THE UNIVERSITY OF BRITISH COLUMBIA

Topics in AI (CPSC 532S):
Multimodal Learning with Vision, Language and Sound

Lecture 19: Generative Models [part 4]

Logistics

Research Paper Presentations:

— Papers are assigned last week (do not need to do one if you’re auditing)

— Presentations by Friday, **November 25th**

Research Paper Readings:

— Paper 2 is posted due Thursday

— Paper 3 & 4 of your choosing one per week

Logistics

Grading

— Assignment 3 & 4 being graded this week

— Project proposal documents will not be graded until later (do not wait for these)

Plans for the rest of term

— **Option 1**: Project presentations in Class (Dec 1 & 6), alt Dec 12 or 13

— **Option 2**: Discussion of research papers

— **Option 3**: I continue to lecture on additional topics

Generative Models — Overview

Pixel CNN/RNN — An explicit tractable density model. Accurate, easy to train but no latent variables.

VAE — An intractable density model with latent variables. Marginalizes the latent variables, but can only optimize the lower bound.

GANs — No explicit density model, just focused on sampling.

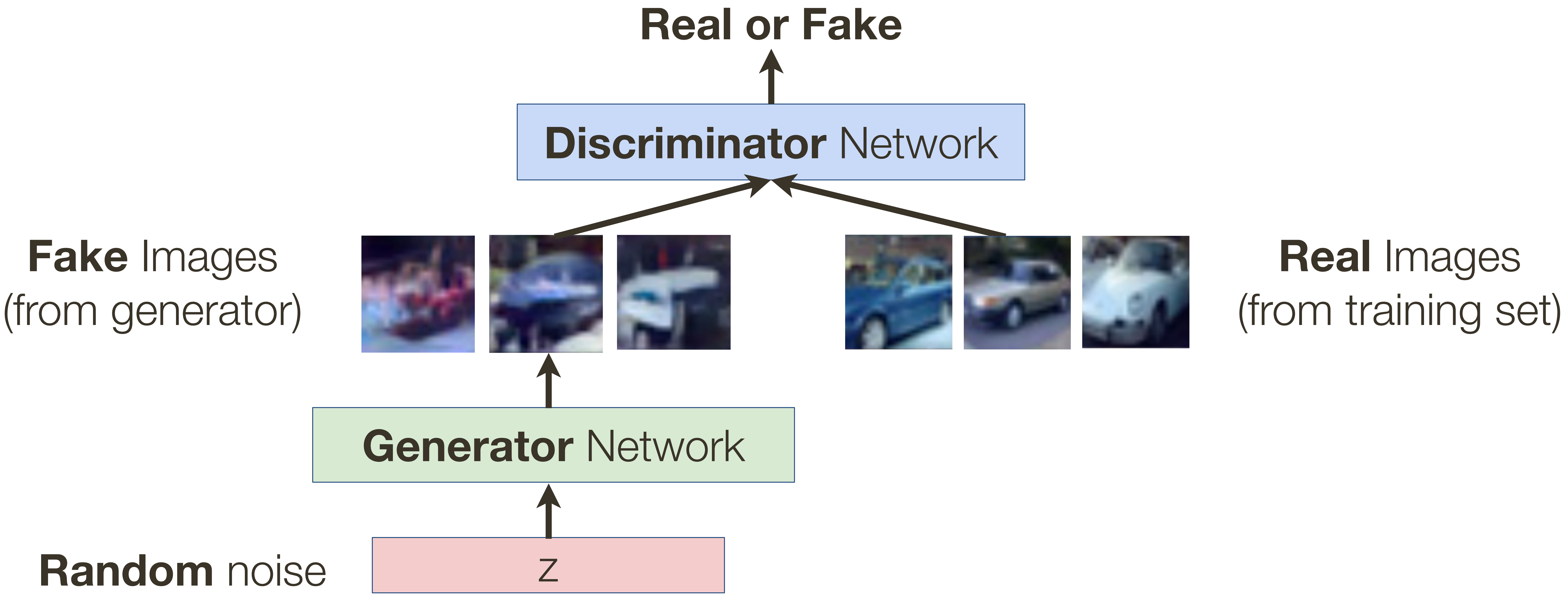
Diffusion Models — No explicit density model, similar to GANs in many ways.

Generative Adversarial Networks (GANs)

Training GANs: Two-player Game

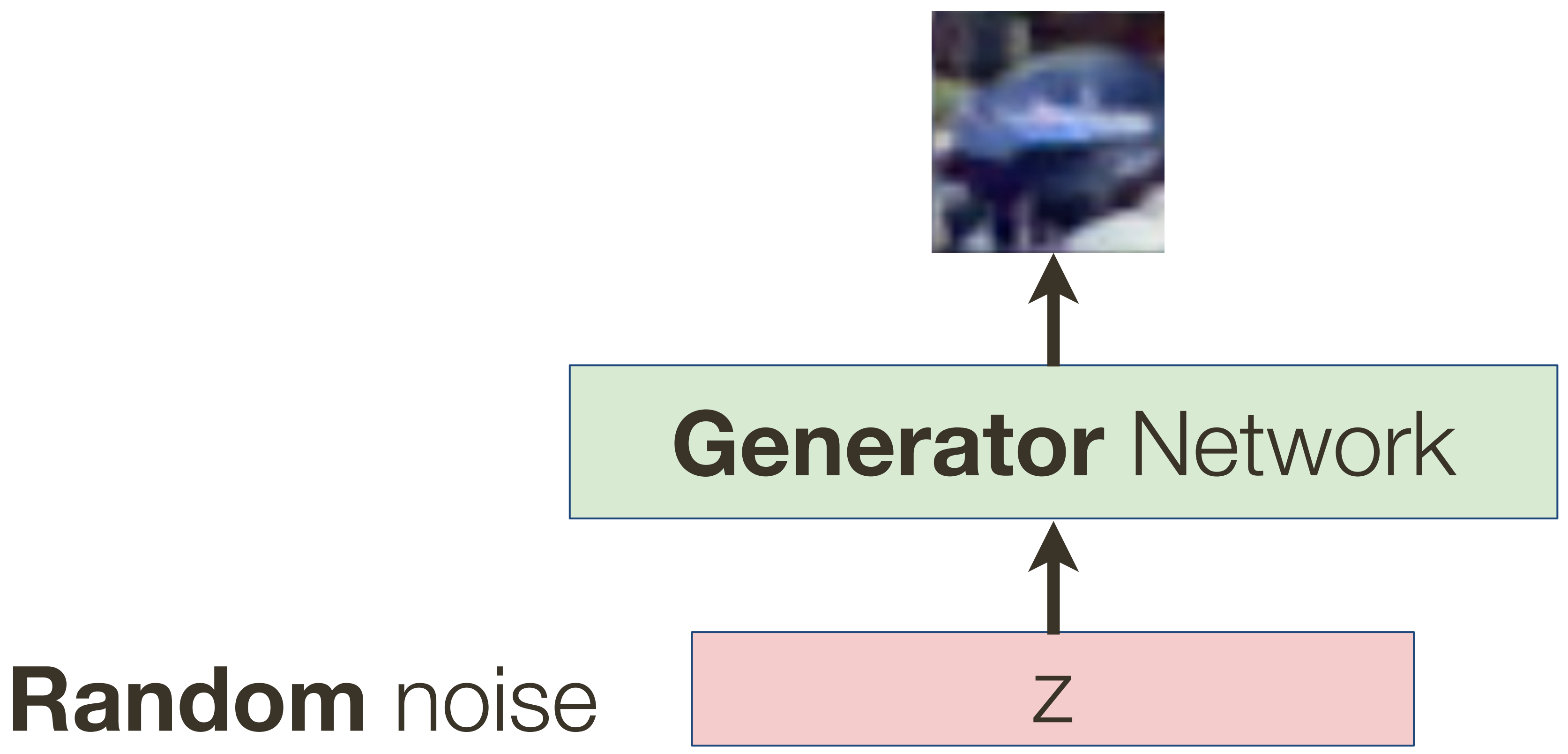
[Goodfellow et al., 2014]

Generator network: try to fool the discriminator by generating real-looking images
Discriminator network: try to distinguish between real and fake images



* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

Sampling **GAN**s



* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

Conditional GAN: Text-to-Image Synthesis

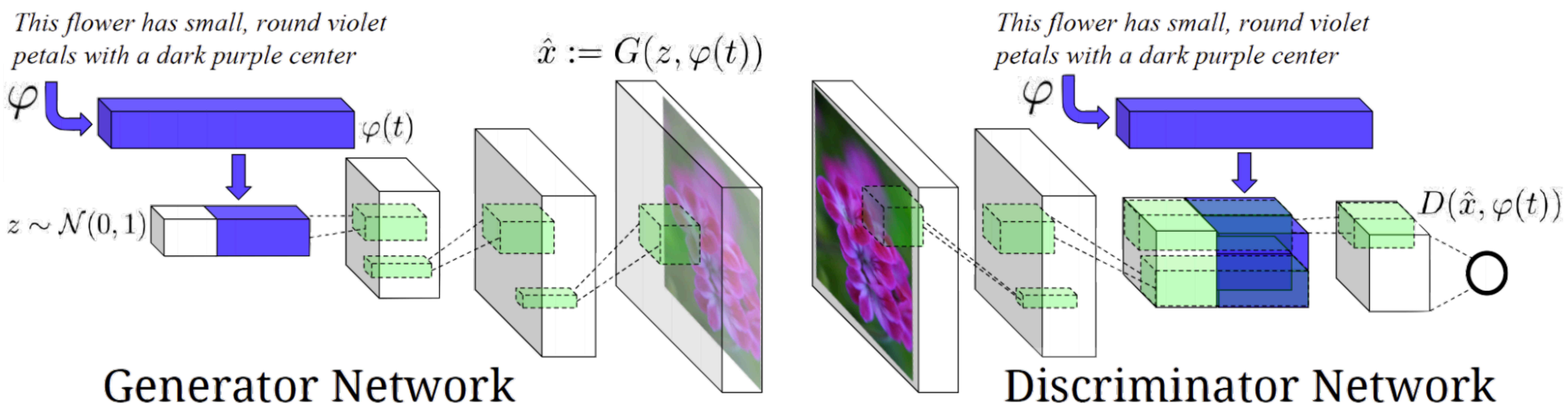


Figure 2 in the original paper.

Positive Example:

Real Image, Right Text

Negative Examples:

Real Image, Wrong Text

Fake Image, Right Text

[Reed et al., ICML 2016]

Conditional GAN: Image-to-Image translation

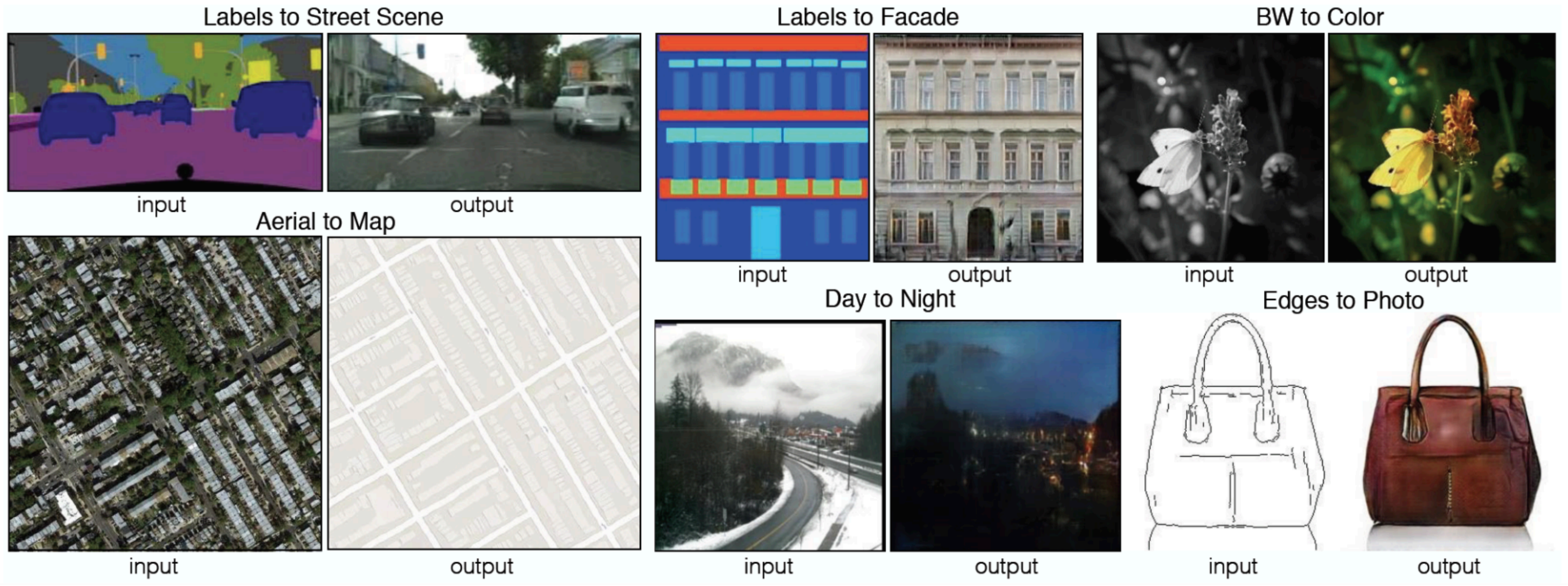


Figure 1 in the original paper.

[Isola et al., 2016]

Conditional GAN: Image-to-Image translation

Architecture: DCGAN-based

Training is conditioned on the **images from the source domain**

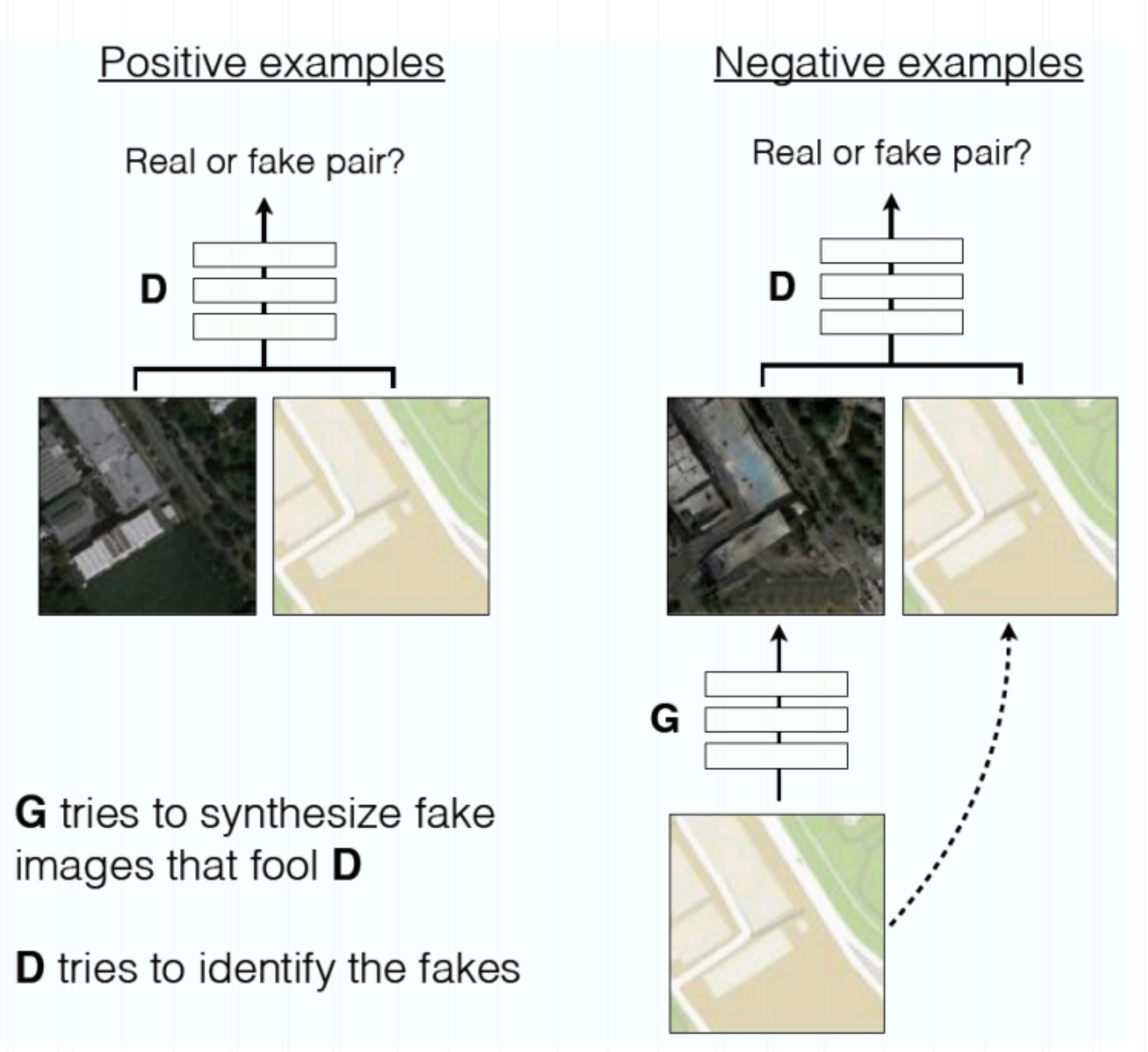
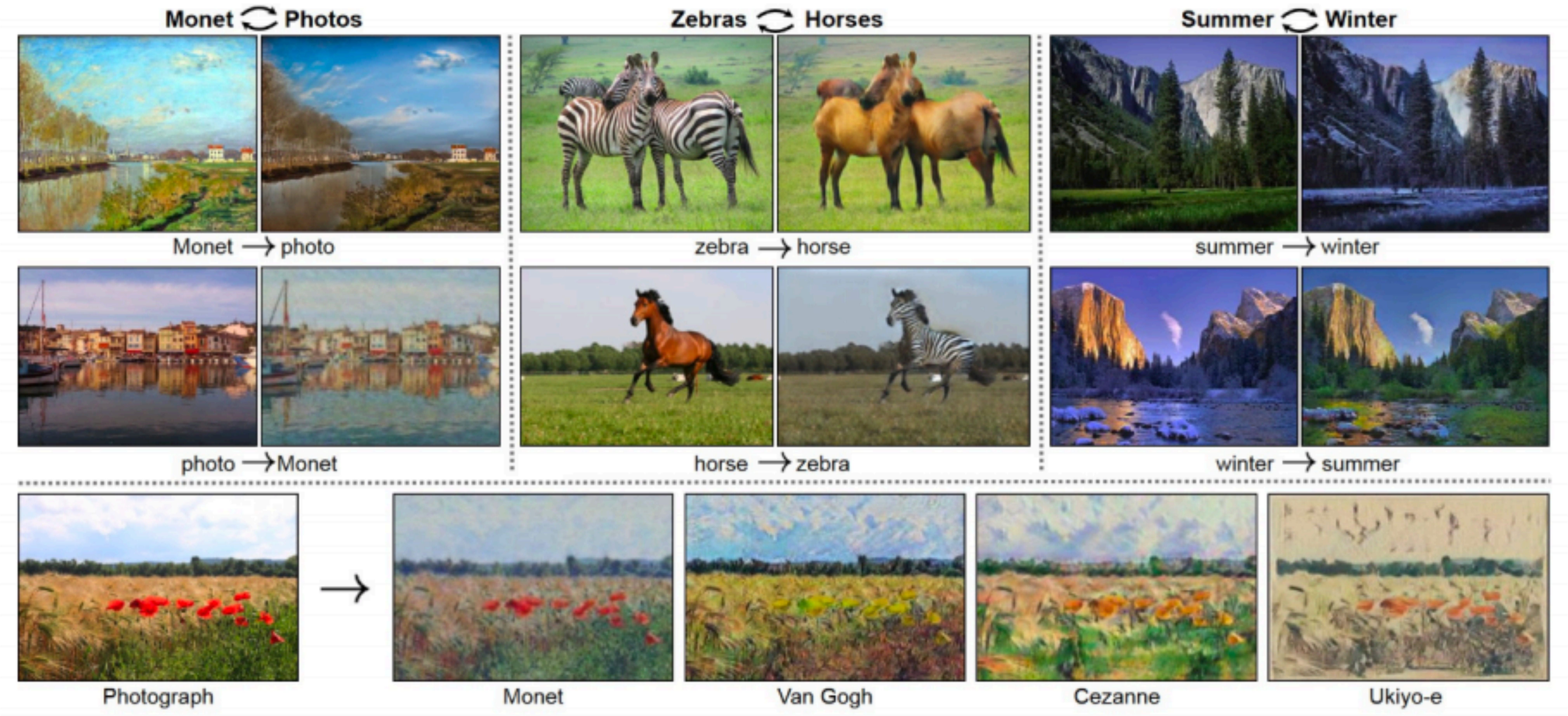


Figure 2 in the original paper.

[Isola et al., 2016]

CycleGAN: Unpaired Image-to-Image translation

Style transfer: change the style of an image while preserving the content



Data: two unrelated collections of image, one for each style

[Zhu et al., 2017]

CycleGAN: Unpaired Image-to-Image translation

Style transfer: change the style of an image while preserving the content

— Train **two different generator networks** to go from Style 1 to Style 2 and vice versa

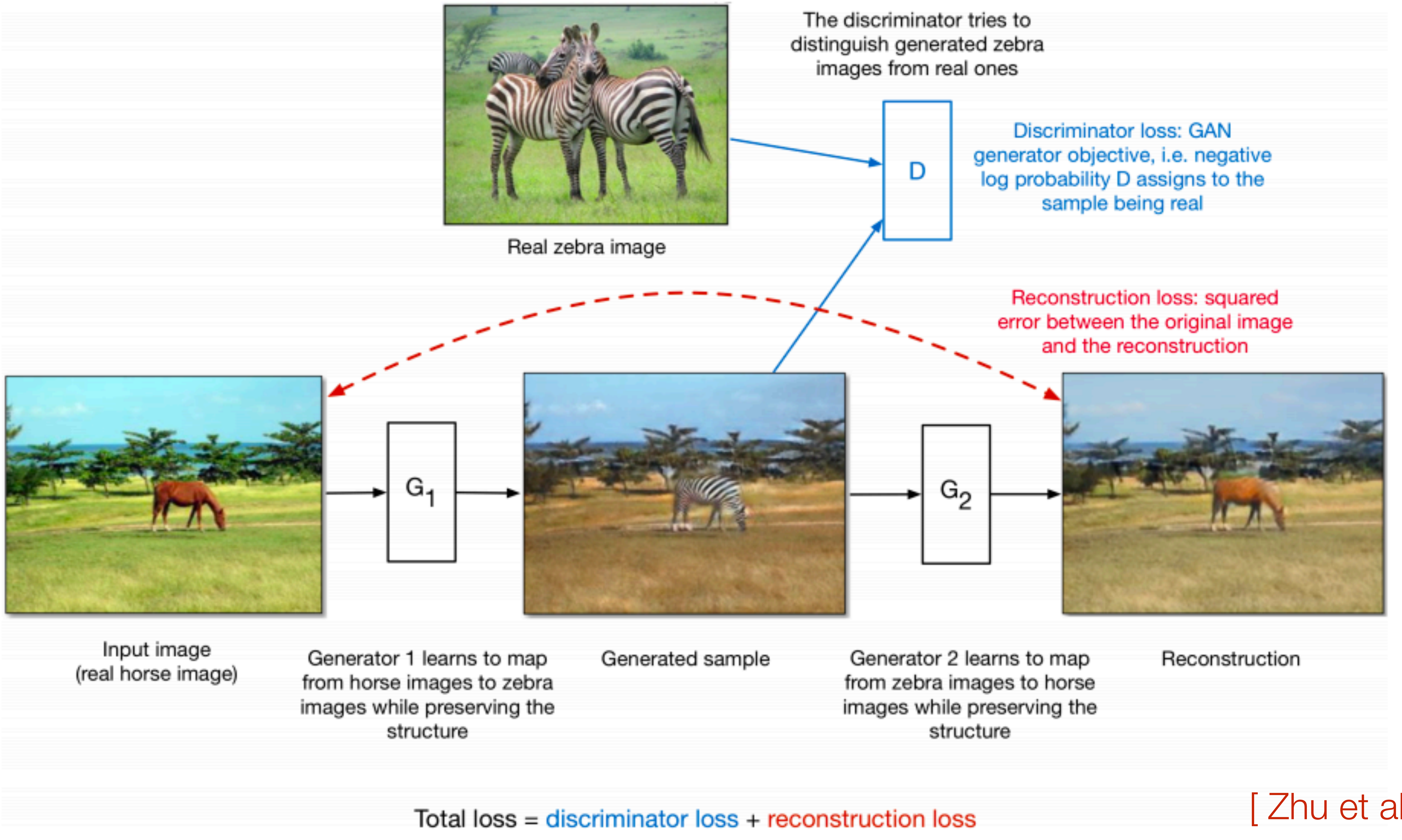
— Make sure the generated (translated) samples of Style 2 are indistinguishable from real images of Style 2 by a discriminator network

— Make sure the generated (translated) samples of Style 1 are indistinguishable from real images of Style 1 by a discriminator network

— Make sure the generators are **cycle-consistent**: mapping Style1 -> Style 2 -> Style 1 should give close to the original image

[Zhu et al., 2017]

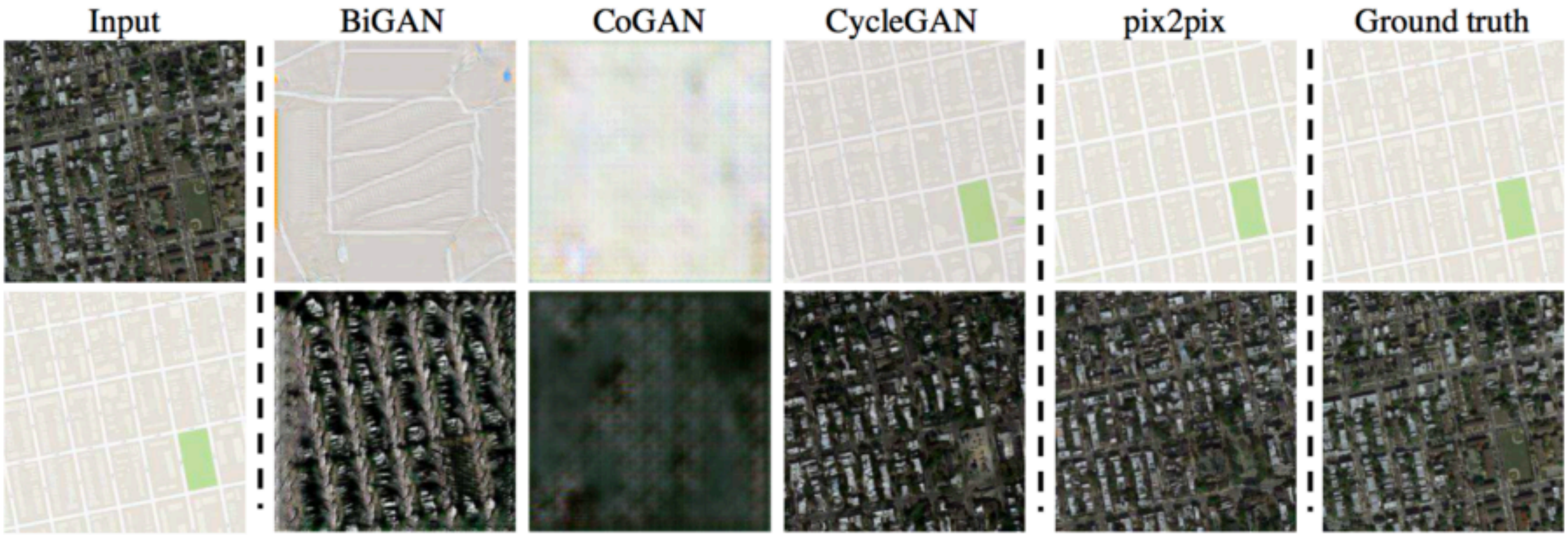
CycleGAN: Unpaired Image-to-Image translation



[Zhu et al., 2017]

CycleGAN: Unpaired Image-to-Image translation

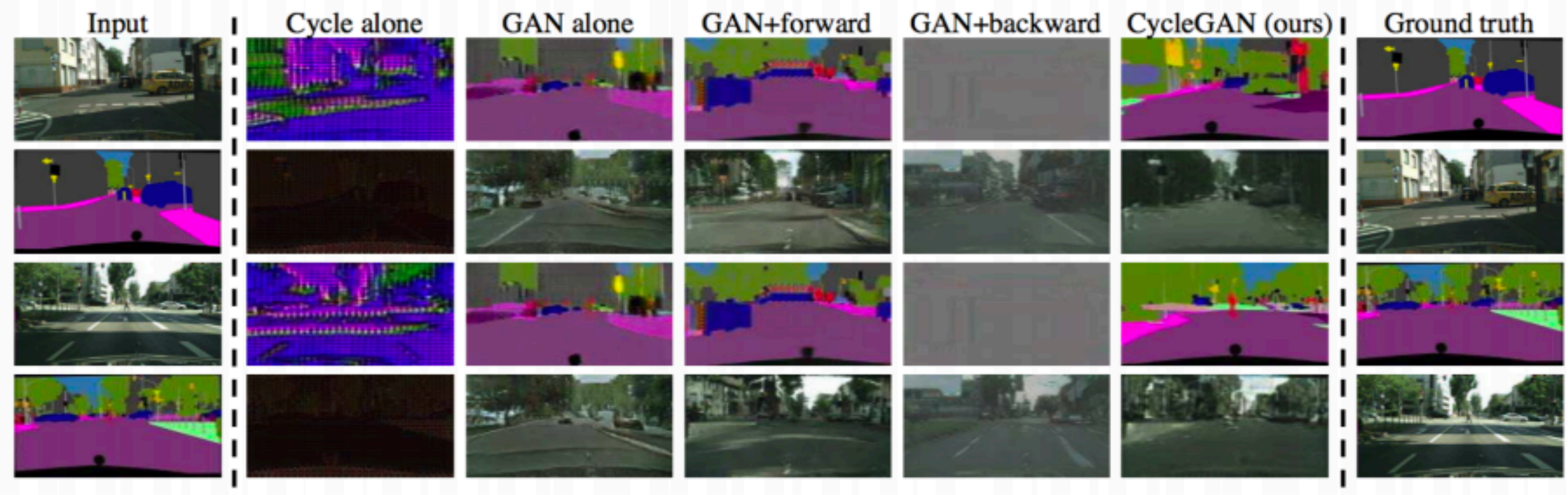
Ariel photos to **maps:**



[Zhu et al., 2017]

CycleGAN: Unpaired Image-to-Image translation

Images to **semantic segmentation:**



[Zhu et al., 2017]

Laplacian Pyramid GAN

Building a **Laplacian Pyramid** — a loss-less hierarchical representation of image

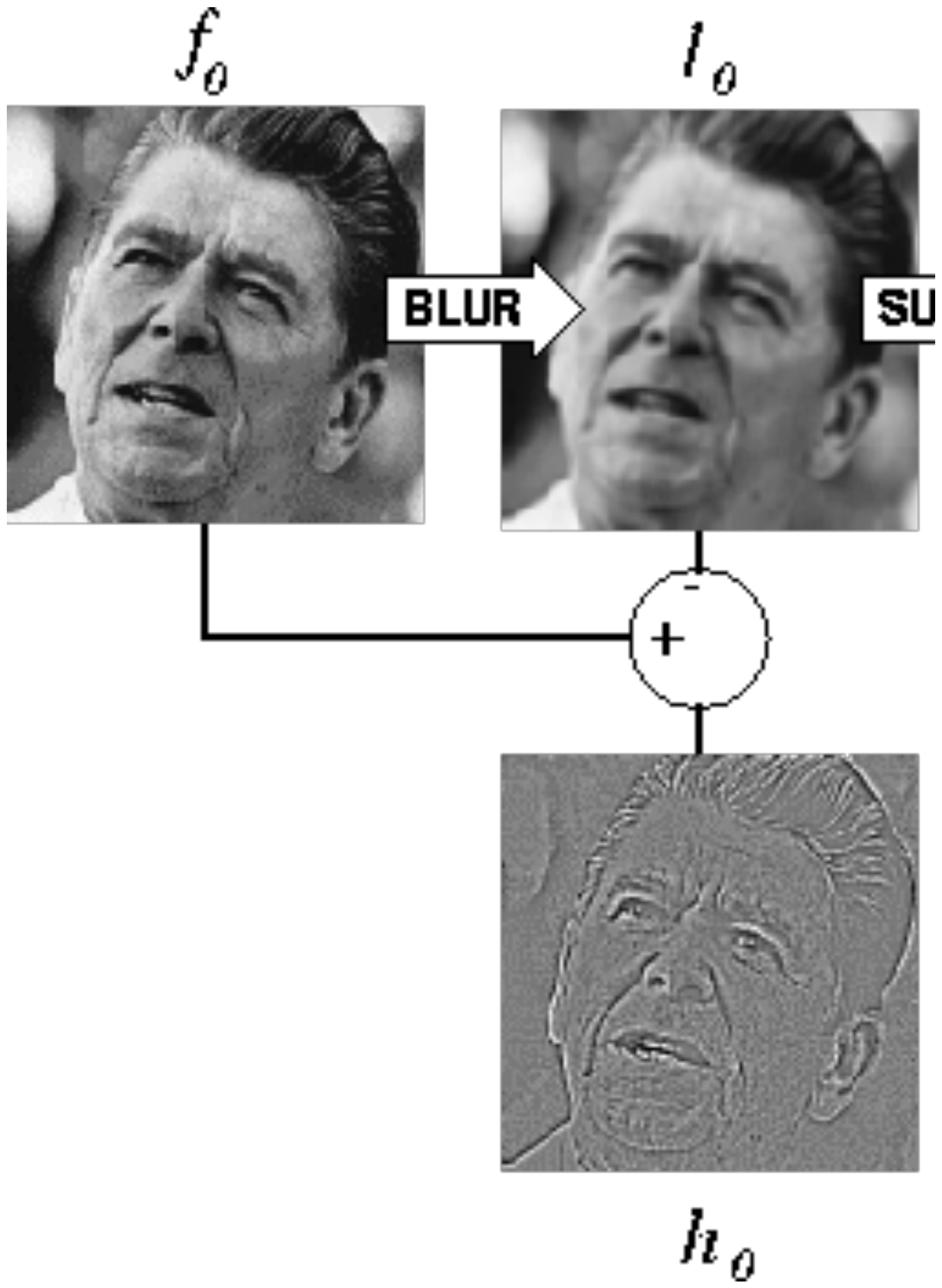
f_0

Laplacian Pyramid GAN

Building a **Laplacian Pyramid** — a loss-less hierarchical representation of image

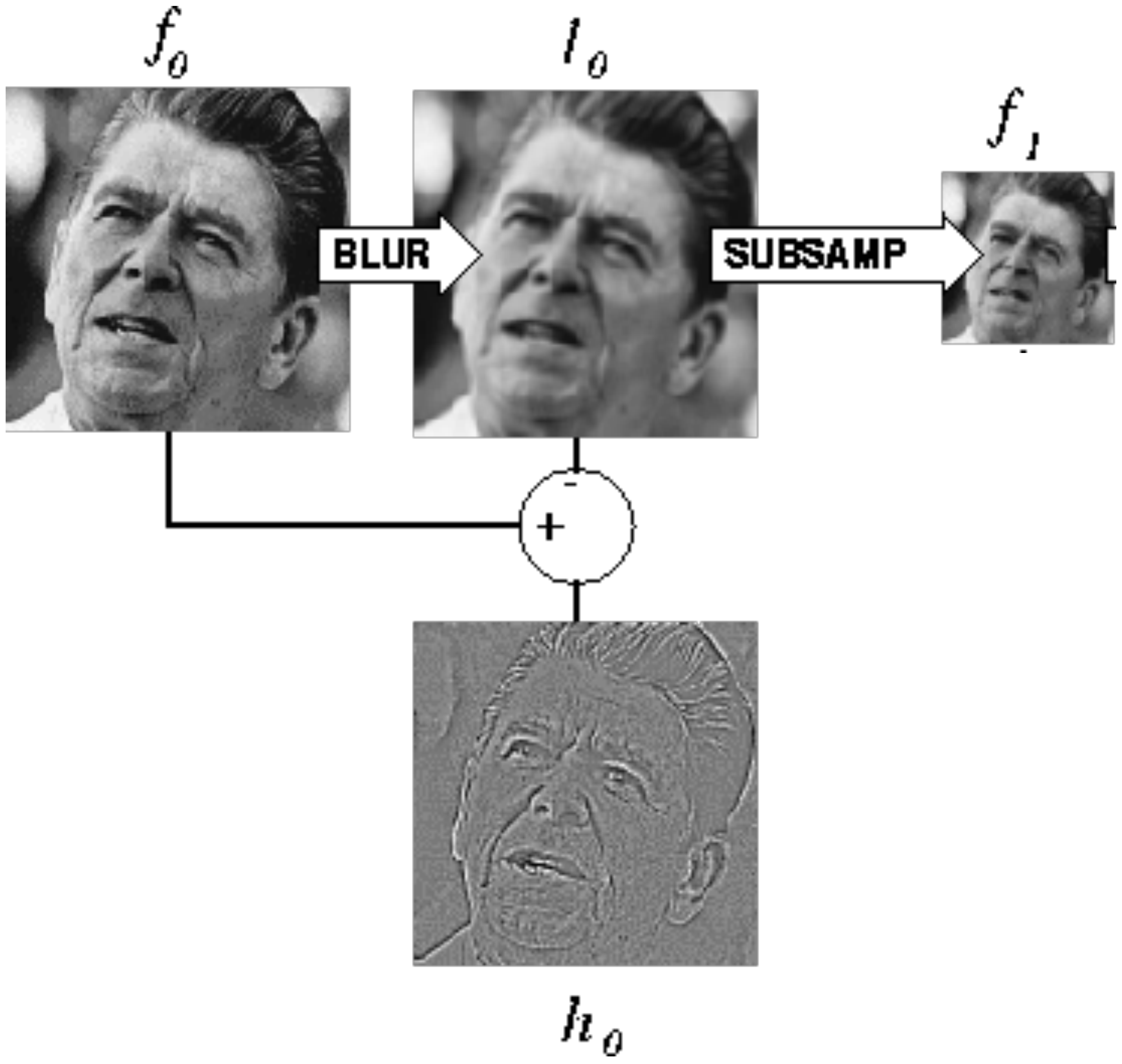
Laplacian Pyramid GAN

Building a **Laplacian Pyramid** — a loss-less hierarchical representation of image



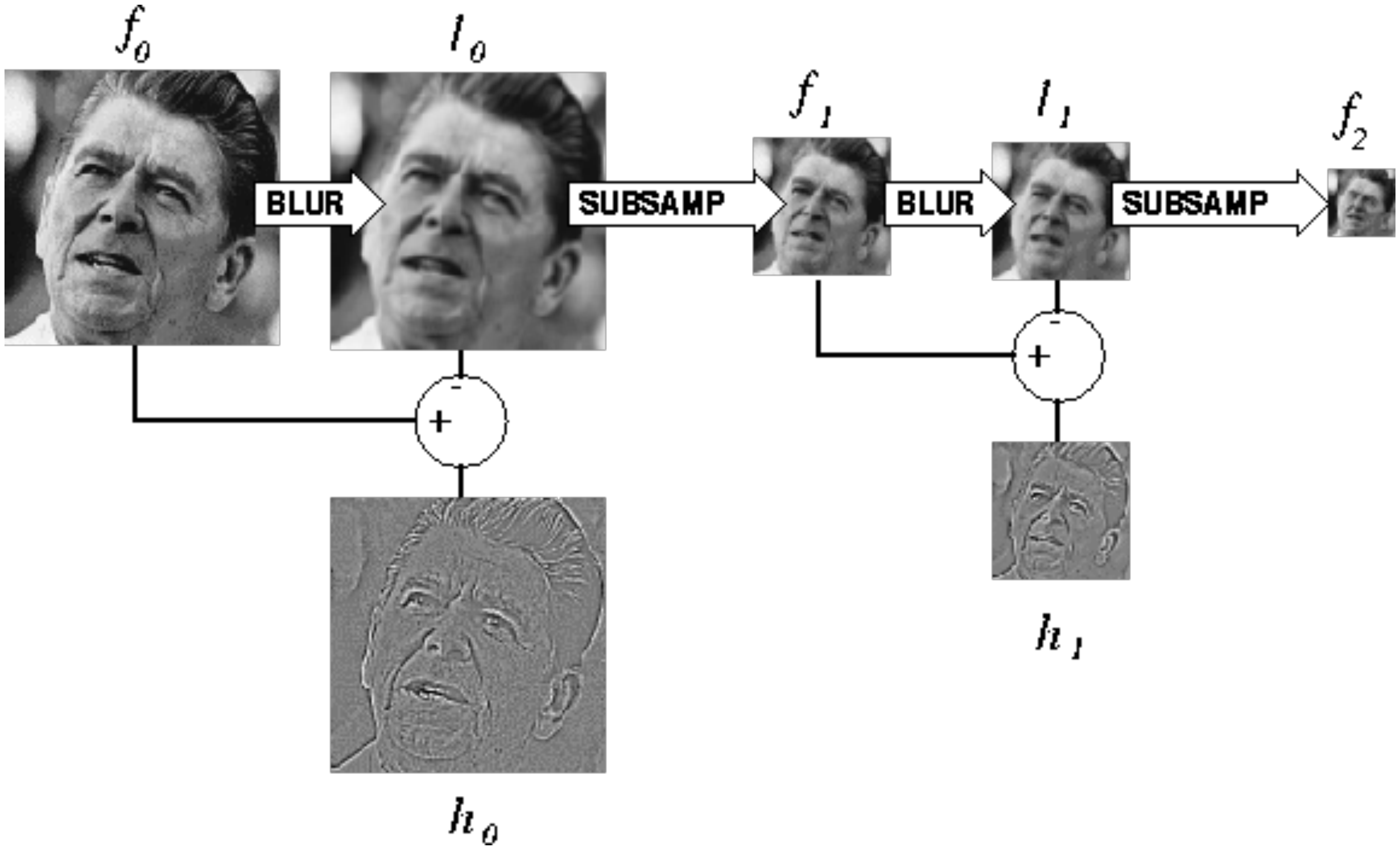
Laplacian Pyramid GAN

Building a **Laplacian Pyramid** — a loss-less hierarchical representation of image



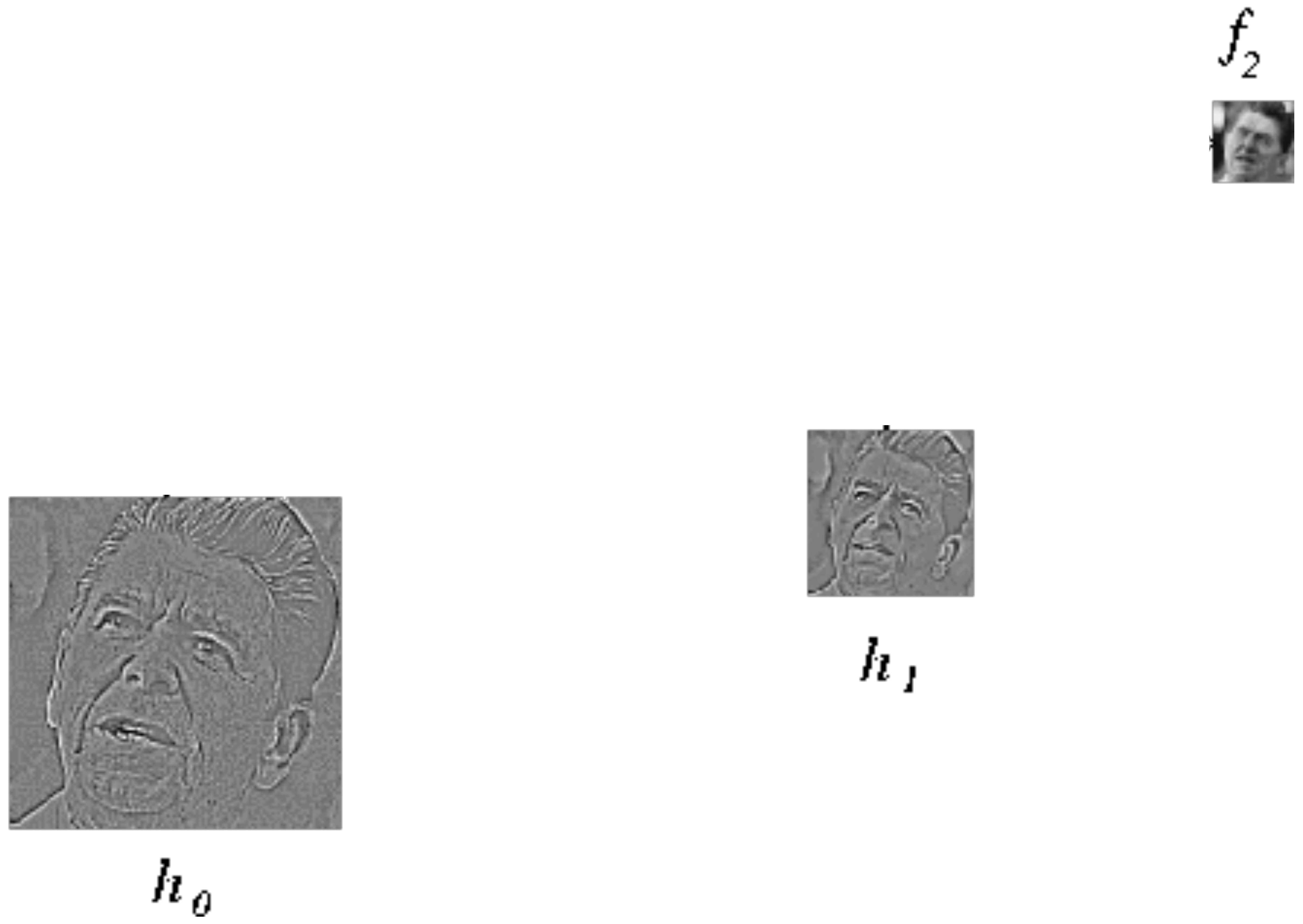
Laplacian Pyramid GAN

Building a **Laplacian Pyramid** — a loss-less hierarchical representation of image



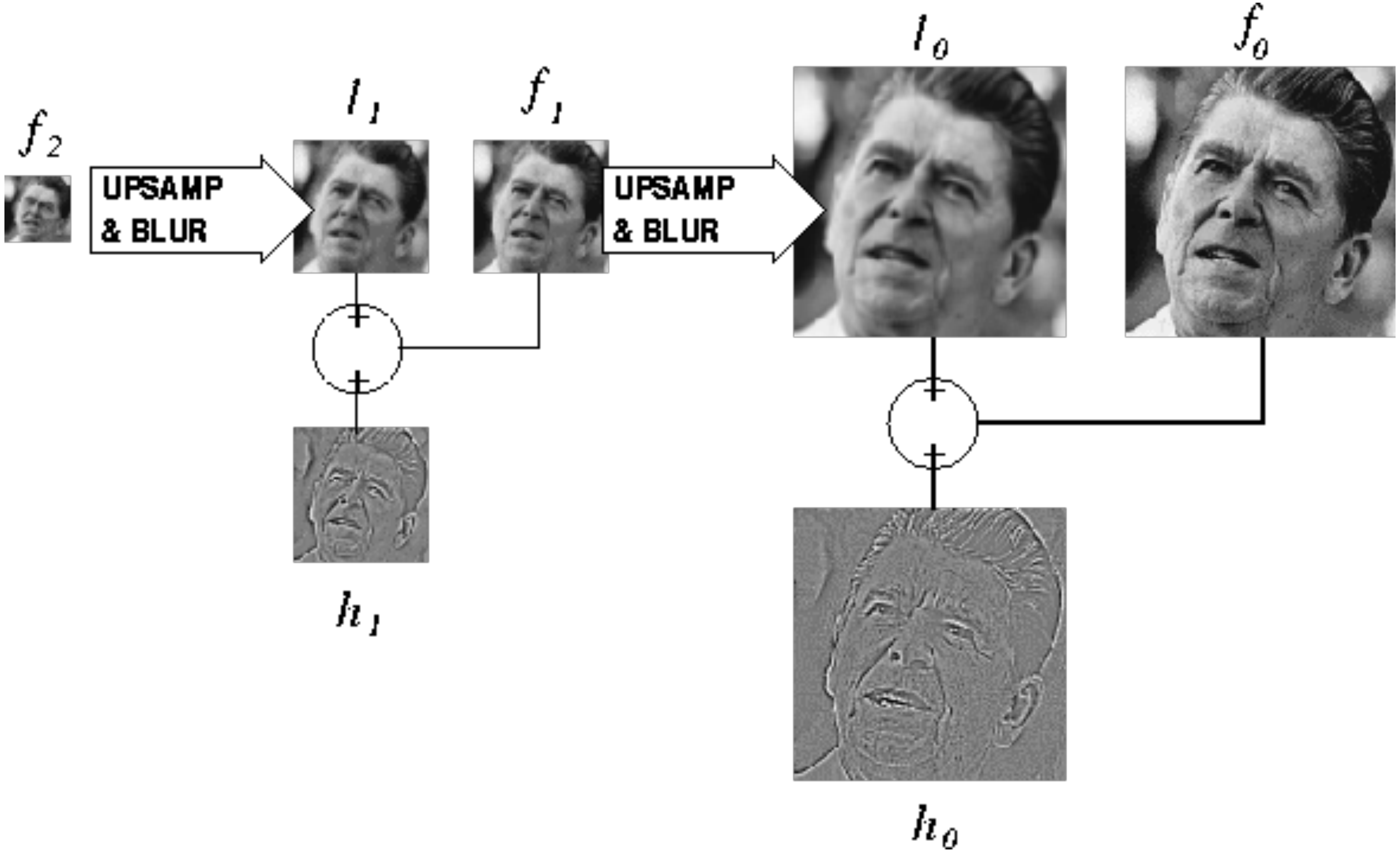
Laplacian Pyramid GAN

Building a **Laplacian Pyramid** — a loss-less hierarchical representation of image



Laplacian Pyramid GAN

Reconstructing a **Laplacian Pyramid**



Laplacian Pyramid GAN

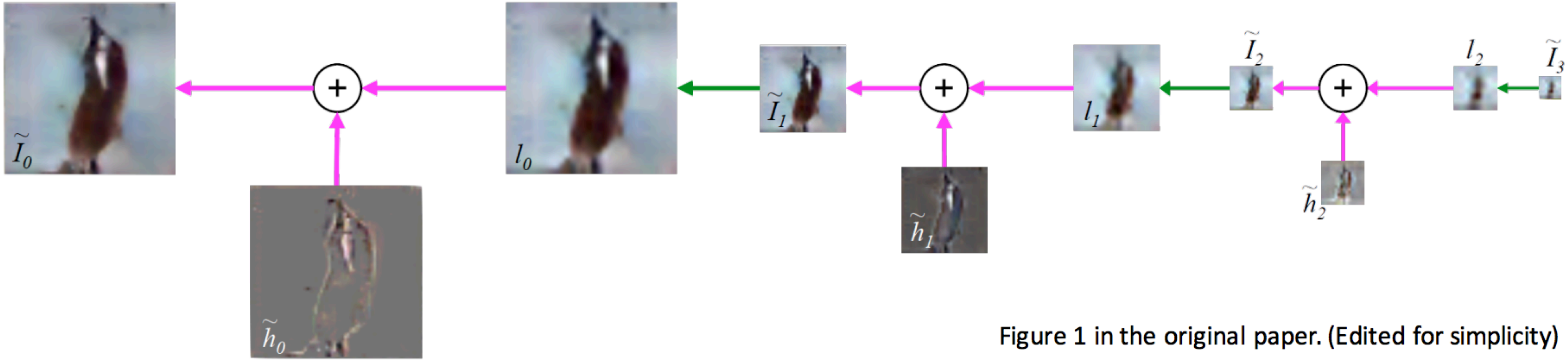


Figure 1 in the original paper. (Edited for simplicity)

— Based on the **Laplacian Pyramid** representation of images

— Generates high resolution images by using **hierarchical set of GANs** by iteratively increasing image resolution and quality

[Denton et al., 2015]

Laplacian Pyramid GAN

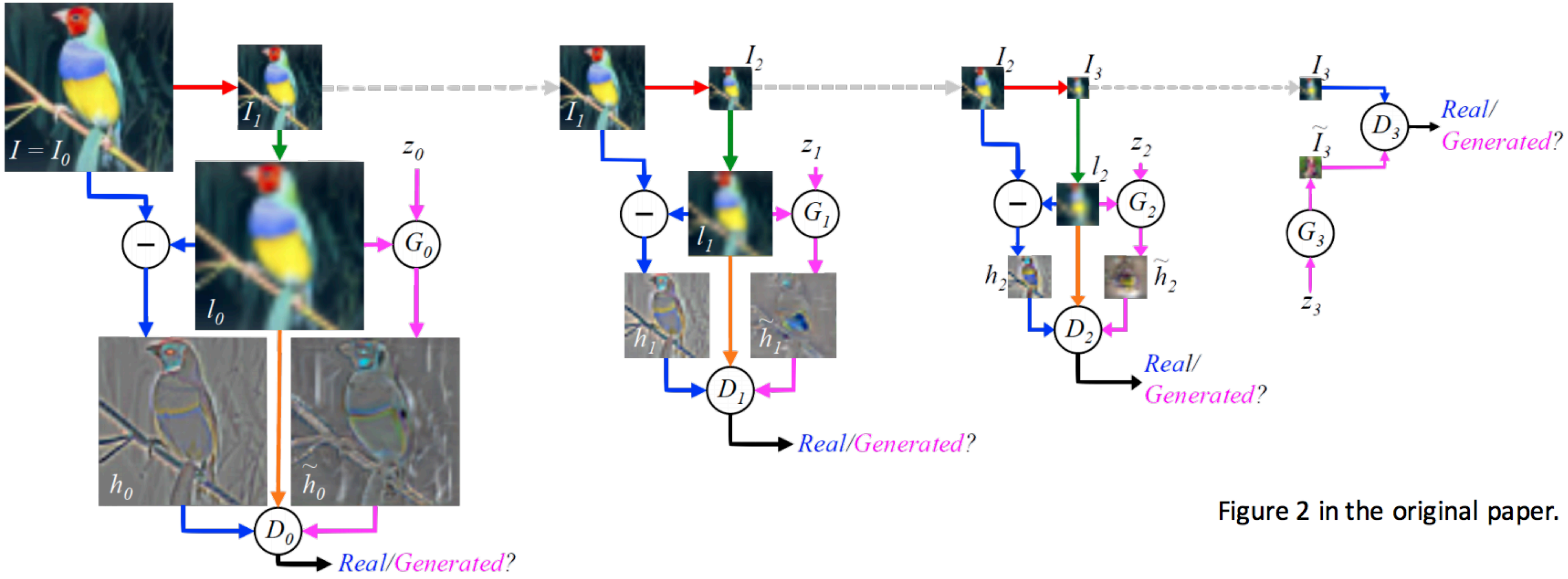


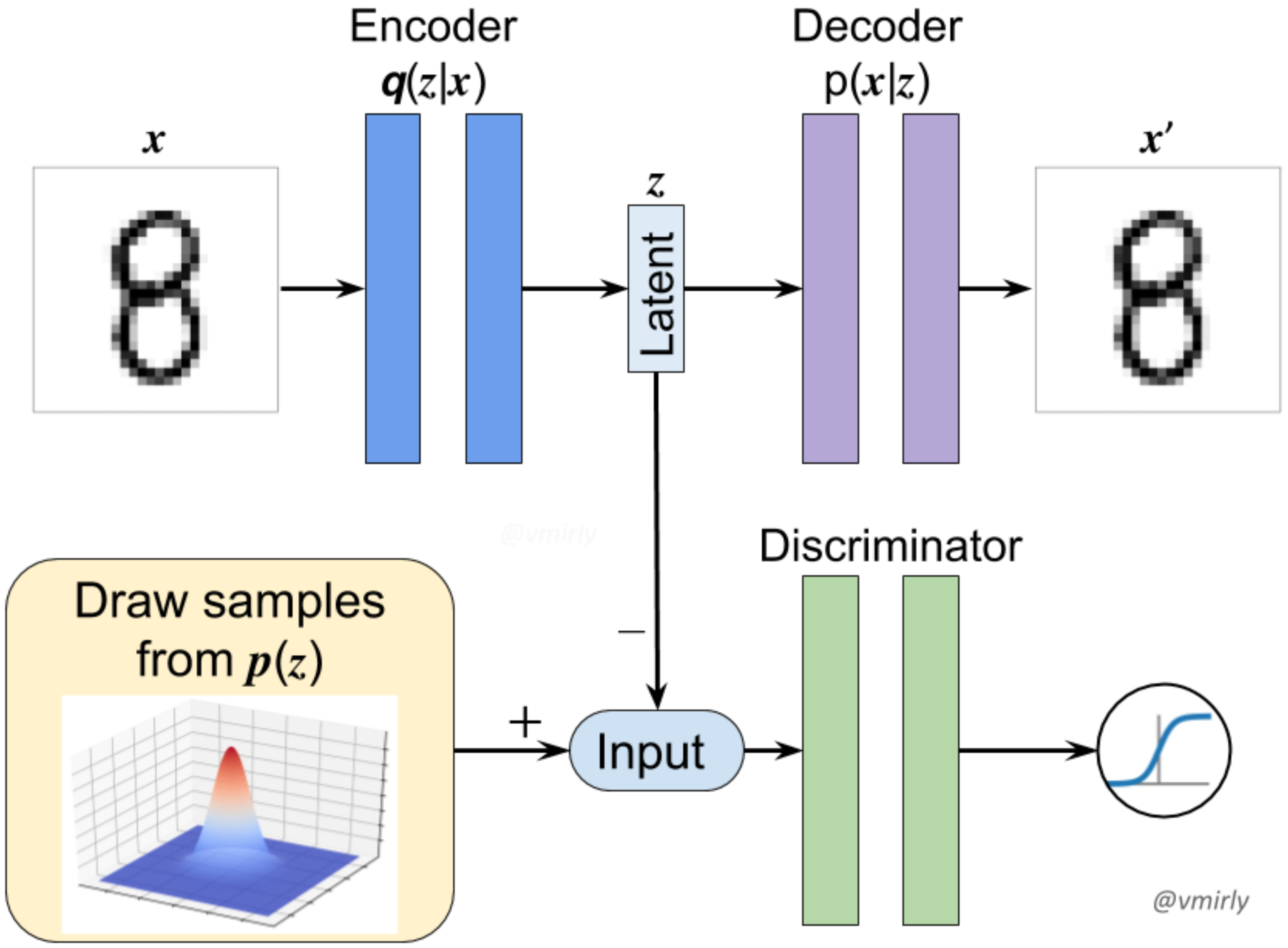
Figure 2 in the original paper.

— Based on the **Laplacian Pyramid** representation of images

— Generates high resolution images by using **hierarchical set of GANs** by iteratively increasing image resolution and quality

[Denton et al., 2015]

Adversarial Autoencoder (GAN + VAE)



[Makhzani et al., 2015]

CVPR LONG BEACH CALIFORNIA
June 16-20, 2019

Image Generation from Layout


Bo Zhao


Lili Meng

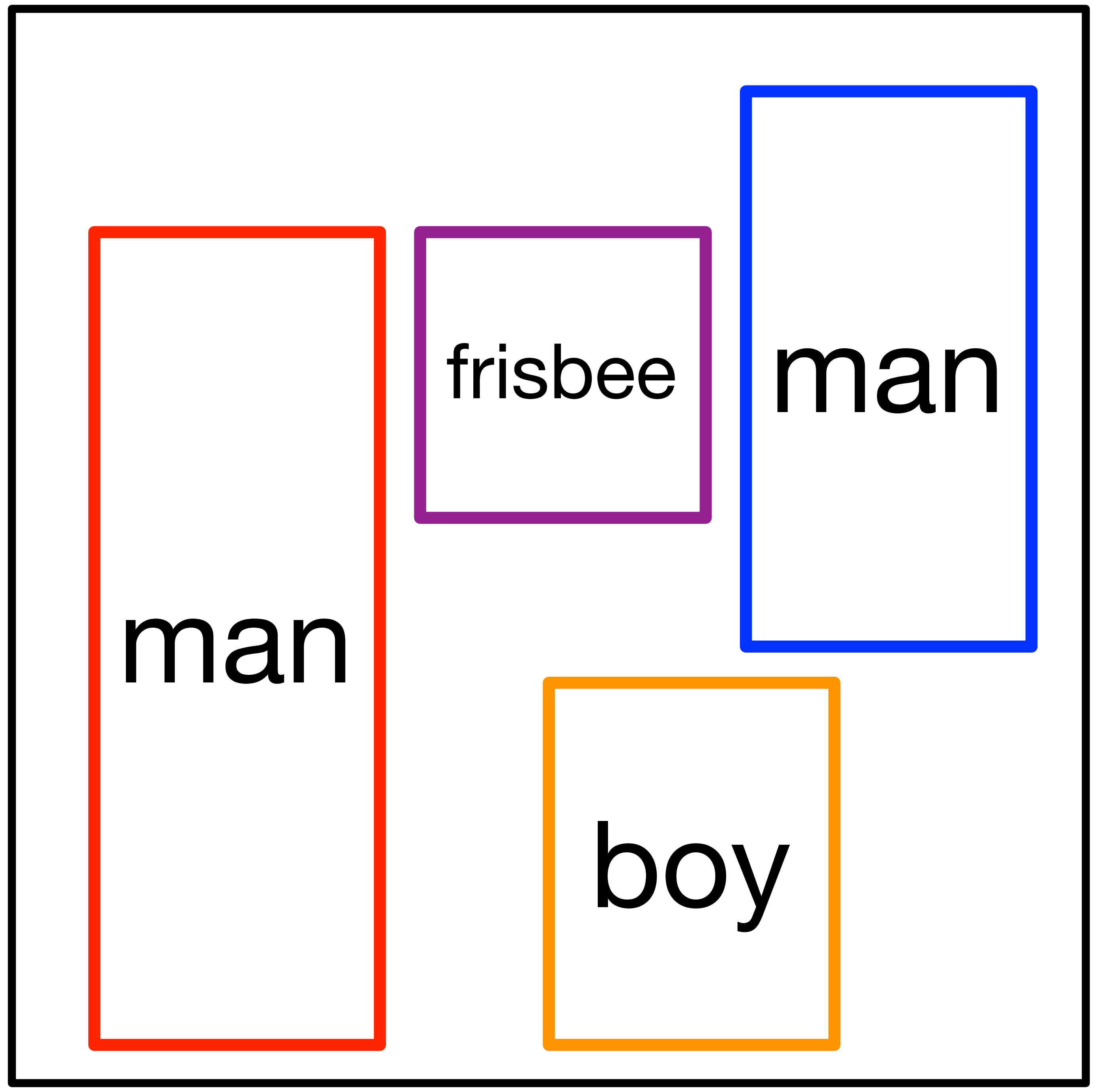

Weidong Yin


Leonid Sigal

UBC

Image Generation from Layout

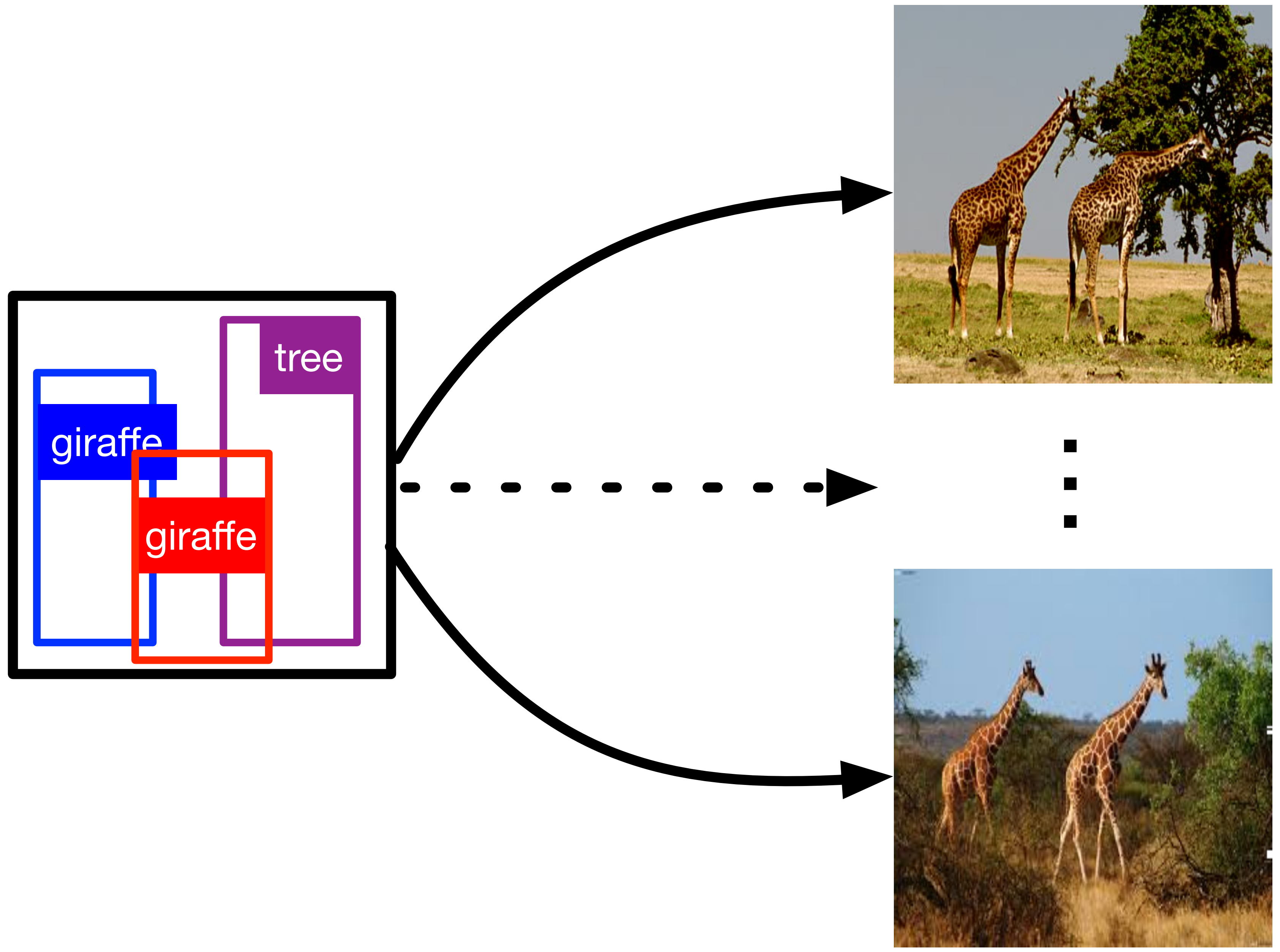
Image Generation from Layout



Layout

Results

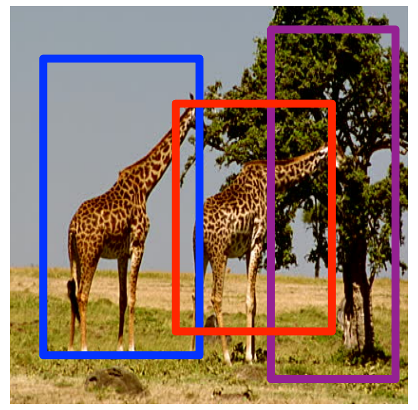
Image Generation from Layout: **Challenges**



- One-to-many mapping
- Information in layout is limited (but important)
- Important interactions between objects in overlap regions and with scene

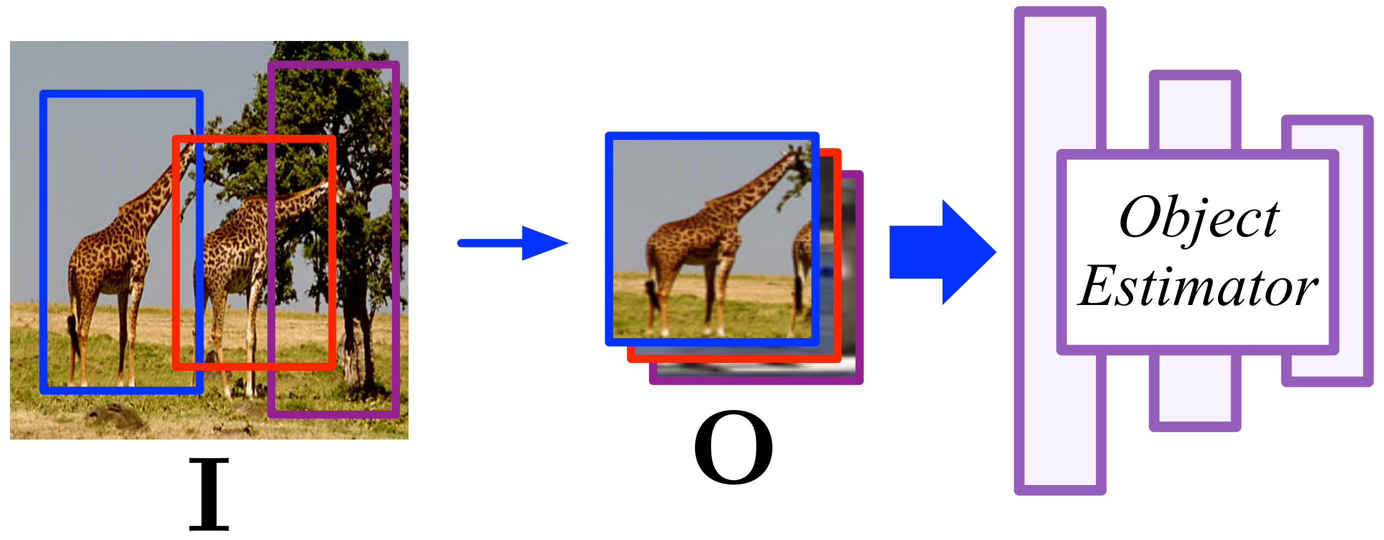
Model **Architecture**: Training

Model **Architecture**: Training

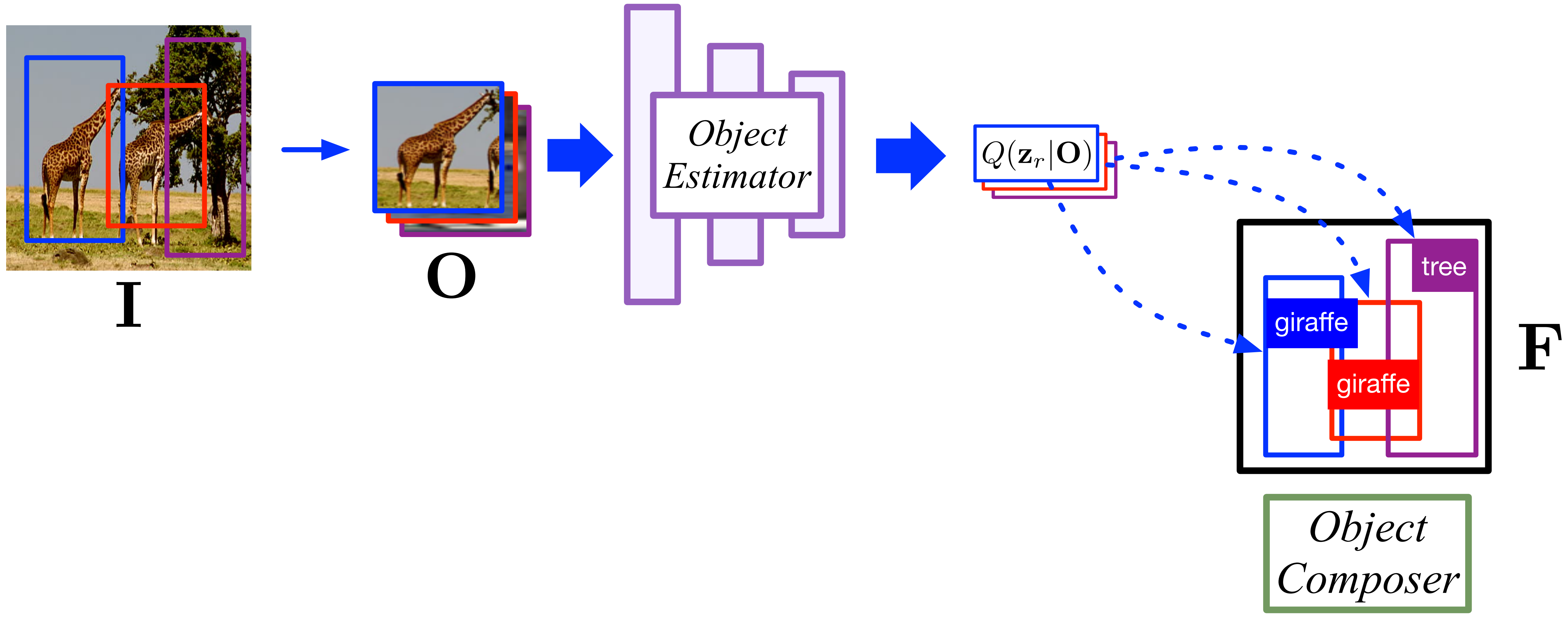


$\mathbf{I}$

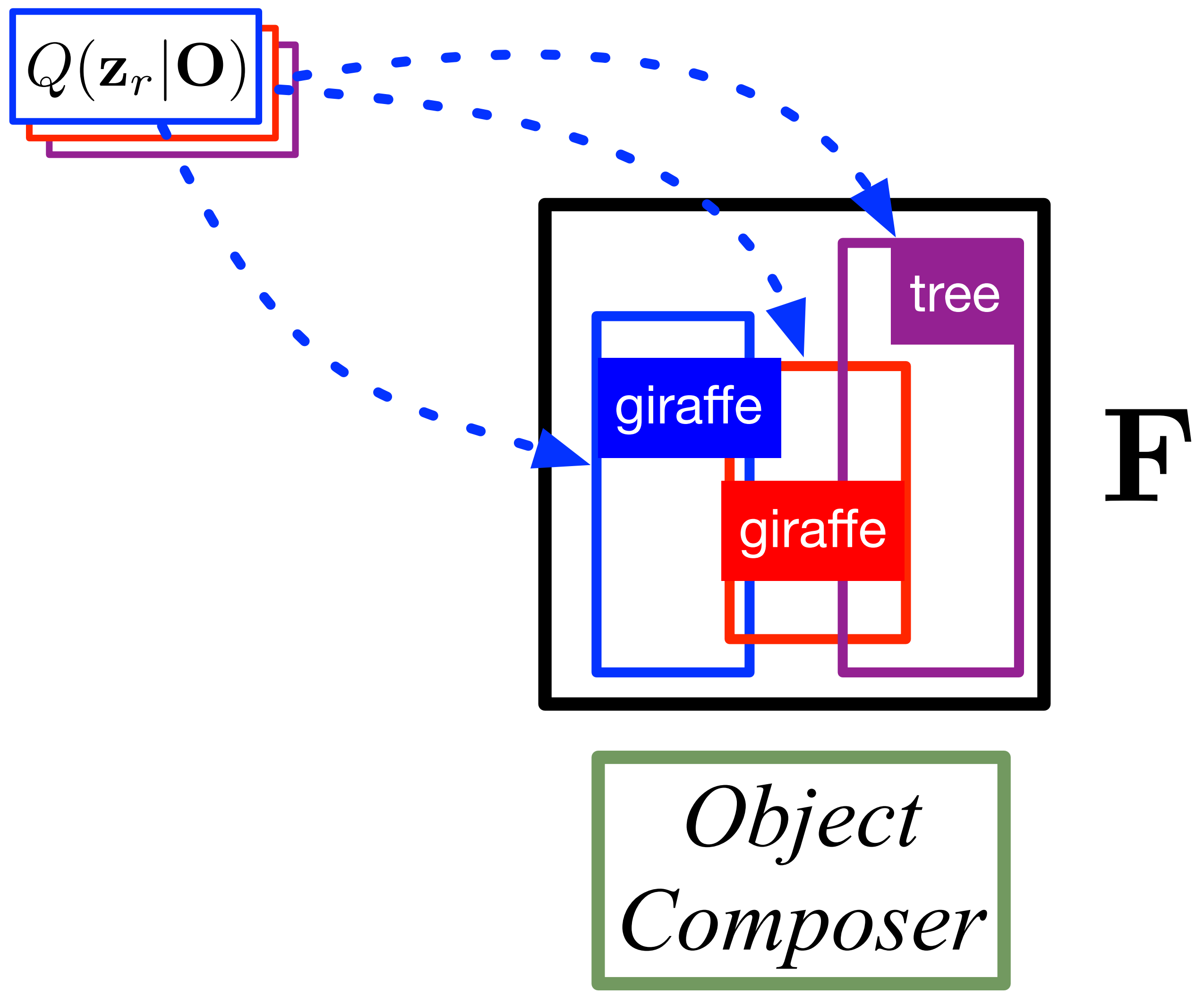
Model **Architecture**: Training



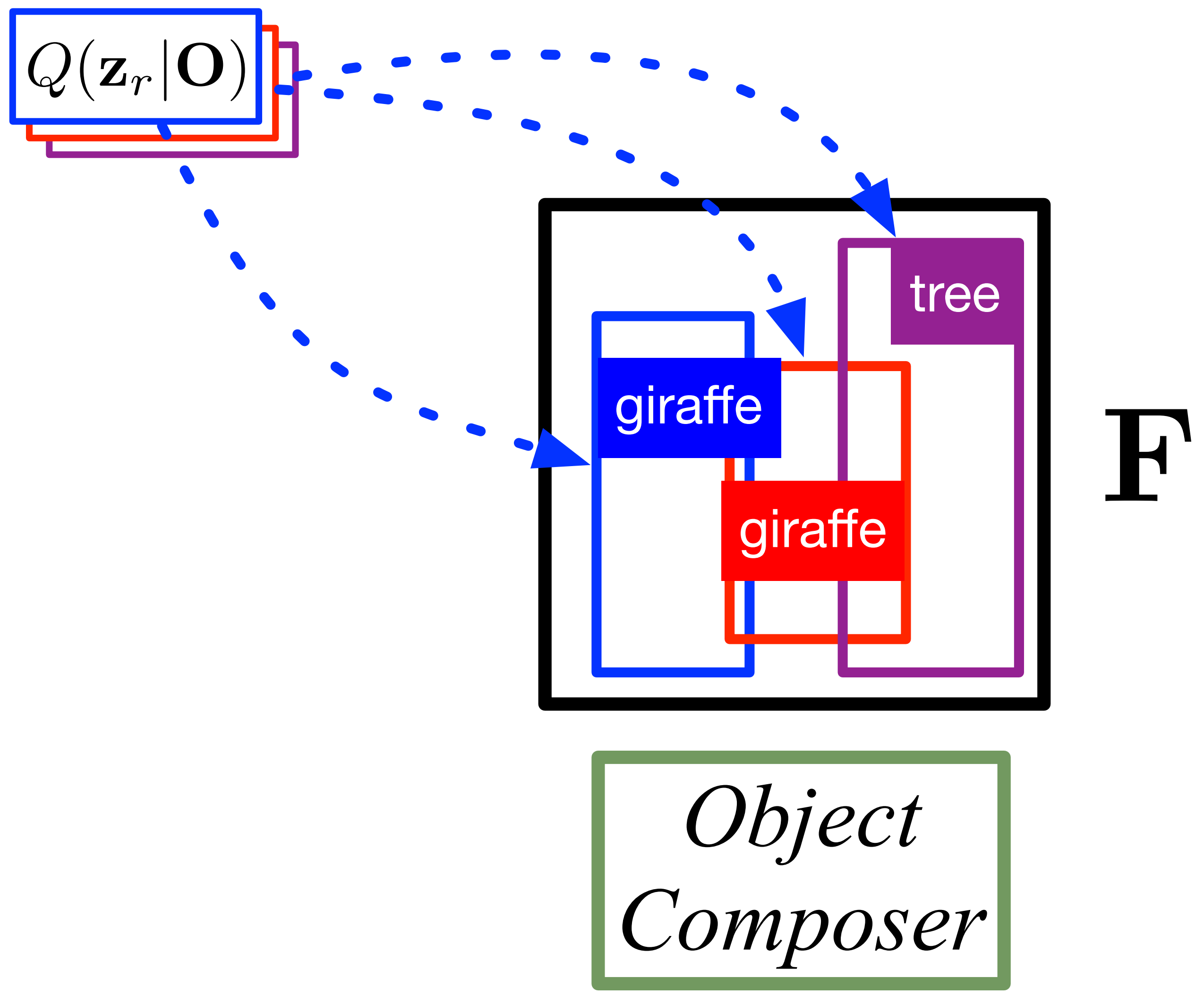
Model **Architecture**: Training

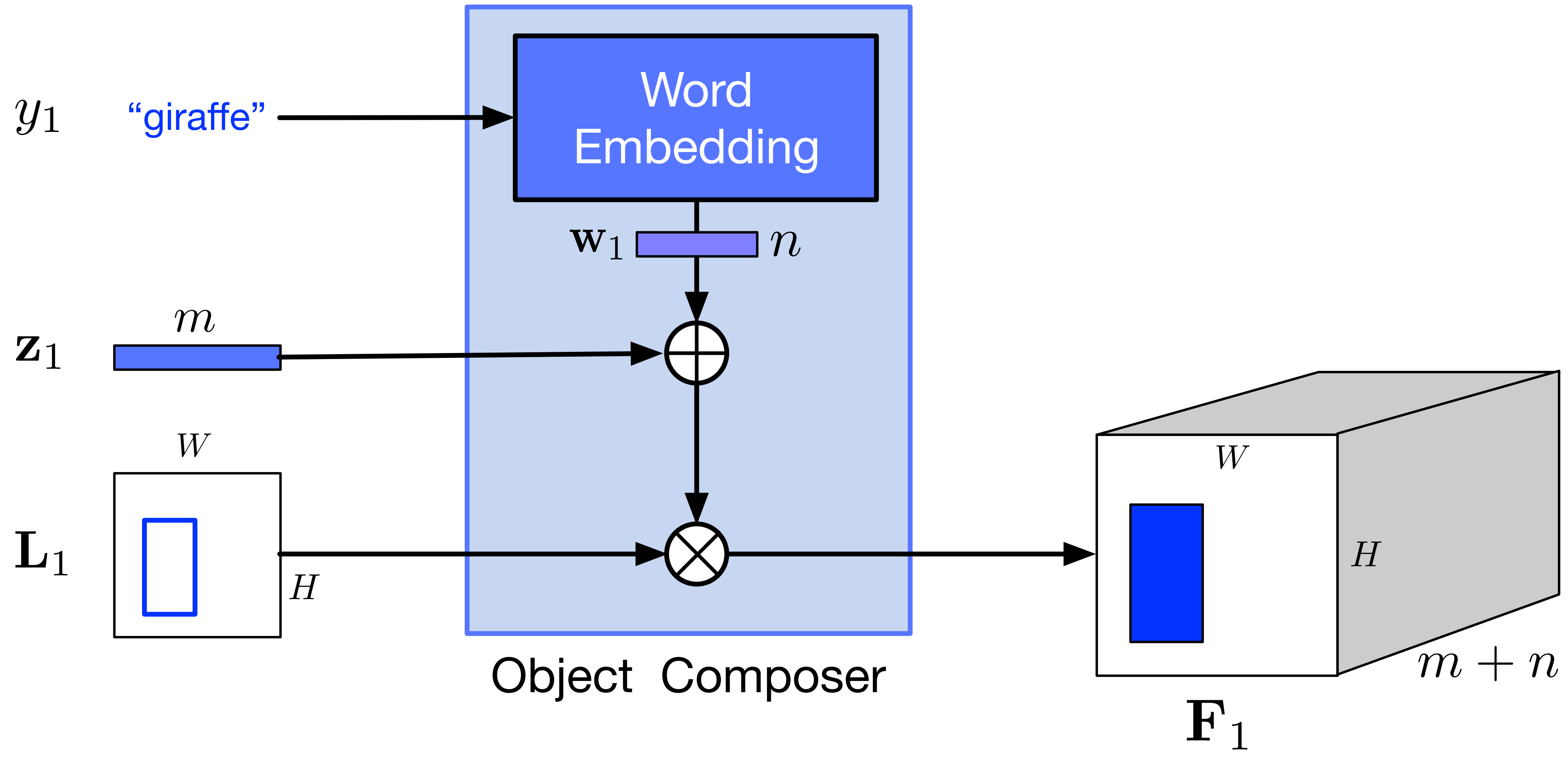


Model **Architecture**: Training

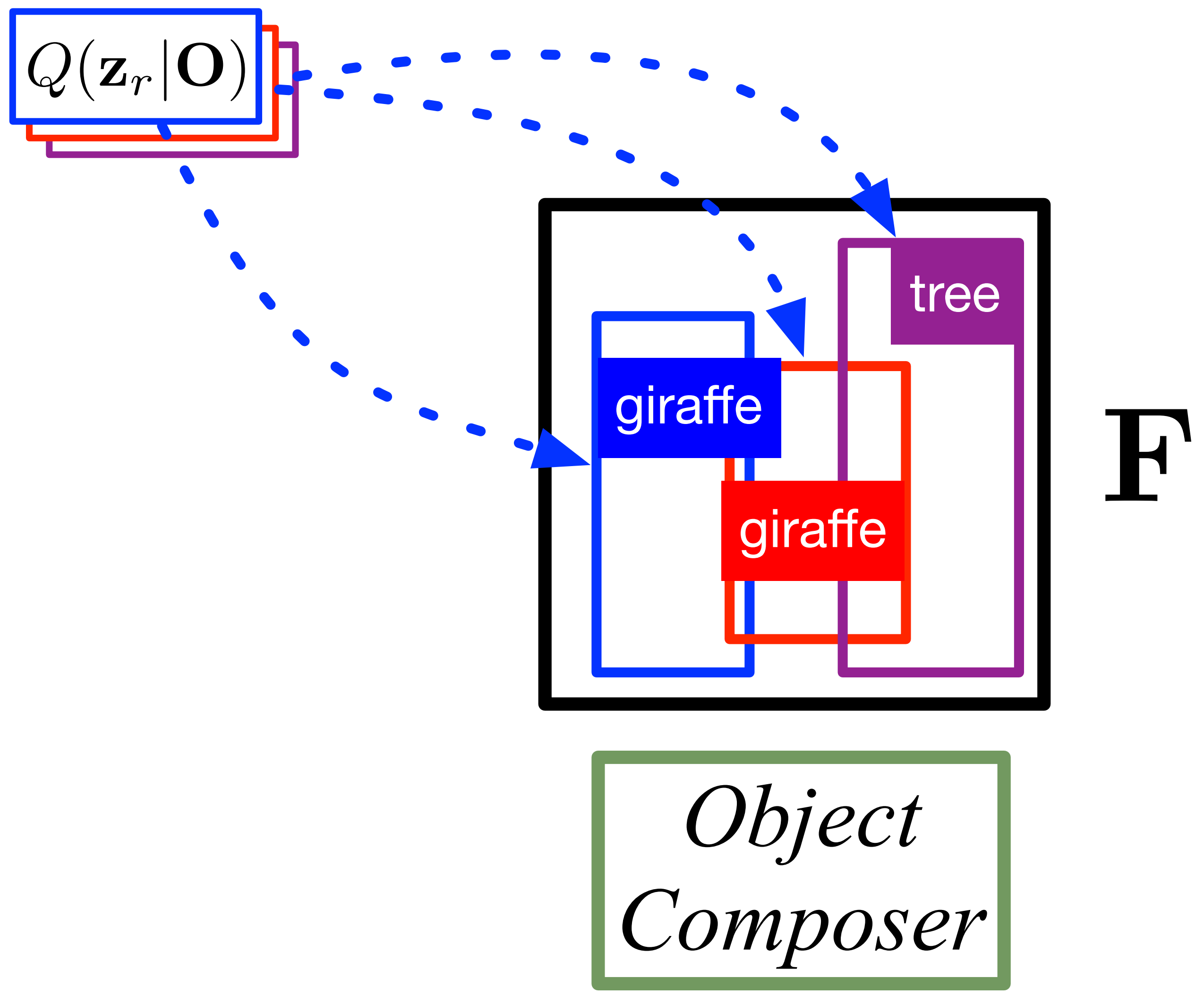


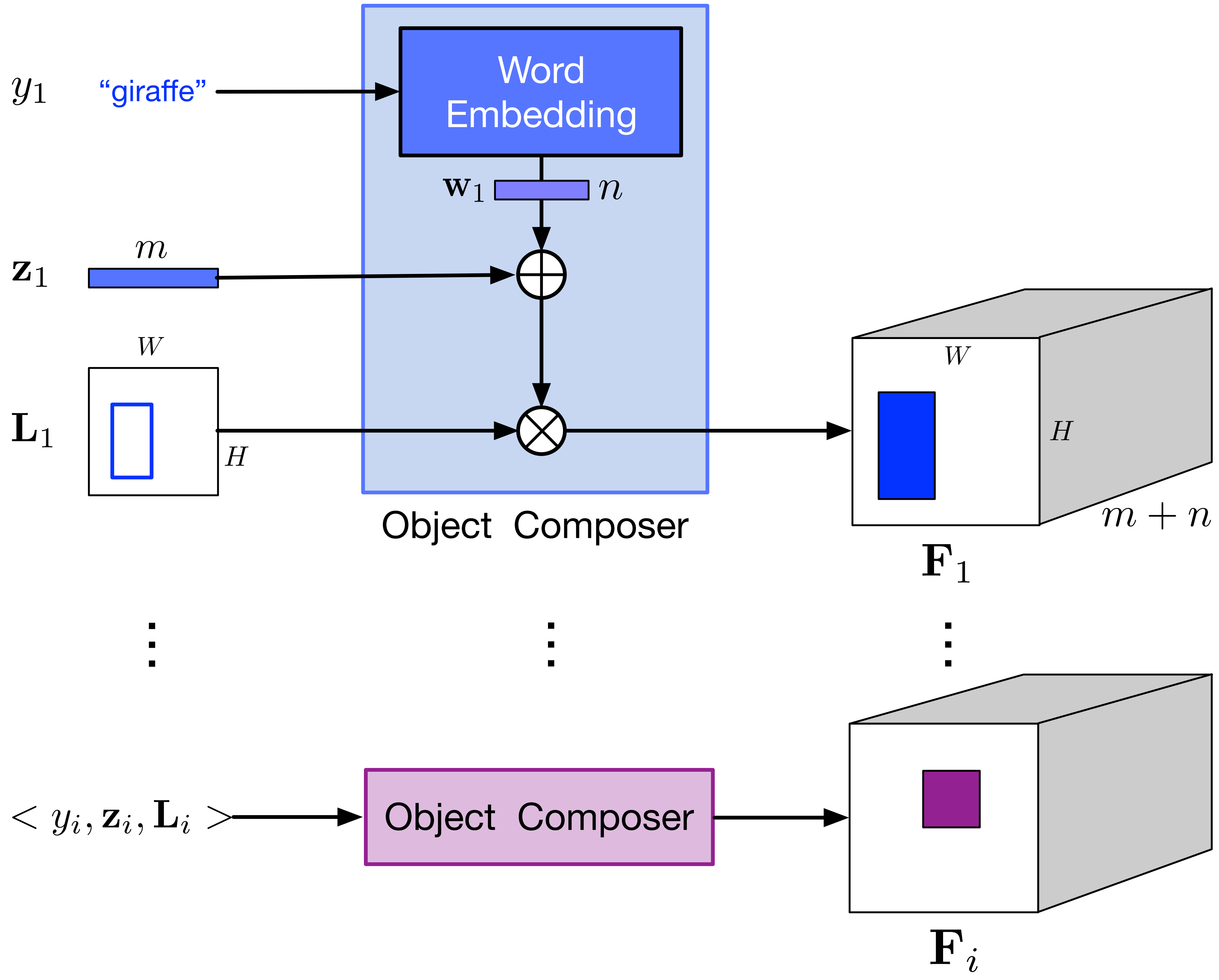
Model **Architecture**: Training



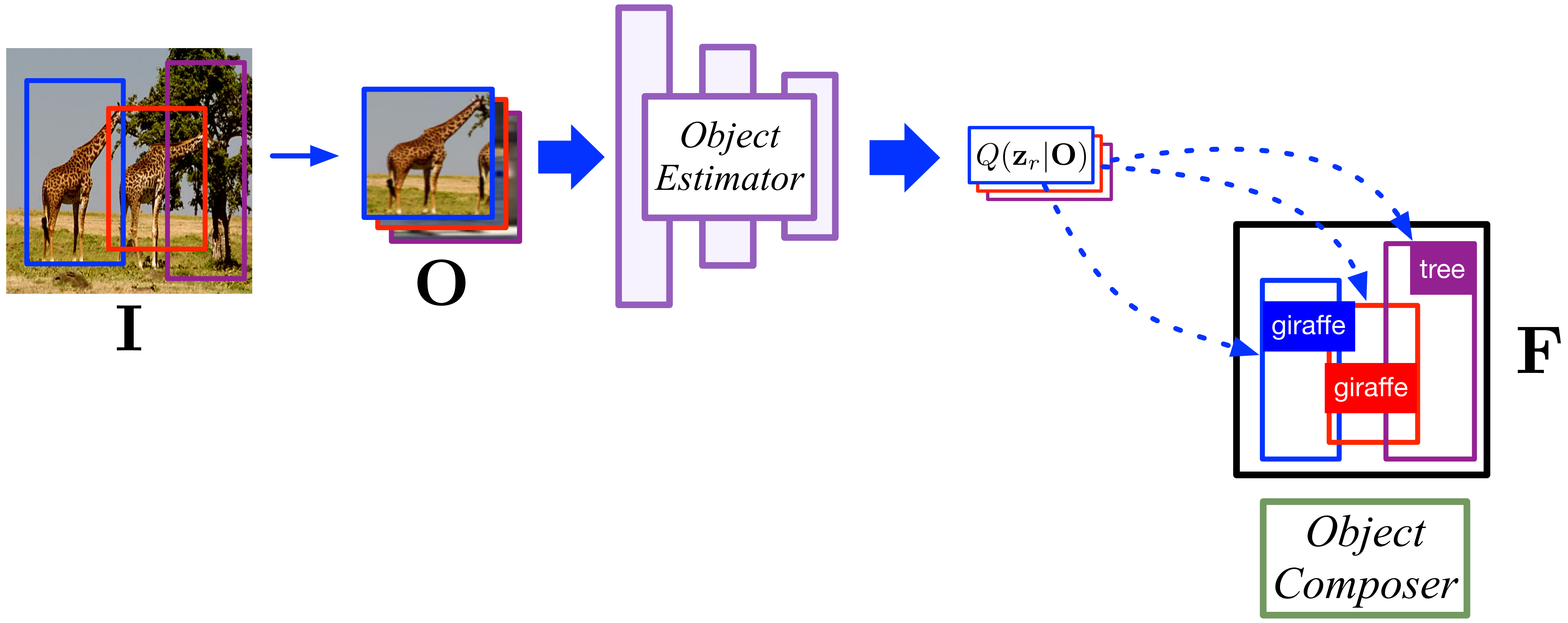


Model **Architecture**: Training

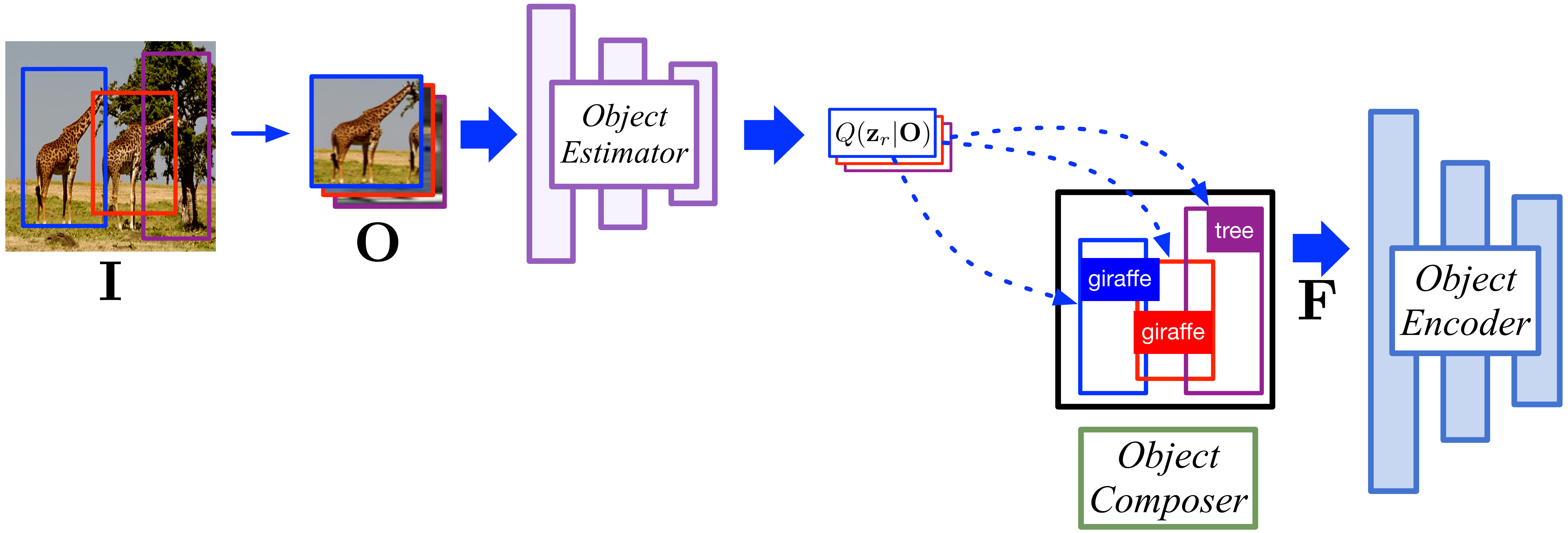




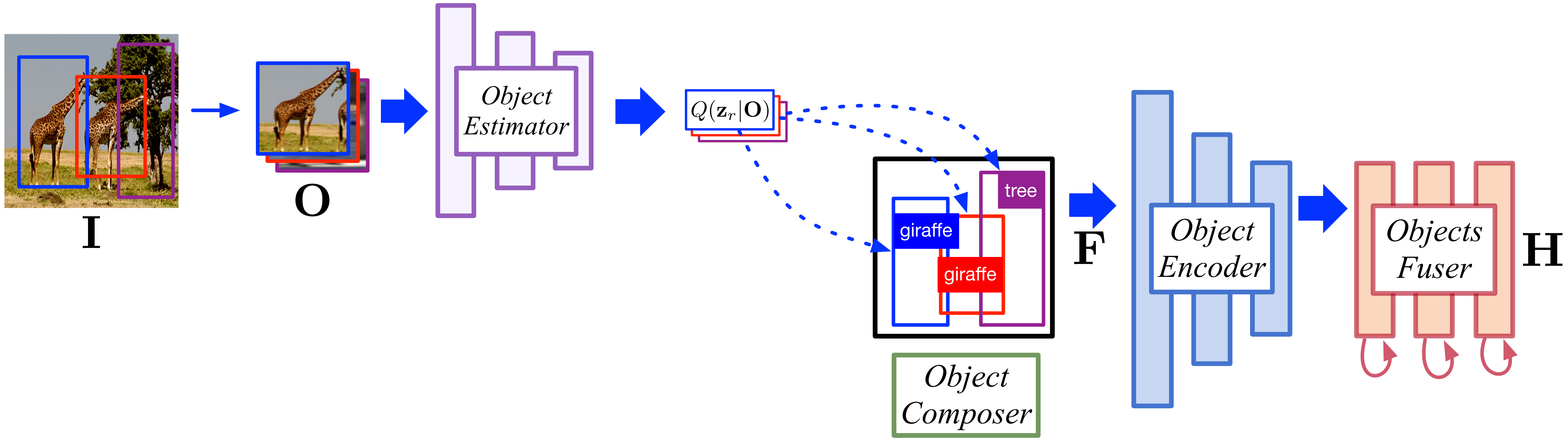
Model **Architecture**: Training



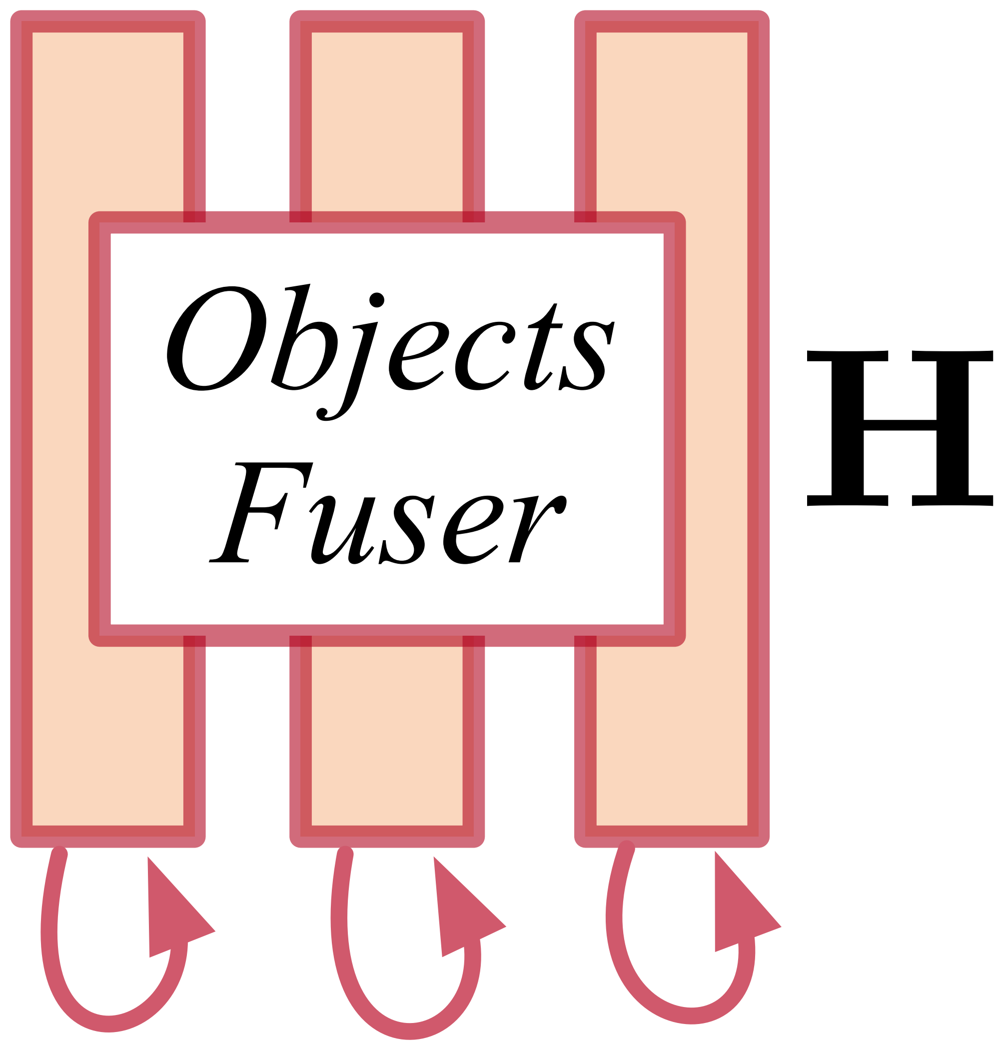
Model **Architecture**: Training

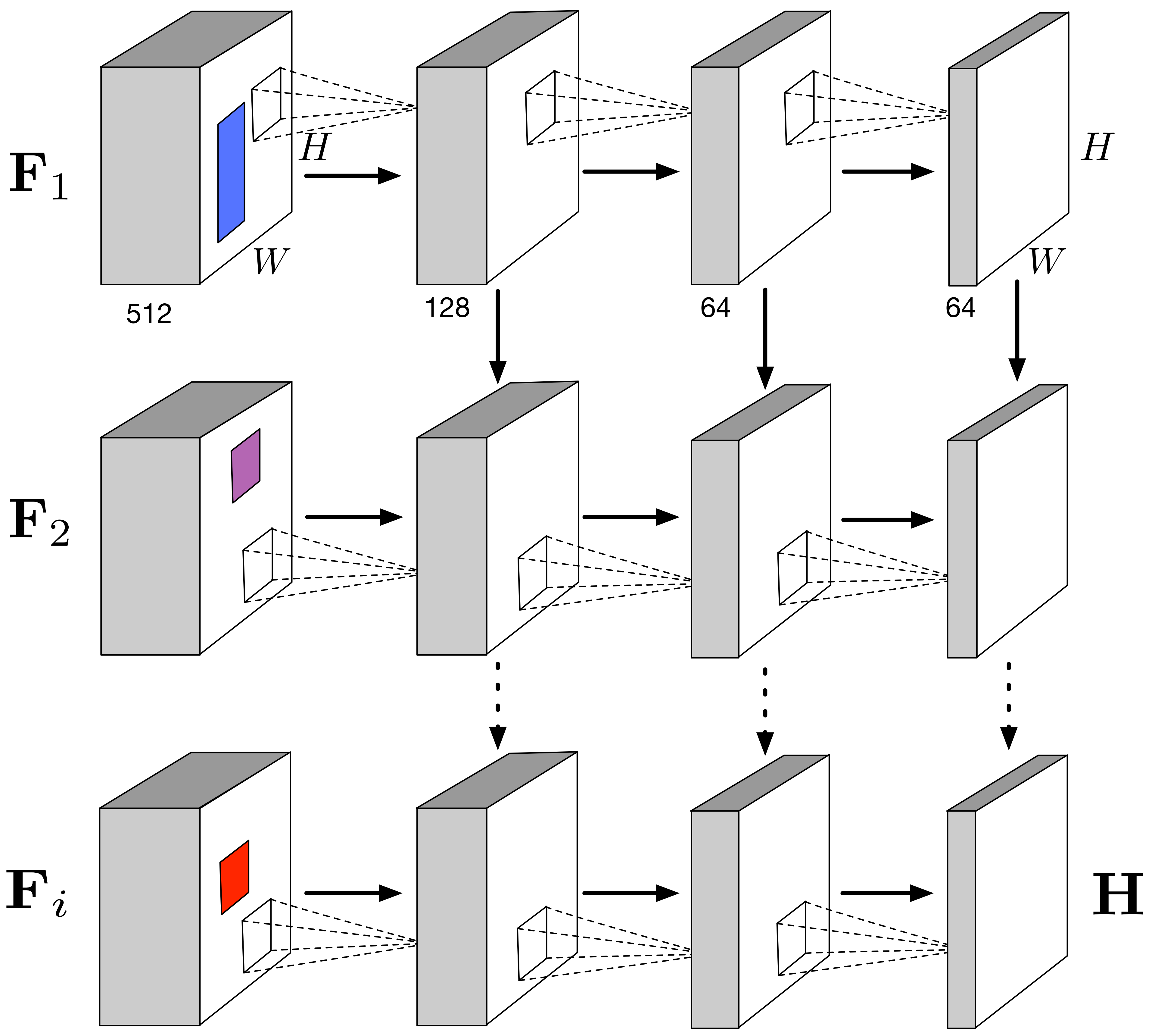


Model **Architecture**: Training

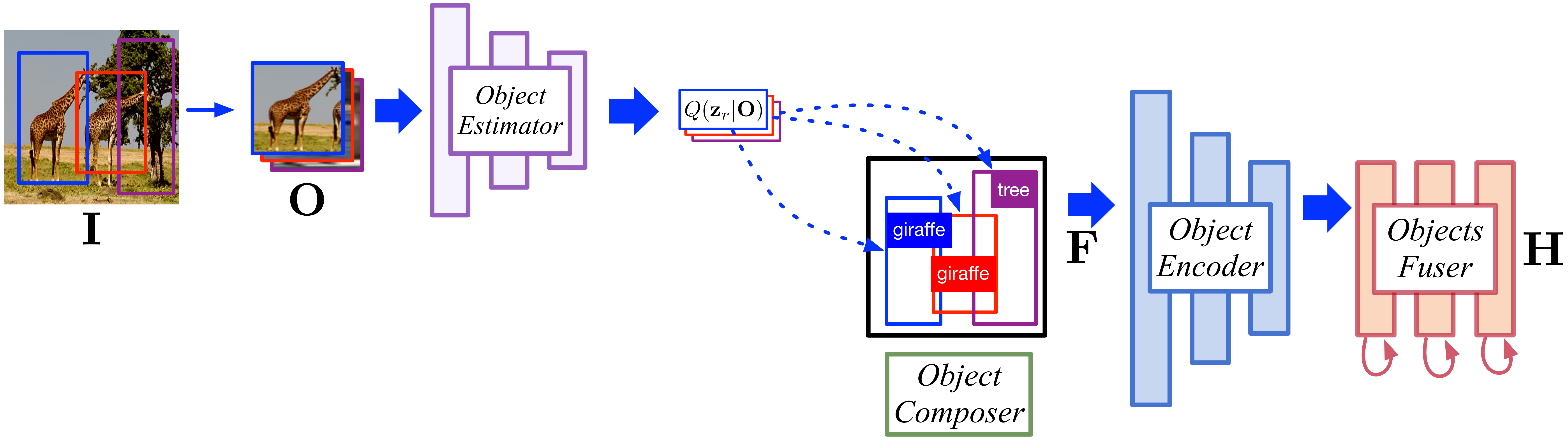


Model **Architecture**: Training

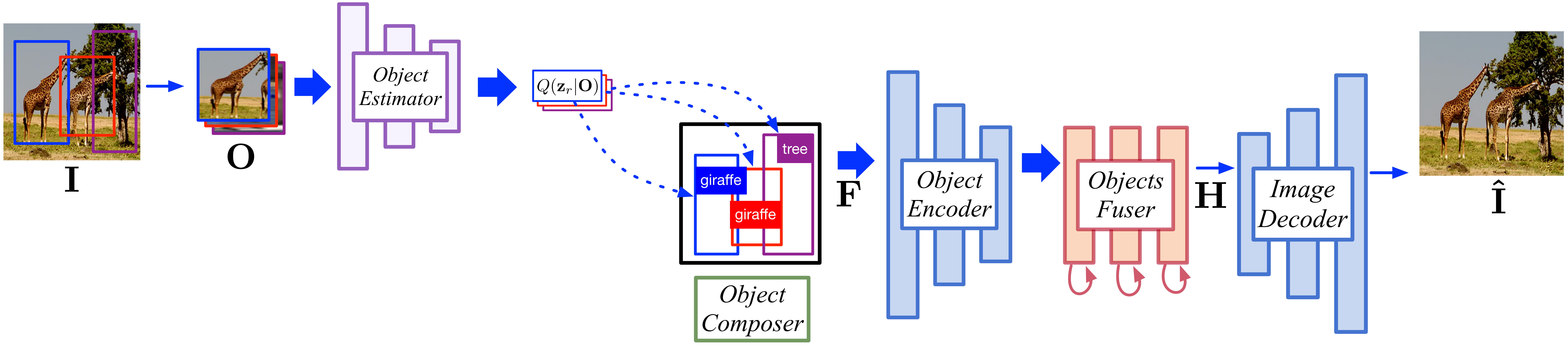




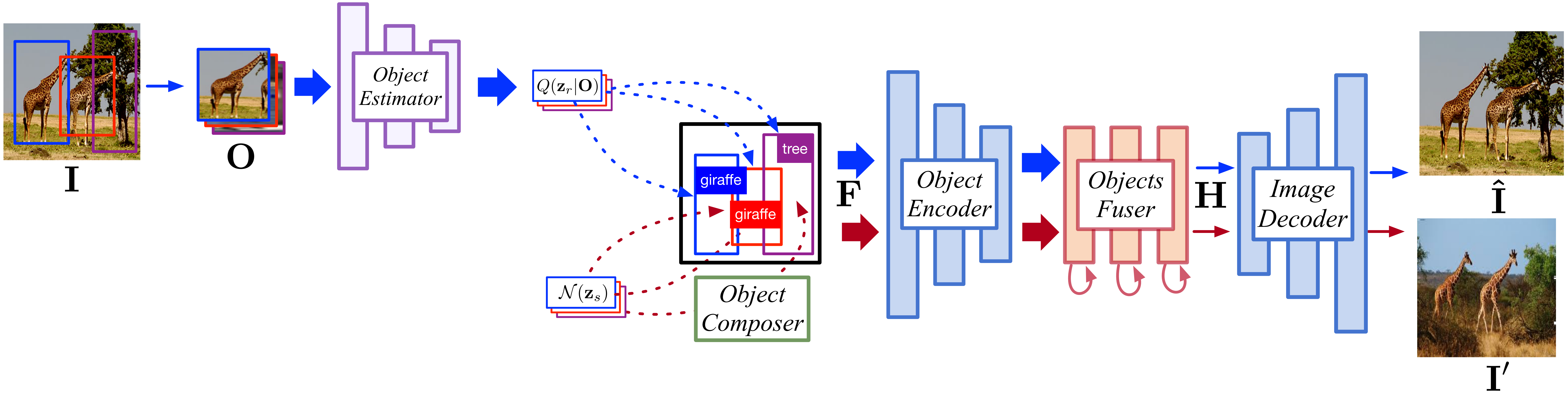
Model **Architecture**: Training



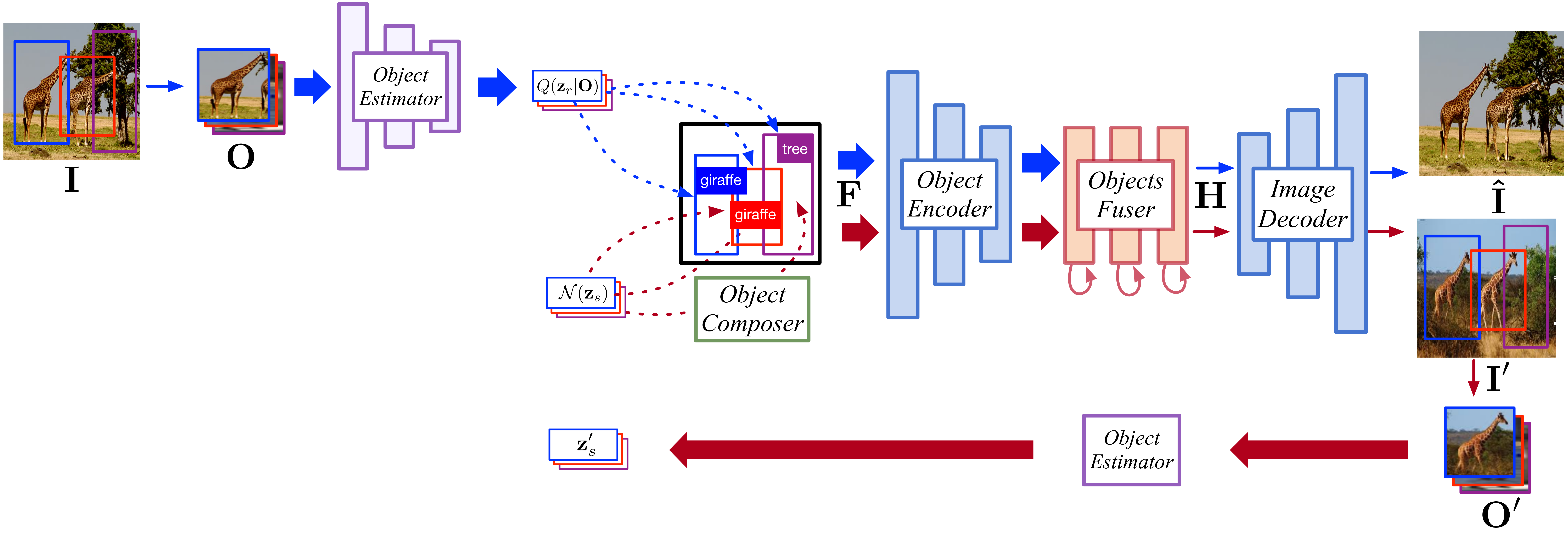
Model **Architecture**: Training



Model **Architecture**: Training

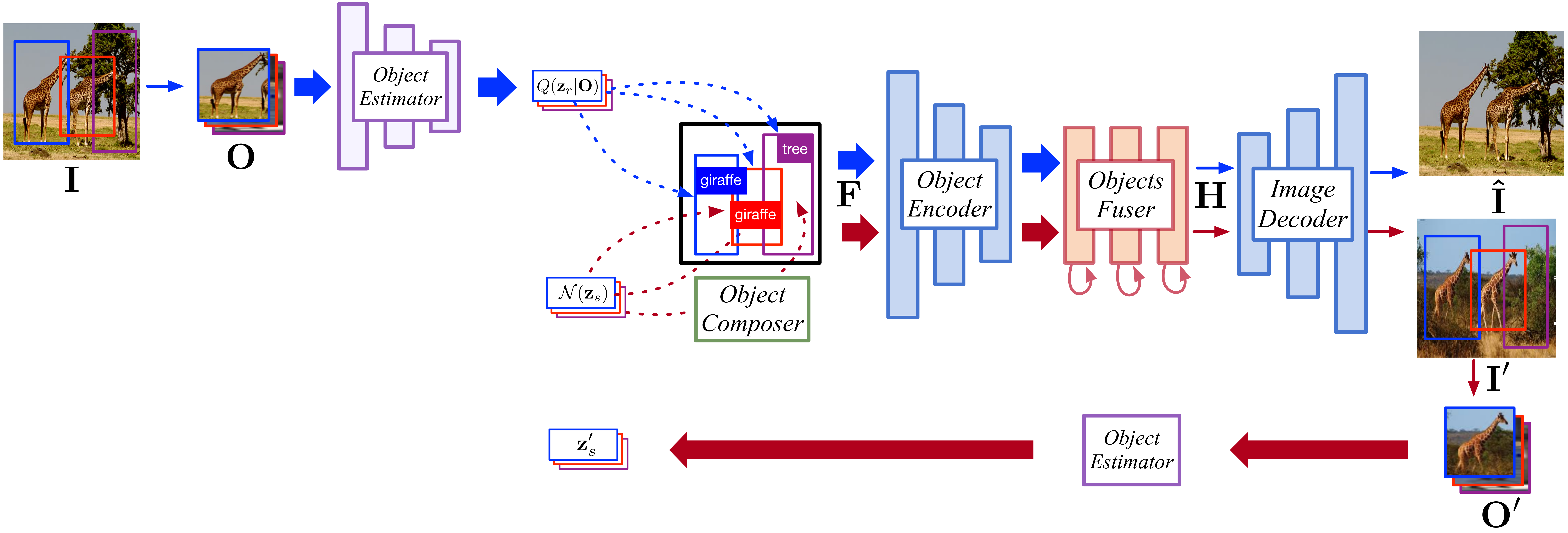


Model **Architecture**: Training



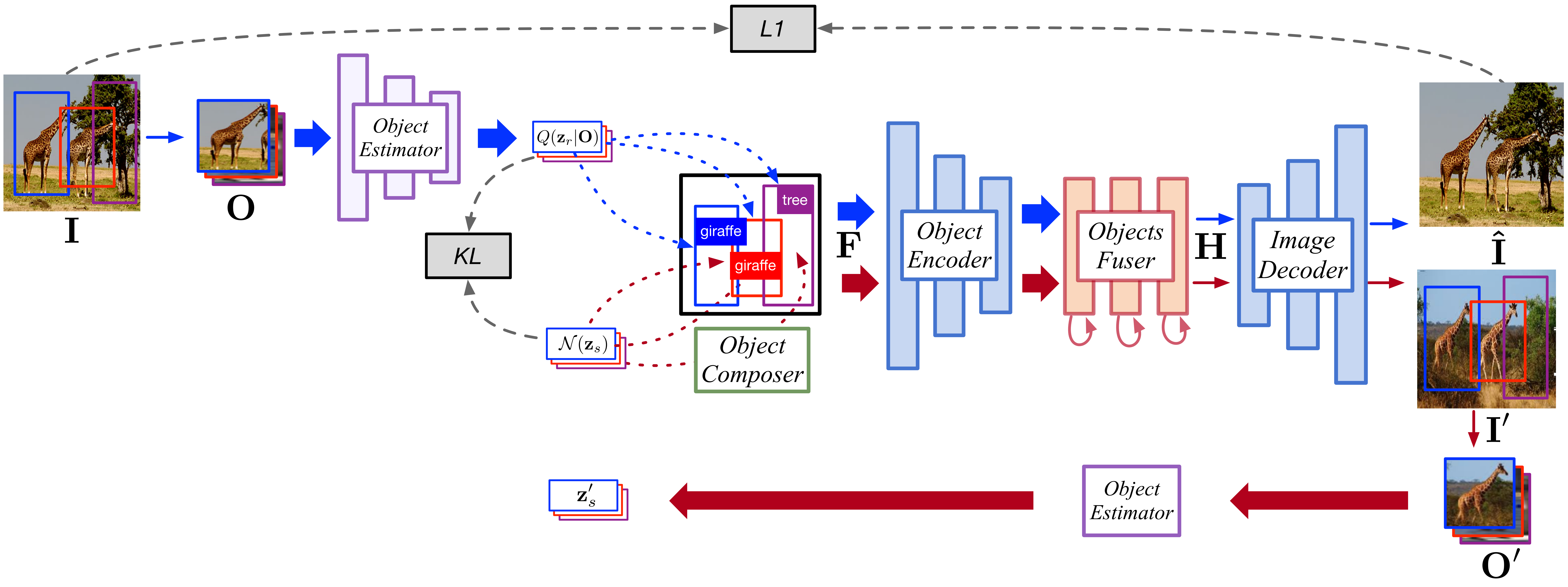
Model **Architecture**: Training

Losses



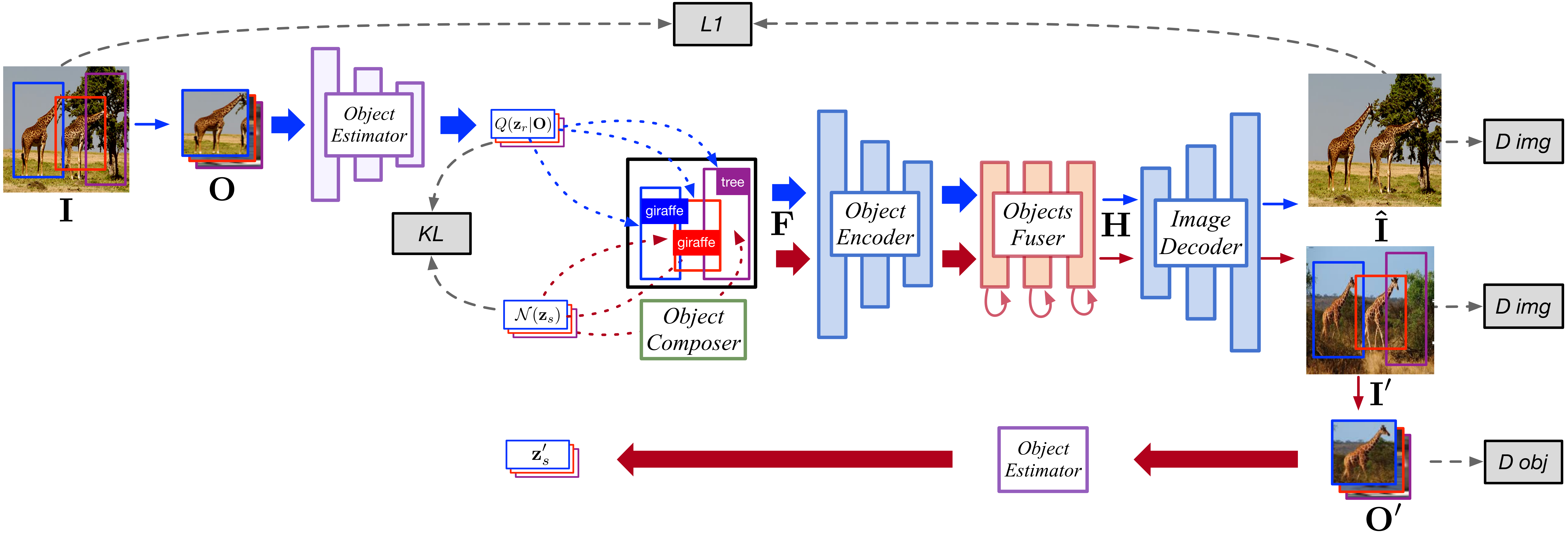
Model **Architecture**: Training

Losses



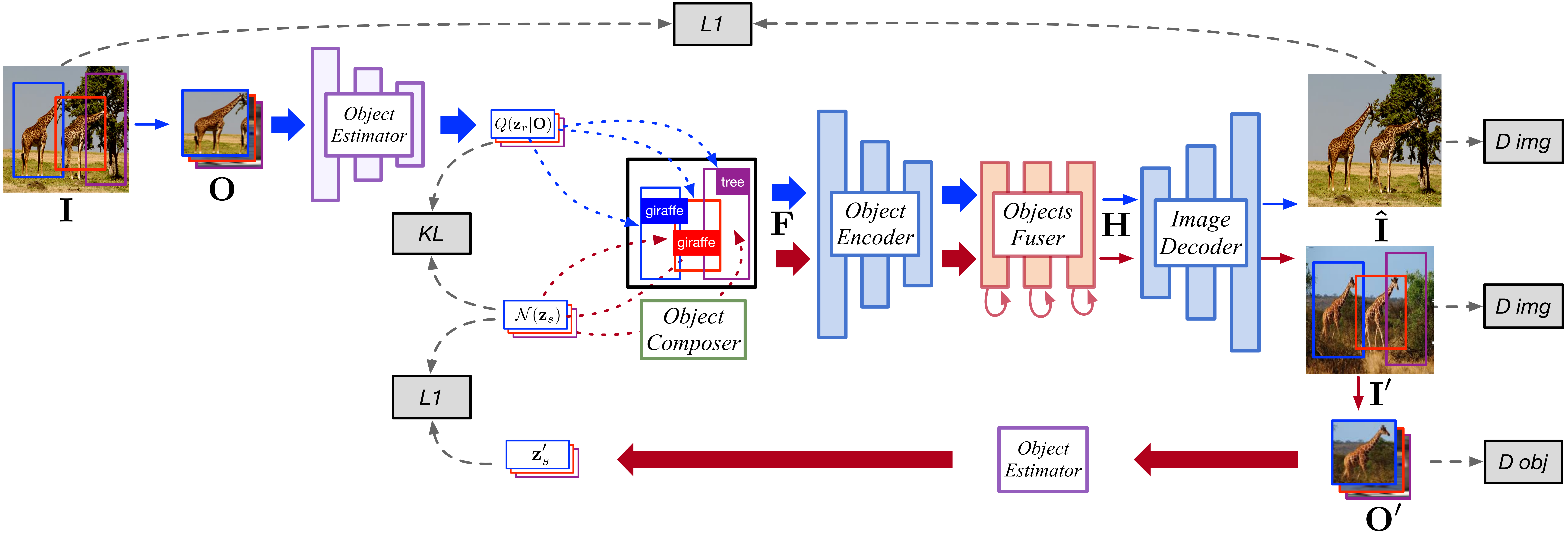
Model **Architecture**: Training

Losses



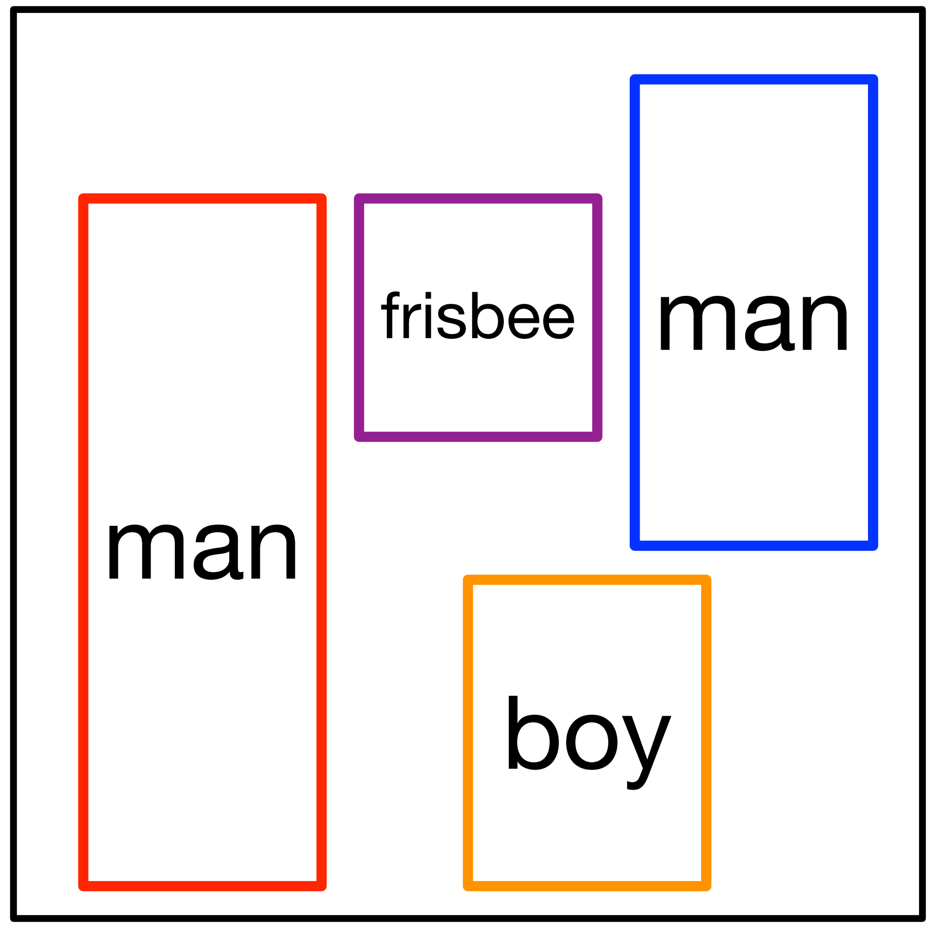
Model **Architecture**: Training

Losses

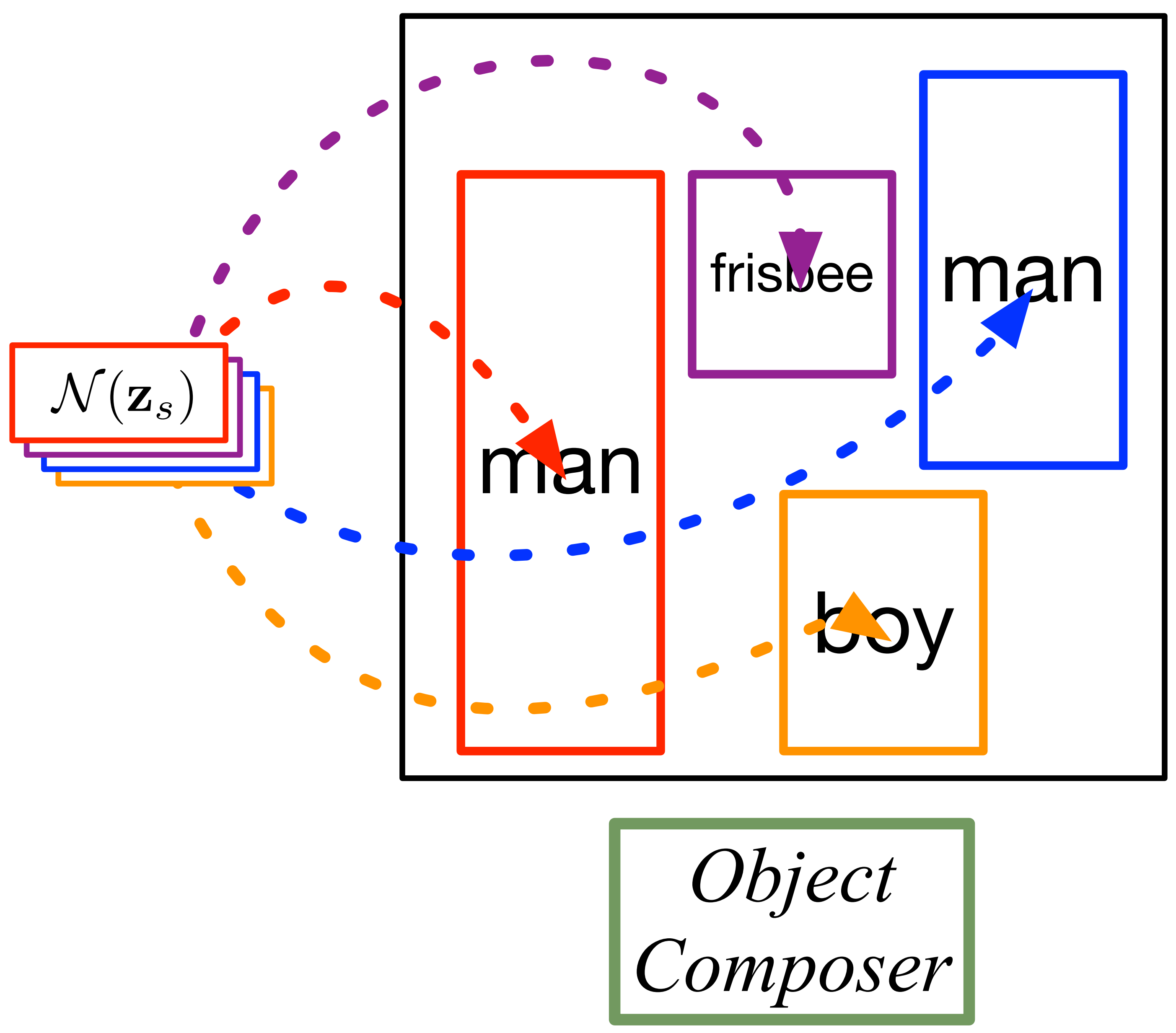


Model **Architecture**: Runtime

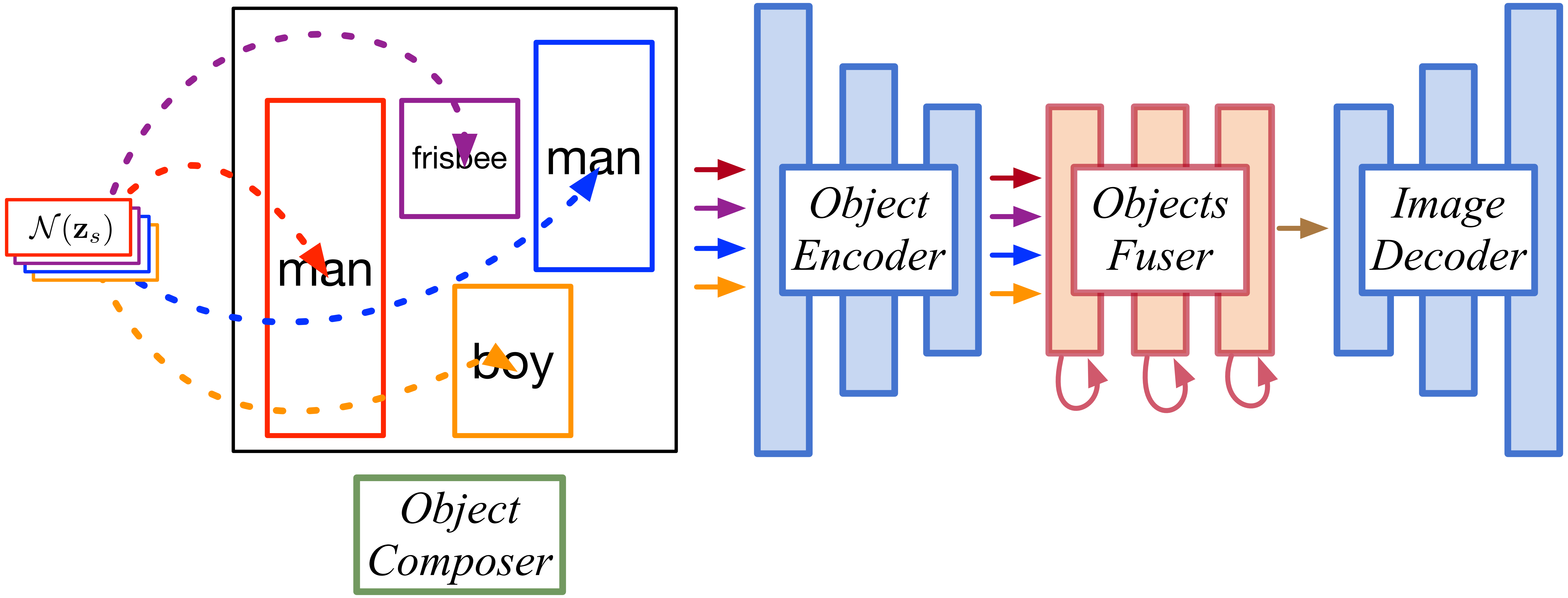
Model **Architecture**: Runtime



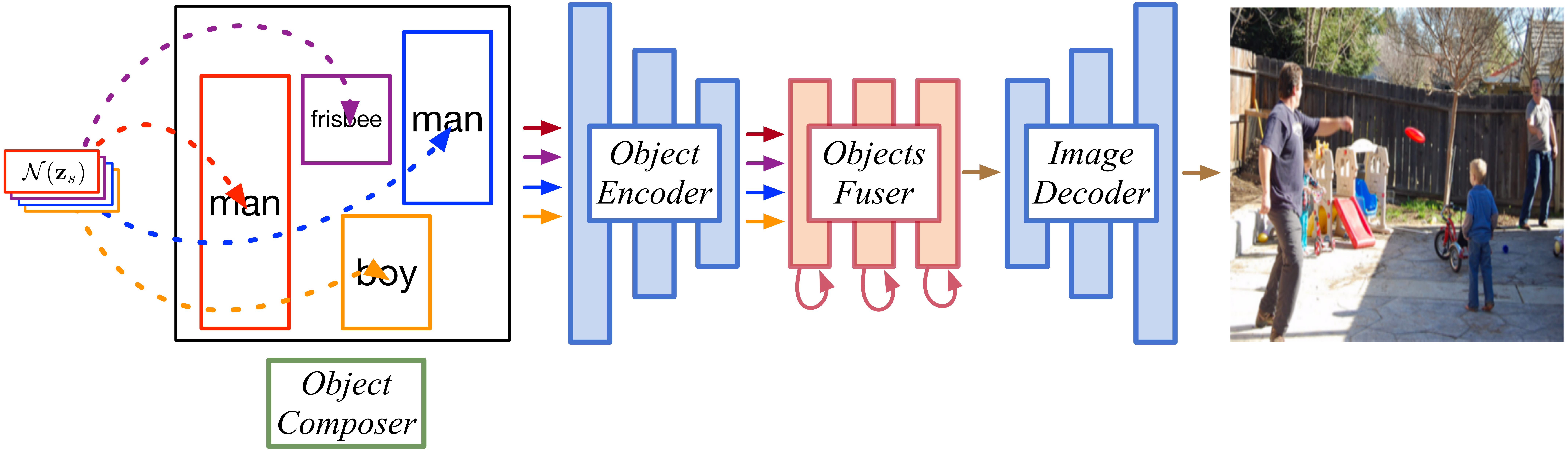
Model **Architecture**: Runtime



Model **Architecture**: Runtime



Model **Architecture**: Runtime



Experiments: Quantitative Results

Datasets:

Dataset	Train	Val.	Test	# Obj.	# Obj. in Image
COCO [1]	24,972	1,024	2,048	171	$3 \sim 8$
VG [18]	62,565	5,506	5,088	178	$3 \sim 30$

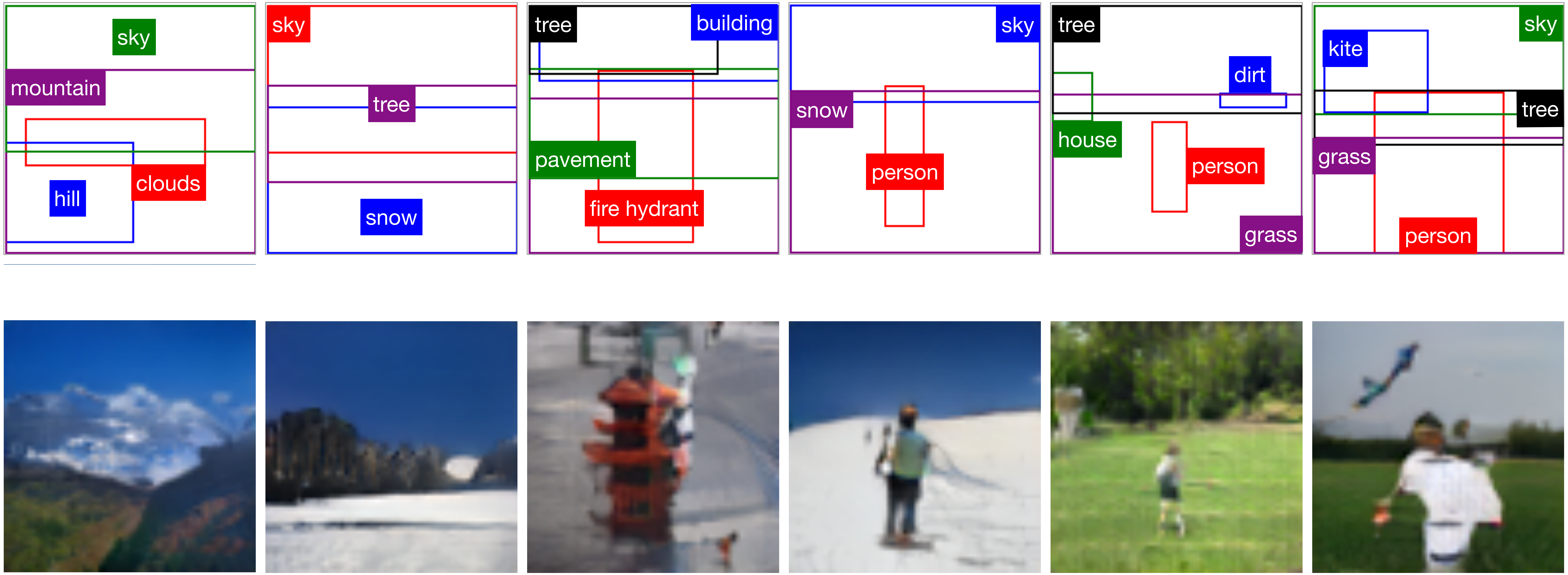
Evaluation:

Method	**Inception** Score		Object **Classification** Score		**Diversity** Score	
	COCO	VG	COCO	VG	COCO	VG
Real Images (64×64)	16.3 ± 0.4	13.9 ± 0.5	55.16	49.13	-	-
pix2pix [12]	3.5 ± 0.1	2.7 ± 0.02	12.06	9.20	0	0
sg2im (GT Layout) [13]	7.3 ± 0.1	6.3 ± 0.2	30.04	40.29	0.02 ± 0.01	0.15 ± 0.12
Ours	$\mathbf{9.1 \pm 0.1}$	$\mathbf{8.1 \pm 0.1}$	**50.84**	**48.09**	$\mathbf{0.15 \pm 0.06}$	$\mathbf{0.17 \pm 0.09}$

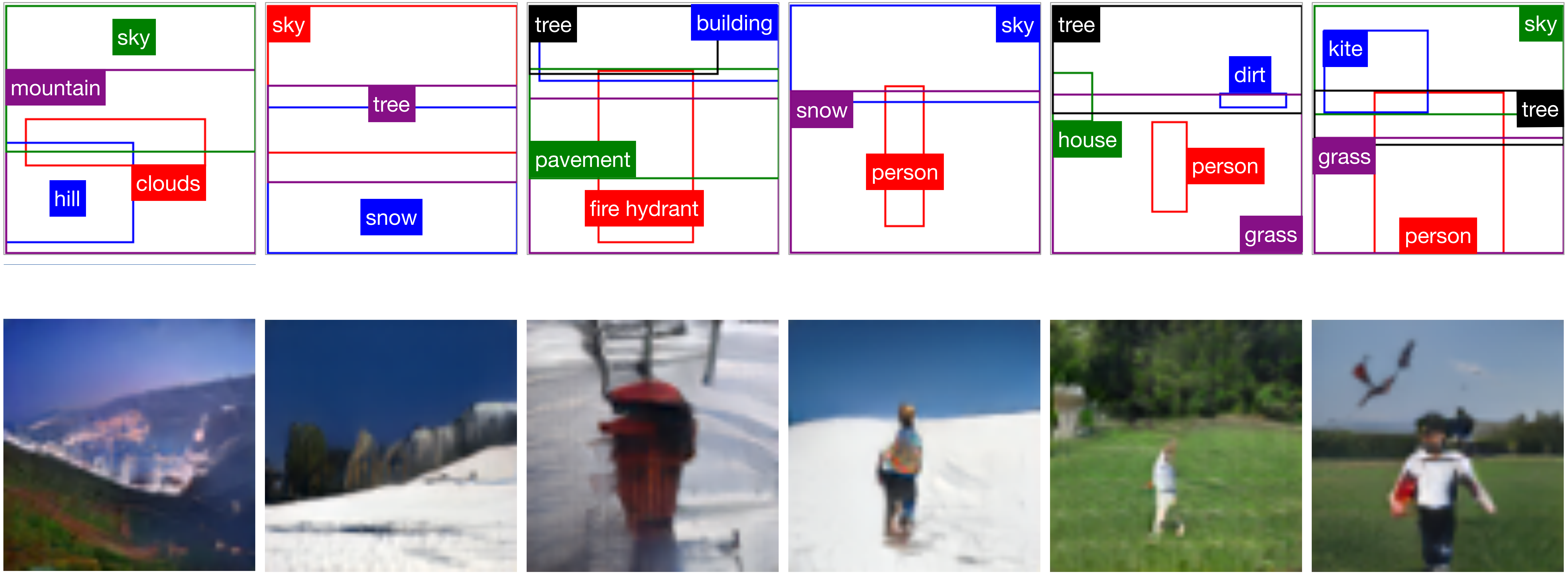
Results on COCO

Results on Visual Genome

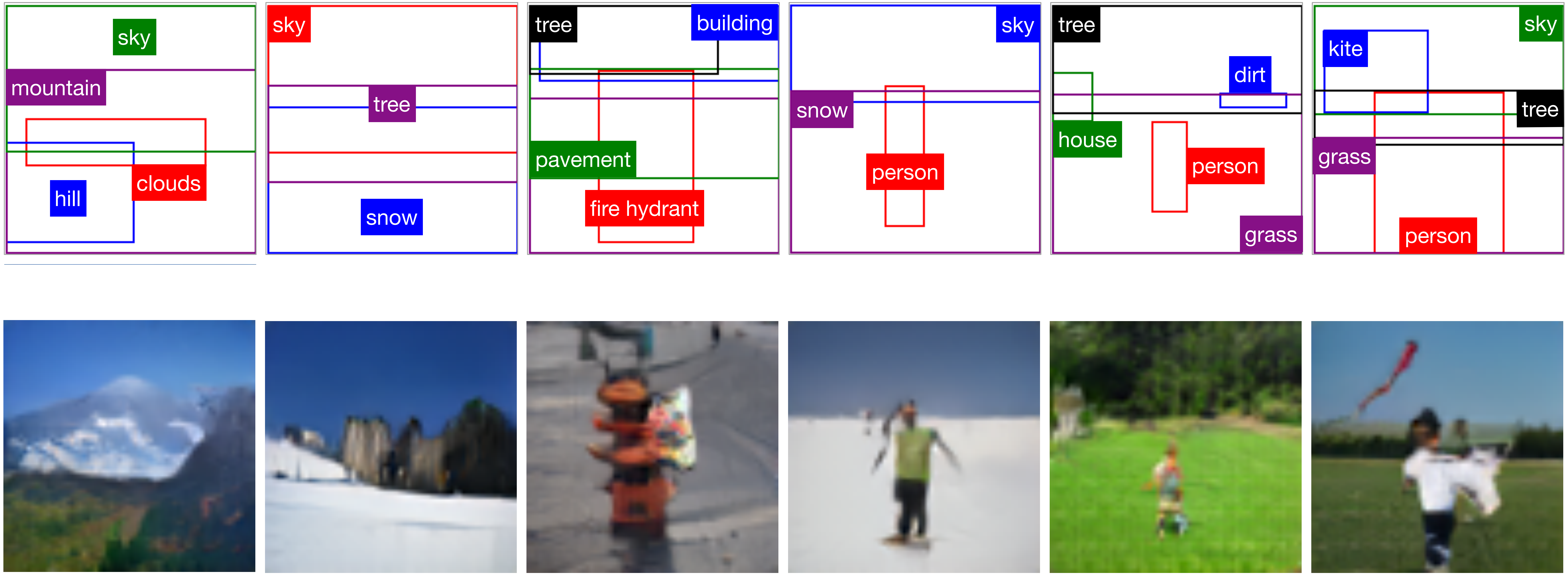
Results: Diversity



Results: Diversity



Results: Diversity



Layout to Image

Drag to draw bounding boxes and assign labels or simply load a pre-defined layout.

PERSON PERSONS INDOOR BEACH FOOD BOAT WINDOW CAR COW MONITOR

Labels

Layout

Images

GENERATE START OVER

Image Generation from Layout, Bo Zhao, Lili Meng, Weidong Yin and Leonid Sigal, CVPR 2019.

Web Application Developed by Mark (Ke) Ma

Layout to Image

Drag to draw bounding boxes and assign labels or simply load a pre-defined layout.

PERSON | PERSONS | INDOOR | BEACH | FOOD | BOAT | WINDOW | CAR | COW | MONITOR

Labels

Layout

Images

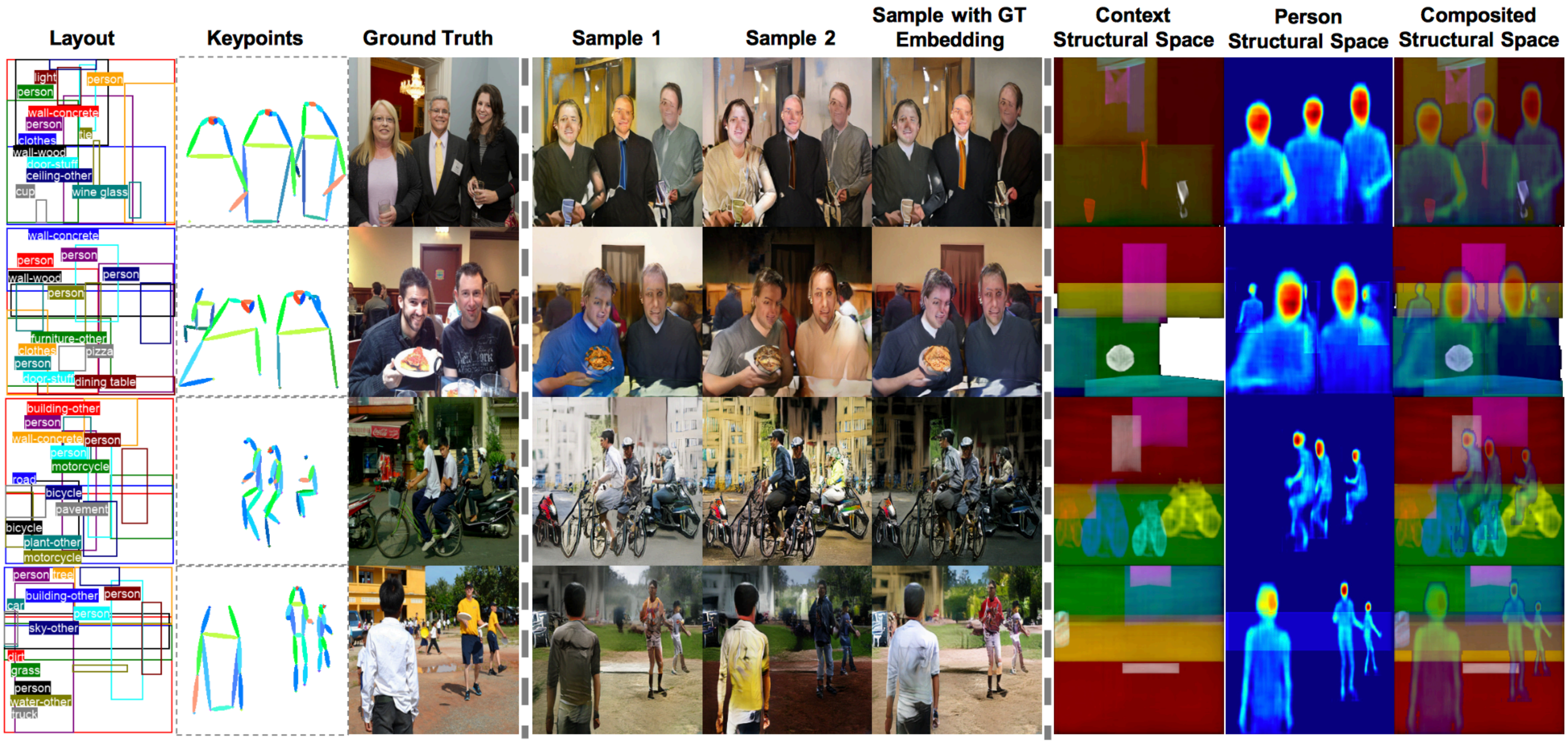
GENERATE | START OVER

Image Generation from Layout, Bo Zhao, Lili Meng, Weidong Yin and Leonid Sigal, CVPR 2019.

Web Application Developed by Mark (Ke) Ma

Extensions

[Yin et al., WACV 2021]



GANs

Don't work with an explicit density function

Take game-theoretic approach: learn to generate from training distribution through 2-player game

Pros:

— Beautiful, state-of-the-art samples!

Cons:

— Trickier / more unstable to train

— Can't solve inference queries such as p(x), p(z|x)

Active area of research:

— Better loss functions, more stable training (Wasserstein GAN, LSGAN, many others)

— Conditional GANs, GANs for all kinds of applications

Non-Convergence

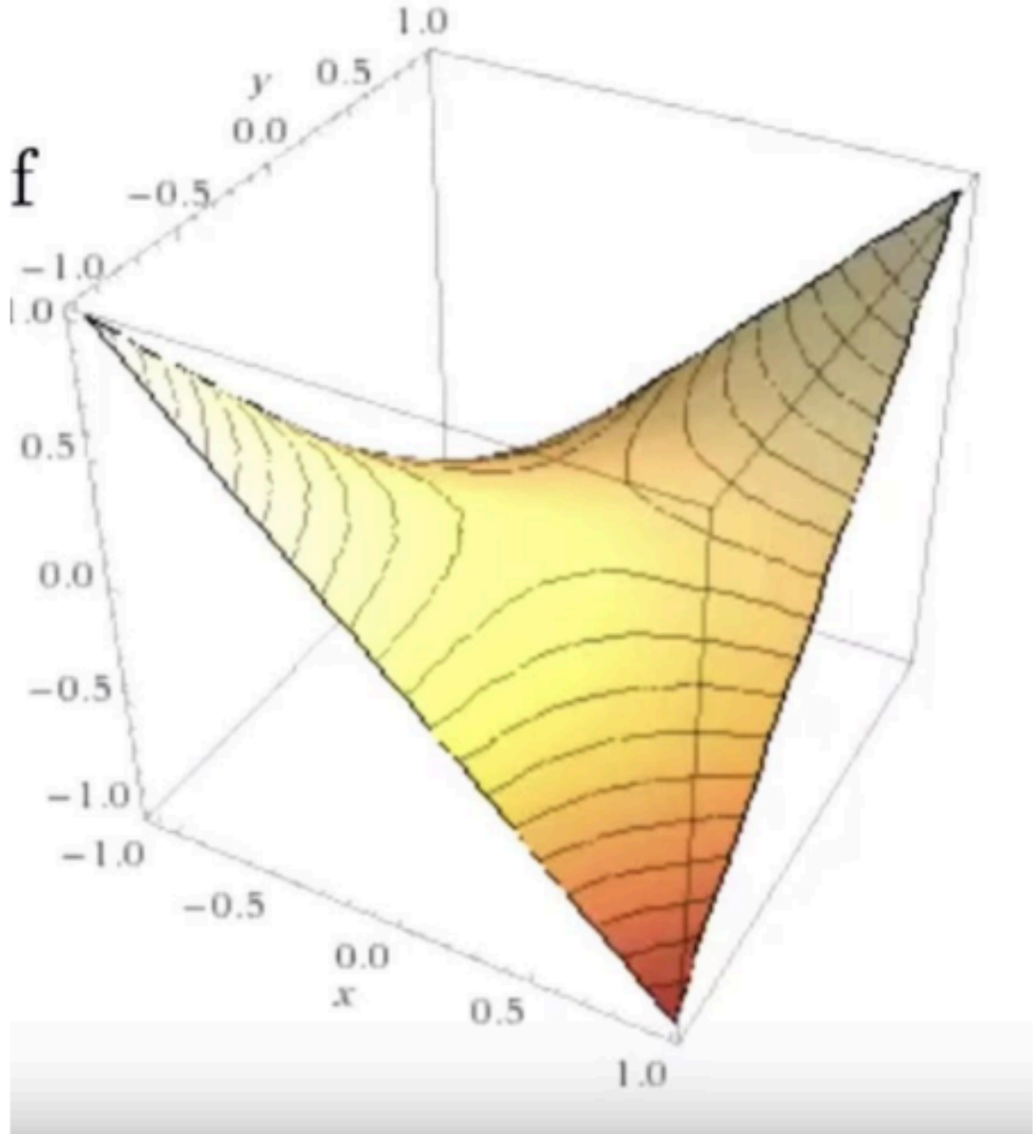
- Differential Equation's solution has sinusoidal terms
- Even with a small learning rate, it will not converge
- Discrete time gradient descent can spiral outward for large step size

D & G nullifies each others learning in every iteration

Train for a long time – without generating good quality samples

$$V(x, y) = xy$$

$$x = 0, \qquad y = 0$$



$$V(x(t), y(t)) = x(t)y(t)$$

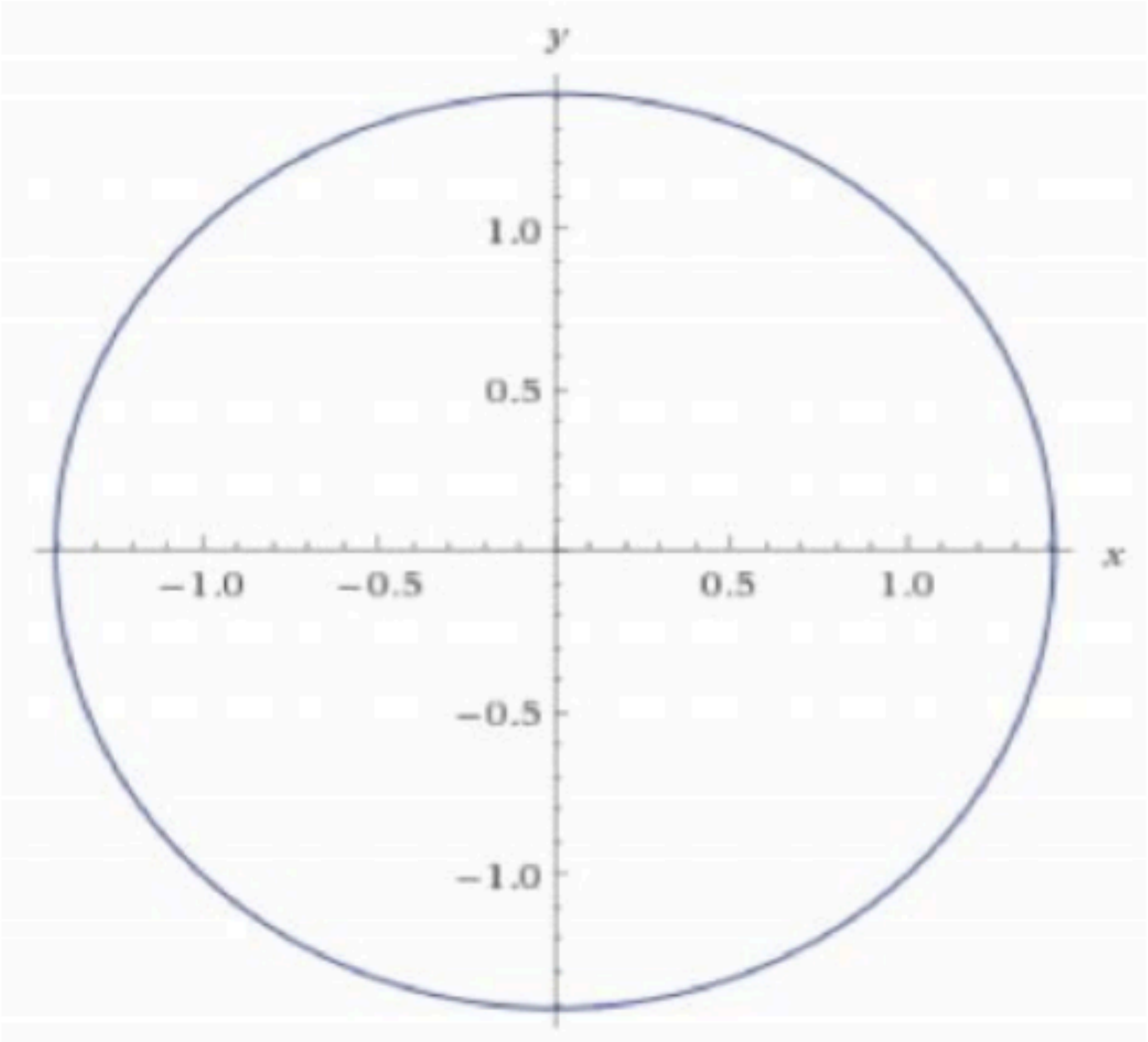
$$\frac{\partial x}{\partial t} = -y(t)$$

$$\frac{\partial y}{\partial t} = x(t)$$

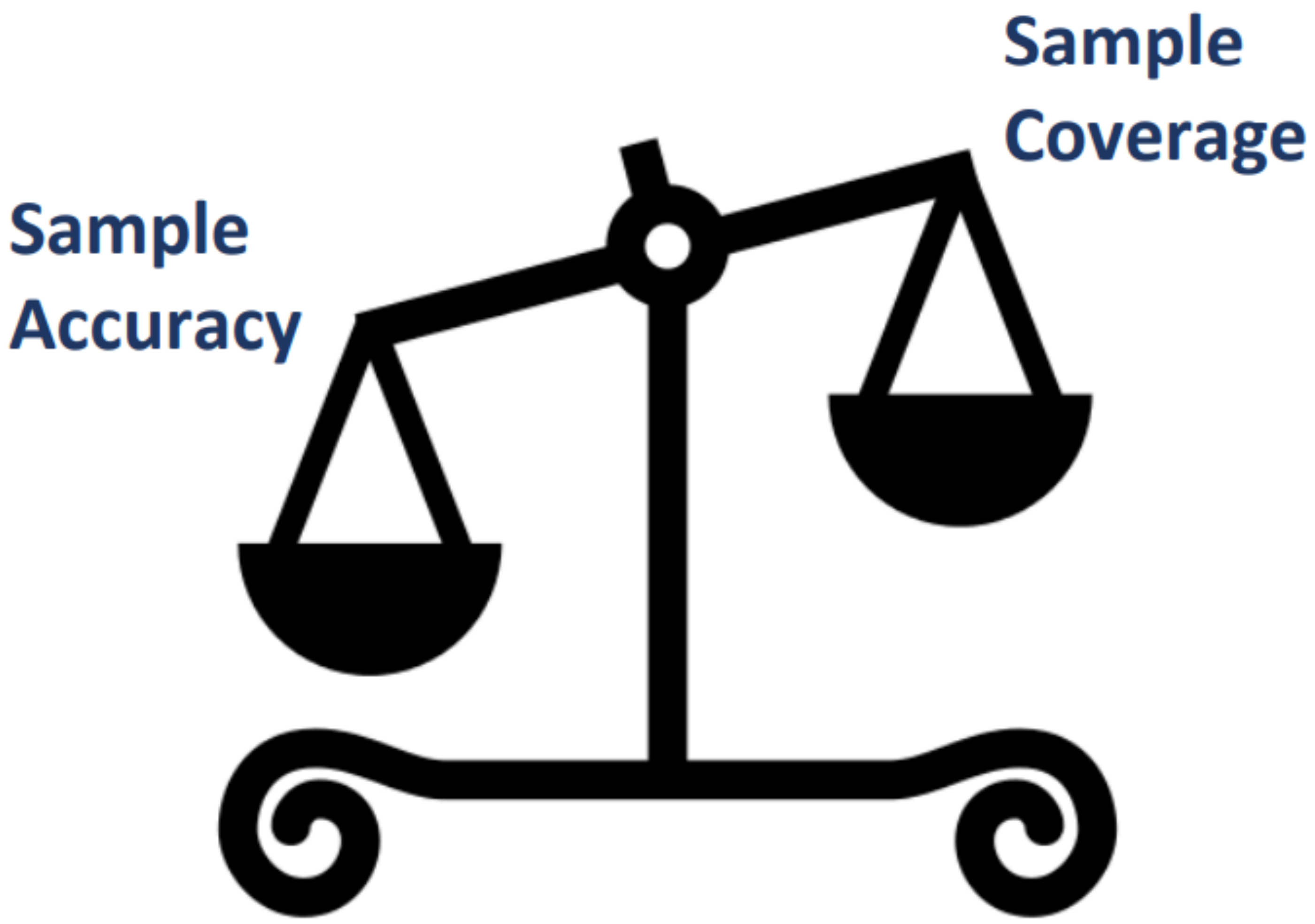
$$\frac{\partial^2 y}{\partial t^2} = \frac{\partial x}{\partial t}. = -y(t)$$

$$x(t) = x(0)cost(t) - y(0)\sin(t)$$

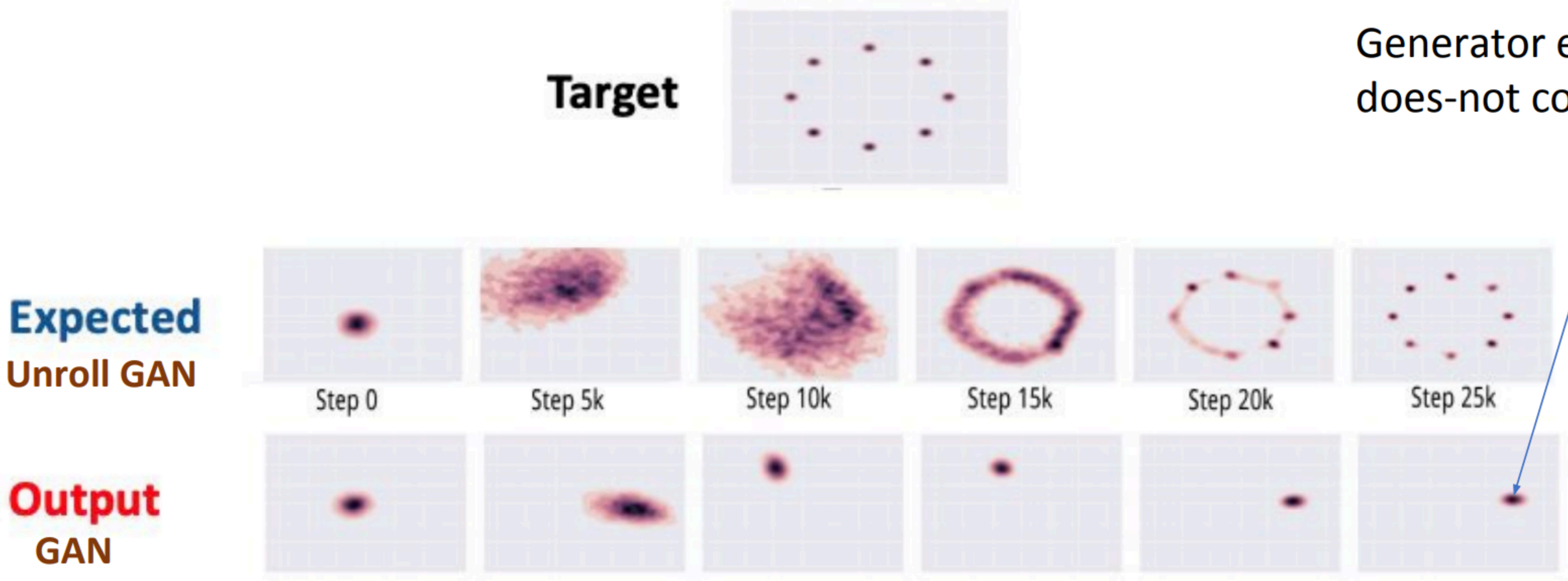
$$y(t) = x(0)cost(t) - y(0)\sin(t)$$



Mode Collapse



Generator excels in a subspace but does-not cover entire real distribution



Luke et al. 2016

Why **GANs** are hard to train?

- — Generator keeps generating similar images — so nothing to learn
- — Maintain trade-off of generating more **accurate** vs. high **coverage** samples
- — Two learning tasks need to have balance to achieve stability
  - — If the **discriminator** is not sufficiently trained — it can worsen generator
  - — If the **discriminator** is too good — will produce no gradients

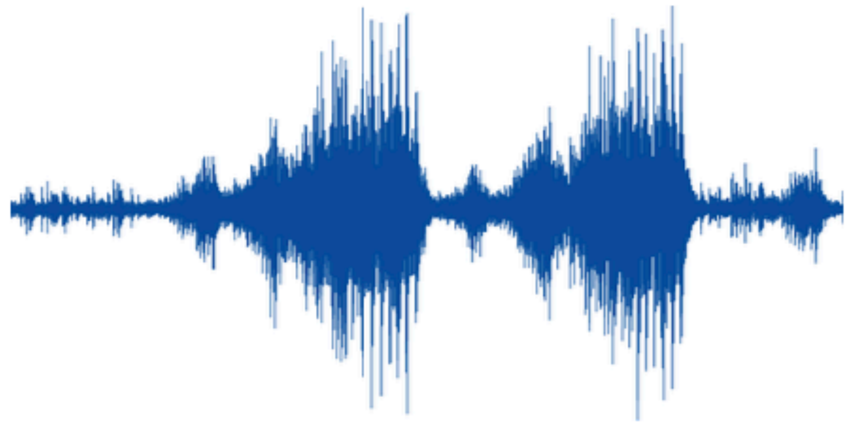
Topics in AI (CPSC 532S):
Multimodal Learning with Vision, Language and Sound

Lecture 19: Graph Neural Networks

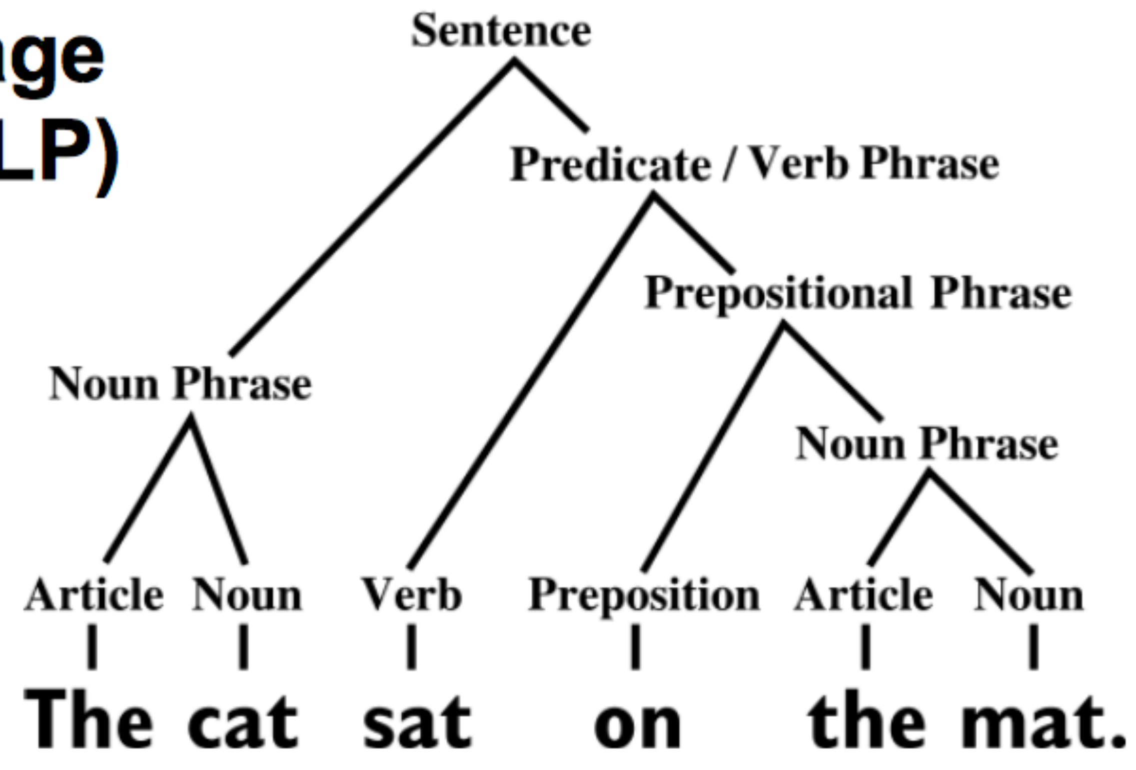
Traditional **Neural Networks**

IMAGENET

Speech data

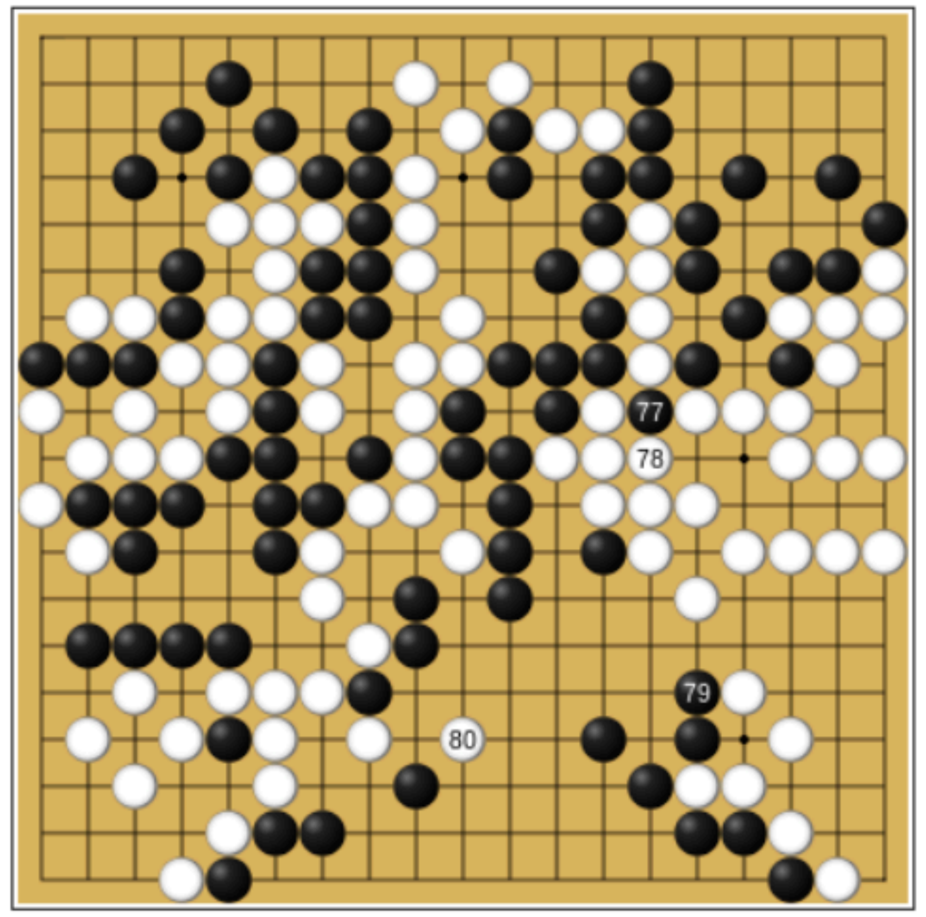


Natural language processing (NLP)



...

Grid games



Deep neural nets that exploit:

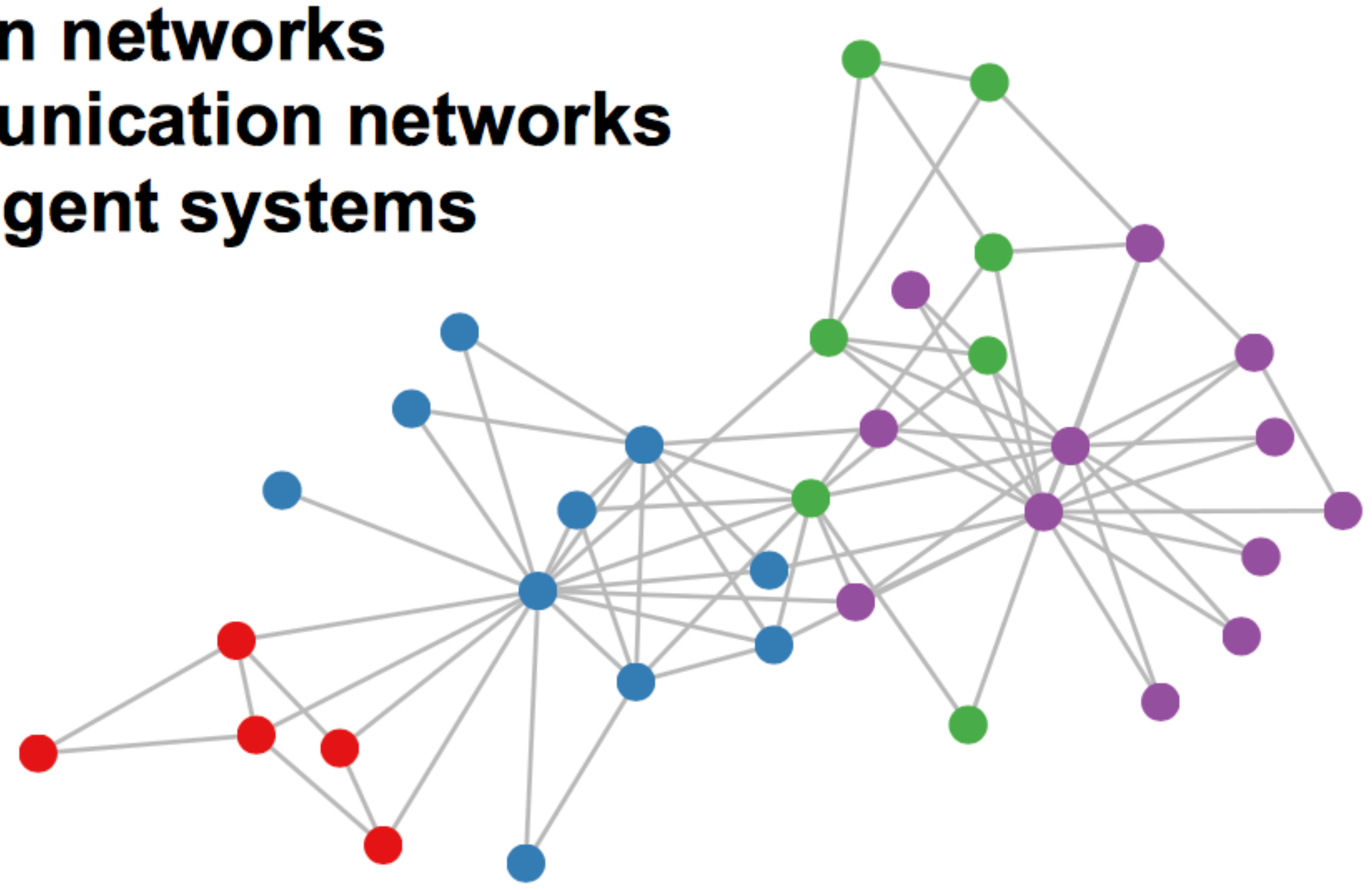
- translation equivariance (weight sharing)
- hierarchical compositionality

* slide from Thomas Kipf, **University of Amsterdam**

Graph-structured Data

A lot of real-world data does not “live” on grids

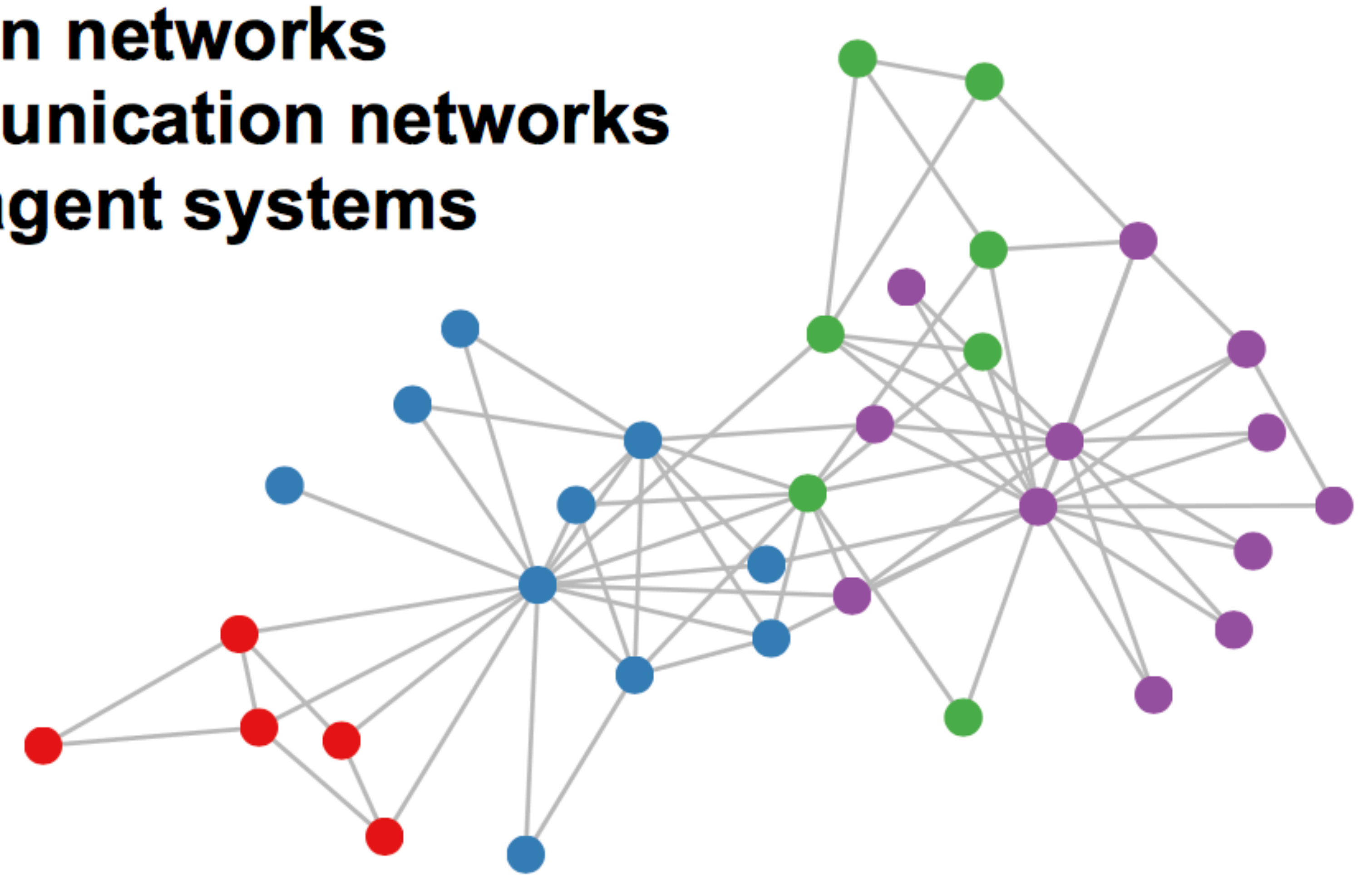
Social networks
Citation networks
Communication networks
Multi-agent systems



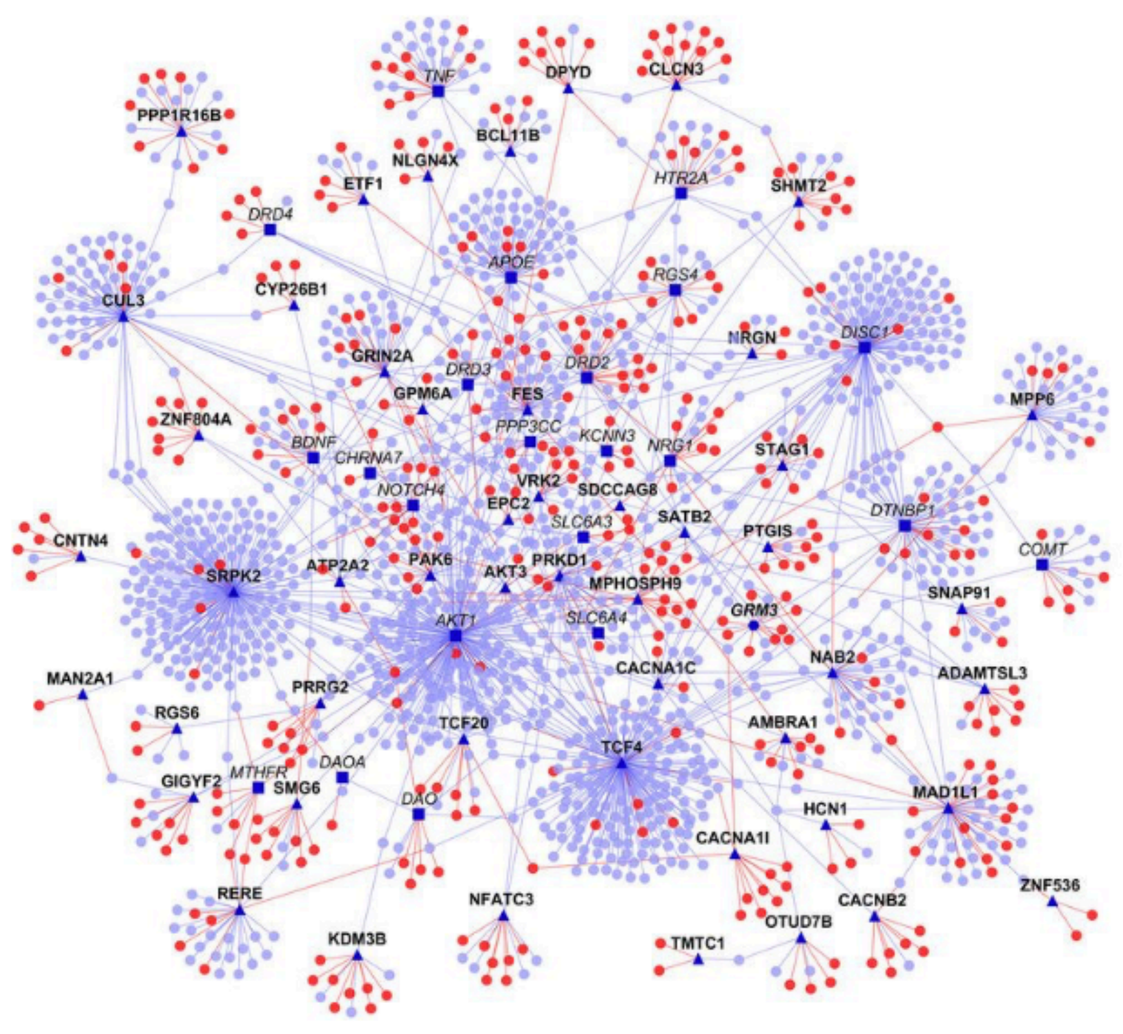
Graph-structured Data

A lot of real-world data does not “live” on grids

Social networks
Citation networks
Communication networks
Multi-agent systems



Protein interaction networks

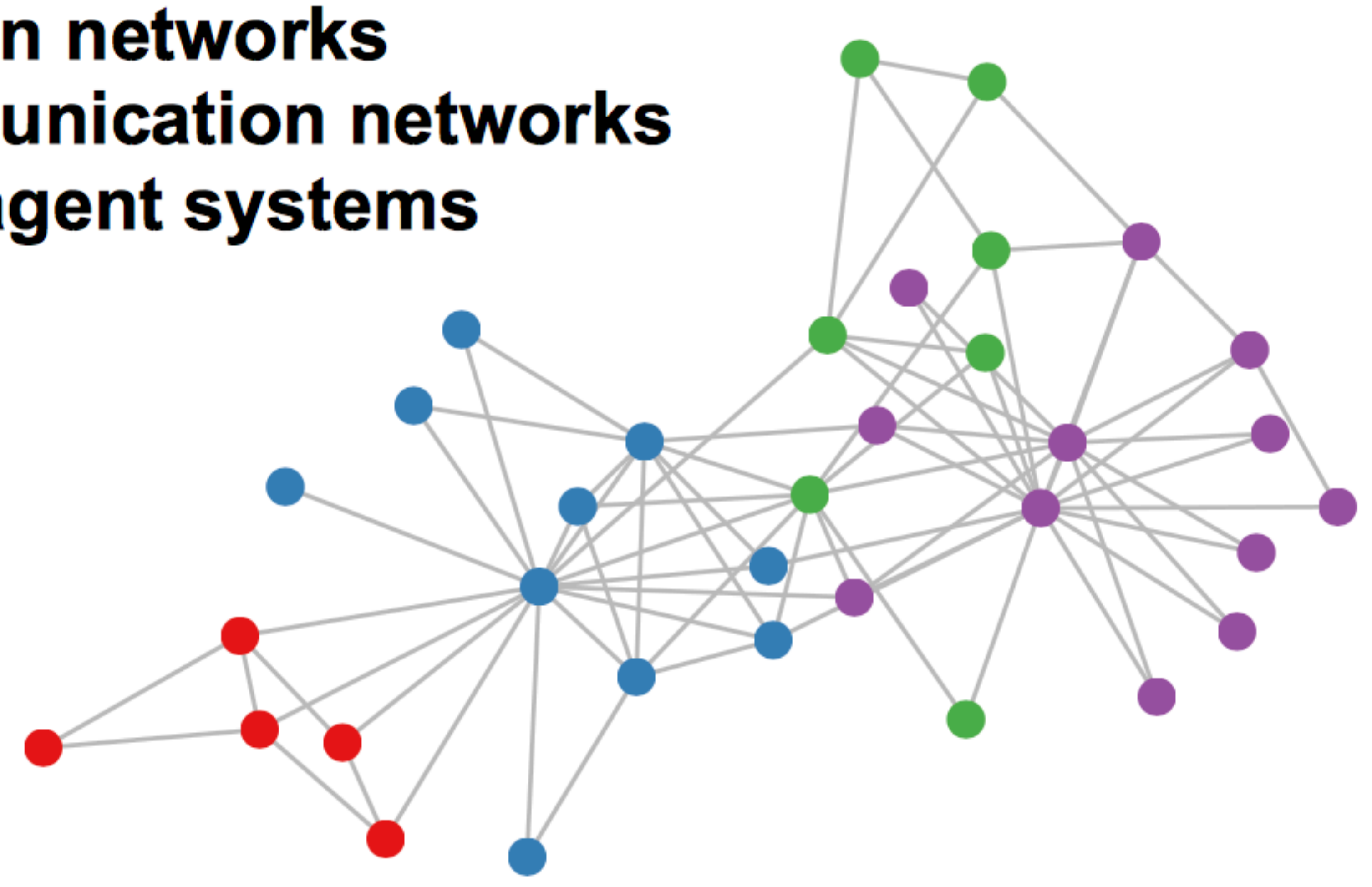


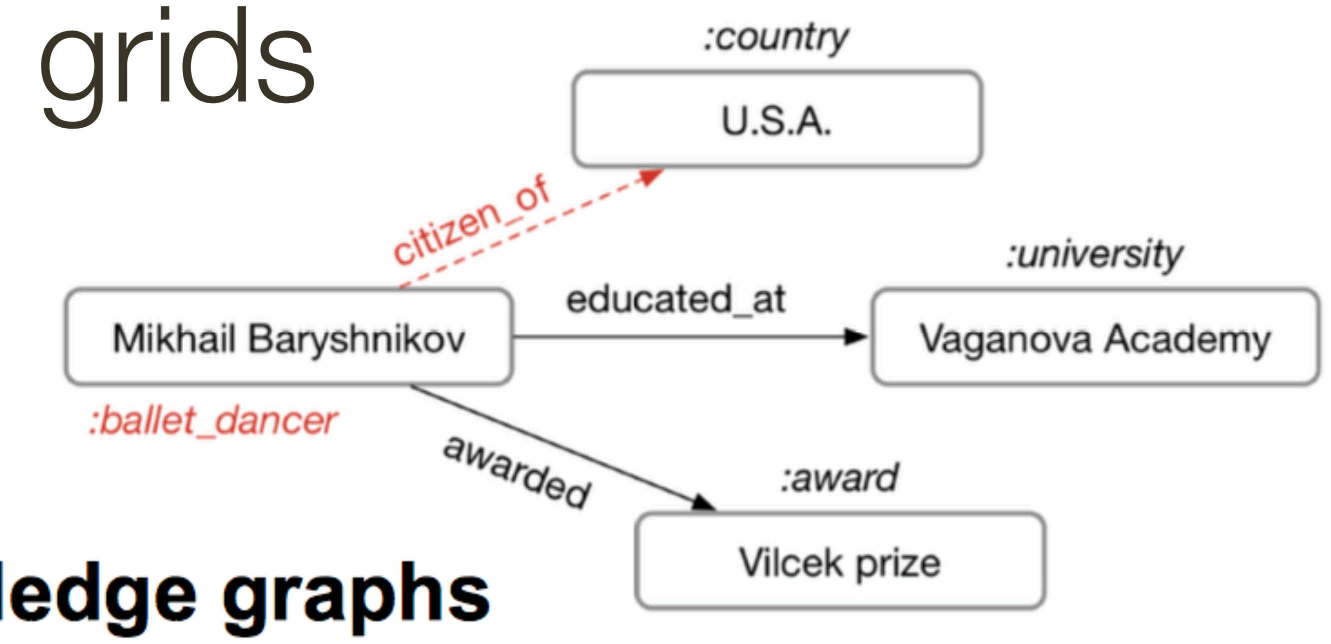
* slide from Thomas Kipf, **University of Amsterdam**

Graph-structured Data

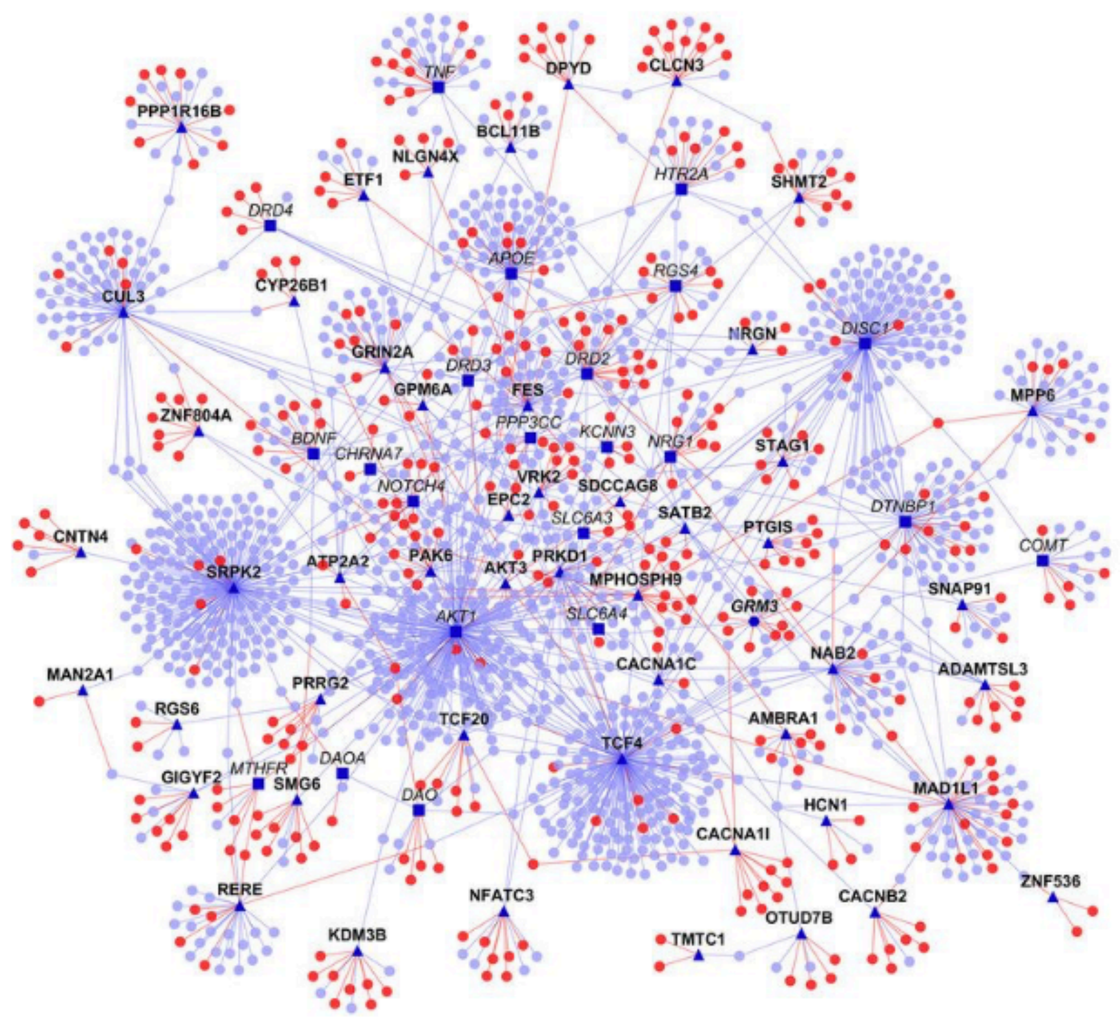
A lot of real-world data does not "live" on grids

Social networks
Citation networks
Communication networks
Multi-agent systems





Knowledge graphs



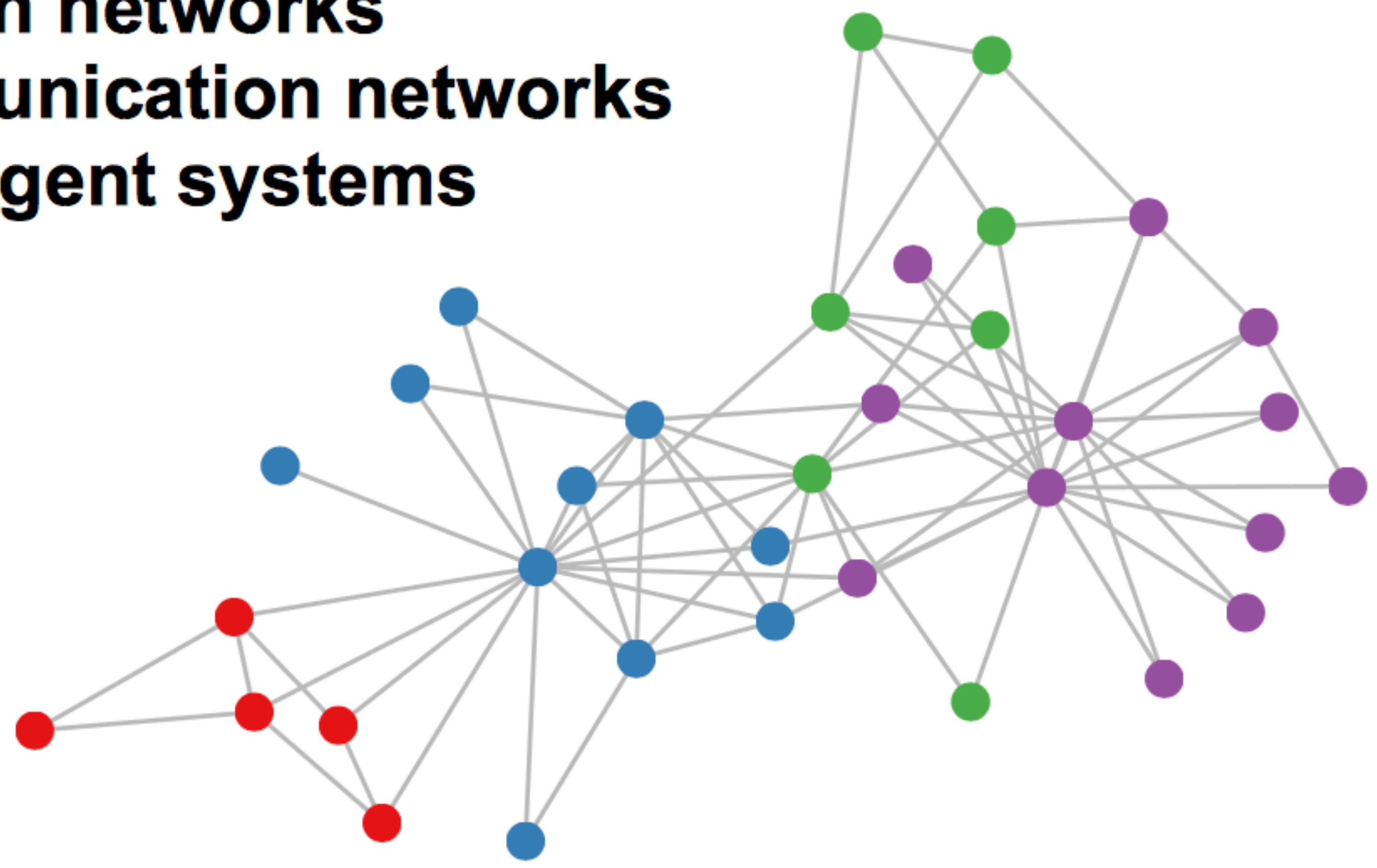
Protein interaction networks

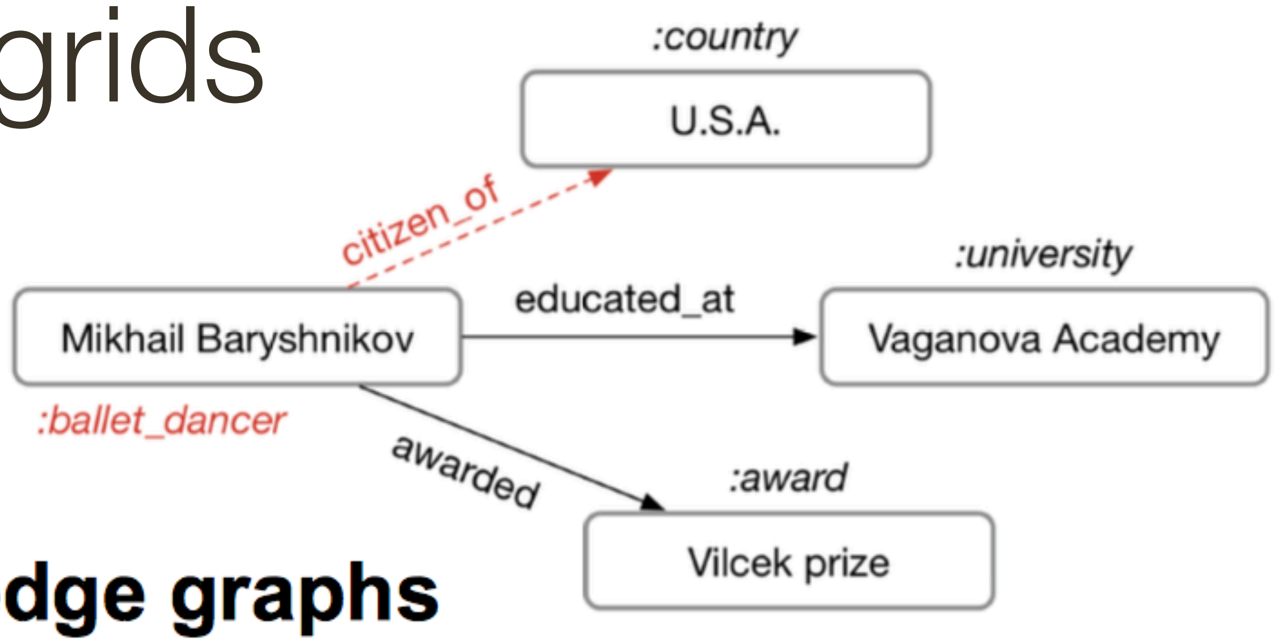
* slide from Thomas Kipf, **University of Amsterdam**

Graph-structured Data

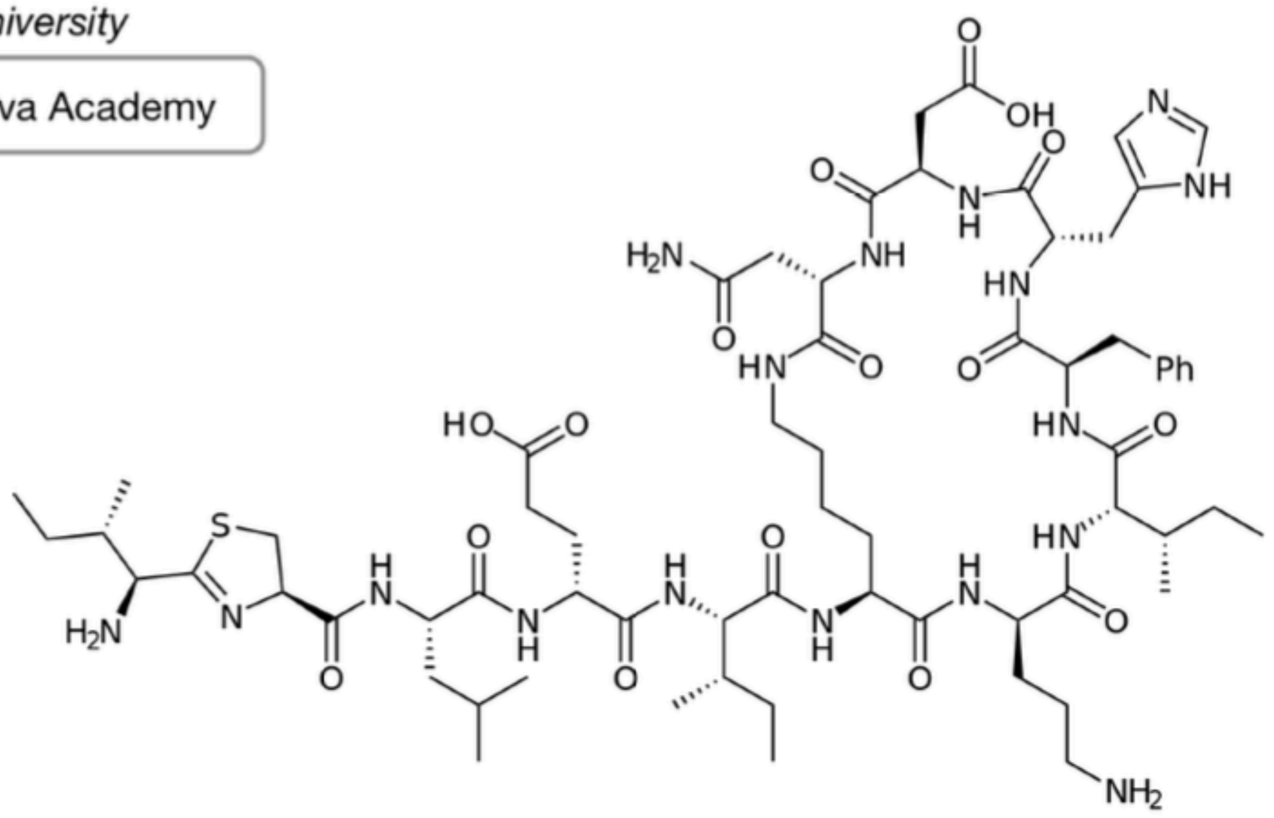
A lot of real-world data does not “live” on grids

Social networks
Citation networks
Communication networks
Multi-agent systems



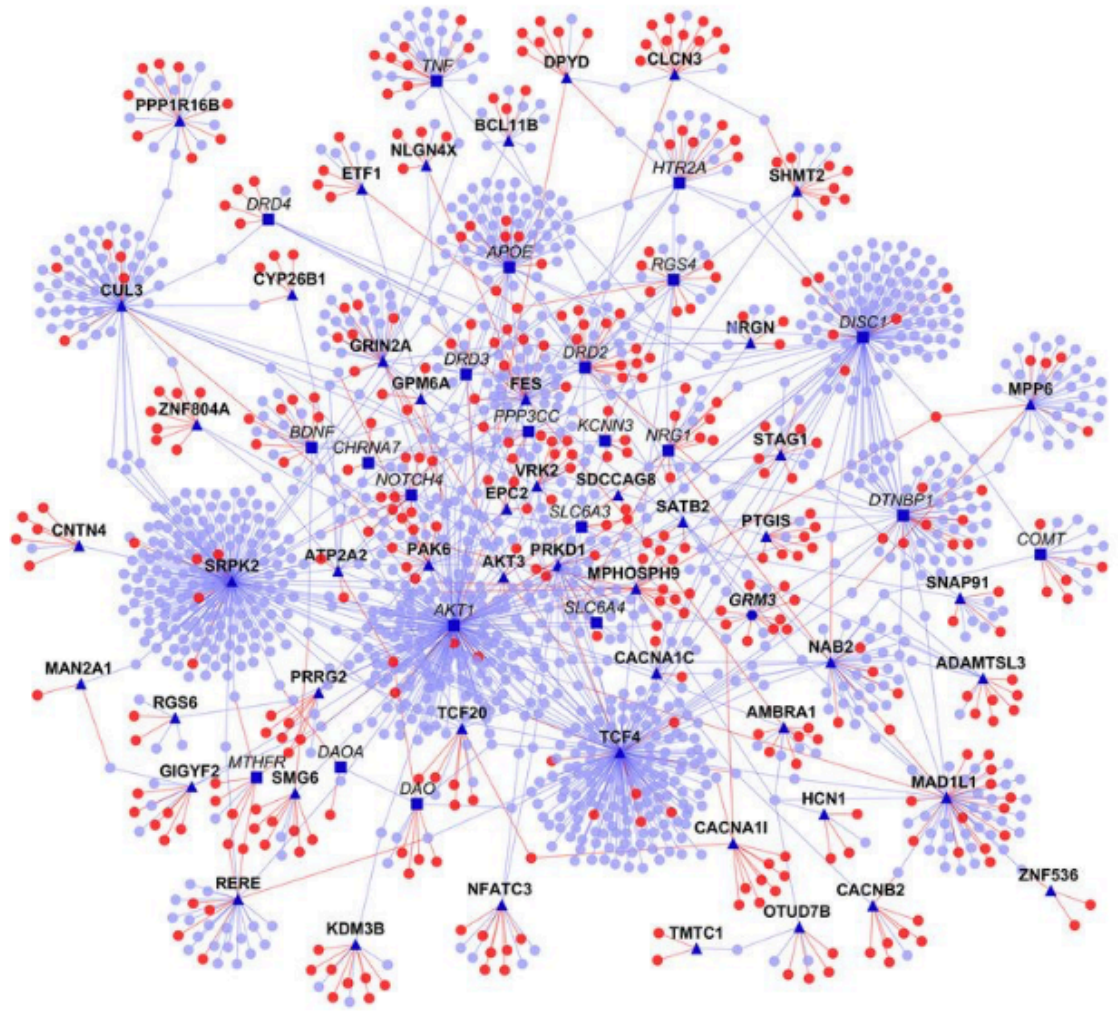


Knowledge graphs

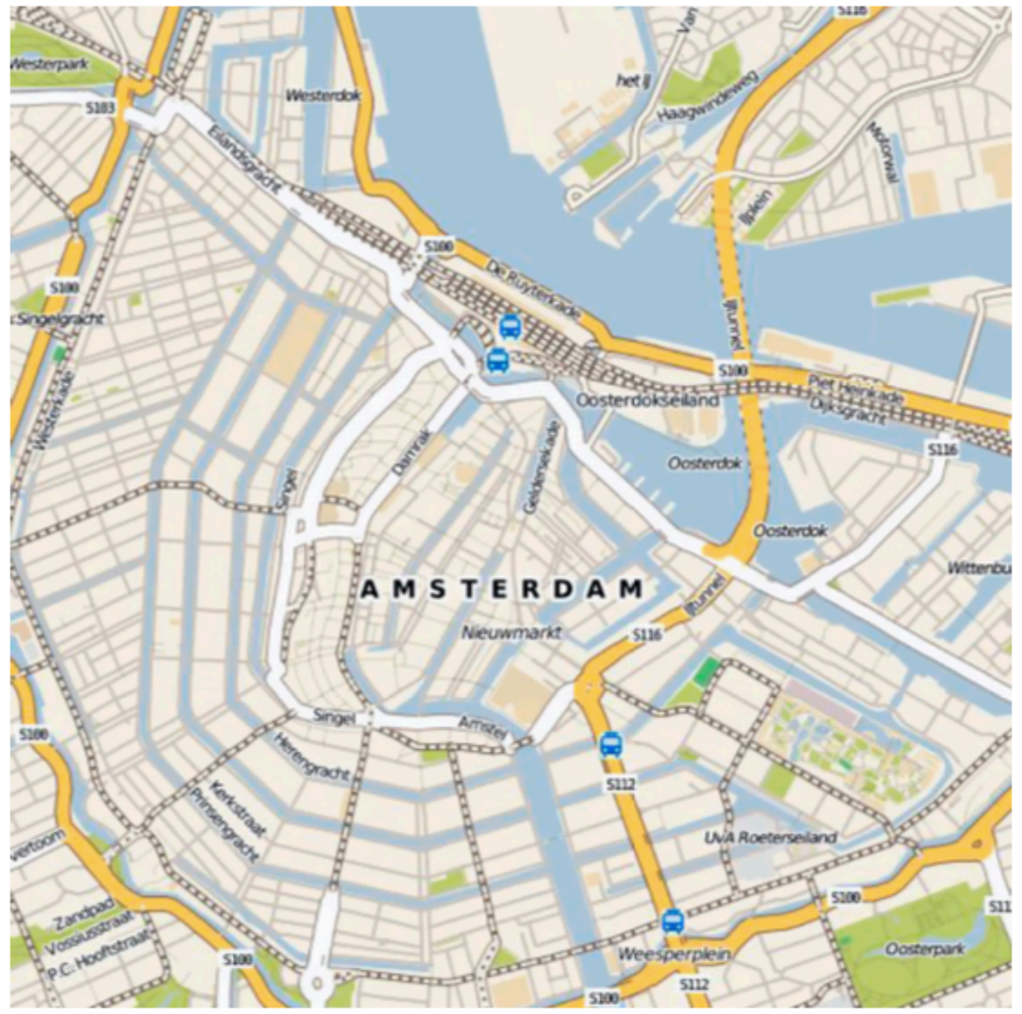


Molecules

Protein interaction networks



Road maps

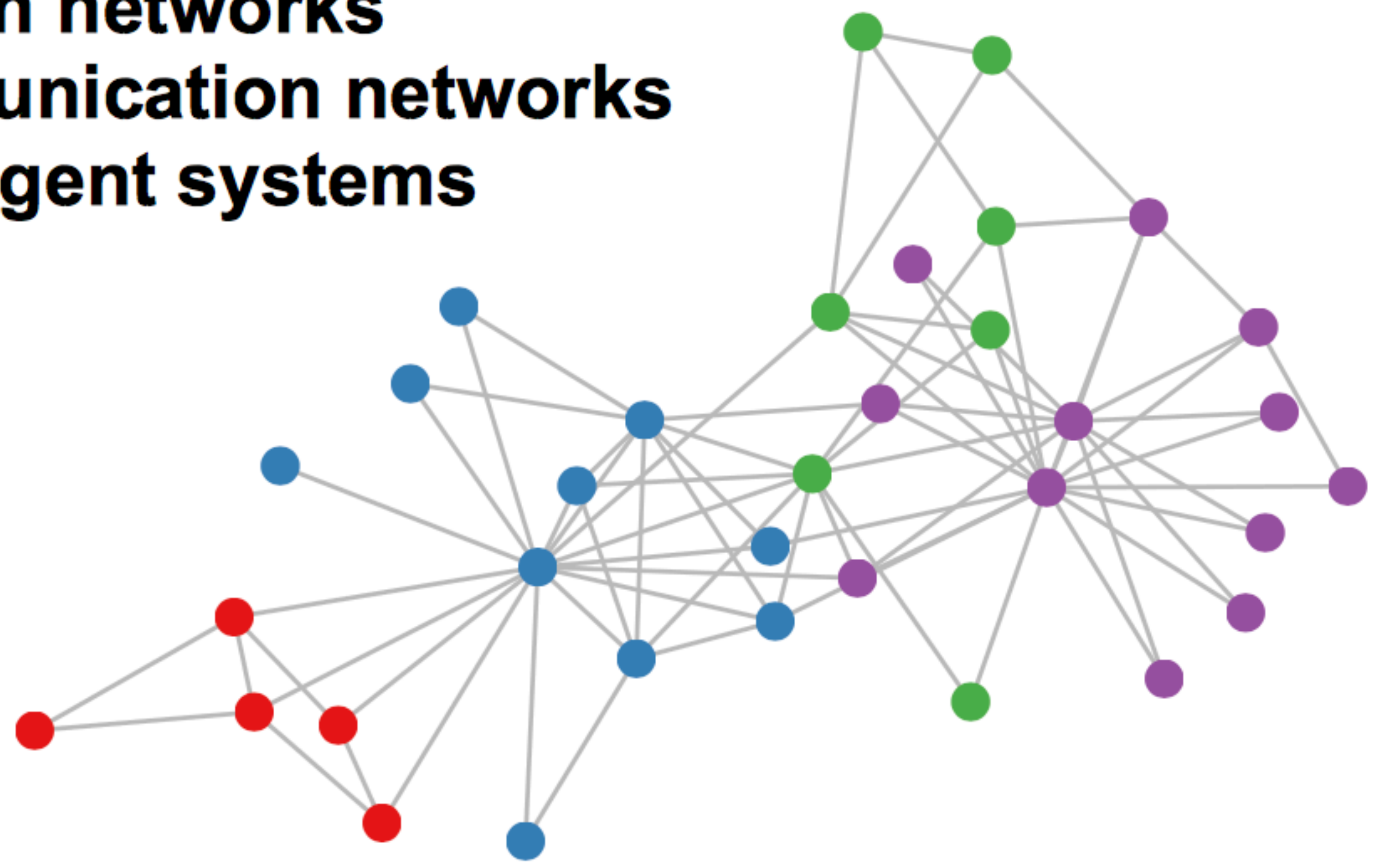


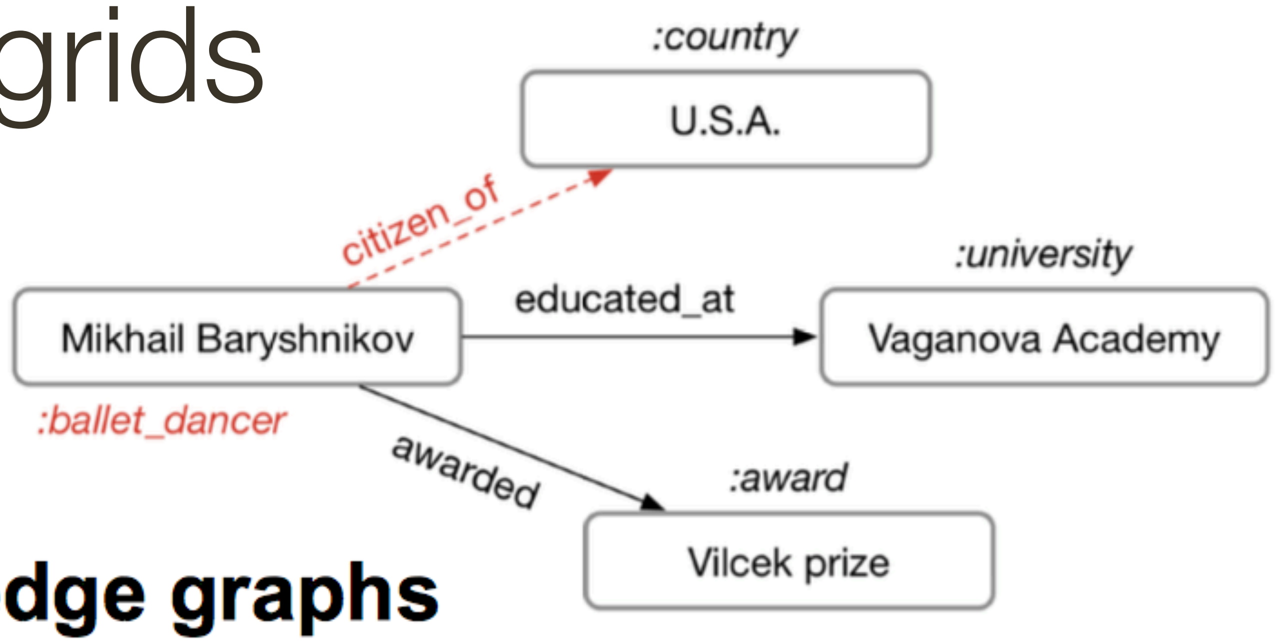
* slide from Thomas Kipf, **University of Amsterdam**

Graph-structured Data

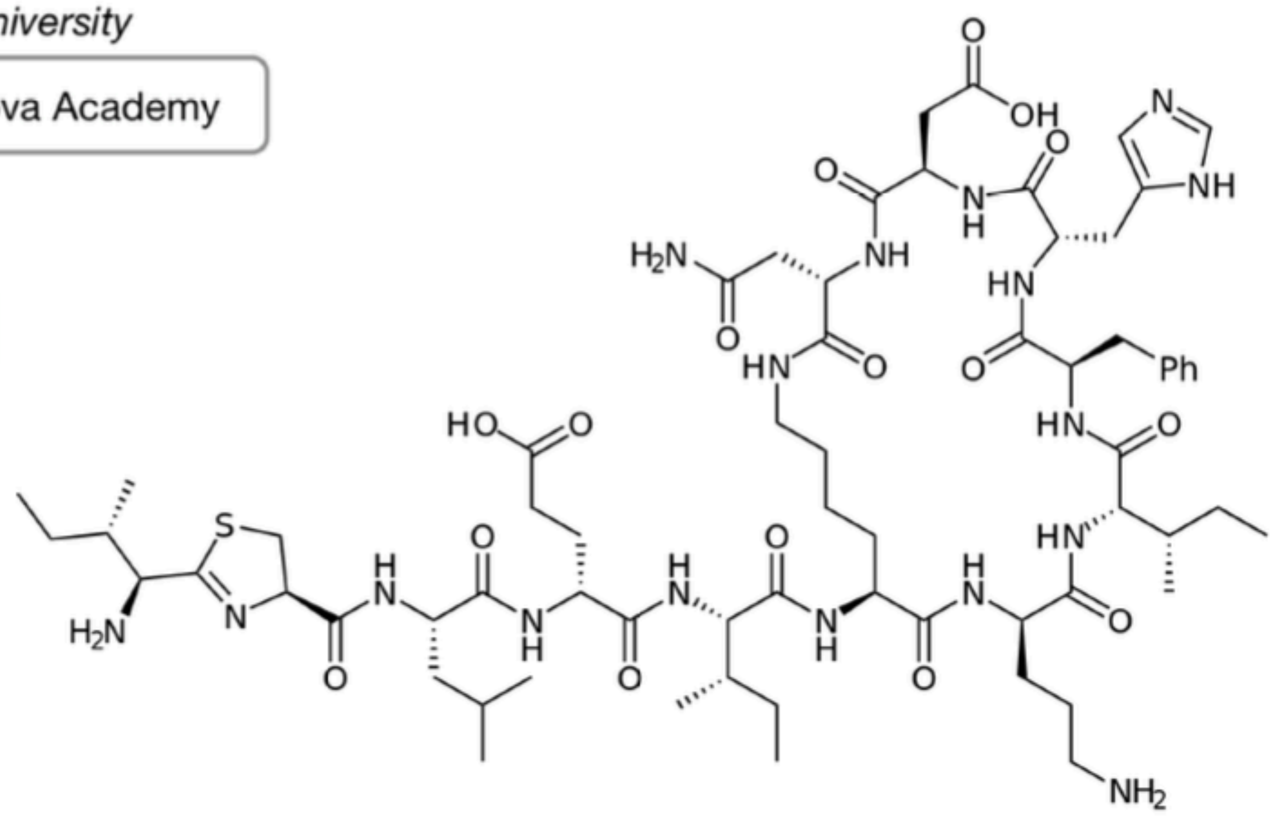
A lot of real-world data does not “live” on grids

Social networks
Citation networks
Communication networks
Multi-agent systems

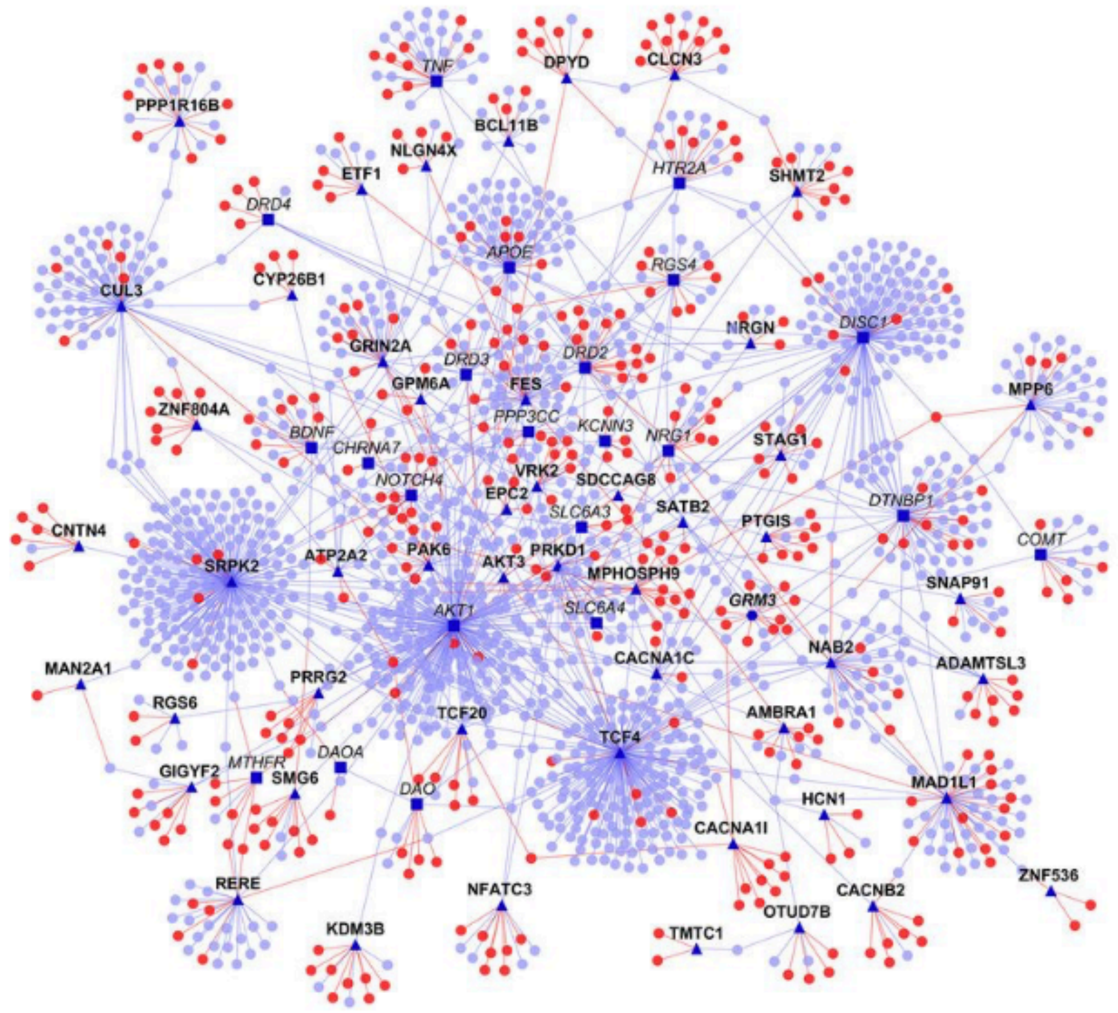




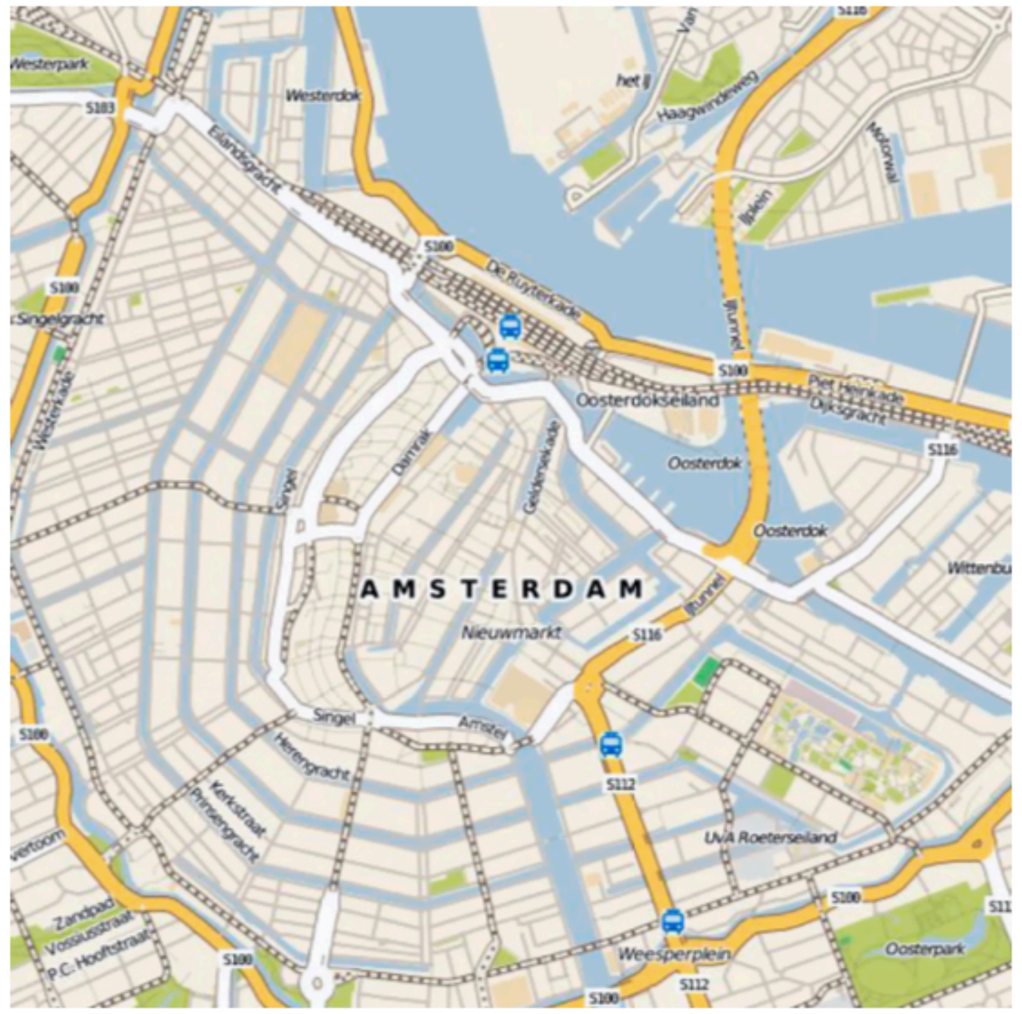
Knowledge graphs



Molecules



Protein interaction networks

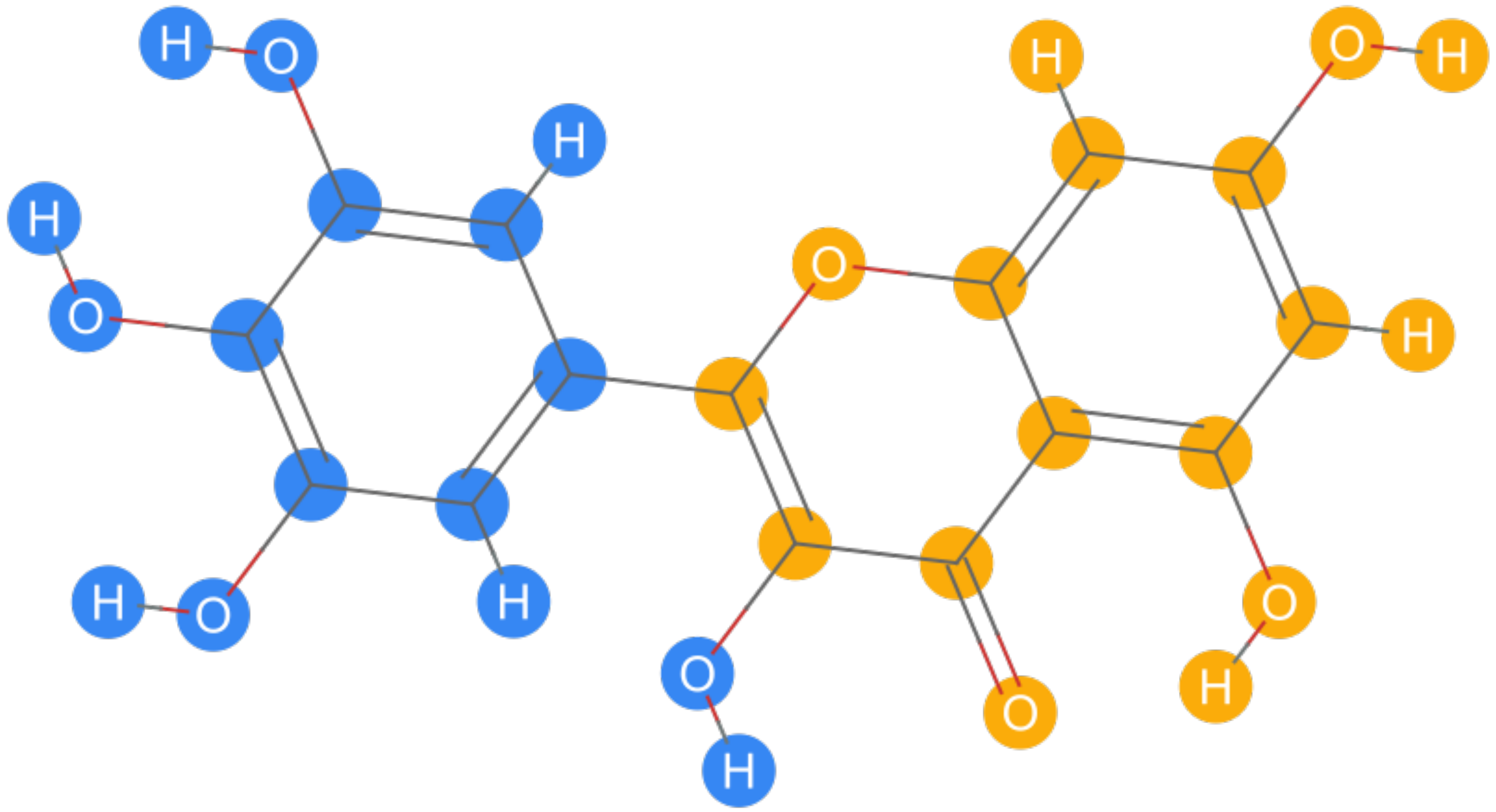


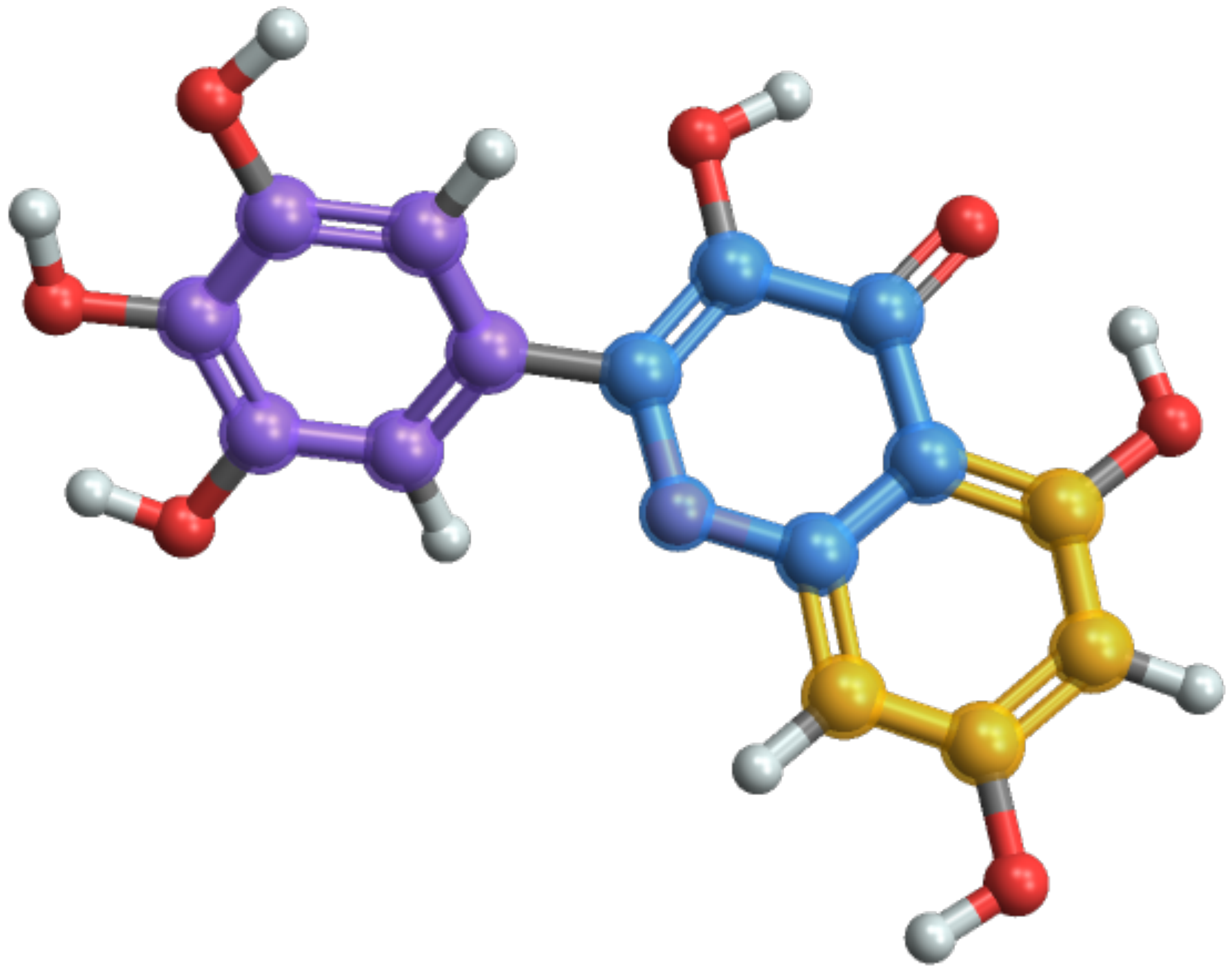
Road maps

Standard **CNN** and **RNN** architectures don’t work on this data

* slide from Thomas Kipf, **University of Amsterdam**

Graph-structured Data — Molecular Data





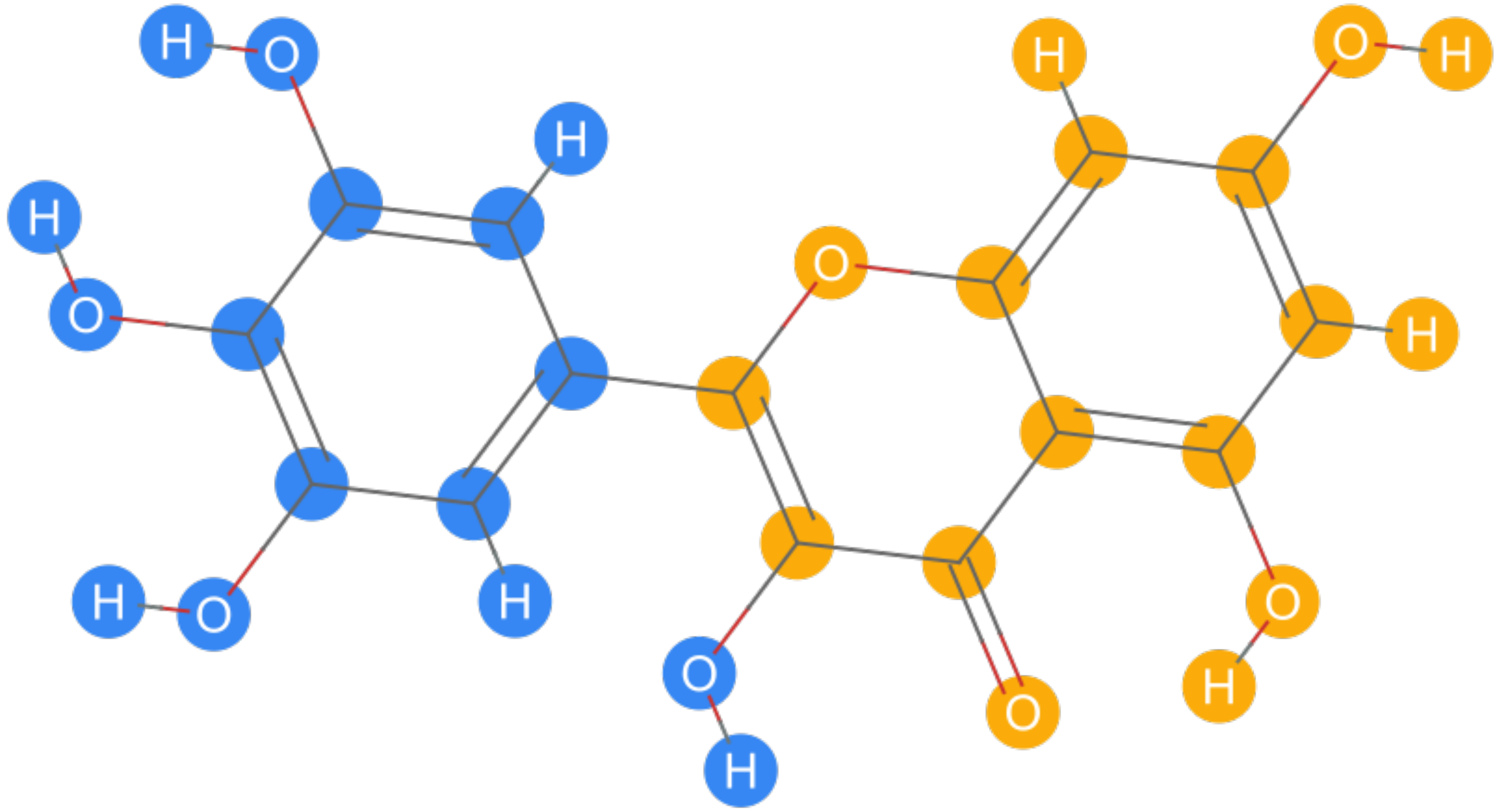
Characteristics:

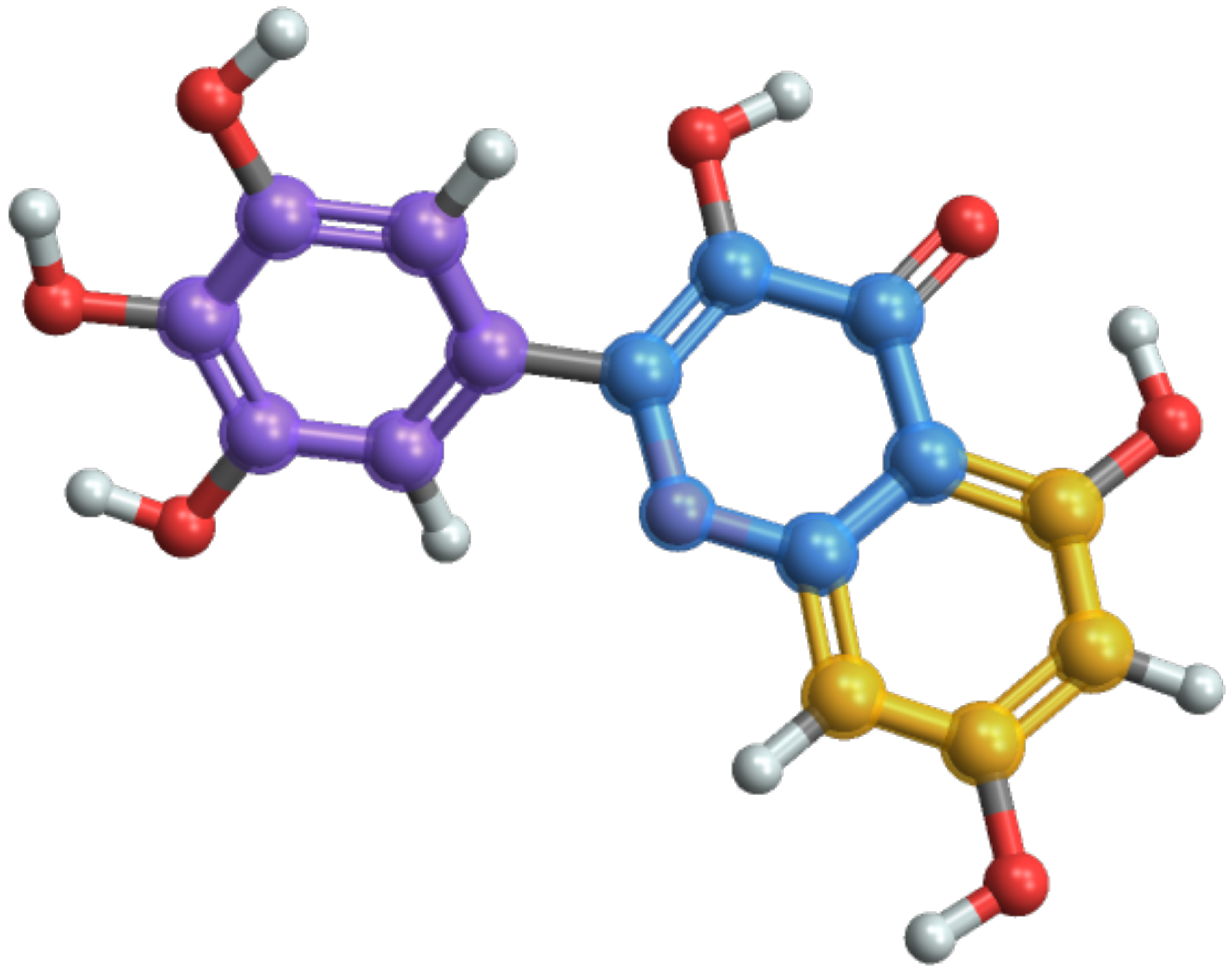
— Nodes have different types (atoms)

— Edges have types and there could be multiple edges (types of bonds)

Image Credit: https://www.wolfram.com/language/12/molecular-structure-and-computation/molecule-graphs.html.en?product=mathematica
https://www.wolfram.com/language/12/molecular-structure-and-computation/molecule-graphs.html.en?product=mathematica

Graph-structured Data — Molecular Data



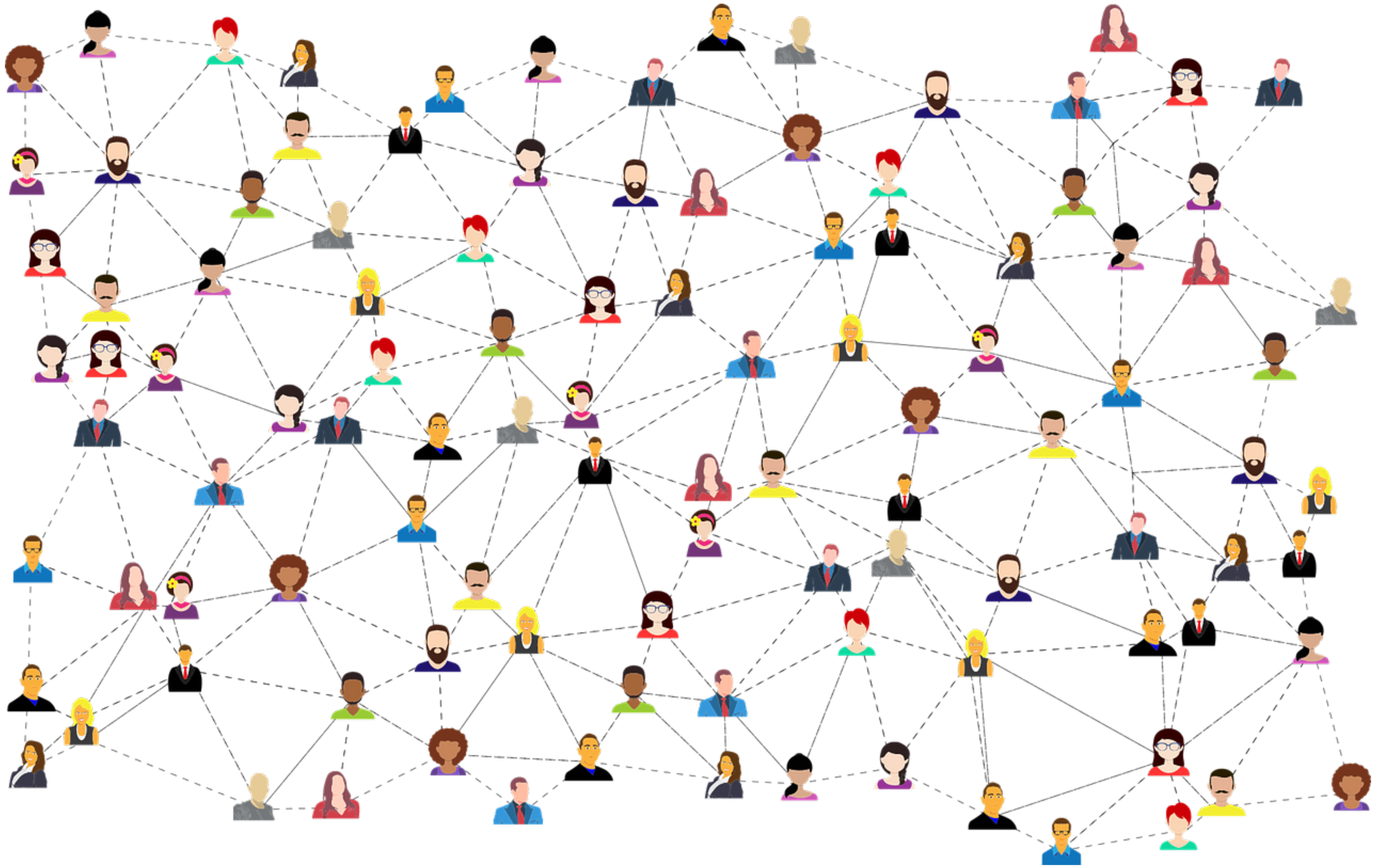


Problems:

- Molecule property prediction (physiology, physical chemistry, quantum mechanics)
- Molecular scoring or docking (pharmacology — screening of drug candidates)
- Molecular dynamics (e.g., position or motion of atoms)

Image Credit: https://www.wolfram.com/language/12/molecular-structure-and-computation/molecule-graphs.html.en?product=mathematica
https://www.wolfram.com/language/12/molecular-structure-and-computation/molecule-graphs.html.en?product=mathematica

Graph-structured Data — Social Networks



Problems:

Link prediction

Image Credit: http://blog.soton.ac.uk/hive/2012/05/10/recommendation-system-of-hive/

Graph-structured Data — Network Recommendation

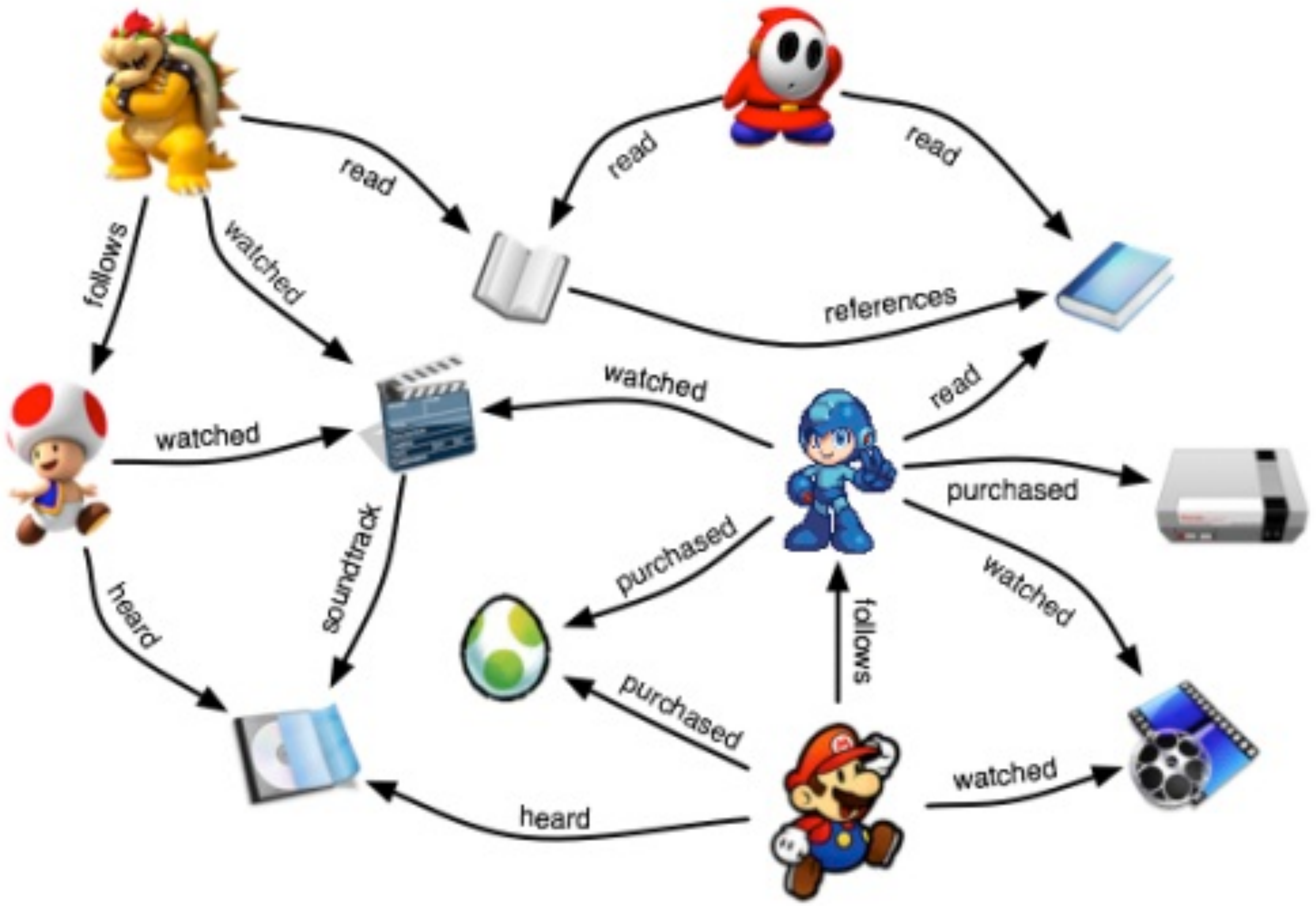


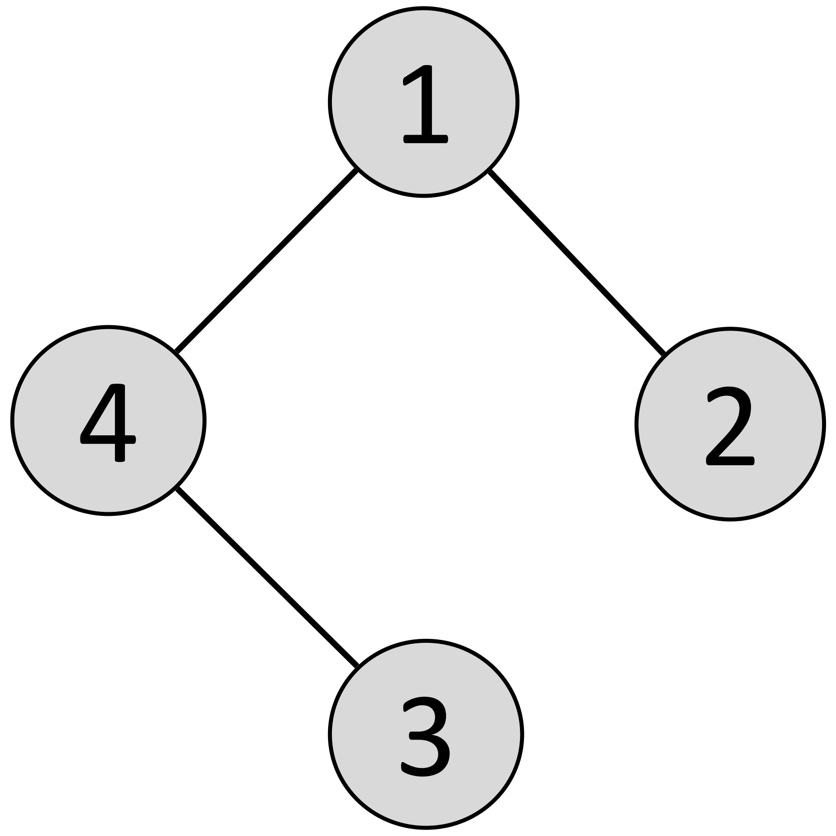
Image Credit: https://www.science.org/doi/10.1126/sciadv.aau4212

Graph **Representations**

Connectivity

— Adjacency List: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

— Adjacency Matrix: A (sometimes we have weights)



V = {1,2,3,4}, E = {(1,2), (1,4), (4,3)}

	1	2	3	4
1	0	1	0	1
2	1	0	0	0
3	0	0	0	1
4	1	0	1	0

Feature

— Node Feature: X

— Edge Feature

— Graph Feature

Graph Data: (A, X)

- Adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$
- Feature matrix $\mathbf{X} \in \mathbb{R}^{N \times F}$

Slide from Renjie Liao

Graph **Isomorphism**

Permutation

V = [1,2,3,4] => V’ = [2,1,3,4]

E = [(1,2), (1,4), (4,3)] => E’ = [(2,1), (2,4), (4,3)]

Permute Rows

	1	2	3	4
1	0	1	0	0
2	1	0	0	0
3	0	0	1	0
4	0	0	0	1

Permutation Matrix

	1	2	3	4
1	0	1	0	1
2	1	0	0	0
3	0	0	0	1
4	1	0	1	0

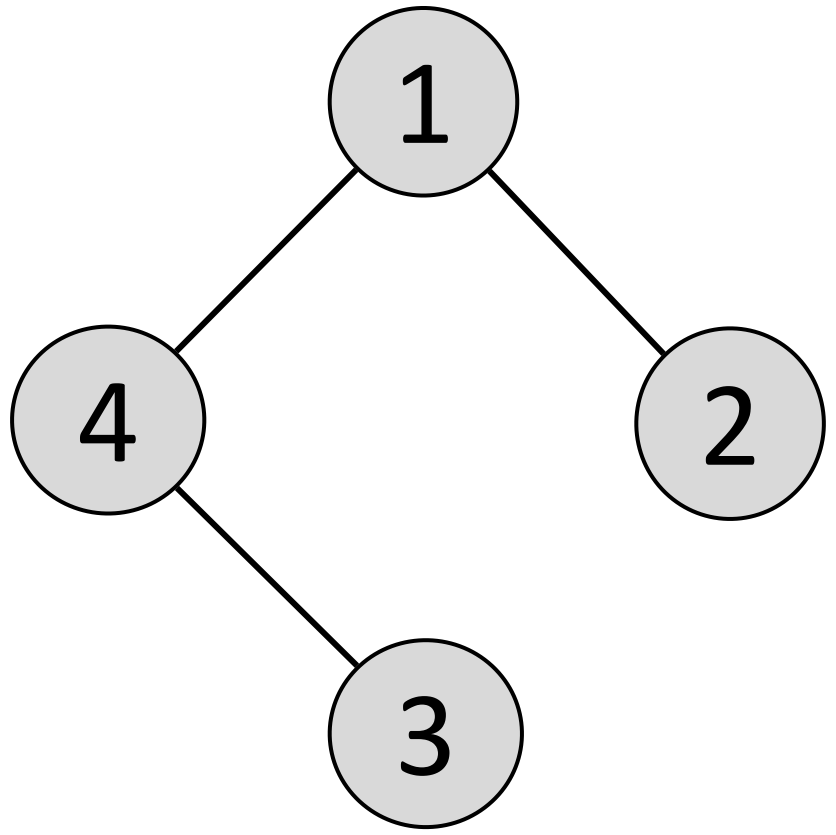
Original Adj Matrix

Permute Columns

	1	2	3	4
1	0	1	0	0
2	1	0	0	0
3	0	0	1	0
4	0	0	0	1

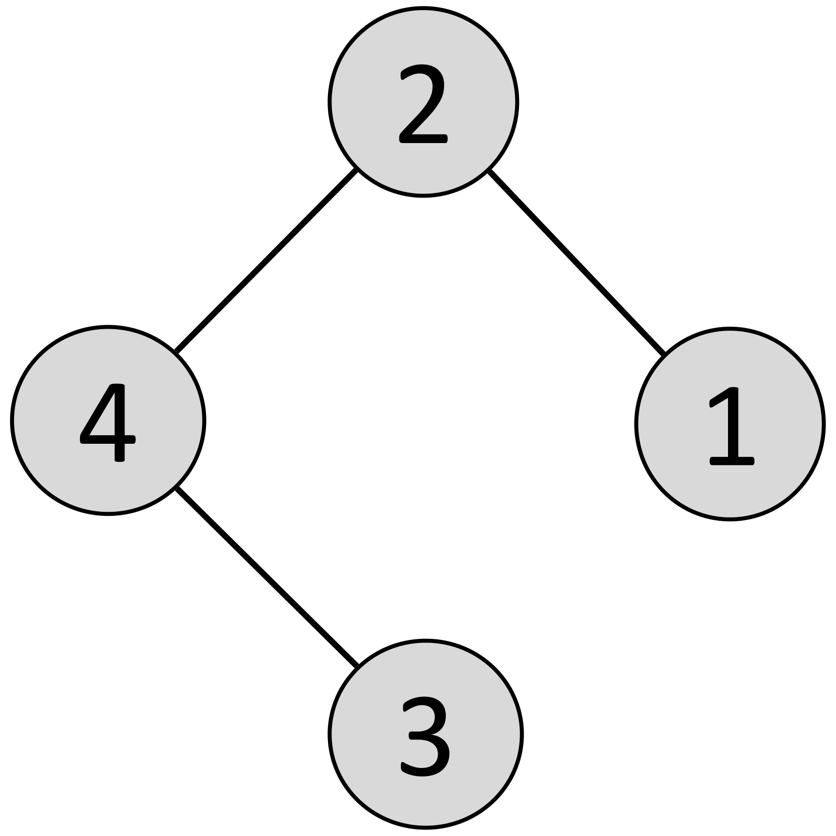
Transposed Permutation Matrix

=



	1	2	3	4
1	0	1	0	1
2	1	0	0	0
3	0	0	0	1
4	1	0	1	0

V = {1,2,3,4}, E = {(1,2), (1,4), (4,3)}

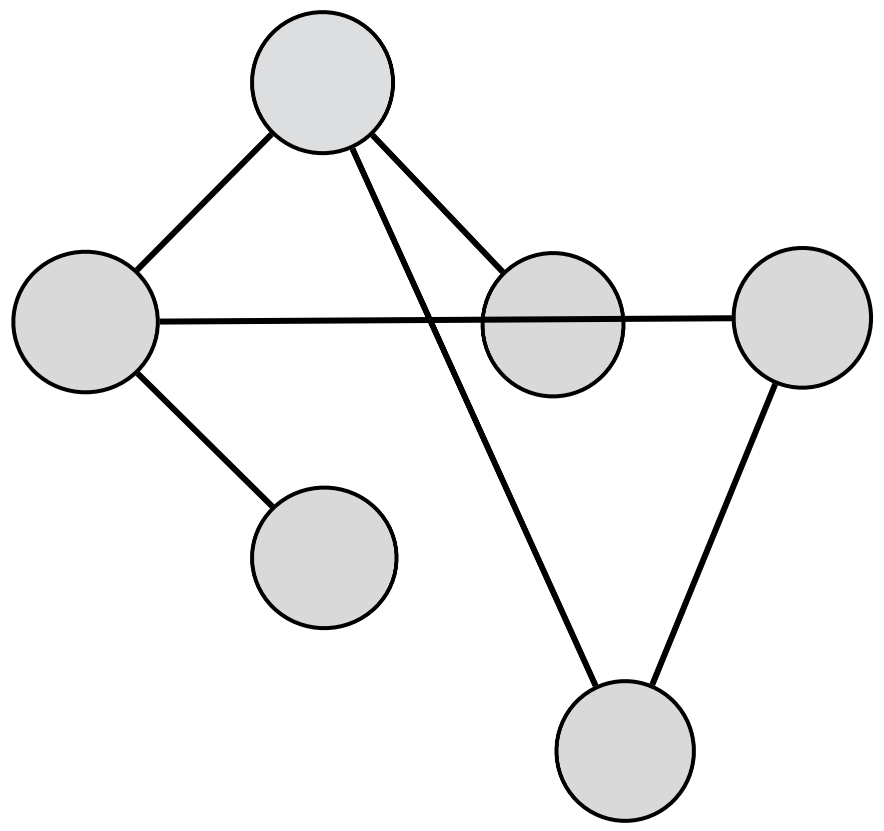


	1	2	3	4
1	0	1	0	1
2	1	0	0	0
3	0	0	0	1
4	1	0	1	0

V = {2,1,3,4}, E = {(2,1), (2,4), (4,3)}

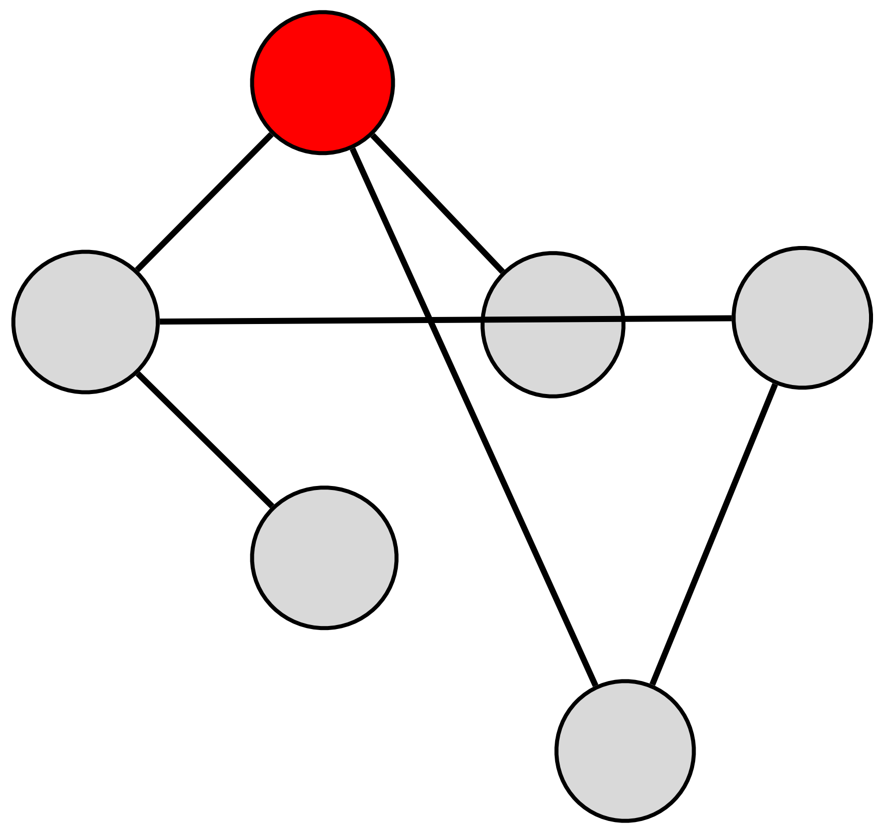
Slide from Renjie Liao

Challenge #1: **Unordered** Neighborhoods



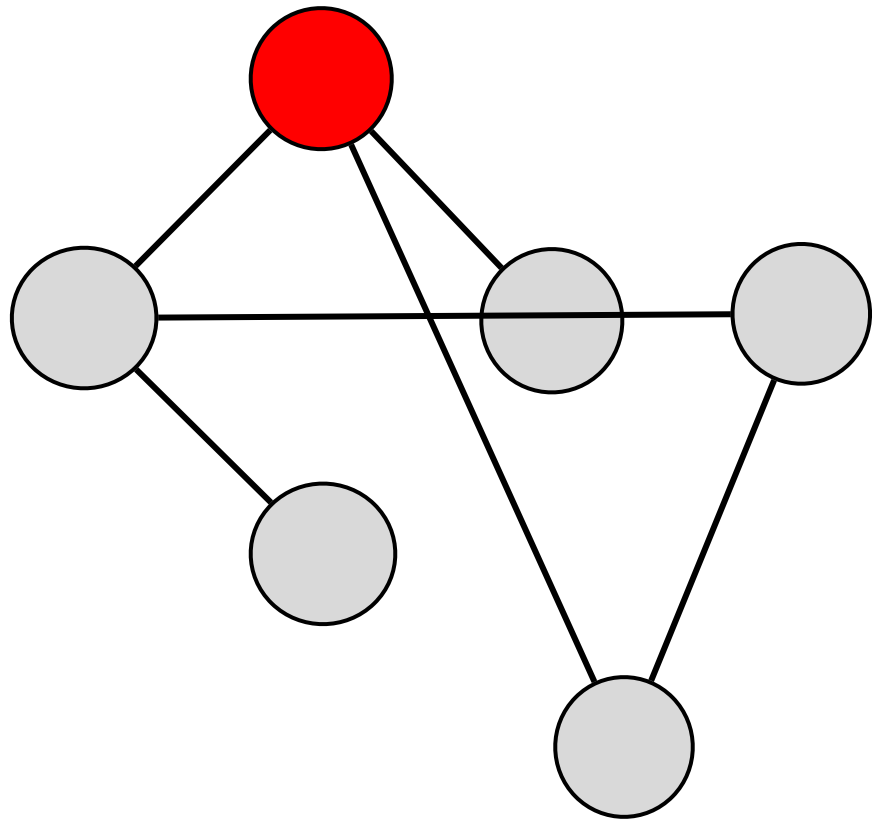
Slide from Renjie Liao

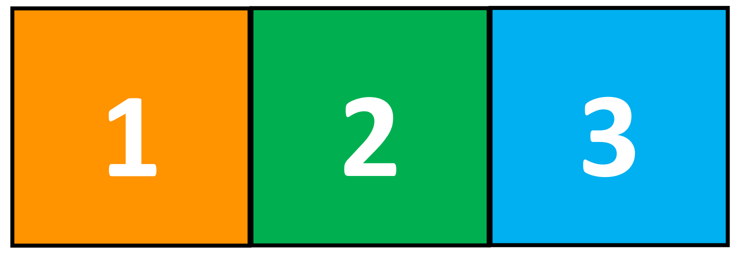
Challenge #1: **Unordered** Neighborhoods



Slide from Renjie Liao

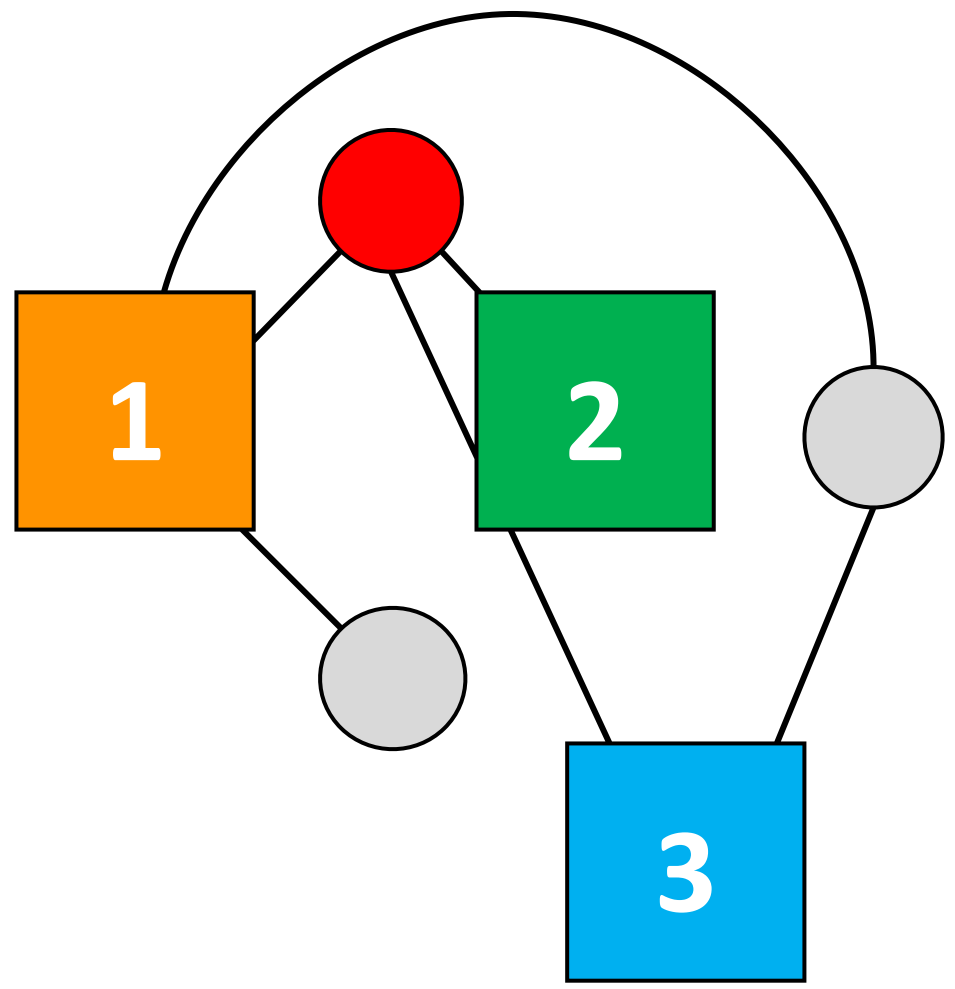
Challenge #1: **Unordered** Neighborhoods

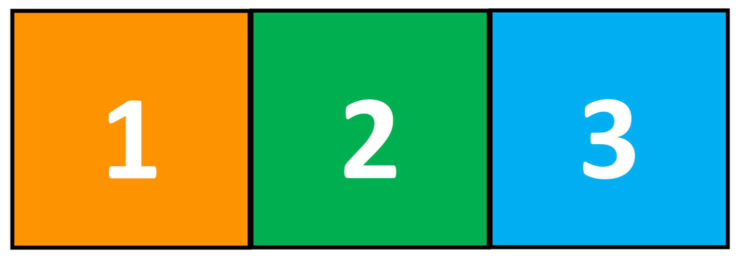




Slide from Renjie Liao

Challenge #1: **Unordered** Neighborhoods

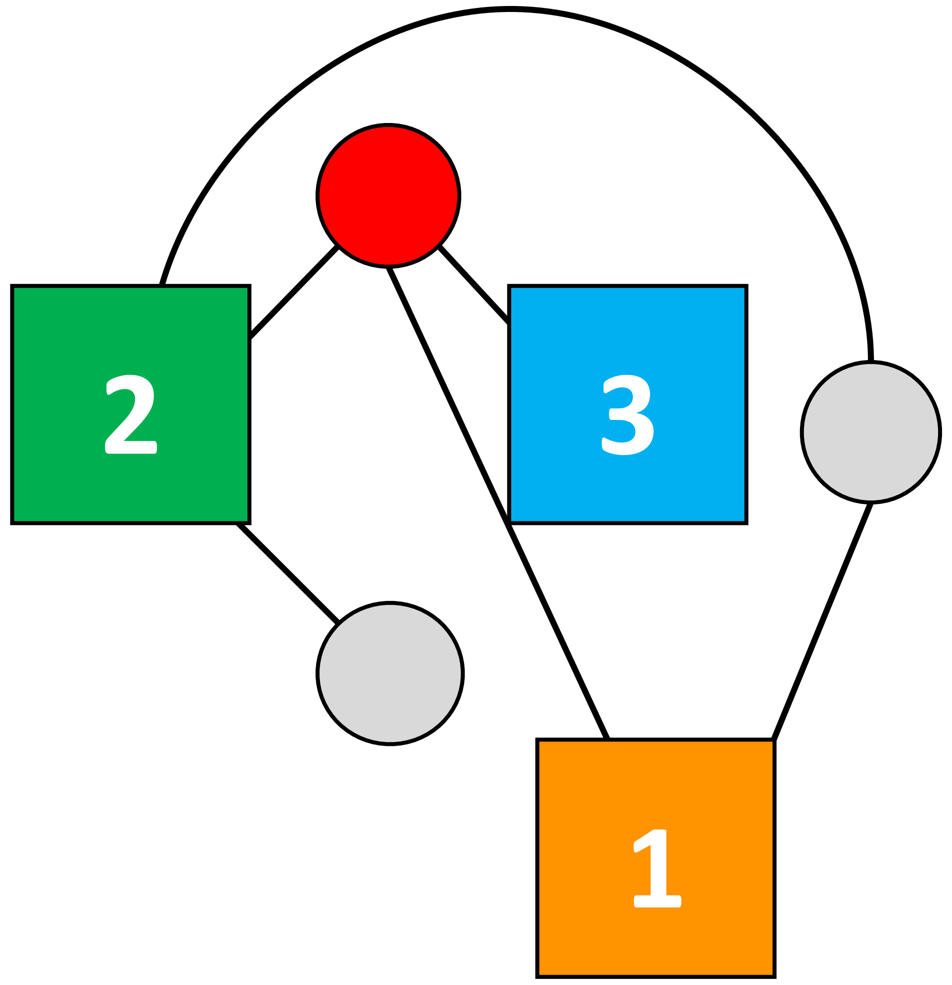




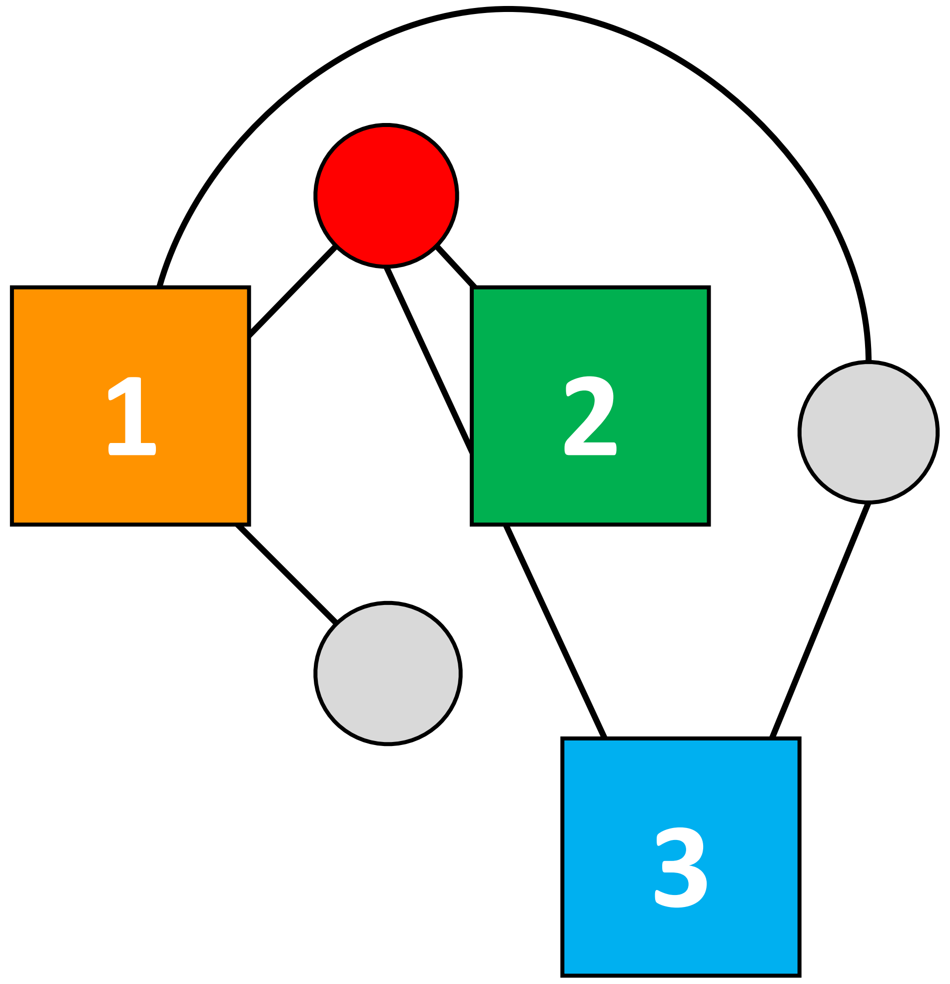
Option 1

Slide from Renjie Liao

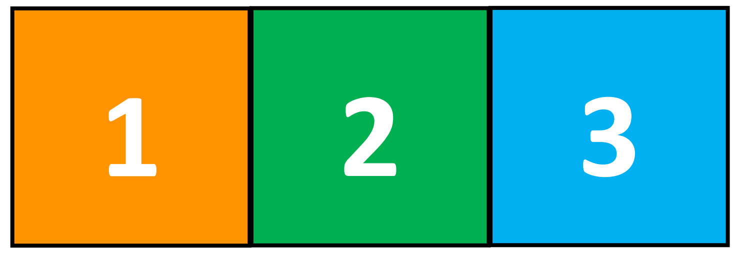
Challenge #1: **Unordered** Neighborhoods



Option 2

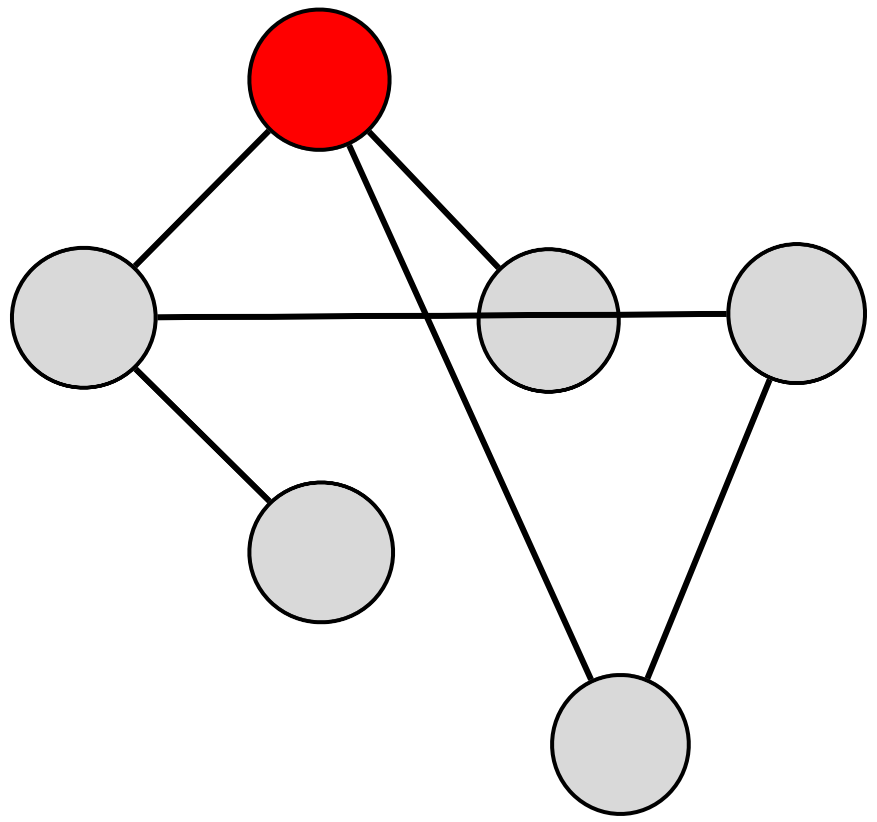


Option 1



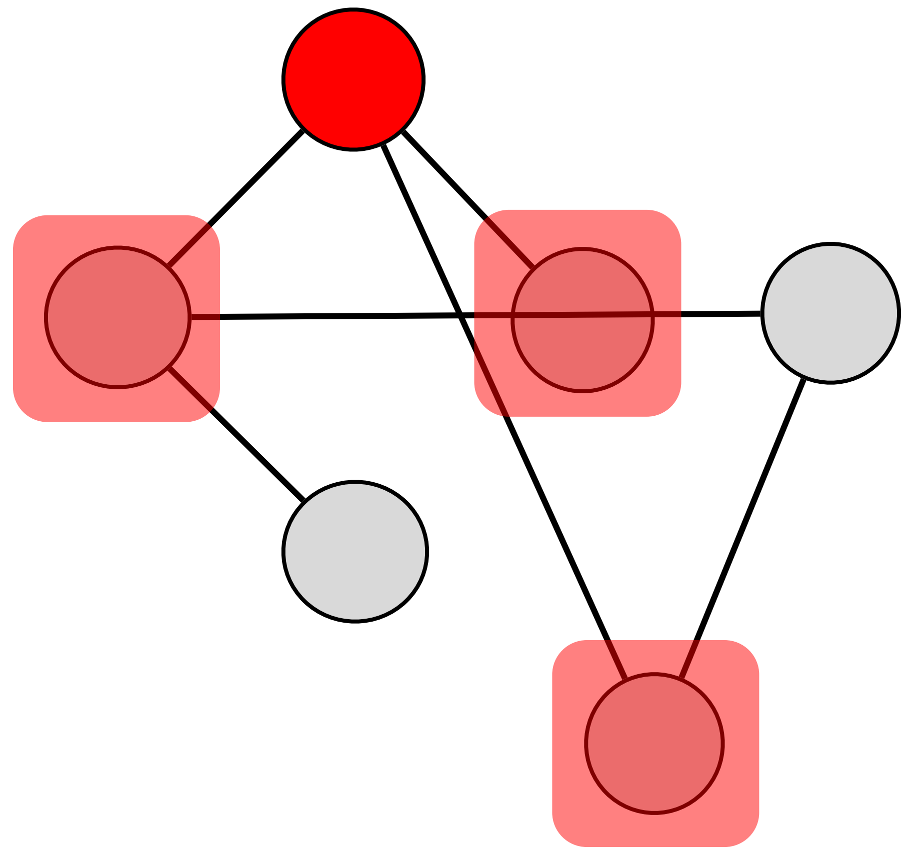
Slide from Renjie Liao

Challenge #2: **Varying** Neighborhood Sizes



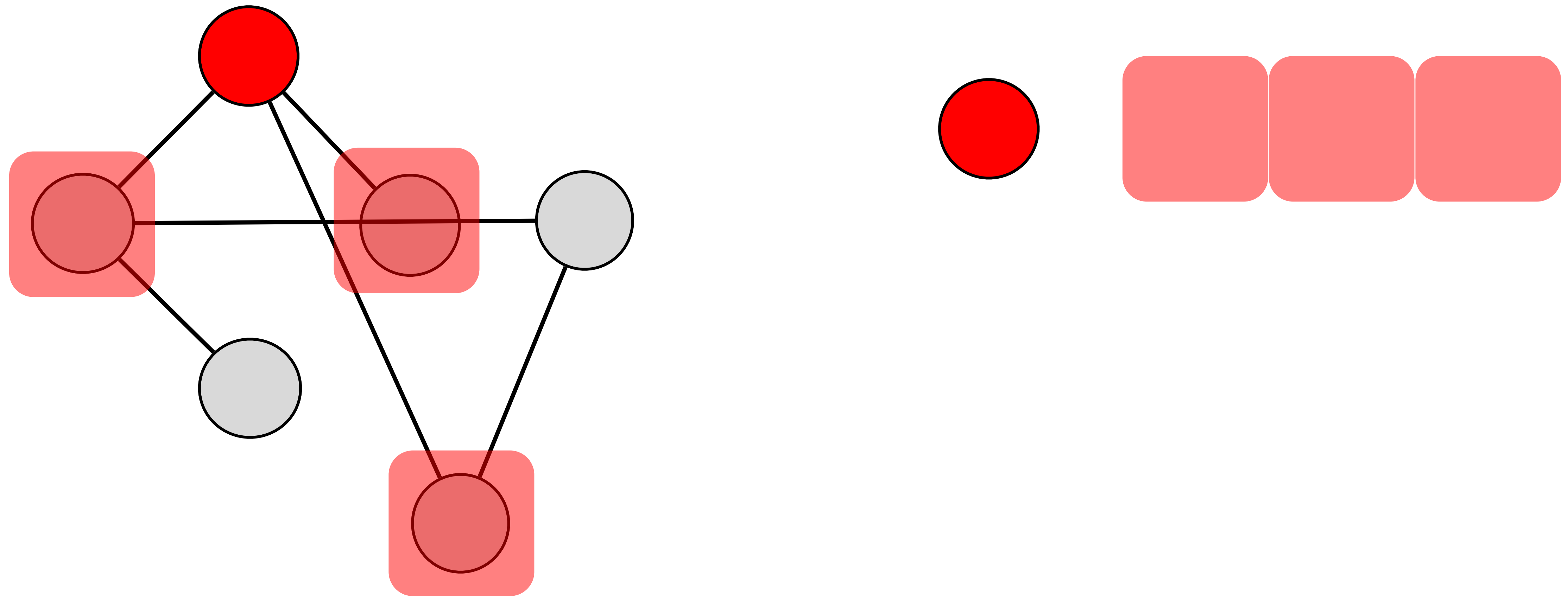
Slide from Renjie Liao

Challenge #2: **Varying** Neighborhood Sizes



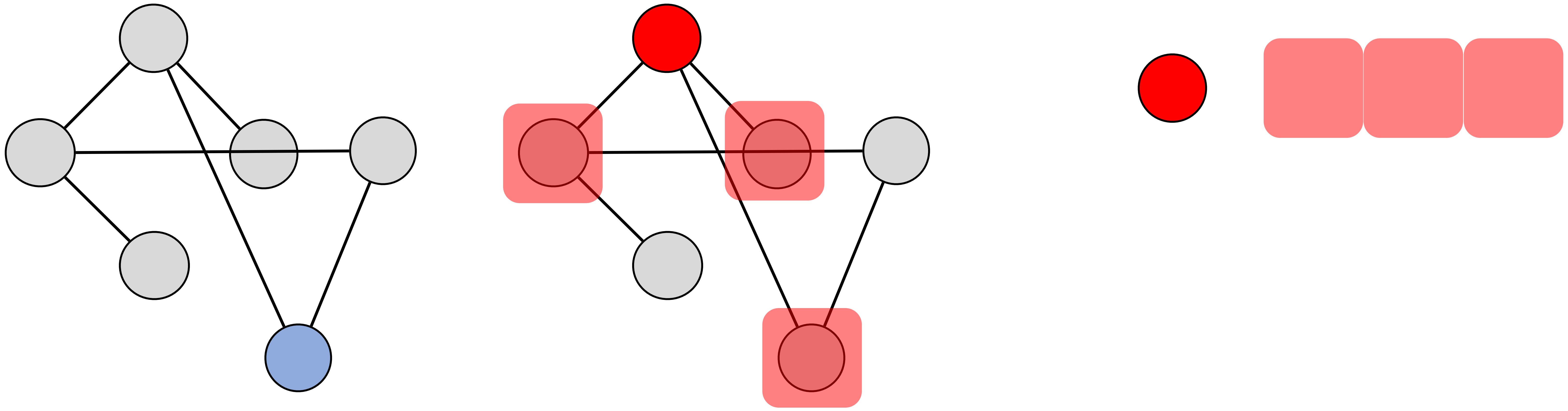
Slide from Renjie Liao

Challenge #2: **Varying** Neighborhood Sizes



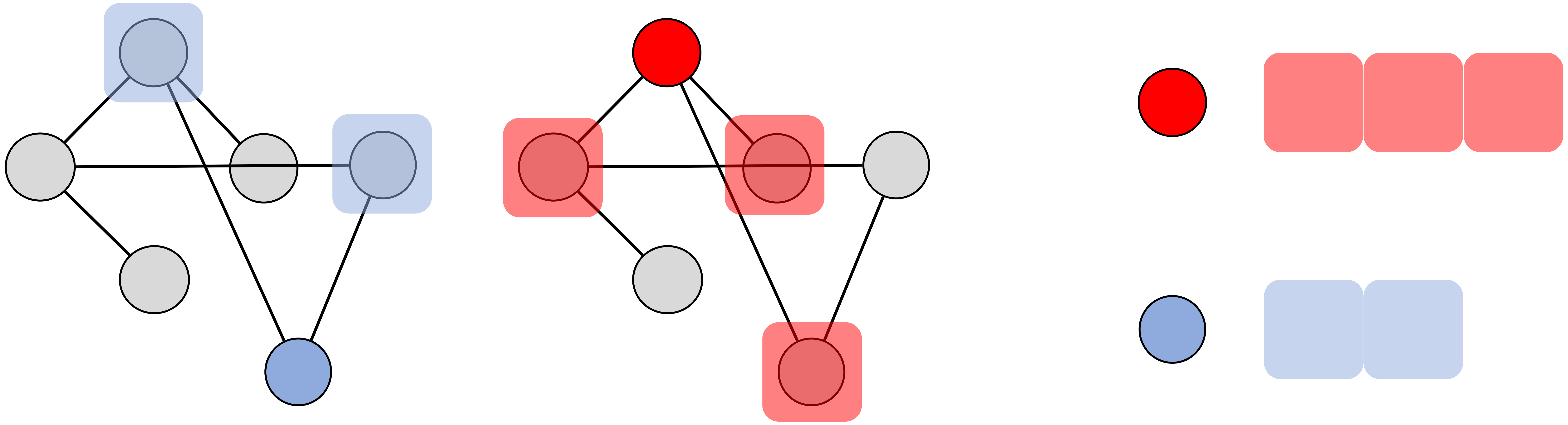
Slide from Renjie Liao

Challenge #2: **Varying** Neighborhood Sizes



Slide from Renjie Liao

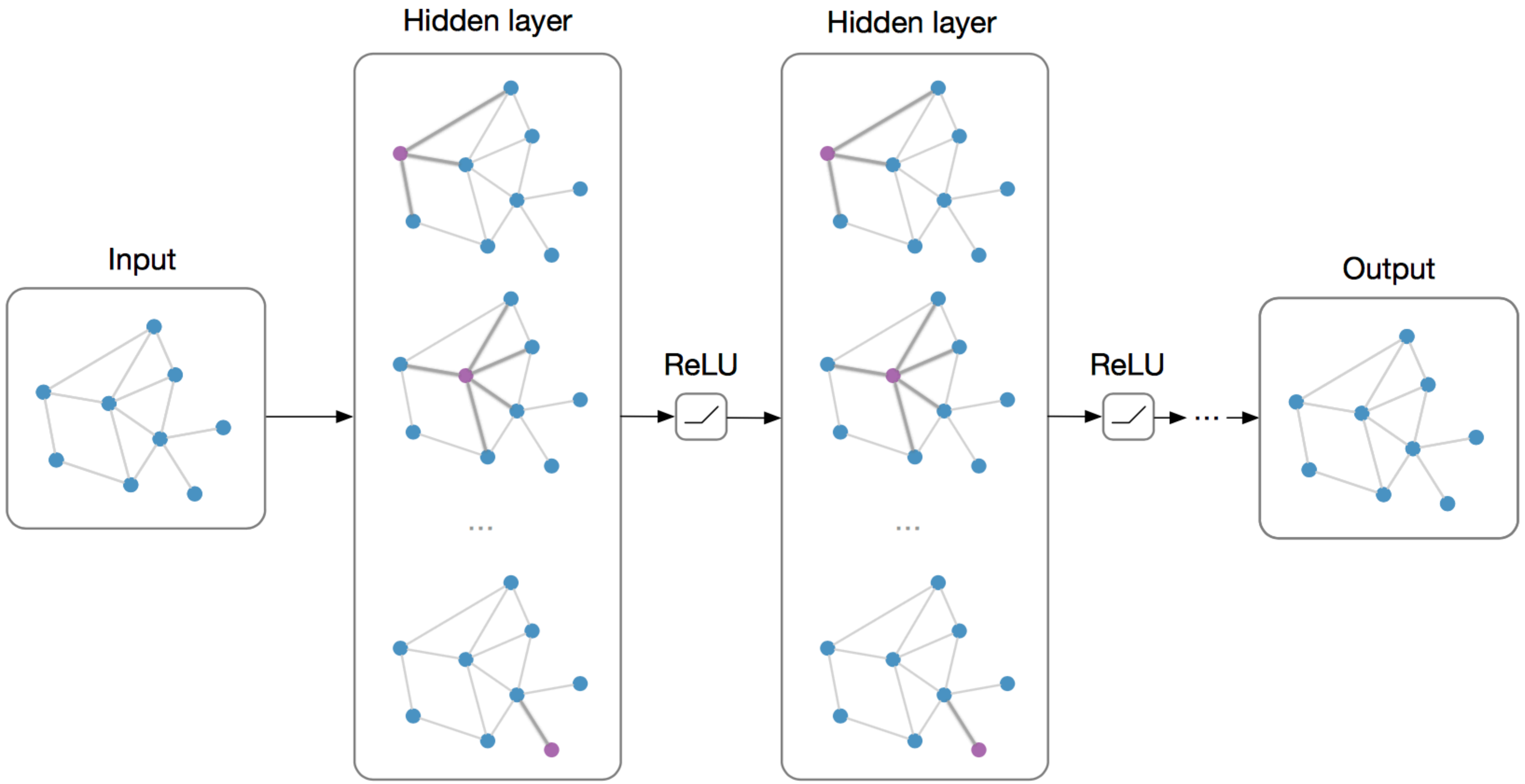
Challenge #2: **Varying** Neighborhood Sizes



Slide from Renjie Liao

Nearly all (if not all) Graph Neural Networks can be expressed as **Message Passing** paradigms

Graph Neural Networks (GNNs)



Main Idea: Pass massages between pairs of nodes and agglomerate

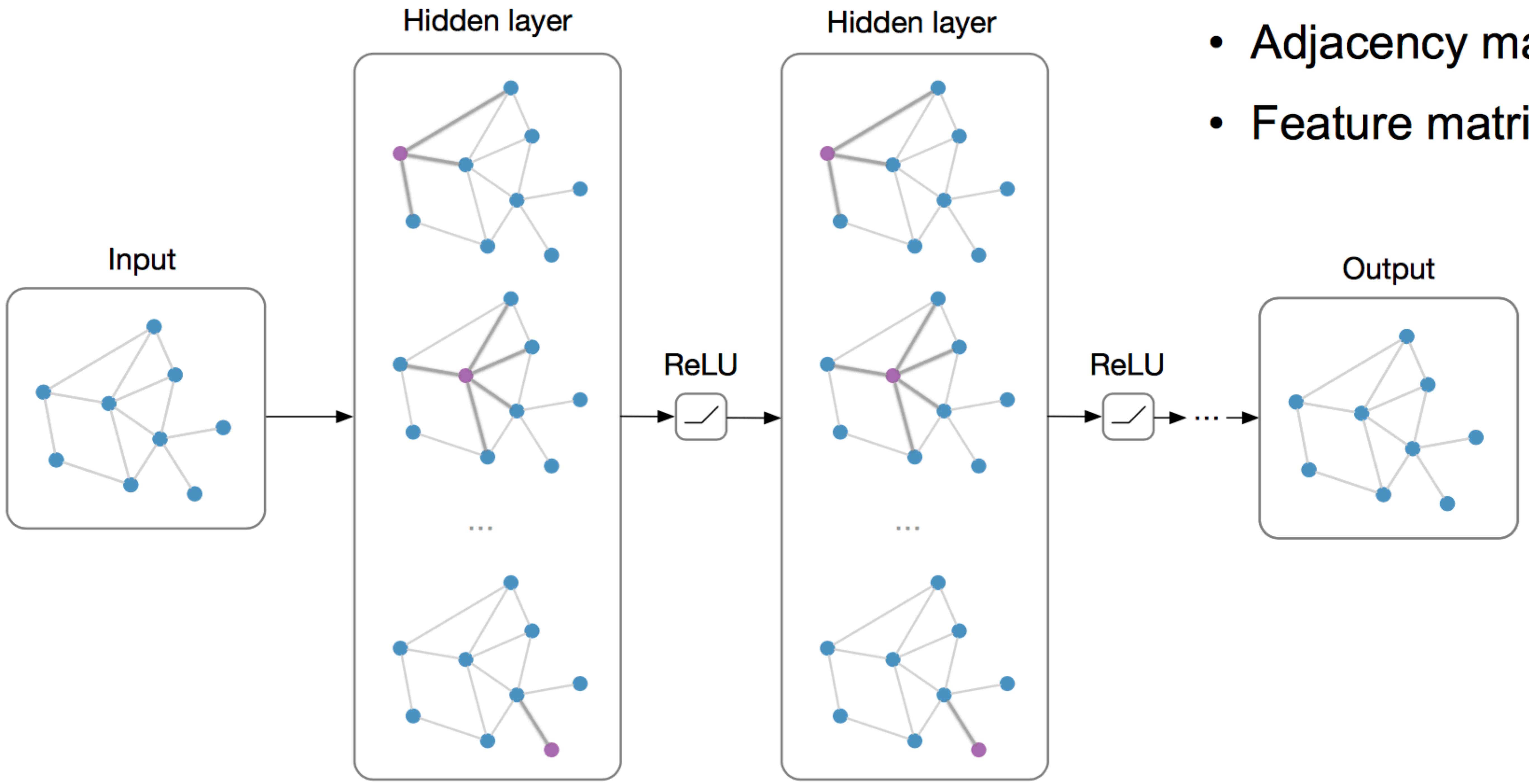
Alternative Interpretation: Pass massages between nodes to refine node (and possibly edge) representations

* slide from Thomas Kipf, **University of Amsterdam**

Graph Neural Networks (GNNs)

Notation: $\mathcal{G} = (\mathbf{A}, \mathbf{X})$

- Adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$
- Feature matrix $\mathbf{X} \in \mathbb{R}^{N \times F}$

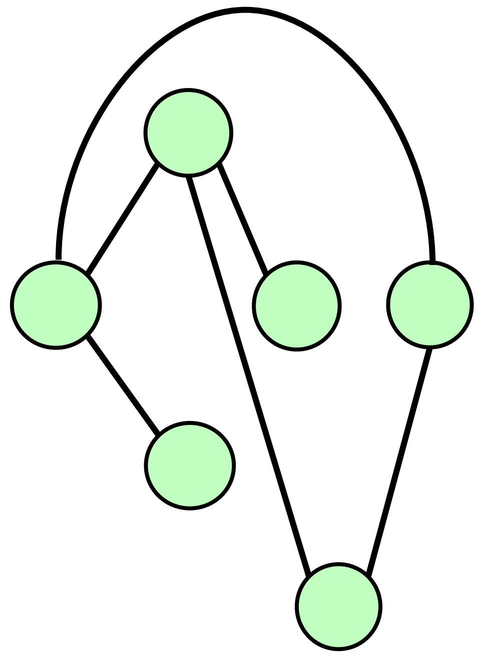


Main Idea: Pass massages between pairs of nodes and agglomerate

Alternative Interpretation: Pass massages between nodes to refine node (and possibly edge) representations

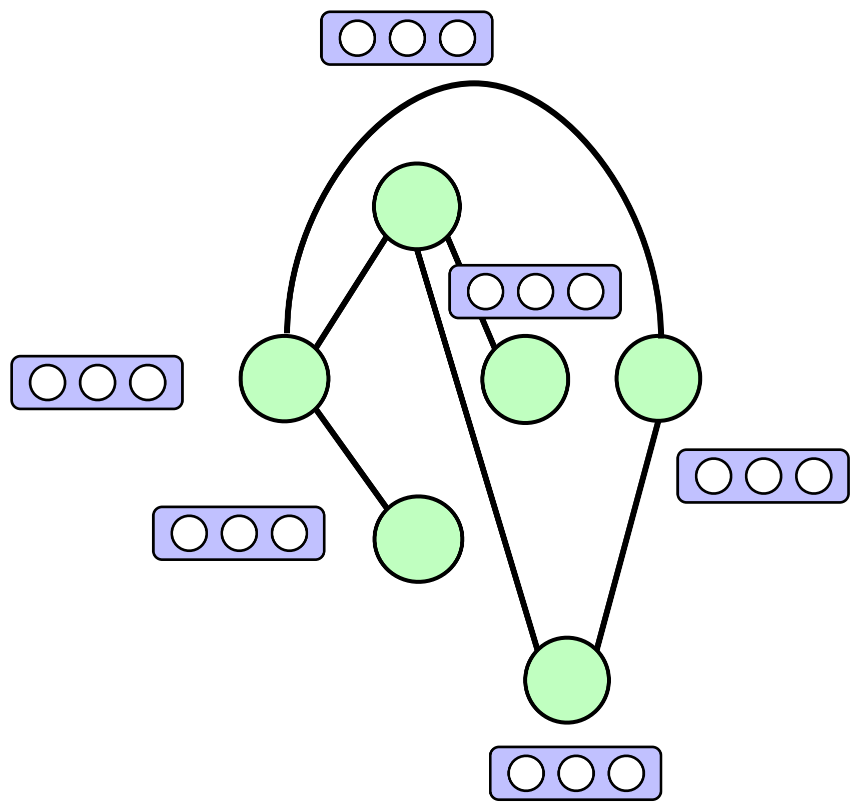
* slide from Thomas Kipf, **University of Amsterdam**

Graph Neural Networks (GNNs)



Slide from Renjie Liao

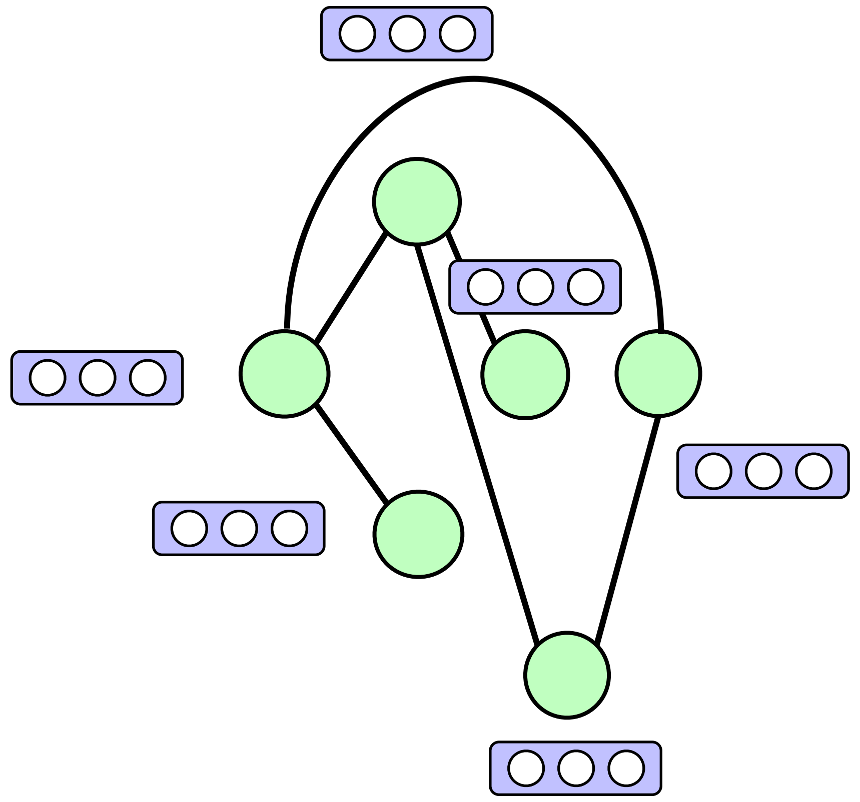
Graph Neural Networks (GNNs)



Input Encoding

Slide from Renjie Liao

Graph Neural Networks (GNNs)



Input Encoding

1. Node Feature

   — *If it is unavailable, use 1-of-K, random, index encoding of node*

2. Edge Feature

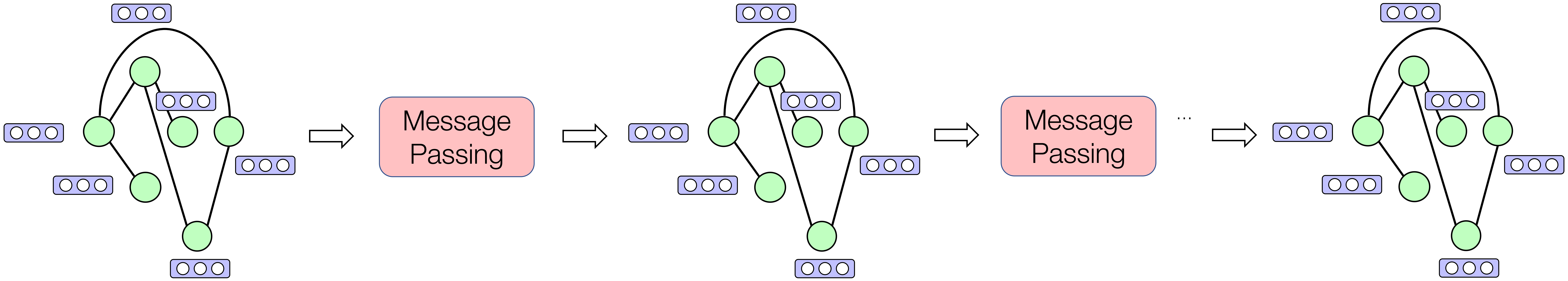
   — *Feed it to message network*

3. Graph Feature

   — *Treat it as a super node in your graph*

   — *Feed graph feature to readout layer*

Graph Neural Networks (GNNs)

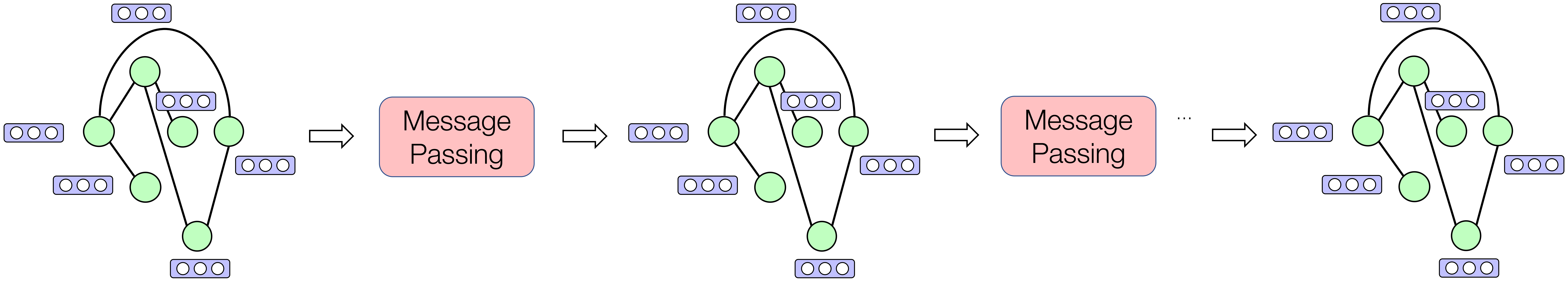


Input Encoding

Message Passing Layers/Steps

Slide from Renjie Liao

Graph Neural Networks (GNNs)



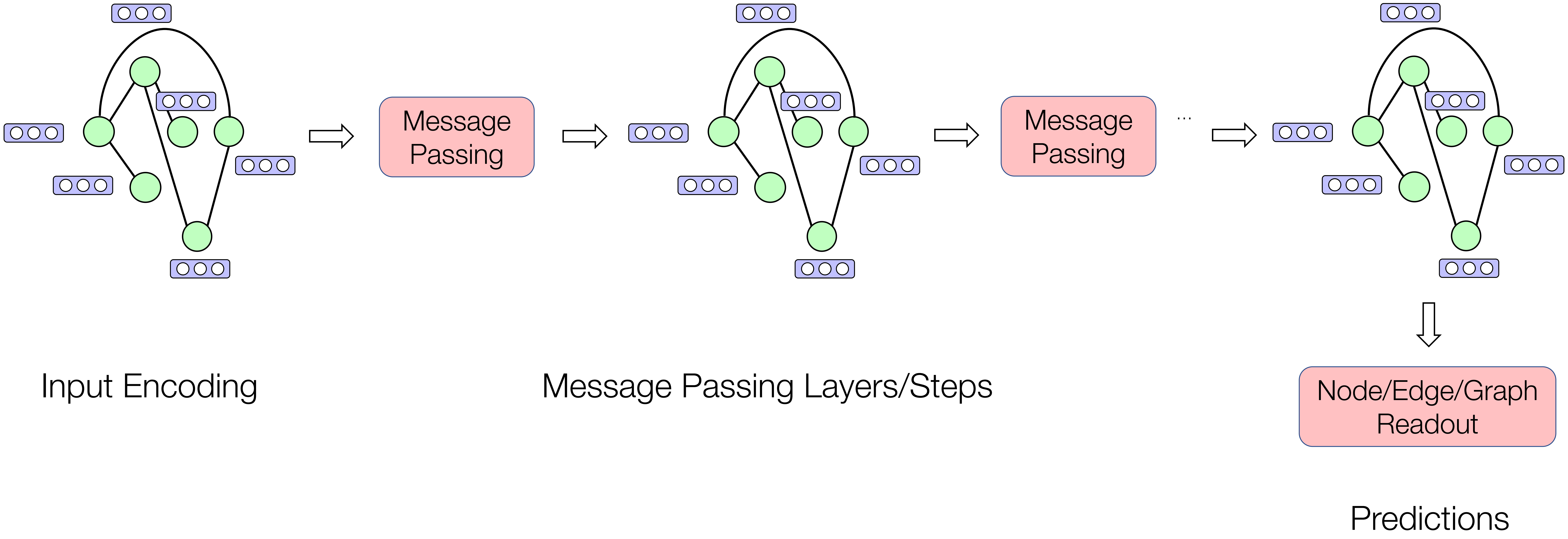
Input Encoding

Message Passing Layers/Steps

Steps: share message passing module (Recurrent Networks)
Layers: do not share message passing module (Feedforward Networks)

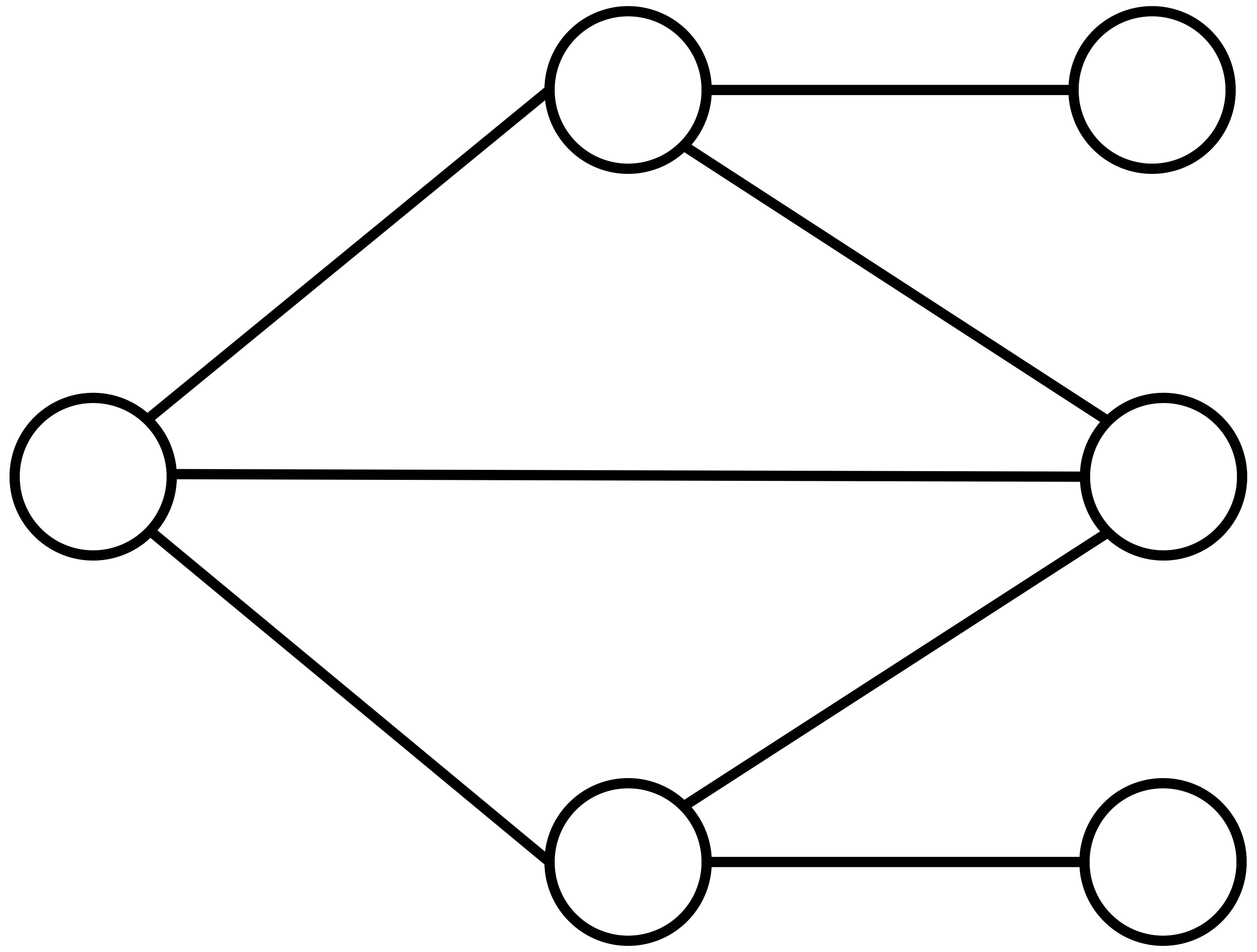
Slide from Renjie Liao

Graph Neural Networks (GNNs)



Message Passing in GNNs

(t+1)-th message passing step/layer

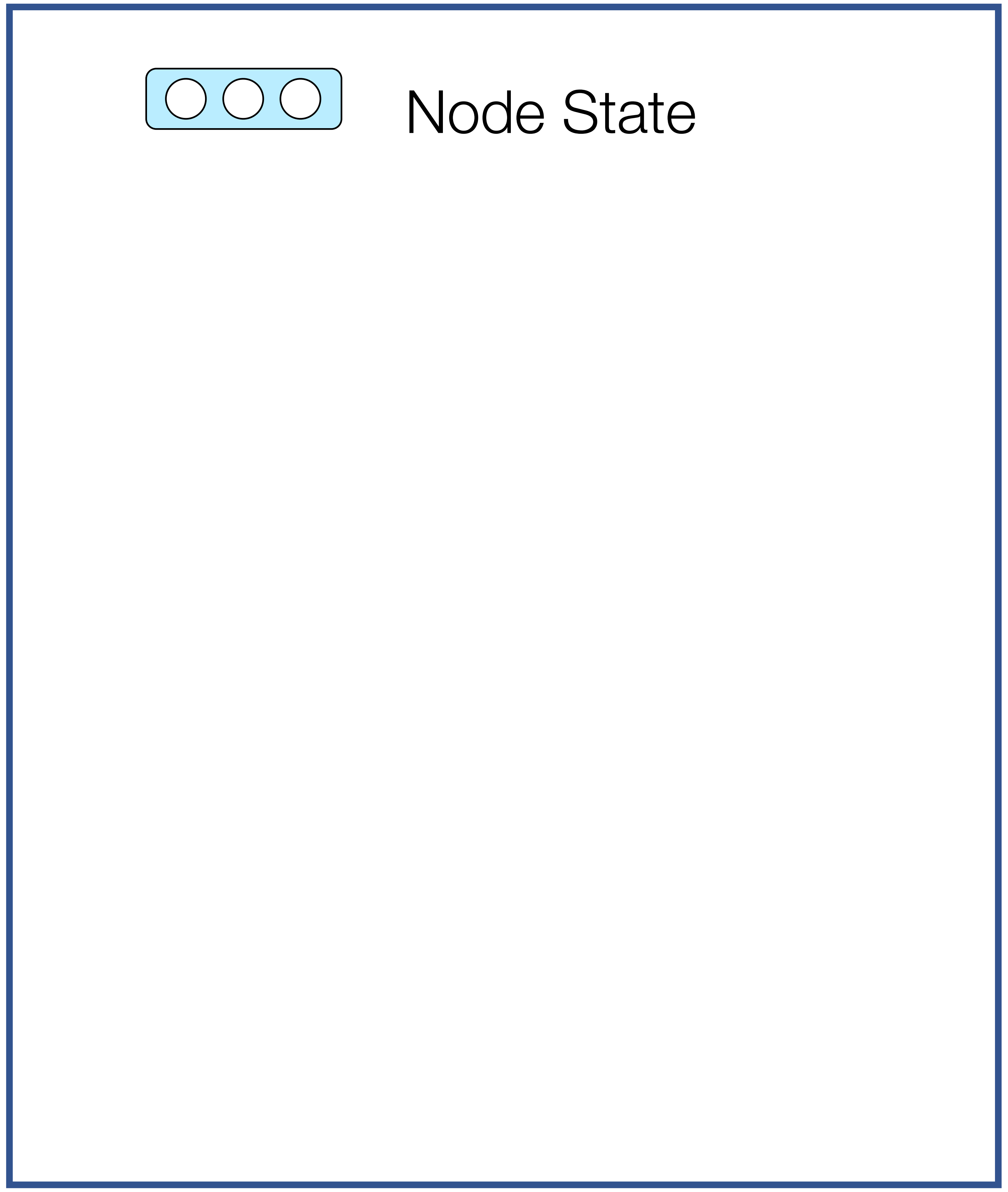


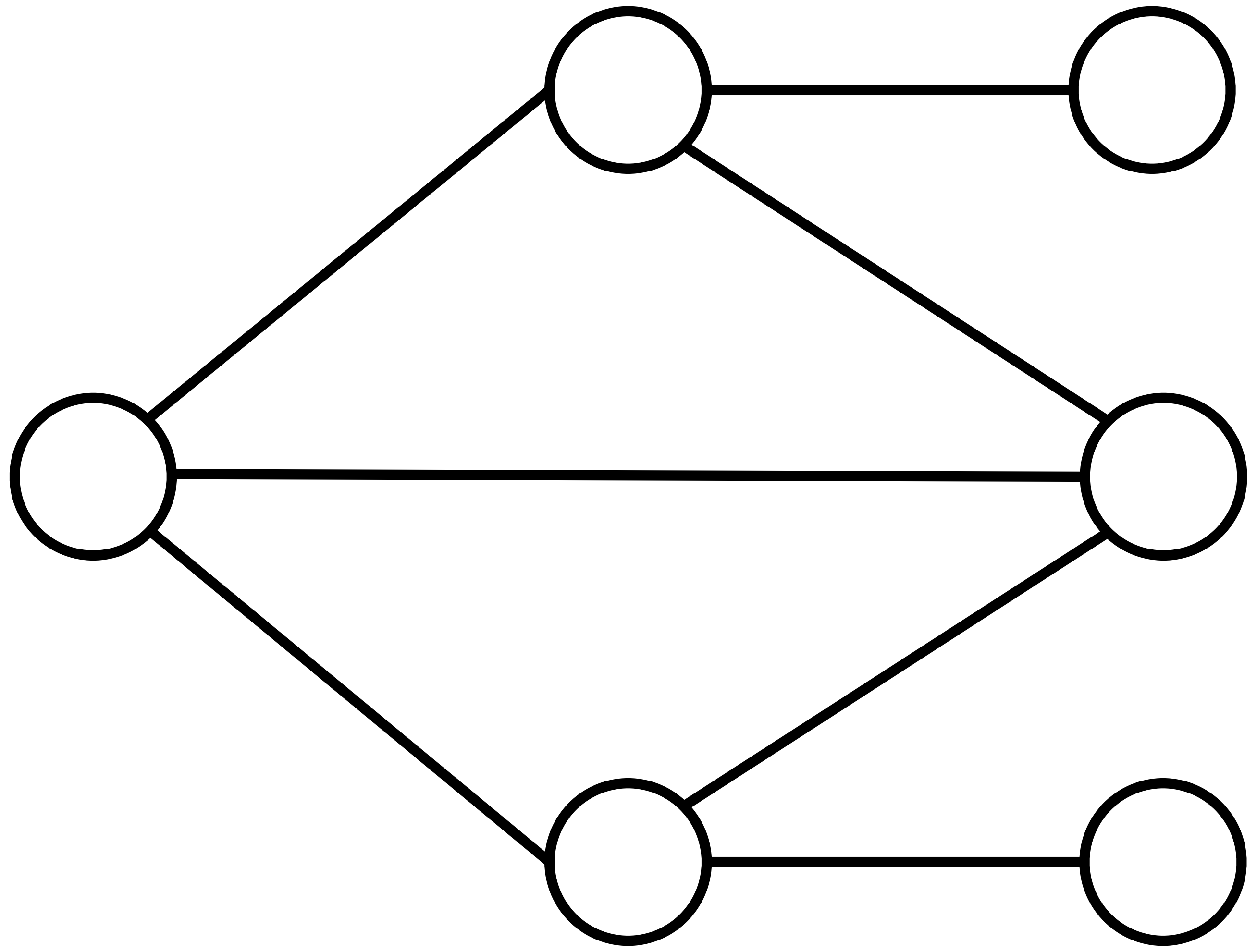
Slide from Renjie Liao

Message Passing in GNNs

(t+1)-th message passing step/layer

$\mathbf{h}_i^t$



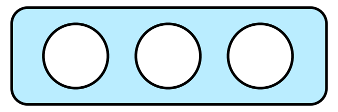


Slide from Renjie Liao

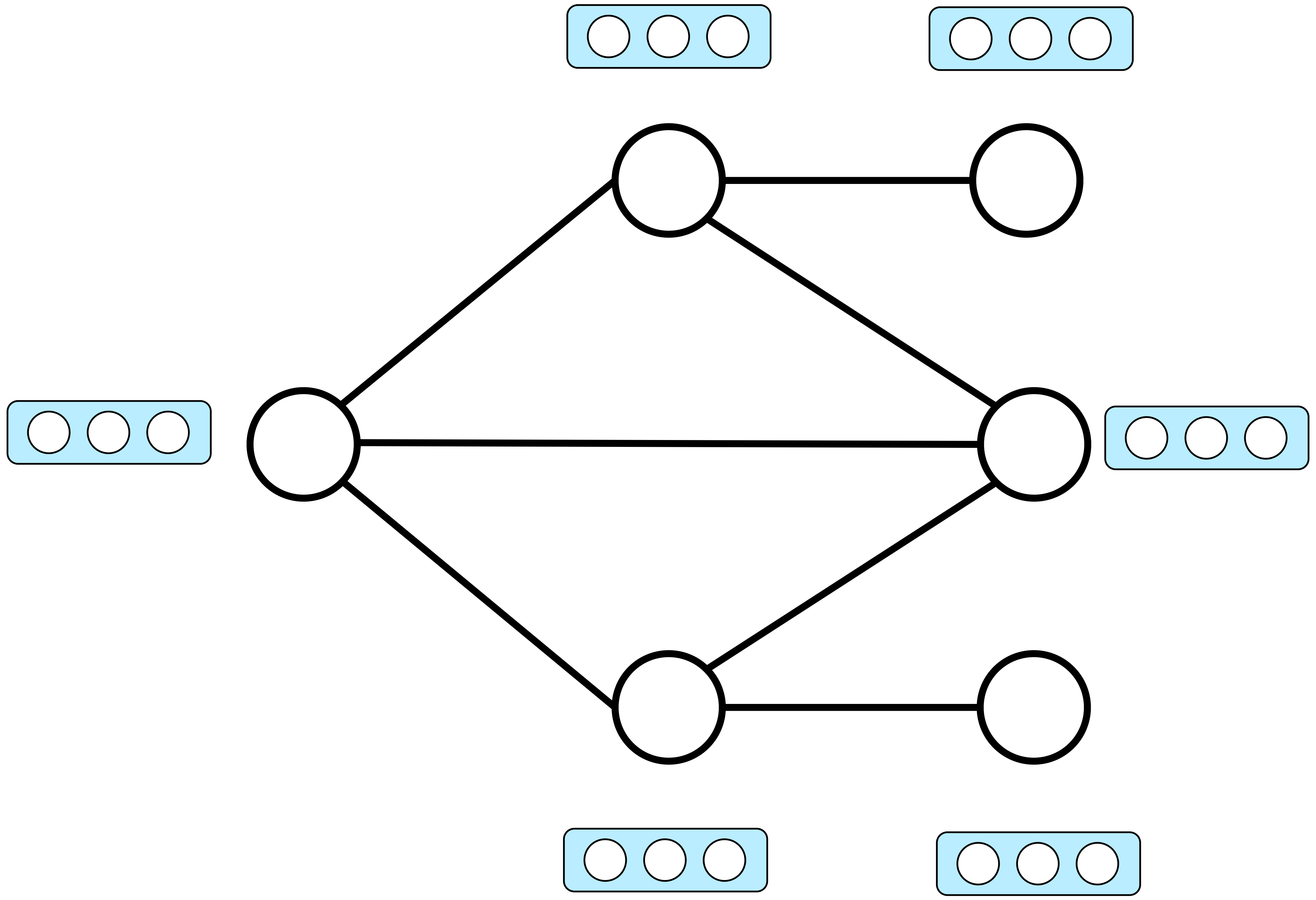
Message Passing in GNNs

(t+1)-th message passing step/layer

$\mathbf{h}_i^t$



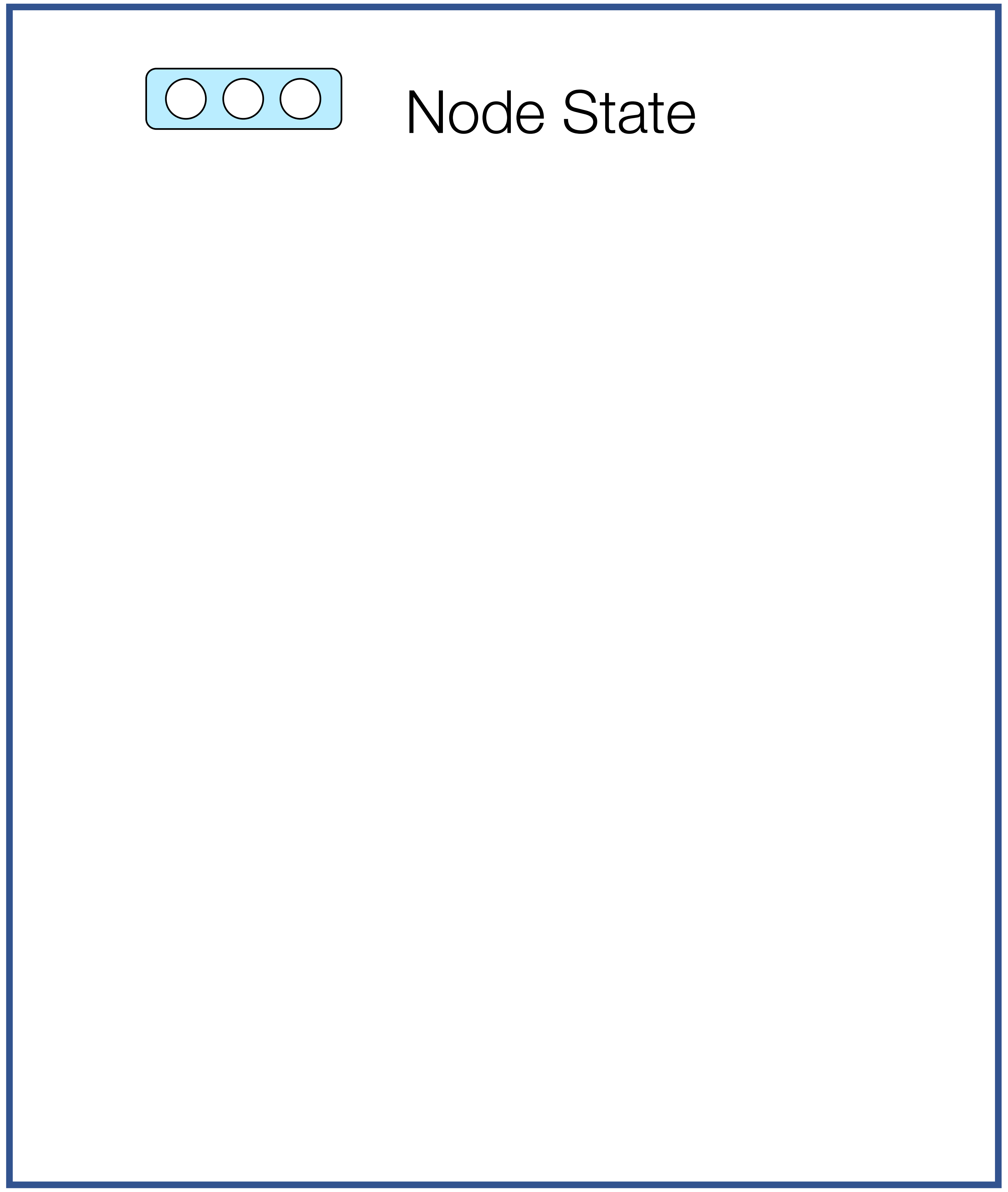
Node State

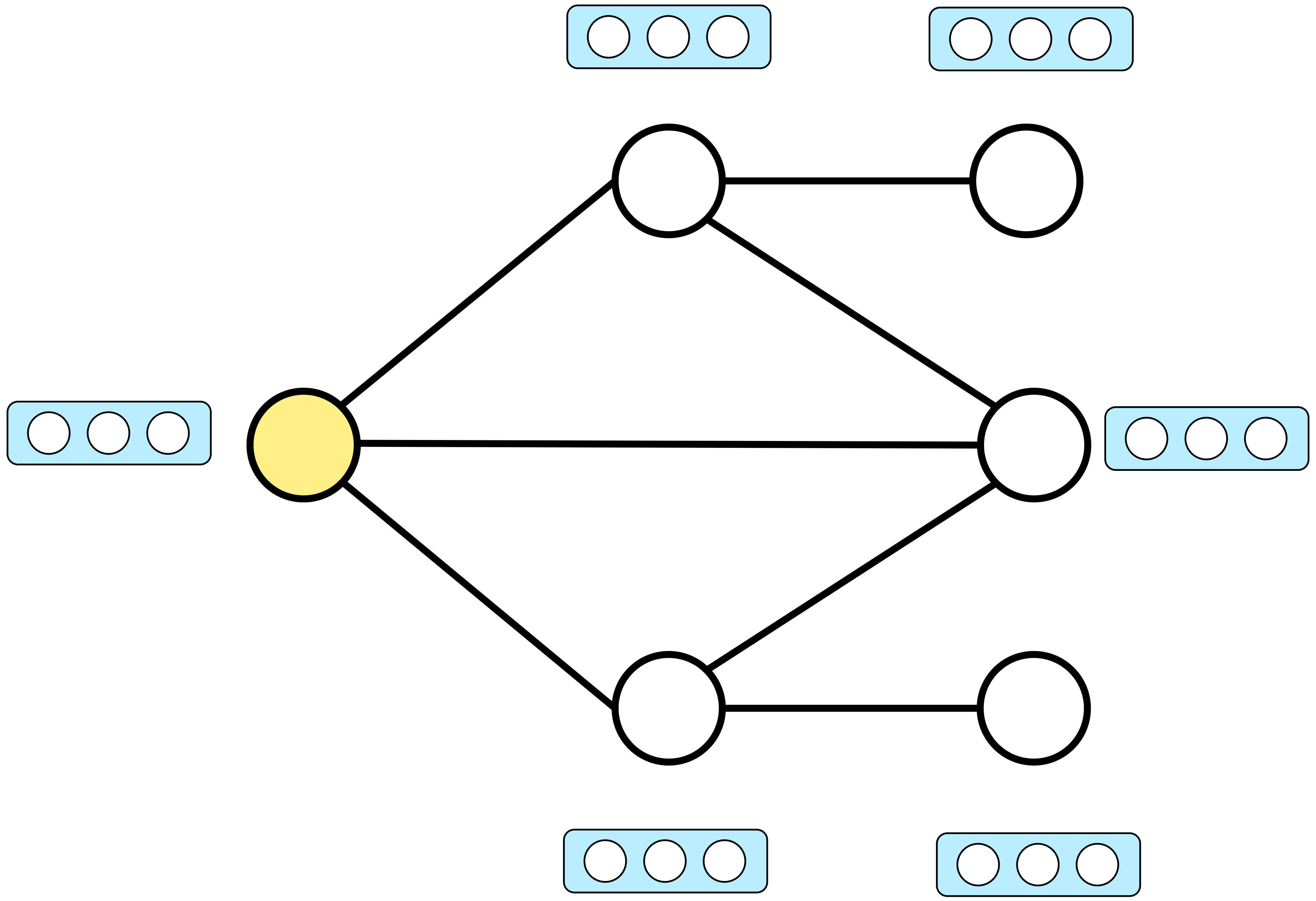


Message Passing in GNNs

(t+1)-th message passing step/layer

$\mathbf{h}_i^t$



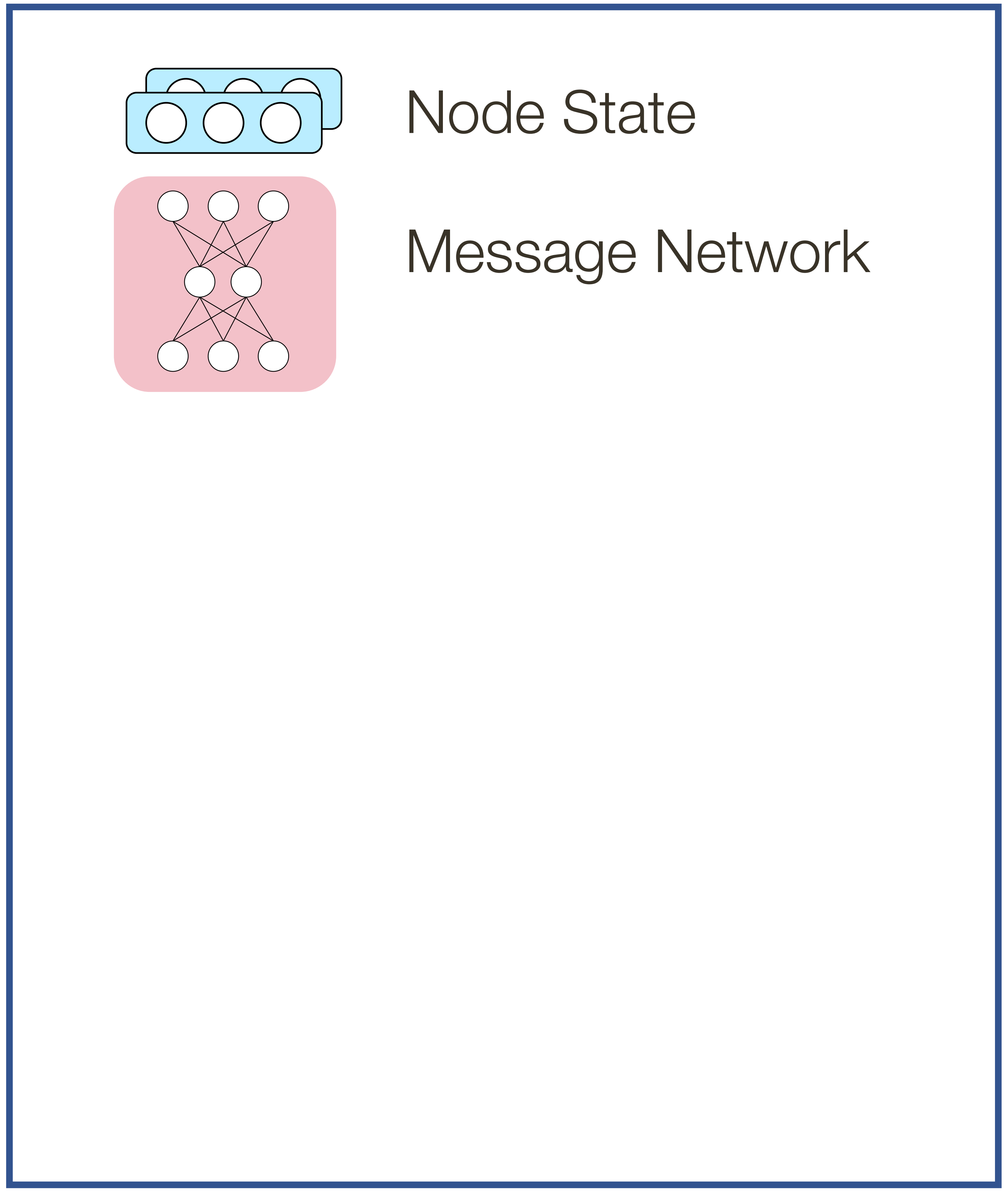


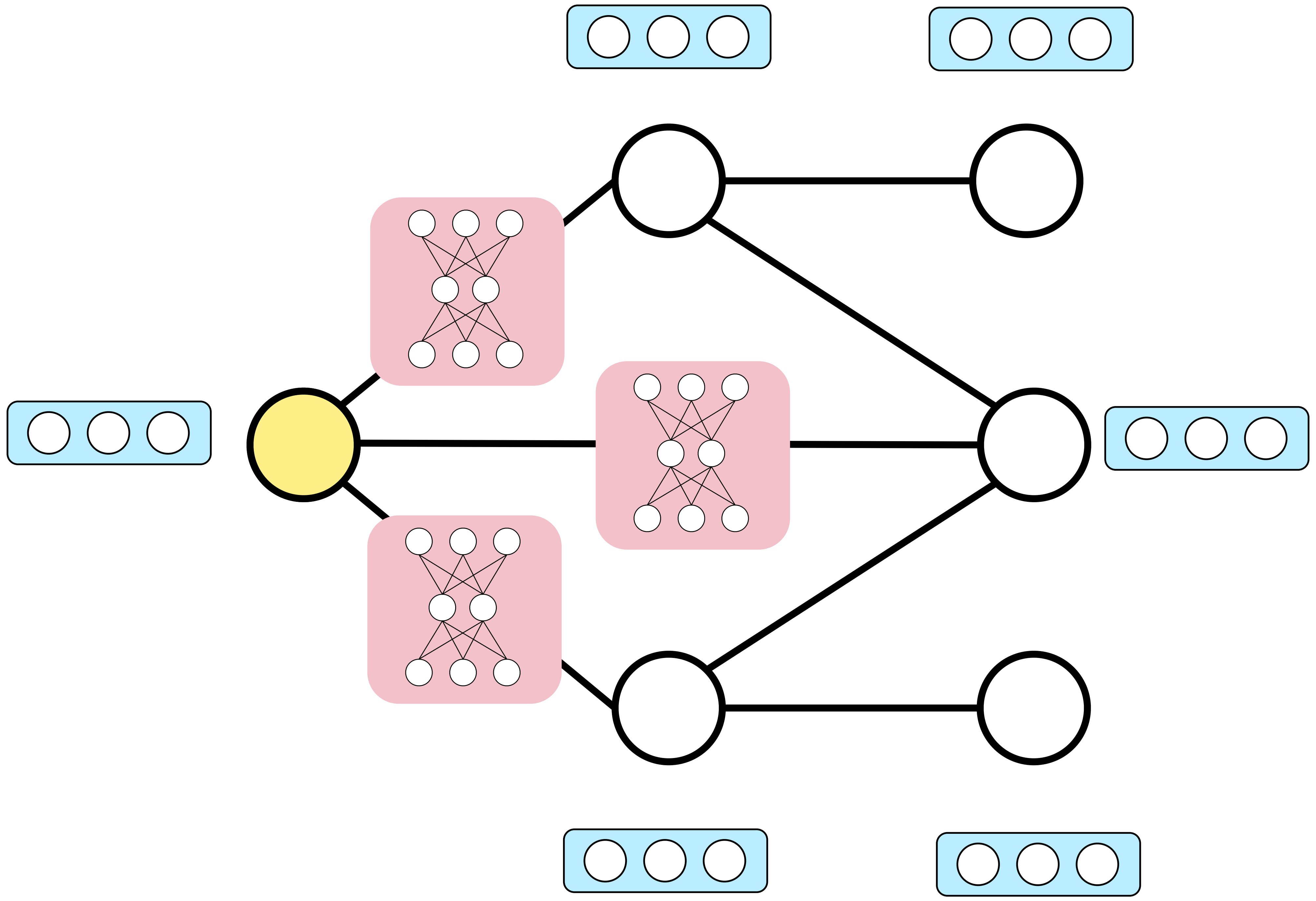
Slide from Renjie Liao

Message Passing in GNNs

(t+1)-th message passing step/layer

$\mathbf{h}_i^t \quad \mathbf{h}_j^t$





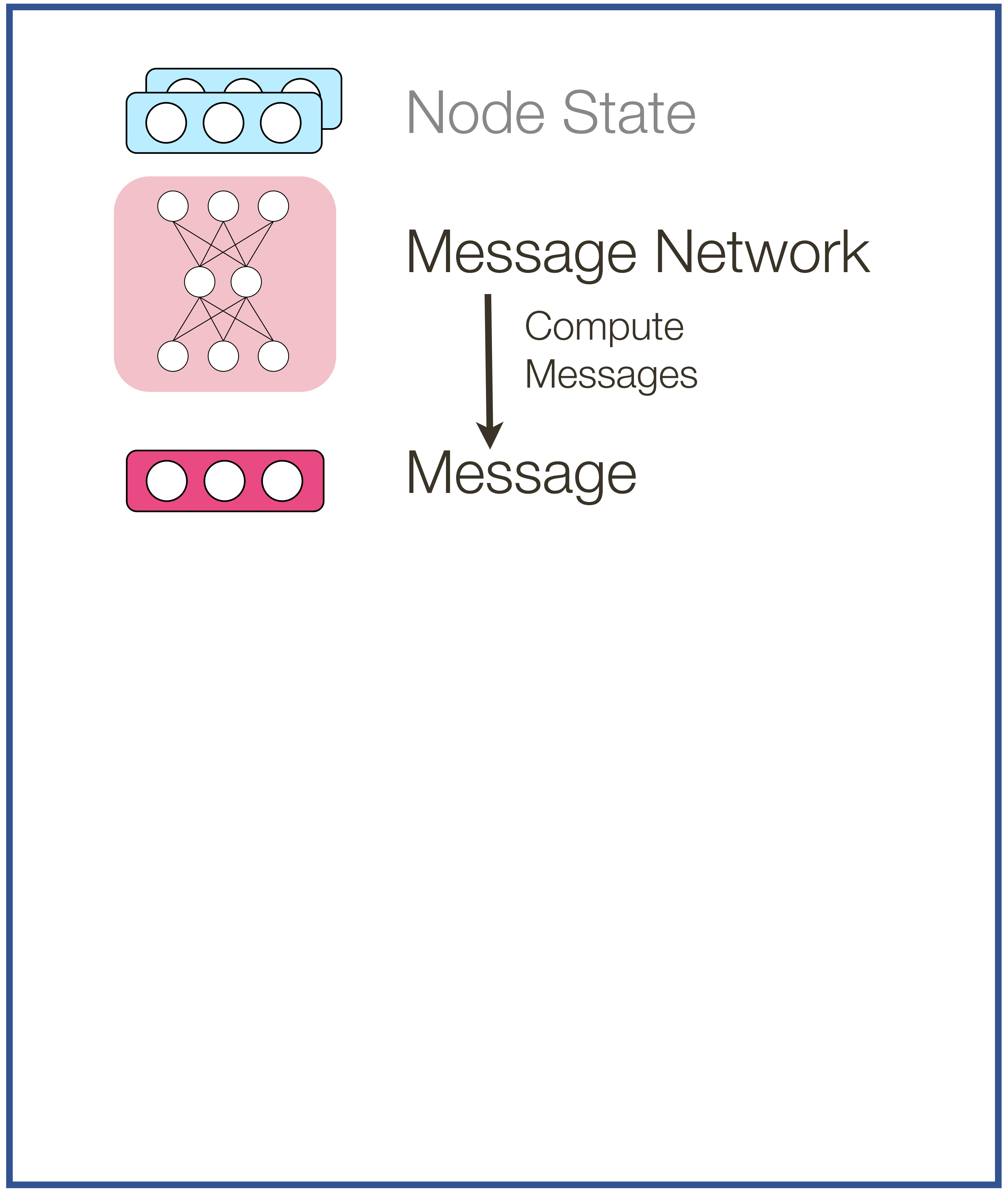
Slide from Renjie Liao

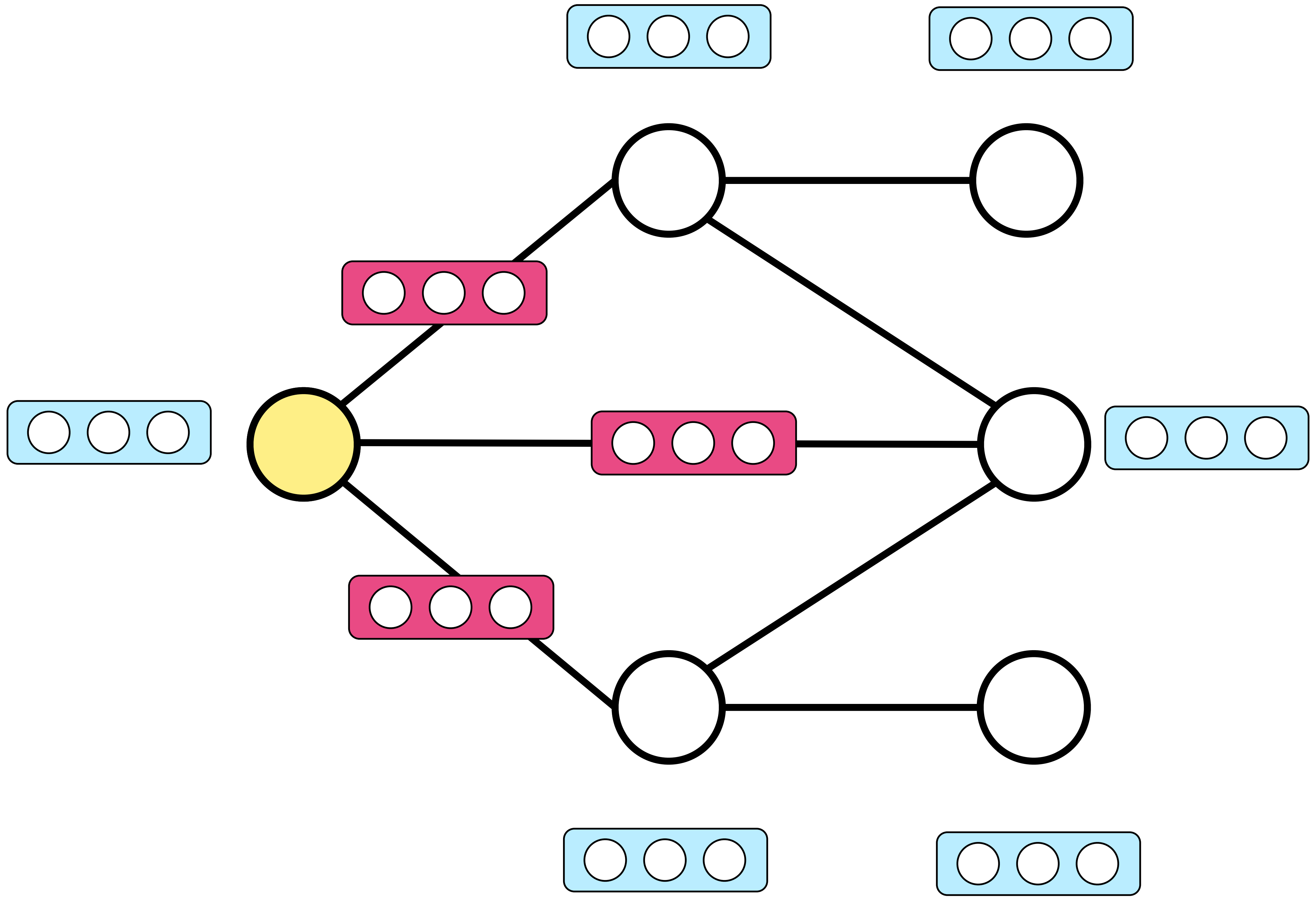
Message Passing in GNNs

(t+1)-th message passing step/layer

$\mathbf{h}_i^t \quad \mathbf{h}_j^t$

$$\mathbf{m}_{ji}^t = f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t)$$



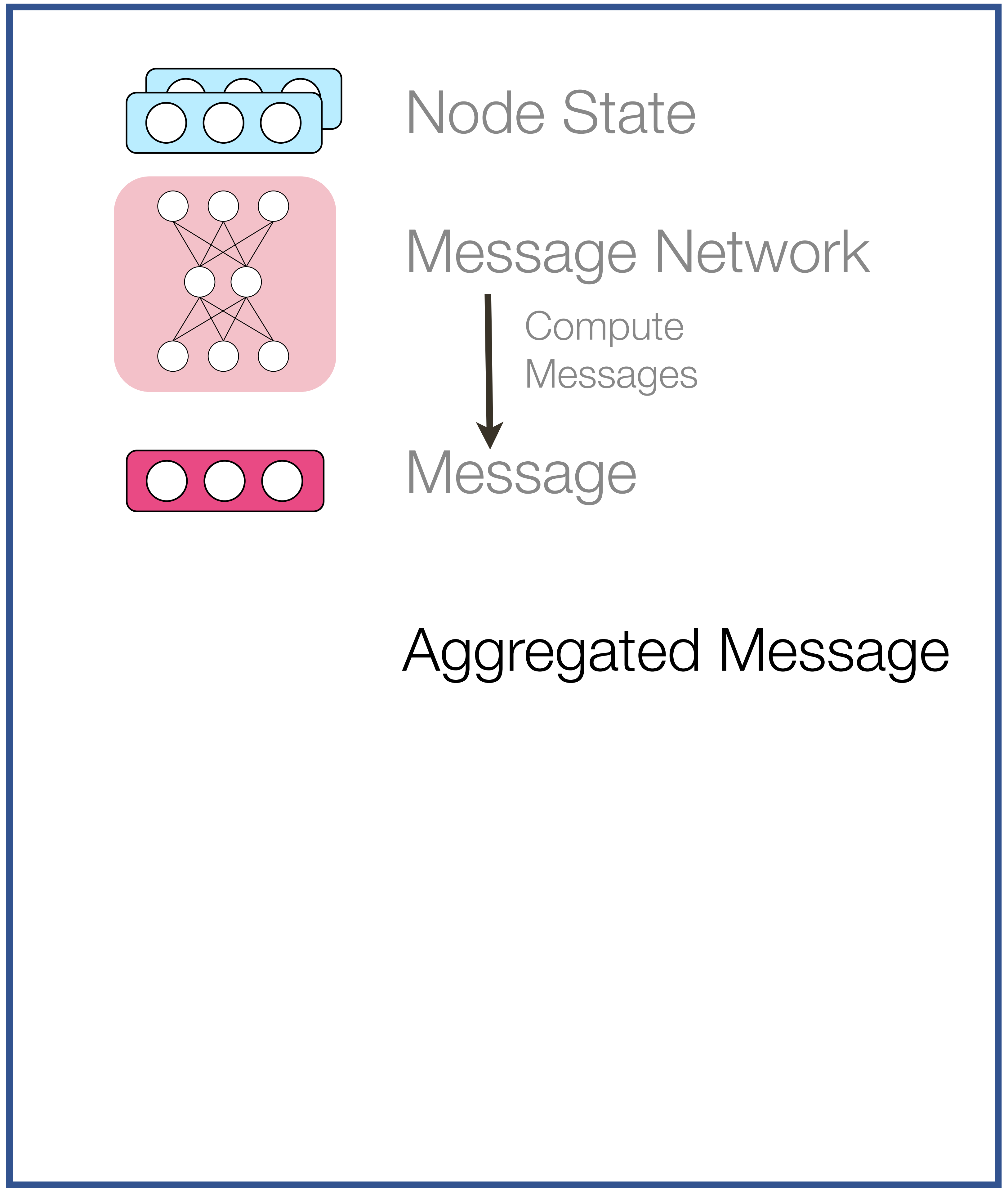


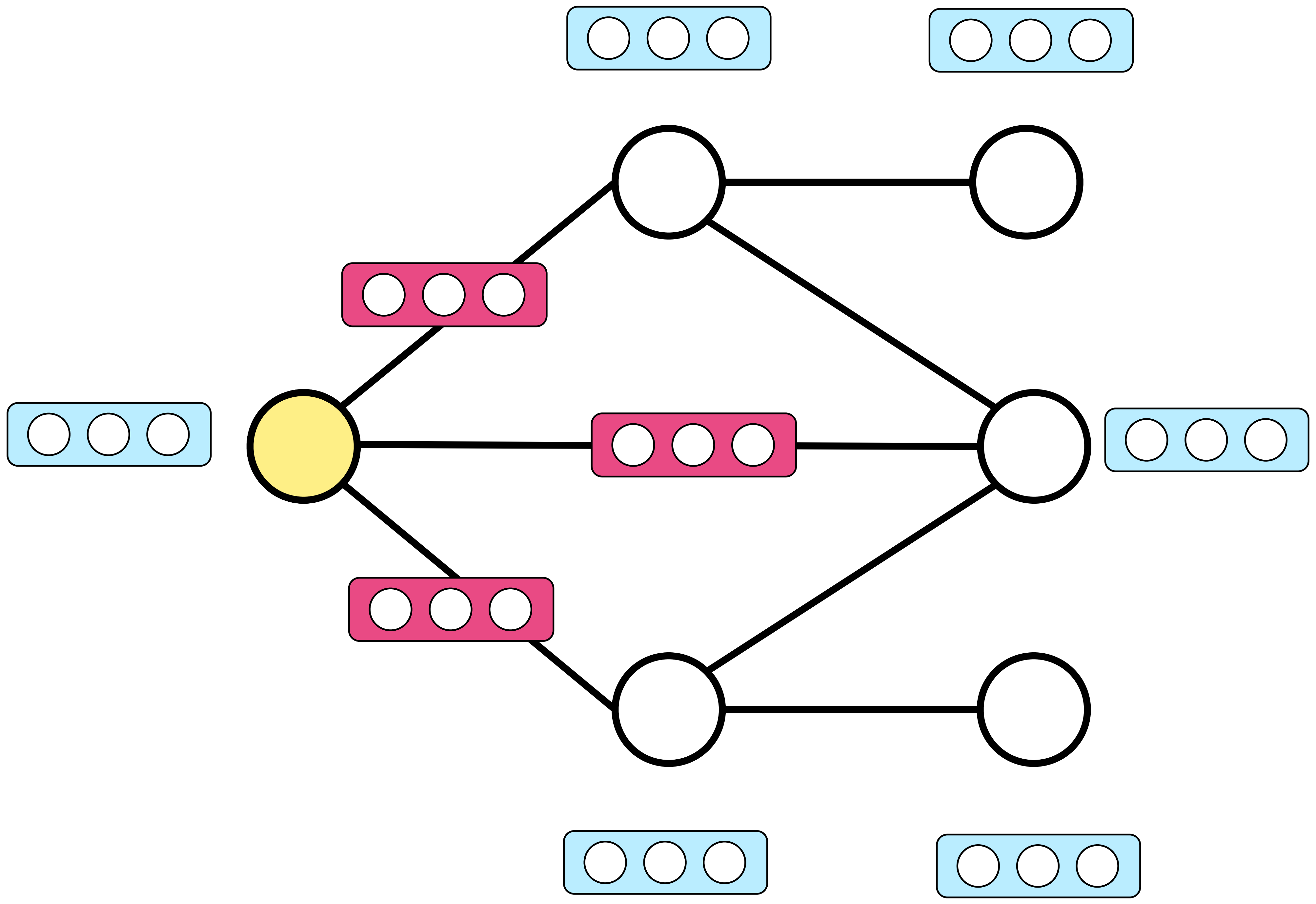
Message Passing in GNNs

(t+1)-th message passing step/layer

$\mathbf{h}_i^t \quad \mathbf{h}_j^t$

$$\mathbf{m}_{ji}^t = f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t)$$



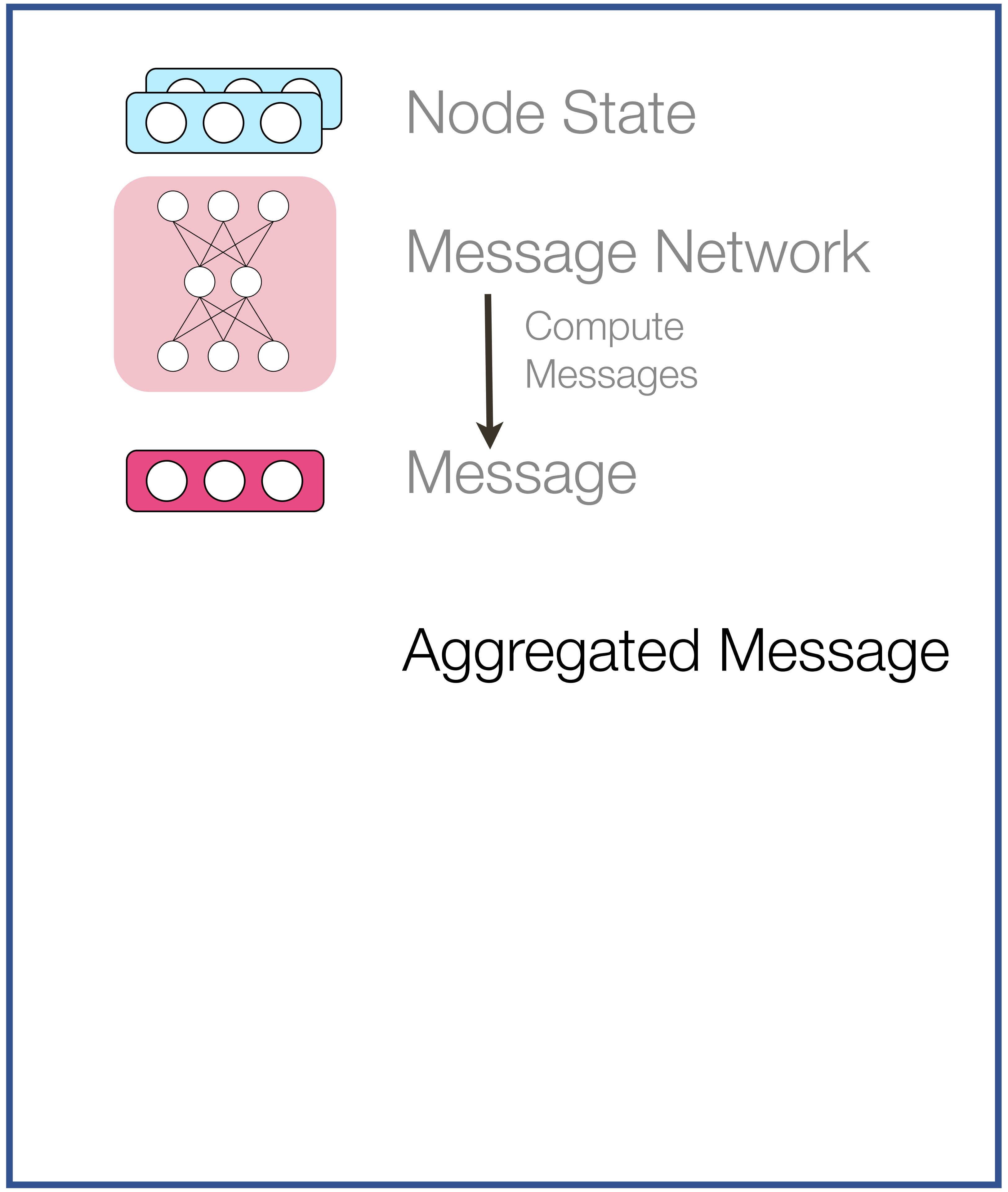


Slide from Renjie Liao

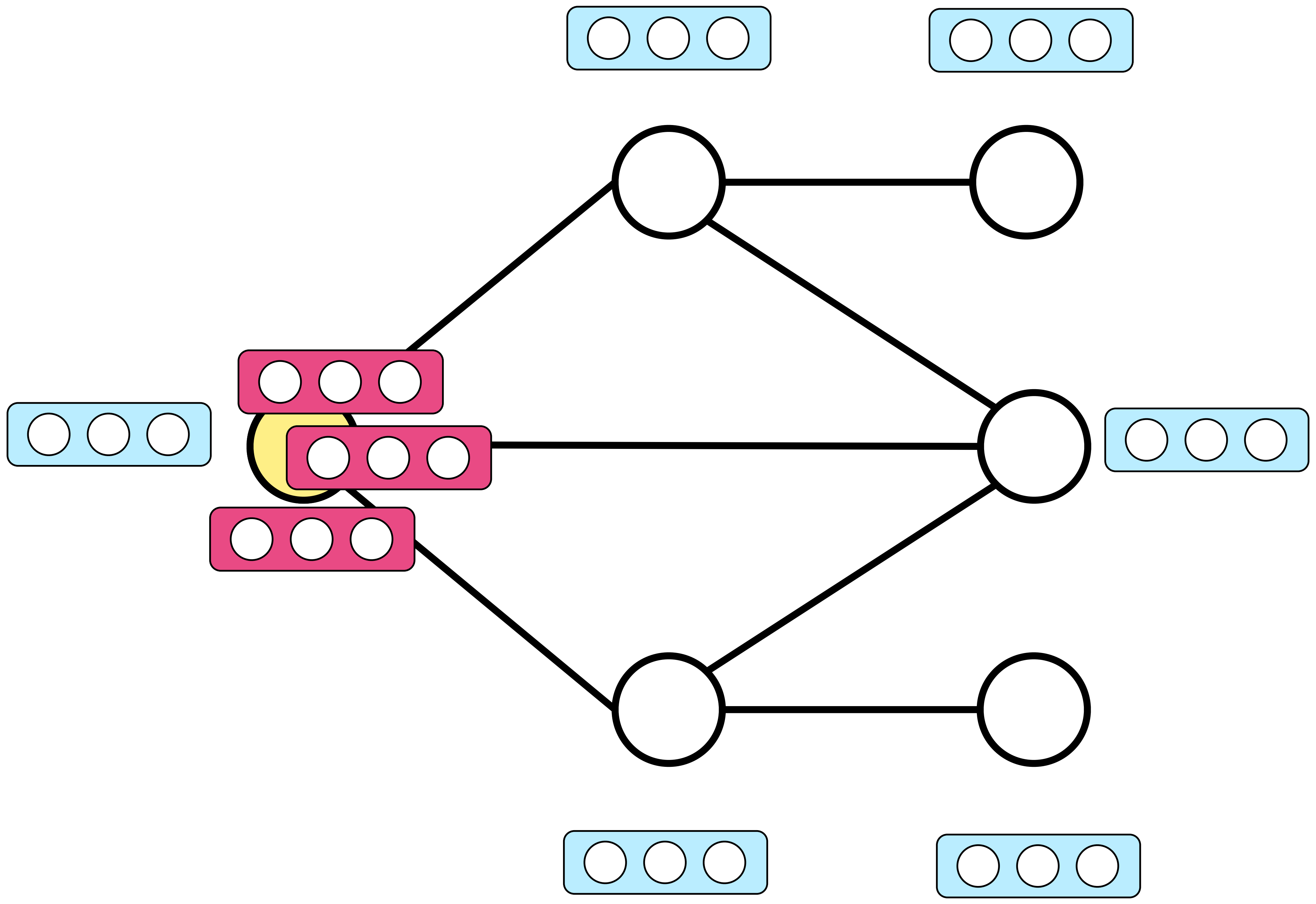
Message Passing in GNNs

$\mathbf{h}_i^t \quad \mathbf{h}_j^t$

$$\mathbf{m}_{ji}^t = f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t)$$



(t+1)-th message passing step/layer



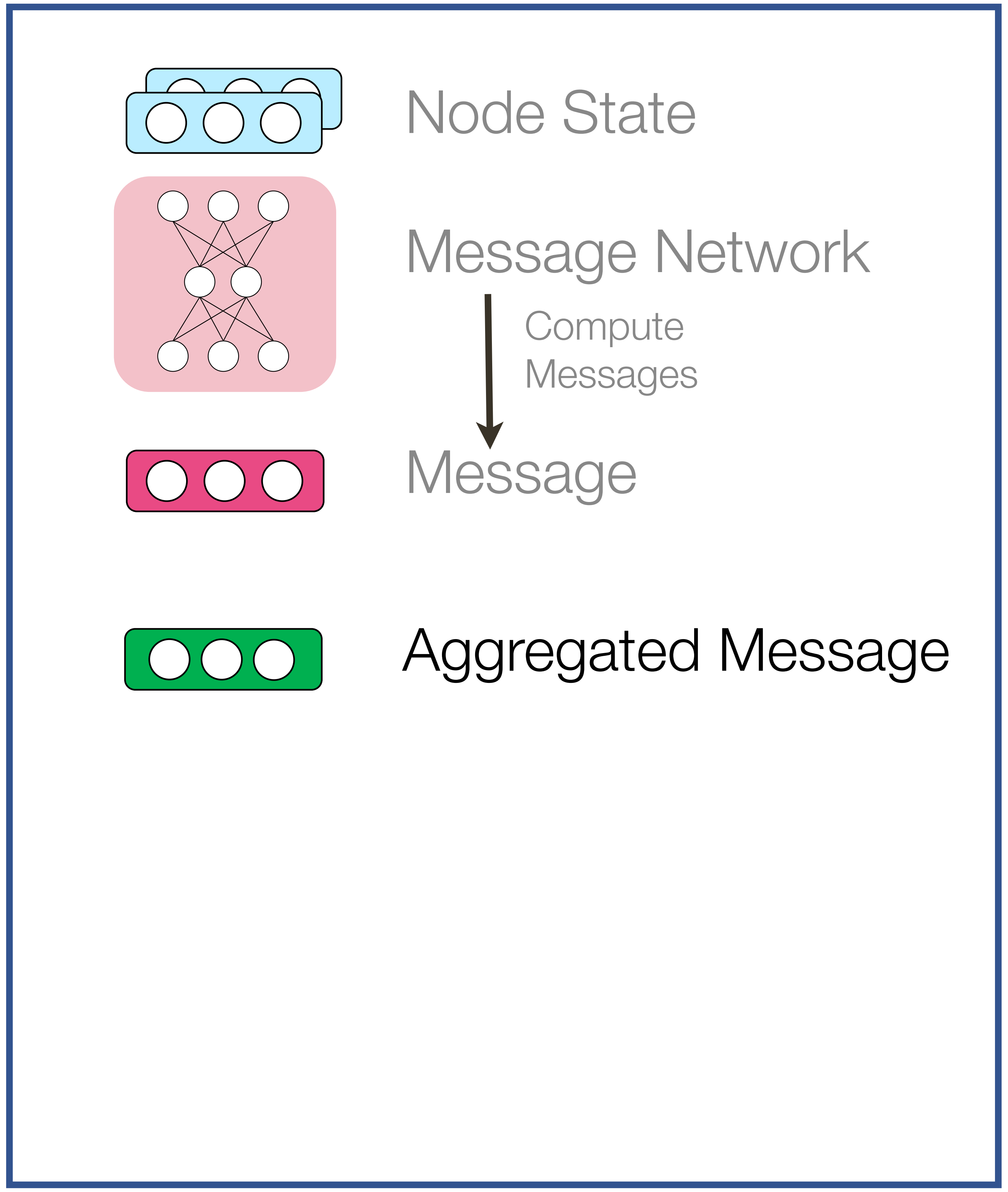
Message Passing in GNNs

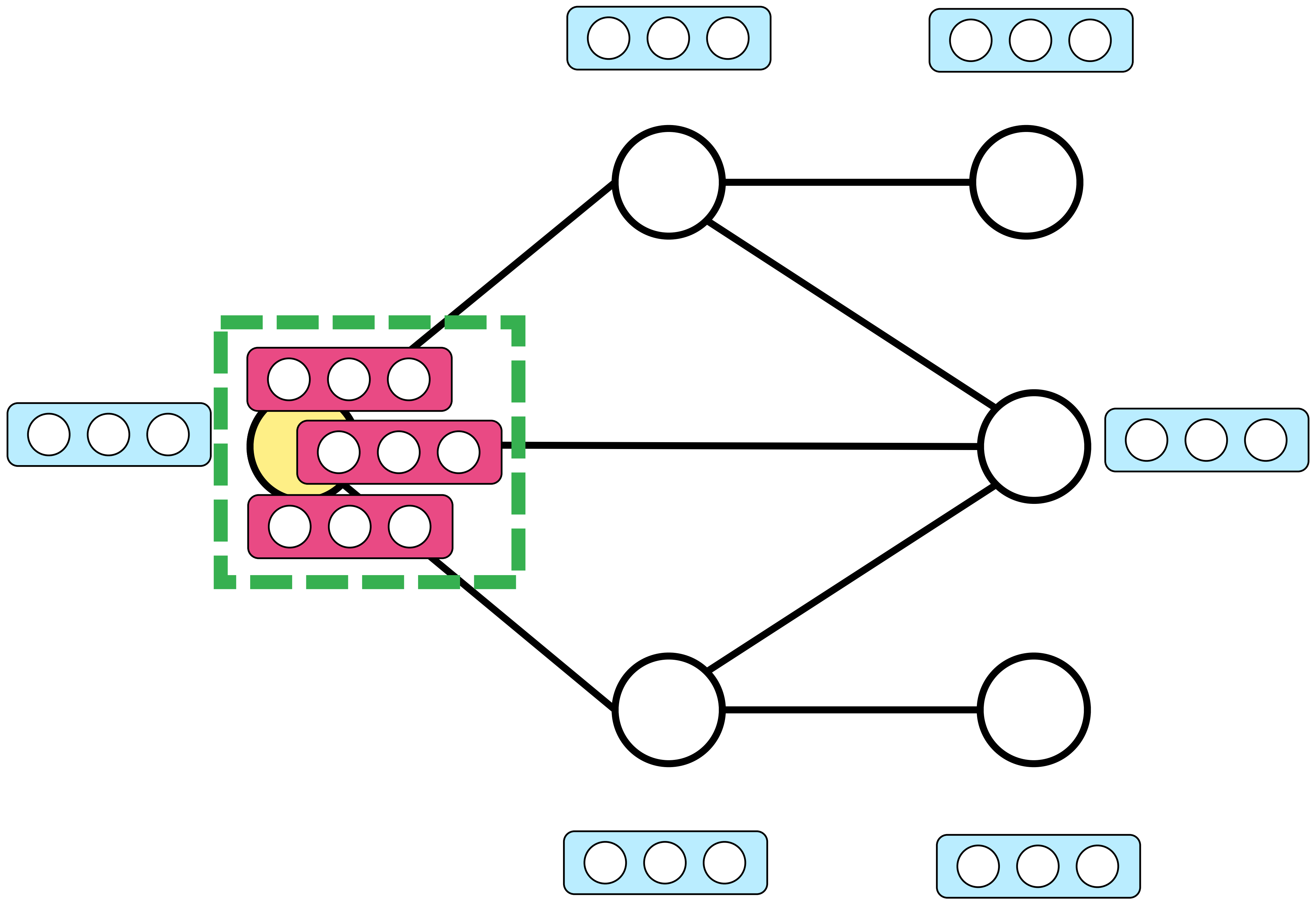
(t+1)-th message passing step/layer

$$\mathbf{h}_i^t \quad \mathbf{h}_j^t$$

$$\mathbf{m}_{ji}^t = f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t)$$

$$\bar{\mathbf{m}}_i^t = f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right)$$





Slide from Renjie Liao

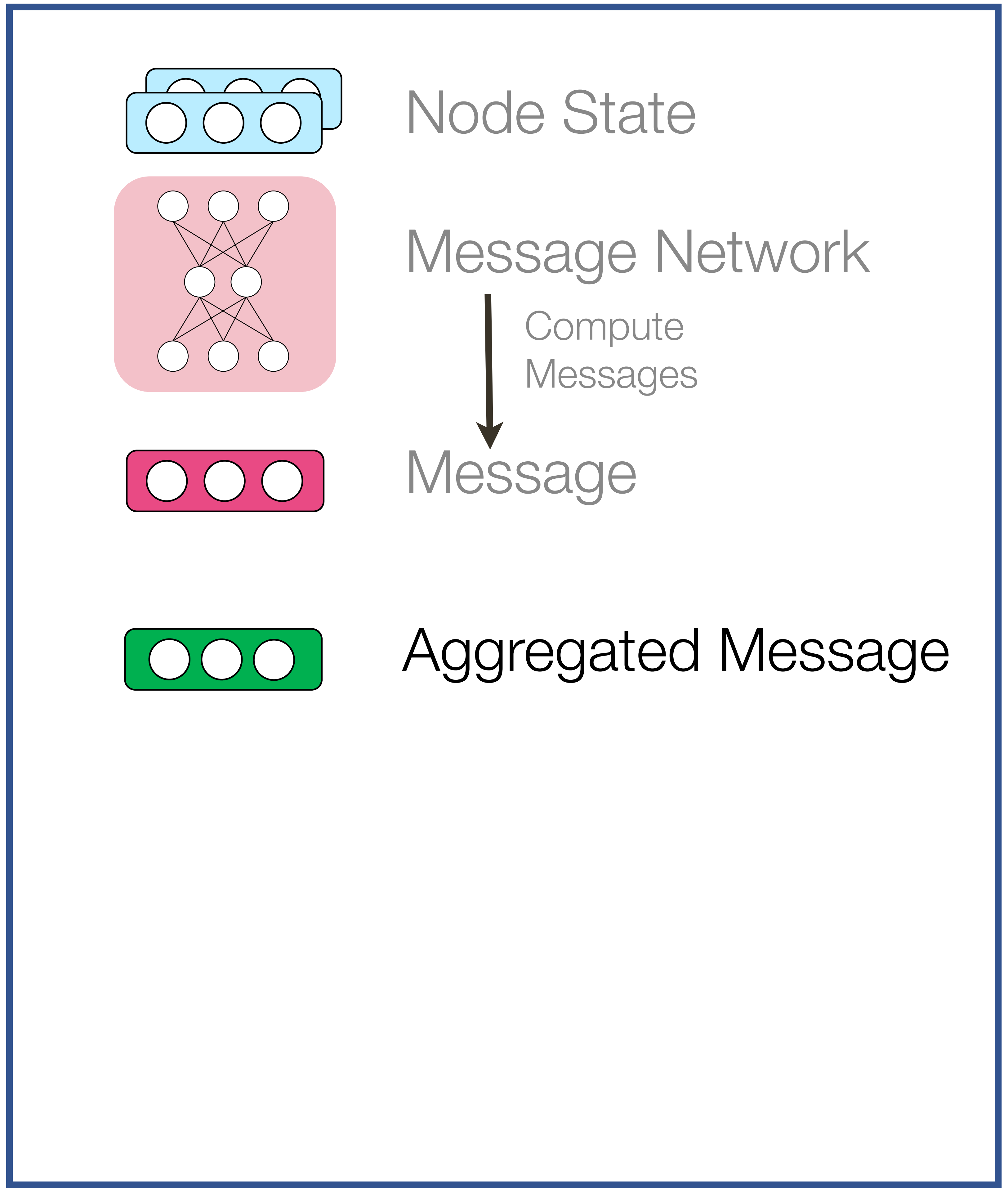
Message Passing in GNNs

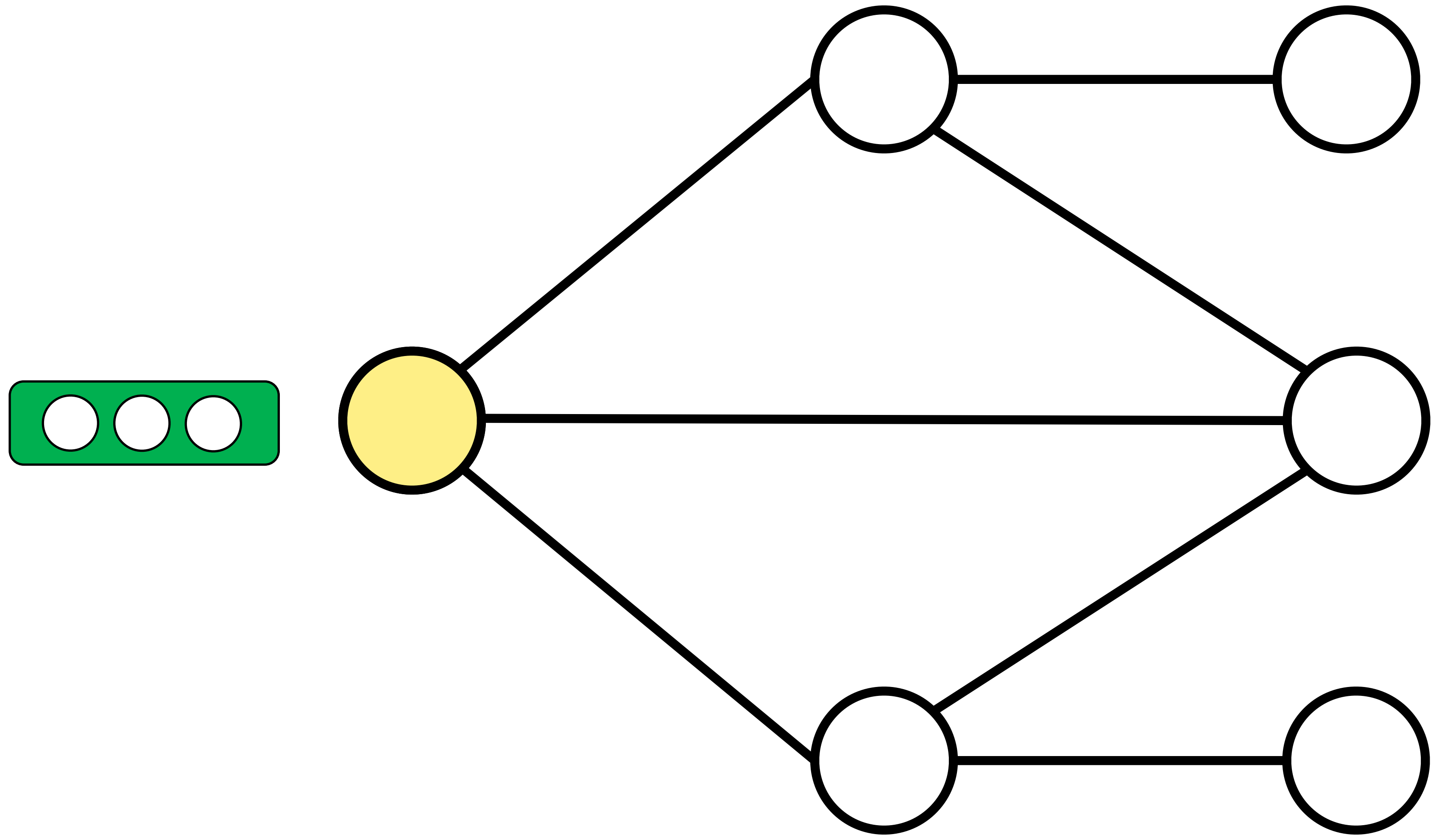
(t+1)-th message passing step/layer

$\mathbf{h}_i^t \quad \mathbf{h}_j^t$

$$\mathbf{m}_{ji}^t = f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t)$$

$$\bar{\mathbf{m}}_i^t = f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right)$$





Slide from Renjie Liao

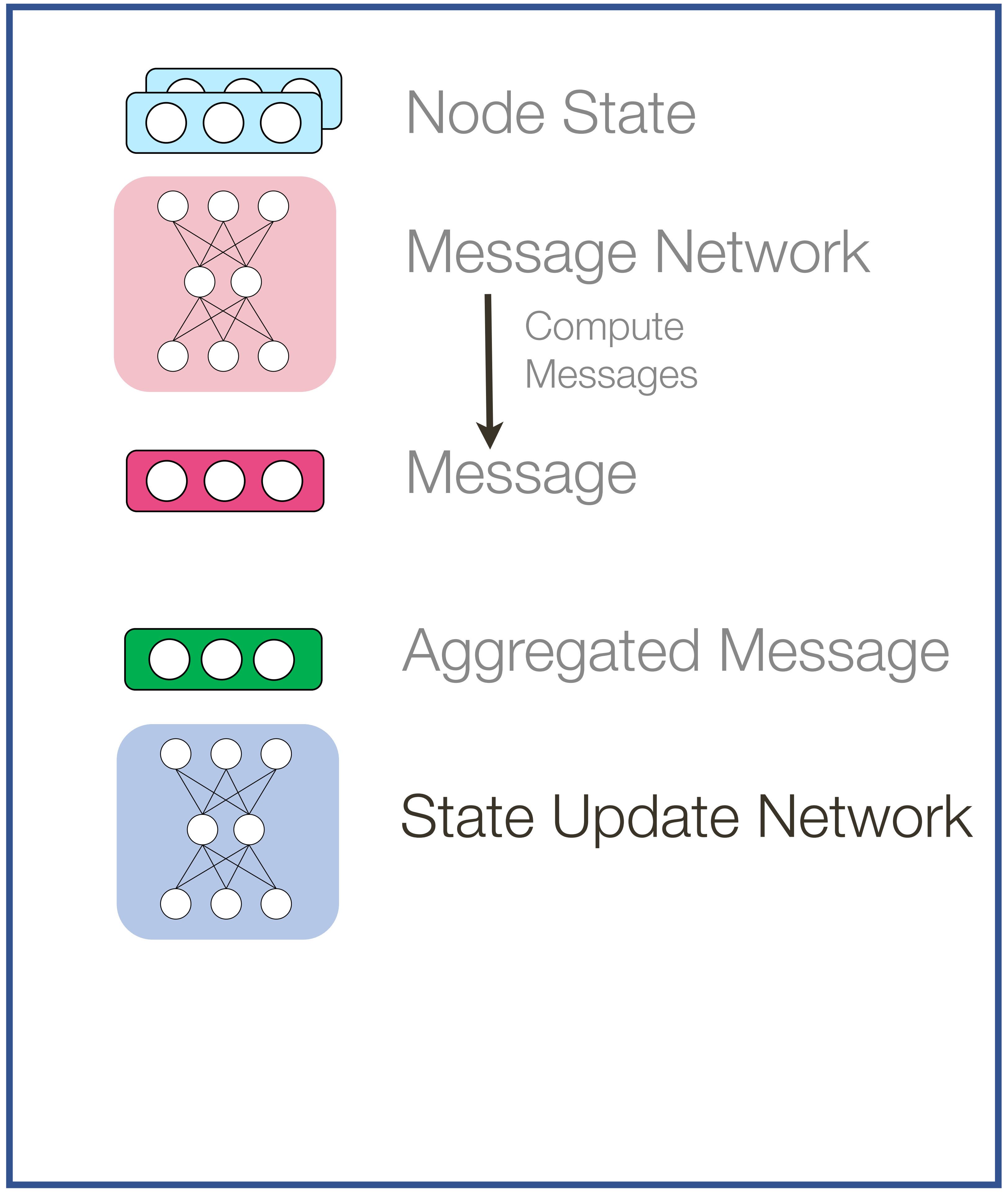
Message Passing in GNNs

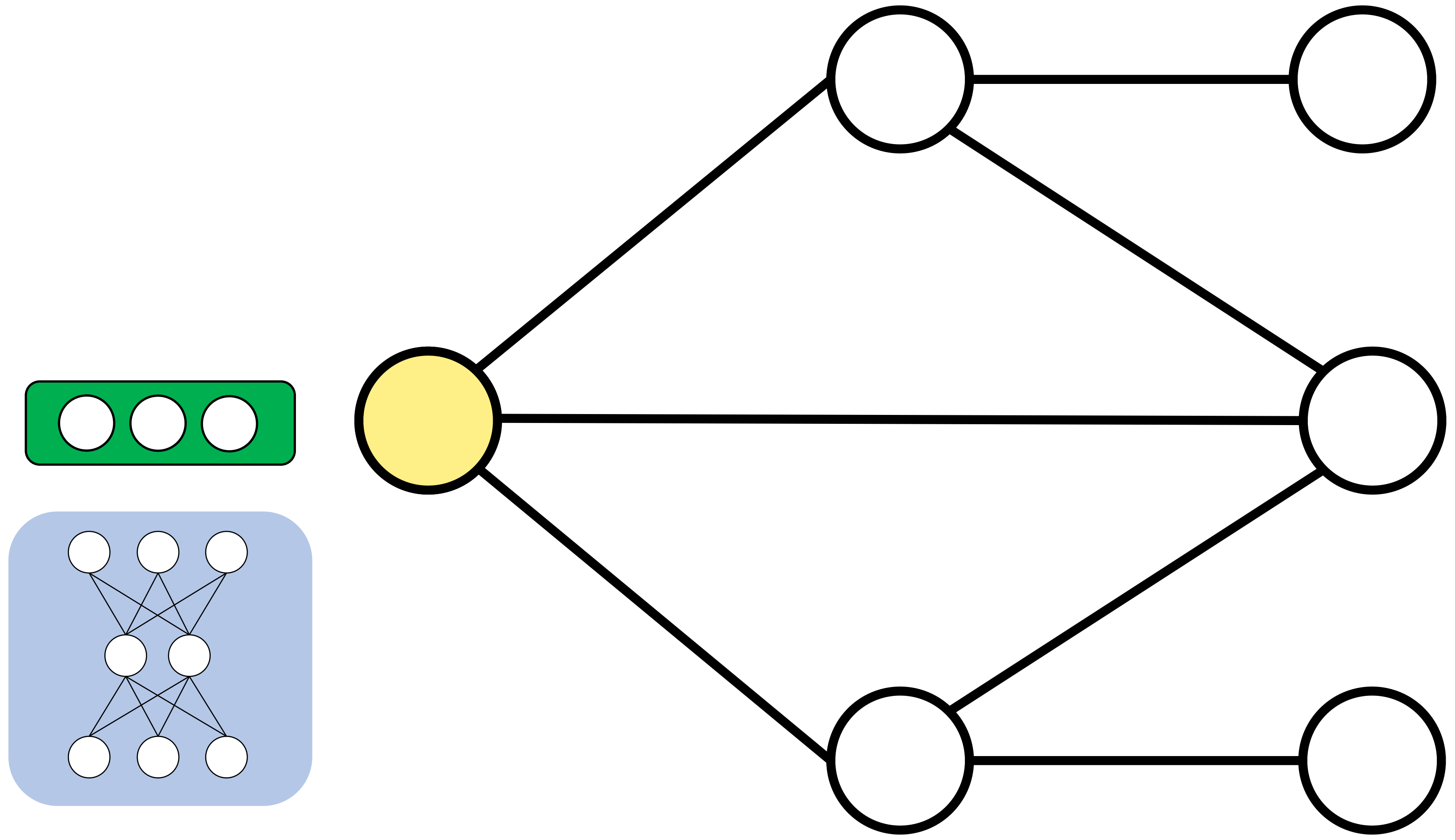
(t+1)-th message passing step/layer

$\mathbf{h}_i^t \quad \mathbf{h}_j^t$

$$\mathbf{m}_{ji}^t = f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t)$$

$$\bar{\mathbf{m}}_i^t = f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right)$$





Message Passing in GNNs

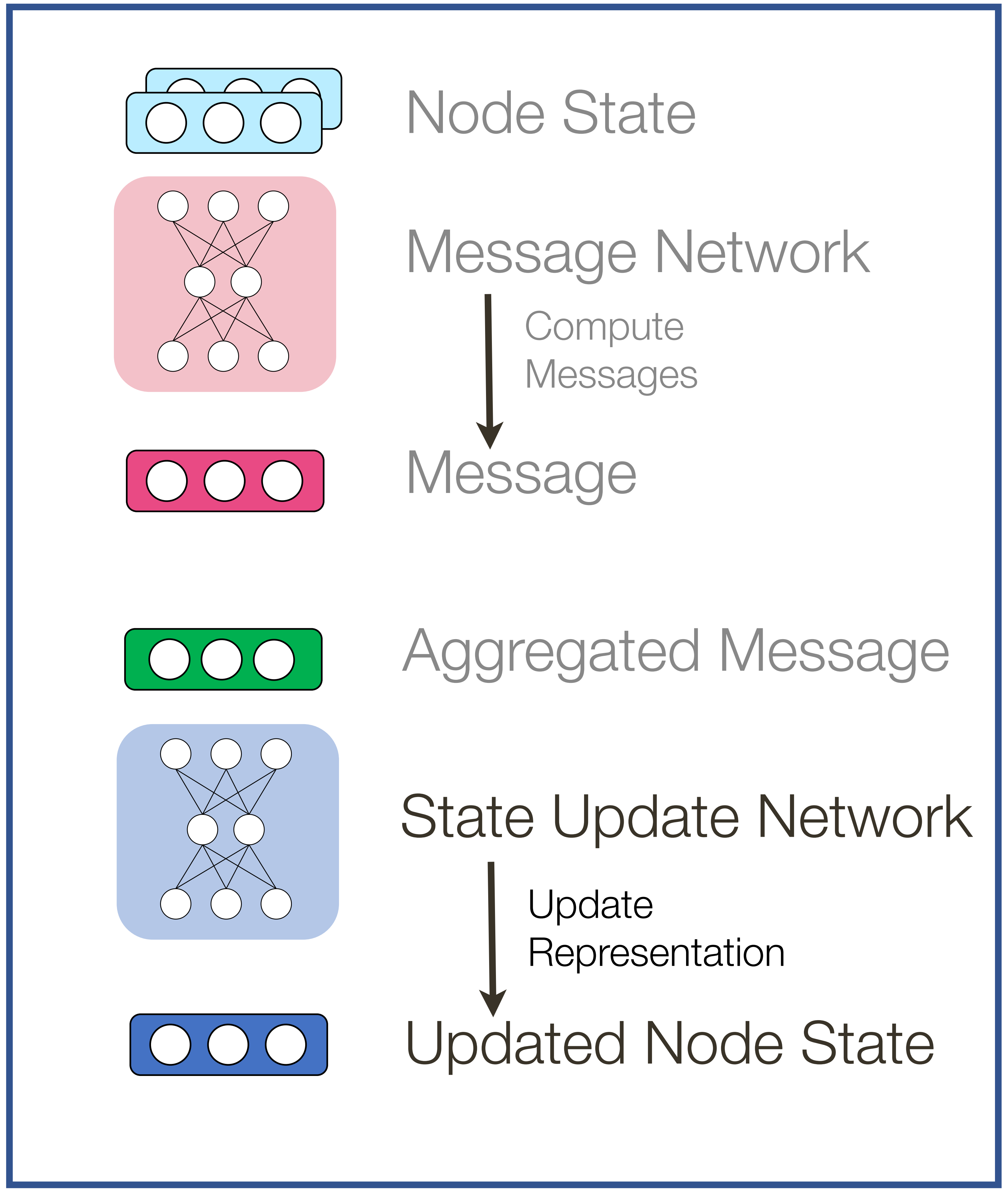
(t+1)-th message passing step/layer

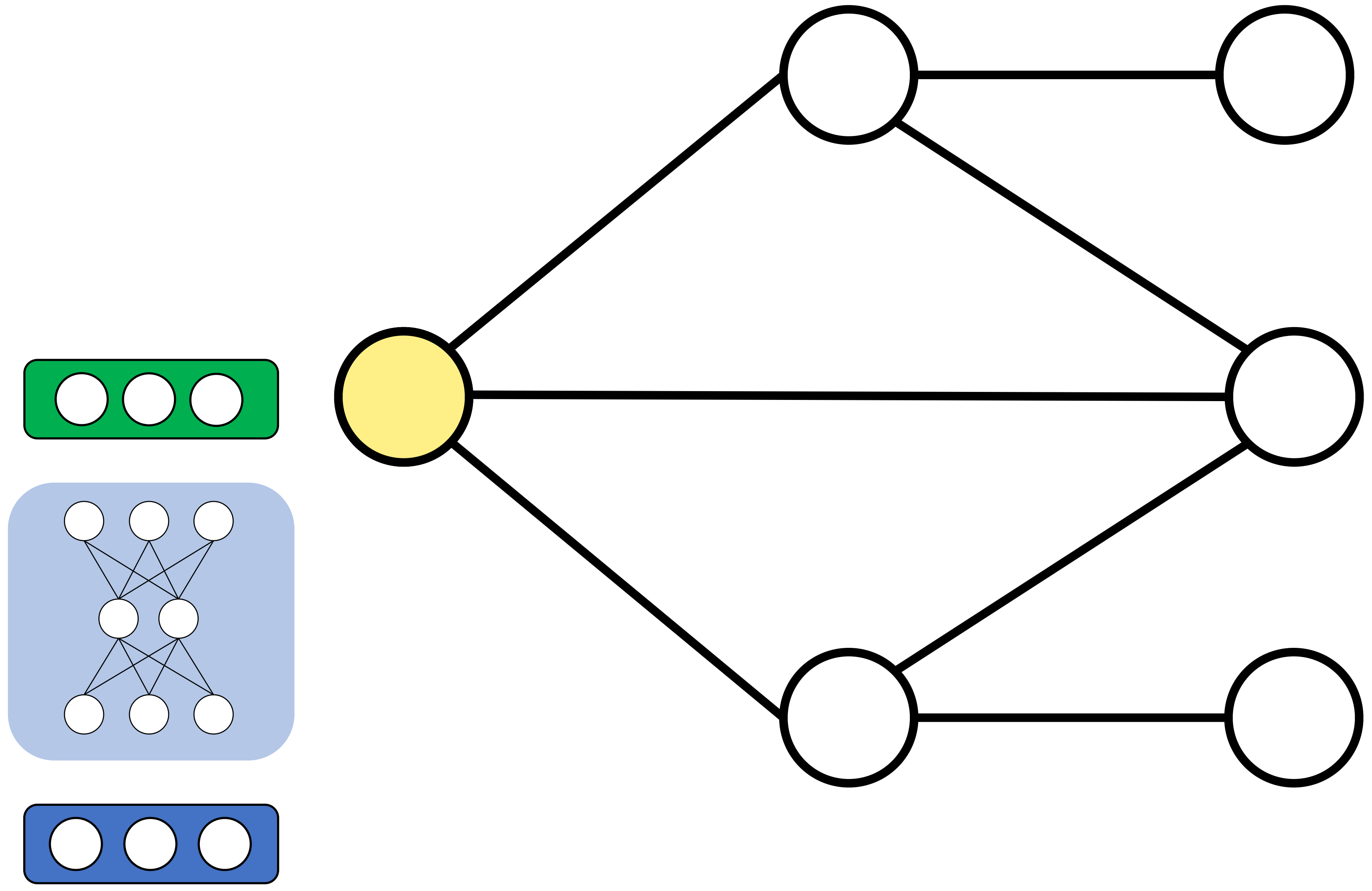
$$\mathbf{h}_i^t \quad \mathbf{h}_j^t$$

$$\mathbf{m}_{ji}^t = f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t)$$

$$\bar{\mathbf{m}}_i^t = f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right)$$

$$\mathbf{h}_i^{t+1} = f_{\text{update}}(\mathbf{h}_i^t, \bar{\mathbf{m}}_i^t)$$





Slide from Renjie Liao

Message Passing in GNNs

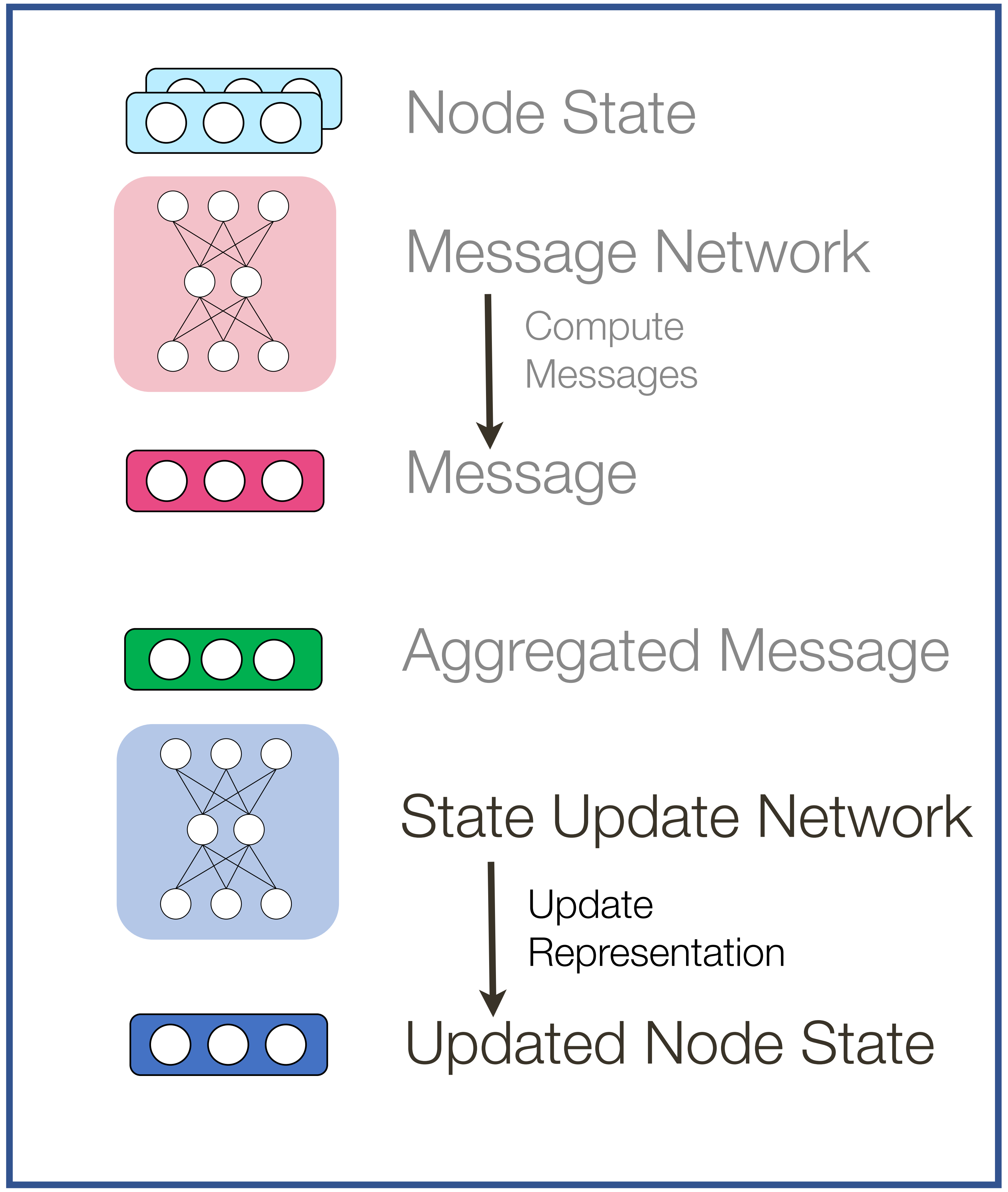
(t+1)-th message passing step/layer

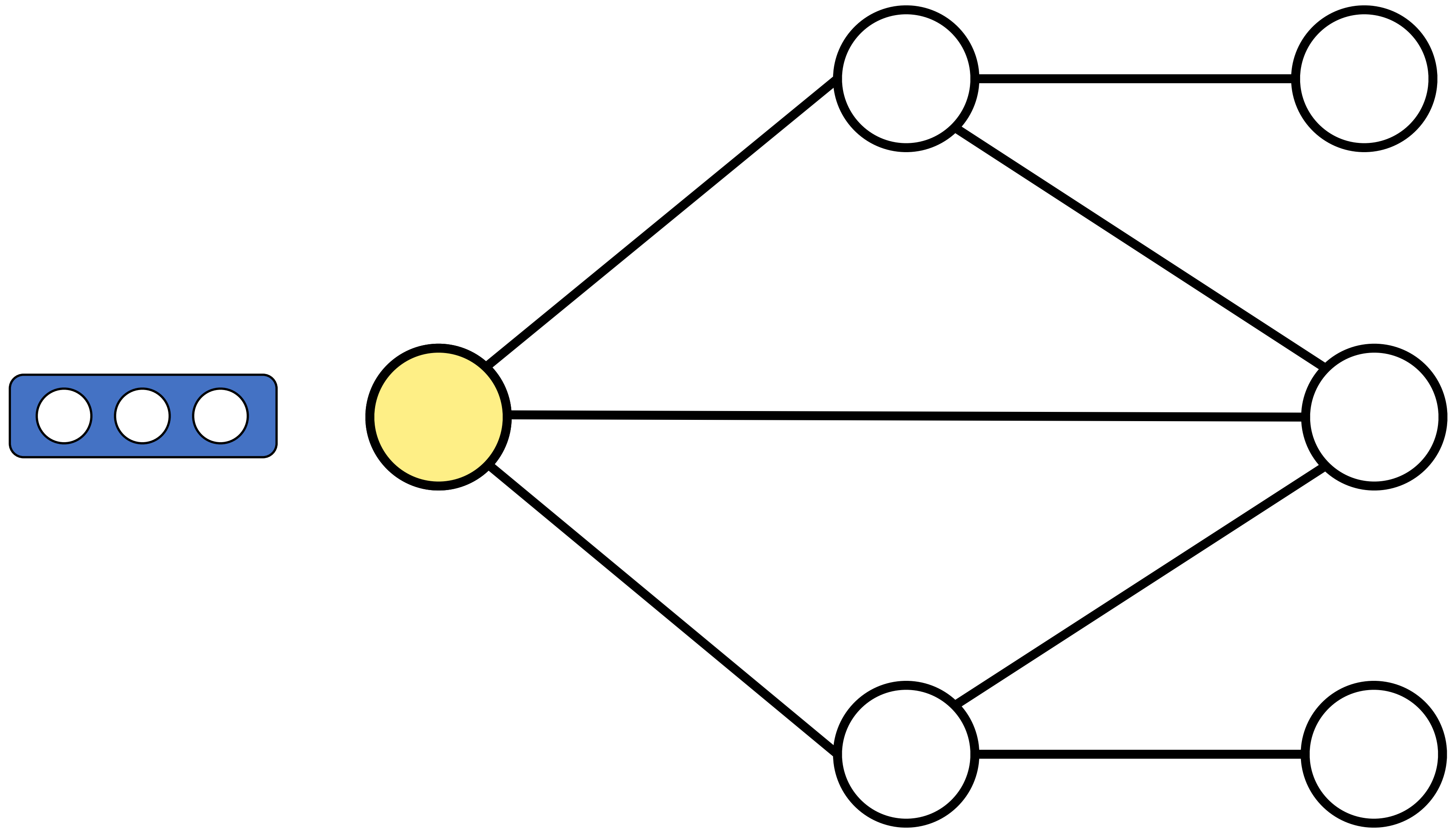
$\mathbf{h}_i^t \quad \mathbf{h}_j^t$

$\mathbf{m}_{ji}^t = f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t)$

$\bar{\mathbf{m}}_i^t = f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right)$

$\mathbf{h}_i^{t+1} = f_{\text{update}}(\mathbf{h}_i^t, \bar{\mathbf{m}}_i^t)$





Slide from Renjie Liao

Message Passing in GNNs

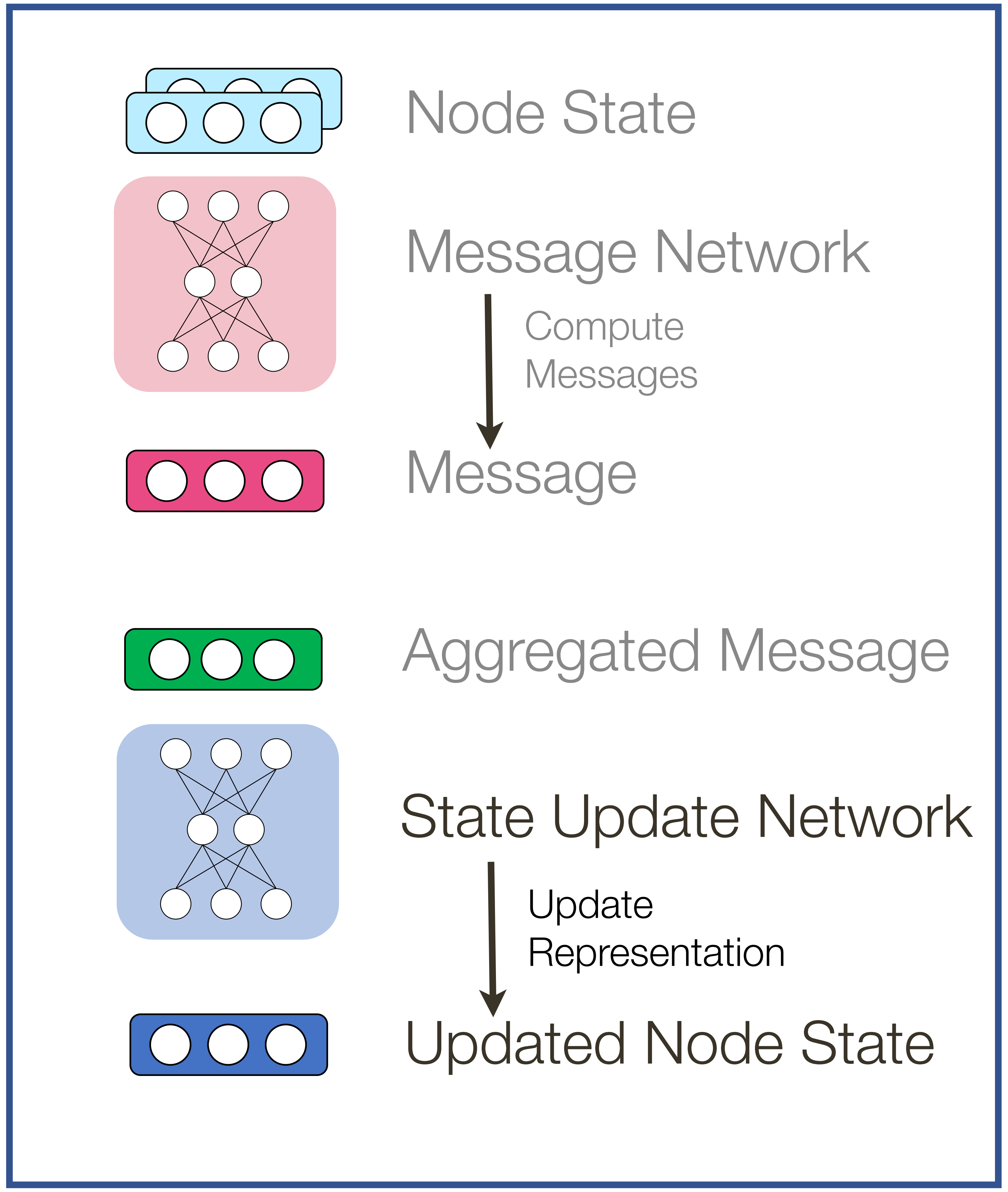
(t+1)-th message passing step/layer

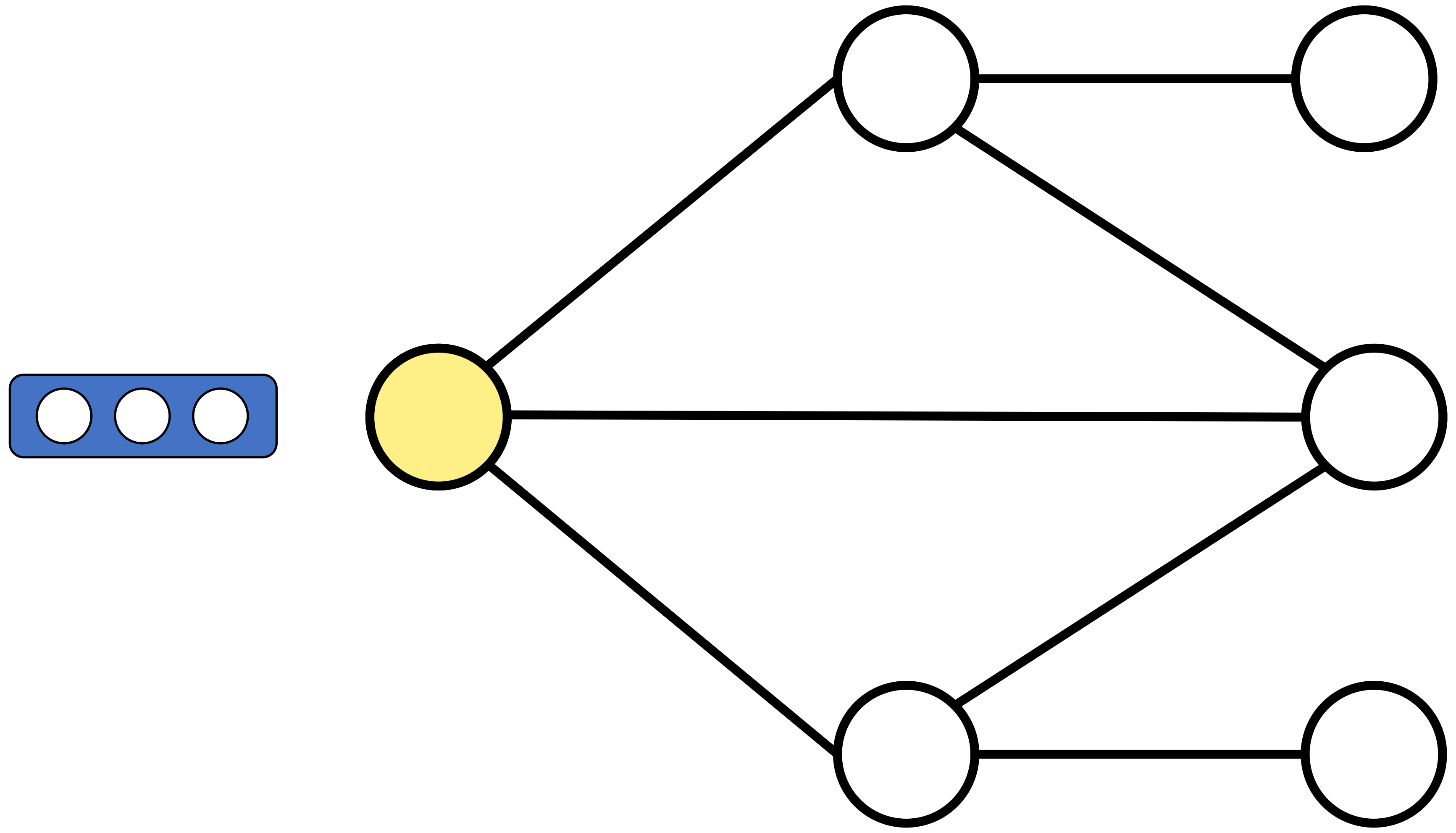
$$\mathbf{h}_i^t \quad \mathbf{h}_j^t$$

$$\mathbf{m}_{ji}^t = f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t)$$

$$\bar{\mathbf{m}}_i^t = f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right)$$

$$\mathbf{h}_i^{t+1} = f_{\text{update}}(\mathbf{h}_i^t, \bar{\mathbf{m}}_i^t)$$





Note: We can do all updates in parallel! (but can also be serial)

Slide from Renjie Liao

GNN **Instantiations**

1. Compute Messages

$$\mathbf{m}_{ji}^{t} = f_{\mathrm{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t)$$

2. Aggregate Messages

$$\bar{\mathbf{m}}_i^t = f_{\mathrm{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right)$$

3. Update Node Representations

$$\mathbf{h}_i^{t+1} = f_{\mathrm{update}}(\mathbf{h}_i^t, \bar{\mathbf{m}}_i^t)$$

GNN **Instantiations**

1. **Compute Messages**

$$\mathbf{m}_{ji}^{t} = f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t)$$

$$f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t) = \mathrm{MLP}([\mathbf{h}_j^t, \mathbf{h}_i^t])$$ [4]

Slide from Renjie Liao

GNN **Instantiations**

1. **Compute Messages**

$$\mathbf{m}_{ji}^t = f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t)$$

$$f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t) = \text{MLP}([\mathbf{h}_j^t, \mathbf{h}_i^t])$$ [4]

$$f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t) = \mathbf{h}_j^t$$ [5]

GNN **Instantiations**

1. **Compute Messages**

$$\mathbf{m}_{ji}^t = f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t)$$

$$f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t) = \text{MLP}([\mathbf{h}_j^t, \mathbf{h}_i^t])$$ [4]

$$f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t) = \mathbf{h}_j^t$$ [5]

$$f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t, \boxed{\mathbf{e}_{ji}}) = \text{MLP}([\mathbf{h}_j^t, \mathbf{h}_i^t, \mathbf{e}_{ji}])$$ [4]

Edge Feature

GNN **Instantiations**

1. Compute Messages

$$\mathbf{m}_{ji}^t = f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t)$$

$$f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t) = \text{MLP}([\mathbf{h}_j^t, \mathbf{h}_i^t])$$ [4]

$$f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t) = \mathbf{h}_j^t$$ [5]

$$f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t, \boxed{\mathbf{e}_{ji}}) = \text{MLP}([\mathbf{h}_j^t, \mathbf{h}_i^t, \mathbf{e}_{ji}])$$ [4]

Edge Feature

2. **Aggregate Messages**

$$\bar{\mathbf{m}}_i^t = f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right)$$

$$f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right) = \sum_{j \in \mathcal{N}_i} \mathbf{m}_{ji}^t$$ [4,5,7]

Slide from Renjie Liao

GNN **Instantiations**

1. Compute Messages

$$\mathbf{m}_{ji}^t = f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t)$$

$$f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t) = \text{MLP}([\mathbf{h}_j^t, \mathbf{h}_i^t])$$ [4]

$$f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t) = \mathbf{h}_j^t$$ [5]

$$f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t, \boxed{\mathbf{e}_{ji}}) = \text{MLP}([\mathbf{h}_j^t, \mathbf{h}_i^t, \mathbf{e}_{ji}])$$ [4]

Edge Feature

2. **Aggregate Messages**

$$\bar{\mathbf{m}}_i^t = f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right)$$

$$f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right) = \sum_{j \in \mathcal{N}_i} \mathbf{m}_{ji}^t$$ [4,5,7]

$$f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right) = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \mathbf{m}_{ji}^t$$ [6]

GNN **Instantiations**

1. Compute Messages

$$\mathbf{m}_{ji}^t = f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t)$$

$$f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t) = \text{MLP}([\mathbf{h}_j^t, \mathbf{h}_i^t])$$ [4]

$$f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t) = \mathbf{h}_j^t$$ [5]

$$f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t, \boxed{\mathbf{e}_{ji}}) = \text{MLP}([\mathbf{h}_j^t, \mathbf{h}_i^t, \mathbf{e}_{ji}])$$ [4]

Edge Feature

2. **Aggregate Messages**

$$\bar{\mathbf{m}}_i^t = f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right)$$

$$f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right) = \sum_{j \in \mathcal{N}_i} \mathbf{m}_{ji}^t$$ [4,5,7]

$$f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right) = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \mathbf{m}_{ji}^t$$ [6]

$$f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right) = \max_{j \in \mathcal{N}_i} \mathbf{m}_{ji}^t$$ [6]

GNN **Instantiations**

1. Compute Messages

$$\mathbf{m}_{ji}^t = f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t)$$

$$f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t) = \text{MLP}([\mathbf{h}_j^t, \mathbf{h}_i^t]) \quad [4]$$

$$f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t) = \mathbf{h}_j^t \quad [5]$$

$$f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t, \boxed{\mathbf{e}_{ji}}) = \text{MLP}([\mathbf{h}_j^t, \mathbf{h}_i^t, \mathbf{e}_{ji}]) \quad [4]$$

Edge Feature

2. **Aggregate Messages**

$$\bar{\mathbf{m}}_i^t = f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right)$$

$$f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right) = \sum_{j \in \mathcal{N}_i} \mathbf{m}_{ji}^t \quad [4,5,7]$$

$$f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right) = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \mathbf{m}_{ji}^t \quad [6]$$

$$f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right) = \max_{j \in \mathcal{N}_i} \mathbf{m}_{ji}^t \quad [6]$$

$$f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right) = \text{LSTM}\left([\mathbf{m}_{ji}^t | j \in \mathcal{N}_i]\right) \quad [6]$$

GNN **Instantiations**

1. Compute Messages

$$\mathbf{m}_{ji}^t = f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t)$$

$$f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t) = \text{MLP}([\mathbf{h}_j^t, \mathbf{h}_i^t])$$ [4]

$$f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t) = \mathbf{h}_j^t$$ [5]

$$f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t, \boxed{\mathbf{e}_{ji}}) = \text{MLP}([\mathbf{h}_j^t, \mathbf{h}_i^t, \mathbf{e}_{ji}])$$ [4]

Edge Feature

2. Aggregate Messages

$$\bar{\mathbf{m}}_i^t = f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right)$$

$$f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right) = \sum_{j \in \mathcal{N}_i} \mathbf{m}_{ji}^t$$ [4,5,7]

$$f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right) = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \mathbf{m}_{ji}^t$$ [6]

$$f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right) = \max_{j \in \mathcal{N}_i} \mathbf{m}_{ji}^t$$ [6]

$$f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right) = \text{LSTM}\left([\mathbf{m}_{ji}^t | j \in \mathcal{N}_i]\right)$$ [6]

3. **Update Node Representations**

$$\mathbf{h}_i^{t+1} = f_{\text{update}}(\mathbf{h}_i^t, \bar{\mathbf{m}}_i^t)$$

$$f_{\text{update}}(\mathbf{h}_i^t, \bar{\mathbf{m}}_i^t) = \text{GRU}(\mathbf{h}_i^t, \bar{\mathbf{m}}_i^t)$$ [4,7]

Slide from Renjie Liao

GNN **Instantiations**

1. Compute Messages

$$\mathbf{m}_{ji}^t = f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t)$$

$$f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t) = \text{MLP}([\mathbf{h}_j^t, \mathbf{h}_i^t])$$ [4]

$$f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t) = \mathbf{h}_j^t$$ [5]

$$f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t, \boxed{\mathbf{e}_{ji}}) = \text{MLP}([\mathbf{h}_j^t, \mathbf{h}_i^t, \mathbf{e}_{ji}])$$ [4]

Edge Feature

2. Aggregate Messages

$$\bar{\mathbf{m}}_i^t = f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right)$$

$$f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right) = \sum_{j \in \mathcal{N}_i} \mathbf{m}_{ji}^t$$ [4,5,7]

$$f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right) = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \mathbf{m}_{ji}^t$$ [6]

$$f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right) = \max_{j \in \mathcal{N}_i} \mathbf{m}_{ji}^t$$ [6]

$$f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right) = \text{LSTM}\left([\mathbf{m}_{ji}^t | j \in \mathcal{N}_i]\right)$$ [6]

3. **Update Node Representations**

$$\mathbf{h}_i^{t+1} = f_{\text{update}}(\mathbf{h}_i^t, \bar{\mathbf{m}}_i^t)$$

$$f_{\text{update}}(\mathbf{h}_i^t, \bar{\mathbf{m}}_i^t) = \text{GRU}(\mathbf{h}_i^t, \bar{\mathbf{m}}_i^t)$$ [4,7]

$$f_{\text{update}}(\mathbf{h}_i^t, \bar{\mathbf{m}}_i^t) = \text{MLP}_1(\mathbf{h}_i^t) + \text{MLP}_2(\bar{\mathbf{m}}_i^t)$$ [5]

GNN **Instantiations**

1. Compute Messages

$$\mathbf{m}_{ji}^t = f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t)$$

$$f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t) = \text{MLP}([\mathbf{h}_j^t, \mathbf{h}_i^t]) \quad [4]$$

$$f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t) = \mathbf{h}_j^t \quad [5]$$

$$f_{\text{msg}}(\mathbf{h}_j^t, \mathbf{h}_i^t, \boxed{\mathbf{e}_{ji}}) = \text{MLP}([\mathbf{h}_j^t, \mathbf{h}_i^t, \mathbf{e}_{ji}]) \quad [4]$$

Edge Feature

2. Aggregate Messages

$$\bar{\mathbf{m}}_i^t = f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right)$$

$$f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right) = \textstyle\sum_{j \in \mathcal{N}_i} \mathbf{m}_{ji}^t \quad [4,5,7]$$

$$f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right) = \frac{1}{|\mathcal{N}_i|} \textstyle\sum_{j \in \mathcal{N}_i} \mathbf{m}_{ji}^t \quad [6]$$

$$f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right) = \max_{j \in \mathcal{N}_i} \mathbf{m}_{ji}^t \quad [6]$$

$$f_{\text{agg}}\left(\{\mathbf{m}_{ji}^t | j \in \mathcal{N}_i\}\right) = \text{LSTM}\left([\mathbf{m}_{ji}^t | j \in \mathcal{N}_i]\right) \quad [6]$$

3. **Update Node Representations**

$$\mathbf{h}_i^{t+1} = f_{\text{update}}(\mathbf{h}_i^t, \bar{\mathbf{m}}_i^t)$$

$$f_{\text{update}}(\mathbf{h}_i^t, \bar{\mathbf{m}}_i^t) = \text{GRU}(\mathbf{h}_i^t, \bar{\mathbf{m}}_i^t) \quad [4,7]$$

$$f_{\text{update}}(\mathbf{h}_i^t, \bar{\mathbf{m}}_i^t) = \text{MLP}_1(\mathbf{h}_i^t) + \text{MLP}_2(\bar{\mathbf{m}}_i^t) \quad [5]$$

$$f_{\text{update}}(\mathbf{h}_i^t, \bar{\mathbf{m}}_i^t) = \text{MLP}([\mathbf{h}_i^t, \bar{\mathbf{m}}_i^t]) \quad [6]$$

Slide from Renjie Liao

GNN **Readout**

1. Node Readout

$$\mathbf{y}_i = f_{\text{readout}}(\mathbf{h}_i^T)$$

2. Edge Readout

$$\mathbf{y}_{ij} = f_{\text{readout}}(\mathbf{h}_i^T, \mathbf{h}_j^T)$$

3. Graph Readout

$$\mathbf{y} = f_{\text{readout}}(\{\mathbf{h}_i^T\})$$

Slide from Renjie Liao

GNN **Readout**

1. **Node Readout**

$$\mathbf{y}_i = f_{\text{readout}}(\mathbf{h}_i^T) \qquad f_{\text{readout}}(\mathbf{h}_i^T) = \text{MLP}(\mathbf{h}_i^T)$$

GNN **Readout**

1. Node Readout

$$\mathbf{y}_i = f_{\text{readout}}(\mathbf{h}_i^T)$$

$$f_{\text{readout}}(\mathbf{h}_i^T) = \text{MLP}(\mathbf{h}_i^T)$$

2. **Edge Readout**

$$\mathbf{y}_{ij} = f_{\text{readout}}(\mathbf{h}_i^T, \mathbf{h}_j^T)$$

$$f_{\text{readout}}(\mathbf{h}_i^T, \mathbf{h}_j^T) = \text{MLP}([\mathbf{h}_i^T, \mathbf{h}_j^T])$$

$$f_{\text{readout}}(\mathbf{h}_i^T, \mathbf{h}_j^T, \boxed{e_{ij}}) = \text{MLP}([\mathbf{h}_i^T, \mathbf{h}_j^T, e_{ij}])$$

Edge Feature

Slide from Renjie Liao

GNN **Readout**

1. Node Readout

$$\mathbf{y}_i = f_{\text{readout}}(\mathbf{h}_i^T)$$

$$f_{\text{readout}}(\mathbf{h}_i^T) = \text{MLP}(\mathbf{h}_i^T)$$

2. Edge Readout

$$\mathbf{y}_{ij} = f_{\text{readout}}(\mathbf{h}_i^T, \mathbf{h}_j^T)$$

$$f_{\text{readout}}(\mathbf{h}_i^T, \mathbf{h}_j^T) = \text{MLP}([\mathbf{h}_i^T, \mathbf{h}_j^T])$$

$$f_{\text{readout}}(\mathbf{h}_i^T, \mathbf{h}_j^T, \boxed{e_{ij}}) = \text{MLP}([\mathbf{h}_i^T, \mathbf{h}_j^T, e_{ij}])$$

Edge Feature

3. **Graph Readout**

$$\mathbf{y} = f_{\text{readout}}(\{\mathbf{h}_i^T\})$$

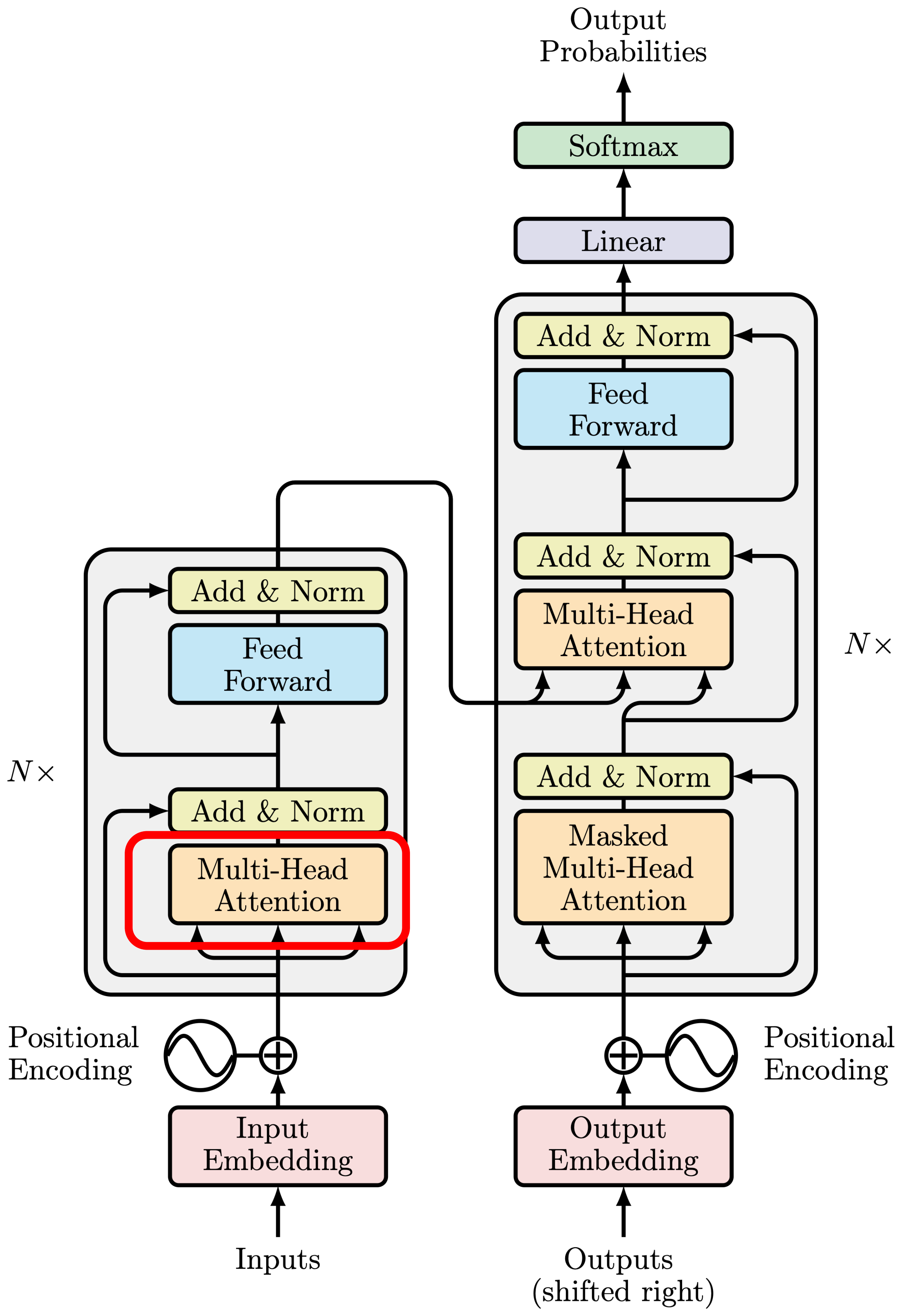
$$f_{\text{readout}}(\{\mathbf{h}_i^T\}) = \sum_i \sigma(\text{MLP}_1(\mathbf{h}_i^T))\text{MLP}_2(\mathbf{h}_i^T)$$

$$f_{\text{readout}}(\{\mathbf{h}_i^T\}, \boxed{\mathbf{g}}) = \sum_i \sigma(\text{MLP}_1(\mathbf{h}_i^T, \mathbf{g}))\text{MLP}_2(\mathbf{h}_i^T, \mathbf{g})$$

Graph Feature

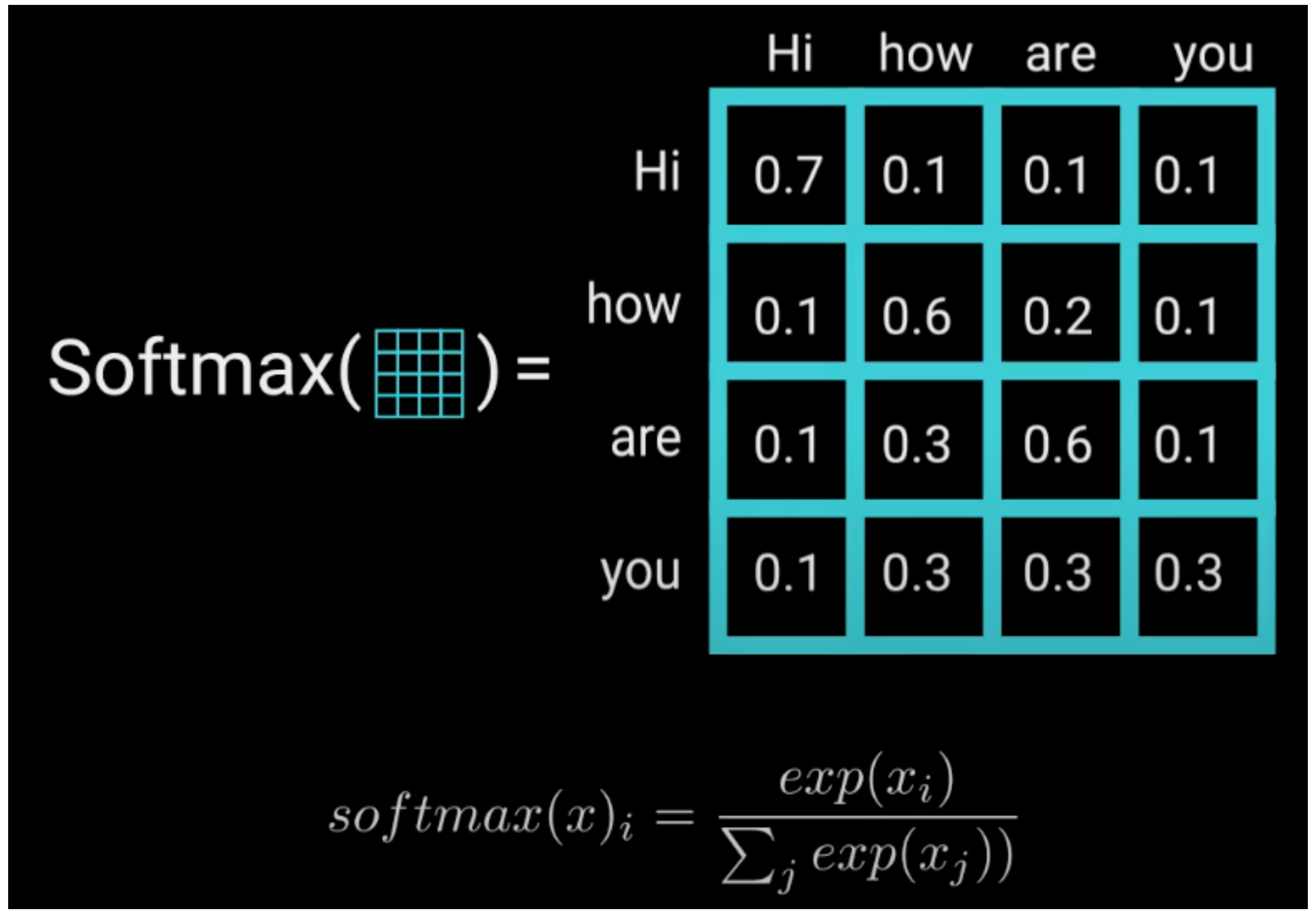
Slide from Renjie Liao

GNN **Relationship** to **Transformers**



— Attention can be viewed as the weighted adjacency matrix of a fully connected graph!

— Transformers (esp. encoder) can be viewed as GNNs applied to fully connected graphs!



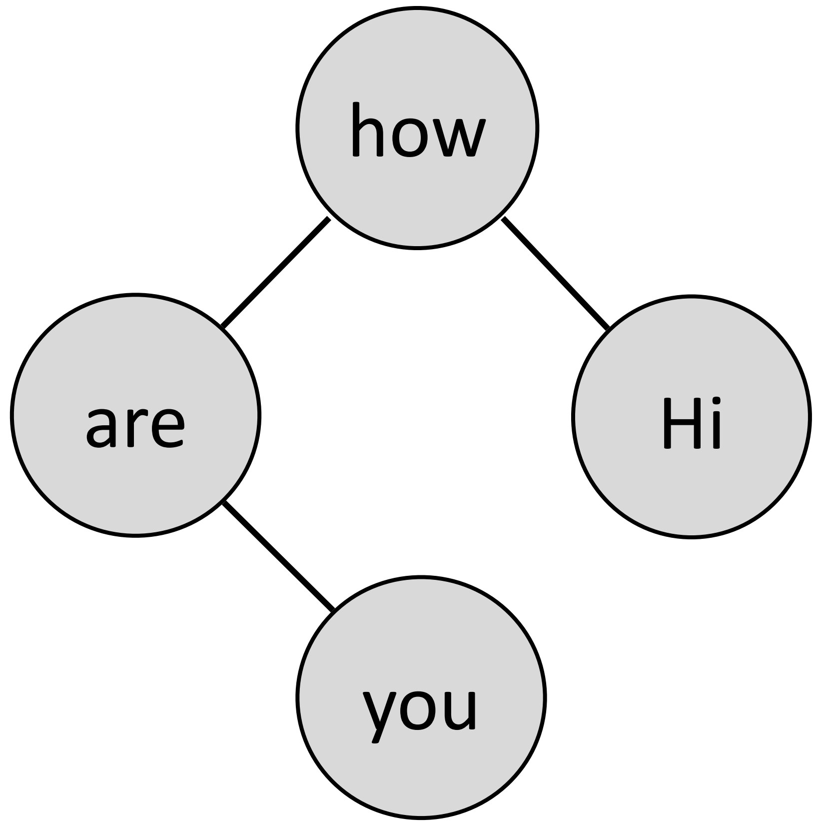
Slide from Renjie Liao

Image Credit: [9] & https://towardsdatascience.com/illustrated-guide-to-transformers-step-by-step-explanation-f74876522bc0

GNN **Relationship** to **Transformers**

— Apply the adjacency matrix as a mask to the attention and renormalize it, is like Graph Attention Networks (GAT) [10]

— Encoder connectivities/distances as bias of the attention [11]



	Hi	how	are	you
Hi	0	1	0	1
how	1	0	0	0
are	0	0	0	1
you	1	0	1	0

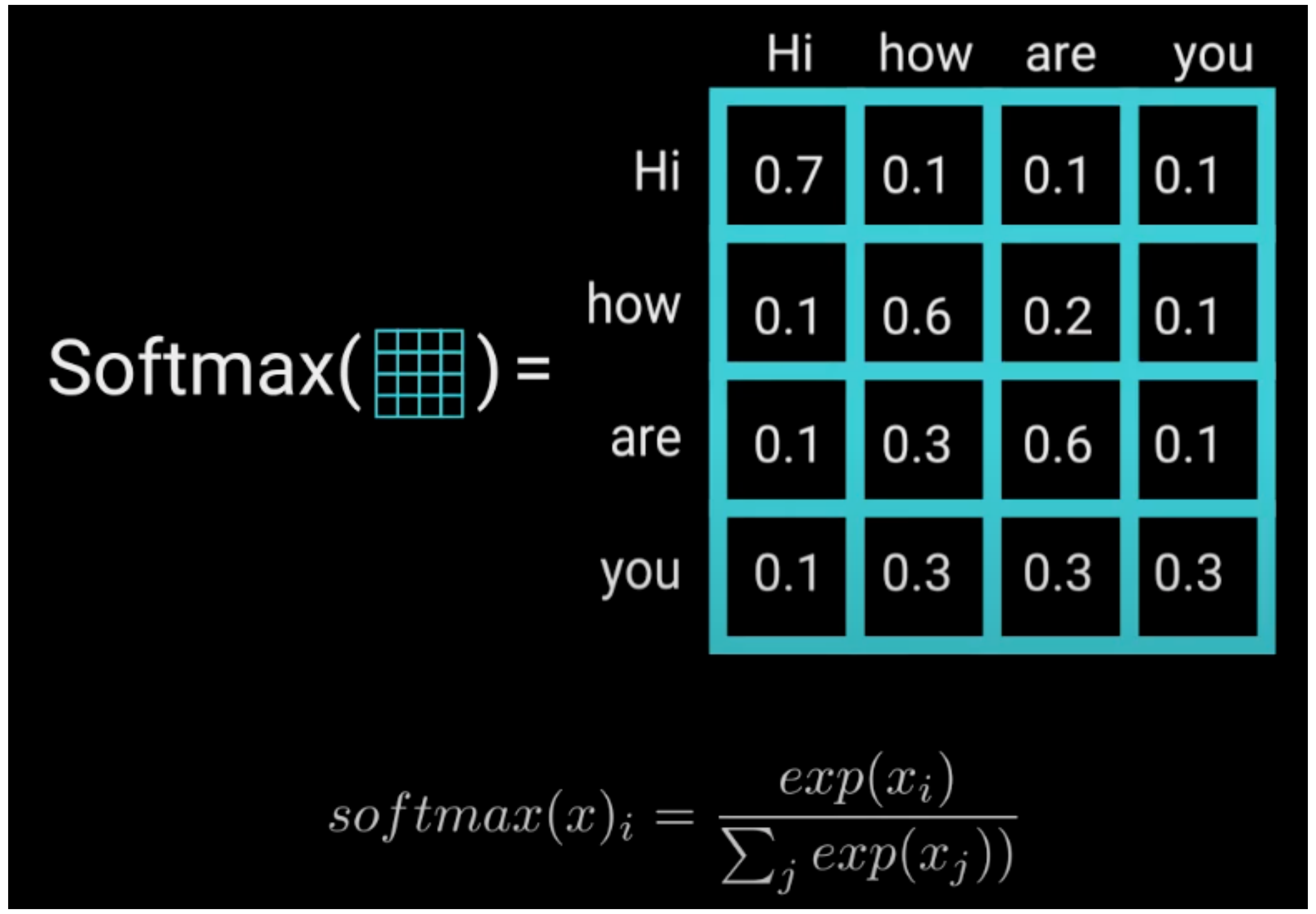


Image Credit: [9] & https://towardsdatascience.com/illustrated-guide-to-transformers-step-by-step-explanation-f74876522bc0

References

[1] Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M. and Monfardini, G., 2008. The graph neural network model. IEEE transactions on neural networks, 20(1), pp.61-80.

[2] Goller, C. and Kuchler, A., 1996, June. Learning task-dependent distributed representations by backpropagation through structure. In Proceedings of International Conference on Neural Networks (ICNN'96) (Vol. 1, pp. 347-352). IEEE.

[3] Ackley, D.H., Hinton, G.E. and Sejnowski, T.J., 1985. A learning algorithm for Boltzmann machines. Cognitive science, 9(1), pp.147-169.

[4] Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O. and Dahl, G.E., 2017, July. Neural message passing for quantum chemistry. In International conference on machine learning (pp. 1263-1272). PMLR.

[5] Morris, C., Ritzert, M., Fey, M., Hamilton, W.L., Lenssen, J.E., Rattan, G. and Grohe, M., 2019, July. Weisfeiler and leman go neural: Higher-order graph neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, No. 01, pp. 4602-4609).

[6] Hamilton, W.L., Ying, R. and Leskovec, J., 2017, December. Inductive representation learning on large graphs. In Proceedings of the 31st International Conference on Neural Information Processing Systems (pp. 1025-1035).

[7] Li, Y., Tarlow, D., Brockschmidt, M. and Zemel, R., 2015. Gated graph sequence neural networks. arXiv preprint arXiv:1511.05493.

[8] Liao, R., Brockschmidt, M., Tarlow, D., Gaunt, A.L., Urtasun, R. and Zemel, R., 2018. Graph partition neural networks for semi-supervised classification. arXiv preprint arXiv:1803.06272.

[9] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).

[10] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P. and Bengio, Y., 2017. Graph attention networks. arXiv preprint arXiv:1710.10903.

[11] Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y. and Liu, T.Y., 2021. Do Transformers Really Perform Bad for Graph Representation?. arXiv preprint arXiv:2106.05234.