Bootstrap Latent Representations for Multi-modal Recommendation

Xin Zhou
xin.zhou@ntu.edu.sg
Nanyang Technological University
Singapore, Singapore

Hongyu Zhou
hongyu.zhou@ntu.edu.sg
Nanyang Technological University
Singapore, Singapore

Yong Liu
stephenliu@ntu.edu.sg
Nanyang Technological University
Singapore, Singapore

Zhiwei Zeng
zhiwei.zeng@ntu.edu.sg
Nanyang Technological University
Singapore, Singapore

Chunyan Miao
ascymiao@ntu.edu.sg
Nanyang Technological University
Singapore, Singapore

Pengwei Wang
hoverwang.wpw@alibaba-inc.com
Alibaba Group
Beijing, China

Yuan You
youyuan.yy@alibaba-inc.com
Alibaba Group
Hangzhou, China

Feijun Jiang
feijun.jiangfj@antgroup.com
Alibaba Group
Hangzhou, China

ABSTRACT

This paper studies the multi-modal recommendation problem, where the item multi-modality information (e.g., images and textual descriptions) is exploited to improve the recommendation accuracy. Besides the user-item interaction graph, existing state-of-the-art methods usually use auxiliary graphs (e.g., user-user or item-item relation graph) to augment the learned representations of users and/or items. These representations are often propagated and aggregated on auxiliary graphs using graph convolutional networks, which can be prohibitively expensive in computation and memory, especially for large graphs. Moreover, existing multi-modal recommendation methods usually leverage randomly sampled negative examples in Bayesian Personalized Ranking (BPR) loss to guide the learning of user/item representations, which increases the computational cost on large graphs and may also bring noisy supervision signals into the training process. To tackle the above issues, we propose a novel self-supervised multi-modal recommendation model, dubbed BM3, which requires neither augmentations from auxiliary graphs nor negative samples. Specifically, BM3 first bootstraps latent contrastive views from the representations of users and items with a simple dropout augmentation. It then jointly optimizes three multi-modal objectives to learn the representations of users and items by reconstructing the user-item interaction graph and aligning modality features under both inter- and intra-modality perspectives. BM3 alleviates both the need for contrasting with negative examples and the complex graph augmentation from an additional target network for contrastive view generation. We show BM3 outperforms prior recommendation models on three datasets with number of nodes ranging from 20K to 200K, while achieving a 2-9× reduction in training time. Code implementation is located at: https://github.com/enoche/BM3.

CCS CONCEPTS

• Information systems → Recommender systems.

KEYWORDS

Multi-modal Recommendation, Bootstrap, Self-supervised learning

1 INTRODUCTION

In fast-growing e-commerce businesses, recommender systems play a critical role in helping users discover products or services they may like among millions of offerings. In practice, deep learning techniques have been widely applied in recommendation systems, mainly for exploiting historical user-item interactions, to model users’ preferences on items and produce item recommendations to users [41]. However, the rich multi-modal content information (e.g., texts, images, and videos) of items has still not been fully explored.

To improve the recommendation accuracy, recent work on multi-modal recommendation have studied effective means to integrate item multi-modal information into the traditional user-item recommendation paradigm. For example, some methods concatenate multi-modal features with the latent representations of items [9] or leverage attention mechanisms [4, 17] to capture users’ preferences on items’ multi-modal features. With a surge of research on graph-based recommendations [32, 37, 47], another line of research uses Graph Neural Networks (GNNs) to exploit item multi-modal...
information and enhance the learning of user and item representations [31, 33, 34]. For instance, [34] uses graph convolutional networks to separately propagate and aggregate different item multimodal information on the user-item interaction graph. To further improve recommendation performance, other auxiliary graph structures, e.g., the user-user relation graph [31] and item-item relation graph [39], have also been exploited to enhance the learning of user and item representations from the multi-modal information.

Although existing GNN-based multi-modal methods [31, 33, 39, 44, 45] can achieve state-of-the-art recommendation accuracy, the following issues may hinder their applications in scenarios involving large-scale graphs. First, they often learn the user and item representations based on pair-wise ranking losses, e.g., the Bayesian Personalized Ranking (BPR) loss [24], which treats observed user-item interaction pairs as positive samples and randomly sampled user-item pairs as negative samples. Such a negative sampling strategy may incur a prohibitive cost on large graphs [30] and bring noisy supervision signals into the training process. For example, previous research [48] has verified that the default uniform sampling [42] in LightGCN [10] incurs more than 25% training time per epoch. Second, methods utilizing auxiliary graph structures may incur prohibitive memory cost when building and/or training on large-scale auxiliary graphs. More analyses on the computational complexity of existing graph-based multi-modal methods can be found in Table 1 and Table 4.

Self-Supervised Learning (SSL) [5, 7] provides a possible solution for learning the representations of users and items without negative samples. Research in various domains ranging from Computer Vision (CV) to Natural Language Processing (NLP), has shown that SSL is possible to achieve competitive or even better results than supervised learning [3, 7, 38]. The main idea of SSL is to maximize the similarity of representations obtained from different distorted versions of a sample using two asymmetry networks, i.e., the online network and the target network. However, training with only positive samples will lead the model into a trivial constant solution [5]. To tackle this collapsing problem, BYOL [7] and SimSiam [5] introduce an additional "predictor" network to the online network and a special "stop gradient" operation on the target network. Recently, BUIR [15] transfers BYOL into the recommendation domain and shows competitive performance on the evaluation datasets.

In this paper, we propose a Bootstrapped Multi-Modal Model, dubbed BM3, for multi-modal recommendation. It first simplifies the current SSL framework by removing its target network, which can reduce half of the model parameters. Moreover, to retain the similarity between different augmentations, BM3 incorporates a simple dropout mechanism to perturb the latent embeddings generated from the online network. This is different from current SSL paradigm that perturbs the inputs via graph augmentation [7, 15] or image augmentation [26]. The design eases both the memory and computational cost of conventional graph augmentation techniques [15, 35], as it does not introduce any auxiliary graphs. Last but not least, we design a loss function that is specialized for multi-modal recommendation. It minimizes the reconstruction loss of the user-item interaction graph as well as aligns the learned features under both inter- and intra-modality perspectives.

We summarize our main contributions as follows. First, we propose BM3, a novel self-supervised learning method for multi-modal recommendation. In BM3, we use a simple latent representation dropout mechanism instead of graph augmentation to generate the target view of a user or an item for contrastive learning without negative samples. Second, to train BM3 without negative samples, we design a Multi-Modal Contrastive Loss (MMCL) function that jointly optimizes three objectives. In addition to minimizing the classic user-item interaction graph reconstruction loss, MMCL further aligns the learned features between different modalities and reduces the dissimilarity between representations of different augmented views from a specific modality. Finally, we validate the effectiveness and efficiency of BM3 on three datasets with the number of nodes ranging from 20K to 200K. The experimental results show that BM3 achieves significant improvements over the state-of-the-art multi-modal recommendation methods, while training 2-9× faster than the baseline methods.

2 RELATED WORK

2.1 Multi-modal Recommendation

2.1.1 Deep Learning-based Models. Due to the success of the Collaborative Filtering (CF) method, most early multi-modal recommendation models utilize deep learning techniques to explore users’ preferences on top of the CF paradigm. For example, VBPR [9], which builds on top of the BPR method, leverages the visual features of items. It utilizes a pre-trained convolutional neural network to obtain the visual features of items and linearly transforms them into a latent visual space. To make predictions, VBPR represents an item by concatenating the latent visual features with its ID embedding. Moreover, DeepStyle [19] augments the representations of items with both visual and style features within the BPR framework. To capture the users’ preference on multi-modal information, the attention mechanism has also been adopted in recommendation models. For instance, VECCF [4] utilizes the VGG model [27] to perform pre-segmentation on images and captures the user’s attention on different image regions. MAML [17] uses a two-layer neural network to capture the user’s preference on textual and visual features of an item.

2.1.2 Graph-based Multi-modal Models. More recently, another line of research introduces GNNs into recommendation systems [37], which can greatly enhance the user and item representations by incorporating the structural information in the user-item interaction graph and auxiliary graphs.

To exploit the item multi-modal information, MMGCN [34] adopts the message passing mechanism of Graph Convolutional Networks (GCNs) and constructs a modality-specific user-item bipartite graph, which can capture the information from multi-hop neighbors to enhance the user and item representations. Based on MMGCN, DualGNN [31] introduces a user co-occurrence graph with a model preference learning module to capture the user’s preference for features from different modalities of an item. As the user-item graph may encompass unintentional interactions, GRCN [33] introduces a graph refine layer to refine the structure of the user-item interaction graph by identifying the noise edges and corrupting the false-positive edges. To explicitly mine the semantic information between items, LATTICE [39] constructs item-item
relation graphs for each modality and fuses them together to obtain a latent item graph. It dynamically updates the graph after items’ information is propagated and aggregated from their highly connected affinities using GCNs. FREEDOM [45] further detects that the learning of item-item graphs are negligible and freezes the graph for effective and efficient recommendation. [43] provides a comprehensive survey of multi-modal recommender systems with taxonomy, evaluation and future directions.

Although graph-based multi-modal models achieve new state-of-the-art recommendation accuracy, they often require auxiliary graphs for user and item augmentations and also a large number of negatives for representation learning with BPR loss. Both requirements can lead to high computational complexity and prohibitive memory cost as the graph size increases, limiting the efficiency of these models in scenarios involving large-scale graphs.

2.2 Self-supervised Learning (SSL)
SSL-based methods have achieved competitive results in various CV and NLP tasks [12, 20]. As our model is based on SSL that only uses observed data, our review of SSL methods focuses on those that do not require negative sampling.

Current SSL frameworks are derived from Siamese networks [1], which are generic models for comparing entities [5]. BYOL [7] and SimSiam [5] use asymmetric Siamese network to achieve remarkable results. Specifically, BYOL proposes two coupled encoders (i.e., the online encoder and the target encoder) that are optimized and updated iteratively. The online encoder is optimized towards the target encoder, while the target encoder is a momentum encoder. Its parameters are updated as an exponentially moving average of the online encoder. BYOL uses both a predictor on the online encoder and a “stop gradient” operator on the target encoder to avoid network collapse. SimSiam verifies that a “stop gradient” operator is crucial for preventing collapse. However, it shares the parameters between the online and target encoders. On the contrary, Barlow Twins [38] uses a symmetric architecture by designing an innovative objective function that can align the cross-correlation matrix computed from two contrastive representations as close to the identity matrix as possible.

Derived from BYOL, the recently proposed self-supervised framework, BUIR [15], learns the representations of users and items solely from positive interactions. It introduces different views and leverages a slow-moving average network to update the parameters of the target encoder with the online encoder. Same inputs are fed into the online encoder and the target encoder (that are optimized and updated) to generate the contrastive views. BYOL uses both a predictor on the online encoder and the target encoder) that are optimized and updated as an exponentially moving average of the online encoder. ByOL uses both a predictor on the online encoder and the target encoder. Its parameters are updated as an exponentially moving average of the online encoder. BYOL uses both a predictor on the online encoder and the target encoder) that are optimized and updated as an exponentially moving average of the online encoder. ByOL uses both a predictor on the online encoder and the target encoder. Its parameters are updated as an exponentially moving average of the online encoder.

2.3 Multi-modal Representation Learning

3.1 Multi-modal Latent Space Converter
Let $e_u, e_i \in \mathbb{R}^d$ denote the input ID embeddings of the user $u \in \mathcal{U}$ and the item $i \in \mathcal{I}$, where $d$ is the embedding dimension, and $\mathcal{U}, \mathcal{I}$ are the sets of users and items, respectively. Their cardinal numbers are set as $|\mathcal{U}|$ and $|\mathcal{I}|$, respectively. We denote the modality-specific features obtained from the pre-trained model as $e_m \in \mathbb{R}^{dm}$, where $m \in M$ denotes a specific modality from the full set of modalities $M$, and $d_m$ denotes the dimension of the features. The cardinal number of $M$ is denoted by $|M|$. In this paper, we consider two modalities: vision $v$ and text $t$. However, the model can be easily extended to scenarios with more than two modalities. As the multi-modal feature spaces are different from each other, we first convert the multi-modal features and ID embeddings into the same latent space.

3.1.1 Multi-modal Features. The features of an item obtained from different modalities are of different dimensions and in different feature spaces. For a multi-modal feature vector $e_m$, we first project it into a latent low dimension using a projection function $f_m$ based on Multi-Layer Perceptron (MLP). Then, we have:

$$h_m = f_m W_m + b_m, \tag{1}$$

where $W_m \in \mathbb{R}^{dm \times d}$, $b_m \in \mathbb{R}^d$ denote the linear transformation matrix and bias in the MLP of $f_m$. In this way, each uni-modal latent representation $h_m$ shares the same space with ID embeddings.

3.1.2 ID Embeddings. Previous work [39] has verified the crucial role of ID embeddings in multi-modal recommendation. Although the ID embeddings of users and items can be directly initialized within the latent space, they do not encode any structural information about the user-item interaction graph. Inspired by the recent success of applying GCN for recommendation, we use a backbone network of LightGCN [10] with residual connection to encode the structure of the user-item interaction graph.

Suppose $G = (\mathcal{V}, \mathcal{E})$ be a given graph with node set $\mathcal{V} = \mathcal{U} \cup \mathcal{I}$ and edge set $\mathcal{E}$. The number of nodes is denoted by $|\mathcal{V}|$, and the number of edges is denoted by $|\mathcal{E}|$. The adjacency matrix is denoted by $A \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$, and the diagonal degree matrix is denoted by $D$. In $G$, the edges describe observed user-item interactions. If a user has interactions with an item, we build an edge between the user node and the item node. Moreover, we use $H^l \in \mathbb{R}^{|\mathcal{V}| \times d}$ to denote the ID embeddings at the $l$-th layer by stacking all the embeddings of users and items at layer $l$. Specifically, the initial ID embeddings $H^0$ is a collection of embeddings $e_u$ and $e_i$ from all users and items. A typical feed forward propagation GCN [14] to calculate the hidden ID embedding $H^{l+1}$ at layer $l + 1$ is recursively conducted as:

$$H^{l+1} = \sigma \left( \hat{A} H^l W^l \right), \tag{2}$$

where $\sigma(\cdot)$ is a non-linear function, e.g., the ReLu function, $\hat{A} = \bar{D}^{-1/2} (A + I) \bar{D}^{-1/2}$ is the re-normalization of the adjacency matrix $A$, and $\bar{D}$ is the diagonal degree matrix of $A + I$. For node classification, the last layer of a GCN is used to predict the label of a node via a softmax classifier.
On top of the vanilla GCN, LightGCN simplifies its structure by removing the feature transformation $W^I$ and non-linear activation $\sigma(\cdot)$ layers for recommendation. As they found these two layers impose adverse effects on recommendation performance. The simplified graph convolutional layer in LightGCN is defined as:

$$H^{l+1} = (D^{-1/2}AD^{-1/2})H^l,$$

where the node embeddings of the $(l + 1)$-th hidden layer are only linearly aggregated from the $l$-th layer with a transition matrix $D^{-1/2}AD^{-1/2}$. The transition matrix is exactly the weighted adjacency matrix mentioned above.

We use a readout function to aggregate all representations in hidden layers for user and item final representations. However, GCNs may suffer from the over-smoothing problem [2, 16, 18]. Following LATTICE [39], we also add a residual connection [2, 14] to the item initial embeddings $H^0_i$ to obtain the final representations of items. That is:

$$H_{u_i} = \text{READOUT}(H_{u_i}^0, H_{u_i}^1, H_{u_i}^2, ..., H_{u_i}^L);$$
$$H_{i} = \text{READOUT}(H_{i}^0, H_{i}^1, H_{i}^2, ..., H_{i}^L) + H_{i}^0,$$

where the READOUT function can be any differentiable function. We use the default mean function of LightGCN for its final ID embedding updating.

With the multi-modal latent space converter, we can obtain three types of latent embeddings: user ID embeddings, item ID embeddings, and unimodal item embeddings. In the following section, we illustrate the design of losses in B3M for efficient parameter optimization without negative samples.

### 3.2 Multi-modal Contrastive Loss

Previous studies on SSL use the stop-gradient strategy to prevent the model from resulting in a trivial constant solution [5, 7]. Besides, they use online and target networks to make the model parameters learn in a teacher-student manner [29]. B3M simplifies the current SSL paradigm [5, 7] by postponing the data augmentation after the encoding of the online network. We first illustrate the data augmentation in B3M.

#### 3.2.1 Contrastive View Generator

Prior studies [15, 30, 36] use graph augmentations to generate two alternate views of the original graph for self-supervised learning. Input features are encoded through both graphs to generate the contrastive views. To reduce the computational complexity and the memory cost, B3M removes the requirement of graph augmentations with a simple latent embedding dropout technique that is analogous to node dropout [28]. The contrastive latent embedding $\tilde{h}$ of $h$ under a dropout ratio $p$ is calculated as:

$$\tilde{h} = h \cdot \text{Bernoulli}(p).$$

Following [5, 7], we also place stop-gradient on the contrastive view $\tilde{h}$. Whilst we feed the original embedding $h$ into a predictor of MLP.

$$\hat{h} = hW_p + b_p,$$

where $W_p \in \mathbb{R}^{d \times d}, b_p \in \mathbb{R}^d$ denote the linear transformation matrix and bias in the predictor function $f_p$.

#### 3.2.2 Graph Reconstruction Loss

B3M takes a positive user-item pair $(u, i)$ as input. With the generated contrastive view $(\tilde{h}_u, \tilde{h}_i)$ of the online representations $(h_u, h_i)$, we define a symmetrized loss function as the negative cosine similarity between $(\tilde{h}_u, \tilde{h}_i)$ and $(h_u, h_i)$:

$$L_{rec} = C(\tilde{h}_u, \tilde{h}_i) + C(h_u, h_i).$$
3.3 Top-K Recommendation

To generate item recommendations for a user, we first predict the interaction scores between the user and candidate items. Then, we rank candidate items based on the predicted interaction scores in descending order, and choose top-ranked items as recommendations to the user. Classical CF methods recommend top-K items by ranking scores of the inner product of a user embedding with all candidate item embeddings. As our MMCL can learn a good predictor on user and item latent embeddings, we use the embeddings transformed by the predictor $f_p$ for inner product. That is:

$$s(h_u, h_i) = h_u^T \cdot \tilde{h}_i^T. \quad (13)$$

A high score suggests that the user prefers the item.

### 3.4 Computational Complexity

The computational cost of $\text{BM3}$ mainly occurs in linear propagation of the normalized adjacency matrix $\hat{A}$. The analytical complexity of $\text{LightGCN}$ and $\text{BM3}$ are in the same magnitude with $O(2|E|d/B)$ on graph convolutional operation, where $L$ is the number of LightGCN layers, and $B$ is the training batch size. However, $\text{BM3}$ has additional costs on multi-modal feature projection and prediction. The projection cost is $O(|\sum_{m \in M} |I|d_m|)$ on all modalities. The contrastive loss cost is $O((2 + 2|M|)d_B)$. The total computational cost for $\text{BM3}$ is $O(2|E|d/B + |\sum_{m \in M} |I|d_m|) + (2 + 2|M|)d_B)$. We summarize the computational complexity of the graph-based multi-modal methods in Table 1. Both $\text{MMGCN}$ and $\text{DualGNN}$ use a two-layer MLP for multi-modal feature projection. On the contrary, $\text{BM3}$ uses intra-modality feature masked loss to further encourage the learning of predictor with sparse representations of latent embeddings. Sparse is verified scale efficient in large transformers [11, 25]. We randomly mask out a subset of the latent embedding $h_m$ by dropout with the stop gradient operator as in [5, 7], and implement the operator $\frac{\partial}{\partial f} (\cdot)$ with $(\cdot)^\top$.

$$L_{\text{mask}} = C(\tilde{h}_{m, h_m}^T). \quad (11)$$

Additionally, we add regularization penalty on the online embeddings (i.e., $h_u$ and $h_i$). Our final loss function is:

$$L = L_{\text{rec}} + L_{\text{align}} + L_{\text{mask}} + \lambda \cdot (||h_u||_2^2 + ||h_i||_2^2). \quad (12)$$

3.3 Top-K Recommendation

To generate item recommendations for a user, we first predict the interaction scores between the user and candidate items. Then, we rank candidate items based on the predicted interaction scores in descending order, and choose $K$ top-ranked items as recommendations to the user. Classical CF methods recommend top-K items by ranking scores of the inner product of a user embedding with all candidate item embeddings. As our MMCL can learn a good predictor on user and item latent embeddings, we use the embeddings transformed by the predictor $f_p$ for inner product. That is:

$$s(h_u, h_i) = h_u^T \cdot \tilde{h}_i^T. \quad (13)$$

A high score suggests that the user prefers the item.
4.1 Experimental Datasets
Following previous studies [9, 39], we use the Amazon review dataset [8] for experimental evaluation. This dataset provides both product descriptions and images simultaneously, and it is publicly available and varies in size under different product categories. To ensure as many baselines can be evaluated on large-scale datasets, we choose three per-category datasets, i.e., Baby, Sports and Outdoors (denoted by Sports), and Electronics, for performance evaluation. In these datasets, each review rating is treated as a record of positive user-item interaction. This setting has been widely used in previous studies [9, 10, 39, 40]. The raw data of each dataset are pre-processed with a 5-core setting on both items and users, and their 5-core filtered results are presented in Table 2, where the data sparsity is measured as the number of interactions divided by the product of the number of users and the number of items. The pre-processed datasets include both visual and textual modalities. Following [39], we use the 4,096-dimensional visual features that have been extracted and published in [21]. For the textual modality, we extract textual embeddings by concatenating the title, descriptions, categories, and brand of each item and utilize sentence-transformers [23] to obtain 384-dimensional sentence embeddings.

4.2 Baseline Methods
To demonstrate the effectiveness of BM3, we compare it with the following state-of-the-art recommendation methods, including general CF recommendation models and multi-modal recommendation models.

- **BPR** [24]: This is a matrix factorization model optimized by a pair-wise ranking loss in a Bayesian way.
- **LightGCN** [10]: This is a simplified graph convolution network that only performs linear propagation and aggregation between neighbors. The hidden layer embeddings are averaged to calculate the final user and item embeddings for prediction.
- **BIUIR** [15]: This self-supervised framework uses asymmetric network architecture to update its backbone network parameters. In BUIR, LightGCN is used as the backbone network. It is worth noting that BUIR does not rely on negative samples for learning.
- **VBPR** [9]: This model incorporates visual features for user preference learning with BPR loss. Following [31, 39], we concatenate the multi-modal features of an item as its visual feature for user preference learning.
- **MMGCN** [34]: This method constructs a modal-specific graph to learn user preference on each modality leveraging GCN. The final user and item representations are generated by combining the learned representations from each modality.
- **GRCN** [33]: This method improves previous GCN-based models by refining the user-item bipartite graph with removal of false-positive edges. User and item representations are learned on the refined bipartite graph by performing information propagation and aggregation.
- **DualGNN** [31]: This method builds an additional user-user correlation graph from the user-item bipartite graph and uses it to fuse the user representation from its neighbors in the correlation graph.
- **LATTICE** [39]: This method mines the latent structure between items by learning an item-item graph from their multi-modal features. Graph convolutional operations are performed on both item-item graph and user-item interaction graph to learn user and item representations.

We group the first three baselines (i.e., BPR, LightGCN, and BUIR) as general models, because they only use implicit feedback (i.e., user-item interactions) for recommendation. The other multi-modal models utilize both implicit feedback and multi-modal features for recommendation. Analogously, we categorize BUIR into the self-supervised model and the others as supervised models as they are using negative samples for representation learning. The proposed BM3 model is within the self-supervised multi-modal domain.

4.3 Setup and Evaluation Metrics
For a fair comparison, we follow the same evaluation setting of [31, 39] with a random data splitting 8:1:1 on the interaction history of each user for training, validation and testing. Moreover, we use Recall@K and NDCG@K to evaluate the top-K recommendation performance of different recommendation methods. Specifically, we use the all-ranking protocol instead of the negative-sampling protocol to compute the evaluation metrics for recommendation accuracy comparison. In the recommendation phase, all items that have not been interacted by the given user are regarded as candidate items. In the experiments, we empirically report the results of K at 10, 20 and abbreviate the metrics of Recall@K and NDCG@K as R@K and N@K, respectively.

4.4 Implementation Details
Same as other existing work [10, 39], we fix the embedding size of both users and items to 64 for all models, initialize the embedding parameters with the Xavier method [6], and use Adam [13] as the optimizer with a learning rate of 0.001. For a fair comparison, we carefully tune the parameters of each model following their published papers. The proposed BM3 model is implemented by PyTorch [22]. We perform a grid search across all datasets to conform to its optimal settings. Specifically, the number of GCN layers is tuned in {1, 2}. The dropout rate for embedding perturbation is chosen from [0.3, 0.5], and the regularization coefficient is searched in [0.1, 0.01]. For convergence consideration, the early stopping and total epochs are fixed at 20 and 1000, respectively. Following [39], we use R@20 on the validation data as the training stopping indicator. We have integrated our model and all baselines into the unified multi-modal recommendation platform, MMRec [46].

4.5 Effectiveness of BM3 (RQ1)
The performance achieved by different recommendation methods on all three datasets are summarized in Table 3. From the table, we have the following observations. First, the proposed BM3 model significantly outperforms both general recommendation methods and state-of-the-art multi-modal recommendation methods on each dataset. Specifically, BM3 improves the best baselines by 3.68%, 6.15%,

---

1Datasets are available at http://jmcauley.ucsd.edu/data/amazon/links.html
Table 3: Overall performance achieved by different recommendation methods in terms of Recall and NDCG. We mark the global best results on each dataset under each metric in boldface and the second best is underlined.

| Datasets | Metrics | General models | Multi-modal models |
|----------|---------|----------------|-------------------|
|          |         | BPR LightGCN BUIR | VBPR MMGCN GRCN DualGNN LATTICE BM3 |
| Baby     | R@10   | 0.0357 0.0479 0.0506 | 0.0423 0.0378 0.0532 0.0448 0.0544 0.0564 |
|          | R@20   | 0.0575 0.0754 0.0788 | 0.0663 0.0615 0.0824 0.0716 0.0848 0.0883 |
|          | N@10   | 0.0192 0.0257 0.0269 | 0.0223 0.0200 0.0282 0.0240 0.0291 0.0301 |
|          | N@20   | 0.0249 0.0328 0.0342 | 0.0284 0.0261 0.0358 0.0309 0.0369 0.0383 |
| Sports   | R@10   | 0.0432 0.0569 0.0467 | 0.0558 0.0530 0.0559 0.0568 0.0618 0.0656 |
|          | R@20   | 0.0653 0.0864 0.0733 | 0.0856 0.0605 0.0877 0.0859 0.0947 0.0980 |
|          | N@10   | 0.0241 0.0311 0.0260 | 0.0307 0.0193 0.0306 0.0310 0.0337 0.0355 |
|          | N@20   | 0.0298 0.0387 0.0329 | 0.0384 0.0254 0.0389 0.0385 0.0422 0.0438 |
| Electronics | R@10 | 0.0235 0.0363 0.0332 | 0.0293 0.0307 0.0349 0.0363 - 0.0439 |
|          | R@20   | 0.0367 0.0540 0.0514 | 0.0458 0.0331 0.0529 0.0541 - 0.0648 |
|          | N@10   | 0.0127 0.0204 0.0185 | 0.0159 0.0109 0.0195 0.0202 - 0.0247 |
|          | N@20   | 0.0161 0.0250 0.0232 | 0.0202 0.0141 0.0241 0.0248 - 0.0302 |

"-" indicates the model cannot be fitted into a Tesla V100 GPU card with 32 GB memory.

Table 4: Efficiency comparison of BM3 against the baselines.

| Datasets | Metrics | General models | Multi-modal models |
|----------|---------|----------------|-------------------|
|          |         | BPR LightGCN BUIR | VBPR MMGCN GRCN DualGNN LATTICE BM3 |
| Baby     | Memory (GB) | 1.59 1.69 2.29 | 1.89 2.69 2.95 2.05 4.53 2.11 |
|          | Time (s/epoch) | 0.47 0.99 0.77 | 0.57 3.48 2.36 7.81 1.61 0.85 |
| Sports   | Memory (GB) | 2.00 2.24 3.75 | 2.71 3.91 4.49 2.81 19.93 3.03 |
|          | Time (s/epoch) | 0.95 2.86 2.19 | 1.28 16.60 6.74 10.71 2.69 2.95 |
| Electronics | Memory (GB) | 3.69 4.92 10.13 | 6.20 14.34 17.38 8.85 - 8.28 |
|          | Time (s/epoch) | 6.75 67.49 63.77 | 14.20 470.15 152.68 341.02 - 73.31 |

"-" denotes the model cannot be fitted into a Tesla V100 GPU card with 32 GB memory.

Apart from the comparison of recommendation accuracy, we also report the efficiency of BM3 against the baselines, in terms of utilized memory and training time per epoch. It is worth noting that all models are firstly evaluated on a GeForce RTX 2080 Ti with 12GB memory, and the model will be advanced to a Tesla V100 GPU with 32 GB memory if it cannot be fitted into the 12 GB memory. The efficiencies of different methods are summarized in Table 4. From the table, we can have the following two observations. First, from both the general model and multi-modal model perspectives, graph-based models usually consume more memory than classic CF models (i.e., BPR and VBPR). Specifically, classic CF models require a minimum GPU memory cost for representation learning of users and items. Whilst graph-based models usually need to
To fully understand the behaviors of BM3, we perform ablation studies on both the multi-modal features and different parts of the multi-modal contrastive loss.

### 4.7 Ablation Study (RQ3 & RQ4)

To fully understand the behaviors of BM3, we perform ablation studies on both the multi-modal features and different parts of the multi-modal contrastive loss.

#### 4.7.1 Multi-modal Features (RQ3)

We evaluate the recommendation accuracy of BM3 by feeding individual modal features into the model. Specifically, we design the following variants of BM3.

- **BM3\(_{w/o \text{ v&t}}\):** This variant degrades to a general recommendation model that exploits only the user-item interactions for recommendation.
- **BM3\(_{w/o \text{ v}}\):** This variant of BM3 learns the representations of users and items without the visual features of items.
- **BM3\(_{w/o \text{ t}}\):** This variant of BM3 is trained without the input from the textual features.

Table 5 summarizes the recommendation performance of BM3 and its variants, i.e., BM3\(_{w/o \text{ v&t}}\), BM3\(_{w/o \text{ v}}\), and BM3\(_{w/o \text{ t}}\) on all three experimental datasets.

As shown in Table 5, the importance of textual and visual features varies with datasets. BM3\(_{w/o \text{ v}}\) leveraging only on the textual features gains slight better recommendation accuracy than BM3\(_{w/o \text{ t}}\) on Baby dataset. However, on the other datasets, the differences between these two variations are negligible. Moreover, we can observe that the context features from either textual or visual modality can boost the performance of BM3\(_{w/o \text{ v&t}}\) on Baby and Sports datasets. However, this statement does not hold under the large dataset, i.e., Electronics. By combining both the textual and visual features, BM3 achieves the best recommendation accuracy on all three datasets.

#### 4.7.2 Multi-modal Contrastive Loss (RQ4)

As the loss function plays a critical role in learning the model parameters of BM3 without negative samples. We further study the behaviors of BM3 by removing different parts of the multi-modal contrastive loss. Specifically, we consider the following variants of BM3 for experimental evaluation.

- **BM3\(_{w/o \text{ mm}}\):** This variant only uses the interaction graph reconstruction loss to train the model parameters, i.e., trained without the multi-modal losses. It is worth noting that this variant is equivalent to BM3\(_{w/o \text{ v&t}}\).
- **BM3\(_{w/o \text{ inter}}\):** In this variant, the representations of users and items are learned without considering the inter-modality alignment loss.
- **BM3\(_{w/o \text{ intra}}\):** This variant learns the representations of users and items without considering the feature masked loss.

Table 6 shows the performance achieved by BM3, BM3\(_{w/o \text{ mm}}\), BM3\(_{w/o \text{ inter}}\), and BM3\(_{w/o \text{ intra}}\) on all three datasets.

From Table 6, we find a similar pattern as that shown in the ablation study on multi-modal features. That is, the multi-modal losses can improve the recommendation accuracy of BM3 on Baby and Sports datasets. However, BM3 leveraging either the inter-modality alignment loss or the intra-modality feature masked loss degrades its performance on the Electronics dataset. Moreover, the importance of inter- and intra-modality losses also varies with the datasets.

From the ablation studies on features and the loss function, we find that the multi-modal features and the multi-modal contrastive loss play a critical role in learning the model parameters of BM3 without negative samples. We further study the behaviors of BM3 by removing different parts of the multi-modal contrastive loss. Specifically, we consider the following variants of BM3 for experimental evaluation.
to learn good representations of users and items. Adding coarse multi-modal signals to BM3w/o mn does not help improve the recommendation accuracy.

5 CONCLUSION
This paper proposes a novel self-supervised learning framework, named BM3, for multi-modal recommendation. BM3 removes the requirement of randomly sampled negative examples in modeling the interactions between users and items. To generate a contrastive view in self-supervised learning, BM3 utilizes a simple yet efficient latent embedding dropout mechanism to perturb the original embeddings of users and items. Moreover, a novel learning paradigm based on the multi-modal contrastive loss has also been devised. Specifically, the contrastive loss jointly minimizes: a) the reconstruction loss of the user-item interaction graph, b) the alignment loss between ID embeddings of items and their multi-modal features, and c) the masked loss within a modality-specific feature. We evaluate the proposed BM3 model on three real-world datasets, including one large-scale dataset, to demonstrate its effectiveness and efficiency in recommendation tasks. The experimental results show that BM3 achieves significant accuracy improvements over the state-of-the-art multi-modal recommendation methods, while training 2-9× faster than the baseline methods.

ACKNOWLEDGMENTS
This work was supported by Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba-NTU Singapore Joint Research Institute (JRI), Nanyang Technological University, Singapore.

6 APPENDICES
6.1 Hyper-parameter Sensitivity Study
To guide the selection of hyper-parameters of BM3, we perform a hyper-parameter sensitivity study with regard to the recommendation accuracy, in terms of Recall@20. We use at least two datasets to evaluate the performance of BM3 under different hyper-parameter settings. Specifically, we consider the following three hyper-parameters, i.e., the number of GCN layers $L$, the ratio of embedding dropout, and the regularization coefficient factor $\lambda$.

![Figure 2: Top-20 recommendation accuracy of BM3 varies with the number of GCN layers in the backbone network.](image1)

6.1.1 The Number of GCN Layers. The number of GCN layers $L$ in BM3 is varied in $\{1, 2, 3, 4\}$. Fig. 2 shows the performance trends of BM3 with respect to different settings of $L$. As shown in Fig. 2, BM3 shows relatively slow performance degradation as the number of layers increases, on small-scale datasets (i.e., Baby and Sports). However, the recommendation accuracy can be improved with more than one GCN layer in the backbone network on Electronics dataset.

6.1.2 The Dropout Ratio and Regularization Coefficient. We vary the dropout ratio of BM3 from 0.1 to 0.5 with a step of 0.1, and vary the regularization coefficient $\lambda$ in $\{0.0001, 0.001, 0.01, 0.1\}$. Fig. 3 shows the performance achieved by BM3 under different combinations of the embedding dropout ratio and regularization coefficient. We note that a larger dropout ratio of BM3 on a relative small-scale dataset (i.e., Sports) usually helps BM3 achieve better recommendation accuracy. Moreover, the performance of BM3 is less sensitive to the settings of regularization coefficient on the large dataset (i.e., Electronics). In Fig. 3(b), it is worth noting that BM3 achieves competitive recommendation accuracy when the dropout ratio is larger than 0.2. This verifies the stability of BM3 in the recommendation task, i.e., the recommendation performance of BM3 is not just a consequence of random seeds.

![Figure 3: The Performance achieved by BM3 with respect to different combinations of the latent embedding dropout ratio and regularization coefficient on three datasets. Darker background indicates better recommendation accuracy.](image2)

REFERENCES
[1] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a ’siamese’ time delay neural network. Advances in neural information processing systems 6 (1993), 737–744.
