

---

# SemGCN: Semantic Initialization and Alignment in Graph-based Recommendation Systems

---

Huy Gia Nguyen

Institute for Artificial Intelligence  
VNU University of Engineering and Technology  
Hanoi, Vietnam  
23020377@vnu.edu.vn

## Abstract

Large language models (LLMs) and pretrained transformers provide rich semantic representations of item content, yet effectively integrating natural language semantics into collaborative recommendation remains challenging due to the mismatch between language modeling objectives and ranking-oriented recommendation losses. We present **SemGCN**, a simple semantic-augmented graph recommender built on top of LightGCN that injects item descriptions into graph-based collaborative filtering. SemGCN first encodes item metadata (title and description) with a pretrained sentence transformer and projects the resulting vectors into the recommendation embedding space. These semantic vectors are then used to warm-start item embeddings and to regularize training via an explicit alignment loss that encourages collaborative item representations to remain close to their frozen semantic targets while optimizing the standard Bayesian Personalized Ranking objective. Experiments on the Amazon *Video Games* dataset show that SemGCN consistently improves ranking quality over vanilla LightGCN, yielding relative gains of about 5–6% on Recall and NDCG at  $K \in \{10, 20\}$  under full-ranking evaluation.

## 1 Introduction

Recommender systems are a core component of modern information platforms, helping users navigate large item catalogs by predicting personalized preferences. In implicit-feedback settings, where interactions such as clicks, purchases, and ratings are sparse and noisy, collaborative filtering (CF) methods learn user and item representations from historical user-item interactions. Despite strong performance, pure CF models often struggle when interaction data are limited (e.g., long-tail items) and when items exhibit semantic similarity that is not fully expressed by co-interaction patterns.

A natural complementary signal is item-side semantics available in metadata (titles, descriptions, attributes). Such information can provide a meaningful prior over item similarity and improve representation learning, especially under sparsity. Motivated by this, we study how to incorporate lightweight semantic signals into graph-based collaborative filtering.

This project builds upon LightGCN (4), a simplified and effective graph convolutional model for recommendation. We propose **SemGCN**, a semantic-enhanced LightGCN variant that (i) warm-starts item embeddings using text-derived semantic vectors and (ii) regularizes collaborative item representations to remain aligned with these semantic targets during training. SemGCN is simple to implement, adds minimal overhead via offline text encoding and caching, and can be trained end-to-end with the standard Bayesian Personalized Ranking (BPR) objective (1).

Our contributions are summarized as follows:

- We implement a semantic-augmented graph recommender (SemGCN) that integrates frozen text embeddings into LightGCN through initialization and an explicit alignment loss.
- We provide a reproducible experimental pipeline on the Amazon *Video Games* domain (13), including full-ranking evaluation for top- $K$  recommendation.
- We empirically show that adding semantic priors improves ranking quality over the vanilla LightGCN baseline.

## 2 Related Work

### 2.1 Collaborative Filtering and Matrix Factorization.

Collaborative Filtering (CF) remains the foundational strategy for modern recommender systems. Early memory-based approaches relied on identifying neighborhood similarities between users or items. However, latent factor models, most notably Matrix Factorization (MF) (5), revolutionized the field by projecting users and items into a shared low-dimensional vector space to capture latent preferences. To handle implicit feedback (e.g., clicks or views), pairwise learning-to-rank objectives such as Bayesian Personalized Ranking (BPR) (1) were developed to optimize the relative order of positive and negative items. With the advent of deep learning, Neural Collaborative Filtering (NCF) (2) proposed replacing the linear dot product of MF with multi-layer perceptrons (MLPs) to capture non-linear interactions, though recent studies suggest that simple dot products remain highly competitive when embeddings are learned effectively.

### 2.2 Graph-based Recommendation.

Since user-item interactions naturally form a bipartite graph, Graph Neural Networks (GNNs) have become the state-of-the-art for CF. These methods leverage the graph structure to propagate information among connected nodes, capturing high-order connectivity (e.g., "users who bought this also bought that"). Neural Graph Collaborative Filtering (NGCF) (3) explicitly modeled this recursive signal propagation by incorporating feature transformation and non-linear activation at each propagation layer. However, subsequent analysis in LightGCN (4) demonstrated that the feature transformation and non-linearities essential for node classification tasks are redundant and even harmful for collaborative filtering. LightGCN simplifies the architecture to linear neighborhood aggregation, significantly improving training efficiency and generalization. SemGCN adopts this streamlined architecture as its backbone.

### 2.3 Semantic and Text-aware Recommendation.

Pure CF methods suffer from the cold-start problem and fail to capture item similarity when interactions are sparse. Text-aware recommendation addresses this by utilizing item metadata (titles, descriptions, reviews). Early approaches like DeepCoNN (6) utilized Convolutional Neural Networks (CNNs) to extract features from review text. Recently, the emergence of Pretrained Language Models (PLMs) such as BERT (7) has shifted the paradigm, allowing for universal semantic representations to be transferred to downstream tasks.

More recent works have focused on bridging the gap between Collaborative Filtering and Large Language Models (LLMs). For instance, **CoLLM** (9) integrates collaborative information into LLMs by mapping embeddings from an external traditional model (like LightGCN) into the LLM's input token space, treating collaborative signals as a distinct modality to enhance warm-start performance. Similarly, **CLLM4Rec** (10) proposes a generative recommender that tightly integrates the ID paradigm with the LLM paradigm. It extends the LLM vocabulary with user/item tokens and employs a soft+hard prompting strategy to learn collaborative and content semantics jointly. While these methods (CoLLM, CLLM4Rec) typically focus on enhancing the LLM with collaborative signals (Graph/ID  $\rightarrow$  LLM), our work (SemGCN) takes the inverse, lightweight approach: we inject rich semantic priors from language models into a graph-based collaborative learner (LLM  $\rightarrow$  Graph), ensuring efficiency and scalability while retaining the benefits of semantic understanding.

### 3 SemGCN

We describe SemGCN, a semantic-augmented LightGCN instantiated in this project as LightGCN\_Semantic. SemGCN operates on a user–item bipartite graph built from implicit interactions and integrates item-side semantics extracted from text metadata.

#### 3.1 Problem formulation

Let  $\mathcal{U}$  denote the set of users and  $\mathcal{I}$  the set of items. Given observed implicit interactions  $\mathcal{R} \subseteq \mathcal{U} \times \mathcal{I}$ , the goal is to learn a scoring function  $f(u, i)$  that ranks the held-out relevant item(s) above irrelevant items for each user.

#### 3.2 LightGCN backbone

We construct a bipartite graph with adjacency matrix  $\mathbf{A}$  and use the normalized message passing operator

$$\mathbf{A} = \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}, \quad (1)$$

where  $\mathbf{D}$  is the degree matrix. LightGCN initializes user embeddings  $\mathbf{e}_u^{(0)} \in \mathbb{R}^d$  and item embeddings  $\mathbf{e}_i^{(0)} \in \mathbb{R}^d$ , and performs  $L$  layers of neighborhood aggregation:

$$\mathbf{E}^{(\ell+1)} = \tilde{\mathbf{A}} \mathbf{E}^{(\ell)}, \quad \ell = 0, \dots, L-1, \quad (2)$$

where  $\mathbf{E}^{(\ell)}$  concatenates all user and item embeddings at layer  $\ell$ . The final representation is obtained by averaging embeddings across layers:

$$\mathbf{E} = \frac{1}{L+1} \sum_{\ell=0}^L \mathbf{E}^{(\ell)}. \quad (3)$$

The recommendation score is computed by dot product  $f(u, i) = \langle \mathbf{e}_u, \mathbf{e}_i \rangle$ .

#### 3.3 Semantic item representations

For each item  $i \in \mathcal{I}$ , we obtain a text description (title and description fields) from the metadata file. We encode the text using a pretrained transformer encoder and mean pooling to obtain a sentence embedding  $\mathbf{h}_i$ . Since the transformer hidden size may differ from  $d$ , we apply a linear projection to the model embedding space:

$$\mathbf{s}_i = \mathbf{W} \mathbf{h}_i \in \mathbb{R}^d. \quad (4)$$

In our implementation, the encoder is frozen and semantic vectors  $\{\mathbf{s}_i\}$  are computed offline and cached for efficiency. If an item lacks metadata, we fall back to a placeholder text string for encoding.

#### 3.4 Semantic warm-start

SemGCN uses semantic vectors to initialize item embeddings:

$$\mathbf{e}_i^{(0)} \leftarrow \mathbf{s}_i, \quad (5)$$

while user embeddings are initialized as trainable parameters. This warm-start injects a semantic prior into the collaborative model before any graph propagation.

#### 3.5 Alignment regularization

To maintain consistency between collaborative signals and semantics, SemGCN adds an alignment term between the final collaborative item embeddings  $\mathbf{e}_i$  and their semantic targets  $\mathbf{s}_i$ . For a set of items  $\mathcal{B}$  appearing in the current training batch (positives and negatives), we define

$$\mathcal{L}_{\text{align}} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbf{e}_i - \mathbf{s}_i^2, \quad (6)$$

which corresponds to mean-squared error (MSE) alignment (cosine alignment is also supported by configuration).

Table 1: Statistics of the Video Games dataset used in this project.

Split	#Users	#Items	#Interactions
Train	94,762	25,527	625,062
Valid	94,762	18,799	94,762
Test	94,762	18,219	94,762

### 3.6 Training objective

SemGCN is trained with the BPR loss (1) over sampled triplets  $(u, i, j)$ , where  $i$  is a positive item for user  $u$  and  $j$  is a sampled negative item:

$$\mathcal{L}_{\text{BPR}} = - \sum_{(u, i, j)} \log \sigma(f(u, i) - f(u, j)). \quad (7)$$

We add standard  $\ell_2$  regularization  $\mathcal{L}_{\text{reg}}$  on the base embeddings and combine losses as

$$\mathcal{L} = \mathcal{L}_{\text{BPR}} + \mathcal{L}_{\text{reg}} + \lambda_{\text{align}} \mathcal{L}_{\text{align}}. \quad (8)$$

The hyperparameter  $\lambda_{\text{align}}$  controls the strength of semantic guidance.

## 4 Experiments

### 4.1 Experimental setup

**Dataset.** We conduct experiments on the Amazon *Video Games* domain (13). The interaction files provided in this project (`Video_Games.train.csv`, `Video_Games.valid.csv`, `Video_Games.test.csv`) follow a leave-one-out style split: each user has exactly one interaction in validation and one interaction in test. In our current training pipeline, the model is trained on the training split and evaluated on the test split. Table 1 reports basic statistics.

**Task and evaluation protocol.** We study implicit-feedback top- $K$  recommendation. For each test user, we compute scores for *all* items and filter out items the user interacted with in the training set. We then rank the remaining items and report metrics at  $K \in \{10, 20\}$ . Following the evaluation implementation in this project, we report Hit Ratio (HR), Precision, Recall, and NDCG. Since the test split contains one held-out interaction per user, HR is numerically identical to Recall under this protocol.

**Baselines.** We compare the proposed semantic-enhanced graph model against two collaborative filtering baselines implemented in this repository: (1) LightGCN (4), and (2) LightGCN\_Semantic.

**Models and hyperparameters.** All methods use embedding dimension  $d = 64$ , batch size 2048, learning rate  $10^{-3}$  (Adam), weight decay (L2 regularization)  $10^{-4}$ , and are trained for 100 epochs. LightGCN uses  $L = 3$  graph convolution layers, while LightGCN\_Semantic uses  $L = 2$  layers. Training uses the pairwise BPR objective (1) with a rating threshold of 3.0 to determine positive interactions during sampling. During training, we evaluate every 5 epochs and keep the best-performing checkpoint according to real-time top- $K$  ranking performance.

**Semantic augmentation (LightGCN\_Semantic).** LightGCN\_Semantic extends LightGCN by injecting item semantics derived from metadata. Specifically, we load item texts from `cleaned_meta_Video_Games.jsonl` using `parent_asin` identifiers, and construct each item description by concatenating title and description fields. We encode item texts using a Sentence-Transformers model (`a11-MiniLM-L6-v2`) (8; 11) implemented via the HuggingFace Transformers stack (12). Text embeddings are mean-pooled and linearly projected to  $d = 64$  dimensions, cached to disk, and treated as frozen semantic targets. These semantic vectors are used to (i) warm-start item embeddings and (ii) regularize collaborative item embeddings via an alignment loss:

$$\mathcal{L} = \mathcal{L}_{\text{BPR}} + \mathcal{L}_{\text{reg}} + \lambda_{\text{align}} \mathcal{L}_{\text{align}}, \quad (9)$$

where  $\mathcal{L}_{\text{align}}$  is mean-squared error (MSE) between collaborative and semantic item embeddings and  $\lambda_{\text{align}} = 0.1$ .

Table 2: Top- $K$  recommendation performance on Video Games (best runs recorded by this project).

Model	Recall@10	NDCG@10	Recall@20	NDCG@20
LightGCN	0.05450	0.02937	0.08265	0.03644
LightGCN_Semantic	0.05784	0.03095	0.08777	0.03847

**Main results.** Table 2 reports the best run saved in `results/` for each method. LightGCN\_Semantic improves over LightGCN by about 5–6% relatively on both Recall and NDCG.

## 5 Conclusions

This project implements and evaluates a semantic-augmented variant of LightGCN for implicit-feedback recommendation. The proposed LightGCN\_Semantic warm-starts item embeddings using frozen text-derived semantic representations and further constrains collaborative item embeddings via an MSE alignment regularizer. On the Amazon Video Games dataset, LightGCN\_Semantic consistently improves ranking quality over the vanilla LightGCN baseline, achieving relative gains of approximately 5–6% on both Recall and NDCG at  $K \in \{10, 20\}$ . These results support the hypothesis that lightweight semantic priors from item metadata can complement collaborative signals and improve recommendation accuracy.

## References

### References

- [1] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of UAI*, 2009.
- [2] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural Collaborative Filtering. In *Proceedings of WWW*, 2017.
- [3] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural Graph Collaborative Filtering. In *Proceedings of SIGIR*, 2019.
- [4] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of SIGIR*, 2020.
- [5] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer*, 42(8):30–37, 2009.
- [6] Lei Zheng, Vahid Noroozi, and Philip S. Yu. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. In *Proceedings of WSDM*, 2017.
- [7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*, 2019.
- [8] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of EMNLP-IJCNLP*, 2019.
- [9] Yang Zhang, Fuli Feng, Jizhi Zhang, Keqin Bao, Qifan Wang, and Xiangnan He. CoLLM: Integrating Collaborative Embeddings into Large Language Models for Recommendation. *arXiv preprint arXiv:2310.19488*, 2023.
- [10] Yaochen Zhu, Liang Wu, Qi Guo, Liangjie Hong, and Jundong Li. Collaborative Large Language Model for Recommender Systems. *arXiv preprint arXiv:2311.01343*, 2023.
- [11] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

- [12] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierrick Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, and Jamie Brew. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of EMNLP: System Demonstrations*, 2020.
- [13] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based Recommendations on Styles and Substitutes. In *Proceedings of SIGIR*, 2015.