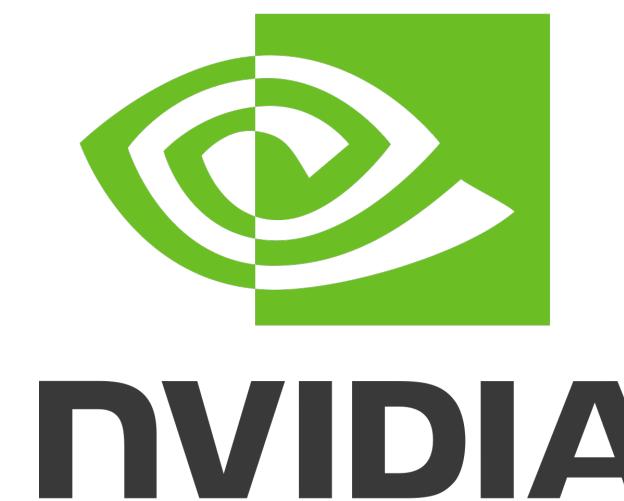


Provably Powerful Graph Neural Networks

Haggai Maron^{*}, Heli Ben-Hamu^{*}, Hadar Serviansky^{*}, Yaron Lipman

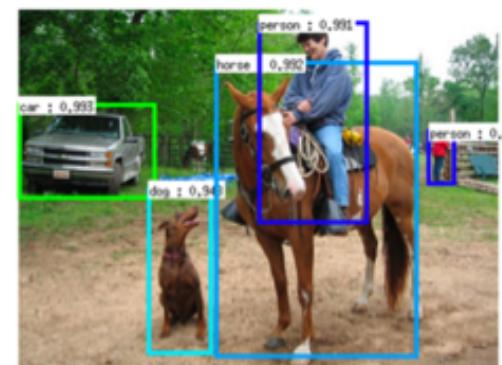
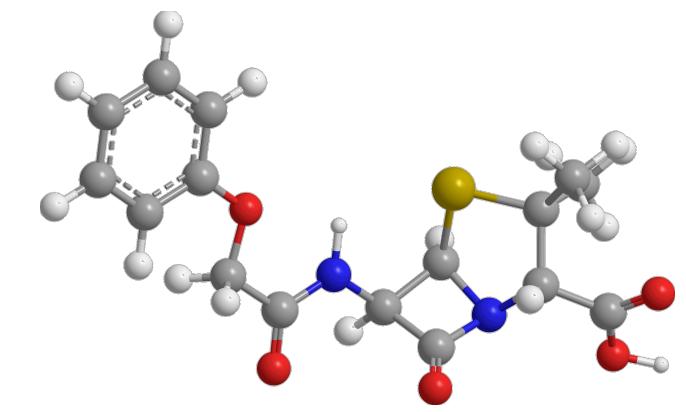
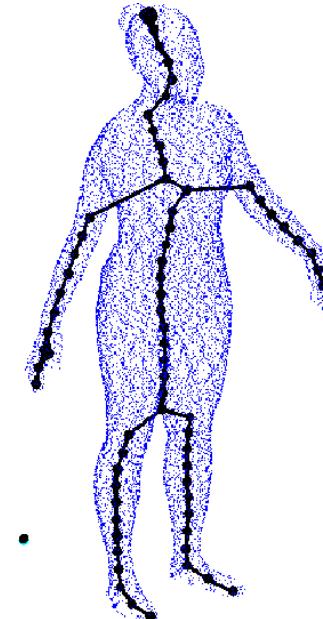
^{*}equal contributors ^{*}currently at NVIDIA Research



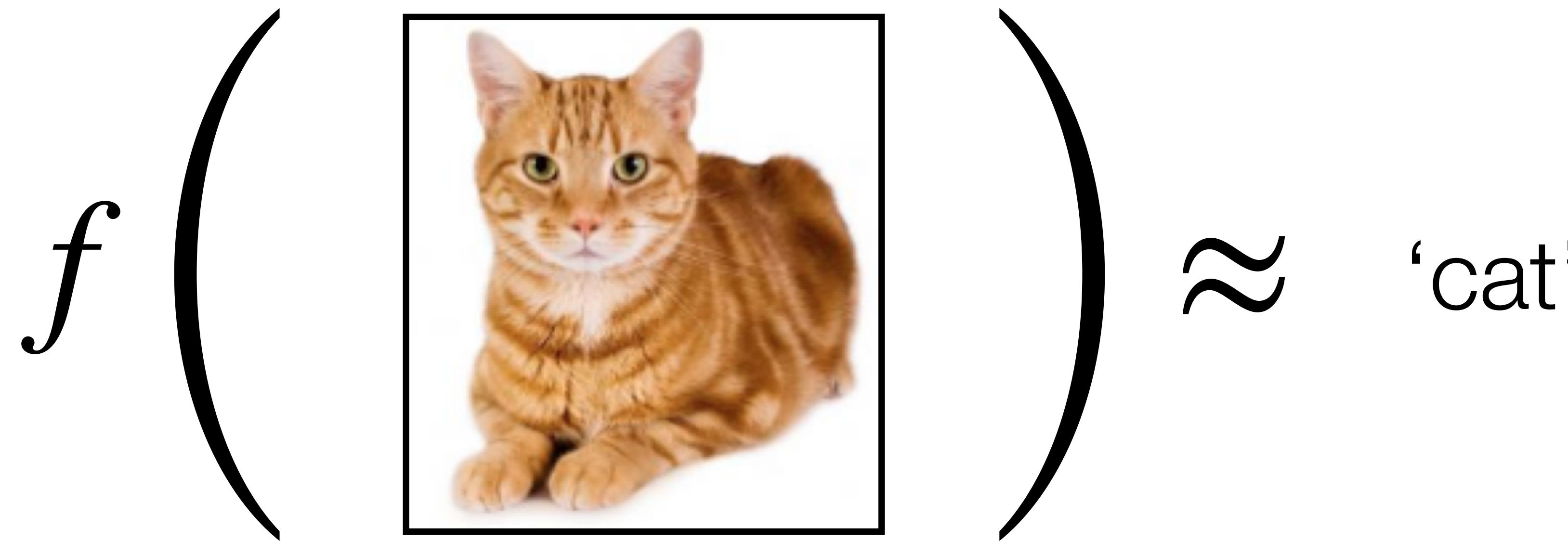
מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

Graph learning

- 3D Shapes
- Molecules and chemical compounds
- Social Networks
- Scenes in images



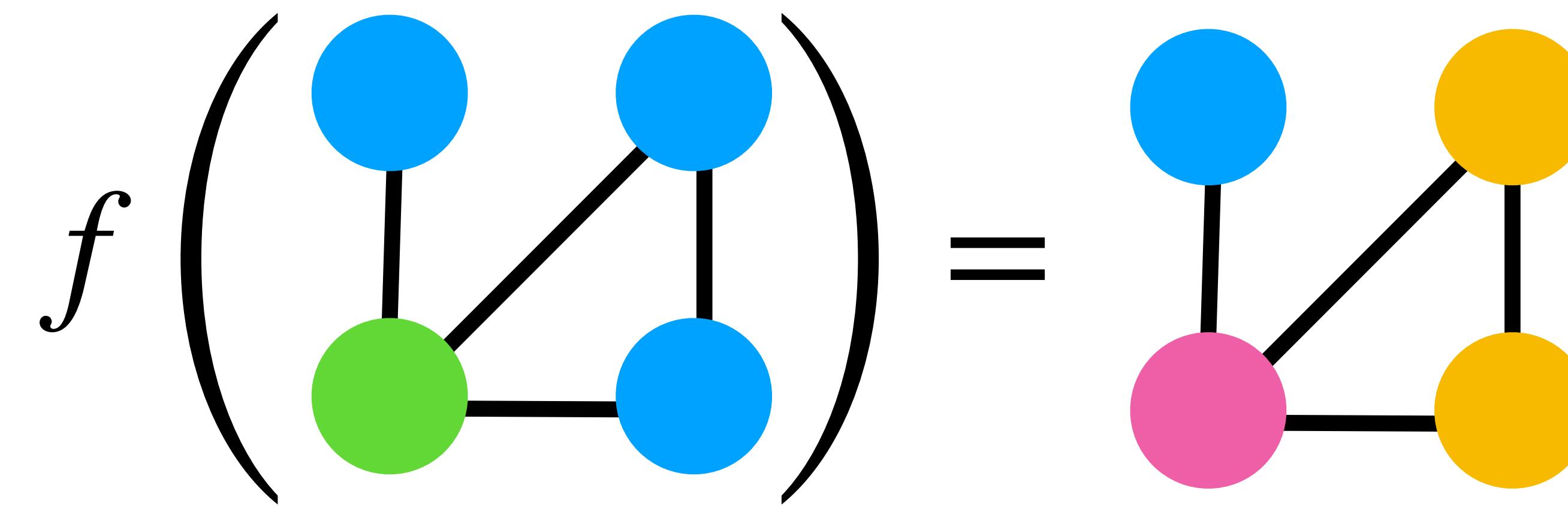
Supervised learning on graphs



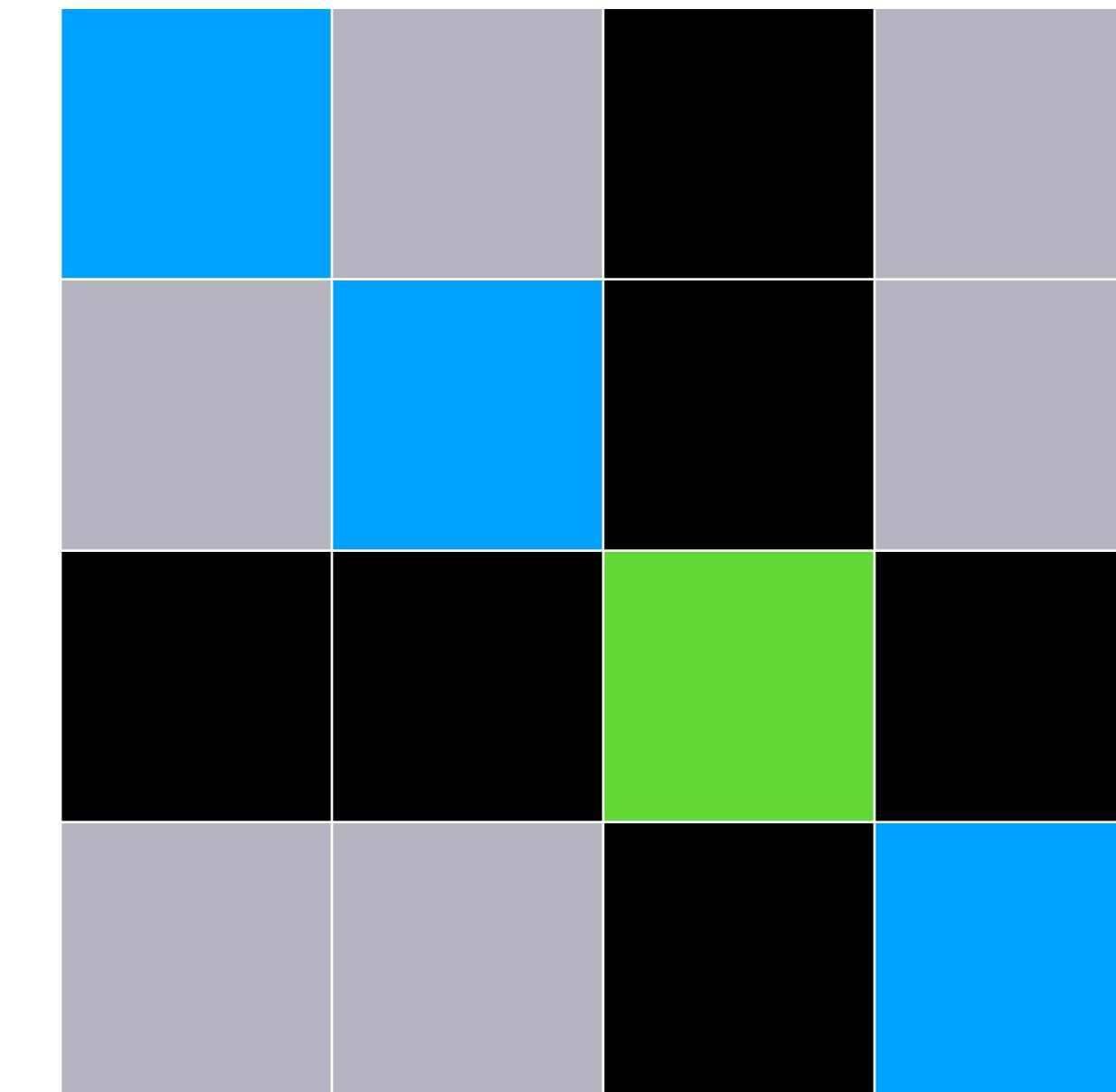
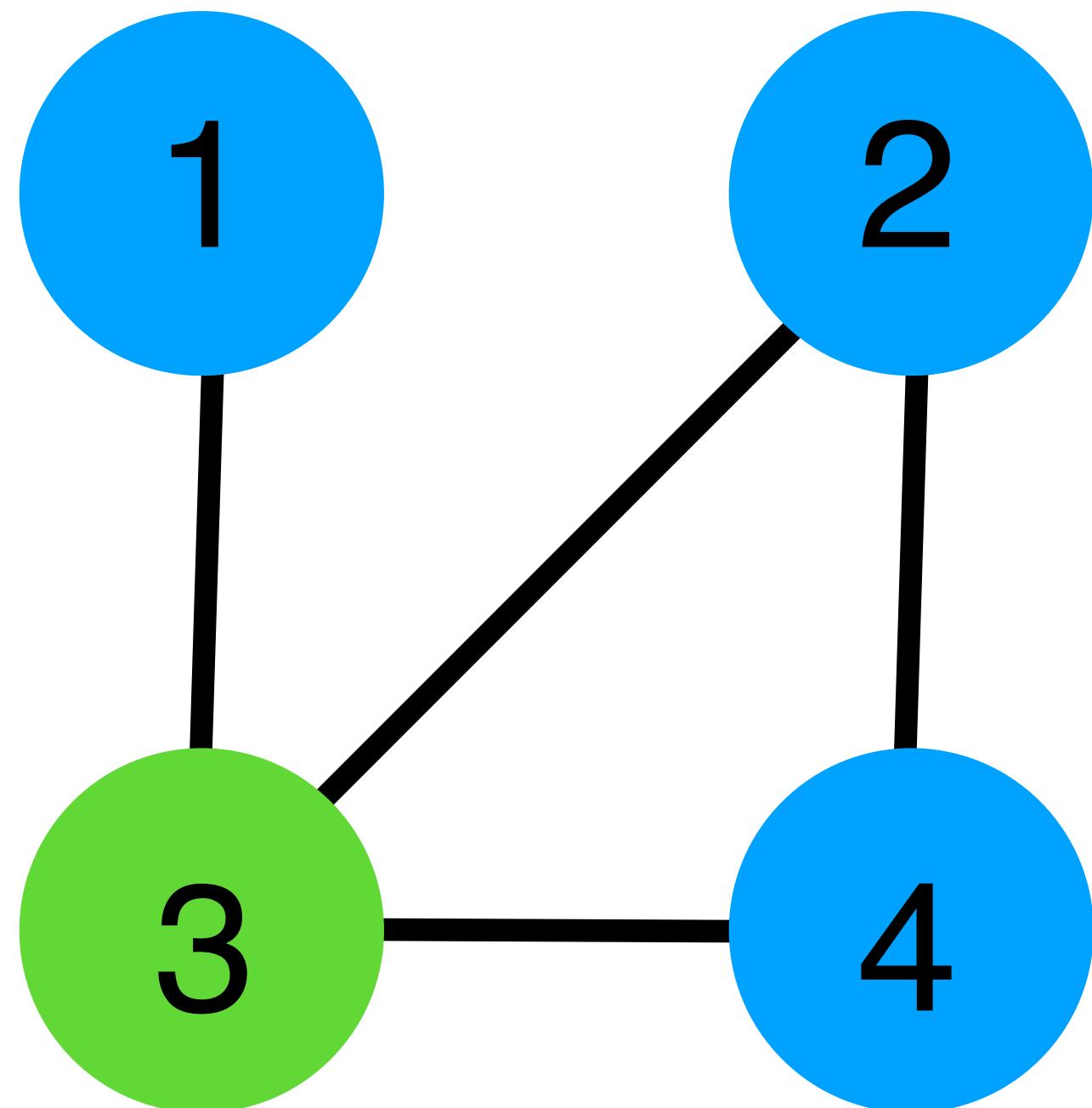
Supervised learning on graphs

$$f \left(\begin{array}{c} \text{graph} \\ \text{(4 nodes, 3 edges)} \end{array} \right) = \text{'label'}$$

Supervised learning on graphs



Graphs encoded as matrices



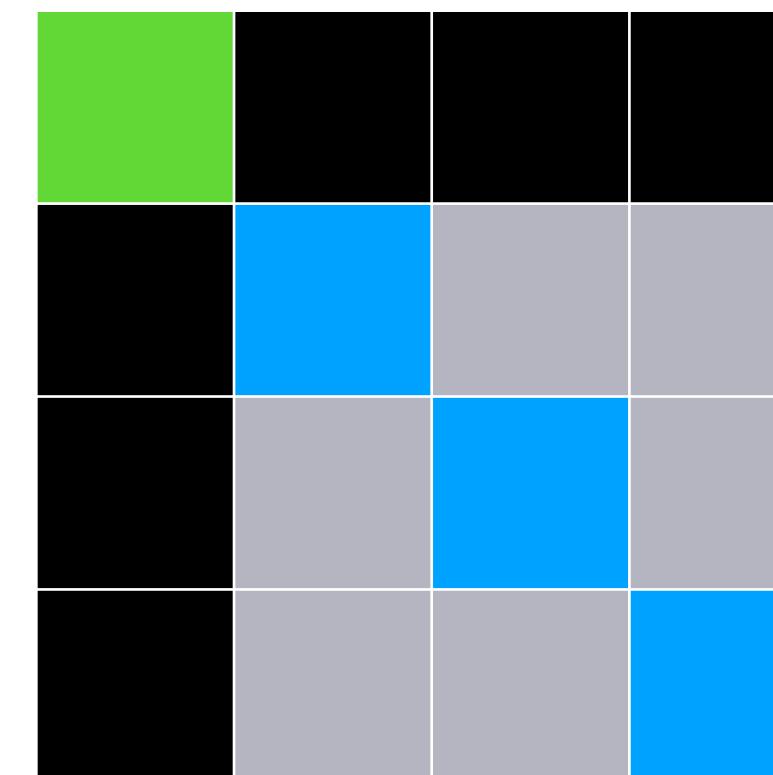
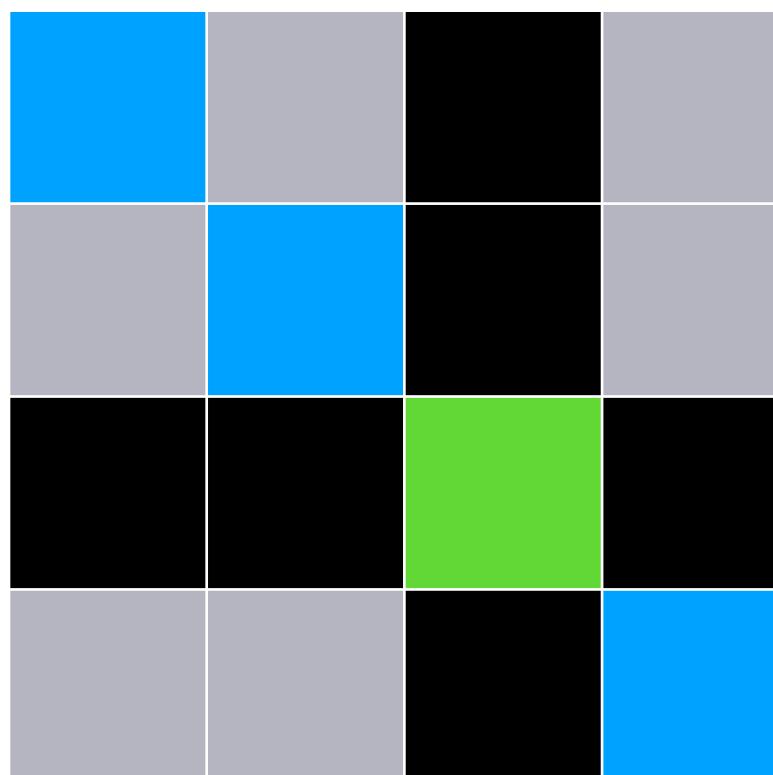
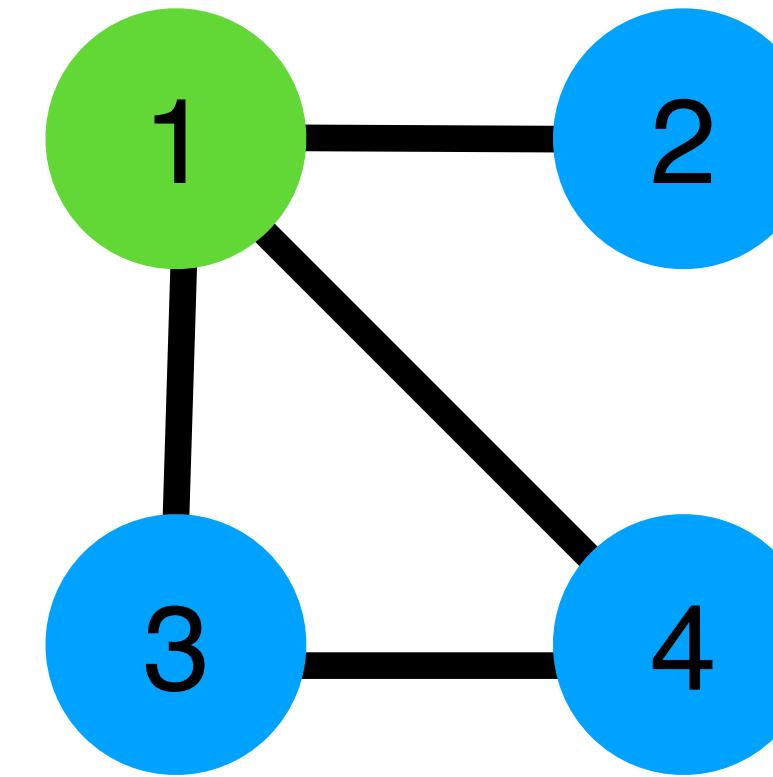
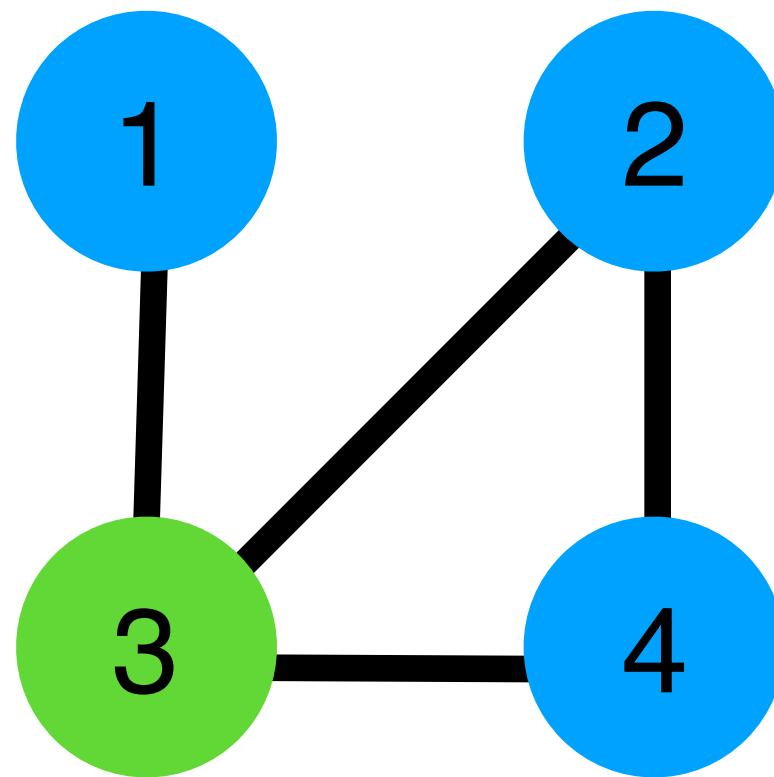
X

Graph learning

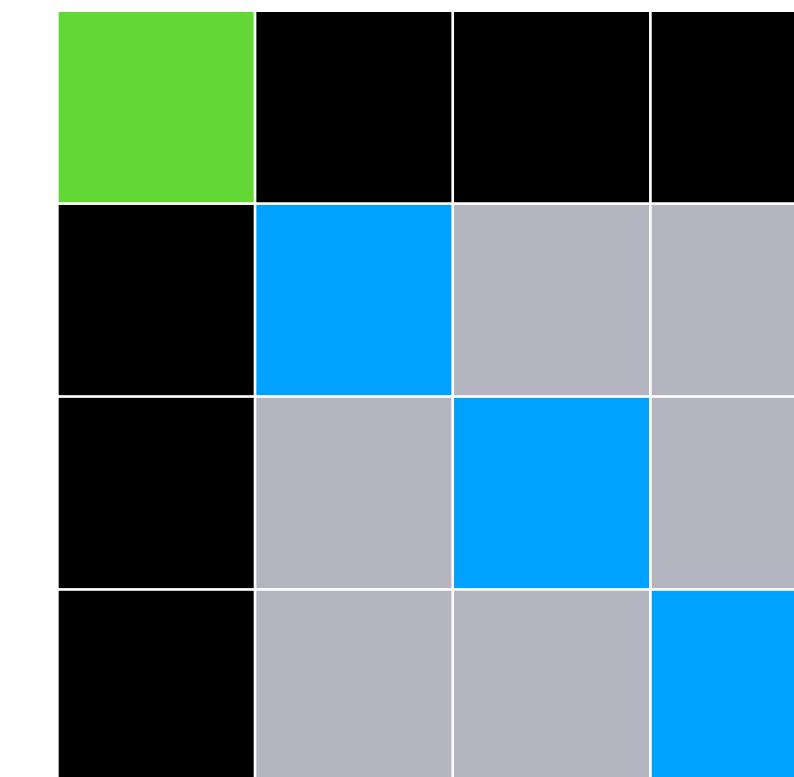
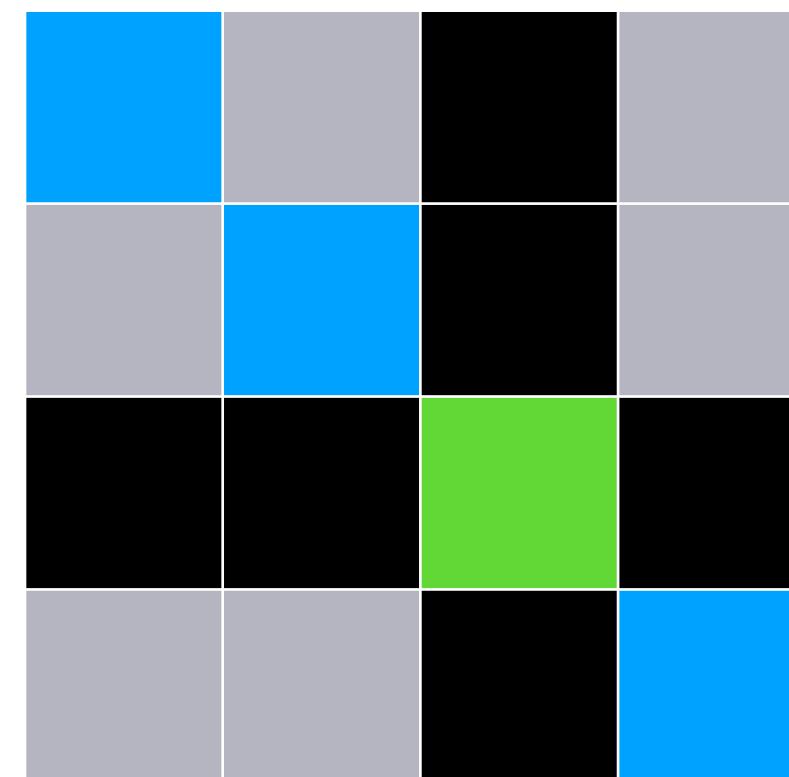
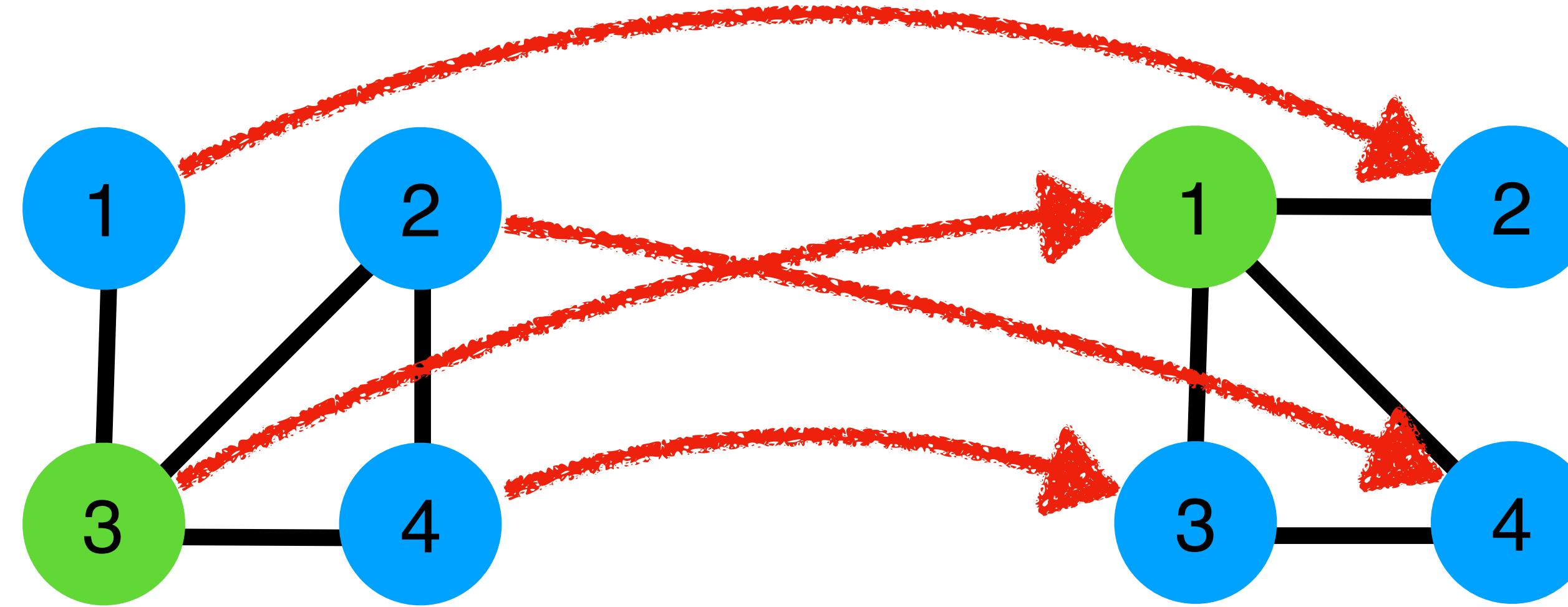
$$f \left(\begin{array}{cccc} \text{green} & \text{black} & \text{black} & \text{black} \\ \text{black} & \text{blue} & \text{grey} & \text{grey} \\ \text{black} & \text{grey} & \text{blue} & \text{black} \\ \text{black} & \text{black} & \text{blue} & \text{blue} \end{array} \right) = \begin{array}{c} \text{pink} \\ \text{blue} \\ \text{yellow} \\ \text{yellow} \end{array}$$

A red arrow points from a question mark (?) to the input matrix, indicating the function f takes a graph as input.

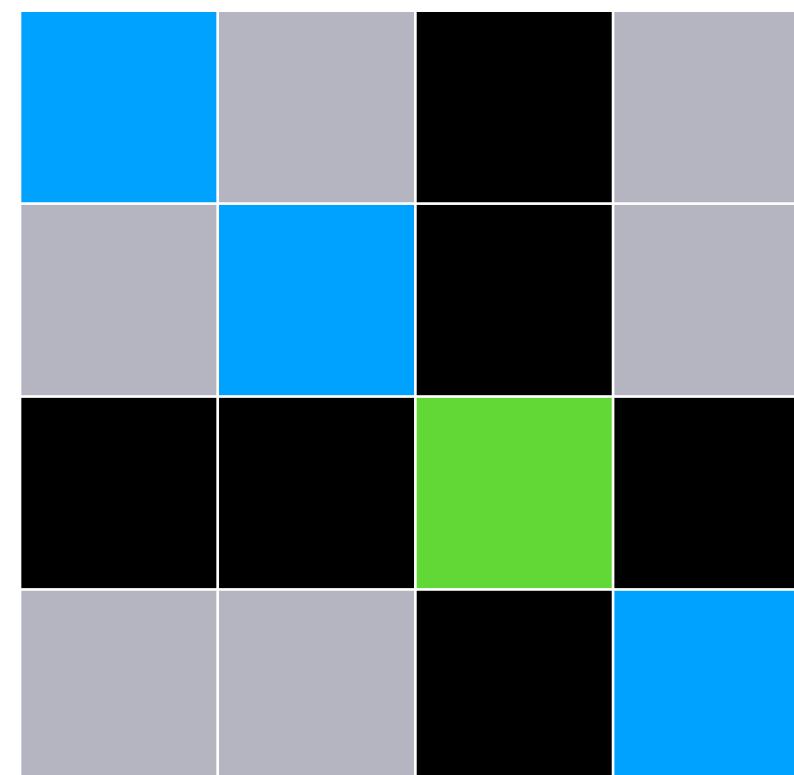
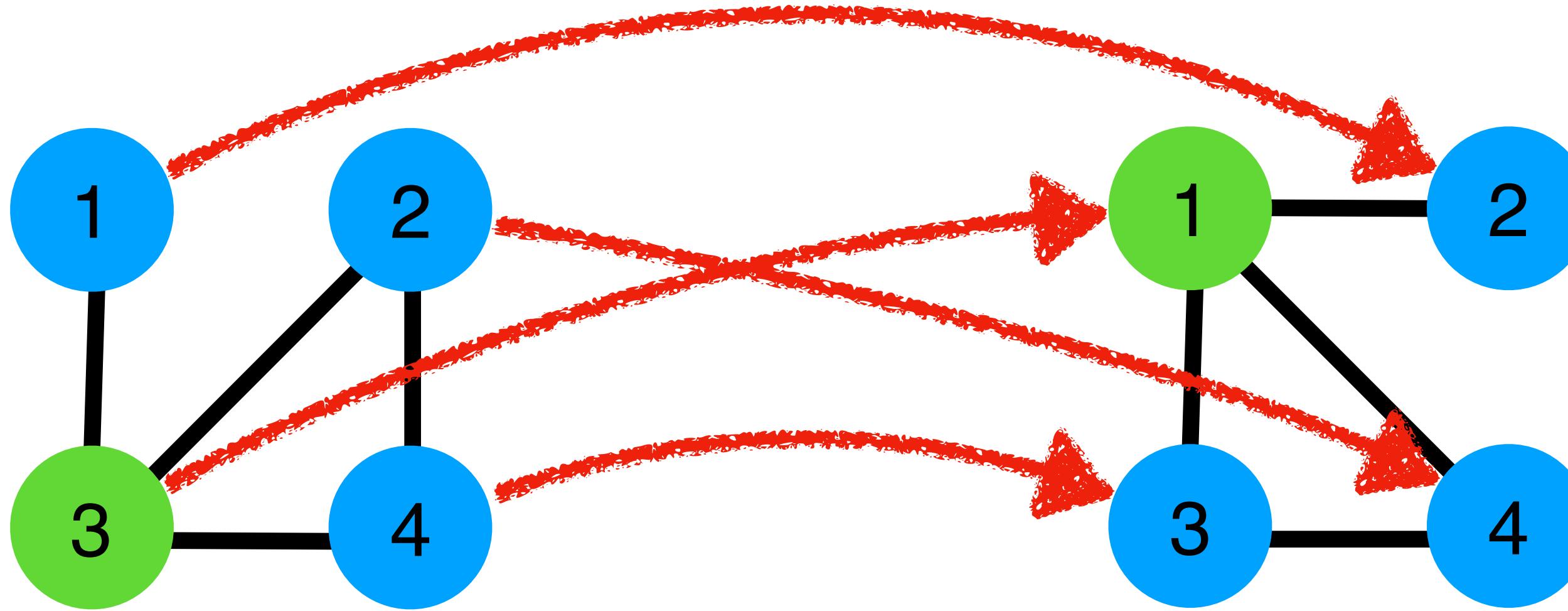
Graph isomorphism



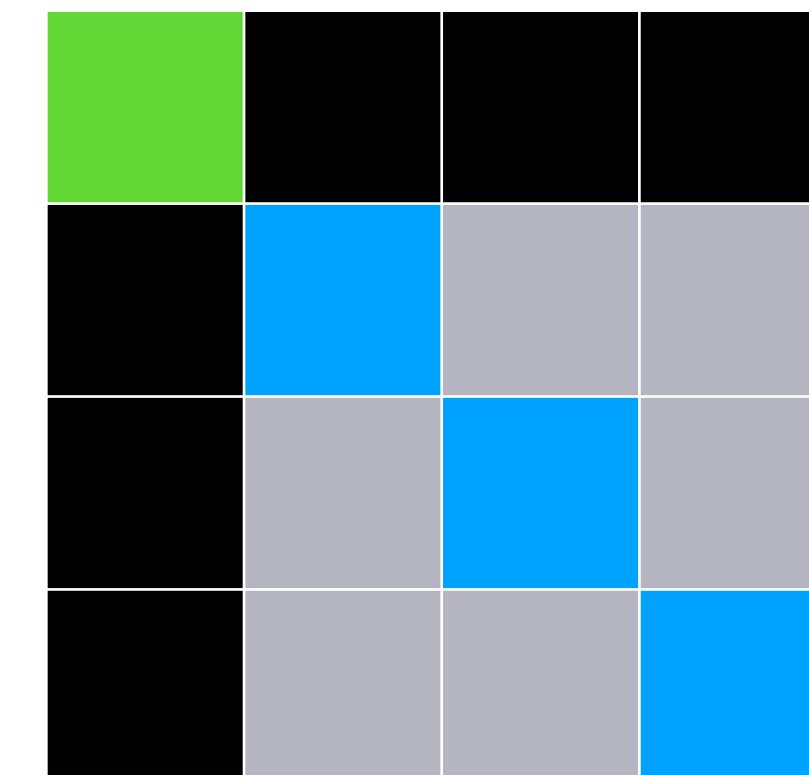
Graph isomorphism



Graph isomorphism



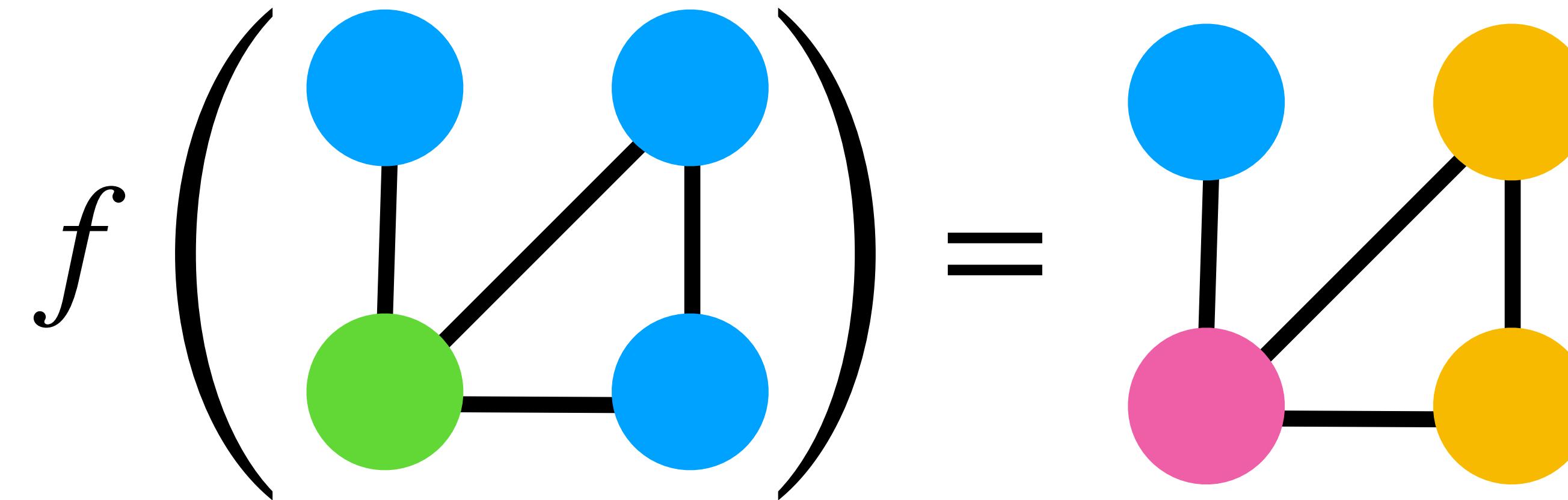
$$= P$$



$$P^T$$

Basic requirement: equivariance

$$f(\mathbf{P} \mathbf{X} \mathbf{P}^T) = \mathbf{P} f(\mathbf{X}) \mathbf{P}^T$$

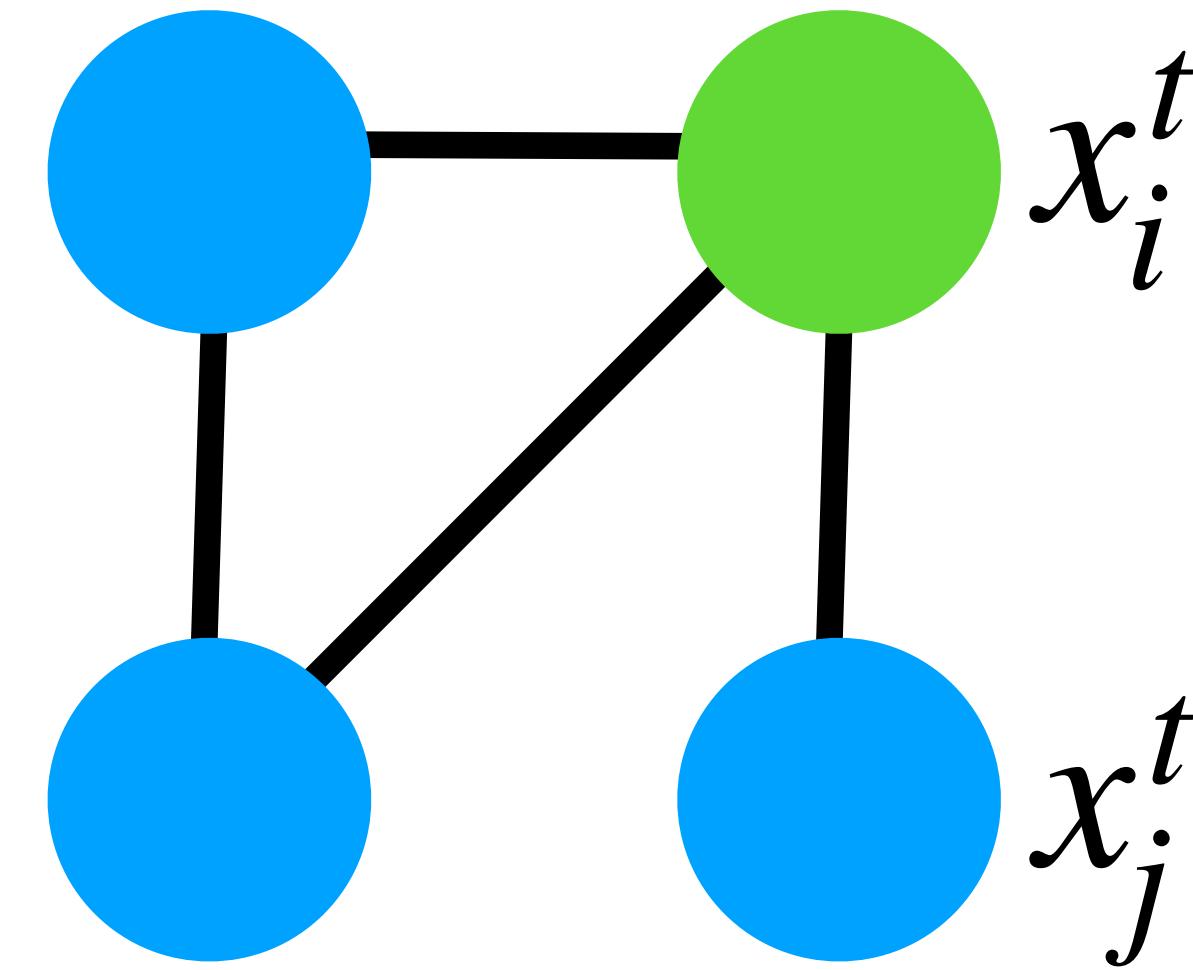


Equivariance: additional benefits

- Often reduces the number of parameters in the model (e.g. conv layers)
- Improves generalization
- Allows us to work with larger data
- (Sometimes) allows us to work with arbitrary object size

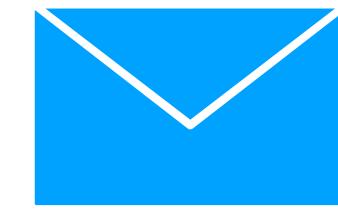
$$f(\textcolor{red}{P} X \textcolor{black}{P}^T) = \textcolor{black}{P} f(X) \textcolor{red}{P}^T$$

Message Passing

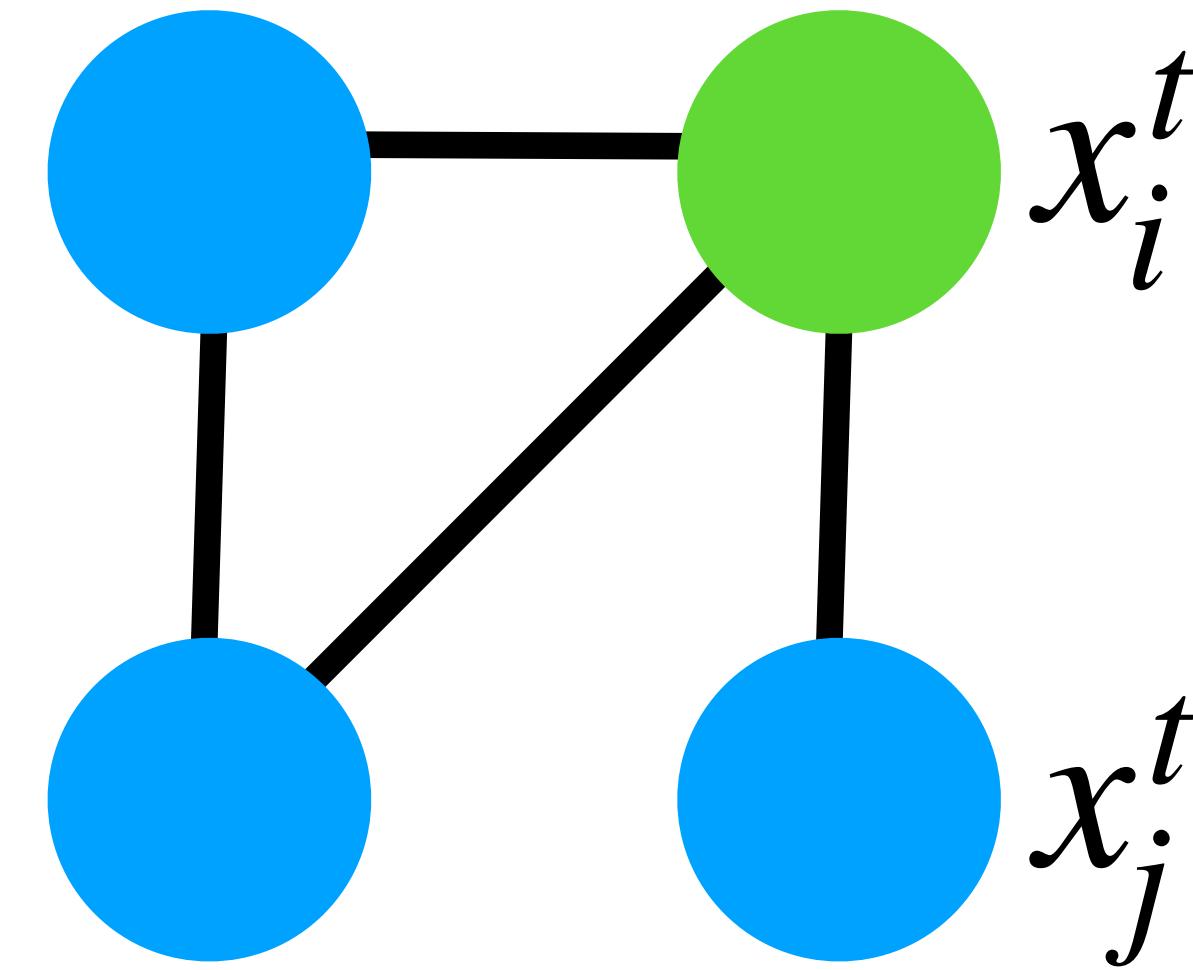


[Duvenaud et al. 2015; Kipf and Welling, 2016; Atwood and Towsley 2016; Niepert et al. 2016; Hamilton et al. 2017b; Velickovic et al. 2017; Monti et al. 2018; Gilmer et al., 2017; Morris et al. 2018; Xu et al. 2019]

Message Passing

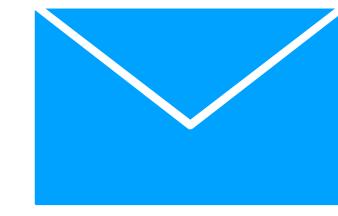


$$\sum_{j \in N(i)} M(x_i^t, x_j^t)$$

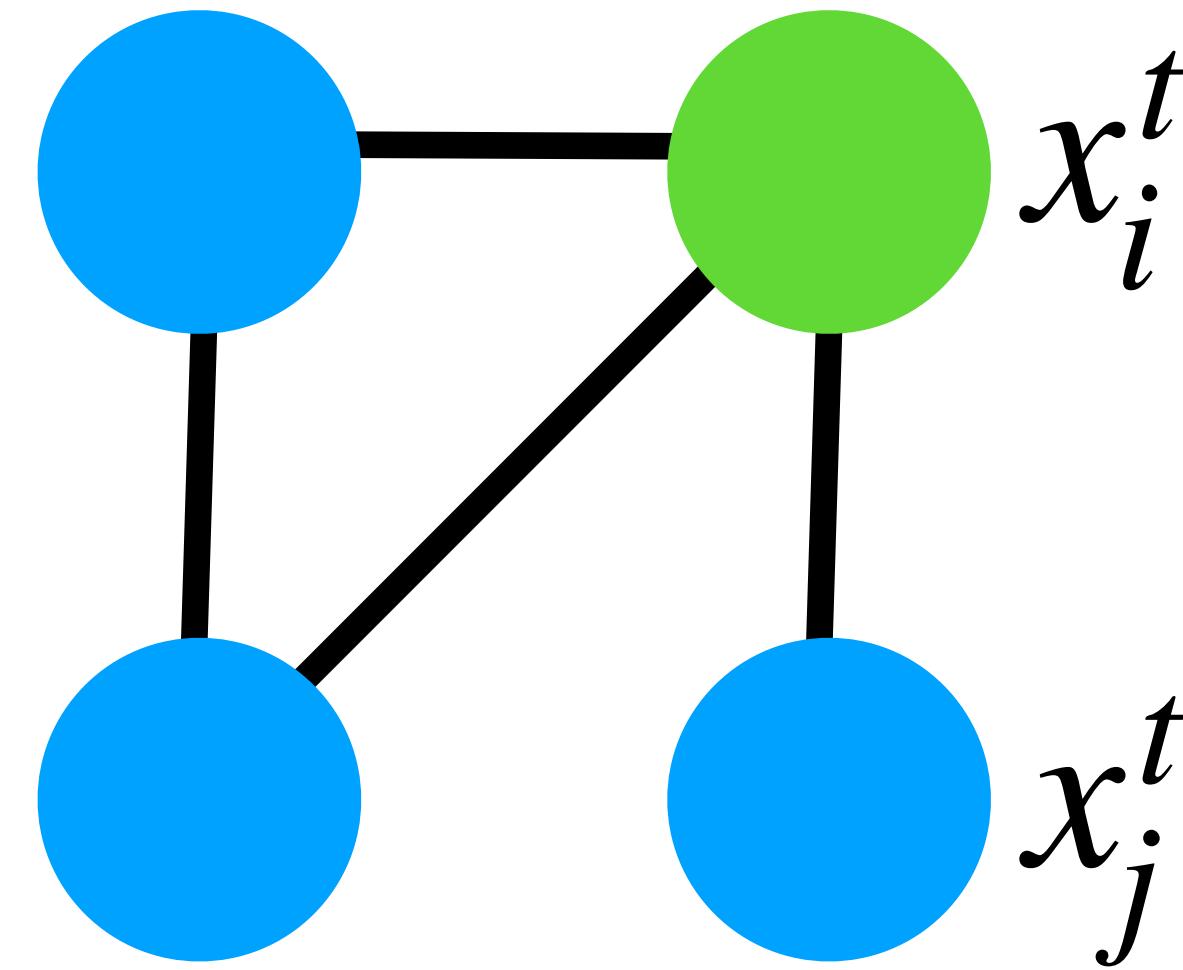


[Duvenaud et al. 2015; Kipf and Welling, 2016; Atwood and Towsley 2016; Niepert et al. 2016; Hamilton et al. 2017b; Velickovic et al. 2017; Monti et al. 2018; Gilmer et al., 2017; Morris et al. 2018; Xu et al. 2019]

Message Passing



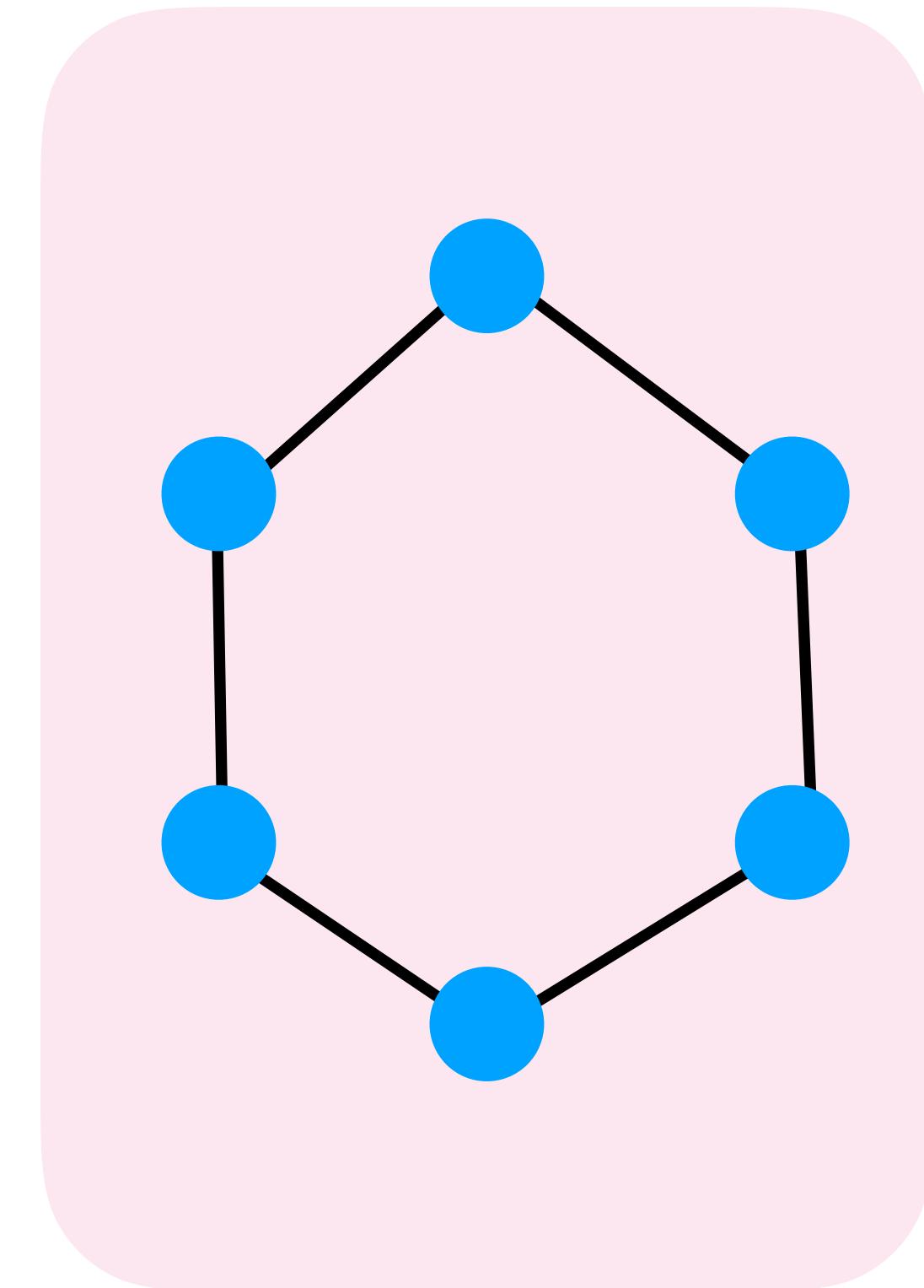
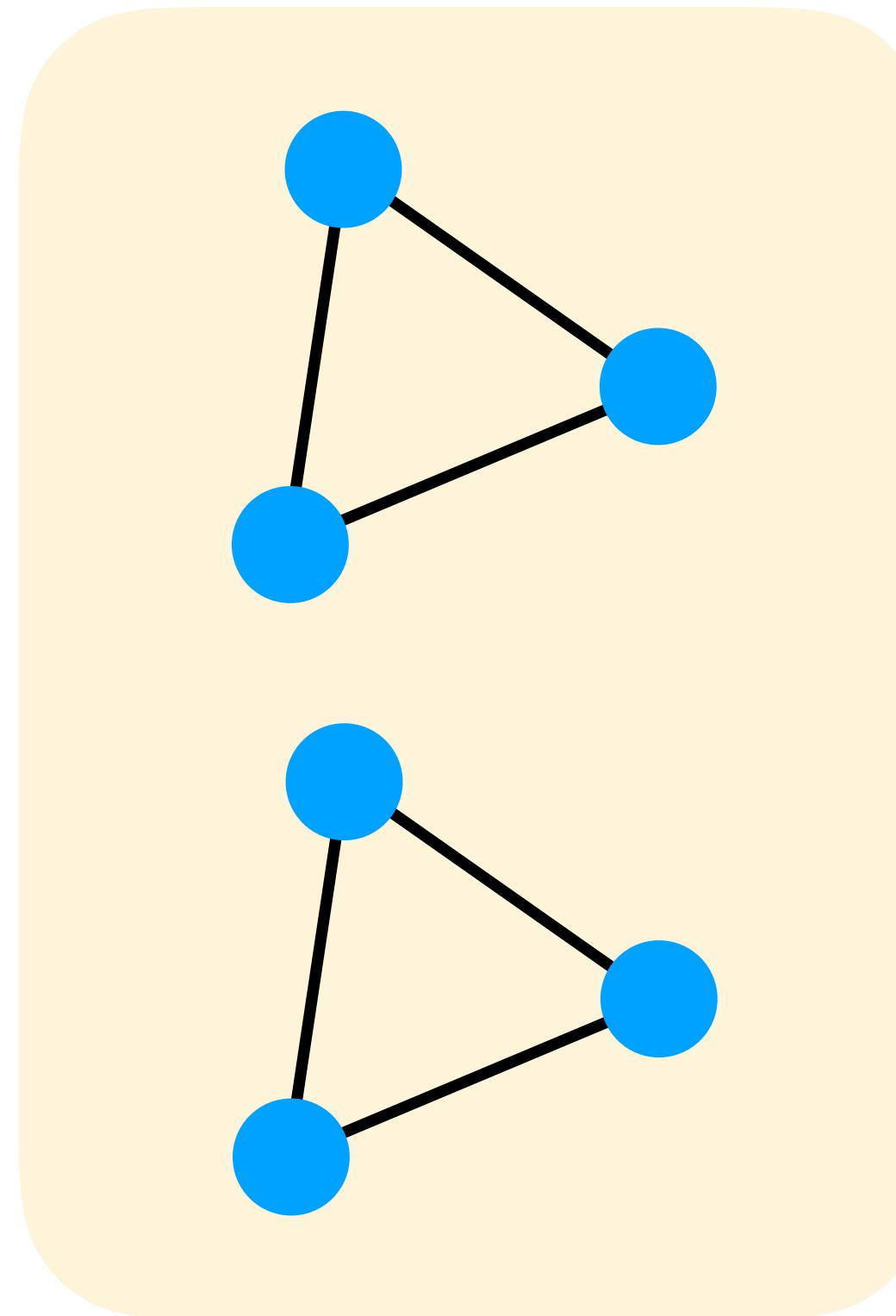
$$x_i^{t+1} = U\left(x_i^t, \sum_{j \in N(i)} M(x_i^t, x_j^t)\right)$$



[Duvenaud et al. 2015; Kipf and Welling, 2016; Atwood and Towsley 2016; Niepert et al. 2016; Hamilton et al. 2017b; Velickovic et al. 2017; Monti et al. 2018; Gilmer et al., 2017; Morris et al. 2018; Xu et al. 2019]

Message Passing: limitations

$$x_i^{t+1} = U\left(x_i^t, \sum_{j \in N(i)} M(x_i^t, x_j^t)\right)$$



Questions:

1. How to measure expressive **power** of graph neural networks?

2. Can we find a simple model which is stronger?

1. How to measure expressive **power** of graph neural networks?

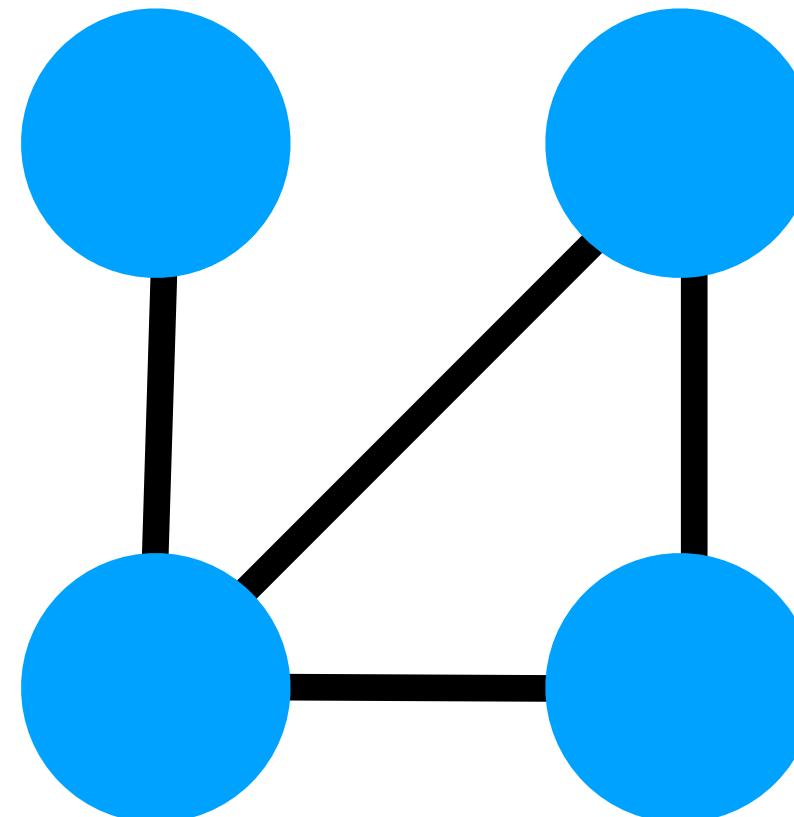
↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓

k -Weisfeiler Lehman

- Polynomial algorithms to test graph isomorphism

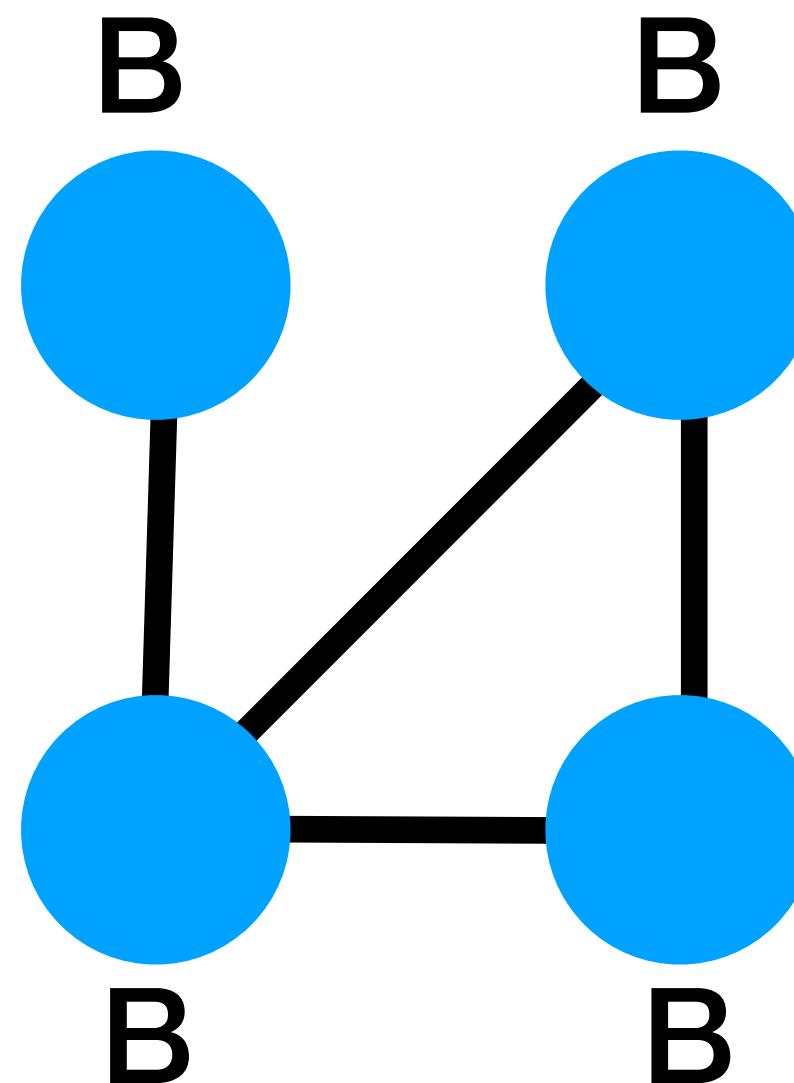
k -Weisfeiler Lehman

- Polynomial algorithms to test graph isomorphism
- The first in the hierarchy is **color refinement**:



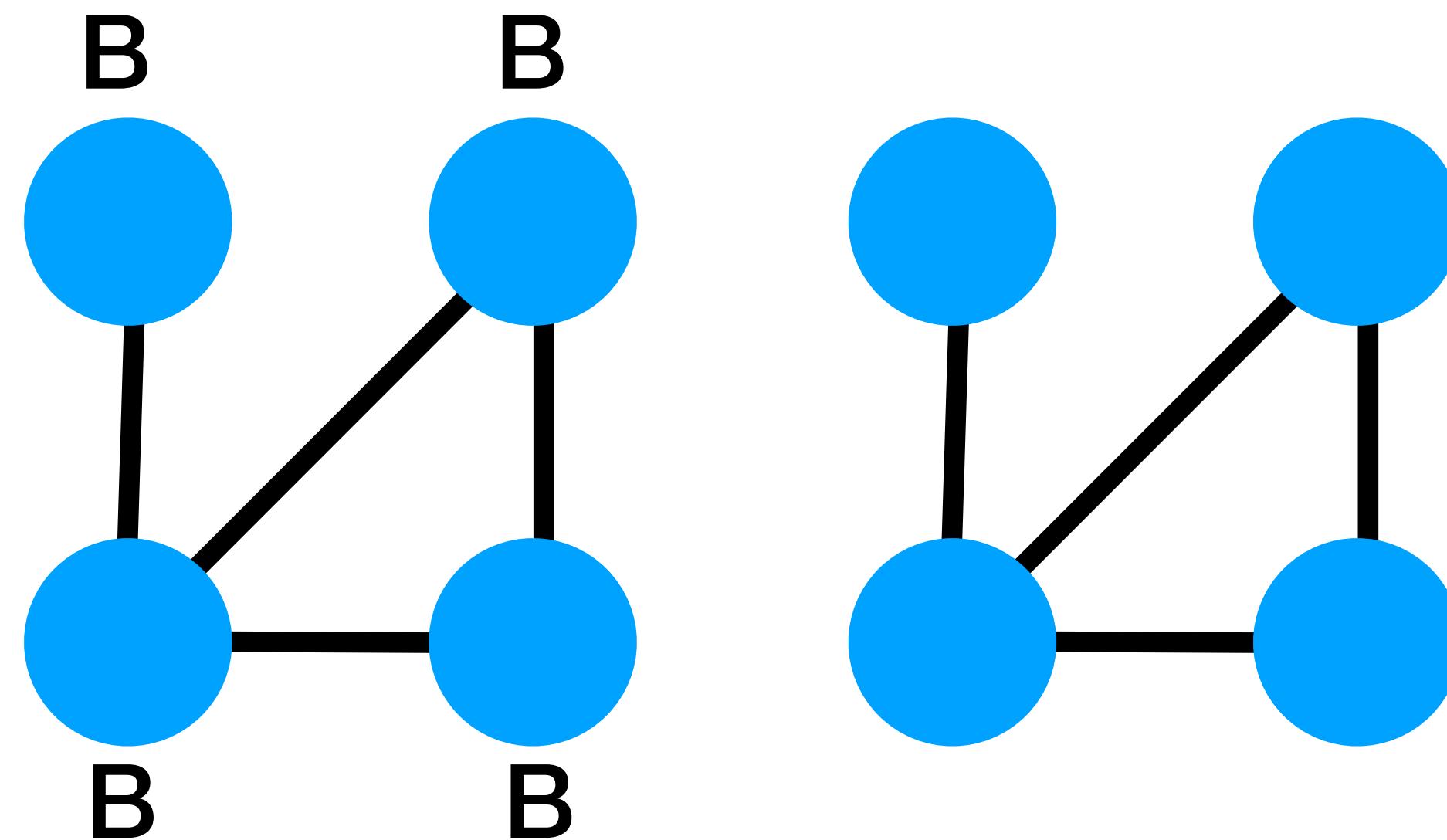
k -Weisfeiler Lehman

- Polynomial algorithms to test graph isomorphism
- The first in the hierarchy is **color refinement**:



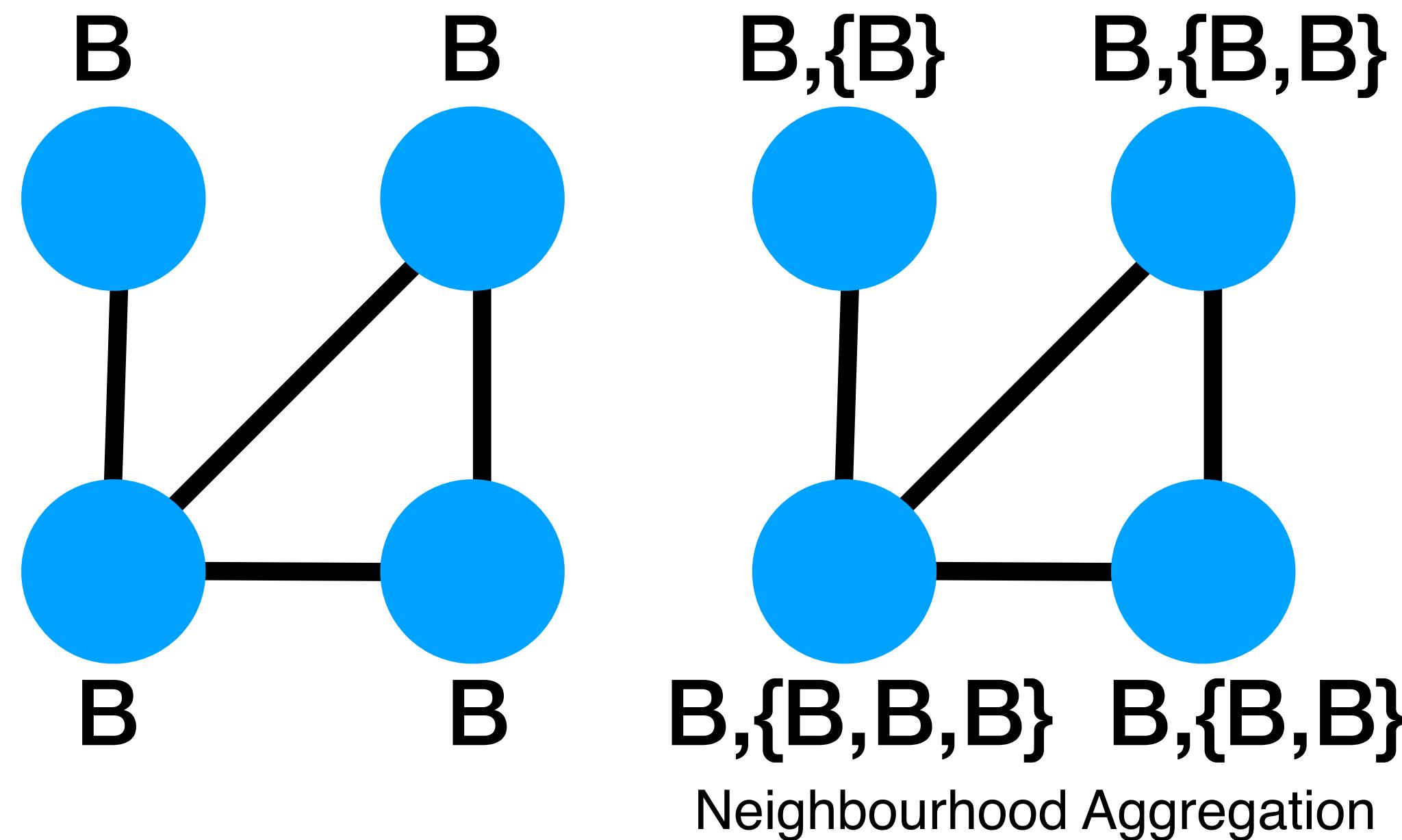
k -Weisfeiler Lehman

- Polynomial algorithms to test graph isomorphism
- The first in the hierarchy is **color refinement**:



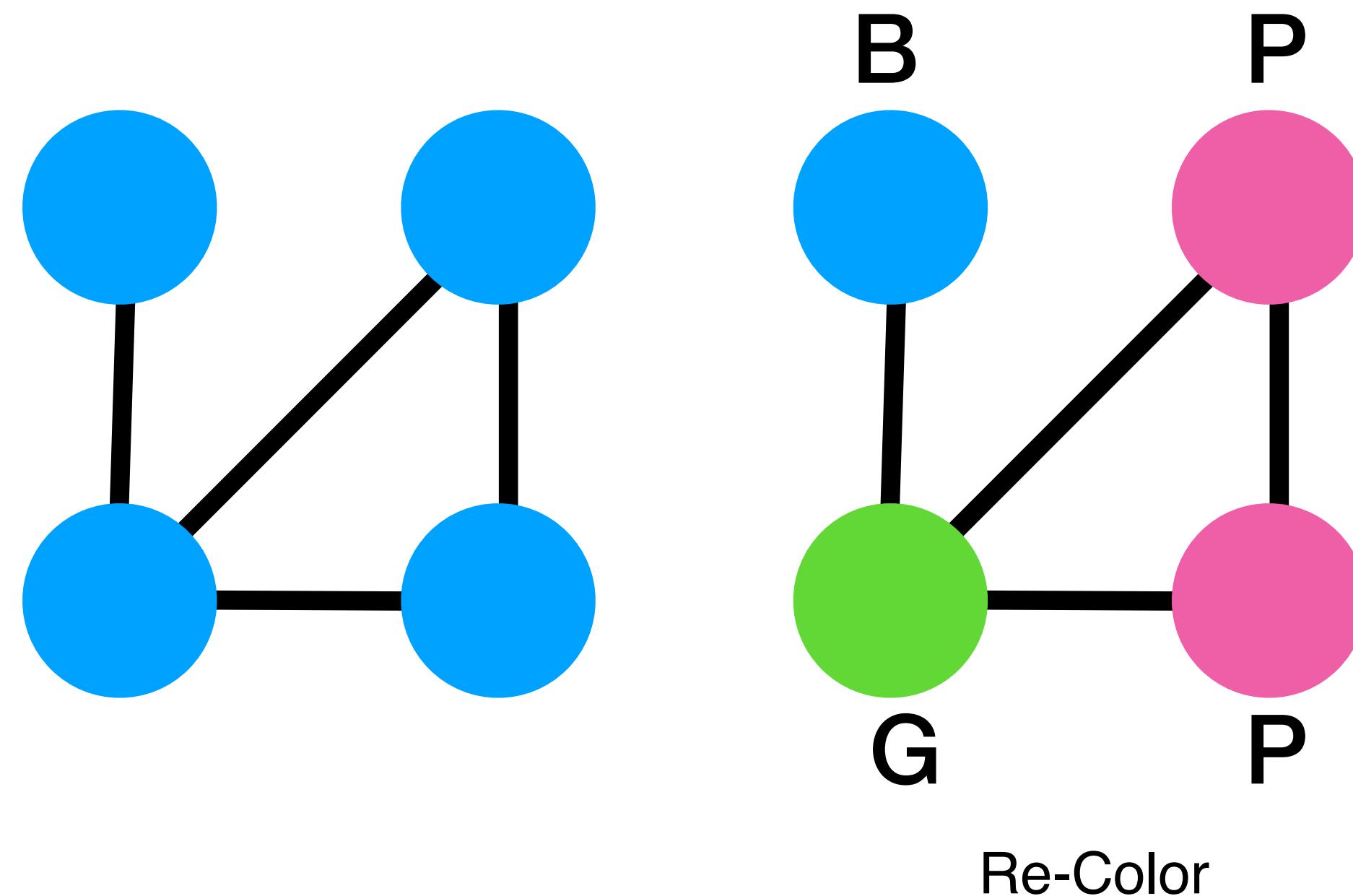
k -Weisfeiler Lehman

- Polynomial algorithms to test graph isomorphism
- The first in the hierarchy is **color refinement**:



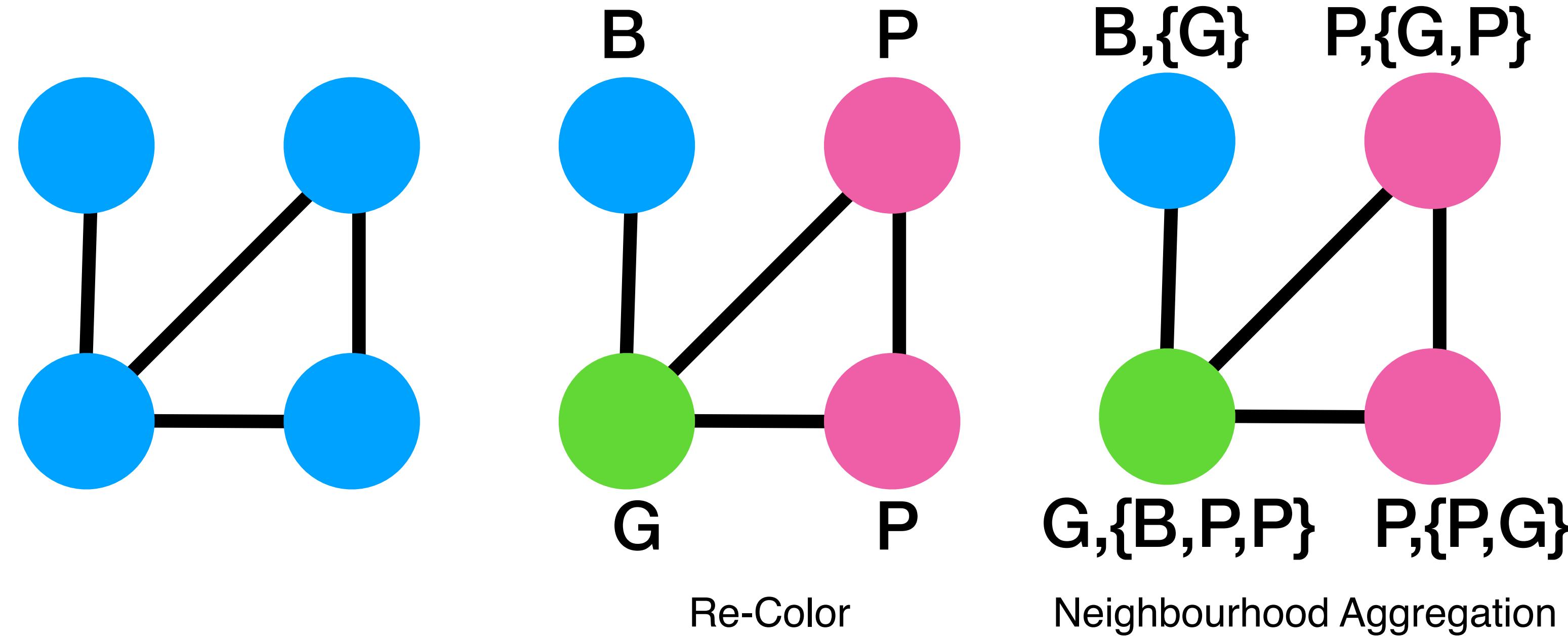
k -Weisfeiler Lehman

- Polynomial algorithms to test graph isomorphism
- The first in the hierarchy is **color refinement**:



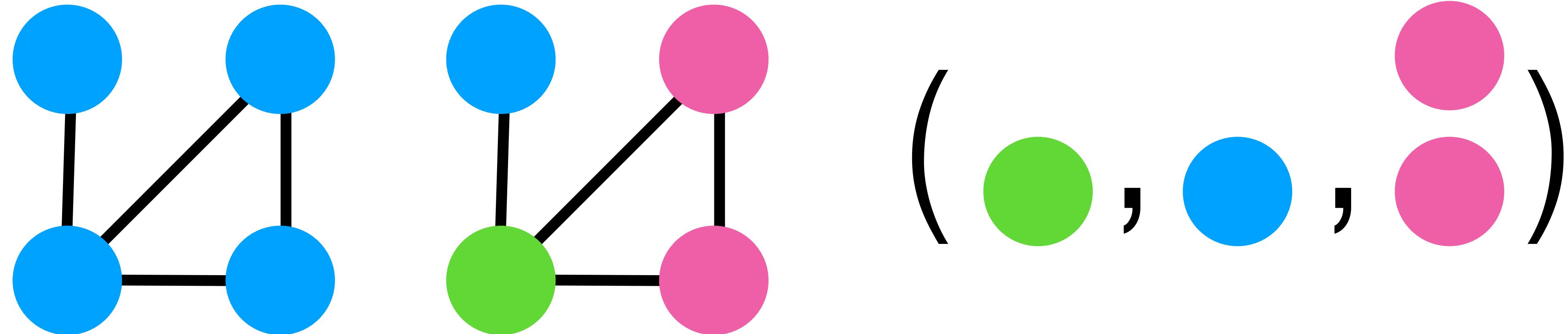
k -Weisfeiler Lehman

- Polynomial algorithms to test graph isomorphism
- The first in the hierarchy is **color refinement**:



k-Weisfeiler Lehman

- Polynomial algorithms to test graph isomorphism
- The first in the hierarchy is **color refinement**:

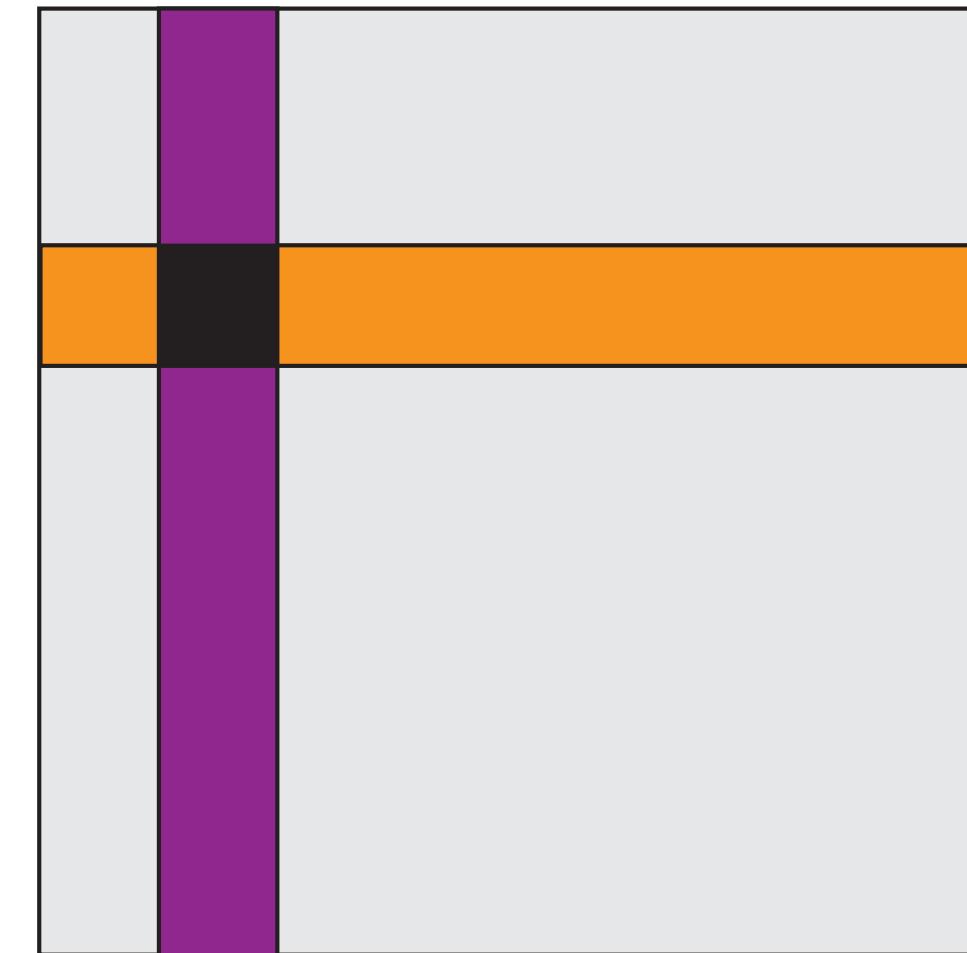


k -Weisfeiler Lehman

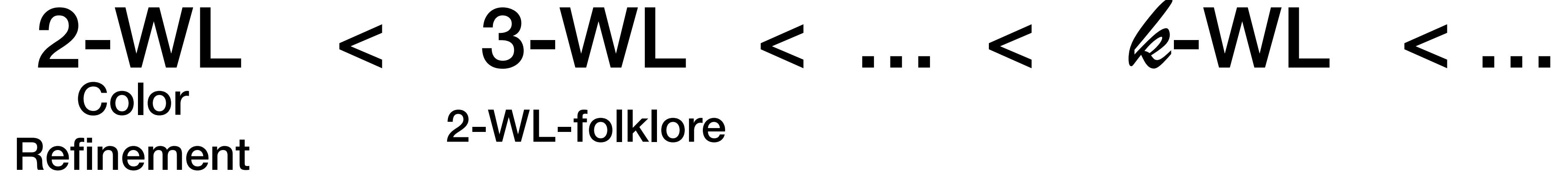
- Polynomial algorithms to test graph isomorphism
- The first in the hierarchy is **color refinement**
- k -Weisfeller Lehman is a generalization to data on **k -tuples of vertices**

k -Weisfeiler Lehman

- Polynomial algorithms to test graph isomorphism
- The first in the hierarchy is **color refinement**
- k -Weisfeller Lehman is a generalization to data on **k -tuples of vertices**

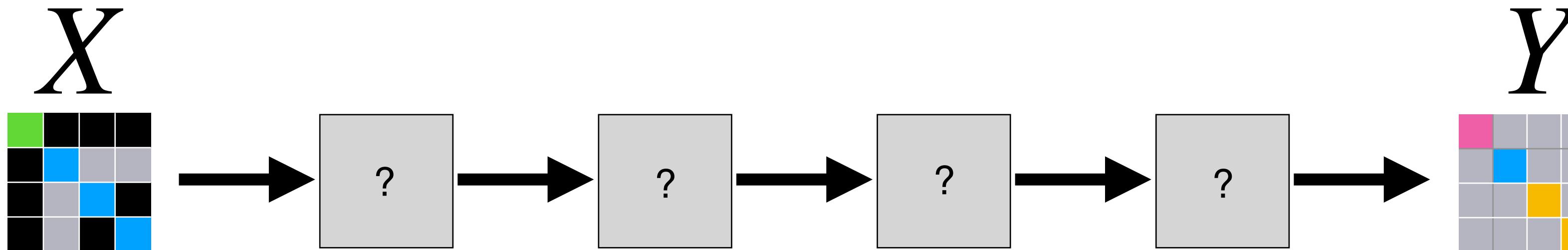


Message passing



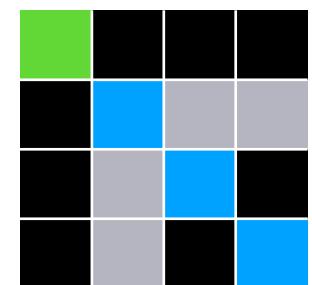
2. Can we find a simple model which is stronger?

A simple powerful model

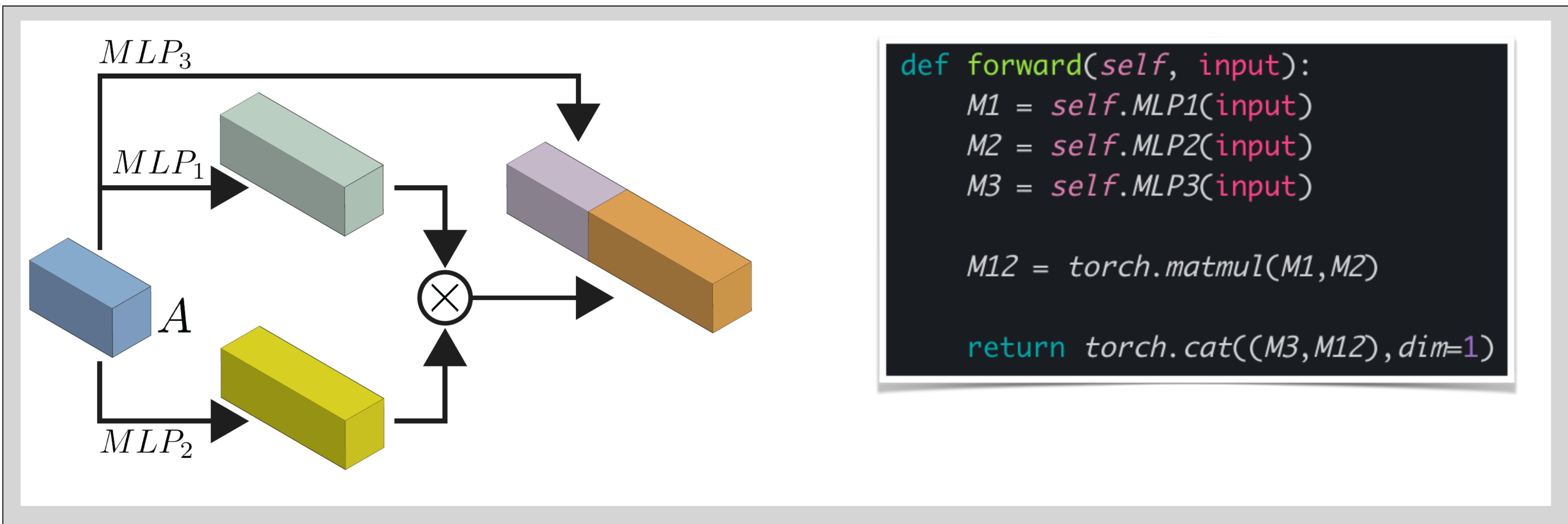
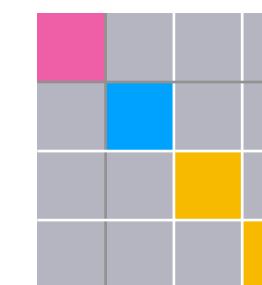


A simple powerful model

X

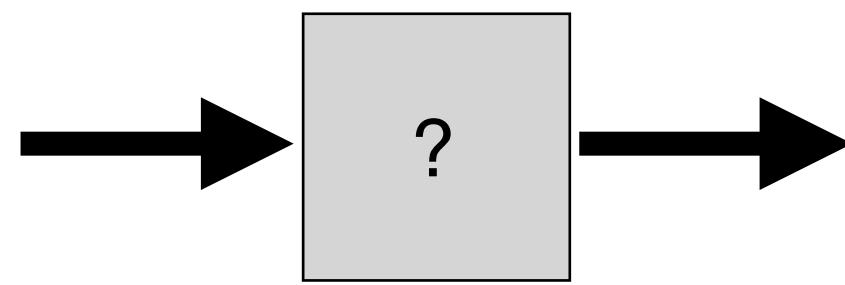
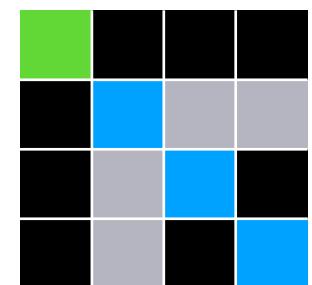


Y

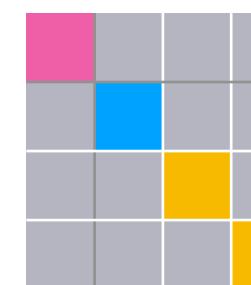


A simple powerful model

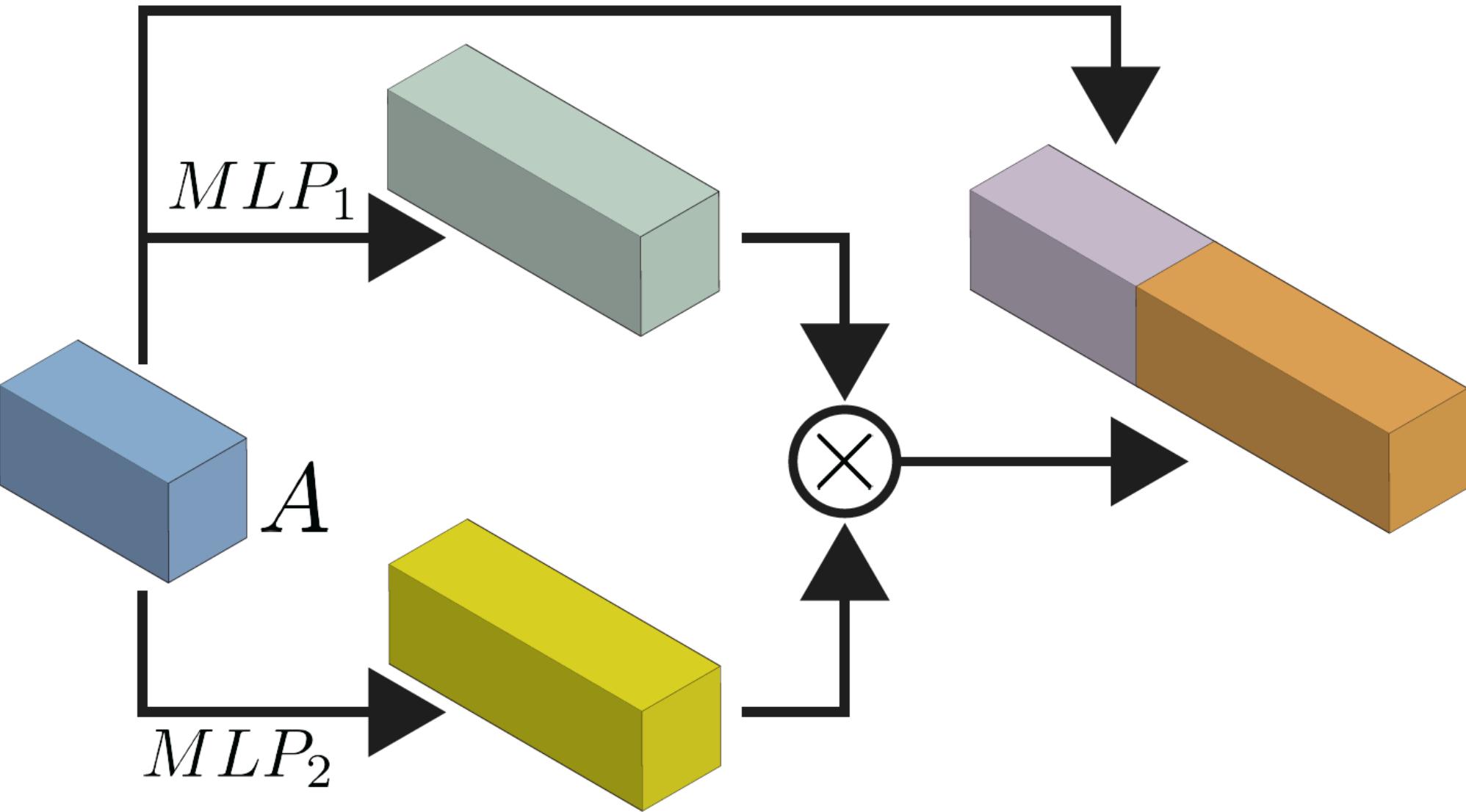
X



Y



MLP_3



```
def forward(self, input):
    M1 = self.MLP1(input)
    M2 = self.MLP2(input)
    M3 = self.MLP3(input)

    M12 = torch.matmul(M1, M2)

    return torch.cat((M3, M12), dim=1)
```

$$X \mapsto (M_3(X), M_1(X)M_2(X))$$

Equivariance?

$$X \mapsto (M_3(X), M_1(X)M_2(X))$$

Equivariance?

$$PXP^T \mapsto (M_3(PXP^T), M_1(PXP^T)M_2(PXP^T))$$

Equivariance?

$$PXP^T \mapsto (PM_3(X)P^T, PM_1(X)P^TPM_2(X)P^T)$$

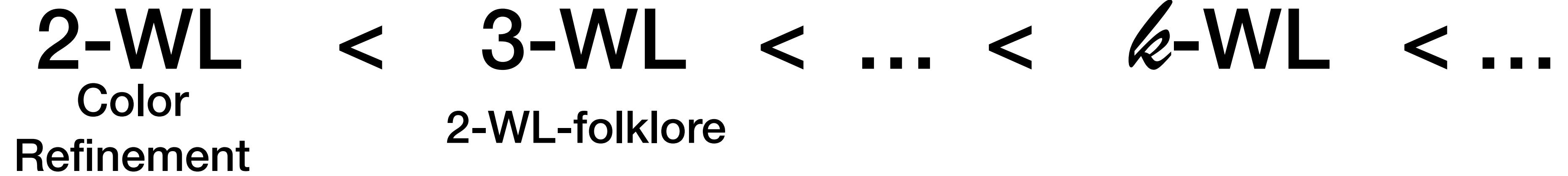
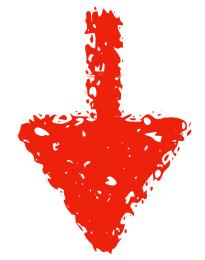
Equivariance?

$$PXP^T \mapsto (PM_3(X)P^T, PM_1(X)M_2(X)P^T)$$

Equivariance?

$$PXP^T \mapsto P(M_3(X), M_1(X)M_2(X))P^T$$

Message passing



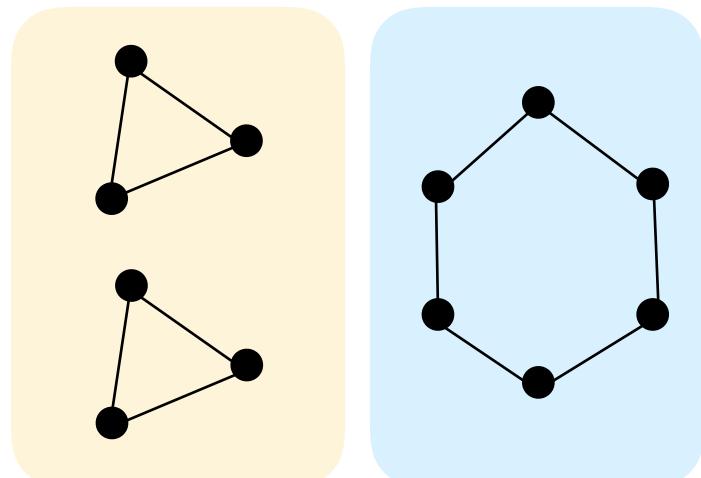
Message passing



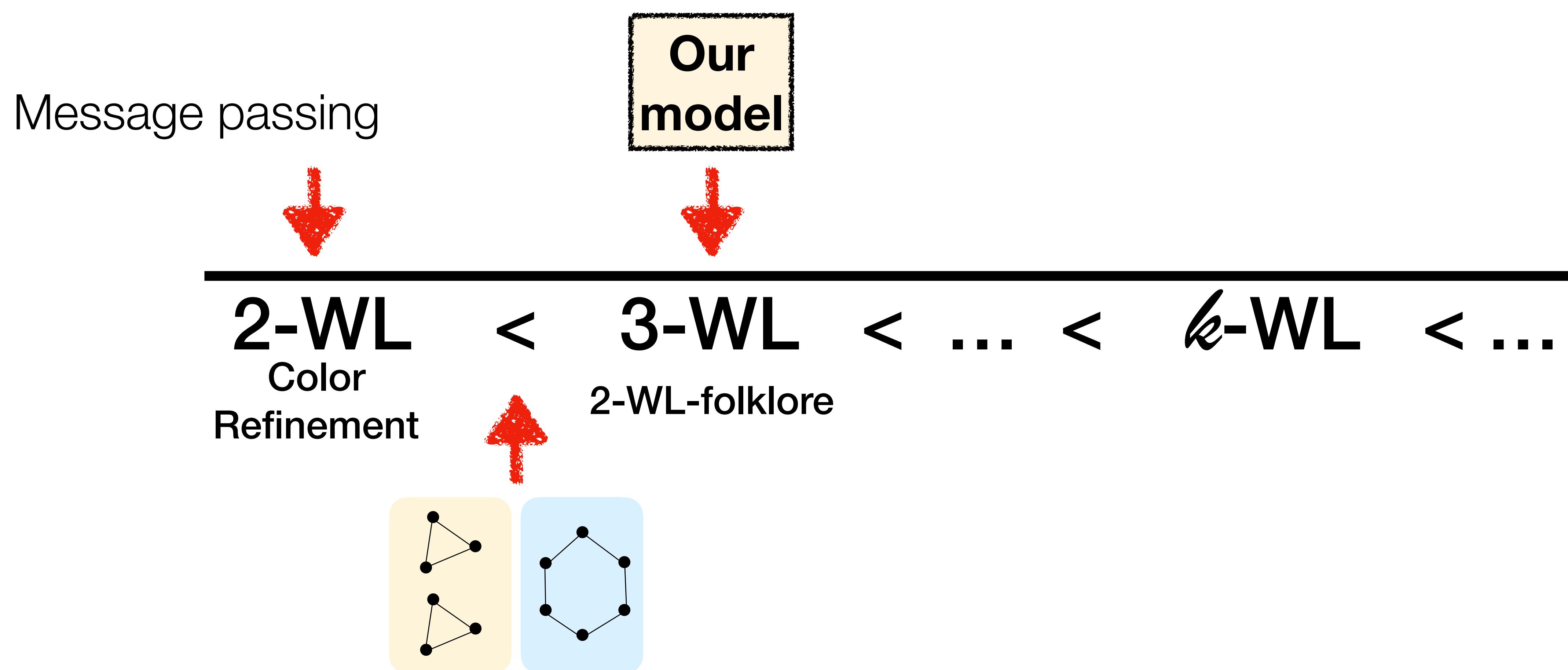
2-WL < **3-WL** < ... < ***k*-WL** < ...

Color
Refinement

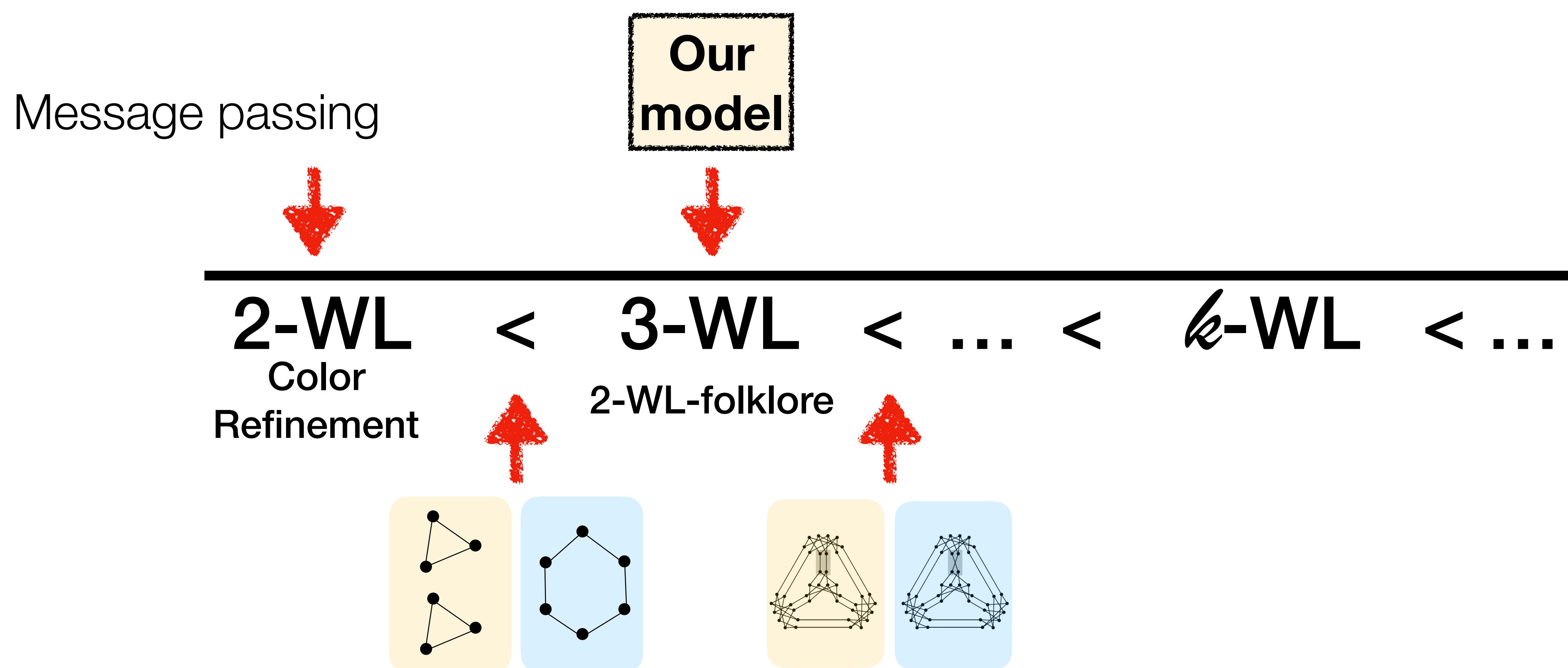
2-WL-folklore

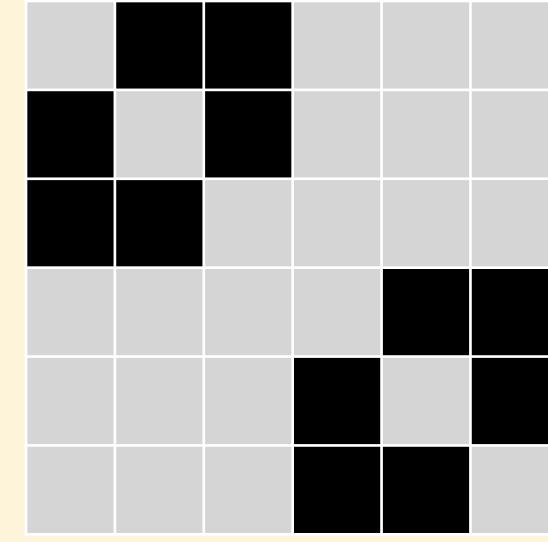
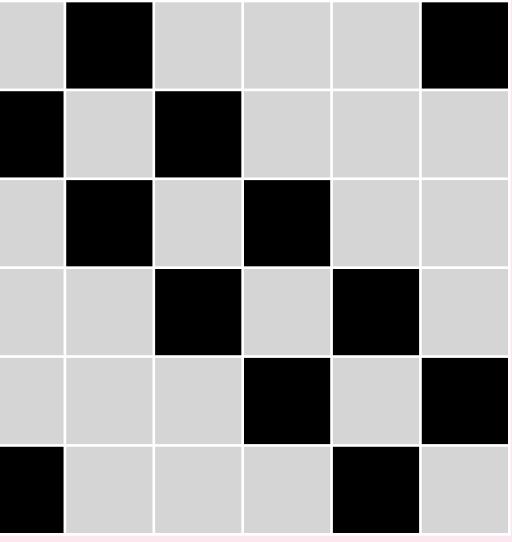
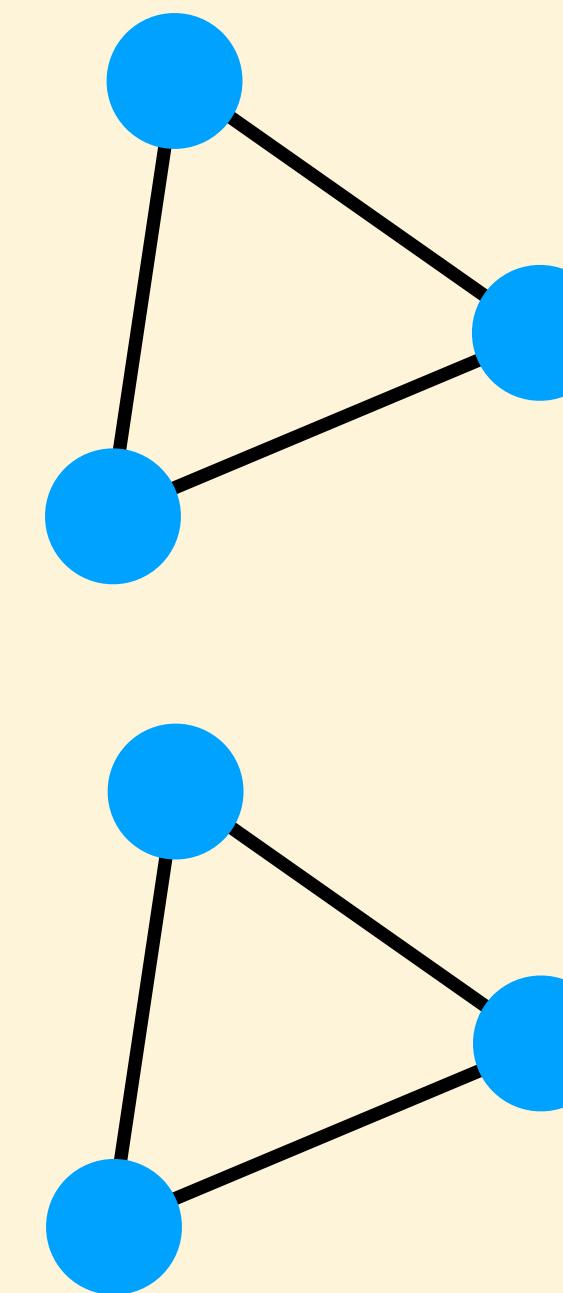
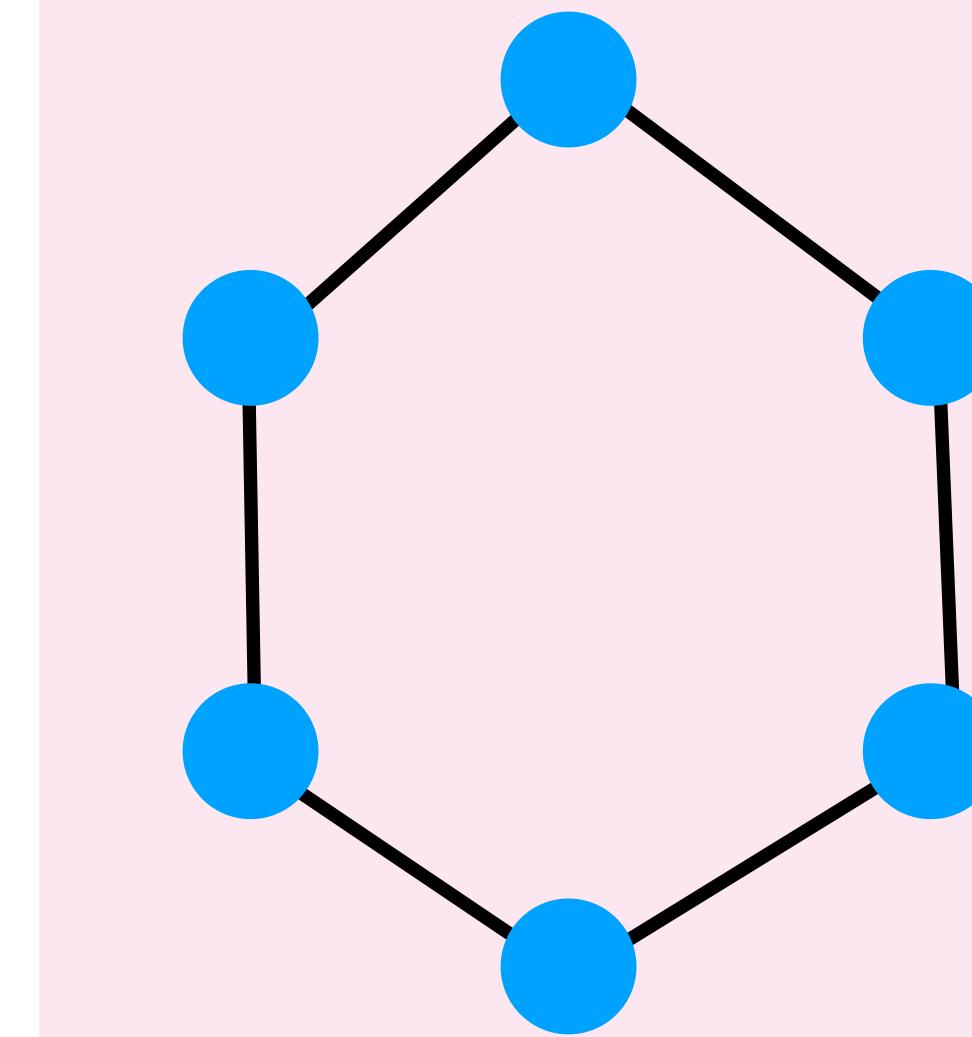


Theorem [Maron, Ben-Hamu, Serviansky, Lipman]: The simple graph network has 3-WL power in distinguishing non-isomorphic graphs.

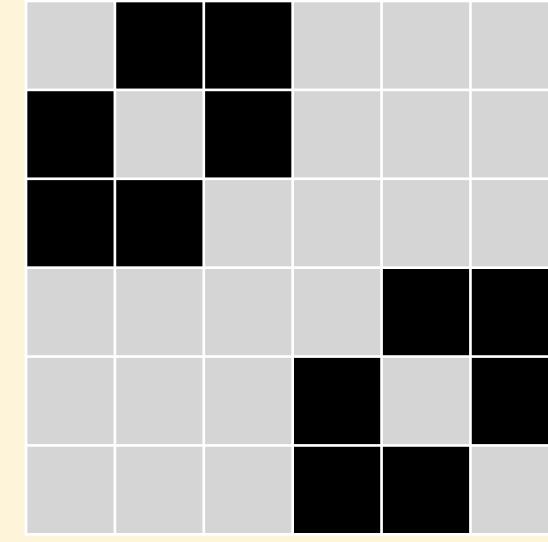
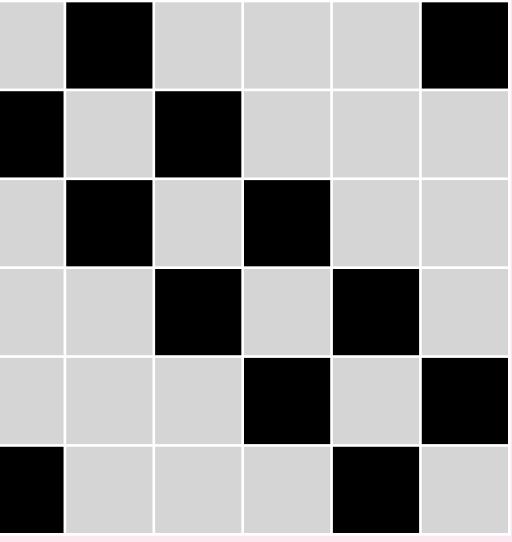
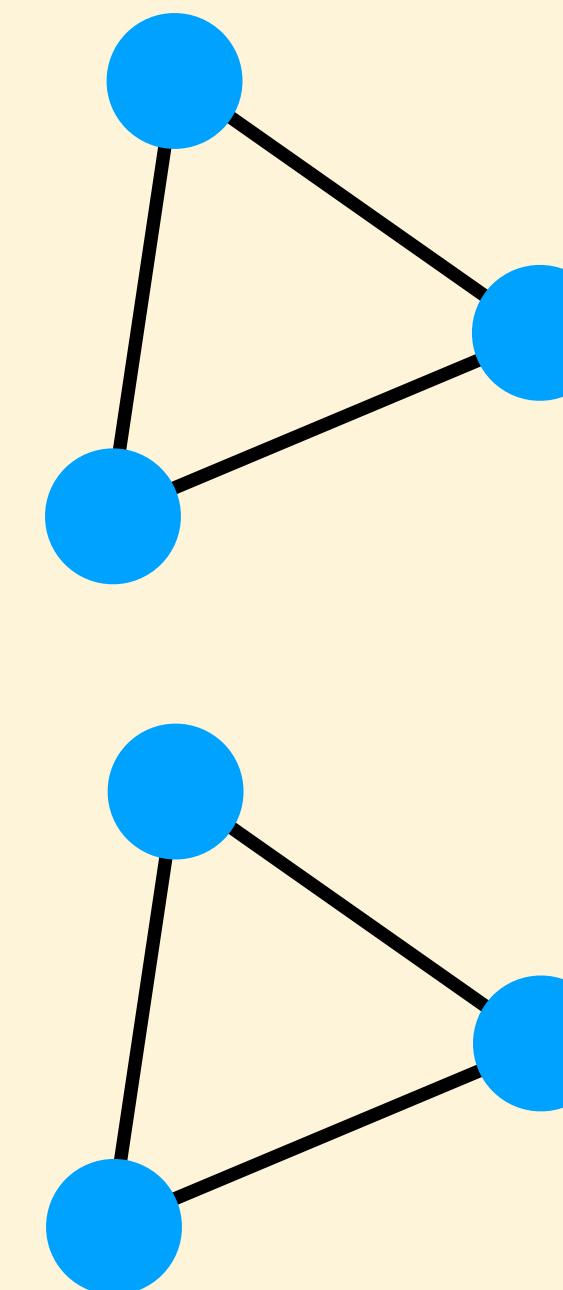
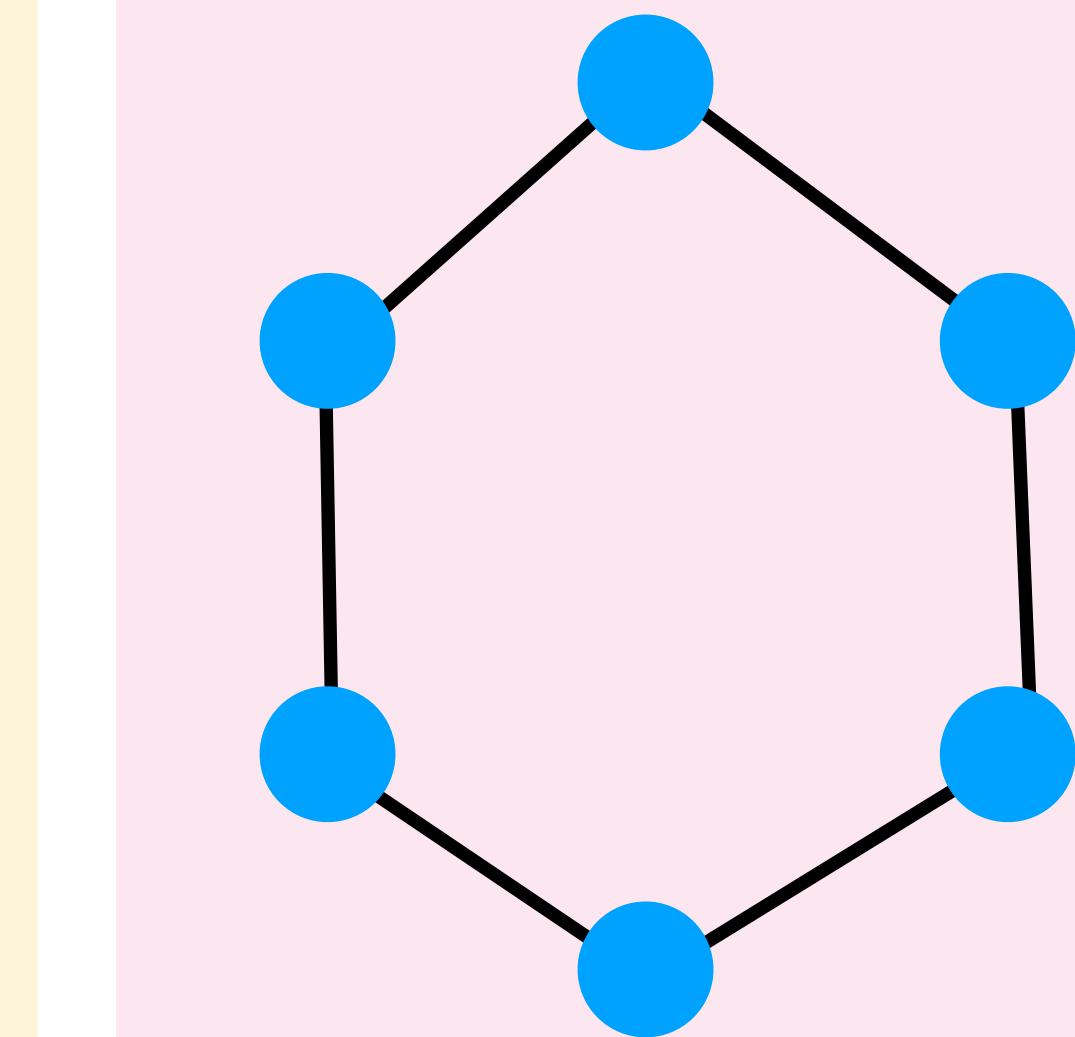


Theorem [Maron, Ben-Hamu, Serviansky, Lipman]: The simple graph network has 3-WL power in distinguishing non-isomorphic graphs.



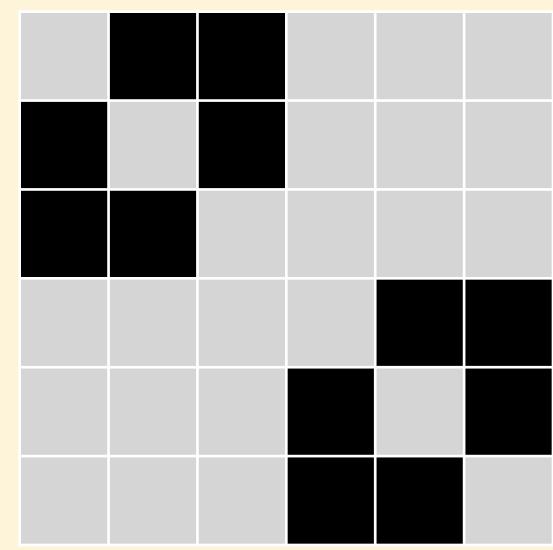
 X  X 

$$X \mapsto (M_3(X), M_1(X)M_2(X))$$

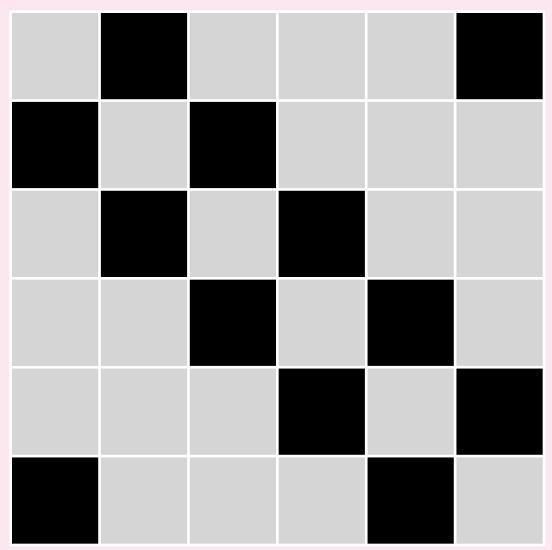
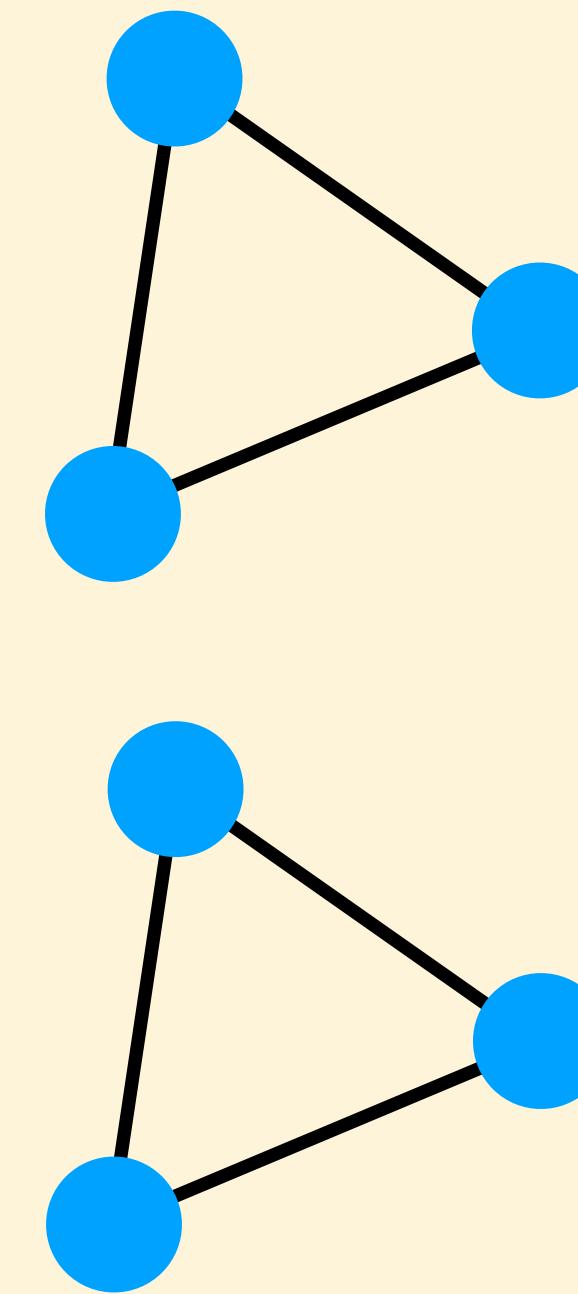
 X  X 

$$X \mapsto (M_3(X), M_1(X)M_2(X))$$

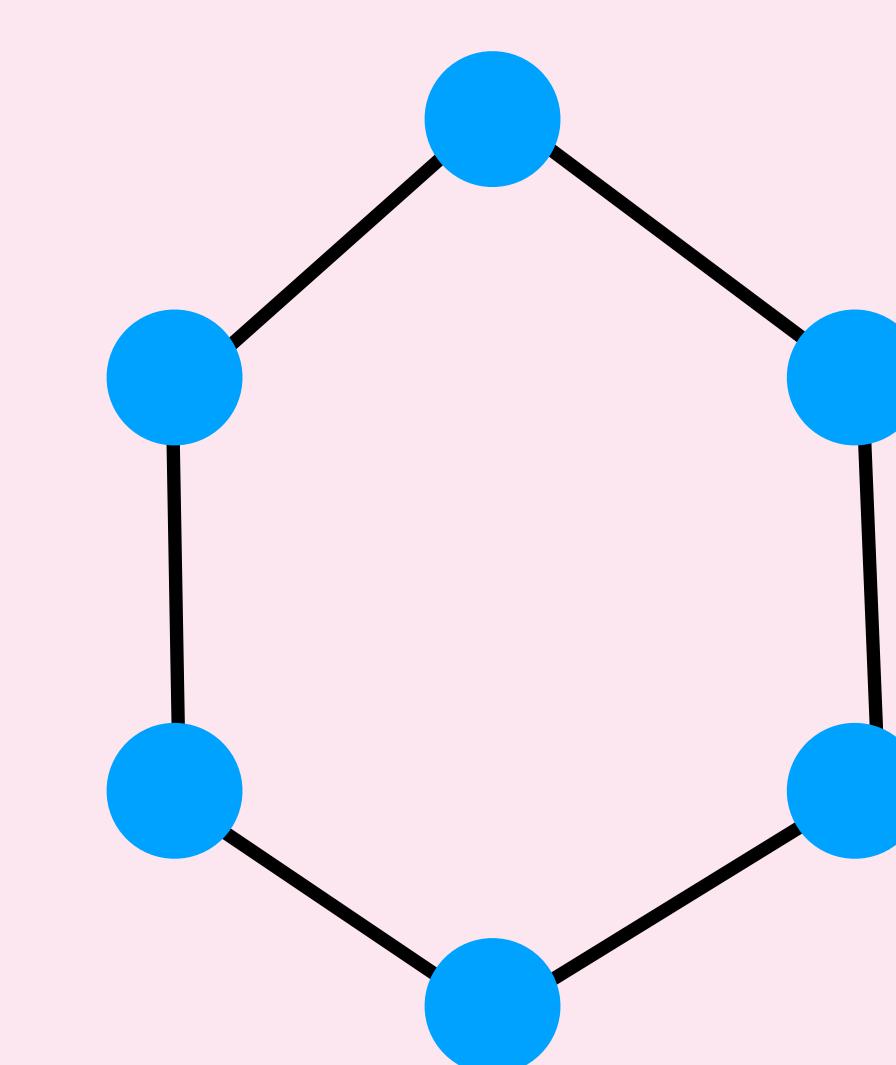
$$X \rightarrow (X, X^2) \rightarrow X^3$$



X

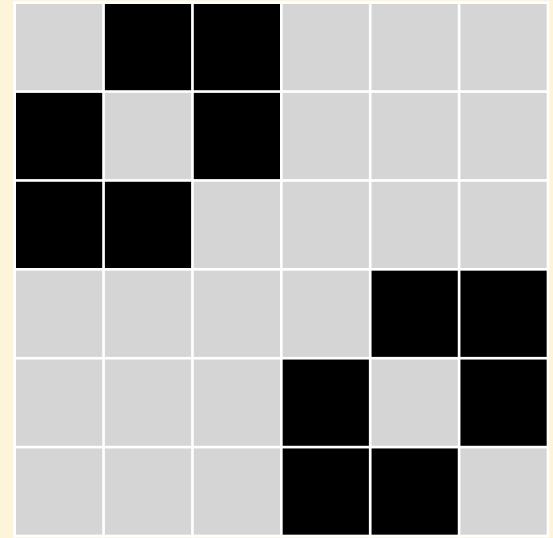


X

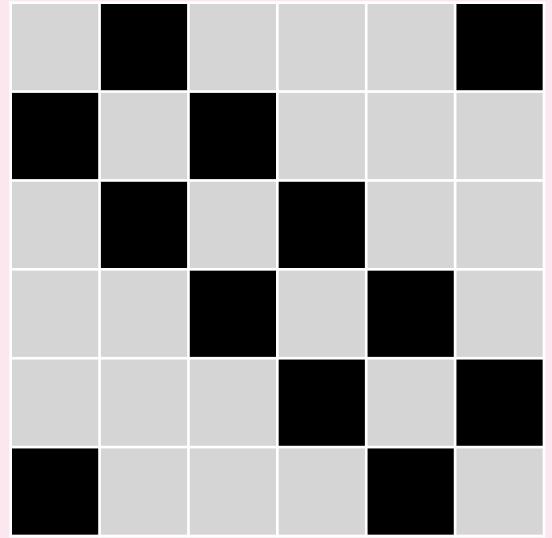
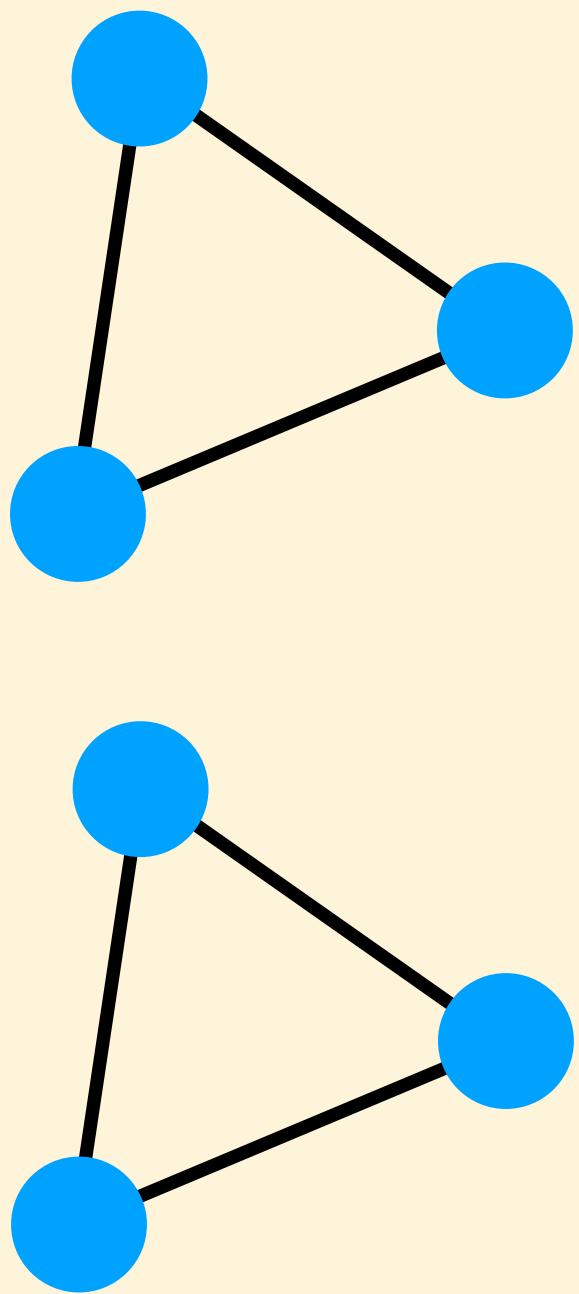


$X \rightarrow X^3$

$$(X^3)_{i,i} = 2$$



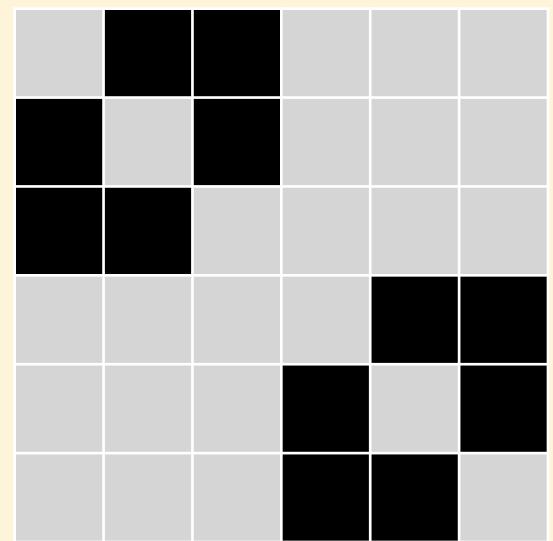
X



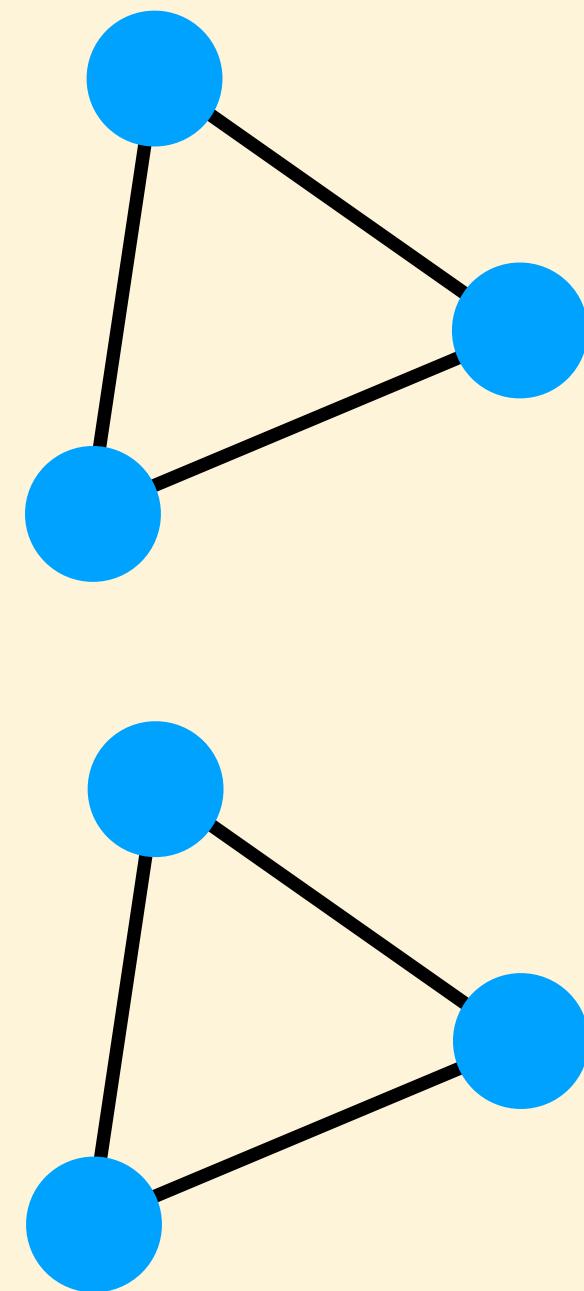
X

$$X \rightarrow X^3$$

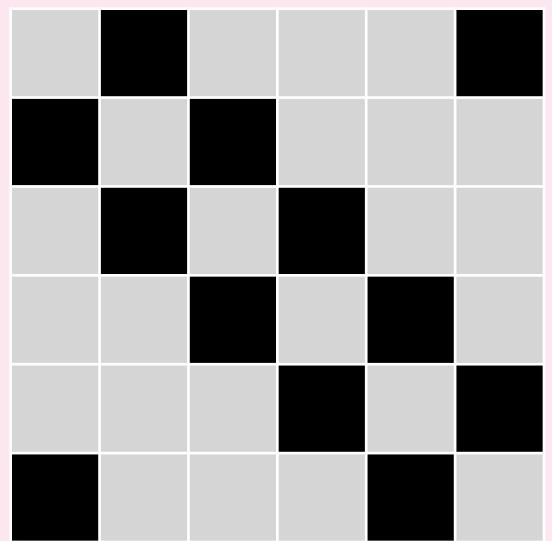
$$(X^3)_{i,i} = 2$$



X



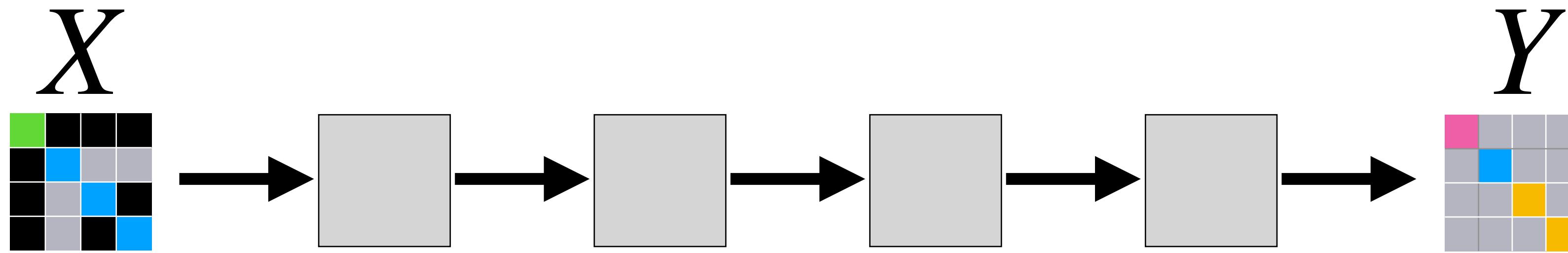
$$(X^3)_{i,i} = 0$$



X

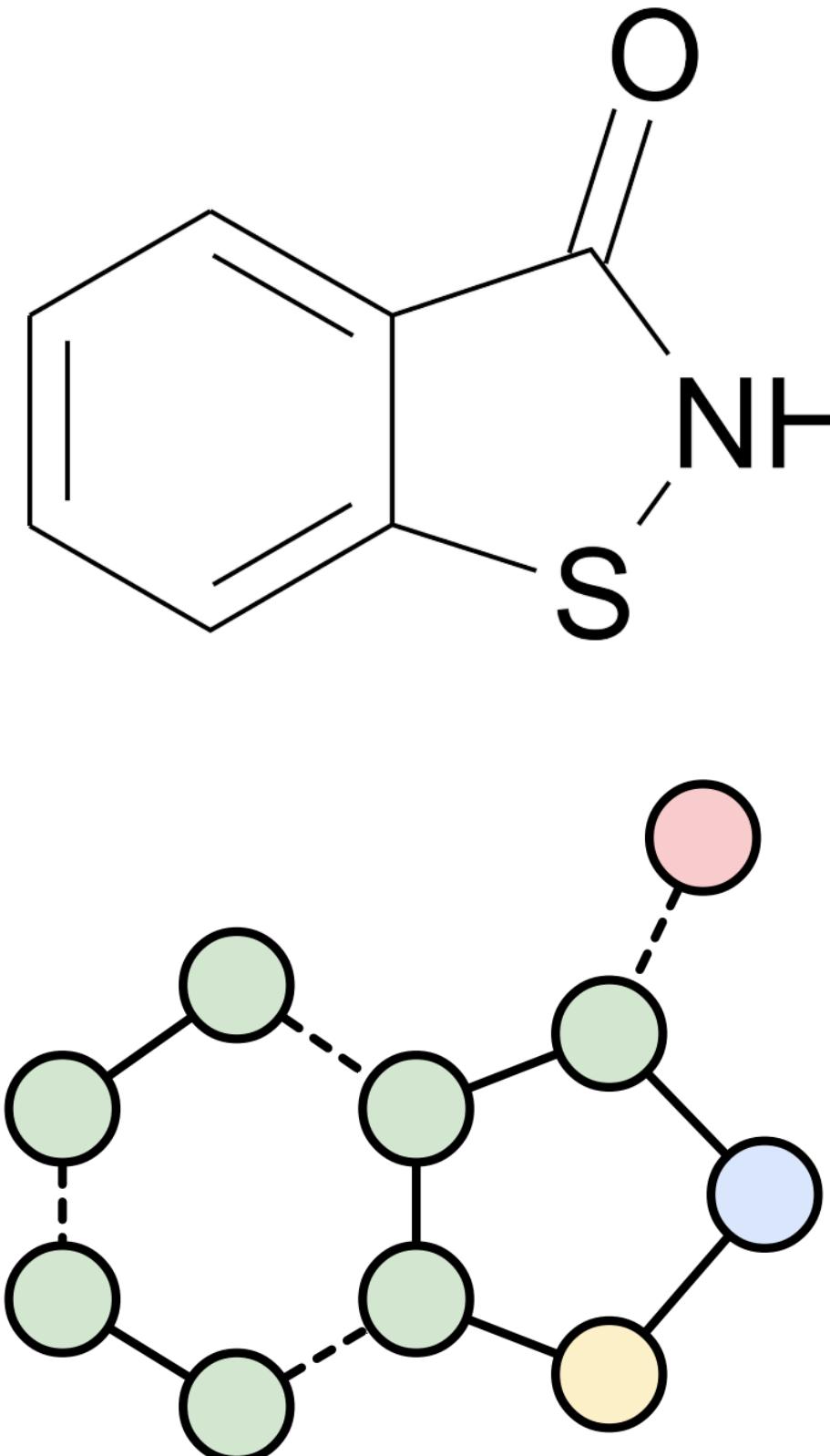
$$X \rightarrow X^3$$

Limitations



- Works with full matrix representation
 - Suitable for **dense small graphs**
 - Not applicable for **large sparse graphs**

Molecule dataset (QM9)



Target	DTNN	MPNN	123-gnn	Ours 1	Ours 2
μ	0.244	0.358	0.476	0.231	0.0934
α	0.95	0.89	0.27	0.382	0.318
ϵ_{homo}	0.00388	0.00541	0.00337	0.00276	0.00174
ϵ_{lumo}	0.00512	0.00623	0.00351	0.00287	0.0021
Δ_ϵ	0.0112	0.0066	0.0048	0.00406	0.0029
$\langle R^2 \rangle$	17	28.5	22.9	16.07	3.78
ZPVE	0.00172	0.00216	0.00019	0.00064	0.000399
U_0	2.43	2.05	0.0427	0.234	0.022
U	2.43	2	0.111	0.234	0.0504
H	2.43	2.02	0.0419	0.229	0.0294
G	2.43	2.02	0.0469	0.238	0.024
C_v	0.27	0.42	0.0944	0.184	0.144

Conclusion

Efficiency

Color
Refinement = $2\text{-WL} \prec 3\text{-WL} \prec \dots \prec k\text{-WL} \prec \dots$

Power

Conclusion

Efficiency

Color
Refinement = $2\text{-WL} < 3\text{-WL} < \dots < k\text{-WL} < \dots$

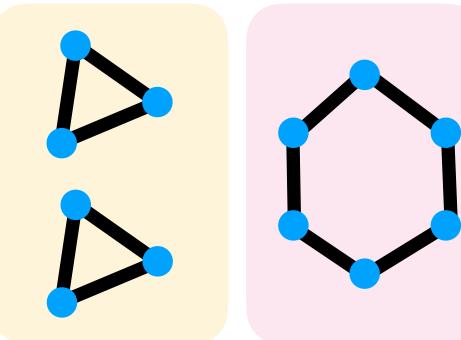
Power



Message passing

[Morris et al.18', Xu et al.19']

Conclusion



Color Refinement = $2\text{-WL} < 3\text{-WL} < \dots < k\text{-WL} < \dots$



Message passing

[Morris et al.18', Xu et al.19']

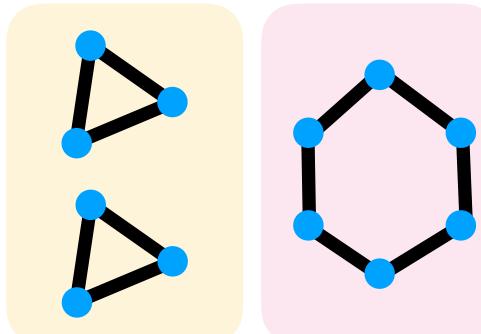
Power

Conclusion

Efficiency

Color Refinement = $2\text{-WL} < 3\text{-WL} < \dots < k\text{-WL} < \dots$

Power



Message passing
[Morris et al.18', Xu et al.19']



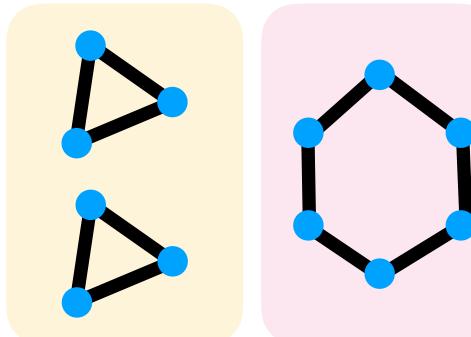
Our
model

Conclusion

Efficiency

Color Refinement = $2\text{-WL} < 3\text{-WL} < \dots < k\text{-WL} < \dots$

Power



Message passing
[Morris et al.18', Xu et al.19']



Our model



ϵ -IGN
Theorem
[Maron, Ben-Hamu, Serviansky, Lipman]

Blog:
irregulardeep.org

Code is online!

Paper:
Provably Powerful Graph Networks (NeurIPS 2019)

Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, Yaron Lipman

Funding:
ERC Consolidator Grant 771136 (“LiftMatch”)
ISRAEL SCIENCE FOUNDATION (grant No. ISF 1830/17)

