FPT UNIVERSITY

Prompt-Based Fashion Outfits Retrieval System

Author: Nguyen Quoc Dung, Pham Hoang Nam, Dao Duy Hung

Thesis Supervisor: Tran Van Ha

Notification of paper acceptance

AIABI

Dear authors,

We are pleased to inform you that your paper has been accepted for publication. The decision was made following the guidelines set by our review committee.

We kindly ask you to submit, by October 7th, the updated version of your paper based on the comments and reviews provided. You can find the instructions to participate in the workshop at the following link: http://www.aixia2023.cnr.it/registration.

Should you have any questions or concerns, please do not hesitate to contact us at: info@socialthingum.com.

Best regards

SUBMISSION: 16
TITLE: Prompt-Based Fashion Outfits Retrieval and Recommender System Using Binary Hashing

---------------------- REVIEW 1 ---------------------
SUBMISSION: 16
TITLE: Prompt-Based Fashion Outfits Retrieval and Recommender System Using Binary Hashing
AUTHORS: Quocdung Nguyen, Hoangnam Pham, Duyhung Dao, Quangmanh Do and Vanha Tran

----------- Overall evaluation -----------
SCORE: 2 (accept)
----- TEXT:
Interesting project. I feel that the paper misses a more detailed and technical description of the methods utilized (which are cited).

It is worth clarifying how the image embeddings are linked to the word embeddings. In particular, I do not see how in the training phase these components are optimized toward the same goal. It seems to me that the two models are trained separately optimizing the clustering of images and the text representation alone. I suggest to clarify this part

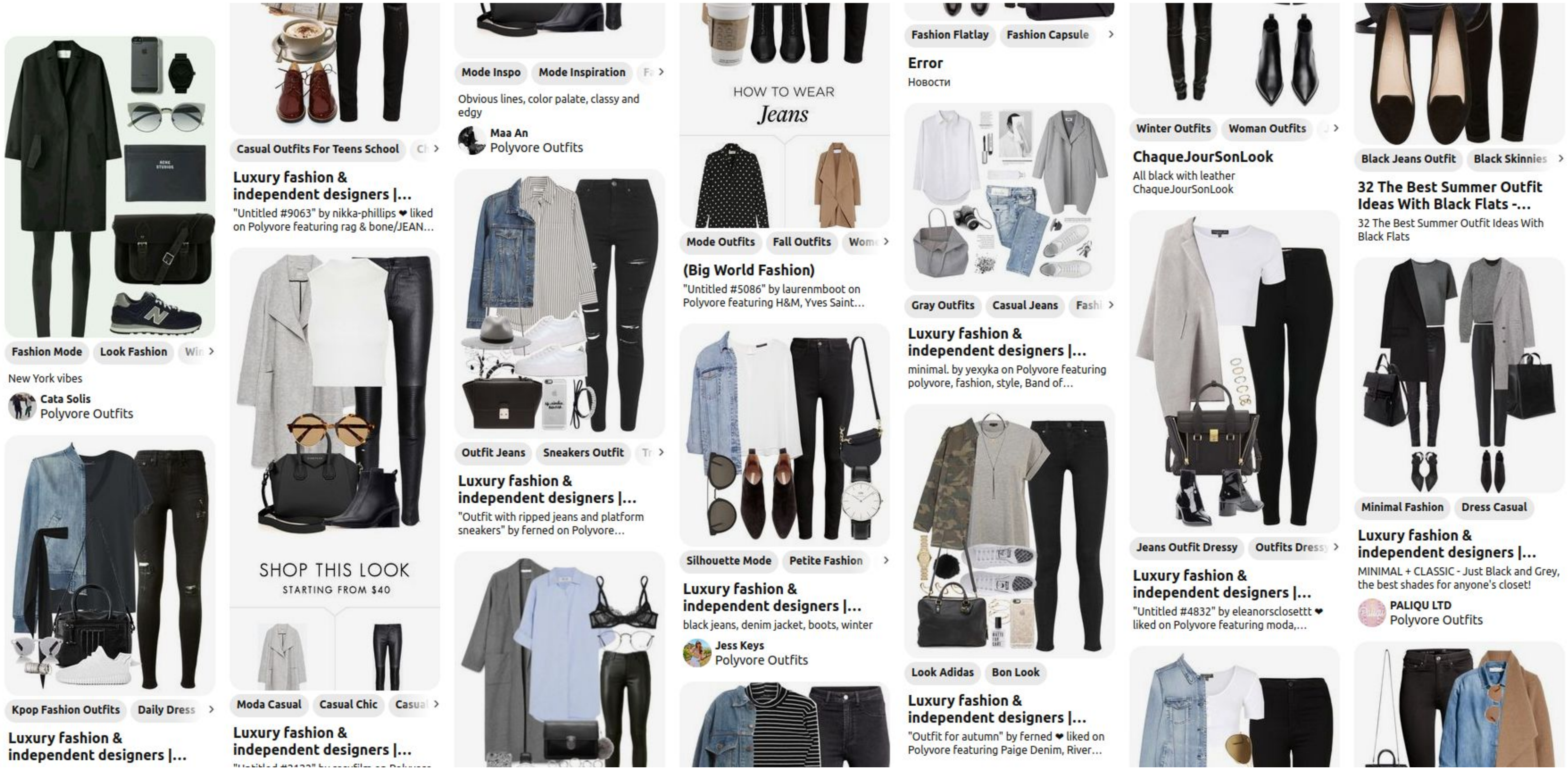
TABLE OF CONTENTS

INTRODUCTION

1. Problem
2. Related works
3. Motivation
4. Contribution

Problem

- A plethora of fashion outfits on the internet, how to search for one matching an use case?

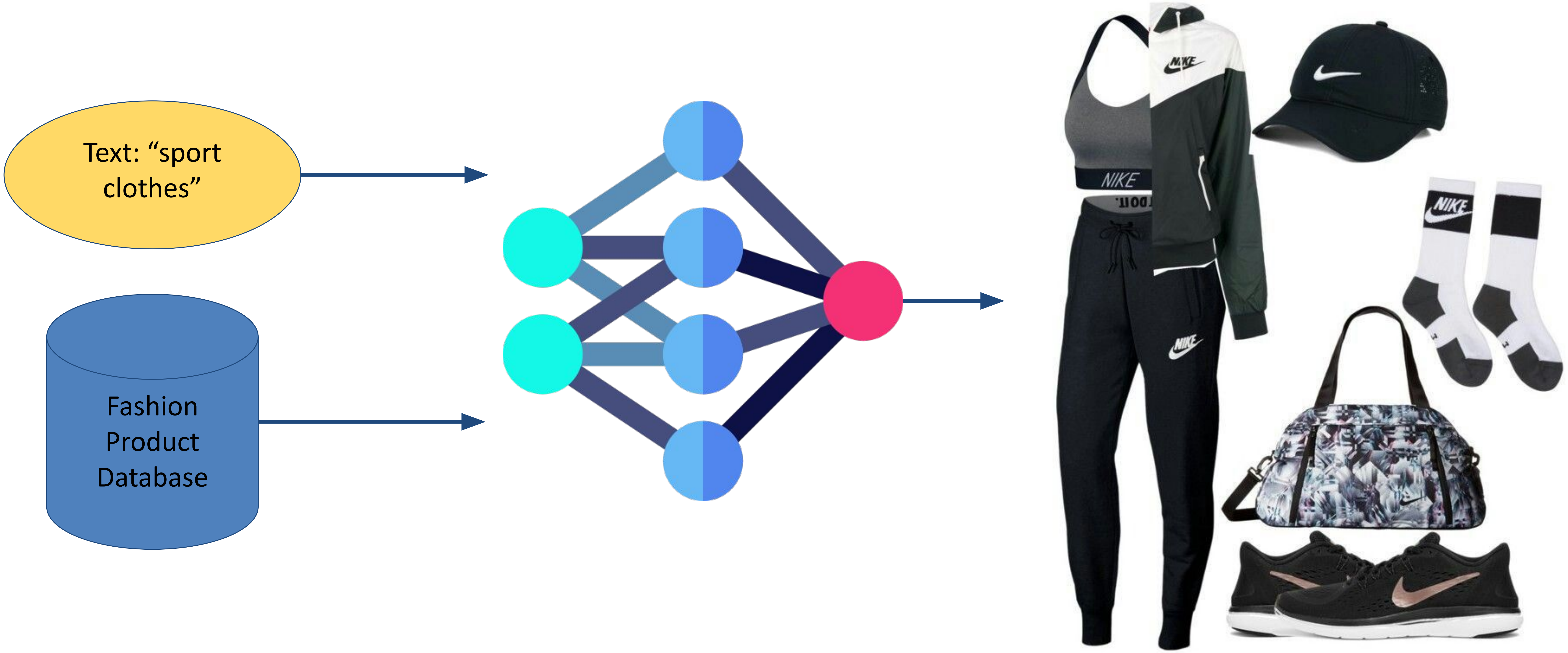


Polyvore outfits, from Polyvore [1]

[1] https://polyvore.ch/

Problem

- Goal
  - Compose outfits matching an user prompt from a large collection of fashion garments



Problem

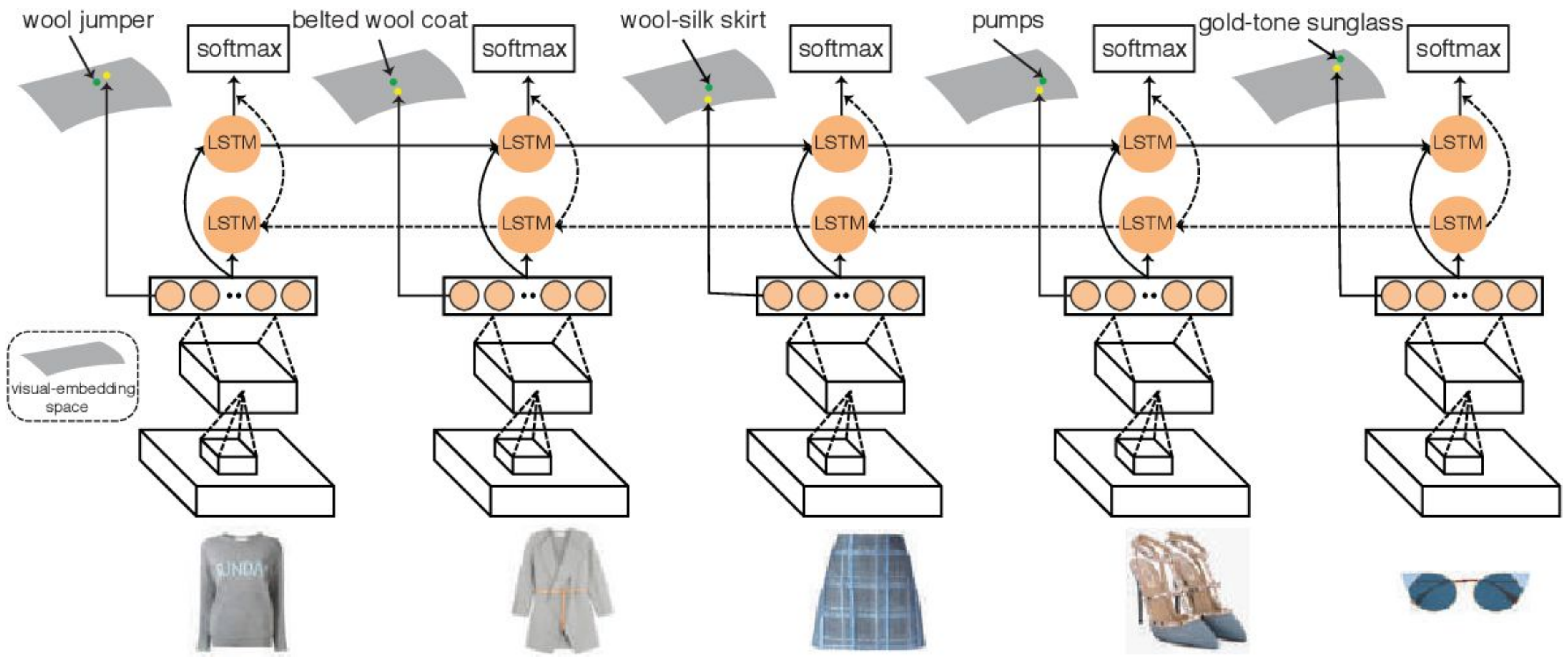
- Example:

Problem

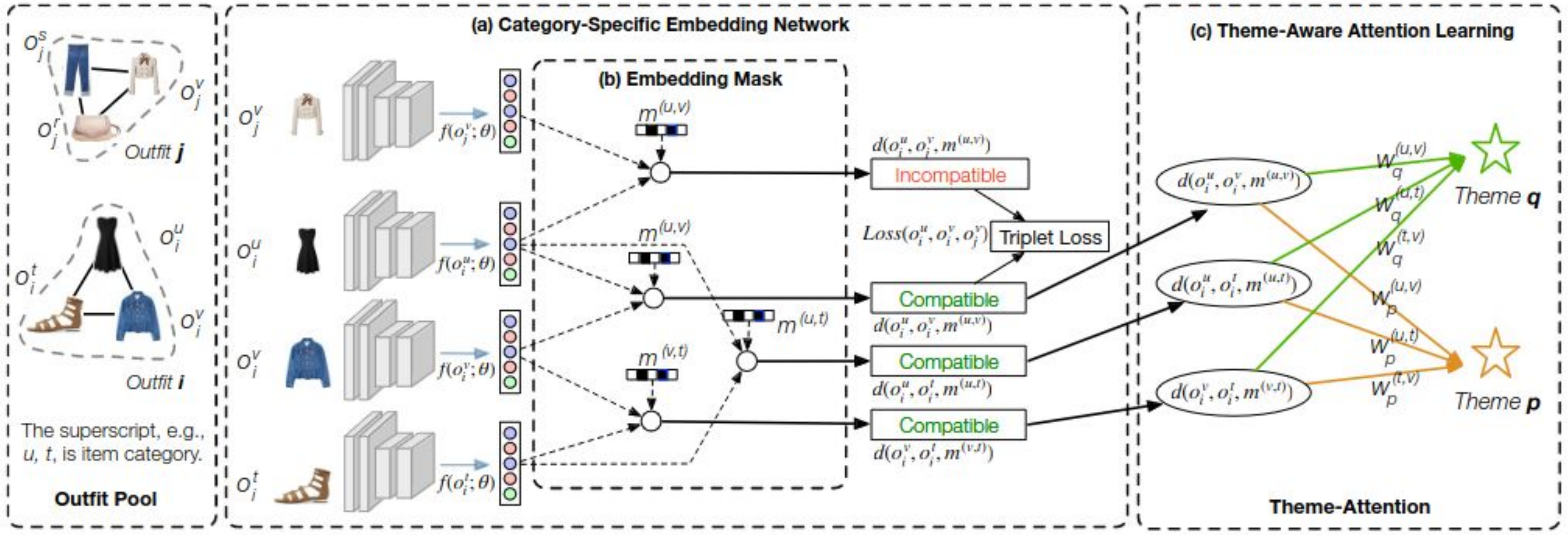
- Challenges
  - How to model the relationship between an ensemble of image items and its respective descriptions?
  - Number of possible outfits is huge → efficient retrieval method

Related works

- Han, X. et al. (2017) Learning fashion compatibility with bidirectional LSTMs



- Lai, J.-H. et al. (2020) Theme-matters: Fashion compatibility learning via theme attention



Motivation

- The rise of multi-modal model: CLIP, Stable Diffusion, and DALL-E
- Potential feature of an AI Chatbot used in Fashion domain

CLIP from OpenAI [2]

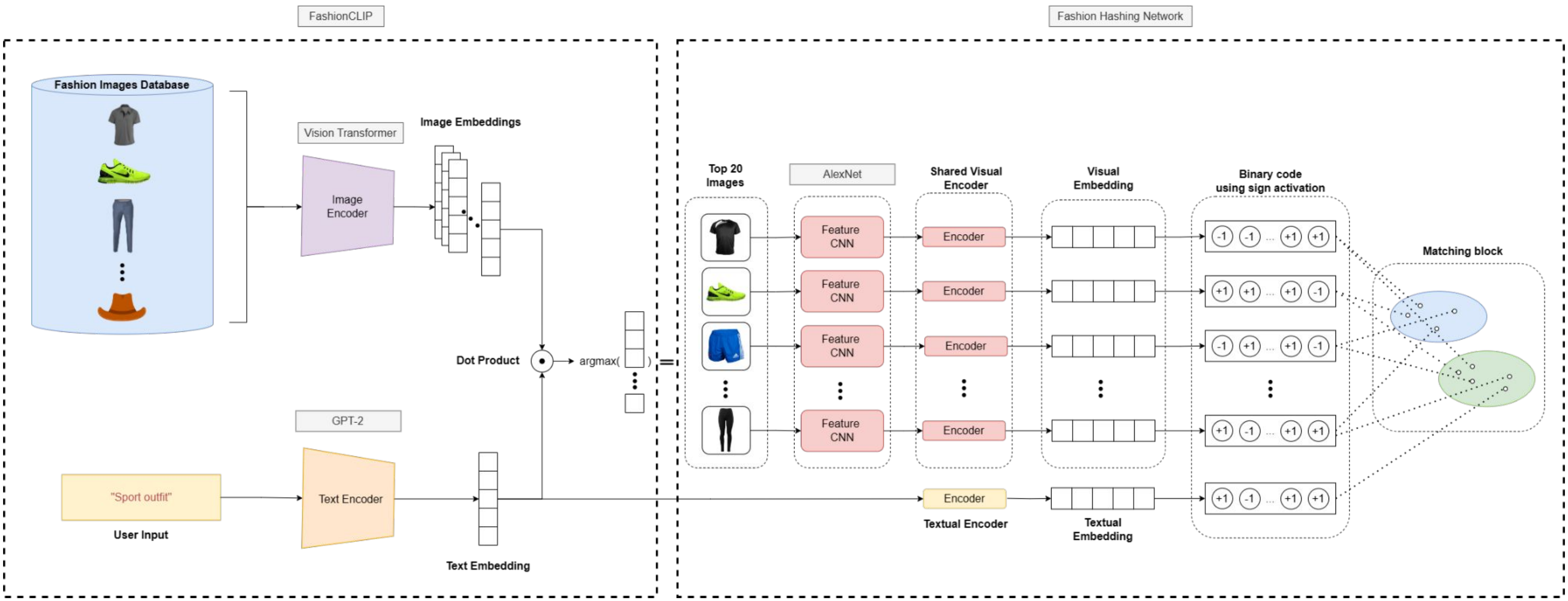
[2] https://openai.com/research/clip

Contribution

- Combines two models for recommending fashion outfits based on textual prompts
- Conduct experiments and demonstrations to assess the effectiveness of our proposed approach

METHODOLOGY

Overall Framework



Multimodal Model: CLIP

- The Text Encoder is a standard Transformer model with GPT2-style modifications
- The Image Encoder can be either a ResNet or a Vision Transformer

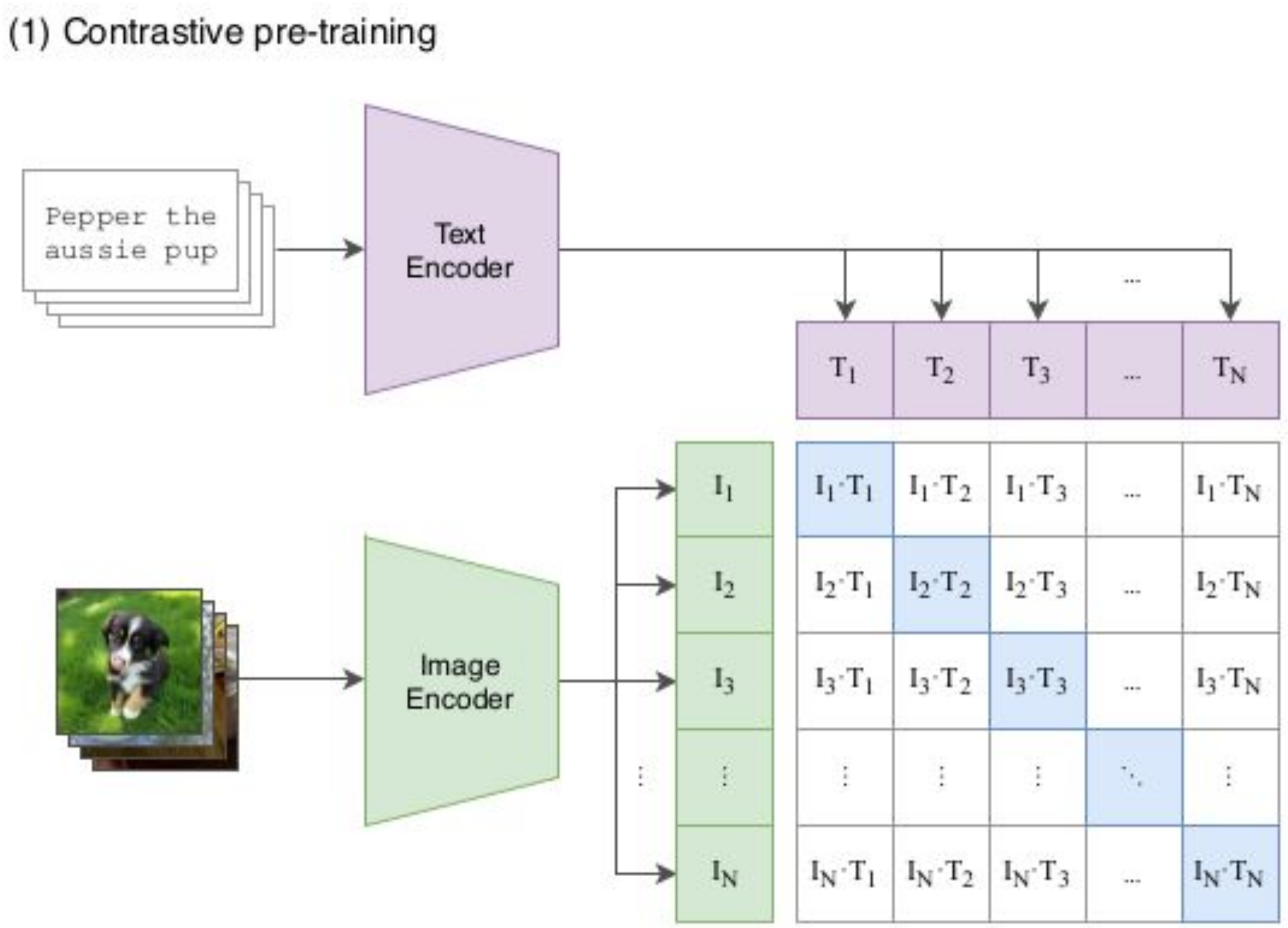


Figure of CLIP Architecture, from [4]

[4] Radford, A. et al. Learning transferable visual models from natural language supervision. In ICML (2021).

Multimodal Model: CLIP

- Contrastive Pre-training aims to jointly train an Image and a Text Encoder that produce image embeddings $[I_1, I_2 \dots I_N]$ and text embeddings $[T_1, T_2, \dots, T_n]$, in a way that:
  - The cosine similarities of the correct <image-text> embedding pairs $<I_1,T_1>$, $<I_2,T_2>$, ..., $<I_i,T_j>$ (where $i = j$) are maximized
  - The cosine similarities of dissimilar pairs $<I_1,T_2>$, $<I_2,T_3>$, ..., $<I_i,T_j>$ (where $i \neq j$) are minimized

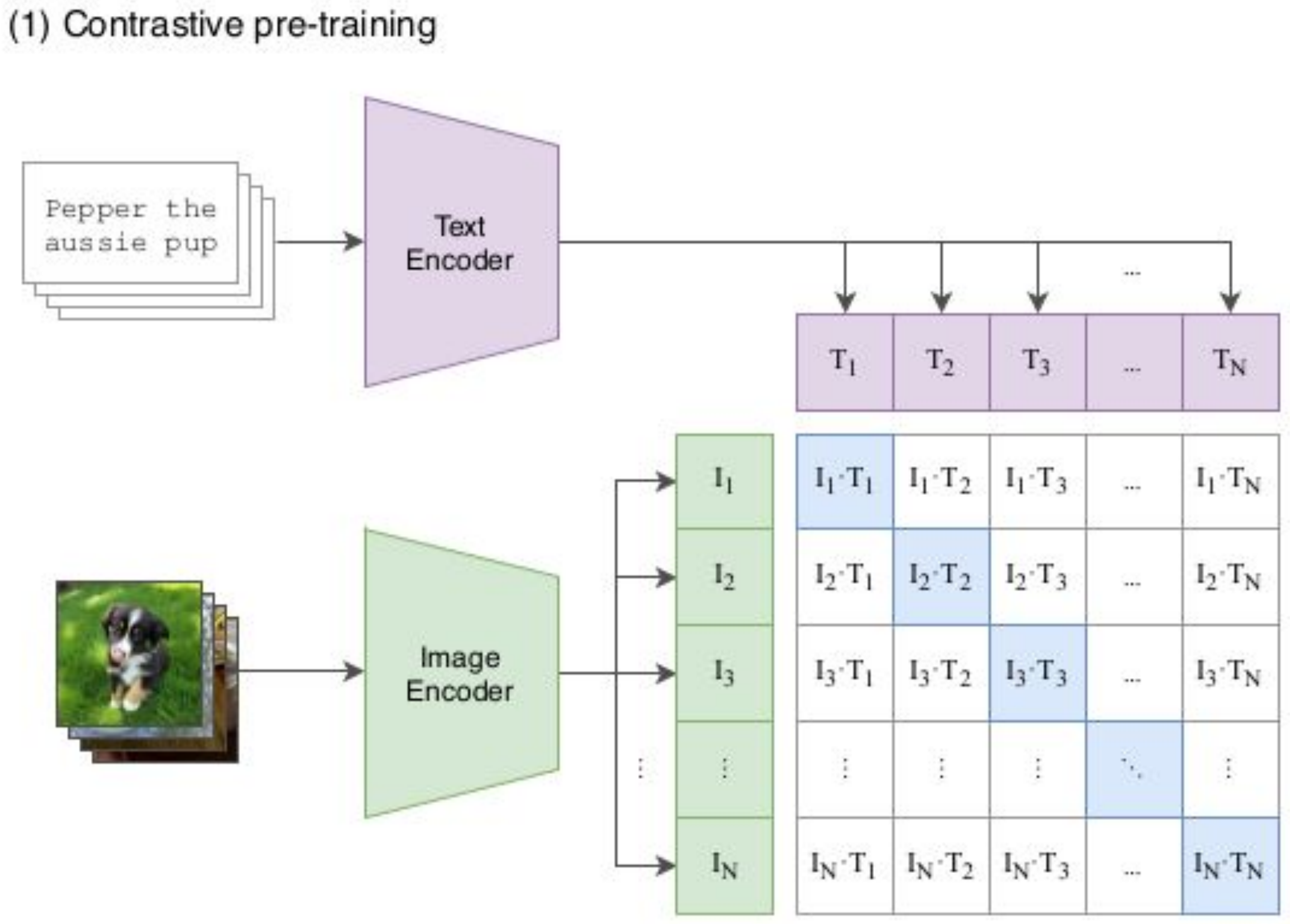
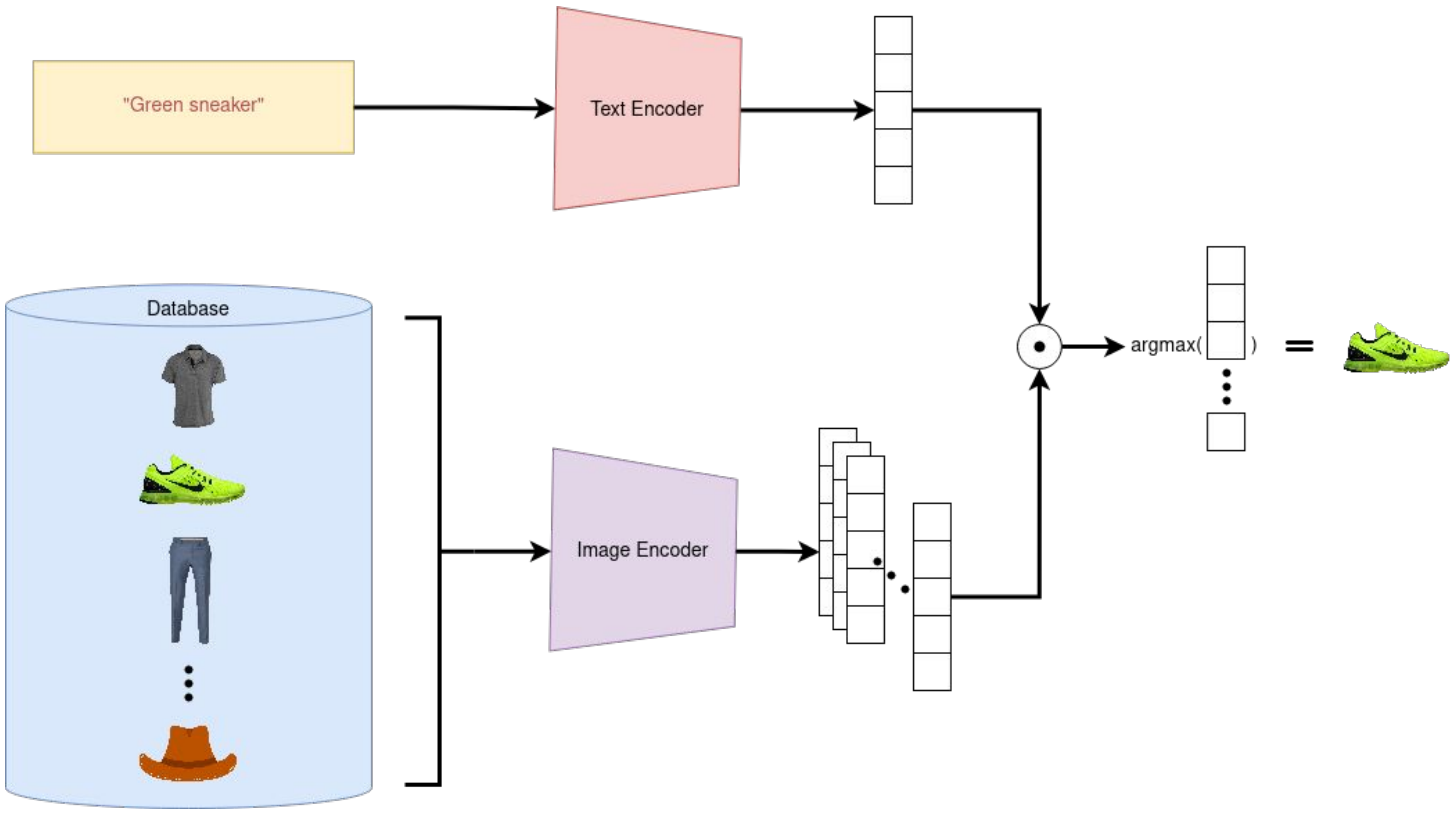


Figure of CLIP Architecture, from [4]

[4] Radford, A. et al. Learning transferable visual models from natural language supervision. In ICML (2021).

FashionCLIP Model



Transformer Architecture

- Revolutional model from the paper "Attention is All You Need" in 2017
- Key innovation is self-attention mechanism

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

Scaled Dot-Product Attention

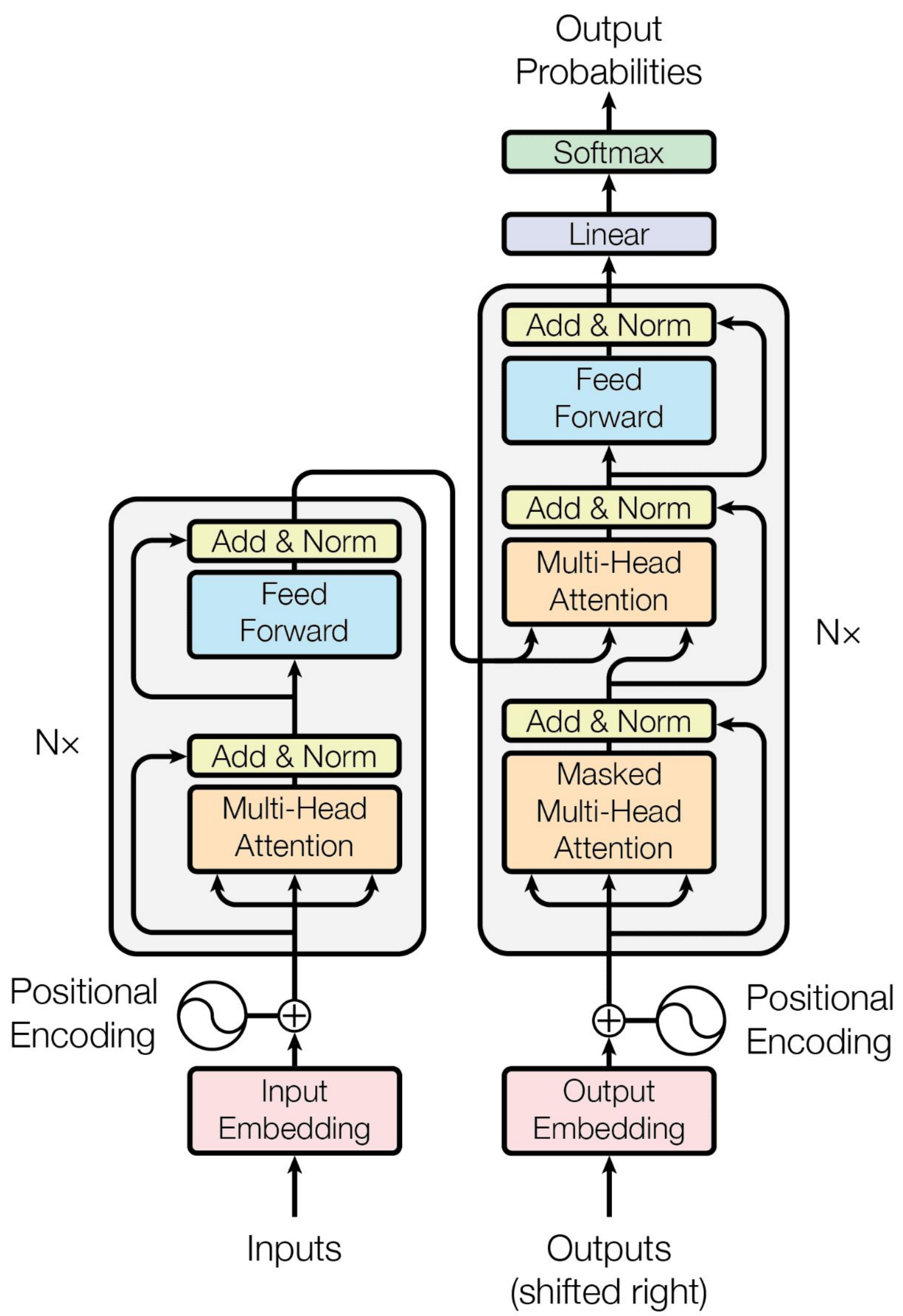


Figure of Transformer Architecture, from [3]

[3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need, 2017

GPT-2

- Foundation of the recent ChatGPT, GPT-4 which are popular today.
- GPT-2 using Decoder from Transformer with small change.

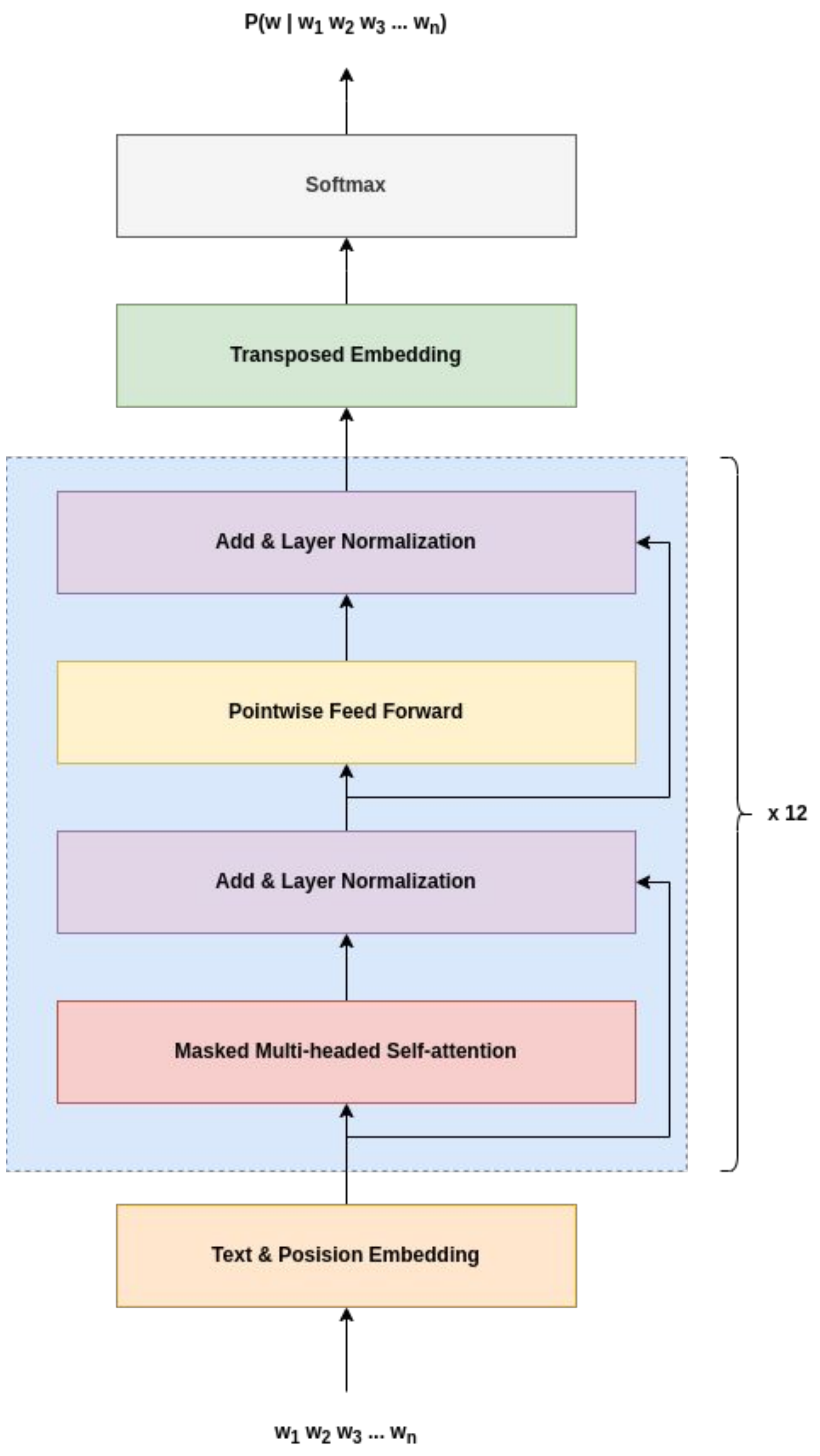


Figure of Small GPT-2 Architecture

Vision Transformer

- Key idea: An image is split into fixed-size patches, each of them are then linearly embedded, position embeddings are added, and the resulting sequence of vectors is fed to a standard Transformer encoder

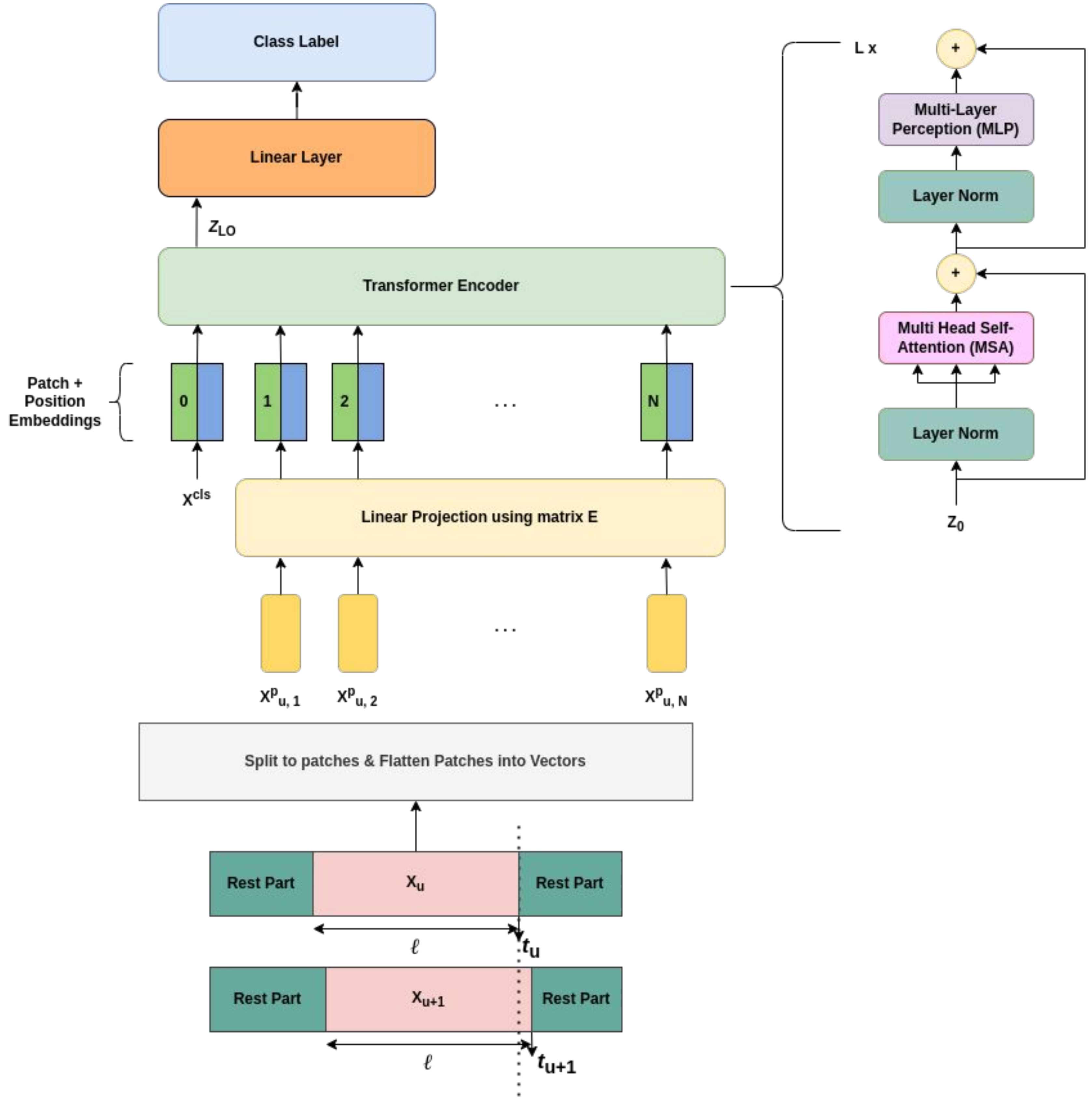


Figure of ViT Architecture

Original Fashion Hashing Network (FHN)

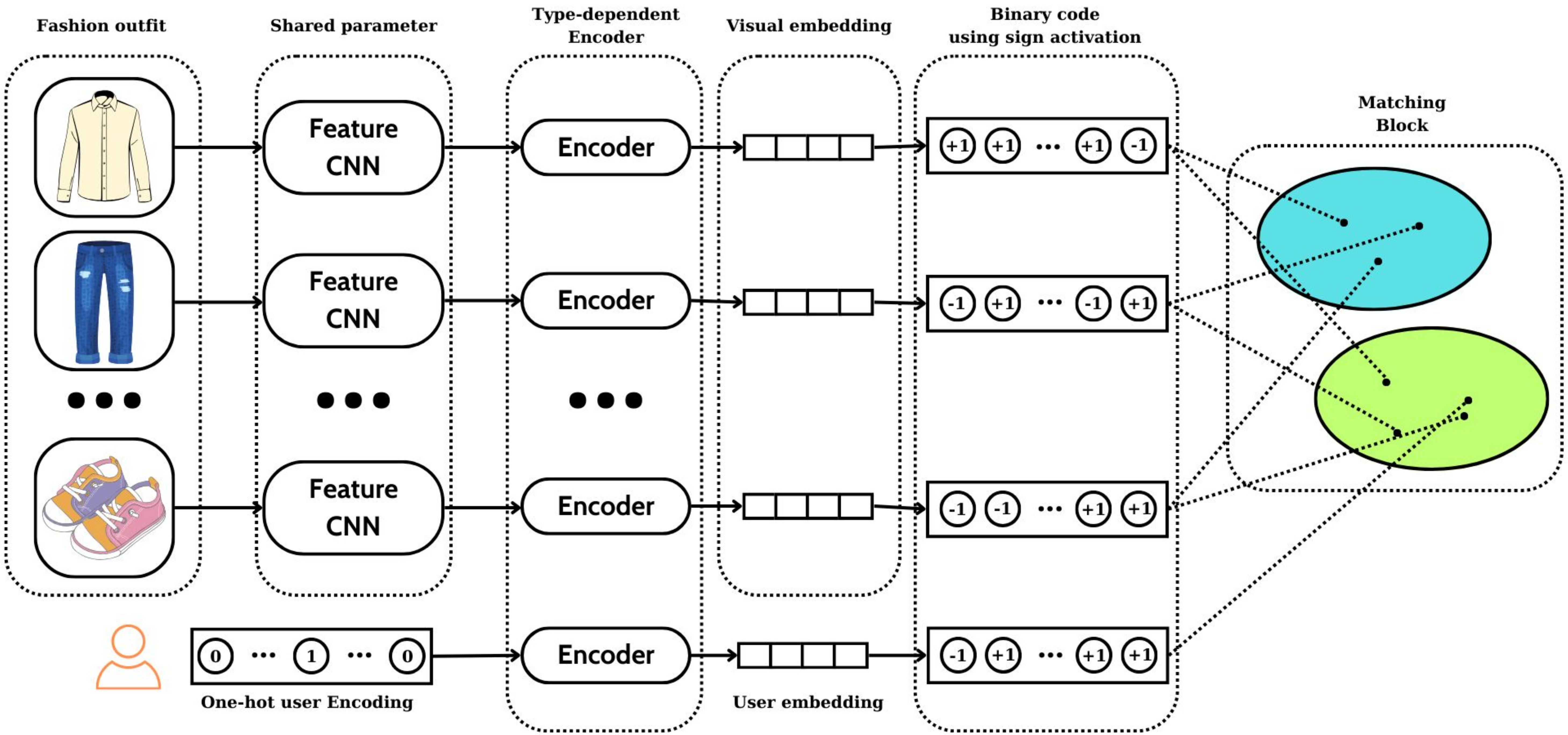


Figure of FHN Architecture

AlexNet Architecture

- A Convolutional Neural Network Architecture that won the LSVRC competition in 2012
- Backbone of one of our main models

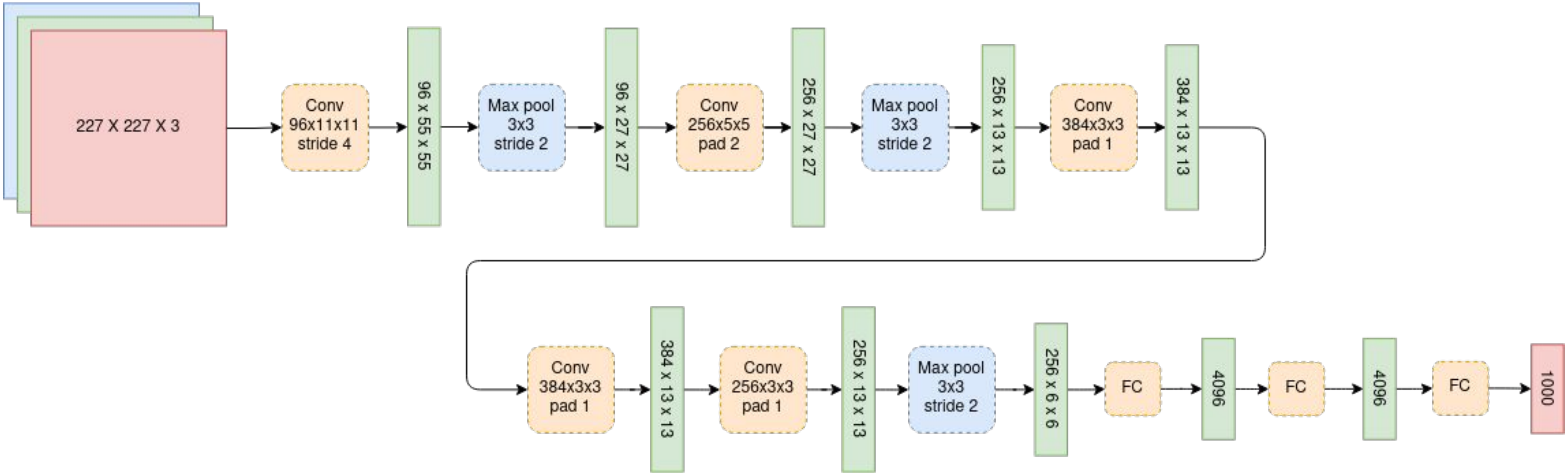


Figure of AlexNet Architecture

The Matching Block

- All items in the *n-th* category

$$X^{(n)} = \{x_1^{(n)}, x_2^{(n)}, \ldots, x_{L_n}^{(n)}\}$$

- Outfit with *N* items, with each from one category

$$O_i = \{x_{i_1}^{(1)}, x_{i_2}^{(2)}, \ldots, x_{i_N}^{(N)}\}$$

$(i_1, i_2, \ldots, i_N)$ is the index tuple

The Matching Block

- Converting to binary codes

$$b^{n}_{i_n} = sign(h^{n}_{i_n}); b^{t}_{t} = sign(h^{t}_{t})$$

  where sign(x) = 1 if x >= 0 and -1 otherwise

- The compatibility score of two objects

$$m_{ij} = b_i^T \Lambda b_j$$

The Matching Block

- The score of outfit O_i corresponding to a textual description

$$r_{O_i}^{(i)} = \frac{1}{z} \sum_n \sum_m b_{i_n}^{(n)T} \Lambda^{(i)} b_{i_m}^{(m)}$$

$$r_{t,O_i}^{(t)} = \frac{1}{z} \sum_n b_{i_n}^{(n)T} \Lambda^{(t)} b_t^{(t)}$$

Binary hashing code for items with different categories

Number of pairs

Binary hashing code for outfit description embedding

- The score for outfit Oi concerning prompt t

$$r_{t,O_i} = \alpha \cdot r_{t,O_i}^{(t)} + r_{O_i}^{(i)}$$

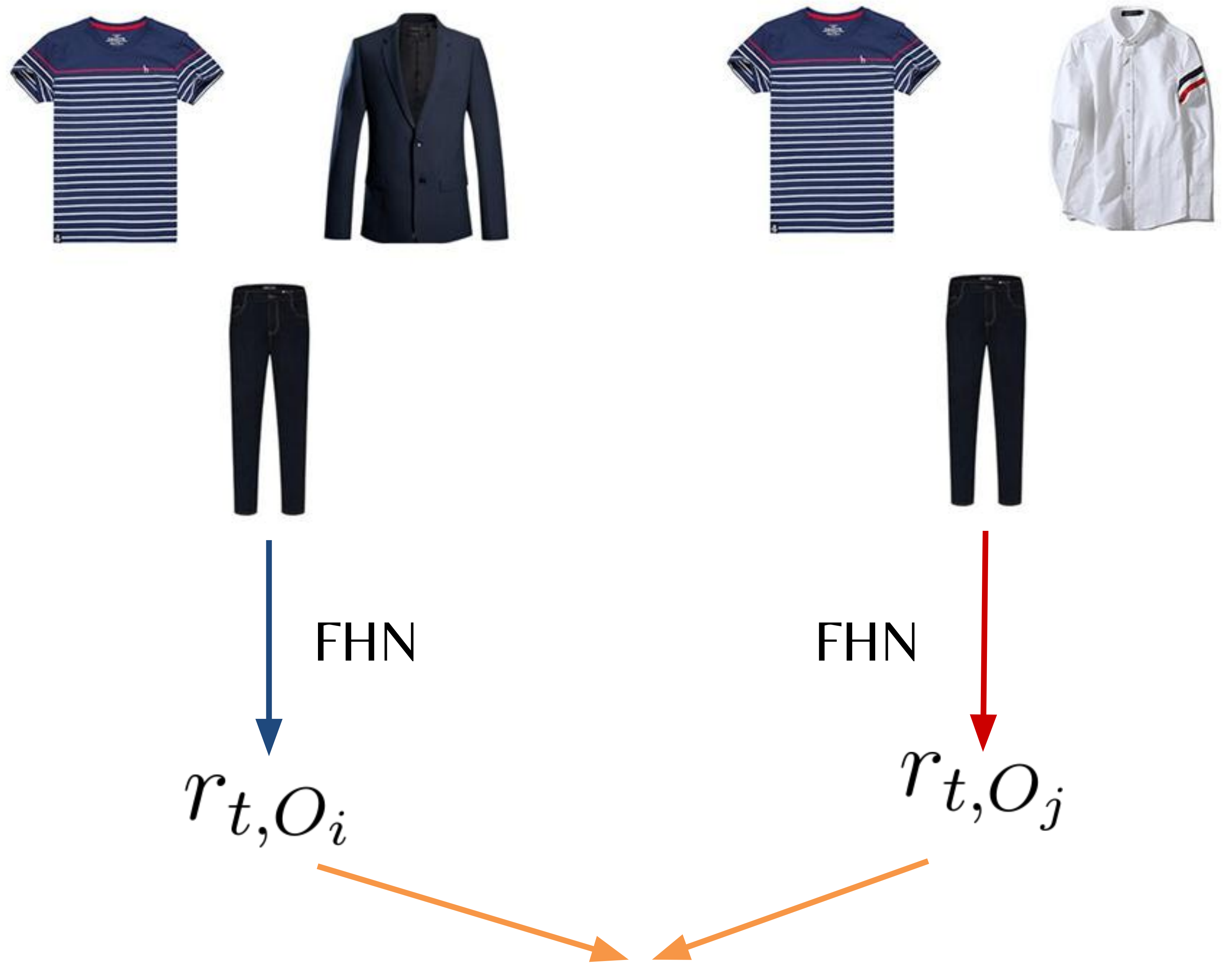
where α = 0 if not incorporating outfit textual embedding into training

Objective Function

- Positive samples $r_{t,Oi}$: complete outfits matching description t (positive outfit)
- Negative samples $r_{t,Oj}$: different outfits from positive outfits

$$\mathcal{P} \equiv \{(t,i,j) | r_{t,O_i} > r_{t,O_j}\}$$

Training outfit pairs



Soft Margin Loss

$$\mathcal{L}_{BPR} = \sum_{(t,i,j)\in\mathcal{P}} log(1 + exp(-(r_{t,O_i} - r_{t,O_j})))$$

Figure of training pipeline

DATASET

1. Polyvore dataset
2. Fashion32 dataset
3. Data preprocessing

Polyvore dataset

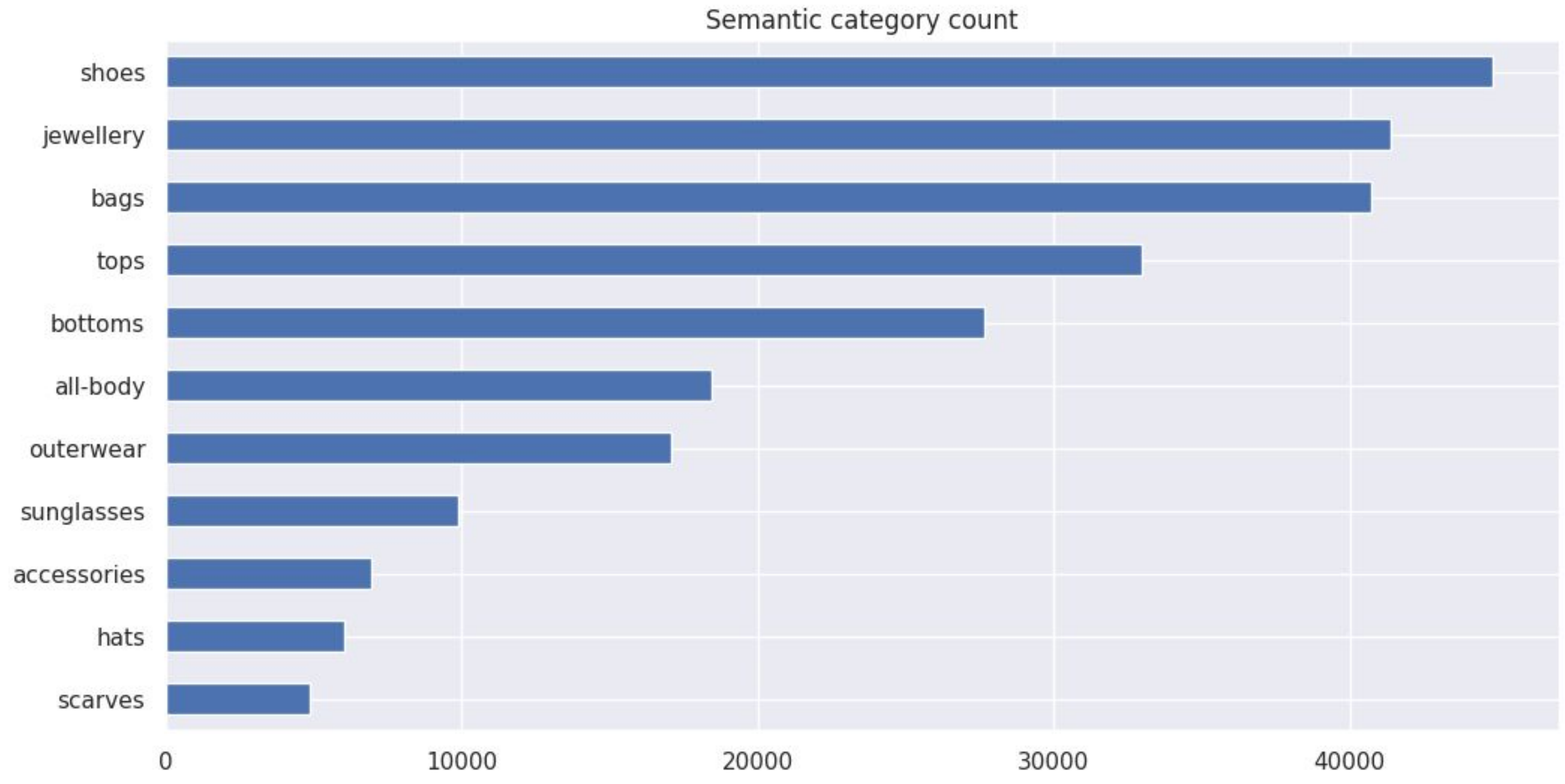
- The Polyvore dataset contains of about 261k images of items with their metadata. We use the images and category for this thesis

	img_name	semantic_category
0	100004189.jpg	sunglasses
1	100005237.jpg	accessories
2	100007550.jpg	all-body
3	100010397.jpg	shoes
4	100010564.jpg	shoes

Some example images of the Polyvore dataset and the metadata used

Polyvore dataset

- The Polyvore dataset contains of about 261k images of items with their metadata. We use the images and category for this thesis



Distribution of categories in Polyvore dataset

Polyvore dataset

- The images are combined into outfits which are classified into 2 outfit datasets: disjoint and nondisjoint. We only focus on the disjoint dataset in this thesis

	all-body	bottom	top	outerwear	bag	shoe	accessory	scarf	hat	sunglass	jewellery	compatible
0	172312529	-1	-1	-1	132621870	153967122	-1	-1	-1	-1	-1	1
1	172482221	-1	-1	-1	162715806	171888747	-1	-1	-1	-1	-1	1
2	-1	181657245	-1	165695205	180028994	182218570	-1	-1	-1	-1	-1	1
3	195973920	-1	-1	-1	198643069	206048471	-1	-1	-1	-1	-1	1
4	-1	204650506	200313980	-1	200139640	156489567	-1	-1	-1	-1	-1	1

Sample outfit items in CSV format table of disjoint outfit dataset

Fashion32 dataset

- The Fashion32 dataset contains about 41k images of items with metadatas which are combined into about 14k outfits

Sample of outfit in the Fashion32 dataset

Fashion32 dataset

- The Fashion32 dataset contains about 41k images of items which are combined into about 14k outfits

	outfit_id	top	outerwear	bottom	full-body	bag	accessory	footwear
0	10269	-1	10269_9708_31264127289.jpg	-1	10269_9719_30906140243.jpg	-1	-1	10269_9772_22469632534.jpg
1	774	774_9732_13730321818.jpg	-1	774_9736_14020491171.jpg	-1	-1	-1	774_6908_14193670097.jpg
2	14484	14484_1348_41318973794.jpg	-1	14484_9735_41318976248.jpg	-1	-1	-1	-1
3	3091	3091_1354_25690065742.jpg	-1	3091_9720_25689993723.jpg	-1	-1	-1	3091_9772_24614335454.jpg
4	13912	13912_9713_32104014616.jpg	-1	13912_9720_33587227013.jpg	-1	-1	-1	-1

Sample of outfit items in CSV format table of the Fashion32 dataset

Preprocessing

- For the Polyvore dataset:
  - Based on the certain category for each outfit item, classify them into 11 groups: all-body, bottom, top, outerwear, bag, shoe, accessory, scarf, hat, sunglass, jewelry
  - The resulting items are then combined based on the metadata and store in CSV files, where each row matches with an outfit along with its various items and the compatible attribute which determines if the outfit is well-matched

Preprocessing

- For the Fashion32 dataset:
  - Employ the Google Translate API to convert the metadata (original in Chinese) into English
  - Based on the certain tags for each outfit item, classify them into 7 groups: top, outerwear, bottom, full-body, bag, accessory, and footwear
  - The resulting items are then combined based on the metadata and store in CSV files, where each row corresponds to an outfit along with its various items

Negative outfits generation

- The outfits from the dataset are labeled as **positive**
- For each positive outfit, randomly select items from the dataset to construct an incompatible outfit, labeled as a **negative** outfit, ensuring it does not match the corresponding positive outfit

EXPERIMENTS

1. Evaluation metrics
2. Benchmark
3. Demonstration

Evaluation metrics

- For outfit recommendations:
  - Area Under the ROC (AUC) score
  - Normalized Discounted Cumulative Gain (NDCG) score
  - Fill-in-the-blank (FITB) visualization score

Evaluation metrics

- For outfit recommendations:
  - Area Under the ROC (AUC) score

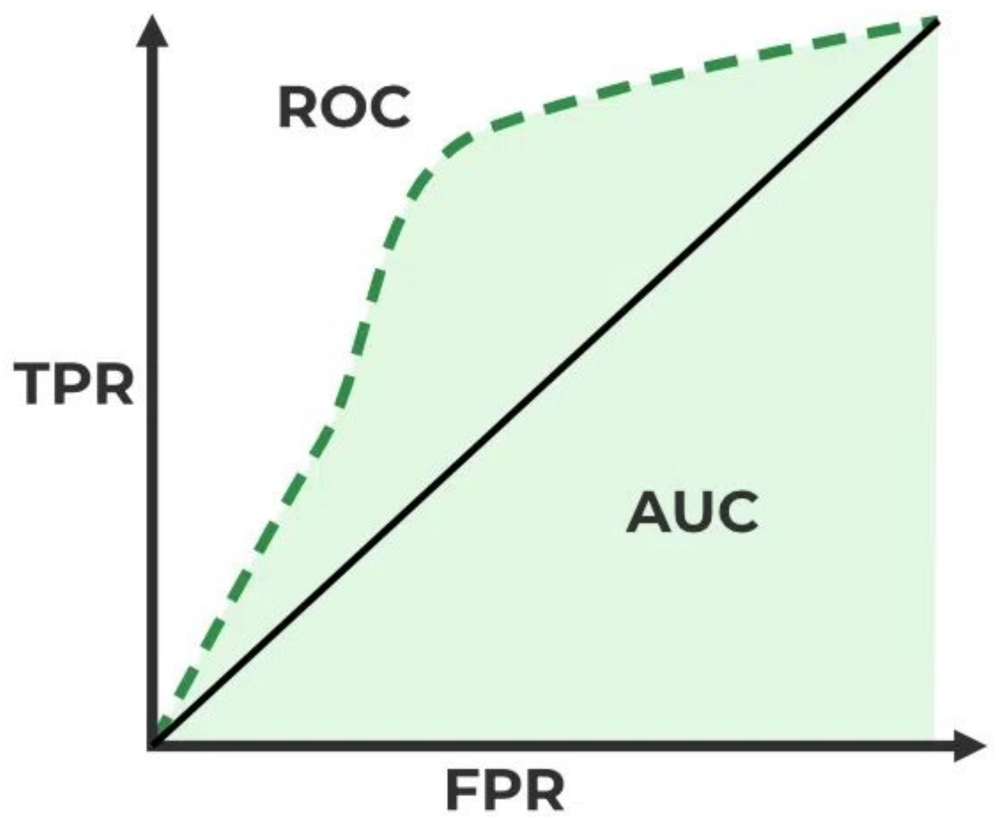
True Positive Rate:

$$TPR = \frac{TP}{TP + FN}$$

False Positive Rate:

$$FPR = \frac{FP}{FP + TN}$$

The ROC-AUC curve:



Evaluation metrics

- For outfit recommendations:
  - Normalized Discounted Cumulative Gain (NDCG) score

$$\mathrm{NDCG@m} = (N_m)^{-1} \sum_{i=1}^{m} \frac{2^{y_{\pi'(i)}} - 1}{\log_2(\max(2, i))}$$

Evaluation metrics

- For outfit recommendations:
  - Fill-in-the-blank (FITB) visualization score

(a) Some examples of the correct cases obtained by our method

(b) Some example of the failed cases obtained by our method

Visualization of the FITB task

Benchmark

- 3 different models comparison (100 epochs):
  - FHN-T3 (Visual - Polyvore): the FHN model trained on item images of the Polyvore dataset
  - FHN-T3 (Visual): the FHN model trained on item images of the Fashion32 dataset
  - FHN-T3 (Visual + Outfit semantic): the FHN model trained on item images of the Fashion32 dataset and also outfit textual description embedding accompanying each outfit

Method	Accuracy	AUC	NDCG	FITB
FHN-T3 (Visual - Polyvore)	0.6232	0.6115	0.7153	0.3520
FHN-T3 (Visual)	0.8191	**0.8150**	**0.8518**	**0.5542**
FHN-T3 (Visual + Outfit semantic)	**0.8706**	0.7416	0.7982	0.5071

CONCLUSIONS AND FUTURE WORKS

Conclusions

- Introduce FashionCLIP model
- Introduce Fashion Hashing Network model
- Combine these two models together

Future Works

- Inference speed
- Incorporate more fashion categories, like accessories, ...
- Aesthetic capabilities
- Potential expansions: room design, ...

DEMONSTRATION

Demonstration

Search:

male casual outfit to go out on sunday night

top bottom bag outerwear shoe

Demonstration

Search:

male outfit for an interview

top | bottom | bag | outerwear | shoe

Demonstration

Search:

male casual outfit to go out on sunday night

top	bottom	bag	outerwear	shoe

Q&A

THANK YOU

FOR YOUR LISTENING!