STAT 461 Project Proposal: Self-Supervised Sequential Recommendation with Hard Negative Graph Contrastive Learning

Sharon Lin
Northwestern University
Evanston, Illinois
sharonlin2025@u.northwestern.edu

Faith Cramer
Northwestern University
Evanston, Illinois
faithcramer2025@u.northwestern.edu

Zihan Zhao
Northwestern University
Evanston, Illinois
zihanzhao2024@u.northwestern.edu

Jeffrey Yuan
Northwestern University
Evanston, Illinois
jeffreyyuan2026.1@u.northwestern.edu

1 Description of the Problem

Accurately predicting user actions based on recent behavior remains a core challenge in sequential recommendation, particularly as users demonstrate both enduring preferences and rapidly shifting short-term interests. While recent advances have turned to self-supervised learning (SSL) to alleviate the limitations of sparse supervision in recommendation systems, two key challenges persist: the neglect of collaborative short-term behavior across users, and the presence of noise in real-world short-term interactions arising from temporary intent or misclicks.

Addressing these issues, SelfGNN (1) introduces a self-supervised graph neural network framework that models user behavior at multiple temporal resolutions. It constructs short-term user-item graphs segmented by time intervals and uses GNNs to learn high-order collaborative patterns among users' recent actions. In parallel, it captures long-term user and item preferences via multi-level sequential encoders with interval fusion and dynamic behavior modeling. Crucially, SelfGNN incorporates a personalized self-augmented denoising mechanism that supervises noisy short-term representations using long-term signals, thereby enhancing robustness and adaptability. Extensive empirical results across four real-world datasets confirm that SelfGNN consistently outperforms state-of-the-art baselines, validating its ability to model complex temporal dynamics and mitigate the impact of behavioral noise in sequential recommendation.

The authors of the SelfGNN paper highlight that future extensions may focus on more accurately capturing short-term behavioral characteristics to further enhance recommendation performance. One promising direction involves integrating contrastive learning with hard negative sampling (2) to refine short-term modeling. Hard negative sampling selects negative examples that are difficult to distinguish from positives, typically those close to the anchor in embedding space but not actually relevant, which forces the model to learn more precise decision boundaries. While the current framework aligns short-term and long-term representations to suppress noise, it does not explicitly distinguish between subtle but irrelevant interactions, particularly among items that appear similar but do not reflect true user intent. Incorporating hard negative sampling can help the model better separate meaningful signals from misleading ones, strengthening the discriminative power of short-term embeddings. This enhancement is especially valuable in datasets with high behavioral variability or noisy feedback, offering a principled way to improve recommendation accuracy and generalization.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

2 Description of the Data

2.1 Experimental Datasets

The experiments will be conducted on four open-source datasets.

- **Amazon-book**: User ratings of Amazon books from 2014.
- **Gowalla**: Check-in data from Gowalla, a location-based social network, from 2010.
- **Movielens**: Movie ratings collected from 2002 to 2009.
- **Yelp**: Venue review data from 2009 to 2019.

All 4 of the datasets will be processed with a 5-core setting, ensuring that all users and items have at least 5 interactions. The temporal train-test split strategy will hold the last interaction of each user for test, the second-to-last for validation, and the remaining for training. Negative sampling will only be implemented during training with one positive and four negatives per user.

3 Plan Towards Completion

3.1 Baseline Testing

To complete this project, we will first familiarize ourselves with the SelfGNN code-base and data pre-processing pipeline. This will serve as the baseline for the modifications that we will implement. We will replicate standard top-K recommendation metrics, Hit Rate (HR), and Normalized Discounted Cumulative Gain (NDCG) for the Amazon-book dataset in the original SelfGNN paper to confirm correct implementation. All other hyperparameters and configurations (e.g., learning rate = 0.001, embedding dimension = 64, LightGCN layers = 2) will remain consistent with those reported by the original authors (1).

3.2 Implement Hard Negative Sampling

After confirming our baseline results, we will implement a hard negative sampling strategy to enhance the discriminative power of the short-term user representations (2). For each training instance, we will compute the cosine similarity between a user's short-term embedding as the anchor and all the other embeddings. Items interacting with the anchor will be excluded, and the top-K most similar non-interacted items will be selected to serve as hard negatives. This aims to prioritize items that are more confusing, to create contrastive signals that encourage more discriminative decision boundaries.

In order to ensure that our sampling strategy is working as expected, we will begin by isolating the contrastive loss component and measure how the model is able to distinguish the short-term anchor, its positive (next interacted item), and hard negatives with InfoNCE loss and the Amazon-book dataset. This will validate that the selected negatives are close in embedding space and that the model can learn to separate them. Once the sampling strategy is confirmed, the contrastive loss objective function will be integrated into the overall training objective. The contrastive loss will be balanced with the existing supervised loss components using a weighting parameter λ. Sampling temperature (τ) and the number of negatives will be tuned with initial values of $\tau = 0.1$ and $K = 5$ hard negatives per anchor. We refer to the resulting contrastive-enhanced recommendation model as HardGNN.

3.3 Experimentations

Once the final method is made, hyperparameters will be tuned, and experiments will be performed and analyzed across all four datasets. Key hyperparameters include contrastive temperature, number of GNN layers, embedding size, and loss weights. The model will be evaluated using HR and NDCG metrics, and ablation studies comparing the SelfGNN baseline, HardGNN without hard negatives (random negatives only), and the full HardGNN model. The results, visualizations, and tables will be included in the final report and presentation.

References

[1] Wang, Yujie, et al. "SelfGNN: Self-Supervised Graph Neural Networks for Sequential Recommendation." arXiv preprint arXiv:2405.20878 (2024).

[2] Robinson, Joshua, et al. "Contrastive Learning with Hard Negative Samples." arXiv preprint arXiv:2010.04592 (2021).