

Revisiting LightGCN: Unexpected Inflexibility, Inconsistency, and A Remedy Towards Improved Recommendation (Supplementary Document)

Geon Lee
KAIST
Seoul, Republic of Korea
geonlee0325@kaist.ac.kr

Kyungho Kim
KAIST
Seoul, Republic of Korea
kkyungho@kaist.ac.kr

Kijung Shin
KAIST
Seoul, Republic of Korea
kijungs@kaist.ac.kr

1 FURTHER ANALYSIS OF LIGHTGCN

In this section, we provide additional results on our analysis of LightGCN discussed in the main paper. The LightGCN’s neighbor aggregation rule for each item i can be written as :

$$\mathbf{e}_i^{(k+1)} = \frac{1}{\sqrt{|\mathcal{N}_i|}} \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_u|}} \mathbf{e}_u^{(k)}. \quad (1)$$

1.1 Further Details on Observation 1

OBSERVATION 1. Empirically, the norm of the unscaled aggregated neighbor embeddings in Eq. (1) at each k^{th} ($k \geq 0$) layer exhibits a near linear relationship with the number of neighbors. Specifically,

$$\left\| \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_u|}} \mathbf{e}_u^{(k)} \right\| \approx |\mathcal{N}_i|, \quad (2)$$

where the symbol \approx denotes a strong positive linear correlation, with a high Pearson correlation coefficient.

Additional observational results. In Figure 1, we present visualizations of Observation 1 across five datasets. These results empirically support our observation that the norm of the unscaled aggregated embedding exhibits a near linear relationship with the number of neighbors, i.e., Eq. (2), for all $k \in \{0, \dots, K\}$. In addition, in Table 1, we measure the Pearson correlation coefficient between $\left\| \sum_{u \in \mathcal{N}_i} |\mathcal{N}_u|^{-0.5} \mathbf{e}_u^{(k)} \right\|$ and $|\mathcal{N}_i|$, which reveals a high correlation across all layers and datasets.

1.2 Detailed Observations

In this subsection, we provide our analysis of all datasets, including those not covered in the main paper.

Properties of LightGCN. We uncover unexpected inflexibility and inconsistency in the embedding behavior of LightGCN, and here are the properties.

PROPERTY 1. Empirically, the norms of the agg-based embedding $\mathbf{e}_i^{(k)}$ ’s at each k^{th} ($k \geq 1$) layer tend to satisfy:

$$\|\mathbf{e}_i^{(k)}\| \approx \sqrt{|\mathcal{N}_i|}.$$

This relationship symmetrically applies to the agg-based embeddings of each user u , i.e., $\|\mathbf{e}_u^{(k)}\| \approx \sqrt{|\mathcal{N}_u|}$ for $k \geq 1$.

PROPERTY 2. Empirically, the norms of the agg-free embedding $\mathbf{e}_i^{(0)}$ ’s do **NOT** exhibit a strong linear correlation with $\sqrt{|\mathcal{N}_i|}$ ’s, in contrast to agg-based embeddings (Property 1).

Table 1: The Pearson correlation between the norm of the unscaled aggregated embedding $\left\| \sum_{u \in \mathcal{N}_i} |\mathcal{N}_u|^{-0.5} \mathbf{e}_u^{(k)} \right\|$ and the number of neighbors $|\mathcal{N}_i|$ is high for $k \geq 0$ (Observation 1).

| Dataset | $k = 0$ | $k = 1$ | $k = 2$ | $k = 3$ |
|-----------|---------|---------|---------|---------|
| LastFM | 0.9944 | 0.9800 | 0.9986 | 0.9909 |
| MovieLens | 0.9829 | 0.9433 | 0.9993 | 0.9984 |
| Gowalla | 0.9799 | 0.9429 | 0.9747 | 0.9316 |
| Yelp | 0.9905 | 0.9086 | 0.9956 | 0.9014 |
| Amazon | 0.9831 | 0.9289 | 0.9852 | 0.9334 |

Table 2: The Pearson correlation between the norm of the scaled aggregated embedding $\|\mathbf{e}_i^{(k)}\|$ and $\sqrt{|\mathcal{N}_i|}$ is observed to be high for $k \geq 1$ (Property 1). However, this correlation is low (i.e., uncorrelated) for $k = 0$ (Property 2).

| Dataset | $k = 0$ | $k = 1$ | $k = 2$ | $k = 3$ |
|-----------|---------|---------------|---------------|---------------|
| LastFM | 0.3459 | 0.9509 | 0.7545 | 0.9802 |
| MovieLens | -0.2476 | 0.9372 | 0.8164 | 0.9982 |
| Gowalla | -0.1515 | 0.8266 | 0.7132 | 0.8283 |
| Yelp | -0.0115 | 0.9223 | 0.6778 | 0.9778 |
| Amazon | 0.1190 | 0.8867 | 0.7442 | 0.9277 |

PROPERTY 3 (NEAR-UNIFORM EFFECTIVE WEIGHTS). Empirically, effective weights at each k^{th} ($k \geq 1$) layer tend to be near uniform across neighboring users, i.e.,

$$\|\mathbf{e}_u^{(k)}\| / \sqrt{|\mathcal{N}_u|} \approx \sqrt{|\mathcal{N}_u|} / \sqrt{|\mathcal{N}_u|} = 1$$

PROPERTY 4. Empirically, for $k = 0$, effective weights tend to decrease with respect to the degrees of neighbors.

PROPERTY 5. Agg-free ($k = 0$) and agg-based embeddings ($1 \leq k \leq K$) are added with a specific weight ratio of $1 : K$.

Norm scaling of LightGCN. In Figure 2, we present the visualizations that validate Properties 1 and 2 of LightGCN across five datasets. We can visually confirm that, empirically, the norm of the scaled aggregated embedding $\mathbf{e}_i^{(k)}$ at each k^{th} ($k \geq 1$) layer tends to satisfy $\|\mathbf{e}_i^{(k)}\| \approx \sqrt{|\mathcal{N}_i|}$ (Property 1) while this relationship does not hold when $k = 0$ (Property 2). This is numerically validated in Table 2, where the Pearson correlation coefficients are high when $k \geq 1$, but are low when $k = 0$.

Neighbor weighting of LightGCN. In Figure 3, we present the visualizations that empirically validate Properties 3 and 4 of LightGCN across five datasets. We can visually confirm that the effective

Table 3: The Pearson correlation between the norm of the unscaled aggregated embedding $\|\sum_{u \in \mathcal{N}_i} |\mathcal{N}_u|^{\alpha-1} \mathbf{e}_u^{(k)}\|$ and $|\mathcal{N}_i|$ is observed to be high for various α s for $k \geq 1$ (Observation 2). Specifically, $k = 2$ is used below.

| Dataset | $\alpha = 0.0$ | $\alpha = 0.2$ | $\alpha = 0.4$ | $\alpha = 0.6$ | $\alpha = 0.8$ | $\alpha = 1.0$ |
|-----------|----------------|----------------|----------------|----------------|----------------|----------------|
| LastFM | 0.9907 | 0.9967 | 0.9982 | 0.9984 | 0.9985 | 0.9989 |
| MovieLens | 0.9968 | 0.9993 | 0.9998 | 0.9984 | 0.9957 | 0.9917 |
| Gowalla | 0.9361 | 0.9432 | 0.9615 | 0.9729 | 0.9803 | 0.9822 |
| Yelp | 0.9777 | 0.9888 | 0.9940 | 0.9958 | 0.9953 | 0.9954 |
| Amazon | 0.9671 | 0.9724 | 0.9751 | 0.9765 | 0.9822 | 0.9864 |

weight $\|\mathbf{e}_u^{(k)}\|/\sqrt{|\mathcal{N}_u|}$ at each k^{th} ($k \geq 1$) layer tend to be near uniform across neighboring users (Property 3) while when $k = 0$, it tends to decrease with the degree of the neighbor (Property 4).

2 FURTHER DETAILS ON LIGHTGCN++

In this section, we offer additional information about LightGCN++ to provide a deeper understanding of its mechanisms.

2.1 Generalized Norm Scaling

We first provide further details about the generalized neighbor aggregation for LightGCN, which extends its capabilities by allowing flexible norm scaling through the following aggregation rule.

$$\mathbf{e}_i^{(k+1)} = \frac{1}{|\mathcal{N}_i|^\alpha} \sum_{u \in \mathcal{N}_i} \frac{1}{|\mathcal{N}_u|^{1-\alpha}} \mathbf{e}_u^{(k)}, \quad (3)$$

where $\alpha \in [0, 1]$ is a controllable hyperparameter that provides flexibility and adaptability in scaling embedding norms.

To build upon this, we first establish that Observation 1 can be generalized to varying values of α in Eq. (3).

OBSERVATION 2. Empirically, the norm of the unscaled aggregated neighbor embeddings in Eq. (3) at each k^{th} ($k \geq 0$) layer exhibits a near linear relationship with the number of neighbors, across varying values of $\alpha \in [0, 1]$. Specifically,

$$\left\| \sum_{u \in \mathcal{N}_i} \frac{1}{|\mathcal{N}_u|^{1-\alpha}} \mathbf{e}_u^{(k)} \right\| \propto |\mathcal{N}_i|.$$

Results on Observation 2. In Figure 4, we present visualizations of Observation 2 across five datasets. The results empirically support our observation that the norm of the unscaled aggregated embedding in Eq. (3) exhibits a near-linear relationship with the number of neighbors. Specifically,

$$\left\| \sum_{u \in \mathcal{N}_i} \frac{1}{|\mathcal{N}_u|^{1-\alpha}} \mathbf{e}_u^{(k)} \right\| \propto |\mathcal{N}_i|$$

for $k \geq 0$. Additionally, Table 3 presents the Pearson correlation coefficient between $\|\sum_{u \in \mathcal{N}_i} |\mathcal{N}_u|^{\alpha-1} \mathbf{e}_u^{(k)}\|$ and $|\mathcal{N}_i|$, using $k = 2$ as an example. The results demonstrate high correlations across various values of α , numerically validating Observation 2.

Preservation of Properties 3 and 4. We explain how Properties 3 and 4 of LightGCN are maintained by the generalized neighbor aggregation rule. Specifically, Eq. (3) can be rewritten as follows:

$$\mathbf{e}_i^{(k+1)} = \frac{1}{|\mathcal{N}_i|^\alpha} \sum_{u \in \mathcal{N}_i} \frac{1}{|\mathcal{N}_u|^{1-\alpha}} \mathbf{e}_u^{(k)} = \frac{1}{|\mathcal{N}_i|^\alpha} \sum_{u \in \mathcal{N}_i} \frac{\|\mathbf{e}_u^{(k)}\|}{|\mathcal{N}_u|^{1-\alpha}} \frac{\mathbf{e}_u^{(k)}}{\|\mathbf{e}_u^{(k)}\|}.$$

This leads to the observation that the effective weight of neighbor u is $\|\mathbf{e}_u^{(k)}\|/|\mathcal{N}_u|^{1-\alpha}$. The near linear relation $\|\mathbf{e}_u^{(k)}\| \propto |\mathcal{N}_u|^{1-\alpha}$ for $k \geq 1$ leads to near-uniform effective weight $\|\mathbf{e}_u^{(k)}\|/|\mathcal{N}_u|^{1-\alpha} \propto |\mathcal{N}_u|^{1-\alpha}/|\mathcal{N}_u|^{1-\alpha} = 1$ for each neighbors u for $k \geq 1$ (Property 3 of LightGCN). On the other hand, when $k = 0$, there is no discernible pattern between $\|\mathbf{e}_u^{(k)}\|$ and $|\mathcal{N}_u|^{1-\alpha}$, and thus due to the denominator $|\mathcal{N}_u|^{1-\alpha}$, the effective weights tend to decrease w.r.t. the degree of neighbors (Property 4 of LightGCN).

Norm scaling w.r.t. α . In Figure 5, we demonstrate the controllable hyperparameter α in Eq. (3) allows for flexible adjustment of the norms of the agg-based embedding $\mathbf{e}_i^{(k)}$'s ($k \geq 1$). Specifically, for $k \geq 1$, $\|\mathbf{e}_i^{(k)}\| \propto |\mathcal{N}_i|^{1-\alpha}$ holds.

Effects of α . In Figure 6, we measure NDCG@20 across a range of α values in Eq. (3). The results demonstrate that adhering strictly to $\alpha = 0.5$, as defined in LightGCN, may not yield optimal performance across different datasets. This highlights the importance of adaptable and flexible norm scaling tailored to each dataset to achieve more accurate recommendations.

3 THEORETICAL ANALYSIS

In this section, we provide the theoretical analysis that supports our empirical observations. We first examine the norm of the aggregated embedding of uncorrelated (i.e., independent) neighbors. Then, we generalize this to neighbors with correlations. Lastly, we discuss the space complexity of LightGCN++.

3.1 Norm of Aggregated Embedding of Uncorrelated Neighbors

To demonstrate that our Observation 1 is not mathematically trivial, we show that embeddings from simple distributions do not exhibit the observation. For example, if we simply assume that neighbor embeddings are independently sampled from a normal distribution, the expected norm of the unscaled aggregated embeddings is not linear w.r.t. the number of neighbors.

THEOREM 1. Let M be a set of d -dimensional embeddings $\{\mathbf{x}_i\}_{i=1}^{|M|}$, where each dimension of $\mathbf{x}_i \in M$ is independently drawn from a normal distribution $N(0, \sigma^2)$. If the embeddings are uncorrelated (i.e., independent) sample-wise, the expected L2 norm of the unscaled aggregated embeddings in M is proportional to $\sqrt{|M|}$:

$$\mathbb{E} \left[\left\| \sum_{\mathbf{x}_i \in M} \mathbf{x}_i \right\| \right] \propto \sqrt{|M|}$$

Proof. Let $\mathbf{X}_j^{(M)} = \sum_{\mathbf{x}_i \in M} \mathbf{x}_{ij}$. Based on the property of the sum of normal distributions, $\mathbf{X}_j^{(M)} \sim N(0, |M|\sigma^2)$. Thus, $\mathbf{X}_1^{(M)}/\sqrt{|M|\sigma^2}, \dots, \mathbf{X}_d^{(M)}/\sqrt{|M|\sigma^2}$ are d independent random variables from $N(0, 1)$. This implies that the following statistic is distributed according to the chi distribution with d degrees of freedom:

$$\sqrt{\sum_{j=1}^d \left(\frac{\mathbf{X}_j^{(M)}}{\sqrt{|M|\sigma^2}} \right)^2} = \frac{1}{\sqrt{|M|\sigma^2}} \|\mathbf{X}^{(M)}\|.$$

From the expected value of the chi distribution:

$$\mathbb{E} \left[\frac{1}{\sqrt{|M|\sigma^2}} \|\mathbf{X}^{(M)}\| \right] = \sqrt{2} \frac{\Gamma\left(\frac{d+1}{2}\right)}{\Gamma\left(\frac{d}{2}\right)},$$

where $\Gamma(\cdot)$ is the gamma function. Thus, the expected value of L2 norm $\|\mathbf{X}^{(M)}\|$ of $\mathbf{X}^{(M)}$ is:

$$\mathbb{E} \left[\|\mathbf{X}^{(M)}\| \right] = \sqrt{2|M|\sigma^2} \frac{\Gamma\left(\frac{d+1}{2}\right)}{\Gamma\left(\frac{d}{2}\right)} \propto \sqrt{|M|}.$$

This completes the proof. \square

From Theorem 1, it is evident that embeddings that are simply sampled independently from a normal distribution do not exhibit the linear relationship in Eq. (9), implying that Observation 1 is indeed non-trivial. Next, we examine the variance of the L2 norm of the aggregated embedding.

THEOREM 2. Let M be a set of d -dimensional embeddings $\{\mathbf{x}_i\}_{i=1}^{|M|}$, where each dimension of $\mathbf{x}_i \in M$ is independently drawn from a normal distribution $N(0, \sigma^2)$. If the embeddings are uncorrelated (i.e., independent) sample-wise, the variance of the L2 norm of the unscaled aggregated embeddings in M is:

$$\text{Var} \left[\left\| \sum_{\mathbf{x}_i \in M} \mathbf{x}_i \right\| \right] = |M|\sigma^2 \left(d - \left(\sqrt{2} \frac{\Gamma\left(\frac{d+1}{2}\right)}{\Gamma\left(\frac{d}{2}\right)} \right)^2 \right)$$

Proof. Let $\mathbf{X}_j^{(M)} = \sum_{\mathbf{x}_i \in M} \mathbf{x}_{ij}$. Then, from the proof of Theorem 1, $\mathbf{X}_1^{(M)}/\sqrt{|M|\sigma^2}, \dots, \mathbf{X}_d^{(M)}/\sqrt{|M|\sigma^2}$ are d independent random variables from $N(0, 1)$. This implies that the following statistic is distributed according to the chi distribution with d degrees of freedom. From the variance of the chi distribution:

$$\text{Var} \left[\frac{1}{\sqrt{|M|\sigma^2}} \|\mathbf{X}^{(M)}\| \right] = d - \mathbb{E} \left[\frac{1}{\sqrt{|M|\sigma^2}} \|\mathbf{X}^{(M)}\| \right]^2 = d - \left(\sqrt{2} \frac{\Gamma\left(\frac{d+1}{2}\right)}{\Gamma\left(\frac{d}{2}\right)} \right)^2.$$

From the properties of variance, we have:

$$\text{Var} \left[\|\mathbf{X}^{(M)}\| \right] = |M|\sigma^2 \left(d - \left(\sqrt{2} \frac{\Gamma\left(\frac{d+1}{2}\right)}{\Gamma\left(\frac{d}{2}\right)} \right)^2 \right).$$

This completes the proof. \square

Since the expected value of the chi distribution is close to $\sqrt{d - \frac{1}{2}}$ for large d , the following approximation holds in the high-dimensional space:

$$\text{Var} \left[\left\| \sum_{\mathbf{x}_i \in M} \mathbf{x}_i \right\| \right] \approx \frac{|M|\sigma^2}{2}.$$

This indicates that the variance linearly increases with the number of neighbors, i.e., $|M|$.

3.2 Norm of Aggregated Embedding of Correlated Neighbors (Generalization)

We have observed that when embeddings are independently sampled from a normal distribution, the expected norm of the unscaled aggregated embeddings is sublinear w.r.t. the number of neighbors. Here, we demonstrate how the correlation between neighbors affects the linearity between the L2 norm of the aggregated embedding and the number of neighbors. Specifically, we introduce the correlation coefficient ρ between neighbors when computing the L2 norm of the aggregated pairs.

THEOREM 3. Let M be a set of d -dimensional embeddings $\{\mathbf{x}_i\}_{i=1}^{|M|}$ consisting of random variables. Assume that (1) at each j^{th} dimension, $\mathbf{x}_{1j}, \dots, \mathbf{x}_{|M|j}$ are drawn from a multivariate normal distribution $N(0, \Sigma)$ with $\Sigma_{ii} = \sigma^2 \forall i$, $\Sigma_{ik} = \rho\sigma^2 \forall i \neq k$ for some $\rho \geq 0$ (i.e., each $\mathbf{x}_{ij} \sim N(0, \sigma^2)$ and each pair \mathbf{x}_{ij} and \mathbf{x}_{kj} have correlation coefficient ρ), and (2) the dimensions are mutually independent and thus i.i.d. Then, the expected L2 norm of the unscaled aggregated embeddings in M follows the proportionality:

$$\mathbb{E} \left[\left\| \sum_{\mathbf{x}_i \in M} \mathbf{x}_i \right\| \right] \propto \sqrt{|M|(1-\rho) + |M|^2\rho}.$$

Note that if $\rho = 1$, then $\mathbb{E} \left[\left\| \sum_{\mathbf{x}_i \in M} \mathbf{x}_i \right\| \right] \propto |M|$.

Proof. Let $\mathbf{X}_j^{(M)} = \sum_{\mathbf{x}_i \in M} \mathbf{x}_{ij}$. Based on the property of the sum of normal distributions, the expected value of $\mathbf{X}_j^{(M)}$ is 0. The variance of $\mathbf{X}_j^{(M)}$ is:

$$\begin{aligned} \text{Var} \left[\mathbf{X}_j^{(M)} \right] &= \sum_{\mathbf{x}_i \in M} \text{Var} \left[\mathbf{x}_{ij} \right] + \sum_{\substack{\mathbf{x}_i, \mathbf{x}_k \in M \\ i \neq k}} \text{Cov} \left(\mathbf{x}_{ij}, \mathbf{x}_{kj} \right) \\ &= |M|\sigma^2 + |M|(|M| - 1)\rho\sigma^2 \\ &= \sigma^2 \left(|M|(1 - \rho) + |M|^2\rho \right). \end{aligned}$$

Thus, $\mathbf{X}_j^{(M)}/\sqrt{\sigma^2(|M|(1 - \rho) + |M|^2\rho)}$ for $j = 1 \dots d$ are d independent random variables from $N(0, 1)$. This implies that the following statistic is distributed according to the chi distribution with d degrees of freedom:

$$\sqrt{\sum_{j=1}^d \left(\frac{\mathbf{X}_j^{(M)}}{\sqrt{\sigma^2(|M|(1 - \rho) + |M|^2\rho)}} \right)^2} = \frac{\|\mathbf{X}^{(M)}\|}{\sqrt{\sigma^2(|M|(1 - \rho) + |M|^2\rho)}}.$$

From the expected value of the chi distribution:

$$\mathbb{E} \left[\frac{\|\mathbf{X}^{(M)}\|}{\sqrt{\sigma^2(|M|(1 - \rho) + |M|^2\rho)}} \right] = \sqrt{2} \frac{\Gamma\left(\frac{d+1}{2}\right)}{\Gamma\left(\frac{d}{2}\right)},$$

where $\Gamma(\cdot)$ is the gamma function. Thus, the expected value of L2 norm $\|\mathbf{X}^{(M)}\|$ of $\mathbf{X}^{(M)}$ is:

$$\begin{aligned} \mathbb{E} \left[\|\mathbf{X}^{(M)}\| \right] &= \sqrt{2\sigma^2(|M|(1 - \rho) + |M|^2\rho)} \frac{\Gamma\left(\frac{d+1}{2}\right)}{\Gamma\left(\frac{d}{2}\right)} \\ &\propto \sqrt{|M|(1 - \rho) + |M|^2\rho}. \end{aligned}$$

This completes the proof. \square

Theorem 3 suggests that if the embeddings are independent of each other (i.e., $\rho = 0$), as assumed in Theorem 1, the norm of the aggregated embedding is sublinear w.r.t. the number of neighbors, i.e., $\mathbb{E} \left[\left\| \mathbf{X}^{(M)} \right\| \right] \propto \sqrt{|M|}$. Conversely, if the embeddings exhibit complete linear relationships with each other (i.e., $\rho = 1$), the norm of the aggregated embedding is linear with the number of neighbors, i.e., $\mathbb{E} \left[\left\| \mathbf{X}^{(M)} \right\| \right] \propto |M|$. This indicates how the degree of correlation between embeddings influences the linearity of their aggregated norm. We conjecture that Observation 1 is attributed to the strong correlations between user/item embeddings. Next, we examine the variance of the L2 norm of the aggregated embedding.

THEOREM 4. *Let M be a set of d -dimensional embeddings $\{\mathbf{x}_i\}_{i=1}^{|M|}$ consisting of random variables. Assume that (1) at each j^{th} dimension, $\mathbf{x}_{1j}, \dots, \mathbf{x}_{|M|j}$ are drawn from a multivariate normal distribution $N(0, \Sigma)$ with $\Sigma_{ii} = \sigma^2 \forall i$, $\Sigma_{ik} = \rho\sigma^2 \forall i \neq k$ for some $\rho \geq 0$ (i.e., each $\mathbf{x}_{ij} \sim N(0, \sigma^2)$ and each pair \mathbf{x}_{ij} and \mathbf{x}_{kj} have correlation coefficient ρ), and (2) the dimensions are mutually independent and thus i.i.d. Then, the variance of the L2 norm of the unscaled aggregated embeddings in M is:*

$$\text{Var} \left[\left\| \sum_{\mathbf{x}_i \in M} \mathbf{x}_i \right\| \right] = \sigma^2 \left(|M|(1 - \rho) + |M|^2 \rho \right) \left(d - \left(\sqrt{2} \frac{\Gamma\left(\frac{d+1}{2}\right)}{\Gamma\left(\frac{d}{2}\right)} \right)^2 \right)$$

Proof. Let $\mathbf{X}_j^{(M)} = \sum_{\mathbf{x}_i \in M} \mathbf{x}_{ij}$. Then, from the proof of Theorem 3, $\mathbf{X}_j^{(M)} / \sqrt{\sigma^2 (|M|(1 - \rho) + |M|^2 \rho)}$ for $j = 1, \dots, d$ are d independent random variables from $N(0, 1)$. This implies that the following statistic is distributed according to the chi distribution with d degrees of freedom. From the variance of the chi distribution:

$$\begin{aligned} \text{Var} \left[\frac{1}{\sqrt{\sigma^2 (|M|(1 - \rho) + |M|^2 \rho)}} \left\| \mathbf{X}^{(M)} \right\| \right] &= \\ d - \mathbb{E} \left[\frac{1}{\sqrt{\sigma^2 (|M|(1 - \rho) + |M|^2 \rho)}} \left\| \mathbf{X}^{(M)} \right\| \right]^2 &= d - \left(\sqrt{2} \frac{\Gamma\left(\frac{d+1}{2}\right)}{\Gamma\left(\frac{d}{2}\right)} \right)^2. \end{aligned}$$

From the properties of variance, we have:

$$\text{Var} \left[\left\| \mathbf{X}^{(M)} \right\| \right] = \sigma^2 \left(|M|(1 - \rho) + |M|^2 \rho \right) \left(d - \left(\sqrt{2} \frac{\Gamma\left(\frac{d+1}{2}\right)}{\Gamma\left(\frac{d}{2}\right)} \right)^2 \right).$$

This completes the proof. \square

Since the expected value of the chi distribution is close to $\sqrt{d - \frac{1}{2}}$ for large d , the following approximation holds in the high-dimensional space:

$$\text{Var} \left[\left\| \sum_{\mathbf{x}_i \in M} \mathbf{x}_i \right\| \right] \approx \frac{\sigma^2 (|M|(1 - \rho) + |M|^2 \rho)}{2}.$$

4 DETAILS ON EXPERIMENTAL SETTINGS

In this section, we offer a detailed description of the settings used in our experiments.

4.1 Experimental Settings

Datasets. We used five benchmark datasets, LastFM, MovieLens, Gowalla, Yelp, and Amazon to conduct experiments. The statistics of each dataset are reported in Table 4.

Baselines. We compared LightGCN++ against the following baseline methods:

- **BPRMF** [6] optimizes the BPR loss to learn the embeddings for users and items using matrix factorization (MF).
- **NeuMF** [4] uses an MLP instead of the dot product in the MF to learn the matching function between users and items.
- **NGCF** [7] incorporates both feature transformation and nonlinearities in its GNN framework.
- **LR-GCCF** [2] removes nonlinearities while incorporating feature transformation within its GNN framework.
- **HCCF** [8] exploits contrastive learning to integrate an explicit interaction graph with the learned implicit hypergraph structure.
- **LightGCL** [1] utilizes an SVD-reconstructed graph as an augmented view in its contrastive learning framework.
- **LightGCN** [3] simplifies NGCF by removing feature transformation and nonlinearities (see Section 2.1 in the main paper for more details).
- **NCL** [5] incorporates both structural and semantic neighbors for each node to construct contrastive pairs.
- **SimGCL** [10] applies random noise into embeddings to create an augmented view for its contrastive learning framework.
- **XSimGCL** [9] simplifies SimGCL by directly applying noises into the embeddings that are used for making predictions.

The source code utilized for each baseline is listed in Table 5. For LR-GCCF [2], we manually implemented it based on the official PyTorch code of LightGCN. For the methods that equip LightGCN (i.e., NCL [5], SimGCL [10], XSimGCL [9]), we integrated their original contrastive loss functions into the LightGCN's framework.

Hyperparameter search. The dataset was divided into training, validation, and test sets following a 7:1:2 ratio. We then searched the hyperparameter settings for each method that yields the best NDCG@20 for the validation set. We initialized the learnable parameters (i.e., embeddings \mathbf{E} at the initial layer) following a normal distribution. The embedding dimension is set to 64, the batch size is set to 2048, and the learning rate is set to 0.001 with a regularization coefficient λ of 0.0001. For GNN-based models, we used the number of layers $K = 2$. The search space for model-specific hyperparameters for each method is conducted as follows:

- **HCCF** [8]: The four hyperparameters, λ_1 (weight decay), λ_2 (weight for contrastive learning loss), τ (temperature for contrastive loss), γ (dropout edge preservation ratio) are selected from $\{0, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\}$, $\{10^{-3}, 10^{-2}, 10^{-1}, 0.1, 0.2, 0.3\}$, $\{0.1, 0.3, 1, 3, 10\}$, and $\{0.25, 0.5, 0.75, 1.0\}$, respectively.
- **LightGCL** [1]: The four hyperparameters, λ_1 (weight of contrastive loss), λ_2 (weight of L2 regularization¹), τ (temperature for contrastive loss), and p (dropout rate) are selected from $\{0.001, 0.01, 0.1\}$, $\{10^{-8}, 10^{-7}\}$, $\{0.2, 0.5\}$, and $\{0.0, 0.25\}$.
- **NCL** [5]: There are three hyperparameters, λ_1 (weight of structure-contrastive loss), λ_2 (weight of prototype-contrastive loss), and

¹In the official implementation of LightGCL, the computation of L2 regularization differs from that of LightGCN. Thus, we manually tuned this hyperparameter instead of using the default setting (i.e., 0.0001).

Table 4: Dataset Statistics.

| Dataset | # User | # Item | # Interaction | Density |
|-----------|--------|--------|---------------|---------|
| LastFM | 1,885 | 17,388 | 91,779 | 0.00280 |
| MovieLens | 6,039 | 3,628 | 836,478 | 0.03817 |
| Gowalla | 29,858 | 40,981 | 1,027,370 | 0.00084 |
| Yelp | 31,668 | 38,048 | 1,561,406 | 0.00129 |
| Amazon | 52,643 | 91,599 | 2,704,860 | 0.00056 |

Table 5: Source code URLs of baselines.

| Method | Source Code URL |
|--------------|---|
| BPRMF [6] | https://github.com/gusye1234/LightGCN-PyTorch |
| NeuMF [4] | https://github.com/guoyang9/NCF |
| NGCF [7] | https://github.com/huangtinglin/NGCF-PyTorch |
| LR-GCCF [2] | https://github.com/gusye1234/LightGCN-PyTorch |
| HCCF [8] | https://github.com/akaxlh/HCCF |
| LightGCL [1] | https://github.com/HKUDS/LightGCL |
| LightGCN [3] | https://github.com/gusye1234/LightGCN-PyTorch |
| NCL [5] | https://github.com/RUCAIBox/NCL |
| SimGCL [10] | https://github.com/Coder-Yu/SELFRec |
| XSimGCL [9] | https://github.com/Coder-Yu/SELFRec |

K (number of prototypes). We tune them from $\{0.001, 0.0001\}$, $\{0.001, 0.0001\}$, and $\{100, 1000\}$, respectively.

- **SimGCL** [10]: The two hyperparameters, ϵ (magnitude of the noise) and λ (weight of contrastive loss) are selected from $\{0.01, 0.05, 0.1, 0.5\}$ and $\{0.01, 0.05, 0.1, 0.5\}$, respectively.
- **XSimGCL** [9]: The three hyperparameters, ϵ (magnitude of the noise), λ (weight of contrastive loss), and l^* (target layer to contrast with the final embedding) are selected from $\{0.01, 0.05, 0.1, 0.5\}$, $\{0.01, 0.05, 0.1, 0.5\}$, and $\{1, 2\}$, respectively.
- **LightGCN++ (proposed)**: The three hyperparameters, α , β , and γ are selected from $\{0.4, 0.5, 0.6\}$, $\{-0.1, 0.0, 0.1\}$, and $\{0.0, 0.1, 0.2\}$.

Implementation. We implemented LightGCN++ based on the official PyTorch implementation of LightGCN². It is easy to implement based on LightGCN's framework, with the following modifications:

- Instead of normalizing with $\mathbf{D}^{-0.5}\mathbf{A}\mathbf{D}^{-0.5}$, where \mathbf{A} is the adjacency matrix and \mathbf{D} is a diagonal degree matrix, we apply normalization using $\mathbf{D}^{-\alpha}\mathbf{A}\mathbf{D}^{-\beta}$ using hyperparameters α and β .
- At the beginning of each k^{th} ($k \geq 0$) layer, we normalize the embedding $\mathbf{e}_i^{(k)}$ to a unit vector $\mathbf{e}_i^{(k)} / \|\mathbf{e}_i^{(k)}\|$.
- After aggregating neighbors over K layers, we apply a weighted sum to $\mathbf{e}_i^{(0)}$ and $\mathbf{e}_i^{(1)} + \dots + \mathbf{e}_i^{(K)}$ using the hyperparameter γ .

For **reproducibility**, we make our code and dataset available at <https://github.com/geon0325/LightGCNpp>.

4.2 Hyperparameter Tuning Strategies

Here, we share some simple strategies for tuning α , β , and γ in LightGCN++ for its practical usability.

- **Tuning α .** We advise users to adjust α based on the long-tailed characteristics of the item popularity (i.e., degree) distribution. Specifically, increasing α (i.e., $\alpha \rightarrow 1$) leads to a fairer recommendation that is equally likely to recommend both popular and unpopular items. In contrast, decreasing α (i.e., $\alpha \rightarrow 0$) biases the system towards recommending more popular items.

²<https://github.com/gusye1234/LightGCN-PyTorch>

Table 6: The searched hyperparameters for each dataset.

| Dataset | α | β | γ |
|-----------|----------|---------|----------|
| LastFM | 0.6 | -0.1 | 0.2 |
| MovieLens | 0.5 | 0.0 | 0.0 |
| Gowalla | 0.6 | -0.1 | 0.2 |
| Yelp | 0.6 | -0.1 | 0.1 |
| Amazon | 0.6 | -0.1 | 0.2 |

- **Tuning β .** According to our experiments, setting $0 \geq \beta \geq 1$ generally leads to improvements. This suggests that reducing the influence of high-degree neighbors enhances accuracy.
- **Tuning γ .** We recommend that users begin by tuning γ from 0, which excludes the embeddings at the initial layer (i.e., $\mathbf{e}_i^{(0)}$) from the layer-wise aggregation. Then, gradually increasing γ may enhance performance, depending on the dataset.

5 ADDITIONAL EXPERIMENTAL RESULTS

In this section, we provide additional experimental results.

5.1 Additional Results of LightGCN++

Tables 9 and 10 present the performance of LightGCN++ and its baselines across five datasets, evaluating the performance in terms of top-10 and top-40 recommendations, respectively.

5.2 Parameter Sensitivity Analysis

We examine the influence of the controllable hyperparameters α , β , and γ on the performance of LightGCN++. We evaluate the performance of LightGCN++ for $\alpha \in \{0.0, 0.1, \dots, 1.0\}$, $\beta \in \{-0.25, -0.2, \dots, 0.25\}$, and $\gamma \in \{0.0, 0.1, \dots, 1.0\}$. We include results in Figures 7, 8, and 9, regarding the sensitivity analysis for parameters α , β , and γ , respectively. The results indicate the importance of the flexible and adaptive adjustment for each dataset.

Optimal α , β , and γ . In Table 6, we report the best hyperparameters searched from $\alpha \in \{0.4, 0.5, 0.6\}$, $\beta \in \{-0.1, 0.0, 0.1\}$, and $\gamma \in \{0.0, 0.1, 0.2\}$, for each dataset. Notably, $\alpha = 0.6$, $\beta = -0.1$, and $\gamma = 0.2$ worked best in three out of five datasets, and we recommend the users use this configuration as default.

Potential rationales. Since α and β control the embedding norms and the effective weights of neighbors based on item/user degree, we hypothesize that the optimal values for α , β , and γ are influenced by the skewness of the degree distributions. The skewness values are 4.384 for LastFM, 2.679 for MovieLens, 7.851 for Gowalla, 4.238 for Yelp, and 5.656 for Amazon. The lower skewness in MovieLens indicates a more balanced degree distribution, leading to different hyperparameters ($\alpha = 0.5$, $\beta = 0.0$, $\gamma = 0.0$) compared to the other datasets ($\alpha = 0.6$, $\beta = -0.1$, $\gamma = 0.1$ or 0.2).

5.3 Potential Extensions of LightGCN++

He et al. [3] reveal that feature transformation (FT) significantly reduces the effectiveness of GNN-based recommender systems, introducing unnecessary complexity. In this subsection, we further elaborate on the potential extension of adopting feature transformation (FT) and nonlinear activations (NA).

Feature Transformation in LightGCN++. LightGCN++ can be

Table 7: While LightGCN encounters significant performance degradation (in terms of NDCG@20) upon incorporating feature transformation (FT) and/or nonlinear activation (NA), LightGCN++ effectively mitigates this degradation.

| Dataset | LastFM | MovieLens | Gowalla | Yelp | Amazon |
|-------------|--------------|--------------|--------------|--------------|--------------|
| LightGCN | 0.2427 | 0.3010 | 0.1426 | 0.0449 | 0.0274 |
| + FT | 0.1869 | 0.2814 | 0.1075 | 0.0388 | 0.0201 |
| Degradation | 22.99% | 6.51% | 24.61% | 13.58% | 26.64% |
| + FT + NA | 0.1980 | 0.2862 | 0.1083 | 0.0377 | 0.0188 |
| Degradation | 18.41% | 4.91% | 24.05% | 16.03% | 31.38% |
| LightGCN++ | 0.2624 | 0.3275 | 0.1469 | 0.0529 | 0.0294 |
| + FT | 0.2429 | 0.3098 | 0.1420 | 0.0494 | 0.0282 |
| Degradation | 7.43% | 5.40% | 3.33% | 6.61% | 4.08% |
| + FT + NA | 0.2457 | 0.3099 | 0.1362 | 0.0486 | 0.0280 |
| Degradation | 6.36% | 5.37% | 7.28% | 8.12% | 4.76% |

Table 8: LightGCN++’s adaptive pooling for layer-wise embedding aggregation, tuning γ , is more effective than both mean pooling and learnable pooling approaches.

| Dataset | LastFM | MovieLens | Gowalla | Yelp | Amazon |
|-------------------|---------------|---------------|---------------|---------------|---------------|
| Mean Pooling | 0.2614 | 0.3217 | 0.1436 | 0.0507 | 0.0282 |
| Learnable Pooling | 0.2590 | <u>0.3227</u> | 0.1389 | <u>0.0524</u> | <u>0.0283</u> |
| Adaptive Pooling | 0.2624 | 0.3275 | 0.1469 | 0.0529 | 0.0294 |

extended by incorporating FT as follows:

$$\mathbf{e}_i^{(k+1)} = \frac{1}{|\mathcal{N}_i|^\alpha} \sum_{u \in \mathcal{N}_i} \frac{1}{|\mathcal{N}_u|^\beta} \frac{\mathbf{W}^{(k)} \mathbf{e}_u^{(k)}}{\|\mathbf{W}^{(k)} \mathbf{e}_u^{(k)}\|}$$

where $\mathbf{W}^{(k)}$ is the trainable weight matrix used for FT at each k^{th} layer. Note that the projected embedding of each neighbor u is normalized, and an adjustable effective weight of $1/|\mathcal{N}_u|^\beta$ is applied. Moreover, the term, $1/|\mathcal{N}_i|^\alpha$ is used for adjusting embedding norms. **Nonlinear activation in LightGCN++.** LightGCN++ can be extended by incorporating NA together with FT as follows:

$$\mathbf{e}_i^{(k+1)} = \phi \left(\frac{1}{|\mathcal{N}_i|^\alpha} \sum_{u \in \mathcal{N}_i} \frac{1}{|\mathcal{N}_u|^\beta} \frac{\mathbf{W}^{(k)} \mathbf{e}_u^{(k)}}{\|\mathbf{W}^{(k)} \mathbf{e}_u^{(k)}\|} \right),$$

where $\phi(\cdot)$ is the nonlinear activation function (e.g., Leaky ReLU), and $\mathbf{W}^{(k)}$ is the weight matrix at the k^{th} layer.

Experimental results. In Table 7, we report the performance (in terms of NDCG@20) for both LightGCN and LightGCN++ when equipped with FT and/or NA. For NA, we used the Leaky ReLU. The results show that LightGCN++ effectively reduces the performance degradation encountered in LightGCN with FT and/or NA.

5.4 Learnable Embedding Pooling

To evaluate the effectiveness of the adaptive layer-wise embedding aggregation approach used in LightGCN++, we compare it with two intuitive pooling methods. Specifically, mean pooling assigns equal importance across all layers, whereas learnable pooling learns individual weights for each layer. As shown in Table 8, adaptive pooling, upon fine-tuning γ , achieves the best results in terms of NDCG@20. This indicates that addressing the disparities in norm scaling between agg-free and agg-based embeddings by properly balancing them is effective for generating the final embeddings.

REFERENCES

- [1] Xuheng Cai, Chao Huang, Lianghao Xia, and Xubin Ren. 2022. LightGCL: Simple Yet Effective Graph Contrastive Learning for Recommendation. In *ICLR*.
- [2] Lei Chen, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. 2020. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In *AAAI*.
- [3] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*.
- [4] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*.
- [5] Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. 2022. Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In *WWW*.
- [6] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*.
- [7] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *SIGIR*.
- [8] Lianghao Xia, Chao Huang, Yong Xu, Jiashu Zhao, Dawei Yin, and Jimmy Huang. 2022. Hypergraph contrastive collaborative filtering. In *SIGIR*.
- [9] Junliang Yu, Xin Xia, Tong Chen, Lizhen Cui, Nguyen Quoc Viet Hung, and Hongzhi Yin. 2023. XSimGCL: Towards extremely simple graph contrastive learning for recommendation. *TKDE* (2023).
- [10] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 2022. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *SIGIR*.

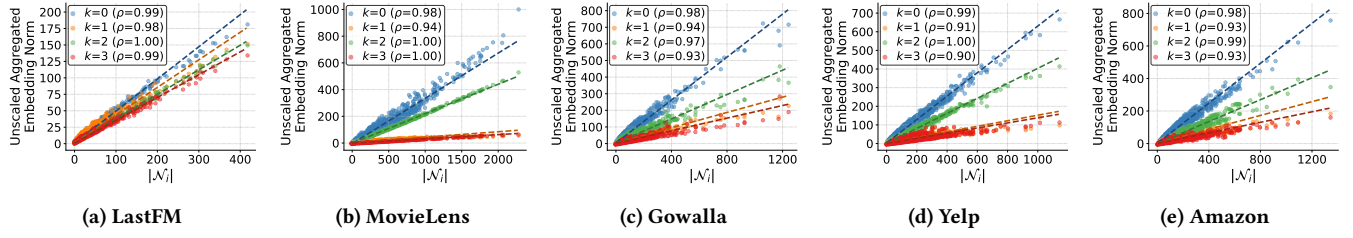


Figure 1: The norm of the unscaled aggregated embedding ($\sum_{u \in N_i} |N_u|^{-0.5} \mathbf{e}_u^{(k)}$) in Eq. (1) tends to be proportional to the number of neighbors $|N_i|$ for $k \geq 0$ (Observation 1). The symbol ρ represents a Pearson correlation coefficient.

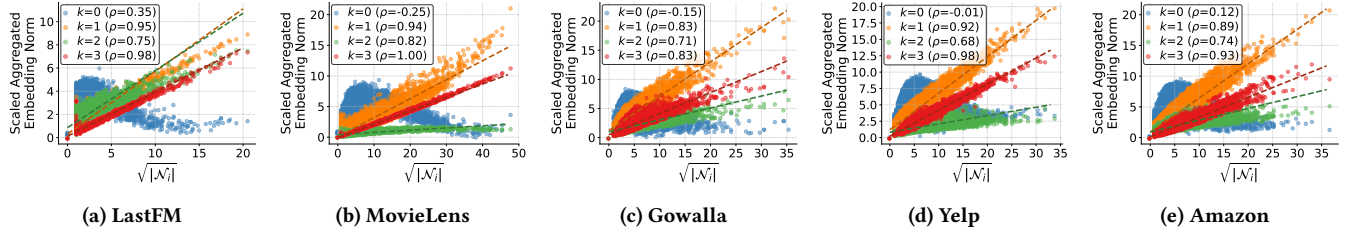


Figure 2: The norm of the scaled aggregated embedding ($\mathbf{e}_i^{(k)}$) in Eq. (1) tends to be proportional to $\sqrt{|N_i|}$ for $k \geq 1$ (Property 1). This relationship does not hold when $k = 0$ (Property 2). The symbol ρ represents a Pearson correlation coefficient.

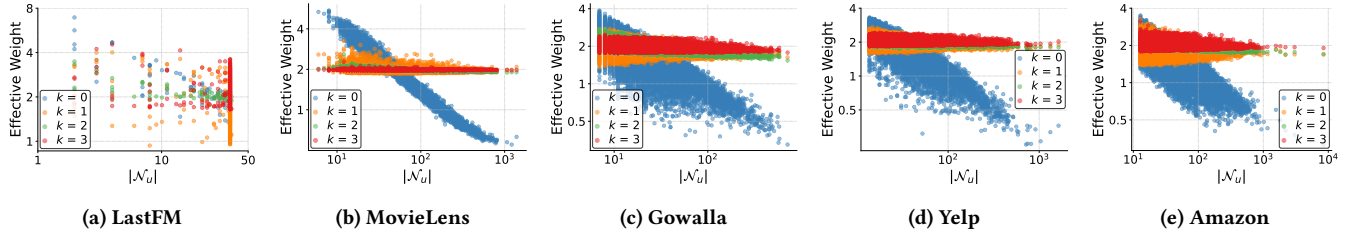


Figure 3: The effective weight $\|\mathbf{e}_u^{(k)}\|/\sqrt{|N_u|}$ of neighbor u tends to be uniform when $k \geq 1$ (Property 3). When $k = 0$, the effective weight decreases with respect to the degree (Property 4).

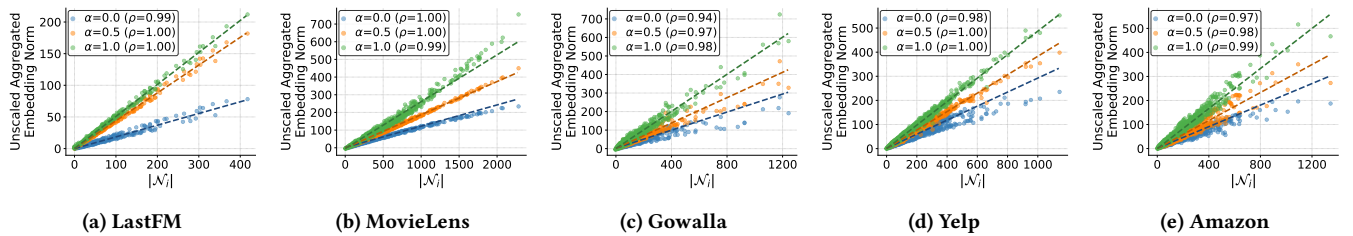


Figure 4: The norm of the aggregated neighbor embeddings ($\sum_{u \in N_i} |N_u|^{\alpha-1} \mathbf{e}_u^{(k)}$) in Eq. (3) is proportional to the number of neighbors $|N_i|$ for $k \geq 0$ (Observation 2) in five datasets. The symbol ρ represents a Pearson correlation coefficient.

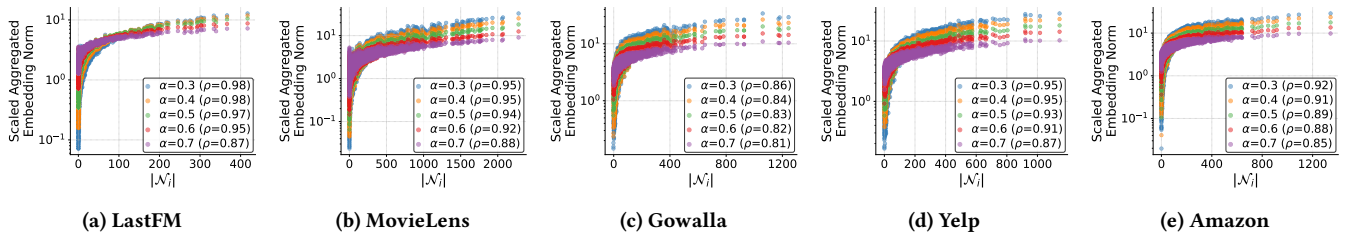


Figure 5: The controllable parameter α in Eq. (3) allows for flexible adjustment of norm scaling of agg-based embeddings. The symbol ρ represents a Pearson correlation coefficient.

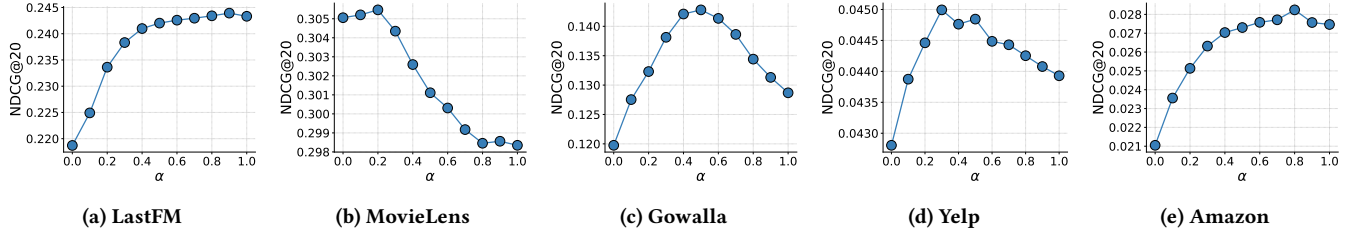


Figure 6: The value of α in Eq. (3), which determines the scaling of embedding norms, varies to achieve the best performance (in terms of NDCG@20) across different datasets.

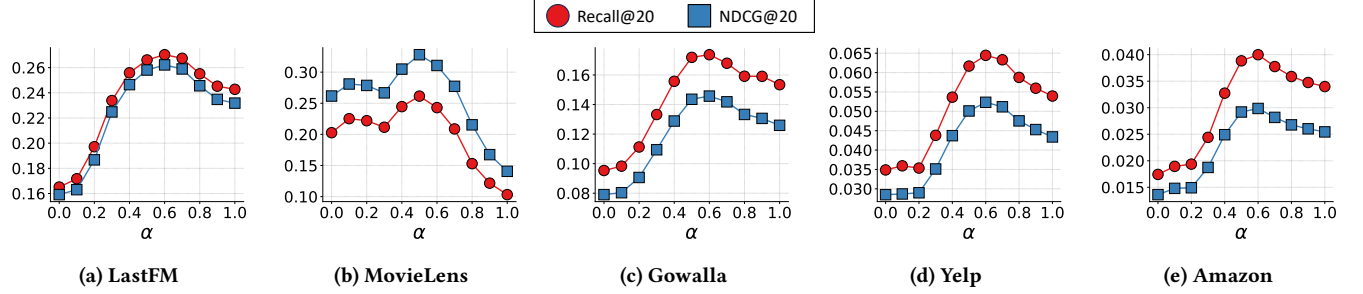


Figure 7: Impact of α on the performance of LightGCN++. The optimal α is different across datasets.

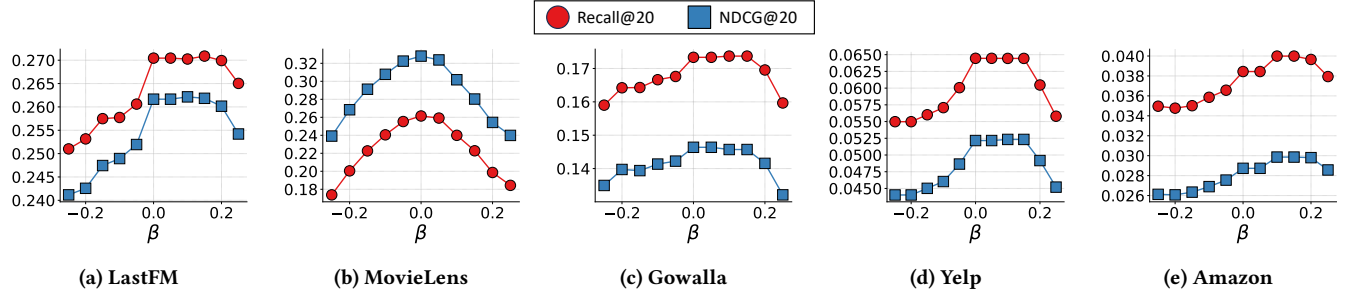


Figure 8: Impact of β on the performance of LightGCN++. The optimal β is different across datasets.

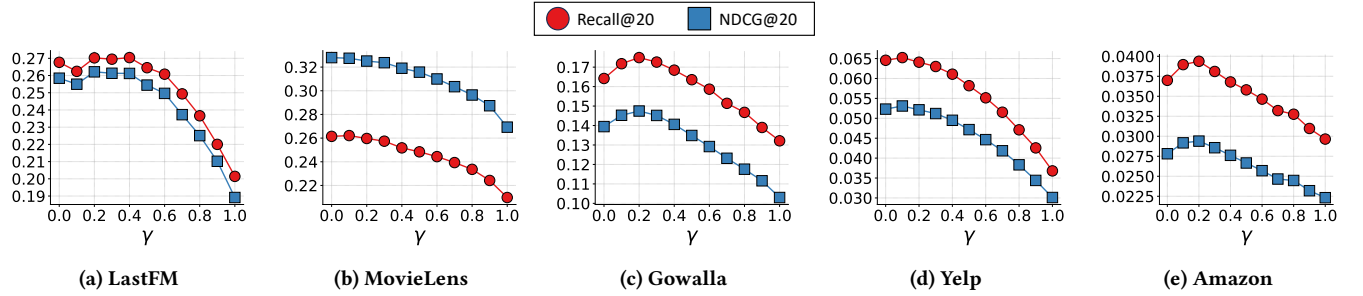


Figure 9: Impact of γ on the performance of LightGCN++. The optimal γ is different across datasets.

Table 9: LightGCN++ consistently and significantly outperforms LightGCN in terms of Recall@10 and NDCG@10. State-of-the-art methods enhanced with LightGCN++ (i.e., NCL++, SimGCL++, and XSimGCL++) also outperform their counterparts with LightGCN. * and ** denote $p < 0.01$ and $p < 0.001$ for a one-tailed t-test, indicating that a method with LightGCN++ significantly outperforms its counterpart with LightGCN. For each dataset, the best performance is in bold and the second-best is underlined.

| Dataset Metric | LastFM | | MovieLens | | Gowalla | | Yelp | | Amazon | |
|-------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| | Recall@10 | NDCG@10 | Recall@10 | NDCG@10 | Recall@10 | NDCG@10 | Recall@10 | NDCG@10 | Recall@10 | NDCG@10 |
| BPRMF [6] | 0.1361 | 0.1532 | 0.1291 | 0.2677 | 0.0895 | 0.0899 | 0.0209 | 0.0242 | 0.0165 | 0.0170 |
| NeuMF [4] | 0.1498 | 0.1688 | 0.1345 | 0.2707 | 0.0842 | 0.0871 | 0.0207 | 0.0232 | 0.0131 | 0.0135 |
| NGCF [7] | 0.1562 | 0.1769 | 0.1420 | 0.2872 | 0.0916 | 0.0942 | 0.0264 | 0.0294 | 0.0176 | 0.0177 |
| LR-GCCF [2] | 0.1367 | 0.1584 | 0.1025 | 0.2281 | 0.0675 | 0.0753 | 0.0224 | 0.0260 | 0.0095 | 0.0109 |
| HCCF [8] | 0.1520 | 0.1730 | 0.1397 | 0.2900 | 0.0827 | 0.0912 | 0.0365 | 0.0414 | 0.0190 | 0.0197 |
| LightGCL [1] | 0.1741 | 0.1997 | 0.1460 | 0.2907 | 0.1183 | 0.1274 | 0.0364 | 0.0411 | 0.0239 | 0.0243 |
| LightGCN [3] | 0.1741 | 0.1996 | 0.1495 | 0.2977 | 0.1182 | 0.1284 | 0.0321 | 0.0363 | 0.0206 | 0.0209 |
| LightGCN++ | 0.1889** | 0.2171** | 0.1667** | 0.3243** | 0.1219** | 0.1321** | 0.0380** | 0.0428** | 0.0221** | 0.0223** |
| Improvement | 8.50% | 8.77% | 11.51% | 8.94% | 3.13% | 2.88% | 18.38% | 17.91% | 7.28% | 6.70% |
| NCL [5] | 0.1768 | 0.2024 | 0.1507 | 0.3002 | 0.1193 | 0.1283 | 0.0342 | 0.0385 | 0.0221 | 0.0222 |
| NCL++ | 0.1894** | 0.2177** | 0.1677** | 0.3257** | 0.1234** | 0.1331** | 0.0401** | 0.0451** | 0.0241** | 0.0239** |
| Improvement | 7.13% | 7.56% | 11.28% | 8.49% | 3.44% | 3.74% | 17.25% | 17.14% | 9.05% | 7.66% |
| SimGCL [10] | 0.1795 | 0.2049 | 0.1647 | 0.3186 | 0.1185 | 0.1274 | 0.0379 | 0.0427 | 0.0239 | 0.0242 |
| SimGCL++ | 0.1880** | 0.2151** | 0.1679** | 0.3254** | 0.1191 | 0.1283 | 0.0385** | 0.0435** | <u>0.0259**</u> | <u>0.0258**</u> |
| Improvement | 4.74% | 4.98% | 1.94% | 2.13% | 0.51% | 0.71% | 1.58% | 1.87% | 8.37% | 6.61% |
| XSimGCL [9] | 0.1801 | 0.2062 | 0.1657 | 0.3216 | 0.1173 | 0.1253 | 0.0380 | 0.0427 | 0.0226 | 0.0227 |
| XSimGCL++ | 0.1893** | <u>0.2173**</u> | 0.1671* | 0.3242* | 0.1188** | 0.1283** | <u>0.0397**</u> | <u>0.0447**</u> | 0.0262** | 0.0263** |
| Improvement | 5.11% | 5.38% | 0.84% | 0.81% | 1.28% | 2.39% | 4.47% | 4.68% | 15.93% | 15.86% |

Table 10: LightGCN++ consistently and significantly outperforms LightGCN in terms of Recall@40 and NDCG@40. State-of-the-art methods enhanced with LightGCN++ (i.e., NCL++, SimGCL++, and XSimGCL++) also outperform their counterparts with LightGCN. * and ** denote $p < 0.01$ and $p < 0.001$ for a one-tailed t-test, indicating that a method with LightGCN++ significantly outperforms its counterpart with LightGCN. For each dataset, the best performance is in bold and the second-best is underlined.

| Dataset Metric | LastFM | | MovieLens | | Gowalla | | Yelp | | Amazon | |
|-------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| | Recall@40 | NDCG@40 | Recall@40 | NDCG@40 | Recall@40 | NDCG@40 | Recall@40 | NDCG@40 | Recall@40 | NDCG@40 |
| BPRMF [6] | 0.2784 | 0.2235 | 0.3192 | 0.2914 | 0.1950 | 0.1238 | 0.0631 | 0.0401 | 0.0507 | 0.0303 |
| NeuMF [4] | 0.3065 | 0.2457 | 0.3320 | 0.2998 | 0.1767 | 0.1163 | 0.0652 | 0.0401 | 0.0409 | 0.0244 |
| NGCF [7] | 0.3218 | 0.2586 | 0.3439 | 0.3134 | 0.1986 | 0.1281 | 0.0794 | 0.0496 | 0.0544 | 0.0322 |
| LR-GCCF [2] | 0.2717 | 0.2248 | 0.2591 | 0.2433 | 0.1352 | 0.0948 | 0.0668 | 0.0427 | 0.0280 | 0.0179 |
| HCCF [8] | 0.3152 | 0.2529 | 0.3366 | 0.3131 | 0.1680 | 0.1165 | 0.1046 | 0.0672 | 0.0590 | 0.0355 |
| LightGCL [1] | 0.3480 | 0.2854 | 0.3540 | 0.3219 | 0.2389 | 0.1643 | 0.1035 | 0.0662 | 0.0697 | 0.0422 |
| LightGCN [3] | 0.3481 | 0.2854 | 0.3586 | 0.3265 | 0.2364 | 0.1640 | 0.0928 | 0.0592 | 0.0620 | 0.0371 |
| LightGCN++ | 0.3745** | 0.3083** | 0.3852** | 0.3534** | <u>0.2448**</u> | <u>0.1693**</u> | 0.1074** | 0.0690** | 0.0665** | 0.0398** |
| Improvement | 7.58% | 8.02% | 7.42% | 8.24% | 3.55% | 3.23% | 15.73% | 16.55% | 7.26% | 7.28% |
| NCL [5] | 0.3502 | 0.2878 | 0.3597 | 0.3281 | 0.2389 | 0.1647 | 0.0969 | 0.0621 | 0.0658 | 0.0394 |
| NCL++ | <u>0.3754**</u> | <u>0.3093**</u> | 0.3861** | 0.3545** | 0.2490** | 0.1712** | 0.1120** | 0.0720** | 0.0706** | 0.0422** |
| Improvement | 7.20% | 7.47% | 7.34% | 8.05% | 4.23% | 3.95% | 15.58% | 15.94% | 7.29% | 7.11% |
| SimGCL [10] | 0.3547 | 0.2914 | 0.3810 | 0.3480 | 0.2388 | 0.1641 | 0.1080 | 0.0692 | 0.0680 | 0.0416 |
| SimGCL++ | 0.3734** | 0.3068** | <u>0.3858**</u> | <u>0.3541**</u> | 0.2405* | 0.1652* | 0.1081 | 0.0697* | <u>0.0716**</u> | <u>0.0439**</u> |
| Improvement | 5.27% | 5.28% | 1.26% | 1.75% | 0.71% | 0.67% | 0.09% | 0.72% | 5.29% | 5.53% |
| XSimGCL [9] | 0.3575 | 0.2937 | 0.3830 | 0.3504 | 0.2339 | 0.1610 | 0.1074 | 0.0689 | 0.0657 | 0.0397 |
| XSimGCL++ | 0.3758** | 0.3094** | 0.3856* | 0.3534** | 0.2419** | 0.1657** | <u>0.1112**</u> | 0.0716** | 0.0742** | 0.0452** |
| Improvement | 5.12% | 5.35% | 0.68% | 0.86% | 3.42% | 2.92% | 3.54% | 3.92% | 12.94% | 13.85% |