

Constructing a Joint Prediction Model for Advertising Click-Through Rate and Conversion Rate Based on a Multi-Task Learning Framework

Shihua Zuo¹

¹Department of Economics and Trade, Hunan Mechanical and Electrical Polytechnic, Changsha 410151, Hunan, China

Corresponding author: Shihua Zuo (e-mail: zuoshihua2005@126.com)

ABSTRACT Click-through rate and conversion rate prediction in online advertising systems are constrained by sample bias and selection bias because conversion-rate training is usually conducted only on clicked samples. To address this problem, this paper constructs a multi-task learning framework for joint CTR and CVR modeling. User, advertisement, and contextual features are first modeled through a shared representation layer. A CTR branch is then designed to generate a stable click signal to assist CVR prediction. A causal bias correction module based on inverse propensity scoring and weighted loss is introduced to correct the distribution discrepancy between click samples and impression samples. Experimental results show that the proposed framework achieves an AUC of 0.912 for joint CTR/CVR prediction, compared with 0.828 for the baseline model. LogLoss decreases to 0.365 and 0.381 for the corresponding tasks, while NDCG@1 for CTR improves to 0.538. Online experiments show CTR and CVR improvement rates of 10.2% and 9.6%, respectively, and revenue increases by 9.9%, demonstrating effective mitigation of sample and selection bias.

INDEX TERMS online advertising, click-through rate prediction, conversion rate prediction, multi-task learning, causal bias correction

I. INTRODUCTION

THE pervasive development of online advertising has fundamentally driven the rapid evolution of large-scale personalized recommendations and precision marketing.

The core of ad ranking and revenue optimization lies in high-precision predictions of users' click-through rate (CTR) and conversion rate (CVR). These predictions not only directly determine the priority of ad display but also influence the overall commercial revenue structure [1,2]. CTR and CVR prediction tasks essentially involve modeling user behavior and inferring advertising effectiveness. Their accuracy is crucial to the efficient allocation of traffic resources and the value of advertising placement [3,4]. However, in existing methods, CTR and CVR are often modeled independently.

CTR relies on a large number of impressions and click samples, while CVR training data is derived solely from conversion samples generated after clicks. Conversion labels for non-clicks are missing, resulting in significant sample bias and selection bias [5,6]. Factors such as differences in ad placement, dynamic changes in user interests, and diversity in display environments further exacerbate the problem, making it difficult for traditional methods to ensure robustness and generalization capabilities in real business

scenarios.

Academia and industry have proposed various explorations to alleviate the above problems. In the field of digital advertising, Singh et al. [7] first proposed a reinforcement learning framework to optimize advertising display strategies and maximize user clicks through a dynamic decision-making process. Subsequently, Xu et al. [8] designed a core interest network to improve CTR prediction accuracy. This model improves feature expression capabilities by mining user core interest points. Finally, Cheng et al. [9] identified three major click behavior patterns by segmenting the search engine user journey, providing a behavioral basis for advertising trigger design. However, most of these methods still assume that the sample distribution is consistent. When the actual distribution of post-click samples differs significantly from that of exposure samples, the prediction results are still biased.

At the feature engineering level of mobile advertising recommendation systems, Zhu et al. [10] systematically proposed a method for extracting and optimizing user behavior features, and significantly improved the accuracy of personalized recommendations through multimodal feature fusion. In terms of user perception, Lim et al. [11] revealed the cognitive patterns of digital natives on online advertis-

ing through cross-cultural empirical research, providing a decision-making basis for the precision marketing of small and medium-sized enterprises. These studies have promoted the development of digital advertising from multiple dimensions, including feature engineering, user psychology, experience innovation, platform strategy, and technical implementation, and provided solid theoretical support and technical reference for the construction of a multi-task joint prediction framework in this study.

This paper aims to address the sample bias and selection bias issues in the joint prediction of CTR and CVR by proposing an improved framework based on multi-task learning. This framework utilizes a shared representation layer to uniformly model user, ad, and contextual features to enhance input consistency and feature expressiveness. Furthermore, an independent CTR prediction branch is designed to generate a stable click signal as auxiliary information for CVR modeling. A causal bias correction module is applied to mitigate the distribution discrepancy between click and impression samples using an inverse propensity score and a weighted loss. Guided by both the CTR signal and bias correction, a CVR prediction branch is constructed to achieve transformation inference across the entire impression space. The designed causal bias correction module weights click samples based on the inverse propensity score and controls the variance through weight truncation and a sliding average strategy. This module is optimized collaboratively with the multi-task framework to directly correct selection bias when inferring CVR in the impression space. This multi-task learning framework innovatively integrates a shared representation layer, CTR auxiliary signal guidance, and the causal bias correction module. Through a dynamically weighted multi-task loss and gradient coordination mechanism, it significantly mitigates negative transfer while maintaining task specificity, overcoming the limitations of classic Entire Space Multi-Task Model (ESMM) / Multi-gate Mixture-of-Experts (MMoE) models in correcting sample bias. While existing methods such as ESMM and MMoE have achieved some success in multi-task CTR/CVR modeling, they still fall short in handling extreme sample sparsity and the problem of broken click-transformation causal chains. Although ESMM alleviates sample selection bias in CVR through full-space modeling, its shared tower structure struggles to capture the differentiated feature interactions between CTR and CVR tasks. While MMoE introduces expert mechanisms to improve task specificity, it does not explicitly model higher-order combinatorial effects in click behavior and lacks explicit bias correction for the CVR inference space. To address this, this paper introduces a Cross Network in the CTR branch to explicitly capture higher-order feature interactions while maintaining consistency in shared representations, and integrates a theory-driven Inverse Propensity Scoring (IPS) bias correction mechanism in the CVR branch, thereby surpassing existing methods in both task decoupling and bias control. This framework improves prediction consistency and robustness and provides

a methodological basis for advertising ranking and revenue optimization, and offering methodological support for user behavior prediction in complex interactive environments.

II. JOINT PREDICTION FRAMEWORK DESIGN

This section describes the overall design of the joint CTR and CVR prediction framework. This framework aims to organically combine CTR and CVR prediction through a multi-task learning mechanism, preserving differences between tasks while sharing feature representations, thereby improving model generalization and prediction accuracy. Through a rational network structure design and bias correction strategies, the framework can more accurately capture user behavior patterns, assisting with ad ranking and revenue optimization.

Figure 1 shows the core structure of the multi-task learning ad prediction model. The input layer receives user, advertisement, and context features, and performs feature embedding and fusion through a shared representation layer. The CTR branch processes click-specific features and generates CTR predictions through a cross-network and deep network. The causal bias correction module uses inverse propensity scoring and weight truncation to address sample bias. The CVR branch receives shared features and CTR signals and generates CVR predictions under a deep network and prior constraints. The joint optimization module dynamically balances the CTR and CVR tasks through a weighted loss function and gradient coordination mechanism. It uses the AdamW optimizer for end-to-end training, effectively addressing sample bias and negative transfer issues.

A. BUILDING A SHARED REPRESENTATION LAYER

1) Feature Embedding and Parameterized Design

In the multi-task joint prediction framework, the uniform representation of user features, advertising features, and contextual features is a prerequisite for building a shared layer. In order to eliminate the differences in dimension and scale between different types of features, an embedding layer is first used to map discrete features into a continuous vector space. For the user feature set $U = \{u_1, u_2, \dots, u_m\}$, the advertising feature set $A = \{a_1, a_2, \dots, a_n\}$, and the contextual feature set $C = \{c_1, c_2, \dots, c_k\}$, embedding matrices $E_U \in \mathbb{R}^{|U| \times d}$, $E_A \in \mathbb{R}^{|A| \times d}$, and $E_C \in \mathbb{R}^{|C| \times d}$ are constructed respectively, where d is the embedding dimension, which is set to 64 in this paper to ensure expressiveness while controlling computational complexity [12,13]. The representation of the discrete feature x_i in the embedding space is defined as shown in Equation (1):

$$e_i = E_x \cdot 1_{x_i} \quad (1)$$

Here, 1_{x_i} represents the one-hot vector of feature indices, and E_x represents the embedding matrix of the corresponding feature. During parameter initialization, Xavier initialization is used to ensure consistent variance across dimensions, preventing vanishing or exploding gradients

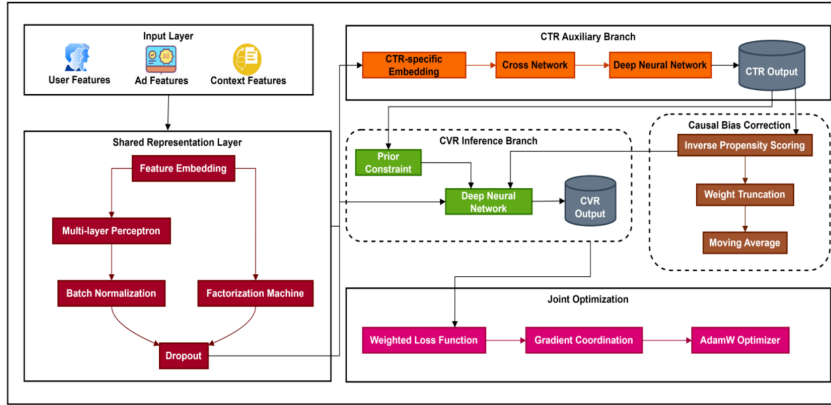


FIGURE 1. The core structure of the multi-task learning advertising prediction model

during early training. The parameters of the embedding layer are shared between the CTR and CVR tasks, and are collaboratively updated through joint backpropagation.

This design avoids the problem of insufficient training in the CVR task due to sample sparsity, enabling the CVR task to obtain stable feature representations using the large-scale CTR data.

2) Multi-Source Feature Fusion and Representation Optimization

After completing the basic embedding, it is necessary to fuse the multi-source information from users, ads, and context to form a shared representation vector h_{shared} [14]. This paper uses a nonlinear transformation based on a multi-layer perceptron (MLP) to achieve feature interaction. Specifically, the three types of embedding vectors are cascaded and input into a two-layer fully connected network, with the number of neurons in each layer set to 256 and 128 respectively, and a ReLU activation function and batch normalization layer are connected after the hidden layer. The process is defined as Equation (2):

$$h_{\text{shared}} = f_{\text{MLP}}([e_U; e_A; e_C]) \quad (2)$$

$[\cdot]$ represents the vector concatenation operation, and $f_{\text{MLP}}(\cdot)$ represents the MLP mapping with nonlinear activation.

To enhance the modeling capability of high-order interactions and reduce the dimensionality expansion problem caused by feature combinations, a factorization machine (FM) structure is applied as an auxiliary in the input layer to calculate the second-order interaction terms between embedded vectors, as shown in Equation (3):

$$FM(e) = \frac{1}{2} \sum_{i=1}^n \sum_{j=i+1}^n \langle v_i, v_j \rangle x_i x_j \quad (3)$$

v_i and v_j are feature embedding vectors, and x_i and x_j are input feature values. The second-order interaction term calculated by the FM is a scalar value. To fuse it with the 128-dimensional vector output by the MLP, we first copy

this scalar to a 128-dimensional vector (i.e., all dimensions have the same value), and then perform element-wise addition with the MLP output, rather than concatenation. This operation maintains the consistency of feature dimensions while injecting high-order interaction information into the shared representation, avoiding dimensionality explosion and enhancing nonlinear expressive power. The final shared representation $h_{\text{shared}} \in \mathbb{R}^{128}$ can be directly input into subsequent multi-task branches. The shared representation layer obtained through the above steps not only provides a unified feature expression for the CTR and CVR tasks but also forms a bidirectional constraint through the multi-task backpropagation mechanism during the training process. The large-scale samples of the CTR task promote rapid convergence of the embedding layer and improve parameter stability; the feedback of the CVR task embeds a signal sensitive to the conversion behavior in the shared layer, compensating for the estimation bias that may be caused by relying solely on the CTR signal [15]. This design solves the problem of sparse CVR samples while alleviating the selection bias caused by inconsistent distribution.

In Figure 2(a), the horizontal, vertical, and depth axes represent the first three principal components obtained from PCA, explaining 32.4%, 21.7%, and 15.3% of the variance, respectively. This demonstrates the separability of user and ad features in low-dimensional space. Samples from different categories A and B exhibit clear clustering in the embedding space, demonstrating that the shared representation layer effectively captures behavioral differences. In Figure 2(b), the horizontal axis represents activation values, and the vertical axis represents probability density. The blue histogram shows the empirical distribution of the model's shared layer, and the red curve is the Gaussian Mixture Model (GMM) fitting result. The distribution exhibits a bimodal nature, corresponding to low- and high-intensity feature activations, respectively. This indicates that this layer is capable of distinguishing feature signals of varying strengths during representation learning.

B. CTR-ASSISTED MODELING

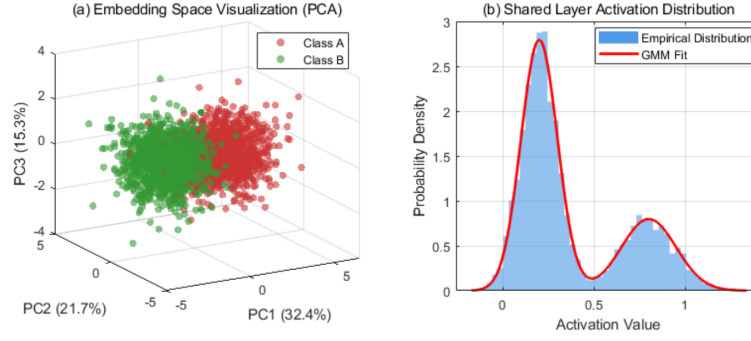


FIGURE 2. Visualization of the shared representation layer. (a). 3D Principal Component Analysis (PCA) visualization of the feature embedding space; (b). Activation distribution and Gaussian mixture fitting curve of the shared representation layer

1) Click Feature Interaction and Deep Structure Construction
After receiving unified input from the shared representation layer, the CTR branch independently builds a prediction network, focusing on modeling the deep interaction of click behavior features. First, the shared layer output h_{shared} is retained in the input phase and combined with a set of CTR-specific features, such as exposure position, ad display frequency, and user click history [16,17]. For these features, the embedding dimension $d = 64$ is consistent with that of the shared layer, and vector quantization is performed using an independent embedding matrix E_{ctr} to ensure that the CTR branch has greater specificity in capturing click patterns. The shared representation and CTR-specific features are then concatenated into a vector $z_{\text{ctr}} = [h_{\text{shared}}; e_{\text{ctr}}]$. Although the CTR branch introduces CTR-specific features such as exposure location and ad display frequency to enhance click pattern modeling capabilities, to avoid these task-specific features interfering with the shared representation layer's general representation ability for CVR tasks, we constrain the fusion method of the shared layer and CTR-specific features: CTR-specific features only enter the cross-network and deep network of the CTR branch through independent embedding matrices and do not participate in the update of the shared representation layer. The shared representation layer h_{shared} consists entirely of three general features: user, ad, and context, and is optimized only through gradient backpropagation between the CTR and CVR backbone networks during multi-task joint training. This design ensures that h_{shared} focuses on stable features that generalize across the entire exposure space, rather than being dominated by CTR-specific signals, thus preserving the ability to model click behavior while ensuring the reliability of the shared representations on which CVR inference depends. In terms of structural design, the CTR branch adopts the form of combining a cross network and a deep network. The cross network is used to explicitly model high-order interactions. For example, as shown in Equation (4):

$$x_{l+1} = x_0 x_l^T w_l + b_l + x_l \quad (4)$$

x_0 represents the input vector, x_l represents the output of the layer l , and w_l and b_l are learnable parameters. This design can gradually form second-order and higher-order

interactive expressions within a limited number of layers, avoiding the dimensionality expansion caused by feature enumeration. The deep network uses a three-layer fully connected structure with neuron sizes of 512, 256, and 128, respectively.

The activation function uses ReLU, and batch normalization and 0.2 dropout are added after each layer to stabilize training and reduce the risk of overfitting. By combining cross-layers with deep layers, the CTR branch can simultaneously capture explicit and implicit click pattern features.

In Figure 3(a), the horizontal axis represents the true CTR value, and the vertical axis represents the predicted CTR value. The blue scatter plots show the correspondence between the model predictions and the true values. The red dashed line represents the ideal prediction baseline, and the light blue area represents the 95% confidence interval. As the true CTR increases, the predicted values tend to converge overall, but there is still some fluctuation in the high CTR range, indicating that the model slightly overestimates samples with high CTR. Figure 3(b) shows that the mean μ of the prediction error distribution is approximately 0.213, and the standard deviation σ is approximately 0.189. This result indicates that the model has a systematic underestimation overall, and the error magnitude is relatively large, rather than a slight bias. Although the distribution shape is close to normal and the standard deviation is small, reflecting that the prediction stability is acceptable, the significant negative bias indicates that the model's response to high CTR samples is still insufficient. The results indicate the presence of a residual bias in the current CTR branch, suggesting that a calibration module could be introduced in future work to further improve the reliability of the click signal.

2) Generation of Stable Click Signals and Prior Guidance

The final output of the CTR branch is passed through the sigmoid function to obtain the click probability estimate $\hat{y}_{\text{ctr}}$, as shown in Equation (5):

$$\hat{y}_{\text{ctr}} = \sigma(W_{\text{out}} h_{\text{ctr}} + b_{\text{out}}) \quad (5)$$

h_{ctr} is the output of the last layer of the deep network, W_{out} and b_{out} are the output layer parameters. The CTR

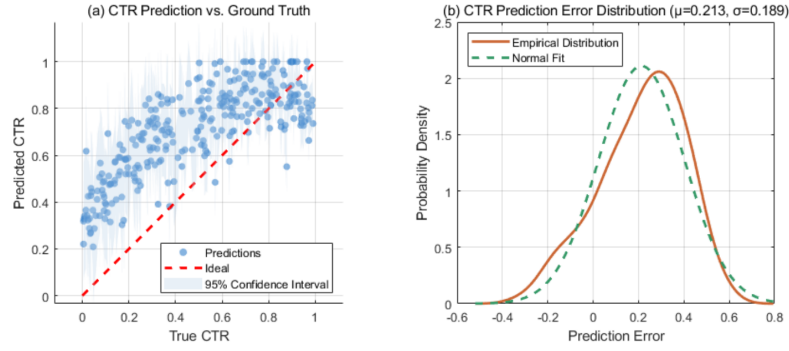


FIGURE 3. CTR-assisted modeling results. (a). Comparison of predicted and true CTR values; (b). CTR prediction error distribution

prediction task is trained using the logarithmic loss function, as shown in Equation (6):

$$L_{\text{ctr}} = -\frac{1}{N} \sum_{i=1}^N [y_i \log \hat{y}_{\text{ctr},i} + (1 - y_i) \log(1 - \hat{y}_{\text{ctr},i})] \quad (6)$$

y_i represents the real click label, and $\hat{y}_{\text{ctr},i}$ represents the model prediction result. The CTR branch relies on large-scale exposure data during training, and can achieve stable convergence within a short iteration. This feature enables the CTR branch to form a reliable click signal at the beginning of training, thereby providing available prior information for the CVR branch. In order to effectively transmit the CTR signal to the CVR branch, this paper adopts two methods in training: one is to use parameter sharing to use the feature representation of the first few layers of the CTR branch as an additional vector of the CVR input; the other is to use soft label guidance to use the CTR output probability as a priori constraint for CVR modeling [18].

Specifically, a regularization term is applied in the CVR loss function, as shown in Equation (7):

$$L'_{\text{cvr}} = L_{\text{cvr}} + \lambda \|\hat{y}_{\text{cvr}} - g(\hat{y}_{\text{ctr}})\|^2 \quad (7)$$

$\hat{y}_{\text{cvr}}$ represents the CVR prediction probability, $g(\cdot)$ is the nonlinear transformation function, and λ is the balancing coefficient (set to 0.1 in this paper). This design ensures that the CVR prediction does not deviate from the prior information provided by CTR, alleviating learning instability caused by sparse conversion samples. The nonlinear transformation function in soft-label guidance is defined as $g(z) = \gamma z + \delta$, where $\gamma = 0.9$ and $\delta = 0.05$. This function aims to map the CTR output probability to the CVR probability scale, since CTR and CVR represent click and conversion probabilities, respectively, which differ significantly in magnitude. Through linear mapping, prior constraints $\lambda \|\hat{y}_{\text{cvr}} - g(\hat{y}_{\text{ctr}})\|^2$ ensure that CVR predictions do not deviate from the signal provided by CTR, while smoothly handling probability differences and enhancing the model's stability under sparse samples.

C. CAUSAL BIAS CORRECTION MODULE

1) Inverse Propensity Score Estimation and Weight Assignment

The propensity score is defined as the CTR model's predicted click probability for any given ad impression sample. This estimation is based on all impression samples, not just those clicked. According to the causal inference theory of inverse propensity weighting (IPW), the true expected CVR can be unbiasedly estimated by dividing the conversion label of a clicked sample by its corresponding propensity score. Therefore, when training the CVR model, we introduce an inverse propensity weight for each click sample [19,20]. This weight is equal to the reciprocal of the propensity score, with a very small constant added to avoid numerical instability. It is important to emphasize that this weight is only used to weight the samples in the CVR loss function, thereby correcting for selection bias caused by training based solely on click samples; we no longer append this weight as an additional feature to the input of the CVR branch. This adjustment makes the model structure more strictly adhere to the theoretical framework of IPW, avoiding training instability issues that might arise from misusing the correction weight as a prediction signal. Furthermore, to control the variance of the weighted loss, we retain the strategies of truncating the weights (limited to between 0 and 20) and using moving averages, but explicitly state that these operations are only used to improve training stability, not as a core component of the bias correction mechanism.

In Figure 4(a), the horizontal axis represents the true CVR value, and the vertical axis represents the predicted CVR value. The red curve shows the prediction results of the uncorrected model, and the green curve shows the prediction results after causal bias correction. It can be seen that the corrected curve is closer to the true trend and has smaller fluctuations, indicating that the correction has effectively reduced the systematic bias.

In Figure 4(b), the horizontal axis represents the iteration number, while the vertical axis represents the loss value on a logarithmic scale. The blue curve illustrates the convergence of the loss after IPS weighting, whereas the orange curve represents the convergence of the standard loss during training. IPS Loss decreases rapidly after about 200 iterations and eventually stabilizes below 0.1, while standard Loss

converges more slowly, eventually reaching about 0.1. This shows that the IPS method has advantages in reducing bias and accelerating convergence.

D. CVR INFERENCE BRANCH

1) Structure Construction and Multi-Source Feature Fusion
Under the constraints of CTR-assisted modeling and causal bias correction signals, the CVR branch independently constructs a prediction network to estimate the conversion probability of all exposure samples. The input consists of the global feature vector h_{shared} output by the shared representation layer, the stable click representation h_{ctr} transmitted by the CTR branch, and the weight constraints provided by the causal correction module. The CVR input representation Equation (8) is formed through concatenation operations:

$$z_{\text{cvr}} = [h_{\text{shared}}; h_{\text{ctr}}] \quad (8)$$

w_{IPS} represents the weight signal after causal correction. This design ensures that CVR modeling not only relies on a shared low-dimensional representation but also applies click priors and distribution correction, thereby indirectly constraining unclicked samples at the input stage.

The CVR branch uses a four-layer fully connected network with hidden layer dimensions of 512, 256, 128, and 64, respectively. The activation function is ReLU, and each layer is followed by batch normalization and a dropout of 0.3 to enhance generalization. Compared to the CTR branch, the CVR branch network is deeper, enabling the extraction of higher-order transformation feature patterns under sparse supervision. During parameter initialization, weights are initialized using the Xavier method to reduce the risk of vanishing or exploding gradients.

2) Conversion Probability Modeling and Loss Constraints

Since non-click samples lack conversion labels, CVR training is based only on click samples. However, through the aforementioned IPS correction and CTR prior constraints, the prediction results can be consistent across the entire exposure space. The CVR loss function is designed to combine cross entropy, IPS weighting, and prior guidance. The specific form is Equation (9):

$$L_{\text{cvr}} = \alpha \left(- \sum_{i=1}^N w(x_i) [y_i \log \hat{y}_{\text{cvr},i} + (1 - y_i) \log(1 - \hat{y}_{\text{cvr},i})] \right) + \beta \|\hat{y}_{\text{cvr}} - g(\hat{y}_{\text{ctr}})\|^2 \quad (9)$$

The first term is the IPS-weighted cross entropy, used for distribution correction; the second term is a prior constraint to ensure consistency between CVR predictions and CTR outputs: α and β take values of 0.7 and 0.3, respectively, to balance the two constraints. Function $g(\cdot)$ uses a linear mapping $g(z) = \gamma z + \delta$ with parameters $\gamma=0.9$ and $\delta=0.05$ to ensure that the CTR prior is applied in a smoothed manner

during CVR training. During optimization, the AdamW optimizer is used with a learning rate of 5×10^{-4} and a weight decay coefficient of 1×10^{-2} . To mitigate the imbalanced effects of transforming sparse samples on training, a positive-negative resampling strategy is implemented, increasing the proportion of positive samples to 20% and dynamically downsampling negative samples to ensure a stable distribution within the batch.

E. MULTI-TASK JOINT OPTIMIZATION

1) Joint Loss Function Construction

In the joint training process of CTR and CVR tasks, if only independent loss functions are relied upon for parallel optimization, gradient conflicts and task imbalance occur. The CTR task has sufficient labels, so the model converges quickly. However, the CVR task has a slower convergence rate because the supervision signal only exists in the click samples. As a result, the shared representation layer gradually shifts towards the CTR optimization direction during training, thereby weakening the prediction ability of CVR [20]. To solve this problem, this paper performs a weighted fusion of the CTR cross entropy loss and the CVR loss corrected by IPS, and achieves adaptive balance through a dynamic weight adjustment mechanism. The overall optimization objective is defined as Equation (10):

$$L_{\text{joint}} = \lambda_1 L_{\text{ctr}} + \lambda_2 L_{\text{cvr}} \quad (10)$$

L_{ctr} is the click prediction loss, L_{cvr} is the conversion prediction loss after IPS correction and prior constraints, and λ_1 and λ_2 are dynamic weight coefficients. In the early stages of training, the weights are set to a balanced ratio so that both tasks are fully trained. As CTR gradually stabilizes and converges, the system adjusts the weights through uncertainty estimation methods, giving CVR a larger proportion in mid- and late-stage optimization, thereby ensuring its predictive power in the exposure space. In specific implementation, the dynamic adjustment of weights does not rely on manual setting, but rather automatically adjusts the coefficient size based on fluctuations through real-time estimation of the variance and gradient amplitude of the task loss. This mechanism ensures that the model can fully utilize the stable signal provided by the CTR task during the joint optimization process.

2) Optimization Process and Gradient Coordination

To mitigate potential gradient conflicts between CTR and CVR tasks during shared representation layer updates, we introduce a gradient coordination mechanism using the Projecting Conflicting Gradients (PCGrad) algorithm. The core idea is that when the gradient directions of two tasks for shared parameters conflict (i.e., the angle between them is greater than 90 degrees), the gradient of one task is projected onto a direction orthogonal to the gradient of the other task, thus eliminating the conflicting components. In practice, we use the gradient of the CTR task as the dominant direction,

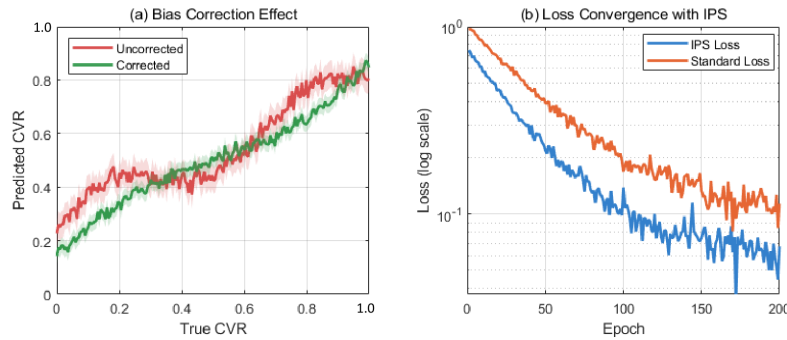


FIGURE 4. Results of the Causal Bias Correction Module. (a). Comparison of Prediction Results Before and After Bias Correction; (b). Convergence Curves of IPS and Standard Loss

correcting the CVR gradient only when a negative correlation is detected (i.e., a negative inner product), ensuring that the shared parameters are updated without affecting CTR optimization. This strategy effectively suppresses negative transfer in multi-task learning while preserving the potential positive knowledge transfer between tasks. Furthermore, we set a cosine similarity threshold of 0 to ensure that the projection operation is triggered only when a genuine gradient conflict occurs, avoiding unnecessary interference with the collaborative optimization process.

AdamW is used as the optimizer to improve the stability of parameter updates. The learning rate is set to 1×10^{-3} , and a learning rate decay strategy is incorporated during training to ensure gradual convergence in mid- and late-stage iterations. Furthermore, to prevent insufficient training of the CVR task due to sample sparsity, a stratified sampling strategy is used to balance the ratio of positive and negative samples, and gradient clipping is combined to suppress gradient explosion caused by fluctuations in IPS weights.

III. JOINT PREDICTION PERFORMANCE EVALUATION

For the joint prediction performance evaluation, the experimental data is derived from real advertising logs, covering user characteristics, ad characteristics, and contextual information over a period of thirty consecutive days. The dataset contains over one billion impression records, of which click samples account for approximately 8% of the total impressions, and conversion samples account for less than 0.5%, exhibiting significant sparseness. User characteristics primarily include demographic attributes and behavioral interests, ad characteristics cover categories, bids, and placements, and contextual information encompasses time, device, and region. To ensure the reliability of the experiment, the data underwent rigorous preprocessing to remove outliers and missing values, and high-dimensional sparse features are processed using hash encoding and embedding vectorization. Training and evaluation are performed using a time-sliced method, with the first 25 days of data serving as the training set, the middle two days serving as the validation set, and the final three days serving as the testing set. This design ensures that the model accurately reflects the distributional and temporal characteristics of the online

system during evaluation.

A. AUC EVALUATION

Area Under the Receiver Operating Characteristic Curve (AUC) measures a model's ability to distinguish between positive and negative samples. In the joint CTR and CVR prediction, AUC calculates the ranking capabilities of click prediction and conversion prediction, respectively. The evaluation first constructs predicted probability and true label pairs, then performs binary sorting on the samples. Based on the sorting results, the true positive rate (TPR) and false positive rate (FPR) are calculated at different thresholds. The model's performance across the full threshold range is statistically analyzed to reflect its overall discrimination effectiveness. Due to the large sample size of CTR, AUC evaluation results are more stable.

However, due to the sparse labels of CVR, its AUC value can reflect the actual improvement of the model after bias correction and task fusion. This metric does not rely on a fixed threshold and is suitable as a global performance benchmark in advertising prediction tasks. Evaluation models include: Logistic Regression (LR), Gradient Boosting Decision Tree (GBDT), FM, Deep Neural Network (DNN), ESMM, MMoE, IPW, and the proposed framework.

Figure 5 shows the performance evaluation of the multi-task CTR/CVR prediction model. Figure 5(a) shows the eight comparison models on the horizontal axis and the AUC score on the vertical axis, which measures the accuracy of CTR and CVR predictions. The results show that the proposed model achieves the highest AUCs on both the CTR and CVR tasks, 0.812 and 0.763, respectively, significantly outperforming traditional DNN, LR, and FM models. This demonstrates the model's superior ability to distinguish between clicks and conversions. Figure 5(b) shows a comparison of the ROC curves, with the FPR on the horizontal axis and the TPR on the vertical axis. The ROC curve for the proposed model is generally higher than that of the baseline model (traditional DNN) and far away from the random reference line. The AUC value reaches 0.912, while the baseline model's is only 0.828, demonstrating that the proposed model is more reliable in classification and discrimination. The data in Figure 5 demonstrates that the

proposed model has a clear advantage in the joint CTR/CVR prediction task.

B. LOGLOSS EVALUATION

LogLoss evaluates the model's probability prediction accuracy, specifically the degree of discrepancy between the predicted probability and the actual label. In the CTR and CVR tasks, the difference between the predicted probability and the actual click and conversion results is calculated, respectively. During evaluation, the predicted probability of all samples is compared with the label, and the loss caused by the prediction deviation for each sample is calculated and averaged across all samples. A smaller LogLoss indicates that the predicted probability is closer to the true distribution, and the model's reference value in ad ranking and revenue optimization scenarios is higher. This metric is particularly critical in multi-task scenarios, as it reveals whether the stable modeling of click probability by the CTR branch provides a reliable prior for the CVR branch. Figure 6 shows the LogLoss performance of different models on the CTR and CVR prediction tasks. The horizontal axis shows the LogLoss value (smaller values are better), and the vertical axis shows the different models. The overall trend shows that traditional models such as LR, GBDT, and FM generally have higher LogLoss, while DNN models and multi-task models such as ESMM, MME, and IPW have improved. MME and IPW both outperform DNN, indicating that multi-task modeling can better capture the correlation between CTR and CVR. In particular, the proposed model achieves the lowest LogLoss on both tasks (CTR 0.365, CVR 0.381), a significant decrease compared to the baseline model, demonstrating its clear advantage in improving prediction accuracy and reducing error.

C. CALIBRATION EVALUATION

Calibration is primarily used to verify the consistency between the model's predicted probabilities and actual occurrence frequencies. In the joint prediction framework, CTR and CVR predictions are bucketed, samples within the predicted probability range are counted, and the proportion of true positive samples within that range is calculated. The model's probability output is evaluated for plausibility by comparing the difference between the predicted values and the actual CVR. If the predicted values and the actual proportions are consistent, the model is well calibrated and suitable for revenue estimation and budget allocation in advertising systems. The evaluation process also focuses on the calibration curves for clicks and conversions to ensure that the CTR and CVR predictions maintain probabilistic consistency under joint modeling.

Figure 7 illustrates the calibration effect of the model in CTR and CVR prediction, as well as the distribution of calibration errors. In Figure 7(a), the horizontal axis shows predicted CTR, and the vertical axis shows actual CTR. The blue dotted curve represents the calibration result of the proposed model, the orange square curve represents the

baseline model ESMM, and the black dashed line represents the ideal perfect calibration line.

It can be seen that the proposed model's predictions are closer to the ideal line, with smaller errors, especially in the low CTR range. Figure 7(b), the horizontal axis shows predicted CVR, and the vertical axis shows actual CVR. Similarly, the proposed model (blue) is closer to the ideal line than the baseline (orange), indicating that causal bias correction and the shared representation layer design effectively improve the reliability of CVR predictions. Figure 7(c), the horizontal axis shows calibration error (predicted value minus true value), and the vertical axis shows probability density. This shows the error distribution of the two models. The blue bars represent the proposed model, and the orange bars represent the baseline. The fitting curves further demonstrate that the proposed model has more concentrated errors and smaller variance, indicating that the model predictions have higher credibility and stability, which helps reduce bias and uncertainty in advertising delivery.

D. NDCG ASSESSMENT

Normalized Discounted Cumulative Gain (NDCG) is used to measure the overall effectiveness of a model in ad ranking scenarios. The evaluation first ranks ads based on the model's predicted values, then calculates the revenue gain at different ranking positions, and finally normalizes the results. NDCG not only considers the accuracy of predictions but also emphasizes the importance of high-ranking ads, thus truly reflecting the actual benefits of the ad system in ranking. For a joint CTR and CVR prediction model, NDCG evaluation can demonstrate whether multi-task optimization improves the final ranking results. In particular, after bias correction, the model can provide more accurate predictions for high-visibility positions.

Table 1 shows the NDCG performance of different models on the CTR and CVR tasks. It can be seen that ranking performance gradually improves with increasing model complexity. LR performs relatively poorly, with an NDCG@1 (CTR) of only 0.421 and an NDCG@5 (CVR) of 0.389. DNN, however, improves to 0.468 and 0.437, respectively, demonstrating that deep learning can better capture ranking features. ESMM and MME further improve, with MME reaching 0.471 for NDCG@5 (CVR). The proposed model achieves the best overall performance, with NDCG@1 (CTR) of 0.538 and NDCG@10 (CVR) of 0.523, significantly outperforming the baseline model and demonstrating the advantages of joint optimization in ranking tasks.

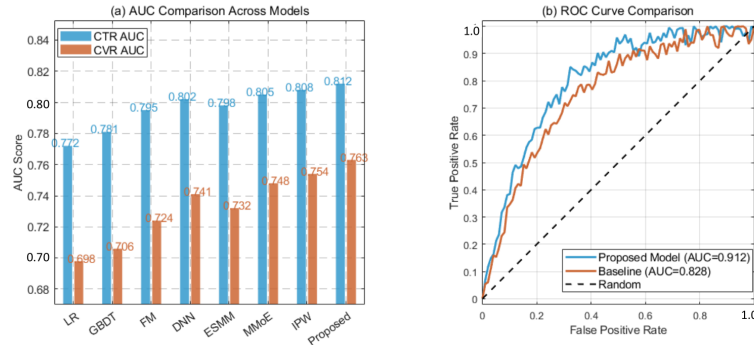


FIGURE 5. AUC Evaluation Chart. (a). AUC Comparison of Different Models; (b). ROC Curve Comparison

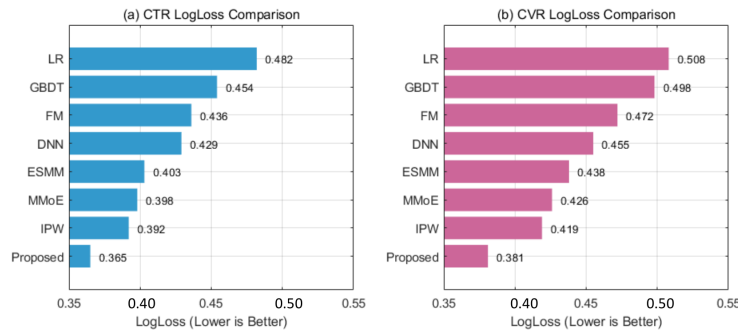


FIGURE 6. Comparison of CTR and CVR Prediction LogLoss. (a). CTR LogLoss Comparison; (b). CVR LogLoss Comparison

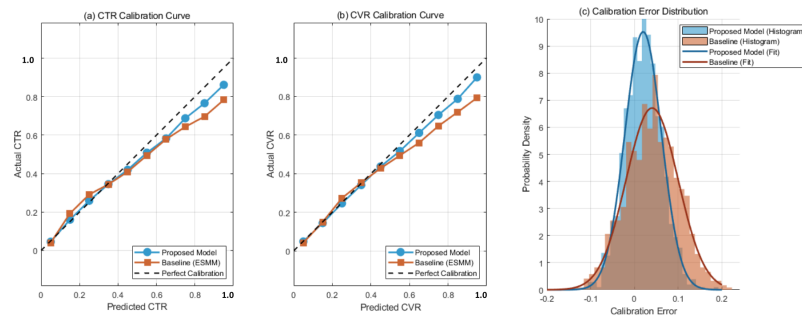


FIGURE 7. Calibration Evaluation. (a). CTR Calibration Curve; (b). CVR Calibration Curve; (c). Calibration Error Distribution

TABLE 1. NDCG performance of different models on CTR and CVR tasks

Model	NDCG@1 (CTR)	NDCG@3 (CTR)	NDCG@5 (CVR)	NDCG@10 (CVR)
LR	0.421	0.436	0.389	0.412
GBDT	0.439	0.452	0.401	0.426
FM	0.448	0.463	0.413	0.437
DNN	0.468	0.482	0.437	0.455
ESMM	0.492	0.506	0.463	0.478
MMoE	0.501	0.517	0.471	0.486
IPW	0.509	0.523	0.479	0.493
Proposed	0.538	0.552	0.509	0.523

E. UPLIFT EVALUATION

Uplift measures a model's direct contribution to business objectives, specifically whether the prediction model can generate additional CTR and CVR gains. This evaluation involves deploying different prediction models in experimental and control groups. The CTR and CVR performance of the two models for the same ad placement are then compared, and the relative improvement ratio is used as the

evaluation result. Unlike traditional offline metrics, Uplift is more closely aligned with business value, demonstrating the advantages of the multi-task framework for revenue optimization and reflecting whether joint CTR and CVR modeling delivers additional gains in real-world applications.

Table 2 shows the model's gains in CTR, CVR, and final revenue. Significant ($p < 0.05$) indicates whether the difference is significant when compared to the proposed model. The traditional LR model, used as the baseline, achieved zero improvement in all categories, indicating no additional benefit. However, GBDT and FM achieved CTR lift of 2.8% and 3.5%, respectively, demonstrating the advantages of tree models and FM in optimizing CTR. The DNN model further improved, achieving a CTR lift of 4.1% and a CVR lift of 3.6%, validating its ability in feature representation. The advantages of the multi-task

learning framework are even more pronounced, with ESMM and MMoE achieving revenue gains of 6.2% and 7.5%, respectively. MMoE achieved a significant CVR lift of 7.2%, demonstrating its effectiveness. The bias-corrected IPW performed even better, achieving a CTR lift of 8.4% and a revenue lift of 8.1%, demonstrating its effectiveness in mitigating exposure bias. Ultimately, the proposed model achieved optimal performance, with a CTR lift of 10.2%, a CVR lift of 9.6%, and a revenue lift of 9.9%, all of which are statistically significant, indicating that this method can bring significant revenue increases in actual advertising businesses.

TABLE 2. Uplift effects of different models in online experiments

Model	CTR Lift (%)	CVR Lift (%)	Revenue Lift (%)	Significant ($p < 0.05$)
LR	0	0	0	No
GBDT	2.8	2.4	2.6	Yes
FM	3.5	3.1	3.3	Yes
DNN	4.1	3.6	3.8	Yes
ESMM	6.5	5.9	6.2	Yes
MMoE	7.8	7.2	7.5	Yes
IPW	8.4	7.9	8.1	Yes
Proposed	10.2	9.6	9.9	—

IV. CONCLUSION

This paper proposes a robust modeling framework based on multi-task learning for the joint prediction of CTR and CVR in online advertising systems. The method first implements unified modeling of user, ad, and contextual features through a shared representation layer. A CTR auxiliary branch is then constructed to provide a stable click signal. A causal bias correction module is applied to mitigate the distribution discrepancy between click samples and the impression space using an inverse propensity score. Furthermore, the CVR inference branch, guided by the CTR prior and bias correction signal, completes full conversion modeling, addressing the unsupervised nature of unclicked samples. Finally, weighted multi-task joint optimization is implemented to dynamically balance the CTR and CVR tasks, mitigating the risk of negative transfer. This integrated method achieves a complete closed-loop from input modeling, causal correction, and optimization strategy, improving the consistency and robustness of the joint prediction. Experimental evaluation demonstrates that the proposed multi-task framework significantly improves the overall performance of joint CTR/CVR prediction. Compared to the baseline model with an AUC of only 0.828, this method achieved an AUC of 0.912, reduced LogLoss to 0.365 and 0.381, respectively, and achieved NDCG@1 of 0.538 on the CTR task. Online experiments further validated its effectiveness: CTR and CVR increased by 10.2% and 9.6%, respectively, for an overall gain of 9.9%, effectively mitigating sample bias and selection bias.

AUTHOR CONTRIBUTIONS

Shihua Zuo designed, collected and analyzed the data, and drafted the manuscript. Shihua Zuo conducted the study, critically revised the manuscript for important intellectual

content, and gave final approval of the version to be published. Shihua Zuo participated fully in the work, take public responsibility for appropriate portions of the content, and agreed to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

CONFLICTS OF INTEREST

The author declares no conflict of interest.

FUNDING

This research received no external funding.

ACKNOWLEDGEMENTS

Not applicable.

REFERENCES

- [1] T. Sangsawang, "Predicting ad click-through rates in digital marketing with support vector machines," *Journal of Digital Market and Digital Currency*, vol. 1, no. 3, pp. 225-246, 2024, doi: 10.47738/jdmcdc.v1i3.20.
- [2] U. Rahardja and Q. Aini, "Evaluating the effectiveness of digital marketing campaigns through conversion rates and engagement levels using anova and chi-square tests," *Journal of Digital Market and Digital Currency*, vol. 2, no. 1, pp. 26-45, 2025, doi: 10.47738/jdmcdc.v2i1.27.
- [3] R. Zhou, C. Liu, J. Wan, Q. Fan, Y. Ren, J. Zhang, et al., "A hybrid neural network architecture to predict online advertising click-through rate behaviors in social networks," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 4, pp. 3061-3072, 2021, doi: 10.1109/tNSE.2021.3102582.
- [4] P. Song, C. Chen, and L. Zhang, "Evaluation model of click rate of electronic commerce advertising based on fuzzy genetic algorithm," *Mobile Networks and Applications*, vol. 27, no. 3, pp. 936-945, 2022, doi: 10.1007/s11036-022-01916-8.
- [5] D. Riana, "Deep Neural Network for Click-Through Rate Prediction," *International Journal of Software Engineering and Computer Systems*, vol. 8, no. 2, pp. 33-42, 2022, doi: 10.15282/ijsecs.8.2.2022.4.0101.
- [6] F. Pratama S and D. Sugianto, "Temporal Patterns in User Conversions: Investigating the Impact of Ad Scheduling in Digital Marketing," *Journal of Digital Market and Digital Currency*, vol. 1, no. 2, pp. 165-182, 2024, doi: 10.47738/jdmcdc.v1i2.10.
- [7] V. Singh, B. Nanavati, K. Kar A, and A. Gupta, "How to maximize clicks for display advertisement in digital marketing? A reinforcement learning approach," *Information Systems Frontiers*, vol. 25, no. 4, pp. 1621-1638, 2023, doi: 10.1007/s10796-022-10314-0.
- [8] E. Xu, Z. Yu, B. Guo, and H. Cui, "Core interest network for click-through rate prediction," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 15, no. 2, pp. 1-16, 2021, doi: 10.1145/3428079.
- [9] M. Cheng and K. Anderson C, "Search engine consumer journeys: exploring and segmenting click-through behaviors," *Cornell Hospitality Quarterly*, vol. 62, no. 2, pp. 198-214, 2021, doi: 10.1177/1938965520924649.
- [10] L. Zhu and C. Zhang, "User Behavior Feature Extraction and Optimization Methods for Mobile Advertisement Recommendation," *Artificial Intelligence and Machine Learning Review*, vol. 4, no. 3, pp. 16-29, 2023, doi: 10.69987/AIMLR.2023.40302.
- [11] M. Lim W, S. Gupta, A. Aggarwal, J. Paul, and P. Sadhna, "How do digital natives perceive and react toward online advertising? Implications for SMEs," *Journal of Strategic Marketing*, vol. 32, no. 8, pp. 1071-1105, 2024, doi: 10.1080/0965254x.2021.1941204.
- [12] S. Wang, C. Hu, and G. Jia, "Deep Learning-Based Saliency Assessment Model for Product Placement in Video Advertisements," *Journal of Advanced Computing Systems*, vol. 4, no. 5, pp. 27-41, 2024, doi: 10.69987/jacs.2024.40503.
- [13] X. Liu and F. Qi, "Research on advertising content recognition based on convolutional neural network and recurrent neural network," *International Journal of Computational Science and Engineering*, vol. 24, no. 4, pp. 398-404, 2021, doi: 10.1504/ijcse.2021.117022.

- [14] K. Zhang, S. Xing, and Y. Chen, "Research on Cross-Platform Digital Advertising User Behavior Analysis Framework Based on Federated Learning," *Artificial Intelligence and Machine Learning Review*, vol. 5, no. 3, pp. 41-54, 2024, doi: 10.69987/aimlr.2024.50304.
- [15] X. Wang, Y. Qiao, J. Xiong, Z. Zhao, N. Zhang, and M. Feng, "Advanced network intrusion detection with tabtransformer," *Journal of Theory and Practice of Engineering Science*, vol. 4, no. 03, pp. 191-198, 2024, doi: 10.53469/jtpes.2024.04(03).18.
- [16] E. Häglund and J. Björklund, "AI-driven contextual advertising: Toward relevant messaging without personal data," *Journal of Current Issues & Research in Advertising*, vol. 45, no. 3, pp. 301-319, 2024, doi: 10.1080/10641734.2024.2334939.
- [17] J. Xie and Z. Chen, "Hierarchical transformer with spatio-temporal context aggregation for next point-of-interest recommendation," *ACM Transactions on Information Systems*, vol. 42, no. 2, pp. 1-30, 2023, doi: 10.1145/3597930.
- [18] P. Lorenz-Spreen, L. Oswald, S. Lewandowsky, and R. Hertwig, "A systematic review of worldwide causal and correlational evidence on digital media and democracy," *Nature human behaviour*, vol. 7, no. 1, pp. 74-101, 2023, doi: 10.1038/s41562-022-01460-1.
- [19] N. Ihzaturrahma and N. Kusumawati, "Influence of integrated marketing communication to brand awareness and brand image toward purchase intention of local fashion product," *International Journal of Entrepreneurship and Management Practices*, vol. 4, no. 15, pp. 23-41, 2021, doi: 10.35631/ijemp.415002.
- [20] R. Andriani, E. Pratiwi D P, and M. Santika I D A D, "Verbal and Non-Verbal Signs in Facial Wash Advertisements: A Semiotic Analysis," *Yavana Bhasha: Journal of English Language Education*, vol. 4, no. 2, pp. 24-29, 2021, doi: 10.25078/yb.v4i2.2768.