

Supplementary Materials: CIRP: Cross-Item Relational Pre-training for Multimodal Product Bundling

Anonymous Authors

1 DETAILED LOSS FORMULATIONS

Section 3.3.2 presents the pre-train objectives of the proposed CIRP. In this section, we explain how ego-item contrastive loss (ITC) and cross-item contrastive loss (CIC) are formulated in detail.

Following the setting of ALBEF [7], the similarity between the input image and text pair is represented by the inner product of the corresponding image representation $g_v(\mathbf{v}_{cls})$ and text representation $g_t(\mathbf{t}_{cls})$, and is formulated as :

$$s = g_v(\mathbf{v}_{cls})^\top g_t(\mathbf{t}_{cls}), \quad (1)$$

where g_v and g_t are linear transformations mapping the CLS embeddings to normalized low dimensional representations. Following the design of MoCo [1], we maintain two queues storing the most recent M image representations $g'_v(\mathbf{v}'_{cls})$ and text representations $g'_t(\mathbf{t}'_{cls})$ respectively from the momentum unimodal encoders.

For each input image and text, the softmax-normalized image-to-text and text-to-image similarity is calculated as:

$$p_m^{i2t}(I) = \frac{\exp(s(I, T_m)/\tau)}{\sum_{m=1}^M \exp(s(I, T_m)/\tau)}, \quad (2)$$

$$p_m^{t2i}(T) = \frac{\exp(s(T, I_m)/\tau)}{\sum_{m=1}^M \exp(s(T, I_m)/\tau)}, \quad (3)$$

where τ is a learnable temperature parameter and $s(I, T_m)$ and $s(T, I_m)$ are defined as:

$$s(I, T_m) = g_v(\mathbf{v}_{cls})^\top g'_t(\mathbf{t}'_{cls}), \quad (4)$$

$$s(T, I_m) = g_t(\mathbf{t}_{cls})^\top g'_v(\mathbf{v}'_{cls}). \quad (5)$$

Let $\mathbf{y}^{i2t}(I)$ and $\mathbf{y}^{t2i}(T)$ denote the ground-truth one-hot similarity, where negative image-text pairs have a probability of 0 and the probability of positive ones are 1. In ITC loss, the image-text pairs of the same items are regarded as positive pairs. The image-text contrastive loss is defined as the cross-entropy H between the softmax-normalized image-text similarities \mathbf{p} and ground-truth labels \mathbf{y}_{ego} of the image-text pairs of the same items:

$$\mathcal{L}^{itc} = H(\mathbf{y}_{ego}^{i2t}(I), \mathbf{p}^{i2t}(I)) + H(\mathbf{y}_{ego}^{t2i}(T), \mathbf{p}^{t2i}(T)). \quad (6)$$

And in CIC loss, the cross-item image-text pairs from correlated items $(i, j) \in \tilde{\mathcal{G}}$ are also regarded as positive pairs. We align the cross-item image-text similarities \mathbf{p} with the co-purchase relations \mathbf{y}_{rel} from correlated cross-item image-text pairs:

$$\mathcal{L}^{cic} = H(\mathbf{y}_{rel}^{i2t}(I), \mathbf{p}^{i2t}(I)) + H(\mathbf{y}_{rel}^{t2i}(T), \mathbf{p}^{t2i}(T)). \quad (7)$$

2 BASELINE IMPLEMENTATION DETAILS

In this section, we present the implementation details of each baseline model.

2.1 Relation-only Pre-training

This type of methods learn pre-trained item representations from the item-item relations.

- **MFBPR** [9] learns item representations using the user-item interaction bipartite graph, which is from the Amazon review dataset and used for pre-training. Factorized from the user-item interaction matrix, the item embeddings are optimized with BPR loss through the pairwise user-item interaction prediction task.
- **LightGCN** [2] is one of the SOTA graph learning methods. It captures the high-order neighbor information by linearly propagating the node embeddings. We train the LightGCN model on the item-item relation graph with the task of link prediction. BPR loss is adopted as the optimization target.
- **SGL** [11] follows the graph learning paradigm of LightGCN. In addition, it generates two views of item representations using different data augmentations and applies self-supervised contrastive learning between item representations from two different views. Similar to the implementation of LightGCN, we train SGL on the item-item relation graph with link prediction task using BPR loss.
- **Caser** [10] is a CNN-based method that models the sequential patterns by adopting convolutional operations on the embedding matrix of a few most recent items. It achieves competitive performance on tasks like sequential recommendation. We train Caser with the item purchase sequence from pre-train data and the model is optimized with BPR loss.
- **GRU4Rec** [3] applies GRU modules to model the sequential patterns within the input item purchase sequences. By inputting the sequences of items purchased by the same user from pre-train dataset, the BPR loss is utilized to train the model to predict the next item given prior item sequence.
- **SASRec** [4] is a popular sequential method commonly used in fields like sequential recommendation. It models the item purchase sequences through self-attention modules, where the user's interest is captured dynamically. The SASRec is trained using the sequences of items purchased by the same user from pre-train dataset with BCE loss.

For all the six methods above, we optimize the model using Adam optimizer and adopt grid search to find the best hyper-parameters. The learning rate is searched from 0.01 to 1.0×10^{-4} , weight decay is searched from 1.0×10^{-4} to 1.0×10^{-7} , and embedding size is tuned in range of [16, 32, 64, 128]. For the three graph-based methods, the number of graph propagation layers in LightGCN and SGL is tuned within [1, 2, 3]. The temperature used in contrastive loss is tuned from 0.05 to 0.4 and the weighting coefficient applied on contrastive loss is tuned from 0.01 to 0.5. For the sequence-based methods, the maximum sequence length is set to 20 and zero-padding is applied to complete the short sequences. All these methods are trained on a single NVIDIA A5000 (24G) GPU.

2.2 Semantic-only Pre-training

These methods generate the item representations using the image and textual descriptions of the items. We adopt two pre-train vision-language models, CLIP [8] and BLIP [6], as the multimodal feature extractor. The item representation is obtained by averaging the normalized visual feature and the normalized textual feature of each item. To further enhance the effectiveness of the pre-trained vision-language models under the product bundling task, we finetuned both CLIP and BLIP using the image-text pair of items in the Amazon review dataset.

Both CLIP and BLIP are finetuned using AdamW optimizer with the learning rate set to 3.0×10^{-5} and decayed linearly at the rate of 0.9 in each epoch. The weight decay in finetuning CLIP is set to 0.01 and weight decay in finetuning BLIP is set to 0.05. The batch size is set to 64 when finetuning CLIP and set to 24 when finetuning BLIP. Both models are finetuned on four NVIDIA A5000 (24G) GPUs.

2.3 Relational and Semantic Pre-training

These methods learn item representations from both semantic and relational information of the items. We first extract the multimodal features from both item texts and images using the finetuned BLIP, which is the best baseline model for multimodal feature extraction. The extracted semantic features will be used as input for training REL-SEM methods.

- **Feature Fusion (FF)** generates item representations by pooling relational and semantic item features. The pooling methods could be simple concatenation, average pooling or more advanced attention-based pooling. In this paper, we implement feature fusion by summing the normalized relational features with the normalized semantic features. FF-LightGCN fuses the features from the relational-only LightGCN model with semantic features extracted by the finetuned BLIP, and FF-SGL fuses the features from SGL and the finetuned BLIP.
- **GL-GCN** [5] combines relational features with semantic features by initializing node embeddings with multimodal features extracted by pre-trained vision-language models. The high-order item relations are modeled through graph propagation in GCN. The model is optimized by the task of link prediction with BPR loss.
- **GL-GCL** [11] follows the setting of GL-GCN, while the difference is that an additional graph contrastive learning is applied over two augmented views of GCN. Similarly, GL-GCL is trained with BPR loss.

Both GL-GCN and GL-GCL are optimized with Adam optimizer and the best hyper-parameter is determined by grid search. The learning rate is searched from 0.03 to 1.0×10^{-4} with weight decay searched from 3.0×10^{-5} to 1.0×10^{-7} , embedding size searched in range of [16, 32, 64, 128], number of propagation layers searched within [1, 2, 3]. Temperature of contrastive loss is tuned from 0.05 to 0.4 and the corresponding weighting coefficient is tuned from 0.001 to 0.5. The GL-GCN and GL-GCL are trained on a single NVIDIA A5000 (24G) GPU.

REFERENCES

- [1] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings*

- of the IEEE/CVF conference on computer vision and pattern recognition*. 9729–9738.
- [2] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *SIGIR*. ACM, 639–648.
- [3] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *ICLR (Poster)*.
- [4] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *ICDM*. IEEE Computer Society, 197–206.
- [5] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR (Poster)*. OpenReview.net.
- [6] Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. 2022. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In *ICML (Proceedings of Machine Learning Research, Vol. 162)*. PMLR, 12888–12900.
- [7] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Gotmare, Shafiq R. Joty, Caiming Xiong, and Steven Chu-Hong Hoi. 2021. Align before Fuse: Vision and Language Representation Learning with Momentum Distillation. In *NeurIPS*. 9694–9705.
- [8] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *ICML (Proceedings of Machine Learning Research, Vol. 139)*. PMLR, 8748–8763.
- [9] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).
- [10] Jiayi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *WSDM*. ACM, 565–573.
- [11] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised Graph Learning for Recommendation. In *SIGIR*. ACM, 726–735.

175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232