Self-supervised Learning for User Sequence Modeling

Yuhan Liu*
Cornell University
yl2976@cornell.edu

Lin Ning†
Google Research
linning@google.com

Neo Wu
Google Research
neowu@google.com

Karan Singhal
Google Research
karansinghal@google.com

Philip Andrew Mansfield
Google Research
memes@google.com

Devora Berlowitz
Google Research
devorab@google.com

Sushant Prakash
Google Research
sush@google.com

Bradley Green
Google Research
brg@google.com

Abstract

Self-supervised learning (SSL) has proven to be very effective in learning representations from unlabeled data, especially in vision and NLP tasks. We aim to transfer this success to user sequence modeling, where users perform a sequence of actions from a large discrete domain (e.g. video views, movie ratings). Since the data type is completely different from images or natural language, we can no longer use pretrained foundation models and must find an efficient way to train from scratch. In this work, we propose an adaptation of Barlow Twins, with a suitable augmentation method and architecture for user sequence data. We evaluate our method on the MovieLens 1M, MovieLens 20M, and Yelp datasets, observing an 8%-20% improvement in accuracy on three downstream tasks compared to the dual encoder model, which is commonly used for user modeling in recommendation systems. Our method can help to learn useful sequence-level information for user modeling, and it is especially beneficial with limited labeled data.

1 Introduction

Self-supervised learning (SSL) aims to learn useful and meaningful representations on large datasets without human-generated labels. The high-level idea is to apply transformations to the original data and learn representations that capture the essential underlying information invariant to the data transformations. It has achieved notable success in various domains such as computer vision [6, 5, 23, 10] and natural language processing [8, 14, 4, 17, 2], and it is a major driving force in recent advances of powerful foundation models [3, 17]. Common techniques in SSL include generative modeling [19, 18], reconstruction tasks [8, 12], and contrastive learning [5, 9].

We wish to transfer the great success of SSL in vision and NLP tasks to user sequence modeling, where users perform a sequence of actions from a large discrete domain (e.g. video clicking, rate a movie). A better model of user activities can significantly improve the performance of recommendation systems, which are ubiquitous among mobile applications and online services. Moreover, in many cases where visual, text, or voice input is limited, existing foundation models for vision and language may no longer be useful, and thus we must leverage information from other user activities.

*Work was completed during an internship at Google Research
†Corresponding author

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

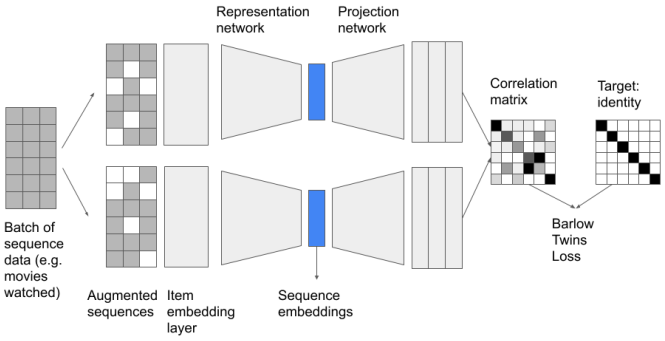

Figure 1: Illustration of Barlow Twins for user sequence data. Two independent augmentations are applied to the same batch, and the loss function enforces statistically independent components.

Performing SSL on user sequence data is not trivial. First, user activities are often sporadic and the intrinsic structure may not be as straightforward as images and natural language (e.g. images contain shapes, and languages have grammar). Moreover, there are technical challenges. In computer vision, many distortions (e.g. cropping, color jittering) used in SSL [5, 12] are specific to images and do not apply to user sequence data. Contrastive learning is used in both vision [6] and NLP tasks [9, 15, 16], but requires large batch with many negative samples. Several works [15, 13] use pretrained model weights [8, 14] to facilitate training, but such resourse is unavailable for user sequence data. Dual encoders [22] are commonly adopted in recommendation systems, which can also learn sequence-level representations. However, they are usually trained on one specific task and may not perform well for other purposes.

We propose to apply ideas from Barlow Twins [23] to user sequence modeling. Although originally proposed for computer vision, we demonstrate that with suitable data augmentation, our adaptation of Barlow Twins can learn meaningful sequence-level representations useful for a variety of downstream tasks. Moreover, it enjoys the same benefits as discussed in [23]: no need to use negative samples, robust to small batch sizes, and naturally avoids trivial (constant) embeddings.

On sequence data, Barlow Twins has been applied to audio [1], but the augmentation is in the spectral domain which does not apply to our problem. Thus, we view our adaptation of Barlow Twins to user sequence data as an important technical contribution.

2 Barlow Twins for user sequence data

Figure 1 shows an illustration of Barlow Twins adapted to user sequence modeling. The input is a sequence of user items with length ℓ, denoted by $\mathbf{u} = (u_1, \ldots, u_\ell) \in \mathbb{N}^\ell$. Each u_i is an integer-valued identifier that uniquely represents a user's action (e.g. a movie watched by the user). Then $\mathbf{u}$ is passed through an item embedding layer $E : \mathbb{N}^\ell \mapsto \mathbb{R}^{d_e \times \ell}$, which converts each integer ID into a d_e-dimensional embedding vector. Then, a representation network $R : \mathbb{R}^{d_e \times \ell} \mapsto \mathbb{R}^{d_r}$ transforms each sequence of embeddings into a sequence-level representation with d_r dimensions. Finally, we lift the sequence-level representation into higher dimensions using a projection network $P : \mathbb{R}^{d_r} \mapsto \mathbb{R}^{d_p}$. We denote the full model as $\text{BT} := P \circ R \circ E$.

During training, for each batch of sequences $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_b]$, we apply two independent augmentations and obtain two batches of augmented sequences $\mathbf{U}_1, \mathbf{U}_2$ (in our experiments, we use random masking in which each item in the sequence is masked independently with probability $p = 0.2$). Two augmented batches are passed through the same model BT and the output is subsequently mean-centered along the batch dimension, denoted by $\mathbf{Y}^i = [\mathbf{y}^i_1, \ldots, \mathbf{y}^i_{d_p}], i = 1, 2$. We minimize the Barlow Twins loss,

$$\mathcal{L}_{BT} := \sum_{i=1}^{d_p} (1 - \mathcal{C}_{ii}) + \lambda \sum_{i \neq j} \mathcal{C}_{ij}^2, \quad \mathcal{C}_{ij} := \frac{\sum_{j=1}^{b} \mathbf{y}^1_{1,j} \mathbf{y}^1_{2,j}}{\sqrt{\sum_{j=1}^{b} (\mathbf{y}_{1,j})^2} \sqrt{\sum_{j=1}^{b} (\mathbf{y}_{2,j})^2}} \tag{1}$$

where $\mathcal{C}$ is the cross correlation matrix along the batch dimension, and λ is a hyperparameter that balances the two terms. The loss enforces $\mathcal{C}$ to be close to an identical matrix, which guides the model to learn statistically independent components.

3 Experiments

We evaluate the quality of representations learned by the self-supervised learning algorithm with the MovieLens 1M, MovieLens 20M, and Yelp tips datasets. More specifically, we train a representation model with each dataset and evaluate the representations on two different types of downstream tasks: sequence-level classification tasks and next movie prediction. We only show part of the results for MovieLens 1M in this section. More experiments on other datasets and tasks are in the appendix.

3.1 Setup

Dataset The Movielens 1M dataset contains about 1 million movie ratings created by 5839 users over 3012 movies. Each movie is associated with a unique movie ID, the title, the year, and a list of genres. Each user is labeled with gender, age group, and occupation. For training and evaluation, we break down each user's movie sequence into smaller sequences of length 16 which *only* contain movie IDs, creating a training set with 800k sequences and a validation and test set with 100k sequences.

Pre-training We train a representation model on the training set with both the Barlow Twins model and a dual encoder baseline. The item embedding (i.e., embedding for the movies) dimension is set to 16. We vary the batch size over 128, 256, 512, and 1024. For the Barlow Twins model, we apply random masking on each pair of input sequences where each item in the sequence is masked out with probability 0.2. The representation network is a 2-layered 1D-CNN, where each layer has 16 convolution filters of size 3 followed by max pooling of size 3. The projection network is an MLP where the size of the layers is $[256, 256]$. We set the trade-off parameter $\lambda = 10$.

The baseline dual encoder has a context and an item tower. The context tower uses the same item embedding and representation network structure as the Barlow Twins model, followed by an MLP of size [32,16]. The item tower is simply the item embedding layer that shares weights with that in the context tower. During training, a batch of user sequence is passed through the context tower, and the batch of ground truth next movie is passed through the item tower. The outputs of the two towers are then used to compute the contrastive loss which is minimized during training.

Downstream evaluation We perform evaluation on a variety of downstream tasks. The input to these tasks is a sequence of movies watched by a user.

Sequence-level classification: predict the most frequent movie genre in the sequence (**Favorite genre prediction**, 18 genres) and occupation label (**Occupation classification**, 21 categories)

Next movie prediction (for Barlow Twins): predict the next movie.

For sequence classification, we take the pretrained Barlow Twins and dual encoder models up to the sequence embedding and add a 2-layered MLP with 20 hidden units and the same number of output units as the number of label categories in each task. The models are trained with either fixed or trainable weights for the sequence representation layers. As a baseline, we train a model with the same architecture from scratch. During training, 1%, or 100% of training data is used with batch size of 64, and the model is validated on the *full* validation dataset. Note that with less than 10% training data, the data used for training is (significantly) less than the validation/test dataset.

To evaluate the Barlow Twins model on the next movie prediction task, we initialize the item embedding and representation network in a dual encoder model with Barlow Twins pretrained weights. The model is trained on the full dataset with either trainable or fixed weights for sequence embedding and is compared to the performance of the dual encoder.

3.2 Results

Sequence-level classification Table 1 shows the best validation accuracy for the sequence classification tasks trained with 1% training data. The item embedding dimension=16 and convolution filters [32, 32]. Using Barlow Twins pretrained weights consistently outperforms other methods.

Task	SSL batch size	BT trainable	BT fixed	DE trainable	DE fixed	Baseline
FG	128	0.8247	**0.8405**	0.7325	0.7133	0.7350
	256	0.8235	**0.8462**	0.7460	0.7077	
	512	0.8100	**0.8402**	0.7549	0.7072	
	1024	0.8222	**0.8505**	0.7405	0.7049	
Occ	128	0.1534	**0.1540**	0.1384	0.1335	0.1407
	256	0.1430	**0.1558**	0.1324	0.1361	
	512	**0.1563**	0.1558	0.1345	0.1324	
	1024	0.1517	**0.1548**	0.1402	0.1330	

Table 1: Best validation accuracy of different models after training on 1% training data.

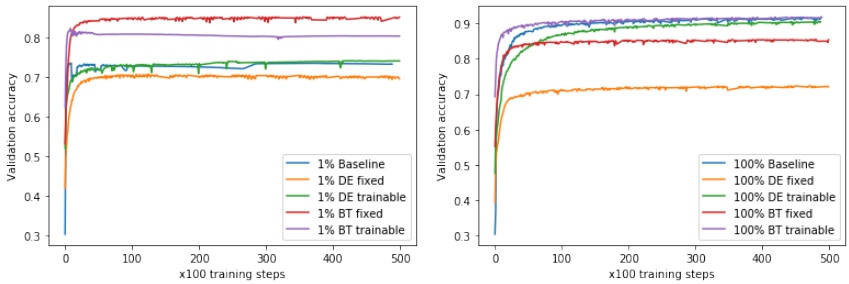

Figure 2: Favorite genre prediction with 1% (left) and 100% (right) training data. The batch size for SSL pretraining is 1024.

Interestingly, using fixed weights is generally preferred over trainable weights. We will argue in Figure 2 that this is because using fixed weights prevents overfitting. More results are in the appendix.

We show the validation curves for favorite genre prediction on 1% and 100% training data in Figure 2. With only 1% training data, models with trainable weights suffer from over-fitting, largely due to a very small training set compared to the validation set. Training with fixed weights from the Barlow twins model prevents over-fitting and yields the best and most stable performance. With 100% training data, using fixed weights yields bad performance, and the final validation accuracy of Barlow Twins and the baseline gradually converge. This is expected because with enough labeled data for a task, directly training from scratch is sufficient. However, we observe that using Barlow Twins weights, the accuracy curve converges much faster than the baseline, and consistently outperforms the model with weights from pretrained dual encoder (the latter even underperforms the baseline with fully trainable weights), which again demonstrates the advantage of Barlow Twins.

Next movie prediction In this experiment we use Barlow twins pretrained weights to build a dual-encoder model and compare with the dual encoder model which is trained specifically for this task. The error metric is top k recall where $k = 1, 5, 10$, which represents the percentage that ground-truth movie is in the top k nearest movies in terms of cosine similarity. Figure 3 shows the validation curves. Despite that the dual encoder model is trained specifically for the task, applying Barlow Twins pretrained weights even slightly improves the performance. We observe that with the pretrained weights fixed, the performance is significantly worse. This may be due to the fact that the Barlow Twins loss directly optimizes sequence-level embeddings and only indirectly affects item embeddings. More discussion is in Appendix C.

4 Conclusion

We explore applying self-supervised learning to learning general-purpose sequence-level representations in user modeling tasks. Our experiments show that Barlow Twins enjoys several practical benefits: 1. Barlow Twins learns useful sequence-level representations that are versatile for various downstream tasks; 2: it improves model performance and reduces overfitting with limited training data for downstream tasks; 3. it facilitates training without hurting the performance when the downstream task has sufficient data. The disadvantage is that the method focuses more on sequence-level

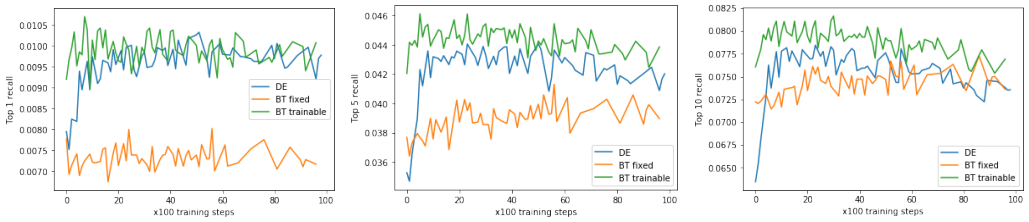

Figure 3: Top k validation recall for next movie prediction

representation rather than item-level representations, and therefore we may need to incorporate other techniques to improve the quality of item embeddings.

References

[1] Jonah Anton, Harry Coppock, Pancham Shukla, and Björn W Schuller. Audio barlow twins: Self-supervised audio representation learning. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[2] Alexei Baevski, Arun Babu, Wei-Ning Hsu, and Michael Auli. Efficient self-supervised learning with contextualized target representations for vision, speech and language. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 1416–1429. PMLR, 23–29 Jul 2023.

[3] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 2020.

[6] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 15750–15758. Computer Vision Foundation / IEEE, 2021.

[7] Yongjun Chen, Zhiwei Liu, Jia Li, Julian J. McAuley, and Caiming Xiong. Intent contrastive learning for sequential recommendation. In Frédérique Laforest, Raphaël Troncy, Elena Simperl, Deepak Agarwal, Aristides Gionis, Ivan Herman, and Lionel Médini, editors, *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 2172–2182. ACM, 2022.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[9] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6894–6910. Association for Computational Linguistics, 2021.

[10] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent - A new approach to self-supervised learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[11] F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4):19:1–19:19, 2016.

[12] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 15979–15988. IEEE, 2022.

[13] Fangyu Liu, Ivan Vulic, Anna Korhonen, and Nigel Collier. Fast, effective, and self-supervised: Transforming masked language models into universal lexical and sentence encoders. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 1442–1459. Association for Computational Linguistics, 2021.

[14] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[15] Yu Meng, Chenyan Xiong, Payal Bajaj, Saurabh Tiwary, Paul Bennett, Jiawei Han, and Xia Song. COCO-LM: correcting and contrasting text sequences for language model pretraining. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 23102–23114, 2021.

[16] Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David Schnurr, Felipe Petroski Such, Kenny Hsu, Madeleine Thompson, Tabarak Khan, Toki Sherbakov, Joanne Jang, Peter Welinder, and Lilian Weng. Text and code embeddings by contrastive pre-training. *CoRR*, abs/2201.10005, 2022.

[17] OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023.

[18] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China, 22–24 Jun 2014. PMLR.

[19] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In William W. Cohen, Andrew McCallum, and Sam T. Roweis, editors, *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, volume 307 of *ACM International Conference Proceeding Series*, pages 1096–1103. ACM, 2008.

[20] Lianghao Xia, Chao Huang, Chunzhen Huang, Kangyi Lin, Tao Yu, and Ben Kao. Automated self-supervised learning for recommendation. In Ying Ding, Jie Tang, Juan F. Sequeda, Lora Aroyo, Carlos Castillo, and Geert-Jan Houben, editors, *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, pages 992–1002. ACM, 2023.

[21] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pages 1259–1273. IEEE, 2022.

[22] Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernández Ábrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Multilingual universal sentence encoder for semantic retrieval. In Asli Celikyilmaz and Tsung-Hsien Wen, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, pages 87–94. Association for Computational Linguistics, 2020.

[23] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12310–12320. PMLR, 2021.

A Detailed Experiment Setup

We evaluate the self-supervised learning pipeline on three datasets: Movielens 1M, Movielens 20M [11], and Yelp. For each dataset, we first train a representation model, and then evaluate it on different downstream tasks. The Barlow Twins model is compared to a dual encoder baseline solely trained for next-item prediction.

A.1 Datasets

The Movielens 1M dataset contains about 1 million movie ratings created by 6040 users over 3952 movies. Each movie is associated with a unique movie ID, the title, the year, and a list of genres. Each user is labeled with gender, age group, and occupation. The Movielens 20M dataset contains 20 million movie ratings over 27278 movies created by 138493 users. The Yelp dataset contains user check-in and review data for 150346 businesses. We only use a small subset which contains 908915 tips made by 301758 users.

For training and evaluation, we break down each user's activity history into smaller sequences of length 16 which contain at least item IDs. Sequences with less than 16 items are padded with the empty [mask] token. Users with less than 10 actions are omitted. Note that the sequences only contain IDs and do not contain any other attributes of the items. The train-validation-test split is 80%-10%-10%. The dataset sizes are listed in Table 2

A.2 Pre-training

Models We train a representation model on the training set with both the Barlow Twins model and a dual encoder baseline. The item embedding (i.e., embedding for the movies) dimension is set to 16. We vary the batch sizes over 128, 256, 512, and 1024.

For the Barlow Twins model, the sequence representation network U is a 2-layered 1D-CNN, where each layer has 32 convolution filters of size 3 followed by max pooling of size 3. The projection network is an MLP where the size of the layers is $[256, 256]$. We set the trade-off parameter $\lambda = 10$. For augmentation methods, we explored random masking with $p = 0.2, 0.4, 0.6, 0.8$, segment masking with $p = 0.2$, and permutation.

The baseline dual encoder has a context and an item tower. The context tower uses the same item embedding and representation network structure as the Barlow Twins model, followed by an MLP of size [32,16]. The item tower is simply the item embedding layer that shares weights with that in the context tower. During training, a batch of user sequence is passed through the context tower, and the

	MovieLens-1M	MovieLens-20M	Yelp tips
# Users	6040	138493	301758
# Items	3952	27278	150436
# Actions	$\sim 10^6$	$\sim 2 \times 10^7$	908915
# Categories	18	18	1311
Train	795335	10776260	253317
Val and test	99417	1347032	28146

Table 2: Dataset statistics and the number of train/val/test sequences after preprocessing.

batch of ground truth next movie is passed through the item tower. The outputs of the two towers are then used to compute the contrastive loss which is minimized during training.

Augmentation Methods In our experiments, we tested on three different augmentation: random masking and segment masking.

1. Random masking (RM): each item in the sequence replaced with the mask token $[\text{mask}]$ independently with probability $p \in (0, 1)$.
2. Segment masking (SM): randomly select a subsequence of length $\lfloor p\ell \rfloor, p \in (0, 1)$ and replace all items in the subsequence with the mask token $[\text{mask}]$
3. Permutation: the input sequence is permuted uniformly at random. This augmentation method may be useful for position-invariant downstream tasks.

A.3 Downstream Evaluation

A.3.1 Tasks

we evaluate the quality of sequence representations on two types of tasks: sequence-level classification tasks and next item prediction. For the former, the performance metric is prediction accuracy, and for the latter, the metric is top k recall (or hit-ratio) where $k = 1, 5, 10$.

For MovieLens 1M, we can perform a variety of sequence-level classification tasks along with next item prediction. For MovieLens 20M and Yelp, we only perform favorite genre/category and next item prediction.

Sequence-level classification:

- **Favorite category prediction**: predict the most frequent movie category in the sequence (18 genres in MovieLens-1M and 20M, 1000+ categories in Yelp)
- **User classification (MovieLens-1M only)**: predict age group and occupation labels. There are 7 age groups and 21 occupation categories.

Next item prediction: predict the next item given a user's history interactions with items.

A.3.2 Model architecture and training setup

For sequence classification, we take the pretrained Barlow Twins and dual encoder models up to the sequence embedding and add a 2-layered MLP with 20 hidden units and the same number of output units as the number of label categories in each task. The models are trained with either fixed or trainable weights for the sequence representation layers U. As a baseline, we train a model with the same architecture (i.e. item embedding, 2-layered CNN, 2-layered MLP) from scratch.

During training, only a very small proportion of the training data is used with batch size of 64, and the model is validated on the *full* validation dataset. We ise 1% for MovieLens-1M, 1%,0.1%,0.01% for MovieLens-20M, and 5%,1% for Yelp tips. Note that with only 1% training data, the data used for training is (significantly) less than the validation/test dataset.

To evaluate the Barlow Twins model on the next item prediction task, we initialize the item embedding and representation network in a dual encoder model with Barlow Twins pretrained weights. The model is trained on the full dataset with either trainable or fixed weights for sequence embedding and is compared to the performance of the dual encoder.

Task	SSL BS	RM Train	RM Fixed	SM Train	SM Fixed	Per Train	Per Fixed	DE train	DE fixed	Baseline
FG	128	0.8247	0.8405	0.8474	**0.861**	0.7984	0.8002	0.7325	0.7133	0.7350
	256	0.8235	0.8462	0.8392	**0.8511**	0.7968	0.8003	0.7460	0.7077	
	512	0.8100	0.8402	0.8375	**0.8465**	0.7954	0.7949	0.7549	0.7072	
	1024	0.8222	**0.8505**	0.8485	0.8405	0.7844	0.7812	0.7405	0.7049	
Occ	128	0.1534	0.154	0.1471	**0.1548**	0.1359	0.1523	0.1384	0.1355	0.1407
	256	0.1430	**0.1558**	0.1403	**0.1558**	0.1483	0.1557	0.1324	0.1361	
	512	**0.1563**	0.1558	0.1446	0.1535	0.1488	0.1533	0.1345	0.1324	
	1024	0.1517	0.1548	0.1508	0.154	0.1457	**0.1556**	0.1402	0.133	
Age	128	**0.4055**	0.4015	0.4035	0.4004	0.4	0.4001	0.4017	0.397	0.3992
	256	**0.4078**	0.3998	0.4	0.4002	0.4023	0.4001	0.4034	0.397	
	512	**0.4112**	0.4017	0.4092	0.4004	0.4016	0.3975	0.4011	0.397	
	1024	0.403	0.3993	**0.4079**	0.4002	0.3996	0.3992	0.3973	0.397	

Table 3: Best validation accuracy of different models after training on 1% training data on MovieLens-1M. FG: favorite genre, Occ: occupation, SSL BS: SSL batch size, RM: random masking, SM: segment masking, Per: permutation, DE: dual encoder, Train: trainable. The highest accuracy in each row is in bold.

Task	SSL BS	R0.2 Train	R0.2 Fixed	R0.4 Train	R0.4 Fixed	R0.6 Train	R0.6 Fixed	R0.8 Train	R0.8 Fixed
FG	128	0.8247	**0.8405**	0.8047	0.8135	0.799	0.7366	0.7653	0.7411
	256	0.8235	**0.8462**	0.8163	0.7989	0.7766	0.7604	0.7612	0.6694
	512	0.8100	**0.8402**	0.8062	0.7953	0.7742	0.7328	0.7503	0.6747
	1024	0.8222	**0.8505**	0.7994	0.7928	0.7735	0.7236	0.7768	0.6591
Occ	128	0.1534	**0.154**	0.1389	0.1523	0.145	0.1515	0.1464	0.1411
	256	0.1430	**0.1558**	0.1462	0.1505	0.1406	0.1491	0.1348	0.1434
	512	**0.1563**	0.1558	0.1391	0.1516	0.134	0.15	0.1426	0.1408
	1024	0.1517	**0.1548**	0.1522	0.1544	0.1402	0.1481	0.138	0.1401
Age	128	**0.4055**	0.4015	0.4026	0.3974	0.4001	0.3999	0.4089	0.3978
	256	**0.4078**	0.3998	0.402	0.3989	0.403	0.3979	0.3975	0.3973
	512	**0.4112**	0.4017	0.4086	0.3978	0.4106	0.3989	0.4083	0.3973
	1024	0.403	0.3993	**0.413**	0.3995	0.4078	0.3977	0.4026	0.3973

Table 4: Best validation accuracy of random masking with different masking ratios after training on 1% training data on MovieLens 1M. R0.2, R0.4, R0.6, and R0.8 refer to the masking ratios of 0.2, 0.4, 0.6, and 0.8, respectively.

B Results

B.1 Sequence-level Classification

Table 3 and shows the best validation accuracy for the sequence classification tasks trained with 1% training data on MovieLens 1M. The item embedding dimension=16 and convolution filters [32, 32].

We observe that the best performing model for *all* tasks uses weights from Barlow Twins. For favorite genre and occupation, using Barlow Twins pretrained weights consistently outperforms the baseline methods (dual encoder fixed/trainable, baseline from scratch) by a large margin. Interestingly, using fixed weights is generally preferred over trainable weights. We will argue in Figure 5 that this is because using fixed weights prevents overfitting with extremely limited labeled training data.

For age classification, the advantage of Barlow Twins is less significant compared to the other two classification tasks, and we do not observe that Barlow Twins with fixed weights outperforms the fully trainable counterpart. However, models using Barlow Twins weights generally outperforms the baseline and their dual encoder counterpart (i.e., BT trainable > DE trainable, and BT fixed > DE fixed).

BS	Training Data Ratio								
	0.01			0.001			0.0001		
	Baseline	RM Train	RM Fixed	Baseline	RM Train	RM Fixed	Baseline	RM Train	RM Fixed
128	0.812	**0.817**	0.714	0.6447	**0.7207**	0.7059	0.4871	0.5765	**0.662**
256		**0.8172**	0.6986		**0.7165**	0.6903		0.5841	**0.6532**
512		**0.8203**	0.6973		**0.7085**	0.6923		0.5587	**0.6504**
1024		**0.8195**	0.6807		**0.7068**	0.6713		0.571	**0.6304**

Table 5: Best validation accuracy for favorite genre prediction on MovieLens-20M. Segment masking and permutation have similar performance to random masking.

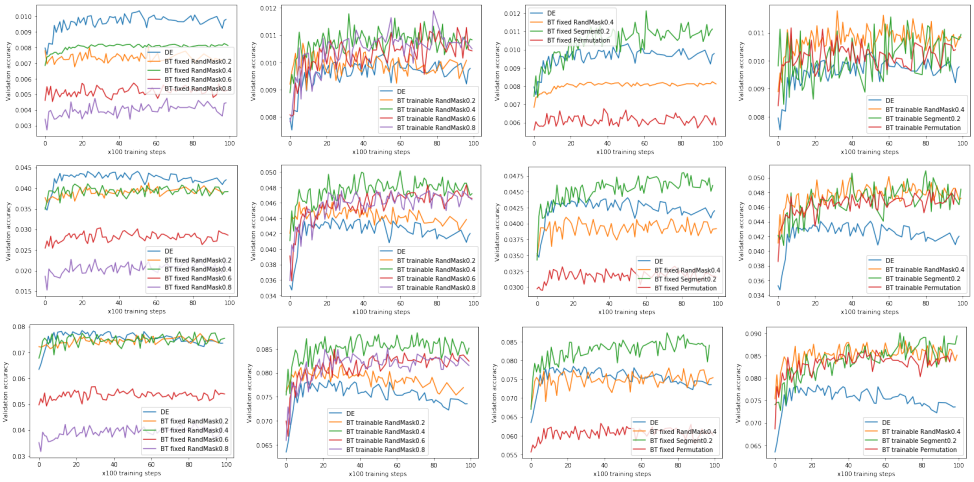

Figure 4: Validation recall (hit-ratio) for next movie prediction on MovieLens-1M. Barlow Twins/dual-encoder batch size=128. Three types of augmentations for Barlow Twins. *Top to bottom*: top 1, 5, 10 recall. *Left to Right*: random masking+ fixed weight, random masking + trainable weight, all augmentations + fixed weight, all augmentations + trainable weight.

On the contrary, models using representations from the dual encoder sometimes even under-performs the baseline with fully trainable weights, suggesting that dual encoder representation transfers poorly to other tasks.

Favorite category results for MovieLens-20M and Yelp tips dataset are reported in Table 5 and 6. For random masking we only report $p = 0.2$ which achieves the best result among all mask ratios for all classification tasks on MovieLens-1M. Similar to MovieLens-1M, using Barlow Twins weights is preferred over training from scratch. We vary the percentage of training data and discover that as the training data size decrease, using fixed weights becomes more advantageous.

B.2 Next Item Prediction

In this experiment we use Barlow twins pretrained weights to build a dual-encoder model and compare with the dual encoder model which is trained specifically for this task. The error metric is top k recall where $k = 1, 5, 10$, which represents the percentage that ground-truth movie is in the top k nearest movies in terms of cosine similarity.

For MovieLens-1M, Figure 4 shows the validation curves using different augmentations for Barlow Twins. The plots reported use a batch size of 128. Despite that the dual encoder model is trained specifically for the task, applying Barlow Twins pretrained weights can further improve the performance with trainable weights. Remarkably, with segment masking at $p = 0.2$, even using fixed weights (i.e., just training a very small MLP head in the context tower) can beat the dual encoder baseline.

We report the top-5 and 10 recalls for MovieLens-20M and Yelp tips dataset in Table 7. We omitted top-1 recall because these two datasets have a very large number of items as shown in Table 2. Thus precisely identifying the exact next item becomes a formidable task. Despite no longer observing the miracle that Barlow Twins with fixed weights outperform the baseline, we still observe significant improvement over dual encoders for models pretrained using Barlow Twins with fully trainable weights .

C Discussion

C.1 Effect of Different Augmentation Methods

We only discuss the results with *fixed weights* as it directly implies the quality of the learned representations.

BS	Training Data Ratio					
	0.01			0.05		
	Baseline	RM T	RM F	Baseline	RM T	RM F
128	0.2082	0.208	**0.2113**	0.4061	**0.4468**	0.2193
256		0.2081	**0.2177**		**0.4536**	0.2237
512		0.217	**0.2142**		**0.4438**	0.2241

Table 6: Best validation accuracy for favorite category prediction on Yelp dataset. RM T: random masking trainable, RM F: random masking fixed. Segment masking and permutation have similar performance to random masking.

Dataset	Metric	SSL BS	RM Train	RM Fixed	SM Train	SM Fixed	Per Train	Per Fixed	DE (baseline)
MovieLens 20M	Top-5	128	0.0291	0.0183	**0.0301**	0.0218	0.0291	0.0183	0.0265
		256	0.0264	0.0148	**0.0301**	0.0232	0.0264	0.0148	
		512	0.0292	0.0134	**0.0303**	0.0191	0.0292	0.0134	
		1024	0.0294	0.0132	**0.0298**	0.021	0.0294	0.0132	
	Top-10	128	0.0544	0.0352	**0.0557**	0.0404	0.0544	0.0352	0.0503
		256	0.0505	0.0276	**0.0562**	0.0432	0.0505	0.0276	
		512	0.0554	0.0268	**0.0555**	0.0372	0.0554	0.0268	
		1024	**0.0548**	0.0259	**0.0548**	0.0395	0.0548	0.0259	
Yelp	Top-5	128	0.0753	0.0483	**0.0756**	0.0608	0.0639	0.0309	0.0578
		256	**0.0751**	0.0483	0.0737	0.066	0.0605	0.0388	
		512	**0.0768**	0.0512	0.0751	0.0577	0.068	0.0505	
		1024	0.0709	0.0458	**0.0709**	0.0492	0.0619	0.048	
	Top-10	128	0.0935	0.07	**0.0954**	0.0847	0.0817	0.0492	0.0717
		256	**0.0931**	0.0669	**0.0931**	0.0899	0.0776	0.0601	
		512	**0.0966**	0.0672	0.0939	0.076	0.0867	0.0708	
		1024	0.0889	0.0577	**0.0891**	0.0616	0.0795	0.0647	

Table 7: Best validation top-5 and top-10 recall (HR) of next item prediction task on MovieLens 20M and Yelp datasets. SSL BS: SSL batch size, RM: random masking, SM: segment masking, Per: permutation, DE: dual encoder, Train: trainable. The highest accuracy in each row is in bold.

For random masking, high masking ratio ($p = 0.6, 0.8$) leads to poor performance as shown in the first column of Figure 4. We argue that when the ratio is too high, a lot of information in the sequence is discarded and thus it is hard to learn useful representations. $p = 0.2$ and $p = 0.4$ achieves decent performance for sequence-level classification tasks. However, for next item prediction, the performance is worse than the dual encoder baseline.

Segment masking at $p = 0.2$ is overall the best augmentation method, and is the only method that outperforms dual encoder baseline in next movie prediction. Interestingly, it outperforms random masking with the same mask ratio. We hypothesize that recovering a consecutive sub-sequence is more helpful for learning higher-level information about the user than recovering random isolated points in the sequence, as the former requires deeper understanding of user behaviors (e.g., user intention, habits, preferences), while the latter could be easily done by inferring from neighboring tokens.

Permutation is generally not preferred, even on permutation-invariant tasks like favorite genre prediction. A possible reason is that permutation breaks the temporal dependency in the sequence, which is crucial in understanding user behaviors.

C.2 Effect of SSL on the Training Process of Downstream Tasks

We show the validation curves on MovieLens 1M for favorite genre prediction on 1% and 100% training data in Figure 5. With only 1% training data, there are only less than 8k sequences for training while the validation set is more than 10 times as large. In this extreme but practically relevant setting, models with trainable weights suffer from over-fitting. This is further confirmed in Table 8 which shows the validation accuracy at the final step of training. Observe that using fixed weights from Barlow Twins consistently achieves the best final performance while only suffering very modest performance drop compared to the best validation accuracy. Thus we can conclude that training with fixed weights from the Barlow twins model prevents over-fitting when training on limited labeled data and yields the best and most stable performance.

Task	SSL batch size	BT trainable	BT fixed	DE trainable	DE fixed	Baseline
FG	128	0.7885	**0.8394**	0.7314	0.7122	0.7223
	256	0.7936	**0.8451**	0.7461	0.7041	
	512	0.7894	**0.8385**	0.7528	0.7027	
	1024	0.8028	**0.8494**	0.7402	0.6971	
Occ	128	0.1317	**0.1522**	0.1258	0.1296	0.1293
	256	0.1308	**0.1535**	0.1266	0.1356	
	512	0.1313	**0.1523**	0.1199	0.1275	
	1024	0.1307	**0.1515**	0.1282	0.1298	

Table 8: Final validation accuracy of sequence-level classification for different models with 1% training data on MovieLens-1M.

With 100% training data, using fixed weights yields bad performance, and the final validation accuracy of Barlow Twins and the baseline gradually converge. This is expected because with enough labeled data for a task, directly training from scratch is sufficient. However, we observe that using Barlow Twins weights, the accuracy curve converges much faster than the baseline, and consistently outperforms the model with weights from pretrained dual encoder (the latter even underperforms the baseline with fully trainable weights), which again demonstrates the advantage of Barlow Twins.

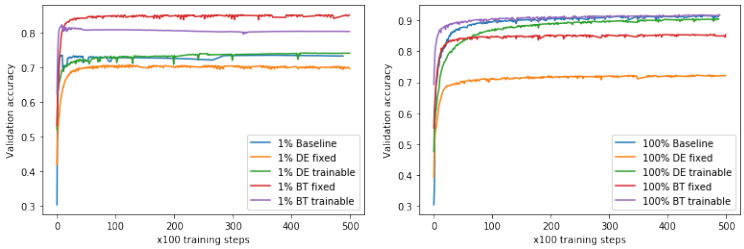

Figure 5: Favorite genre prediction with 1% (left) and 100% (right) training data. The batch size for SSL pretraining is 1024.

On next item prediction, we also observe that SSL pretraining can help reduce overfitting. The fact is most evident in the last column of Figure 4, where the accuracy of dual encoder gradually drops, while the performance of SSL-pretrained models remains steady and even slightly increases.

C.3 Effect of SSL Batch Size

From Tables 3 to 8 we can see that small batch sizes have no significant adverse effect on both types of downstream tasks. In rare cases, smaller batch size even has higher performance than large batch size (e.g. Table 3 FG, SM fixed; Table 7 Yelp dataset, SM Train). These results echo the findings in the original Barlow Twins paper [23]. In practice, training with smaller batch size requires less computational resource and increases convergence speed.

C.4 Item Embedding Visualization

To qualitatively evaluate the item embeddings, we draw the t-SNE plots (Figure 6) of the movie embeddings from three different genres (romance, horror, sci-fi) obtained from dual encoder and Barlow Twins with $p = 0.2$ random masking and segment masking trained on MovieLens-1M dataset. Intuitively, these three genres are drastically different and a good item embedding should be able to separate them into clusters. We can see that the three genres are well separated for the dual encoder, but poorly separated for Barlow twins.

We speculate that the reason for poor item embeddings of Barlow Twins is that the item embedding layers receive insufficient training signals from Barlow Twins loss, which is only applied at the very end of the model. On the other hand, the item embeddings receives direct signal in the item tower when minimizing the contrastive loss.

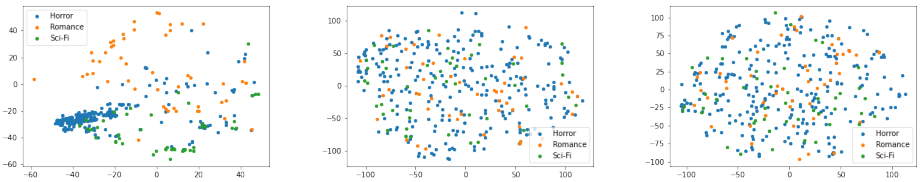

Figure 6: t-SNE plots of movie embeddings from 3 movie genres. Left: dual encoder. Middle: Barlow Twins with random masking. Right: Barlow Twins with segment masking.

We believe that the quality of sequence-level representations can be further improved with better item embeddings. Possible methods to improve item embeddings include reconstructing the masked actions similar to BERT [8] and training Barlow Twins in conjunction with next item prediction objective as was done in several previous works [21, 7, 20].