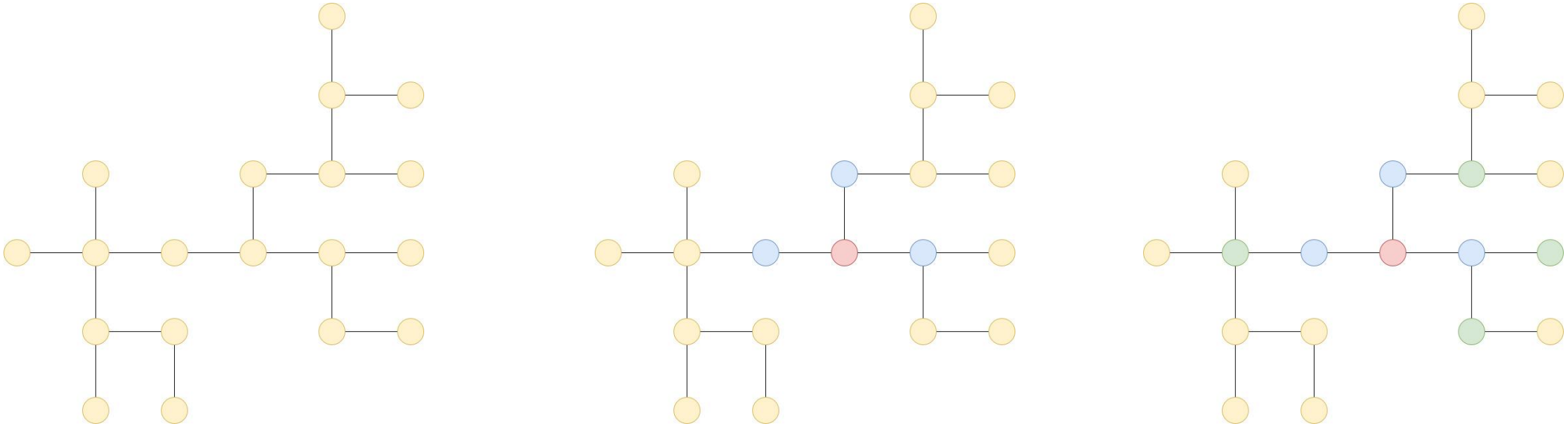
(b) Inception module with dimension reductions

Stack different layer transformations in parallel, resulting in nets that were **simultaneously deep** (many layers) and **wide** (many parallel operations).

https://towardsdatascience.com/an-intuitive-guide-to-deep-network-architectures-65fdc477db41

Inception

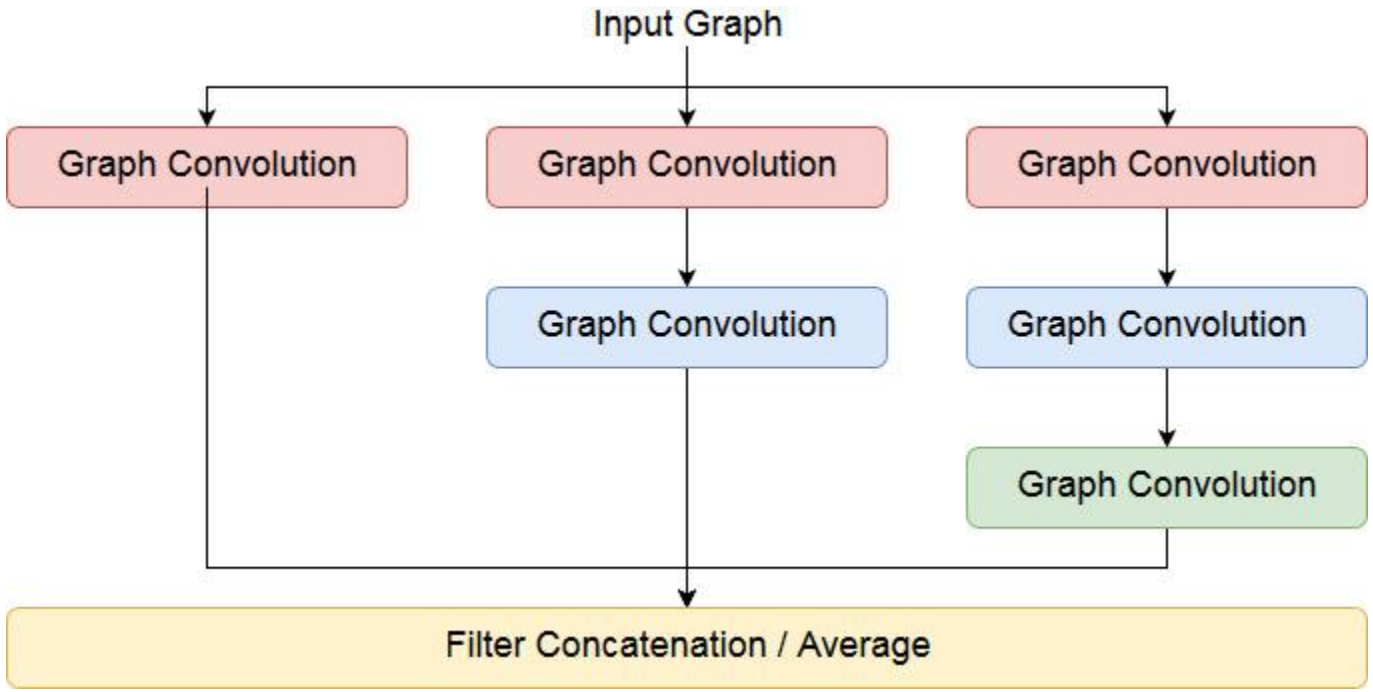
Single graph convolution reflects the first nearest neighbor information and subsequent **multiple graph convolution can deliver information of distant atoms.**

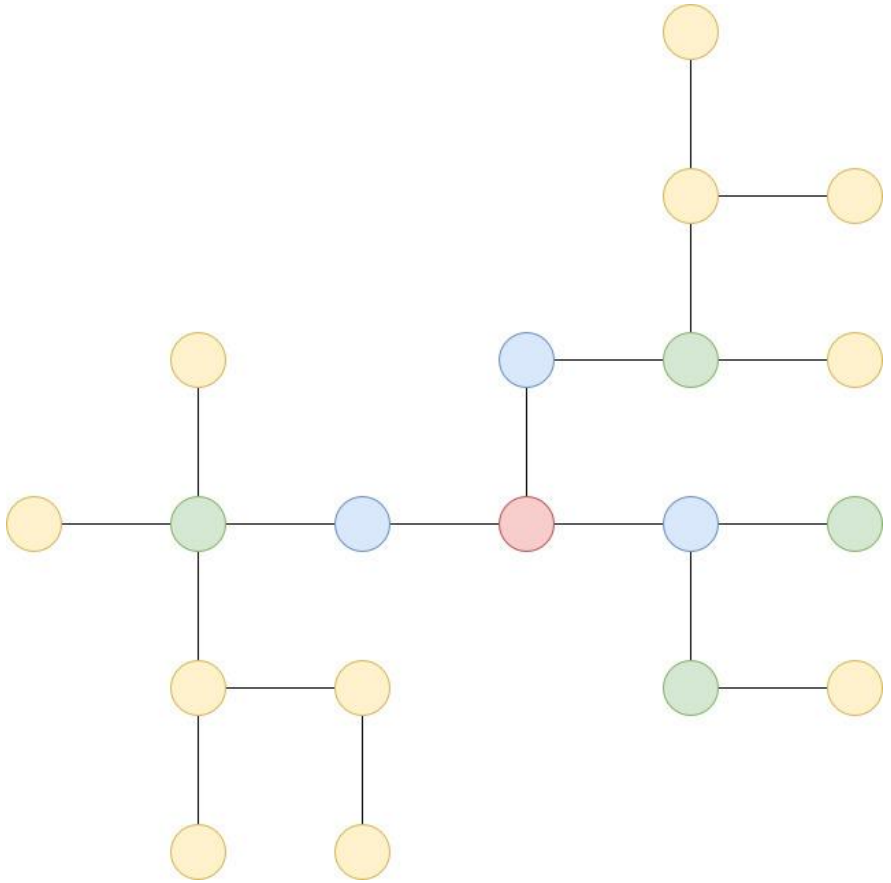


$$H^{(1)} = \sigma\left(AH^{(0)}W^{(0)}\right) \qquad H^{(2)} = \sigma\left(AH^{(1)}W^{(1)}\right)$$

Inception

Inception module in GCN

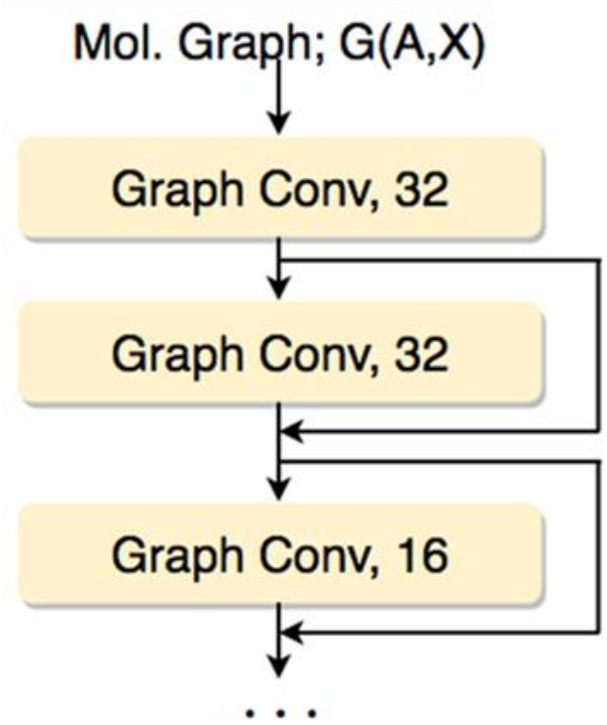




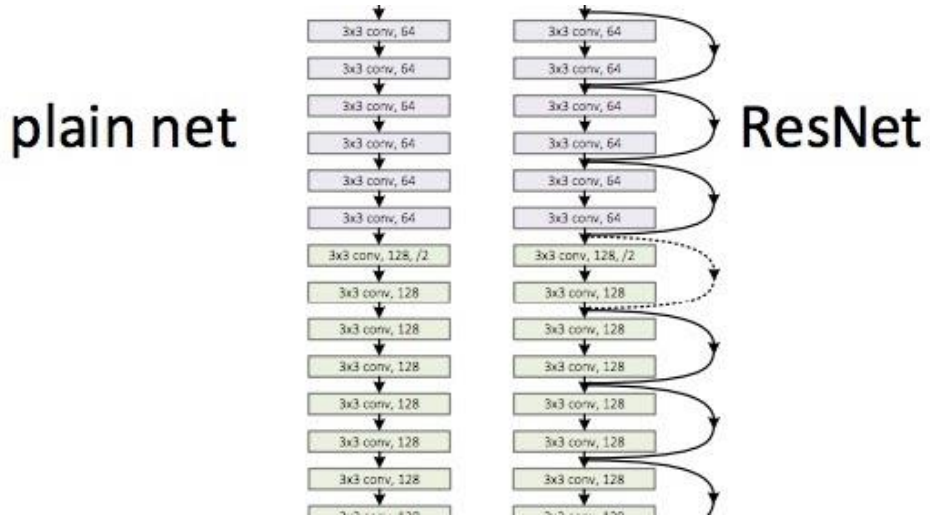
- Make network **wider**
- Avoid **vanishing gradient**

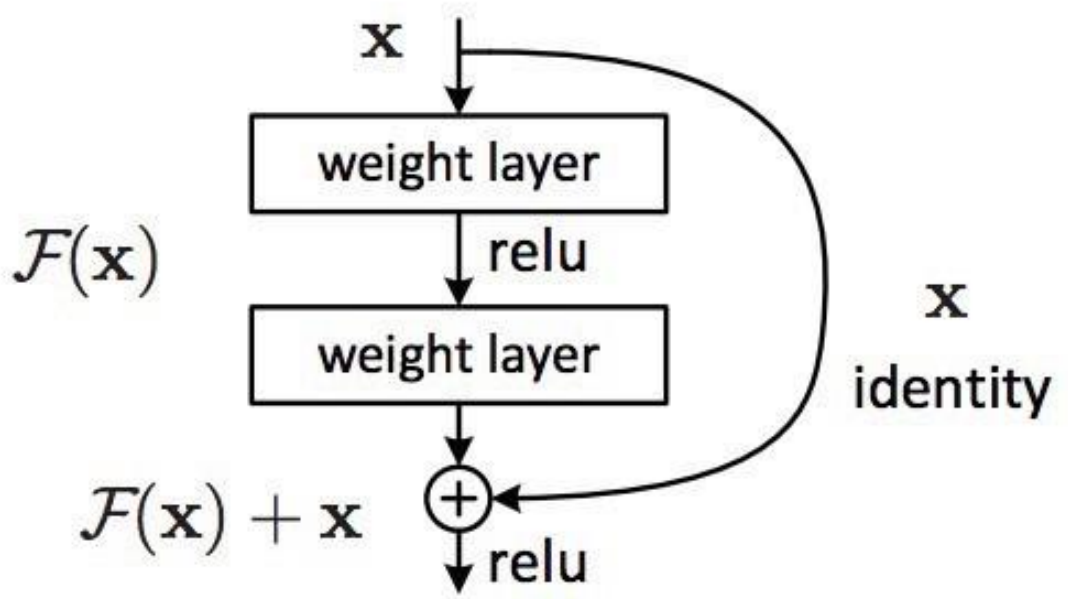
Skip connection

ResNet – Winner of 2015 ImageNet Challenge



$$y = H_i^{(l+1)} + H_i^{(l)}$$

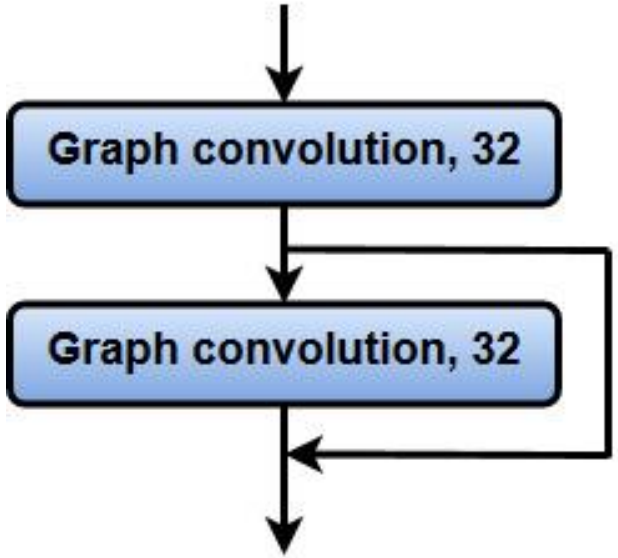




https://towardsdatascience.com/an-intuitive-guide-to-deep-network-architectures-65fdc477db41

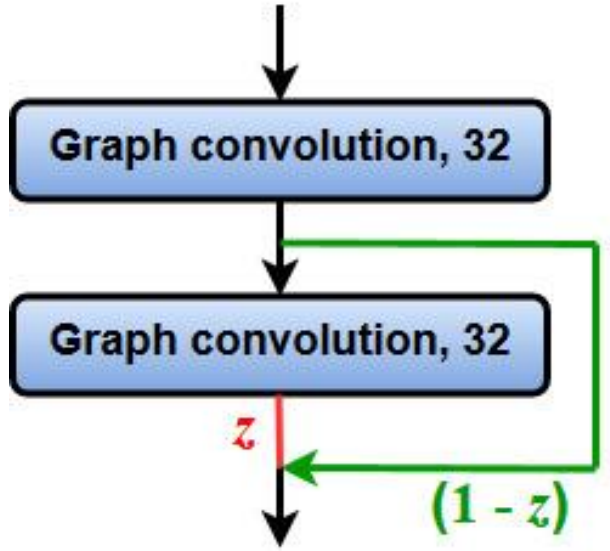
Gated skip connection

However, naïve skip-connection unintentionally mix the information.



$$H_{i,sc}^{(l+1)} = H_i^{(l+1)} + H_i^{(l)}$$

Instead, one may use a gated-skip connection, which mixes the information with appropriate ratio, z.



$$H_{i,gsc}^{(l+1)} = z_i \odot H_i^{(l+1)} + (1 - z_i) \odot H_i^{(l)}$$

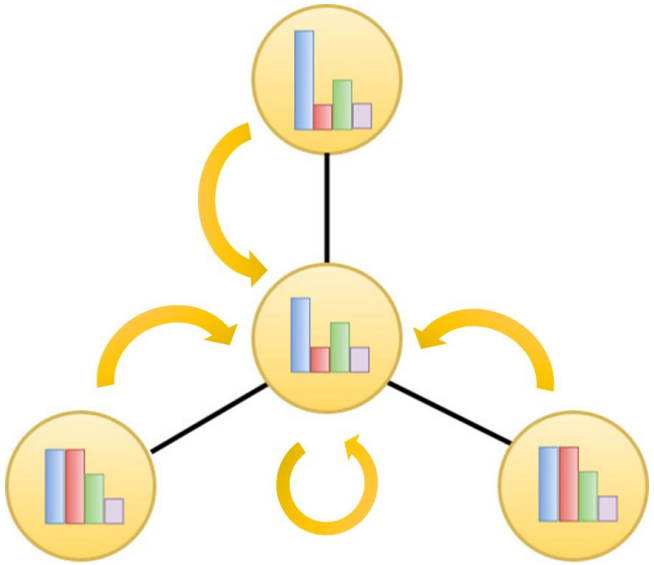
$$z_i = \sigma\left(U_{z,1} H_i^{(l+1)} + U_{z,2} H_i^{(l)} + b_z\right)$$

Ryu, Seongok, Jaechang Lim, Seung Hwan Hong and Woo Youn Kim. "Deeply learning molecular structure-property relationships using attention- and gate-augmented graph convolutional network." *arXiv preprint arXiv:1805.10988* (2018).

KAIST

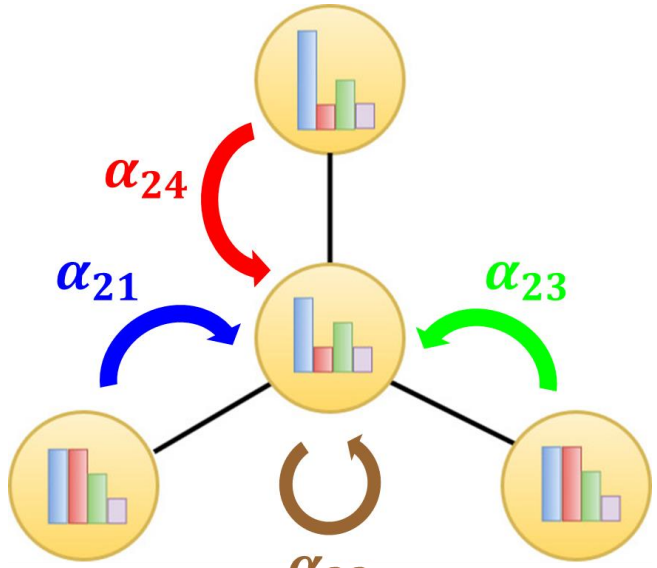
Attention

Vanilla GCN updates information of neighbor atoms **with same importance**.



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

Attention mechanism enables it to update nodes **with different importance**



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

Ryu, Seongok, Jaechang Lim, Seung Hwan Hong and Woo Youn Kim. "Deeply learning molecular structure-property relationships using attention- and gate-augmented graph convolutional network." *arXiv preprint arXiv:1805.10988* (2018).

Attention

Learnable parameters : Convolution weight and attention coefficient

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

- Velickovic, Petar, et al. – network analysis

$$\alpha_{ij} = softmax(e_{ij}) = \frac{e_{ij}}{exp\left(\sum_{k \in N(i)} e_{ik}\right)} \qquad e_{ij} = LearkyReLU(a^T[H_i W, H_j W])$$

Velickovic, Petar, et al.
"Graph attention networks." *arXiv preprint arXiv:1710.10903* (2017).

- Seongok Ryu, et al. – molecular applications

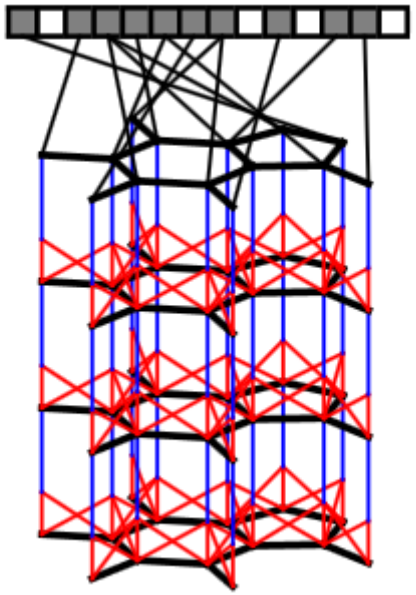
$$\alpha_{ij} = \tanh\left((H_i W)^T C(H_j W)\right)$$

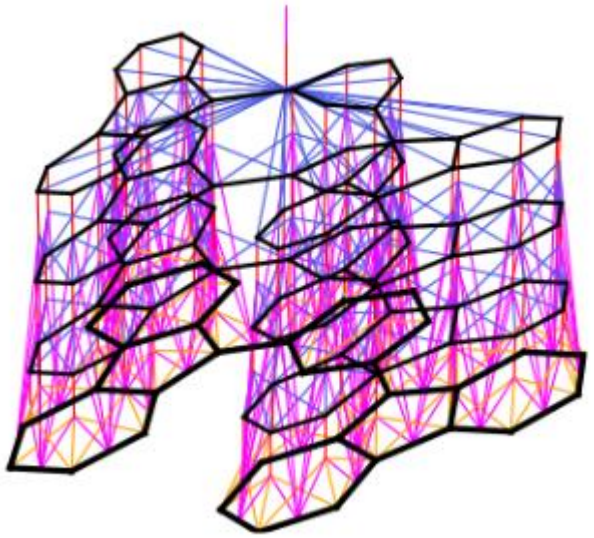
Ryu, Seongok, Jaechang Lim, Seung Hwan Hong and Woo Youn Kim.
"Deeply learning molecular structure-property relationships using attention- and gate-augmented graph convolutional network." *arXiv preprint arXiv:1805.10988* (2018).

GCN in Chemistry

Convolutional Networks on Graphs for Learning Molecular Fingerprints

David Duvenaud†, Dougal Maclaurin†, Jorge Aguilera-Iparraguirre
Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, Ryan P. Adams
Harvard University





Duvenaud, David K., et al. "Convolutional networks on graphs for learning molecular fingerprints." *Advances in neural information processing systems*. 2015.

Morgan fingerprint

Neural fingerprint

Algorithm 1 Circular fingerprints

Input: molecule, radius R, fingerprint length S
Initialize: fingerprint vector $\mathbf{f} \leftarrow \mathbf{0}_S$
for each atom a in molecule
  $\mathbf{r}_a \leftarrow g(a)$ ▷ lookup atom features
for $L = 1$ to R ▷ for each layer
  for each atom a in molecule
    $\mathbf{r}_1 \dots \mathbf{r}_N = \text{neighbors}(a)$
    $\mathbf{v} \leftarrow [\mathbf{r}_a, \mathbf{r}_1, \dots, \mathbf{r}_N]$ ▷ concatenate
    $\mathbf{r}_a \leftarrow \text{hash}(\mathbf{v})$ ▷ hash function
    $i \leftarrow \text{mod}(r_a, S)$ ▷ convert to index
    $\mathbf{f}_i \leftarrow 1$ ▷ Write 1 at index
Return: binary vector $\mathbf{f}$

Algorithm 2 Neural graph fingerprints

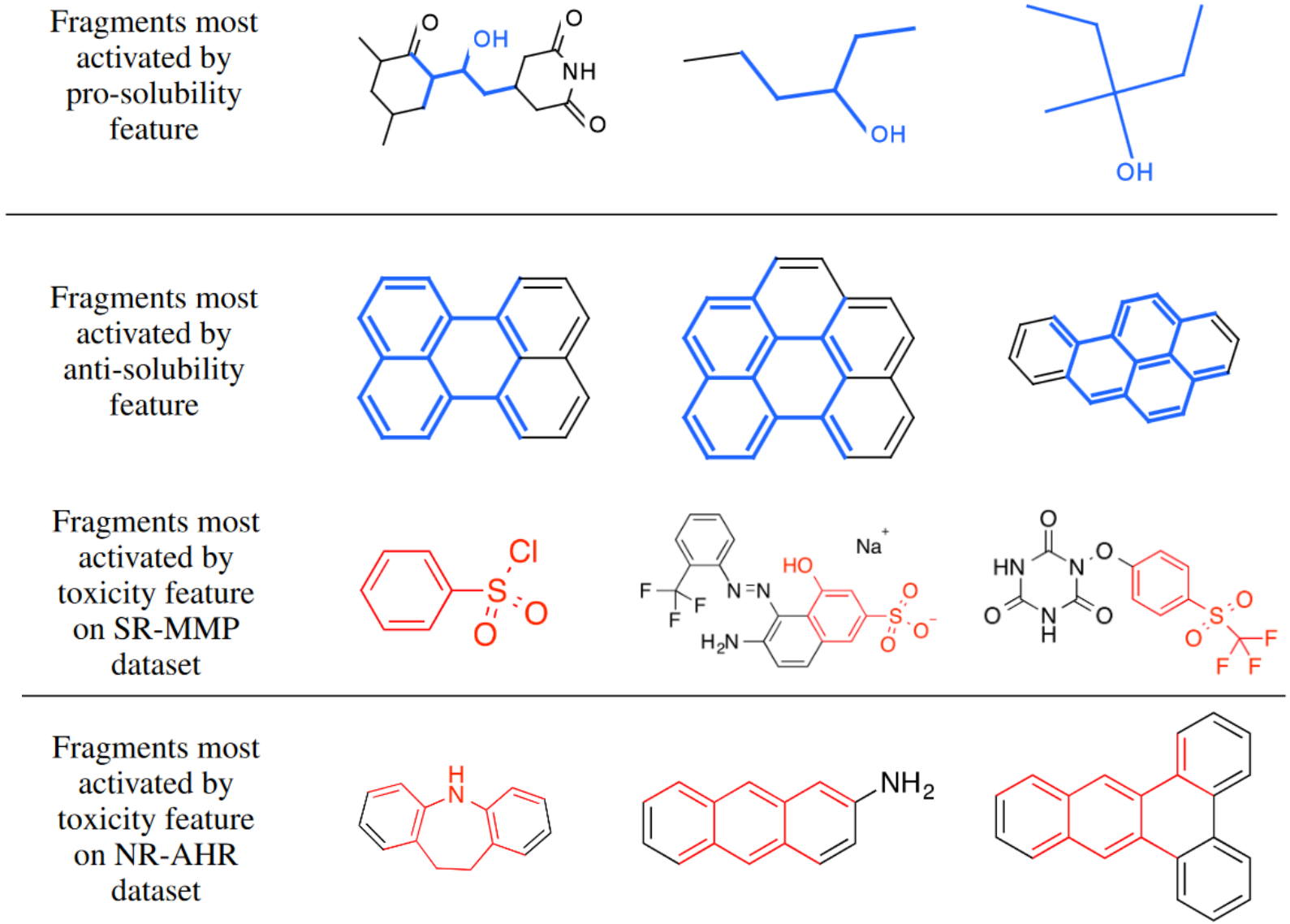
Input: molecule, radius R, hidden weights $H_1^1 \dots H_R^5$, output weights $W_1 \dots W_R$
Initialize: fingerprint vector $\mathbf{f} \leftarrow \mathbf{0}_S$
for each atom a in molecule
  $\mathbf{r}_a \leftarrow g(a)$ ▷ lookup atom features
for $L = 1$ to R ▷ for each layer
  for each atom a in molecule
    $\mathbf{r}_1 \dots \mathbf{r}_N = \text{neighbors}(a)$
    $\mathbf{v} \leftarrow \mathbf{r}_a + \sum_{i=1}^{N} \mathbf{r}_i$ ▷ sum
    $\mathbf{r}_a \leftarrow \sigma(\mathbf{v} H_L^N)$ ▷ smooth function
    $\mathbf{i} \leftarrow \text{softmax}(\mathbf{r}_a W_L)$ ▷ sparsify
    $\mathbf{f} \leftarrow \mathbf{f} + \mathbf{i}$ ▷ add to fingerprint
Return: real-valued vector $\mathbf{f}$

Figure 2: Pseudocode of circular fingerprints (*left*) and neural graph fingerprints (*right*). Differences are highlighted in blue. Every non-differentiable operation is replaced with a differentiable analog.

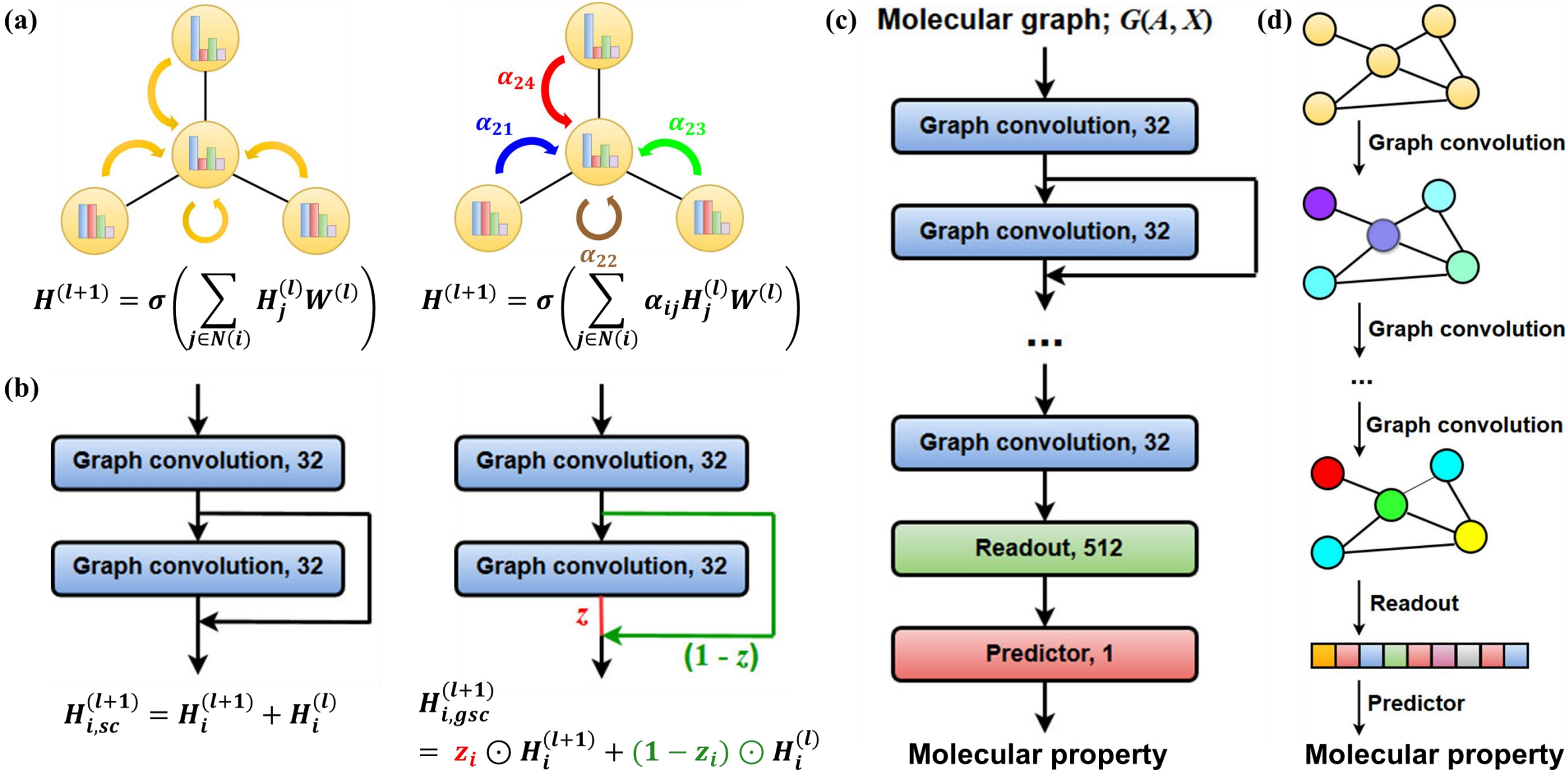
Morgan algorithm uses hash function to encode(featurize) molecular structure.
However, a neural network which encodes with proper weight and algorithms can generate better fingerprints.

Duvenaud, David K., et al. "Convolutional networks on graphs for learning molecular fingerprints." *Advances in neural information processing systems*. 2015.

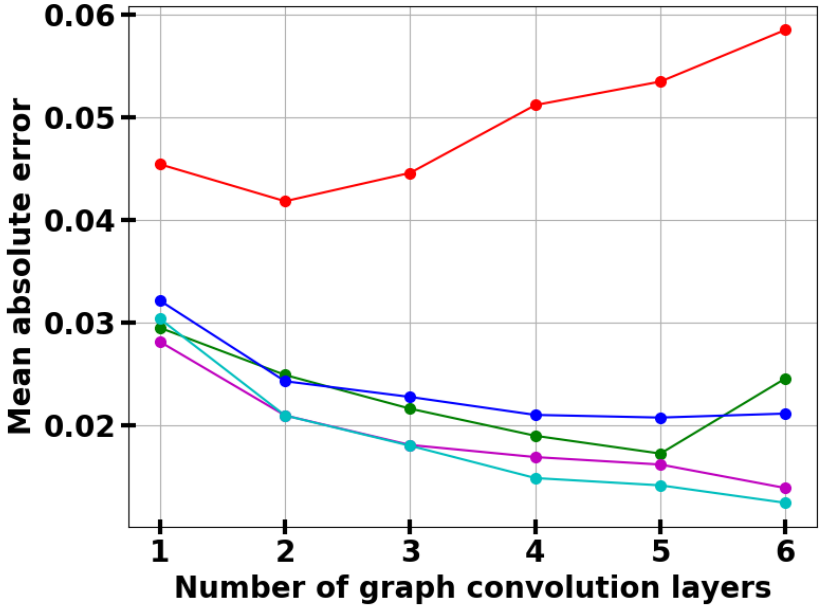
The neural network can recognize key substructures determining molecular properties by itself.
→ Quantitative structure property relationships (QSPR)

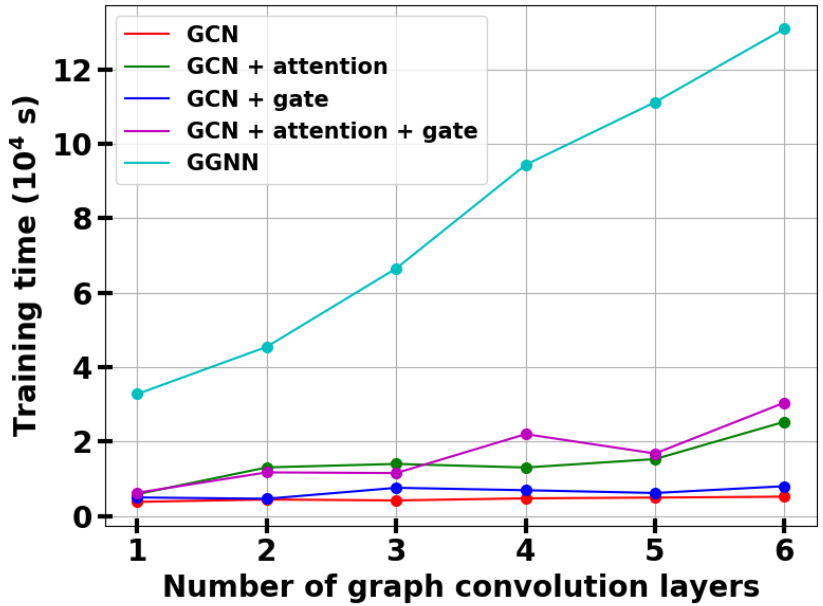


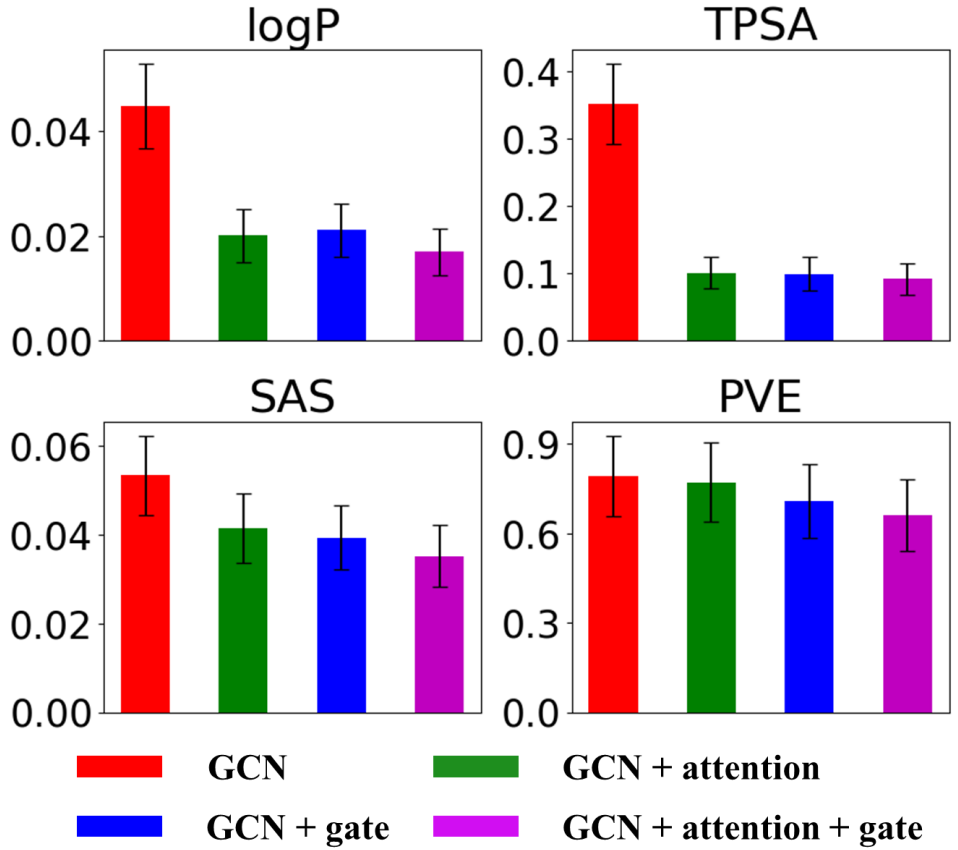
Duvenaud, David K., et al. "Convolutional networks on graphs for learning molecular fingerprints." *Advances in neural information processing systems*. 2015.



Ryu, Seongok, Jaechang Lim, Seung Hwan Hong and Woo Youn Kim.
"Deeply learning molecular structure-property relationships using attention- and gate-augmented graph convolutional network." *arXiv preprint arXiv:1805.10988* (2018).

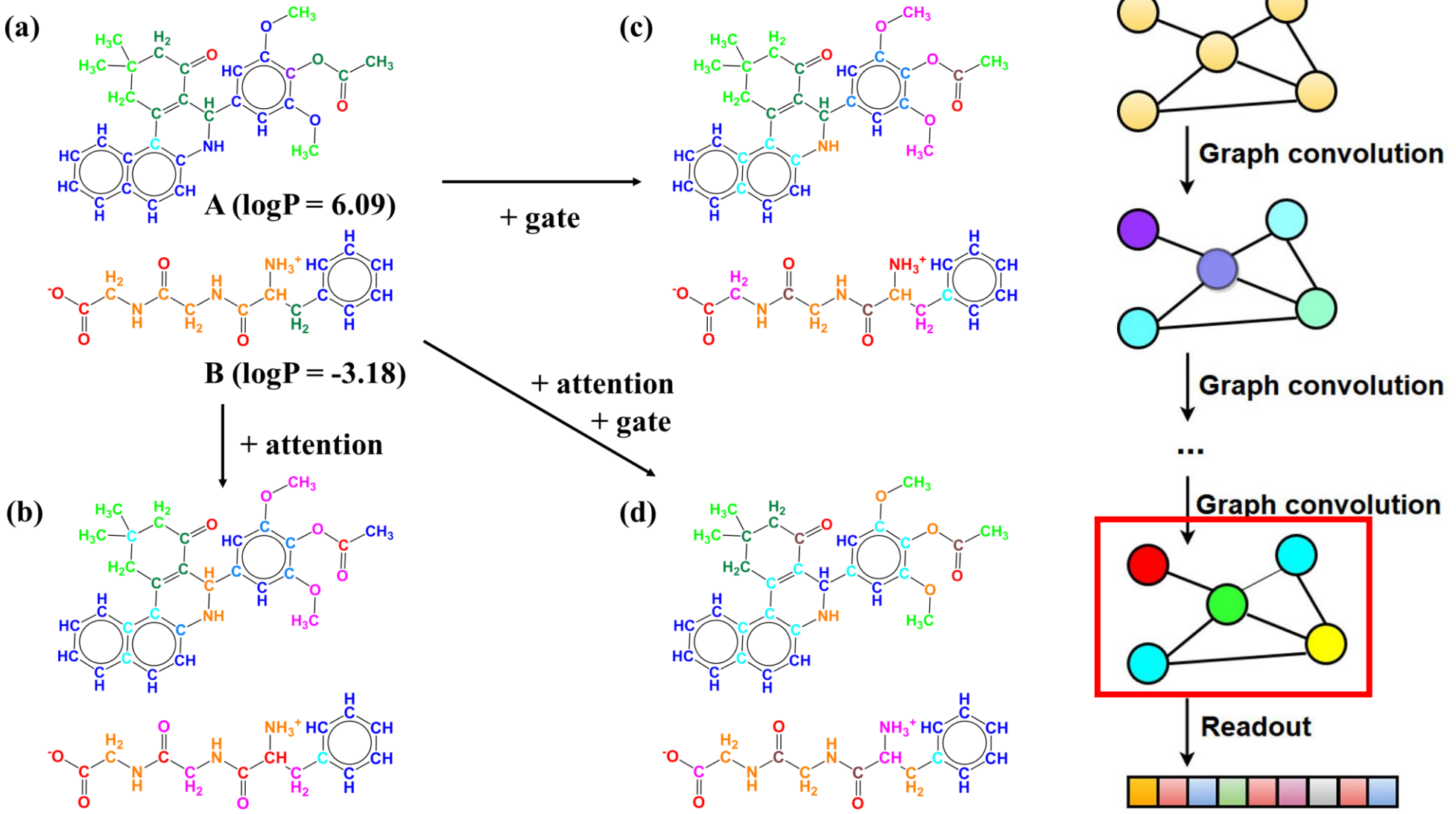






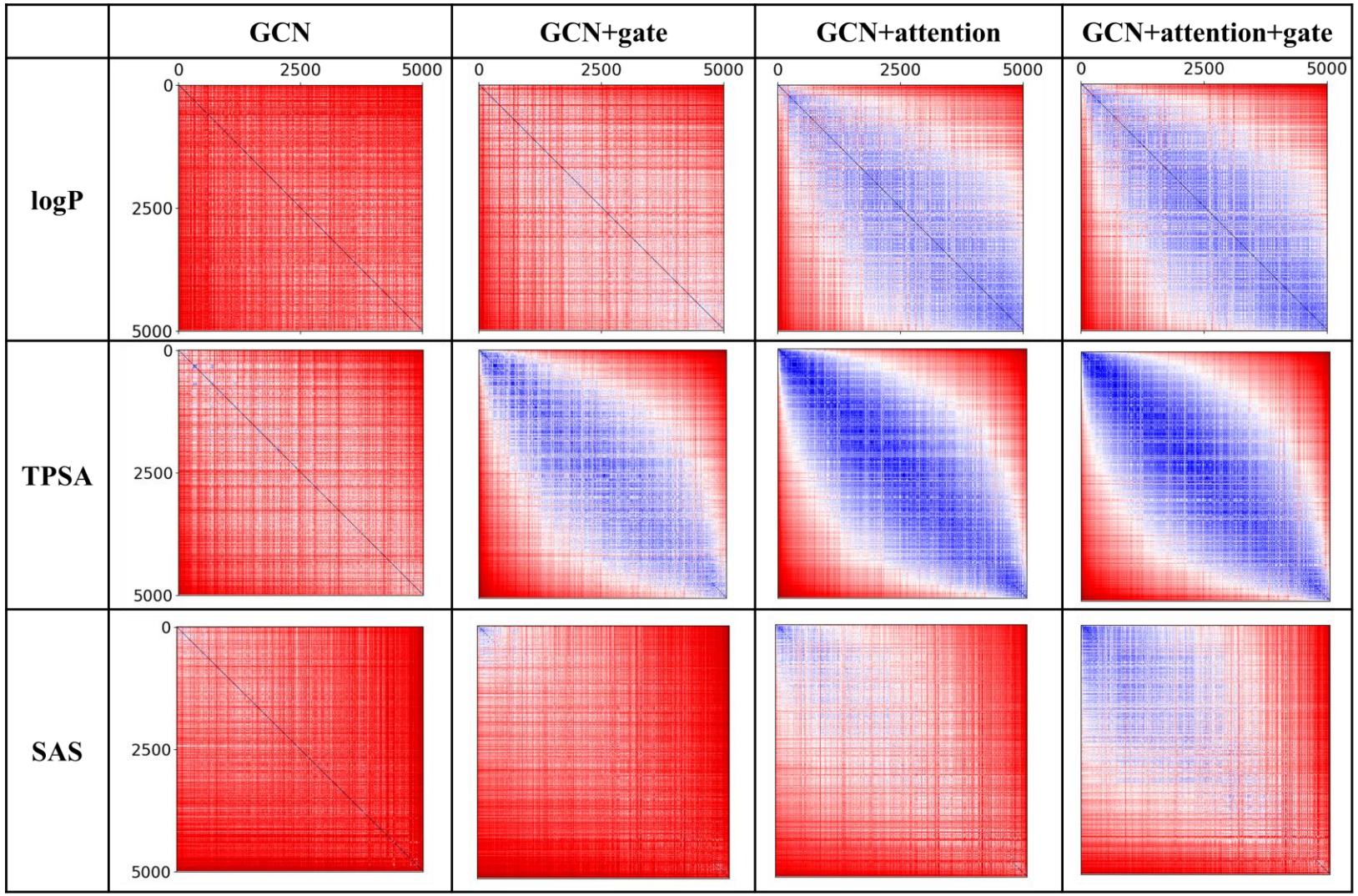
- The GCN+attention+gate improves the vanilla GCN
- It shows comparable results and requires much lower computational costs than GGNN.

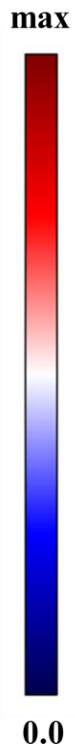
Ryu, Seongok, Jaechang Lim, Seung Hwan Hong and Woo Youn Kim. "Deeply learning molecular structure-property relationships using attention- and gate-augmented graph convolutional network." *arXiv preprint arXiv:1805.10988* (2018).

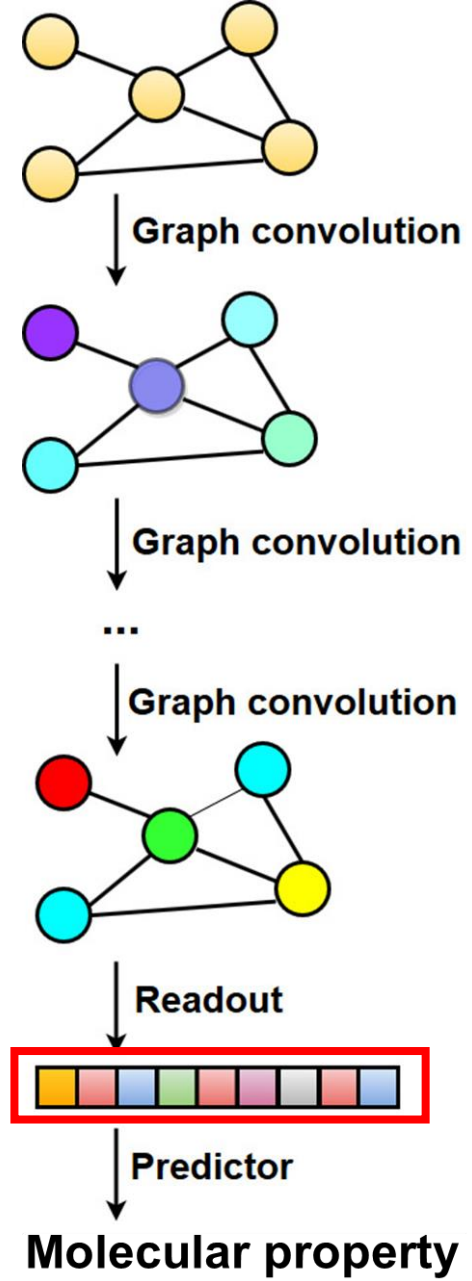


- The machine can recognize atoms in a similar environment.
- Quantitative structure property relationships (QSPR)

Ryu, Seongok, Jaechang Lim, Seung Hwan Hong and Woo Youn Kim. "Deeply learning molecular structure-property relationships using attention- and gate-augmented graph convolutional network." *arXiv preprint arXiv:1805.10988* (2018).







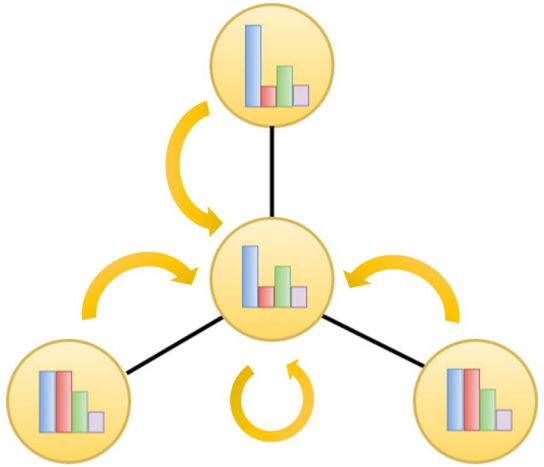
- Molecules with similar property locate close with each other in a latent space.
- It is key for *de novo* molecular engineering.

KAIST

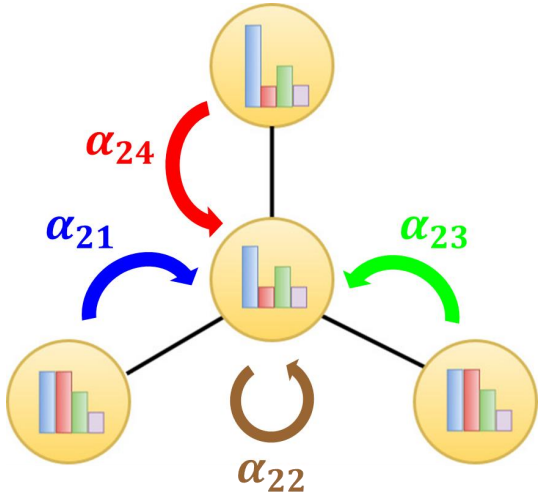
Ryu, Seongok, Jaechang Lim, Seung Hwan Hong and Woo Youn Kim. "Deeply learning molecular structure-property relationships using attention- and gate-augmented graph convolutional network." *arXiv preprint arXiv:1805.10988* (2018).

Summary

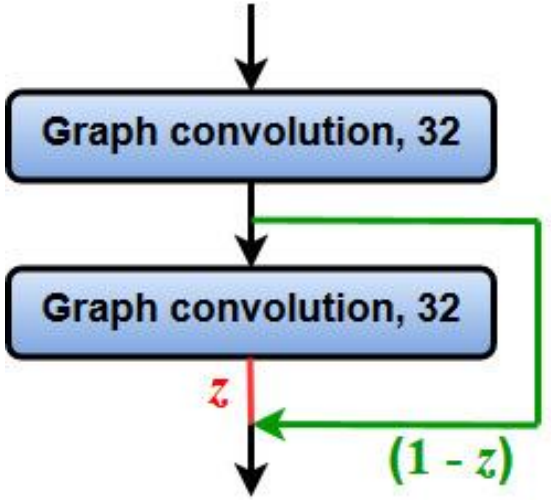
Graph convolution shares weights for all nodes in a graph

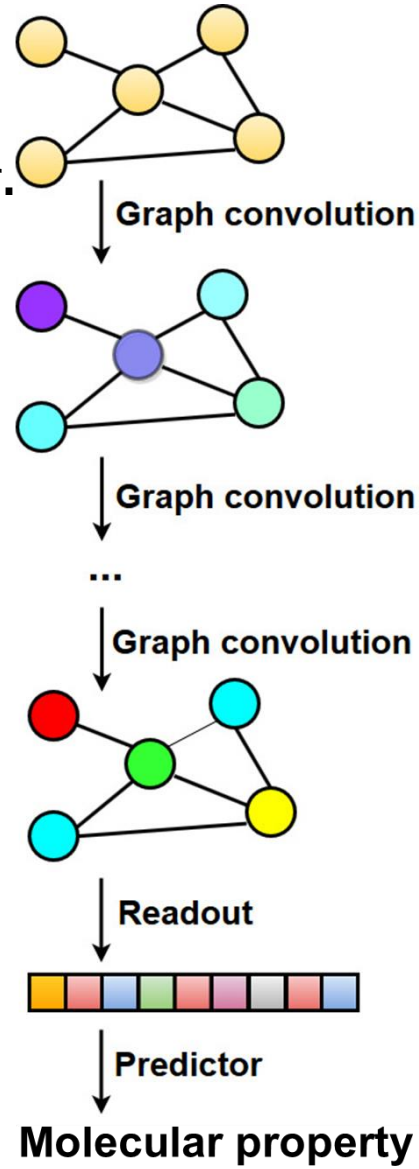


Attention mechanism mimics molecular interactions

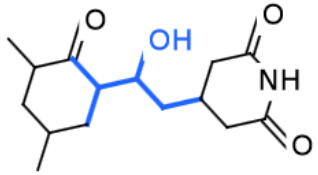


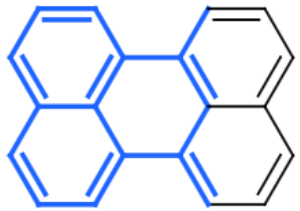
Using gated skip-connection updates atom features much better.

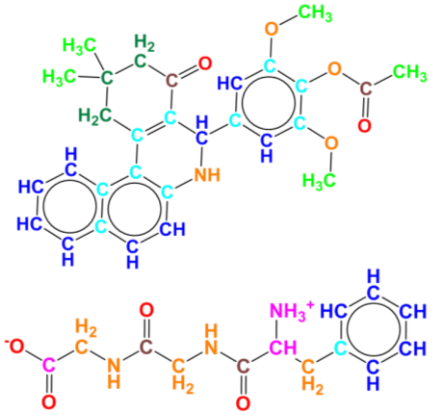


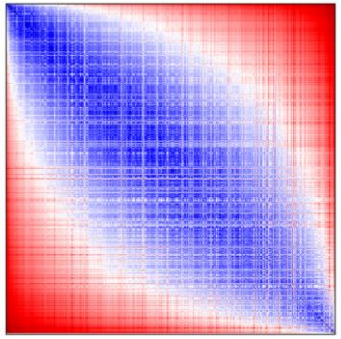


Neural machines can recognize the key features for molecular property predictions. In addition, **A better-developed model** performs **better** in **updating properties** and **predicting properties**.









KAIST

New terms

- Graph neural network
- Graph convolutional network
- Readout and permutation invariance
- Gated skip-connection
- Attention mechanism
- Quantitative structure property relationships (QSPR)
- Neural fingerprint
- Atom features
- Graph features