Deconstructing Multi-Task Learning in Search & Recommender Systems: A Combinatorial Study Project Proposal for NLP Course, Winter 2025

Wojciech Kutak
Warsaw University of Technology
wojciech.kutak.stud@pw.edu.pl

Abstract

Multi-Task Learning (MTL) is fundamental to modern e-commerce search and recommender systems, yet existing literature often conflates distinct architectural roles into monolithic models. This paper proposes a systematic deconstruction of MTL architectures into three functional pillars: the *Bottom* (semantic representation learning), the *Towers* (cross-task information routing), and the *Top* (probabilistic inference). Leveraging this taxonomy, we conduct an extensive combinatorial study of state-of-the-art components—including **MMOE**, **PLE**, **ResFlow**, and **AITM**—to identify synergistic combinations that optimize the trade-off between semantic relevance and user engagement. Furthermore, we introduce a novel **Task Attention (TA)** module within the Tower layer. By utilizing multi-head self-attention, the TA module explicitly captures latent sequential dependencies between ranking tasks (e.g., CTR and CVR). Extensive experiments on the public AliExpress dataset and a large-scale proprietary dataset from Allegro demonstrate that our modular framework and the proposed Task Attention mechanism significantly mitigate negative transfer and the "seesaw phenomenon," outperforming monolithic baselines in high-sparsity conversion funnels.

1 Introduction

The fundamental interaction in large-scale e-commerce is linguistic: a user expresses intent via a query, and the system retrieves items described by text. This creates a complex multi-objective environment. The system must satisfy *Semantic Relevance* (ensuring the retrieved "bijoux" matches the query "jewelry" across languages) while simultaneously maximizing *User Engagement* (predicting clicks and purchases).

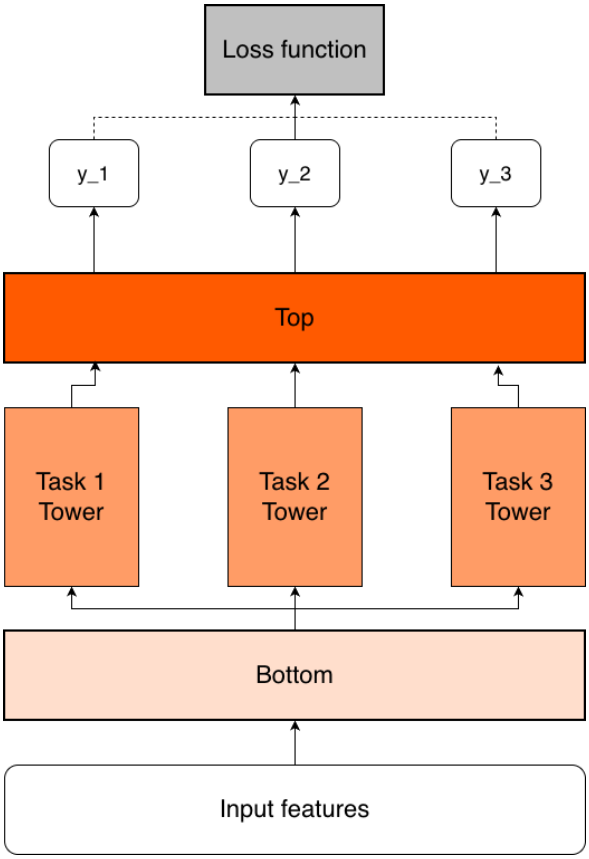

Figure 1: General Multi-task model framework.

Historically, Multi-Task Learning (MTL) research has treated these signals as generic numerical targets. However, recent works indicate that treating Semantic Relevance as a distinct, foundational task within the MTL framework significantly improves performance.

We argue that architectural innovations in this space are best understood by how they propagate this textual understanding. For instance, *PLE* focuses on separating semantic signals from behavioral noise at the representation level, while *Res-*

Flow explicitly routes relevance scores to boost conversion predictions.

To systematize this, we propose a tripartite taxonomy:

1. **The Bottom:** The shared encoder layer responsible for synthesizing dense semantic vectors from textual inputs (Query/Item Titles).

2. **The Towers:** The intermediate processing layers that route information between the "Semantic" pathway and the "Engagement" pathway.

3. **The Top:** The inference layer dealing with the probabilistic dependencies of the final labels.

This proposal outlines a combinatorial study to determine which architectural combinations best preserve semantic understanding while optimizing for downstream commercial metrics.

2 Related Work

2.1 Multi-task Learning

Multi-task learning (MTL) is essential for modern recommender systems, enhancing generalization and mitigating data sparsity. Initial methods relied on **Hard Parameter Sharing** (**Shared-Bottom**), sharing a common bottom layer for efficiency. This is susceptible to negative transfer when tasks are loosely correlated.

To improve on this, **Soft Parameter Sharing** models emerged. The **Multi-gate Mixture-of-Experts** (**MMOE**) (Ma et al.(2018b)) introduced task-specific gating networks to selectively combine shared expert submodels, capturing both shared and task-specific patterns. However, this can lead to the "seesaw phenomenon," where one task's improvement hurts another's performance. The **Progressive Layered Extraction** (**PLE**) (Tang et al.(2020)) model addresses this by explicitly separating task-shared and task-specific experts, using a progressive routing mechanism to avoid harmful parameter interference. Beyond architecture, probabilistic approaches like the **Entire Space Multi-Task Model** (**ESMM**) (Ma et al.(2018a)) and **Deep Bayesian Multi-Target Learning** (**DBMTL**) (Wang et al.(2019)) model logical dependencies between labels (e.g., $P(CTCVR) = P(CTR) \times P(CVR)$) to leverage the full sample space.

2.2 Sequence Learning in E-commerce

User behavior in e-commerce is inherently sequential (e.g., *Impression* $\rightarrow$ *Click* $\rightarrow$ *Conversion*). Ignoring this sequence leads to suboptimal performance, as dense upstream tasks provide critical signals for sparse downstream predictions. **ResFlow** (Fu et al.(2024)) adopts a lightweight approach, introducing residual connections between corresponding layers of task networks to facilitate an additive flow of information from higher-frequency to sparser tasks. The **SEQ+MD** (Wang et al.(2024)) framework treats MTL as a sequence generation problem, utilizing **Recurrent Neural Networks** (**RNNs**) to predict later tasks conditioned on the hidden states of earlier ones, effectively decomposing complex ranking into simpler sequential sub-tasks.

2.3 Attention Mechanism

Recent works explicitly model this sequential dependence. The **Adaptive Information Transfer Multi-task** (**AITM**) (Xi et al.(2021)) framework uses an attention-based module to learn the optimal amount and type of information to transfer between adjacent task towers for multi-step conversions. Attention mechanisms enable dynamic focus on relevant information and are commonly implemented in MTL via gating networks. In **MMOE** and **PLE**, gating networks calculate Softmax weights to determine the contribution of different expert modules to a task. The **AITM** framework uses a more explicit attention-based **Adaptive Information Transfer** (**AIT**) module. This module weighs the importance of the transferred information against the current task's representation, learning distinct transfer weights adapted to different conversion stages and audiences. While **SEQ+MD** models sequence via recurrence, attention remains a powerful alternative for structuring complex dependencies within task towers.

Table 1: Taxonomy of MTL Architectures

Component	Function	Key Models
Bottom (Encoder)	Semantic Representation	MMOE PLE
Towers (Routing)	Information Flow	ResFlow SEQ+MD
Top (Inference)	Probabilistic Dependence	ESMM, AITM DBMTL

3 Proposed Methodology

In this section we provide the description of the two novelties of the work, the high-level framework for combining different MTL architectures and new architecture which captures the sequential dependencies of the task with attention mechanisms.

3.1 Combinations of architectures

We define an MTL model as a tuple $M = \langle \mathcal{B}, \mathcal{T}, \mathcal{P} \rangle$.

3.1.1 The Bottom Section ($\mathcal{B}$): The Semantic Encoder

The Bottom component acts as the primary **Semantic Encoder**. It ingests a heterogeneous feature space consisting of high-dimensional textual features x_{text} (e.g., Query Embeddings, Item Titles) and numerical features x_{num} (e.g., price, historical stats).

$$\mathcal{B}(x_{text}, x_{num}) \rightarrow \{b_{CTR}, b_{CVR}\} \quad (1)$$

The goal of this layer is to model the semantic interaction between the user query and the candidate items. It outputs latent representations that must balance general semantic understanding with task-specific utility. We will evaluate architectures like **PLE** and **MMOE** here to determine which structure best preserves semantic signals while learning behavioral interaction features.

3.1.2 The Towers ($\mathcal{T}$): Sequential Modeling

The Tower component handles **Sequential Task Modeling**. While traditional MTL treats tasks as parallel independent outputs, the user journey is inherently sequential (Cognitive Relevance Assessment $\rightarrow$ Click Action $\rightarrow$ Purchase Decision).

$$\mathcal{T}(\{b_{CTR}, b_{CVR}\}) \rightarrow \{t_{CTR}, t_{CVR}\} \quad (2)$$

This layer is responsible for routing the "Semantic Relevance" signal into downstream behavioral predictions. We investigate architectures that support this sequential information flow, comparing the residual connections of **ResFlow** against the recurrent structures of **SEQ**.

3.1.3 The Top ($\mathcal{P}$): Causal Consistency

Finally, the Top component $\mathcal{P}$ transforms features into probabilities, enforcing the causal constraints of the user journey (e.g., *ESMM*'s probability decomposition).

$$\mathcal{P}(\{t_{CTR}, t_{CVR}\}) \rightarrow \{p_{CTR}, p_{CVR}\} \quad (3)$$

We perform an exhaustive study of the Cartesian product $\mathcal{B} \times \mathcal{T} \times \mathcal{P} \in \{\textbf{MMOE}, \textbf{PLE}\} \times \{\textbf{MLP}, \textbf{ResFlow}, \textbf{SEQ}\} \times \{\text{Identity}, \textbf{ESMM}, \textbf{AITM}\}$ to find if and which combination of architectures yields the most significant improvements in performance of the model.

3.2 Task Attention Towers

We introduce the **Task Attention** (**TA**) module which utilizes self-attention mechanism to capture sequential dependencies between the task. Given the input to the Towers section as described in 2, let's define a vector $\mathbf{b} = [b_1, b_2, \ldots, b_S] \in R^{S \times D}$, where S is the number of tasks and D is the dimensionality of the vectors from the towers section. To explicitly model the dependencies and interactions between different tasks, we employ a mechanism inspired by the self-attention found in Transformer architectures, which has been shown to effectively capture relationships in multi-task settings.

First, we incorporate task-specific information by adding learned task-embeddings $\mathbf{T} \in R^{S \times D}$ to the input vector. Let $\mathbf{H}$ denote the input to the self-attention layer:

$$\mathbf{H} = \mathbf{b} + \mathbf{T} \quad (4)$$

To capture diverse task relationships from different representation subspaces, we utilize Multi-Head Attention. Instead of a single attention function, we project the queries, keys, and values h times with different, learned linear projections. We then perform the attention function in parallel on each of these projected versions of queries, keys, and values.

For each head i ($1 \leq i \leq h$), we compute the Query ($\mathbf{Q}_i$), Key ($\mathbf{K}_i$), and Value ($\mathbf{V}_i$) matrices:

$$\mathbf{Q}_i = \mathbf{H}\mathbf{W}_i^Q, \quad \mathbf{K}_i = \mathbf{H}\mathbf{W}_i^K, \quad \mathbf{V}_i = \mathbf{H}\mathbf{W}_i^V \quad (5)$$

where $\mathbf{W}_i^Q \in R^{D \times d_k}$, $\mathbf{W}_i^K \in R^{D \times d_k}$, and $\mathbf{W}_i^V \in R^{D \times d_v}$ are learnable weight matrices specific to the i-th head. The scaled dot-product attention for each head is calculated as:

$$\text{head}_i = \text{Attention}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i), \quad (6)$$

where

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} \tag{7}$$

The outputs from all h heads are then concatenated and linearly projected again to result in the final values:

$$\text{MultiHead}(\mathbf{H}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\mathbf{W}^O \tag{8}$$

where $\mathbf{W}^O \in R^{hd_v \times D}$ is the output projection matrix.

To ensure training stability and regularization, we apply Dropout and Layer Normalization. This results in the intermediate representation $\mathbf{Z}$:

$$\mathbf{Z} = \text{LayerNorm}\left(\mathbf{H} + \text{Dropout}(\text{MultiHead}(\mathbf{H}))\right) \tag{9}$$

Subsequently, the matrix $\mathbf{Z}$ is fed into a position-wise Multi-Layer Perceptron (MLP) layer. Consistent with standard MLP structures used in recommender systems, this consists of two linear transformations with a non-linear activation function (ReLU) in between:

$$\text{MLP}(\mathbf{Z}) = \text{ReLU}(\mathbf{Z}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \tag{10}$$

where $\mathbf{W}_1 \in R^{D \times d_{ff}}$ and $\mathbf{W}_2 \in R^{d_{ff} \times D}$ are weights, and $\mathbf{b}_1, \mathbf{b}_2$ are bias terms.

Finally, we apply a second residual connection followed by Layer Normalization to obtain the output vectors $\mathbf{t} \in R^{S \times D}$:

$$\mathbf{t} = \text{LayerNorm}\left(\mathbf{Z} + \text{Dropout}(\text{MLP}(\mathbf{Z}))\right) \tag{11}$$

The resulting vectors $\mathbf{t}$ are then forwarded to the top section of the architecture.

4 Experiments

To evaluate the effectiveness of the proposed modular architecture decomposition (*Bottom*, *Tower*, *Top*) and measure the inter-architectural influence, we conduct extensive experiments on a large-scale public benchmark. We aim to answer the following research questions:

- **RQ1:** How do different combinations of *Bottom*, *Tower*, and *Top* modules affect performance compared to monolithic SOTA architectures?
- **RQ2:** Can sequential modeling in the *Tower* layer (e.g., via RNNs or Attention) effectively capture task dependencies in e-commerce settings?
- **RQ3:** Does the proposed modular approach alleviate negative transfer and the seesaw phenomenon?

4.1 Baselines

To rigorously evaluate our proposed modular framework, we compare it against a comprehensive set of strong baselines. We categorize these methods based on their primary mechanism for handling multi-task relationships: representation sharing, architectural sequentiality, and probabilistic dependency modeling.

4.1.1 Bottom section

These models focus on optimizing the shared hidden representations between tasks to mitigate negative transfer and the seesaw phenomenon.

- **Shared-Bottom (SB)**: The fundamental hard-parameter sharing architecture where bottom feature extraction layers are shared across all tasks, followed by task-specific tower networks. It serves as a standard baseline to measure the benefits of advanced sharing mechanisms.
- **PLE (Progressive Layered Extraction)** (Tang et al.(2020)): An advanced extraction architecture that explicitly separates shared experts from task-specific experts to mitigate harmful parameter interference. PLE utilizes a progressive routing mechanism to extract deeper semantic knowledge.

4.1.2 Towers section

These models introduce architectural inductive biases (e.g., residual flows or recurrent units) to explicitly model the sequential nature of the tasks (e.g., *Impression* $\rightarrow$ *Click* $\rightarrow$ *Conversion*).

- **SEQ+MD** (Wang et al.(2024)): A framework that treats multi-task learning as a sequence generation problem using Recurrent Neural Networks (RNNs), rather than parallel outputs. We omit the multi-distribution (MD) module in order to make a fair comparison to other models.

4.1.3 Top section

These models focus on capturing the causal or probabilistic dependencies between target events, often to address specific challenges like Sample Selection Bias (SSB) and Data Sparsity (DS) in conversion funnels.

- **DBMTL (Deep Bayesian Multi-Target Learning)** (Wang et al.(2019)): A framework that models target events as a Bayesian network. Directed links between tasks are parameterized by hidden layers and learned from data, allowing the model to capture arbitrary causal relationships among targets.

- **AITM (Adaptive Information Transfer Multi-task)** (Xi et al.(2021)): This model addresses the sequential dependence among multi-step conversions by employing an Adaptive Information Transfer (AIT) module. The AIT module dynamically learns what and how much information should be transferred from a former task step to a latter one to improve end-to-end conversion estimation.

4.2 Datasets

We utilize benchmarks that capture the interplay between text relevance and user behavior.

4.2.1 Public AliExpress (AE) dataset

We utilize the public **AliExpress (AE)** dataset (Li et al.(2020)), which is derived from real-world traffic logs of the Alibaba search system. This dataset is commonly used for multi-task learning benchmarks in e-commerce (Fu et al.(2024); Yuan et al.(2025); Zou et al.(2022)). It contains user behavior logs with two sequential actions: *Click* and *Conversion*. The dataset is split into five subsets partitioned by country of origin: Russia (RU), Spain (ES), France (FR), Netherlands (NL), and the United States (US). Table 2 summarizes the statistics of the datasets. In order to ensure the robustness of our method and to follow the widely used convention (Fu et al.(2024); Li et al.(2020); Dai et al.(2025); Yuan et al.(2025); Zou et al.(2022)), we report metrics separately for each subset.

The datasets for each country are further split into two subsets: train and test. For the purposes of parameter hypertuning, we set aside 10% of listings randomly chosen for validation.

4.2.2 Proprietary Allegro dataset

TBD

4.3 Experimental Procedure

To ensure a rigorous and equitable comparison across the architectures presented in Table 1, we implemented a standardized experimental protocol. We conducted hyperparameter optimization using the training and validation subsets. Specifically, sixty (60) distinct model instances were trained for each architecture, with each instance utilizing a unique set of hyperparameters. The instance yielding the minimum validation loss was identified as the optimal model. This selected model was subsequently evaluated on an independent test set to determine final performance metrics. Detailed specifications regarding the hyperparameter search spaces, along with the specific configurations of the best-performing models, are provided in Appendix A. To ensure reproducible results final models were trained with fixed known seed and all hyperparameters of model architectures were saved in configuration files. For training the combinations of the architectures we select the architectures with the best results in parameter hypertraining and train the model from scratch.

4.4 Loss Function

We optimize the model parameters by minimizing a joint loss function L, defined as the weighted sum of the individual losses for each task. Formally, the global objective is given by:

$$L = \sum_{k \in \mathcal{T}} \lambda_k \mathcal{L}_k \tag{12}$$

where $\mathcal{T}$ represents the set of tasks, $\mathcal{L}_k$ is the loss function for task k, and λ_k is a scalar hyperparameter balancing the contribution of each task. In our experiments, we assign equal importance to all tasks, setting $\lambda_k = 1$ for all k.For the binary classification tasks considered in this study (CTR, ATCR, CVR), we employ the Binary Cross-Entropy (BCE) loss:

$$\mathcal{L}k = -\frac{1}{N} \sum_{i=1}^{N} \left(y_{i,k} \log(\hat{y}_{i,k}) + (1 - y_{i,k}) \log(1 - \hat{y}_{i,k}) \right) \tag{13}$$

where $y_{i,k} \in \{0, 1\}$ is the ground truth label and $\hat{y}_{i,k}$ is the predicted probability for the i-th sample on task k.The specific task sets $\mathcal{T}$ for each dataset are defined as follows:

- **AliExpress (AE) Dataset:** The model is optimized for two sequential tasks: Click-Through Rate (CTR) and Conversion Rate (CVR). Thus, $\mathcal{T}_{AE} = \{\text{CTR}, \text{CVR}\}$.

Table 2: Statistics and probability of clicks given impression (CTR), purchases given clicks (CVR) and purcheses given impressions (CTCVR) of AliExpress Datasets.

Subset	# Impressions	# Clicks	# Purchases	CTR (%)	CVR (%)	CTCVR (%)
AE-RU	129,919,753	3,618,492	61,898	2.79	1.72	0.05
AE-ES	31,669,427	842,055	19,096	2.66	2.27	0.07
AE-FR	27,035,601	542,753	14,430	2.01	2.66	0.06
AE-NL	17,717,195	381,078	13,815	2.16	3.63	0.08
AE-US	27,392,613	449,608	10,830	1.65	2.41	0.04

Table 3: Comparison of the performance of the baseline architectures on AliExpress datasets. All the results are presented as the relative improvement against the baseline *Shared Bottom* architecture. The best result is bolded, whilst second best result is underlined.

Model	AE-RU		AE-ES		AE-FR		AE-NL		AE-US	
	CTR	CVR	CTR	CVR	CTR	CVR	CTR	CVR	CTR	CVR
Shared Bottom	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
PLE	-	-	-	-	-	-	-	-	0.17%	-0.37%
SEQ+MD	-	-	-	-	-	-	-	-	**0.51%**	<u>0.40%</u>
DBMTL	-	-	-	-	-	-	-	-	-	-
AITM	-	-	-	-	-	-	-	-	<u>0.46%</u>	0.24%
Task Attention	-	-	-	-	-	-	-	-	0.32%	**0.59%**

- **Allegro Dataset:** The model extends to three sequential tasks, including the intermediate Add-To-Cart action: Click-Through Rate (CTR), Add-To-Cart Rate (ATCR), and Conversion Rate (CVR). Thus, $\mathcal{T}_{Allegro} = \{\text{CTR}, \text{ATCR}, \text{CVR}\}$.

Depending on the choice of the *Top* architecture (e.g., ESMM), the specific formulation of the CVR and ATCR components may involve entire-space estimation (CTCVR, CTATCR) to handle sample selection bias (Ma et al.(2018a)). However, the overarching training objective remains the aggregation of these task-specific signals.

4.5 Evaluation Metrics

Following standard practices in industrial recommendation, we use **AUC** (Area Under the ROC Curve) as the primary evaluation metric for CTR (Click-Through Rate), ATCR (Add-To-Cart Rate) and CVR (Conversion Rate) tasks. AUC measures the probability that a randomly chosen positive sample is ranked higher than a randomly chosen negative sample.

4.6 Results

Table 4 presents the performance of the baseline models on the AliExpress and Allegro dataset. ***Conclusions will be updated when all data is available.***

4.7 Exploratory Data Analysis

An analysis of the AliExpress datasets reveals the severe challenges inherent in modeling user conversion behaviors due to extreme class imbalance and regional heterogeneity.

Extreme Class Imbalance and Data Sparsity. As shown in Table 2, the signal for the primary target, purchase, is exceptionally scarce across all subsets. The Click-Through and Conversion Rate (CTCVR) ranges from a mere 0.04% (AE-US) to 0.08% (AE-NL). This indicates that the positive class (purchase) constitutes a negligible fraction of the sample space, creating a "needle in a haystack" scenario where the model must distinguish rare conversion events from massive amounts of negative feedback. Furthermore, comparing the # Clicks to # Purchases highlights the Data Sparsity (DS) problem; for example, in the AE-US subset, despite over 27 million impressions, there are only roughly 10,000 purchases, making the fitting of conversion models difficult due to limited positive supervision.

Regional Heterogeneity. The data exhibits significant inconsistencies in class proportions across different regions, confirming the presence

Table 4: Comparison of the performance of the baseline architectures on proprietary Allegro dataset. All the results are presented as the relative improvement against the baseline *Shared Bottom* architecture.The best result is bolded, whilst second best result is underlined.

Model	CTR	ATCR	CVR
Shared Bottom	0.00%	0.00%	0.00%
PLE	-	-	-
SEQ+MD	-	-	-
DBMTL	-	-	-
AITM	-	-	-
Task Attention	-	-	-

Table 5: Comparison of the performance of the combinations of architectures on AliExpress datasets. All the results are presented as the relative improvement against the baseline *Shared Bottom* architecture.The best result is bolded, whilst second best result is underlined.

Model	**AE-RU**		**AE-ES**		**AE-FR**		**AE-NL**		**AE-US**	
	CTR	CVR	CTR	CVR	CTR	CVR	CTR	CVR	CTR	CVR
SB + IT	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
PLE + IT	-	-	-	-	-	-	-	-	-	-
PLE + IT + AITM	-	-	-	-	-	-	-	-	-	-
PLE + SEQ+MD + DBMTL	-	-	-	-	-	-	-	-	-	-
PLE + SEQ+MD + AITM	-	-	-	-	-	-	-	-	-	-
AITM	-	-	-	-	-	-	-	-	-	-
SB + TA + AITM	-	-	-	-	-	-	-	-	-	-
SB + TA + DBMTL	-	-	-	-	-	-	-	-	-	-

Table 6: Comparison of the performance of the combinations of architectures on proprietary Allegro dataset. All the results are presented as the relative improvement against the baseline *Shared Bottom* architecture.The best result is bolded, whilst second best result is underlined.

Model	CTR	ATCR	CVR
SB + IT	0.00%	0.00%	0.00%
PLE + IT	-	-	-
PLE + IT + AITM	-	-	-
PLE + SEQ+MD + DBMTL	-	-	-
PLE + SEQ+MD + AITM	-	-	-
AITM	-	-	-
SB + TA + AITM	-	-	-
SB + TA + DBMTL	-	-	-

of multi-distribution shifts. While AE-NL demonstrates a high conversion efficiency with a CVR of 3.63%, AE-RU shows a much lower CVR of 1.72% despite having the highest volume of traffic (approx. 130 million impressions). Such variations suggest distinct shopping preferences and cultural influences in different markets, where feature importance and distributions shift drastically between regions.

Problem Hardness. The combination of extremely low positive signal (CTCVR $< 0.1\%$) and significant regional distribution shifts poses a dual challenge. The model must be robust enough to handle severe sparsity without overfitting to the majority negative class, while simultaneously possessing the flexibility to adapt to inconsistent user behaviors across different geographic domains.

4.8 Ablation study

This section will be updated when all the data and results are available.

5 Conclusions

This section will be updated when all the data and results are available.

References

Quanyu Dai, Jiaren Xiao, Zhaocheng Du, Jieming Zhu, Chengxiao Luo, Xiao-Ming Wu, and Zhenhua Dong. 2025. MCNet: Monotonic Calibration Networks for Expressive Uncertainty Calibration in Online Advertising. In *Proceedings of the ACM on Web Conference 2025 (WWW '25)*. ACM, 4408–4419. https://doi.org/10.1145/3696410.3714802doi:10.1145/3696410.3714802

Cong Fu, Kun Wang, Jiahua Wu, Yizhou Chen, Guangda Huzhang, Yabo Ni, Anxiang Zeng, and Zhiming Zhou. 2024. Residual Multi-Task Learner for Applied Ranking. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Barcelona, Spain) *(KDD '24)*. Association for Computing Machinery, New York, NY, USA, 4974–4985. https://doi.org/10.1145/3637528.3671523doi:10.1145/3637528.3671523

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 28th ACM International Conference on Information and Knowledge Management, CIKM 2020, Virtual Event, Ireland, October 19- 23,2019*. ACM, New York,NY,USA.

Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H. Chi. 2018b. 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* (London, United Kingdom) *(KDD '18)*. Association for Computing Machinery, New York, NY, USA, 1930–1939. https://doi.org/10.1145/3219819.3220007doi:10.1145/3219819.3220007

Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. 2018a. 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* (Ann Arbor, MI, USA) *(SIGIR '18)*. Association for Computing Machinery, New York, NY, USA, 1137–1140. https://doi.org/10.1145/3209978.3210104doi:10.1145/3209978.3210104

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* (Virtual Event, Brazil) *(RecSys '20)*. Association for Computing Machinery, New York, NY, USA, 269–278. https://doi.org/10.1145/3383313.3412236doi:10.1145/3383313.34122

Qi Wang, Zhihui Ji, Huasheng Liu, and Binqiang Zhao. 2019. Deep Bayesian Multi-Target Learning for Recommender Systems. *ArXiv* abs/1902.09154 (2019). https://api.semanticscholar.org/CorpusID:67856758

Siqi Wang, Audrey Zhijiao Chen, Austin Clapp, Sheng-Min Shih, and Xiaoting Zhao. 2024. SEQ+MD: Learning Multi-Task as a SEQuence with Multi-Distribution Data. *ArXiv* abs/2408.13357 (2024). https://api.semanticscholar.org/CorpusID:271957174

Dongbo Xi, Zhen Chen, Peng Yan, Yinger Zhang, Yongchun Zhu, Fuzhen Zhuang, and Yu Chen. 2021. Modeling the Sequential Dependence among Audience Multi-step Conversions with Multi-task Learning in Targeted Display Advertising. arXiv:2105.08489 [cs.AI] https://arxiv.org/abs/2105.08489

Jun Yuan, Guohao Cai, and Zhenhua Dong. 2025. A Parameter Update Balancing Algorithm for Multi-task Ranking Models in Recommendation Systems. arXiv:2410.05806 [cs.IR] https://arxiv.org/abs/2410.05806

Xinyu Zou, Zhi Hu, Yiming Zhao, Xuchu Ding, Zhongyi Liu, Chenliang Li, and Aixin Sun. 2022. Automatic Expert Selection for Multi-Scenario and Multi-Task Search. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*. ACM, 1535–1544. https://doi.org/10.1145/3477495.3531942doi:10.1145/3477495.35319

A Model Hyperparameters

Ranges of parameter values tested during parameter hypertuning on US subset of AliExpress dataset and proprietary Allegro dataset are presented in table 7.

Table 7: Ranges of parameters hypertuned.

Parameter	**Range**
Optimizer	
Learning rate	$[10^{-5}, 10^{-1}]$
Bottom section	
Shared	
MLP layer sizes	$\{(128, 64, 32, (64, 64, 64), (64, 64, 32)$ $(64, 32, 32), (64, 64), (64, 32), (64)\}$
Dropout	$[0.0, 0.5]$
Batch normalization	$\{False, True\}$
PLE	
Experts per task	$\{4, 8, 10\}$
Shared experts	$\{2, 4, 8\}$
Number of layers	$\{2, 3, 4\}$
Expert output dimension	$\{32, 64, 128\}$
Expert dropout	$[0.0, 0.5]$
Expert batch normalization	$\{False, True\}$
Towers section	
Independent	
MLP layer sizes	$\{(64, 64, 64), (64, 64, 32), (64, 64),$ $(64, 32), (32, 32)\}$
Dropout	$[0.0, 0.5]$
Batch normalization	$\{False, True\}$
SEQ	
Number of layers	$\{2, 3, 4\}$
Hidden size	$\{32, 64, 128\}$
Task Attention	
Model dimension	$\{32, 64, 96\}$
Number of heads	$\{2, 4, 8\}$
Number of layers	$\{2, 3, 4\}$
MLP ratio	$\{1, 2\}$
Dropout	$[0.0, 0.5]$
Top section	
DBMTL	
MLP layer sizes	$\{24, 32, 64\}$
Dropout	$[0.0, 0.5]$
AITM	
Information module dimension	$\{24, 32, 64, 96\}$