Prototype-Guided Representation Projection for Multi-Domain Multi-Task Recommendation

Binrui Wu*
Fudan University
Shanghai, China
23210180068@m.fudan.edu.cn

Haochen Sui*
University of Michigan
Ann Arbor, USA
hcsui@umich.edu

Jiaye Lin*
Tsinghua University
Beijing, China
linjiaye2222@gmail.com

Jiechao Gao†
Center for SDGC, Stanford University
CA, USA
jiechao@stanford.edu

Ting Xu
University of Massachusetts Boston
Boston, USA
ting.xu001@umb.edu

Keyan Jin
Macao Polytechnic University
Macao, China
p2317001@mpu.edu.mo

Xuesong Zhang
KuaiShou Technology
Beijing, China
zhangxuesong@kuaishou.com

Abstract

Multi-domain and multi-task learning enhance the efficiency and performance of industrial recommendation systems by integrating information from different domains/tasks to model user interests uniformly. However, existing methods suffer from the problem of representation entanglement, which limits the effective handling of commonality and specificity among various domains/tasks. In this paper, we propose a Prototype-guided Representation Projection (PRP) model to address this issue, which explores a novel direction of applying prototype learning to deal with complex domain/task relationships in the recommendation field. To identify inter-domain/task commonality, PRP initially uses a shared Mixture of Experts (MoE) architecture to learn representations for each sample, projecting them into a common prototype space across all domains/tasks. For domain/task specificity, specific feature extraction experts are employed, and sample representations are projected to the corresponding prototype spaces, constrained by an orthogonal loss to ensure the independence of those spaces. Moreover, PRP utilizes Optimal Transport (OT) to guide the correct representation projection within the prototype spaces, employing the linear combination of prototypes as the new sample representation. We conduct offline experiments on two open-source datasets and deploy our approach in an online system for A/B testing. Extensive experimental results consistently demonstrate that our approach outperforms existing methods.

CCS Concepts

• **Information systems → Recommender systems**.

*Equal contributions.
†Corresponding Author.

MM '25, Dublin, Ireland

ACM ISBN 979-8-4007-2035-2/2025/10
https://doi.org/10.1145/3746027.3755224

Keywords

Multi-Domain Recommendation, Multi-Task Recommendation

ACM Reference Format:
Binrui Wu, Haochen Sui, Jiaye Lin, Jiechao Gao, Ting Xu, Keyan Jin, and Xuesong Zhang. 2025. Prototype-Guided Representation Projection for Multi-Domain Multi-Task Recommendation. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25), October 27–31, 2025, Dublin, Ireland*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3746027.3755224

1 Introduction

With the rapid growth of Web 2.0 and user-generated data, recommendation systems have become a key tool to overcome information overload and enhance decision-making quality by providing personalized services to users aligned with their interests [5, 18, 19, 33, 37, 38]. These systems are typically designed for specific domains due to domain-specific data distribution and the corresponding feature space [45]. However, in mainstream Internet applications like KuaiShou[1], TikTok[2], and TaoBao[3], recommendations under various domains exist (e.g., homepage feeds and city-based feeds), each characterized by distinct data distributions [34, 42]. Maintaining a single model for each domain not only neglects the overarching user behavior and shared knowledge across different domains but also increases the cost and maintenance burden for the company [43]. Furthermore, even within a single domain, different interaction behaviors with items, such as "click" and "purchase", may not always be positively correlated [32]. Therefore, it is particularly important to develop a unified Multi-Domain Multi-Task Recommendation (MDMTR) [3, 16, 30, 46] system to model and leverage relationships between different domains/tasks.

Recently, significant progress has been made separately in the fields of Multi-Domain Recommendation (MDR) [10, 21, 25, 28] and Multi-Task Recommendation (MTR) [2, 24, 32]. However, real-world applications often require a combination of both, posing new challenges for information transfer and target balancing due to

[1]https://www.kuaishou.com
[2]https://www.tiktok.com
[3]https://www.taobao.com

domain-task interplay. While some existing studies [3, 16, 28] have explored the integration of MDR and MTR, they primarily focus on the Mixture of Experts (MoE) [24] architecture. However, these methods suffer from the problem of representation entanglement, where the learned representations across different domains/tasks are entangled because they project all information into the same feature space, hindering the effective capture of complex relationships in multi-domain and multi-task setups.

To this end, we propose a Prototype-based Representation Projection (PRP) model that incorporates prototype learning into Multi-Domain Multi-Task Recommendation (MDMTR) to effectively address representation entanglement and handle complex domain/task relationships, i.e., commonality and specificity. Prototype learning, which has been widely utilized and successful in computer vision [40, 47] and few-shot learning [15, 29], forms the foundation of our approach. The core concept of PRP involves mapping each sample's hidden representation to different prototype spaces to learn the common and specific features of domains/tasks separately. Specifically, PRP consists of three parts: (i) To identify inter-domain/task commonality, PRP first learns each sample representation using the domain/task-shared MoE. We then construct a shared prototype space, which consists of a set of vectors. Sample representations are projected into this space, and a linear combination of prototypes is used as the sample's common feature representation. (ii) To identify domain/task specificity, PRP creates independent feature extractors for each domain/task and assigns a separate prototype space to each, projecting sample representations into their respective prototype spaces. To prevent the convergence of different prototype spaces, which is undesirable, we introduce an orthogonal loss to maintain the independence of these spaces. (iii) Finally, to learn optimal projections, PRP employs Optimal Transport (OT) [6, 26] to guide the projection of sample representations into prototype spaces, using linear combinations of prototypes as representations of commonality or specificity.

For MDMTR, we construct corresponding PRP models for domains and tasks, namely D-PRP and T-PRP, respectively. The D-PRP model takes sample features and domain ID as inputs, while output vectors represent both common and specific representations of domains. Subsequently, the T-PRP model receives the outputs from D-PRP and generates common representations across all tasks as well as specific representations for each task. Finally, these representations are combined to predict the target for different tasks. To summarize, we make the following contributions in this paper:

- We propose to use prototype learning to handle the complex inter-domain/task relationships in Multi-Domain Multi-Task Recommendation (MDMTR). *To the best of our knowledge, this is the first application of prototype learning in MDMTR.*
- We introduce a Prototype-guided Representation Projection (PRP) model, utilizing a prototype-based Optimal Transport (OT) module to project sample representations into prototype spaces to learn commonality and specificity, respectively.
- We conduct extensive experiments on two offline datasets and implement our approach in a video platform for online A/B testing. Experimental results demonstrate the effectiveness of the PRP model in MDMTR for real-world applications.

2 Related Works

2.1 Multi-Domain Recommendation

There has been growing interest in MDR due to the capability to transfer knowledge across various business domains [7, 14]. Approaches like MARIA [35] capture domain information through feature scaling modules and dynamically use the MoE structure to analyze signals. HMoE [20] projects multi-domain data into the same feature space and optimizes using MTR methods. CausalInt [36] employs disentangled representation learning to extract domain-specific information and integrates a TransNet to leverage data from various domains. In the field of domain modeling, STAR [28] introduces a comprehensive model that integrates both domain-specific and domain-shared components, adeptly capturing both unique and common information to boost performance. Unlike MDR methods, our PRP leverages prototype learning to simultaneously capture multi-domain and multi-task aspects.

2.2 Multi-Task Recommendation

The core idea of MTR frameworks is to combine different but related tasks, improving overall model performance by sharing knowledge and information [22, 24, 32]. These methods can be divided into architecture-based approaches [2, 24] and optimization-based approaches [27]. Shared Bottom [2] is one of the earliest architecture-based methods, enhancing performance by sharing hard structures between tasks. MoE [13] proposed sharing experts at the base level and combining them via a gating network. MMoE [24] extends this by using task-specific gates to achieve varying fusion weights. PLE [32] introduces task-specific experts alongside shared ones. SNR [23] divides shared layers into sub-networks and learns their connections using latent variables. AdaTT [17] adds a branch that linearly integrates task-specific experts for joint task modeling. Another research direction focuses on enhancing MTR optimization. To address negative transfer, GradNorm [4] adjusts gradient magnitudes for balanced training rates. Gradient surgery [41] resolves conflicting gradients by projecting them onto each other's normal plane. MetaBalance [12] flexibly balances gradient magnitude proximity between auxiliary and target tasks. AdaTask [39] tackles gradient conflicts using task-specific optimization. Compared with existing MTR methods, we comprehensively consider complex domain relationships, including commonality and specificity, to further enhance the optimization of multiple objectives.

2.3 Multi-Domain Multi-Task Recommendation

Recently, a trend is emerging that focuses on MDMTR [3, 16, 42, 44], aiming to simultaneously harness the advantages of both domains and tasks. M2M [42] uses meta-learning to extract domain-specific information by applying dynamic weights that take into account inter-domain correlations. PEPNet [3] employs personalized prior information and gating mechanisms to dynamically scale units, enabling personalized adaptation across multiple domains and tasks. M3Rec [16] introduces a meta-item-embedding generator and a user-preference transformer to unify user and item representations, which effectively captures non-i.i.d. behaviors across different domains. M3oE [44] utilizes three MoE modules to disentangle and

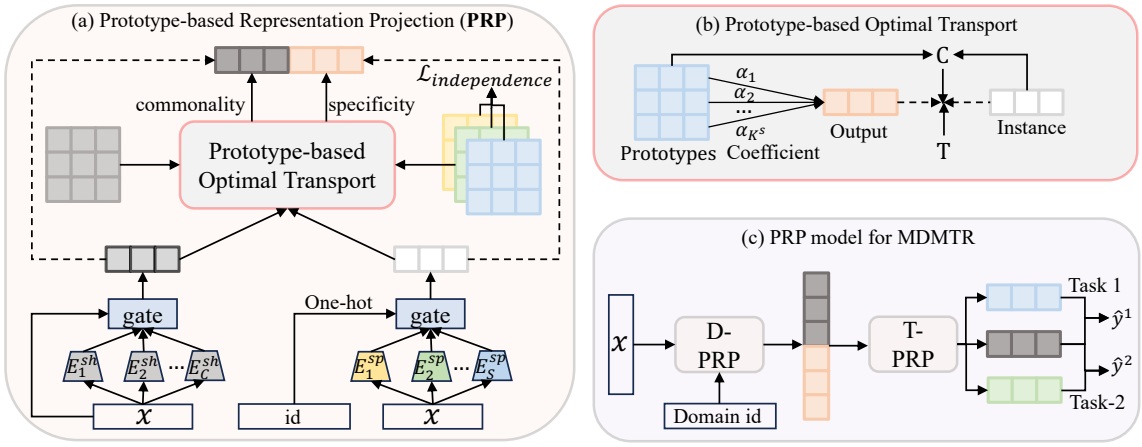


Figure 1: The illustration of our method, which mainly consists of three modules: (a) The Prototype-based Representation Projection (PRP) model diagram, which learns commonality and specificity representations from input x. (b) The Prototype-based Optimal Transport (POT) diagram, illustrating how representations are projected to corresponding spaces with Optimal Transport (OT). (c) PRP model for MDMTR, depicting the use of D-PRP for multi-domains and T-PRP for multi-tasks.

learn user preferences across multiple domains and tasks, incorporating a two-level fusion mechanism and AutoML for dynamic structure optimization. However, existing methods typically project information into the same feature space, failing to fully capture the complex relationships in an MDMTR setup. In contrast, our framework incorporates prototype learning to better address representation entanglement and handle complex domain/task relationships.

3 Method

3.1 Problem Definition

MDMTR seeks to optimize T recommendation tasks across S domains simultaneously. Each sample i is represented by the tuple (x_i, s, y_i^t), where x_i includes a set of comprehensive features, e.g., user information, item characteristics, and contextual data. The variable $s \in \{1, \ldots, S\}$ denotes the domain to which the sample x_i belongs. The label y_i^t is a binary indicator for task $t \in \{1, \ldots, T\}$, where $y_i^t \in \{0, 1\}$ specifies whether the user associated with sample i provides positive feedback for task t within domain s. The goal is to learn a function $\hat{y}_i^t = f^t(x_i, s; \Theta)$ for each task t and domain s, aiming to accurately predict the binary label y_i^t based on the input features, where Θ represents the trainable parameter of f.

3.2 Representation Extraction Layer

Many previous methods based on the MoE architecture [24] project all information into a single feature space, often failing to adequately capture the complex relationships inherent in MDMTR. To address this limitation, our proposed PRP introduces two distinct types of representation extraction layers designed to both identify inter-domain/task commonality and extract domain/task-specific characteristics. These layers form the foundation for enhancing performance in the higher layers of the model. We refer to them as the shared representation extraction layer and the specific representation extraction layer.

3.2.1 ***Shared Representation Extraction Layer.*** Given the overlapping user interactions and items, shared information exists among data from multiple domains and tasks. Inspired by the MoE architecture [8, 13], we design a shared representation extraction layer. Specifically, the output of this layer is formulated as:

$$J(x) = \sum_{c=1}^{C} g_c^{sh}(x) E_c^{sh}(x), \tag{1}$$

where x represents the input, E_c^{sh} denotes the c-th expert network, composed of a Multi-Layer Perceptron (MLP) with ReLU activation function, and C is the number of experts. The term $g_c^{sh}(x)$ denotes the output of the gating network for c-th expert, which is a simple linear transformation of x followed by a Softmax layer:

$$g_c^{sh}(x) = \text{Softmax}(xW_c^{sh}), \tag{2}$$

where W_c^{sh} is the trainable parameter for c-th expert.

3.2.2 ***Specific Representation Extraction Layer.*** In addition to extracting specific information, we continue to utilize the MoE architecture [8, 13], dividing specific information into domain-specific and task-specific with different extraction outputs.

For domain-specific information, since each sample belongs to only one domain, we establish a separate expert network for each domain and adopt a one-hot vector to set a gate to ensure that only the expert corresponding to the domain of the current sample is activated during each forward pass. The output $H(\cdot)$ of the specific representation extraction layer is formulated as follows:

$$H(x, \text{id}) = \sum_{j=1}^{S} \mathbb{I}(\text{id} == j) E_j^{sp}(x), \tag{3}$$

where E_j^{sp} denotes the j-th domain-specific expert network consisting of an MLP with ReLU activation function, and S is the number of domains. Here, $\mathbb{I}(\text{id} == j)$ is an indicator function that activates the j-th expert if the domain index of the current input x matches j. For domain-specific information extraction, id refers to the domain

identity s of the sample x_i. For task-specific information, the difference lies in the need to predict T tasks simultaneously for each sample. In other words, each task t needs to output a task-specific feature representation $H^t(x) = E_t^{sp}(x)$.

3.3 Prototype-Based Representation Alignment

Inspired by prototype learning [1, 31, 40], a critical step in PRP is mapping the hidden representation of each sample into distinct prototype spaces. This allows for the independent learning of common and specific features across domains and tasks after obtaining the shared representations $J(x)$ and specific representations $H(x)$.

3.3.1 ***Commonality Representation.*** Assuming that after passing through the shared representation extraction layer, we obtain a vector $J(x) \in \mathbb{R}^d$ including shared information for sample x, where d is the hidden dimension. To further learn the commonality between different domains/tasks, it is essential to transform this shared representation into a more unified space. We now generate a set of learnable prototypes $\mathbf{P} = \{\mathbf{p}_k\}_{k=1}^K \in \mathbb{R}^{K\times d}$, which together form a structured prototype space. Here, K is the number of prototypes, and $\mathbf{p}_k$ represents the k-th prototype. These prototypes act as anchor vectors that collectively span the prototype space, providing a basis for comparing different representations.

The next step involves projecting the shared representation $J(x)$ of all samples into the same prototype space $\mathbf{P}$ to capture commonality. This projection results in a new, more informative representation $\mathbf{J}_i$ for sample x_i, named commonality representation. Specifically, the calculation of $\mathbf{J}$ is carried out as follows: (i) The coordinates in prototype space are determined as $\mathbf{a} = \{\alpha_k\}_{k=1}^K \in \mathbb{R}^K$, where each α_k, generated by $\phi(J(x)) = W_p J(x)$, reflects the contribution of the corresponding prototype $\mathbf{p}_k$ in representing $J(x)$, with W_p as the trainable weight. (ii) $\mathbf{J}$ is computed as a weighted sum of these prototypes, given by $\mathbf{J} = \sum_{k=1}^K \alpha_k \mathbf{p}_k$..

3.3.2 ***Specificity Representation.*** While the commonality representation is designed to capture shared features across different domains/tasks, it is often insufficient when the goal is to model the nuanced differences that exist between these domains/tasks. Therefore, to address this limitation, we introduce the specificity representation for each sample.

Similar to the commonality representation, the specificity representation also involves mapping each sample's features into a prototype space. However, instead of using a single set of global prototypes, we generate distinct prototype spaces for each domain/task to capture unified domain-specific or task-specific characteristics.

For domain-specific representation, given S domains, we generate a set of prototype spaces $\mathcal{P} = \{\mathbf{P}^j\}_{j=1}^S$, where each $\mathbf{P}^j$ is associated with a specific domain. Each $\mathbf{P}^j$ consists of a set of prototypes represented as $\mathbf{P}^j = \{\mathbf{p}_k^j\}_{k=1}^K \in \mathbb{R}^{K\times d}$, with K as the number of prototypes and d as the hidden dimension. Similar to the process used in commonality representation, the calculation of the specificity representation $\mathbf{H}$ is carried out as follows: (i) Given $H(x_i, s) \in \mathbb{R}^d$ for the s-th domain of sample x_i, it is projected into the prototype space $\mathbf{P}^s = \{\mathbf{p}_k^s\}_{k=1}^K$ for domain s, where the coordinates $\mathbf{a}^s = \{\alpha_k^s\}_{k=1}^K \in \mathbb{R}^K$ are determined by $\phi(H(x_i, s)) = W^s H(x_i, s)$. (ii) The final specificity representation $\mathbf{H}_i^s$ for sample x_i on domain s is given by the weighted sum as follow:

$$\mathbf{H}_i^s = \sum_{k=1}^K \alpha_k^s \mathbf{p}_k. \tag{4}$$

Given sample x_i, for task-specific representation $\mathbf{H}_i^t$ in task t, the alignment process is the same as domain-specific representation. Specifically, we generate a corresponding prototype space for each task and project the task-specific features $H^t(x)$ into this space, ensuring that the unique characteristics of each task are effectively captured and represented.

3.3.3 ***Prototype-Based Optimal Transport.*** After calculating the representations in the prototype space for shared/specific sample representations, e.g., obtaining the commonality representation $\mathbf{J}_i^s$ for the shared representation $J(x_i^s)$ of sample x_i in domain s, the next step is to compute the optimal projection between $J(x_i^s)$ and $\mathbf{J}_i^s$. This process ensures that the two representations are well-aligned. To achieve this, we use OT [26] method to construct the optimal projection. Suppose we have a sample representation $\mathbf{h}^{sample}$, and its projection representation in prototype space is $\mathbf{h}^{prototype} = \sum_{k=1}^K \alpha_k \mathbf{p}_k$. Let $p = \delta(\mathbf{h}^{sample})$ and $q = \sum_{k=1}^K \alpha_k \delta(\mathbf{p}_k)$ represent the discrete distributions of the sample representation $\mathbf{h}^{sample}$ and the projection representation $\mathbf{h}^{prototype}$, respectively, with δ as the Dirac function. OT seeks to identify the matching matrix $\mathbf{M}$ that minimizes the distance between these two distributions. For discrete distributions, this can be formalized by considering the cost associated with individual elements as follows:

$$\begin{aligned} \mathrm{OT}(p, q) &= \min_{\mathbf{M}_i \in \Pi(p,q)} \langle \mathbf{M}_i, \mathbf{C}_i \rangle \\ &= \min_{\mathbf{M}_i \in \Pi(p,q)} \sum_{k=1}^K \mathbf{M}_{ik}\mathbf{C}_{ik} + \epsilon \sum_{k=1}^K \mathbf{M}_{ik} \log \mathbf{M}_{ik}, \end{aligned} \tag{5}$$

where $\langle \cdot, \cdot \rangle$ is the Frobenius dot-product and $\mathbf{C}$ is the cost matrix with each element calculated as $\mathbf{C}_{ik} = 1 - \cos(\mathbf{h}^{sample}, \mathbf{p}_k)$, representing the distance between the i-th sample representation and the k-th prototype vector. $\mathbf{M}$ is the transport probability matrix. The second term is entropic regularization [6], allowing the optimization at a small computational cost in sufficient smoothness. ϵ is the strength of entropic regularization.

Since the OT process is differentiable, it allows for an end-to-end training process of the model network. The average OT distance between μ_i and υ_i across the entire training set can be treated as a loss function to be optimized:

$$\mathcal{L}_{ot}(\mathbf{h}^{sample}, \mathbf{h}^{prototype}) = \frac{1}{N} \sum_{i=1}^N \mathrm{OT}(p_i, q_i), \tag{6}$$

where N is the training set size. By optimizing Equation (6), we drive the sample distribution p towards the prototype distribution q to learn the optimal projection.

3.3.4 ***Prototype Constraint Loss.*** When learning specific representations $\mathbf{H}$, we need to maintain multiple prototype spaces. To ensure that each prototype space $\mathbf{P}^j$ effectively captures the unique characteristics of its corresponding domain, we introduce an independence loss. This loss encourages the prototype spaces for different domains to remain distinct, minimizing the overlap

between them and thus enhancing their ability to represent domain-specific features. The independence loss is defined as:

$$\mathcal{L}_{in}(\mathcal{P}) = \sum_{i \neq j} \text{cosine}^2(\mathbf{P}^i, \mathbf{P}^j), \tag{7}$$

where $\text{cosine}(\mathbf{P}^i, \mathbf{P}^j)$ calculates the cosine similarity between the prototypes of domains i and j, ensuring the representations remain distinct and focus on specificity.

3.4 Training

3.4.1 ***PRP Model for MDMTR.*** To implement MDMTR, we construct corresponding PRP models for domains and tasks, namely D-PRP and T-PRP, respectively. The D-PRP model takes sample features x and domain id s as inputs to obtain a common representation $\mathbf{J}^{dom}$ and specific representations $\mathbf{H}^s$ for corresponding domain s. Subsequently, the T-PRP model receives the output $\mathbf{J}^{dom} \oplus \mathbf{H}^s$ from D-PRP and generates a common representation $\mathbf{J}^{task}$ among all tasks and specific representations $\mathbf{H}^t$ for each task t. Finally, the predicted value $\hat{y}^t$ of task t is obtained by the dot product between the common representation $\mathbf{J}^{task}$ and the task-specific representation $\mathbf{H}^t$, which can be formulated as follow:

$$\hat{y}^t = \sigma(\mathbf{J}^{task}\mathbf{H}^t). \tag{8}$$

3.4.2 ***Training Process.*** We use binary cross-entropy loss to measure the prediction error for each task t, described as follows:

$$\mathcal{L}^t_{BCE} = \frac{1}{N}\sum_{i=1}^{N} y_i^t \log \hat{y}_i^t + (1 - y_i^t)\log(1 - \hat{y}_i^t), \tag{9}$$

where N is the training set size. In a single PRP, there are two Prototype-based Optimal Transport (POT) modules, which are designed to learn the common representation $\mathbf{J}$ and the specificity representation $\mathbf{H}$, respectively. Additionally, an independent constraint is introduced for the prototypes used to learn $\mathbf{H}$. Thus, our minimization objective for PRP is defined as follows:

$$\mathcal{L}_{PRP} = \mathcal{L}_{OT}(J(x), \mathbf{J}) + \mathcal{L}_{OT}(H(x), \mathbf{H}) + \alpha\mathcal{L}_{in}(\mathcal{P}), \tag{10}$$

where α serves as the loss weight for independence loss.

Finally, training an MDMTR model with T tasks requires two PRP modules, resulting in a total loss formulated as:

$$\mathcal{L}_{total} = \sum_{t=1}^{T} \mathcal{L}^{(t)}_{BCE} + \mathcal{L}_{D-PRP} + \mathcal{L}_{T-PRP}. \tag{11}$$

3.5 Complexity Analysis

According to the Sinkhorn algorithm [6], the time complexity for approximating the OT distance between two discrete distributions of size n with ε-accuracy is $n^2 * \log(1/\varepsilon)$. In the POT module, for each instance x_i, the OT distance is minimized to align its representation distribution p_i with the corresponding projection distribution q_i. When PRP is applied to MDMTR, POT is executed four times, resulting in an overall time complexity of $O(K\log(1/\varepsilon))$ for a single instance, where K is the number of prototypes. Notably, the time complexity of PRP is independent of the number of tasks T and domains S, which enhances its scalability.

The space complexity of PRP in MDMTR is $O((S+1)Kd + (T+1)Kd) = O((S+T+2)Kd)$, representing the memory cost of storing the prototype matrix. Since the number of prototypes K depends on data characteristics and is usually a small constant, it does not significantly increase the overall complexity in our setting. We conducted an analysis of the experiments, showing that our approach improves performance while ensuring acceptable model efficiency.

Table 1: Statistics of datasets comprising various domains.

Dataset	Domain	#Users	#Items	#Inters.	Perc.
KuaiRand	D1	961	1,596,491	2,407,352	21.76%
	D2	991	2,741,383	7,760,237	70.15%
	D3	171	332,210	895,385	8.09%
Movielens	D1	1,325	3,429	210,747	21.07%
	D2	2,096	3,508	395,556	39.55%
	D3	2,619	3,595	39,3906	39.38%

4 Experiments

4.1 Experimental Setup

4.1.1 Datasets. We conduct offline evaluations on two open-source datasets, i.e., KuaiRand [9] and Movielens [11], to assess the performance of PRP in MDMTR. Statistical information about these datasets is provided in Table 1.

- **KuaiRand**: This unbiased dataset is collected from the short video platform KuaiShou, which consists of various interactions on the app corresponding to different domains, denoted by the feature "tab" ranging from 0 to 14. For simplicity, we select the top three domains with the largest data volume. "Click" and "long-view" are used as multi-task targets.
- **Movielens**: This dataset contains one million ratings for approximately 3900 movies, along with 7 user attributes and 2 item attributes. It features diverse data, including ratings and user information across different domains and tasks, and records demographic details about users, such as gender, age, and occupation. We utilize the "age" attribute to categorize the dataset into three domains and derive the tasks of "click" and "like".

4.1.2 Baselines. To comprehensively evaluate performance, we compare our approach with several advanced baselines across MDR, MTR, and MDMTR settings. We configure the models as follows: Multi-Task (-MTR) runs S times for S domains, Multi-Domain (-MDR) runs T times for T tasks, and Multi-Domain Multi-Task (-MDMTR) handles both simultaneously. This diverse set of baselines ensures a thorough comparison across various real-world scenarios: **Shared Bottom [2]**, **MMoE [24]**, **PLE [32]**, **STAR [28]**, **M2M [42]**, **HiNet [46]**, **PEPNet [3]** and **M^{3}oE [44]**.

4.1.3 Evaluation Metrics. (i) AUC: Area Under the Curve (AUC) refers to the area under the ROC curve, used to evaluate the performance of binary classification models. The AUC value ranges from 0 to 1, with higher values indicating stronger classification ability; an AUC of 0.5 means the model's performance is equivalent to random guessing, while an AUC of 1 indicates perfect classification. (ii) LogLoss: Logarithmic Loss (LogLoss) is a loss function used to evaluate classification models, particularly in binary and multi-class problems. It measures the difference between the predicted probabilities of the model and the actual labels, with smaller values, indicating more accurate predictions by the model.

Table 2: Performance comparison of different methods across three domains on two tasks with unit of $*10^{-2}$. The best and second-best results are highlighted in bold and underlined, respectively. Higher AUC is better, while lower Logloss is better.

Type	Model	KuaiRand (AUC ↑)								Movielens (AUC ↑)							
		Click (T1)				Long-view (T2)				Click (T1)				Like (T2)			
		D1	D2	D3	ALL	D1	D2	D3	ALL	D1	D2	D3	ALL	D1	D2	D3	ALL
MTR	ShBot-MTR	71.73	72.56	77.25	77.01	91.69	76.74	79.84	82.56	78.93	80.30	78.99	79.54	81.44	81.07	80.17	80.80
	MMoE-MTR	71.88	72.57	78.18	77.17	91.47	76.73	80.03	82.48	79.21	80.28	79.08	79.64	81.48	80.94	80.15	80.72
	PLE-MTR	71.75	72.60	78.13	77.08	91.66	76.78	79.63	82.50	79.15	80.35	78.78	79.55	81.84	81.03	80.26	80.92
MDR	ShBot-MDR	71.89	72.09	78.36	77.07	91.17	76.59	80.22	82.14	80.41	81.09	79.77	80.49	81.35	81.92	79.49	80.85
	MMoE-MDR	71.96	72.13	78.29	77.09	90.54	76.63	80.16	82.08	80.33	81.15	79.85	80.54	81.16	81.74	79.54	80.76
	PLE-MDR	71.84	72.08	78.36	77.12	91.32	76.64	80.28	82.32	80.05	81.07	79.95	80.48	81.07	81.71	79.65	80.77
	STAR	71.73	72.15	78.28	76.62	90.39	76.24	79.95	81.71	80.46	81.21	80.07	80.66	81.82	82.35	79.81	81.25
MDMTR	ShBot-MDMTR	71.78	72.30	78.45	77.05	91.82	76.59	80.31	82.54	80.28	81.29	80.81	80.56	82.57	80.63	80.29	81.48
	MMoE-MDMTR	71.84	72.34	78.48	77.08	91.79	76.61	80.23	82.55	80.76	82.10	81.07	80.85	82.73	80.27	80.44	81.73
	PLE-MDMTR	71.81	72.41	78.42	77.12	91.89	76.63	80.21	82.58	80.74	81.87	81.10	80.90	82.92	80.33	80.70	81.80
	M2M	71.73	72.31	78.04	76.91	91.59	76.24	79.92	82.21	80.14	81.14	79.84	80.48	82.22	80.47	79.81	80.97
	HiNet	71.57	73.07	78.01	77.10	91.72	76.38	80.04	82.39	80.48	81.55	80.22	80.61	82.31	82.89	80.48	82.08
	PEPNet	71.45	72.92	78.45	77.09	91.40	76.91	80.24	82.67	81.26	81.95	80.56	80.78	82.40	82.92	80.40	82.05
	M^3oE	71.87	72.31	78.46	77.06	91.84	76.54	80.33	82.50	80.62	82.08	81.03	80.72	82.86	80.01	80.32	81.71
Ours	PRP	**72.72**	**73.25**	**79.37**	**78.00**	**92.83**	**77.50**	**81.48**	**83.49**	**82.04**	**82.13**	**81.13**	**81.75**	**84.07**	**83.18**	**82.06**	**82.88**

Type	Model	KuaiRand (LogLoss ↓)								Movielens (LogLoss ↓)							
		Click (T1)				Long-view (T2)				Click (T1)				Like (T2)			
		D1	D2	D3	ALL	D1	D2	D3	ALL	D1	D2	D3	ALL	D1	D2	D3	ALL
MTR	ShBot-MTR	33.65	60.59	59.29	54.40	17.18	52.94	53.06	44.80	54.89	52.68	53.26	53.38	41.37	40.61	43.34	41.81
	MMoE-MTR	33.54	60.59	55.77	54.24	18.36	52.96	53.78	45.11	54.57	53.02	53.28	53.45	41.48	41.21	43.22	42.05
	PLE-MTR	33.58	60.56	58.33	54.32	17.60	52.89	53.60	44.87	54.87	52.80	53.75	53.60	41.14	40.73	43.31	41.84
MDR	ShBot-MDR	33.48	60.76	53.65	54.29	17.45	53.03	49.24	44.79	53.18	51.94	52.59	52.45	41.65	40.27	44.25	42.12
	MMoE-MDR	33.50	60.74	53.65	54.28	19.28	53.10	49.41	45.26	53.08	51.83	52.31	52.28	41.70	40.32	44.18	42.13
	PLE-MDR	33.48	60.71	53.79	54.25	18.04	53.04	49.25	44.94	53.35	51.96	52.28	52.38	41.67	40.25	43.95	42.00
	STAR	34.82	60.97	53.98	54.75	19.63	53.49	49.71	45.63	52.97	51.79	52.22	52.19	41.30	39.88	44.27	41.89
MDMTR	ShBot-MDMTR	33.53	60.79	53.65	54.31	16.80	53.06	49.34	44.66	53.09	51.63	52.26	52.18	41.41	39.62	43.37	41.43
	MMoE-MDMTR	33.49	60.76	53.54	54.28	16.77	53.03	49.36	44.63	52.45	51.46	51.77	51.79	40.69	39.59	43.22	41.12
	PLE-MDMTR	33.52	60.72	53.55	54.25	16.72	53.00	49.29	44.60	52.59	51.42	51.79	51.82	41.20	39.49	43.06	41.11
	M2M	33.60	60.79	53.94	54.54	17.11	53.18	49.39	44.75	53.47	51.82	52.38	52.39	41.52	41.02	43.60	41.74
	HiNet	33.76	60.66	54.27	54.28	16.89	52.98	49.41	44.62	53.42	51.89	52.48	52.46	40.91	39.67	43.85	41.34
	PEPNet	33.49	60.59	53.88	54.12	18.29	53.37	49.97	45.48	53.43	51.73	51.99	52.23	40.73	39.51	43.54	41.28
	M^3oE	33.56	60.76	53.69	54.31	16.81	53.11	49.51	44.71	52.88	51.60	52.06	52.06	40.93	39.54	43.45	41.24
Ours	PRP	**33.08**	**60.09**	**53.39**	**53.69**	**16.47**	**52.41**	**48.74**	**44.10**	**51.90**	**51.18**	**51.46**	**51.44**	**39.13**	**39.44**	**42.40**	**40.54**

Table 3: Ablation results on the effects of different distribution measurements in PRP.

Distribution Alignment	KuaiRand (AUC ↑)								Movielens (AUC ↑)							
	Click (T1)				Long-view (T2)				Click (T1)				Like (T2)			
	D1	D2	D3	ALL	D1	D2	D3	ALL	D1	D2	D3	ALL	D1	D2	D3	ALL
OT	**72.72**	**73.25**	**79.37**	**78.00**	**92.83**	**77.50**	**81.48**	**83.49**	**82.04**	**82.13**	**81.13**	**81.75**	**84.07**	**83.18**	**82.06**	**82.88**
Manhattan	72.09	73.15	79.01	77.57	92.12	77.08	80.57	83.11	81.59	82.12	81.11	81.05	83.74	83.04	81.67	82.47
Euclidean	71.98	73.13	78.69	77.44	91.93	77.11	80.63	82.86	81.17	82.07	81.09	81.29	83.27	83.01	81.07	82.28

4.1.4 Implementation Details. For all methods, we use Adam optimizer with a learning rate of $1e^{-3}$. The batch size is 4096 for training and testing. In D-PRP, we set the number of experts to the number of domains, and the specific prototype is also equal to the number of domains. For each expert, we use a two-layer MLP with 64 and 32 hidden layers. In T-PRP, we set the number of experts to the number of tasks, and the specific prototype is also equal to the number of tasks. For each expert, we use a two-layer MLP with 64 and 32 hidden layers. We set the K of prototype learning to 32 and the weight of independence loss α to 0.5.

4.2 Performance Comparison

Table 2 shows the performance comparison of PRP and other baselines in MDMTR on two widely used metrics, respectively. In Table 2, we consider three domains and two tasks on KuaiRand and

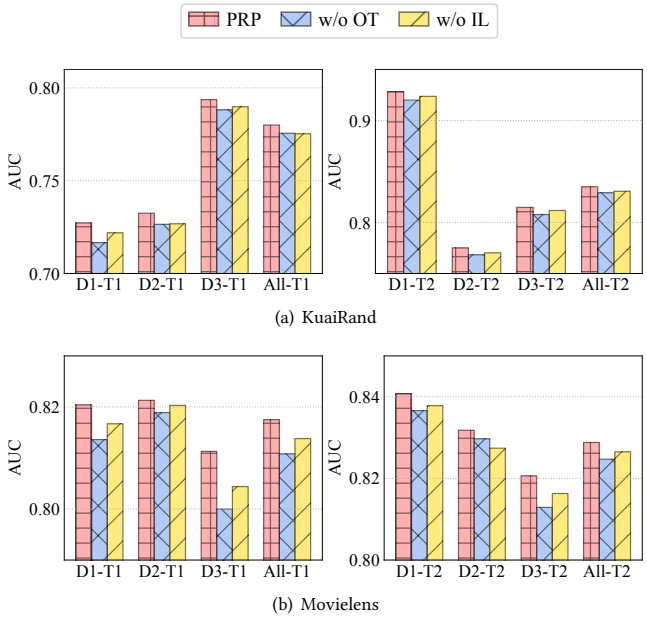


Figure 2: Results of the ablation study under D1, D2, and D3 for T1 and T2. "All" denotes all domains.

Movielens, respectively. We report the AUC and LogLoss of two tasks for each domain and the whole domain.

From the experimental results, we find the following observations: (i) For the baselines, MTR, MDR, and MDMTR all exhibit seesaw problems to varying degrees, i.e., they cannot maintain consistent results across multiple domains/tasks, with one usually performing better while the other performs worse. (ii) Compared with different baselines, PRP gains significant improvements in various domains and different tasks, mitigating the unbalance among MDR and MTR. This benefit comes from PRP extracting commonality and specificity separately, mapping them to distinct spaces to avoid representation entanglement. (iii) By capturing the common and specific characteristics of different fields and tasks, PRP can effectively balance the stable performance in multiple domains and multiple tasks, and will not have strengths in one domain and weaknesses in another.

4.3 Ablation Study

In this part, we conduct the ablation study to explore the contribution of each designed module in PRP. Therefore, we compare PRP with two different variants: (i) w/o OT, where the optimal transportation for aligning the instance into the prototype space is not used. (ii) w/o IL, without using independence loss to ensure different prototype networks are orthogonal.

These results illustrate two facts: (i) All designed modules are important for PRP. If any module is removed, AUC and LogLoss will decrease. For example, in Figure 2(a) and Figure 2(b), each variant performs poorly on KuaiRand and Movielens. (ii) Optimal transfer has the greatest impact on performance, because aligning different domains and tasks to corresponding prototype networks is an important basis for achieving stable MDMTR in PRP. Compared

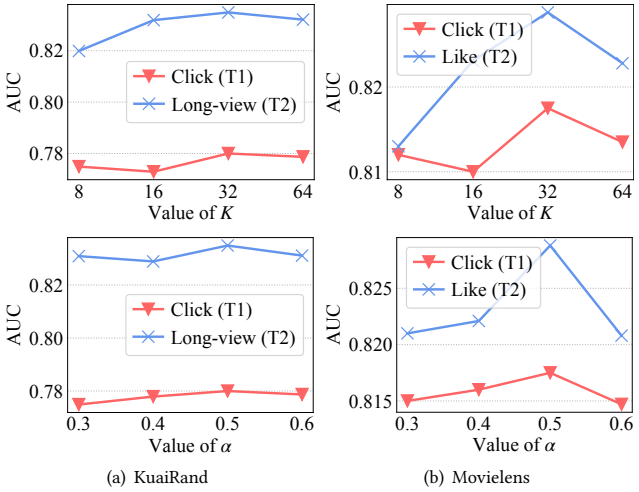


Figure 3: Performance comparison under different hyperparameter settings on KuaiRand and Movielens.

with existing methods that map features to the same space, our method ensures the ability to fully capture common features and characteristic features in different feature spaces. In addition, the addition of an orthogonal mechanism based on independent loss can ensure the correctness of feature mapping. Moreover, Table 3 compares the effects of different distribution measurements in PRP, including OT, Manhattan, and Euclidean. The results show that OT has the best performance, further highlighting its role in our proposed PRP.

4.4 Efficiency Analysis

In this section, we further compare the efficiency. We conduct an efficiency analysis of each model. Table 4 presents specific statistics on the number of parameters and the running time per epoch. The results indicate that PRP exhibits similar parameter numbers and running time as baselines, such as PEPNet and M^{3}oE. Compared with existing methods, our framework improves accuracy while ensuring acceptable model efficiency.

4.5 Hyperparameters Analysis

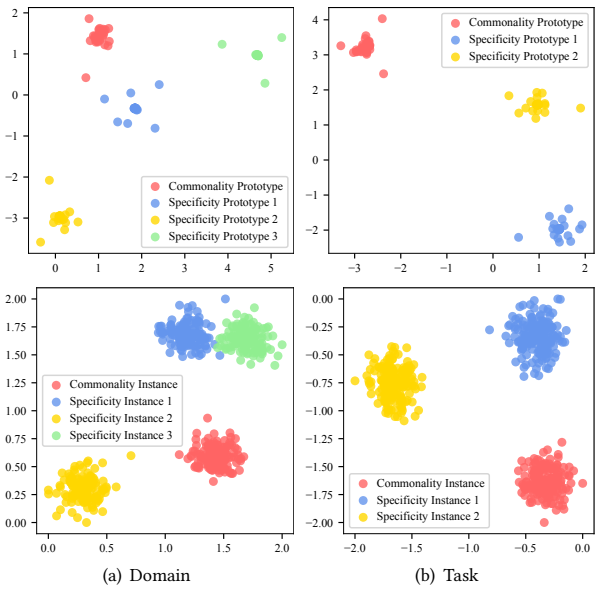
In Figure 3, we investigate the impact of several key hyperparameters on the recommendation performance for PRP.

4.5.1 ***Effects of*** K. We first evaluate the recommendation performance of PRP under different values of K, one crucial setting for representation alignment. From the results, we can find that the metrics first rise and then fall, with the optimal settings of $K = 32$ on KuaiRand and Movielens. Too small K will lead to underfitting of spatial mapping, while too large K will hinder generalization.

4.5.2 ***Effects of*** α. Additionally, we investigate PRP's performance under different values of the loss weight α for independence loss. For α, it can effectively regularize OT to prevent overfitting on the samples' locations. Results indicate that $\alpha = 0.5$ yields superior performance. Larger values may impede effective model updates, while smaller values compromise model robustness.

Table 4: Efficiency Analysis. We statistics the number of parameters and the running time per epoch for each model.

Dataset	Metric	Model							
		ShBot-MDMTR	MMoE-MDMTR	PLE-MDMTR	M2M	HiNet	PEPNet	M^3oE	PRP
KuaiRand	#Parameters	65.32M	65.48M	65.78M	72.08M	68.82M	66.88M	65.57M	65.54M
	Time Cost (min)	3.4	3.48	3.47	3.9	3.67	3.77	3.6	3.65
Movielens	#Parameters	0.24M	0.24M	0.28M	0.57M	0.68M	0.66M	0.25M	0.36M
	Time Cost (min)	0.8	0.95	1.12	1.37	1.23	1.25	1.17	1.21



(a) Domain (b) Task

Figure 4: T-SNE results of domain prototypes/instances (left) and task prototypes/instances (right) on KuaiRand.

Table 5: Online A/B test results on a video platform.

Domain	Metric	Confidence Interval	Improvement
Main	Like	[−0.02%, +0.30%]	+0.142%
	Long-view	[+0.05%, +0.35%]	+0.204%
Home	Like	[−0.23%, +0.27%]	+0.126%
	Long-view	[+0.22%, +1.10%]	+0.422%
Slide	Like	[−0.46%, +0.88%]	+0.283%
	Long-view	[+0.03%, +0.40%]	+0.214%

4.6 Visualization and Further Analysis

To further evaluate the impact of PRP in MDMTR, we conduct experiments on KuaiRand. In Figure 4, we compare D-PRP and T-PRP. For each PRP, we visualize the distribution of prototype vectors and instances in the feature spaces, employing T-SNE for dimensionality reduction. The visualizations demonstrate successful differentiation of different distributions and the presence of distinct boundaries in the space. In contrast to the conventional approach of mapping features from different domains/tasks into a single space, our method enhances the synergy between domains and tasks by mapping information into multiple feature spaces.

4.7 Online Development and A/B Test

To validate the effectiveness of the PRP model in real-world applications, we conduct A/B testing on an online video platform. We integrate the PRP model into the fully connected module of the ranking model and test it across three different domains (Main, Home, and Slide) with two tasks (like and long-view). As shown in Table 5, the PRP model significantly improves performance in all domains. Specifically, it not only increases the user-like rate but also significantly extends the time users spend watching videos. This demonstrates the broad applicability of the PRP model in various application domains and highlights its great potential in enhancing user experience.

5 Conclusion

In this paper, we propose a Prototype-guided Representation Projection (PRP) model for enhancing Multi-Domain Multi-Task Recommendation (MDMTR). Our method leverages prototype learning to handle commonality and specificity among various domains/tasks. By employing a shared Mixture of Experts (MoE) for common representation and specific feature extraction experts for domain/task-specific projections, constrained by the independence loss, PRP effectively manages complex domain/task relationships. Different from existing methods, Optimal Transport (OT) is utilized to guide the projection of representations within various prototype spaces, avoiding representation entanglement. Our extensive offline experiments on two open-source datasets, along with online A/B testing, consistently demonstrate that PRP outperforms existing methods in real-world applications across different domains/tasks.

Acknowledgement

This research is partially sponsored by funding from Yonghua Foundation.

References

[1] Jacob Bien and Robert Tibshirani. 2011. Prototype selection for interpretable classification. (2011).
[2] Rich Caruana. 1997. Multitask learning. *Machine learning* 28 (1997), 41–75.
[3] Jianxin Chang, Chenbin Zhang, Yiqun Hui, Dewei Leng, Yanan Niu, Yang Song, and Kun Gai. 2023. Pepnet: Parameter and embedding personalized network for infusing with personalized prior information. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3795–3804.
[4] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*. PMLR, 794–803.
[5] Chao Cui, Shisong Tang, Fan Li, Jiechao Gao, and Hechang Chen. [n. d.]. Calibrating Video Watch-time Predictions with Credible Prototype Alignment. In *Forty-second International Conference on Machine Learning*.
[6] Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems* 26 (2013).

[7] Mark Dredze, Alex Kulesza, and Koby Crammer. 2010. Multi-domain learning by confidence-weighted parameter combination. *Machine Learning* 79 (2010), 123–149.

[8] David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. 2013. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314* (2013).

[9] Chongming Gao, Shijun Li, Yuan Zhang, Jiawei Chen, Biao Li, Wenqiang Lei, Peng Jiang, and Xiangnan He. 2022. Kuairand: an unbiased sequential recommendation dataset with randomly exposed videos. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3953–3957.

[10] Jingtong Gao, Xiangyu Zhao, Bo Chen, Fan Yan, Huifeng Guo, and Ruiming Tang. 2023. AutoTransfer: instance transfer for cross-domain recommendations. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1478–1487.

[11] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* (2015), 1–19.

[12] Yun He, Xue Feng, Cheng Cheng, Geng Ji, Yunsong Guo, and James Caverlee. 2022. Metabalance: improving multi-task recommendations via adapting gradient magnitudes of auxiliary tasks. In *Proceedings of the ACM Web Conference 2022*. 2205–2215.

[13] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural computation* 3, 1 (1991), 79–87.

[14] Mahesh Joshi, Mark Dredze, William Cohen, and Carolyn Rose. 2012. Multi-domain learning: when do domains matter?. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. 1302–1312.

[15] Steinar Laenen and Luca Bertinetto. 2021. On episodes, prototypical networks, and few-shot learning. *Advances in Neural Information Processing Systems* 34 (2021), 24581–24592.

[16] Zerong Lan, Yingyi Zhang, and Xianneng Li. 2023. M3REC: A Meta-based Multi-scenario Multi-task Recommendation Framework. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 771–776.

[17] Danwei Li, Zhengyu Zhang, Siyang Yuan, Mingze Gao, Weilin Zhang, Chaofei Yang, Xi Liu, and Jiyan Yang. 2023. AdaTT: Adaptive Task-to-Task Fusion Network for Multitask Learning in Recommendations. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4370–4379.

[18] Fan Li, Xu Si, Shisong Tang, Dingmin Wang, Kunyan Han, Bing Han, Guorui Zhou, Yang Song, and Hechang Chen. 2024. Contextual distillation model for diversified recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 5307–5316.

[19] Meng Li and Haochen Sui. 2025. Causal Recommendation via Machine Unlearning with a Few Unbiased Data. In *AAAI 2025 Workshop on Artificial Intelligence with Causal Techniques*. https://openreview.net/forum?id=7btnaN4Std

[20] Pengcheng Li, Runze Li, Qing Da, An-Xiang Zeng, and Lijun Zhang. 2020. Improving multi-scenario learning to rank in e-commerce by exploiting task relationships in the label space. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2605–2612.

[21] Xiaopeng Li, Fan Yan, Xiangyu Zhao, Yichao Wang, Bo Chen, Huifeng Guo, and Ruiming Tang. 2023. Hamur: Hyper adapter for multi-domain recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 1268–1277.

[22] Junning Liu, Xinjian Li, Bo An, Zijie Xia, and Xu Wang. 2022. Multi-faceted hierarchical multi-task learning for recommender systems. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3332–3341.

[23] Jiaqi Ma, Zhe Zhao, Jilin Chen, Ang Li, Lichan Hong, and Ed H Chi. 2019. Snr: Sub-network routing for flexible parameter sharing in multi-task learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 216–223.

[24] 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*. 1930–1939.

[25] Wentao Ning, Xiao Yan, Weiwen Liu, Reynold Cheng, Rui Zhang, and Bo Tang. 2023. Multi-domain Recommendation with Embedding Disentangling and Domain Alignment. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 1917–1927.

[26] Filippo Santambrogio. 2015. Optimal transport for applied mathematicians. *Birkäuser, NY* 55, 58-63 (2015), 94.

[27] Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems* 31 (2018).

[28] Xiang-Rong Sheng, Liqin Zhao, Guorui Zhou, Xinyao Ding, Binding Dai, Qiang Luo, Siran Yang, Jingshan Lv, Chi Zhang, Hongbo Deng, et al. 2021. One model to serve all: Star topology adaptive recommender for multi-domain ctr prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4104–4113.

[29] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems* 30 (2017).

[30] Derun Song, Enneng Yang, Guibing Guo, Li Shen, Linying Jiang, and Xingwei Wang. 2024. Multi-scenario and multi-task aware feature interaction for recommendation system. *ACM Transactions on Knowledge Discovery from Data* 18, 6 (2024), 1–20.

[31] Ichigaku Takigawa, Mineichi Kudo, and Atsuyoshi Nakamura. 2009. Convex sets as prototypes for classifying patterns. *Engineering Applications of Artificial Intelligence* 22, 1 (2009), 101–108.

[32] Hongyan Tang, Junning Liu, Ming Zhao, and Xudong 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*. 269–278.

[33] Shisong Tang, Qing Li, Xiaoteng Ma, Ci Gao, Dingmin Wang, Yong Jiang, Qian Ma, Aoyang Zhang, and Hechang Chen. 2022. Knowledge-based temporal fusion network for interpretable online video popularity prediction. In *Proceedings of the ACM Web Conference 2022*. 2879–2887.

[34] Shisong Tang, Qing Li, Dingmin Wang, Ci Gao, Wentao Xiao, Dan Zhao, Yong Jiang, Qian Ma, and Aoyang Zhang. 2023. Counterfactual video recommendation for duration debiasing. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4894–4903.

[35] Yu Tian, Bofang Li, Si Chen, Xubin Li, Hongbo Deng, Jian Xu, Bo Zheng, Qian Wang, and Chenliang Li. 2023. Multi-Scenario Ranking with Adaptive Feature Learning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 517–526.

[36] Yichao Wang, Huifeng Guo, Bo Chen, Weiwen Liu, Zhirong Liu, Qi Zhang, Zhicheng He, Hongkun Zheng, Weiwei Yao, Muyu Zhang, et al. 2022. Causalint: Causal inspired intervention for multi-scenario recommendation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4090–4099.

[37] Yuhao Wang, Ha Tsz Lam, Yi Wong, Ziru Liu, Xiangyu Zhao, Yichao Wang, Bo Chen, Huifeng Guo, and Ruiming Tang. 2023. Multi-task deep recommender systems: A survey. *arXiv preprint arXiv:2302.03525* (2023).

[38] Yuntian Wu, Yuntian Yang, Jiabao Sean Xiao, Chuan Zhou, Haochen Sui, and Haoxuan Li. 2024. Invariant Spatiotemporal Representation Learning for Cross-patient Seizure Classification. In *The First Workshop on NeuroAI @ NeurIPS2024*. https://openreview.net/forum?id=Ex6wAivo7G

[39] Enneng Yang, Junwei Pan, Ximei Wang, Haibin Yu, Li Shen, Xihua Chen, Lei Xiao, Jie Jiang, and Guibing Guo. 2023. Adatask: A task-aware adaptive learning rate approach to multi-task learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 10745–10753.

[40] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. 2018. Robust classification with convolutional prototype learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3474–3482.

[41] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems* 33 (2020), 5824–5836.

[42] Qianqian Zhang, Xinru Liao, Quan Liu, Jian Xu, and Bo Zheng. 2022. Leaving no one behind: A multi-scenario multi-task meta learning approach for advertiser modeling. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 1368–1376.

[43] Yingyi Zhang, Xianneng Li, Yahe Yu, Jian Tang, Huanfang Deng, Junya Lu, Yeyin Zhang, Qiancheng Jiang, Yunsen Xian, Liqian Yu, et al. 2023. Meta-generator enhanced multi-domain recommendation. In *Companion Proceedings of the ACM Web Conference 2023*. 485–489.

[44] Zijian Zhang, Shuchang Liu, Jiaao Yu, Qingpeng Cai, Xiangyu Zhao, Chunxu Zhang, Ziru Liu, Qidong Liu, Hongwei Zhao, Lantao Hu, et al. 2024. M3oE: Multi-Domain Multi-Task Mixture-of Experts Recommendation Framework. *arXiv preprint arXiv:2404.18465* (2024).

[45] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 5941–5948.

[46] Jie Zhou, Xianshuai Cao, Wenhao Li, Lin Bo, Kun Zhang, Chuan Luo, and Qian Yu. 2023. Hinet: Novel multi-scenario & multi-task learning with hierarchical information extraction. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 2969–2975.

[47] Tianfei Zhou, Wenguan Wang, Ender Konukoglu, and Luc Van Gool. 2022. Rethinking semantic segmentation: A prototype view. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2582–2593.