B4I: BLACKBOX PLATFORM INTERPRETATION FOR ITEM-LEVEL ASSESSMENT WITH BEHAVIOR FEEDBACK

Anonymous authors
Paper under double-blind review

ABSTRACT

Many creator-facing and analytics-driven online platforms operate as blackbox systems for external stakeholders: ranking and exposure mechanisms, internal models, and the ways in which these components interact with user behavior are inaccessible, while only final feedback such as clicks or conversions is observable. Under this constraint, stakeholders must infer item-level outcomes without access to platform internals or direct model outputs. Existing approaches often rely on opaque neural predictors or aggregate observed statistics, offering limited insight into how user-conditioned interaction patterns shape item-level outcomes. To address this problem, we introduce feature-based neural–logical components into an interpretable behavioral surrogate that models the observable mapping from user and item features to platform-mediated feedback. Building on this idea, we propose B4I, a two-stage framework. Stage I learns the surrogate from historical user–item interactions, producing user-conditioned interaction scores and neural–logical explanatory signals. Stage II evaluates the surrogate over a fixed historical user-group panel and aggregates the resulting outputs to predict item-level popularity labels. B4I thereby transfers behavioral knowledge learned from historical user–item interactions to item-level assessment without requiring observed user–item interaction feedback for the target item, enabling interpretable item-level assessment under blackbox platform constraints. The code is anonymously available at `https://anonymous.4open.science/r/B4I-2B2C/`.

1 INTRODUCTION

Modern online platforms rely on complex ranking, recommendation, and exposure mechanisms to mediate interactions between users and items. For external stakeholders such as content creators, advertisers, and analysts, however, these mechanisms typically operate as a black box. Internal models, parameters, feature representations, ranking policies, and exposure mechanisms are inaccessible, while only observable user and item features together with final interaction feedback, such as clicks or conversions, may be available.

Despite the pervasiveness of this "blackbox platform" setting, extracting meaningful insights for **item-level assessment** from behavioral user–item interaction data remains poorly understood. In practice, creators and analysts are often interested not only in aggregate performance metrics but also in why certain items perform well or poorly, which user groups drive this performance, and how item attributes interact with user characteristics. However, existing analytics pipelines typically rely on simple aggregation heuristics, such as average click-through rate or popularity-based ranking, which fail to capture the structured interaction patterns induced by complex recommendation systems. This motivates the need for principled modeling approaches that are explicitly designed for blackbox platform scenarios where only user behaviors are observable.

A natural alternative is therefore to learn from the observable behavior of the platform-mediated interaction process. Rather than attempting to recover inaccessible platform internals, we seek to model the observable relationship between user features, item features, and historical interaction feedback. Such a behavioral surrogate can capture how different user states respond to different item characteristics while remaining independent of the platform's inaccessible internal architecture.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Interpretability is particularly important in this setting. Conventional deep neural interaction models (Rendle, 2010; Cheng et al., 2016; Guo et al., 2017; Lian et al., 2018; Song et al., 2019) are expressive and capable of learning complex user–item feature interactions, but their learned representations are often difficult to inspect. For creator-facing and analytical applications, predicting whether an item will perform well is only part of the objective; it is also useful to understand which combinations of user and item characteristics contribute to the predicted behavior. Neural–logical models provide a suitable mechanism for this purpose by combining nonlinear interaction modeling with structured explanatory signals.

In this work, we propose a two-stage framework, **B4I**, for item-level assessment under blackbox platform constraints. In **Stage I**, B4I learns a feature-based neural–logical behavioral surrogate from historical user–item interactions. The surrogate produces interaction scores together with structured rule-based explanatory signals, exposing how observable user and item features jointly relate to platform-mediated feedback. In **Stage II**, B4I evaluates each target item over a fixed historical user-group panel and aggregates the resulting user-conditioned signals into an item-level popularity prediction. Conceptually, B4I models an item-level outcome as a function of its predicted response profile across users rather than solely as a function of item attributes:

$$\hat{y}_i = g_\phi\left(\{f_\theta(\mathbf{x}_u, \mathbf{x}_i) \mid u \in \mathcal{U}_{\text{panel}}\}\right), \tag{1}$$

where f_θ is the learned behavioral surrogate, $\mathcal{U}_{\text{panel}}$ is the fixed historical user-group panel, and g_ϕ aggregates the resulting user-conditioned signals for item-level prediction.

Additional discussion of related work appears in Appendix D.

Our contributions are summarized as follows:

- We formulate interpretable item-level assessment under blackbox platform constraints, where platform internals and direct model outputs are inaccessible and only historical platform-mediated interaction feedback is observable.
- We introduce an interpretable neural–logical behavioral surrogate that learns user-conditioned interaction patterns from observable user and item features while providing structured explanatory signals.
- We propose a two-stage framework that transfers interaction-level behavioral knowledge to item-level assessment by evaluating target items over a fixed user-group panel and aggregating the resulting model-generated responses.
- We empirically evaluate B4I on item-level popularity prediction and investigate both its predictive performance and its ability to expose user–item feature patterns associated with item-level outcomes.

2 METHODOLOGY

2.1 PROBLEM SETTING

We study item-level prediction for a platform whose ranking and exposure mechanisms are inaccessible. Historical observations pair user-group features $\mathbf{x}_u$, item features $\mathbf{x}_i$, and platform-mediated feedback y_{ui} (e.g., a click or conversion). An item label y_i is constructed from its aggregate outcome over a predefined observation period. At prediction time, B4I receives the target item's observable features but no realized interaction feedback for that item. It learns behavioral responses from historical interactions and aggregates predicted responses to assess unseen items.

2.2 FRAMEWORK OVERVIEW

B4I trains its stages separately. Stage I learns a neural–logical surrogate of observable platform-mediated feedback and is then frozen. Stage II evaluates each target item over a fixed historical user-group panel and aggregates predicted scores and rule encodings into an item-level label. These evaluations use no observed target-item interactions; the shared panel makes item profiles comparable.

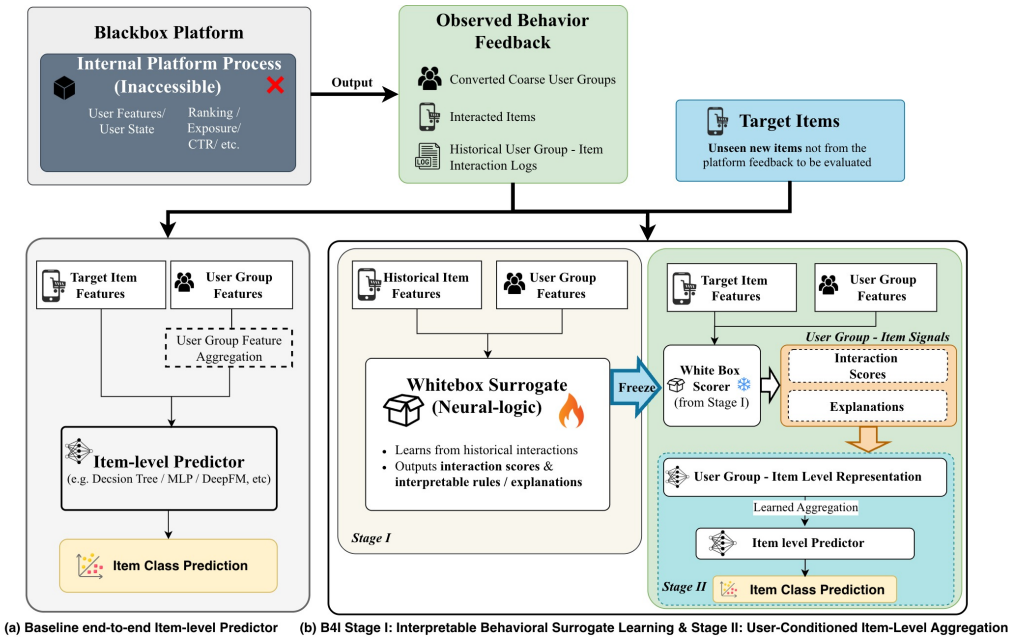


Figure 1: B4I framework and direct baseline. Stage I learns neural–logical rules from historical feedback and uses their activations to compute predicted interaction scores. For each unseen item, Stage II aggregates these scores and encodings of the same rules over a fixed historical user-group panel.

2.3 Stage I: Behavioral Surrogate and Rules

Within the surrogate, user-group and item features first activate learned neural–logical rules; those activations are then used to compute the interaction score:

$$\mathbf{r}_{ui} = R_\theta(\mathbf{x}_u, \mathbf{x}_i), \qquad \hat{s}_{ui} = S_\theta(\mathbf{x}_u, \mathbf{x}_i; \mathbf{r}_{ui}). \tag{2}$$

Thus the rules participate in scoring; they are not explanations generated after a score has been produced. Stage I learns this mapping from historical feedback y_{ui}. An encoding $\mathbf{e}_{ui}^{\text{rule}} = \text{RuleEnc}(\mathbf{r}_{ui})$ summarizes the same activated rules. Stage II receives both the score and this rule encoding. The rules explain predicted user-group responses and affect the item label through Stage II's learned aggregation. Further details appear in Appendix A.

2.4 Stage II: User-Conditioned Aggregation

Let $\mathcal{U}_{\text{panel}}$ be the fixed historical user-group panel. For each target item i, the frozen surrogate uses its activated rules to produce a predicted score and encodes those same rules for every $u \in \mathcal{U}_{\text{panel}}$. The main B4I configuration combines both signals into a user-conditioned representation $\mathbf{z}_{ui} = \phi(\hat{s}_{ui}, \mathbf{e}_{ui}^{\text{rule}})$ and predicts

$$\hat{y}_i = q_\omega(\mathcal{A}(\{g_\psi(\mathbf{z}_{ui}) : u \in \mathcal{U}_{\text{panel}}\})), \tag{3}$$

Here ϕ combines the score with its rule encoding, g_ψ is a shared encoder applied to each user-group representation, $\mathcal{A}$ pools the encoded representations independently of panel order, and q_ω is the item-level classification head. Thus, the item prediction uses predicted user-group responses and their underlying rule evidence. Score-only inputs are considered in ablations; further details appear in Appendix A.

Table 1: Statistics of the processed datasets used for item-level assessment.

	CommAds	**iPinYou** (Liao et al., 2014; Zhang et al., 2014a)
Interactions	1,971,914	20,176
Items	9,707	714
User groups	1,676	8,629
Regions	19	35

3 EXPERIMENTS

3.1 EXPERIMENTAL SETUP

We evaluate B4I on item-level popularity prediction under blackbox platform constraints. The experiments investigate whether behavioral patterns learned from historical user–item interactions can be transferred to improve item-level assessment and whether the resulting intermediate signals provide interpretable evidence for the predictions. The experimental pipeline follows the two-stage framework. In Stage I, the behavioral surrogate is trained on historical user–item interaction records using observable user and item features together with interaction feedback. After training, the surrogate is frozen. In Stage II, each item is evaluated against the same fixed historical user-group panel to generate user-conditioned interaction scores and neural–logical signals. These model-generated signals are then aggregated to predict the item's popularity label. Items are partitioned into disjoint training, validation, and test sets to simulate assessment of unseen items. Target-item interaction feedback from the validation and test partitions is not provided to the Stage II model during inference. Instead, the Stage I surrogate generates the interaction-level signals used for assessment from the target item's observable features and the fixed historical user-group panel.

3.1.1 DATASET

We evaluate our framework on one industrial advertising dataset and one public real-time bidding dataset. The two datasets differ substantially in platform, user-group representation, item features, and performance signals, allowing us to evaluate whether the proposed method generalizes across advertising environments.

CommAds is a proprietary dataset derived from the Google Ads platform, aggregating campaign-level performance feedback delivered to advertisers via the Google Ads analytics dashboard. It contains interactions between advertising creatives and anonymized user groups, together with user, campaign, and creative metadata. User-group attributes include age range, country group, application language, and site information. Item attributes describe the advertisement and its campaign, including content and creative types, product categories, production source, advertiser information, image and video properties, campaign budget, bid amount, and optimization objective. The dataset additionally contains automatically extracted semantic attributes describing the content, visual design, narrative structure, presentation style, and packaging of each creative.

iPinYou The iPinYou dataset (Liao et al., 2014; Zhang et al., 2014a) is a public real-time bidding benchmark collected from a large Chinese demand-side platform. The original dataset contains impression and click logs from multiple advertiser campaigns, together with user, advertisement, and auction-context information. The available fields include geographic and behavioral user attributes, creative and advertiser identifiers, ad-exchange information, slot properties, floor prices, timestamps, domains, and URLs.

Table 1 summarizes the scale of the two datasets in terms of interactions, items, user groups, and regions.

3.2 DATA PROCESSING AND ITEM-LABEL CONSTRUCTION

We process the two datasets into user-group–item interaction records and observable feature representations. For each item, we construct one of four ordered popularity labels from its aggregate feedback during the observation period, with larger labels indicating greater observed popularity. We

split items into disjoint training, validation, and test segments, and use a fixed historical user-group panel to produce the Stage II inputs. The dataset-specific processing and label rules are summarized in Appendix B.1 for COMMADS and Appendix B.2 for iPinYou.

3.3 BASELINES

We compare B4I with 12 direct item-label baselines: Logistic Regression, Decision Tree, Random Forest (Breiman, 2001), MLP, FM (Rendle, 2010), Wide&Deep (Cheng et al., 2016), DeepFM (Guo et al., 2017), FinalMLP (Mao et al., 2023), DCNv2 (Wang et al., 2021), AutoInt (Song et al., 2019), FT-Transformer (Gorishniy et al., 2021), and TabM (Gorishniy et al., 2024). These baselines predict one label per item from item attributes and aggregated historical user-group features using the same item partitions. At inference, they do not observe interaction feedback for the target item. Model descriptions and available implementation details appear in Appendix C.

3.4 EVALUATION METRICS

We evaluate item-level assessment using both ordinal and classification metrics. For ordinal evaluation, we report **Mean Squared Error (MSE)** and **Mean Absolute Error (MAE)** between the predicted and ground-truth class indices. Since the four popularity classes are ordered, these metrics reflect not only whether a prediction is incorrect, but also how far it deviates from the true class.

For classification evaluation, we report the weighted-average **F1 Score**, where the contribution of each class is weighted by its support to account for class imbalance. We additionally report **macro-averaged AUC (AUC$_{\mathrm{macro}}$)**, computed under a one-vs-rest formulation and averaged equally across the four classes. AUC$_{\mathrm{macro}}$ measures the model's ability to discriminate each popularity class from the remaining classes while giving equal importance to all classes regardless of their frequency.

3.5 OVERALL PERFORMANCE

Tables 2 and 3 report item-level prediction on CommAds and iPinYou. We compare one B4I configuration with direct item-label baselines. Each entry is the mean $\pm$ standard deviation over three random seeds.

CommAds B4I improves all four metrics over the best direct baseline for each metric. It achieves an MSE of 0.8648 versus 0.9556 for Random Forest, an MAE of 0.7632 versus 0.7957 for TabM, an F1 Score of 0.4041 versus 0.3165 for Wide&Deep, and an AUC$_{\mathrm{macro}}$ of 0.6561 versus 0.5672 for Random Forest.

iPinYou B4I obtains the lowest MSE (0.6806, versus 0.7500 for DeepFM) and MAE (0.5814, versus 0.6411 for FinalMLP). Its AUC$_{\mathrm{macro}}$ is 0.8118, compared with 0.7689 for MLP. The F1 Score gain is smaller: 0.5105 for B4I versus 0.5081 for DeepFM.

3.6 ABLATION STUDY

We replace B4I's Stage I neural–logical scorer with nine interaction models while holding the Stage II aggregation method fixed. We also compare the default scorer with and without passing its rule encodings to Stage II (Table 4). All configurations receive predicted interaction scores; only the B4I scores-plus-rules row additionally uses rule encodings during aggregation.

With scores alone, the B4I scorer has the lowest MSE (0.8624) and MAE (0.7613), and exceeds every replacement on F1 and AUC$_{\mathrm{macro}}$. Adding rule inputs improves F1 from 0.3583 to 0.4041 and AUC$_{\mathrm{macro}}$ from 0.6416 to 0.6561, while MSE changes from 0.8624 to 0.8648 and MAE from 0.7613 to 0.7632. Thus, the rule inputs improve classification measures here without improving ordinal error. Both B4I variants retain Stage I rules that explain predicted user-group–item responses; the replacement scorers do not generate these neural–logical explanations. The table compares complete Stage I signal configurations, not the effect of any individual rule.

Table 2: Ordinal-error metrics (mean ± standard deviation).

Method	CommAds		iPinYou	
	MSE ↓	MAE ↓	MSE ↓	MAE ↓
Logistic Regression	1.2012 ± 0.0000	0.8502 ± 0.0000	0.9444 ± 0.0000	0.7484 ± 0.0000
Decision Tree	1.1796 ± 0.0000	0.8352 ± 0.0002	1.3333 ± 0.0000	0.8054 ± 0.0000
Random Forest	0.9556 ± 0.0140	0.8042 ± 0.0005	0.9444 ± 0.0000	0.7520 ± 0.0015
MLP	1.0399 ± 0.0062	0.8028 ± 0.0041	0.8426 ± 0.0424	0.6747 ± 0.0216
FM	1.0234 ± 0.0445	0.8019 ± 0.0134	0.7870 ± 0.0526	0.6776 ± 0.0117
Wide&Deep	1.0437 ± 0.0263	0.8013 ± 0.0075	0.8380 ± 0.0802	0.6744 ± 0.0099
DeepFM	1.0382 ± 0.0345	0.8060 ± 0.0174	0.7500 ± 0.0417	0.6658 ± 0.0114
FinalMLP	1.0636 ± 0.0422	0.8008 ± 0.0068	0.8519 ± 0.0656	0.6411 ± 0.0371
DCNv2	1.0475 ± 0.0345	0.8045 ± 0.0102	0.8704 ± 0.0212	0.6792 ± 0.0624
AutoInt	1.0292 ± 0.0153	0.8004 ± 0.0057	0.8981 ± 0.1005	0.7496 ± 0.0354
FT-Transformer	1.0282 ± 0.0305	0.7979 ± 0.0128	0.9398 ± 0.0212	0.7847 ± 0.0260
TabM	1.0120 ± 0.0060	0.7957 ± 0.0039	0.8889 ± 0.0278	0.7098 ± 0.0533
B4I	**0.8648 ± 0.0269***†	**0.7632 ± 0.0076****†	**0.6806 ± 0.0695**	**0.5814 ± 0.0570**

Table 3: Classification metrics (mean ± standard deviation).

Method	CommAds		iPinYou	
	F1 ↑	$\text{AUC}_{\text{macro}}$ ↑	F1 ↑	$\text{AUC}_{\text{macro}}$ ↑
Logistic Regression	0.3124 ± 0.0000	0.5424 ± 0.0000	0.4278 ± 0.0000	0.7510 ± 0.0000
Decision Tree	0.3006 ± 0.0000	0.5348 ± 0.0014	0.3653 ± 0.0000	0.6843 ± 0.0000
Random Forest	0.3068 ± 0.0055	0.5672 ± 0.0016	0.4649 ± 0.0121	0.7429 ± 0.0069
MLP	0.3135 ± 0.0059	0.5446 ± 0.0029	0.4917 ± 0.0455	0.7689 ± 0.0153
FM	0.2914 ± 0.0249	0.5567 ± 0.0087	0.4952 ± 0.0065	0.7585 ± 0.0025
Wide&Deep	0.3165 ± 0.0079	0.5575 ± 0.0089	0.4588 ± 0.0392	0.7517 ± 0.0293
DeepFM	0.2896 ± 0.0249	0.5555 ± 0.0067	0.5081 ± 0.0073	0.7672 ± 0.0052
FinalMLP	0.2975 ± 0.0239	0.5476 ± 0.0072	0.5014 ± 0.0405	0.7613 ± 0.0108
DCNv2	0.2973 ± 0.0102	0.5482 ± 0.0029	0.4925 ± 0.0284	0.7545 ± 0.0051
AutoInt	0.2921 ± 0.0254	0.5526 ± 0.0059	0.3585 ± 0.0534	0.6766 ± 0.0322
FT-Transformer	0.2956 ± 0.0320	0.5476 ± 0.0196	0.4156 ± 0.0616	0.6830 ± 0.0245
TabM	0.2885 ± 0.0122	0.5631 ± 0.0025	0.4279 ± 0.0309	0.7577 ± 0.0164
B4I	**0.4041 ± 0.0157****†	**0.6561 ± 0.0104****†	**0.5105 ± 0.0021**	**0.8118 ± 0.0149**

Bold: best; *underline*: second best within each dataset and metric. CommAds B4I seed-level significance: * raw $p < 0.05$, ** raw $p < 0.01$; † survives Holm correction.

Table 4: Stage I scorer and rule-input ablation on CommAds with the Stage II aggregation method fixed (mean ± standard deviation).

Stage I configuration	MSE ↓	MAE ↓	F1 ↑	$\text{AUC}_{\text{macro}}$ ↑
AutoInt	0.9835 ± 0.0116	0.7860 ± 0.0117	0.2990 ± 0.0184	0.5850 ± 0.0049
DCNv2	0.9608 ± 0.0179	0.7739 ± 0.0119	0.3131 ± 0.0126	0.6112 ± 0.0028
DeepFM	1.0348 ± 0.0203	0.8040 ± 0.0007	0.2931 ± 0.0124	0.5590 ± 0.0027
FinalMLP	0.9673 ± 0.0305	0.7695 ± 0.0033	0.3236 ± 0.0162	0.5910 ± 0.0146
FM	1.0062 ± 0.0206	0.7954 ± 0.0120	0.3002 ± 0.0040	0.5803 ± 0.0126
FT-Transformer	1.0372 ± 0.0127	0.8061 ± 0.0085	0.2816 ± 0.0047	0.5662 ± 0.0073
MLP	1.0024 ± 0.0124	0.7887 ± 0.0008	0.3155 ± 0.0080	0.5810 ± 0.0041
TabM	1.0361 ± 0.0182	0.7986 ± 0.0077	0.2817 ± 0.0027	0.5651 ± 0.0033
Wide&Deep	0.9604 ± 0.0052	0.7776 ± 0.0136	0.2976 ± 0.0372	0.6020 ± 0.0043
B4I (scores only)	$\mathbf{0.8624 \pm 0.0192}$	$\mathbf{0.7613 \pm 0.0065}$	0.3583 ± 0.0359	0.6416 ± 0.0319
B4I (scores + rules)	0.8648 ± 0.0269	0.7632 ± 0.0076	$\mathbf{0.4041 \pm 0.0157}$	$\mathbf{0.6561 \pm 0.0104}$

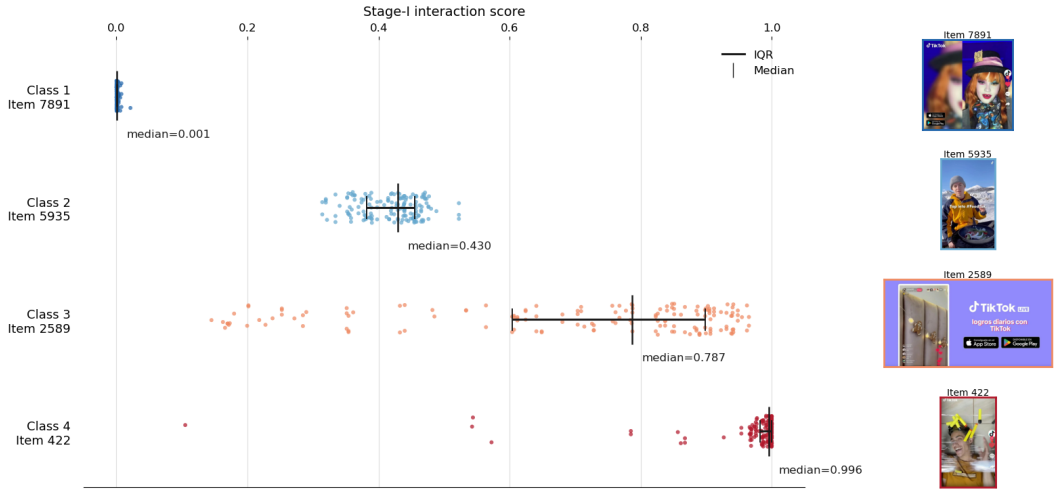


Figure 2: Stage-I interaction-score profiles for four correctly predicted items over the same historical user-group panel. Each point is one predicted user-group–item score; black intervals show the interquartile range (IQR), and vertical markers show the median. Thumbnails show the image creative or an image associated with a video creative.

3.7 Case Study on Interpretability

Beyond aggregate predictive performance, we examine how the interpretable Stage-I surrogate provides evidence for B4I's item-level predictions. We focus on four correctly predicted items from the CommAds test set, one from each item class. The Stage-I surrogate evaluates each item against the same historical user-group panel. We first compare the resulting score distributions, then examine their neural–logical rule signals.

Case 1: Behavioral profiles across item classes The samples in Fig. 2 span an Argentina fashion-themed image creative (Item 7891), a Malaysia outdoor-cooking video (Item 5935), a Mexico Spanish-language TikTok promotion (Item 2589), and a Brazil Portuguese comedy video with app-download prompts (Item 422). These brief descriptions come from creative metadata and annotations; all four items share the same hard-sell vertical category.

Their Stage-I score profiles differ markedly. The Class-1 image creative has scores concentrated near zero (median 0.001). The Class-2 outdoor-cooking video has a moderate, more dispersed profile (median 0.430). The Class-3 promotion has a still broader profile with both low- and high-scoring user groups (median 0.787). The Class-4 comedy video is concentrated near one, apart from a few low-scoring user groups (median 0.996).

The class ordering of these selected examples is consistent with the shift in their predicted response profiles. Stage II uses the full set of user-conditioned scores and their rule-based representations to predict the item label; the medians shown in the figure summarize only one aspect of that input.

Case 2: Aggregating Neural–Logical Evidence at the Item Level We next investigate whether the UI-level neural–logical explanations can be aggregated into an interpretable account of an item's behavioral profile. For each neural–logical rule R_r, Stage I produces a signed contribution $c_{ui,r}$ for user–item pair (u, i). Since each target item is evaluated over the same fixed user group $\mathcal{U}$, we summarize the evidence associated with rule R_r as

$$\bar{c}_{i,r} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} c_{ui,r}, \qquad P^{+}_{i,r} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \mathbb{I}[c_{ui,r} > 0],$$

where $\bar{c}_{i,r}$ measures the average signed evidence associated with the rule and $P^{+}_{i,r}$ measures how consistently it provides positive evidence across the user group. Together, these statistics provide an item-level summary of the UI-level neural–logical evidence.

For Item 422, the analysis reveals strong positive neural–logical evidence across the user group. As shown in Fig. 2, the item is predicted as Class 4 and exhibits a predominantly high-response behavioral profile, with a median Stage-I interaction score of 0.996 and 99.35% of scores above 0.9. Two representative positive rules have mean signed contributions of 2.754 and 2.978 across the user group. Representative conditions in one rule associated with this positive evidence are

$$\begin{aligned}&\text{Region} = \text{Brazil} \wedge \text{Content type} = \text{Mago UGC} \wedge \text{Category} = \text{Hard sell}\\ &\wedge \text{Aspect ratio} = \text{Portrait} \wedge \text{Story} = \text{Comedy} \wedge \text{Creative element} = \text{Interaction}\\ &\wedge \text{Editing} = \text{Fast-paced / beat-synchronized} \wedge \text{Sound} = \text{Transition effects}.\end{aligned}$$

The aggregated contribution indicates that the represented neural–logical evidence repeatedly supports the UI-level responses that constitute the item's high-response behavioral profile. This provides an item-level interpretation of the behavioral evidence associated with its Class-4 prediction.

Item 2589 provides a contrasting example. The item is predicted as Class 3 and exhibits a more heterogeneous behavioral profile, with a median interaction score of 0.787 and 78.1% of scores above 0.5. Its positive rules have mean signed contributions of 2.013, 1.366, and 0.730, while several opposing rules have mean contributions of -1.071, -0.491, -1.665, and -0.961. Thus, aggregation of UI-level explanations exposes both supporting and opposing neural–logical evidence underlying the behavioral profile.

Case 3: Cross-User Interpretation of Neural–Logical Evidence While Case 2 aggregates neural–logical evidence to characterize an item as a whole, we next examine how this evidence varies across the individual user contexts that constitute the behavioral profile. We focus on Item 2589 because its Stage-I scores exhibit substantial cross-user variation. We select actual users near the 10th, 50th, and 90th percentiles of its score distribution, whose interaction scores are 0.252, 0.787, and 0.939, respectively.

The rule-level evidence also varies substantially across these user contexts. For one representative positive rule, the signed contribution is 0.450, 1.450, and 1.898 for the low-, median-, and high-response users, respectively. Another rule contributes 0.171, 0.583, and 1.477. A third rule contributes 1.028, 2.391, and 2.353, showing strong positive evidence for the latter two user contexts.

The corresponding rule instances further illustrate the differences between these evaluations. Because the target item is unchanged, its item-side conditions remain fixed, including

$$\text{Region} = \text{Mexico}, \qquad \text{Video Style} = \text{Modern}, \qquad \text{Special Effects} = \text{Text Effects}$$

In contrast, the observed user-side conditions differ:

	Country Group	App Language
Low Response User Group	Sub-Saharan	English
Median Response User Group	Middle East	Turkish
High Response User Group	Latin America	Spanish

These examples illustrate how the same item can receive different neural–logical evidence under different user contexts. Importantly, the comparison is observational: multiple user-side characteristics may differ between the selected users, so the score differences should not be attributed to any individual feature. Instead, the analysis shows how heterogeneous UI-level explanations collectively give rise to the item's behavioral profile, which is subsequently aggregated for item-level assessment.

Case 4: Rule filtering versus learned aggregation for class-4 discovery On the same CommAds test items, we compare a deterministic class-4 filter using Stage I rules with B4I's learned Stage II aggregation of predicted user-group scores and rule representations (Table 5).

Table 5: Class-4 discovery on CommAds: deterministic filtering with B4I's learned rules versus B4I's Stage II aggregation. R, P, and F1 denote class-4 recall, precision, and F1.

	Global			Mexico			Brazil			Malaysia		
Method	R	P	F1	R	P	F1	R	P	F1	R	P	F1
Rule filter	0.8293	0.2064	0.3305	0.8148	0.2500	0.3826	0.7308	0.2754	0.4000	0.9444	0.2297	0.3696
B4I scores + rules	0.4512	0.4302	0.4405	0.4074	0.4074	0.4074	0.3077	0.5000	0.3810	0.6111	0.4583	0.5238

The two approaches offer different class-4 retrieval tradeoffs. Globally, the rule filter recalls 0.8293 of class-4 items versus 0.4512 for B4I's aggregator, but its precision is 0.2064 versus 0.4302 and its F1 is 0.3305 versus 0.4405. The filter's precision is only modestly above the global class-4 prevalence of 0.169, so the high recall comes with many non-class-4 candidates. This recall advantage persists across Mexico, Brazil, and Malaysia. The aggregator achieves higher precision in all three regions and higher F1 globally and in Mexico and Malaysia; the rule filter has a slightly higher F1 in Brazil (0.4000 versus 0.3810). Thus, reusing Stage I rules offers a broad screening signal, whereas the learned aggregation yields a smaller, more precise class-4 selection. The rule filter is a diagnostic use of B4I's intermediate rules, not an independent trained competitor.

4 Discussion

B4I should be interpreted as a model of observable platform-mediated behavior, rather than a reconstruction or causal explanation of an inaccessible platform. The interaction feedback used to train Stage I may jointly reflect user preferences, ranking and exposure policies, interface effects, and other unobserved platform mechanisms. Consequently, the learned neural–logical rules describe regularities in the observed interaction process, summarizing which combinations of item characteristics and user-group contexts are associated with stronger or weaker predicted responses

These rules also support exploratory uses beyond explaining individual predicted scores. For an item, their contributions across the fixed user-group panel can identify groups for which Stage I predicts a stronger response, providing hypotheses for audience targeting. Conversely, comparing rule contributions across items can reveal which observable item characteristics are associated with stronger predicted responses from a particular user group. The rules can also serve as transparent filters for retrieving candidate items with specified patterns. As Case 4 illustrates, such filtering may favor recall over precision, so it is best treated as a screening tool rather than a substitute for learned aggregation. These uses describe patterns in predicted platform-mediated responses, not established user preferences or guaranteed targeting outcomes.

5 Conclusion

We introduced **B4I** for item-level popularity assessment from historical platform-mediated feedback. Stage I learns a neural–logical behavioral surrogate that produces predicted user-group–item scores and their rule-based evidence. Stage II evaluates each target item against a fixed historical user-group panel and aggregates the resulting scores and rule signals into a class prediction, without observing the target item's future interactions.

B4I achieves the best mean MSE, MAE, F1, and $\mathrm{AUC}_{\mathrm{macro}}$ among the compared methods on CommAds and iPinYou. The CommAds improvements have seed-level significance after Holm correction; on iPinYou, the F1 margin is small. The CommAds ablation shows that the choice of Stage I scorer affects item-level results and that supplying its rules improves F1 and $\mathrm{AUC}_{\mathrm{macro}}$ while slightly increasing ordinal error. The case studies show how the learned rules reveal patterns in predicted responses across user groups and can inform item filtering and targeting.

AI Use Statement

The authors used generative AI tools to help draft portions of the manuscript, including descriptions of the methodology and results, and to polish and proofread the text for clarity, grammar, and con-

sistency. The authors reviewed and revised the AI-assisted writing and take responsibility for the final content of this paper, including its claims and citations.

REFERENCES

Osbert Bastani, Carolyn Kim, and Hamsa Bastani. Interpretability via model extraction. *arXiv preprint arXiv:1706.09773*, 2017.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS)*, pp. 7–10. ACM, 2016.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. In *Advances in Neural Information Processing Systems*, volume 34, pp. 18932–18943, 2021.

Yury Gorishniy, Akim Kotelnikov, and Artem Babenko. TabM: Advancing tabular deep learning with parameter-efficient ensembling. *arXiv preprint arXiv:2410.24210*, 2024.

Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: A factorization-machine based neural network for ctr prediction. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pp. 1725–1731. International Joint Conferences on Artificial Intelligence Organization, 2017. doi: 10.24963/ijcai.2017/239.

Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 1661–1670. ACM, 2015. doi: 10.1145/2806416.2806504.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1754–1763. ACM, 2018. doi: 10.1145/3219819.3220023.

Hairen Liao, Linglong Peng, Zhiyong Liu, and Xuehua Shen. iPinYou global RTB bidding algorithm competition dataset. In *Proceedings of the 8th International Workshop on Data Mining for Online Advertising (ADKDD)*, pp. 1–6. ACM, 2014.

Kelong Mao, Jieming Zhu, Liangcai Su, Guohao Cai, Yuru Li, and Zhenhua Dong. Finalmlp: An enhanced two-stream mlp model for ctr prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 4552–4560, 2023.

Steffen Rendle. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pp. 995–1000. IEEE, 2010. doi: 10.1109/ICDM.2010.127.

Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1161–1170. ACM, 2019. doi: 10.1145/3357384.3357925.

Ruoxi Wang, Rakesh Shivanna, Derek Z. Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed H. Chi. Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *Proceedings of the Web Conference 2021*, pp. 1785–1797, 2021.

Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. Explainable reasoning over knowledge graphs for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 5329–5336, 2019. doi: 10.1609/aaai.v33i01.33015329.

David H. Wolpert. Stacked generalization. In *Neural Networks*, volume 5, pp. 241–259, 1992. doi: 10.1016/S0893-6080(05)80023-1.

Yikun Xian, Zuohui Fu, S. Muthukrishnan, Gerard de Melo, and Yongfeng Zhang. Reinforcement knowledge graph reasoning for explainable recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 285–294. ACM, 2019. doi: 10.1145/3331184.3331203.

Yikun Xian, Zuohui Fu, Handong Zhao, Yingqiang Ge, Xu Chen, Qiaoying Huang, Shijie Geng, Zhou Qin, Gerard de Melo, S. Muthukrishnan, and Yongfeng Zhang. Cafe: Coarse-to-fine neural symbolic reasoning for explainable recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 1645–1654. ACM, 2020. doi: 10.1145/3340531.3412038.

Zhichao Xu, Hansi Zeng, Juntao Tan, Zuohui Fu, Yongfeng Zhang, and Qingyao Ai. A reusable model-agnostic framework for faithfully explainable recommendation and system scrutability. *ACM Transactions on Information Systems*, 42(1):1–29, 2023. doi: 10.1145/3605357.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep sets. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Weinan Zhang, Shuai Yuan, and Jun Wang. Real-time bidding benchmarking with iPinYou dataset. In *arXiv preprint arXiv:1407.7073*, 2014a.

Xiaoyu Zhang, Shaoyun Shi, Yishan Li, Weizhi Ma, Peijie Sun, and Min Zhang. Feature-enhanced neural collaborative reasoning for explainable recommendation. *ACM Transactions on Information Systems*, 43(1):1–33, 2024. doi: 10.1145/3690381.

Yongfeng Zhang and Xu Chen. Explainable recommendation: A survey and new perspectives. *Foundations and Trends in Information Retrieval*, 14(1):1–101, 2020. doi: 10.1561/1500000066.

Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 83–92. ACM, 2014b. doi: 10.1145/2600428.2609579.

A Additional Methodology Details

Stage I objective The training set is $\mathcal{D}_{\text{int}} = \{(\mathbf{x}_u, \mathbf{x}_i, y_{ui})\}$, where feedback is an observable outcome of platform-mediated interaction, not necessarily the output of an internal prediction model. For binary feedback, Stage I minimizes

$$\mathcal{L}_{\text{int}} = -\sum_{(u,i)\in\mathcal{D}_{\text{int}}} [y_{ui}\log\hat{s}_{ui} + (1-y_{ui})\log(1-\hat{s}_{ui})]. \tag{4}$$

Rule contributions and interaction scores The learned rules yield $\mathbf{r}_{ui} = [r_{ui}^{(1)}, \ldots, r_{ui}^{(L)}]$, where $r_{ui}^{(\ell)}$ is the activation of rule ℓ for pair (u, i). These activations are internal to the surrogate's score computation. For each activated rule, we denote its signed scoring evidence by $c_{ui,\ell}$: positive values support a higher interaction score, negative values oppose it, and the magnitude describes strength on the scorer's internal scale. The surrogate combines this rule-level evidence to produce the predicted interaction score, which we write schematically as

$$\hat{s}_{ui} = \text{Score}_\theta\left(\mathbf{x}_u, \mathbf{x}_i; \mathbf{c}_{ui}\right), \qquad \mathbf{c}_{ui} = (c_{ui,1}, \ldots, c_{ui,L}). \tag{5}$$

Here Score_θ denotes the learned combination and output mapping, not an assumed unweighted sum of the contributions. Thus the rule evidence participates in producing $\hat{s}_{ui}$ rather than being added as a post-hoc explanation. The signed contributions analyzed in the case studies refer to this Stage I prediction; they do not directly decompose Stage II's item-class output.

Rule encoding and panel evaluation An encoding of the same activated rules, $\mathbf{e}_{ui}^{\text{rule}} = \text{RuleEnc}(\mathbf{r}_{ui})$, is also passed to Stage II alongside $\hat{s}_{ui}$. For the panel $\mathcal{U}_{\text{panel}} = \{\mathbf{x}_{u_1}, \ldots, \mathbf{x}_{u_N}\}$, Stage I evaluates the pairs $\mathcal{S}_i = \{(\mathbf{x}_u, \mathbf{x}_i) : u \in \mathcal{U}_{\text{panel}}\}$. These constructed pairs need not be observed interactions. Stage II applies a shared transformation to each $\mathbf{z}_{ui}$, aggregates over the fixed panel, and predicts the class from the resulting item representation.

Stage II objective With $p_{i,c}$ denoting the predicted probability of class c and $y_i \in \{1, \ldots, K\}$ the item label, the aggregation model is trained with

$$\mathcal{L}_{\text{CE}} = -\frac{\sum_i w_{y_i} \log p_{i,y_i}}{\sum_i w_{y_i}}, \qquad w_c = \frac{N}{K n_c}, \tag{6}$$

$$\mathcal{L}_{\text{ord}} = \frac{1}{B} \sum_i \left(\sum_{c=1}^{K} c\, p_{i,c} - y_i \right)^2, \tag{7}$$

$$\mathcal{L}_{\text{Stage II}} = \mathcal{L}_{\text{CE}} + \lambda_{\text{MSE}} \mathcal{L}_{\text{ord}}. \tag{8}$$

Here N is the number of training items, n_c is the number in class c, and B is the batch size. The ordinal term is optional ($\lambda_{\text{MSE}} = 0$ gives weighted cross-entropy alone).

B Dataset Processing and Item-Label Construction

We transform both datasets into a common two-stage learning setting. Each processed dataset consists of: (1) user–item interactions for training the Stage I scorer, (2) a fixed user-group panel, (3) item-level labels for training Stage II, and (4) training, validation, and test item segments. The Stage I model estimates the response of a user-group profile to an item, while Stage II aggregates the resulting user-group response scores into an item-level prediction.

B.1 CommAds Processing

In CommAds, an item corresponds to an advertising creative and a user corresponds to an anonymized user group. We retain the available user, campaign, creative, semantic-content, and packaging features after categorical encoding. Identifier fields are excluded from the direct item-level baselines to prevent the models from relying on item or user-group memorization.

Item-label assignment. Let q_i denote the calibrated seven-day user count of item i. Because advertising performance can differ substantially across markets, items are ranked separately within each of the 18 largest campaign regions. Within every region, items are sorted in descending order of q_i and partitioned using the cumulative percentile boundaries

$$5\%,\ 10\%,\ 16.7\%,\ 33.3\%,\ 50\%,\ 66.7\%,\ 83.3\%,\ 90\%,\ 95\%,\ 100\%.$$

The ten resulting groups define fine-grained ordinal labels, with label 10 assigned to the highest-performing items and label 1 to the lowest-performing items. For the four-class task reported in our experiments, these fine-grained labels are mapped to coarse labels as

$$y_i = \begin{cases} 1, & y_i^{\text{fine}} \in \{1, 2, 3\}, \\ 2, & y_i^{\text{fine}} \in \{4, 5\}, \\ 3, & y_i^{\text{fine}} \in \{6, 7\}, \\ 4, & y_i^{\text{fine}} \in \{8, 9, 10\}. \end{cases} \tag{9}$$

Accordingly, a larger label represents stronger post-delivery item performance.

Data partitioning and use. The processed CommAds benchmark contains 3,884 training items, 970 validation items, and 969 test items. The validation and test items are disjoint from the training items. For the training items, interactions are further divided into two partitions. The first partition is used to train the Stage I interaction scorer, whereas the second is reserved for constructing the inputs used to train the Stage II aggregation network. This prevents Stage II from being trained on the same interaction observations used to optimize Stage I.

A fixed user-group panel containing 155 user groups is used in the reported experiments. After Stage I training, every item is paired with every member of this panel to produce a set of user-group response scores. The response sets for the training items are used to train Stage II, while the validation and test sets are used for model selection and final evaluation, respectively.

B.2 iPinYou Processing

The original iPinYou records describe individual advertising opportunities rather than item-level examples. We therefore define an item as a *contextualized creative*, identified by the combination of creative ID, ad exchange, slot width, slot height, slot visibility, and slot format. This definition distinguishes materially different delivery contexts for the same creative without introducing the severe sparsity caused by including raw domains and slot identifiers in the item identity.

Each item is associated with categorical metadata describing its exchange, domain, slot, floor price, key page, and advertiser. When a contextual attribute varies across the impressions associated with an item, we use its modal value. Raw iPinYou tracking identifiers are not treated as persistent users. Instead, a user group is defined by a unique combination of region, city, ad exchange, and normalized user-tag set. Thus, a user group represents an interpretable user group rather than an individual user.

Item-label assignment Let c_i and n_i denote the total numbers of clicks and impressions for item i, respectively. We first construct a reliable zero-click group. Items with no observed clicks are eligible only when they have at least ten impressions, preventing items with insufficient exposure from being labeled as poor-performing. The eligible zero-click items are ranked by impression count, and the most exposed items are selected to form approximately one quarter of the final item collection. These items receive label 1.

The items with at least one click are sorted by c_i and divided into three approximately equal-cardinality tiers. The tiers receive labels 2, 3, and 4 from lower to higher click popularity:

$$y_i = \begin{cases} 1, & c_i = 0 \text{ and } n_i \geq 10, \\ 2, & c_i > 0 \text{ and } i \text{ is in the lower click-count tier}, \\ 3, & c_i > 0 \text{ and } i \text{ is in the middle click-count tier}, \\ 4, & c_i > 0 \text{ and } i \text{ is in the upper click-count tier}. \end{cases} \tag{10}$$

The resulting label measures absolute click popularity rather than click-through rate. This construction produces 714 items, comprising 537 clicked items and 177 high-exposure zero-click items.

Data partitioning and use The iPinYou items are stratified by label and divided in a 6:2:1:1 ratio into four mutually disjoint segments: 427 items for training the Stage I scorer, 143 items for training the Stage II aggregator, 72 validation items, and 72 test items. Because the split is performed at the item level, neither the scorer nor the aggregation network observes validation or test items during training.

Only genuine click events are retained as positive user–item interactions. During Stage I training, unclicked candidates are sampled from the scorer-training item segment. The scorer-training manifest explicitly includes zero-click items so that they can serve as valid negative candidates without introducing items assigned to the aggregation, validation, or test segments. The four-class item labels are not used to train Stage I.

The fixed iPinYou user-group panel consists of 2,048 user groups observed in the aggregation-training data. The trained Stage I scorer is applied to the Cartesian product of this panel and the items in the aggregation-training, validation, and test segments. This yields 292,864 panel–item scores for aggregation training and 147,456 scores for each evaluation segment.

C Direct Prediction Baselines

Inputs and shared implementation Each direct baseline maps an item's observable attributes and aggregated historical user-group features to one of the four item-level labels. Categorical features undergo rare-category capping; one-hot encoded input is used where required by the model. The baselines use the same item-level training, validation, and test partitions as the B4I comparisons.

Target-item interaction feedback is unavailable at prediction time. Results are reported over three random seeds.

Models

- **Logistic Regression (LR):** A linear softmax classifier over one-hot encoded input features.
- **Decision Tree (DT):** A single CART classifier with axis-aligned splits over input features.
- **Random Forest (RF)** (Breiman, 2001): An ensemble of decision trees using sample and feature bagging.
- **MLP:** A feed-forward network combining learned categorical representations and dense features through nonlinear layers.
- **FM** (Rendle, 2010): Models pairwise feature interactions through inner products of latent feature vectors.
- **Wide&Deep** (Cheng et al., 2016): Combines a linear component with a deep network.
- **DeepFM** (Guo et al., 2017): Combines a factorization machine with a deep network for low- and high-order feature interactions.
- **FinalMLP** (Mao et al., 2023): Uses two MLP streams with feature selection and interaction aggregation.
- **DCNv2** (Wang et al., 2021): Uses an explicit feature cross network alongside a deep network.
- **AutoInt** (Song et al., 2019): Uses self-attention to learn interactions among feature fields.
- **FT-Transformer** (Gorishniy et al., 2021): Applies Transformer layers to tokenized tabular feature fields.
- **TabM** (Gorishniy et al., 2024): Uses parameter-efficient ensembling of MLP-based tabular models.

D Related Work

D.1 Blackbox Surrogate Modeling

Blackbox surrogate modeling aims to approximate the observable behavior of an inaccessible system using a separate model learned from available input–output observations. Related approaches include knowledge distillation (Hinton et al., 2015), model extraction (Bastani et al., 2017), and post-hoc surrogate explanation (Xu et al., 2023). Conventional knowledge distillation typically assumes access to predictions or representations produced directly by a teacher model, while model extraction similarly attempts to reproduce the behavior of a target model through query access.

Our setting differs from conventional teacher–student distillation. We do not assume access to the outputs, parameters, or architecture of an internal recommendation or ranking model. Instead, the observable supervision consists of historical interaction feedback generated through the platform, which may jointly reflect ranking and exposure mechanisms, user responses, and other inaccessible components of the platform process. We therefore treat the platform as a system-level black box and learn a behavioral surrogate of its observable user–item interaction patterns rather than attempting to reconstruct a particular internal model.

D.2 Interpretable Recommendation Models

Interpretable recommendation aims to expose the reasoning behind user–item predictions rather than only producing ranking scores (Zhang & Chen, 2020). Existing approaches generate explanations in different forms, including aspect-level preferences extracted from reviews (Zhang et al., 2014b; He et al., 2015), reasoning paths over knowledge graphs (Wang et al., 2019; Xian et al., 2019; 2020), and feature- or rule-level explanations. Recent neural–logical models further combine neural representation learning with logic-inspired reasoning structures. In particular, FENCR represents user–item interactions through feature-enhanced neural collaborative reasoning, producing

both prediction scores and rule-based explanatory signals (Zhang et al., 2024). These methods provide structured signals that can make interaction predictions more interpretable. In contrast, B4I uses interpretable user–item modeling as an intermediate step toward a different objective: item-level assessment. The interaction-level scores and neural–logical signals generated by the behavioral surrogate are evaluated across a common user-group panel and aggregated to characterize the expected response profile of each target item.

D.3 Aggregation and Tabular Classification

Item-level outcomes in interactive platforms can be viewed as population-level phenomena arising from heterogeneous user responses. This motivates representing an item not only through its own attributes but also through a collection of user-conditioned interaction signals.

Our aggregation stage is related to stacked prediction, set representation learning, and multiple-instance learning. In the stacking formulation (Wolpert, 1992), the user-conditioned scores generated for an item form a fixed-dimensional representation that can be processed by a downstream predictor. Alternatively, an item can be represented as an unordered set of user-conditioned responses. DeepSets (Zaheer et al., 2017) provides a general formulation for permutation-invariant learning over such sets, while attention-based multiple-instance learning allows individual instances to contribute unequally to the final bag-level representation. Recent tabular deep learning work further shows that well-tuned MLP-based models remain strong for structured data; TabM, for example, improves MLP-based tabular prediction through parameter-efficient ensembling (Gorishniy et al., 2024).

B4I differs from generic set-based or tabular prediction in the construction of these instances. The user-conditioned signals are not observed target-item interactions. Instead, they are generated by evaluating a behavioral surrogate learned from historical interactions over a fixed historical user-group panel. The aggregation stage therefore transfers historical interaction knowledge to target-item assessment without requiring realized interaction feedback for the target item.