

Week 6 Theory: Hypergraph Neural Networks

Presented by Xiaohang Yu



Outline

PhD Project

Research Background

Hypergraph Neural Networks (GNNs):

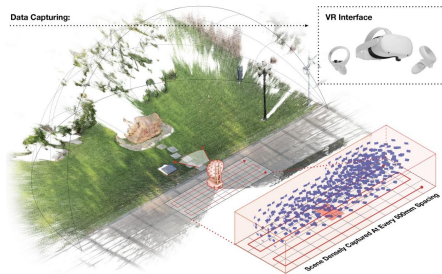
- Background
- Taxonomy
- Encoding design
- Training design
- Applications
- Conclusions

PhD Project

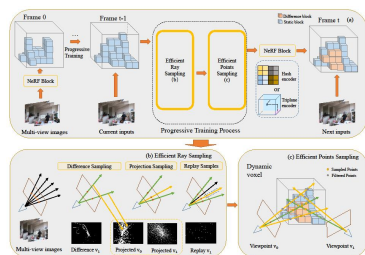
- I am a first-year PhD student at Mathis Laboratory of adaptive intelligence at EPFL, Switzerland. My advisor is Prof. Mackenzie Mathis.
- I got my Master's degree of Data Science from Tsinghua University, China. My master's thesis focused on large-scale scene reconstruction and rendering and the corresponding applications in VR to improve the sense of immersion.
- Now, I am focusing on 3D/4D reconstruction, particularly recovering shape and motion from partial observations like videos.
- For any inquiries, feel free to reach out to me via mail!

Research Background: 2D/3D Vision and Beyond

3D – novel view synthesis Immersive light field reconstruction



Capture & render:
Dataset and VR application



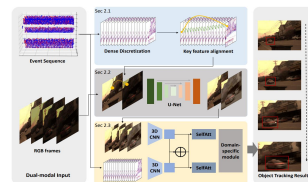
Representation:
dynamic reconstruction

Render:
Detail generation



Representation:
Large-scale scene reconstruction

2D – object tracking & behavior recognition

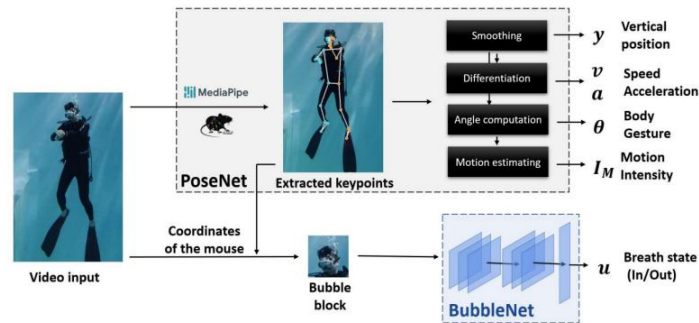


Object tracking
on drones



Behavior recognition
for farmers

Vision-based agent training



AI coach for divers

ACM KDD 2024

A Survey on Hypergraph Neural Networks: An In-Depth and Step-by-Step Guide

Sunwoo Kim*

KAIST

Seoul, Republic of Korea

kswoo97@kaist.ac.kr

Soo Yong Lee*

KAIST

Seoul, Republic of Korea

syleetolow@kaist.ac.kr

Yue Gao

Tsinghua University

Beijing, China

gaoyue@tsinghua.edu.cn

Alessia Antelmi

University of Turin

Turin, Italy

alessia.antelmi@unito.it

Mirko Polato

University of Turin

Turin, Italy

mirko.polato@unito.it

Kijung Shin[†]

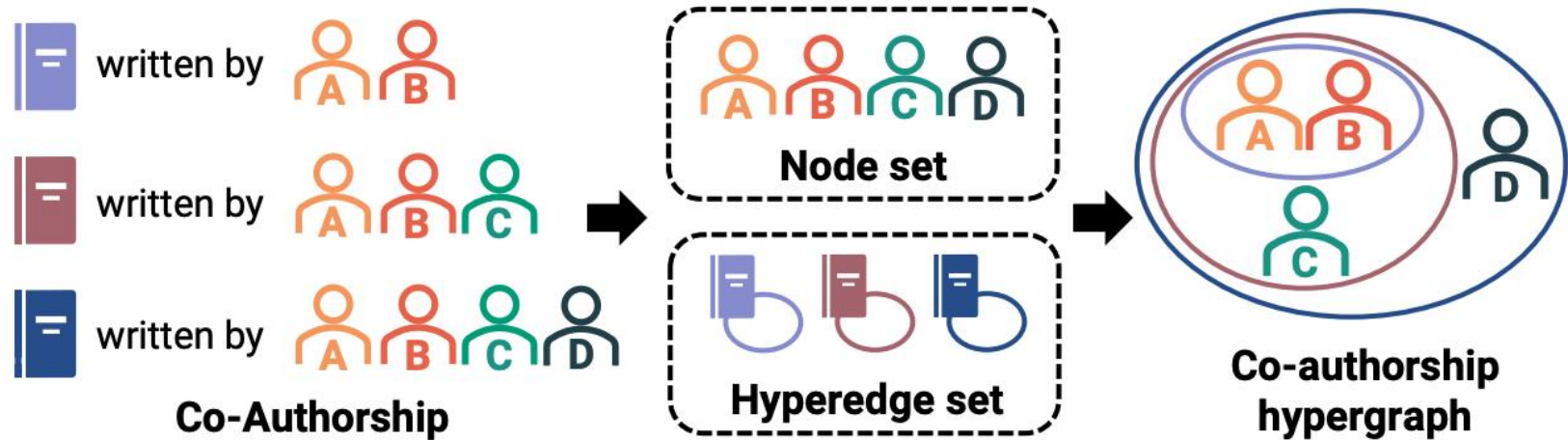
KAIST

Seoul, Republic of Korea

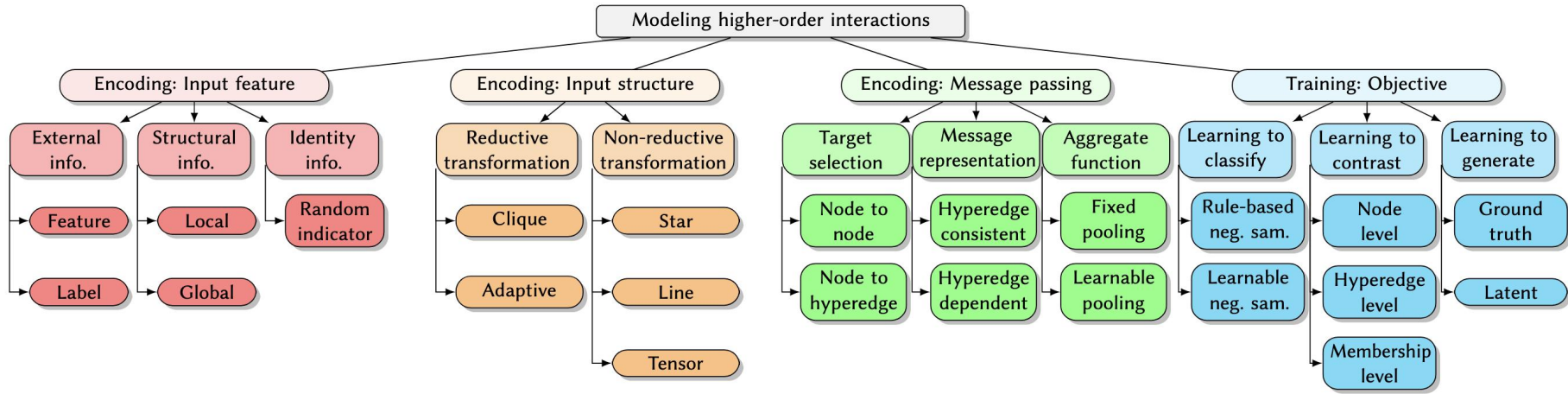
kijungs@kaist.ac.kr

Background: Hypergraphs

- Higher-order interactions are commonly modeled as a **hypergraph**.
- A hypergraph consists of a **node** set and a **hyperedge** set.
- A **hyperedge** (i.e., a subset of nodes) models a higher-order interaction



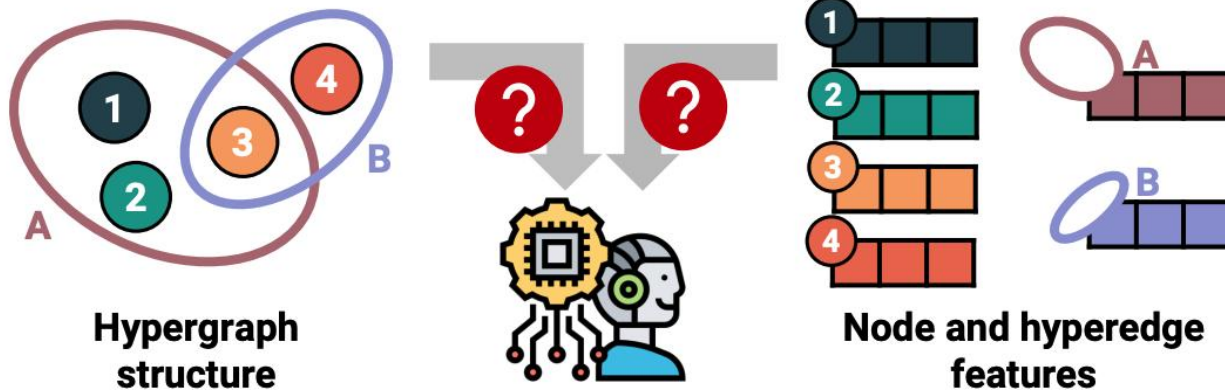
Taxonomy of HNNs



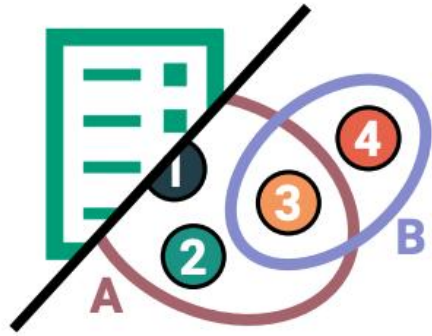
- Encoding: how do HNNs effectively capture HOIs
- Training: how to encode HOIs with training objectives, especially when external labels are scarce or absent

Encoding design - Overview

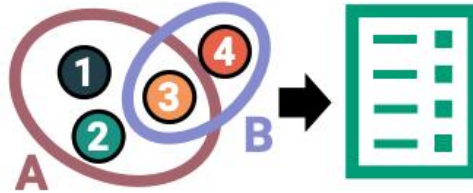
- Typical inputs of hypergraph neural networks (HNNs) are **hypergraph structure** and **node (and/or hyperedge) feature vectors**.
- The quality of the inputs can be critical for the effectiveness of HNNs
- **Key questions** regarding HNN inputs are:
 - Q1. **What** input features can be used for nodes and hyperedges?
 - Q2. **How** can hypergraph structures be represented?



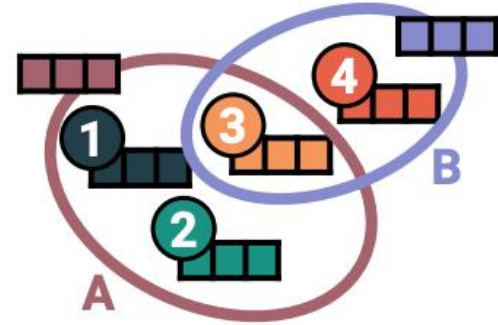
Encoding design – Step1. Design Features to Reflect HOIS



External features



Structural features

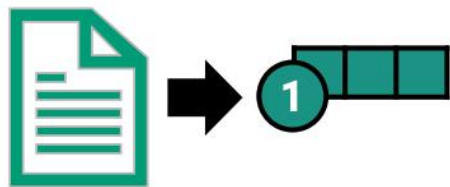


Identity features

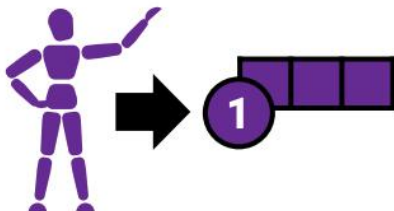
Encoding design – Step1. Design Features to Reflect HOIS

External features are additionally given information that is **not directly derived from** the input hypergraph structure.

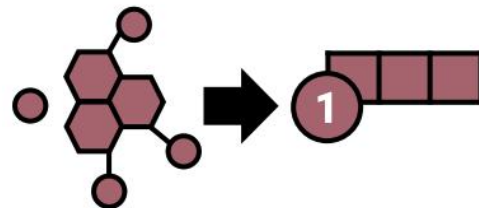
- They complement structural information reflected in hypergraphs.
- External node features in popular benchmark datasets include:
 - 1) bag-of-words vectors for article nodes
 - 2) visual object embeddings for image nodes
 - 3) the counts of atoms for molecule nodes.



Bag-of-words



**Visual object
embedding**

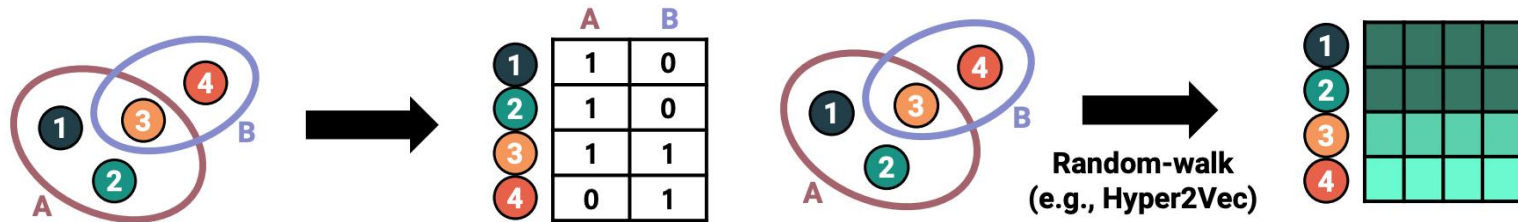


Count of atoms

Encoding design – Step1. Design Features to Reflect HOIS

Structural features are derived from the input hypergraph structure.

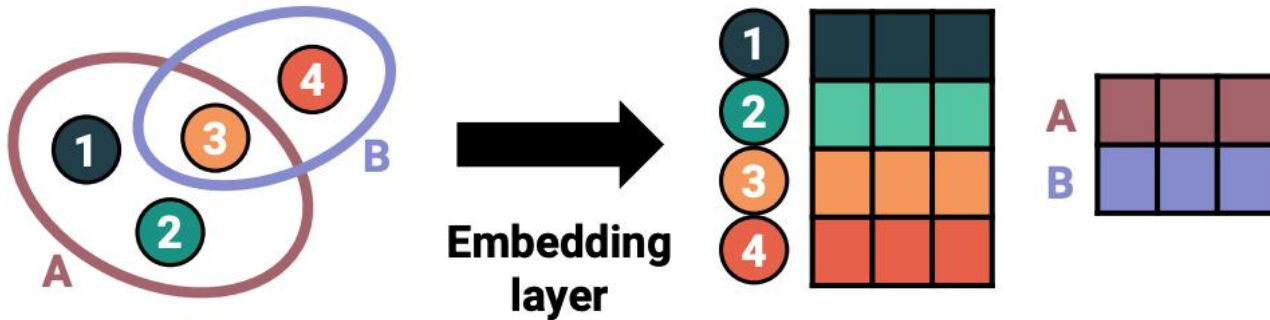
- They typically capture structural proximity or similarity among nodes.
- Structural features are either **local** or **global**.
- Local structural features capture node-hyperedge membership.
- leveraged the rows of the incidence matrix as part of their node features.
- Global structural features capture proximity or similarity among nodes beyond direct connections.
- leveraged random walks to capture such proximity and similarity



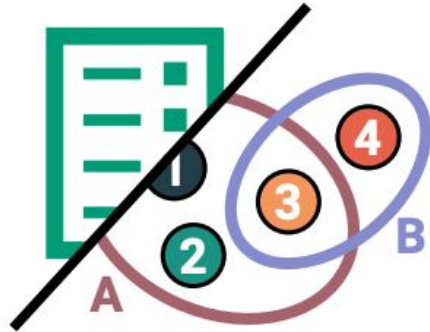
Encoding design – Step1. Design Features to Reflect HOIS

Identity features are uniquely assigned to each node and hyperedge.

- They encourage HNNs to distinguish different nodes and hyperedges.
- some work “learned” identity features using an embedding layer.
- Each node and hyperedge has a feature vector that is learned during training



Encoding design – Step1. Input: Design Features to Reflect HOIS



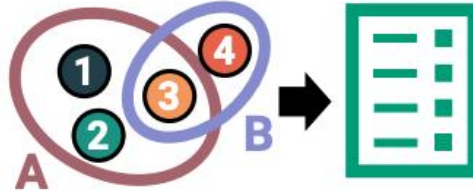
External features



Informative



Costly to obtain



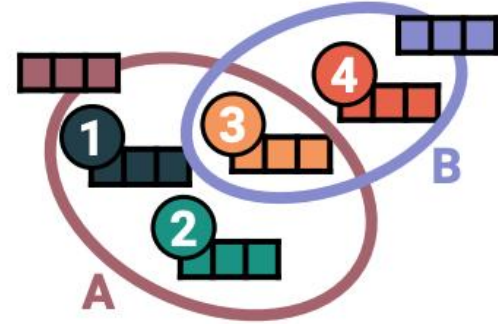
Structural features



Readily obtainable



Not scalable



Identity features

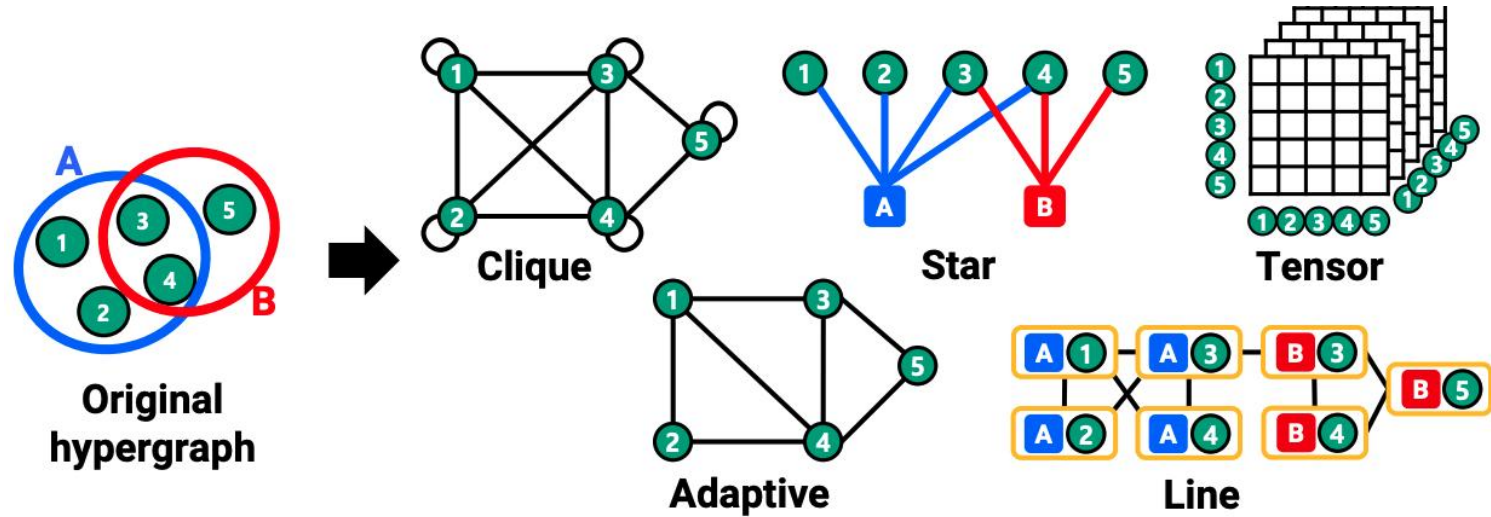


**Readily obtainable
& Scalable**



Non-inductive

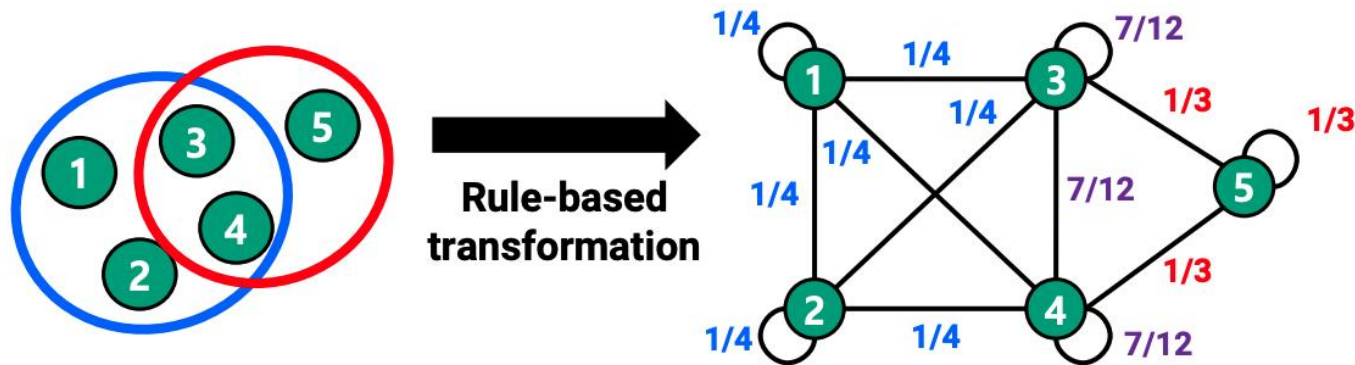
Encoding design – Step 2. Express Hypergraphs to Reflect HOIs



Encoding design – Step 2. Express Hypergraphs to Reflect HOIs

Clique expansion of a hypergraph is a homogeneous, pairwise graph.

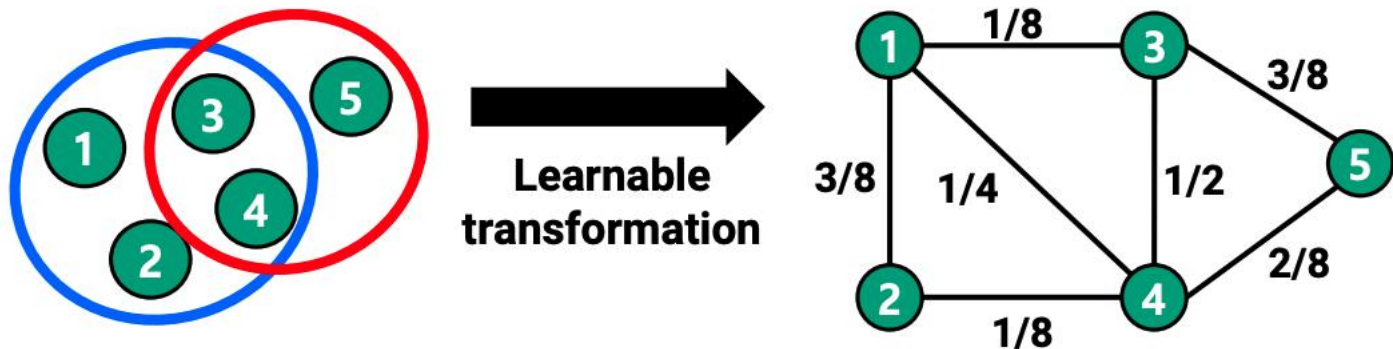
- Each hyperedge is converted into a clique of its member nodes.
- Optionally, self-loops can be added.
- Often, edges are weighted by the (normalized) count of hyperedges
Normalization considers the size of each hyperedge.
- Some work further weighed edges with learnable parameters



Encoding design – Step 2. Express Hypergraphs to Reflect HOIs

Adaptive expansion of a hypergraph is a homogeneous, pairwise graph.

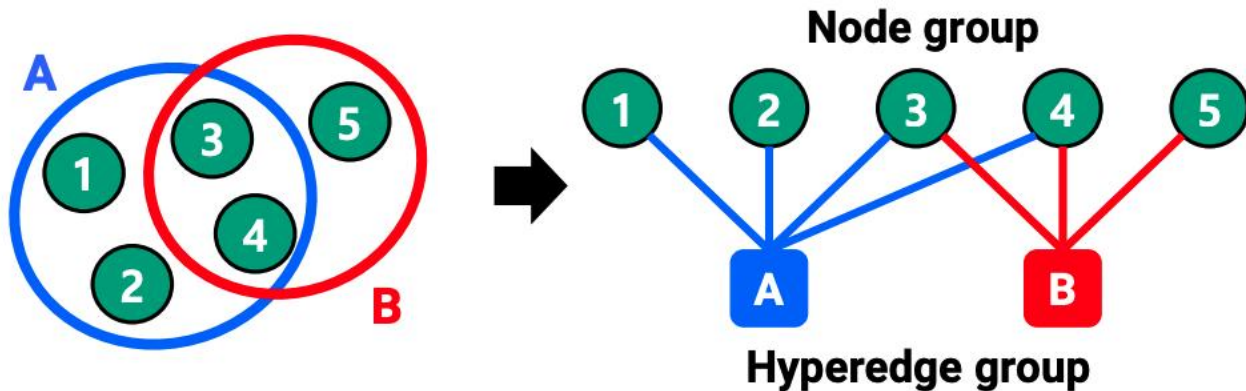
- Each hyperedge is converted into pairwise edge(s) via learnable rules.
- some work created and weighed edges based on node features.



Encoding design – Step 2. Express Hypergraphs to Reflect HOIs

Star expansion of a hypergraph is a bipartite, pairwise graph.

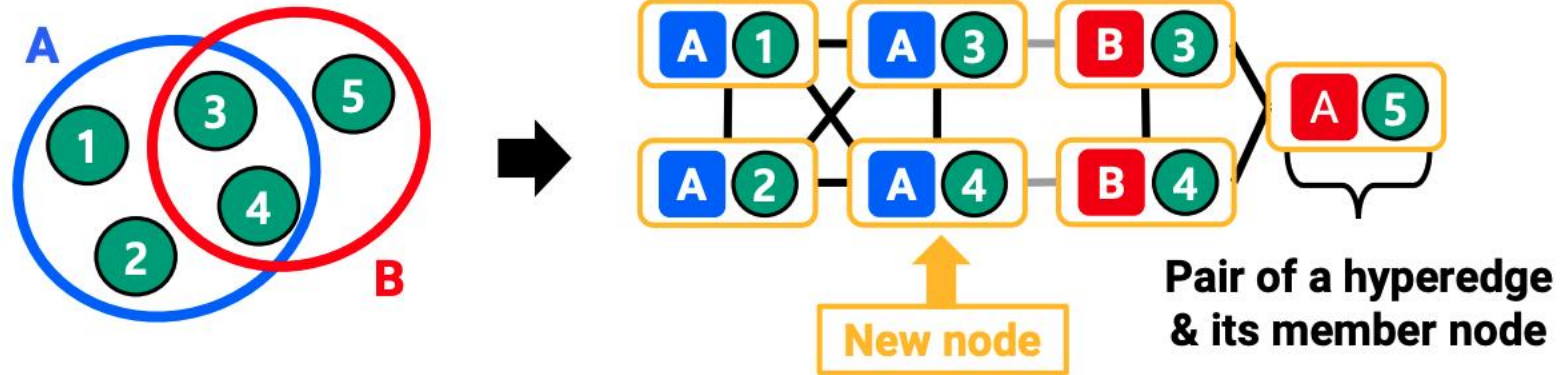
- Each hyperedge is converted into a new node.
- Each hyperedge (i.e., new node) and each of its member nodes are joined by a pairwise edge.



Encoding design – Step 2. Express Hypergraphs to Reflect HOIs

Line expansion of a hypergraph is a homogeneous, pairwise graph.

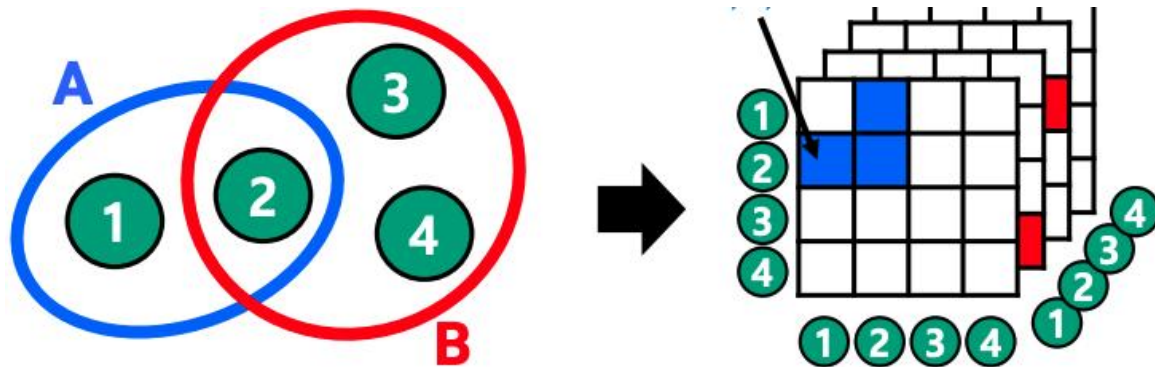
- Each pair of a hyperedge and its node is converted into a new node.
- Two new nodes are joined by a pairwise edge if they share a hyperedge or a node.



Encoding design – Step 2. Express Hypergraphs to Reflect HOIs

A hypergraph can be represented as a **tensor**

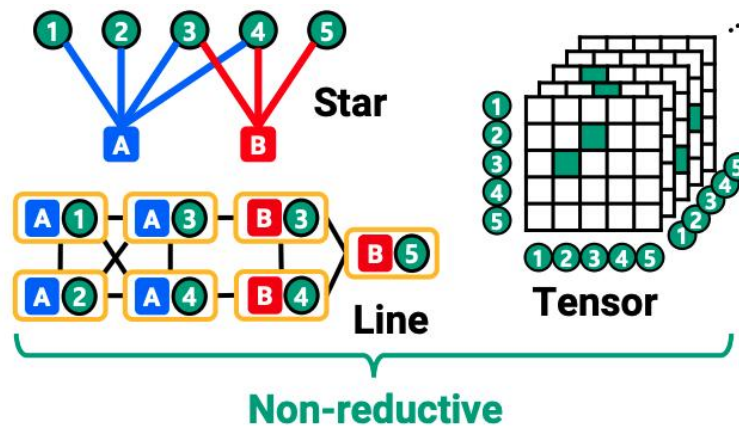
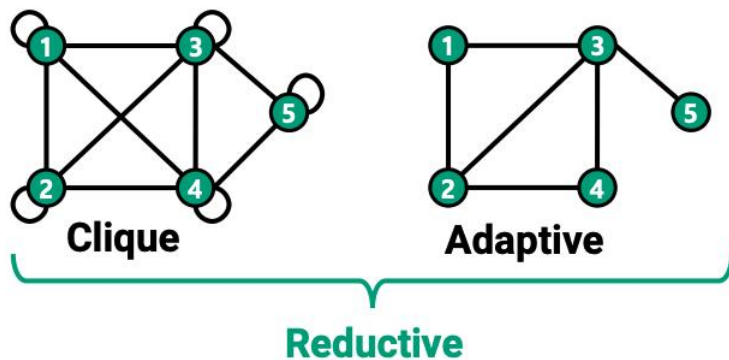
- The order of the tensor equals the maximum hyperedge size
- The dimensionality of each mode equals the node count.
- Each tensor entry is non-zero if there exists a hyperedge containing all its mode indices.



Encoding design – Step 2. Express Hypergraphs to Reflect HOIs

These methods are either **reductive** or **non-reductive**.

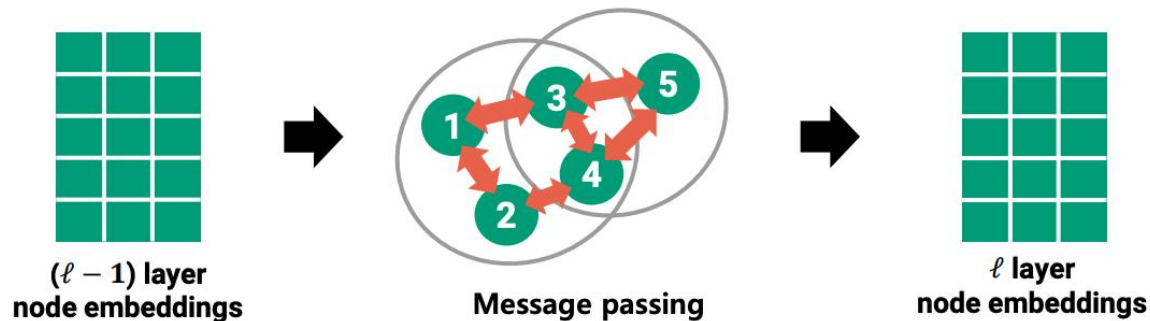
- Reductive methods may incur **information loss** after transformation
- However, they provide simple and straightforward graph structures.
- Non-reductive methods incur no information loss.
- However, they are often more complex and, thus, difficult to handle.



EPFL

Encoding design – Step 3. Pass Message to Reflect HOIs - Overview

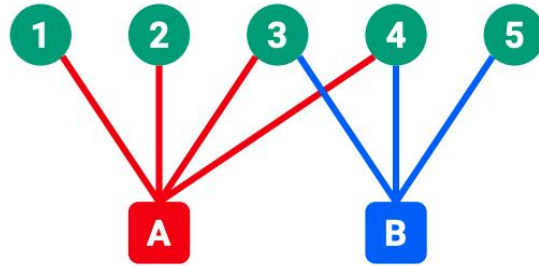
- HNNs learn **node (and hyperedge)** embeddings by **aggregating information** from other nodes (and hyperedges).
- This process is called message passing.
- In HNNs' message passing, some of the key issues involve:
 - Q1) **Whose** messages to aggregate
 - Q2) **What** messages to aggregate
 - Q3) **How** to aggregate messages



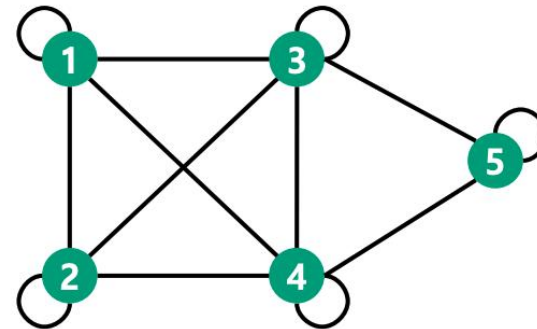
Encoding design – Step 3. Pass Message to Reflect HOIs

Target Selection can be determined by how the input hypergraph is expressed.

- The **star-expansion** transforms a hypergraph into a bipartite graph.
 - Two groups of nodes: Node group and Hyperedge group with **two-stage passing**
 - From the node group to the hyperedge group
 - From the hyperedge group to the node group
- **Clique-expansion** transforms a hypergraph into a weighted graph.
 - Akin typical GNNs, it perform message passing between nodes.



Star-expanded graph

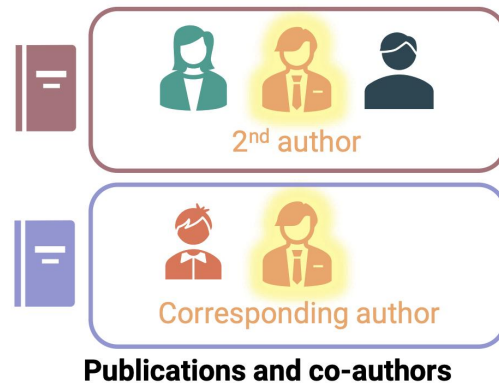
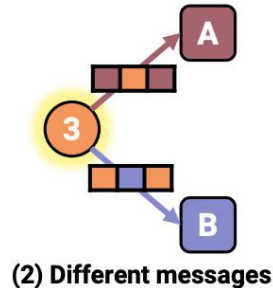
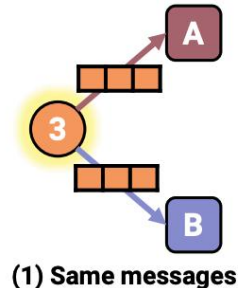
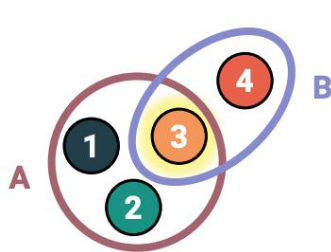


Clique-expanded graph

Encoding design – Step 3. Pass Message to Reflect HOIs

What messages to aggregate. Possible message representations:

- Hyperedge-consistent messages:
 - the node representation remains the same across all aggregations.
- Hyperedge-dependent messages:
 - The role of a node may vary based on the hyperedges it is involved in

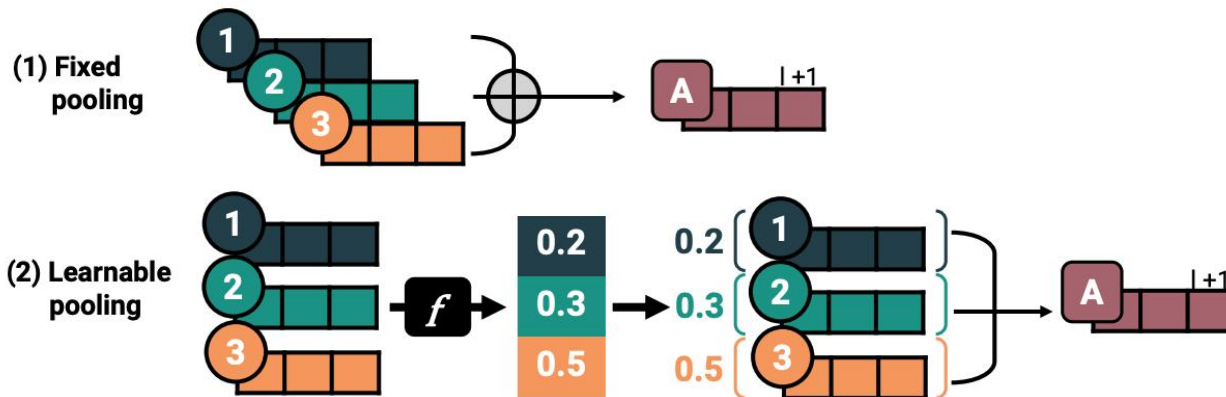
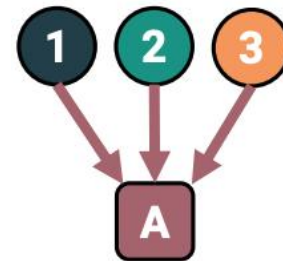


Publications and co-authors

Encoding design – Step 3. Pass Message to Reflect HOIs

Aggregate Function

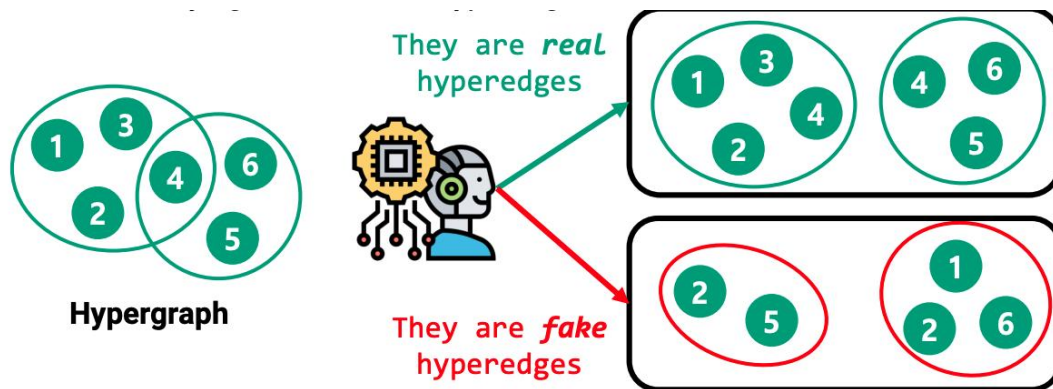
- Fixed pooling function
 - Notable examples are **summation** or **average**.
- Learnable pooling function
 - **adaptively** aggregate node/hyperedge messages
 - **attention** mechanism is widely used to assign **different weights** to message



Encoding design – Learning to Classify

Hyperedge prediction (a.k.a. Link Prediction) is a task that **predicts** future or unobserved hyperedges

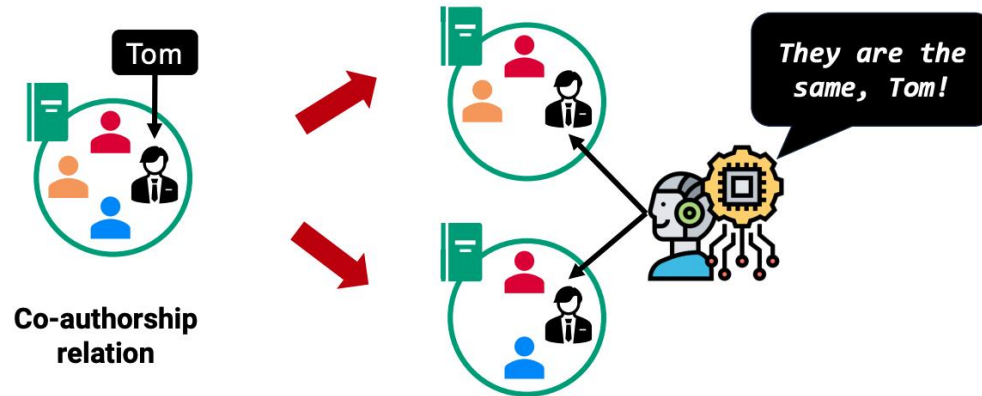
- **Classifying** real and fake hyperedges
- Representative fake hyperedges are crucial
 - heuristic fake hyperedge generation
 - learnable fake hyperedge generation



Training design – Learning to contrast

Contrastive learning (CL) aims to contrast multiple **views** of a hypergraph.

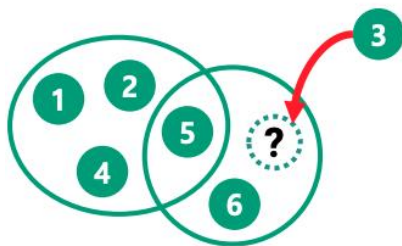
- View **generation** and **encoding**
- Construct contrastive **loss**:
 - maximize the embedding similarity of identical nodes/edges
 - minimize the embedding similarity of different ones
 - Node-level (node - node) contrastive losses
 - Hyperedge-level (hyperedge - hyperedge) contrastive losses



Training design – Learning to Generate

Trains the network to **generate hyperedges** that reflect the characteristics of the input data.

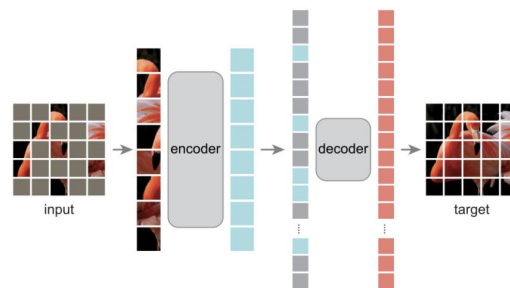
- **Generation** is an effective SSL strategy for neural networks.
- GPT [OpenAI, 2023] and MAE [He et al., 2022] are notable examples.
- In generative SSL, two major aspects should be focused on:
- Q1) **What** to generate: Generative task
- Q2) **How** to generate: Generative method



Generative SSL



GPT in NLP



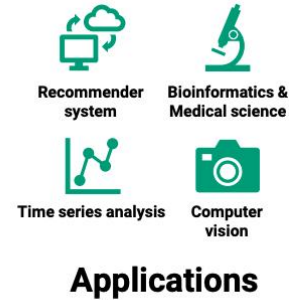
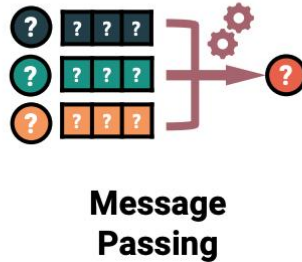
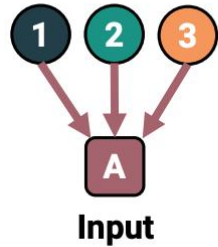
MAE in Computer Vision

Application of HNNs

- Recommendation systems: recommend products to customers based on their past purchases and preferences
- Bioinformatics and medical science: predict the structure of proteins and other biological molecules
- Time series analysis: forecast taxi demands, gas pressures, vehicle speeds, traffic, electricity consumptions, meteorological measures, stocks, and crimes.
- Computer vision: recognize objects in images

Take Homes

- HNNs are a type of neural network that's specifically designed to work with hypergraphs.
- They're able to learn from the structure of the graph and make predictions about the relationships between the different nodes.
- They're being used in a lot of different applications, including recommendation systems, bioinformatics, and computer vision.



Thank You



Week 6:

Hypergraph Neural Networks: Application

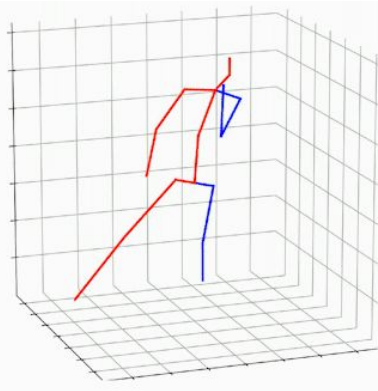
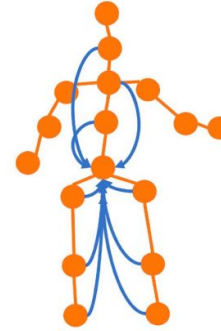
Presented by Ti Wang



Previous Research as a Master's Student

3D Human Pose and Shape Estimation:

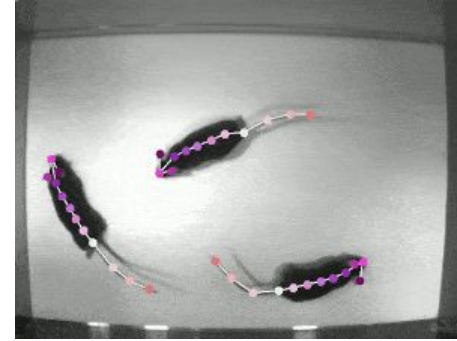
- Transformer-based Methods
- Graph-based Methods



- ❑ <https://github.com/Vegetebird/MHFormer/tree/main>
- ❑ <https://github.com/kasvii/PMCE/tree/main>

Current Research as a PhD Student in EDEE

- Focusing on 2D/3D animal pose estimation
- Foundation Model for Animal



<https://github.com/DeepLabCut/DeepLabCut?tab=readme-ov-file>
<https://actu.epfl.ch/news/unifying-behavioral-analysis-through-animal-founda/>

Hypergraph factorization for multi-tissue gene expression imputation

Ramon Viñas, Chaitanya K. Joshi, Dobrik Georgiev, Phillip Lin, Bianca Dumitrascu, Eric R. Gamazon, Pietro Liò——Nature Machine Intelligence, 2023

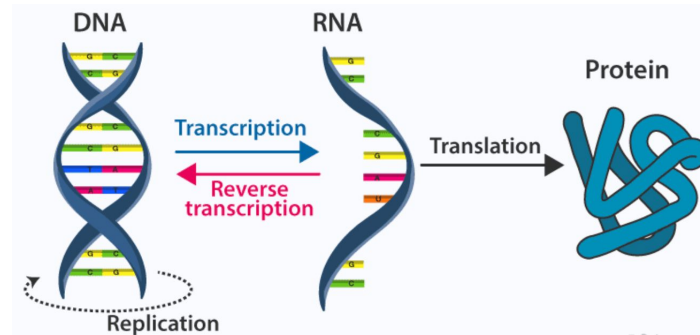
<https://www.nature.com/articles/s42256-023-00684-8>

What is Gene Expression?

- Gene expression is how cells use information from genes to create proteins.
- Every tissue (like the brain or liver) uses different genes to perform its specific functions.

Why It Matters

- Studying gene expression across tissues helps us understand diseases and develop treatments.



The Problem: Missing Data

- It's hard to collect gene expression data from all tissues, especially difficult ones like brain tissue.
 - This leaves gaps in understanding how different tissues work together.
- Genotype data is not always available due to privacy concerns.
- Tissues collected for different individuals vary, leading to gaps in data.

Traditional Methods

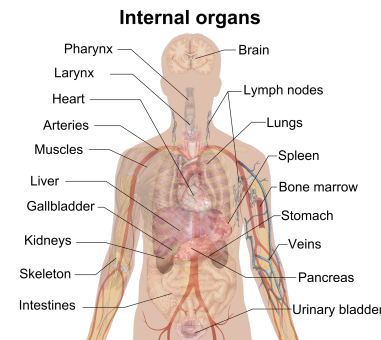
- Traditional ways to fill these gaps depend on genetic data, which may not always be available for privacy or technical reasons.
- Existing methods cannot capture complex, non-linear relationships between tissues.

What is the Task?

- predict gene expression in tissues where data is missing or difficult to collect (e.g., brain, heart).
- fill in these gaps using data from tissues that are easier to collect (e.g., blood, skin).

Challenges

- **Gene expression varies across tissues**
 - Each tissue has a unique **gene expression pattern** based on its function. Understanding this helps us study how tissues work together and how diseases progress.
- **Limited access to some tissues:**
 - Certain tissues are hard or invasive to sample (like brain tissue), leaving gaps in our understanding of these critical tissues.
- **Nonlinear and Complex Relationships:** Gene expression across tissues isn't a simple 1-to-1 relationship. The relationships between different tissues are complex and nonlinear.



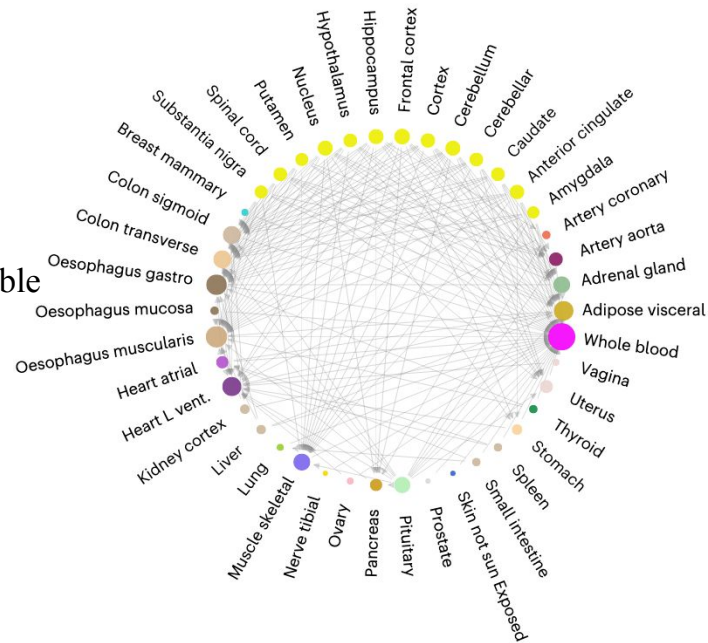
Why Use a Hypergraph for Gene Expression Imputation?

Limitations of Existing Approaches:

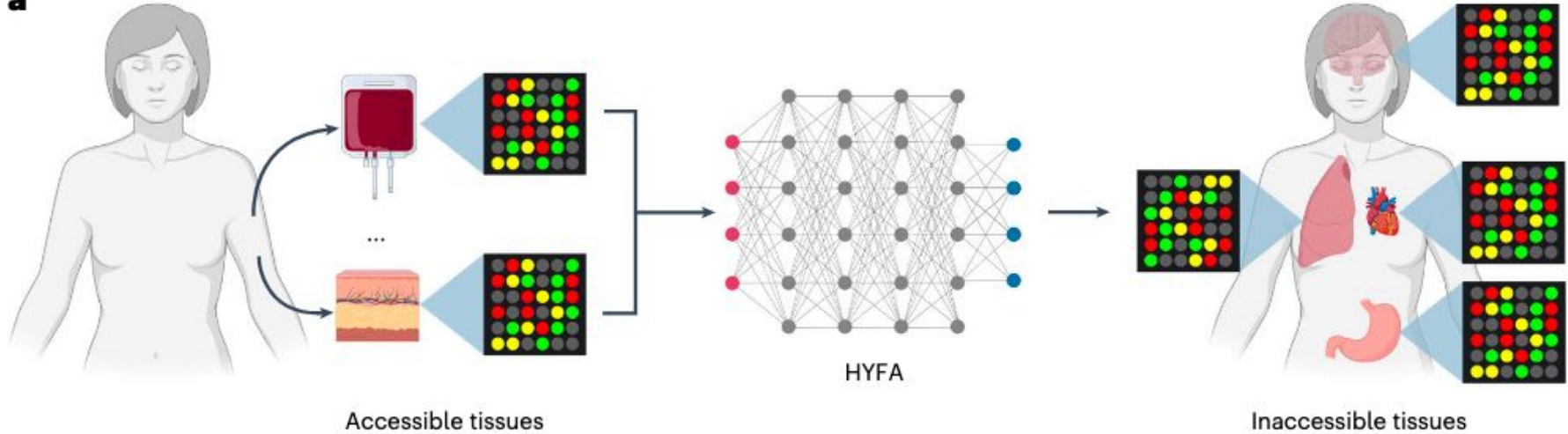
- **Linear methods** (like PCA and regression) fail to capture the **nonlinear** relationships between tissues.
- These approaches rely heavily on **genotype data**, which is often unavailable due to privacy or technical issues.

Benefits of Hypergraph:

- **Higher-Order Relationships:** Hypergraphs can model **complex interactions** between multiple tissues, genes, and individuals all at once.
- **Nonlinear Patterns:** Unlike traditional graphs, a hypergraph can capture **nonlinear gene expression patterns** across different tissues.
- **Flexibility:** Supports a **variable number of tissues**, allowing for input flexibility when not all tissue data is available.



HYFA: Hypergraph Factorization for Gene Expression Imputation

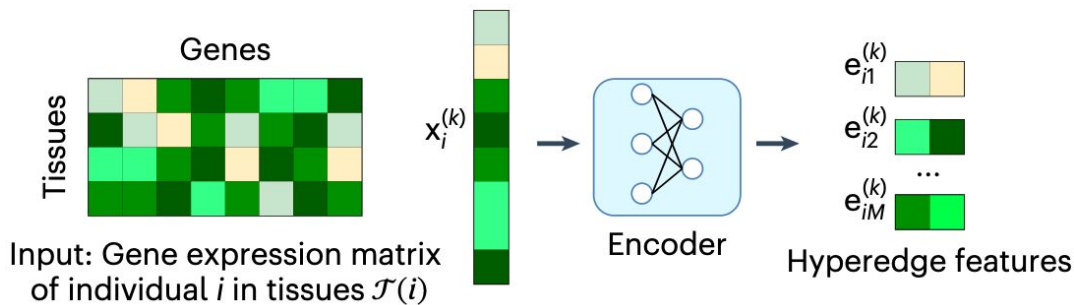
a

What is HYFA?

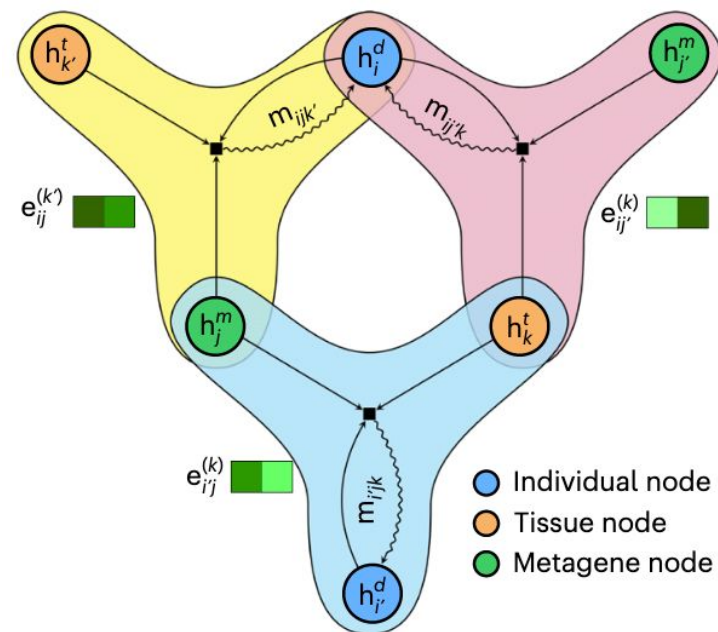
- HYFA (Hypergraph Factorization) is a method for imputing missing gene expression across multiple tissues.
- It uses a **hypergraph** to model complex relationships between individuals, tissues, and genes.

Workflow of HYFA

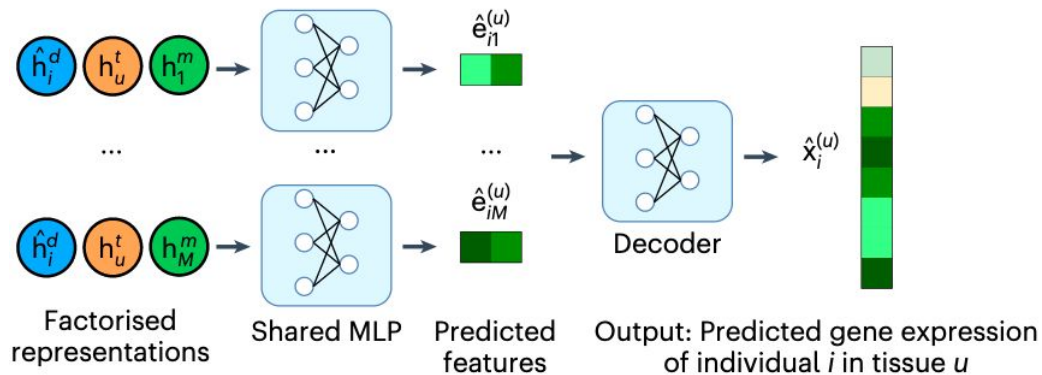
1. Encoder learns low-dimensional representation



2. Message passing computes factorised representations

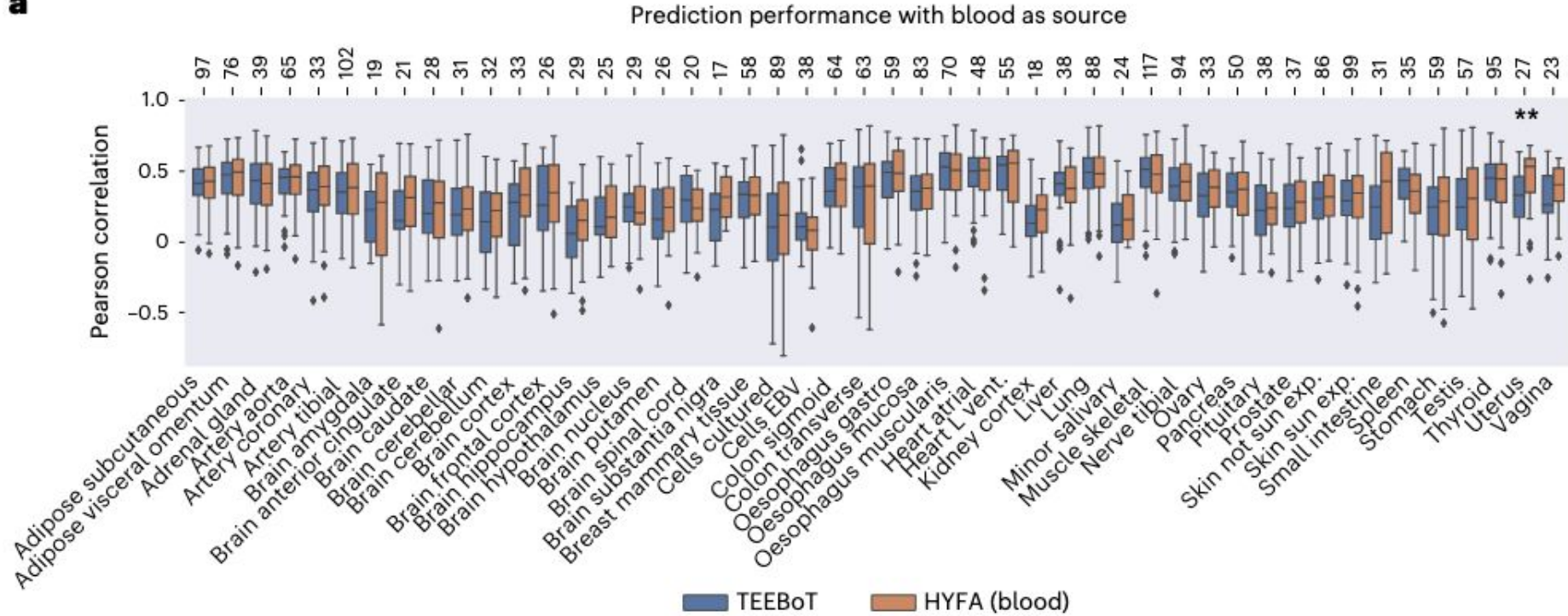


3. Decoder recovers gene expression of uncollected tissue u



Prediction Performance

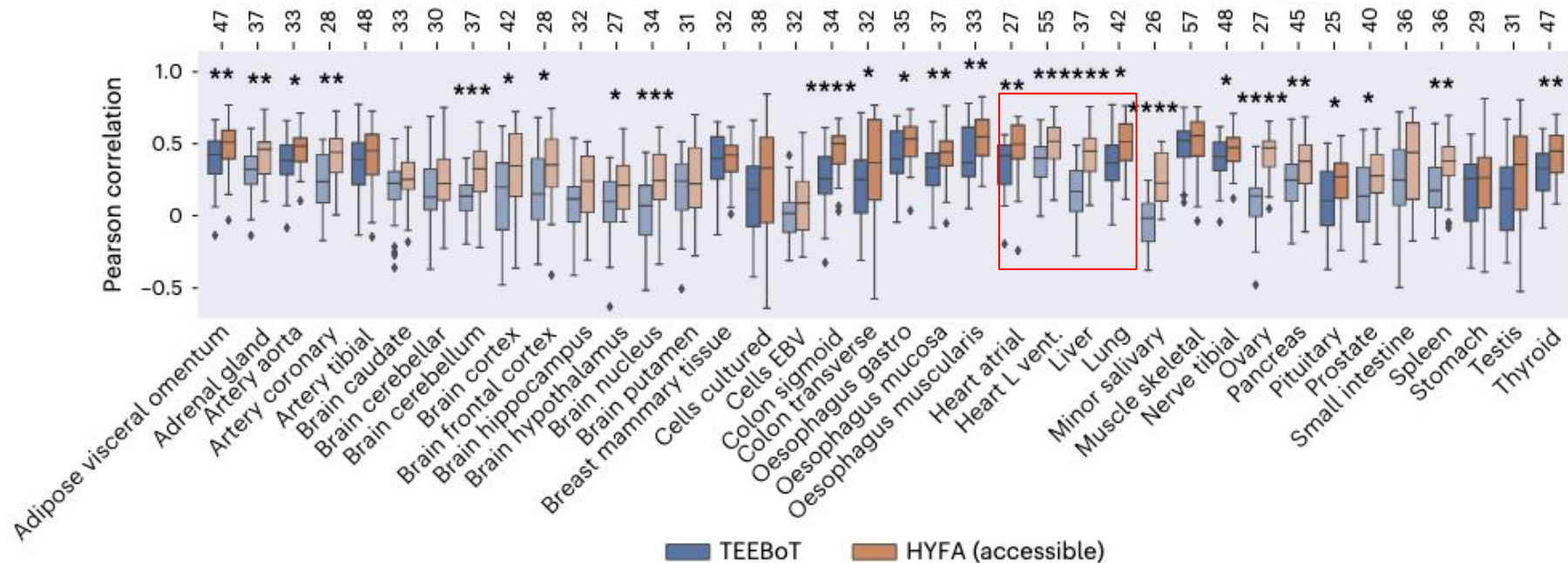
a



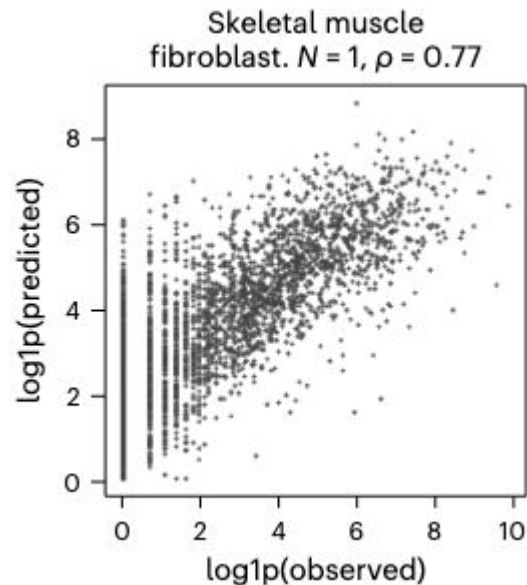
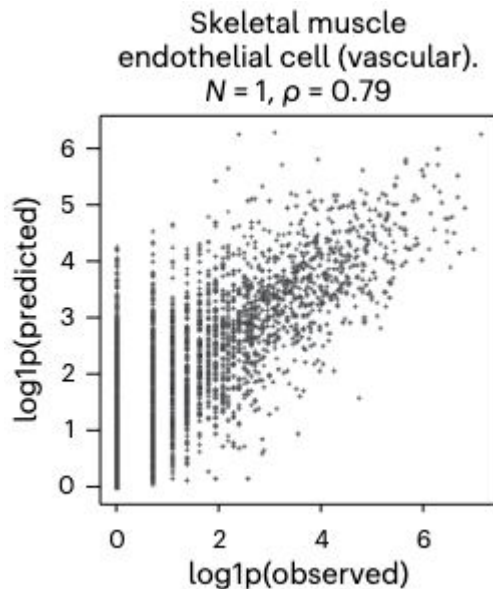
Prediction Performance

b

Prediction performance with accessible tissues as source tissues (whole blood, skin and adipose subcutaneous)



Cell-Type Specific Gene Expression



- HYFA accurately predicts gene expression for specific cell types, even in tissues not used for training.
- These results demonstrate the **effectiveness of HYFA's factorized tissue representations**, allowing it to accurately infer gene expression at the **cell-type level**.

Summary

Problem Solved:

- HYFA addresses the challenge of imputing **missing gene expression** in hard-to-reach tissues by leveraging data from accessible tissues, without relying on genotype information.

Key Features of HYFA

- **Hypergraph Structure:** Captures **higher-order relationships** across tissues, genes, and individuals, allowing more accurate predictions.
- **Cell-Type Specific Predictions:** HYFA can infer **cell-type-specific gene expression** even for tissues not used during training, as demonstrated with vascular endothelial cells and fibroblasts.

Superior Performance:

- HYFA consistently **outperforms traditional methods** (like TEEBoT), especially when using multiple reference tissues.
- Achieves strong correlations for difficult-to-predict cell types, highlighting its robustness and accuracy.

Broader Implications:

- HYFA's ability to infer accurate gene expression across tissues and cell types is crucial for **precision medicine** and understanding complex diseases.

<https://github.com/rvinas/HYFA>

Hypergraph Transformers for EHR-based Clinical Predictions

Ran Xu, Mohammed K. Ali, Joyce C. Ho, Carl Yang——AMIA Jt Summits Transl Sci Pro, 2023

What are Electronic Health Records (EHR)?

- Digital records of patient health information collected over time.
- Include data such as:
 - Diagnoses, treatments, and medications.
 - Medical procedures, lab test results, and doctor's notes.
 - Vital signs, allergies, and medical imaging results.



Growing Importance of EHRs

- EHR systems are now standard in most healthcare facilities, providing large, diverse datasets.
- Used extensively for clinical decision-making and research, particularly in personalized medicine.

Example of EHR Usage:

- EHRs help track the **progress of chronic diseases** like diabetes and heart disease, allowing healthcare providers to monitor **long-term health trends** across multiple visits.

Challenges with EHR Data

- **Diverse and Complex Data:** Each patient visit contains multiple medical codes (e.g., diagnoses, medications), creating a **high-dimensional and heterogeneous dataset**.
- **Irregularity and Sparsity:** EHR data is often incomplete, with irregular time intervals and missing information.

Limitations of Traditional Models

- **Pairwise Relations:** Graph-based models capture only pairwise interactions between medical codes, ignoring the broader context of co-occurring codes.
- **Expert-Defined Rules:** Rule-based systems are **labor-intensive** and **lack generalizability** across datasets.

Need for Accurate Models

- Effective EHR modeling is crucial for improving personalized medicine and population health strategies.

Main Task

- Predict patient outcomes (e.g., disease progression, cardiovascular risk) using **Electronic Health Records (EHR)** data.

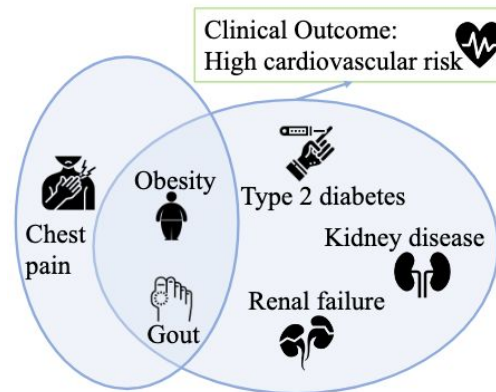
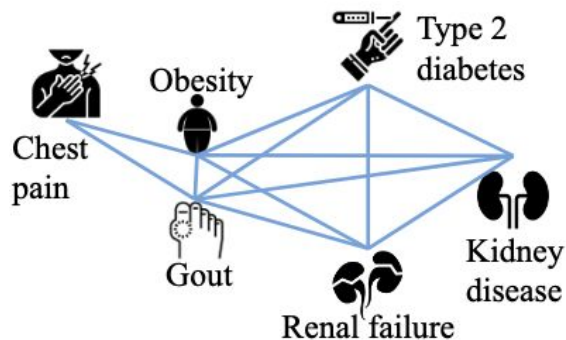
Data Structure:

- Each patient visit contains multiple **medical codes** (diagnoses, medications, procedures).
- These codes co-occur and interact in complex ways, requiring advanced methods to capture relationships.

Goal

- Accurately model the interactions among medical codes within each visit to support clinical outcome predictions.

Why Use Hypergraphs for EHR Data?



Limitations of Traditional Graphs:

- Traditional graphs only capture **pairwise** relationships between medical codes, missing the **higher-order** interactions crucial in complex patient visits.
- Example: a patient visit with multiple diagnoses (e.g., diabetes, kidney disease) is reduced to simple pairwise edges, losing the broader context.

Benefits of Hypergraphs:

- Hypergraphs can capture **higher-order interactions**, where a single visit connects multiple medical codes simultaneously.
- Example: all relevant medical codes (nodes) in a visit are connected via a **hyperedge**, preserving the full context and allowing more accurate predictions.

Model Overview

- HypEHR is a **hypergraph-based model** aimed at predicting clinical outcomes from EHR data by capturing complex interactions between medical codes within a patient visit.

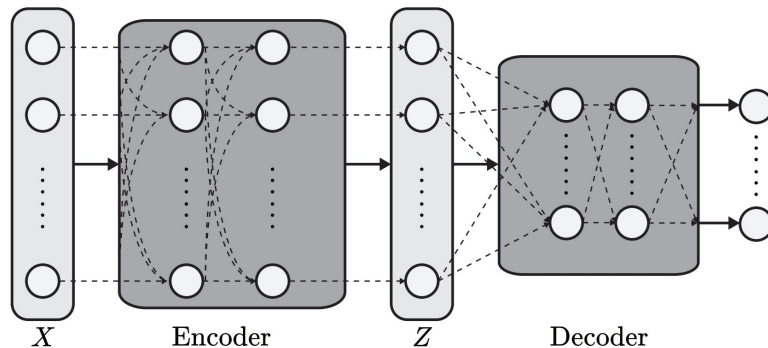
Hypergraph Construction

- Medical codes (diagnoses, procedures, medications) are represented as **nodes**.
- Each patient visit forms a **hyperedge**, connecting all the relevant medical codes.
- This preserves both the relationships between the codes and the context of the visit, providing a richer data structure for predictions.

How HypEHR Solves EHR Challenges

- **Higher-order Interactions:** Captures relationships that occur between multiple codes, not just pairs.
- **Flexible Representation:** Handles the diverse and irregular nature of EHR data by using hypergraphs, which are better suited than traditional pairwise graphs.

HypEHR: Hypergraph Transformer Model



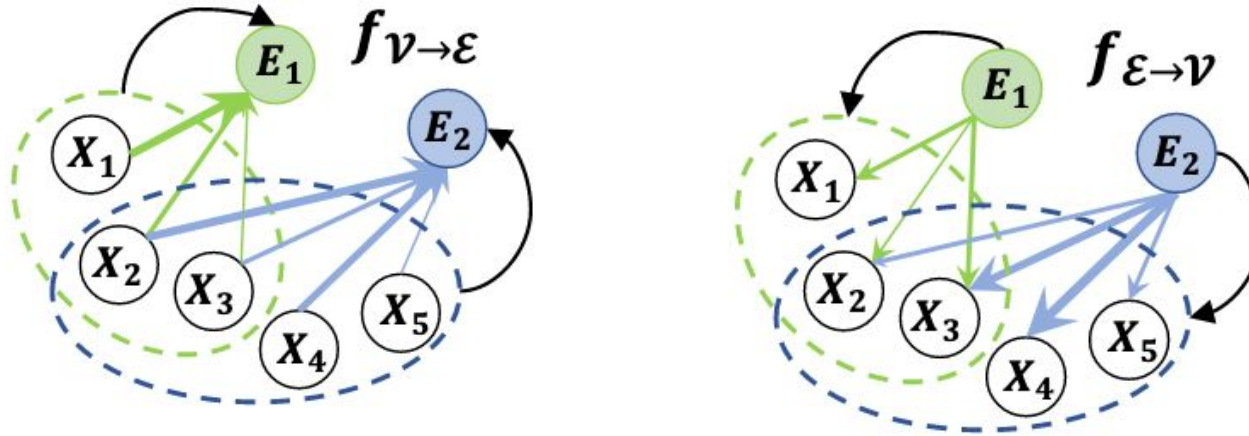
Set Transformer

- Used to model the relationships between medical codes (nodes) and patient visits (hyperedges).
- Can handle sets of any size, making it flexible for various patient records.

Self-Attention Mechanism

- HypEHR employs **self-attention** to identify the **most important medical codes** within a patient visit.
- By assigning **higher weights** to relevant medical codes, it filters out irrelevant ones, improving the model's focus on clinically significant information.
- This also makes the model more **interpretable**, allowing us to gain insights into which medical codes are the most important for specific predictions.

HypEHR: Hypergraph Transformer Model



Message Passing (Aggregation)

- Two key steps:
 - Node to Hyperedge Aggregation** ($f_{v \rightarrow \epsilon}$): Information from individual medical codes (nodes) is aggregated to form **hyperedge (visit) embeddings**.
 - Hyperedge to Node Aggregation** ($f_{\epsilon \rightarrow v}$): The hyperedge embeddings are then used to update the representations of the **medical codes (nodes)**.
- This process helps the model understand how **medical codes interact** within the broader context of a visit, leading to more accurate predictions.

Results

Model	MIMIC-III				CRADLE			
	ACC	AUROC	AUPR	F1	ACC	AUROC	AUPR	F1
LR ³⁶	68.66 ± 0.24	64.62 ± 0.25	45.63 ± 0.32	13.74 ± 0.40	76.22 ± 0.30	57.22 ± 0.28	25.99 ± 0.26	42.18 ± 0.35
SVM ³⁷	72.02 ± 0.12	55.10 ± 0.14	34.19 ± 0.17	32.35 ± 0.21	68.57 ± 0.13	53.57 ± 0.11	23.50 ± 0.15	52.34 ± 0.22
MLP ³⁸	70.73 ± 0.24	71.20 ± 0.22	52.14 ± 0.23	16.39 ± 0.30	77.02 ± 0.17	63.89 ± 0.18	33.28 ± 0.23	45.16 ± 0.26
GCT ²²	76.58 ± 0.23	78.62 ± 0.21	63.99 ± 0.27	35.48 ± 0.34	77.26 ± 0.22	67.08 ± 0.19	35.90 ± 0.20	56.66 ± 0.25
GAT ³⁹	76.75 ± 0.26	78.89 ± 0.12	66.22 ± 0.29	34.88 ± 0.33	77.82 ± 0.20	66.55 ± 0.27	36.06 ± 0.18	56.43 ± 0.26
HyperGCN ¹²	78.01 ± 0.23	80.34 ± 0.15	67.68 ± 0.16	39.29 ± 0.20	78.18 ± 0.11	67.83 ± 0.18	38.28 ± 0.19	60.24 ± 0.21
HCHA ¹³	78.07 ± 0.28	80.42 ± 0.17	68.56 ± 0.15	37.78 ± 0.22	78.60 ± 0.15	68.05 ± 0.17	39.23 ± 0.13	59.26 ± 0.21
HypEHR	79.07 ± 0.31*	82.19 ± 0.13*	71.08 ± 0.17*	41.51 ± 0.25*	79.76 ± 0.18*	70.07 ± 0.13*	40.92 ± 0.12*	61.23 ± 0.18*

Datasets:

- **MIMIC-III:** Predicting 25 clinical phenotypes based on patient data from ICU visits.
- **CRADLE:** Predicting cardiovascular disease (CVD) risk for diabetic patients.

Performance Metrics: Accuracy (ACC), Area Under ROC Curve (AUROC), and Area Under Precision-Recall Curve (AUPR).

Key Results

- HypEHR significantly outperforms traditional models like Logistic Regression (LR), SVM, and Graph Neural Networks.

Key Contributions

- HypEHR introduces a **hypergraph-based approach** to model complex interactions in EHR data, outperforming traditional models.
- The use of **self-attention** allows the model to identify the most relevant medical codes, improving both accuracy and interpretability.

Significant Results

- **Improved performance:** HypEHR achieves notable improvements in both **AUROC** and **AUPR**, outperforming traditional and hypergraph-based baselines.

Broader Implications

- The ability to model higher-order interactions makes **HypEHR** applicable to other domains with complex, multi-relational data, such as **social networks** and **biological networks**.

Future Work

- Potential extensions include incorporating **chronological information** to better model time-sequences in EHR data or applying HypEHR to other prediction tasks beyond clinical outcomes.

Thank You

