# AutoHERI: Automated Hierarchical Representation Integration for Post-Click Conversion Rate Estimation

Penghui Wei†, Weimin Zhang†, Zixuan Xu, Shaoguo Liu, Kuang-chih Lee and Bo Zheng*
Alibaba Group
{wph242967,dutan.zwm,xuzixuan.xzx,shaoguo.lsg,kuang-chih.lee,bozheng}@alibaba-inc.com

## ABSTRACT

Post-click conversion rate (CVR) estimation is a crucial task in online advertising and recommendation systems. To address the sample selection bias problem in traditional CVR models trained in click space, recent studies perform entire space multi-task learning based on the probability of events in user behavior funnels like "impression-click-conversion". However, those models learn the feature representation of each task independently, and omit potential inter-task correlations that can help improve the CVR estimation performance. In this paper, we propose AutoHERI, an entire space CVR model with automated hierarchical representation integration, which leverages the interplay across multi-tasks' representation learning. It performs neural architecture search to learn optimal connections between layer-wise representations of different tasks. Besides, AutoHERI achieves better search efficiency with one-shot search algorithm, and thus it can be easily extended to new scenarios that have more complex user behaviors. Both offline and online experimental results on large-scale real-world datasets verify that AutoHERI outperforms previous entire space models significantly.

## CCS CONCEPTS

• **Information systems** → **Online advertising**; • **Computing methodologies** → **Multi-task learning**; **Neural networks**.

## KEYWORDS

Online advertising; Post-click conversion rate; Hierarchical representation integration; Neural architecture search

**ACM Reference Format:**
Penghui Wei†, Weimin Zhang†, Zixuan Xu, Shaoguo Liu, Kuang-chih Lee and Bo Zheng*. 2021. AutoHERI: Automated Hierarchical Representation Integration for Post-Click Conversion Rate Estimation. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3459637.3482061

## 1 INTRODUCTION

Post-click conversion rate (CVR) estimation plays an important role for online advertising and recommendation systems. Considering the complete period of an item's transaction in e-commerce sites, the user behavior funnel is typically composed of several chronological phrases, e.g., impression, click and conversion. The goal of post-click conversion rate estimation is to predict the probability of conversion after a user has clicked an item.

A main challenge in CVR estimation is the sample selection bias problem [16]. Traditional CVR models are trained in click space, that is, positive samples are the impressions that were clicked and purchased, and negative samples are the ones that were clicked only. However, at online inference stage, the trained models are utilized to estimate CVR in the entire impression space. Note that a large fraction of impressions will not be clicked, and thus this intrinsic difference between distributions of training and test data limits the generalization of CVR models. To address this, Ma et al. [8] proposed an entire space multi-task model, which incorporates the estimation tasks of click-through rate (CTR) and click-through conversion rate (CTCVR) to indirectly train the CVR estimation task in entire impression space. Follow-up work [12] considered more behaviors between click and conversion as additional learning tasks to further address data sparsity [5], and achieve better results.

Although current entire space CVR models have achieved state-of-the-art results, they learn the feature representation of each task using independent neural network (e.g., multi-layer perceptron), and omit potential inter-task correlations that can help improve the performance. The two learning tasks induced from the funnel "impression-click-conversion" form a task hierarchy, where estimating CTR / CVR can be regarded as the preceding / succeeding task. It is natural that representations learned from the preceding task(s) can be selectively integrated into the succeeding task(s) to improve their estimation performance. However, previous studies independently model each task, thus they do not exploit the hierarchical nature of them and only achieve suboptimal CVR estimation result.

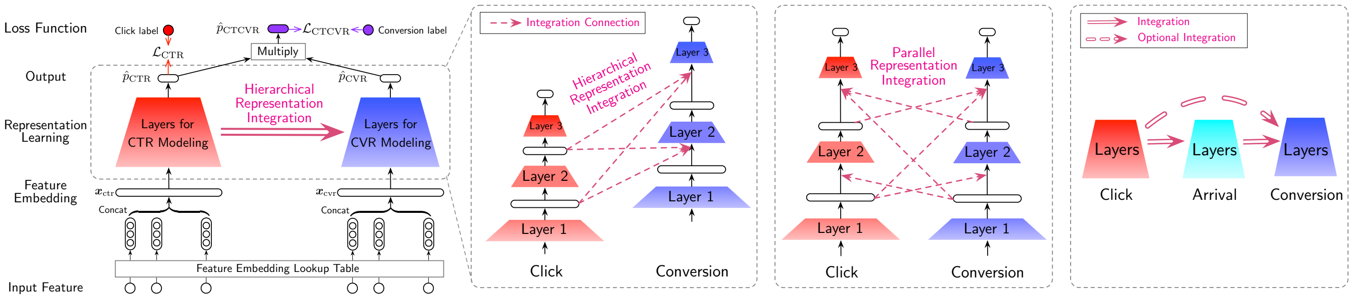
To implement such representation integration scheme, a straightforward way is to manually design layer-wise connections from the layers of preceding task(s) to those of succeeding task(s). However, the number of possible connection combinations grows exponentially as the increasing numbers of layers and estimation tasks, thus it is too costly to obtain the optimal connection combination via trial-and-error. Besides, even if we can obtain a good connection combination for one scenario, it may not be applicable to new scenarios having different behavior funnels. When we are faced with CVR estimation for new scenarios, it is intractable if we repeat the manual trial-and-error procedure for each one.

In this paper, we propose AutoHERI, an entire space CVR estimation model with automated hierarchical representation integration, which exploits the task hierarchy induced from behavior funnels and leverages the interplay across multi-tasks' representations via neural architecture search (NAS). We add integration connections

†Equal contribution. *Corresponding author.


*CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia*

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482061




(a) Hierarchical Representation Integration (b) Parallel Representation Integration (c) Extension to More Behaviors

**Figure 1: Overview of AutoHERI for CVR estimation. Subfigure (a) takes the scenario following "impression-click-conversion" as an illustration, and the right part shows the *hierarchical* representation integration scheme. Subfigure (b) shows the *parallel* representation integration scheme that omits the task hierarchy. Subfigure (c) shows the extension to more behaviors.**

between layer-wise representations from preceding tasks to succeeding tasks, learning effective integration pattern in a hierarchical fashion. Motivated by the intuition that each scenario has its specific integration pattern based on its behavior funnel, AutoHERI automates the procedure of obtaining optimal connection combination with one-shot architecture search, achieving better efficiency. Thus, AutoHERI can be easily extended to new scenarios having complex behavior funnels. The contributions of this paper are:

- To our knowledge, this is among the first work that incorporates automated hierarchical inter-task representation integration for CVR estimation in entire impression space.
- We propose a CVR estimation model AutoHERI, which performs automated hierarchical representation integration via one-shot architecture search. It can be easily adopted to various scenarios with specific behavior funnels.
- Both offline and online experiments verify that AutoHERI significantly outperforms the state-of-the-art CVR models.

## 2 PROBLEM FORMULATION

An impression is represented using a feature set that contains user features, item features and other context features. Most features are categorical, where each one can be denoted as a one-hot encoding. An impression's feature set is transformed to a sparse feature vector (composed of various one-hot encodings). Consider an impression sample $(x, y_{\text{click}}, y_{\text{conv}})$, where $x$ is the sparse feature vector and $y_{\text{click}} \in \{0, 1\}$ / $y_{\text{conv}} \in \{0, 1\}$ denotes its click label / post-view conversion label. The estimations of CTR, CVR and CTCVR aim at predicting the following probabilities respectively: $p_{\text{CTR}} = p(y_{\text{click}} = 1 \mid x)$, $p_{\text{CVR}} = p(y_{\text{conv}} = 1 \mid y_{\text{click}} = 1, x)$, and $p_{\text{CTCVR}} = p(y_{\text{conv}} = 1, y_{\text{click}} = 1 \mid x) = p_{\text{CTR}} \times p_{\text{CVR}}$.

## 3 PROPOSED METHOD

AutoHERI exploits the task hierarchy induced from user behavior funnels on impressions, and learns hierarchical representation integration scheme to improve succeeding tasks in behavior funnels.

Without loss of generality, we first describe AutoHERI for a simplified scenario where behavior funnels follow "impression-click-conversion" (§ 3.1 – 3.2). Fig. 1 (a) shows an overview. We then introduce how to extend it for more behaviors (§ 3.3).

### 3.1 Backbone of AutoHERI

For the scenario of "impression-click-conversion", CTR / CVR estimation is the preceding / succeeding task in the task hierarchy.

In this case, the backbone of AutoHERI consists of a feature embedding layer and two networks that learn high-level representations for estimating CTR and CVR respectively, as shown in Fig. 1 (a). Given an impression's sparse feature vector used for CTR estimation task, the feature embedding layer transforms each one-hot feature to a dense embedding, and then concatenates these embeddings to obtain the feature embedding $\boldsymbol{x}_{\text{ctr}}$. Similarly, for CVR estimation task we can obtain the feature embedding $\boldsymbol{x}_{\text{cvr}}$.

Two networks further learn the representations for CTR and CVR estimation tasks, where each network consists of multiple layers (such as MLPs). The network of CTR task takes the feature embedding $\boldsymbol{x}_{\text{ctr}}$ as input, and produces the predicted CTR $\hat{p}_{\text{CTR}}$ after multiple layers and a logistic function. The network of CVR task has a similar architecture and produces $\hat{p}_{\text{CVR}}$. To further improve the performance of CVR estimation, AutoHERI introduces the following hierarchical representation integration scheme.

### 3.2 Automated Hierarchical Representation Integration

With the aim of better exploiting the representations learned from preceding tasks to improve succeeding tasks, we propose a hierarchical representation integration scheme to leverage the interplay across multiple tasks. We automated the integration procedure by means of neural architecture search, which makes AutoHERI can be easily adopted to different scenarios with various behavior funnels.

*3.2.1* ***Hierarchical Representation Integration.*** As illustrated in Fig. 1 (a), based on the task hierarchy of CTR and CVR estimation, AutoHERI adds integration connections between the two networks in a hierarchical and layer-wise fashion. It selectively integrates the output representations of CTR network into the representations of CVR network, aiming to learn effective integration patterns and obtain the optimal connection combination for CVR estimation.

Formally, suppose that the numbers of layers for CTR and CVR tasks are $N_{\text{ctr}}$ and $N_{\text{cvr}}$. We denote the output representation of the $i$-th layer for CTR task as $\boldsymbol{h}^i_{\text{ctr}}$, and similarly the $j$-th layer for CVR task as $\boldsymbol{h}^j_{\text{cvr}}$. To model the inter-task representation integration, each possible integration connection points from $\boldsymbol{h}_{\mathrm{ctr}}^{i}$ to $\boldsymbol{h}_{\mathrm{cvr}}^{j}$ (where $i \in \{1, 2, \ldots, N_{\mathrm{ctr}}\}$ and $j \in \{1, 2, \ldots, N_{\mathrm{cvr}}\}$). It forms a connection set containing $N_{\mathrm{ctr}} \times N_{\mathrm{cvr}}$ possible integration connections.

Let $z_{ij} \in \{0, 1\}$ denote whether there is an integration connection between $\boldsymbol{h}_{\mathrm{ctr}}^{i}$ and $\boldsymbol{h}_{\mathrm{cvr}}^{j}$. Considering the $j$-th layer for CVR task (parameterized by $\mathbf{W}_j$ and $\mathbf{b}_j$), its output representation is:

$$\boldsymbol{h}_{\mathrm{cvr}}^{j} = \mathrm{Trans}\left(\mathrm{ReLU}\left(\mathbf{W}_j \boldsymbol{h}_{\mathrm{cvr}}^{j-1} + \mathbf{b}_j\right), z_{ij}\boldsymbol{h}_{\mathrm{ctr}}^{i}\right) \quad (1)$$

where $\mathrm{Trans}(\cdot, \cdot)$ is a linear transformation operation that maps the dimension of $[\mathrm{ReLU}(\mathbf{W}_j \boldsymbol{h}_{\mathrm{cvr}}^{j-1} + \mathbf{b}_j); z_{ij}\boldsymbol{h}_{\mathrm{ctr}}^{i}]$ to $\boldsymbol{h}_{\mathrm{cvr}}^{j}$.

There are $2^{N_{\mathrm{ctr}} \times N_{\mathrm{cvr}}}$ candidate connection combinations between the two tasks' networks. Next, we detail how to automatically learn the optimal connection combination via neural architecture search.

*3.2.2* ***Automated Integration via One-Shot Architecture Search***. AutoHERI employs one-shot architecture search to automate the procedure of obtaining optimal connection combination. One-shot algorithms [3, 6, 15] make the search process more effectively than traditional ones by relaxing the binary connections as continuous probabilities. Thus, the problem is reduced to optimize such probabilities, which can be jointly trained with network parameters.

Formally, we relax the binary connection values $\boldsymbol{z} = \{z_{ij}\}$ to continuous connection probabilities $\boldsymbol{\alpha} = \{\alpha_{ij}\}$, where each $\alpha_{ij}$ is the mean of a Bernoulli distribution, and the term $z_{ij}\boldsymbol{h}_{\mathrm{ctr}}^{i}$ in Eq. 1 is also replaced by $\alpha_{ij}\boldsymbol{h}_{\mathrm{ctr}}^{i}$. During training, we adopt gradient descent to jointly optimize: 1) the network parameters $\Theta$ for representation learning, and 2) the integration probabilities $\boldsymbol{\alpha}$ for searching the optimal connection combination. Given an impression sample $(x, y_{\mathrm{click}}, y_{\mathrm{conv}})$, AutoHERI's outputs ($\hat{p}_{\mathrm{CTR}}$ and $\hat{p}_{\mathrm{CVR}}$) depend on both $\Theta$ and $\boldsymbol{\alpha}$. Based on the equation of $p_{\mathrm{CTCVR}} = p_{\mathrm{CTR}} \times p_{\mathrm{CVR}}$, we compute the cross-entropy losses of CTR task and CTCVR task in impression space respectively. The overall loss function is:

$$\mathcal{L}(\Theta, \boldsymbol{\alpha}) = \ell_{\mathrm{xent}}(\hat{p}_{\mathrm{CTR}}, y_{\mathrm{click}}) + \lambda \ell_{\mathrm{xent}}(\hat{p}_{\mathrm{CTR}} \times \hat{p}_{\mathrm{CVR}}, y_{\mathrm{conv}}) \quad (2)$$

where $\ell_{\mathrm{xent}}(\cdot, \cdot)$ denotes cross-entropy loss, and $\lambda$ is a factor.[1]

We optimize the network parameters $\Theta$ and the connection probabilities $\boldsymbol{\alpha}$ with the following bi-level optimization:

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} \mathcal{L}\left(\Theta^*(\boldsymbol{\alpha}), \boldsymbol{\alpha}\right) \\ &\text{s.t. } \Theta^*(\boldsymbol{\alpha}) = \arg\min_{\Theta} \mathcal{L}(\Theta, \boldsymbol{\alpha}^*) \end{aligned} \quad (3)$$

At each iteration, we first optimize the network parameters $\Theta$ while the connection probabilities $\boldsymbol{\alpha}$ are fixed. We then fix the updated $\Theta^*(\boldsymbol{\alpha})$ and optimize $\boldsymbol{\alpha}$. Note that the two terms $\mathcal{L}\left(\Theta^*(\boldsymbol{\alpha}), \boldsymbol{\alpha}\right)$ and $\mathcal{L}(\Theta, \boldsymbol{\alpha}^*)$ need to be computed using distinct data at each iteration. Following [3], we add an entropy regularization loss term $\sum_{i,j}\left(-\alpha_{ij}\log\alpha_{ij} - (1-\alpha_{ij})\log(1-\alpha_{ij})\right)$ on the connection probabilities $\boldsymbol{\alpha}$ to ensure better convergence.

## 3.3 Extension to Modeling More Behaviors

We use the advertising scenario where user behavior funnels follow "impression-click-arrival-conversion" to introduce how we can extend AutoHERI to more scenarios. In this case, a post-click landing page is the page users arrive at after they click the corresponding ads, and conversion happens in the landing page.

[1]For $(x, y_{\mathrm{click}} = 1, y_{\mathrm{conv}} = 0)$, it is a positive/negative sample of CTR/CTCVR task. To avoid gradient conflict, for CTR network we can stop the gradient of CTCVR loss.

Based on such behavior funnel, an impression sample has three labels $(y_{\mathrm{click}}, y_{\mathrm{arrival}}, y_{\mathrm{conv}})$ that denote click label, post-view arrival label and post-view conversion label. The induced task hierarchy contains three estimation tasks for CTR, post-click arrival rate (AR) and post-arrival conversion rate (VR): $p_{\mathrm{CTR}} = p(y_{\mathrm{click}} = 1 \mid x)$, $p_{\mathrm{AR}} = p(y_{\mathrm{arrival}} = 1 \mid y_{\mathrm{click}} = 1, x)$, and $p_{\mathrm{VR}} = p(y_{\mathrm{conv}} = 1 \mid y_{\mathrm{click}} = 1, y_{\mathrm{arrival}} = 1, x)$. Thus, the CVR is as $p_{\mathrm{CVR}} = p_{\mathrm{AR}} \times p_{\mathrm{VR}}$, and we still have $p_{\mathrm{CTCVR}} = p_{\mathrm{CTR}} \times p_{\mathrm{CVR}}$.

For this scenario, the backbone of AutoHERI has three networks to produce $\hat{p}_{\mathrm{CTR}}$, $\hat{p}_{\mathrm{AR}}$ and $\hat{p}_{\mathrm{VR}}$ for estimating CTR, AR and VR. Given an impression, it gives the estimation for CVR as $\hat{p}_{\mathrm{AR}} \times \hat{p}_{\mathrm{VR}}$.

The hierarchical representation integration scheme can add three groups of integration connections from preceding tasks' representations to succeeding tasks' ones: from CTR to AR, from AR to VR and from CTR to VR (see Fig. 1 (c)). The third group is optional, and in practice incorporating connections between two adjacent networks is enough. The loss function is defined as follows, which contains three cross-entropy losses with two factors $\lambda_1$ and $\lambda_2$:

$$\begin{aligned} \mathcal{L}(\Theta, \boldsymbol{\alpha}) = &\ell_{\mathrm{xent}}(\hat{p}_{\mathrm{CTR}}, y_{\mathrm{click}}) + \lambda_1 \ell_{\mathrm{xent}}(\hat{p}_{\mathrm{CTR}} \times \hat{p}_{\mathrm{AR}}, y_{\mathrm{arrival}}) \\ &+ \lambda_2 \ell_{\mathrm{xent}}(\hat{p}_{\mathrm{CTR}} \times \hat{p}_{\mathrm{AR}} \times \hat{p}_{\mathrm{VR}}, y_{\mathrm{conv}}) . \end{aligned} \quad (4)$$

# 4 EXPERIMENTS

## 4.1 Experimental Setup

*4.1.1* ***Datasets***. We conduct experiments on two datasets. The first is a public dataset named **Ali-CCP** [8], which contains click and conversion labels. Because there is no available dataset containing more behaviors, we collected user feedback logs from Alibaba advertising system, named **Ads** dataset. It contains click, arrival and conversion labels. We split the dataset to training/validation/test sets using timestamp with a proportion of 5:1:1. Table 1 lists the statistics. As in previous work, we report the metrics of AUC and negative log likelihood (NLL) on estimating CVR and CTCVR.

*4.1.2* ***Comparative Models***. We compare AutoHERI with the following strong baseline models for CVR estimation: (1) DNN [1] uses a network trained in click space to produces $\hat{p}_{\mathrm{CVR}}$. (2) ESMM [8] uses two networks to model CTR and CVR. It is trained with CTR and CTCVR tasks in impression space. (3) Multi-DR [17] is the state-of-the-art model for scenarios of two behaviors (click and conversion). It employs doubly robust to produce unbiased estimation. (4) ESM$^2$ [12] augments ESMM for scenarios having more than two behaviors. We employ three networks to model CTR, AR and VR respectively. (5) DBMTL [11] considers inter-task relationships with manually designed sequential operations on the top of networks. (6) AITM [13] is a contemporaneous work that models multi-task interactions with attention and probability constraint loss.

For fair comparison, in all models, each network is a three-layered MLP, with output sizes [1024, 512, 256] (Ads dataset) or [200, 80, 20] (Ali-CCP). We use Adam optimizer with 256 batch size and 0.005 learning rate. Factors are $\lambda = 5.0$, $\lambda_1 = 1.0$ and $\lambda_2 = 5.0$.

**Table 1: Statistics of two datasets.**

| Dataset | # impression | # click | # arrival | # conversion |
|---|---|---|---|---|
| Ali-CCP | 84 mil. | 3.4 mil. | – | 18 k |
| Ads | 0.4 bil. | 63 mil. | 34 mil. | 0.1 mil. |

**Table 2: Results of CVR estimation on Ads dataset and Ali-CCP dataset. Δ denotes performance gain w.r.t. DNN.**

| Representation Integration | Model | Ads Dataset | | | | Ali-CCP Dataset | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $AUC_{CVR}$ (Δ) | $NLL_{CVR}$ | $AUC_{CTCVR}$ (Δ) | $NLL_{CTCVR}$ | $AUC_{CVR}$ (Δ) | $NLL_{CVR}$ | $AUC_{CTCVR}$ (Δ) | $NLL_{CTCVR}$ |
| No | DNN [1] | 66.78 | 0.01319 | 73.26 | 0.00277 | 65.96 | 0.03335 | 60.84 | 0.00206 |
| | ESMM [8] | 67.67 (↑0.89) | 0.01314 | 73.72 (↑0.46) | 0.00277 | 67.25 (↑1.29) | 0.03320 | 62.89 (↑2.05) | 0.00205 |
| | Multi-DR [17] | 67.91 (↑1.13) | 0.01317 | 73.70 (↑0.44) | 0.00276 | 67.74 (↑1.78) | 0.03307 | 63.22 (↑2.38) | 0.00205 |
| | $ESM^2$ [12] | 67.81 (↑1.03) | 0.01315 | 74.00 (↑0.74) | 0.00276 | - | - | - | - |
| Yes | DBMTL [11] | 67.99 (↑1.21) | 0.01312 | 74.01 (↑0.75) | 0.00276 | 66.91 (↑0.95) | 0.03353 | 63.36 (↑2.52) | 0.00209 |
| | AITM [13] | 68.22 (↑1.44) | 0.01311 | 74.29 (↑1.03) | **0.00275** | 67.61 (↑1.65) | **0.03299** | 63.27 (↑2.43) | **0.00204** |
| | AutoHERI | **68.74 (↑1.96)** | **0.01307** | **74.47 (↑1.21)** | **0.00275** | **67.83 (↑1.87)** | 0.03301 | **63.79 (↑2.95)** | **0.00204** |

**Table 3: Ablation tests.**

| Model | $AUC_{CVR}$ | $AUC_{CTCVR}$ | Training |
|---|---|---|---|
| AutoHERI | 68.74 | 74.47 | ~290min |
| w/o hierarchical, w/ parallel | 68.36 | 74.18 | ~320min |
| w/o one-shot, w/ evolutionary | 68.65 | 74.49 | >2 days |
| AITM (non-NAS-based) | 68.22 | 74.29 | ~210min |

**Table 4: Online results (relative improvements).**

| Model | CVR | CTCVR | Cost-per-conversion |
|---|---|---|---|
| AutoHERI | **+4.98%** | **+6.77%** | **-5.86%** |

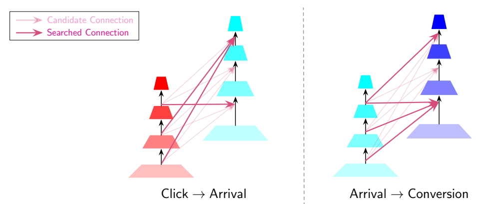



**Figure 2: The searched hierarchical connections.**

## 4.2 Results and Discussion

*4.2.1* ***Main Results.*** Table 2 shows the results of all models on two datasets. Compared to DNN that is trained in click space, entire space models perform better than it, indicating that entire space multi-task modeling is effective for CVR estimation. On the Ads dataset, $ESM^2$ outperforms ESMM and Multi-DR, which shows that modeling more behaviors can achieve higher improvements.

DBMTL and AITM outperform $ESM^2$, and thus explicitly modeling inter-task relationships can boost CVR estimation. AutoHERI incorporates hierarchical integration and automatically learns the optimal connection combination, achieving the best performance with a significant improvement. The results verify that AutoHERI benefits from exploiting the task hierarchy induced by behavior funnels to enhance representation learning of CVR estimation.

*4.2.2* ***Effectiveness of AutoHERI.*** We further compare AutoHERI with two variants to verify the effectiveness. The first variant replaces the integration connection scheme from the hierarchical fashion to a parallel one. As shown in Fig. 1 (b), this variant also leverages the interplay of multiple tasks, but does not consider the task hierarchy induced from behavior funnels. The second variant is optimized via evolutionary algorithm [9] other than one-shot search. Table 3 reports their results. AutoHERI performs better than the parallel-based variant on both CVR and CTCVR estimation, demonstrating that taking the task hierarchy into account during multitasks' representation learning can further boost the estimation performance. Besides, AutoHERI performs on par with the evolutionary-based variant, and thus one-shot algorithm for searching layer-wise connections achieves competitive results.

*4.2.3* ***Efficiency of AutoHERI.*** We evaluate the efficiency by comparing training time. As in the last column of Table 3, compared to evolutionary-based variant, AutoHERI significantly reduces the searching time. Compared to the non-NAS-based competitor, the training time of AutoHERI is also comparable and it achieves better AUC performance. Therefore, AutoHERI has advantages in both effectiveness and efficiency for entire space CVR estimation.

*4.2.4* ***Illustration of Searched Connections.*** Fig. 2 shows the final architecture of AutoHERI, which is difficult to be designed manually. We observe some interesting patterns that it prefers the two types of connections: from lower layers of click modeling to the higher layers of arrival modeling, and from some layers of arrival modeling to the lower layers of conversion modeling.

## 4.3 Online A/B Test

We deployed AutoHERI to Alibaba advertising system for one week. Table 4 reports the relative improvements compared to the current state-of-the-art model deployed online on CVR, CTCVR and cost-per-conversion (smaller is better). AutoHERI achieves significantly gain on all metrics, verifying its effectiveness in industrial system.

# 5 RELATED WORK

Current entire space CVR models [8, 12] omit the task hierarchy during learning representations. Although a few studies consider multiple behaviors [2, 11, 14], each of them is for a specific scenario with manually-designed network architecture. Recently, some studies employed automated machine learning for embedding size [4] and interaction function searching [7, 10], focusing on searching a better model for one task. In contrast, our work aims at taking the hierarchy of multiple tasks into account and learns the hierarchical representation integration scheme.

# 6 CONCLUSION

In this paper, we propose AutoHERI for post-click conversion rate estimation in impression space. It performs automated hierarchical representation integration scheme via one-shot architecture search, which exploits the task hierarchy induced from user behavior funnels and can be easily adopted to different CVR estimation scenarios. Both offline and online experimental results verify that AutoHERI significantly outperforms state-of-the-art CVR models.

## REFERENCES


[1] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for YouTube recommendations. In *Proceedings of RecSys*.

[2] Chen Gao, Xiangnan He, Dahua Gan, Xiangning Chen, Fuli Feng, Yong Li, Tat-Seng Chua, and Depeng Jin. 2019. Neural multi-task recommendation from multi-behavior data. In *Proceedings of ICDE*.

[3] Yuan Gao, Haoping Bai, Zequn Jie, Jiayi Ma, Kui Jia, and Wei Liu. 2020. MTL-NAS: Task-agnostic neural architecture search towards general-purpose multi-task learning. In *Proceedings of CVPR*.

[4] Manas R Joglekar, Cong Li, Mei Chen, Taibai Xu, Xiaoming Wang, Jay K Adams, Pranav Khaitan, Jiahui Liu, and Quoc V Le. 2020. Neural input search for large scale recommendation models. In *Proceedings of KDD*.

[5] Kuang-chih Lee, Burkay Orten, Ali Dasdan, and Wentong Li. 2012. Estimating conversion rate in display advertising from past performance data. In *Proceedings of KDD*.

[6] Hanxiao Liu, Karen Simonyan, and Yiming Yang. 2019. DARTS: Differentiable architecture search. In *Proceedings of ICLR*.

[7] Yuanfei Luo, Mengshuo Wang, Hao Zhou, Quanming Yao, Wei-Wei Tu, Yuqiang Chen, Wenyuan Dai, and Qiang Yang. 2019. AutoCross: Automatic feature crossing for tabular data in real-world applications. In *Proceedings of KDD*.

[8] 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 *Proceedings of SIGIR*.

[9] Esteban Real, Sherry Moore, Andrew Selle, Saurabh Saxena, Yutaka Leon Suematsu, Jie Tan, Quoc V Le, and Alexey Kurakin. 2017. Large-scale evolution of image classifiers. In *Proceedings of ICML*. 2902–2911.

[10] Qingquan Song, Dehua Cheng, Hanning Zhou, Jiyan Yang, Yuandong Tian, and Xia Hu. 2020. Towards automated neural interaction discovery for click-through rate prediction. In *Proceedings of KDD*.

[11] Qi Wang, Zhihui Ji, Huasheng Liu, and Binqiang Zhao. 2019. Deep Bayesian Multi-Target Learning for Recommender Systems. *arXiv preprint arXiv:1902.09154* (2019).

[12] Hong Wen, Jing Zhang, Yuan Wang, Fuyu Lv, Wentian Bao, Quan Lin, and Keping Yang. 2020. Entire space multi-task modeling via post-click behavior decomposition for conversion rate prediction. In *Proceedings of SIGIR*.

[13] 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 KDD*.

[14] Lianghao Xia, Chao Huang, Yong Xu, Peng Dai, Bo Zhang, and Liefeng Bo. 2020. Multiplex Behavioral Relation Learning for Recommendation via Memory Augmented Transformer Network. In *Proceedings of SIGIR*.

[15] Sirui Xie, Hehui Zheng, Chunxiao Liu, and Liang Lin. 2019. SNAS: Stochastic neural architecture search. In *Proceedings of ICLR*.

[16] Bianca Zadrozny. 2004. Learning and evaluating classifiers under sample selection bias. In *Proceedings of ICML*.

[17] Wenhao Zhang, Wentian Bao, Xiao-Yang Liu, Keping Yang, Quan Lin, Hong Wen, and Ramin Ramezani. 2020. Large-scale Causal Approaches to Debiasing Post-click Conversion Rate Estimation with Multi-task Learning. In *Proceedings of WWW*.