

Personalized Task Dependency Graphs for Mitigating Signal Erosion in Multi-Task Recommendation

Fuyuan Liu*
liufuyuan2@huawei.com
Huawei Technologies Co., Ltd.
Shanghai, China

Tiandeng Wu*[†]
wutiandeng1@huawei.com
Huawei Technologies Co., Ltd.
Shanghai, China

Yaqun Fang
fangyaqun@huawei.com
Huawei Technologies Co., Ltd.
Dongguan, Guangdong, China

Wei Zhou
zhouwei281@huawei.com
Huawei Technologies Co., Ltd.
Nanjing, Jiangsu, China

Zehao Zhou
zhouzehao@huawei.com
Huawei Technologies Co., Ltd.
Shanghai, China

Wenping Chen
chenwenping15@huawei.com
Huawei Technologies Co., Ltd.
Shanghai, China

Qishun Mei
meiqishun1@huawei.com
Huawei Technologies Co., Ltd.
Shanghai, China

Jiaxin Zhou
zhoujiaxin15@huawei.com
Huawei Technologies Co., Ltd.
Shanghai, China

Heng Chang
changh17@tsinghua.org.cn
Huawei Technologies Co., Ltd.
Beijing, China

Yi Cao
caoyi23@huawei.com
Huawei Technologies
Shanghai, China

Jiandong Ding
dingjiandong2@huawei.com
Huawei Technologies
Shanghai, China

Abstract

Optimizing multiple conversion objectives is a core challenge in industrial recommendation, often limited by signal erosion in rigid architectures. Existing Multi-Task Learning (MTL) methods typically enforce uniform dependency strengths across a static conversion funnel, overlooking how task correlations naturally vary based on item characteristics. Hierarchical message passing along these fixed chains leads to cumulative signal attenuation, which degrades performance on sparse, deep-funnel objectives. To address this, we propose the Personalized Task Dependency Graphs (PTDG). While respecting necessary physical causal constraints (e.g., *Click* $\rightarrow$ *Pay*), PTDG dynamically “rewires” the *intensity* of dependency pathways for each item via low-rank approximation to ensure structural robustness. We implement a GCN-based propagation with hard causal masking to establish adaptive information shortcuts. Additionally, we introduce an Adaptive Progressive Masking (APM) strategy that decouples shared parameters according to task sparsity, helping to stabilize optimization. Experiments on KuaiRand1K and an industrial dataset show that PTDG significantly improves AUC on sparse conversion tasks by up to 1.45%, while maintaining comparable performance on dense objectives. Online A/B testing shows PTDG

improves Conversion Rate (CVR) by 1.2% and effective Cost Per Mille (eCPM) by 1.9% relative to the baseline.

CCS Concepts

• Information systems $\rightarrow$ Recommender systems.

Keywords

Recommender Systems, Multi-Task Learning, Task Dependency, Graph Convolutional Networks

ACM Reference Format:

Fuyuan Liu, Tiandeng Wu, Yaqun Fang, Wei Zhou, Zehao Zhou, Wenping Chen, Qishun Mei, Jiaxin Zhou, Heng Chang, Yi Cao, and Jiandong Ding. 2026. Personalized Task Dependency Graphs for Mitigating Signal Erosion in Multi-Task Recommendation. In *Proceedings of the 35th ACM International Conference on Information and Knowledge Management (CIKM '26)*, November 07–11, 2026, Rome, Italy. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3799682.3839876>

1 Introduction

In modern industrial recommender systems, the focus has shifted from maximizing single interaction metrics to optimizing the overall effective Cost Per Mille (eCPM). This requires simultaneously predicting multiple heterogeneous user behaviors, ranging from shallow interactions (e.g., *Click*) to deep conversion objectives (e.g., *Payment*). Multi-task learning (MTL) enables efficient multi-target prediction in recommendation by sharing knowledge across tasks, which improves performance and lowers computational cost. While MTL architectures like MMoE [7] and AITM [14] are standard for representation learning, they typically rely on a rigid *inductive bias*: assuming universal dependency strengths within the conversion funnel across all traffic.

*Both authors contributed equally to this research (co-first authors).

[†]Corresponding author.



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

CIKM '26, Rome, Italy

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2539-5/2026/11

<https://doi.org/10.1145/3799682.3839876>

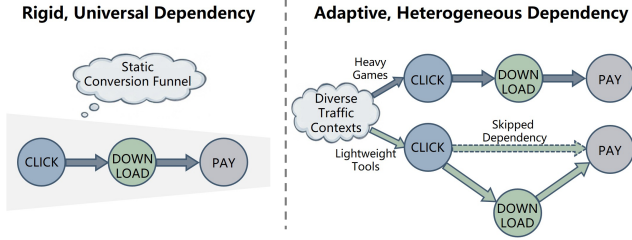


Figure 1: Contrast between rigid funnels (left) with uniform dependency and our personalized dependency graph (right) where edge intensities adapt per instance.

However, this assumption overlooks the *structural heterogeneity* inherent in diverse industrial applications, that is, the systematic differences in dependency patterns across conversion tasks. While the physical causal order (e.g., *Click* precedes *Payment*) is necessary, the *relevance intensity* between these nodes varies significantly. For instance, heavy games follow a strong causal path (*Click* → *Download* → *Pay*), whereas lightweight tools often exhibit weaker dependencies where users skip intermediate steps (*Click* → *Pay*). Enforcing a monolithic topology on such divergent instances limits model expressiveness. Although recent approaches like PMTRec [15] attempt to personalize optimization weights, they fail to account for these explicit structural variations between task nodes. Furthermore, hierarchical message passing along fixed chains often leads to *cumulative signal attenuation* in deep dependency paths, which significantly degrades signals for sparse downstream tasks. To address these limitations, we propose Personalized Task Dependency Graph (PTDG), which treats task dependency as a dynamic, learnable topology. Our main contributions are as follows:

- **Dynamic Topology Learning:** PTDG utilizes a low-rank graph generation mechanism to adaptively construct instance-specific dependency intensities, effectively modeling heterogeneous correlations within causal constraints.
- **Controllable Propagation:** To mitigate signal erosion, we design a GCN-based message passing module with hard causal masking, which establishes adaptive shortcuts based on relevance.
- **Adaptive Optimization:** We introduce an Adaptive Progressive Masking (APM) strategy. Unlike numerical optimization methods, APM decouples parameters based on label density to resolve gradient conflicts. This approach structurally resolves the gradient conflict while reducing computational complexity.

2 Related Work

Research on Multi-Task Learning (MTL) for recommendation has primarily focused on mitigating negative transfer. Architectures such as MMoE [7], PLE [10], and AITM [14] employ gating networks to disentangle shared and specific representations. Recent work PaDiRec[3] employs a diffusion process-based adapter to generate corresponding model parameters conditioned on dynamic task preference weights. Optimization strategies such as GradNorm[2], PMTRec [15] and MoCoGrad [1] address gradient conflicts through

re-weighting and momentum calibration, respectively. However, these methods predominantly optimize the numerical loss landscape, treating tasks as parallel objectives or using globally-shared dependency patterns, thereby neglecting the structural heterogeneity across different industrial applications.

Given the sequential nature of user actions, modeling task dependencies has become essential. ESMM [8] and ESCM2 [12] address sample selection bias by modeling conversion rate over the entire space. AITM [14] introduces an information transfer module along a fixed conversion chain. HTLNet [11] investigates hierarchical cascading paths but enforces fixed level assignments, limiting its ability to capture instance-level variation. CSRL [17] attempts to learn causal structures, but its iterative search incurs prohibitive costs ($O(T^2)$ or higher), rendering it incompatible with industrial latency constraints. MIT [18] explores cross-task correlations but relies on globally-shared task relationships. In contrast, PTDG learns instance-specific, low-rank dependency graphs via efficient GCNs [5, 6, 13], balancing structural flexibility with the latency constraints of industrial systems.

3 Method

As illustrated in Figure 2, the architecture consists of three main components: (1) Personalized Topology Learning, which constructs task-to-task dependency graphs using low-rank factorization; (2) Adaptive Progressive Masking (APM), which structurally decouples parameter sharing to mitigate gradient conflicts; and (3) Causally Optimized Message Passing, which employs GCNs on hard causal masked dependency graphs with shortcut connections to prevent semantic backflow (e.g., *Pay* → *Click*) and signal erosion.

3.1 Problem Formulation

We formulate multi-objective conversion prediction as a supervised learning task. Let $\mathcal{D} = \{(d_1, Y_1), \dots, (d_N, Y_N)\}$ denote the training dataset, where each data point d_n is a triplet (f_i, f_u, f_s) representing item, user, and scenario features. The objective is to learn a mapping function $\Phi : \mathcal{X} \rightarrow \mathcal{Y}$, which estimates the probabilities $Y_n = \{y_n^{(1)}, y_n^{(2)}, \dots, y_n^{(T)}\}$ for T distinct conversion goals.

3.2 Personalized Task Dependency Graph

Unlike approaches that enforce static dependency strengths, PTDG learns an item-specific graph structure within physical constraints.

Item-Specific Dependency Graph. To prevent overfitting on noise-prone industrial interactions and reduce parameter complexity, we adopt the low-rank approximation. The item-specific adjacency matrix A_{item} is defined as:

$$A_{item} = Z_i^L (Z_i^R)^\top \quad (1)$$

where $Z_i^L, Z_i^R \in \mathbb{R}^{T \times q}$ ($q < \frac{T}{2}$) are latent factors derived from the i -th item embeddings (extracted by two backbone models such as DeepFM [4] or MLP).

Personalized Bias Injection. To capture context-specific dependencies, we dynamically modulate the graph structure. Specifically, we generate a personalization bias vector $\gamma_{u,s} \in \mathbb{R}^q$ from features of user u and scenario s using an MLP. This bias is then broadcast across all T tasks to modulate the item-specific latent

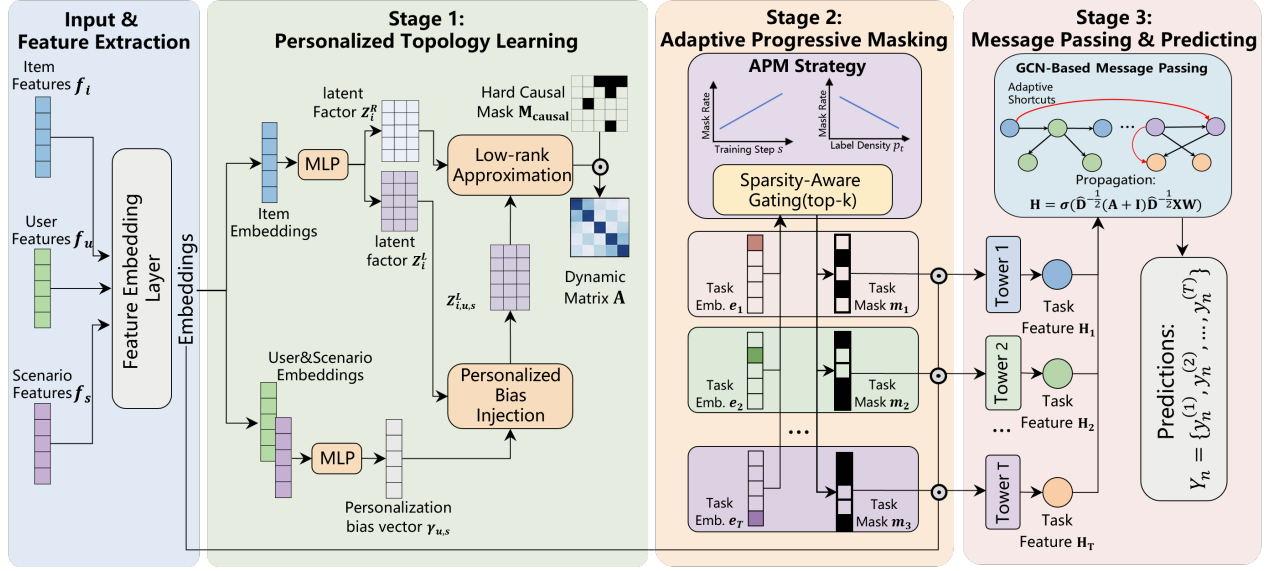


Figure 2: Overall structure of PTDG. The approach can be divided into three stages: personalized topology learning, adaptive progressive masking and message passing & predicting.

Table 1: Statistics of Experiment Datasets.

Dataset	Samples	Tasks	Positive Ratio (range)
KuaiRand1K	8.4M	6	0.02% ~ 37.9%
Industrial	13.2M	7	(Confidential)

factors:

$$\mathbf{A}_{dyn} = \mathbf{Z}_{i,u,s}^L (\mathbf{Z}_i^R)^\top, \quad \mathbf{Z}_{i,u,s}^L = \mathbf{Z}_i^L \odot (\mathbf{1}_T \otimes \gamma_{u,s}^\top) \quad (2)$$

where $\mathbf{A}_{dyn}$ represents the personalized dynamic dependency matrix, $\mathbf{1}_T$ is an all-ones column vector, and $\otimes$ denotes the outer product. This mechanism provides a lightweight way to capture instance-level diversity while alleviating the overfitting problem. **Causal Pruning & Message Passing.** We apply a hard causal mask $\mathbf{M}_{causal}$ to enforce behavior order constraints and avoid semantic backflow, followed by GCN-based propagation which operates on the original task feature matrix $\mathbf{X}$:

$$\mathbf{A} = \mathbf{A}_{dyn} \odot \mathbf{M}_{causal} \quad (3)$$

$$\mathbf{H} = \sigma \left(\hat{\mathbf{D}}^{-\frac{1}{2}} (\mathbf{A} + \mathbf{I}) \hat{\mathbf{D}}^{-\frac{1}{2}} \mathbf{X} \mathbf{W} \right) \quad (4)$$

where $\mathbf{H}$ represents the extracted task node feature matrix, which is used for the final CVR prediction. $\mathbf{I} \in \mathbb{R}^{T \times T}$ is an identity matrix, $\hat{\mathbf{D}}$ represents the degree matrix of $(\mathbf{A} + \mathbf{I})$, and $\mathbf{W}$ is a learnable parameter matrix. In contrast to static sequential chains, personalized dependency graph creates adaptive shortcuts between tasks without causal relation, which prevents the signal erosion caused by redundant intermediate steps in fixed hierarchies.

We employ the extracted task node feature matrix to predict the conversion probability $\hat{y}_t$ of the user for the target t :

$$\hat{y}_t = \text{Sigmoid}(\text{MLP}(h_t)) \quad (5)$$

where h_t represents the feature of node t for task t in the extracted task node feature matrix $\mathbf{H}$.

3.3 Optimization Objective

The overall training objective $\mathcal{L}$ of PTDG is a weighted sum of per-task binary cross-entropy losses:

$$\mathcal{L} = \sum_{t=1}^T \beta_{t,s} \cdot \mathcal{L}_{t,s}, \quad \mathcal{L}_{t,s} = \text{BCE}(y_t, \hat{y}_{t,s}) \quad (6)$$

$$\hat{\mathcal{L}}_{t,s} = \alpha \mathcal{L}_{t,s} + (1 - \alpha) \hat{\mathcal{L}}_{t,s-1}, \quad \beta_{t,s} = \frac{1}{\hat{\mathcal{L}}_{t,s}} \quad (7)$$

where $\beta_{t,s}$ is the weight of task t at training step s , and $\hat{y}_{t,s}$ is the predicted probability for task t at step s . We obtain the task loss weights $\beta_{t,s}$ by dividing each task loss by their exponential moving average (EMA), bringing them all to a unified scale to stabilize training.

3.4 Adaptive Progressive Masking (APM)

Information transfer between multiple tasks may exacerbate the gradient conflict problem inherent in multi-task models. To address this, we propose a structural masking strategy that learns a task-specific mask $\mathbf{m}_t$ to select shared parameters, which is computed using the task embedding e_t we add for each task.

Sparsity-Aware Mask. A binary mask $\mathbf{m}_t$ is generated by retaining only the top- k_t elements of a learned gate:

$$\mathbf{m}_t^{(j)} = \mathbb{I} \left(\text{rank}(\mathbf{w}_t^{(j)}) \leq k_t \right), \quad \mathbf{w}_t = \text{Gate}_t(e_t) \quad (8)$$

where $\mathbf{m}_t^{(j)}$ represents the j -th parameter of mask $\mathbf{m}_t$. The mask is applied to the parameters shared between tasks through the Hadamard product to obtain the parameters visible to task t . **Adaptive Progressive Masking.** The parameter mask rate $r_{target}^{(t)}$ is

Table 2: Performance comparison in terms of AUC on KuaiRand dataset and industrial dataset. Values are means over 20 different random seeds. Bold indicates the best, underline the second best.

Dataset	Task	MMoE	PLE	STEM	MCGrad	PMTR	MIT	PTDG
KuaiRand1K	Click	0.8706	0.8637	0.8732	0.8743	<u>0.8772</u>	0.8766	0.8801
	Like	0.8879	0.8903	0.8998	0.9009	0.9017	<u>0.9028</u>	0.9085
	Follow	0.9033	0.9024	0.9073	<u>0.9130</u>	0.9124	0.9129	0.9193
	Comment	0.9051	0.8986	0.9081	0.9072	0.9085	<u>0.9101</u>	0.9172
	Forward	0.8984	0.8933	0.9028	0.9011	<u>0.9042</u>	0.9038	0.9097
	Hate	0.9041	0.8960	0.9055	0.9066	0.9072	<u>0.9085</u>	0.9139
	Avg.	<u>0.8949</u>	<u>0.8907</u>	<u>0.8995</u>	<u>0.9005</u>	<u>0.9019</u>	<u>0.9025</u>	0.9081
Industrial	Task 1	0.8410	0.8424	0.8494	<u>0.8501</u>	0.8481	0.8471	0.8527
	Task 2	0.8395	0.8433	0.8457	0.8462	<u>0.8500</u>	0.8468	0.8511
	Task 3	0.8601	0.8379	0.8482	0.8550	<u>0.8745</u>	0.8541	0.8872
	Task 4	0.9417	0.9471	0.9457	<u>0.9468</u>	0.9462	0.9452	0.9464
	Task 5	0.8699	0.8678	0.8613	0.8630	<u>0.8729</u>	0.8732	0.8751
	Task 6	0.8837	0.8796	0.8557	0.8610	0.8843	<u>0.8899</u>	0.8993
	Task 7	0.8912	0.8974	0.8915	0.8952	0.8967	<u>0.8979</u>	0.9028
	Avg	<u>0.8753</u>	<u>0.8736</u>	<u>0.8710</u>	<u>0.8739</u>	<u>0.8818</u>	0.8791	0.8878

determined by the label positive rate p_t of task t to balance easy (dense) and hard (sparse) tasks:

$$r_{target}^{(t)} = r_{min} + (r_{max} - r_{min}) \cdot \left(1 - \frac{p_t - p_{min}}{p_{max} - p_{min}}\right) \quad (9)$$

where p_{min} and p_{max} represent the minimum and maximum label positive rate, r_{max} and r_{min} are hyper parameters. We also implement a linear warm-up mechanism with max warm-up step S_{warm} to adjust the learning difficulty gradually based on the training step s and the number of shared parameters d to stabilize training:

$$k_{t,s} = d - d \cdot \min\left(r_{target}^{(t)}, \frac{s}{S_{warm}} \cdot r_{target}^{(t)}\right) \quad (10)$$

3.5 Complexity Analysis

PTDG uses low-rank factorization ($O(T \times q)$) as structural regularization rather than directly learning a full adjacency matrix, which prevents overfitting to spurious correlations on sparse industrial data. The low-rank formulation ensures both adjacency calculation and GCN-based message passing remain linear in T . In production, PTDG increases end-to-end latency by only 8 ms over the MMoE baseline, within serving SLA. Unlike gradient-based de-conflicting methods (e.g., PCGrad [16] with $O(T^2)$ projections), APM introduces negligible training cost and no inference overhead as masks are fixed after training.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate PTDG on KuaiRand1K (6 tasks) and an industrial dataset (7 tasks). Details are provided in Table 1.

Baselines. We compare PTDG with strong industrial baselines and state-of-the-art methods: (1) MMoE [7]: A widely adopted multi-gate mixture-of-experts architecture. (2) PLE [10]: A widely used industrial standard for parameter sharing. (3) STEM [9]: An advanced optimization-based MTL framework. (4) MoCoGrad [1]: A method addressing gradient conflicts via momentum calibration. (5)

Table 3: Ablation Study on Industrial Dataset.

Exp.	Variant Model	Avg AUC
1	PTDG (Full)	0.8878
2	w/o Personalization ($\gamma_{u,s}$)	0.8771
3	w/o GCN (use Concat)	0.8796
4	w/o Causal Mask	0.8814
5	w/o APM (Adaptive Masking)	0.8826
6	w/o Progressive Warmup	0.8841

PMTRec [15]: A personalized multi-task approach using gradient re-weighting. (6) MIT [18]: A recent model focusing on cross-task heterogeneous correlation modeling.

Implementation Details. We use Adam as our optimizer with a learning rate of $1e-3$, a dropout rate of 0.1, and a batch size of 2048. In all experiments, we repeat each setting 20 times using 20 different random seeds. The rank q of item-specific latent factors is set to 2. The parameter α in the EMA-based multi-task loss weight optimization is set to 0.1. The maximum and minimum mask rates r_{max} and r_{min} of APM were set to 0.7 and 0.2, respectively. The warm-up process in APM was conducted with $p_{min} = 500$ and $p_{max} = 2000$. All baseline method parameters were kept consistent with those in the original paper. All experiments were conducted on a single NVIDIA Tesla V100 GPU.

4.2 Performance Comparison (RQ1)

Comparison with SOTAs. We conducted 20 runs using 20 different random seeds. As shown in Table 2, PTDG achieves the highest average AUC on both datasets. PTDG surpasses MIT [18] by 1.05% on Task 6—the key deep conversion task—because, unlike MIT’s rigid global correlations, it adapts task relationships per instance, enabling GCN-based message passing that preserves signal strength across deeper tasks. PTDG also beats MoCoGrad [1] by 1.59% on average. Unlike MoCoGrad’s post-hoc momentum tuning, our structural APM directly resolves gradient conflicts at the parameter level, yielding more effective and efficient multi-task optimization.

Performance on Deep Tasks. A critical observation is PTDG’s strong performance on sparse, deep-conversion objectives. On dense, shallow tasks (e.g., Task 4, click), PTDG achieves comparable but not significantly better performance, which is expected as signal erosion primarily affects sparse deep-funnel objectives. On harder tasks, however, PTDG significantly outperforms state-of-the-art methods. For instance, on the sparse Industrial Task 3, PTDG surpasses PMTRec by 1.45%. Similarly, on the follow task in KuaiRand1K, it exceeds our online baseline model MMoE by 1.77%. This confirms that our personalized graph topology effectively mitigates signal erosion for deep nodes without compromising head-task performance.

4.3 Ablation Study (RQ2)

To verify the contribution of each component, we conduct an ablation study on the Industrial Dataset (Table 3). Removing instance-level dependency graph decreases Avg AUC by 1.21%, confirming

that personalizing task dependencies is effective for capturing heterogeneous multi-task relationships. Replacing the GCN with simple concatenation lowers AUC by 0.93%, verifying that GCN-based message passing effectively preserves signal strength across tasks compared to passing through the static conversion funnel. Disabling adaptive masking (w/o APM) reduces performance by 0.59%, underscoring its effectiveness in mitigating gradient conflicts.

4.4 Sensitivity Analysis

We vary the low-rank dimension q from 1 to 7, performance peaks at $q = 2$ and degrades with larger q , confirming that low-rank approximation effectively controls overfitting while using fewer parameters than a full adjacency matrix. Empirically, $q \approx T/3$ works well across our settings. Formal scaling analysis is left to future work. We also conduct masking rate experiments on tasks with maximum and minimum masking rates ranging from 0 to 0.9 in the APM strategy. The experimental results show that the model achieves the best performance when set to $r_{min} = 0.2$ for high positive sample rate tasks and $r_{max} = 0.7$ for low positive sample rate tasks.

4.5 Online A/B Testing

We deployed PTDG as the ranking model of a mainstream app distribution platform, serving over 100 million daily active users, for a two-week online A/B test. The baseline model was the currently deployed MME. This is the same MME architecture included in our offline comparison (Table 2). Traffic was split into two buckets: 10% for the base bucket and 10% for the experimental bucket. The remaining 80% of traffic continued to use the original production model to ensure service stability. To ensure statistical significance, we computed daily aggregated metrics for each bucket and performed two-sample t-tests with a significance threshold of $p < 0.05$. Results show that PTDG achieved statistically significant improvements on core business metrics, with a +1.2% lift in CVR and +1.9% lift in eCPM. This demonstrates the industrial viability of PTDG as a multi-task learning framework that delivers consistent gains in real-world recommender systems.

5 Conclusion

This work addresses signal erosion in industrial multi-task recommendation through the proposed PTDG framework. Unlike static funnel-based approaches, PTDG employs a low-rank dynamic graph learning method to model instance-level dependencies under strict latency constraints. By combining causal message passing with structural parameter decoupling, the model mitigates signal attenuation for deep-conversion objectives. Experiments demonstrate that PTDG achieves a favorable trade-off, improving performance on sparse, high-value tasks (up to 1.45% lift) while maintaining accuracy on dense interactions. Future work will explore scaling to larger task sets and extending the framework to cross-domain settings.

GenAI Usage Disclosure

Generative AI tools assisted with language editing and code debugging. All experimental results were produced by author-written scripts and verified against saved artifacts.

References

- [1] Heyan Chai, Zeyu Liu, Yongxin Tong, Ziyi Yao, Binxing Fang, and Qing Liao. 2024. Towards Task-Conflicts Momentum-Calibrated Approach for Multi-task Learning. In *40th IEEE International Conference on Data Engineering (ICDE)*. 939–951.
- [2] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*. 794–803.
- [3] Yifan Gao, Peilin Zhou, Mikko Laukkanen, and Yong Li. 2024. PaDiRec: Parameter Diffusion for Controllable Multi-Task Recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 123–132.
- [4] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*. 1725–1731.
- [5] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongjin Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 639–648.
- [6] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.
- [7] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H. Chi. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. 1930–1939.
- [8] Xiao Ma, Lu Zhao, Guan Huang, Zhi Wang, Zelan He, Yong Zhu, Kai Lian, Xiaoyi Liu, and Tianyang Xie. 2018. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR)*. 1137–1140.
- [9] Liangcai Su, Junwei Pan, Ximei Wang, Xi Xiao, Shijie Quan, Xihua Chen, and Jie Jiang. 2023. STEM: Unleashing the Power of Embeddings for Multi-Task Recommendation. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI)*. 11515–11523.
- [10] Hongyan Tang, Junrui Liu, Ming Zhao, and Xiuqi Gong. 2020. Progressive layered extraction (PLE): A novel multi-task learning (MTL) model for personalized recommendations. In *Proceedings of the 14th ACM Conference on Recommender Systems (RecSys)*. 269–278.
- [11] Xing Tang, Yang Qiao, Fuyuan Lyu, Dugang Liu, and Xiuqiang He. 2024. Touch the Core: Exploring Task Dependence Among Hybrid Targets for Recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems (RecSys)*. 329–339.
- [12] Hao Wang, Tai-Wei Chang, Tianqiao Liu, Jianmin Huang, Zhichao Chen, Chao Yu, Ruopeng Li, and Wei Chu. 2022. ESCM²: Entire Space Counterfactual Multi-Task Model for Post-Click Conversion Rate Estimation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 2360–2365.
- [13] Tiandeng Wu, Qijiong Liu, Yi Cao, Yao Huang, Xiao-Ming Wu, and Jiandong Ding. 2023. Continual graph convolutional network for text classification. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*.
- [14] Dongbo Xi, Fuzhen Zhu, Qian Wang, Yulong Chen, Fumin Chen, Yichao Guo, Zhi Li, Qing Wu, et al. 2021. Modeling the sequential dependence among audience multi-step conversions with multi-task learning. In *Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. 3745–3755.
- [15] Liangwei Yang, Zhiwei Liu, Jianguo Zhang, Rithesh Murthy, Shelby Heinecke, Huan Wang, Caiming Xiong, and Philip S. Yu. 2024. Personalized Multi-task Training for Recommender System. In *Proceedings of the ACM Web Conference 2024 (WWW)*. 2136–2146.
- [16] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient Surgery for Multi-Task Learning. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*. 5704–5716.
- [17] Peng Zhang, Yancheng Wang, Qingyao Sun, et al. 2023. CSRL: Causal Structure Learning for Multi-Task Recommendation. In *Proceedings of the ACM Web Conference 2023 (WWW)*. 3662–3672.
- [18] Rui Zhou, Hao Wang, Wei Guo, Qinglin Jia, Wenjia Xie, Xiang Xu, Yong Liu, Defu Lian, and Enhong Chen. 2025. MIT: A Multi-Tower Information Transfer Framework Based on Hierarchical Task Relationship Modeling. In *Companion Proceedings of the ACM on Web Conference 2025*. 651–660.