IGD: Token Decisiveness Modeling via Information Gain in LLMs for Personalized Recommendation

Zijie Lin[1], **Yang Zhang**[1†], **Xiaoyan Zhao**[2], **Fengbin Zhu**[1†], **Fuli Feng**[3], **Tat-Seng Chua**[1]
[1]National University of Singapore [2]The Chinese University of Hong Kong
[3]University of Science and Technology of China
zijie.lin@u.nus.edu, zyang1580@gmail.com
xzhao@se.cuhk.edu.hk, zhfengbin@gmail.com
fulifeng93@gmail.com, dcscts@nus.edu.sg
[†]Corresponding author

Abstract

Large Language Models (LLMs) have shown strong potential for recommendation by framing item prediction as a token-by-token language generation task. However, existing methods treat all item tokens equally, simply pursuing likelihood maximization during both optimization and decoding. This overlooks crucial token-level differences in decisiveness—many tokens contribute little to item discrimination yet can dominate optimization or decoding. To quantify token decisiveness, we propose a novel perspective that models item generation as a decision process, measuring token decisiveness by the Information Gain (IG) each token provides in reducing uncertainty about the generated item. Our empirical analysis reveals that most tokens have low IG but often correspond to high logits, disproportionately influencing training loss and decoding, which may impair model performance. Building on these insights, we introduce an Information Gain-based Decisiveness-aware Token handling (IGD) strategy that integrates token decisiveness into both tuning and decoding. Specifically, IGD downweights low-IG tokens during tuning and rebalances decoding to emphasize tokens with high IG. In this way, IGD moves beyond pure likelihood maximization, effectively prioritizing high-decisiveness tokens. Extensive experiments on four benchmark datasets with two LLM bbones demonstrate that IGD consistently improves recommendation accuracy, achieving significant gains on widely used ranking metrics compared to strong baselines. Our codes are available at https://github.com/ZJLin2oo1/IGD.

1 Introduction

Recommendation systems [1, 2, 3] play a crucial role in helping users discover relevant and personalized content. With recent advances in Large Language Models (LLMs) [4, 5, 6], there is growing interest in leveraging LLMs' strong language understanding and reasoning capabilities [7, 8, 9] for recommendation tasks [10, 11, 12, 13, 14, 15], giving rise to a new paradigm known as LLM4Rec [16, 17, 18]. In this paradigm, recommendation is typically formulated as a natural language problem: user history and task context are encoded into a prompt, based on which the LLM is tuned to generate the top-K recommended items via autoregressive token decoding [19]. This approach has demonstrated strong performance in capturing user intent and generating personalized outputs [20, 21, 22].

Despite its promise, we argue that the existing token handling strategy in LLM4Rec does not fully align with the item generation process in recommendation. The current approach is primarily likelihood-driven, treating each item token equally and simply focusing on: 1) optimizing token

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

likelihood during fine-tuning for data fitting [23, 24, 21], and 2) maximizing token likelihood during inference for generation [19, 20]. However, not all tokens are equally important. Some tokens are more decisive in defining the item, while others serve grammatical or filler functions with low decisiveness. Low-decisiveness tokens do not reduce uncertainty in item generation, making their focus less meaningful. Moreover, low-decisiveness tokens may have high logits—such as "ghost tokens" defined in [19], which have generation probabilities close to 1 for a given prefix. These tokens can dominate likelihood-maximizing decoding, introducing bias toward items with more such tokens [19]. To improve both tuning and decoding, it is crucial to quantify and incorporate token decisiveness, rather than relying solely on likelihood optimization.

To measure token decisiveness, we introduce a novel perspective that frames token-by-token item generation in LLM4Rec as a decision process. In this framework, the uncertainty of the recommendation outcome at each generation step is quantified by the entropy [25] of the item distribution conditioned on the tokens generated so far. The decisiveness of a token is defined as the reduction in this uncertainty when selecting the token, *i.e.*, its Information Gain (IG) [26]. Based on this definition, we observe that: 1) in the studied real-world datasets, over 50% of item tokens exhibit zero IG; and 2) under the existing token strategy, LLM4Rec models may be misled by these zero-IG tokens. Specifically, as shown in Figure 5, models tend to over-optimize zero-IG tokens while under-emphasizing non-zero-IG tokens during tuning. Furthermore, as shown in Figure 4, low IG tokens, especially zero-IG tokens, often correspond to high logits, which can bias the likelihood-maximizing decoding process toward items containing more such tokens.

To incorporate token decisiveness into tuning and decoding, we propose an *Information Gain-based Decisiveness-aware Token Handing* (IGD) strategy that goes beyond simply optimizing/maximizing token likelihood. During tuning, IGD downweights zero-IG tokens to prioritize informative (non-zero-IG) tokens that aid in item discrimination, enabling more effective learning. At decoding, IGD increases the influence of high IG tokens along the autoregressive path by adjusting focus toward their logits, rather than blindly following the likelihood-maximizing principle, thereby reducing bias toward items dominated by high-logit but low-decisiveness tokens. In this way, IGD reshapes token importance by incorporating token decisiveness, leading to improved recommendation accuracy. We evaluate IGD on four benchmark datasets using two LLM backbones. Results demonstrate that IGD consistently enhances recommendation performance, with average gains of 18.89% in HR@10 and 20.15% in NDCG@10 over strong baselines.

The main contributions are: (1) We emphasize the importance of quantifying and incorporating token decisiveness in both tuning and decoding, and propose framing the item generation process as a decision process, defining token decisiveness based on information gain. (2) We introduce IGD, a simple yet effective token handling strategy that leverages token-level information gain to guide both tuning and decoding, going beyond mere likelihood optimization/maximization to prioritize high-decisiveness tokens, thereby enhancing recommendation performance. (3) We conduct extensive experiments on four benchmark datasets using two LLM backbones. The results show that IGD consistently improves recommendation performance, achieving significant gains in HR@10 and NDCG@10 over strong baselines.

2 Preliminary

This section introduces the typical tuning and decoding approaches in LLM4Rec.

2.1 Tuning

To enable next-item prediction using LLMs, supervised fine-tuning (SFT) is usually applied to teach LLMs to map items to token sequences. Given an input prompt x transformed from user history and task description, and a list of target item tokens $y = (y_1, \ldots, y_m)$, the model is trained to minimize the token-level cross-entropy loss:

$$\mathcal{L} = \sum_{t=1}^{m} \ell(f_\theta(x, y_{<t}); y_t), \tag{1}$$

where f_θ denotes the LLM with parameters θ, and ℓ is the cross-entropy between the predicted token distribution and the ground-truth token y_t at position t.

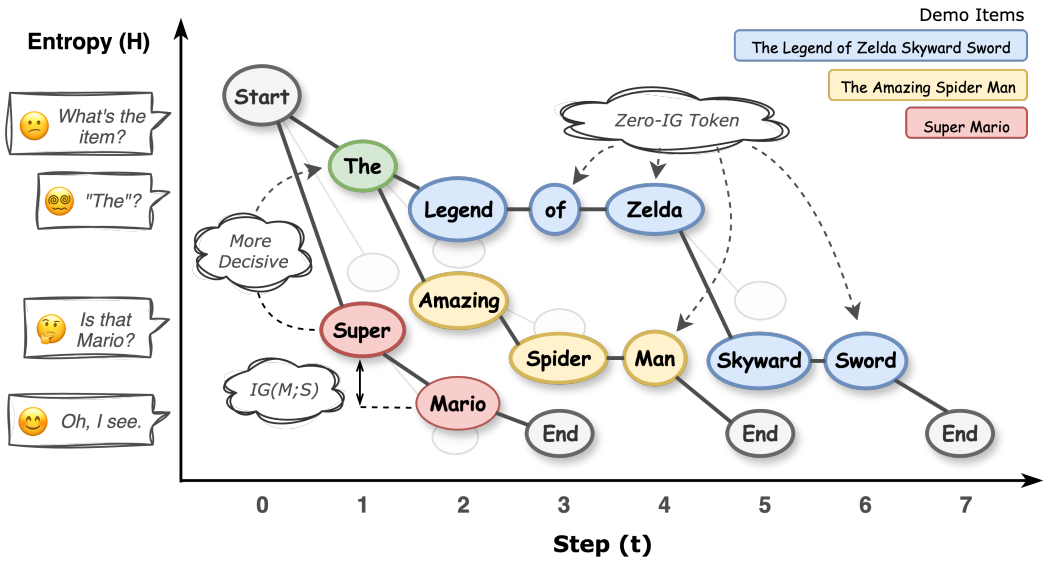

Figure 1: Illustration of LLM4Rec autoregressive token generation as a sequential decision process. As tokens are generated, the entropy of the remaining sequence gradually decreases. The information gain (IG) quantifies this reduction, e.g., IG(M; S) measures the IG of token "Mario" given prefix "Super". Tokens shared across many items (e.g., "The") exhibit lower decisiveness with lower IG, while more decisive tokens (e.g., "Super") lead to larger IG. Tokens with IG=0—such as "of", "Zelda", "Sword", and "Man"—are referred to as zero-IG tokens.

2.2 Decoding

Autoregressive Decoding. At inference time, the model generates item sequences autoregressively. The conditional probability of a full sequence y given x is factorized as:

$$p(y|x) = \prod_{t=1}^{m} p(y_t|x, y_{<t}) \tag{2}$$

This formulation enables step-by-step generation but highly relies on local token-level probabilities.

Beam Search Decoding. During inference, existing methods commonly use beam search to generate multiple item candidates simultaneously. At each decoding step t, the model expands each partial sequence $y_{\leq t-1}$ by considering the top-ranked token candidates, and updates the cumulative score using:

$$S(y_{\leq t}) = S(y_{\leq t-1}) + \log p(y_t|x, y_{<t}), \tag{3}$$

where $S(y_{\leq t})$ denotes the log-probability of the sequence prefix $y_{\leq t}$, and $p(y_t|x, y_{<t})$ is the conditional probability of the token in step t.

In standard natural language generation tasks, a length penalty is often introduced to avoid overly long outputs. However, recent work [19] reveals that applying such penalties in recommendation scenarios tends to favor longer item sequences, introducing bias into the final selection. Therefore, we follow [19] and set the length penalty to zero, keeping the score computation purely based on accumulated log-probabilities as shown above.

3 Token Decisiveness Modeling

3.1 Token Decisiveness Measurement

We model the token generation process in LLM4Rec as a sequential decision-making procedure, where each autoregressive decoding step progressively refines the target item from the full item space. Let $\mathcal{I}$ denote the full item collection, and $\mathcal{I}^{y_{\leq t}} \subseteq \mathcal{I}$ represent the set of candidate items consistent with the generated token prefix $y_{\leq t}$ at step t. Following information theory [27], we quantify the uncertainty of the recommendation at step t using Shannon entropy [25] over the candidate item distribution:

$$H(y_{\leq t}) = -\sum_{\mathcal{I}_i \in \mathcal{I}^{y_{\leq t}}} p_r(\mathcal{I}_i) \log p_r(\mathcal{I}_i), \tag{4}$$

where $p_r(\mathcal{I}_i)$ denotes the empirical prior probability of item $\mathcal{I}_i$ estimated from the training data. The entropy is computed over the candidate set $\mathcal{I}^{y_{\leq t}}$ compatible with the current token prefix.

At each step, a new token y_t reduces the candidate space. We measure the decisiveness of y_t using its **Information Gain (IG)** —the reduction in uncertainty it induces:

$$\text{IG}(y_t; y_{<t}) = H(y_{\leq t-1}) - H(y_{\leq t}) \tag{5}$$

This formulation quantifies how much y_t contributes to identifying the target item, where a higher IG indicates greater decisiveness.

3.2 Statistical Analysis on Token Decisiveness

Table 1: Dataset Statistics and zero-IG Token Proportion

Dataset	Items	Train	Valid	Test	Tokens	zero-IG Tokens (%)
CDs	14,239	148,685	18,586	18,587	805,786	450,960 (55.96%)
Games	11,037	201,613	25,202	25,203	2,128,430	1,292,171 (60.71%)
Toys	11,252	112,755	14,095	14,096	1,530,370	1,098,070 (71.75%)
Books	41,722	682,998	85,376	85,376	7,183,839	5,241,997 (72.97%)

We conduct a statistical analysis of the proposed IG metrics on four public datasets, summarized in Table 1. First, each item title is tokenized into a sequence of tokens using the Qwen2.5 tokenizer [28]. Then, we construct a token prefix tree and compute, for each prefix, its corresponding entropy as well as the IG for each token. All statistics are computed exclusively on the training set. From Table 1, we observe that zero-IG tokens—tokens that yield no reduction in entropy are predominant, constituting 55.96% to 72.97% of all tokens across datasets.

3.3 Token-level Biases

To investigate how LLMs interact with token decisiveness, we analyze tuning dynamics and decoding paths using the D3 [19] method with the Qwen2.5-1.5B model [28]. In model tuning, we respectively monitor the tuning loss for zero-IG tokens and non-zero-IG tokens across the training set; In model decoding, we gather both ground-truth items and top-10 predicted items from the test set, compute the average entropy of the prefix at each decoding step, and compare the average entropies between predicted and ground-truth prefixes.

We observe the following biases:

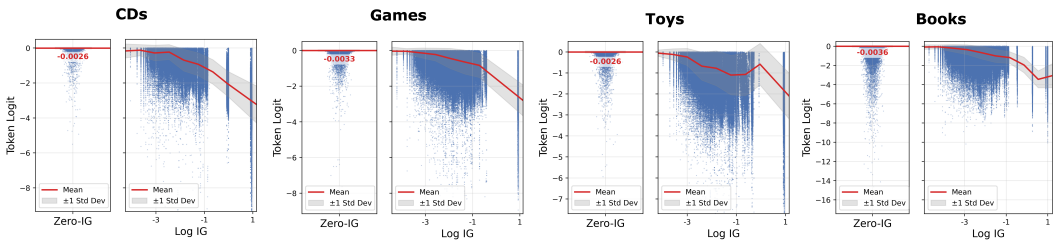

Figure 4: Relationship between IG values and logits of tokens in decoding. For each dataset, the left subfigure shows that zero-IG tokens are associated with extremely high logits (close to 0). The right subfigure illustrates a negative correlation between IG values and logit magnitudes for non-zero-IG tokens.

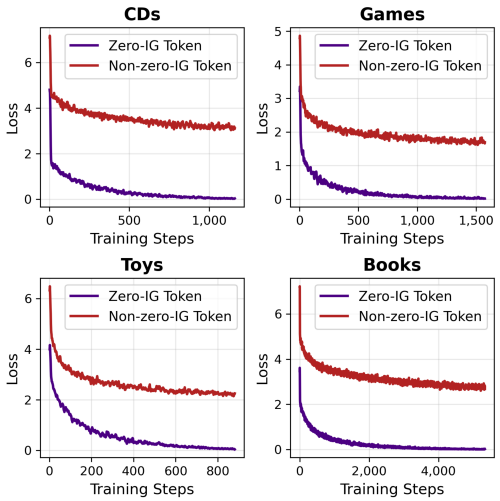

Figure 2: Loss comparison between zero-IG and non-zero-IG tokens in model tuning (epoch 1)

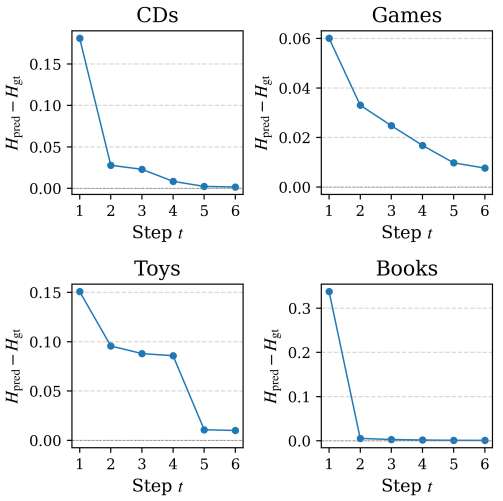

Figure 3: Entropy difference in decoding: model prediction vs. ground-truth

Tuning Bias: During model tuning, models tend to over-optimize zero-IG tokens while under-emphasizing non-zero-IG tokens. As shown in Figure 2, the model rapidly minimizes the loss on zero-IG tokens, while the loss on non-zero-IG tokens remains relatively high. This imbalance leads to a biased learned distribution, causing the LLM to favor less decisive tokens. Therefore, a more effective training approach that emphasizes decisive tokens is necessary.

Decoding Bias: As shown in Figure 4, LLMs tend to assign higher logits to low IG tokens, particularly those with zero IG. Since beam search selects top-k candidates solely based on token-level logit scores, these low IG tokens are favored during decoding. As a result, the generated item prefixes exhibit higher average entropy than the ground-truth prefixes (Figure 3), indicating a shift toward less informative predictions. This reveals a decoding bias, where the model systematically prefers less decisive tokens due to the likelihood-based decoding objective.

4 Information Gain-based Decisiveness-aware Token Handing (IGD)

To mitigate the token-level bias during both tuning and decoding phases, we propose a two-stage method, **I**nformation **G**ain-based **D**ecisiveness-aware Token Handling (**IGD**). Specifically, we leverage IG to quantify token decisiveness and adjust training dynamics and inference process accordingly.

IGD-Tuning. To mitigate learning bias towards non-decisive tokens, we introduce a token-level reweighting scheme into the loss function when tuning LLMs. The revised objective is:

$$\mathcal{L}_{\text{IGD}} = \frac{1}{\Omega} \sum_{t=1}^{|y|} w_t \cdot \ell(f_\theta(x, y_{<t}), y_t), \tag{6}$$

where w_t is a weight assigned to token y_t, Ω is the sum of w_t across all predicted tokens. w_t is defined as:

$$w_t = \begin{cases} \beta, & \text{if IG}(y_t; y_{<t}) = 0 \\ 1, & \text{if IG}(y_t; y_{<t}) > 0 \end{cases} \tag{7}$$

Here, $\beta \in [0, 1]$ is a hyperparameter that controls the penalty on non-decisive tokens, thereby reducing the learning focus on tokens with zero IG. Setting $\beta = 1$ recovers the standard cross-entropy loss.

IGD-Decoding. To address decoding bias that favors generic or homogeneous tokens, we modify the standard beam search scoring function to promote decisive tokens. The revised score at step t is computed as:

$$S(y_{\leq t}) = S(y_{\leq t-1}) + w_d \cdot \log p(y_t|x, y_{<t}), \tag{8}$$

with the reweighting factor w_d computed as:

$$w_d = 1 - \alpha \cdot \widetilde{\text{IG}}(y_t) \tag{9}$$

Here, $\widetilde{\text{IG}}(y_t)$ denotes the max-min normalized IG of token y_t within the current beam step, scaled to $[0, 1]$ across all candidates. If all candidates have zero-IG, their normalized values are set to zero. The hyperparameter $\alpha \in [0, 1]$ controls the strength of decisiveness calibration. When $\alpha = 1$, the method falls back to standard beam search scoring.

5 Experiments

In this section, we aim to address the following research questions (RQs): **RQ1:** Does IGD improve the recommendation accuracy of LLM4Rec? **RQ2:** How does each stage of IGD contribute to performance improvements? **RQ3:** How does IGD influence the tuning and decoding to enhance performance? **RQ4:** Is IGD effective across LLMs of different scales and tokenization schemes?

5.1 Experimental Setup

Datasets. We evaluate IGD on four publicly available Amazon review datasets [29]: *CDs*, *Games*, *Toys*, and *Books*, covering data from May 1996 to October 2018. Dataset statistics are summarized in Table 1. Following the preprocessing procedure in the D3 paper [19], we truncate the data based on timestamps and filter out infrequent users and items, ensuring that each user and each item has at least 5 interactions.

Compared Methods. For standard recommendation settings, we compare our method against: (1) two representative sequential recommendation models—**GRU4Rec**[30], **SASRec**[31] and **LRURec**[32]; and (2) two state-of-the-art (SOTA) LLM-based recommendation approaches—**BIGRec**[20] and **D3**[19]. Our IGD strategy can be integrated into both BIGRec and D3 for fair comparison. To further evaluate the effectiveness of IGD as a token handling strategy, we compare it with two token handing baselines that can also be seamlessly integrated into LLM4Rec frameworks: 1) **Position Normalization (Pos)**, which assigns higher weight to earlier item tokens to mitigate position bias; and 2) **Causal Fine-tuning (CFT)** [33], which builds upon Pos by introducing an additional causal loss term to enhance the modeling of causal effects at the token level. See Appendix A for detailed descriptions of all baselines. By default, we implement all LLM-based methods using Qwen2.5-1.5B [28]. More implementation details, including hyperparameter tuning settings, can be found in Appendix B.

Evaluation Metrics. To evaluate the model's top-K recommendation performance, we adopt two widely-used metrics: Hit Ratio (HR@K) and Normalized Discounted Cumulative Gain (NDCG@K) [34]. Both metrics are computed under the all-ranking protocol [35], where the model ranks all candidate items for each user. In our experiments, we report results for $K = 5$ and $K = 10$.

Table 2: Recommendation performance of the compared methods evaluated on four benchmark datasets. H@K and N@K denote HR@K and NDCG@K, respectively. *Improvement* indicates the relative performance gain of IGD over the corresponding LLM4Rec backbone without any token reweighting. The best results are bold.

Methods	**CDs**				**Games**			
	N@5	N@10	H@5	H@10	N@5	N@10	H@5	H@10
GRU4Rec	0.0248	0.0288	0.0342	0.0467	0.0169	0.0221	0.0261	0.0423
SASRec	0.0477	0.0535	0.0647	0.0824	0.0237	0.0290	0.0338	0.0502
LRURec	0.0540	0.0586	0.0680	0.0824	0.0298	0.0363	0.0421	0.0621
BIGRec	0.0502	0.0553	0.0623	0.0782	0.0317	0.0381	0.0430	0.0631
+Pos	0.0511	0.0566	0.0632	0.0802	0.0319	0.0396	0.0423	0.0665
+CFT	0.0509	0.0566	0.0631	0.0810	0.0349	0.0414	0.0482	0.0686
+**IGD**	**0.0540**	**0.0593**	**0.0669**	**0.0833**	**0.0423**	**0.0507**	**0.0576**	**0.0833**
Improvement	*+7.78%*	*+7.82%*	*+9.33%*	*+9.04%*	*+33.4%*	*+33.1%*	*+34.0%*	*+32.0%*
D3	0.0716	0.0767	0.0882	0.1040	0.0415	0.0477	0.0581	0.0773
+Pos	0.0729	0.0779	0.0902	0.1053	0.0429	0.0489	0.0581	0.0767
+CFT	0.0736	0.0786	0.0917	0.1069	0.0437	0.0499	0.0613	0.0806
+**IGD**	**0.0748**	**0.0801**	**0.0929**	**0.1092**	**0.0518**	**0.0598**	**0.0705**	**0.0946**
Improvement	*+4.47%*	*+4.43%*	*+5.33%*	*+5.00%*	*+25.6%*	*+29.2%*	*+26.7%*	*+22.7%*

Methods	**Toys**				**Books**			
	N@5	N@10	H@5	H@10	N@5	N@10	H@5	H@10
GRU4Rec	0.0200	0.0238	0.0275	0.0392	0.0060	0.0078	0.0094	0.0149
SASRec	0.0356	0.0398	0.0473	0.0745	0.0097	0.0123	0.0146	0.0226
LRURec	0.0358	0.0404	0.0463	0.0608	0.0257	0.0277	0.0319	0.0383
BIGRec	0.0553	0.0623	0.0736	0.0951	0.0190	0.0211	0.0245	0.0309
+Pos	0.0561	0.0631	0.0741	0.0958	0.0197	0.0218	0.0255	0.0319
+CFT	0.0561	0.0630	0.0746	0.0961	0.0195	0.0218	0.0250	0.0321
+**IGD**	**0.0577**	**0.0656**	**0.0771**	**0.1014**	**0.0267**	**0.0294**	**0.0334**	**0.0419**
Improvement	*+4.34%*	*+5.30%*	*+4.76%*	*+6.62%*	*+41.3%*	*+40.0%*	*+36.9%*	*+36.0%*
D3	0.0634	0.0698	0.0833	0.1029	0.0212	0.0228	0.0266	0.0315
+Pos	0.0644	0.0702	0.0850	0.1029	0.0221	0.0237	0.0275	0.0324
+CFT	0.0640	0.0704	0.0840	0.1036	0.0219	0.0236	0.0275	0.0327
+**IGD**	**0.0658**	**0.0726**	**0.0868**	**0.1082**	**0.0291**	**0.0313**	**0.0356**	**0.0424**
Improvement	*+3.79%*	*+4.01%*	*+4.20%*	*+5.15%*	*+37.3%*	*+37.3%*	*+33.8%*	*+34.6%*

5.2 Main Results (RQ1)

In this section, we evaluate whether the proposed IGD method improves overall recommendation performance. Table 2 summarizes the performance of all compared methods. For token handling strategies (IGD, CFT, and Pos), we implement each on top of both BIGRec and D3 for comparison. For traditional baselines such as GRU4Rec and SASRec, we adopt the reported results from the D3 paper [19] to ensure consistency. For LRURec, we utilize the official PyTorch implementation [32] and evaluate it using our dataset split. Our key observations are as follows:

- IGD achieves notable improvements on both LLM4Rec methods (BIGRec and D3). Specifically, it yields an average improvement of **20.9%** in HR@10 and **21.5%** in NDCG@10 on BIGRec, and **18.9%** in HR@10 and **20.1%** in NDCG@10 on D3.
- Compared with other token-handling methods (Pos and CFT), IGD consistently delivers better performance across all evaluation metrics on both the LLM4Rec backbone, showing its effectiveness and generalizability. The superiority of the proposed IGD method can be attributed to its specific consideration of token decisiveness.
- LLM4Rec methods (BIGRec, D3) clearly outperform traditional recommendation models (GRU4Rec, SASRec). Although LRURec outperforms them on the CDs and Books datasets, BIGRec and D3 regain the lead when combined with token reweighting techniques, highlighting the advantages of leveraging LLMs in recommendation tasks.

Table 3: Ablation results of IGD on D3. Here, "w/o" w_d, "w/o w_t", and "w/o Both" denote the removal of decoding reweighting, tuning reweighting, and both components, respectively.

Methods	CDs				Games			
	N@5	N@10	H@5	H@10	N@5	N@10	H@5	H@10
IGD	0.0748	0.0801	0.0929	0.1092	0.0518	0.0598	0.0705	0.0946
w/o w_d	0.0751	0.0800	0.0926	0.1077	0.0514	0.0594	0.0695	0.0942
w/o w_t	0.0718	0.0768	0.0887	0.1041	0.0414	0.0484	0.0575	0.0790
w/o Both	0.0716	0.0767	0.0882	0.1040	0.0415	0.0477	0.0581	0.0773
Methods	Toys				Books			
	N@5	N@10	H@5	H@10	N@5	N@10	H@5	H@10
IGD	0.0658	0.0726	0.0868	0.1082	0.0291	0.0313	0.0356	0.0424
w/o w_d	0.0653	0.0719	0.0861	0.1063	0.0290	0.0312	0.0355	0.0422
w/o w_t	0.0640	0.0711	0.0843	0.1060	0.0212	0.0229	0.0268	0.0318
w/o Both	0.0634	0.0698	0.0833	0.1029	0.0212	0.0228	0.0266	0.0315

5.3 Ablation Study (RQ2)

In this section, we analyze the individual contributions of the two stages of IGD reweighting—Tuning and Decoding—to the overall improvement in recommendation accuracy. The results of the ablation study are presented in Table 3. We observe the following: (1) Both the tuning-stage and the decoding-stage reweighting contribute positively to performance gains; (2) The tuning stage brings a more substantial improvement than the decoding stage. Notably, even when using only the tuning-stage reweighting (w/o w_d), IGD still outperforms the CFT and Pos baselines.

5.4 In-depth Analysis of IGD's Effect (RQ3)

In this subsection, we investigate how IGD influences the underlying mechanisms of tuning and decoding, and how these effects contribute to the improvement of recommendation performance. We follow a similar experimental strategy as described in Section 3.3 to compare tuning loss and prefix entropy.

1. **IGD-Tuning**: As shown in Fig. 5, after incorporating the weight term w_t, the training loss of zero-IG tokens decreases more slowly, whereas the loss for non-zero-IG tokens decreases more rapidly. This indicates that IGD-Tuning encourages learning on non-zero-IG tokens while avoiding overfitting zero-IG ones.

2. **IGD-Decoding**: As illustrated in Figure 6, the entropy gap between the predicted and ground truth prefixes is reduced under IGD-Decoding, indicating a more decisive and better-aligned decoding process. Note that increasing α (see Equation (9)) to mitigate decoding bias does not always improve recommendation accuracy, suggesting a trade-off between maximizing the likelihood of the token sequence and prioritizing high-decisiveness tokens.

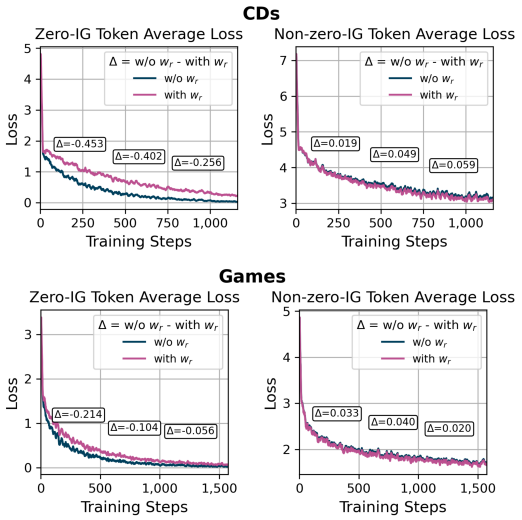

Figure 5: Loss comparison on CDs and Games datasets: IGD-Tuning effect on zero-IG and non-zero-IG tokens (epoch 1). The results on the other two datasets are in Appendix C.

5.5 Generalizability (RQ4)

To demonstrate the generalizability of our proposed IGD method, we evaluate it under different tokenization strategies and model scales. Specifically, we conduct experiments using the LLaMA3-8B model [36], with the prefix tree constructed using the LLaMA3 tokenizer [36]. To enable efficient

Table 4: Performance comparison of D3+IGD vs. original D3 on LLaMA3-8B backbone on CDs and Games. The results on the other two datasets are in Appendix E.

Methods	CDs				Games			
	N@5	N@10	H@5	H@10	N@5	N@10	H@5	H@10
D3	0.0742	0.0790	0.0917	0.1062	0.0456	0.0528	0.0611	0.0832
+IGD	**0.0791**	**0.0850**	**0.0994**	**0.1179**	**0.0645**	**0.0734**	**0.0863**	**0.1137**
Improvement	*+6.60%*	*+7.59%*	*+8.40%*	*+11.0%*	*41.5%*	*+39.0%*	*+41.2%*	*+36.7%*

fine-tuning, we employ 4-bit QLoRA [37] for both training and inference, with the rank parameter r set to 32, the scaling factor α set to 64, and the dropout rate set to 0.05. As shown in Table 4, IGD yields consistent improvements across all four datasets when applied to D3, with an average gain of 26.8% in HR@10 and 26.4% in NDCG@10. In addition to the Amazon dataset, we also evaluate IGD on the Steam dataset (see Appendix E), where it demonstrates consistent improvements. These results confirm the generalizability of our IGD approach across different model architectures and datasets.

6 Related Work

LLM-based Recommendation: Based on how LLMs are utilized, existing LLM4Rec methods can be broadly categorized into two groups: (1) using LLMs to augment traditional recommendation models [38, 39, 40], and (2) employing LLMs directly as recommendation systems [20, 19, 41]. Our work aligns with the second category. Early approaches in this direction typically followed a discriminative paradigm [24], which has since shifted toward a generative paradigm [20]. Building on this, recent studies have explored enhancing collaborative modeling [42, 43, 44], optimizing tokenization schemes [22], and improving inference efficiency [45], *etc*. However, few works have investigated recommendations at the token level. Our work specifically focuses on this.

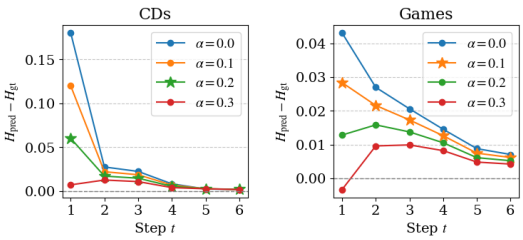

Figure 6: Entropy difference: prediction vs. ground truth after IGD-decoding. "Start" indicates optimal α selected based on HR@10. The results on the other two datasets are in Appendix D.

Token-level Biases in LLM4Rec: Autoregressive LLMs generate items token-by-token, misaligning with item-level recommendation objectives [46, 19, 47, 48]. Multi-token items suffer from token-level biases: high-probability *ghost tokens* dominate decoding without aiding discrimination, inflating scores for longer items [19]; and *common tokens* share across many items (e.g., 'The') dilute the impact of rare and informative tokens, reducing diversity [46, 19].

Token-level Bias Mitigation Strategies: To reduce amplification bias caused by *ghost tokens*, D3 removes length normalization in decoding [19], but *ghost tokens* still skew beam search. Position normalization reweights tokens by position, assuming early tokens are more uncertain [33], yet *ghost tokens* appear throughout sequences and are not confined to later positions. To prevent *common token* domination, some methods leverage traditional model guidance [47, 19], but this limits flexibility and scalability. Different from them, we propose a token-handling strategy by considering token decisiveness, which helps debiasing.

Token Prefix Trie for LLM4Rec: Recent works leverage token prefix trie for LLM-based recommendation, each employing the trie in a different way to guide token generation and learning. MSL [48] uses the trie as a constraint: a masked softmax prunes infeasible next tokens, reducing negative optimization signals and focusing training on valid continuations. Flower [47] uses the trie for process-guided supervision: rewards are stored at nodes and propagated along paths during tuning, encouraging trajectories that obtain higher rewards. Ours uses the trie for decisiveness-aware weighting: each node stores an IG score estimated from data, and these scores are applied to debias both tuning and decoding, highlighting discriminative tokens while down-weighting ambiguous ones.

7 Conclusion & Limitation

In this work, we introduced a decision-process perspective for token-by-token generation in LLM4Rec, quantifying token decisiveness using IG. We identified tuning and decoding biases where current models misallocate focus between decisive and non-decisive tokens. Our proposed IGD method addressed these issues through token reweighting during both training and inference. Experiments across four datasets, two LLM4Rec backbones, and two LLM architectures demonstrated IGD's effectiveness, achieving significant improvements in recommendation accuracy.

Our current experiments focus exclusively on IG-based token scoring. However, the IG values of tokens are highly skewed (see Figure 4), which makes it challenging to design practical reweighting methods (as discussed in Appendix G). Future work may explore alternative decision metrics, such as Gini impurity, Gain Ratio, or Chi-squared statistics [49, 50, 51], which may provide complementary perspectives on token decisiveness.

In addition, our present analysis is restricted to text tokens. Extending decisiveness modeling to semantic ID tokens [34, 52, 53] and multimodal tokens [54, 55, 56] is a promising direction for future research.

8 Acknowledgement

This research/project is supported by the National Research Foundation, Singapore under its National Large Language Models Funding Initiative (AISG Award No: AISG-NMLP-2024-002) and A*STAR under its Japan-Singapore Joint Call: Japan Science and Technology Agency (JST) and Agency for Science, Technology and Research (A*STAR) 2024 (Award R24I6IR142). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the A*STAR.

References

[1] Y. Zhang, F. Feng, X. He, T. Wei, C. Song, G. Ling, and Y. Zhang, "Causal intervention for leveraging popularity bias in recommendation," in *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, 2021, pp. 11–20.

[2] C. Gao, S. Wang, S. Li, J. Chen, X. He, W. Lei, B. Li, Y. Zhang, and P. Jiang, "Cirs: Bursting filter bubbles by counterfactual interactive recommender system," *ACM Transactions on Information Systems (TOIS)*, vol. 42, no. 1, aug 2023. [Online]. Available: https://doi.org/10.1145/3594871

[3] Y. Zhang, Z. Hu, Y. Bai, J. Wu, Q. Wang, and F. Feng, "Recommendation unlearning via influence function," *ACM Transactions on Recommender Systems*, vol. 3, no. 2, pp. 1–23, 2024.

[4] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[5] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi *et al.*, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *arXiv preprint arXiv:2501.12948*, 2025.

[6] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[7] X. Zhao, M. Yan, Y. Zhang, Y. Deng, J. Wang, F. Zhu, Y. Qiu, H. Cheng, and T.-S. Chua, "Reinforced strategy optimization for conversational recommender systems via network-of-experts," *arXiv e-prints*, pp. arXiv–2509, 2025.

[8] X. Zhao, L. Wang, Z. Wang, H. Cheng, R. Zhang, and K.-F. Wong, "Pacar: Automated fact-checking with planning and customized action reasoning using large language models," in *International Conference on Language Resources and Evaluation*, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:269804391

[9] Y. Qiu, T. Shi, X. Zhao, F. Zhu, Y. Zhang, and F. Feng, "Latent inter-user difference modeling for llm personalization," *arXiv preprint arXiv:2507.20849*, 2025.

[10] J. Harte, W. Zorgdrager, P. Louridas, A. Katsifodimos, D. Jannach, and M. Fragkoulis, "Leveraging large language models for sequential recommendation," in *Proceedings of the 17th ACM Conference on Recommender Systems*, 2023, pp. 1096–1102.

[11] J. Lin, R. Shan, C. Zhu, K. Du, B. Chen, S. Quan, R. Tang, Y. Yu, and W. Zhang, "Rella: Retrieval-enhanced large language models for lifelong sequential behavior comprehension in recommendation," in *Proceedings of the ACM Web Conference 2024*, 2024, pp. 3497–3508.

[12] Y. Zhang, W. Xu, X. Zhao, W. Wang, F. Feng, X. He, and T.-S. Chua, "Reinforced latent reasoning for llm-based recommendation," *arXiv preprint arXiv:2505.19092*, 2025.

[13] X. Zhao, Y. Deng, W. Wang, H. lin, H. Cheng, R. Zhang, S.-K. Ng, and T.-S. Chua, "Exploring the impact of personality traits on conversational recommender systems: A simulation with large language models," *ArXiv*, vol. abs/2504.12313, 2025. [Online]. Available: https://api.semanticscholar.org/CorpusID:277857404

[14] L. Li, Y. Zhang, and L. Chen, "Prompt distillation for efficient llm-based recommendation," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 1348–1357.

[15] W. Liu, Z. Du, H. Zhao, W. Zhang, X. Zhao, G. Wang, Z. Dong, and J. Xu, "Inference computation scaling for feature augmentation in recommendation systems," *ArXiv*, vol. abs/2502.16040, 2025. [Online]. Available: https://api.semanticscholar.org/CorpusID:276575263

[16] C. Liu, Y. Bai, X. Zhao, Y. Zhang, F. Feng, and W. Rong, "Discrec: Disentangled semantic-collaborative modeling for generative recommendation," *arXiv preprint arXiv:2506.15576*, 2025.

[17] J. Harte, W. Zorgdrager, P. Louridas, A. Katsifodimos, D. Jannach, and M. Fragkoulis, "Leveraging large language models for sequential recommendation," p. 1096–1102, 2023. [Online]. Available: https://doi.org/10.1145/3604915.3610639

[18] K. Bao, J. Zhang, Y. Zhang, W. Wenjie, F. Feng, and X. He, "Large language models for recommendation: Progresses and future directions," ser. SIGIR-AP '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 306–309. [Online]. Available: https://doi.org/10.1145/3624918.3629550

[19] K. Bao, J. Zhang, Y. Zhang, X. Huo, C. Chen, and F. Feng, "Decoding matters: Addressing amplification bias and homogeneity issue for llm-based recommendation," *arXiv preprint arXiv:2406.14900*, 2024.

[20] K. Bao, J. Zhang, W. Wang, Y. Zhang, Z. Yang, Y. Luo, C. Chen, F. Feng, and Q. Tian, "A bi-step grounding paradigm for large language models in recommendation systems," *ACM Transactions on Recommender Systems*, 2023.

[21] J. Zhang, R. Xie, Y. Hou, X. Zhao, L. Lin, and J.-R. Wen, "Recommendation as instruction following: A large language model empowered recommendation approach," *ACM Transactions on Information Systems*, 2023.

[22] B. Zheng, Y. Hou, H. Lu, Y. Chen, W. X. Zhao, M. Chen, and J.-R. Wen, "Adapting large language models by integrating collaborative semantics for recommendation," in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 2024, pp. 1435–1448.

[23] J. Lin, X. Dai, Y. Xi, W. Liu, B. Chen, H. Zhang, Y. Liu, C. Wu, X. Li, C. Zhu *et al.*, "How can recommender systems benefit from large language models: A survey," *ACM Transactions on Information Systems*, vol. 43, no. 2, pp. 1–47, 2025.

[24] K. Bao, J. Zhang, Y. Zhang, W. Wang, F. Feng, and X. He, "Tallrec: An effective and efficient tuning framework to align large language model with recommendation," in *Proceedings of the 17th ACM Conference on Recommender Systems*, 2023, pp. 1007–1014.

[25] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[26] R. Baeza-Yates, B. Ribeiro-Neto *et al.*, *Modern information retrieval*. ACM press New York, 1999, vol. 463, no. 1999.

[27] T. M. Cover, *Elements of information theory*. John Wiley & Sons, 1999.

[28] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei *et al.*, "Qwen2. 5 technical report," *arXiv preprint arXiv:2412.15115*, 2024.

[29] J. Ni, J. Li, and J. McAuley, "Justifying recommendations using distantly-labeled reviews and fine-grained aspects," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 188–197. [Online]. Available: https://aclanthology.org/D19-1018/

[30] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," *arXiv preprint arXiv:1511.06939*, 2015.

[31] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *2018 IEEE International Conference on Data Mining (ICDM)*, 2018, pp. 197–206.

[32] Z. Yue, Y. Wang, Z. He, H. Zeng, J. Mcauley, and D. Wang, "Linear recurrent units for sequential recommendation," in *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, ser. WSDM '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 930–938. [Online]. Available: https://doi.org/10.1145/3616855.3635760

[33] Y. Zhang, J. You, Y. Bai, J. Zhang, K. Bao, W. Wang, and T.-S. Chua, "Causality-enhanced behavior sequence modeling in llms for personalized recommendation," *arXiv preprint arXiv:2410.22809*, 2024.

[34] S. Rajput, N. Mehta, A. Singh, R. Keshavan, T. Vu, L. Heidt, L. Hong, Y. Tay, V. Q. Tran, J. Samost, M. Kula, E. H. Chi, and M. Sathiamoorthy, "Recommender systems with generative retrieval," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, ser. NIPS '23. Red Hook, NY, USA: Curran Associates Inc., 2023.

[35] W. Krichene and S. Rendle, "On sampled metrics for item recommendation," ser. KDD '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1748–1757. [Online]. Available: https://doi.org/10.1145/3394486.3403226

[36] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[37] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," *Advances in neural information processing systems*, vol. 36, pp. 10 088–10 115, 2023.

[38] Y. Xi, W. Liu, J. Lin, X. Cai, H. Zhu, J. Zhu, B. Chen, R. Tang, W. Zhang, and Y. Yu, "Towards open-world recommendation with knowledge augmentation from large language models," in *Proceedings of the 18th ACM Conference on Recommender Systems*, 2024, pp. 12–22.

[39] X. Ren, W. Wei, L. Xia, L. Su, S. Cheng, J. Wang, D. Yin, and C. Huang, "Representation learning with large language models for recommendation," in *Proceedings of the ACM Web Conference 2024*, 2024, pp. 3464–3475.

[40] L. Sheng, A. Zhang, Y. Zhang, Y. Chen, X. Wang, and T.-S. Chua, "Language models encode collaborative signals in recommendation," 2024.

[41] J. Zhang, R. Xie, Y. Hou, X. Zhao, L. Lin, and J.-R. Wen, "Recommendation as instruction following: A large language model empowered recommendation approach," *ACM Transactions on Information Systems*, 2023.

[42] Y. Liu, J. Zhang, Y. Dang, Y. Liang, Q. Liu, G. Guo, J. Zhao, and X. Wang, "Cora: Collaborative information perception by large language model's weights for recommendation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 12, 2025, pp. 12 246–12 254.

[43] Y. Zhang, F. Feng, J. Zhang, K. Bao, Q. Wang, and X. He, "Collm: Integrating collaborative embeddings into large language models for recommendation," *IEEE Transactions on Knowledge and Data Engineering*, 2025.

[44] Y. Zhang, K. Bao, M. Yang, W. Wang, F. Feng, and X. He, "Text-like encoding of collaborative information in large language models for recommendation," in *Annual Meeting of the Association for Computational Linguistics*, 2024.

[45] X. Lin, C. Yang, W. Wang, Y. Li, C. Du, F. Feng, S.-K. Ng, and T.-S. Chua, "Efficient inference for large language model-based generative recommendation," *arXiv preprint arXiv:2410.05165*, 2024.

[46] C. Gao, R. Chen, S. Yuan, K. Huang, Y. Yu, and X. He, "Sprec: Leveraging self-play to debias preference alignment for large language model-based recommendations," *arXiv preprint arXiv:2412.09243*, 2024.

[47] C. Gao, M. Gao, C. Fan, S. Yuan, W. Shi, and X. He, "Process-supervised llm recommenders via flow-guided tuning," *arXiv preprint arXiv:2503.07377*, 2025.

[48] B. Wang, F. Liu, J. Chen, X. Lou, C. Zhang, J. Wang, Y. Sun, Y. Feng, C. Chen, and C. Wang, "Msl: Not all tokens are what you need for tuning llm as a recommender," in *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '25. New York, NY, USA: Association for Computing Machinery, 2025, p. 1912–1922. [Online]. Available: https://doi.org/10.1145/3726302.3730041

[49] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[50] W.-Y. Loh, "Classification and regression trees," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 14–23, 2011.

[51] D. J. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining*. MIT press, 2001.

[52] A. Singh, T. Vu, N. Mehta, R. Keshavan, M. Sathiamoorthy, Y. Zheng, L. Hong, L. Heldt, L. Wei, D. Tandon *et al.*, "Better generalization with semantic ids: A case study in ranking for recommendations," in *Proceedings of the 18th ACM Conference on Recommender Systems*, 2024, pp. 1039–1044.

[53] J. Deng, S. Wang, K. Cai, L. Ren, Q. Hu, W. Ding, Q. Luo, and G. Zhou, "Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment," *arXiv preprint arXiv:2502.18965*, 2025.

[54] X. Long, J. Zeng, F. Meng, Z. Ma, K. Zhang, B. Zhou, and J. Zhou, "Generative multi-modal knowledge retrieval with large language models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 17, 2024, pp. 18 733–18 741.

[55] H. Liu, Y. Wei, X. Song, W. Guan, Y.-F. Li, and L. Nie, "Mmgrec: Multimodal generative recommendation with transformer model," *arXiv preprint arXiv:2404.16555*, 2024.

[56] M. Lao, Z. Li, Y. Guo, X. Zhang, S. Cai, Z. Ding, and H. Li, "Boosting discriminability for robust multimodal entity linking with visual modality missing," in *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2025, pp. 989–999.

[57] X. Zhao, J. You, Y. Zhang, W. Wang, H. Cheng, F. Feng, S.-K. Ng, and T.-S. Chua, "Nextquill: Causal preference modeling for enhancing llm personalization," *arXiv preprint arXiv:2506.02368*, 2025.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The statement is supported by the experimental results and clearly explained in the methodology section, ensuring the claim is well-founded..

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of our work are explicitly discussed in section 7, where we acknowledge potential constraints and directions for future improvements.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: See section 3. Our paper rigorously formulates token decisiveness using information theory, clearly stating all underlying assumptions.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: See Section 5, we provide detailed descriptions of the experimental setup, datasets, model configurations. All hyperparameters and evaluation metrics are clearly reported.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: A link to the GitHub repository containing all code and data, along with detailed instructions for reproducing the experiments, is provided at the end of the abstract.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Please see Section 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   As discussed in Section 7, our method yields a substantial improvement (about 20%) compared to baselines, which strongly supports the significance of the results. Due to the high computational cost of running multiple randomized trials, we report results based on a fixed random seed without error bars.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Basic information on the computational setup is included in the README of the code repository. We plan to further elaborate on the compute details in the camera-ready version.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research complies with the NeurIPS Code of Ethics. It does not involve human subjects, sensitive personal data, or potential dual-use concerns.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See 1, the paper highlights the potential positive societal impact of advanced recommendation systems in helping internet user find relevant and personalized information.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not release any new data or models with high risk for misuse. We use only existing, publicly available datasets and models, and follow their original usage guidelines and licenses.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly cite the code and datasets used in our experiments, including [19] and [29], and ensure their usage complies with the respective licenses and terms of use.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release any new datasets, models, or code assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This work does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This work does not involve research with human subjects or crowdsourcing experiments, and thus no IRB approval was required.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Our method LLMs as a core component for recommendation system.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A The Details of Compared Methods

To highlight the modeling strength of LLM4Rec models and provide context for our token-level enhancement method IGD, we first compare them with traditional sequential recommenders.

Traditional Recommendation Methods:

- **GRU4Rec** [30]: A widely-used sequential recommendation method that employs Gated Recurrent Units (GRU) to capture sequential patterns and model user preferences.
- **SASRec** [31]: A representative sequential recommendation method that utilizes a self-attention mechanism for preference modeling, offering powerful representation capabilities for sequential data.
- **LRURec** [32]: A linear recurrent unit-based sequential recommender that enables fast, incremental inference with reduced model size and parallelizable training, achieving strong accuracy and efficiency compared to attention-based baselines.

LLM4Rec Methods:

- **BIGRec** [20]: A representative LLM4Rec system that fine-tunes large language models to generate the next item based on the user's historical behavior. We adopt the constrained beam search decoding paradigm as described in D3 paper [19].
- **D3** [19]: A state-of-the-art LLM4Rec approach that fine-tunes the model similarly to BIGRec but mitigates amplification bias by removing length normalization during beam search decoding. In addition, it incorporates an ensemble design with traditional models, which we omit for a fair comparison.

Compared Token Reweighting Methods To evaluate the effectiveness of IGD as a token reweighting strategy, we compare it against other token-level methods that can be seamlessly integrated into LLM4Rec frameworks. These methods include:

- **Position Normalization (Pos)** [33]: This method reweights tokens during SFT based on their position in the sequence, assigning higher weight to earlier tokens compared to later tokens.
- **Causal Fine-tuning (CFT)** [33]: Building upon the Pos method, CFT introduces an additional context-aware loss term. This loss captures the difference between contextual and non-contextual token predictions (representing causal effects [57]), encouraging the model to emphasize tokens that are more tightly correlated with the input context.

B Implementation Details of Compared Method

For traditional baselines, we directly adopt the settings from the D3 paper [19], as our experimental setup is fully aligned with theirs. For LLM-based methods, we use Qwen2.5-1.5B [28] as the backbone. The batch size is set to 64, the optimizer is AdamW, the learning rate is 1×10^{-4}, and the dropout rate is 0.05. Model selection is based on validation loss, with an early stopping strategy that uses a maximum of 3 epochs and a patience of 1 epoch. All other settings follow the D3 paper. For our proposed IGD method, the tuning hyperparameter β is selected from the set $\{0.08, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 1.0\}$, and the decoding hyperparameter α is selected from $\{0.0, 0.1, 0.2, 0.3, 0.4\}$. We first search for the optimal β based on validation performance, followed by a search for the best α.

C Loss on All Datasets

This section presents a comparison of the training loss for zero-IG and non-zero-IG tokens before and after applying our IGD-tuning across all datasets. The results are summarized in Figure 7.

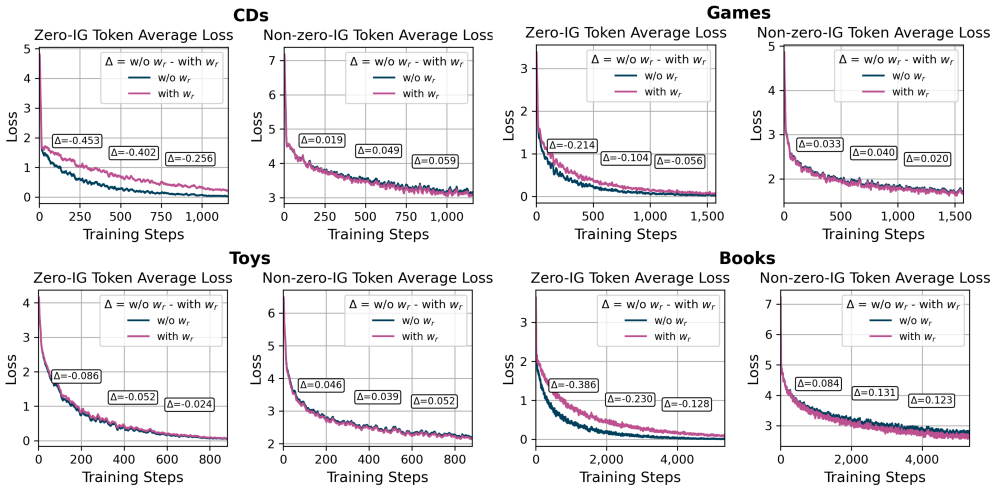

Figure 7: Loss comparison: IGD-Tuning effect on zero-IG and non-zero-IG tokens (epoch 1).

D Entropy difference on All Datasets

This section presents a comparison of entropy differentials across all datasets following the application of IGD-decoding. As the value of α increases, the decoding bias decreases. The results summarized in Figure 8

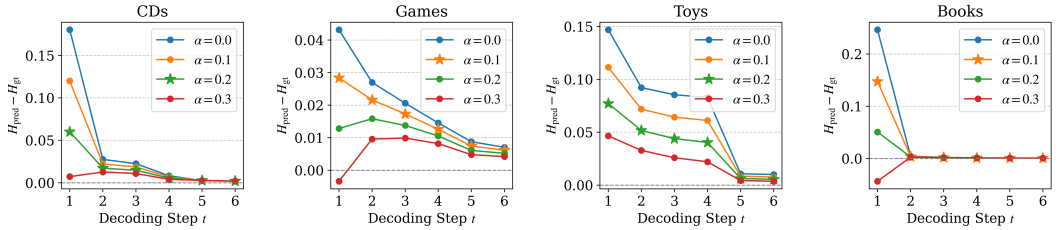

Figure 8: Entropy difference: prediction vs. ground truth after IGD-decoding. "Start" indicates optimal α selected based on HR@10 on all datasets.

E All Results for Generalizability Study

This section presents all the results for the Generalizability study. The results are summarized in Table 5 and Table 6

More dataset. To further assess generalizability, we additionally evaluated the best-performing baseline (D3) and its IGD-enhanced variant on Steam dataset (∼982K interactions). Results in Table 6 show that IGD continues to provide consistent gains over the baseline.

F Item Diversity under IGD-Decoding

We evaluate IGD-D's effect on item diversity with $\alpha \in \{0.0, 0.1, 0.2, 0.3, 0.4\}$.

First Word Repetition Rate (FWR)—lower is better—is the proportion of the most frequent first token among the top-10 recommended items. **Item Score Entropy (ISE)**—higher is better—is computed from the decoding probabilities induced by final item scores using the standard $\sum -p \log p$ aggregation.

As Table 7 is shown, increasing α consistently decreases FWR and increases ISE across all datasets.

Table 5: Performance comparison of D3+IGD vs. original D3 on LLaMA3-8B backbone across different recommendation datasets

Methods	CDs				Games			
	N@5	N@10	H@5	H@10	N@5	N@10	H@5	H@10
D3	0.0742	0.0790	0.0917	0.1062	0.0456	0.0528	0.0611	0.0832
+IGD	**0.0791**	**0.0850**	**0.0994**	**0.1179**	**0.0645**	**0.0734**	**0.0863**	**0.1137**
Improvement	*+6.60%*	*+7.59%*	*+8.40%*	*+11.0%*	*41.5%*	*+39.0%*	*+41.2%*	*+36.7%*
Methods	Toys				Books			
	N@5	N@10	H@5	H@10	N@5	N@10	H@5	H@10
D3	0.0739	0.0797	0.0916	0.1096	0.0198	0.0214	0.0249	0.0299
+IGD	**0.0885**	**0.0932**	**0.1102**	**0.1311**	**0.0282**	**0.0304**	**0.0352**	**0.0418**
Improvement	*+19.8%*	*+16.9%*	*+20.3%*	*+19.6%*	*+42.4%*	*+42.1%*	*+41.4%*	*+39.8%*

Table 6: Results on the Steam dataset ($\sim$982K interactions) under the same evaluation protocol as the main study.

Method	HR@10	NDCG@10
D3	0.1126	0.0782
+IGD	0.1184 (+5.15%)	0.0810 (+3.58%)

Mechanism Analysis: Non-decisive tokens tend to receive high logits and dominate beam expansion, reducing diversity. IGD-D increases the logits of decisive tokens (via α), yielding lower FWR and higher ISE.

G Why IGD-Tuning Adopts a Binary Weighting Scheme

Our binary-based weighting scheme is motivated by the observed behavior of IG in our setting and is not arbitrary. Two empirical observations guide the design:

(1) Distinct zero-IG token group. Tokens with zero IG form a dominant cluster (about 55% of all tokens) and exhibit disproportionately high logits with very low training loss. Treating this group as a separate class and applying a uniform down-weighting factor β effectively mitigates their outsized influence.

(2) Non-linear IG distribution among non-zero tokens. The IG distribution for non-zero tokens is highly skewed and roughly exponential rather than linear. This makes it difficult to craft a smooth, well-calibrated continuous weighting function over IG. We experimented with a simple linear and monotonic alternative:

$$w_t \;=\; \beta \;+\; (1-\beta)\cdot\frac{\mathrm{IG}}{\mathrm{IG}_{\max}}$$

As shown in Table 8, this linear function does not outperform the binary scheme.

Due to the dominance of zero-IG tokens and the non-linear nature of non-zero IG values, a binary separation with a calibrated β provides stronger and more stable improvements than a linear mapping, across all evaluated datasets.

H Effective Hyperparameter Ranges

Our method introduces two parameters: a decoding weight α and a training-time weight β. Since α is easy to tune, we only analyze β's sensitivity while fixing α=0.0. Unless otherwise noted, the search grid for β is $\{0.08, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$; after selecting β, α is tuned over $\{0.1, 0.2\}$.

Optimal β. CDs: β=0.2 (HR@10=0.1077). Games: β=0.2 (HR@10=0.0942). Toys: β=0.5 (HR@10=0.1063). Books: β=0.1 (HR@10=0.0422).

Table 7: Diversity vs. α across datasets. FWR↓ and ISE↑.

α	**CDs**		**Toys**		**Games**		**Books**	
	FWR↓	ISE↑	FWR↓	ISE↑	FWR↓	ISE↑	FWR↓	ISE↑
0.0	0.364	2.65	0.588	2.67	0.340	2.95	0.382	2.75
0.1	0.332	2.70	0.577	2.73	0.328	2.98	0.339	2.81
0.2	0.304	2.76	0.564	2.80	0.321	3.01	0.306	2.87
0.3	0.282	2.84	0.557	2.88	0.307	3.04	0.278	2.93
0.4	0.265	2.92	0.551	2.95	0.296	3.08	0.259	2.99

Table 8: Comparison of linear vs. binary IGD-T weighting across datasets. Best β (Linear): 0.4, 0.9, 0.8, 0.4 for CDs, Games, Toys, Books. Best β (Binary): 0.2, 0.2, 0.5, 0.1 for CDs, Games, Toys, Books.

Dataset	Metric	Baseline (D3)	IGD-T (Linear)	IGD-T (Binary, ours)
CDs	HR@10	0.1040	0.1072	0.1077
	NDCG@10	0.0767	0.0793	0.0800
Games	HR@10	0.0773	0.0800	0.0942
	NDCG@10	0.0477	0.0492	0.0594
Toys	HR@10	0.1029	0.1034	0.1063
	NDCG@10	0.0698	0.0692	0.0719
Books	HR@10	0.0315	0.0339	0.0422
	NDCG@10	0.0228	0.0245	0.0312

Effective ranges (HR@10 and NDCG@10 $>$ baseline). CDs: $[0.1, 0.6]$; Games: $[0.1, 0.4] \cup \{0.6\}$; Toys: $[0.2, 0.6]$; Books: $[0.1, 0.6]$.

Observation. Each dataset exhibits a broad interval where performance exceeds the baseline, making β easy to integrate into existing methods. However, to obtain the optimal β, one still needs to search over a reasonable range.