GraphSASRec: модель последовательных рекомендаций на основе самовнивания, дополненная графовыми представлениями.

A Preprint

Матвеев Артем
Московский государственный университет
имени М. В. Ломоносова
matfu21@ya.ru

Майсурадзе Арчил Ивериевич
Московский государственный университет
имени М. В. Ломоносова
artchil@mail.ru

2023

Abstract

Последовательные модели, решающие задачу предсказания следующего взаимодействия пользователя на основе кодирования его исторических событий, являются популярным решением для построение персонализированных рекомендательных систем как в индустрии, так и в академии. Преимуществами таких моделей являются: учет порядка, в котором следуют исторические события и оценка долгосрочных интересов пользователя. Однако подобные подходы недостаточно эксплуатируют полезный коллаборативный сигнал и плохо представляют объекты из длинного хвоста. Популярным решением этих проблем являются методы, основанные на применении графовых нейронных сетей к двудольному графу взаимодействий пользователей. В работе предлагается с другой стороны взглянуть на модель последовательных рекомендаций как на графовую сеть со стороны пользователя. Предлагается метод увеличения глубины этой сети, сохраняющий длину истории пользователя при минимальных накладных расходах. Предлагается способ корректировки смещения, вызванного семплированием из показательного распределения для задачи link-prediction. Наблюдаемые результаты демонстрируют улучшение с точки зрения метрик ранжирования и разнообразия выдачи.

Keywords Информационный поиск · Последовательные рекомендации · Графовые нейронные сети

1 Введение

Последовательные рекомендательные системы - это класс рекомендательных систем, которые принимают во внимание порядок взаимодействий пользователя и пытаются предсказать следующее его взаимодейсвие. Учет порядка является важной состоявляющей во многих рекомендательных сценариях: если пользователь только что купил мобильный телефон, то следующая покупка с большой долей вероятности будет аксессуаром к нему. Ранее такая задача решалась с помощью подходов, основанных на Марковских цепях [29] или рекуррентных нейронных сетях [5, 6, 14]. Но после появления архитектуры трансформер [33] и его успеха в задачах распознования естественного языка, наиболее успешными стали модели, основанные на обработке последовательностей действий пользователя с помощью трансформера [34, 12, 3].

Другим популярным подходом в рекомендательных системах являются графовые нейронные сети [36, 40, 19, 23, 41]. В этом подходе информация, которая есть в рекомендательной системе, рассматривается в виде графов. Большинство данных в любой рекомендательной системе по существу имееют графовую структуру. Например, данные взаимодействий в рекомендательном сервисе могут быть представлены в виде двудольного графа, вершины одной доли которого - пользователи, другой объекты (например,

товары), а наблюдаемый взаимодействия - ребра. Пусть нам дан такой граф. Ключевая идея графовых нейронный сетей заключается в итеративной агрегации признаковых представлений соседей в графе и объединение этой агрегированной информации с представлением вершины, для которой рассматривается соседство [31]. Такая операция называется распространением сообщений [36, 35]. Формально ее можно записать в следующем виде:

$$Aggregation : n_v^{(l)} = Aggregator_l(\{h_u^l, \forall u \in \mathcal{N}_v\}),$$
$$Update : h_v^{(l+1)} = Updater_l(h_v^{(l)}, n_v^{(l)}),$$

где $h_u^{(l)}$ определяется как представление вершины u после l-ого слоя графовой сети. $Aggregator_l$ и $Update_l$ представляют собой обучаемые функции агрегации соседей и обновления представления вершины на l-ом слое. В качестве функции агрегации могут выступать как простые варианты по типу max-pooling, mean-pooling [35], так и более сложные, например: importance-pooling [26], агрегация на основе контекста [7], механизм внимания [22], агрегация на базе трансформера [30]. В качестве функции обновления могут выступать как простые архитектуры на базе нескольких полносвязных слоев [35, 26], так и более сложные на базе трансформера [30].

Успех графовых подходов в рекомендательных системах можно объяснить тремя причинами [8]. Во-первых, выражая все данные в виде вершин и ребер графа, графовые нейронные сети предоставляют общий способ использовать все имеющиеся данные [2], тогда как традиционные рекомендательные системы чаще всего фокусируются на одном или небольшом количестве источников данных. Во-вторых, подобные модели могут явно утилизировать связи высокого порядка (товар X купил пользователь U, которому понравился товар Y, который в свою очередь похож на товар Z). В классических моделях этот учет происходит только неявно. Причем многие работы показывают, что от увеличения глубины графовой сети (то есть явного учета взаимодействий более высокого порядка) наблюдается рост целевых метрик [36, 35, 9]. В-третьих, целевой сигнал в рекомендательных системах очень разреженный (например, покупка). Графовые подходы позволяют использовать методы, основанные на обучении с частичным привлечением учителя [24], что приводит к улучшению качества моделей.

При внимательном взгляде на модели последовательных рекомендаций, можно увидеть в них графовую нейронную сеть со стороны пользователя. Как было упомянуто выше, графовые нейронные сети выучивают лучшее семантическое пространство, если начинать использовать в модели связи все большего порядка. Однако, на практике, длина последовательности пользователя варьируется от сотен [34, 16], до тысяч [20, 39] исторических событий. В этом случае, при построении соседств следующих уровней, возникает проблема экспоненциального роста их размера [35], что приводит к невозможности применения графовых методов в классическом виде.

Основной вклад заключается в следующем:

- Предлагается с другой стороны взглянуть на задачу последовательных рекомендаций. Найти в ней сходства с подходами, связанными с графовыми нейронными сетями, и использовать методы из этой области для улучшения моделей последовательных рекомендаций.
- Предлагается метод увеличения глубины связей, утилизируемых в моделях последовательных рекомендаций, на оснвое предобученных графовых представлений. При этом сохраняющей длину истории пользователя при минимальных накладных расходах.
- Представляется способ корректироваки, получающий несмещенную оценку градиента для функции потерть в задаче link-prediction. Приводятся результаты, демонстрирующие улучшения с точки зрения метрик полноты по сравнению с решением, не использующим эту корректировку.
- Представляется модифицированная архитектура модели последовательных рекомендаций, учитывающая связи более высоких порядков. Демонстрируются результаты, показывающие, что такой подходи приводит к росту метрик ранжирования.

2 Постановка задачи

2.1 Последовательные рекомендации

В последовательных рекомендациях рассматривается задача предсказания следующего положительного взаимодействия пользователя по последовательности его исторических действий $S^u = (S_1^u, S_2^u, \ldots, S_{|S^u|}^u)$. Во время обучения, на момент времени t, модель предсказывает следующий объект интереса пользователя на основе его взаимодействий, произошедших раньше момента t. На вход модели поступает

последоватлельность $(S_1^u, S_2^u, \ldots, S_{|S^u|-1}^u)$. Ожидаемый выход модели - следующее положительное взаимодействия $S_{|S^u|}^u$. В данной работе рассматривается модель SASRec [34], в основе которой лежит декодер блок трансформера [33], на выходе которого тоже последовательность. Поэтому задачу можно переформулировать в эквивалентном виде, как задачу предсказания сдвинутой версии последовательности $(S_2^u, S_3^u, \ldots, S_{|S^u|}^u)$.

2.2 Предсказание ребра

Дан граф $G = (V, E, X)$, где V - множество вершин, E - множество ребер, $X \in \mathbb{R}^{|V| \times d}$ - d-размерные векторные представления входных вершин. Задача предсказания ребра формулируется как задача определения существования (или появления в будущем, если граф рассматривается как динамический [42]) ребра e_{ij} между вершинами i и j, где $i, j \in V$, и $e_{ij} \notin E$.

3 Сопутствующие работы

3.1 Последовательные рекомендации в индустрии

В реальных рекомендательных системах процесс получения кандидатов для пользователя разбивается на две части: матчинг и ранжирование. На стадии матчинга применяются относительно легкие модели [15, 37], которые из миллионов-миллиардов кандидатов отбирают сотни-тысячи. На стадии ранжирования применяются тяжелые модели [28, 43, 17], уточняющие прогнозы моделей с предыдущей стадии. Модели последовательных рекомендаций применяются на обоих стадиях. Например, модель PinnerFormer [20] применяется на стадии матчинга. Ее особенность заключается в том, что предсказывается не просто следующее взаимодействие пользователя, а сразу несколько следующих взаимодействий, что позволяет учитывать долгосрочные интересы пользователя. Примером модели, которая применяется на стадии ранжирования является UserBody [16]. Ее особенность заключается в двух стадиях обучения: предобучение и дообучении. Предобучение происходит на стандарную для матчинга Sampled-Softmax функцию потерь. Дообучения происходит на задачу ранжирования с классическим попарным лоссом в качестве оптимизируемого критерия.

3.2 Графовые подходы в индустрии

В большинстве академических работ в моделях на основе графов каждой вершине ставится в соответствие обучаемый вектор [36]. Дальше такие модели учатся совместно с этими векторами. Минусом таких моделей является невозможность обобщения на новые вершины, которые со временем появляются в графе. Модели, обладающие таким свойством, называются трансдуктивными. Еще одним минусом трансдуктивного подхода являются огромные матрицы эмбеддингов, совпадающие по размеру с количеством вершин в графе. Это приводит к резкому росту числа параметров модели. Уже при миллионах вершин размеры моделей будут исчисляться миллиардами параметров, тогда как на практике количество вершин в графах может достигать и десятков миллиардов. Такой рост числа параметров приводит к необходимости использовать разреженные методы оптимизации (SparseAdam), что приводит к худшей траектории оптимизации. Именно поэтому в данной работе предлагается фокусироваться на подходах из индустрии.

В противовес к трансдуктивным графовым подходам выделяются индуктивные [35]. Их особенность заключается в том, что во настройки модели выучивается только преобразование над входными данными, что приводит к способности обобщаться на новые данные, которых раньше не было в обучающей выборке.

В индустрии можно выделить два подхода к добавлению графовой информации к моделям: end-to-end обучение вместе с целевой задачей и переиспользование заранее предобученных графовых векторов в последующих моделях для матчинга и ранжирования. Второй подход в свою очередь разбивается на два: трансдуктивные и индуктивные модели для получения предобученных графовых векторов.

3.3 End-to-end подходы

В подходе от Etsy [27] рассматривается двухбашенная модель для задачи матчинга в товарном поиске. Выбирается двудольных граф запрос-товар, где ребро проводится в случае, если с поискового запроса был переход на товар. Графовая структура здесь учитывается следующим образом: в башню над товаром дополнительно подаются запросы, связанные с товаром в двудольном графе. Т.к. таких запросов может

быть много, семплируется фиксированное количество, а дальше это соседство агрегируется с помощью усреднения. Похожие подходы используются у Amazon [13] и Taobao [18]. Еще один подход от Alibaba Group [44] заключается в том, чтобы на стадии обучения сближать представления, выдаваемые моделью последовательных рекомендаций с графовыми представлениями с помощью контрастивного обучения, тем самым заставляя модель выучивать неявным образом графовую структуру.

3.4 Трансдуктивные графовые модели

Как уже было сказано выше, трансдуктивные модели обладают рядом недостатков, что делает их обслуживание в реальных системах затруднительным. Однако одним из успешных примеров трандуктивной модели в индустрии является TwHIN [2] от Twitter. В основе TwHIN лежит модель TransE [4], в которой каждой вершине и каждому типу ребра присваивается свой обучаемый вектор. Для обучения такой модели используется специальный фреймворк PyTorch-BigGraph [1], который использует специальный механизм бакетирования, позволяющий работать с огромными матрицами эмбеддингов. Обучения векторов происходи на задачу link-prediction. Полученные предобученные векторы далее используются во всех рекомендательных моделях Twitter в замороженном виде как источник графовой структуры [38, 32]. Другим примером использования трандуктивной модели является Spotify [11].

3.5 Индуктивные графовые модели

Самым успешным примером индуктивного подхода является Pinterest: PinSage [26], MulitSage [7], MultiBiSage [30]. Получившиеся предобученные векторы далее используются в задача матчинга [20], ранжирования [39] и получении новых векторов [21]. Похожие работы есть у Spotify [9], Amazon [10] и Walmart [25].

4 Метод

Кратко, предлагаемый метод заключается в следующем:

- Строится двудольных граф взаимодействий пользователей с объектами.
- На основе этого графа выучиваются векторные представления вершин на задачу предсказания ребра (link-prediction).
- Предобученные векторные представления с помощью преобразования с нелинейностью переводятся в одно семантическое пространство с обучаемыми векторами для объектов из модели SASRec. К каждому объекту исторической последовательности прибавляется преобразованный графовый вектор. Преобразование над предобученными графовыми векторами обучается вместе с моделью SASRec на задачу предсказания следующего положительного взаимодействия пользователя (см. Рис. 1).

`Предобученные графовые векторы.` В качестве модели, получающей предобученные графовые представления, будет выступать GraphSAGE [35]. Формально, в общем случае, метод можно представить в следующем виде:

$$h^k_{\mathcal{N}(v)} \leftarrow AGGREGATE_k(\{h^{k-1}_u, \forall u \in \mathcal{N}(v)\}),$$
$$h^k_v \leftarrow \sigma(W^k \cdot CONCAT(h^{k-1}_v, h^k_{\mathcal{N}(v)})),$$
$$z_v \leftarrow \frac{h^k_v}{||h^k_v||_2}.$$

Особенностью метода GraphSAGE является семплирование соседства для каждой вершины, что приводит к борьбе с экспоненциальным ростом размера окрестности. Для получения эмбеддингов вершин будет рассматриваться только соседство первого уровня. Обучения происходит в постановке обучения без учителя на предсказание ребра в этом графе. Функция потерь в этому случае выглядит следующим образом:

$$\mathcal{L}(u, i) = -\log(\sigma(z_u^T z_v)) - \mathbb{E}_{v_n \sim P_n(v)} \log(\sigma(-z_u^T z_{v_n})).$$

Матожидание оценивается по Монте-Карло. На практике, семплирование происходит по негативным примерам из очередного батчка, что приводит к смещенной оценке. Решением этой проблемы является корректировка, представленная в секции 5.

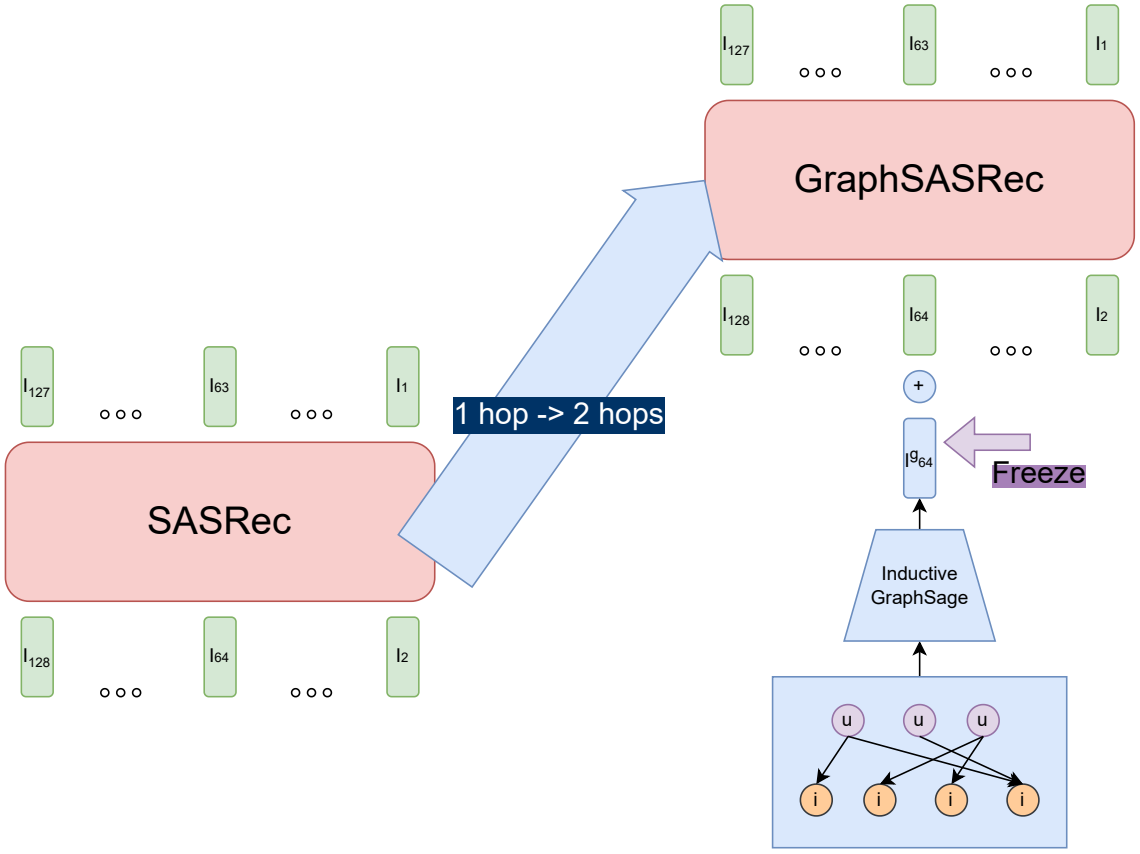

Рис. 1: Переход от модели SASRec к GraphSASRec.

`GraphSASRec`. Модель отличается от SASRec добавлением графовых векторов. Целевая задаче не меняется, все та же бинарная кросс-энтропия.

5 Корректировка смещения, вызванного семплированием

Классическая link-prediction функция потерь выглядит следующим образом:

$$\mathcal{L}(u, v) = -\log(\sigma(z_u^T z_v \times s + b)) - \frac{1}{|I_u|} \sum_{v_n \in I} \log(1 - \sigma(z_u^T z_{v_n} \times s + b)) \approx,$$

где s, b - обучаемые параметры, I_u - множество негативных ребер.

$$\approx -\log(\sigma(z_u^T z_v \times s + b)) - \mathbb{E}_{v_n \sim Unif[1,\dots,|I|]} \log(1 - \sigma(z_u^T z_{v_n} \times s + b)).$$

Вспоминаем, что u - пользователи, v - объекты, с которыми взаимодействуют пользователи в рекомендательной системе. Это число может быть очень большим, поэтому математическое ожидание оцениваем по Монте-Карло. Но каждый раз семплировать элементы для каждого примера в батче тоже долго, поэтому предлагается перейти к семплированию по in-batch негативам. Т.е. из семплирования по равномерному распределению на всех объектах нужно перейти к семплированию из естественного показательного распределения на объектах. Применим важностное семплирование:

$$-\log(\sigma(z_u^T z_v \times s + b)) - \mathbb{E}_{v_n \sim Power[1,\dots,|I|]} \log(1 - \sigma(z_u^T z_{v_n} \times s + b)) \frac{Unif(v_n)}{p(v_n)} \approx$$

Монте-Карло:

$$\approx -\log(\sigma(z_u^T z_v \times s + b)) - \frac{1}{batch\_size} \sum_{v_n \in Batch} \log(1 - \sigma(z_u^T z_{v_n} \times s + b)) \frac{1}{|I| p(v_n)}.$$

Получили, что такая корректировака эквивалента перевзвешиванию по вероятностям. Благодаря тому, что здесь меньше штрафуются популярные объекты, оптимизация на такую функцию потерь приводит к увеличению метрик полноты.

6 Эксперименты

Эксперименты проводились на наборе данных MovieLense-1M. Статистики по набору данных представлены в таблице 1.

Dataset	Users	Items	Interactions
MovieLense-1M	6,040	3,416	999,611

Таблица 1: Экспериментальные наборы данных.

Для SASRec все параметры были взяти из оригинальной статьи [34]. Итоговые результаты представлены в таблице 2.

Model	Recall@1	Recall@10	NDCG@10
SASRec	0.043	0.232	0.135
GraphSASRec	0.079	0.292	0.165

Таблица 2: Результаты.

7 Заключение

Предложенный подход показал улучшение с точки зрения метрик ранжирования. NDCG@10 было увеличиено на 22 процента для задачи предсказания следуюещго взаимодействия пользователя. Дальнейшие улучшения могли бы заключаться в рассмотрении сильно гетерогенных графов и более сложных архитектур графовых нейронных сетей.

Список литературы

[1] L. W. Adam Lerer. Pytorch-biggraph: A large-scale graph embedding system. 2019.

[2] T. M. Ahmed El-Kishky. Twhin: Embedding the twitter heterogeneous information network for personalized recommendation. 2022.

[3] C. M. Aleksandr Petrov. gsasrec: Reducing overconfidence in sequential recommendation trained with negative sampling. 2023.

[4] N. U. Antoine Bordes. Translating embeddings for modeling multi-relational data. 2013.

[5] A. K. B. Hidasi. Session-based recommendations with recurrent neural networks. 2016.

[6] A. A. C. Wu. Recurrent recommender networks. 2017.

[7] A. P. Carl Yang. Multisage: Empowering gcn with contextualized multi-embeddings on web-scale multipartite networks. 2020.

[8] Y. Z. Chen Gao. A survey of graph neural networks for recommender systems: Challenges, methods, and directions. 2023.

[9] A. Damianou. Podcast recommendations and search query using gnns at spotify | graph learning workshop 2022. 2022.

[10] Z. J. Elan Markowitz. Multi-task knowledge enhancement for zero-shot and multi-domain recommendation in an ai assistant application. 2023.

[11] A. D. Enrico Palumbo. Graph learning for exploratory query suggestions in an instant search system. 2023.

[12] J. L. Fei Sun. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. 2019.

[13] Y. H. Hanqing Lu. Graph-based multilingual product retrieval in e-commerce search. 2021.

[14] B. Hidasi and A. Karatzoglou. Recurrent neural networks with top-k gains for session-based recommendations. 2017.

[15] Y. K. Jon Eskreis-Winkler. Xwalk: Random walk based candidate retrieval for product search. 2023.

[16] A. F. Kirill Khrylchenko. Personalized transformer-based ranking for e-commerce at yandex. 2023.

[17] G. G. Liudmila Prokhorenkova. Catboost: unbiased boosting with categorical features. 2017.

[18] C. Z. Longbin Li. Graph contrastive learning with multi-objective for personalized product retrieval in taobao search. 2023.

[19] C. H. Mengru Chen. Heterogeneous graph contrastive learning for recommendation. 2023.

[20] A. Z. Nikil Pancha. Pinnerformer: Sequence modeling for user representation at pinterest. 2022.

[21] H. C. Paul Baltescu. Itemsage: Learning product embeddings for shopping recommendations at pinterest. 2022.

[22] Y. B. Petar Velickov. Graph attention networks. 2018.

[23] T. W. Qiang Cui. Herograph: A heterogeneous graph framework for multi-target cross-domain recommendation. 2020.

[24] Z. H. Qimai Li. Deeper insights into graph convolutional networks for semi-supervised learning. 2018.

[25] R. Y. M. Ramin Giahi. Gnn-gmvo: Graph neural networks for optimizing gross merchandise value in similar item recommendation. 2023.

[26] R. H. Rex Ying. Graph convolutional neural networks for web-scale recommender systems. 2018.

[27] S. S. Rishikesh Jha. Unified embedding based personalized retrieval in etsy search. 2023.

[28] R. S. Ruoxi Wang. Dcn v2: Improved deep and cross network and practical lessons for web-scale learning to rank systems. 2020.

[29] C. F. S. Rendle. Factorizing personalized markov chains for next-basket recommendation. 2010.

[30] N. P. Saket Gurukar. Multibisage: A web-scale recommendation system using multiple bipartite graphs at pinterest. 2022.

[31] F. S. Shiwen Wu. Graph neural networks in recommender systems: A survey. 2022.

[32] P. P. Vanessa Cai. Twerc: High performance ensembled candidate generation for ads recommendation at twitter. 2023.

[33] S. Vaswani. Attention is all you need. 2017.

[34] J. M. Wang-Cheng Kang. Self-attentive sequential recommendation. 2018.

[35] J. L. William L. Hamilton. Inductive representation learning on large graphs. 2018.

[36] K. D. Xiangnan He. Lightgcn: Simplifying and powering graph convolution network for recommendation. 2020.

[37] J. Y. Xinyang Yi. Sampling-bias-corrected neural modeling for large corpus item recommendations. 2019.

[38] Y. M. Xinyang Zhang. Twhin-bert: A socially-enriched pre-trained language model for multilingual tweet representations at twitter. 2023.

[39] P. E. Xue Xia. Transact: Transformer-based realtime user action model for recommendation at pinterest. 2023.

[40] C. H. Xuheng Cai. Lightgcl: Simple yet effective graph contrastive learning for recommendations. 2023.

[41] H. C. Yue Xu. Single-layer graph convolutional networks for recommendation. 2020.

[42] X. C. Yunfei Chu. Dynamic sequential graph learning for click-through rate prediction. 2021.

[43] Q. S. Zhiqiang Wang. Masknet: Introducing feature-wise multiplication to ctr ranking models by instance-guided mask. 2021.

[44] H. L. Ziyang Wang. Multi-level contrastive learning framework for sequential recommendation. 2022.