SEQ+MD: Learning Multi-Task as a SEQuence with Multi-Distribution Data

Siqi Wang[1,*], Audrey Zhijiao Chen[2], Austin Clapp[2], Sheng-Min Shih[2] and Xiaoting Zhao[2]

[1]Boston University, 665 Commonwealth Ave, Boston, MA 02215
[2]Etsy, 117 Adams St, Brooklyn, NY 11201

Abstract

In e-commerce, ranking algorithms based on relevance and engagement signals have often shown improvement in sales and gross merchandise value (GMV). Designing such algorithms becomes particularly challenging when serving customers across diverse regional markets, as shopping preferences and cultural traditions vary significantly. We propose the SEQ+MD framework, which combines sequential learning for multi-task learning (MTL) with a region-based feature mask for handling multi-distribution data. This approach utilizes the sequential order within tasks and accounts for regional heterogeneity, enhancing performance on multi-source data. Unlike traditional sequential models that rely on tracking user interaction histories, SEQ operates on user-item feature pairs and generates task-specific predictions in sequence. Moreover, SEQ supports efficient parameter sharing across tasks and allows new tasks to be added easily. Notably, SEQ trained on data from only two tasks outperforms the baseline model trained on data from all three tasks when evaluated on the full three-task setting. Experiments on in-house data showed significant gains in high-value engagements, including add-to-cart and purchase actions. Furthermore, our multi-regional learning module can be flexibly applied to enhance other MTL applications.

Keywords

Multi-task Learning, Mixed-distribution Learning, E-commerce Search, E-commerce Ranking,

1. Introduction

In e-commerce, the design of item display algorithms is crucial for enhancing the customer shopping experience [1]. When a customer enters a query in the search window, the query typically goes through two stages to render final search results: retrieval and re-ranking. In the first stage, retrieval systems extract thousands of the most relevant items from millions of listings; in the re-ranking step, the thousands of listings are further re-ranked such that the most relevant results are shown at the top. Unlike traditional pattern-searching methods [2], machine learning offers possibilities for more personalized search experiences [3, 4]. The same search query from different users may yield completely different listing displays.

Designing effective machine learning algorithms for global e-commerce involves two major challenges. First, models often need to handle multiple tasks with unevenly distributed data. For example, *click* data is much more abundant than *purchase* data [8]. Multi-task learning (MTL) improves performance by enabling shared learning across tasks [9], as illustrated in Fig.1-(a), but it remains difficult to maintain balanced training and promote effective communication between tasks [10, 11]. Second, regional differences introduce significant variation in data distributions. In global marketplaces, users interact with international listings, yet shopping behaviors differ across countries due to cultural preferences. For instance, buyers in the UK are more likely to purchase cookie boxes as birthday gifts (Fig.2-(a)). These differences influence both the distribution and relevance of features. As shown in Fig.2-(b), some features are informative in certain regions but uninformative in others. Throughout this paper, we use "country" and "region" interchangeably, though a region may refer to any geographic area.

SIGIReCom'25: 2025 SIGIR Workshop on eCommerce, July 17, 2025, Padua, Italy
*Corresponding author.
siqiwang@bu.edu (S. Wang); achen@etsy.com (A. Z. Chen); aclapp@etsy.com (A. Clapp); sshih@etsy.com (S. Shih); xzhao@etsy.com (X. Zhao)
https://cs-people.bu.edu/siqiwang/ (S. Wang)

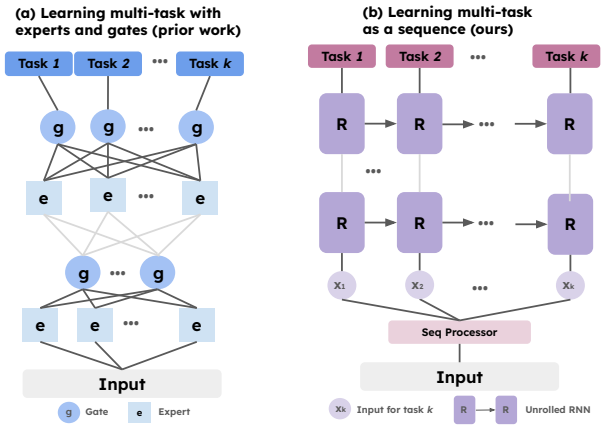

Figure 1: MTL Architecture Comparison. (a) Prior work [5, 6, 7] uses *experts* and *gates* for task knowledge sharing, with variations in whether the expert or gate is shared among tasks. (b) Our SEQ learns multi-task as a sequence, where task knowledge is shared through sequence tokens.

Existing methods usually address these two challenges separately. To the best of our knowledge, no single model currently solves both challenges effectively. Regarding multi-task learning (MTL), many approaches treat tasks independently [12, 13, 14], ignoring their natural sequential structure. Methods that consider task ordering either rely on user interaction sequences to predict the next item [15], or use separate task-specific towers followed by conditional probability modeling [16, 17, 18]. Beyond sharing a base model, interactions between tasks are typically limited to shared experts or gating mechanisms [7, 6, 5]. For region-specific data, most models are trained without accounting for regional variation, despite clear differences in input features across regions as shown in Fig. 2. While incorporating regional information could improve performance, training separate models for each region is inefficient and often ineffective due to imbalanced data availability, especially in regions with limited samples.

To this end, we propose the learning multi-task as a **SEQ**uence + **M**ulti-**D**istribution (**SEQ+MD**) framework, which can tackle the two challenges simultaneously. For the multi-task component, we observe that many user actions follow a natural sequence, such as clicking before purchasing, which can be modeled effectively as a sequential learning problem. Rather than treating each task independently, our SEQ architecture generates task predictions as a sequence, as shown in Fig.1-(b). The input pair of user and item features is first encoded into a sequence, and the model then outputs a probability token for each task in order. The most closely related work, HTLNet[18], also uses the output of earlier tasks as input for later ones. However, their approach relies on separate task towers, while our SEQ model uses a recurrent neural network (RNN) [19] that shares the same weights across tasks. This design supports efficient expansion to new tasks and maintains strong performance without the need for additional training. For handling mixed input distributions, we separate input features into region-invariant and region-dependent groups. The region-dependent features are processed with a country embedding in our multi-distribution (**MD**) learning module, meaning these features are transformed according to their region, and then concatenated with the region-invariant features. An advantage of this approach is that the MD module is easy to plug in and can enhance the performance of any multi-task learning model on multi-source data.

We evaluated our framework on our in-house data offline and observed a 1.8% performance increase in the critical purchase task while keeping the click task performance positive compared to baseline models. In summary, our contributions are:

- We introduced a new framework **SEQ** for multi-task learning with an improvised RNN architecture, specifically designed to handle tasks with sequential order. SEQ not only extracts and utilizes the sequence relation between tasks, reduces redundant computations among related tasks but also demonstrates excellent transferability when adding new tasks. By decomposing a complex task

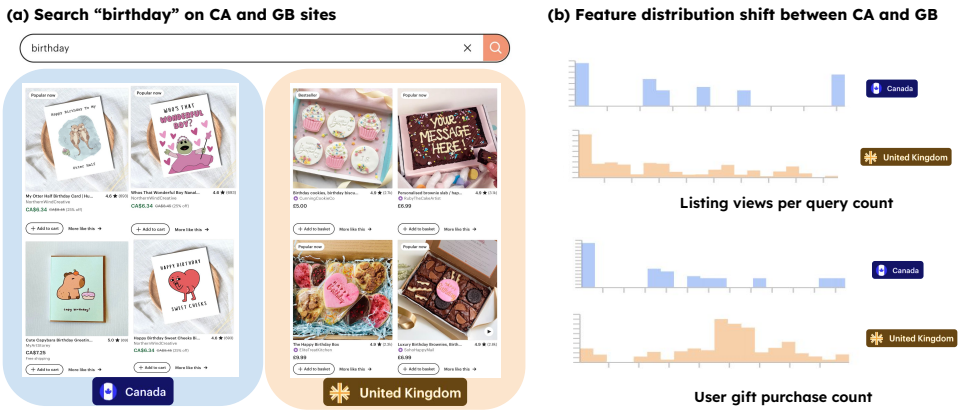

Figure 2: Regional Difference Examples. (a) The same search query on different regional sites should display different listings to reflect local preferences. For example, GB (United Kingdom) shoppers often choose cookie boxes as birthday gifts, while Canadian shoppers favor birthday cards. (b) Feature distribution shifts across countries. In Canada (CA) and the UK (GB), some features display an entirely different distribution pattern, posing a challenge for the model to learn.

into simpler, sequential tasks, SEQ effectively enhances the multi-task learning process.

- We developed a module **MD** for learning regional data with different distributions. The MD module enables the model to capture region-specific features while sharing region-invariant features, allowing for effective training with a more extensive and diverse dataset.
- Our in-house data experiments demonstrate improvements with this new framework.

2. Related Work

Multi-task learning (MTL) trains models on multiple tasks simultaneously. By sharing information across tasks, the model can learn more robust features, leading to improved performance on each individual task. MTL can be categorized into two types: *hard parameter sharing* and *soft parameter sharing*. Hard parameter sharing involves an architecture where certain layers are shared among all tasks in the base model, while other layers remain specific to individual tasks in separate task "towers." The "Shared-bottom" approach [12] is one of the most popular methods within this category. Soft parameter sharing uses trainable parameters to combine each layer's output with linear combinations. This approach often incorporates the concepts of *experts* and *gates*, which are multi-layer perceptrons (MLPs) in the architecture design. *Experts* are responsible for learning with specific attention from the features, while *gates* determine how to combine these attentions. Various methods differ based on whether the *experts* and *gates* are shared among tasks or specific to individual tasks, as shown in Fig. 1-(a). *E.g.* MMoE [5] shares all *experts* and *gates* parameters among the tasks; PLE [6] includes both task-specific and shared *experts* and *gates*; Adatt-sp [7] has task-specific experts, but all gates are shared among tasks. Soft parameter sharing heavily relies on experts and gates for knowledge sharing between multiple tasks. However, many related works often overlook the potential to utilize relationships between tasks in MTL. For tasks with a sequential order, Recurrent Neural Networks (RNNs) offer another method to promote knowledge sharing, which has been less explored.

Sequence learning in e-commerce has been explored to model user behavior patterns [15, 20, 17, 18]. For instance, DPN [21] retrieves target-related user behavior patterns using a target-aware attention mechanism, where user behaviors are represented by their shopping history—a sequence of purchased listings. Similarly, Hidasi *et al.* [22] demonstrates the impressive performance of RNNs over classical methods in session-based recommendations. GRU4Rec [23] takes the listing from the current event in the session and outputs a set of scores indicating the likelihood of each listing being the next in the session. However, these related works primarily focus on learning from listing interactions. To the best of our knowledge, our work is the first to treat tasks themselves as a sequence in the context of

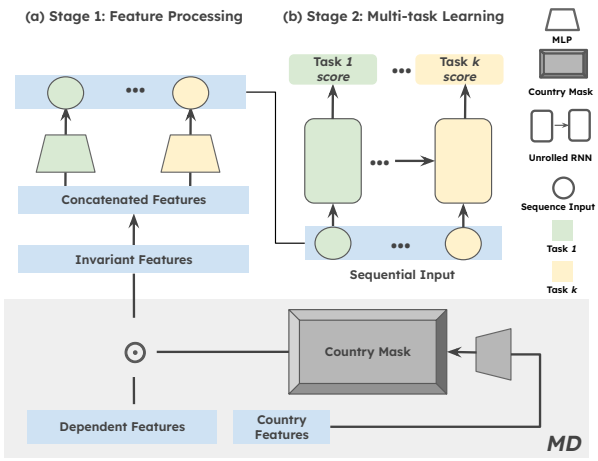

Figure 3: SEQ+MD overall architecture. (a) Feature processing. The input is split into three parts: *country features*, *dependent features*, and *invariant features*. *Country features* and *dependent features* are processed through our multi-distribution (MD) learning module. More details about the multi-distribution adaptor module can be found in Fig. 4. Concatenated features are pro- cessed into a sequence input with MLP blocks. (b) Multi-task Learning. The concatenated features pass through the following RNN layers, providing the model's final output scores for each task. Note that the RNN blocks illustrate the model's architecture, and the number of layers can vary.

e-commerce.

Multi-distribution learning trains models using data from various sources, each with distinct feature distributions. Multi-regional data is an example of multi-distribution input, with prior work largely focusing on language-agnostic approaches to create a unified, unbiased embedding space [24] or on learning consistent similarities across different markets [25, 26]. In contrast, our approach utilizes regionally distinct signals to enhance model **diversification**. Bonab *et al.* [27] propose learning in an MTL setting where each market is treated as a *task*. However, this approach faces challenges when market data is imbalanced, especially for smaller markets with limited data. Model-agnostic meta learning (MAML) [28] tackles this through a dual-loop training process: an inner loop optimizes each market individually, while an outer loop optimizes across markets, but the need for separate parameter fine-tuning for market adaptation makes MAML inefficient in this context. More recently, Market-Aware (MA) models [29] have used market-specific embeddings to create market-adapted item embeddings. Our MD module is similar to MA, yet we observed that not all features are region-specific [30], making it more effective to distinguish between shared and region-specific features.

3. Method

In this section, we introduce our SEQ+MD framework, which includes two model components: a multi-task learning architecture **SEQ** and a multi-distribution learning module **MD**. We provide formal definitions for the problem followed by detailed explanations for our framework in the subsections.

3.1. Problem Definition

Consider an online shopping dataset that records users' queries and interactions (*e.g.*, click, purchase) with the returned listings. Let $D = \{(X_i, Y_i)\}_{i=1}^{n}$ be the dataset with n samples, where $X = (x_u^m, x_l^p)$, x_u^m refers to the m-dimensional features about the *user* and *query*, x_l^p refers to the p-dimensional features about the target *listing*, and $Y = \{y_i\}_{i=1}^{k}$ is the score set for k tasks. The score for each task is calculated based on the user interaction sequences. A complete sequence would be ["click", "add to cart", "purchase"]. The last action in this sequence represents the final step. For example, if the sequence is ["click", "add to cart"], it means the user clicked on the listing and added it to the cart but did not

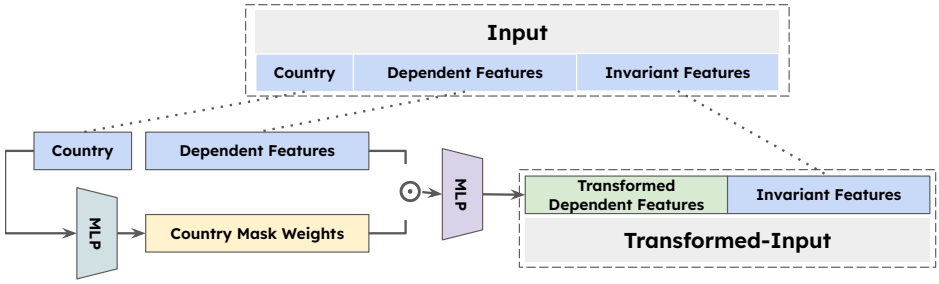

Figure 4: Multi-Distribution Adaptor Module (MD). The input is broken down into three parts: *Country features* (the cause of the distribution difference), *dependent features* (the features with multi-distributions), and *invariant features* (the features with consistent distributions). The *country features* generate a weight mask through an MLP block, which is then element-wise multiplied with the *dependent features*. This product feature is processed through an MLP, producing *transformed dependent features* that are assumed to be invariant. These are then concatenated with the original *invariant features* from the input to create the transformed input. This transformed input can then be passed to any MTL models for further processing.

purchase it. If none of these actions occurred, the sequence is ["no interaction"]. We assign specific scores to each action ("no interaction", "click", "add to cart", "purchase"), and the final task score is a combination of these action scores.

The multi-task learning architecture SEQ focuses on making predictions for the k tasks simultaneously given a single input X. Meanwhile, the multi-distribution learning module MD is designed for unified learning across the entire input set $\{(X_i)\}_{i=1}^{n}$, where the distribution of X for certain regions shows significant differences compared to other regions. (See Fig. 2-(b) for examples.) The multi-task learning architecture and multi-distribution learning module can be applied separately. We combine these two parts in our final framework and Fig. 3 shows the overall structure.

3.2. Learning Multi-Task as A SEQuence

Some tasks naturally form a sequence, e.g., *click*, *add to cart*, *purchase*, where each action occurs in a sequential order, conditional on the previous ones. However, most multi-task learning architectures do not account for the sequential nature of the problem, making the output tasks order-agnostic and interchangeable.

Introducing "order" into multi-task learning offers several benefits. First, sequential ordering allows the model to **prioritize more complex tasks** later in the sequence. In e-commerce, those later tasks (*e.g. purchase*) are often more critical than earlier (*e.g. click*) tasks because of their higher monetization values. At the same time, the data sparsity of the *purchase* task makes it more difficult to optimize. By establishing a sequence, knowledge from earlier (and typically easier) tasks can be used to address later (and often harder) tasks. Second, sequential ordering **facilitates the transfer or addition of new tasks**. Since the model learns tasks in a "continuous" manner, adding new tasks in the sequence requires minimal training cost. Journey Ranker [31] recognized the importance of task order having each task model predict the conditional probability based on the previous task. However, the MLP components in their model are isolated, not fully utilizing the knowledge exchange of the sequential tasks.

To address this, we connect RNNs [19] with multi-sequential-task learning. In RNN [19], the prediction of later tokens is based on previous tokens; similarly, our predictions for later user actions are conditioned on previous actions. In RNN [19], each token position shares the same set of weights (*e.g.* W_{hh}, W_{xh} and W_{hy} in Eq. 2, 3) with the only difference being the input token and the hidden input from previous tokens. In our approach, as shown in Eq. 1, we process the single input feature through an MLP for each token, transforming the input feature specifically for each task (see Fig. 3-(a)). The hidden input can be seen as the knowledge passed down from previous actions. As shown in Eq. 2, the knowledge for the current task j (h_j) is from both the input for task j ($MLP^j(X)$) and knowledge from the previous task $j-1$ (h^{j-1}). The score for task j (y^j) depends on the knowledge (h_j). Gated Recurrent Unit (GRU) [32]

is applied in our SEQ architecture.

$$[X_i^1, ..., X_i^k] = [MLP^1(X_i), ..., MLP^k(X_i)] \tag{1}$$

$$h_i^j = tanh(W_{hh}^T h_i^{j-1} + W_{xh}^T MLP^j(X_i)) \tag{2}$$

$$y_i^j = W_{hy}^T h_i^j \tag{3}$$

Fig. 3 shows our sequential task learning together with MD module. Given a single input feature, the first step is passing it through $k-1$ MLPs to create a length-k sequence, where k is the number of tasks. After passing through multiple layers of RNN, the output scores are in sequence form, with each score token corresponding to a task.

To further strengthen the learning with sequence, we add the **Descending Probability Regularizer** [31]. Based on the prior knowledge that the probability of a sequence of actions decreases from the beginning to the end (i.e., the probability of a user "clicking" the listing is greater than or equal to the probability of "purchasing"), we add a sigmoid multiplication at the end of the output. Each output score is activated with a sigmoid function and then multiplied by the previous sigmoid scores. As shown in Eq. 4, the score for task m, $\tilde{y_m}$ is the product of the sigmoid activations of the logits l from all previous tasks. This ensures that the output probabilities of later actions are always smaller than those of previous actions, aligning with the prior knowledge.

$$\tilde{y_m} = \prod_{i=1}^{i=m} sigmoid(l_i) \tag{4}$$

3.3. Learning with Multi-Distribution Input

Looking at the distribution of each raw input feature, we noticed that there are multi-distributions for certain features (*e.g.* average number of purchases, see examples in Fig.2-(b)). If the goal of training a machine learning model is to learn the transition from a input distribution to the output distribution, then this multi-distribution will pose significant challenges to the model, ultimately leading to a failure in learning [33].

Fig. 4 shows the overall structure of the multi-distribution adaptor module. We first break the input features into three parts: *country features* (which is the deciding factor of the distribution shift), *dependent features* (with distribution shifts across countries), and *invariant features* (which are country-agnostic features). The feature split is done in a heuristic way: country features are manually selected, and the *dependent features* and *invariant features* are separated with a distribution distance threshold. *i.e.*, when the average of the distribution distance among all countries is greater than a certain threshold, the feature is categorized as a *dependent feature*.

After splitting the input features, different operations are applied to these three groups of features. *Country* features are used to generate *country mask weights* for the *dependent features*. *Country mask weights* have the same dimension as the *dependent features*, and elementwise-multiplication is performed between the mask and *dependent features*. The multiplied input is fed into an MLP, which transforms the output into invariant features. These are then concatenated with the *invariant features* from the original input, resulting in a transformed input with consistent distributions.

This multi-distribution adaptor module MD can be easily plugged in for all MTL frameworks. Adding this module directly after the input and then sending the transformed input to the model is clean and simple. We also explore other options for combining this adaptor module with our sequential task learning framework, as shown in Fig. 3. Instead of concatenating the transformed dependent features with the input feature directly, we can concatenate them with the invariant feature model output from the previous layers. Block (b) in Fig. 3 shows how the multi-distribution module works in our sequential learning architecture. Each task has its own country mask. For a single input (*country*

Algorithm 1 SEQ+MD

Input: Feature (x_u, x_l), heuristic feature selector F, network for generating country mask $\text{MLP}_{\text{country mask}}$, k networks for each task input transformation $\{\text{MLP}_{\text{task } k}\}$, sequential learning network $\text{RNN}_{\text{seq task}}$.
Output: Scores $\{y_i\}_{i=1}^{k}$ for k tasks.

// Separate country features, dependent features, and invariant features for the input
$(x_{\text{country}}, x_{\text{dependent}}, x_{\text{invariant}}) \leftarrow F(x_u, x_l)$
// Generate country mask
$m_c \leftarrow \text{MLP}_{\text{country mask}}(x_{\text{country}})$
// Transformed dependent features
$x_{\text{trans d}} \leftarrow m_c \odot x_{\text{dependent}}$ ▷ element-wise multiplication
// Concatenate dependent and invariant features
$x \leftarrow \text{concat}(x_{\text{trans d}}, x_{\text{invariant}})$
// Transform into a feature sequence
$x_{\text{input seq}} \leftarrow []$
for $i = 1$ **to** k **do**
    $x_i \leftarrow \text{MLP}_{\text{task } i}(x)$
    $x_{\text{input seq}}.\text{append}(x_i)$
end for
// Learning multi-task as a sequence
score_logits $\leftarrow \text{RNN}(x_{\text{input seq}})$
// Calculate scores with descending probability regularizer
$score_i \leftarrow 1$
$scores \leftarrow []$
for $i = 1$ **to** k **do**
    $score_i \leftarrow score_i \times \sigma(\text{score\_logits}[i])$ ▷ σ denotes the sigmoid function
    $scores.\text{append}(score_i)$
end for

features, *dependent features*) transformed with k-task country masks, the output is also a length-k input sequence. Concatenated with the invariant feature output, the new input features can be processed with the following sequential learning layers to finally get the task scores.

4. Experiments

To evaluate our methods, we conducted experiments on our offline in-house datasets. Four baseline methods were selected for comparison. The Shared-Bottom model [12] is used as the baseline for all other models, as it represents the most fundamental architecture in multi-task learning (MTL). Results are reported as **changes relative to the Shared-Bottom model, with its performance marked as the 0% reference point**. The other methods implemented for reference are MLMMOE [5], PLE [6], and Adatt [7]. Details of the baselines are described in Sec. 4.1.

We used 14 days of offline in-house data for training and three days of data for evaluation, and we report the relative increase in the average Normalized Discounted Cumulative Gains ($NDCG$) [34] in the result tables (see Sec. 4.2 for more details). Due to the varying nature of different traffic sources, the results are divided into two sections: Webpage search traffic (Web), and Mobile App search traffic (App). We track multi-tasks across all traffic.

The results focus on two main areas: the effectiveness of the sequential learning architecture for MTL and the "plug-in" multi-distribution learning module for SOTA MTL methods. Ablation studies and alternative designs are discussed in Sec. 5.

Table 1
Multi-task learning performance. Results are reported with respect to the shared-bottom model baseline, with the best results marked in bold. State-of-the-art methods are listed in (a) and our models in (b). Our SEQ model outperforms all baselines in (a) across all tasks and platforms. Adding the multi-distribution learning module *MD* further enhances the performance. See Section 4.3 for further discussion.

		Click Task		Purchase Task	
	Platform	Web	App	Web	App
(a)	MLMMoE [5]	-0.043%	-0.315%	+1.027%	+0.553%
	PLE [6]	-0.450%	-0.298%	+0.790%	+0.512%
	AdaTT [7]	-0.001%	-0.541%	+0.572%	+0.556%
(b)	SEQ	**+0.618%**	**+0.476%**	+1.305%	+1.426%
	SEQ+MD	+0.170%	+0.091%	**+1.705%**	**+1.952%**

4.1. Baseline Models

We select a few state-of-the-art multi-task learning methods without any multi-distribution adjustments as the baselines. For multi-distribution learning challenges, most related work [35, 36] focuses on learning invariant features, whereas our goal is to better capture regional preferences. Thus, we use training with single or multi-distribution data as the baselines for multi-distribution learning comparisons.

Shared-bottom [12] is a hard parameter sharing method in MTL. It consists of a shared bottom layer for all tasks, followed by separate "tower" layers for each task, which extend from the shared-bottom output. Both the "bottom" and the "towers" are MLPs, with no knowledge sharing beyond the shared-bottom.

MLMMOE [5] is a soft parameter sharing method in MTL. It features *experts* and *gates*, which are MLPs within the architecture. "ML" refers to multiple layers; except for the top task-specific gates, all other *experts* and *gates* are shared among tasks.

PLE [6] is another soft parameter sharing method in MTL. It includes two types of *experts* and *gates*: task-specific and task-shared. Task-specific *experts* learn only for their individual tasks, and task-specific *gates* accept input exclusively from the same task *expert* or the shared *expert*.

Adatt-sp [7] is also a soft parameter sharing method in MTL. All *experts* are task-specific, while all *gates* take outputs from all *experts* as their input.

4.2. Datasets and Metrics

We exclusively use our in-house data for experiments because public search datasets [37] often omit feature details for data security reasons. This omission makes it difficult to isolate country features and generate accurate country mask weights. Our offline in-house dataset contains over 20 million <user, query, listing> interaction sequences from 10 regions and 2 platforms. Unless otherwise specified, we train the models with data from all regions and platforms. Results are evaluated separately for each platform. Normalized Discounted Cumulative Gain ($NDCG$) [34] is our evaluation metric, commonly used for measuring the effectiveness of search engines by summing the gain of the results, discounted by their ranked positions. The rankings of the search listings are ordered by the output scores from the model, and $NDCG$ is calculated based on the user interaction sequences. As discussed in Sec. 3.2, e-commerce prioritizes the *purchase* task over *click*, making *purchase-ndcg* our prioritized metric for model evaluation.

4.3. Results

SEQ. Table 1 presents the multi-task learning performance on *click* and *purchase* tasks across different platforms. State-of-the-art MTL baseline methods demonstrate various improvements in the *purchase* task but show a slight decline in the *click* task. In contrast, our SEQ model shows improvement across all tasks, adding MD module (SEQ+MD) achieves the best $NDCG$ on the critical *purchase* task. We observed a performance drop in the *click* task after adding the MD module to SEQ, making the final

Table 2
Multi-distribution learning module *MD* performance. Results are reported based on the improvements over the shared-bottom model [12] baseline, with the best results marked in bold. Applying the *MD* module to state-of-the-art MTL methods demonstrates varying degrees of overall improvement (Percentage changes with regard to no-MD baselines are marked in green for improvements and red for declines in performance). See Section 4.3 for further discussion.

	Click Task		Purchase Task	
Platform	Web	App	Web	App
MLMMoE [5]	-0.043%	-0.315%	+1.027%	+0.553%
MLMMoE [5] + MD	**+0.629%** 0.673%	**+0.291%** 0.607%	+0.129% 0.889%	-0.025% 0.575%
PLE [6]	-0.450%	-0.298%	+0.790%	+0.512%
PLE [6] + MD	-0.023% 0.429%	-0.113% 0.186%	**+1.958 %** 1.159%	**+1.891%** 1.372%
AdaTT [7]	-0.001%	-0.541%	+0.572%	+0.556%
AdaTT [7] + MD	+0.477% 0.478%	-0.115% 0.428%	+1.060% 0.486%	+0.863% 0.306%

click performance only slightly positive compared to the share-bottom baseline. This may be due to the *click* data being noisier and having higher variance. Another possible explanation is that the region-dependent features isolated by the MD module are more closely related to user/listing purchase history, which may have a greater impact on the purchase task.

MD: Multi-Distribution Learning Module. Table 2 illustrates the effectiveness of our multi-distribution learning module as a "plug-in" component for state-of-the-art MTL methods. The adapted models demonstrate overall improvements, with PLE [6]+MD achieving the best performance for the *purchase* task across all platforms. These results validate that our MD module can significantly enhance MTL performance.

5. Discussions

5.1. Will the sequential learning model benefit from more tasks?

A significant advantage of learning multi-task sequences lies in the inherent properties of RNNs, where weights are shared across all tokens in the sequence. This has two main benefits. First, it reduces redundant calculations among related tasks. For instance, tasks like *click* and *purchase* share many commonalities in the buyer's decision process, *i.e.* a listing that a user clicks on is also likely to be purchased. Second, by reinforcing the connections between tasks, later tasks in the sequence can be learned more effectively by decomposing them and beginning with easier tasks. As the sequence progresses, task difficulty can be seen as increasing, with earlier tasks acting as processors for the later ones. This recurrent learning process, from easier to harder tasks, is advantageous. For example, predicting which listing is likely to be *purchased* is challenging, but if the model starts by learning *click* behavior, it can learn better. We hypothesize that the sequential learning model will benefit from more tasks. In our experiment, we add an *add to cart* task between the *click* and *purchase* sequence to better reflect the buyer's shopping journey. The results in Table 3 support this hypothesis.

Table 3
Three-task learning performance. Results are reported based on the shared-bottom [12] model baseline, with the best results marked in bold. Upon adding an additional task, *add to cart*, our SEQ+MD model continues to outperform others, demonstrating even larger performance gains compared to the two-task learning scenario. See Sec.5.1 for the discussion.

	Click Task		Add to Cart Task		Purchase Task	
Platform	Web	App	Web	App	Web	App
MLMMoE [5]	-0.025%	-0.627%	+0.885%	+0.883%	+0.596%	+0.769%
PLE [6]	-0.728%	-0.458%	+0.672 %	+0.475%	+1.247 %	+1.396%
AdaTT [7]	**+0.163%**	-0.054%	+0.459%	+0.698%	+0.901%	+1.265%
SEQ+MD	-0.955%	**+0.104%**	**+0.990%**	**+1.029%**	**+1.731%**	**+2.342%**

5.2. Transferability from two-task to three-task

An important consideration for multi-task models is how easily they can adapt to additional tasks, the SEQ+MD model demonstrates a significant advantage. Adding new tasks requires almost no increase in parameters compared to the state-of-the-art models which increase parameter size by 30% on average. Moreover, reusing weights trained on previous tasks can also lead to improved performance in new task evaluations. Figure 5 illustrates the performance comparison of evaluating a three-task setup using weights from a two-task model. The RNN in SEQ+MD uses consistent weights across sequence positions, allowing a new task to be added by simply appending a token to the input sequence. This setup enables predictions for the new task without fine-tuning or additional data. In our three-task evaluation, we averaged the MLP weights from the *click* and *purchase* tasks to initialize the MLP weights for the *add to cart* task. After transforming the inputs separately with three MLPs as a sequence, we applied the RNN using weights trained on only two tasks. Notably, without exposure to *add to cart* data during training, the model still outperforms the baseline trained on three tasks in both *click* and *purchase* tasks. These results support our hypothesis that utilizing the sequential order of tasks can improve multi-task learning effectiveness.

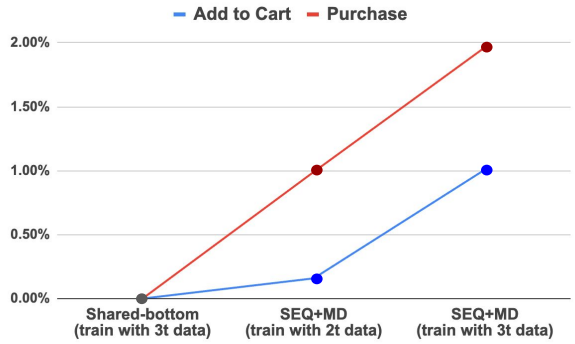

Figure 5: Transferability of SEQ+MD from two-task to three-task models is evaluated by comparing the performance of shared-bottom [12] and SEQ+MD models trained on three-task data with the SEQ+MD model trained on two-task data. Remarkably, despite the SEQ+MD model not being trained on *add to cart* data, it still shows improved performance on the *add to cart* and *purchase* tasks when compared to the shared-bottom [12] model. See Sec. 5.2 for the discussion.

5.3. Ablation studies

Learning multi-task as a sequence not only enhances knowledge sharing among tasks but also simplifies the integration of output regularization. In our SEQ design, we incorporate a *descending probability regularizer* that enforces the model to output task scores in a non-increasing order. This regularization is based on the observation that the probability of a user purchasing a listing cannot exceed the probability of them clicking on it, as a click typically precedes a purchase. The results in Fig. 6 demonstrate the effectiveness of this regularizer.

5.4. How effective is the MD module when compared to models trained with single regional data?

Our SEQ+MD model demonstrates a superior ability to align with regional preferences compared to other baselines. Figure 7 illustrates the changes in the percentage of domestic listings relative to the shared-bottom [12] model baseline (All models are trained with all-regional data.). Our in-house analysis shows distinct regional preferences in CA and GB, where CA buyers tend to favor international listings, while GB buyers lean towards domestic options. However, Fig. 7 shows that PLE [6] consistently returns more domestic listings, while AdaTT [7] consistently returns less, regardless of these regional

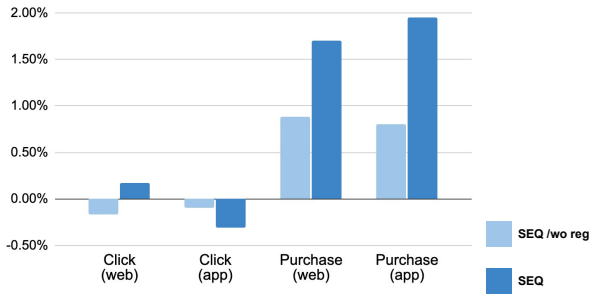

Figure 6: The impact of adding the *descending probability regularizer* in the SEQ model. Results are reported based on the improvements over shared-bottom model [12] baseline. Light blue represents the SEQ model without the regularizer, while dark blue indicates the model with the regularizer. The regularizer enhances performance, with noticeable improvements in the purchase task. See Sec. 5.3 for the discussion.

Figure 7: Domestic listing percentage changes compared to the shared-bottom model [12] baseline are illustrated for two representative regions: CA and GB. CA buyers tend to favor international listings, while GB buyers prefer domestic options. PLE [6] and AdaTT [7] show minimal regional differentiation, with AdaTT [7] consistently returning less domestic listings and PLE [6] returning more. In contrast, our SEQ+MD model consistently aligns better with regional preferences, demonstrating superior performance in fitting local market trends. See Sec. 5.4 for the discussion.

preferences. In contrast, our SEQ+MD model effectively captures these regional trends, providing more accurate rankings that better align with the buyers' preferences.

6. Conclusion

In this paper, we introduce the SEQ+MD framework, which integrates sequential learning for multi-task problems with multi-distribution data. While SEQ and MD can be applied independently, their combination yields stronger results, particularly on complex tasks. The motivation behind learning multi-task as a sequence stems from the natural sequential order of tasks. Our experiments and analyses highlight two primary benefits: First, SEQ reduces redundant computation across tasks and enhances transferability between different task sets, requiring minimal additional parameters while effectively utilizing weights from previous models. Second, by breaking down a complex task into simpler subtasks that serve as processors in the sequence, the model demonstrates improved performance on more challenging tasks. Additionally, our MD module effectively handles multi-distribution data, it can also enhance the performance of state-of-the-art multi-task learning models.

Future work. 1. Improve robustness against noisy data. Even though the primary goal of our approach is to improve performance on complex tasks such as *add to cart* and *purchase*, we see opportunities in making SEQ+MD have a neutral impact on *click* compared to SEQ only. One hypothesis is that click data tends to be noisier than other tasks, with a significant amount of "false clicks" present, particularly on mobile platforms. For example, users may accidentally click on a listing due to the touch screen's sensitivity. Learning with task-specific noise within a multi-task learning framework could be a valuable direction for future research. **2. Generalize multi-distribution data from region-wise to other scenarios.** While this paper focuses on regional differences as an example of multi-distribution, other multi-distribution exists in e-commerce search data. For instance, different platforms (web, app) may show distinct shopping patterns. Extending our MD module to address these scenarios could be a promising research direction.

References

[1] H. A. Lari, K. Vaishnava, K. Manu, Artifical intelligence in e-commerce: Applications, implications and challenges, Asian Journal of Management 13 (2022) 235–244.

[2] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Analysis of recommendation algorithms for e-commerce, in: Proceedings of the 2nd ACM Conference on Electronic Commerce, 2000, pp. 158–167.

[3] A. De Mauro, A. Sestino, A. Bacconi, Machine learning and artificial intelligence use in marketing: a general taxonomy, Italian Journal of Marketing 2022 (2022) 439–457.

[4] H. Yoganarasimhan, Search personalization using machine learning, Management Science 66 (2020) 1045–1070.

[5] J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong, E. H. Chi, 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, 2018, pp. 1930–1939.

[6] H. Tang, J. Liu, M. Zhao, X. Gong, Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations, in: Proceedings of the 14th ACM Conference on Recommender Systems, 2020, pp. 269–278.

[7] D. Li, Z. Zhang, S. Yuan, M. Gao, W. Zhang, C. Yang, X. Liu, J. Yang, 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, 2023, pp. 4370–4379.

[8] J. Oh, S. Bellur, S. S. Sundar, Clicking, assessing, immersing, and sharing: An empirical model of user engagement with interactive media, Communication Research 45 (2018) 737–763.

[9] Y. Li, X. Tian, T. Liu, D. Tao, On better exploring and exploiting task relationships in multitask learning: Joint model and feature learning, IEEE transactions on neural networks and learning systems 29 (2017) 1975–1985.

[10] Y. Zhang, Q. Yang, An overview of multi-task learning, National Science Review 5 (2018) 30–43.

[11] Y. Zhang, Q. Yang, A survey on multi-task learning, IEEE transactions on knowledge and data engineering 34 (2021) 5586–5609.

[12] R. Caruana, Multitask learning, Machine learning 28 (1997) 41–75.

[13] I. Misra, A. Shrivastava, A. Gupta, M. Hebert, Cross-stitch networks for multi-task learning, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 3994–4003.

[14] S. Ruder, J. Bingel, I. Augenstein, A. Søgaard, Latent multi-task architecture learning, in: Proceedings of the AAAI conference on artificial intelligence, volume 33, 2019, pp. 4822–4829.

[15] E. Yuan, W. Guo, Z. He, H. Guo, C. Liu, R. Tang, Multi-behavior sequential transformer recommender, in: Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval, 2022, pp. 1642–1652.

[16] X. Ma, L. Zhao, G. Huang, Z. Wang, Z. Hu, X. Zhu, K. Gai, Entire space multi-task model: An effective approach for estimating post-click conversion rate, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018, pp. 1137–1140.

[17] X. Tao, M. Ha, Q. Ma, H. Cheng, W. Lin, X. Guo, L. Cheng, B. Han, Task aware feature extraction framework for sequential dependence multi-task learning, in: Proceedings of the 17th ACM Conference on Recommender Systems, 2023, pp. 151–160.
[18] X. Tang, Y. Qiao, F. Lyu, D. Liu, X. He, Touch the core: Exploring task dependence among hybrid targets for recommendation, in: Proceedings of the 18th ACM Conference on Recommender Systems, 2024, pp. 329–339.
[19] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078 (2014).
[20] D. Xi, Z. Chen, P. Yan, Y. Zhang, Y. Zhu, F. Zhuang, Y. Chen, Modeling the sequential dependence among audience multi-step conversions with multi-task learning in targeted display advertising, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 3745–3755.
[21] H. Zhang, J. Pan, D. Liu, J. Jiang, X. Li, Deep pattern network for click-through rate prediction, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024, pp. 1189–1199.
[22] B. Hidasi, A. Karatzoglou, Recurrent neural networks with top-k gains for session-based recommendations, in: Proceedings of the 27th ACM international conference on information and knowledge management, 2018, pp. 843–852.
[23] B. Hidasi, A. Karatzoglou, L. Baltrunas, D. Tikk, Session-based recommendations with recurrent neural networks, arXiv preprint arXiv:1511.06939 (2015).
[24] A. Ahuja, N. Rao, S. Katariya, K. Subbian, C. K. Reddy, Language-agnostic representation learning for product search on e-commerce platforms, in: Proceedings of the 13th International Conference on Web Search and Data Mining, 2020, pp. 7–15.
[25] J. Cao, X. Cong, T. Liu, B. Wang, Item similarity mining for multi-market recommendation, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 2249–2254.
[26] X. Li, Z. Qiu, J. Jiang, Y. Zhang, C. Xing, X. Wu, Conditional cross-platform user engagement prediction, ACM Transactions on Information Systems 42 (2023) 1–28.
[27] H. Bonab, M. Aliannejadi, A. Vardasbi, E. Kanoulas, J. Allan, Cross-market product recommendation, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 110–119.
[28] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: International conference on machine learning, PMLR, 2017, pp. 1126–1135.
[29] S. Bhargav, M. Aliannejadi, E. Kanoulas, Market-aware models for efficient cross-market recommendation, in: European Conference on Information Retrieval, Springer, 2023, pp. 134–149.
[30] J. Cao, S. Li, B. Yu, X. Guo, T. Liu, B. Wang, Towards universal cross-domain recommendation, in: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, 2023, pp. 78–86.
[31] C. H. Tan, A. Chan, M. Haldar, J. Tang, X. Liu, M. Abdool, H. Gao, L. He, S. Katariya, Optimizing airbnb search journey with multi-task learning, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 4872–4881.
[32] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv:1412.3555 (2014).
[33] B. Peng, The sample complexity of multi-distribution learning, in: The Thirty Seventh Annual Conference on Learning Theory, PMLR, 2024, pp. 4185–4204.
[34] H. Valizadegan, R. Jin, R. Zhang, J. Mao, Learning to rank by optimizing ndcg measure, Advances in neural information processing systems 22 (2009).
[35] J. Cha, K. Lee, S. Park, S. Chun, Domain generalization by mutual-information regularization with pre-trained models, in: European conference on computer vision, Springer, 2022, pp. 440–457.
[36] J. Cha, S. Chun, K. Lee, H.-C. Cho, S. Park, Y. Lee, S. Park, Swad: Domain generalization by seeking flat minima, Advances in Neural Information Processing Systems 34 (2021) 22405–22418.

[37] P. Li, R. Li, Q. Da, A.-X. Zeng, L. Zhang, 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, 2020, pp. 2605–2612.