1 EFFICIENCY

[R1,R2,R3,R4] Thank you for the valuable feedback and suggestions. As suggested by the reviewer, we explicitly calculate the computation time (second) and GPU memory occupation (MB) at each epoch.

Model	Gowalla			Foursquare		
	Train	Inference	Memory	Train	Inference	Memory
Flashback	37s	302s	10237MB	54s	912s	14697MB
Graph-Flashback	85s	431s	9531MB	120s	1320s	13289MB
SNPM	306s	631s	10887MB	420s	1910s	16365MB
our	8s	100s	5061MB	13s	420s	7467MB

Table 1: Performance about efficiency.

2 COMPLETE ABLATION EXPERIMENTS

[R1, R2, R4, R6] We supplemented ablation experiments by removing the Potential Missing Nodes Completion Strategy (i.e., w/o PMNCS) and the Confidence-based Weight Approach (i.e., w/o confidence weight). Additionally, we added ablation experiments on the Foursquare dataset.

Model	Gowalla				Foursquare			
	Acc@1	Acc@5	Acc@10	MRR	Acc@1	Acc@5	Acc@10	MRR
MLP(student model)	0.1544	0.3484	0.4328	0.2467	0.2821	0.5912	0.6741	0.4221
ST model	0.1567	0.3618	0.4505	0.2523	0.2862	0.6054	0.6842	0.4311
KD	0.1556	0.3517	0.4376	0.2506	0.2866	0.5981	0.6792	0.4291
KD-singel	0.1602	0.3594	0.4487	0.2540	0.2912	0.6058	0.6883	0.4320
w/o mask	0.1662	0.3637	0.4542	0.2578	0.2951	0.6125	0.6957	0.4353
w/o ST distill	0.1656	0.3629	0.4536	0.2572	0.2932	0.6103	0.6923	0.4326
w/o PMNCS	0.1537	0.3480	0.4332	0.2446	0.2801	0.5901	0.6722	0.4207
w/o confidence weight	0.1588	0.3628	0.4516	0.2562	0.2893	0.6083	0.6892	0.4353

Table 2: Performance about ablation study.

3 COMPARED WITH OTHER BASELINE KNOWLEDGE DISTILLATION METHODS

[R2,R6]

Model	Gowalla				Foursquare			
	Acc@1	Acc@5	Acc@10	MRR	Acc@1	Acc@5	Acc@10	MRR
ExplanationIntervention	0.0672	0.1673	0.2238	0.1196	0.2126	0.4827	0.5560	0.3341
KRD	0.1591	0.3522	0.4336	0.2502	0.2902	0.5970	0.6769	0.4281
MMKD	0.1610	0.3554	0.4404	0.2551	0.2912	0.6047	0.6812	0.4334
MLP-POI	**0.1684**	**0.3689**	**0.4561**	**0.2602**	**0.2971**	**0.6167**	**0.6983**	**0.4379**

Table 3: Performance about KD.

Model	Gowalla				Foursquare			
	Acc@1	Acc@5	Acc@10	MRR	Acc@1	Acc@5	Acc@10	MRR
STAN	0.1502	0.3424	0.4261	0.2416	0.2801	0.5789	0.6574	0.4142
Flashback	0.1532	0.3473	0.4301	0.2442	0.2814	0.5813	0.6621	0.4176
GETNext	0.1562	0.3501	0.4342	0.2484	0.2877	0.5955	0.6745	0.4257
Graph-Flashback	0.1596	0.3550	0.4382	0.2533	0.2914	0.6009	0.6790	0.4291
SNPM	0.1634	0.3613	0.4472	0.2576	0.2931	0.6082	0.6873	0.4338

Table 4: Performance about the replacement of multi-teacher.

4 VARIOUS MODELS STRUCTURE OF THE MULTI-TEACHER SPATIAL-TEMPORAL MODEL

[R2, R5, R6] We employed the following four models to substitute for the multi-teacher model.

5 WHY SNPM WASN'T USED FOR INITIALIZATION?

[R2] We tried initializing both the student and teacher models using methods from SNPM, but obtained poor results. Specifically, the poor performance of initializing the student MLP model may be attributed to its initialization method, which focuses on capturing current sequence information, overlapping somewhat with the role of MLP in summarizing current sequence information.

Model	Gowalla				Foursquare			
	Acc@1	Acc@5	Acc@10	MRR	Acc@1	Acc@5	Acc@10	MRR
$\text{Student}_{\text{SNPM}}$	0.1522	0.3415	0.4211	0.2438	0.2787	0.5804	0.5594	0.4176
$\text{Student}_{\text{our}}$	0.1544	0.3484	0.4328	0.2467	0.2839	0.5896	0.6687	0.4213
$\text{Teacher}_{\text{SNPM}}$	0.1534	0.3429	0.4185	0.2478	0.2823	0.5864	0.6652	0.4251
$\text{Teacher}_{\text{our}}$	0.1567	0.3618	0.4505	0.2523	0.2873	0.6082	0.6914	0.4301

Table 5: Performance about using SNMP initialization.

6 COMPARISON BETWEEN THE SELF-ATTENTION MECHANISM (S.A.) AND MLP.

[R3] Our goal is to accelerate the Transformer-based POI algorithm, making the POI algorithm more efficient without compromising performance. Self-attention operations in Transformers typically exhibit a computational complexity of $O(n^2 d)$ and a storage complexity of $O(n^2)$, necessitating pairwise attention score computation across the input sequence of length n. This quadratic dependency on sequence length poses a considerable hurdle when scaling to longer user sequences. To overcome this challenge, we introduce the Effective MLP-based Reliable Distillation for POI Recommendation approach, which significantly reduces computational complexity to $O(n)$, enabling more scalable solutions. Our experimental findings, particularly in terms of efficiency and GPU storage space, further underscore the superiority of our method compared to baseline methods.

7 SUPPLEMENTED BASELINE EXPERIMENTS.

[R4] Thank you for the timely reminder. As shown in Table. 6, we have further supplemented the two robust baseline experiments you mentioned. It is evident that their performance still falls short compared to our method.

8 LACKS IMPLEMENTATION DETAILS AND HYPERPARAMETER TUNING ASPECTS.

[R5, R6] Additionally, regarding hyperparameter tuning, we will include experiments in the appendix based on the random walk length and the number of teacher models, along with the addition of the threshold for random sampling weighting and balancing weight in training strategy.

Model	Gowalla				Foursquare			
	Acc@1	Acc@5	Acc@10	MRR	Acc@1	Acc@5	Acc@10	MRR
HMT-GRN	0.1312	0.3112	0.3921	0.2201	0.2643	0.5578	0.6342	0.3941
STHGCN	0.1543	0.3462	0.4311	0.2461	0.2842	0.5917	0.6702	0.4232
MLP-POI	**0.1684**	**0.3689**	**0.4561**	**0.2602**	**0.2971**	**0.6167**	**0.6983**	**0.4379**

Table 6: Performance about additional baseline.