GRAPH NEURAL NETWORKS & ROTATIONAL EQUIVARIANCE

University of California, Berkeley Fall 2023, CS 189/289A: Introduction to Machine Learning

Rasmus Malik Hoeegh Lindrup

Postdoc, BAIR/ICSI

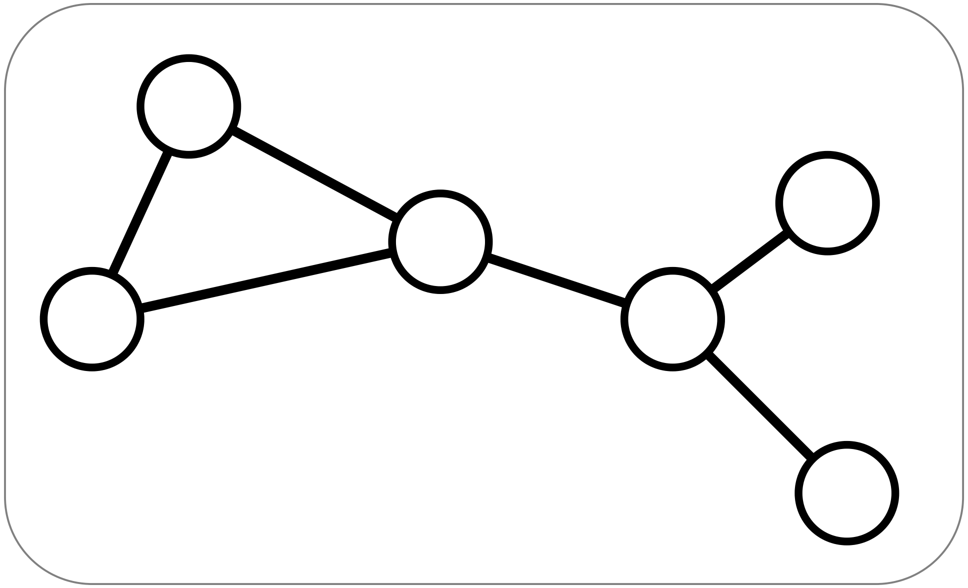
Building on slides originally made by Daniel Rothchild

OUTLINE

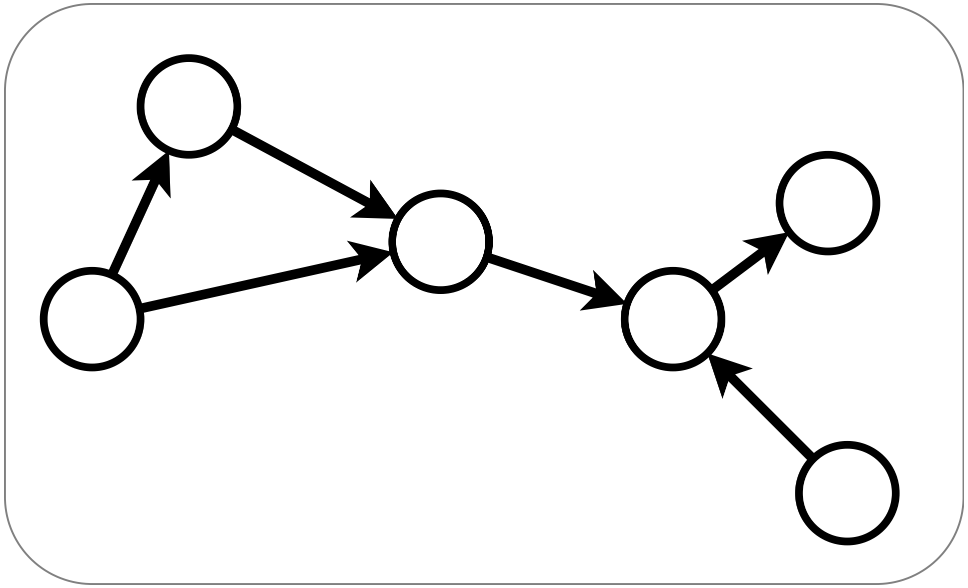
- Lecture 1
  - Graph data
  - Graph tasks
  - Invariance and equivariance
  - Message passing
- Lecture 2
  - Rotational equivariance
  - Equivariant neural networks

GRAPH DATA

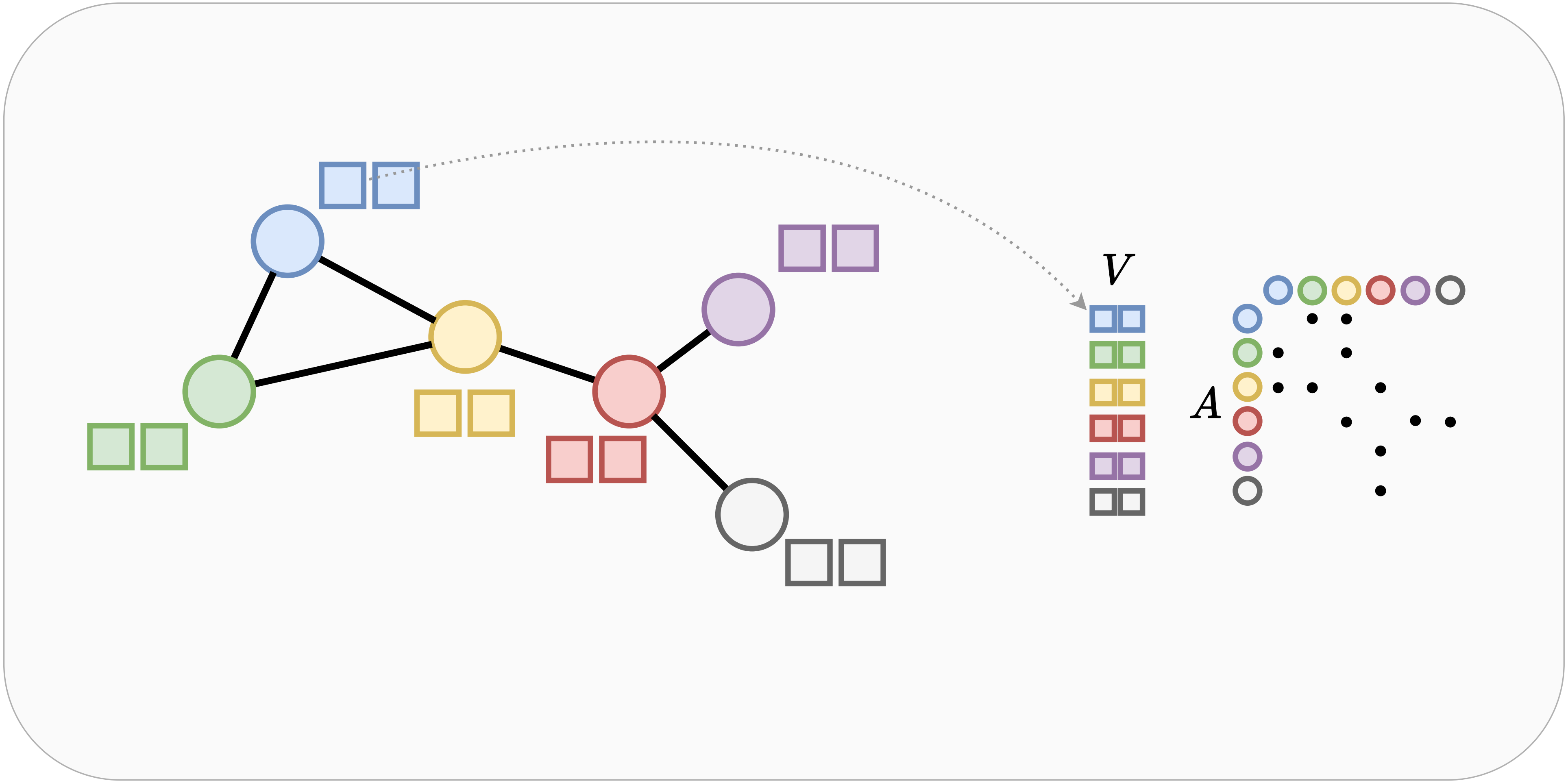
GRAPH



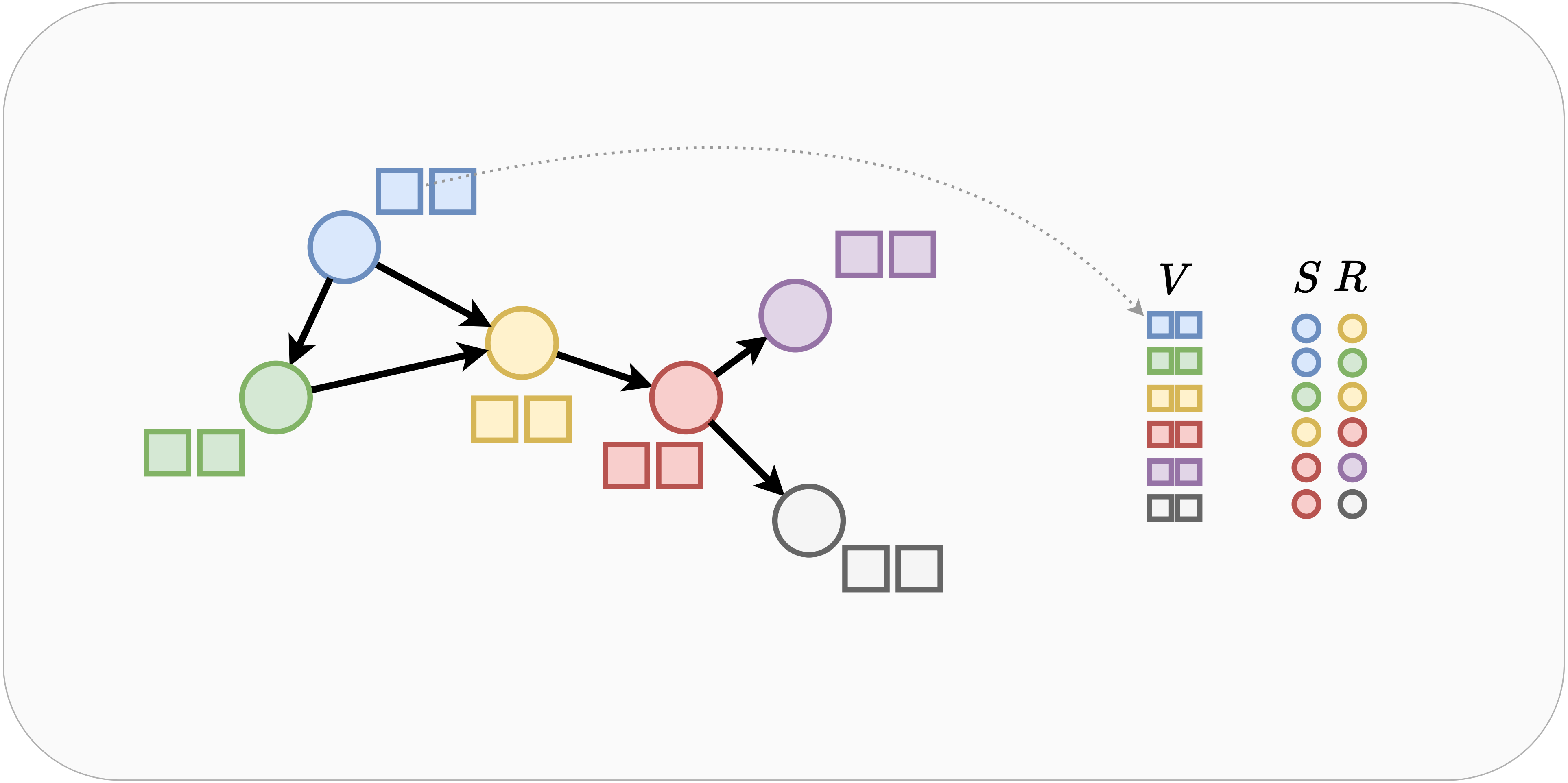
GRAPH



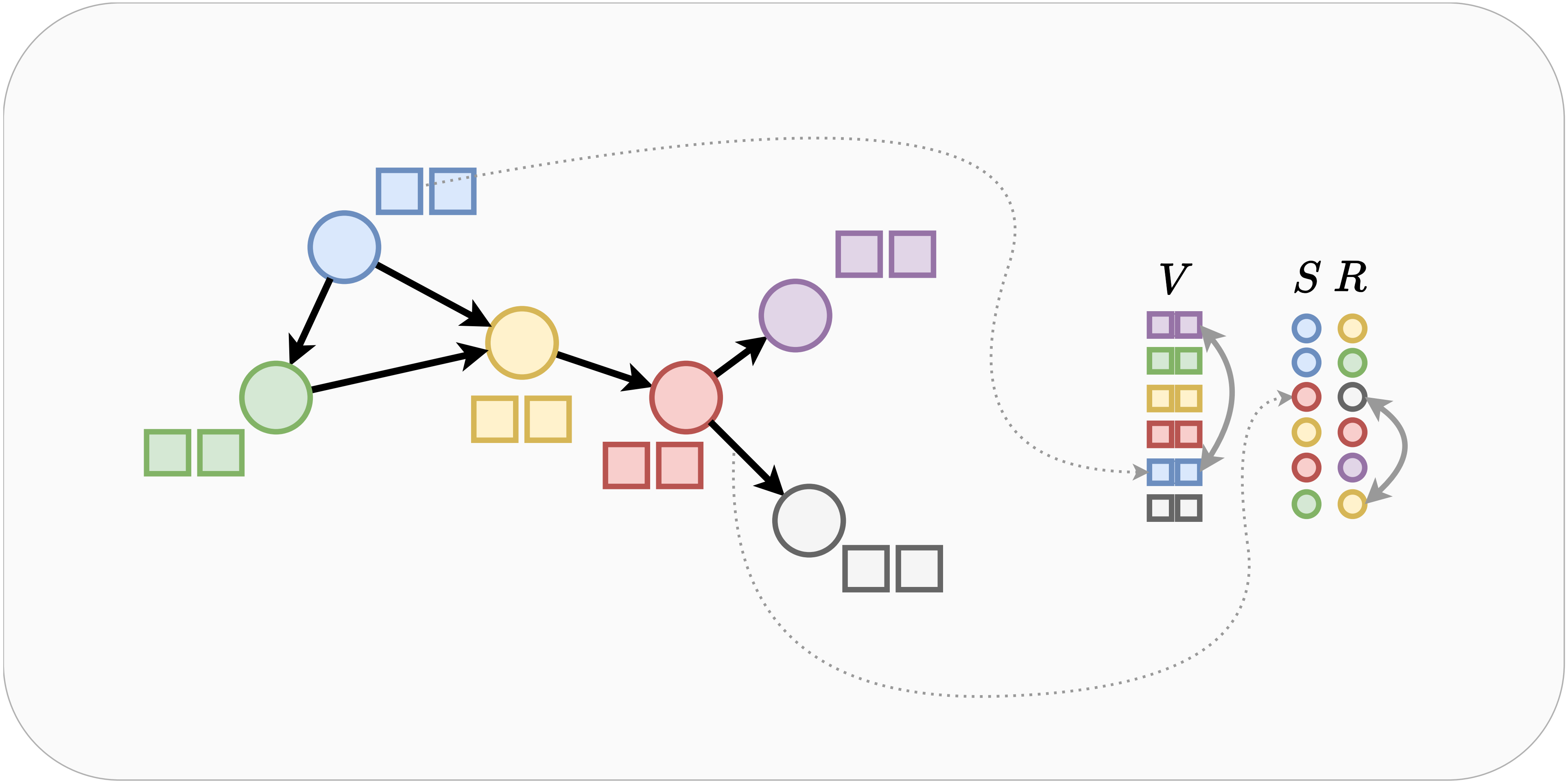
REPRESENTING A GRAPH



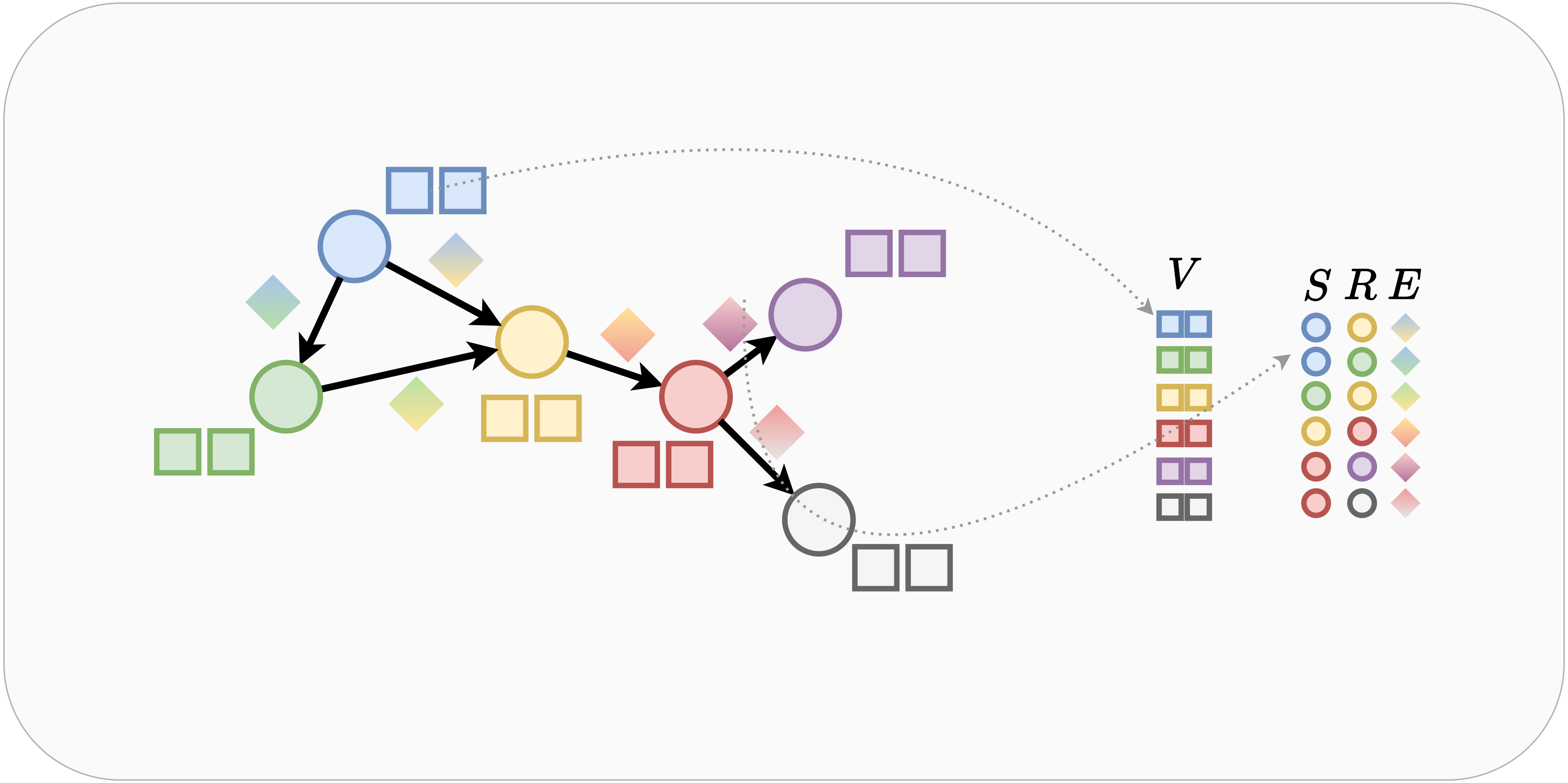
REPRESENTING A GRAPH



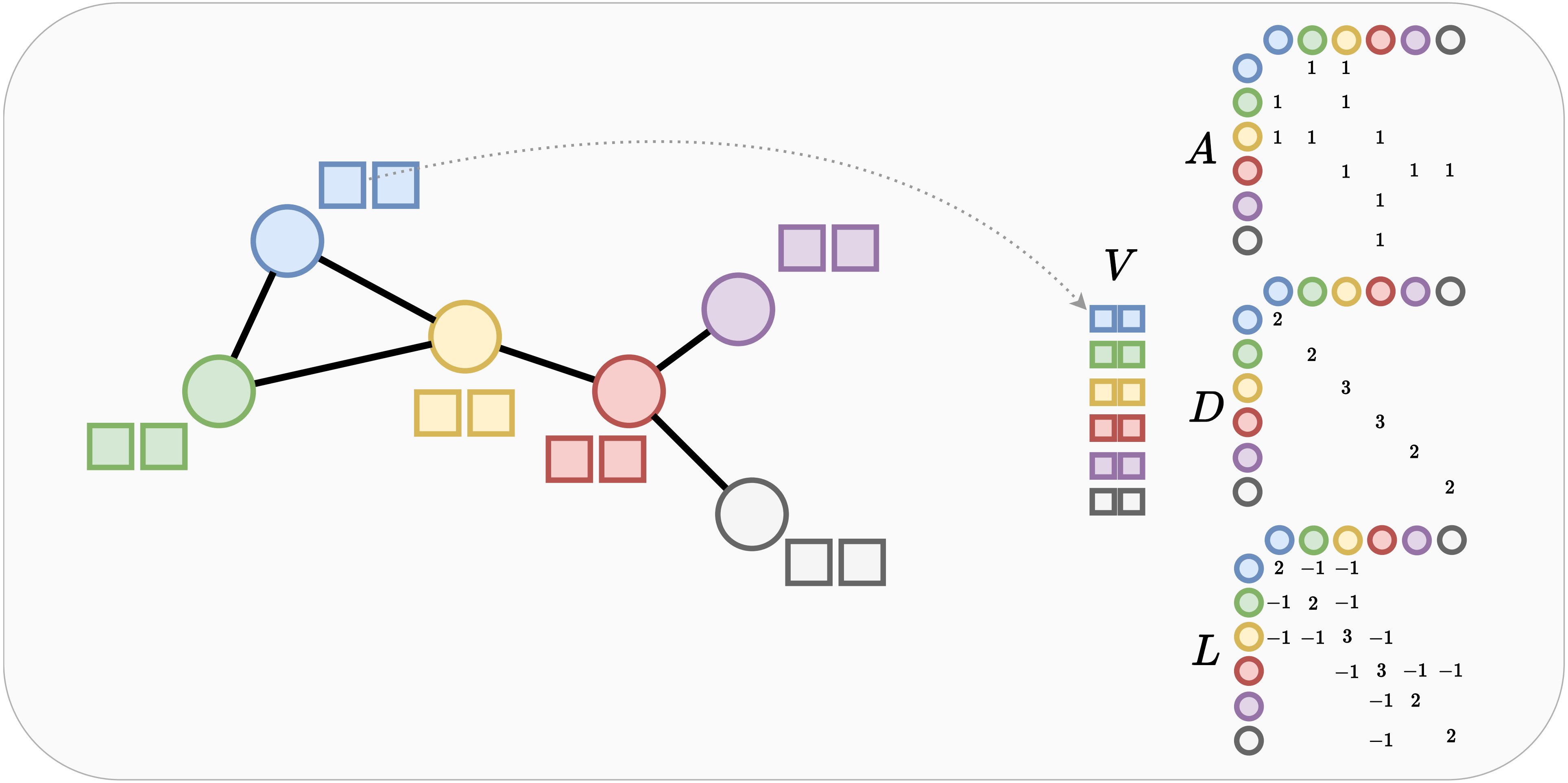
REPRESENTING A GRAPH



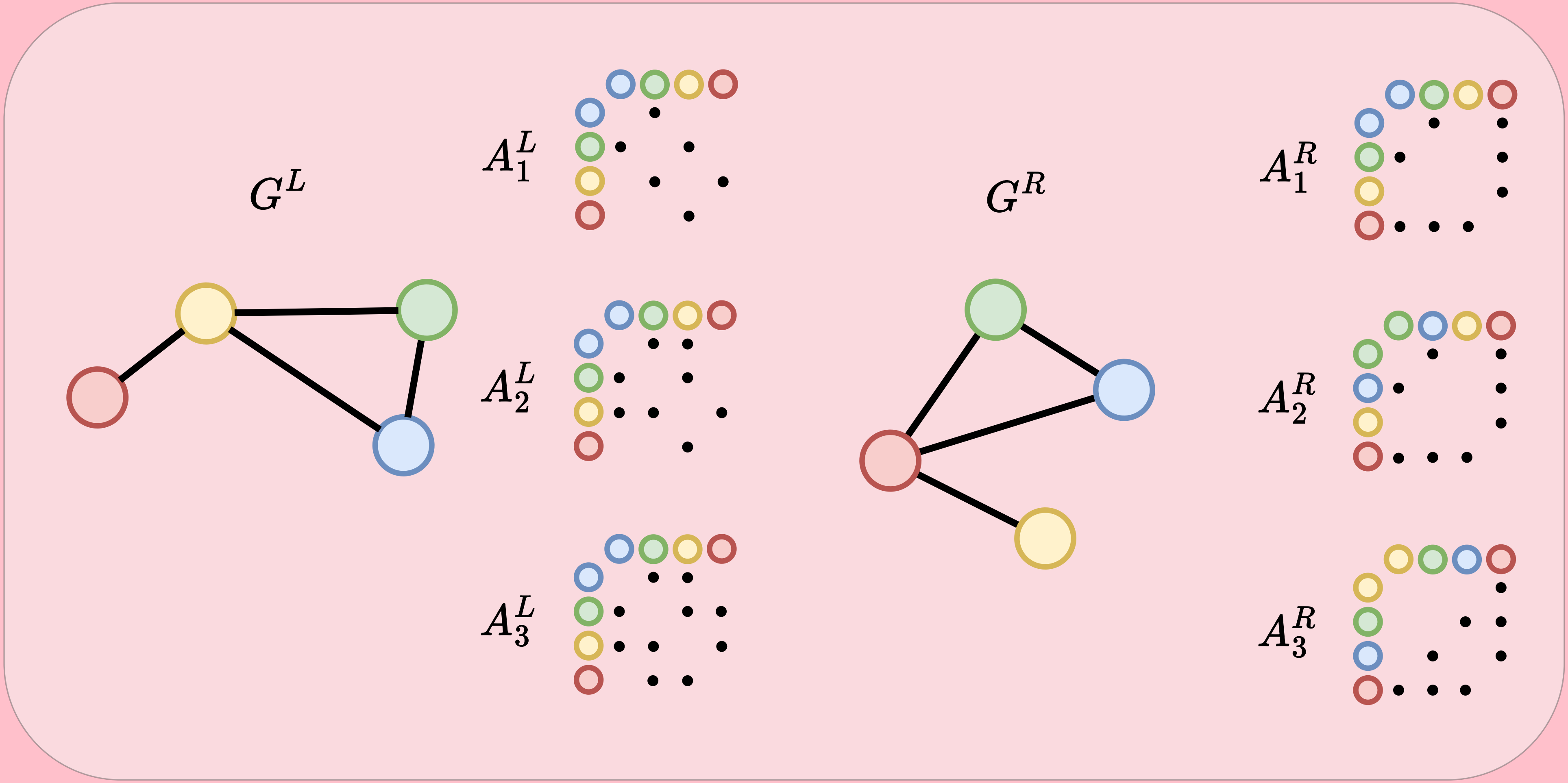
REPRESENTING A GRAPH



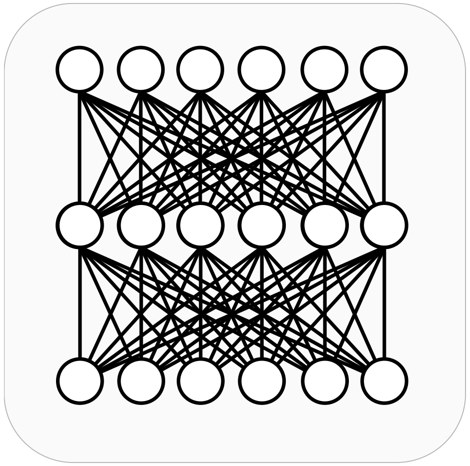
REPRESENTING A GRAPH

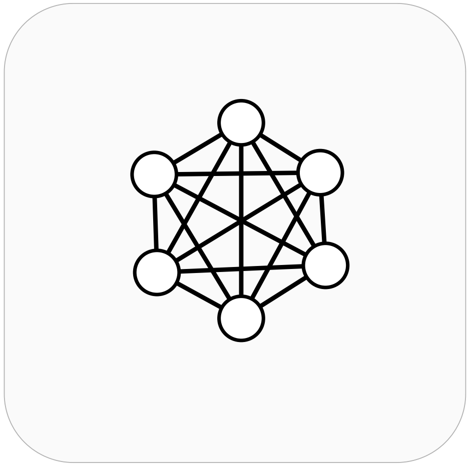


WHICH GRAPH CORRESPONDS TO THESE REPRESENTATIONS?

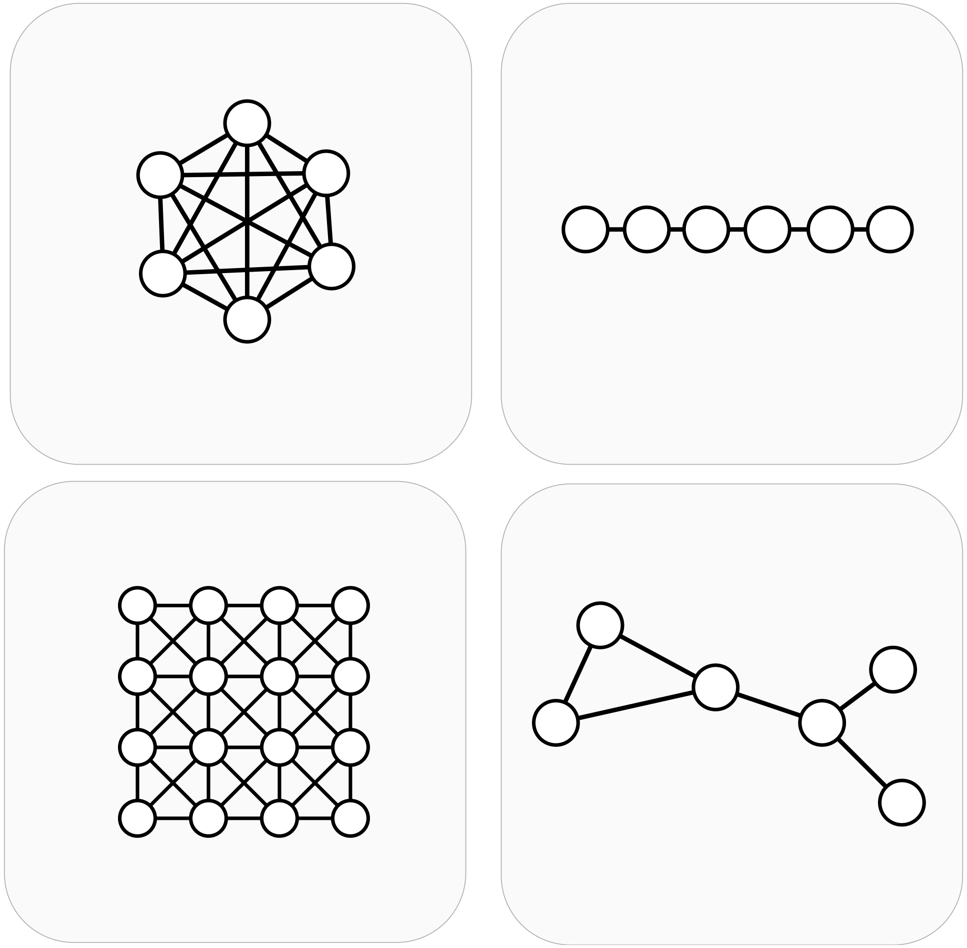


COMMON ARCHITECHTURES

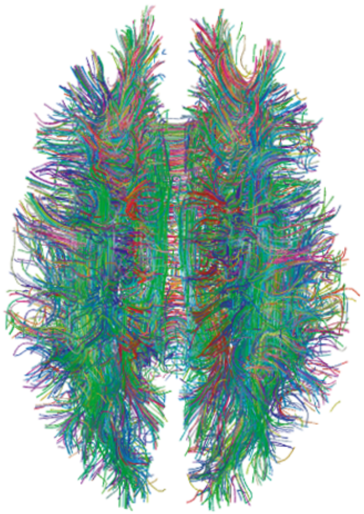


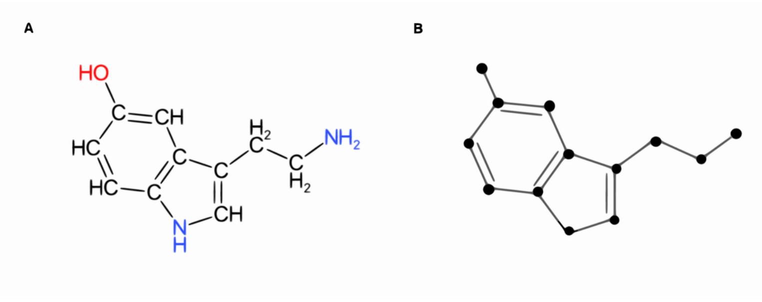


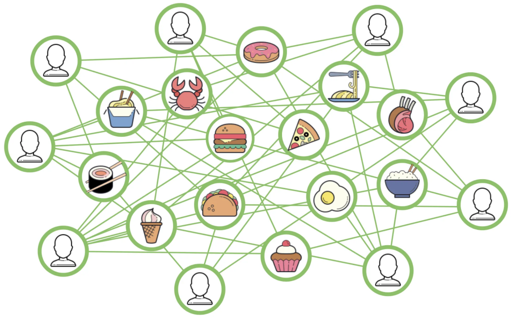
COMMON ARCHITECHTURES

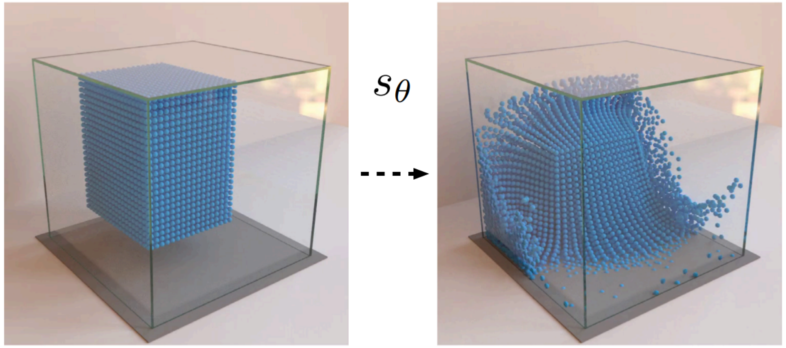


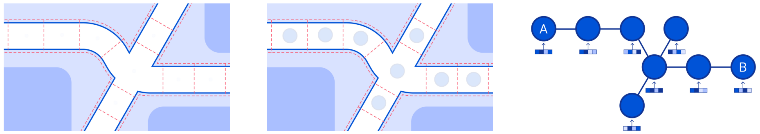
GRAPHS







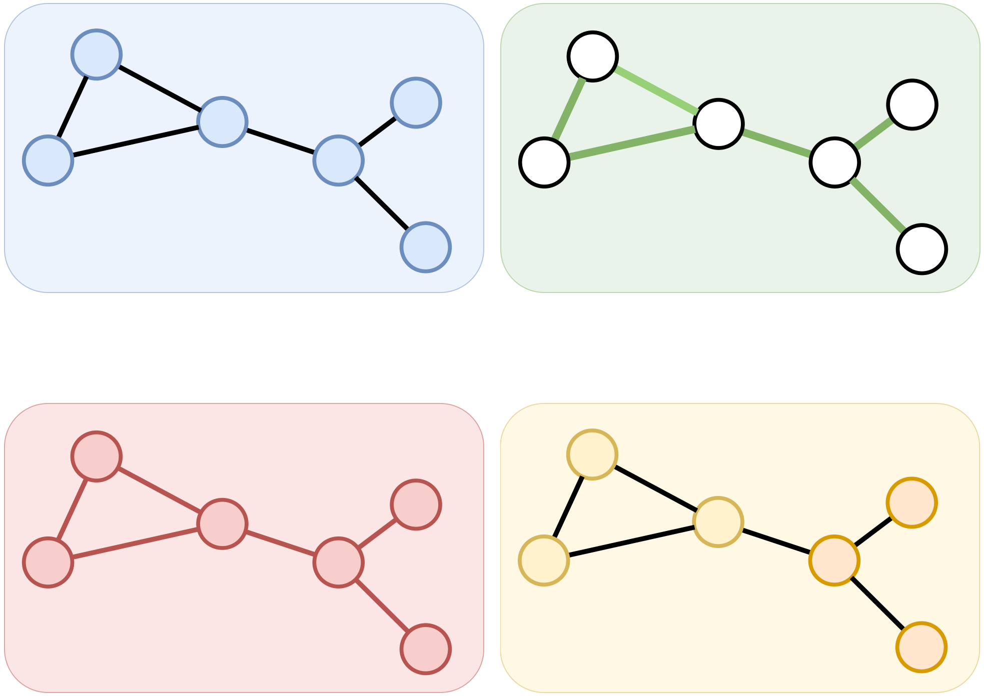




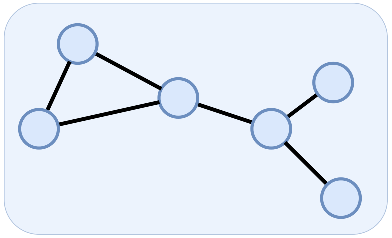
(Connectome: Gigandet X, Hagmann P, Kurant M, Cammoun L, Meuli R, et al. (2008) Estimating the Confidence Level of White Matter Connections Obtained with MRI Tractography. PLoS ONE 3(12): e4006. doi:10.1371/journal.pone.0004006.)
(Particles: Sanchez-Gonzalez, Alvaro, et al. "Learning to simulate complex physics with graph networks." International conference on machine learning. PMLR, 2020.)
(Food social network: https://www.uber.com/blog/uber-eats-graph-learning/)
(Serotonin: Yirik MA, Steinbeck C (2021) Chemical graph generators. PLoS Comput Biol 17(1): e1008504. https://doi.org/10.1371/journal.pcbi.1008504)
(Traffic: Derrow-Pinion, Austin, et al. "ETA prediction with Graph Neural Networks in Google Maps" Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 2021.)

GRAPH TASKS

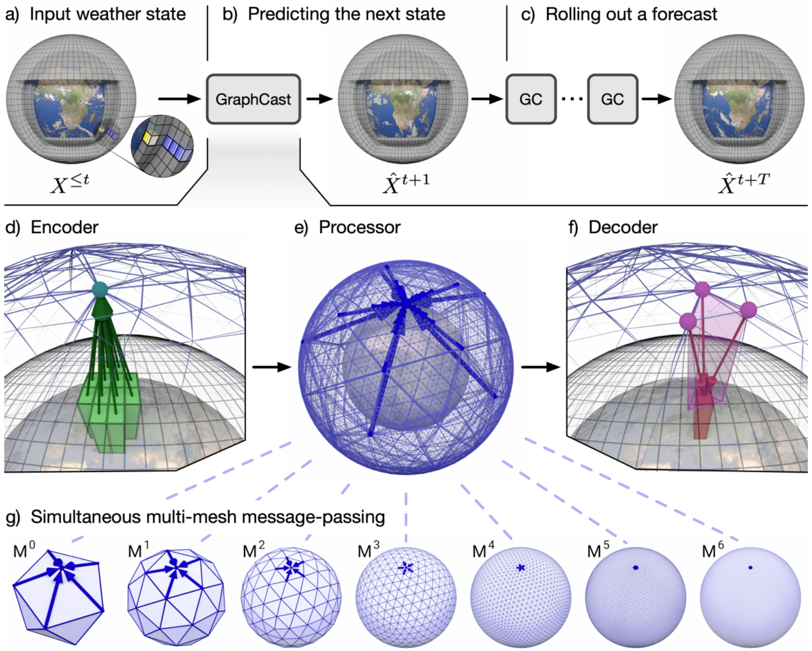
CONCEPTUAL GRAPH TASKS



NODE-LEVEL: WEATHER FORECASTING

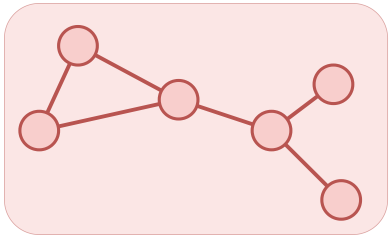


- Data: atmospheric variables like temperatures, wind speeds, pressures, etc., at diffent times, various longitude/latitudes, and levels in the atmosphere
- Represent the global weather state as a graph, model long-range dependencies with multi-mesh
- Task: predict future states of the graph, predicting future node features

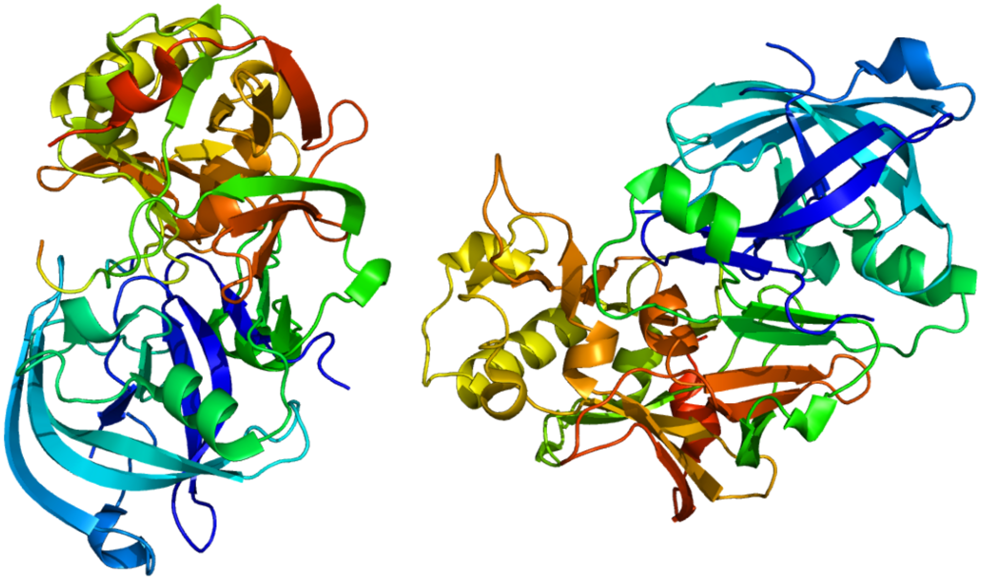


(Lam, Remi, et al. "GraphCast: Learning skillful medium-range global weather forecasting." arXiv preprint arXiv:2212.12794 (2022).)

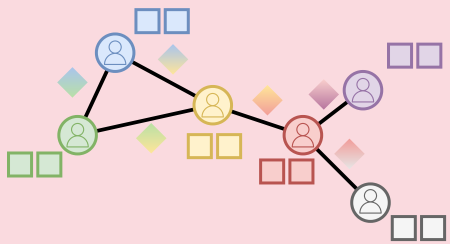
DRUG DISCOVERY: GRAPH-LEVEL



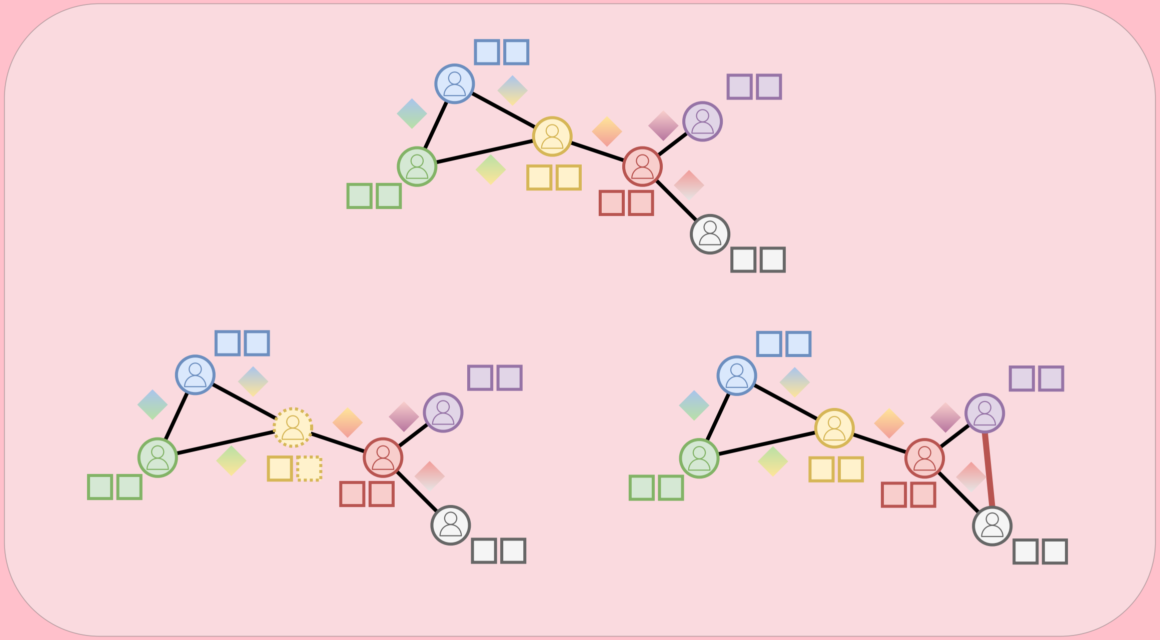
- Alzheimer's disease, Amyloid Beta plaques, beta-secretase 1 protein (BACE1)
- Inhibit BACE1: good candidate for an Alzheimer's drug
- Task: given graph predict BACE1 inhibition (IC50)

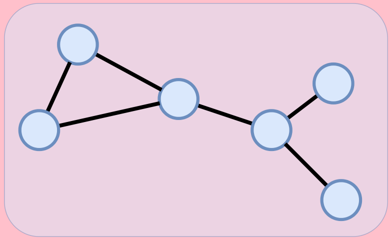


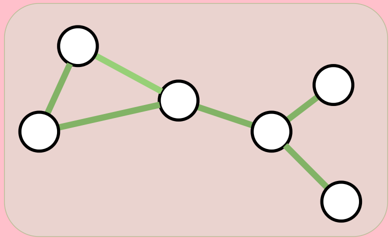
DISCUSSION: GRAPH-, NODE-, OR EDGE-LEVEL?



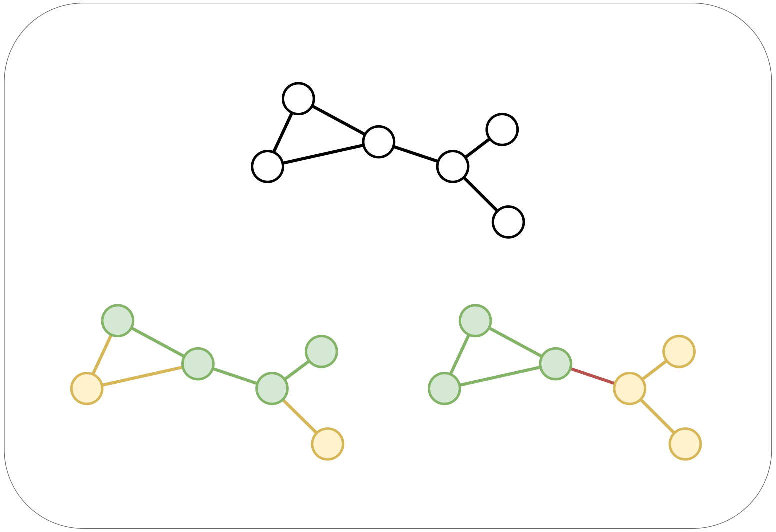
DISCUSSION: GRAPH-, NODE-, OR EDGE-LEVEL?

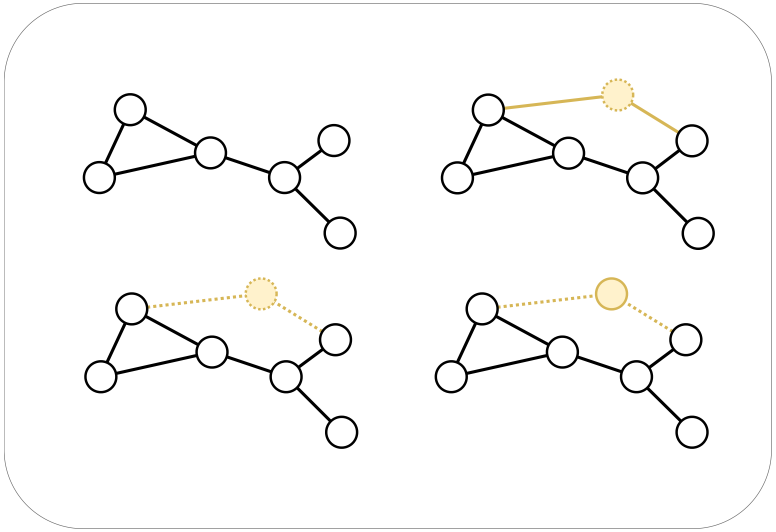






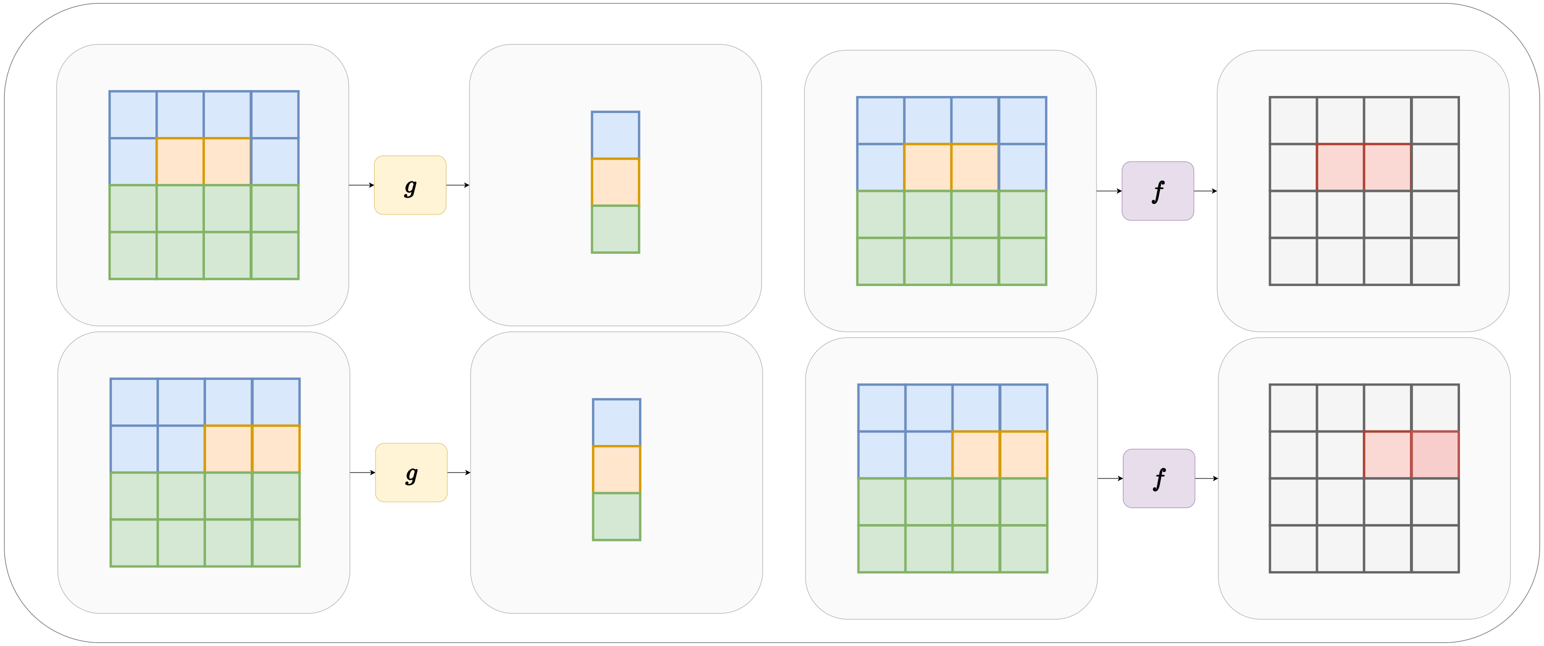
EVALUATION



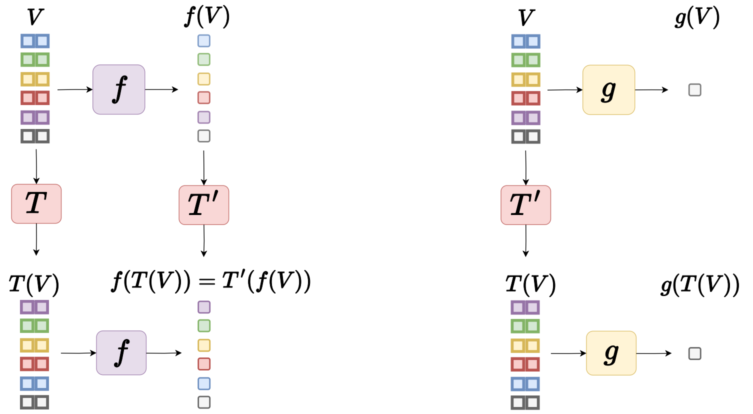


INVARIANCE AND EQUIVARIANCES

TRANSLATIONAL INVARIANCE AND EQUIVARIANCE



PERMUTATION INVARIANCE AND EQUIVARIANCE

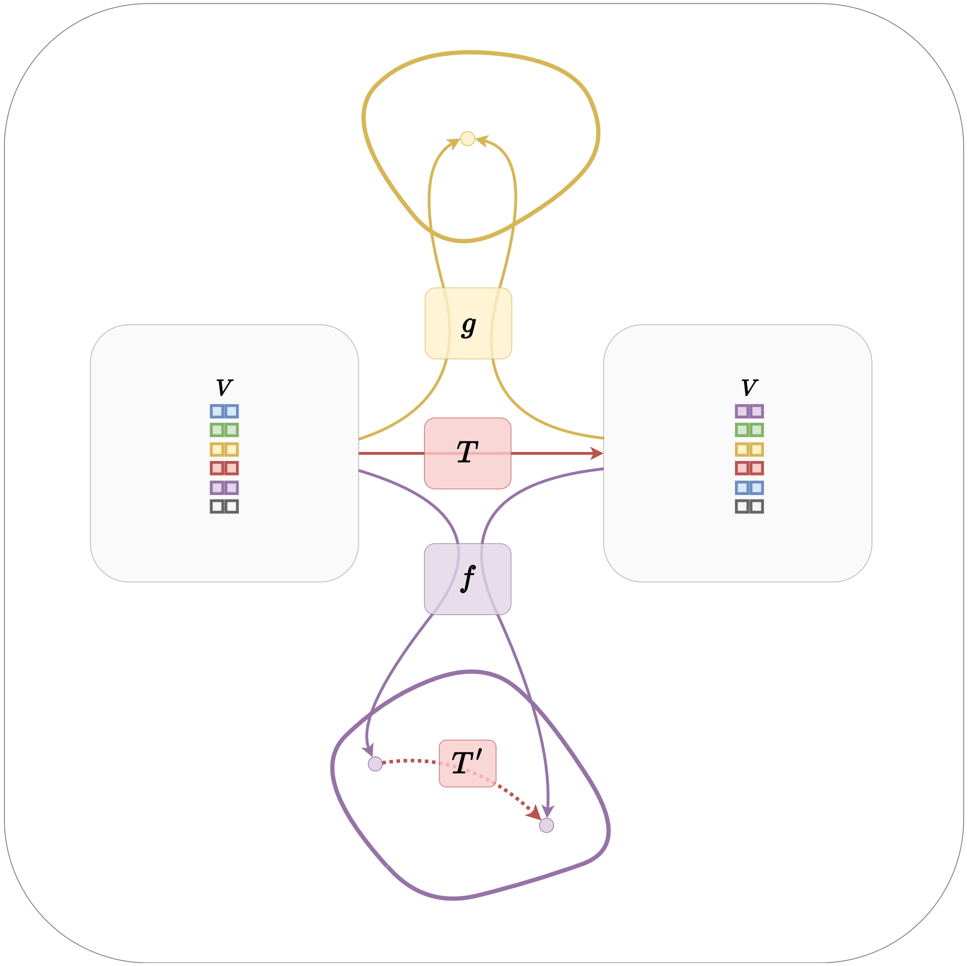


PERMUTATION INVARIANCE AND EQUIVARIANCE

$$g(\mathbf{PAP}^\top) = g(\mathbf{A})$$

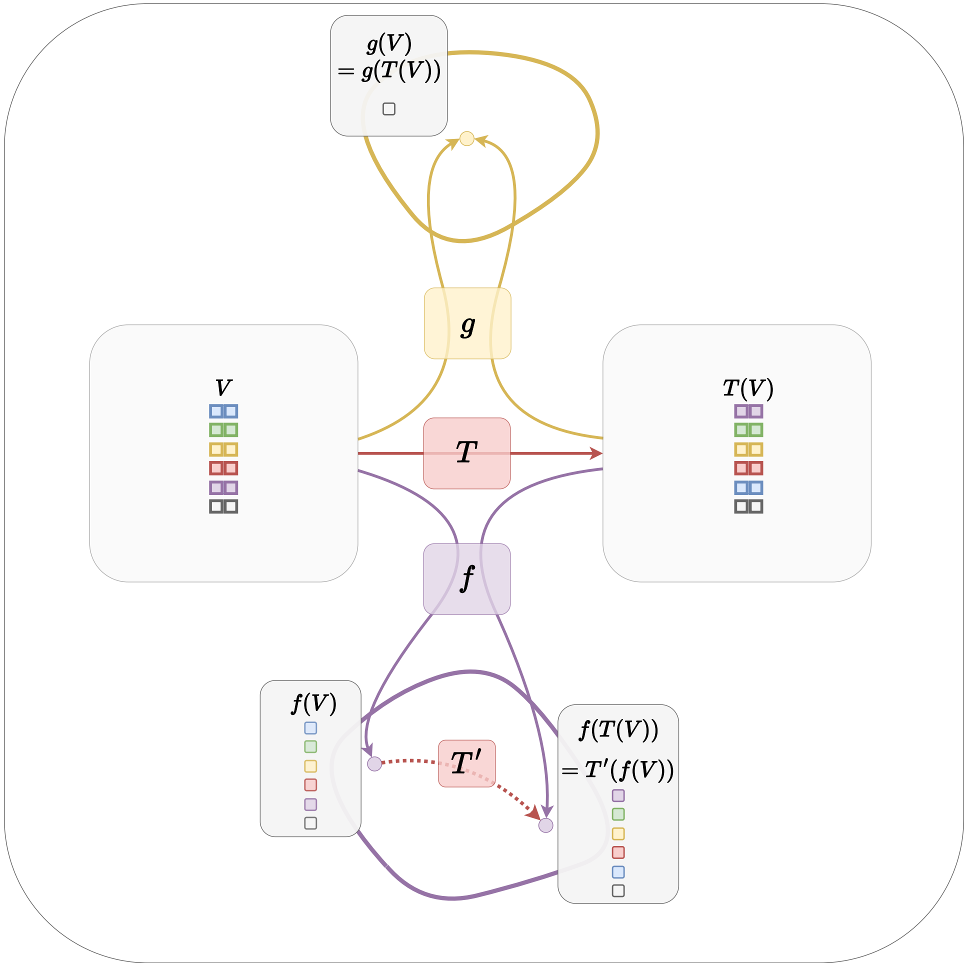
$$f(\mathbf{PAP}^\top) = \mathbf{P}f(A)$$

PERMUTATION INVARIANCE AND EQUIVARIANCE

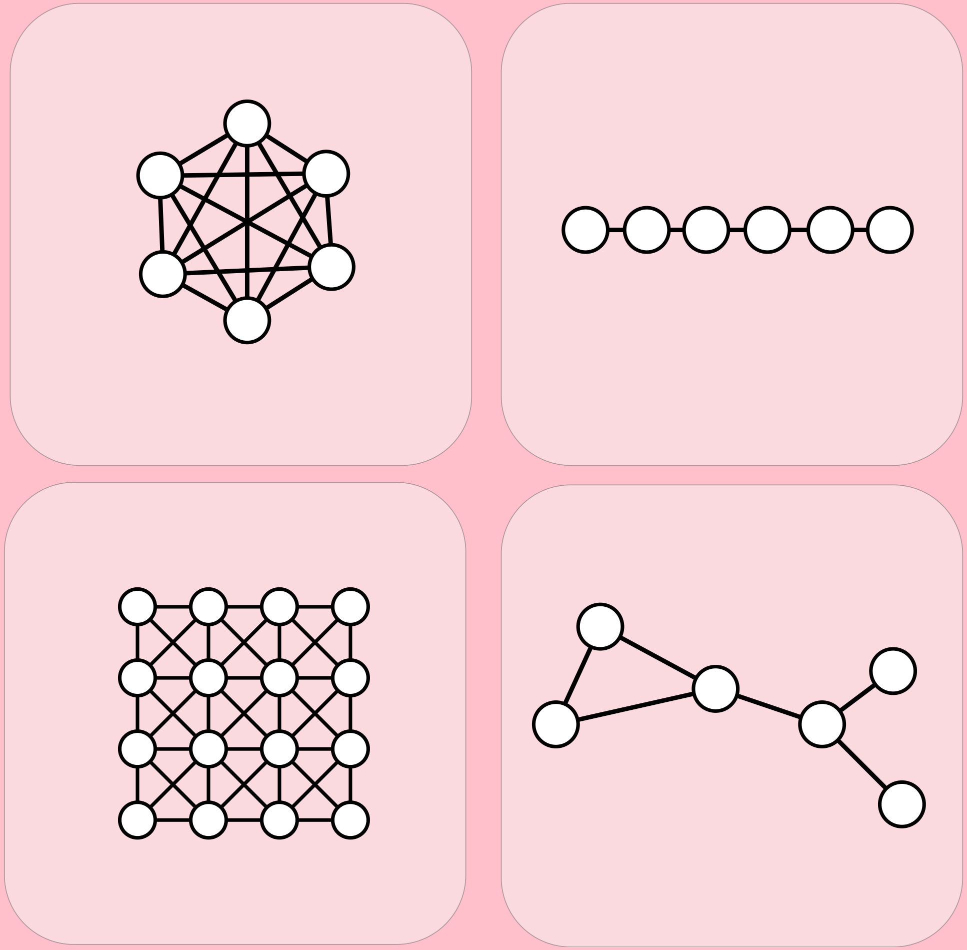


(Adapted from lecture by Mikkel Schmidt given at DTU's course (2022) "Graph Representation Learning")

PERMUTATION INVARIANCE AND EQUIVARIANCE

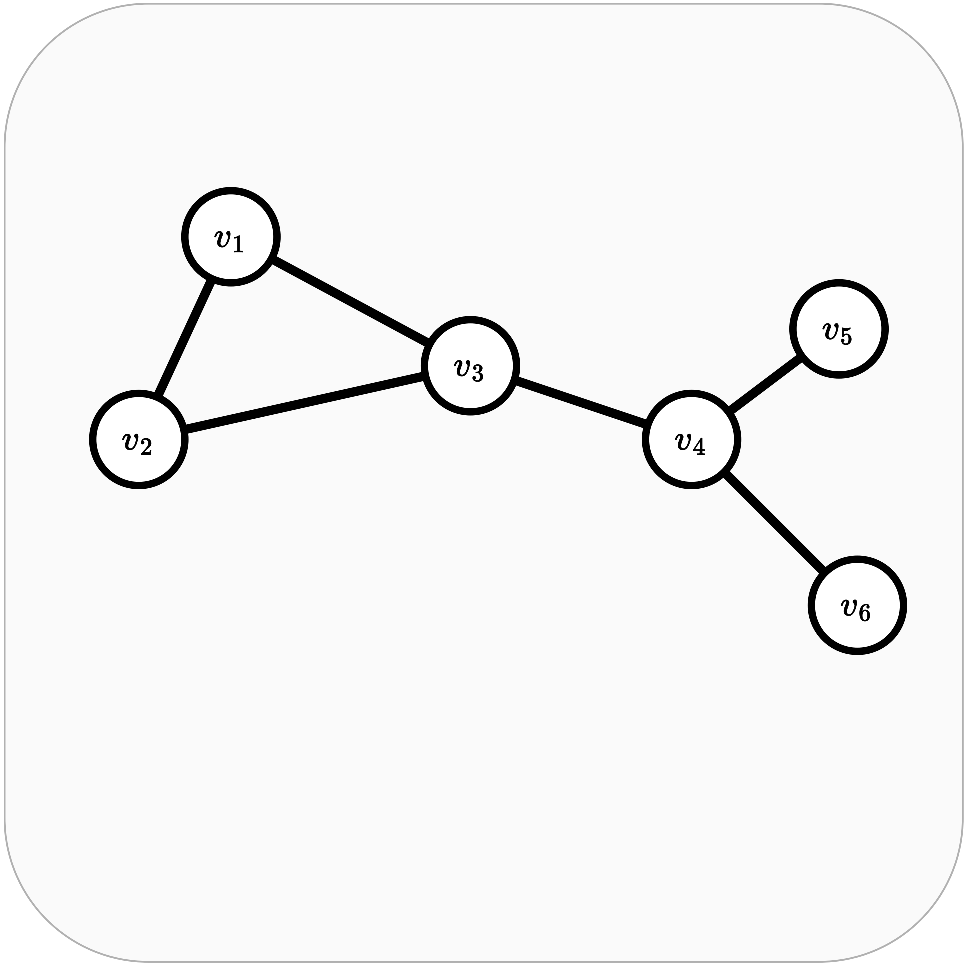


DISCUSSION: IN- AND EQUIVARIANCES OF MLPS, CNNS, RNNS, GNNS?

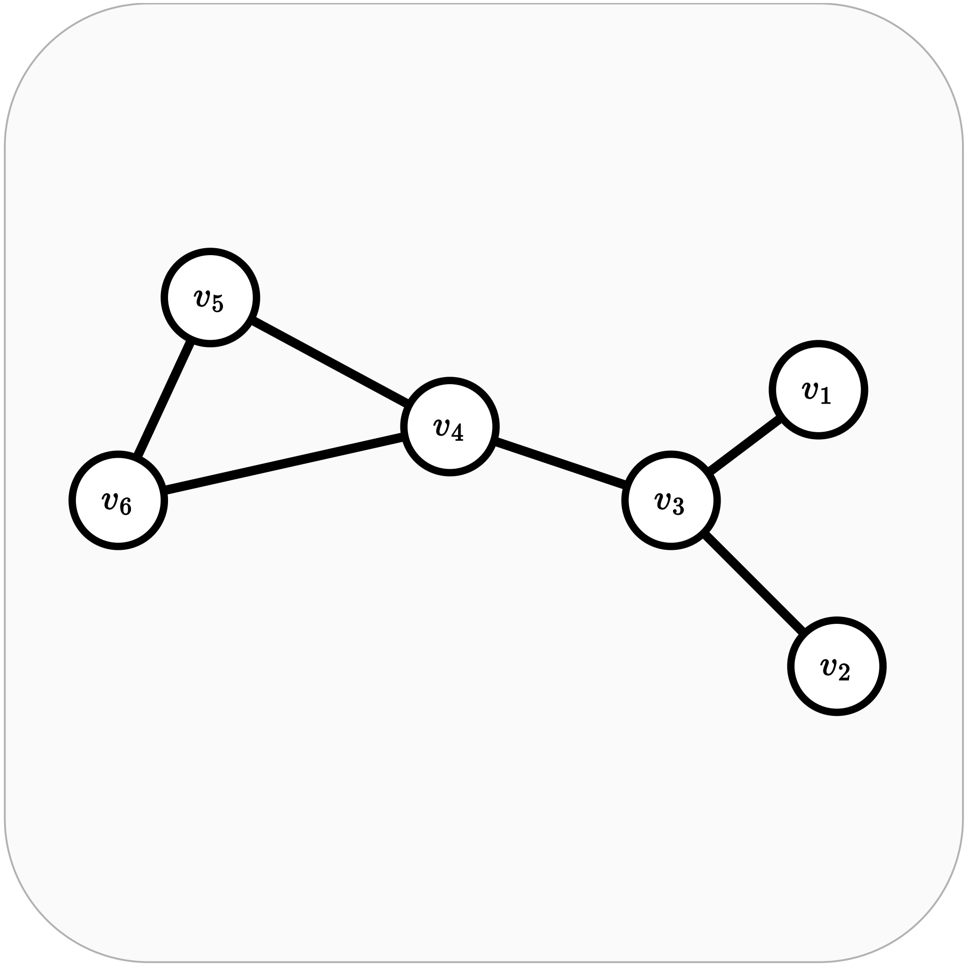


MESSAGE PASSING

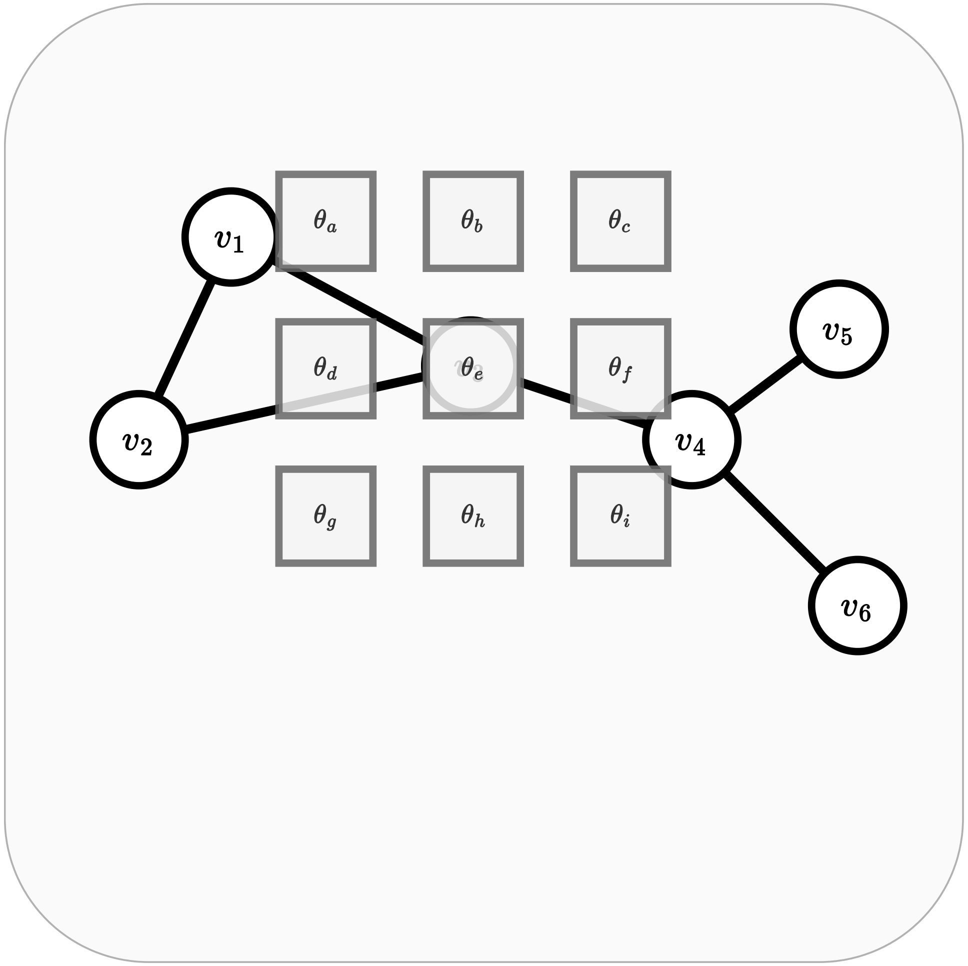
GRAPH WITH ORDERING



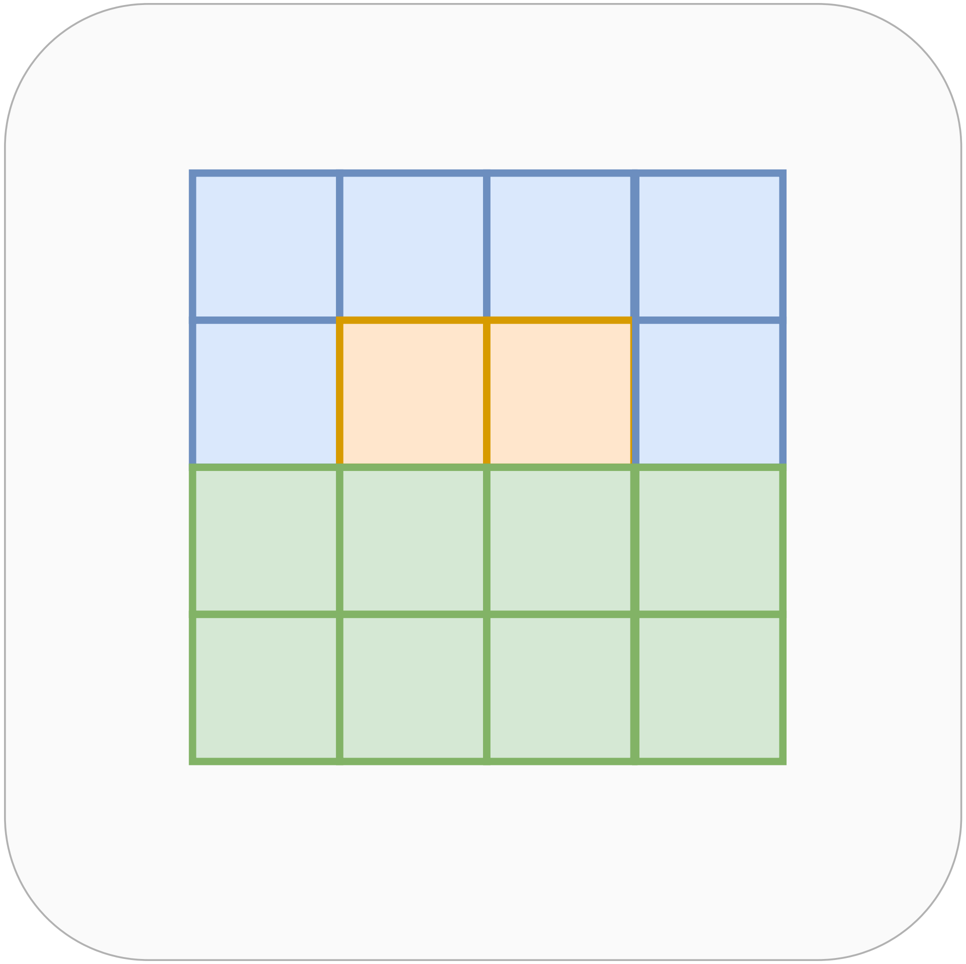
GRAPH WITH ORDERING



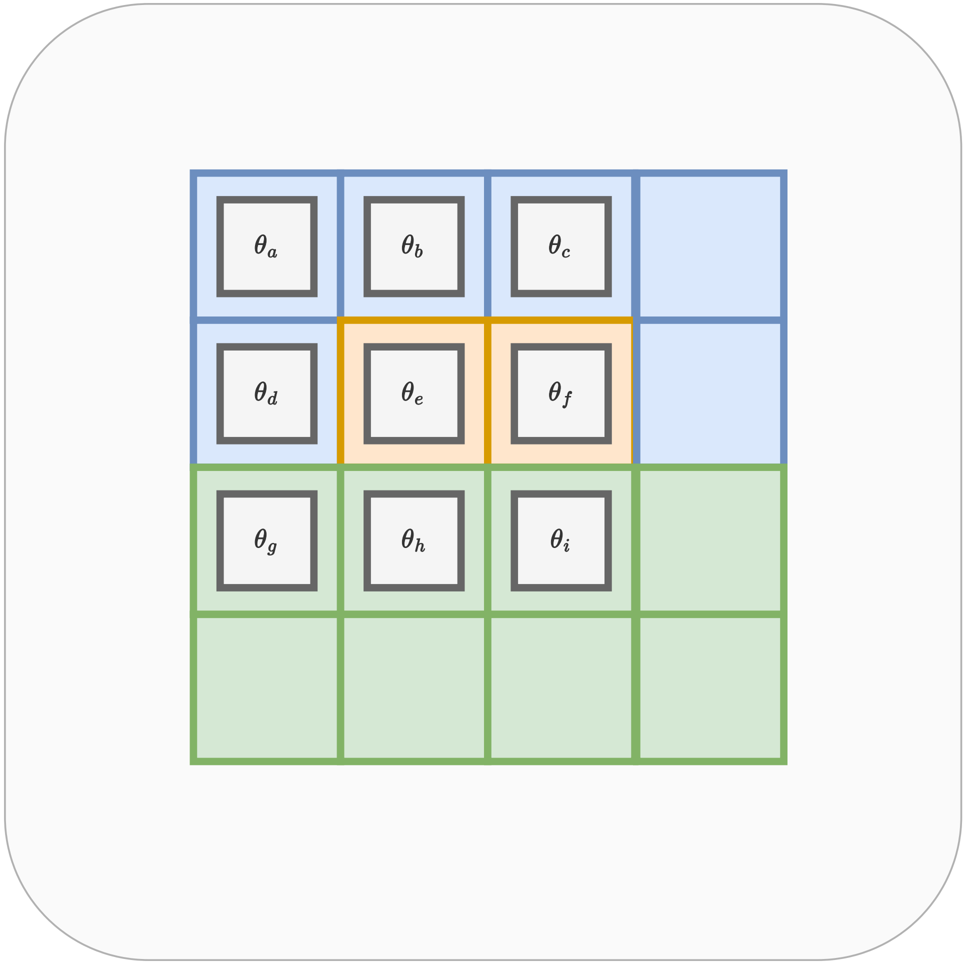
APPLYING CONVOLUTIONAL FILTER?



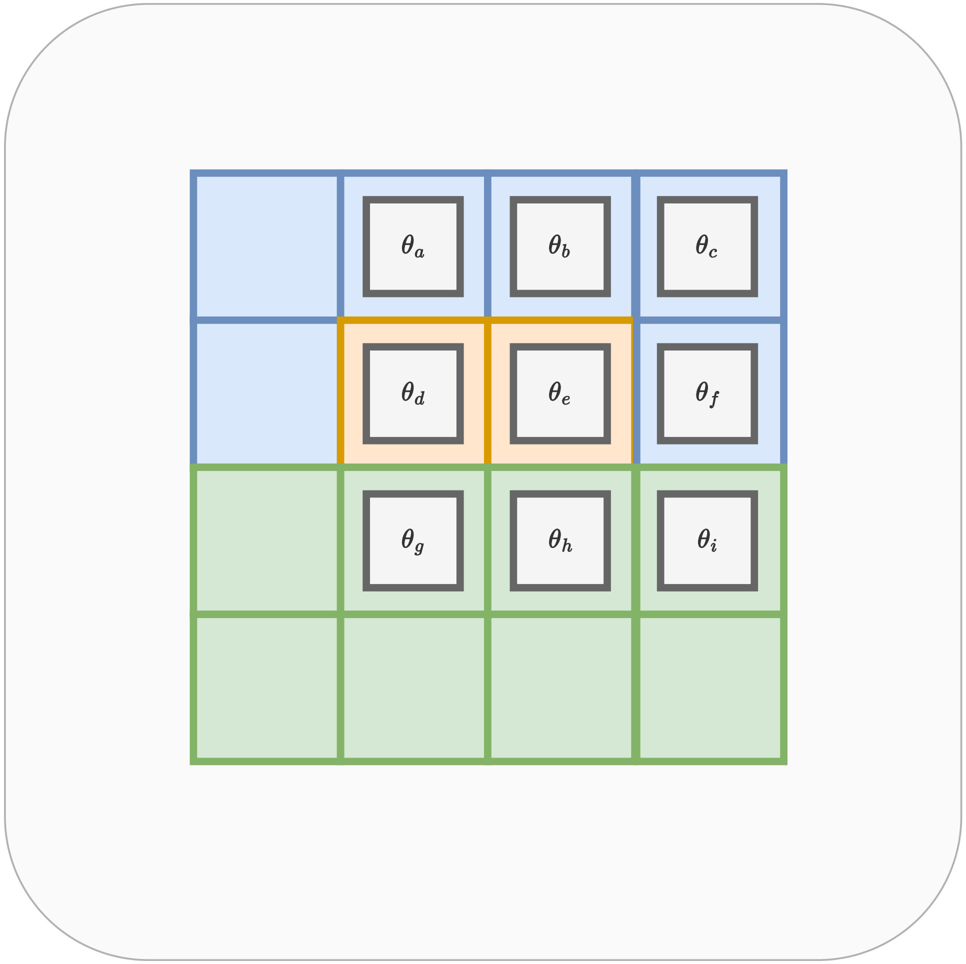
CONVOLUTION AS MESSAGE PASSING



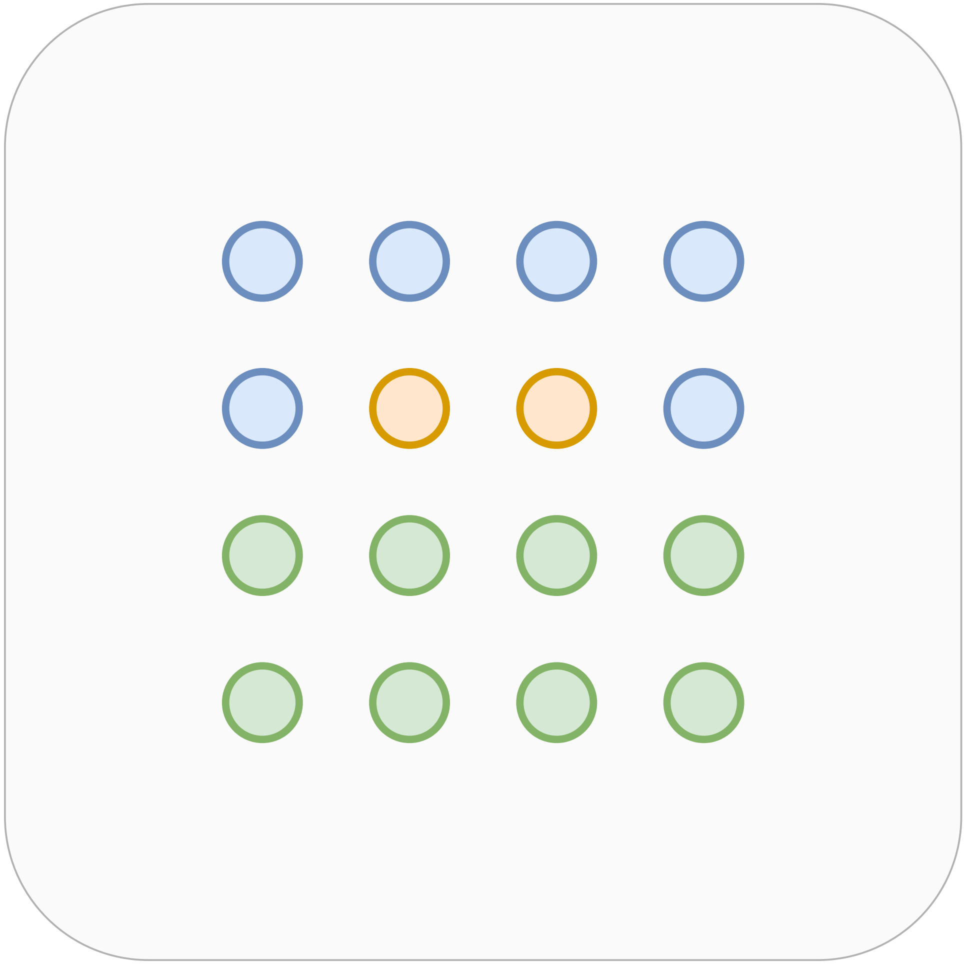
CONVOLUTION AS MESSAGE PASSING



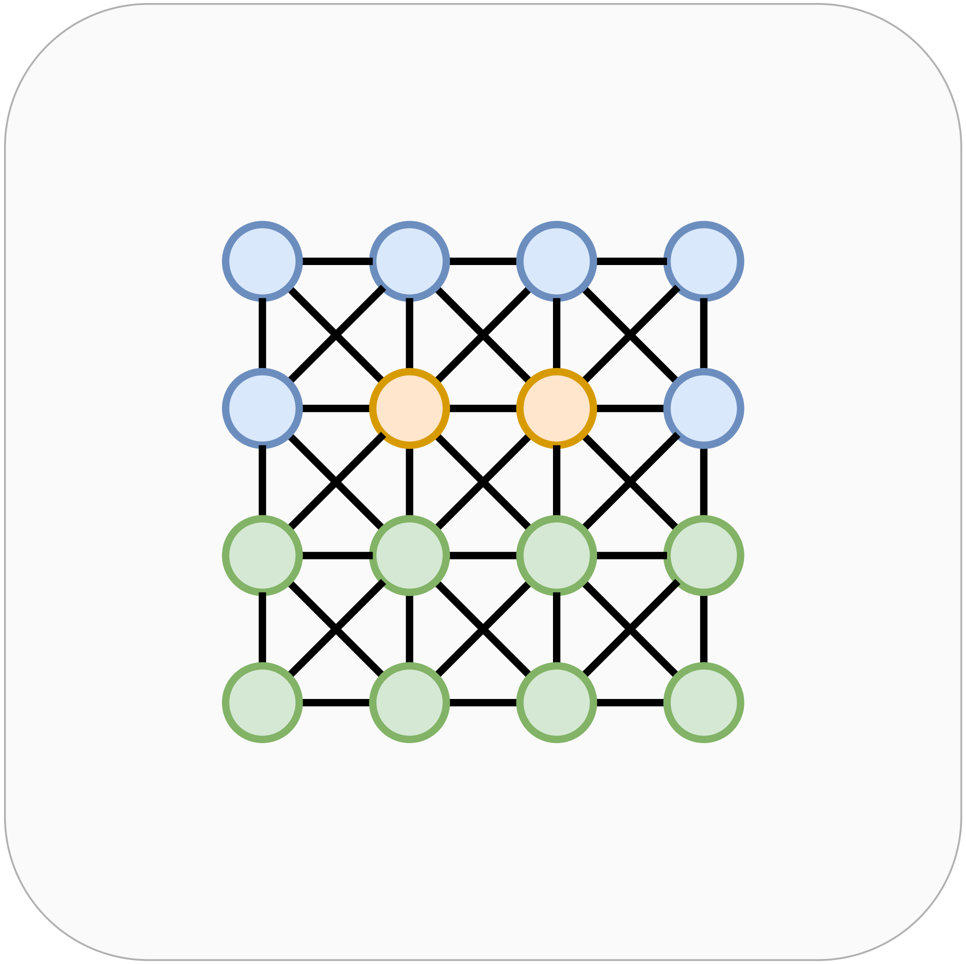
CONVOLUTION AS MESSAGE PASSING



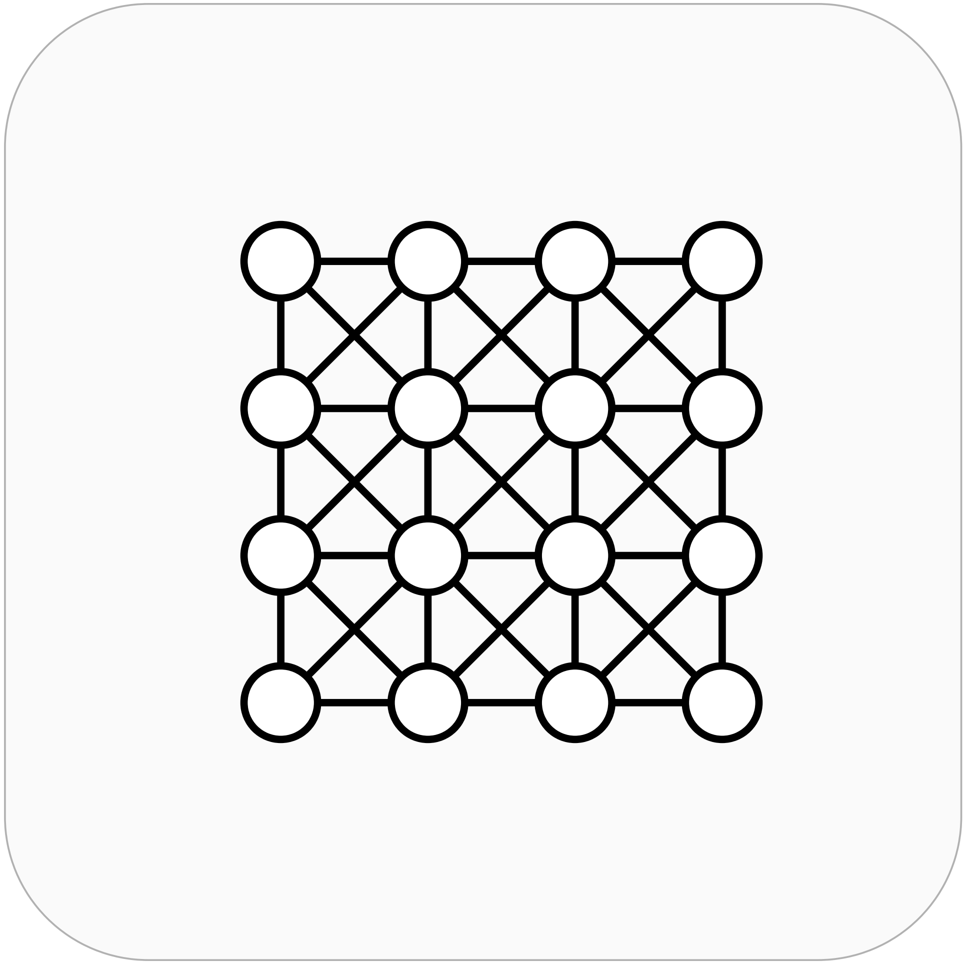
CONVOLUTION AS MESSAGE PASSING



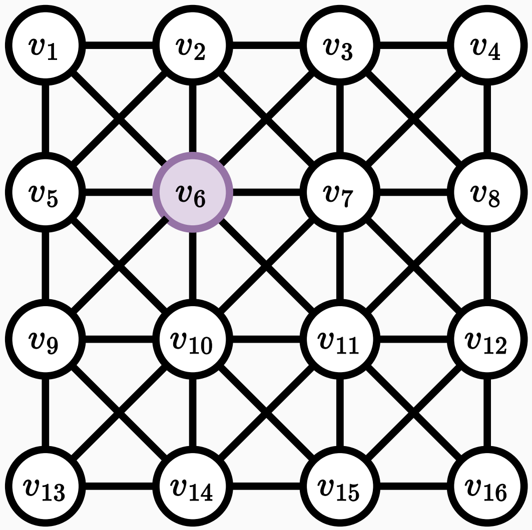
CONVOLUTION AS MESSAGE PASSING



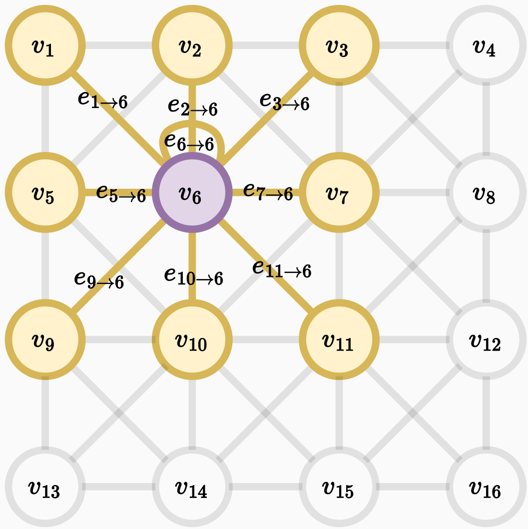
CONVOLUTION AS MESSAGE PASSING



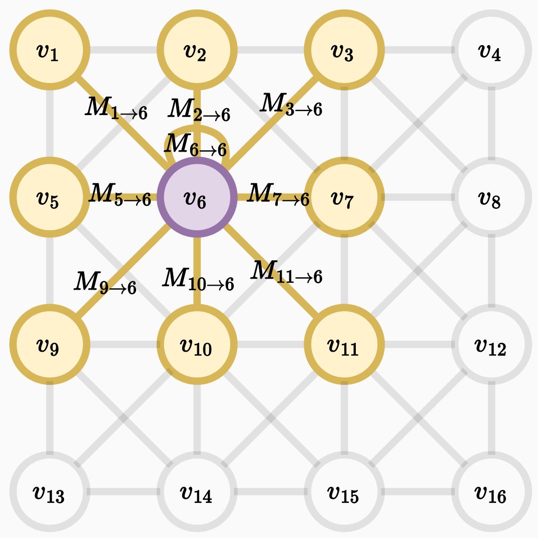
CONVOLUTION AS MESSAGE PASSING



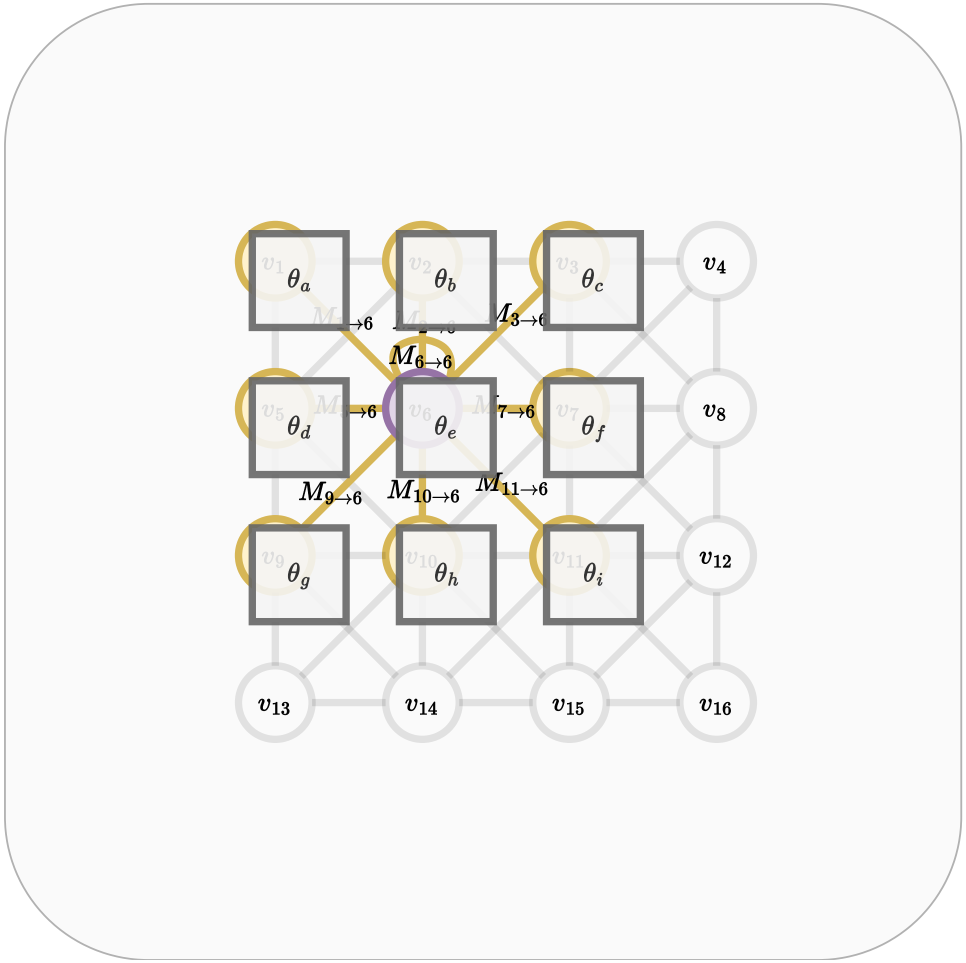
CONVOLUTION AS MESSAGE PASSING



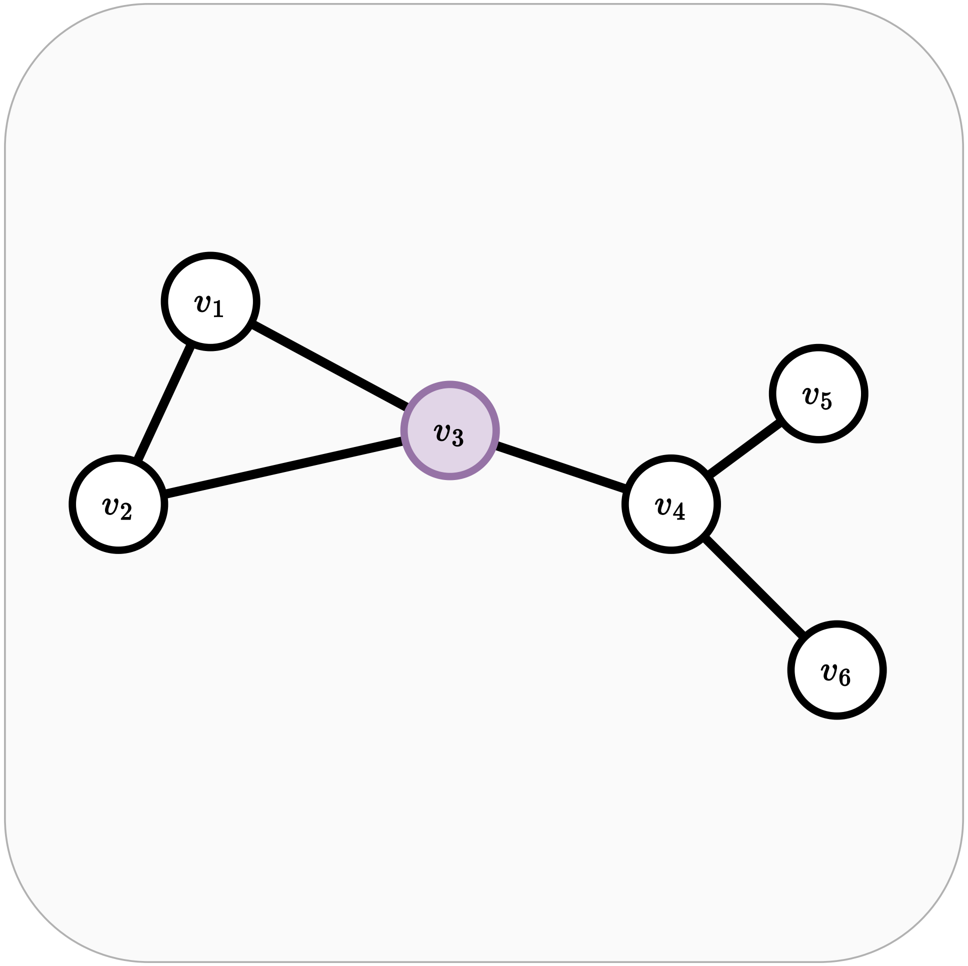
CONVOLUTION AS MESSAGE PASSING



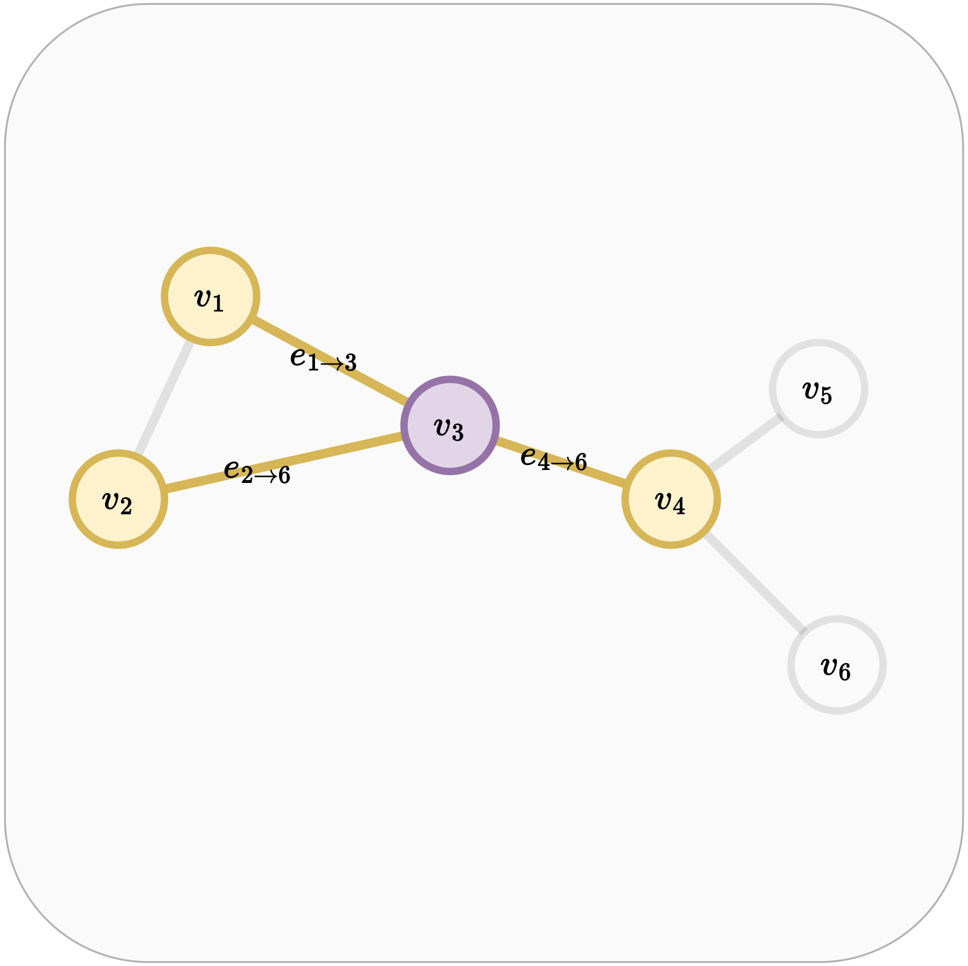
CONVOLUTION AS MESSAGE PASSING



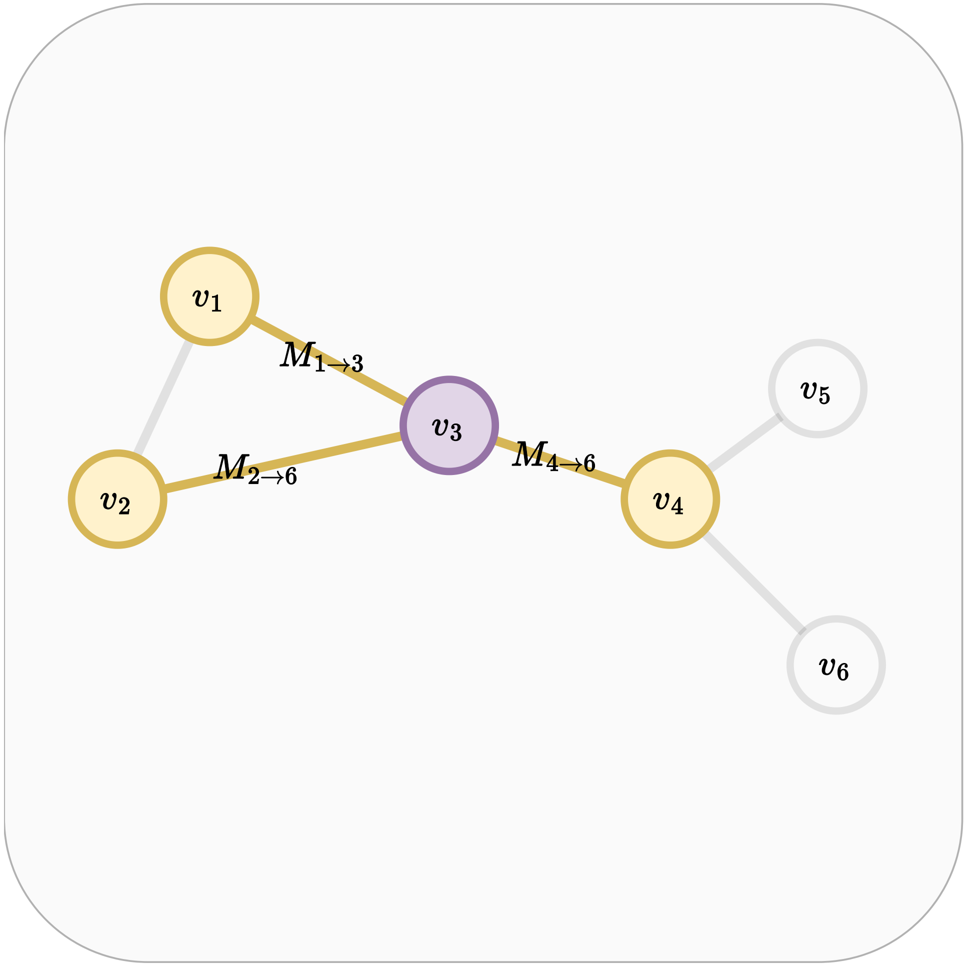
MESSAGE PASSING ON GRAPH



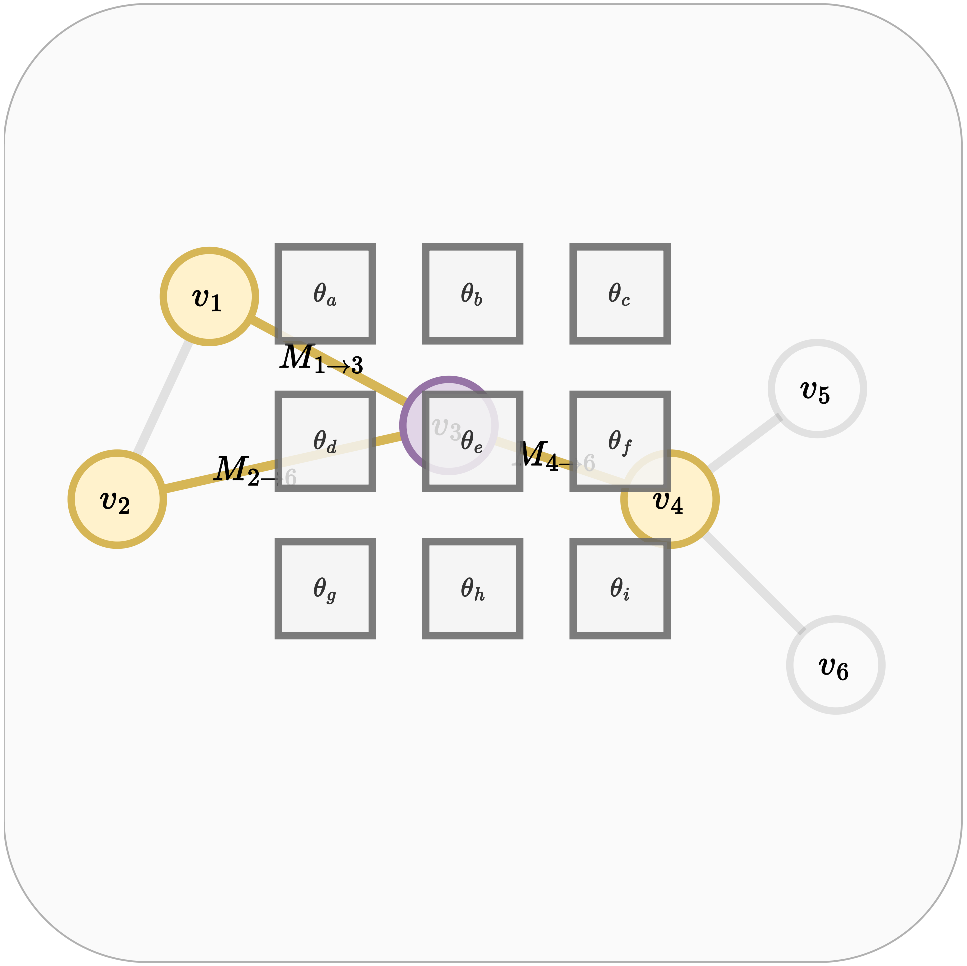
MESSAGE PASSING ON GRAPH



MESSAGE PASSING ON GRAPH



MESSAGE PASSING ON GRAPH



ABSTRACT MESSAGE PASSING

$$\mathbf{M}_{\mathcal{N}(u)}^{(k)} = \mathrm{AGGREGATE}^{(k)}\left(\{\mathbf{h}_v^{(k)}, \forall v \in \mathcal{N}(u)\}\right)$$

$$\mathbf{h}_u^{k+1} = \mathrm{UPDATE}^{(k)}\left(\mathbf{h}_u^{(k)}, \mathbf{M}_{\mathcal{N}(u)}^{(k)}\right)$$

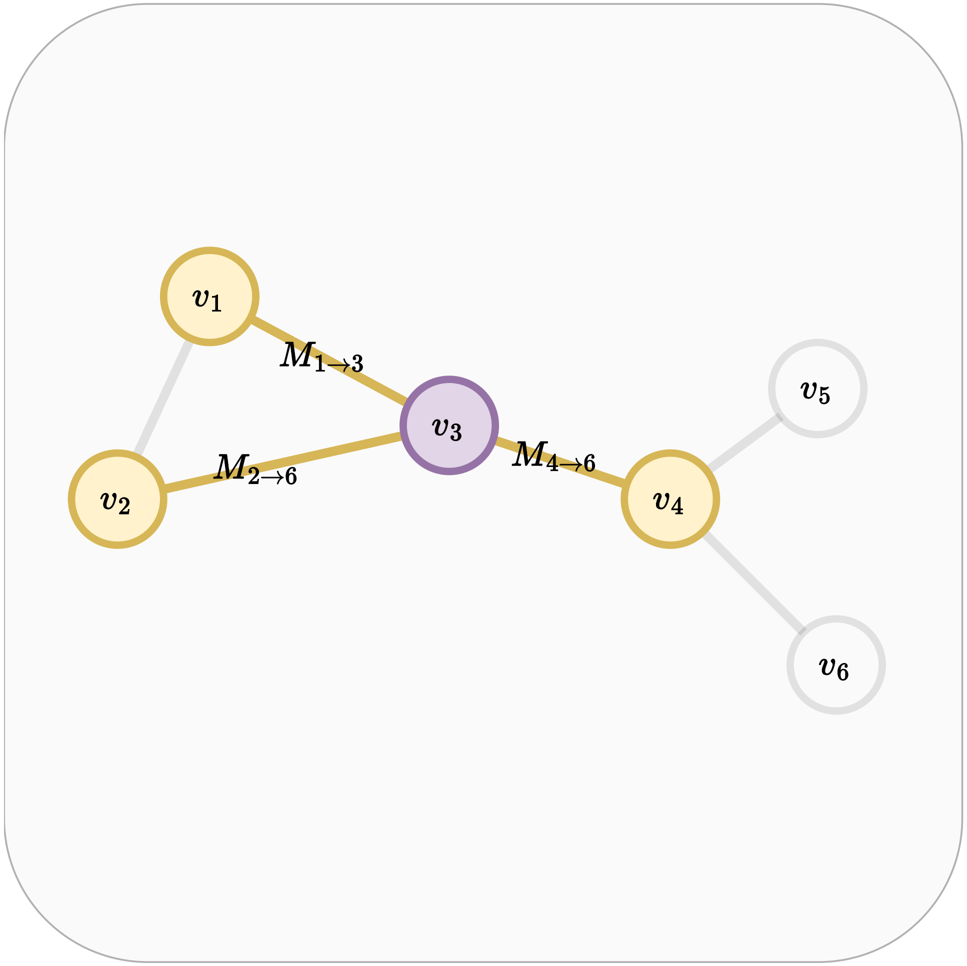
(Based on Chap. 5 in: Hamilton, William L. Graph representation learning. Morgan & Claypool Publishers, 2020.)

BASIC INSTANTIATION OF MESSAGE PASSING

$$\mathbf{h}_u^{(k)} = \sigma\left(\mathbf{W}_{\text{self}}^{(k)}\mathbf{h}_u^{(k-1)} + \mathbf{W}_{\text{neigh}}^{(k)}\sum_{v\in\mathcal{N}(u)}\left(\mathbf{h}_v^{(k-1)} + \mathbf{b}^{(k)}\right)\right)$$

(Based on Chap. 5 in: Hamilton, William L. Graph representation learning. Morgan & Claypool Publishers, 2020.)

CONVOLUTION AS MESSAGE PASSING

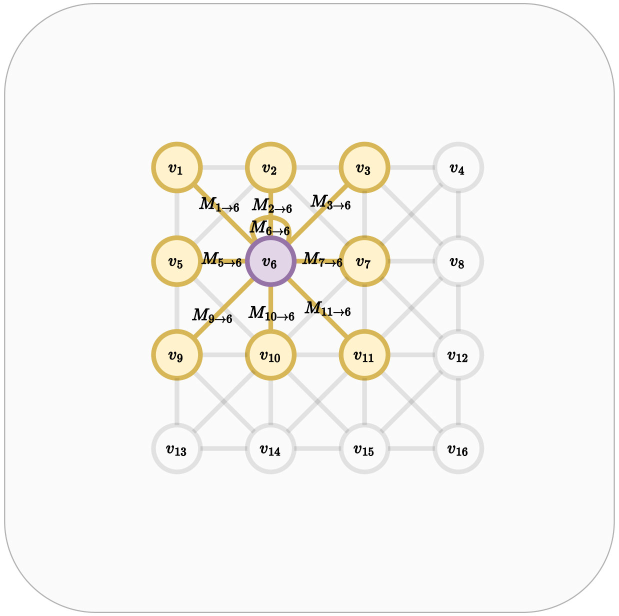


CONVOLUTION AS MESSAGE PASSING ON GRID

Algorithm 1 CNN as message passing

Input: Weight matrix, $\mathbf{W}$, with parameters $\theta_{u \to v}$, neighborhood function, $\mathcal{N}$.
Input: Graph, $\mathcal{G}$ with nodes $\mathcal{V} = \{v_i\}_{i=0}^{V}$ and edges $\mathcal{E} = \{e_{u \to v} | u, v \in \mathcal{V}\}$.
Output: Updated node features $\mathbf{h}_u^{(1)}$ for all nodes u

Initialize $\mathbf{h}_u^{(0)}$ as v_u
for $k \in [0]$ **do**
  for $u \in \mathcal{V}$ **do**
    for $v \in \mathcal{N}(u) \cup \{u\}$ **do**
      Compute messages : $\mathbf{M}_{v \to u} = \theta_{v \to u} \cdot \mathbf{h}_v^{(k)}$
    end for
    Compute total message: $\mathbf{M}_u = \sum_{v \in \mathcal{N}(u)} \mathbf{M}_{v \to u}$
    Update node: $\mathbf{h}_u^{(k+1)} \leftarrow \sigma(\mathbf{M}_u)$
  end for
end for



MESSAGE PASSING ON GRAPH

Algorithm 2 Basic graph message passing

Input: Weight matrices, $\mathbf{W}_{\text{self}}$, $\mathbf{W}_{\text{neigh}}$, and bias, $\mathbf{b}$, neighborhood function, $\mathcal{N}$.

Input: Graph, $\mathcal{G}$ with nodes $\mathcal{V} = \{v_i\}_{i=0}^{V}$ and edges $\mathcal{E} = \{e_{u \to v} | u, v \in \mathcal{V}\}$, and a specified K number of rounds of message passing.

Output: Updated node features $\mathbf{h}_u^{(K+1)}$ for all nodes u

Initialize $\mathbf{h}_u^{(0)}$ as v_u for all nodes u

for $k \in [0, 1, \ldots, K]$ **do**

  for $u \in \mathcal{V}$ **do**

    for $v \in \mathcal{N}(u)$ **do**

      Compute messages : $\mathbf{M}_{v \to u} = \mathbf{W}_{\text{neighbors}} \mathbf{h}_v^{(k)} + \mathbf{b}$

    end for

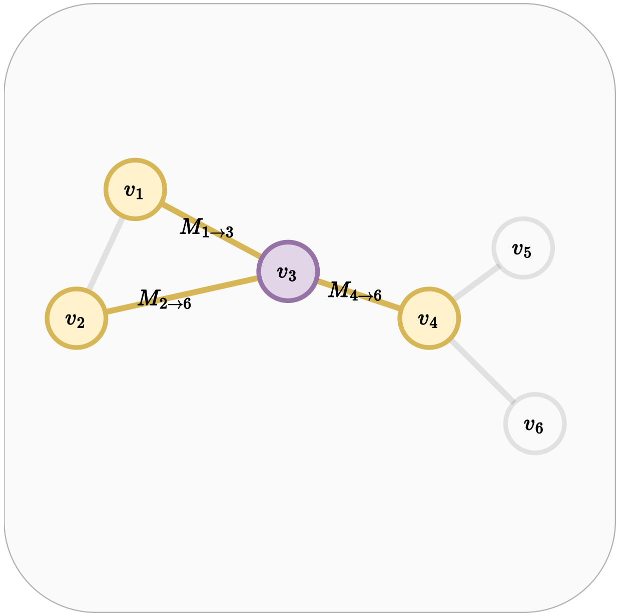
    Compute self message: $\mathbf{M}_{\text{self}} = \mathbf{W}_{\text{self}} \mathbf{h}_u^k$

    Compute total message: $\mathbf{M}_u = \mathbf{M}_{\text{self}} + \sum_{v \in \mathcal{N}(u)} \mathbf{M}_{v \to u}$

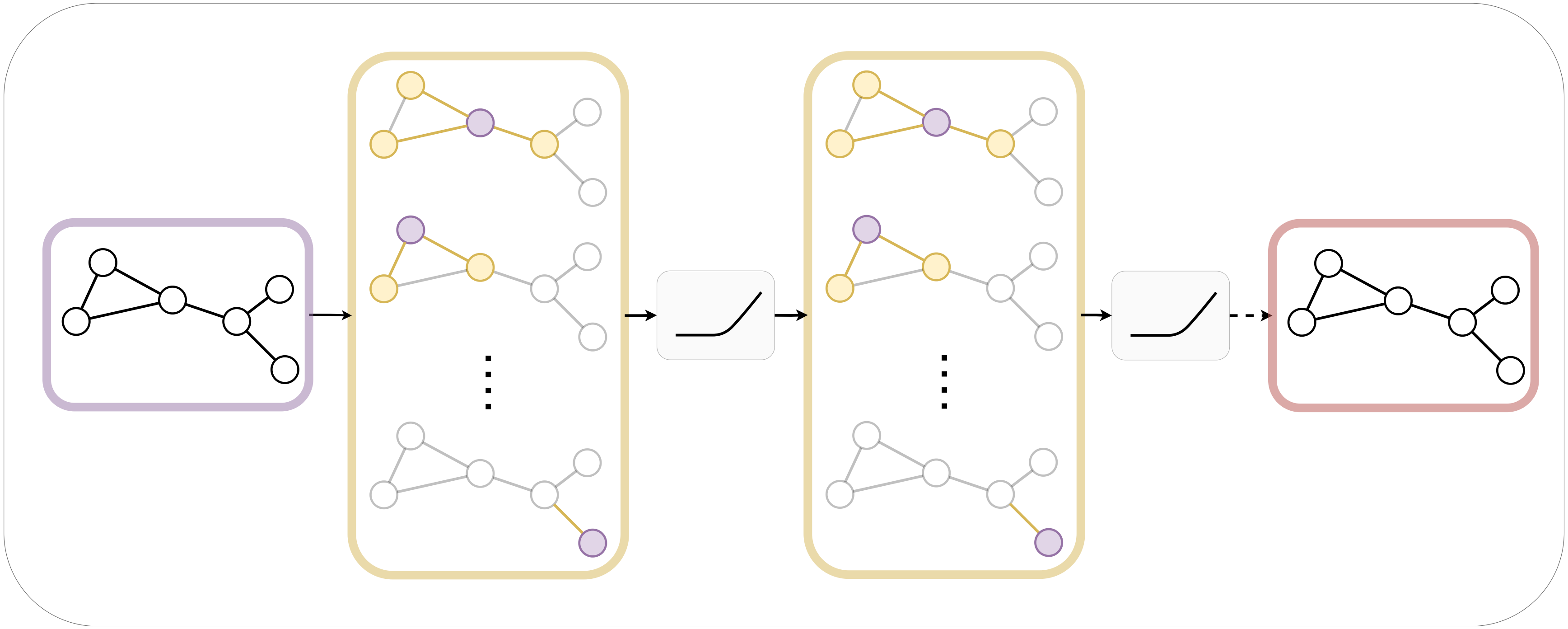
    Update node: $\mathbf{h}_u^{(k+1)} \leftarrow \sigma(\mathbf{M}_u)$

  end for

end for

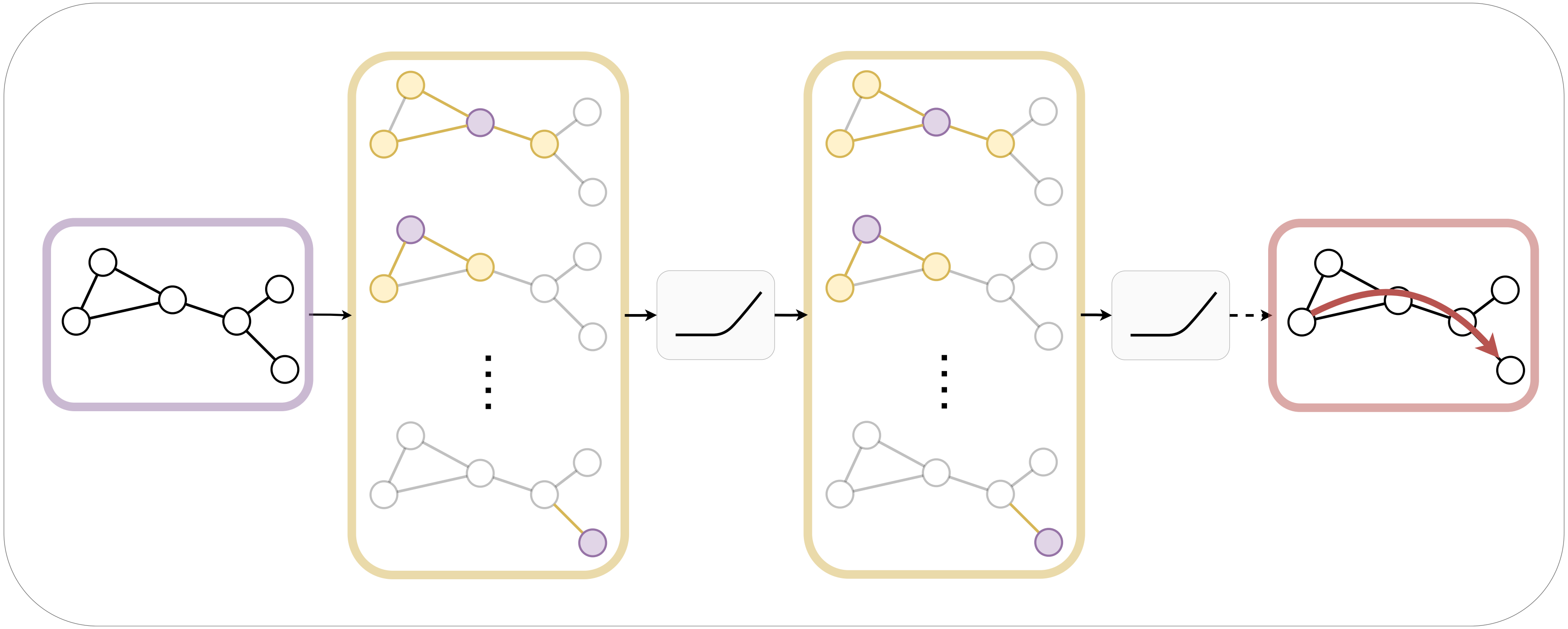


GRAPH MESSAGE PASSING NETWORKS



(Adapted from Thomas Kipf, https://tkipf.github.io/graph-convolutional-networks/)

GRAPH MESSAGE PASSING NETWORKS



DISCUSSION: WHAT ISSUES MIGHT A NAIVE IMPLEMENTATION RUN INTO?

DISCUSSION: WHAT ISSUES MIGHT A NAIVE IMPLEMENTATION RUN INTO?

Graph-level:

$$\mathbf{H}^{(t)} = \sigma\left(\mathbf{A}\mathbf{H}^{(k-1)}\mathbf{W}^{(k)}_{\text{neigh}} + \mathbf{H}^{(k-1)}\mathbf{W}^{k}_{\text{self}}\right)$$

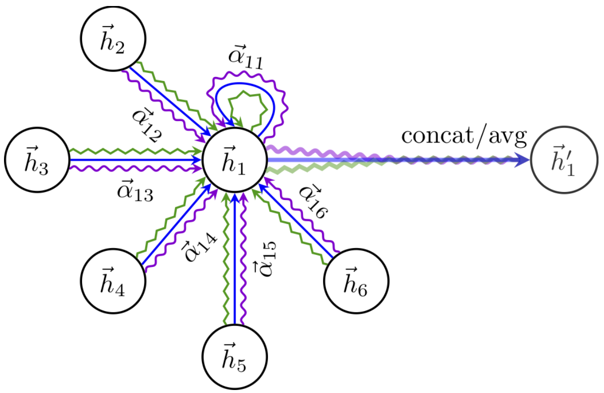
Normalization:

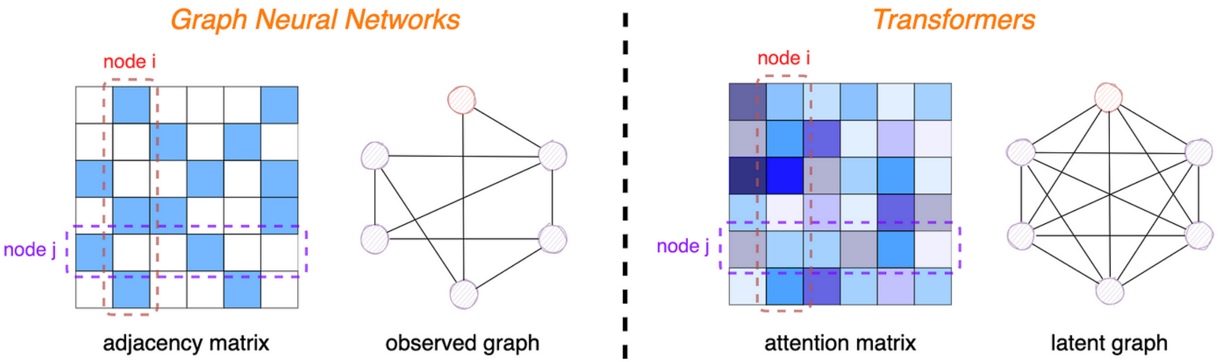
$$\mathbf{h}^{k}_{u} = \sigma\left(\mathbf{W}^{(k)} \sum_{v\in\mathcal{N}(u)\cup\{u\}} \frac{\mathbf{h}_v}{\sqrt{|\mathcal{N}(u)||\mathcal{N}(v)|}}\right)$$

GRAPH ATTENTION NETWORKS AND TRANSFORMERS

$$\mathbf{M}_{\mathcal{N}(u)} = \sum_{v \in \mathcal{N}(u)} \alpha_{u,v} \mathbf{h}_v$$

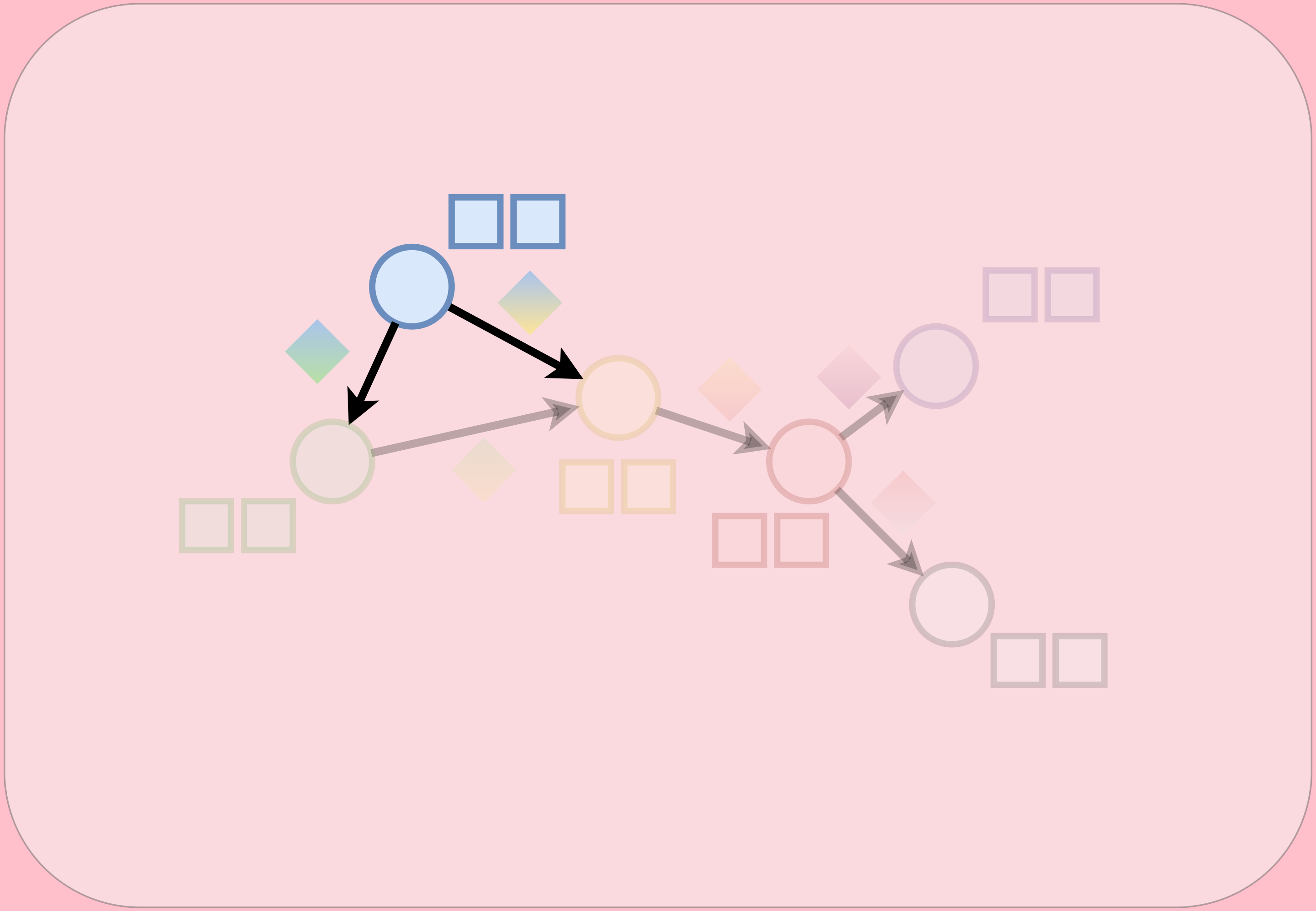
$$\alpha_{u,v} = \frac{\exp\left(\mathbf{a}^\top [\mathbf{W}\mathbf{h}_u \oplus \mathbf{W}\mathbf{h}_v]\right)}{\sum_{v' \in \mathcal{N}(u)} \exp\left(\mathbf{a}^\top [\mathbf{W}\mathbf{h}_u \oplus \mathbf{W}\mathbf{h}_{v'}]\right)}$$



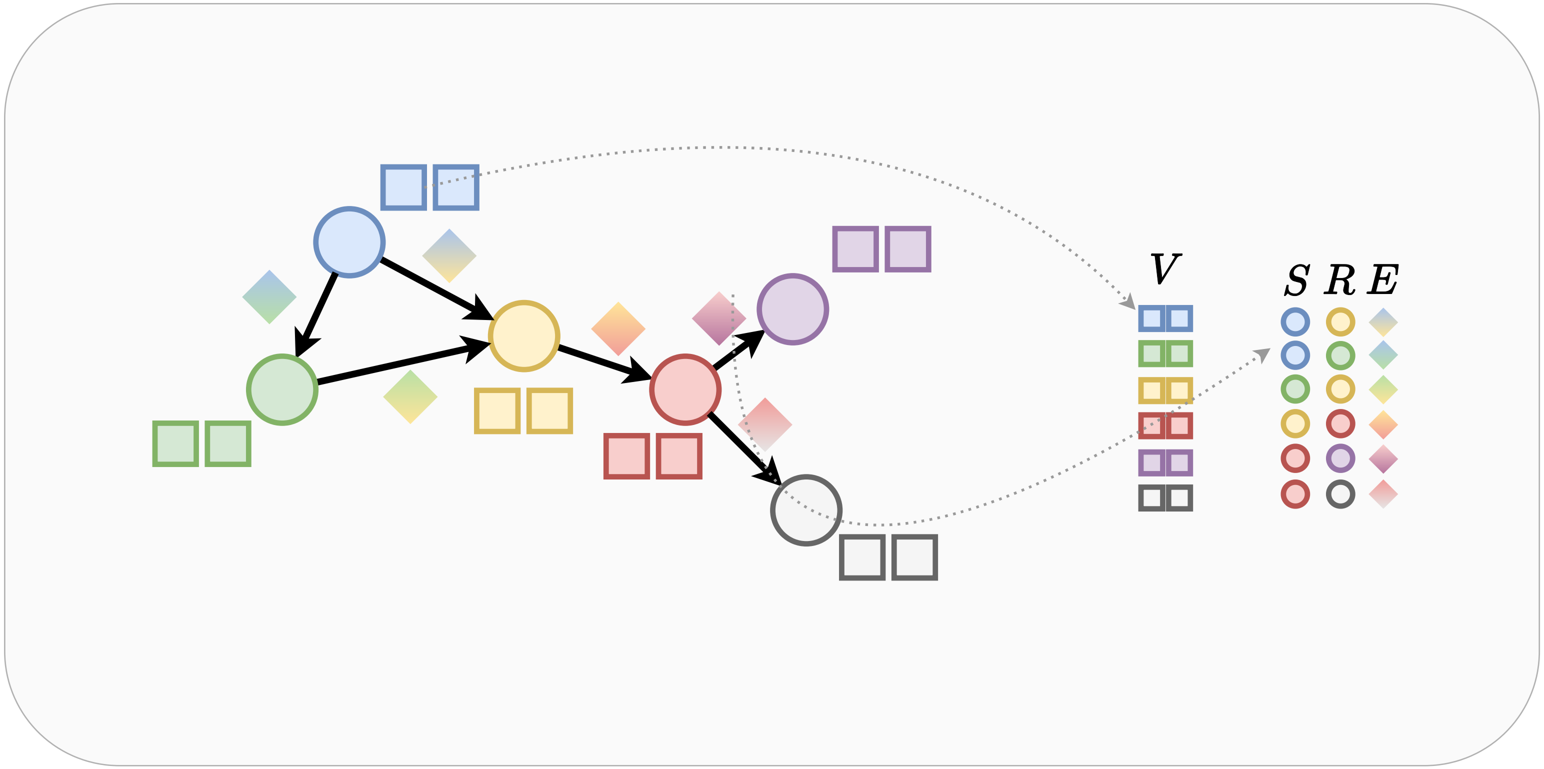


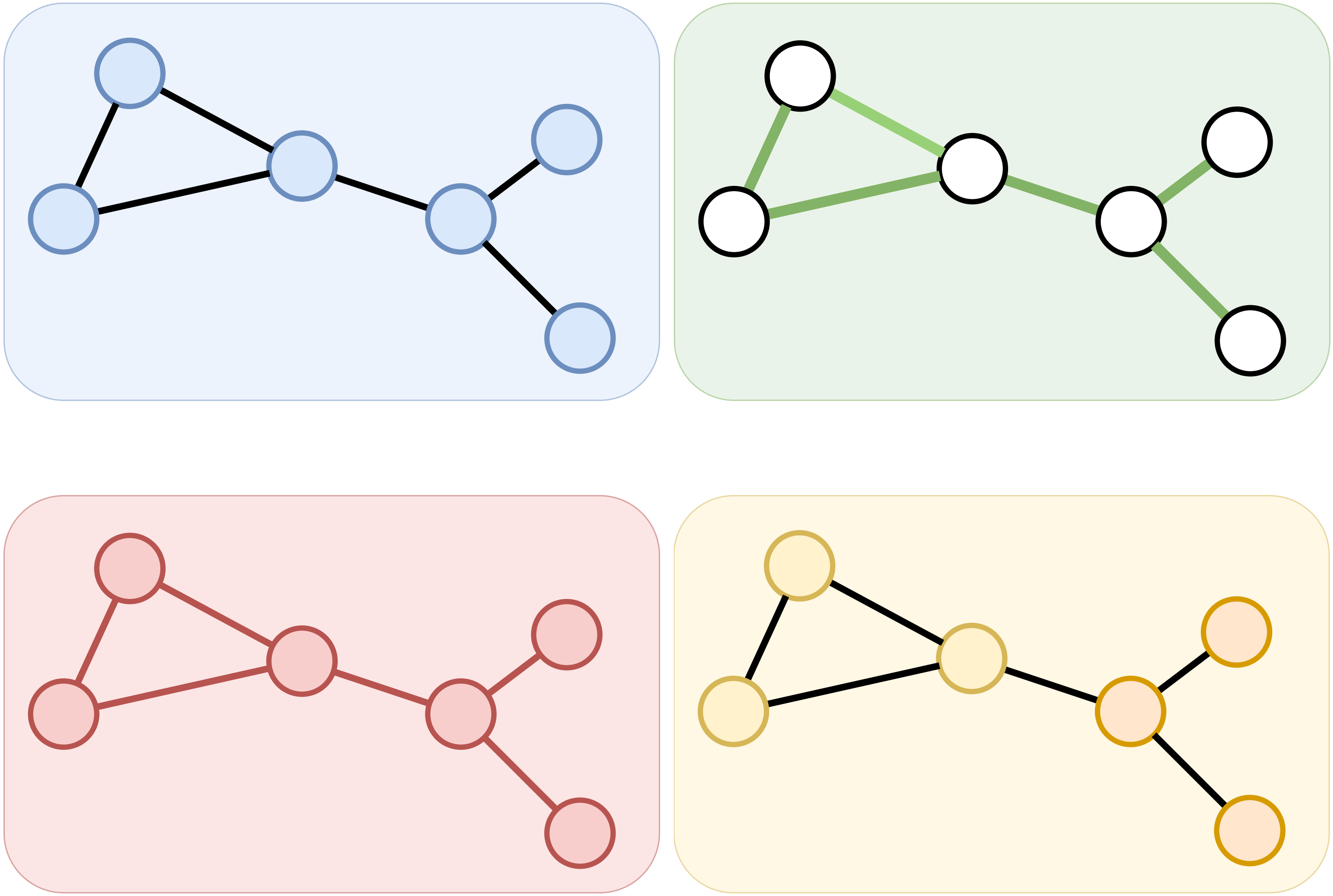
(left: Veličković, Petar, et al. "Graph Attention Networks." arXiv preprint arXiv:1710.10903 (2017))
(right: How to Build Graph Transformers with O(N) Complexity by Qitian Wu in @TDataScience)

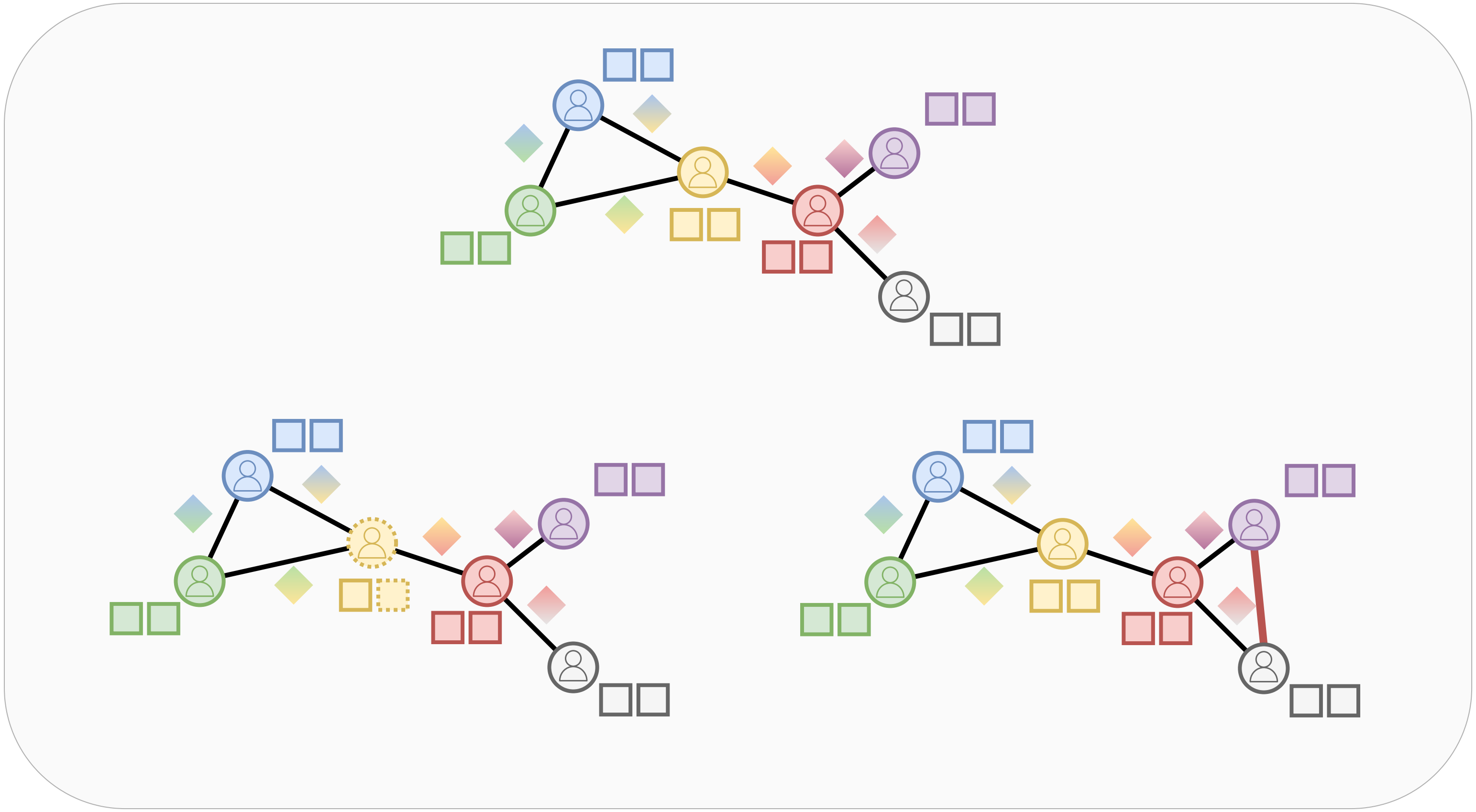
DISCUSSION: LIVE DEMO

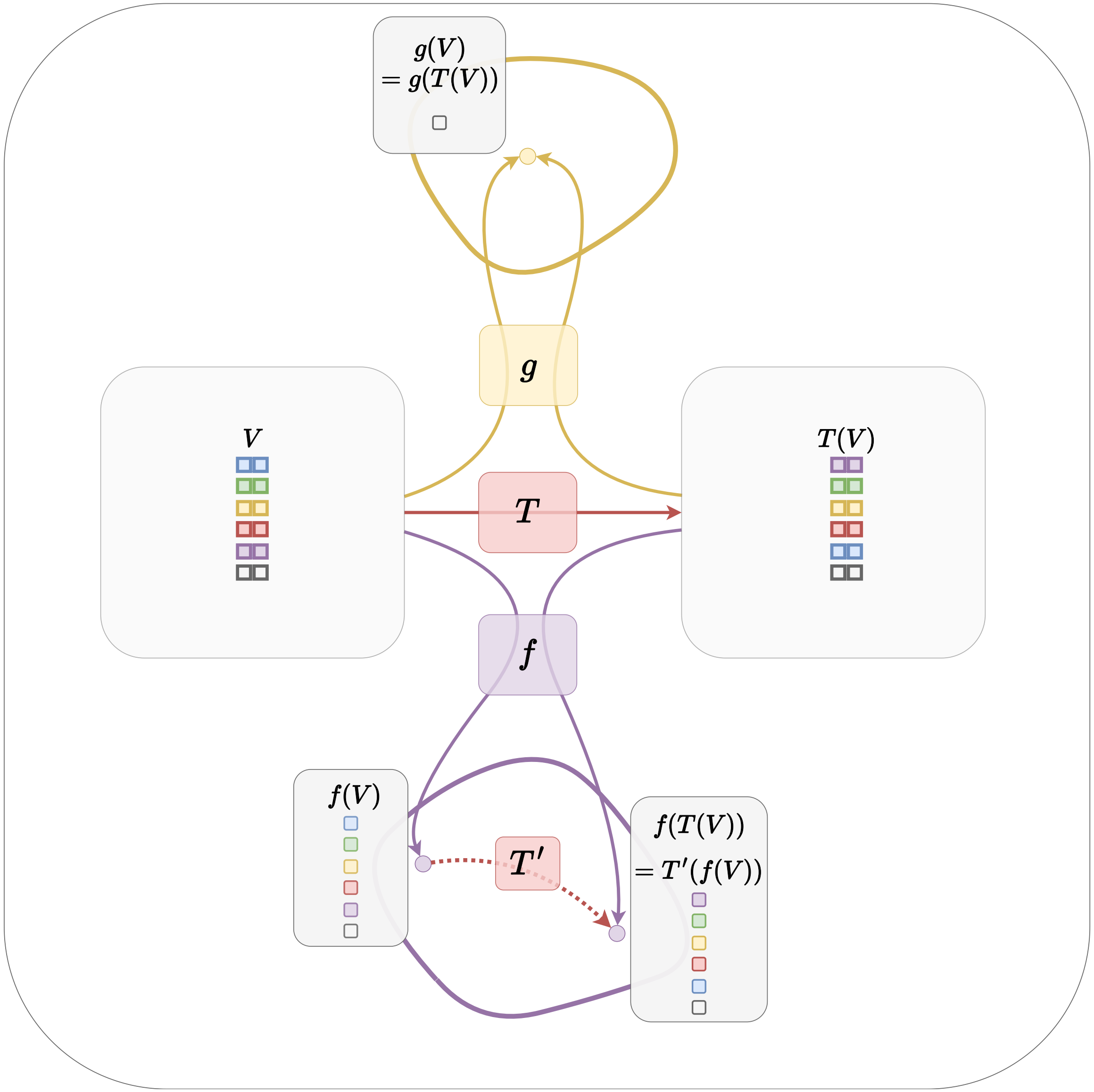


RECAP LECTURE 1

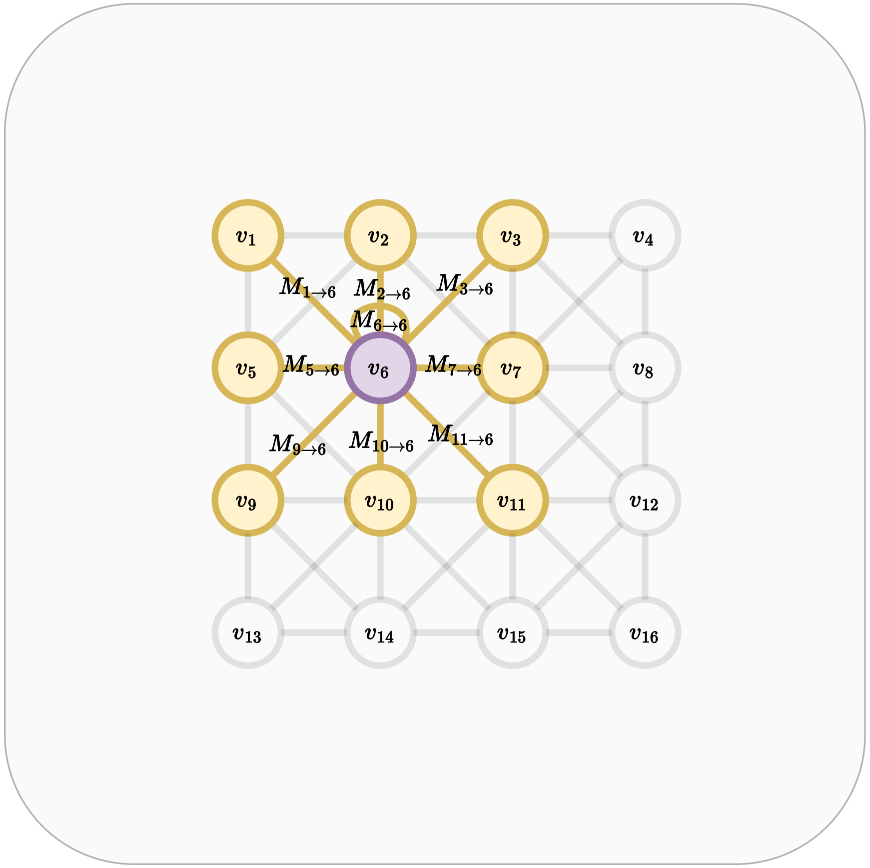

V
S R E

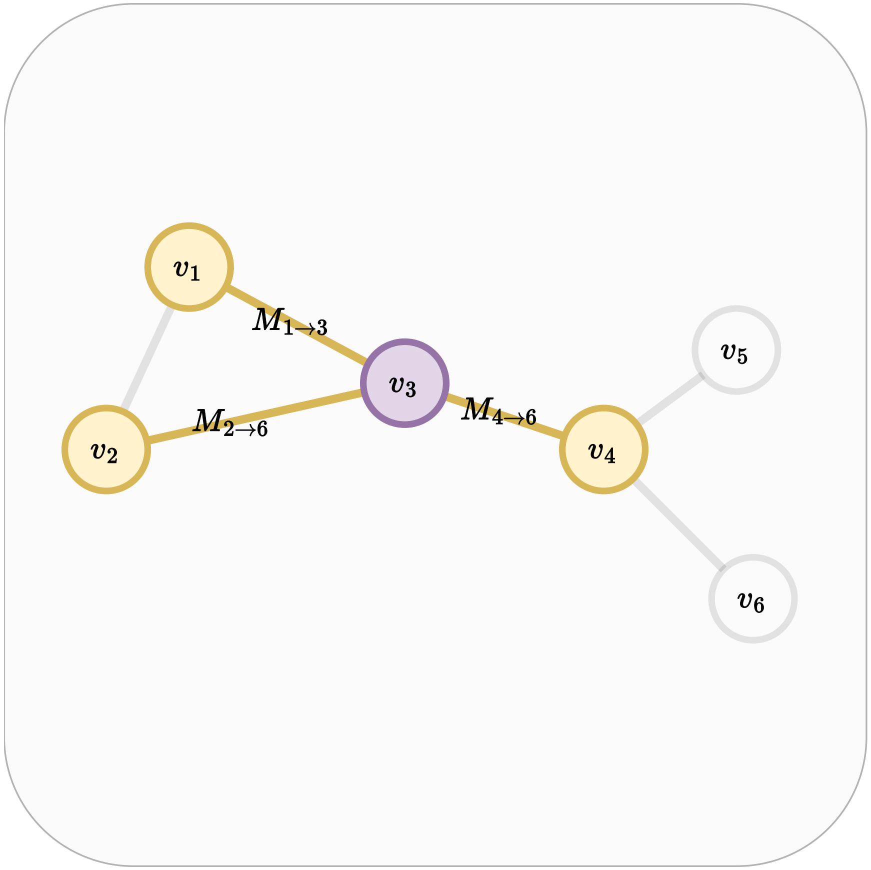


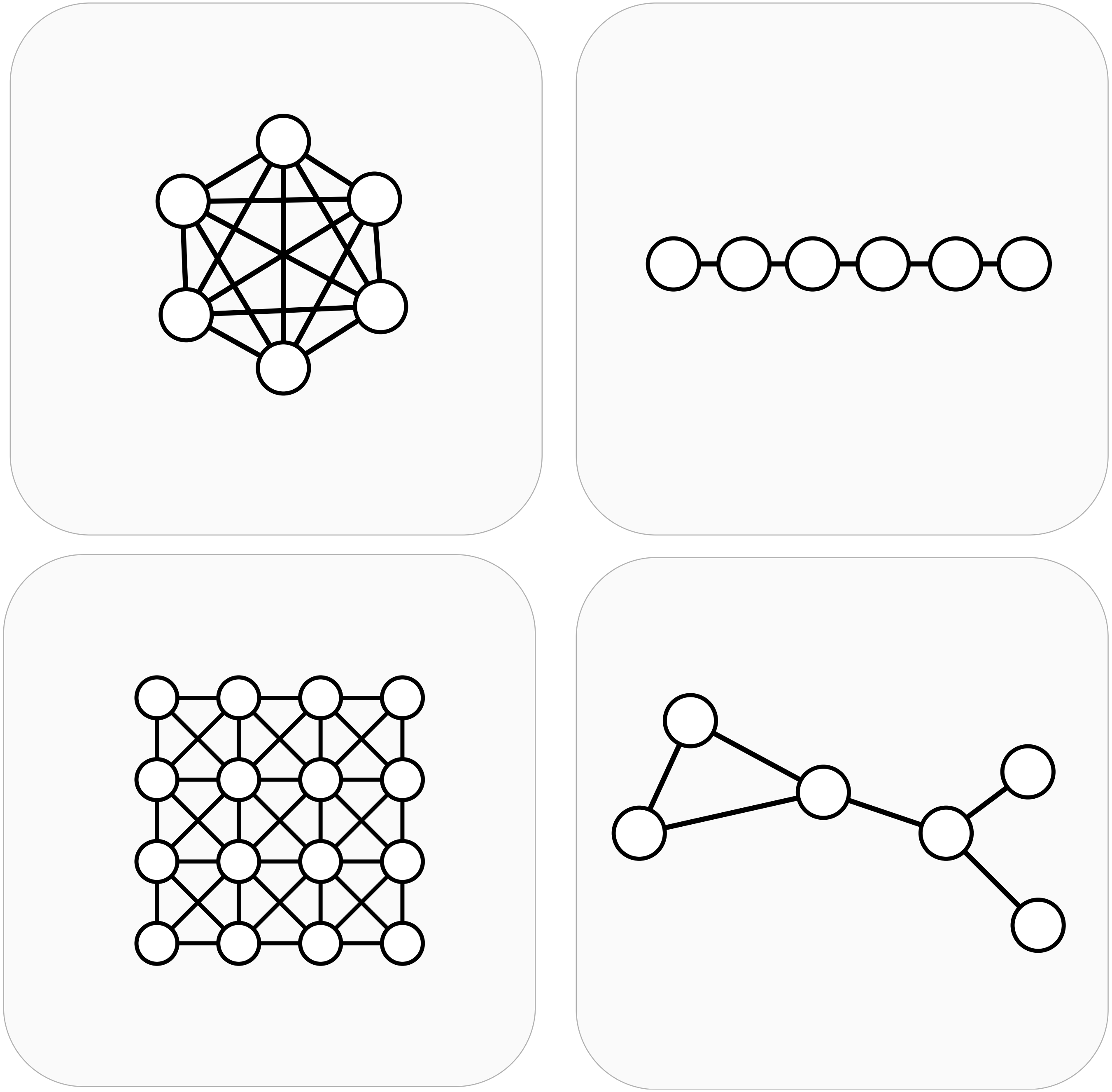




$g(V)$
$= g(T(V))$
g
V
T
$T(V)$
f
$f(V)$
T'
$f(T(V))$
$= T'(f(V))$


v_1 v_2 v_3 v_4
v_5 v_6 v_7 v_8
v_9 v_{10} v_{11} v_{12}
v_{13} v_{14} v_{15} v_{16}
$M_{1\to6}$
$M_{2\to6}$
$M_{3\to6}$
$M_{6\to6}$
$M_{5\to6}$
$M_{7\to6}$
$M_{9\to6}$
$M_{10\to6}$
$M_{11\to6}$


v_1
v_2
v_3
v_4
v_5
v_6
$M_{1\to3}$
$M_{2\to6}$
$M_{4\to6}$



MESSAGE PASSING ON GRAPH

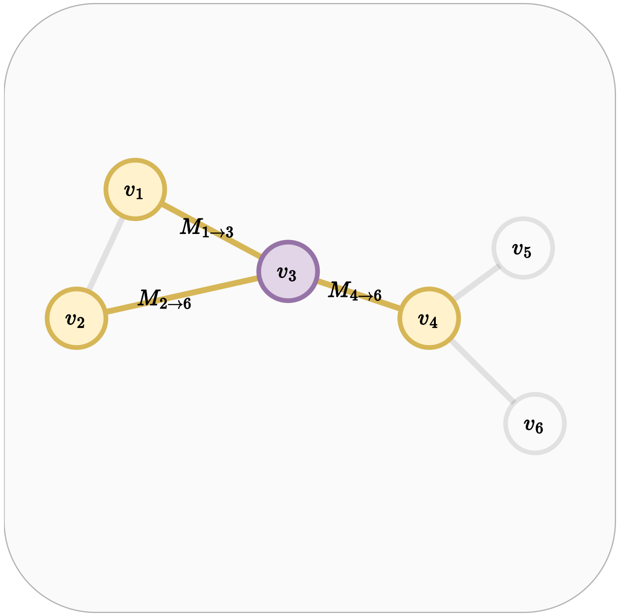
Algorithm 3 Basic graph message passing

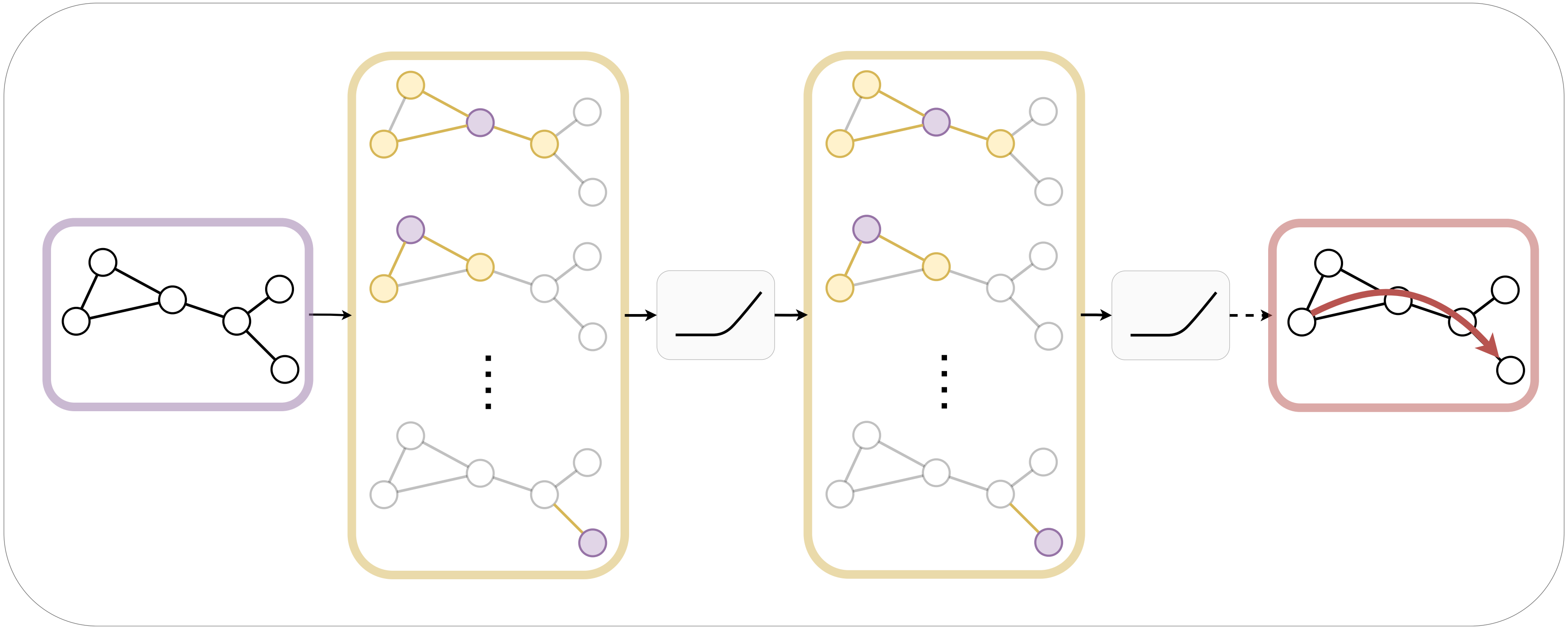
Input: Weight matrices, $\mathbf{W}_{\text{self}}$, $\mathbf{W}_{\text{neigh}}$, and bias, $\mathbf{b}$, neighborhood function, $\mathcal{N}$.

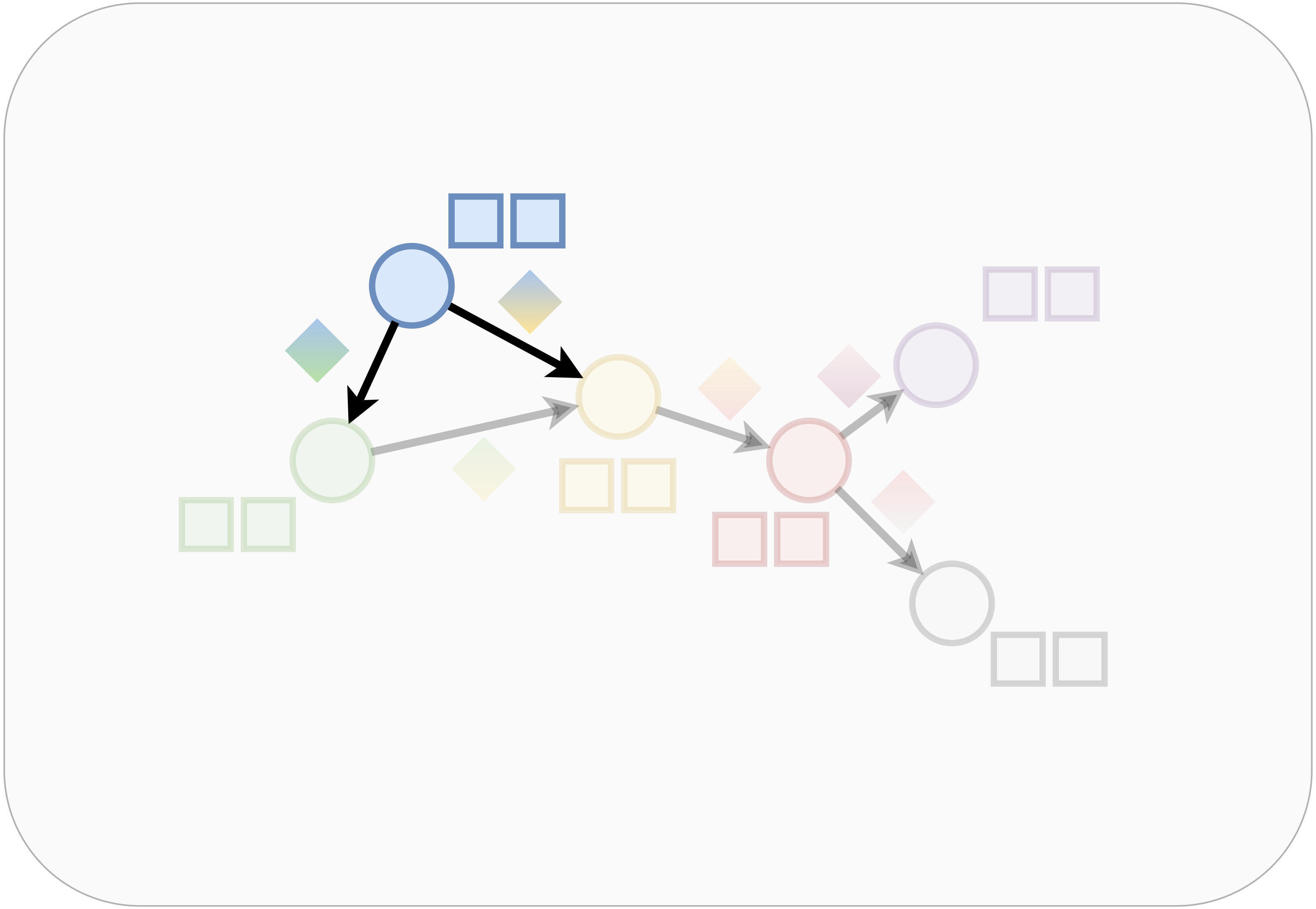
Input: Graph, $\mathcal{G}$ with nodes $\mathcal{V} = \{v_i\}_{i=0}^{V}$ and edges $\mathcal{E} = \{e_{u\to v} | u, v \in \mathcal{V}\}$, and a specified K number of rounds of message passing.

Output: Updated node features $\mathbf{h}_u^{(K+1)}$ for all nodes u

Initialize $\mathbf{h}_u^{(0)}$ as v_u for all nodes u
for $k \in [0, 1, \ldots, K]$ **do**
  for $u \in \mathcal{V}$ **do**
    for $v \in \mathcal{N}(u)$ **do**
      Compute messages : $\mathbf{M}_{v\to u} = \mathbf{W}_{\text{neighbors}} \mathbf{h}_v^{(k)} + \mathbf{b}$
    end for
    Compute self message: $\mathbf{M}_{\text{self}} = \mathbf{W}_{\text{self}} \mathbf{h}_u^{k}$
    Compute total message: $\mathbf{M}_u = \mathbf{M}_{\text{self}} + \sum_{v \in \mathcal{N}(u)} \mathbf{M}_{v\to u}$
    Update node: $\mathbf{h}_u^{(k+1)} \leftarrow \sigma(\mathbf{M}_u)$
  end for
end for

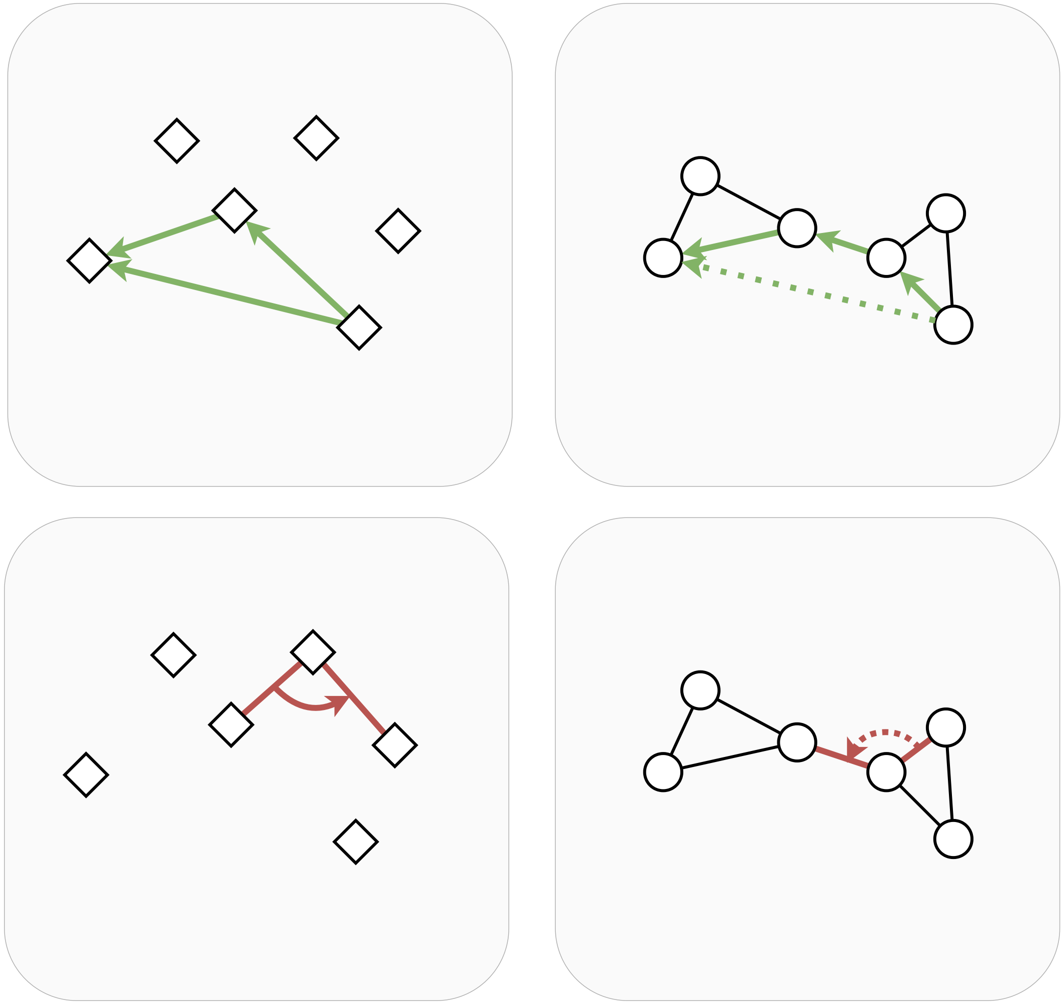




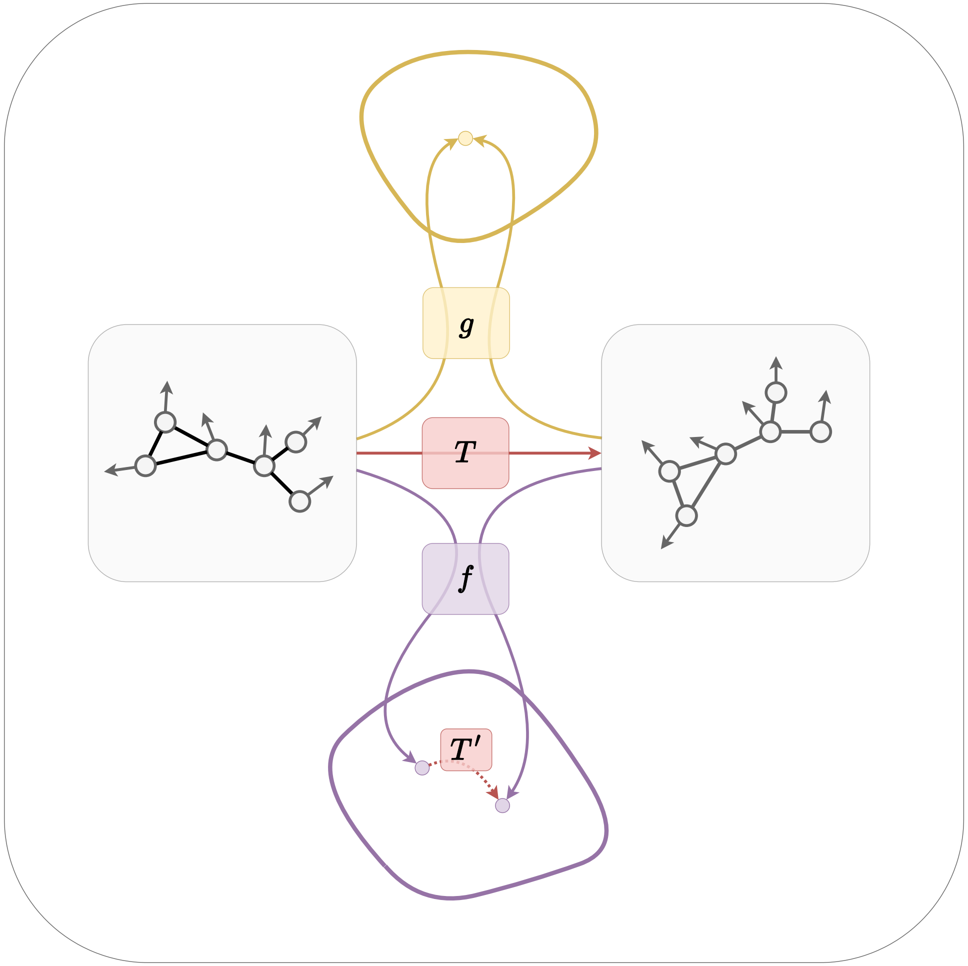


GEOMETRIC INFORMATION

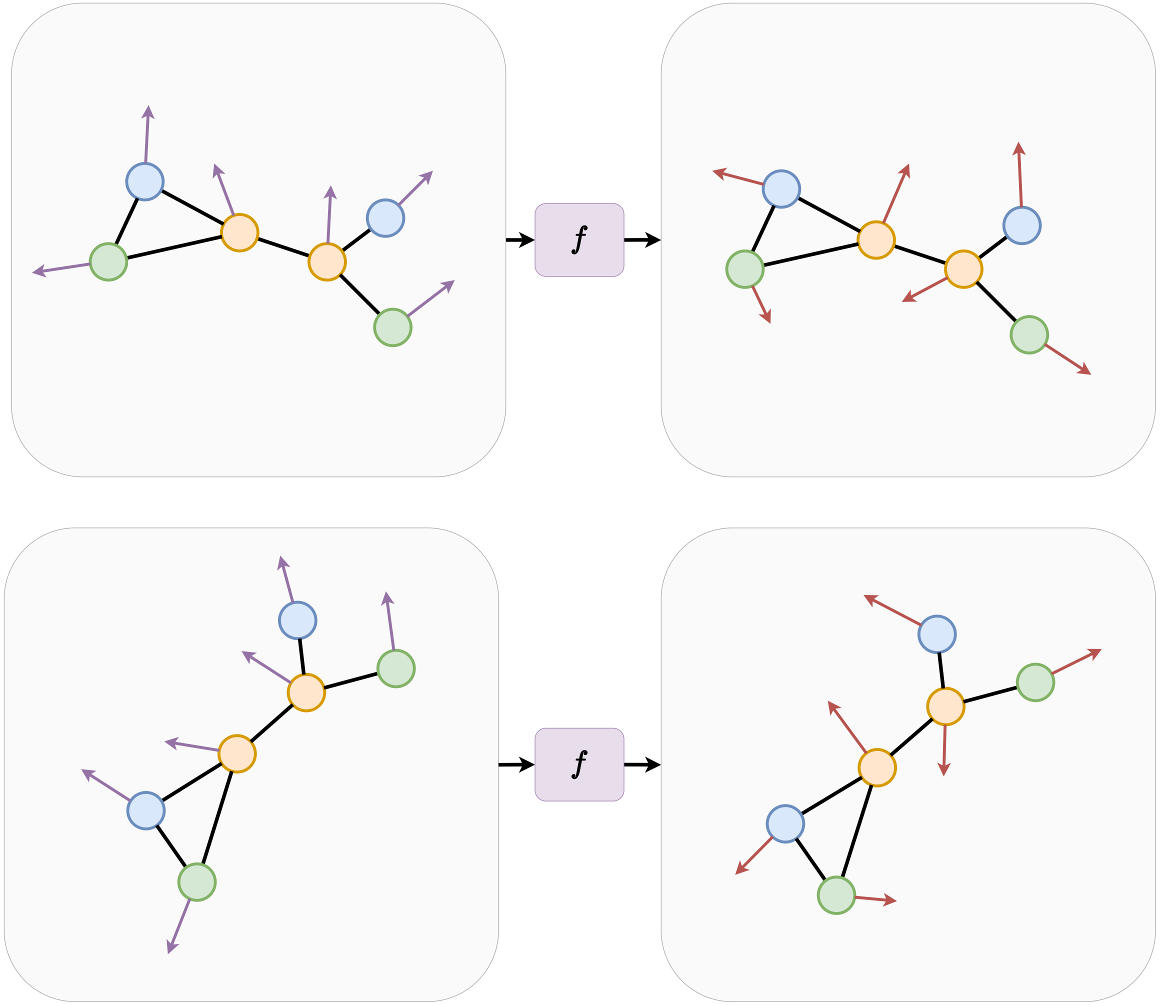
GEOMETRIC INFORMATION



ROTATION INVARIANCE AND EQUIVARIANCE



ROTATION INVARIANCE AND EQUIVARIANCE



ROTATION INVARIANCE AND EQUIVARIANCE