Fashion Recommendation: Outfit Compatibility Using GNN

CSE 6240 Web Search & Text Mining

Saksham Arora, Yeojin Chang,
Samaksh Gulati, Suraj Shourie

Outline

Introduction & Motivation

Problem Statement & Data

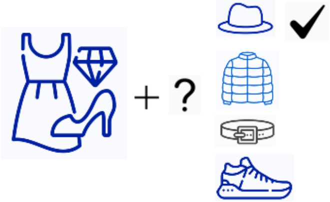
Approaches (Option 1.)

Experiments & Evaluation

Introduction

Is this a good outfit?

Which item would "complete" this outfit?



"Fashion companies that now embed AI into their businesses models could see a **118 percent cumulative increase in cash flow** by 2030. [1]

Example of compatible and incompatible outfits

[1] State of Fashion Technology Report 2022, May 2, 2022, Article by McKinsey & Company

Available Dataset

Polyvore Fashion Dataset

- Polyvore.com - platform for stylists to showcase their outfits

- 21,889 total outfits
- Each item belongs to 1 of 120 clothing/accessory categories

- Images, category information, titles, number of likes

Category	Image	Title
Coats		Irregular lapel blue trench coat
Knee Length Skirts		Beautiful cotton lace skirt yellow
Sandals		Mcq alexander mcqueen mirrored leather sandals
Necklaces		Navy blue yellow statement bib necklace earrings set

Example of outfit in Polyvore dataset

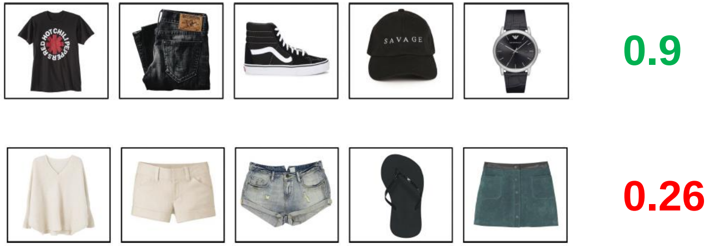
Problem Formulation and Tasks

Fill-In-The-Blank

- Given a set of fashion items (variable length), find most compatible item from a candidate set to fill in the blank
- Metric of evaluation: Accuracy

Outfit Compatibility Prediction

- Predict the compatibility score for any given outfit
- Metric of evaluation: AUC



Data Preparation

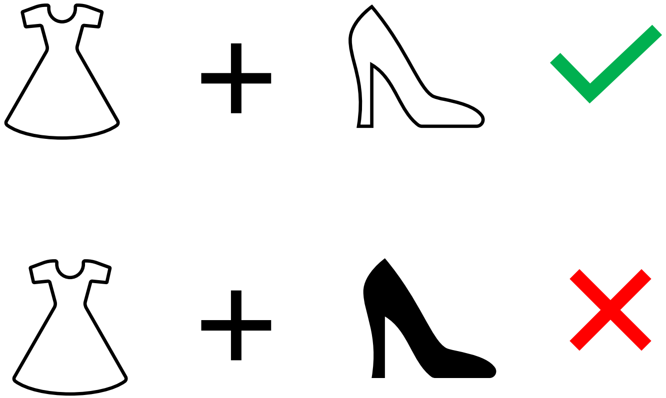
Polyvore Fashion Dataset

- Took a subset of 1600 outfits with 9,429 items
- 70%-30% split for training and test
- We create image embeddings and textual embeddings for each item.
- For Fill-in-the-Blank task, mask 1 item and randomly sample 4 negatives items
- For Outfit Compatibility task, for each actual outfit, create corresponding fake outfit pair with randomly sampled items

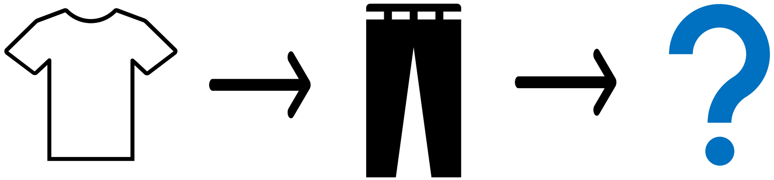
Problem Complexity

Initial solutions and Past work	Pitfalls
Recommend next item based on **simple rules** like colour palette match/fabric match/texture etc.	Hard to define a static & **consistent set of rules**
Use a **Neural Network to learn interactions** between items to score outfits	Different outfits have different number of items – **cannot use fixed input framework** of a simple Neural Networks
Use Sequential NNs like RNN/LSTM to learn interactions between items	**Introduces bias** as outfits don't inherently possess a sequence

Motivation for using Graphs

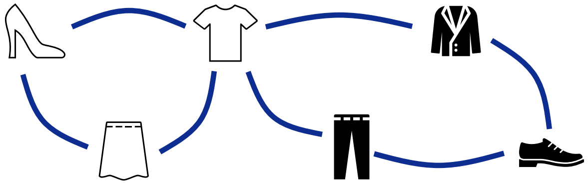


Rule Based



Sequential Representation

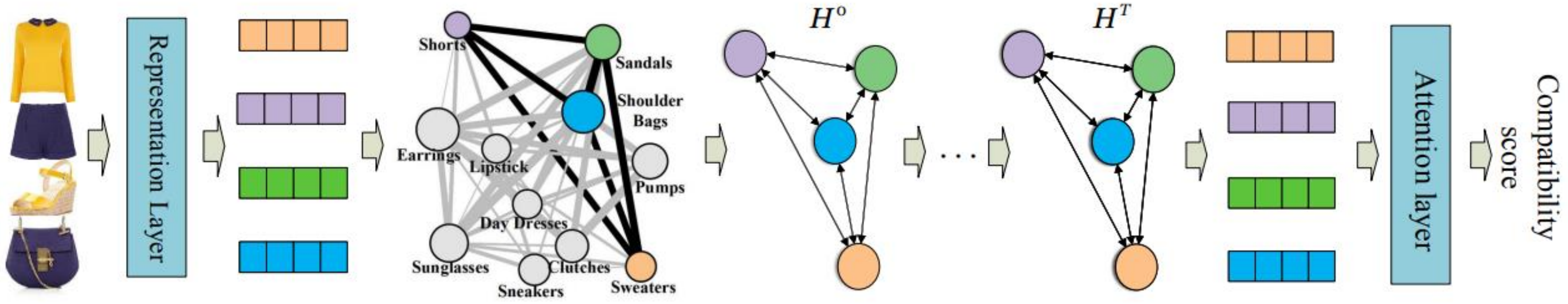
Approach 1
Node-wise Graph NN (NGNN)



Salient Features

- **Circumvents variable length** by representing outfits as subgraphs of a category graph
- **Category specific item mapping** for features of items
- **Weighted edge interaction** instead of parameter sharing used in GNN, here aggregation strategy is modified to use different weights for different category interactions.
- **Attention mechanism** to distil subgraph and compute compatibility score

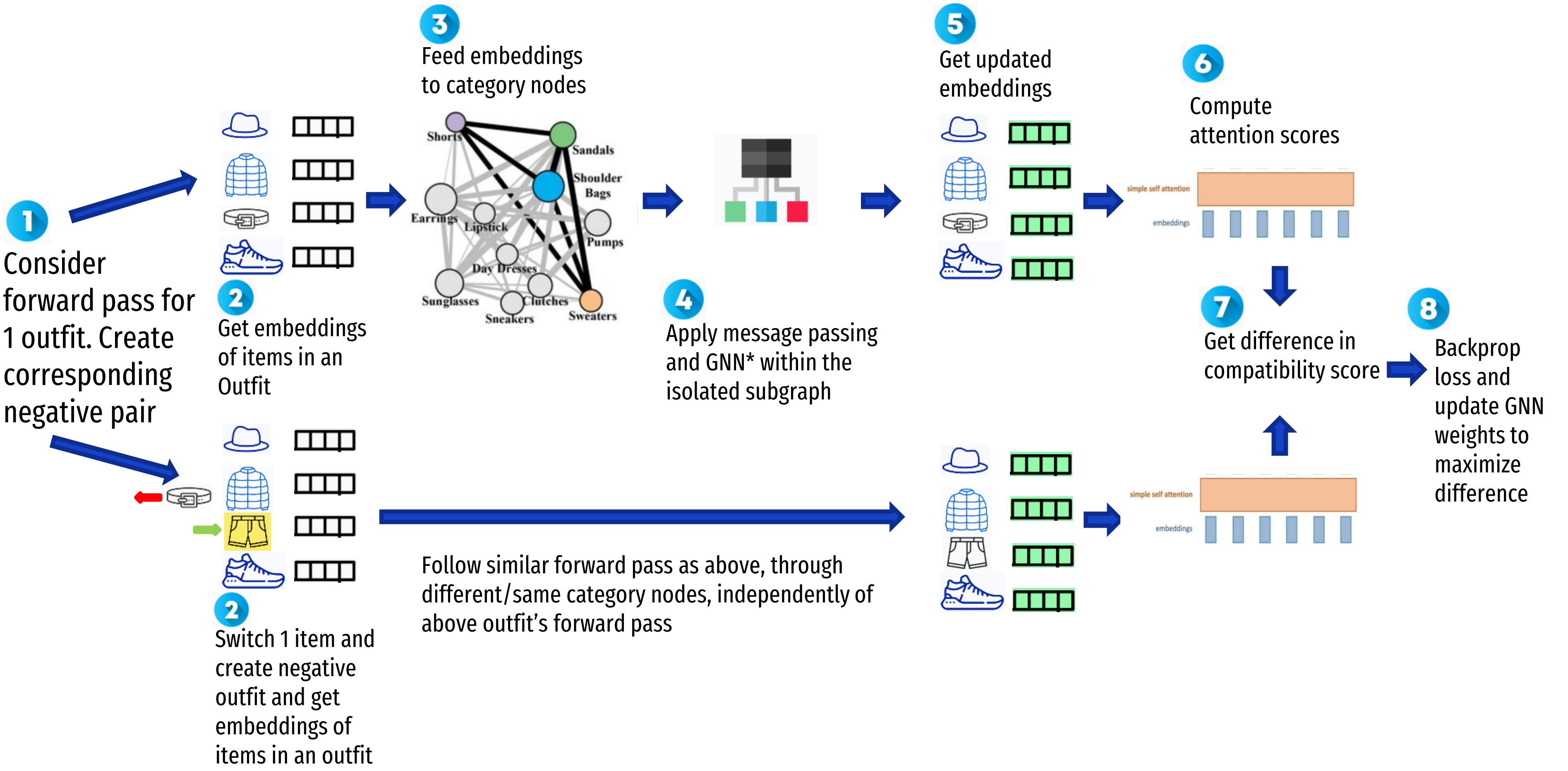
Overview of Method:



We create Fashion Graph where each node represents a category and each edge signifies the interactions between nodes.

Each outfit is treated as a sub-graph, where using image and text embeddings of the items, the NGNN model learns node representations. Finally, an attention layer is used to calculate the outfit compatibility score.

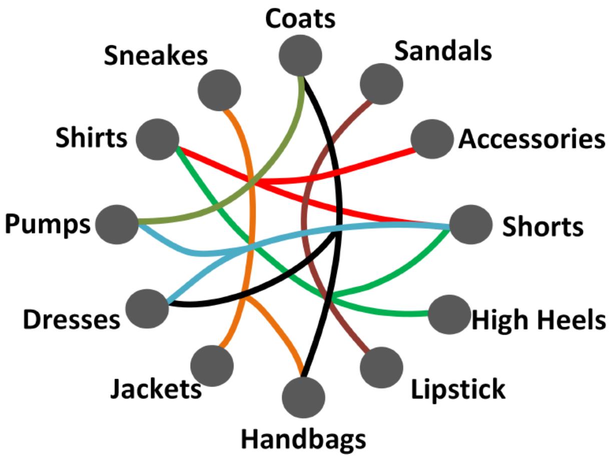
Training flow



Approach 2: Hypergraph Neural Networks

We explored how we can model even more complex relationship in the outfit?

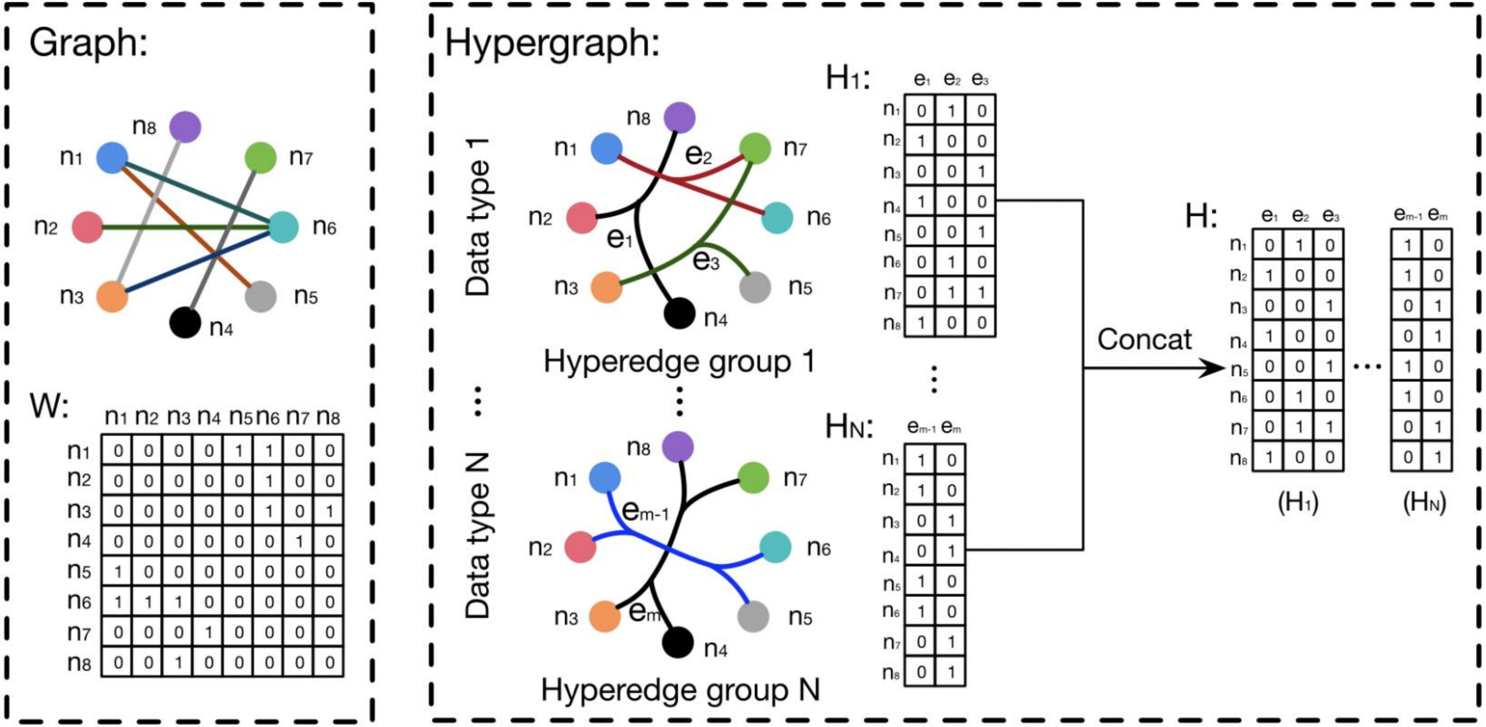
Salient features in addition to the NGNN:



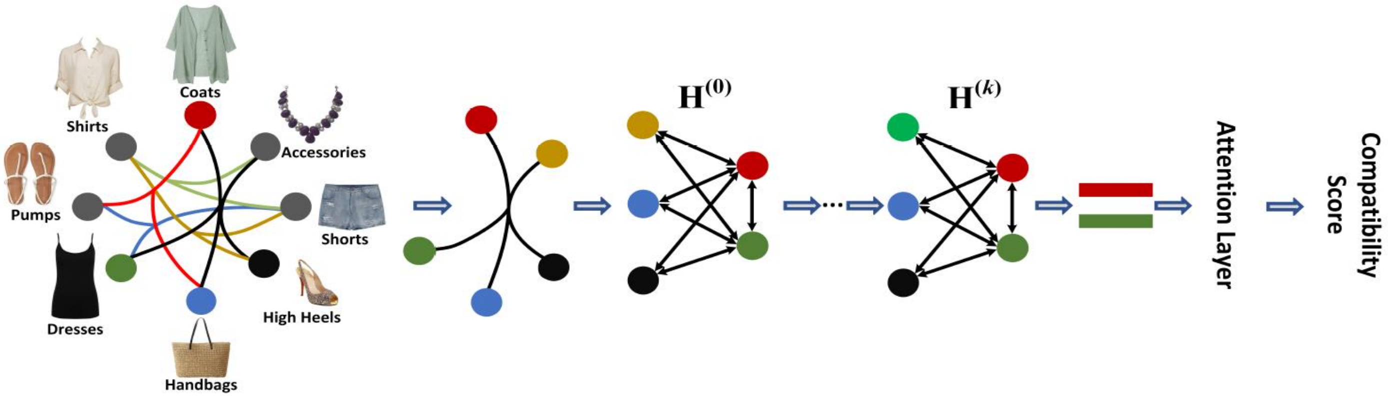
A hyper-egde can represents the matching interactions between multiple categories

- **Better Node Representation:** hyperedges in a hypergraph can connect any number of nodes.
- **Flexible** hypergraph structure has been employed to model high-order correlation among data

What is a Hypergraph?



Overview of Method:



HyperGraph Construction

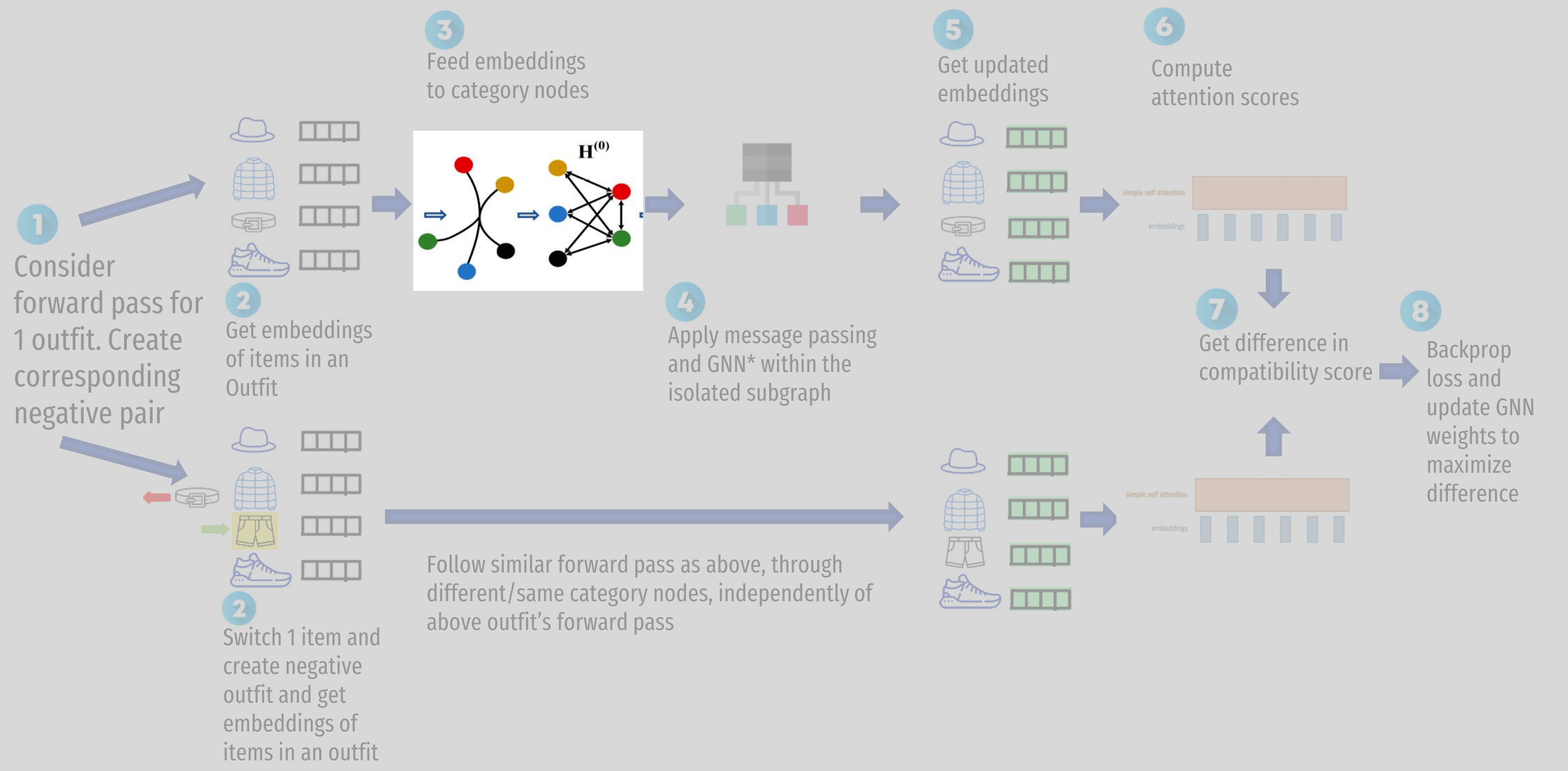
Node Initialization

Converting a Hyperedge to a Graph

State Propagation Updates the Node Status

Model Prediction

Training flow



Experiments and Evaluations

Experimental Setup:

- Replicated using TensorFlow 1.15.
- Hyper-parameters (learning, rate, λ, β, hidden size, propagation steps) were finalized using grid-search strategy.
- Experimented with the following:
  - Using of text and visual features
  - Using VIT image embeddings instead of inception
- We ran until convergence which was around 15 epochs.
- All experiments were run on COC ICE servers with 1 RTX6000 GPUs.

Evaluations:

- FITB Score:

In this task, Given a set of fashion items and a blank, we aim to find the most compatible item from the candidates set to fill in the blank. We compare FITB score across the two methods

- AUC Compatibility score:

We calculate AUC score and evaluate the two methods based on this in the next slide

RESULTS

Method	Accuracy (FITB)	AUC (Compatibility)
Random	24%	0.51
NGNN	38%	0.65
HGNN	39%	0.76
NGNN with VIT	40%	0.68
HGNN with VIT	40%	0.77

Method	Accuracy (FITB)	AUC (Compatibility)
NGNN (visual)	35%	0.62
NGNN (textual)	37%	0.64
NGNN (multi-modal)	38%	0.65

Performance Comparison

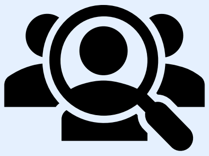
- HGNN performs slightly better than NGNN in both metrics.
- We also compare results for image embeddings created using VIT model and we notice that both metrics increase slightly.

Impact of different modalities

- For NGNN, multi-modal is the best approach, and text-only model is slightly better than the visual model.
- This happens because detailed textual descriptions are concise enough to capture key features of the items.

Future Work

User-specific Compatibility

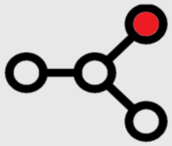


1. Assigning compatibility scores personal to a user
2. Determining style preferences

Better Evaluation Tasks

1. Current Fill-in-the-Blank task formulation might have irrelevant choices
2. For actual recommendation, needs evaluation of ALL items

Further Ablation Study



1. Examine affect of different added features of NGNN paper over vanilla GNN

2. Turn off Category specific feature transformation currently used

THANK YOU! :)