Residual Multi-Task Learner for Applied Ranking

Cong Fu
Shopee Pte. Ltd.
Singapore, Singapore
fc731097343@gmail.com

Kun Wang
Shopee Pte. Ltd.
Shanghai, China
wk1135256721@gmail.com

Jiahua Wu
Shopee Pte. Ltd.
Singapore, Singapore
gauvain.wujiahua@gmail.com

Yizhou Chen
Shopee Pte. Ltd.
Singapore, Singapore
yizhou.chen@shopee.com

Guangda Huzhang
Shopee Pte. Ltd.
Singapore, Singapore
guangda.huzhang@shopee.com

Yabo Ni
Nanyang Technological University
Singapore, Singapore
yabo001@e.ntu.edu.sg

Anxiang Zeng
SCSE, Nanyang Technological
University
Singapore, Singapore
zeng0118@e.ntu.edu.sg

Zhiming Zhou*
ECONCS†, Shanghai University of
Finance and Economics
Shanghai, China
zhouzhiming@mail.shufe.edu.cn

ABSTRACT

Modern e-commerce platforms rely heavily on modeling diverse user feedback to provide personalized services. Consequently, multi-task learning has become an integral part of their ranking systems. However, existing multi-task learning methods encounter two main challenges: some lack explicit modeling of task relationships, resulting in inferior performance, while others have limited applicability due to being computationally intensive, having scalability issues, or relying on strong assumptions. To address these limitations and better fit our real-world scenario, pre-rank in Shopee Search, we introduce in this paper ResFlow, a lightweight multi-task learning framework that enables efficient cross-task information sharing via residual connections between corresponding layers of task networks. Extensive experiments on datasets from various scenarios and modalities demonstrate its superior performance and adaptability over state-of-the-art methods. The online A/B tests in Shopee Search showcase its practical value in large-scale industrial applications, evidenced by a 1.29% increase in OPU (order-per-user) without additional system latency. ResFlow is now fully deployed in the pre-rank module of Shopee Search. To facilitate efficient online deployment, we propose a novel offline metric Weighted Recall@K, which aligns well with our online metric OPU, addressing the longstanding online-offline metric misalignment issue. Besides, we propose to fuse scores from the multiple tasks additively when ranking items, which outperforms traditional multiplicative fusion.

CCS CONCEPTS

• **Computing methodologies** → **Multi-task learning**; *Ranking*; *Neural networks*; • **Applied computing** → *Electronic commerce*; • **Information systems** → *Information retrieval*.

KEYWORDS

Multi-task learning, ranking system, e-commerce, residual learning

1 INTRODUCTION

Modern large-scale recommender systems and search engines heavily rely on modeling diverse user feedback to understand the preferences of users and better provide personalized services. Specifically, for e-commerce platforms like Meituan [31], AliExpress [15], Taobao [18], Walmart [30], and Shopee, estimating the click-through rate (CTR) and the click-through & conversion rate (CTCVR) of a user w.r.t. items have become one of their primary tasks. These metrics serve as their main indicators when ranking items.

Given that these estimation tasks are closely related and that high-commitment user behaviors exhibit significant sparsity, e.g., CTCVR is typically at the level of 0.1%, multi-task learning (MTL) [36, 39] has become an integral part of such systems to boost cross-task information interchange and mitigate the sample sparsity issue.

However, real-world large-scale e-commerce platforms have unique features that make MTL for their ranking systems special and challenging. Firstly, large-scale ranking systems commonly use a multi-stage candidate selection framework [28, 38], e.g., match → pre-rank → rank, to strike a balance between efficiency and accuracy, as depicted in Figure 1. In the early stages, such as pre-rank, the system must quickly sort through millions of items with limited resources, shortlisting a manageable number of candidates that
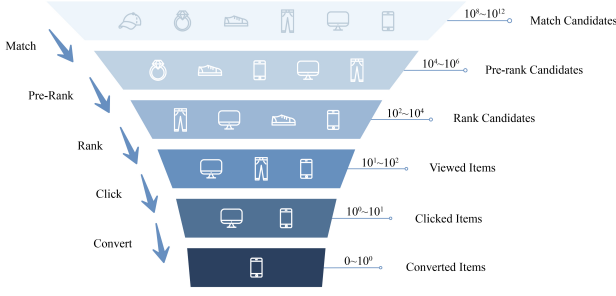
Figure 1: An illustration of the cascading item filtering process by the multi-stage candidate selection framework in large-scale applied ranking systems and subsequent user feedback. It sketches the quantities of samples at each stage, highlighting the sparsity of conversion feedback.

align well with user interests and search keywords for the next stage. This requirement poses a significant challenge to the efficiency of model inference, forbidding the employment of complex models. Secondly, user actions in e-commerce platforms are typically designed to be carried out progressively, e.g., users can only place an order after clicking the item. Effective leverage of such sequential dependence properties may have critical importance.

Among the mainstream multi-task learning methods, MMOE [17] and PLE [22] foster cross-task information sharing through expert-based network architectures. However, they lack explicit modeling of task relationships, showing relatively inferior performance [31]. Towards leveraging the sequential dependence among tasks, AITM [31] proposed to transfer information sequentially from the former task to the latter via an attention-based module. However, its high computational intensity limits its applicability in resource-constrained scenarios like pre-rank.

On the other hand, ESMM [18] proposed to use causal graphs to model sequentially dependent tasks, modeling the conditional probability of the latter step given the former. Thereafter, given the inherent sample selection bias [5] of modeling condition probability, subsequent methods including $ESCM^2$ [27] and DCMT [40] further employed counterfactual regularizers [24] to mitigate this issue. However, they encounter scalability issues when extending beyond two tasks, due to the high variance associated with counterfactual regularizers and their accumulation along the causal dependency chain [27]. Besides, the applicability of these causal methods is limited to scenarios where clear causal relationships exist.

To address the aforementioned limitations and better fit our real-world scenario, we propose *ResFlow*, a lightweight multi-task learning framework that boosts efficient information transfer from one task to another by introducing a set of residual connections between corresponding layers of their networks. ResFlow is hence generally applicable in situations where the information from the former task is beneficial to the latter [13, 26, 30, 31], typically from low-commitment, dense tasks to high-commitment, sparser ones, including sequentially dependent cases, e.g., "click" → "order", and more general ones where tasks show certain progressiveness, e.g., "like" and "forward". It is worth noting that such residual connections can be straightforwardly extended to longer progressive chains,

e.g., "click" → "add-to-cart" → "order", fostering a sufficient and continuous flow of information, as illustrated in Figure 2. With its simplicity and generality, ResFlow can be integrated smoothly into diverse ranking stages and application scenarios.

Comprehensive experiments on various offline datasets, as well as online A/B tests in Shopee Search, have validated the superb effectiveness and scalability of ResFlow. In particular, according to our online A/B tests, ResFlow brings a 1.29% increase in OPU (order-per-user) without extra system latency. ResFlow is now fully deployed in the pre-rank module of Shopee Search.

Furthermore, to facilitate efficient online deployment, we propose a new offline metric Weighted Recall@K, which aligns well with our online metric OPU, addressing the longstanding issue of online-offline metric misalignment. Besides, we propose to fuse the scores from the multiple tasks additively when ranking items, which according to our experiments consistently outperforms traditional multiplicative score fusion.

The contribution of this paper can be summarized as:
- We propose ResFlow, a novel, lightweight, and versatile MTL framework, demonstrating its superior performance and adaptability over state-of-the-art methods and fully deploying it in the pre-rank module of Shopee Search.
- We propose a new offline metric that addresses the longstanding online-offline metric alignment issues in practical deployment. We propose to additively fuse multi-task scores when ranking items, which outperforms traditional multiplicative score fusion.

2 RELATED WORKS

Multi-task learning, originating from Caruana's work [3], has evolved through various techniques [20, 32, 35, 37]. Its application in e-commerce has recently seen a significant increase, which can be mainly categorized into: 1) parameter-sharing based [7, 17, 22]; 2) feature information transfer based [13, 26, 30, 31]; and 3) probability transfer based [18, 27, 40]. ResFlow aligns with the feature information transfer category.

Our online tests are conducted within the pre-rank module of a multi-stage candidate selection framework [28, 38]. Specific stage focused works include: DeepMatch [11] and DSSM [16] for match stage, Wang et al. [28] and Zhang et al. [38] for pre-rank stage. Joint optimization across different stages has also been explored [8, 21].

ResFlow adopted the residual learning concept, a principle established by ResNet [10]. Various studies [14, 23] have investigated its effectiveness, and it remains popular in diverse applications, such as ControlNet [34] in diffusion model control and LoRA [33] in large language model adaptation.

Some tasks in applied ranking, e.g., the CTCVR estimation, confront severe sample imbalance issues. Besides employing multi-task learning, among the typical single-task solutions, such as sampling-based [1], cost-sensitive [4, 19], and kernel-based [29], we further adopt the sampling-based, implemented as sample re-weighting, for its simplicity and compatibility with neural networks.

3 METHODOLOGY

In this section, we introduce the proposed multi-task learning framework ResFlow. As illustrated in Figure 2, it boosts efficient information transfer from one task to another by introducing residual
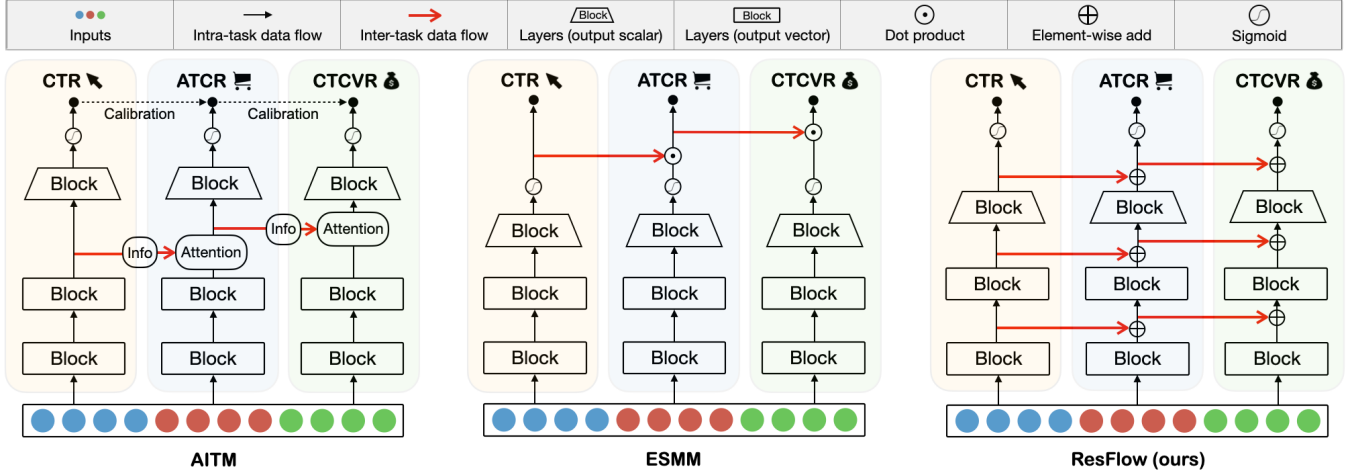
Figure 2: The conceptual architectures of ResFlow and its main competitors. Illustrated with three sequentially dependent tasks: CTR, the click-through rate; ATCR, the post-view add-to-cart rate; CTCVR, the click-through & conversion rate. AITM extracts information from the last feature layer of the former task, transferring it to the latter via an attention-based module. ESMM models the (conditional) probability of each step, estimating ATCR and CTCVR based on the products of these probabilities. ResFlow builds residual connections between corresponding layers among tasks, enabling information transfer sufficiently at various levels in an additive manner, which is not only extremely lightweight but also shows distinguished effectiveness.

connections between corresponding layers of their networks at various levels. ResFlow is generally applicable if only the information from the former task is beneficial to the latter.

Typical application scenarios include sequentially dependent tasks, e.g., "click" → "add-to-cart" → "order", and tasks that show certain progressiveness, e.g., "view", "like" and "forward". Note that, for information boost, it is typically connected from low-commitment, dense tasks to high-commitment, sparser ones (see Section 4.5).

3.1 Multi-Task Optimization

Given a set of related tasks, multi-task learning typically optimizes a joint loss that is a weighted sum of the loss from each task:

$$L(\vartheta, \theta_1, ..., \theta_K) = \sum_{k=1}^{K} \omega_k L_k(\vartheta, \theta_k), \quad (1)$$

where K denotes the number of tasks, ω_k and θ_k denote the loss weight and the task-specific parameters of task k respectively, ϑ denote the possible parameters that shared across tasks, and L_k denotes the loss of task k. More specifically,

$$L_k(\vartheta, \theta_k) = \sum_i \delta_k(y_k^i, \hat{y}_k^i(\vartheta, \theta_k, x_k^i)), \quad (2)$$

where δ_k denotes the loss function, e.g., cross-entropy, of task k, y_k^i and $\hat{y}_k^i$ respectively denote the target and the prediction of x_k^i, the i-th sample of task k.

3.2 ResFlow Architecture

The core element of ResFlow is the residual connections between corresponding layers among tasks, which are also the only extra things that need to be introduced when employing ResFlow.

The residual connection here is a generalized one, which links a position in one task network to a position in another task network, diverging from the traditional pattern, e.g., in He et al. [10] where such connections are within a single network. It allows for information transfer from the former position to the latter through element-wise addition of their values.

Notably, residual connections are directional, it is hence required to ascertain the task relationships and determine how the tasks should be chained beforehand. Regardless, for sequentially dependent tasks, we just need to follow its dependence chain. More general cases will be studied in the experiments (Section 4.5).

Besides, a logically reasonable residual connection would require its two endpoints to have a certain correspondence. ResFlow hence requires that the task networks to be residually connected have an analogous structural topology, to facilitate sensible constructions of residual connections between their corresponding points that share the same functionality and dimensionality.

For clarity, we here formally discuss the setting where the networks for each task are of the same sequential (i.e., with no branch) architecture, e.g., the same multilayer perceptron (MLP), while illustrating some more complex settings in Appendix A.

Assume that each task network consists of L sequentially chained function blocks, denoted by f_k^l with $1 \leq l \leq L$ and $1 \leq k \leq K$ respectively. A function block can simply be a linear transformation followed by a nonlinear activation, while more complex structures are equally applicable. In particular, the output of the last function block would be of the target dimension, typically a scalar.

Denoting the output of the l-th block (after residual, if has) of the k-th task as o_k^l, if the $(k-1)$-th task is residually connected with the k-th task and there is a link behind the l-th function block, then:

$$o_k^l = o_{k-1}^l + f_k^l(o_k^{l-1}), \quad (3)$$

otherwise, it will simply be:

$$o_k^l = f_k^l(o_k^{l-1}), \tag{4}$$

where o_k^0 refers to the input of task k.

Essentially, having a residual link behind would redefine f_k^l as a residual learner that learns to yield o_k^l additively based on the output o_{k-1}^l from the former task. Making use of the feature learned by the former task, which typically has denser positive feedback, could make the latter's learning easier. Otherwise, it needs to learn fully by itself to generate o_k^l.

To seek a maximum information transfer, such residual links can be employed behind all function blocks of chained task networks, including the final ones that output the logits, as in Figure 2. According to our experiments, all of these connections are beneficial.

With these residual connections, ResFlow enables transferring information effectively and sufficiently at various levels, leading to improved performance. The additional residual connections come with minimal extra computational overhead and thus can be used in various scenarios, including those with limited computational resources like pre-rank. Without leveraging any harsh prior knowledge, it is generally applicable for joint learning of tasks where the information from one task benefits another.

3.3 Implementation Details

Although ResFlow can be generally applicable, task definitions can still be important. For example, one can choose to build a task to directly model CTCVR, but given there is an auxiliary CTR task, one can also choose to model CTCVR indirectly via modeling its probability condition on the estimated CTR, like ESMM [18]. According to our experiments, with ResFlow, direct modeling leads to better performance than modeling the conditional. Apart from the sample selection bias issue in modeling conditional probability [5, 27, 40], direct modeling of CTCVR could offer better progressiveness upon the CTR task, making residual connections in-between more acceptable. From another perspective, residual learning could play the role of additively encoding in the conditional probabilities.

For sequentially dependent actions, when modeling the probabilities of progressing from the outset to each of the later stages, their predicted probability should be strictly non-increasing by definition. According to our experiments, in most e-commerce scenarios, we need no extra regularization to ensure this. They will be naturally satisfied after training, possibly due to the ground-truth probabilities having very sharp decreases within each step. Nevertheless, we tested a set of regularizers for this purpose. Among them, forcing the residual logit, i.e., the output of final block $f_k^L(o_k^{L-1})$, to be non-positive by taking a min between it and zero, gives the best performance, which is very stable, shows no harm to the performance, and can bring improvements in some cases.

4 OFFLINE EVALUATION

This section presents experiments conducted on datasets from diverse scenarios and modalities to demonstrate ResFlow's superior performance and distinguished adaptability over existing methods.

Table 1: Statistics of used e-commerce datasets. Vew, Clk, and Cnv denote the number of viewed, clicked, and converted samples, respectively.

Dataset	Split	Vew	Clk	Cnv	CTR	CVR	CTCVR
AliCCP	train	42M	1.6M	9.0K	3.9%	0.55%	0.021%
	test	43M	1.7M	9.4K	3.9%	0.56%	0.022%
S0	train	26M	1.0M	5.7K	3.8%	0.57%	0.021%
	test	26M	1.0M	6.0K	3.8%	0.60%	0.023%
S1	train	16M	0.6M	3.3K	4.0%	0.52%	0.021%
	test	16M	0.7M	3.4K	4.0%	0.52%	0.021%
S2	train	0.3M	14K	0	4.6%	0%	0%
	test	0.3M	14K	0	4.6%	0%	0%
S0&S1	train	42M	1.6M	9.0K	3.9%	0.55%	0.021%
	test	43M	1.7M	9.4K	3.9%	0.56%	0.022%
AE-ES	train	22M	0.6M	13K	2.6%	2.25%	0.058%
	test	9.3M	0.3M	6.1K	2.8%	2.30%	0.066%
AE-FR	train	18M	0.34M	9.0K	1.9%	2.63%	0.049%
	test	8.8M	0.2M	5.3K	2.2%	2.71%	0.061%
AE-NL	train	12.2M	0.25M	8.9K	2.0%	3.63%	0.073%
	test	5.6M	0.14M	4.9K	2.4%	3.61%	0.088%
AE-US	train	20.0M	0.29M	7.0K	1.5%	2.41%	0.035%
	test	7.5M	0.16M	3.9K	2.2%	2.41%	0.052%
AE-RU	train	95M	2.6M	43K	2.7%	1.68%	0.045%
	test	35M	1.0M	19K	3.0%	1.79%	0.054%
Shopee	train	4.1B	348M	8.9M	8.4%	2.57%	0.217%
	test	0.48B	40M	1.0M	8.3%	2.65%	0.220%

Table 2: Statistics of the KuaiRand-Pure-S1 dataset. We list the amount of different user feedback. VV denotes Valid View, defined by the data provider, indicating high confidence in actual user engagement.

Split	Total	VV	Like	Follow	Comment	Forward
train	945K	500K	21K	1.1K	2.8K	1.1K
test	102K	50K	2K	130	318	107

4.1 Experiment Setup

4.1.1 Datasets. The datasets considered include: 1) AliCCP, from real-world traffic logs of the recommender system in Taobao [18], and its scenario-splits S0, S1, S0&S1; 2) AE, from real-world traffic logs of the search system in AliExpress [15], comprising country-wise subsets AE-RU, AE-ES, AE-FR, AE-NL, and AE-US; 3) Shopee, a private dataset from traffic logs of the primary search scenario in Shopee, collected within 10 consecutive days; 4) KuaiRand-Pure-S1, the main scenario of the multi-scenario user interaction dataset collected from the primary recommender system of Kuaishou [9]; 5) MovieLens-1M, a widely used movie ratings dataset. Their statistics are provided in Table 1, 2, and 3.

In particular, the Shopee dataset has three types of labels: click, add-to-cart, and conversion. Shopee-2 indicates results on using only the click and conversion labels, i.e., there are two tasks, CTR and CTCVR estimation, while Shopee-3 indicates results using all three labels with three estimation tasks, CTR, ATCR, and CTCVR.

Table 3: Statistics of the MovieLens-1M dataset. We list the number of samples with different ratings.

Split	Total	1	2	3	4	5
train	$0.80M$	$45.3K$	$84.6K$	$0.207M$	$0.278M$	$0.185M$
test	$0.20M$	$10.8K$	$22.9K$	$54.0K$	$70.6K$	$41.4K$

KuaiRand-Pure-S1 and MovieLens-1M are used to attest Res-Flow's applicability to more general scenarios and other modalities.

4.1.2 Baseline Methods. The baseline methods considered in our experiments include: ESMM [18]; AITM [31]; ESCM^2-IPW [27]; ESCM^2-DR [27]; DCMT [40]. Besides, we include NSE (naive shared embedding), which employs a separate network for each task but shares the embedding table across tasks. We exclude multi-task methods such as MMOE [17] and PLE [25] due to their relatively inferior performance as reported in baselines [27, 31, 40].

4.1.3 Evaluation Metrics. In line with the convention [6, 18, 27, 40], we employ the area under the ROC curve (AUC) as the main evaluation metric for ranking tasks, while adopting the mean square error (MSE) for regression tasks. All experiments are conducted five times with different random seeds, and the average value along with the standard deviation is provided.

4.1.4 Implementations. To ensure a fair comparison, we use the same backbone network architecture (which we mainly considered MLP with two or three hidden layers) and the same basic configurations, e.g., embedding size, optimizer, and learning rate, for all tasks and all methods in our experiments. Besides, we tune the optimum sample re-weighting ratio for each task for all methods.

Note that owing to the particular nature of our considered multi-task settings, e.g., CTR and CTCVR are predicting the rates of the same user w.r.t. the same item, the inputs of different task networks in our experiments are typically the same and shared.

Due to page limitations, we provide more detailed information about the datasets and implementations in Appendix B.

4.2 Performance on E-commerce Datasets

We present in Table 4 the experiment results on the e-commerce datasets in terms of the AUC of the CTCVR estimation task, the primary goal. As we can see, ResFlow consistently outperforms all the baselines on all datasets, achieving an average improvement of 1.54% in CTCVR AUC relative to the best-performing baselines. The results for other tasks, e.g., CTR estimation, are provided in Appendix B, where ResFlow also gets closely the best performances.

Notably, ESMM achieves the best results on the Shopee datasets among the baselines, which was the method deployed online prior to ResFlow. NSE also works reasonably well for CTCVR estimation, which may have suggested that embedding sharing along with sample re-weighting can fairly counter the sample sparsity issue.

Although ESCM^2 tends to have better performance on the AE datasets, it performs less satisfactorily on the Shopee and AliCPP datasets. Such instability in performance may be attributed to the numerical sensitivity of its causal debiasing modules (IPW and DR). The performance of DCMT is relatively much better and more stable, which may be attributed to its more advanced debiasing technique.

However, it still performs worse than ESMM on the Shopee dataset. Besides, according to our experiments when applied to three-task settings, i.e., Shopee-3, the training of all these causal debiasing methods (ESCM^2 and DCMT) consistently fails with numerical errors (NaN), while all other methods work well on Shopee-3 and lead to considerable improvement compared with Shopee-2.

4.3 Ablation Studies & Parameter Sensitivities

We validate the design of ResFlow by ablation studies. As shown in Table 5, residual connections for the feature blocks and the final logit can all lead to significant improvement in performance, while their combination leads to the largest improvement, which suggests the effectiveness of residual connections at various levels. It is worth noticing that when only involving a single residual connection, the performance gradually increases from "NSE + FR H1-only" (only for the first feature block) to "NSE + FR H2-only" (only for the second feature block) to "NSE + LR" (only for logit), which suggests the residual connection for higher-level abstraction is more critical.

To be more comprehensive, we have also tested adding residual connections under the ESMM framework. As shown in Table 5, feature residual can also improve ESMM's performance, which echoes the general effectiveness of residual connections in boosting information transfer. However, further introducing logit residual aggravates its performance, which should be because logit addition can not cooperate well with the probability (sigmoid-of-logit) multiplication (modeling conditional probability). On the other hand, we can see "NSE + LR" is better than "ESMM" and even "ESMM + FR", which implies that logit addition, i.e., residual connection for logit, is a more appropriate choice than probability multiplication.

We study the sensitivity of ResFlow with respect to its key parameters. The results on the AE-RU dataset are presented in Figure 3. We see that ResFlow is not very sensitive to its hyper-parameters. It is worth noting that the optimal positive weight for the CTR task is 1, which means it does not require sample re-weighting, though the positive samples only take around 2.7%, being highly sparse/imbalanced in a sense. Meanwhile, for the CTCVR task, where the positive samples take around 0.045%, we need the positive sample weight to be 500 to gain optimal performance.

4.4 Probing the Residual Learned by ResFlow

To understand how ResFlow transfers information between, for example, CTR and CTCVR, in different layers, we randomly selected 10 users and sampled 3 items of different interaction types for each user forming 3 groups, where samples from group A were clicked with conversion, samples from group B were clicked without conversion, and samples from group C were not clicked.

Figure 4(a) and 4(b) illustrate the hidden layer feature outputs of various modules. We can see that the residual learned by the CTCVR tower (i.e., task network) has a similar value scale as the CTR tower, resulting in the recovered (after element-wise addition) feature scale of CTCVR being relatively similar to the CTR tower. By contrast, if directly learning the CTCVR tower (with NSE), its learned feature pattern appears to be quite more extreme, primarily being either larger positive values or near-zero negative values, which may have indicated a more severe over-fitting. We further provide the feature map of a randomly initialized network for reference.

Table 4: The AUC results of the CTCVR estimation task on offline e-commerce datasets. The best results are presented in bold font, while the second bests are marked with underlines. ResFlow consistently achieves the best.

| Dataset	NSE	PLE	AITM	ESMM	ESCM^2-IPW	ESCM^2-DR	DCMT	ResFlow (ours)
S0	0.619±0.001	0.630±0.005	0.636 ± 0.002	0.632 ± 0.001	0.624 ± 0.007	0.622±0.006	0.640 ± 0.001	**0.656 ± 0.001**
S1	0.621±0.002	0.624±0.003	0.623 ± 0.001	0.623 ± 0.003	0.624 ± 0.002	0.623±0.003	0.632 ± 0.002	**0.647 ± 0.002**
S0&S1	0.623±0.002	0.635±0.001	0.637 ± 0.001	0.637 ± 0.003	0.623 ± 0.005	0.623±0.001	0.634 ± 0.002	**0.661 ± 0.001**
AliCCP	0.624±0.002	0.640±0.002	0.644 ± 0.001	0.641 ± 0.003	0.625 ± 0.005	0.627±0.003	0.643 ± 0.002	**0.664 ± 0.002**
AE-ES	0.861±0.002	0.873±0.002	0.876 ± 0.001	0.867 ± 0.003	0.868 ± 0.003	0.871±0.002	0.886 ± 0.003	**0.893 ± 0.001**
AE-FR	0.842±0.002	0.852±0.002	0.856 ± 0.002	0.851 ± 0.001	0.873 ± 0.007	0.870±0.005	0.874 ± 0.003	**0.885 ± 0.001**
AE-NL	0.831±0.003	0.844±0.002	0.843 ± 0.001	0.840 ± 0.002	0.854 ± 0.006	0.854±0.004	0.858 ± 0.001	**0.864 ± 0.001**
AE-US	0.826±0.001	0.851±0.003	0.843 ± 0.002	0.827 ± 0.002	0.843 ± 0.006	0.841±0.003	0.863 ± 0.003	**0.872 ± 0.001**
AE-RU	0.870±0.003	0.886±0.001	0.882 ± 0.002	0.878 ± 0.002	0.882 ± 0.006	0.880±0.006	0.887 ± 0.001	**0.913 ± 0.002**
Shopee-2	0.865±0.001	0.860±0.003	0.855 ± 0.006	0.882 ± 0.001	0.840 ± 0.012	0.812±0.041	0.867 ± 0.001	**0.902 ± 0.001**
Shopee-3	0.877±0.002	0.877±0.003	0.871 ± 0.003	0.893 ± 0.001	/	/	/	**0.910 ± 0.002**
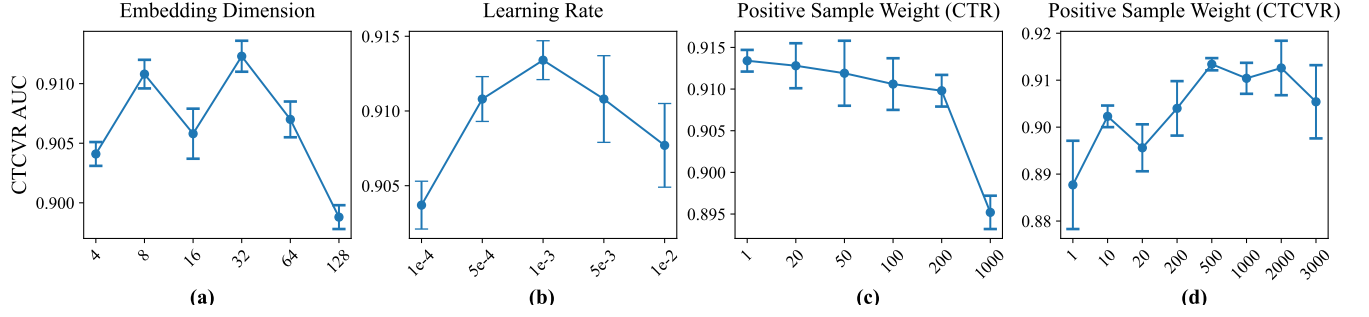
Figure 3: Parameter sensitivity experiment results. From left to right: (a) embedding dimension, (b) learning rate, (c) positive sample weight of CTR task, and (d) positive sample weight of CTCVR task. The negative sample weight is fixed as 1.
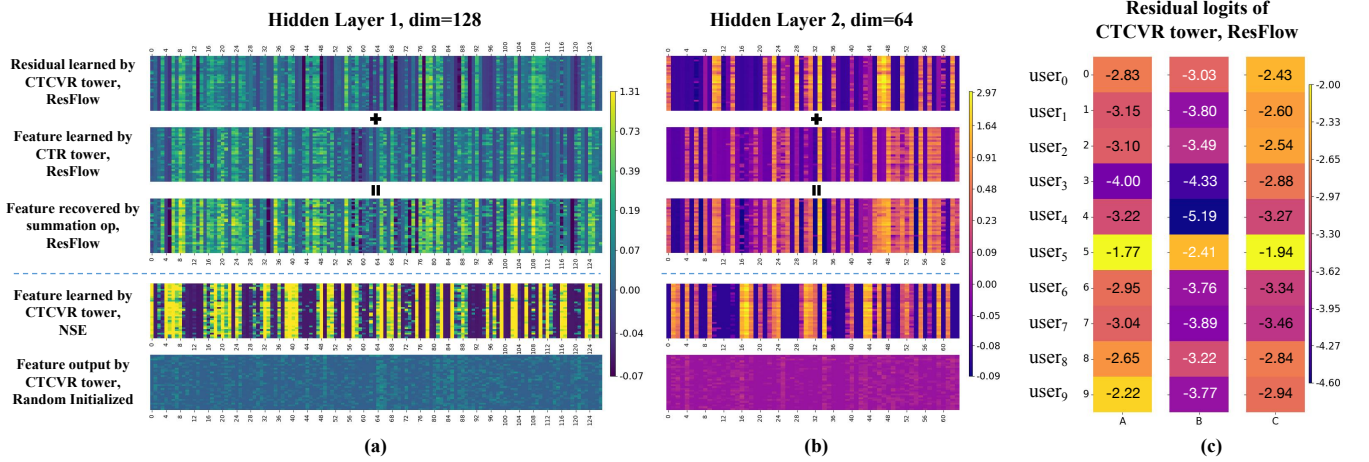
Figure 4: Case study of the learned residual. We randomly selected 10 users and sampled 3 items of different types for each user, forming 3 groups: A (click=1, order=1), B (click=1, order=0), and C (click=0, order=0). The learned residual features and logits of the CTCVR tower, and other related, are shown. Trained on the AliCCP dataset. ResFlow tends to learn less extreme features than NSE. The learned residual logits are all negative, and the values in group B are generally smaller than those in group A.

Figure 4(c) shows the learned residual logits by the CTCVR tower. We see that they are all negative values, and that is exactly how it should be. Because CTCVR should be strictly less than CTR, i.e.,

the logit should be decreased to get a smaller sigmoid probability. Notably, the learned residual logits of group B are generally smaller

Table 5: Ablation results of ResFlow in terms of CTCVR AUC.

Model	AE-RU	Shopee-2
NSE	0.869 ± 0.003	0.865 ± 0.001
NSE + Feature Residual (FR)	0.897 ± 0.001	0.891 ± 0.001
NSE + FR (H1-only)	0.880 ± 0.002	0.868 ± 0.002
NSE + FR (H2-only)	0.898 ± 0.002	0.887 ± 0.001
NSE + Logit Residual (LR)	0.907 ± 0.002	0.896 ± 0.001
ResFlow (NSE + FR + LR)	$\mathbf{0.913 \pm 0.002}$	$\mathbf{0.902 \pm 0.001}$
ESMM	0.878 ± 0.002	0.882 ± 0.001
ESMM + FR	0.899 ± 0.004	0.891 ± 0.002
ESMM + FR + LR	0.896 ± 0.006	0.888 ± 0.001

than group A, which is also reasonable since "click without conversion" instances require larger decreases in the logits to shift from a high probability of clicking to a low probability of conversion.

4.5 More General Multi-Task Scenario

To demonstrate the more general applicability of ResFlow. We experiment with the KuaiRand-Pure-S1 dataset, modeling its user interactions with videos, including: "valid view", "like", "follow", "comment", and "forward".

Given that the logical relationships between "follow", "comment", and "forward" are not that clear, and that there is no significant differentiation in their frequencies, while they are all clearly less frequent than "like", and "like" is clearly less frequent than "valid view". In this experiment, we build residual connections from task "is_valid_view" to task "is_like", from "is_like" to "is_follow", from "is_like" to "is_comment", and from "is_like" to "is_forward", respectively, and jointly train the five tasks. Although strict conditional dependence is lacking, these probability-transfer-based methods (ESMM, etc.) are also included as baselines.

As shown in Table 6, ResFlow leads to consistent improvement and achieves the best across all tasks, demonstrating its capability in modeling such types of progressiveness. Notably, AITM gets an unsatisfactory performance on the "is_follow" task. We consider that having too many parameters in its attention-based information transfer block may have made it more likely to overfit occasionally. The full table with AUCs of all tasks is provided in Appendix B.

To further validate such an intuitive strategy, i.e., building task topologies according to their levels of sample sparsity, is a good choice, we tested a set of other possible task topologies on this dataset. We visualize them in Figure 5 and provide their respective results in Table 7. We can see that "topo1", the aforementioned topology strategy, performs the best. More specifically, we see that sparse tasks can generally benefit from denser ones, e.g., task "is_follow", "is_forward", and "is_comment" in topo1, and task "is_follow" task in topo2. However, chaining tasks of the same sparsity level or putting sparse tasks before denser ones may lead to inferior performance, e.g., task "is_forward" and "is_comment" in topo2, task "is_follow" in topo3, and task "is_forward" and "is_comment" in topo4. We also notice that task "is_forward" seems to work properly when chained after task "is_comment" in topo3, which may be owed to its alignment with users' overall habits on commenting and forwarding, but its accuracy is still less than the one in "topo1".

Table 6: Performance on KuaiRand-Pure-S1 in terms of AUC. The bests are in bold font. The second bests are underlined.

Target	is_follow	is_forward	is_comment
NSE	0.821 ± 0.005	0.750 ± 0.011	0.779 ± 0.003
AITM	0.785 ± 0.014	$\underline{0.764 \pm 0.012}$	0.782 ± 0.005
ESMM	$\underline{0.822 \pm 0.007}$	0.757 ± 0.007	0.782 ± 0.004
ESCM^2-IPW	0.817 ± 0.010	0.754 ± 0.008	0.773 ± 0.004
ESCM^2-DR	0.821 ± 0.009	0.754 ± 0.008	0.782 ± 0.003
DCMT	0.819 ± 0.007	0.762 ± 0.008	$\underline{0.783 \pm 0.003}$
ResFlow	$\mathbf{0.826 \pm 0.002}$	$\mathbf{0.776 \pm 0.006}$	$\mathbf{0.789 \pm 0.003}$
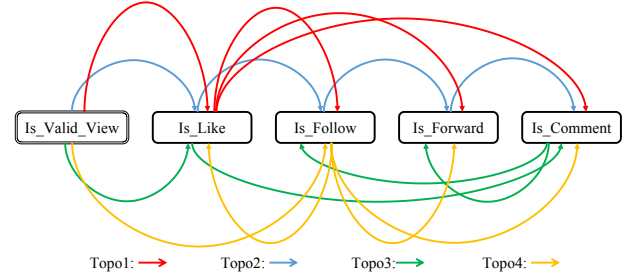
Figure 5: Visualization of different task topologies of KuaiRand-Pure-S1 Multi-Task.

Table 7: Performance on KuaiRand-Pure-S1 in terms of AUC. The bests are in bold font. The second bests are underlined.

Target	is_follow	is_forward	is_comment
ResFlow-topo1	$\mathbf{0.826 \pm 0.002}$	$\mathbf{0.776 \pm 0.006}$	$\mathbf{0.789 \pm 0.003}$
ResFlow-topo2	$\underline{0.825 \pm 0.003}$	0.756 ± 0.010	0.771 ± 0.002
ResFlow-topo3	0.811 ± 0.007	$\underline{0.774 \pm 0.007}$	0.778 ± 0.003
ResFlow-topo4	0.820 ± 0.002	0.748 ± 0.009	0.775 ± 0.002
NSE	0.821 ± 0.005	0.750 ± 0.011	$\underline{0.779 \pm 0.003}$

This all indicates that, for scenarios without explicit dependency among tasks, building the task topology according to their sample sparsities can be a sound and easy-to-follow choice.

4.6 Regression as Progressive Multi-Task

Besides predicting the likelihood of user actions, which can be formulated as classification problems, there are also practical needs to predict numerical values, e.g., predicting the video playtime or the rating of a user to an item. In this section, we showcase that such regression tasks may also be converted as progressive multi-task learning problems and effectively tackled by ResFlow.

Formally, to predict a value that has a bounded range and indicates progressive engagements as it increases, we discretize the target variable if it is not inherently discretized. Subsequently, for each discretized threshold v_k, excluding the minimal one, we define a task $Q(v \geq v_k)$ to predict the likelihood that the actual value equals or surpasses v_k. By definition, these tasks embody progressiveness, necessitating their probabilities non-increase as v_k ascends.[1]

---

[1]Forcing residual logits non-positive, taking min with 0, led to slight gains in this task.

Table 8: Regression performances on KuaiRand-Pure-S1 and MovieLens-1M in terms of MSE. Progressive indicates converting the regression into a progressive multi-task problem.

Dataset	KuaiRand-Pure-S1	MovieLens-1M
Traditional	1719.92 ± 3.82	0.906 ± 0.002
Progressive + NSE	1720.12 ± 4.32	0.906 ± 0.002
Progressive + ResFlow	$\mathbf{1658.44 \pm 5.21}$	$\mathbf{0.894 \pm 0.002}$

For setting the training targets, if the actual value is v, then by the definition, all tasks where $v_k \leq v$ are assigned with a label of 1, and all other tasks receive a label of 0. For prediction purposes, we employ the approximation of the expected value:

$$\mathbb{E}(v) = \sum_{k=0}^{K-1} v_k \cdot \max\left(Q(v \geq v_k) - Q(v \geq v_{k+1}), 0\right) + v_K \cdot Q(v \geq v_K),$$

where $v_0 < v_1 < \ldots < v_K$, v_0 and v_K are the minimal and maximal possible values respectively, $Q(v \geq v_0) = 1$, there are K tasks.

We experiment with: 1) MovieLens-1M, where we predict the users' ratings on movies; and 2) KuaiRand-Pure-S1, where we predict the users' video playtime. We compare the progressive modeling of regression with the traditional regression and test whether ResFlow can help such progressive modeling. Since multi-task solutions need to involve multiple networks and hence more parameters, we search for the best network configuration within the multilayer perception class for all methods for a fair comparison.

As the results presented in Table 8, progressive modeling (with NSE) works almost equally well as traditional regression, while ResFlow can effectively boost the performance of progressive modeling, leading to clear improvements over traditional regressions.

5 ONLINE DEPLOYMENT

We conducted prolonged A/B tests to assess the effectiveness of ResFlow in the online environment and ultimately deployed it in the pre-rank module in the primary search scenario of Shopee. In this section, we detail our investigations about the online deployment.

5.1 The Pre-Rank Module

The pre-rank module is responsible for sorting roughly filtered item candidates from the match module and passes the top-ranked to the rank module. Typically, it needs to handle millions of items within a very short time, strictly less than 150ms in our scenario, hence requiring both efficiency and accuracy.

Following the normal practice [11, 28, 38], we adopt a twin-tower task network architecture for pre-rank: one tower for extracting representation for user-and-query, and one for item, which is illustrated in Figure 7 in the Appendix. It facilitates the separation of user-query representation inference, which requires online processing, from item representations, which can be precomputed offline.

5.2 Online-Offline Metric Alignment

The main online metric in Shopee is OPU, the average number of orders placed by each user (in one day). OPU-driven optimization could contribute to the growth of platform scale and market share.

However, online OPU, as well as other similar online metrics, can only be effectively evaluated online. Regarding offline data, it is a fixed value that cannot be used to tune the model and hyperparameters. Therefore, finding offline metrics that well aligns with the online metric is vital for efficient model optimization in the industry, given the time-intensive and costly nature of online A/B testing. However, commonly used offline metrics do not well align with the online metrics [2, 12, 38]. Tuning configurations for online deployments hence has long been and still is a challenge.

To address this, we propose the offline metric Weighted Recall@K (WR@K), which characterizes how well the model can rank hot items up within the top K, which we found to align fairly well with the online metric for pre-rank. Formally,

$$WR@K = \frac{\sum_{k=0}^{K} W_k}{\sum_{n=0}^{N} W_n}, \quad (5)$$

where K is the given parameter of the metric, N is the total number of items to be ranked, and W_k indicates the number of orders of item k (in one day, under a given query).

As results shown in Figure 6 and Table 10, WR@100 aligns reasonably well with OPU compared to other metrics. In the compared metrics, NDCG is the normalized discounted cumulative gain, while List AUC is the AUC of the ranked list considering items with order labels as positive. We consider that the effectiveness of WR@K stems from its incorporation of collective user feedback (i.e., W_k) that reflects item popularity, while metrics like List AUC, Recall@K, and NDCG tend to focus on individual user responses.

Given that List AUC and NDCG emphasize a different perspective from WR@K and also show a positive correlation with the online metric to a certain extent, we adopt WR@K as the primary metric for evaluating model performance, while using List AUC and NDCG as complementary monitoring metrics [2, 12, 38].

5.3 Score Fusion Strategy

To align with the online metric (OPU in our case) and better provide the service[2], ranking systems in e-commerce platforms typically fuse multiple scores to form the indicator for item ranking. A historically employed formula for score fusion combines[3] estimated CTR and CTCVR multiplicatively [18]:

$$Score = CTR^{\alpha} \times CVR^{\beta}, \quad (6)$$

which can be equivalently formulated as $CTR^{\alpha-\beta} \times CTCVR^{\beta}$. In our experiments, we investigated fusing CTR and CTCVR additively:

$$Score = \alpha \cdot CTR + \beta \cdot CTCVR, \quad (7)$$

which we found to be more effective. Besides, we tried to further additively incorporate semantic relevance indicators from other teams, which also seems valuable.

We search the best weight parameter (e.g., α and β) for all these different fusion formulations w.r.t. offline WR@K for ESMM and ResFlow, and for each of the best, we conduct a two-week online A/B test. The results are shown in Table 9, where we use ESMM with $CTR^1 \times CVR^1$, i.e., directly using CTCVR as the ranking indicator,

---

[2]It may require exposing more diverse items of interest to users including those they won't order. This might be why CTR is typically incorporated in score fusion formulas.
[3]One could include a price term to balance market scale measured by order quantity and gross merchandise value. We mainly consider OPU hence omit it for clarity.
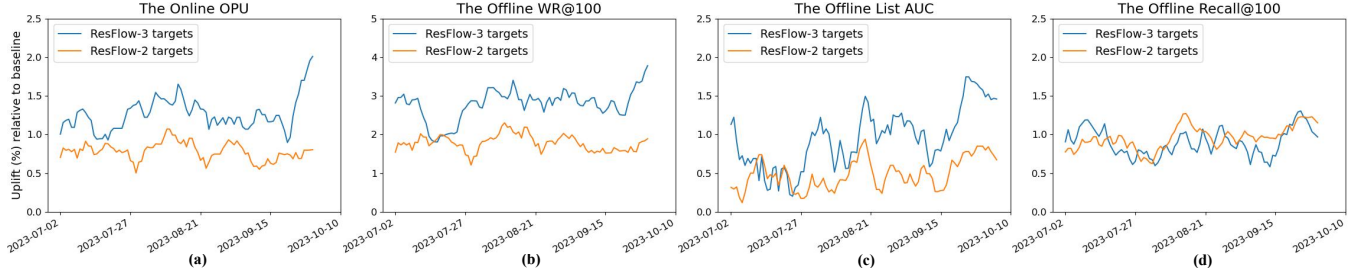
Figure 6: The uplift of online OPU and other offline metrics of ResFlow relative to the formerly deployed method (ESMM) during online A/B test. The curves are smoothed to better show the trend. Offline WR@100 aligns fairly well with online OPU.

Method & Score Formula	WR@100	List AUC	OPU	CTR	CTCVR	BCR@20
ESMM, $CTR^{0.9} \times CVR^{1.1}$	+0.77%	+0.23%	+0.56%	-0.02%	+0.61%	+0.02%
ESMM, CTR*2+CTCVR*19	+2.7%	+0.09%	+0.75%	+0.12%	+0.67%	-0.01%
ResFlow, CTCVR	+1.41%	+1.82%	+0.45%	-0.15%	+0.40%	-0.02%
ResFlow, $CTR^{-0.2} \times CTCVR$	+3.37%	+1.03%	+1.33%	+0.06%	+1.19%	-0.02%
ResFlow, CTR+CTCVR*20	+4.19%	+0.88%	+2.14%	+0.66%	+2.11%	+0.02%
ResFlow, CTR+CTCVR*20+RL	+4.11%	+0.82%	+2.04%	+0.61%	+1.97%	-2.13%
ResFlow, CTR+CTCVR*20+RS	+0.03%	-0.13%	-0.53%	-0.19%	-0.44%	-2.99%
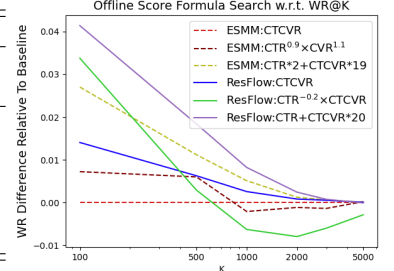
Table 9: Here we use ESMM with fusion strategy $CTR^1 \times CVR^1$, i.e., CTCVR, as the baseline. Left: online A/B test results on various score fusion strategies. We show the relative uplift of offline WR@100 and List AUC, and online OPU, CTR, CTCVR, and BCR@20 (bad case rate @ top 20). RL: weighted relevance level. RS: weighted relevance score. Both ESMM and ResFlow are trained with 3 targets. Right: best offline search results of score fusion formulas. We show their WR@K relative to the baseline.

Table 10: Pearson correlation coefficient (PCC) between offline metric uplift and online metric uplift.

Metrics	2 Targets Uplift		3 Targets Uplift	
	PCC	p-value	PCC	p-value
Recall@100	0.0814	0.4425	0.1974	0.0606
NDCG	0.3719	0.0002	0.3825	0.0002
List AUC	0.4353	1.6×10^{-5}	0.1564	0.1386
WR@100	0.7879	1.9×10^{-20}	0.8666	2.5×10^{-42}

as the baseline, and report the relative performances. We see that: 1) ResFlow with CTCVR outperforms ESMM with CTCVR; 2) score fusions generally lead to better performance than directly using CTCVR; 3) ResFlow with optimal multiplicative fusion outperforms that of ESMM; 4) both ESMM and ResFlow show better performance with additive formulas; 5) ResFlow with optimal additive fusion outperforms that of ESMM; 6) best-performing formulas[4] tend to enhance CTR relative to CVR or CTCVR; 7) adding an optimally weighted relevance score (RS) leads to a decrease in OPU though improves the bad case rate (the rate of top-ranked items mismatching user query, from another team); 8) adding an optimally weighted relevance level (RL, three-level discretized relevance score) degenerates OPU slightly, while notably improving the bad case rate.

We consider that multiplicative fusion may be too sensitive to extreme values, e.g., if one score (e.g., CTCVR) is very small, the others

---

could lose their voting rights, while additive fusion provides milder control. Relevance level helps steadily enhance item relevance.

5.4 Online Performance

With the best score fusion formulas for each, we compared 3-target ResFlow against 3-target ESMM with a two-week online A/B test. ResFlow achieved a 1.29% uplift in OPU, a 0.88% increase in gross merchandise value, a 0.84% rise in the number of buyers, a 0.25% increase in online CTR, and a 1.37% uplift in online CTCVR, compared to ESMM, without an increase in the bad case rate and system latency, with average and 99th percentile latency being 110 ms and 147 ms respectively for ResFlow v.s. 110 ms and 146 ms for ESMM.

6 CONCLUSION

We have proposed ResFlow, a lightweight multi-task learning framework that enables information transfer via inter-task residual connections, and demonstrated its general effectiveness in various application scenarios with extensive offline and online experiments. We have addressed key implementation challenges of deploying multi-task learning in applied ranking systems, including online-offline metric alignment and multi-score fusion strategies. ResFlow is now fully deployed in the pre-rank module of Shopee Search.

---

[4]Note that $CTR^{-0.2} \times CTCVR$ is equivalent to $CTR^{0.8} \times CVR^{1.0}$.

REFERENCES

[1] Gustavo EAPA Batista, Ronaldo C Prati, and Maria Carolina Monard. 2004. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter* 6, 1 (2004), 20–29.

[2] Joeran Beel, Marcel Genzmehr, Stefan Langer, Andreas Nürnberger, and Bela Gipp. 2013. A comparative analysis of offline and online evaluations and discussion of research paper recommender system evaluation. In *Proceedings of the international workshop on reproducibility and replication in recommender systems evaluation*. 7–14.

[3] Rich Caruana. 1997. Multitask learning. *Machine learning* 28 (1997), 41–75.

[4] Nitesh V Chawla, Nathalie Japkowicz, and Aleksander Kotcz. 2004. Special issue on learning from imbalanced data sets. *ACM SIGKDD explorations newsletter* 6, 1 (2004), 1–6.

[5] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2023. Bias and debias in recommender system: A survey and future directions. *ACM Transactions on Information Systems* 41, 3 (2023), 1–39.

[6] Yizhou Chen, Guangda Huzhang, Anxiang Zeng, Qingtao Yu, Hui Sun, Heng-Yi Li, Jingyi Li, Yabo Ni, Han Yu, and Zhiming Zhou. 2023. Clustered Embedding Learning for Recommender Systems. In *Proceedings of the ACM Web Conference 2023*. 1074–1084.

[7] Jing Du, Lina Yao, Xianzhi Wang, Bin Guo, and Zhiwen Yu. 2022. Hierarchical Task-aware Multi-Head Attention Network. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1933–1937.

[8] Luke Gallagher, Ruey-Cheng Chen, Roi Blanco, and J Shane Culpepper. 2019. Joint optimization of cascade ranking models. In *Proceedings of the twelfth ACM international conference on web search and data mining*. 15–23.

[9] Chongming Gao, Shijun Li, Yuan Zhang, Jiawei Chen, Biao Li, Wenqiang Lei, Peng Jiang, and Xiangnan He. 2022. KuaiRand: An Unbiased Sequential Recommendation Dataset with Randomly Exposed Videos. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management* (Atlanta, GA, USA) *(CIKM '22)*. 3953–3957. https://doi.org/10.1145/3511808.3557624

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[11] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. 2333–2338.

[12] Guangda Huzhang, Zhenjia Pang, Yongqing Gao, Yawen Liu, Weijie Shen, Wen-Ji Zhou, Qing Da, Anxiang Zeng, Han Yu, Yang Yu, et al. 2021. AliExpress Learning-To-Rank: Maximizing online model performance without going online. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[13] Jiarui Jin, Xianyu Chen, Weinan Zhang, Yuanbo Chen, Zaifan Jiang, Zekun Zhu, Zhewen Su, and Yong Yu. 2022. Multi-Scale User Behavior Network for Entire Space Multi-Task Learning. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 874–883.

[14] Janghyeon Lee, Donggyu Joo, Hyeong Gwon Hong, and Junmo Kim. 2020. Residual continual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 4553–4560.

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

[16] Zhengdong Lu and Hang Li. 2013. A deep architecture for matching short texts. *Advances in neural information processing systems* 26 (2013).

[17] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1930–1939.

[18] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. 2018. 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*. 1137–1140.

[19] Marcus A Maloof. 2003. Learning when data sets are imbalanced and when costs are unequal and unknown. In *ICML-2003 workshop on learning from imbalanced data sets II*, Vol. 2. 2–1.

[20] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. 2016. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3994–4003.

[21] Jiarui Qin, Jiachen Zhu, Bo Chen, Zhirong Liu, Weiwen Liu, Ruiming Tang, Rui Zhang, Yong Yu, and Weinan Zhang. 2022. RankFlow: Joint Optimization of Multi-Stage Cascade Ranking Systems as Flows. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 814–824.

[22] Zhen Qin, Yicheng Cheng, Zhe Zhao, Zhe Chen, Donald Metzler, and Jingzheng Qin. 2020. Multitask mixture of sequential experts for user activity streams.

[23] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. *Advances in neural information processing systems* 30 (2017).

[24] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In *international conference on machine learning*. PMLR, 1670–1679.

[25] Hongyan Tang, Junning Liu, Ming Zhao, and Xudong Gong. 2020. Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In *Proceedings of the 14th ACM Conference on Recommender Systems*. 269–278.

[26] Xuewen Tao, Mingming Ha, Qiongxu Ma, Hongwei Cheng, Wenfang Lin, Xiaobo Guo, Linxun Cheng, and Bing Han. 2023. Task Aware Feature Extraction Framework for Sequential Dependence Multi-Task Learning. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 151–160.

[27] Hao Wang, Tai-Wei Chang, Tianqiao Liu, Jianmin Huang, Zhichao Chen, Chao Yu, Ruopeng Li, and Wei Chu. 2022. Escm2: Entire space counterfactual multi-task model for post-click conversion rate estimation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 363–372.

[28] Zhe Wang, Liqin Zhao, Biye Jiang, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2020. COLD: Towards the Next Generation of Pre-Ranking System. *Workshop on Deep Learning Practice for High-Dimensional Sparse Data in KDD 2020* (2020).

[29] Gang Wu and Edward Y Chang. 2004. Aligning boundary in kernel space for learning imbalanced dataset. In *Fourth IEEE International Conference on Data Mining (ICDM'04)*. IEEE, 265–272.

[30] Yiqing Wu, Ruobing Xie, Yongchun Zhu, Xiang Ao, Xin Chen, Xu Zhang, Fuzhen Zhuang, Leyu Lin, and Qing He. 2022. Multi-view multi-behavior contrastive learning in recommendation. In *International Conference on Database Systems for Advanced Applications*. Springer, 166–182.

[31] 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. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3745–3755.

[32] Yongxin Yang and Timothy Hospedales. 2017. Trace Norm Regularised Deep Multi-Task Learning. In *International Conference on Learning Representations*.

[33] Yu Yu, Chao-Han Huck Yang, Jari Kolehmainen, Prashanth G Shivakumar, Yile Gu, Sungho Ryu, Roger Ren, Qi Luo, Aditya Gourav, I-Fan Chen, et al. 2023. Low-rank adaptation of large language model rescoring for parameter-efficient speech recognition. *arXiv preprint arXiv:2309.15223* (2023).

[34] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3836–3847.

[35] Yu Zhang and Qiang Yang. 2017. Learning sparse task relations in multi-task learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[36] Yu Zhang and Qiang Yang. 2021. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering* 34, 12 (2021), 5586–5609.

[37] Yu Zhang, Dit-Yan Yeung, and Qian Xu. 2010. Probabilistic multi-task feature selection. *Advances in neural information processing systems* 23 (2010).

[38] Zhixuan Zhang, Yuheng Huang, Dan Ou, Sen Li, Longbin Li, Qingwen Liu, and Xiaoyi Zeng. 2023. Rethinking the Role of Pre-ranking in Large-scale E-Commerce Searching System. *arXiv preprint arXiv:2305.13647* (2023).

[39] Yong Zheng and David Xuejun Wang. 2022. A survey of recommender systems with multi-objective optimization. *Neurocomputing* 474 (2022), 141–153.

[40] Feng Zhu, Mingjie Zhong, Xinxing Yang, Longfei Li, Lu Yu, Tiehua Zhang, Jun Zhou, Chaochao Chen, Fei Wu, Guanfeng Liu, et al. 2023. DCMT: A Direct Entire-Space Causal Multi-Task Framework for Post-Click Conversion Estimation. In *Proceedings of the39th International Conference on Data Engineering*. 3113–3125.

A MORE EXPERIMENT RESULTS

A.1 Residual Links in Twin-Tower Architecture

For the twin-tower architecture, illustrated in Figure 7(a), which is typically employed in the pre-rank module, establishing residual connections is slightly different from that of the single-tower architecture. The main concern is where to put the residual links around the highest-level abstractions, i.e., before or after the inner product. We have tested different choices. The results are shown in Table 13. We can see that positioning the residual link before or after the inner product presents similar performances, while putting it after the inner product, i.e., building a residual connection on the logits, leads to slightly better performance. We consider that putting a residual link after the inner product would treat the inner product operation along with the last function blocks of the two towers as the final residual learner, which offers more degrees of freedom in learning and can leverage a higher degree of abstraction from the former task, whereas adding residual links before the inner product is also fine, which defines the last function block of each tower as a residual learner respectively. In contrast, adding residual links to both before and after the inner product degenerates the performance significantly. This should be because the residual link before the inner product will turn the last function blocks into residual learners respectively for the two towers, but meanwhile, the residual link after the inner product also makes the inner product a residual learner. However, there are no learnable parameters in the inner product operation, hence it lacks the expressiveness needed to be a residual learner, which would not ease the learning process but bring about more burdens to the lower levels. Positioning the residual link after the dot-product operation is adopted for ResFlow in Section 4.2 and in online deployment.

A.2 Enforcing Non-Increase of Probability

In sequentially dependent multi-task settings, e.g., "click" → "add-to-cart" → "order", when modeling the probabilities of progressing from the outset to each of the later stages, it is expected that the predicted probabilities to be non-increasing along the dependency chain since only when the former action takes place the latter could possibly happen. According to our experiments, such non-increase can naturally be learned by the task networks without additional regularizers in most cases, especially in e-commerce settings where the ground truth probability for each task shows significant disparities. Nevertheless, we investigated several regularization techniques for enforcing such a constraint, including:

(1) Penalizing the increase in probabilities via a regularization defined as, akin to the one used in AITM [31]:

$$L_{prob} = \sum_i \sum_{k=2}^{K} \max(\hat{y}_k^i - \hat{y}_{k-1}^i, 0). \qquad (8)$$

(2) Penalizing residual logits exceeding zero:

$$L_{logit} = \sum_i \sum_{k=2}^{K} \max(logit_k^i, 0). \qquad (9)$$

(3) Mandating residual logits to be non-positive, by involving a $\min(logit, 0)$ operator after the logit and before the sigmoid.

Table 11: Results of regularizers for enforcing non-increase of probability. W.O.: ResFlow without non-increase regularizer. M1: ResFlow + penalizing increases in probabilities. M2: ResFlow + penalizing positive residual logits. M3: ResFlow + mandating non-positive residual logits. M4: ESMM + feature residual. We tune the best reg weight for each of them.

Method	Dataset (Task)	Reg Weight	MSE/AUC
W.O.	KuaiRand-Pure-S1 (Regression)	NA	1658.44
M1	KuaiRand-Pure-S1 (Regression)	0.1	1658.45
M2	KuaiRand-Pure-S1 (Regression)	0.1	1656.72
M3	KuaiRand-Pure-S1 (Regression)	NA	1655.83
M4	KuaiRand-Pure-S1 (Regression)	NA	1673.55
W.O.	Shopee-3	NA	0.9101
M1	Shopee-3	0.1	0.9093
M2	Shopee-3	0.1	0.9087
M3	Shopee-3	NA	0.9102
M4	Shopee-3	NA	0.9085
W.O.	AE-RU	NA	0.9134
M1	AE-RU	0.5	0.9133
M2	AE-RU	0.1	0.9097
M3	AE-RU	NA	0.9135
M4	AE-RU	NA	0.9083

Table 12: Pearson correlation coefficient (PCC) between offline metric uplift and online metric uplift.

	2 Targets Uplift		3 Targets Uplift	
	PCC	p-value	PCC	p-value
WR@100	0.7879	1.9×10^{-20}	0.8666	2.5×10^{-42}
WR_{log}@100	0.7359	9.5×10^{-17}	0.8056	6.0×10^{-22}
WR_{sqrt}@100	0.7054	5.8×10^{-15}	0.8281	4.2×10^{-24}
WR_{square}@100	0.6797	1.3×10^{-13}	0.8515	1.1×10^{-26}

(4) Generating the output probability via the multiplication of the output probability of the former task and the probability generated by a new-build task network, i.e., using the ESMM framework, to ensure the non-increase of probability.

The results are shown in Table 11. We can see that mandating non-positive residual logits leads to the best results among the four. It leads to almost the same performance as the non-regularized version in most cases, while in some cases brings slight improvement. Other regularizers, however, may affect the performance.

A.3 WR@K Variants

We have tried different variants of WR@K by replacing the weight term W_k with, for example, $\log(W_k)$, $\sqrt{W_k}$, and W_k^2. The results are shown in Table 12 and Figure 8. Comparing with Table 10, we can see that the original version of WR@K aligns with online OPU best.

A.4 More Ablation and Baseline Results

We provide the ablation results of ResFlow on AliCCP in Table 14. We provide CTCVR AUC the results of more baselines, including the single-task model, MOE, and MMOE, in Table 15.
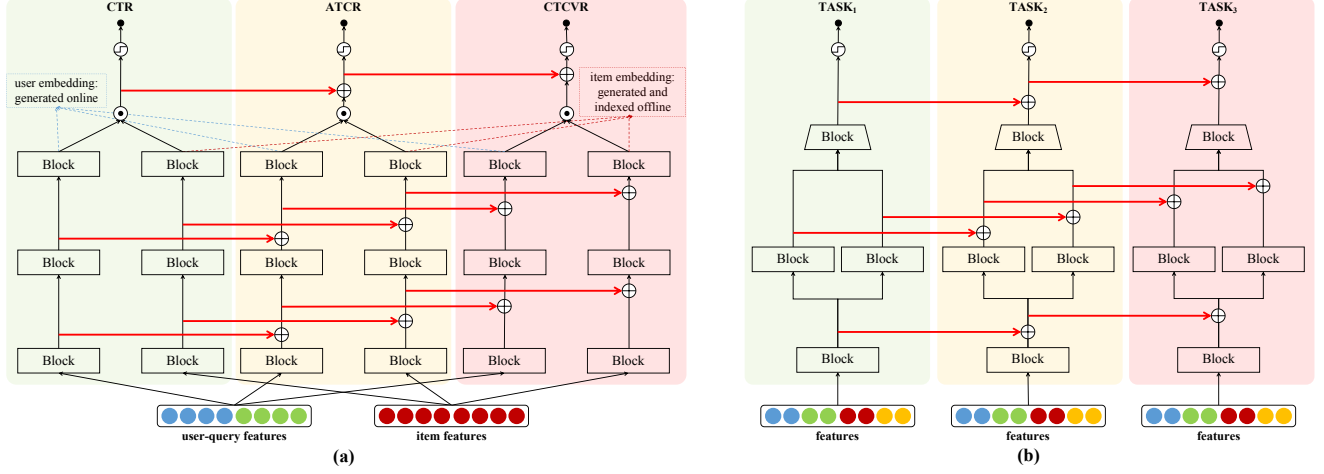
Figure 7: ResFlow with more complex architectures. (a) The illustration of ResFlow builds upon the twin-tower architecture. It is adopted as the backbone for all experiments on the Shopee dataset and our online deployment. (b) Another conceptual ResFlow architecture, where task networks have inner branches. Residual connections can be generally applied behind function blocks that have parameters, while we should be cautious about operations that have no or very few parameters (see Section A.1).
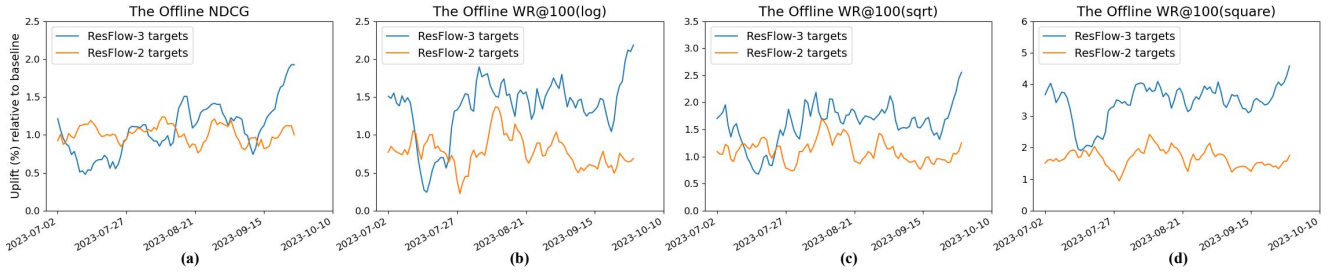

Figure 8: The uplift of more offline metrics of ResFlow relative to the formerly deployed method (ESMM) during the online A/B test. The curves are smoothed to better show the trend.

Table 13: Ablation results of the highest-level residual connections in the twin-tower architecture on Shopee-2.

Strategy	CTR AUC	CTCVR AUC
Before & After	0.8573 ± 0.0018	0.8801 ± 0.0037
Before Dot-Product	0.8661 ± 0.0021	0.9019 ± 0.0022
After Dot-Product	**0.8667 ± 0.0014**	**0.9024 ± 0.0010**

Table 14: Ablation results of ResFlow on AliCCP.

Model	CTCVR AUC
NSE	0.6238 ± 0.0018
NSE + Feature Residual (FR)	0.6482 ± 0.0010
NSE + FR(H1-only)	0.6296 ± 0.0021
NSE + FR(H2-only)	0.6417 ± 0.0015
NSE + Logit Residual (LR)	0.6511 ± 0.0013
ResFlow(NSE + FR + LR)	**0.6642 ± 0.0016**
ESMM	0.6412 ± 0.0031
ESMM + FR	0.6503 ± 0.0028
ESMM + FR + LR	0.6487 ± 0.0043

Table 15: More AUC results of the CTCVR estimation task on offline e-commercial datasets.

Datasets	Single Task	MOE	MMOE
S0	0.609 ± 0.005	0.624 ± 0.002	0.628 ± 0.001
S1	0.609 ± 0.003	0.620 ± 0.002	0.622 ± 0.001
S0&S1	0.610 ± 0.002	0.629 ± 0.002	0.636 ± 0.002
AliCCP	0.613 ± 0.003	0.637 ± 0.001	0.640 ± 0.001
AE-ES	0.844 ± 0.003	0.871 ± 0.001	0.872 ± 0.003
AE-FR	0.834 ± 0.003	0.850 ± 0.003	0.851 ± 0.001
AE-NL	0.813 ± 0.002	0.831 ± 0.001	0.847 ± 0.003
AE-US	0.812 ± 0.004	0.844 ± 0.002	0.851 ± 0.002
AE-RU	0.862 ± 0.004	0.876 ± 0.003	0.886 ± 0.003
Shopee-2	0.859 ± 0.003	0.851 ± 0.001	0.862 ± 0.002
Shopee-3	0.859 ± 0.003	0.876 ± 0.002	0.876 ± 0.003