$\mathtt{M}^2$: Mixed Models with Preferences, Popularities and Transitions for Next-Basket Recommendation (Supplementary Materials)

Bo Peng, Zhiyun Ren, Srinivasan Parthasarathy, *Member, IEEE,* and Xia Ning*, *Member, IEEE*

S1 Next-Basket Recommendation vs Sequential Recommendation

In this paper, we focus on the problem of next-basket recommendation. We do not consider sequential recommendation methods as baselines in our experiments due to the fact that sequential recommendation methods usually assume that single items are interacted/purchased at different timestamps. This assumption does not really hold in next-basket recommendations that items could be interacted/purchased at the same timestamp. Thus, sequential recommendation methods might not be easily adapted for next-basket recommendations. However, our method can be extended to the sequential recommendation problem by designing an appropriate objective and replacing basket-level embeddings (Equation 6 in the main text) with item embeddings in each timestamp to be consistent with the setting of the sequential recommendation, as in the literature [1], [2], [3], [4]. We leave the investigation of extending $\mathtt{M}^2$ to the sequential recommendation as in our future work.

S2 Performance on the order-based split protocol

We apply a widely used protocol [1], [4], [5], [6] to split training, validation and testing sets based on the sequential order, and evaluate the methods on the last basket recommendation task. Specifically, in all the datasets, for each user, we use her/his last and second last basket as the testing and validation basket, respectively. We use the other baskets of each user for training and measure the training error on the last basket of the training data (i.e., the third last basket in the original sequences before split). We tune the parameters using grid search and use the best parameters in terms of recall@5 on the validation set during testing for the $\mathtt{M}^2$ and all the baseline methods. Table S1 presents the overall performance of all the methods in recommending the last basket. In this setting, we only consider users with at least 4 baskets. The number of users used in each dataset is in the parentheses after the dataset in the table. As shown in Table S1, the performance of $\mathtt{M}^2$ and baseline methods in this setting has a similar trend as that in the time-based split settings in the main paper. Overall, $\mathtt{M}^2$-gp^{2}t is still the best performing method. In terms of recall@k, $\mathtt{M}^2$-gp^{2}t achieves the best performance on all the datasets with a significant improvement over the best baseline methods. In terms of NDCG@k, $\mathtt{M}^2$-gp^{2}t could still achieve the best performance on 3 out of 4 datasets, and achieve the second best performance on the TMall dataset. In this setting, overall, $\mathtt{M}^2$-p^2 achieves similar performance with $\mathtt{M}^2$-gp^2, and both of them substantially outperform the best baseline methods on TMall, sTMall and Gowalla datasets. On the TaFeng dataset, $\mathtt{M}^2$-p^2 still achieves superior performance over the best baseline method `Sets2Sets`. These results demonstrate the effectiveness of $\mathtt{M}^2$ under the widely used but questionable order-based split protocol.

S3 Parameters for Reproducibility

We implemented $\mathtt{M}^2$ in python 3.7.3 with PyTorch 1.4.0 (https://pytorch.org). We used Adagrad optimizer with learning rate 1e-2 on all the datasets in all the tasks. We initialized all the learnable parameters using the default initialization methods in PyTorch [1]. The dimension of the hidden representation d, the time-decay parameter γ, and the regularization parameter λ that are specific for each dataset are reported in the $\mathtt{M}^2$-gp^{2}t column of Table S2. During the grid search, we initially searched d from $\{8, 16, 32, 64, 128\}$, γ from $\{0.2, 0.4, 0.6, 0.8, 1.0\}$, and λ from {1e-5, 1e-4, 1e-3, 1e-2} on all the datasets for all the methods which have the corresponding parameters. After that, if a parameter yields the best performance on the validation set when its value is on the boundary of the search

- *Bo Peng is with the Department of Computer Science and Engineering, The Ohio State University, Columbus, OH, 43210.*
E-mail: peng.707@buckeyemail.osu.edu
- *Srinivasan Parthasarathy and Xia Ning are with the Department of Biomedical Informatics, the Department of Computer Science and Engineering, and the Translational Data Analytics Institute, The Ohio State University, Columbus, OH, 43210.*
E-mail: srini@cse.ohio-state.edu, ning.104@osu.edu
- *Zhiyun Ren is with the Department of Biomedical Informatics, The Ohio State University, Columbus, OH, 43210.*
E-mail: ren.685@osu.edu
- **Corresponding author*

Manuscript received April 19, 2005; revised August 26, 2015.

[1] https://discuss.pytorch.org/t/whats-the-default-initialization-methods-for-layers/3157

TABLE S1: Performance Comparison on the Last Basket Recommendation

	method	recall@k			NDCG@k		
		k=5	k=10	k=20	k=5	k=10	k=20
TaFeng (9,629)	POP	0.0647	0.0760	0.1033	0.0894	0.0866	0.0936
	POEP	0.0749	0.1103	0.1534	0.1005	0.1041	0.1164
	Dream	0.0773	0.0912	0.1063	0.0967	0.0945	0.0982
	FPMC	0.0402	0.0520	0.0664	0.0464	0.0474	0.0517
	Sets2Sets	0.0857	0.1232	0.1767	0.1039	0.1112	0.1284
	M^2-p^2	0.0890	0.1261	0.1764	0.1169	0.1205	0.1350
	M^2-gp^2	0.0789	0.1170	0.1650	0.1062	0.1104	0.1245
	M^2-gp^{2}t	†**0.1095**	†**0.1496**	†**0.2038**	†**0.1384**	†**0.1431**	†**0.1594**
	improv	27.8%*	21.4%*	15.3%*	33.2%*	28.7%*	24.1%*
TMall (28,004)	POP	0.0594	0.0618	0.0653	0.0574	0.0584	0.0598
	POEP	0.0978	0.1267	0.1563	0.0703	0.0787	0.0869
	Dream	0.0628	0.0663	0.0735	0.0593	0.0607	0.0630
	FPMC	0.0588	0.0604	0.0663	0.0570	0.0576	0.0596
	Sets2Sets	0.1013	0.1339	0.1667	†0.0875	†0.0984	†0.1078
	M^2-p^2	0.1053	0.1366	0.1624	0.0758	0.0850	0.0920
	M^2-gp^2	0.1039	0.1362	0.1602	0.0750	0.0845	0.0912
	M^2-gp^{2}t	†**0.1108**	†**0.1407**	†**0.1700**	**0.0854**	**0.0943**	**0.1025**
	improv	9.4%*	5.1%*	2.0%*	-2.4%*	-4.2%*	-4.9%*
sTMall (204,206)	POP	0.0644	0.0666	0.0693	0.0624	0.0630	0.0638
	POEP	0.0894	0.1111	0.1220	0.0680	0.0749	0.0779
	Dream	0.0654	0.0690	0.0748	0.0626	0.0637	0.0651
	FPMC	0.0637	0.0656	0.0685	0.0616	0.0622	0.0629
	Sets2Sets	OOM	OOM	OOM	OOM	OOM	OOM
	M^2-p^2	0.0960	0.1205	0.1387	0.0719	0.0796	0.0845
	M^2-gp^2	0.0960	0.1206	0.1393	0.0719	0.0797	0.0847
	M^2-gp^{2}t	†**0.1011**	†**0.1245**	†**0.1397**	†**0.0806**	†**0.0880**	†**0.0922**
	improv	13.1%*	12.1%*	14.5%*	18.5%*	17.5%*	18.4%*
Gowalla (26,574)	POP	0.0112	0.0257	0.0409	0.0065	0.0115	0.0158
	POEP	0.4458	0.5050	0.5473	0.3703	0.3904	0.4019
	Dream	0.0255	0.0422	0.0682	0.0168	0.0225	0.0294
	FPMC	0.0162	0.0330	0.0627	0.0087	0.0145	0.0224
	Sets2Sets	0.3884	0.4722	0.5356	0.3103	0.3390	0.3563
	M^2-p^2	0.4498	0.5094	0.5531	0.3729	0.3932	0.4050
	M^2-gp^2	0.4501	0.5095	0.5505	0.3729	0.3931	0.4042
	M^2-gp^{2}t	†**0.4551**	†**0.5143**	†**0.5588**	†**0.3754**	†**0.3955**	†**0.4076**
	improv	2.1%*	1.8%*	2.1%*	1.4%*	1.3%*	1.4%*

For each dataset, the best performance among our proposed methods (i.e., M^2-p^2, M^2-gp^2 and M^2-gp^{2}t) is in **bold**, the best performance among the baseline methods is underlined, and the overall best performance is indicated by a dagger (i.e., †). The row "improv" presents the percentage improvement of the best performing methods among M^2-p^2, M^2-gp^2 and M^2-gp^{2}t (**bold**) over the best performing baseline methods (underlined) in each column. The numbers in the parentheses after the datasets represent the number of testing users in the datasets. The "OOM" represents the out of memory issue. The * indicates that the improvement is statistically significant at 95 percent confidence level.

range, we will extend the search range of this parameter, if applicable, until a value in the middle yields the best performance, while fixing the range of the other parameters.

For Sets2Sets, we used the implementation provided by the authors in GitHub [2]. We used the default Adam optimizer with learning rate 1e-4. We also used the default weight 10 for the partitioned set margin constraint. The other parameters that are dataset specific are reported in the Sets2Sets column of Table S2.

For Dream and FPMC, since we did not find available implementations provided by the authors online, we implemented Dream and FPMC by ourself. We implemented Dream and FPMC in python 3 with pytorch 1.4.0 (https://pytorch.org). We used Adagrad optimizer with learning rate 1e-2 on all the datasets in all the tasks. The other parameters that are dataset specific are reported in the Dream and FPMC columns of Table S2. The implementations of M^2-gp^{2}t, Dream and FPMC is available in GitHub [3].

TABLE S2: Best Parameters for M^2-gp^{2}t and Baseline Methods

	Dataset	M^2-gp^{2}t			Sets2Sets	Dream		FPMC	
		d	γ	λ	d	d	λ	d	λ
First	TaFeng	32	0.6	1e-2	64	64	1e-2	8	1e-7
	TMall	32	0.8	1e-4	128	512	1e-3	128	1e-3
	sTMall	8	1.0	1e-5	OOM	32	1e-3	16	1e-4
	Gowalla	128	0.6	1e-3	64	128	1e-3	64	1e-4
Second	TaFeng	16	0.6	1e-3	32	64	1e-2	32	1e-6
	TMall	64	0.6	1e-3	8	512	1e-3	32	1e-3
	sTMall	4	1.0	1e-4	OOM	128	1e-4	128	1e-4
	Gowalla	128	0.8	1e-3	128	128	1e-4	128	1e-4
Third	TaFeng	64	0.4	1e-3	128	32	1e-4	64	1e-3
	TMall	8	0.6	1e-3	8	128	1e-4	128	1e-3
	sTMall	64	1.0	1e-3	OOM	128	1e-4	128	1e-4
	Gowalla	64	0.8	1e-2	64	64	1e-3	128	1e-4

In this table, in M^2-gp^{2}t, d, γ and λ are the dimension of the hidden representation, time-decay parameter and regularization parameter. In Sets2Sets, d is the dimension of the hidden representation . In Dream and FPMC, d, λ are the dimension of the hidden representation and regularization parameter. The first column presents the tasks in which the reported parameters are used. The "First", "Second" and "Third" represents the task of recommending the first, second and third next basket. The M^2-gp^{2}t, Sets2Sets, Dream and FPMC columns present the best parameters on validation sets and thus are used in testing for M^2-gp^{2}t, Sets2Sets, Dream and FPMC, respectively.

S4 Overall performance at precision@k

S4.1 Overall performance at precision@k on the first next basket

Table S3 presents the performance of different methods at precision@k in recommending the first next basket. In this table, for each dataset, the best performance among M^2 variants (i.e., M^2-p^2, M^2-gp^2 and M^2-gp^{2}t) is in bold, the best performance among baseline methods (e.g., POP, POEP, Sets2Sets) is underlined, and the overall best performance is indicated by a dagger (i.e., †). Overall, the trend shown in Table S3 is very similar to that shown in Table 3 in the main text. In terms of precision@k, M^2-gp^{2}t is still the best performing method, which achieves the best performance on all the 4 datasets. Compared to the best baseline method on each dataset, M^2-gp^{2}t consistently achieves statistically significant improvement on all the datasets over all the metrics. These results strongly demonstrate the superior performance of M^2-gp^{2}t over the baseline methods.

S4.2 Overall performance at precision@k on the second next basket

Table S4 presents the overall performance at precision@k in recommending the second next basket. Generally, the results in Table S4 also show very similar trend with that in Table 4 in the main text. M^2-gp^{2}t is still the best performing method in this task at precision@k. M^2-gp^{2}t achieves the

[2] https://github.com/HaojiHu/Sets2Sets

[3] https://github.com/BoPeng112/PPT

TABLE S3: Performance Comparison on the Next Basket

	method	precision@k		
		k=5	k=10	k=20
TaFeng (7,227)	POP	0.0630	0.0381	0.0244
	POEP	0.0833	0.0613	0.0432
	Dream	0.0622	0.0368	0.0237
	FPMC	0.0460	0.0284	0.0188
	Sets2Sets	0.0746	0.0610	0.0453
	M^{2}-p^{2}	0.0907	0.0688	0.0487
	M^{2}-gp^{2}	0.0913	†**0.0691**	0.0486
	M^{2}-gp^{2}t	†**0.0946**	†**0.0691**	†**0.0496**
	improv	13.6%*	12.7%*	9.5%*
TMall (14,051)	POP	0.0174	0.0093	0.0055
	POEP	0.0281	0.0180	0.0113
	Dream	0.0180	0.0100	0.0058
	FPMC	0.0178	0.0092	0.0056
	Sets2Sets	0.0292	0.0193	0.0122
	M^{2}-p^{2}	0.0302	0.0197	0.0119
	M^{2}-gp^{2}	0.0303	0.0196	0.0117
	M^{2}-gp^{2}t	†**0.0311**	†**0.0201**	†**0.0126**
	improv	6.5%*	4.2%*	3.3%*
sTMall (94,337)	POP	0.0181	0.0093	0.0049
	POEP	0.0219	0.0133	0.0074
	Dream	0.0177	0.0093	0.0051
	FPMC	0.0176	0.0092	0.0049
	Sets2Sets	OOM	OOM	OOM
	M^{2}-p^{2}	0.0233	0.0146	0.0086
	M^{2}-gp^{2}	0.0233	0.0146	†**0.0087**
	M^{2}-gp^{2}t	†**0.0256**	†**0.0154**	†**0.0087**
	improv	16.9%*	15.8%*	17.6%*
Gowalla (12,975)	POP	0.0035	0.0037	0.0032
	POEP	0.1162	0.0673	0.0371
	Dream	0.0057	0.0047	0.0034
	FPMC	0.0036	0.0041	0.0042
	Sets2Sets	0.0994	0.0610	0.0354
	M^{2}-p^{2}	0.1170	0.0679	0.0372
	M^{2}-gp^{2}	0.1172	0.0676	0.0374
	M^{2}-gp^{2}t	†**0.1175**	†**0.0682**	†**0.0377**
	improv	1.1%*	1.3%*	1.6%*

For each dataset, the best performance among our proposed methods (i.e., M^{2}-p^{2}, M^{2}-gp^{2} and M^{2}-gp^{2}t) is in **bold**, the best performance among the baseline methods is underlined, and the overall best performance is indicated by a dagger (i.e., †). The row "improv" presents the percentage improvement of the best performing methods among M^{2}-p^{2}, M^{2}-gp^{2} and M^{2}-gp^{2}t (**bold**) over the best performing baseline methods (underlined) in each column. The numbers in the parentheses after the datasets represent the number of testing users in the datasets. The "OOM" represents the out of memory issue. The* indicates that the improvement is statistically significant at 95 percent confidence level.

best performance on 3 out of 4 datasets, and achieves the second best performance on the TaFeng dataset. M^{2}-gp^{2} is the second best performing method. It achieves the best performance on TaFeng, and the second best or (near) the second best performance on the TMall, sTMall and Gowalla datasets. It is also worth noting that compared to the best baseline method on each dataset, the best M^{2} variant (i.e., M^{2}-p^{2}, M^{2}-gp^{2} or M^{2}-gp^{2}t) achieves statistically significant improvement on all the datasets at all the metrics except the precision@5 in Gowalla.

TABLE S4: Performance Comparison on the Second Next Basket

	method	precision@k		
		k=5	k=10	k=20
TaFeng (2,801)	POP	0.0660	0.0431	0.0246
	POEP	0.0723	0.0528	0.0368
	Dream	0.0621	0.0353	0.0201
	FPMC	0.0341	0.0232	0.0151
	Sets2Sets	0.0491	0.0445	0.0360
	M^{2}-p^{2}	0.0731	0.0550	0.0396
	M^{2}-gp^{2}	†**0.0820**	†**0.0598**	†**0.0415**
	M^{2}-gp^{2}t	0.0808	0.0569	0.0399
	improv	13.4%*	13.3%*	12.8%*
TMall (5,109)	POP	0.0175	0.0096	0.0052
	POEP	0.0320	0.0214	0.0135
	Dream	0.0192	0.0105	0.0058
	FPMC	0.0183	0.0097	0.0055
	Sets2Sets	0.0340	0.0222	0.0140
	M^{2}-p^{2}	0.0344	†**0.0229**	0.0139
	M^{2}-gp^{2}	0.0347	0.0225	0.0143
	M^{2}-gp^{2}t	†**0.0356**	†**0.0229**	†**0.0144**
	improv	4.7%*	3.2%*	2.9%*
sTMall (29,741)	POP	0.0172	0.0091	0.0049
	POEP	0.0262	0.0168	0.0096
	Dream	0.0179	0.0094	0.0051
	FPMC	0.0178	0.0093	0.0049
	Sets2Sets	OOM	OOM	OOM
	M^{2}-p^{2}	0.0233	0.0146	0.0086
	M^{2}-gp^{2}	0.0281	0.0178	0.0105
	M^{2}-gp^{2}t	†**0.0288**	†**0.0182**	†**0.0106**
	improv	9.9%*	8.3%*	10.4%*
Gowalla (10,032)	POP	0.0041	0.0037	0.0032
	POEP	0.1213	0.0702	0.0386
	Dream	0.0062	0.0052	0.0038
	FPMC	0.0020	0.0025	0.0024
	Sets2Sets	0.0981	0.0613	0.0361
	M^{2}-p^{2}	0.1214	0.0705	0.0388
	M^{2}-gp^{2}	0.1214	0.0707	0.0388
	M^{2}-gp^{2}t	†**0.1219**	†**0.0709**	†**0.0394**
	improv	0.5%	1.0%*	2.1%*

For each dataset, the best performance among our proposed methods (i.e., M^{2}-p^{2}, M^{2}-gp^{2} and M^{2}-gp^{2}t) is in **bold**, the best performance among the baseline methods is underlined, and the overall best performance is indicated by a dagger (i.e., †). The row "improv" presents the percentage improvement of the best performing methods among M^{2}-p^{2}, M^{2}-gp^{2} and M^{2}-gp^{2}t (**bold**) over the best performing baseline methods (underlined) in each column. The numbers in the parentheses after the datasets represent the number of testing users in the datasets. The "OOM" represents the out of memory issue. The* indicates that the improvement is statistically significant at 95 percent confidence level.

S4.3 Overall performance at precision@k on the third next basket

Table S5 presents the overall performance at precision@k in recommending the third next basket. Similar to the trend in recommending the first, and second next basket, in the task of recommending the third next basket, M^{2}-gp^{2}t is still the best performing method. In terms of precision@5, it achieves the best performance on 3 out of 4 datasets. On the TaFeng dataset, it also achieves near the second best performance. M^{2}-gp^{2} is still the second best performing method, which achieves the best performance on TaFeng and sTMall, and the second best performance on the TMall and Gowalla datasets. Compared to the best baseline methods, in this task, the best M^{2} variant (i.e., M^{2}-p^{2}, M^{2}-gp^{2} or M^{2}-gp^{2}t)

TABLE S5: Performance Comparison on the Third Next Basket

	method	precision@k		
		k=5	k=10	k=20
TaFeng (1,099)	POP	0.0477	0.0383	0.0238
	POEP	0.0730	0.0525	0.0351
	Dream	0.0409	0.0258	0.0151
	FPMC	0.0269	0.0191	0.0126
	Sets2Sets	0.0462	0.0386	0.0318
	M^2-p^2	0.0752	0.0552	0.0363
	M^2-gp^2	†**0.0808**	†**0.0565**	†**0.0379**
	M^2-gp^2t	0.0744	0.0535	0.0358
	improv	10.7%*	7.6%*	8.0%*
TMall (1,461)	POP	0.0149	0.0078	0.0041
	POEP	0.0387	0.0258	0.0164
	Dream	0.0151	0.0080	0.0046
	FPMC	0.0162	0.0085	0.0049
	Sets2Sets	0.0385	0.0255	0.0167
	M^2-p^2	0.0398	0.0262	0.0166
	M^2-gp^2	0.0404	†**0.0266**	0.0165
	M^2-gp^2t	†**0.0411**	0.0258	†**0.0169**
	improv	6.2%*	3.1%	1.2%
sTMall (7,561)	POP	0.0165	0.0087	0.0046
	POEP	0.0294	0.0194	0.0117
	Dream	0.0175	0.0092	0.0050
	FPMC	0.0172	0.0090	0.0048
	Sets2Sets	OOM	OOM	OOM
	M^2-p^2	0.0314	†**0.0207**	†**0.0123**
	M^2-gp^2	0.0315	†**0.0207**	†**0.0123**
	M^2-gp^2t	†**0.0320**	0.0205	0.0122
	improv	8.8%*	6.7%*	5.1%*
Gowalla (7,985)	POP	0.0039	0.0041	0.0034
	POEP	0.1346	0.0774	0.0427
	Dream	0.0061	0.0049	0.0037
	FPMC	0.0058	0.0064	0.0061
	Sets2Sets	0.1138	0.0696	0.0404
	M^2-p^2	0.1360	0.0780	0.0429
	M^2-gp^2	0.1361	†**0.0783**	0.0428
	M^2-gp^2t	†**0.1366**	†**0.0783**	†**0.0432**
	improv	1.5%*	1.2%*	1.2%*

For each dataset, the best performance among our proposed methods (i.e., M^2-p^2, M^2-gp^2 and M^2-gp^2t) is in **bold**, the best performance among the baseline methods is underlined, and the overall best performance is indicated by a dagger (i.e., †). The row "improv" presents the percentage improvement of the best performing methods among M^2-p^2, M^2-gp^2 and M^2-gp^2t (**bold**) over the best performing baseline methods (underlined) in each column. The numbers in the parentheses after the datasets represent the number of testing users in the datasets. The "OOM" represents the out of memory issue. The* indicates that the improvement is statistically significant at 95 percent confidence level.

TABLE S6: Frequency Distribution of Recommended items (%)

	method	t1	t2	t3	t4	t5	b5	b4	b3	b2	b1
TaFeng	POEP	10.2	10.2	10.2	10.2	10.2	10.2	10.0	9.9	9.6	9.3
	Dream	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	FPMC	19.4	10.3	9.9	9.8	8.8	8.3	7.9	8.4	8.5	8.7
	Sets2Sets	13.4	13.4	13.3	12.7	11.5	10.0	8.7	7.3	5.6	4.1
	M^2-gp^2	11.8	11.8	11.7	11.5	11.1	10.4	9.8	8.9	7.5	5.5
	M^2-gp^2t	12.3	12.3	12.3	12.1	11.2	10.4	9.6	8.3	6.7	4.9
TMall	POEP	10.2	10.2	10.2	10.2	10.3	10.2	10.2	10.3	10.2	8.0
	Dream	71.1	12.4	3.4	4.5	2.3	1.9	0.4	1.5	0.0	2.6
	FPMC	12.8	10.3	9.9	9.9	9.9	9.5	9.7	9.9	9.1	9.0
	Sets2Sets	10.6	10.5	10.6	10.6	10.6	10.5	10.4	10.3	9.9	6.2
	M^2-gp^2	10.3	10.2	10.2	10.3	10.3	10.3	10.2	10.3	10.2	7.8
	M^2-gp^2t	10.3	10.2	10.3	10.3	10.3	10.3	10.3	10.3	10.2	7.6

The columns t1, t2, t3, t4 and t5 correspond to the top-10%, top 10-20%, top 20-30%, top 30-40% and top 40-50% most frequent items, respectively. The columns b1, b2, b3, b4 and b5 correspond to the bottom-10%, bottom 10-20%, bottom 20-30%, bottom 30-40% and bottom 40-50% most frequent items, respectively.

still achieves statistically significant improvement over the best baseline methods at all the metrics on TaFeng, sTMall and Gowalla. On the TMall dataset, M^2-gp^2t also achieves statistically significant improvement over the best baseline method POEP at precision@5.

S5 Analysis on the diversity of recommendations

We also evaluate the diversity of the recommendations generated by different methods. Specifically, for each method, we consider the top-20 recommend items for each user, and then bin the recommended items into different buckets based on their frequencies in the dataset. In this analysis, we have 10 buckets in total: 5 buckets are for the top-10%, top 10-20%, top 20-30%, top 30-40% and top 40-50% most frequent items; the other 5 buckets are for the bottom 40-50%, bottom 30-40%, . . . , bottom-10% most frequent items. We conduct the analysis on the widely used TaFeng and TMall datasets. All the methods are trained using the training and validation baskets for the task of recommending the first next basket. The frequencies of items are also calculated from the training and validation baskets. In this analysis, for the baseline methods, we do not consider POP since POP only recommends the most frequent items (Section 5.1 in the main text). Among the M^2 variants, we consider the two best performing methods M^2-gp^2 and M^2-gp^2t.

Table S6 presents the percentage distributions of recommended items in different buckets. For example, on TaFeng, for M^2-gp^2t, 12.3% of the recommended items are among the top-10% most frequent items. As shown in Table S6, M^2-gp^2t could generate more diverse recommendations compared to the model-based baseline methods (i.e., Dream, FPMC and Sets2Sets). For example, for M^2-gp^2t, 12.3% of the recommended items are among the top-10% most frequent items, while for Dream, FPMC and Sets2Sets, more recommender items, that is, 100.0%, 19.4% and 13.4%, respectively, are among the top-10% most frequent items. That is, on TaFeng, M^2-gp^2t recommends fewer most frequent items compared to these baseline methods. This result indicates better diversity among the recommended items generate by M^2-gp^2t. We also found a similar trend in TMall. On TMall, 10.3% of the recommended items from M^2-gp^2t are among the top-10% most frequent items, while for Dream, FPMC and Sets2Sets, the percentage increases to 71.1%, 12.8% and 10.6%, respectively. We noticed that the baseline method POEP recommends slightly more diverse items compared to M^2-gp^2t. For example, on TaFeng, 10.2% of the recommended items from POEP are among the top-10% most frequent items, while for M^2-gp^2t, the percentage increases to 12.3%. However, as shown in Table 3 (main text) and Table S3, M^2-gp^2t statistically significantly outperforms POEP on all the datasets over most of the metrics. Considering both the diversity and quality of the recommendations,

$\mathtt{M}^2$-gp^{2}t could still substantially outperform POEP in real applications.

S6 Limitations of the Dunnhumby and Instacart datasets

We notice that in the experiments of Sets2Sets, the authors used the Dunnhumby dataset[4]. However, since this dataset is simulated, the results on this dataset may not necessarily represent the models' performance in real applications. Thus, we do not use this dataset in our experiments. It is worth noting that, although we do not use Dunnhumby in our final experiments, we compared $\mathtt{M}^2$ and baseline methods on this dataset in our preliminary study, and found that $\mathtt{M}^2$ performs at least similarly to the best baseline method. For example, in terms of recall@5 (Section 5.4 in the main text), $\mathtt{M}^2$-gp^{2}t achieves the best performance at 0.1500, and the best baseline method POEP achieves similar performance as 0.1499.

The Instacart dataset[5] is another dataset used in the literature [7]. However, the original version of this dataset is not publicly available now. We found on the Kaggle dataset [6] that there is no absolute time information for each basket, and we cannot conduct time-based split as that presented in the main text (Section 5.3). Considering the above limitations, we also do not use the Instacart dataset in our experiments.

S7 Intra-Basket Item Complementarities

We noticed that some recent publications [7] assume that items in the same basket are complementary and show that modeling this pattern could slightly improve the recommendation performance. However, this pattern actually highly depends on the datasets and how the baskets are defined. For example, in the online shopping scenario (i.e., the most popular recommendation scenario), the baskets are usually defined as all the purchased items in one "cart". The items in the same "cart" could be added in very different timestamps. Thus, it might not always be reasonable to assume they are complementary. Based on our experiments and the existing literature, we did not find concrete evidences to show that most of the items in a basket could be complementary. Actually, we also tried to model the intra-basket item complementarities in $\mathtt{M}^2$ by regularizing items in the same baskets to have similar embeddings. However, it did not improve the performance on benchmark datasets, while significantly increased the training time. Considering the above reasons, we did not model the intra-basket item complementarities in $\mathtt{M}^2$.

References

[1] J. Tang and K. Wang, "Personalized top-n sequential recommendation via convolutional sequence embedding," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018, pp. 565–573.

[2] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 197–206.

[3] C. Ma, P. Kang, and X. Liu, "Hierarchical gating networks for sequential recommendation," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019, pp. 825–833.

[4] B. Peng, Z. Ren, S. Parthasarathy, and X. Ning, "HAM: Hybrid associations models for sequential recommendation," *IEEE Transactions on Knowledge and Data Engineering*, no. 01, p. early access, jan 2021.

[5] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, "A dynamic recurrent model for next basket recommendation," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2016, pp. 729–732.

[6] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 811–820.

[7] M. Wan, D. Wang, J. Liu, P. Bennett, and J. McAuley, "Representing and recommending shopping baskets with complementarity, compatibility and loyalty," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 1133–1142.

[4] https://www.dunnhumby.com/source-files/

[5] https://www.instacart.com/datasets/grocery-shopping-2017

[6] https://www.kaggle.com/c/instacart-market-basket-analysis/data