Bottoms, Towers and Tops: Evaluating Component Interactions in Multi-Task Learning for E-Commerce

Piotr Bajger[1,*,†], Aleksander Wawer[1,2,†] and Wojciech Kutak[1,†]

[1]*Allegro sp. z o.o., ul. Wierzbięcice 1B, Poznań, 61-659, Poland*
[2]*Institute of Computer Science, Polish Academy of Sciences, Jana Kazimierza 5, Warsaw, 01-248 Poland*

Abstract

In industrial e-commerce ranking, multi-task learning (MTL) is the standard approach for modeling the sequential user journey. The majority of literature focuses on the design of specialized modules to refine input representations and label dependencies. These modules are often studied in isolation which makes it difficult to assess how they interact with each other. To address this problem we introduce the Bottom-Tower-Top (BTT) framework, which decouples MTL architectures into representation extraction, task-specific refinement, and label dependency modeling. We examine the literature to identify the state of the art techniques for each stage, as well as propose how transformer-based modules may be adapted to MTL problems by introducing two components: Multi-Task Transformer (MT2) for representation learning and Cross-Task Towers (XTT) for task-specific refinement. We perform 900 hyperparameter tuning trials in an empirical evaluation of 15 different configurations on a public AliExpress dataset with two labels: click and conversion. Our results demonstrate that specific synergies — such as pairing the MT2 bottom with sequential routing — yield a 0.85% relative improvement in click-through rate AUROC. In addition, leveraging the sequential dependency of the tasks in user journey may lift both click and purchase metrics. At the same time, the simplicity of a shared bottom remains a strong baseline and many architectural combinations exhibit the seesaw effect of negative knowledge transfer.

Keywords

e-commerce, machine learning, multi-task learning, CTR prediction

1. Introduction

In the context of information retrieval for industrial e-commerce platforms, the objective has shifted from simple relevance ranking to a simultaneous optimization of multiple performance metrics. Search engines and recommender systems at many large-scale platforms need to be able to accurately predict a variety of user behaviors to offer optimal user experience. For this purpose Amazon [1, 2] Pinterest [3, 4], Taobao [5], AliExpress [6] and Walmart [7] all use Multi-Task Learning (MTL) in their ranking systems.

A fundamental characteristic of e-commerce data is the sparsity of the conversion signal relative to the click signal. Although click-through rate (CTR) data is abundant, conversion rate (CVR) events are typically orders of magnitude less frequent [5]. Standalone CVR models often struggle with convergence due to a limited number of positive examples. MTL aims to mitigate this problem via knowledge transfer between tasks. The systems are designed so that the sparser task (e.g. conversion) model can use knowledge from the denser task (e.g. click) model to improve generalization.

Historically, the development of these architectures focused on how parameters are shared and how information is routed between tasks. Initial approaches relied on Shared Bottom [8] which, while efficient, often suffers from a "seesaw effect" where performance improvements in one task degrade the performance of other tasks. To mitigate this, gating mechanisms were introduced in the form of Multi-gate Mixture of Experts (MMoE) [9] and Progressive Layer Extraction (PLE) [10]. It is noteworthy that all these architectures modify the "bottom" part of the model, before the data are routed to task-specific layers.

ECOM'26: SIGIR Workshop on eCommerce, Jul 24, 2026, Melbourne, Australia
*Corresponding author.
†These authors contributed equally.
piotr.bajger@allegro.com (P. Bajger); aleksander.wawer@allegro.com (A. Wawer); wojtek.kutak@gmail.com (W. Kutak)
0000-0002-9487-8476 (P. Bajger); 0000-0002-7081-9797 (A. Wawer)

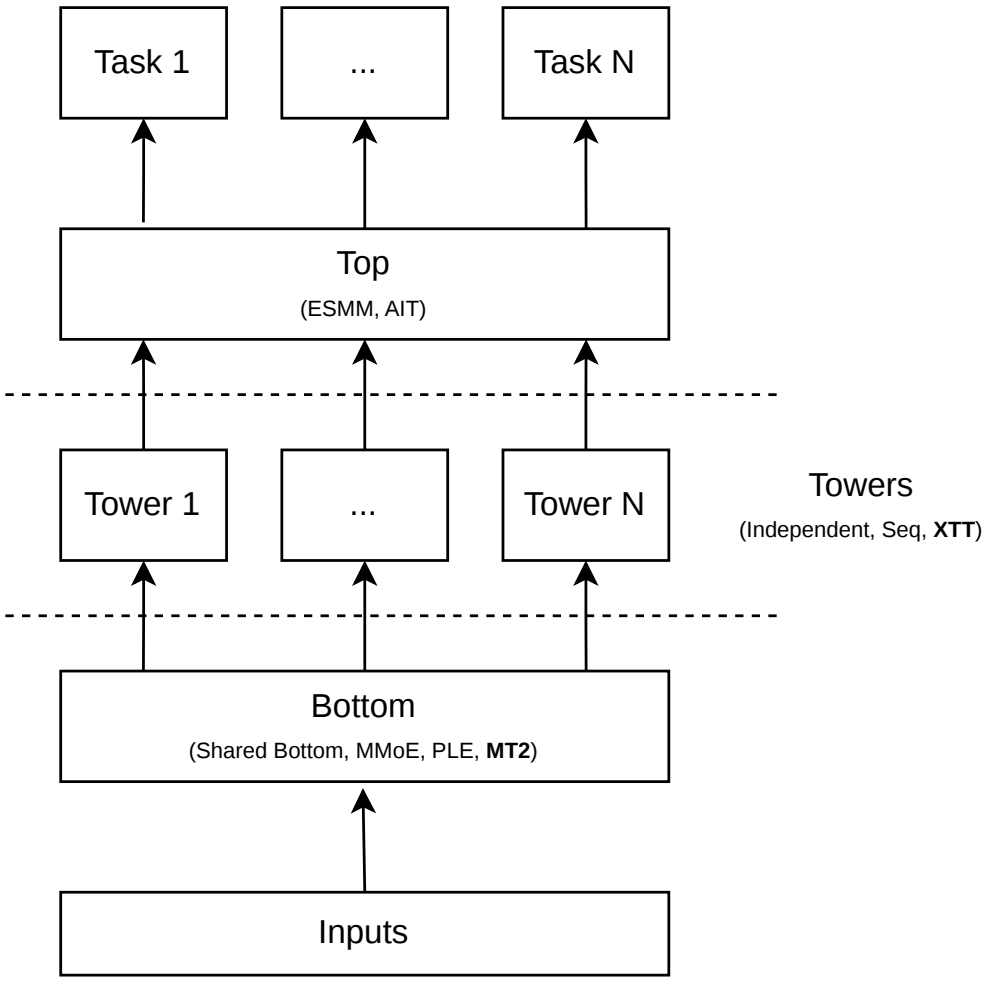

Figure 1: Generic representation of an MTL model. Inputs are passed through a Bottom model (e.g. Shared Bottom or MMoE). The output of the bottom layer is processed via task-specific Towers (Independent, SEQ). The Towers' outputs are then passed to a Top layer (e.g. ESMM or AIT) which outputs final probabilities.

A related research direction focuses on structural dependencies in an e-commerce funnel. Typically the tasks learned in e-commerce settings are sequential: click → add-to-cart → purchase. The Entire Space Multi-task Modelling (ESMM) [5], Adaptive Information Transfer (AIT) [6] and Deep Bayesian Multi-Target Learning (DBMTL) [11] all leverage this sequential structure. These approaches act on the outputs of task-specific layers to calculate the final probabilities of click and conversion. Notably, Sequential Multi-Distribution (SEQ+MD) [12] explicitly uses recurrent neural networks to model this sequence of actions.

Despite these advancements, these methodologies are frequently evaluated in isolation, often using proprietary datasets and platform-specific heuristics. This fragmented landscape makes it difficult to identify synergies between different components of a multi-task model.

Consequently, there is a lack of a unified framework to categorize and compare these diverse approaches. In this paper, we address this gap by proposing a structural taxonomy that decomposes MTL systems into three distinct modules: Bottom, Towers, and Top.

1. The Bottom layer focuses on representation learning and expert-based feature extraction.

2. The Towers facilitate task-specific specialization.
3. The Top layer manages the final multi-objective fusion and label relationship modeling.

The generic Bottom-Towers-Top (BTT) model under our proposed framework is shown in Figure 1. We observe that many MTL solutions proposed in the literature may be classified into Bottom (Shared [8], MMoE [9], PLE [10]), Towers (Independent [8], SEQ [12]) and Top (ESMM [5], AIT [6], DBMTL [11]).

By abstracting these systems into a modular hierarchy, we provide a clearer understanding of the design space and facilitate the study of cross-module interactions. Specifically, we investigate how architectural choices within a given layer influence the performance and behavior of others, exploring the relationships across the entire pipeline. Building upon this taxonomy, we further investigate how transformer-based architectures compare against pre-attention models for e-commerce dataset.

2. Methods

2.1. The Bottom-Towers-Top framework

This section describes the Bottom-Towers-Top (BTT) framework in greater detail. We argue that a systematic decomposition of existing Multi-Task Learning (MTL) approaches reveals potential architectural synergies that are often overlooked when models are viewed in isolation. We categorize the existing MTL literature into three functional modules: **Bottom**, **Towers**, and **Top**. This modularization allows us to examine the interaction between different representation learning techniques and task-specific structures. Building upon this framework, we propose how attention-based models may be adapted to the MTL problem.

2.2. Bottom modules

The Bottom modules aim to facilitate knowledge transfer between the tasks in order to learn effective representation of raw input features. Depending on the architecture bottom modules may return a single shared representation, or a set of N task-specific representations. Below we discuss the key Bottom modules from the literature and propose a simple adaptation of transformer architecture to an MTL problem: the Multi-Task Transformer (MT2) module.

Shared Bottom. [8] This represents the classic hard parameter sharing approach. All tasks share a single set of hidden layers (typically a Multi-Layer Perceptron), which produces a unified representation. While computationally efficient, it is susceptible to "task interference" when objectives are loosely correlated.

PLE [10]: Progressive Layered Extraction refines the mixture-of-experts concept [9] by explicitly separating the expert pool into task-specific experts and shared experts. Organized in a multi-level structure, PLE utilizes a "Customized Gate Control" mechanism that allows task-specific layers to absorb information from both their dedicated experts and the shared experts, effectively mitigating the "seesaw effect" where performance on one task improves at the expense of another.

2.2.1. Multi-Task Transformer (MT2)

Inspired by [13] we propose Multi-Task Transformer (MT2), a Bottom module which adapts the attention mechanism to produce task-specific input representations. Unlike traditional expert-based architectures, MT2 leverages the self-attention mechanism to treat features and task objectives as interacting tokens in a shared latent space.

Feature Tokenization. MT2 first maps heterogeneous input features into a unified sequence of embeddings $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_F]$, where F is the number of feature fields and each $\mathbf{e}_j \in \mathbb{R}^{d_{model}}$. The features are processed according to the following rules:

- **Discrete Features:** Categorical attributes are embedded using lookup tables to produce d_{model}-sized vectors.

- **Continuous Features:** Numerical values are projected into the embedding space via field embeddings [14], where the scalar value is multiplied by a learnable field-specific vector of size d_{model}.

Task Tokens Following [13] we introduce a set of N learnable task tokens $\mathbf{TSK}_1$, $\mathbf{TSK}_2$, ..., $\mathbf{TSK}_N$, where N corresponds to the number of objectives These tokens serve as task-specific "probes" and are prepended to the feature sequence. The input to the first transformer block is thus:

$$\mathbf{Z}^{(0)} = [\mathbf{TSK}_1, \ldots, \mathbf{TSK}_N, \mathbf{e}_1, \ldots, \mathbf{e}_F] \in \mathbb{R}^{(N+F)\times d_{model}}$$

Transformer Processing and Representation Extraction. The sequence $\mathbf{Z}^{(0)}$ is processed through L layers of a Transformer encoder. Through self-attention, each task token $\mathbf{TSK}_k$ aggregates relevant information from the entire feature set and potentially from other task tokens. This allows the model to learn context-aware representations where the interaction between features is conditioned on the specific task requirements.

After L layers, we extract the updated task tokens from the final hidden state:

$$\mathbf{Z}_{1:N}^{(L)} = [\mathbf{h}_1^{(L)}, \mathbf{h}_2^{(L)}, \ldots, \mathbf{h}_N^{(L)}]$$

These N refined vectors $\mathbf{h}_k$ serve as the task-specific input representations for the subsequent Tower modules. By utilizing this token-based approach, MT2 avoids the bottleneck of a single shared representation and enables knowledge transfer between tasks.

2.3. Tower modules

The Tower modules receive the representations generated by the Bottom layer and perform task-specific non-linear transformations. In our BTT framework, Towers represent the intermediate stage where features are specialized before final label prediction. Below we discuss two approaches from the literature and suggest how a transformer-based architecture can be used as the Towers module.

Independent Towers: The predominant approach in MTL literature. Each task is assigned a parallel feed-forward network. While computationally efficient and simple to implement, these towers operate in isolation and cannot capture semantic or sequential correlations between different stages of the user journey.

SEQ Towers [12]: The Sequential + Multi-Distribution (SEQ+MD) architecture introduces a refinement to the tower structure to model the user funnel (e.g., Click $\rightarrow$ Purchase). This allows downstream towers to consume a "state" vector from preceding stages. While the paper introduces also other improvements, our focus will be on the sequential towers.

2.3.1. Cross-Task Towers (XTT)

The Cross-Task Towers (XTT) module is an attention-based relational layer designed to model the dependencies between task-specific representations. Given a set of N task vectors $\mathbf{h}_i \in \mathbb{R}^{d_{in}}$ produced by the Bottom module we first project these features into a shared d-dimensional latent space. That is, for task $i \in \{1, \ldots, N\}$ we have:

$$\mathbf{x}_i^{(0)} = \mathbf{W}_i \mathbf{h}_i + \mathbf{b}_i, \tag{1}$$

where $\mathbf{W} \in \mathbb{R}^{d\times d_{in}}$ and $b_i \in \mathbb{R}^d$ are learnable, task-specific parameters. The resulting vectors are stacked into a vector $\mathbf{X}^{(0)} = [\mathbf{x}_1^{(0)}, \ldots, \mathbf{x}_N^{(0)}] \in \mathbb{R}^{N\times d}$ which serves as the input to a Transformer encoder consisting of L layers. Each layer l performs a multi-head self-attention operation:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}} + \mathbf{M}\right)\mathbf{V} \tag{2}$$

In this formulation, the queries $\mathbf{Q}$, keys $\mathbf{K}$, and values $\mathbf{V}$ are linear transformations of the task representations from the preceding layer. The matrix $\mathbf{M} \in \mathbb{R}^{N\times N}$ represents an optional causal

mask. By setting $\mathbf{M}$ to a strictly lower triangular matrix, XTT enforces the sequential logic of the user journey, ensuring that downstream conversion tasks are conditioned only on information from upstream engagement tasks.

2.4. Top modules

The Top modules operate on the refined task vectors to enforce structural label dependencies. These are particularly critical in e-commerce, where user actions follow a sequential funnel (e.g., Impression $\rightarrow$ Click $\rightarrow$ Purchase). Unlike the Towers, which focus on feature specialization, Top modules ensure that the final probability estimates adhere to the causal constraints of the target space.

ESMM [5]: This is the simplest form of probability transfer for sequential tasks where the scalar outputs of the towers are multiplied cumulatively. For example, the output of the "Purchase" tower is interpreted as the post-click conversion probability. ESMM multiplies this output by the probability of click to obtain the conversion probability: $P(purchase) = P(purchase|click)P(click)$.

AIT [6]: For any task i in the funnel, an Adaptive Information Transfer (AIT) unit aggregates the current tower representation with the "transition" representation from the preceding stage. The information flow is governed by a gating function and lightweight attention.

2.5. Experiments

To evaluate the effectiveness of the proposed modular architecture decomposition (Bottom, Tower, Top) and measure the inter-architectural dependencies, 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 of MTL model?
- **RQ2:** Do attention-based modules outperform other modules for sequential tasks in e-commerce setting?
- **RQ3:** Are there any modules capable of consistently mitigating the problem of negative transfer (the "seesaw effect") across different architecture combinations?

2.6. Dataset

	AliExpress
#Total	223.7M
#Clicked	5.8M
#Purchased	120k
CTR	2.50%
CVR	0.05%
Post-click CVR	2.06%
#Cts. features	47
#Cat. features	9

Table 1
Summary statistics for the public AliExpress US dataset [15].

We perform the experiments on the publicly available AliExpress e-commerce dataset [15]. Summary statistics for this dataset are shown in Table 1. This data was derived from real-world traffic logs of the Alibaba search system. This dataset is commonly used for multi-task learning benchmarks in e-commerce [16, 17, 18]. It contains user behavior logs with two sequential actions: *Click* and *Conversion*. The data set is divided into five subsets partitioned by country of origin: Russia (RU), Spain (ES), France (FR), the Netherlands (NL), and the United States (US). We restricted our analysis to the US dataset as it is the largest of the four.

The original dataset is split into train and test subsets. In order to perform rigorous evaluation, we split the train dataset further by setting aside randomly selected 10% of the train dataset for validation.

2.7. Feature Representation and Preprocessing

The AliExpress dataset has two distinct types of features: continuous and discrete. To accommodate the architectural requirements of both MLP-based and Transformer-based Bottom modules, we apply the following preprocessing strategies:

Continuous Features. For the Shared Bottom and PLE architectures, numerical features are passed directly to the model without additional preprocessing. In contrast, for the MT2 architecture, each continuous feature x_j is mapped to a d_{model}-dimensional latent space using field embeddings.

Categorical Features. All categorical attributes are mapped to integer indices and embedded via learnable lookup tables.

Following these transformations, the feature representations are either concatenated into a single dense vector (Shared and PLE Bottom modules) or treated as a sequence of feature tokens (for the MT2 Bottom module) before being passed to the respective representation learning stage.

2.8. Hyperparameter Optimization

To ensure a rigorous and fair comparison across all architectural combinations within the BTT framework, we conducted a systematic hyperparameter search for 15 different configurations listed in Table 2. For each combination of Bottom, Tower, and Top modules, we executed 60 hyperparameter tuning trials using the Google Cloud Platform (GCP) Vertex AI Hyperparameter Tuning service, yielding a total of 900 training runs. The hyperparameter search was conducted using the service's default Bayesian optimization algorithm.

The optimization objective for the tuning process was defined as the maximization of CVR AUROC on the validation set. The models were trained for 14 epochs. Upon completion of the search, we selected the trial that achieved the highest validation performance for each architecture. The final metrics reported in Table 2 were calculated by evaluating these optimal configurations on the test set.

3. Results and Analysis

We evaluate the proposed Bottom-Towers-Top (BTT) framework on the AliExpress-US dataset, exploring the interplay between different representation layers, task-specific towers, and probability-modeling top modules. We use the shared bottom with independent towers configuration as our experimental baseline, as it represents the most basic implementation of Multi-Task Learning. Detailed performance metrics and relative improvements (Δ) are reported in Table 2, with aggregated factor analysis provided in Figure 2.

3.1. Main Effects Analysis

An analysis of the aggregated main effects across all experimental runs reveals several key trends regarding the impact of each tier in the BTT hierarchy.

Towers. The Choice of tower architecture significantly impacts engagement metrics. The SEQ tower configuration yields the highest relative improvement in CTR AUROC (+0.37% on average). Our proposed attention-based XTT towers also demonstrate a positive mean impact on CTR (+0.11%), though they appear more sensitive to the choice of Bottom module. Interestingly, Independent towers, while lagging in CTR, remain competitive for the more sparse purchase prediction task.

Top Modules. The AIT top module shows the strongest mean performance for CTR AUROC, providing a +0.21% relative gain over other approaches. In contrast, the ESMM module consistently underperforms in this framework, exhibiting a relative decrease of 0.17% in CTR and nearly 1.0% in CVR AUROC. This suggests that the rigid multiplicative constraints of ESMM may be less effective than the adaptive gating found in AIT, particularly when paired with more complex bottom architectures. Notably, for the purchase task (CVR), omitting a top module altogether (—) resulted in the highest mean stability.

Table 2
AliExpress-US: Test AUROC results for 15 different architectural combinations in the BTT framework.

Bottom	Towers	Top	CTR AUROC	Δ CTR	CVR AUROC	Δ CVR
MT2	XTT	—	0.7097	+0.32%	0.8727	+0.51%
MT2	XTT	AIT	0.7089	+0.20%	0.8619	-0.73%
MT2	Independent	—	0.7066	-0.12%	0.8644	-0.44%
MT2	Independent	ESMM	0.7004	-1.00%	0.8642	-0.47%
MT2	SEQ	AIT	0.7135	+0.85%	0.8682	-0.00%
PLE	XTT	ESMM	0.7085	+0.14%	0.8437	-2.82%
PLE	Independent	—	0.7067	-0.12%	0.8633	-0.57%
PLE	Independent	AIT	0.7084	+0.12%	0.8666	-0.19%
PLE	SEQ	—	0.7109	+0.47%	0.8708	+0.30%
Shared	XTT	—	0.7068	-0.10%	0.8696	+0.16%
Shared	XTT	AIT	0.7082	+0.09%	0.8668	-0.17%
Shared	Independent	—	0.7075	—	0.8682	—
Shared	Independent	AIT	0.7089	+0.20%	0.8753	+0.81%
Shared	SEQ	AIT	0.7061	-0.19%	0.8573	-1.25%
Shared	SEQ	ESMM	0.7100	+0.35%	0.8710	+0.32%

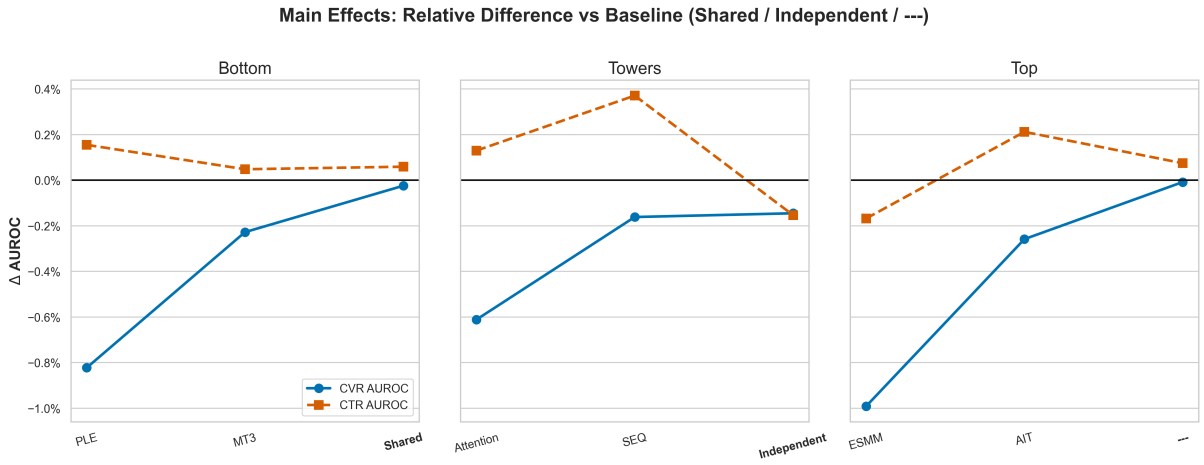

Figure 2: Main effects plot showing the relative difference in AUROC (%) for each factor level compared to the baseline configuration (Shared bottom, Independent towers, no top model). Levels are ordered by increasing CVR AUROC. The solid line with circles represents CVR AUROC and the dashed line with squares represents CTR AUROC.

Bottom Modules. At the representation level, PLE provides the highest relative mean gain for CTR (+0.16%), while the Shared bottom remains slightly more robust for the sparse purchase task. The proposed MT2 architecture remains a strong contender, particularly when paired with sequential or attention-based towers rather than independent MLPs.

3.2. Architectural Synergies

The results in Table 2 demonstrate that the highest gains are achieved by identifying synergetic combinations across the BTT tiers rather than optimizing modules in isolation.

The optimal configuration for engagement prediction (MT2-SEQ-AIT) achieves a +0.85% relative improvement in CTR AUROC over the baseline. This suggests that the high-order feature interactions captured by the MT2 bottom are best utilized when passed through a sequential tower that accounts for the user funnel, further refined by the AIT top module.

For conversion prediction, the best performing model (Shared-Independent-AIT) reaches a +0.81%

relative gain in CVR AUROC. This highlights that while complex transformers are effective for dense signals like clicks, conversion prediction can benefit from simpler feature sharing combined with specialized top-level probability transfer.

Finally, we observe that the combination of MT2 and XTT without any top-level label modeling achieves a significant dual gain: **+0.32% in CTR** and **+0.51% in CVR** relative to the baseline. This confirms the efficacy of XTT in capturing relational task dependencies within the latent space, potentially reducing the need for explicit probability transfer modules like ESMM.

4. Discussion

Our results point at the importance of finding synergistic combinations across the BTT tiers. For engagement prediction, the optimal configuration (MT2-SEQ-AIT) improves CTR AUROC by +0.85% over the baseline. Conversely, the best model for conversion prediction (Shared + Independent + AIT) yields a +0.81% gain in CVR AUROC. It appears that dense signals like clicks require complex feature extractors, sparse conversion tasks benefit more from simpler feature sharing paired with specialized top-level probability transfer. Notably, combining MT2 and XTT without any top-level label modeling produces a strong dual gain (+0.32% CTR, +0.51% CVR).

Regarding **RQ1** and how different combinations of *Bottom*, *Tower*, and *Top* modules affect performance, our results demonstrate that architectural complexity is not additive. Many existing studies rely on simple feedforward independent towers, often ignoring how they interact with the rest of the network. Our analysis shows that a module performance is affected by other modules in the pipeline. For example, high-capacity bottom layers (like MT2 or PLE) are designed to capture a wide variety of patterns in the data. However, our results show that simply extracting these rich patterns is not enough: both MT2-Independent and PLE-Independent variants perform below the baseline in both CTR and CVR metrics. When paired with towers which leverage the sequentiality of the tasks (like XTT or SEQ), these advanced bottom layers deliver accurate predictions.

As for **RQ2** and the question whether attention-based modules outperform other module types for sequential tasks, the answer is that attention-based towers (XTT) do not universally outperform other modules. They only achieve state-of-the-art performance when decoding high-capacity representations, such as the MT2 bottom. However, when applied atop a basic Shared MLP bottom, performance stagnates. We hypothesize this may occur because the shared bottom already forces all tasks into the exact same latent space. Lateral routing on this simple foundation results in a mismatched capacity.

Finally, regarding **RQ3**, there is no single module that eliminates the seesaw effect across all architectures. For instance, while PLE with independent towers was designed specifically to tackle task interference by separating shared and task-specific experts, our results indicate that it actually underperforms when compared to a standard Shared bottom baseline. Ultimately, mitigating negative transfer requires the entire architecture to operate in an optimal manner. Simple networks benefit most from adaptive transfer mechanisms (e.g., Shared + Independent + AIT), whereas high-capacity networks perform best with unconstrained lateral routing (e.g., MT2 + XTT).

5. Conclusions

In this paper, we introduced the Bottom-Towers-Top (BTT) framework to systematically analyze the architectural interactions across multi-task learning pipelines in e-commerce ranking. Our empirical study on the AliExpress-US dataset shows that architectural complexity is not additive.

While our proposed Multi-Task Transformer backbone with Cross-Task Towers (MT2-XTT) is highly effective, yielding a +0.32% CTR and +0.51% CVR improvement over the baseline, stacking the AIT top module onto this combination causes performance to drop to +0.20% in CTR and a significant -0.73% in CVR. Similarly, while advanced configurations like MT2-SEQ-AIT can yield up to a +0.85% relative improvement in CTR AUROC, the simplicity of a shared bottom remains a remarkably robust baseline for downstream conversion predictions. It is worth noting that the best-performing architecture in

terms of CVR (+0.81%) is the relatively simple Shared-Independent-AIT configuration, which utilizes two out of three "baseline" components.

Ultimately, our findings demonstrate that successful multi-task learning in e-commerce depends on functional alignment across the BTT tiers rather than the isolated complexity of individual modules. In future research, we will investigate the specific data dynamics that dictate these architectural interactions instead of focusing solely on the development of isolated, high-capacity components.

Declaration on Generative AI

During the preparation of this work, the authors used Gemini 3 in order to: Generate literature review, Improve writing style, Paraphrase and reword, Grammar and spelling check. After using these tools/services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] M. Momma, A. B. Garakani, Y. Sun, Multi-objective relevance ranking, https://www.amazon.science/publications/relevance-ranking, 2019. Amazon Science Publications.

[2] D. Carmel, E. Haramaty, A. Lazerson, L. Lewin-Eytan, Multi-objective ranking optimization for product search using stochastic label aggregation, in: Proceedings of The Web Conference 2020, WWW '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 373–383. URL: https://doi.org/10.1145/3366423.3380122. doi:10.1145/3366423.3380122.

[3] X. Xia, P. Eksombatchai, N. Pancha, D. D. Badani, P.-W. Wang, N. Gu, S. V. Joshi, N. Farahpour, Z. Zhang, A. Zhai, Transact: Transformer-based realtime user action model for recommendation at pinterest, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '23, ACM, 2023, p. 5249–5259. URL: http://dx.doi.org/10.1145/3580305.3599918. doi:10.1145/3580305.3599918.

[4] X. Xia, S. V. Joshi, K. Rajesh, K. Li, Y. Lu, N. Pancha, D. D. Badani, J. Xu, P. Eksombatchai, Transact v2: Lifelong user action sequence modeling on pinterest recommendation, 2025. URL: https://arxiv.org/abs/2506.02267. arXiv:2506.02267.

[5] 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, SIGIR '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 1137–1140. URL: https://doi.org/10.1145/3209978.3210104. doi:10.1145/3209978.3210104.

[6] 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, 2021. URL: https://arxiv.org/abs/2105.08489. arXiv:2105.08489.

[7] X. Wu, A. Magnani, S. Chaidaroon, A. Puthenputhussery, C. Liao, Y. Fang, A multi-task learning framework for product ranking with bert, in: Proceedings of the ACM Web Conference 2022, WWW '22, ACM, 2022, p. 493–501. URL: http://dx.doi.org/10.1145/3485447.3511977. doi:10.1145/3485447.3511977.

[8] R. Caruana, Multitask learning, Kluwer Academic Publishers, USA, 1998, p. 95–133.

[9] 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, KDD '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 1930–1939. URL: https://doi.org/10.1145/3219819.3220007. doi:10.1145/3219819.3220007.

[10] 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, RecSys '20, Association for Computing Machinery, New

York, NY, USA, 2020, p. 269–278. URL: https://doi.org/10.1145/3383313.3412236. doi:10.1145/3383313.3412236.

[11] Q. Wang, Z. Ji, H. Liu, B. Zhao, Deep bayesian multi-target learning for recommender systems, 2019. URL: https://arxiv.org/abs/1902.09154. arXiv:1902.09154.

[12] S. Wang, A. Z. Chen, A. Clapp, S.-M. Shih, X. Zhao, Seq+md: Learning multi-task as a sequence with multi-distribution data, ArXiv abs/2408.13357 (2024). URL: https://api.semanticscholar.org/CorpusID:271957174.

[13] D. Sinodinos, J. Y. Wei, N. Armanfard, Multitab: A scalable foundation for multitask learning on tabular data, 2025. URL: https://arxiv.org/abs/2511.09970. arXiv:2511.09970.

[14] H. Guo, B. Chen, R. Tang, W. Zhang, Z. Li, X. He, An embedding learning framework for numerical features in ctr prediction, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery amp; Data Mining, KDD '21, ACM, 2021, p. 2910–2918. URL: http://dx.doi.org/10.1145/3447548.3467077. doi:10.1145/3447548.3467077.

[15] 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 28th ACM International Conference on Information and Knowledge Management, CIKM 2020, Virtual Event, Ireland, October 19- 23,2019, ACM, New York,NY,USA, 2020.

[16] C. Fu, K. Wang, J. Wu, Y. Chen, G. Huzhang, Y. Ni, A. Zeng, Z. Zhou, Residual multi-task learner for applied ranking, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 4974–4985. URL: https://doi.org/10.1145/3637528.3671523. doi:10.1145/3637528.3671523.

[17] J. Yuan, G. Cai, Z. Dong, A parameter update balancing algorithm for multi-task ranking models in recommendation systems, 2025. URL: https://arxiv.org/abs/2410.05806. arXiv:2410.05806.

[18] X. Zou, Z. Hu, Y. Zhao, X. Ding, Z. Liu, C. Li, A. Sun, 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, 2022, p. 1535–1544. URL: http://dx.doi.org/10.1145/3477495.3531942. doi:10.1145/3477495.3531942.