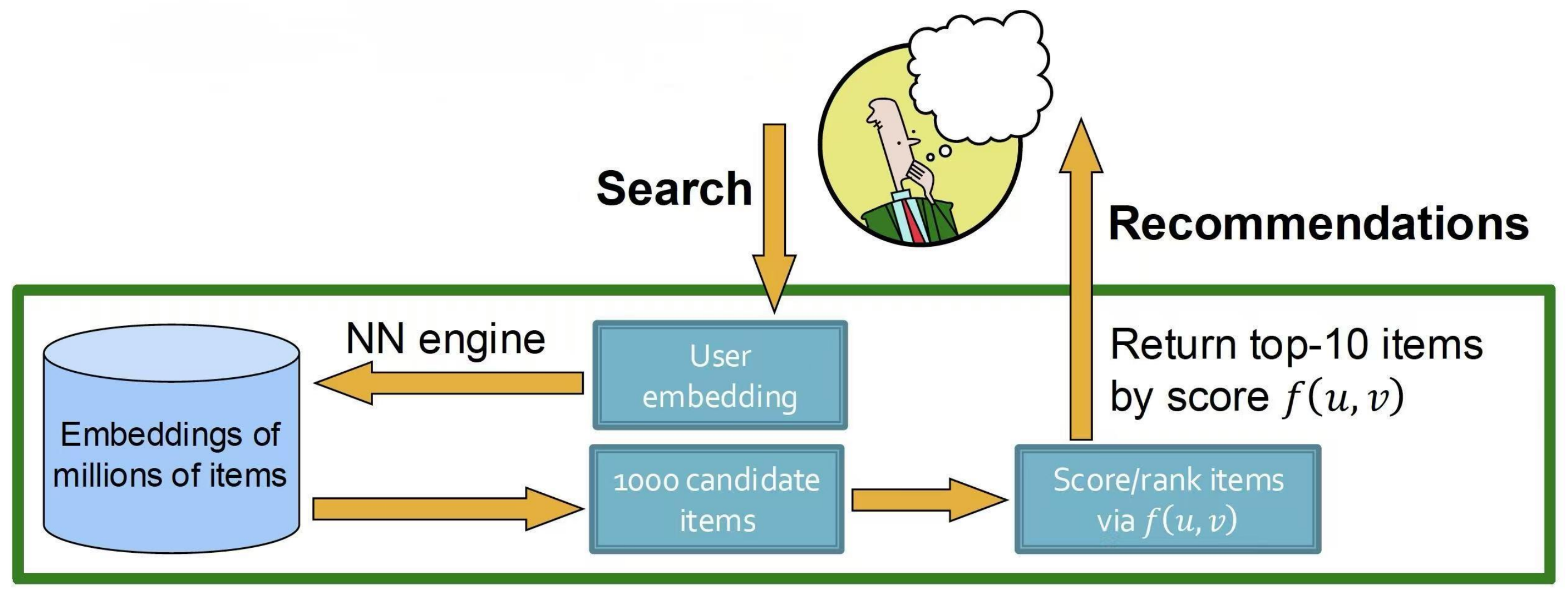
Learning a Multi-Head Value Model for Short-Video Recommendation with GNNs

Haozhan Gao, Stanford Graduate School of Business

Overview

Recommender systems rely on user engagement metrics such as completion, likes, and comments, but these are **noisy signals** of true latent user utility.

- **Hypothesis**: even with the same latent preferences, users behave differently when they hold different beliefs about recommendation quality or follow different search patterns.
- I train a toy short-video recommender with a multi-head value model on the KuaiRec dataset: a GNN predicts four engagement heads, and head weights are estimated by matching per-session average embeddings.
- The toy recommender is meant to **approximate the deterministic part** of the real system so that the residual between observed and simulated rankings can be treated as an **exogenous shock** to help causal analysis.
- The future work will debias the user engagement signals so they only depend on latent preference.


Search
Recommendations
NN engine
User embedding
Return top-10 items by score $f(u,v)$
Embeddings of millions of items
1000 candidate items
Score/rank items via $f(u,v)$

Data and Features

I use the KuaiRec dataset, a fully observed short-video recommendation dataset from the Kuaishou platform. It contains user-video interaction logs with timestamps, play duration and rich side information for three high-traffic bursts between July and September 2020. After filtering and splitting by calendar day, I obtain about 0.8M training interactions and 0.2M interactions each for validation and test. Each interaction is represented by a feature vector that merges four types of information:

- **User features**: Activity level, whether the user is a creator or live-streamer, follow/fan/friend counts (and log/binned versions), and encrypted demographic one-hot features.
- **Item features**: video duration and aspect ratio, upload and visibility type, author and music IDs, and three levels of content category.
- **Context features**: burst and session IDs, time-of-day (sine/cosine of hour), and a weekend indicator.
- **History features**: exponential moving averages of past engagement outcomes, category-level EMAs and entropy, author recency and last-completion flag, and simple session statistics (previous session length and inter-session gap).

A Multi-Head Value Model

For each user–video interaction I predict four binary engagement “heads”:

- $p_{comp} = 1$ if that the user completes the video (watch ratio > 1)
- $p_{long} = 1$ if the watch ratio > 1.5 or the play duration > 12 seconds
- $p_{rewatch} = 1$ if the user rewatches the content (watch ratio ≥ 2)
- $p_{neg} = 1$ if the interaction is negative (play duration < 2 seconds)

A score assigned to each video is a weighted sum of the predicted heads:

$$Score = \sum_{j=1}^{4} w_j \hat{p}_j \quad \text{for j} = \{\text{comp}, \text{long}, \text{rewatch}, \text{negative}\}$$

For each user and session, the recommender computes this score for all candidate videos and ranks them in descending order and recommend the top-K items.

I train a GNN to predict these heads, and estimate head weights by matching the average per-session embeddings between the simulated recommendation and the real one.

GNN Structure

I represent the data as a user–item bipartite graph with learnable node embeddings and edge features for each interaction. A 3-layer GCN performs message passing on this graph to produce updated user and item embeddings. Specifically, each user and item is associated with an embedding vector $h_v^0 \in \mathbb{R}^H$ with hidden dimension H. The graph convolution has the GCN type:

$$h_v^{(\ell+1)} = \sigma\left(\sum_{w \in \mathcal{N}(v)} \alpha_{vw}^{(\ell)} W^{(\ell)}\, h_w^{(\ell)}\right), \ell = 0,1,2$$

where $\mathcal{N}(v)$ is the set of neighbors of v,$W^{(\ell)}$ are learnable weight matrices, $\alpha_{vw}^{(\ell)}$ are normalization coefficients based on node degrees, and σ is the ReLU activation.

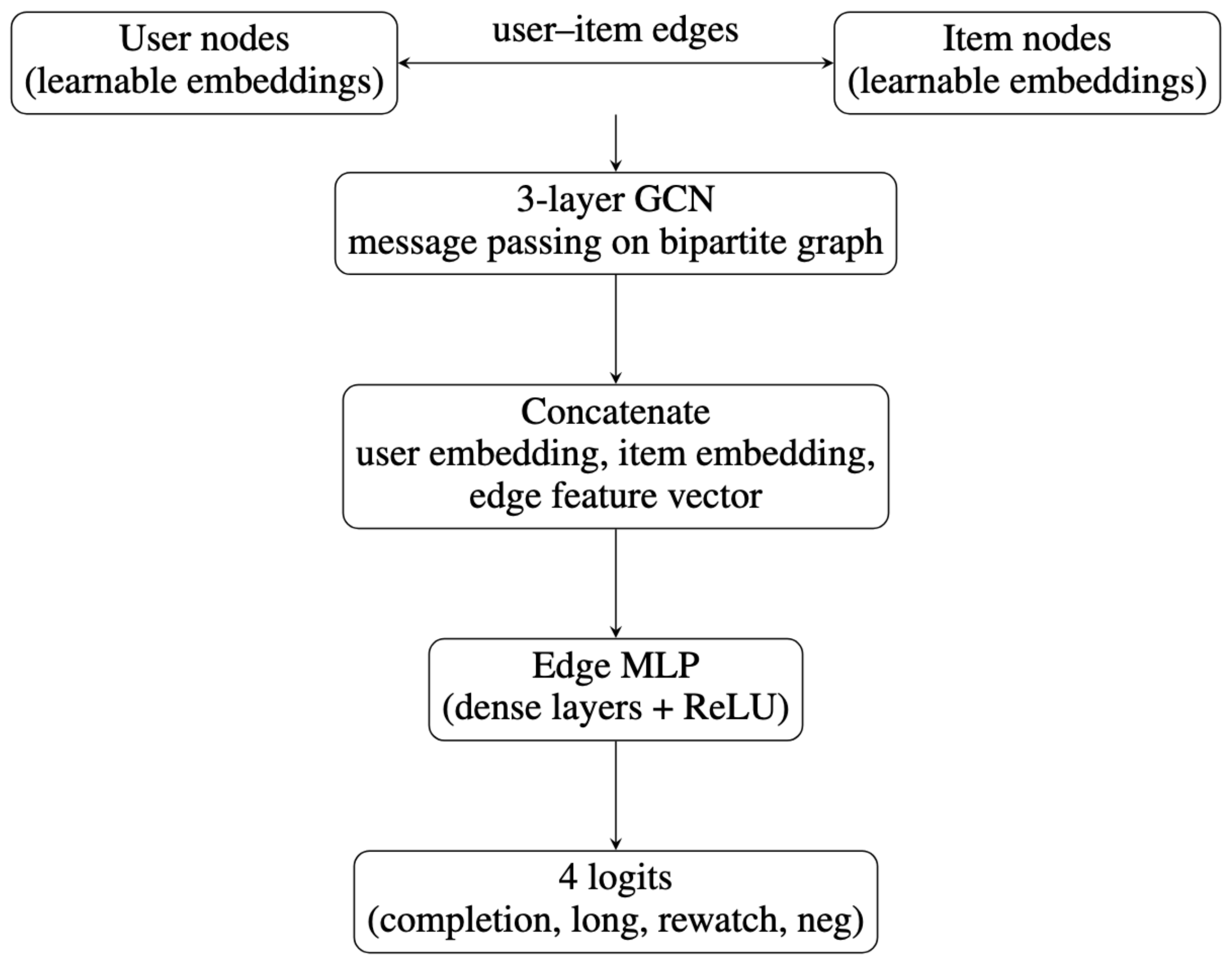
After obtaining the final node embeddings for all users and items, for every edge, I concatenate the user/item embeddings and edge feature vector, feed them through an 3-layer Edge MLP, and obtain 4 logits corresponding to the completion, long-watch, rewatch, and negative-feedback heads (the same prediction head used in the non-graph baselines).

Head Weight Estimation

I estimate the head weights by matching the per-session averaged embedding between the simulated recommendation and the observed one:

$$w^* = \arg\min_w \frac{1}{\mathcal{S}} \sum_{s \in \mathcal{S}} \left|\left|\mu_s^{obs} - \mu_s^{sim}(w)\right|\right|_2^2$$

I obtain the estimate $w^* = (0.22, 0.10, -0.02, -0.66)$ for head {comp, long, rewatch, neg} respectively, which shows that completion and long watch are rewarded, while negative feedback is penalized.



Results and Discussion

I compare prediction performance against two baselines: logistic regression and a feed-forward MLP that use only edge features, and I test two embedding sizes for the GNN (64 vs. 128). Logistic regression already achieves strong AUCs, indicating that the engineered edge features are highly informative. Adding a non-linear Edge MLP and then a GNN yields consistent but modest gains, with the 64-dim GNN giving the best mean test AUC, especially on the harder long-watch and rewatch heads. Increasing the hidden size to 128 provides almost no additional improvement, suggesting limited returns from extra capacity on this dataset.

Model/test AUC	complete	long	rewatch	negative	mean
Logistic regression	0.8297	0.7718	0.8315	0.8617	0.8237
Edge MLP (hidden = 128)	0.8485	0.8198	0.8537	**0.8658**	0.8470
GNN (hidden = 64)	**0.8492**	**0.8213**	**0.8566**	0.8657	**0.8482**
GNN (hidden = 128)	0.8446	0.8194	0.8541	0.8649	0.8458

Future Direction

The pre-trained toy recommender will be used for causal inference. Because it is trained on rich data and features, its simulated rankings approximate the “deterministic” part of the real platform policy; the residual between simulated and observed rankings serves as an exogenous treatment, orthogonal to unobserved drivers of behavior. I will estimate user selectivity in a structural behavioral model and regress this selectivity on the treatment to test whether changes in recommendations causally shift selectivity. If the effect is significant, I will work on debiasing engagement signals so that they reflect latent utility rather than beliefs about the algorithm or search patterns.