← Go to **AAAI 2026 Conference** homepage (/group?id=AAAI.org/2026/Conference)

3 Versions

Inductive Generative Recommendation via Retrieval-based Speculation

(/pdf?id=zw2mfNgLuW)

Yijie Ding (/profile?id=~Yijie_Ding2), Jiacheng Li (/profile?id=~Jiacheng_Li2), Julian McAuley (/profile?id=~Julian_McAuley1), Yupeng Hou (/profile?id=~Yupeng_Hou1)

Published: 07 Nov 2025, Last Modified: 07 Nov 2025 AAAI-26 Oral
Conference, Area Chairs, Senior Program Committee, Program Committee, Publication Chairs, Authors
Revisions (/revisions?id=zw2mfNgLuW) BibTeX

Serve As Reviewer: Yupeng Hou (/profile?id=~Yupeng_Hou1)
Keywords: Generative Recommendation, Sequential Recommendation, Inductive Recommendation
Primary Keyword: DMKM: Recommender Systems
TL;DR: We propose SpecGR, a plug-and-play framework that enables any generative recommendation models to recommend new items in an inductive setting.

Abstract:
Generative recommendation (GR) is an emerging paradigm that tokenizes items into discrete tokens and learns to autoregressively generate the next tokens as predictions. While this token-generation paradigm is expected to surpass traditional transductive methods, potentially generating new items directly based on semantics, we empirically show that GR models predominantly generate items seen during training and struggle to recommend unseen items. In this paper, we propose SpecGR, a plug-and-play framework that enables GR models to recommend new items in an inductive setting. SpecGR uses a *drafter* model with inductive capability to propose candidate items, which may include both existing items and new items. The GR model then acts as a *verifier*, accepting or rejecting candidates while retaining its strong ranking capabilities. We further introduce the guided re-drafting technique to make the proposed candidates more aligned with the outputs of generative recommendation models, improving the verification efficiency. We consider two variants for drafting: (1) using an auxiliary drafter model for better flexibility, or (2) leveraging the GR model's own encoder for parameter-efficient self-drafting. Extensive experiments on three real-world datasets demonstrate that SpecGR exhibits both strong inductive recommendation ability and the best overall performance among the compared methods.

Country Of Institutions: United States
Supplementary Material: zip (/attachment?id=zw2mfNgLuW&name=supplementary_material)
Profile Policy Agreement: I confirm that all authors have up-to-date OpenReview profiles, including their current position, institution-affliated email address, and DBLP URL. I understand that submissions with incomplete author profiles will be subject to desk rejection.
Submission Number: 11437

Filter by reply type... Filter by author... Search keywords... Sort: Newest First
Everyone Program Chairs Submission11437... Submission11437... Submission11437...
9 / 9 replies shown

Submission11437...	Submission11437...	Submission11437...	Submission11437...
Submission11437...	Submission11437...	Submission11437...	✖

Add: **Withdrawal** **Ethics Chair Author Comment**

Paper Decision

Decision by Program Chairs 07 Nov 2025, 10:56 (modified: 07 Nov 2025, 15:11)

Program Chairs, Area Chairs, Senior Program Committee, Program Committee, Authors

Revisions (/revisions?id=6MWKTtTN8x)

Decision: Oral

Add: **Ethics Chair Author Comment**

Phase 2 AC Recommendation by Area Chairs

Phase 2 AC Recommendation by Area Chairs 30 Oct 2025, 00:14 (modified: 07 Nov 2025, 14:24)

Program Chairs, Area Chairs, Senior Program Committee, Program Committee, Authors

Revisions (/revisions?id=oWJWpwjBxt)

Metareview:
Largely agree with SPC.

Acceptance Recommendation: This paper is in the bottom 25%-50% of papers presented at a top tier venue like AAAI. (Accept recommendation.)
Confidence: 4: The AC is confident but not absolutely certain

Add: **Ethics Chair Author Comment**

Phase 2 SPC Recommendation by Senior Program Committee WUB1

Phase 2 SPC Recommendation by Senior Program Committee WUB1

24 Oct 2025, 04:24 (modified: 07 Nov 2025, 16:05)

Program Chairs, Area Chairs, Senior Program Committee, Program Committee, Authors

Revisions (/revisions?id=sune3gFYFf)

Metareview:
All reviewers have provided positive evaluations. It is confirmed that the identified weaknesses can be addressed in the final version. Based on this, the manuscript is clearly recommended for acceptance.

Acceptance Recommendation: This paper is in the bottom 25%-50% of papers presented at a top tier venue like AAAI. (Accept recommendation.)
Confidence: 4: The SPC is confident but not absolutely certain

Add: **Ethics Chair Author Comment**

Rebuttal by Authors

Rebuttal

by Authors (Yupeng Hou (/profile?id=~Yupeng_Hou1), Yijie Ding (/profile?id=~Yijie_Ding2), Jiacheng Li (/profile?id=~Jiacheng_Li2), Julian McAuley (/profile?id=~Julian_McAuley1))

13 Oct 2025, 22:42

Program Chairs, Area Chairs, Senior Program Committee, Program Committee Submitted, Authors

Rebuttal:

We sincerely thank all reviewers for their insightful feedback.

1. SpecGR vs. Hybrid Methods [3Y7o]

1. Hybrid methods like TIGER-C heuristically mix a fixed ratio of unseen items into predictions, while SpecGR uses a *model-based acceptance mechanism* over candidate semantic-IDs for adaptive inductive ability.
2. LIGER ranks output by feature similarity, while SpecGR employs *generative likelihood scores for target-aware re-ranking* (Sec. 3.2.4), capturing more nuanced semantic interactions.

2. GR's Encoder as an Inductive Drafter [3Y7o]

In SpecGR++, the encoder's last hidden states serve as *sequence representations* for retrieval-based drafting (Sec. 3.3.2). Though trained on seen items, *it learns transferable semantic ID representations* that generalize to unseen ones.

3. Complexity and Scalability [6VDv, aWof, 83mn]

Comprehensive analyses are provided in the Appendix.

- [aWof] In F.3, SpecGR achieves a *constant acceleration factor* over TIGER, determined by the average acceptance rate.
- [83mn] In F.1, drafting and verification overheads are minor and become *negligible* with larger model or corpus scales.
- [6Vdv] In F.5, SpecGR remains efficient and scalable under strict latency constraints, suitable for *real-time, large-scale deployment*.

4. Drafter–Verifier Alignment [3Y7o]

SpecGR does not constrain the drafter's modality or feature space, as drafting retrieves item IDs from embedding, and verifier scores them in the semantic ID space. Alignment is unnecessary by design, though higher alignment improves efficiency (F.4, Fig. 5).

5. Integration with Different Drafters [83mn]

SpecGR remains robust across drafters. Integrating drafters (SpecGR++, Semantic-KNN, UniSRec) with TIGER yields consistent overall NDCG@50 gains:

Drafter	N@50
GR Encoder	0.0225
S-KNN	0.0231
UniSRec	0.0239
Avg Δ%	20.1

6. Qualitative Analysis [83mn]

SpecGR (S) correctly outputs unseen items, whereas TIGER (T) only recommends seen ones: User history: [..., VR headset, Xbox controller].

Method	Pred	SIDs	Token Log-Likelihood
T	PlayStation console (seen) ✗	[152,195,350,434]	[-1.03,-1.95,-1.40,-0.00]
S	Xbox console (unseen) ✓	[152,173,373,441]	[-1.03,-2.35,-0.89,-6.54] (Acc. score = -1.42)

7. Additional Datasets, Baselines, and Evaluation [6VDv, aWof, 83mn]

We thank the reviewers for their suggestions on extending datasets [6VDv], adding baselines [aWof], and cold-start analysis [83mn]. We will add these in our future work.

Add: **Ethics Chair Author Comment**

Review for Inductive Generative Recommendation via Retrieval-based Speculation

Official Review by Program Committee 6VDv 06 Oct 2025, 00:20 (modified: 07 Nov 2025, 12:14)

Program Chairs, Area Chairs, Senior Program Committee, Program Committee Submitted, Program Committee 6VDv, Authors

Revisions (/revisions?id=mXid6c3j6f)

Review:

Summary: This paper is of solid quality with a well-motivated problem definition and carefully designed methodology. The writing is clear and logically organized, with illustrative figures that help readers follow the proposed framework. The originality lies in adapting speculative decoding from LLMs to recommendation, not for speed but to enhance inductive capability, which is a creative and non-trivial extension. The significance is considerable, as the work directly addresses a fundamental limitation of generative recommendation models—the difficulty of recommending unseen items—and demonstrates a plug-and-play solution that achieves strong empirical results and shows clear potential for industrial adoption. Overall, the contribution is both technically sound and practically meaningful.

Pros:

1. Well-motivated and important problem: Clearly identifies the gap in GR models' inductive capability and shows empirical evidence of their limitations.
2. Novel framework: Adapts speculative decoding into recommendation in a creative way, repurposing draft-verify for inductive generalization instead of acceleration.
3. Comprehensive experiments: Strong empirical validation across datasets, with careful ablation studies and analyses that support the claims.
4. Practical relevance: The proposed method is plug-and-play and parameter-efficient (SpecGR++), making it feasible for large-scale deployment scenarios.

Cons:

1. Scope of evaluation: Experiments are conducted only on Amazon datasets; evaluation on more dynamic domains (e.g., news, video) would strengthen claims of inductive capability.
2. Theoretical depth: While the framework is well-motivated, there is little formal analysis of why draft-verify improves unseen item coverage beyond empirical evidence.

Question:

How would the proposed framework scale in real-time recommendation scenarios where latency constraints are very strict (e.g., short-video feeds)?

Rating: 7: Good paper, accept
Confidence: 4: The reviewer is confident but not absolutely certain that the evaluation is correct

Add: **Ethics Chair Author Comment** **Author Review Evaluation**

Review for Submission 11437

Official Review by Program Committee aWof 05 Sept 2025, 02:42 (modified: 07 Nov 2025, 12:14)

Program Chairs, Area Chairs, Senior Program Committee, Program Committee Submitted, Program Committee aWof, Authors

Revisions (/revisions?id=TdjHbWN0r5)

Review:

Summary

This paper addresses a major limitation of Generative Recommendation (GR), an emerging sequential recommendation paradigm: while GR models tokenize items into "semantic IDs" and autoregressively generate next tokens for predictions (offering better scalability than traditional transductive methods like SASRec), they fail to recommend unseen items. To solve this, the authors propose SpecGR (Speculative Generative Recommendation), a plug-and-play framework that enables GR models to operate in an inductive setting (recommending unseen items). SpecGR draws on the "drafter-verifier" concept from speculative decoding but integrates models with complementary strengths. The paper also introduces SpecGR++, a parameter-efficient variant where the GR model's own encoder acts as the drafter (eliminating the need for a separate auxiliary model). Experiments on three real-world datasets (split chronologically to simulate new item arrival) confirm SpecGR's effectiveness: it drastically improves GR's ability to recommend unseen items while outperforming existing methods in overall recommendation performance.

Pros

- Unlike prior work that either ignores GR's unseen-item problem or blends GR outputs with non-GR methods (undermining GR's strengths), SpecGR directly addresses the root cause (GR's overfitting to training semantic IDs) via a drafter-verifier design.
- SpecGR is "plug-and-play," meaning it can integrate with any existing GR model without requiring major reengineering. Additionally, the two drafting variants (auxiliary model for flexibility, GR encoder for parameter efficiency) cater to different use cases.
- The authors use chronological dataset splitting (fixed timestamp cut-offs) to simulate the inductive setting—mimicking how new items appear over time in platforms like news or short videos. This is far more realistic than random train-test splits (which do not account for unseen items) and ensures results reflect SpecGR's performance in practical scenarios.

Cons

- While the paper mentions using a "KNN-based inductive model" as the initial drafter, more detailed information, e.g., what side information (e.g., item descriptions, user demographics) the KNN model uses, are suggested to be added, since these features used for drafter could affect the quality of the candidate items.
- SpecGR adds a drafter step to the GR pipeline, which likely introduces computational overhead. Therefore, it would be better to conduct complexity analysis or running time comparison to demonstrate the efficiency of SpecGR.
- More recent baselines (from 2024/2025) are also suggested to be added for a more holistic comparison.

> I would like to thank the authors for their rebuttal and keep my rating.

Rating: 7: Good paper, accept
Confidence: 5: The reviewer is absolutely certain that the evaluation is correct and very familiar with the relevant literature

Add: **Ethics Chair Author Comment** **Author Review Evaluation**

Review of Submission11437

Official Review by Program Committee 3Y7o 01 Sept 2025, 10:25 (modified: 07 Nov 2025, 12:14)

Program Chairs, Area Chairs, Senior Program Committee, Program Committee Submitted, Program Committee 3Y7o, Authors

Revisions (/revisions?id=mLYawHxzDX)

Review:

Summary

SpecGR is a plug-and-play “draft-then-verify” framework for generative recommendation: an inductive drafter proposes candidate items—including unseen ones—then the GR model verifies them with a target-aware query-likelihood that ignores the non-semantic identification token; guided re-drafting and adaptive exiting complete the loop. Evaluated on three benchmarks, SpecGR delivers the best overall results and markedly better cold-start performance—up to +20.64% Recall@50 and +14.8% NDCG@50—while preserving in-sample quality.

Strengths

- This paper Introduces a novel draft-then-verify paradigm for GR—repurposing speculative decoding for inductive recommendation and allowing either an auxiliary KNN drafter or self-drafting via the GR encoder.
- It clearly diagnoses a core GR failure mode—overfitting to seen semantic-ID patterns and near-zero performance on unseen items—thereby motivating an on-the-fly inductive framework.
- The method is technically sound and well evaluated.
- Empirically, SpecGR delivers strong overall results and markedly improves unseen-item performance without compromising in-sample quality.

Weaknesses

- Since the proposed method fundamentally combines discriminative inductive models (drafter) with GR frameworks, the paper should more explicitly articulate how it differs from, and improves upon, existing hybrid approaches.
- SpecGR++ reuses the GR model’s encoder as a self-drafter, yet the explanation of how this enables the discovery of unseen items is vague. Since the encoder is trained on semantic ID sequences of seen items, it is not evident why or how it can generalize effectively to unseen items.
- When using an auxiliary inductive model (e.g., UniSRec) for drafting, it is unclear how the embedding space of the auxiliary model aligns with the GR’s semantic ID–based space. Does this require the modality-based representations and GR embeddings to come from the same encoder, or is there a mapping mechanism to ensure consistency?

Rating: 7: Good paper, accept

Confidence: 4: The reviewer is confident but not absolutely certain that the evaluation is correct

Add: **Ethics Chair Author Comment** **Author Review Evaluation**

the paper presents a valuable and well-executed research contribution that significantly advances generative recommendation by introducing crucial inductive capabilities

Official Review by Program Committee 83mn 31 Aug 2025, 03:45 (modified: 07 Nov 2025, 12:14)

Program Chairs, Area Chairs, Senior Program Committee, Program Committee Submitted, Program Committee 83mn, Authors

Revisions (/revisions?id=cV9EIMEMdN)

Review:

The paper "Inductive Generative Recommendation via Retrieval-based Speculation" proposes SpecGR, a novel framework for generative recommendation (GR) models that addresses their limitation in recommending new, unseen items. The work introduces a draft-and-verify mechanism, inspired by speculative decoding, where an inductive drafter

proposes candidate items (potentially new), and a GR model acts as a verifier to accept or reject them based on likelihood. The paper also explores two drafting strategies: an auxiliary model and a self-drafting approach using the GR model's own encoder.

Quality

The paper is well-structured and provides a clear description of the proposed SpecGR framework. The methodology is detailed, explaining the four core components: Inductive Drafting, Target-aware Verifying, Guided Re-drafting, and Adaptive Exiting. The authors present two distinct drafting strategies, which adds to the comprehensiveness of the approach. Experimental results on three real-world datasets (Video Games, Office Products, Cell Phones and Accessories from Amazon Reviews 2023) demonstrate SpecGR's strong inductive recommendation ability and overall performance compared to existing methods. The ablation study further validates the contribution of each component of the framework.

Clarity

The paper is generally well-written and easy to follow. The introduction clearly articulates the problem of GR models struggling with unseen items and the motivation for SpecGR. Figure 1 provides an excellent visual overview of the problem and the proposed solution, significantly aiding understanding. Figure 2 further illustrates the SpecGR method's workflow in detail. The mathematical formulations are clear, and the explanation of how verification scores are computed for unseen items is particularly helpful. The hyperparameter analysis and the discussion on alternative GR backbones also enhance clarity.

Originality

The core idea of adapting speculative decoding to generative recommendation for inductive capabilities is novel. While speculative decoding has been used for accelerating large language models, its application to recommendation systems to enable new-item recommendations, especially with a draft-and-verify framework that leverages different model paradigms for drafting and verification, is an original contribution. The introduction of guided re-drafting and adaptive exiting specifically for this recommendation context also demonstrates originality.

Significance

The work addresses a crucial limitation of current generative recommendation models, which primarily struggle with recommending items not seen during training. This inductive capability is highly significant for real-world scenarios where new items are constantly introduced (e.g., news, e-commerce, streaming platforms). By enabling GR models to recommend new items effectively, SpecGR has the potential to expand the practical applicability and impact of GR. The strong experimental results, outperforming various state-of-the-art methods, highlight the practical significance of this research. The proposed framework is plug-and-play, making it adaptable to different GR backbones, which further increases its potential impact.

Pros:

1. **Addresses a Critical Problem:** The paper tackles the inherent limitation of generative recommendation models in handling unseen items, which is a major challenge for real-world deployment.
2. **Novel Framework:** The SpecGR framework, inspired by speculative decoding, offers an innovative solution by integrating an inductive drafter with a GR verifier.
3. **Comprehensive Methodology:** The detailed breakdown of SpecGR into four components (Inductive Drafting, Target-aware Verifying, Guided Re-drafting, Adaptive Exiting) provides a thorough understanding of the proposed approach.
4. **Effective Drafting Strategies:** The paper explores two practical drafting strategies (auxiliary model and self-drafting), demonstrating flexibility and parameter efficiency.
5. **Strong Experimental Validation:** Extensive experiments on three public datasets show significant performance improvements over various baselines, particularly in inductive settings.
6. **Ablation Study:** The ablation study effectively demonstrates the contribution of each component and training paradigm, reinforcing the design choices.
7. **Clear Visualizations:** Figures 1 and 2 are highly effective in illustrating the problem and the proposed solution, respectively.

Cons:

1. **Computational Overhead:** While adaptive exiting is mentioned to reduce inference time, the draft-and-verify process, especially with iterative re-drafting, might still introduce notable computational overhead compared to direct generation. A more in-depth analysis or comparison of inference speed against pure GR models, beyond just "Time per Rec. (s)" in Figure 3, could be beneficial.
2. **Complexity of Hyperparameter Tuning:** The framework involves several hyperparameters (δ, γ, β, λ1, λ2), and tuning them for optimal performance could be complex, as indicated by the hyperparameter analysis. A discussion on best practices or more robust tuning strategies would be helpful.
3. **Generalizability of Drafter Model Choice:** While UniSRec is used as an auxiliary drafter, the impact of choosing different inductive models as drafters on overall performance and computational cost could be further explored.
4. **Nature of "New Items":** The paper defines "new items" as those not seen in the training set. A clearer distinction or discussion on whether these are truly *novel* items (e.g., new product launches) or simply existing items that were *unseen* during training for a particular user or segment could add nuance.
5. **Interpretation of "Unseen" Performance:** While SpecGR significantly improves performance on unseen items, the absolute recall/NDCG values for unseen items are still relatively low compared to in-sample items. A discussion on the inherent difficulty and limitations in recommending truly unseen items would be valuable.

Suggestions for Improvement:

- **Further Inference Time Analysis:** Provide a more detailed breakdown of the inference time, possibly comparing the time taken for drafting, verifying, and re-drafting stages. This would give a clearer picture of the practical efficiency gains.
- **Sensitivity Analysis to Drafter Model:** Explore how SpecGR performs with different types of inductive drafter models, beyond UniSRec, to demonstrate the framework's versatility and identify optimal drafter characteristics.
- **Discussion on Cold-Start Scenarios:** Elaborate on how SpecGR addresses different facets of the cold-start problem (e.g., brand-new items with no historical interaction vs. items with very few interactions).
- **Robustness to Drafter Quality:** Discuss the framework's robustness if the drafter model generates low-quality or irrelevant candidates, and how the verifier mitigates this.
- **Qualitative Examples:** Include a few qualitative examples of recommendations for unseen items generated by SpecGR versus a baseline GR model. This could provide intuitive insights into the framework's effectiveness.

Overall, the paper presents a valuable and well-executed research contribution that significantly advances generative recommendation by introducing crucial inductive capabilities. The methodology is sound, and the experimental results are convincing.
Rating: 6: Marginally above acceptance threshold
Confidence: 2: The reviewer is willing to defend the evaluation, but it is quite likely that the reviewer did not understand central parts of the paper

Add: **Ethics Chair Author Comment** **Author Review Evaluation**

AI Review

AI Review by Program Committee AI 27 Aug 2025, 00:16 (modified: 10 Oct 2025, 11:55)
Program Chairs, Area Chairs, Senior Program Committee, Program Committee Submitted, Program Committee AI, Authors
Revisions (/revisions?id=gy93Lx1b5m)

Review:
Title: Inductive Generative Recommendation via Retrieval-based Speculation

Synopsis of the paper The paper proposes SpecGR, a draft-then-verify framework that equips generative recommenders with inductive ability by retrieving candidate items with an inductive drafter and using a generative recommender as a verifier. The verifier scores candidates via a query-likelihood objective, with a key adjustment that ignores a tokenizer-specific identification digit to avoid penalizing unseen items. A guided re-drafting mechanism conditions the drafter on beam-search prefixes emitted by the verifier, and a self-drafting variant (SpecGR++) reuses the

GR encoder trained with contrastive and learning-to-rank objectives. Experiments on timestamp-split Amazon categories show improved overall metrics and stronger performance on unseen items; integration with a second GR backbone indicates plug-and-play generality.

Summary of Review This work addresses a real and underexplored limitation of semantic-ID generative recommenders—weak inductive coverage—by repurposing speculative decoding's drafter–verifier idea to transfer inductive capability from retrieval into GR-based ranking. The design is well-motivated and largely sound, and the empirical results are competitive across multiple baselines with useful ablations. However, there are nontrivial issues that should be resolved: reporting inconsistencies in tables (including incorrect "Improv." percentages and a LIGER metric mismatch), notation errors that can mislead reproduction, and over-strong latency claims that ignore retrieval/verification costs. Key implementation details are missing (e.g., "unseen mask," tokenizer-agnostic handling of identification tokens, and precise guided re-drafting mechanics), and the evaluation would benefit from statistical rigor and additional high-churn domains. Overall, the paper is promising and significant; addressing the issues below would materially strengthen clarity, validity, and reproducibility.

Strengths

- Clear problem and motivation
  - Persuasively demonstrates that current GR models overfit to seen semantic-ID patterns and underperform on unseen items in timestamp-split settings; the analysis connects this to the presence of a non-semantic "identification" digit in tokenization and to autoregressive decoding's preference for seen patterns.
- Conceptual contribution and technical alignment
  - Adapts speculative decoding (Leviathan, Kalman, and Matias, 2023; Chen et al., 2023; He et al., 2023b) from acceleration to capability transfer: a retrieval-based inductive drafter proposes candidates while the GR model verifies and ranks them. This division of labor matches the strengths of each component and avoids heuristic blending.
  - The "target-aware verification" score $V(x_t, X)$ uses a query-likelihood view and explicitly excludes the identification digit for unseen items, directly addressing a tokenizer-specific bias in GR likelihoods.
  - Guided re-drafting aligns the drafter's candidate space with the verifier's token-level preferences via beam prefixes, improving acceptance efficiency.
  - SpecGR++ provides a parameter-efficient self-drafting path by reusing the GR encoder with contrastive pretraining and learning-to-rank finetuning, avoiding an extra drafter.
- Empirical validation and analysis depth
  - Consistent overall gains across three timestamp-split Amazon categories; detailed in-sample vs. unseen breakdowns substantiate inductive benefits.
  - Ablations isolate contributions of inductive drafting, identification-token exclusion, guided re-drafting, verification-based re-ranking, and adaptive exiting; training ablations validate the two-stage encoder learning in SpecGR++.
  - Hyperparameter analyses (draft size δ, beam size β, threshold γ) illuminate controllable trade-offs between unseen coverage, in-sample performance, and latency.
  - Plug-and-play integration with DSI (Tay et al., 2022) suggests broader applicability beyond a single GR backbone.
- Relation to prior work is appropriate
  - Situates the method against TIGER (Rajput et al., 2023), RecFormer (Li et al., 2023), and dense-retrieval blending (Yang et al., 2024), and justifies a GR-as-verifier approach via query-likelihood insights (Hou et al., 2024).

Weaknesses

- Reporting inconsistencies and overstatements that affect validity
  - Improvement ratios in Table 1 are incorrect in multiple rows. For example, on Phones NDCG@50, the best baseline (Table 1, markdown) is LIGER at 0.0089 and SpecGR is 0.0090, a relative gain of about +1.12%, not +14.80%. Similar discrepancies exist for Office R@10/N@10. These errors materially affect interpretation.
  - LIGER's Phones NDCG@50 is inconsistent across tables: Table 1 (markdown) lists 0.0089, but Table 2's subset aggregation implies an overall ≈0.0059, which aligns with the PDF Table 1 value. This contradiction needs resolution.

- The paper states “even in the worst case, SpecGR does not incur additional time overhead compared to decoding with beam search.” This is misleading in wall-clock terms: KNN retrieval and verification add nontrivial latency beyond decoder steps. The claim should be restricted to decoder step counts or revised.
- The statement that “LIGER attains higher performance on unseen items” is overgeneralized; per Table 2, this is dataset-dependent (e.g., RecFormer is stronger on Phones’ Unseen subset).

- Notation errors and ambiguities that hinder reproducibility
  - Semantic ID definition uses inconsistent indices: $\mathrm{ID}_i := [\langle c_1^1 \rangle, \langle c_2^2 \rangle, \ldots, \langle c_i^l \rangle]$ should be $[\langle c_i^1 \rangle, \ldots, \langle c_i^l \rangle]$.
  - In the verification definition, the case conditions in $V(x_t, X)$ refer to $x \in \mathcal{I}$ and $x \in \mathcal{I}^* \setminus \mathcal{I}$ but should be $x_t \in \mathcal{I}$ and $x_t \in \mathcal{I}^* \setminus \mathcal{I}$.
  - Guided re-drafting set construction uses $(c_1^j, \ldots, c_j^j) \in B_j$ without binding to candidate i; it should be $(c_1^{(i)}, \ldots, c_j^{(i)}) \in B_j$ to indicate the candidate’s prefix matches the beam prefix.
  - Minor naming typos (e.g., “Reformer” vs. RecFormer) recur and should be normalized.
- Missing implementation details that are necessary for correctness
  - Unseen detection: Figure 2 suggests an “Unseen Mask,” but there is no formal procedure for determining whether a candidate is unseen at inference time. This is critical because $V(x_t, X)$ changes depending on whether the identification digit is excluded.
  - Tokenizer dependence: The method assumes the last digit is an identification token and drops it for unseen items. This is appropriate for TIGER-like tokenizers but not necessarily for other GR backbones such as DSI. A tokenizer-agnostic or tokenizer-aware protocol is required to generalize the verifier computation.
  - Guided re-drafting mechanics: The paper does not specify how δ (draft size) is allocated across β beam prefixes, how to handle empty or low-yield prefixes, or how δ and β interact across iterations $j \leq l$. Without this, reproducing acceptance-rate gains is difficult.
  - Final ranking calibration: When the K list is filled by combining verified candidates (scored by log-likelihood) and beam-only items (scored by beam search), the paper does not explain any calibration or normalization to ensure comparability.
  - Table 2 contains “-” entries for Unseen metrics for some GR baselines, preventing a complete inductive comparison; if measurement is infeasible, an explanation is needed.
- Evaluation breadth and statistical rigor
  - Only three Amazon categories are used. Timestamp splits are appropriate, but high-churn domains (e.g., news or short-video) would better stress inductive behavior and on-the-fly recommendation.
  - No statistical tests or variation reporting; the reproducibility checklist itself marks statistical analysis as “no.” Multiple runs, variances, and significance tests are required for reliable claims.
  - The review initially stated a “drafter-only” baseline was missing; this is incorrect—UniSRec serves as a drafter-only baseline. However, a “drafter + naive blending” baseline matched to the same drafter is still useful to isolate how much of the gain comes from verification and guided re-drafting, beyond candidate quality. The paper mentions ensemble comparisons in an appendix, but an explicit, like-for-like blending baseline in the main text would clarify attribution.
- Omitted comparisons and broader context
  - Recent strong inductive baselines such as content-based graph reconstruction (Kim et al., 2024) and contrastive cold-start collaborative filtering (Zhou, Zhang, and Yang, 2023) are not included; these could change the relative positioning on unseen items.
  - Discussion of scaling trends (e.g., trillion-parameter GR transducers in Zhai et al., 2024) is brief; it remains unclear how SpecGR’s benefits compare to simply scaling the GR verifier in terms of accuracy/latency trade-offs.
  - It is not explored whether SpecGR’s guided re-drafting and likelihood-based verification remain effective for tokenizations that interleave behavior and item tokens (Liu, Hou, and McAuley, 2024).

Suggestions for Improvement

- Fix reporting, qualify claims, and standardize notation
  - Recompute every “Improv.” percentage in Table 1 from the displayed numbers; correct and highlight any changes. Resolve the LIGER Phones NDCG@50 inconsistency between Table 1 and Table 2 (and the PDF), and ensure all tables reflect the same source runs.
  - Qualify the discussion about LIGER’s unseen performance as dataset-dependent; remove blanket statements.

- Restrict the “no additional time overhead” claim to decoder steps and provide a wall-clock breakdown that includes retrieval (KNN), verification (likelihood scoring), and decoding. Report per-request latency with mean and standard deviation.
- Correct the semantic ID notation to $[\langle c_i^1 \rangle, \ldots, \langle c_i^l \rangle]$, the verification case conditions to $x_t \in \mathcal{I}$ vs. $x_t \in \mathcal{I}^* \setminus \mathcal{I}$, and the guided re-drafting filter to $(c_1^{(i)}, \ldots, c_j^{(i)}) \in B_j$.
- Normalize naming (e.g., RecFormer) and check metric names; specify whether logs in $V(x_t, X)$ use natural or base-10 logarithms and whether scores are length-normalized (as written) consistently across variants.

- Specify the full algorithm with tokenizer-aware details
  - Provide concise pseudocode for the draft–verify loop, including:
    - How Q_j is built each iteration: KNN retrieval, per-prefix budgets (e.g., $\lfloor \delta/\beta \rfloor$ per prefix), handling empty/low-yield prefixes via backoff, and deduplication across rounds.
    - How the “Unseen Mask” is computed for each candidate at inference time (e.g., via lookup against the training item universe or an index built offline).
    - How the identification token is detected and excluded. Propose a tokenizer-agnostic interface: mark non-semantic positions in tokenization metadata and define $V(x_t, X)$ as a sum over semantic positions only. For $x_t \in \mathcal{I}^* \setminus \mathcal{I}$, exclude any non-semantic positions; for $x_t \in \mathcal{I}$, include them. This makes the method portable to GR backbones like DSI.
    - How final ranking is produced when combining verified items (log-likelihoods) and beam-only fill-ins (beam scores). Consider a calibration step (e.g., min–max scaling per batch, temperature scaling, or Platt scaling fit on validation) to put scores on a common scale.
- Strengthen evaluation breadth and rigor
  - Add statistical significance tests (e.g., paired bootstrap on session-level outcomes or Wilcoxon signed-rank across users) and report mean ± standard deviation over multiple seeds.
  - Include a “drafter + naive blending” baseline that merges the same drafter’s candidates with GR outputs via a simple heuristic (e.g., linear score interpolation or interleaving), to isolate the value of verification and guided re-drafting beyond candidate quality. Keep the candidate budget equal to SpecGR.
  - Expand to a high-churn domain such as news (e.g., MIND (Wu et al., 2020)) or short video to stress on-the-fly inductive generalization, or at least discuss expected behavior and potential adaptations (e.g., larger δ, tighter γ) for such domains.
  - Report acceptance rates per iteration j and per-prefix acceptance, and analyze performance by item popularity and recency to understand when inductive verification helps most.
  - Compare against additional strong inductive methods such as CGRC (Kim et al., 2024) and contrastive CF for cold-start (Zhou, Zhang, and Yang, 2023), to position SpecGR among diverse inductive strategies.
- Probe generality and scaling interactions
  - Evaluate SpecGR atop a scaled GR verifier (e.g., stronger decoder depth/width) to quantify the marginal benefit of drafting/verification versus simply scaling the verifier (cf. Zhai et al., 2024). Include accuracy–latency trade-offs under a fixed latency budget.
  - Discuss how guided re-drafting and verification adapt to interleaved behavior–item tokenizations (Liu, Hou, and McAuley, 2024). If feasible, include a small-scale experiment to demonstrate compatibility.
- Deployment guidance
  - Provide practical guidance for choosing δ, β, and γ to target a desired unseen proportion or latency budget. For example, select γ via an “elbow” on the validation trade-off curve between unseen Recall@K and latency; set δ to meet a target unseen acceptance rate; set β to cap per-iteration compute while protecting in-sample accuracy.
  - Include a resource breakdown (CPU/GPU time for retrieval, verifier forward passes, and decoding), and a fairness note on hyperparameter search budgets across baselines.
  - Clarify that adaptive exiting reduces decoding steps but total latency depends on retrieval–verification loops; show how many iterations are typically needed until stopping and how often the worst case ($j = l$) occurs.

References

- Chen, C., Borgeaud, S., Irving, G., Lespiau, J.-B., Sifre, L., & Jumper, J. (2023). Accelerating large language model decoding with speculative sampling. arXiv preprint arXiv:2302.01318.
- He, Z., Zhong, Z., Cai, T., Lee, J. D., & He, D. (2023b). ReST: Retrieval-based speculative decoding. arXiv preprint arXiv:2311.08252.

- Hou, Y., Mu, S., Zhao, W. X., Li, Y., Ding, B., & Wen, J.-R. (2022). Towards universal sequence representation learning for recommender systems. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (pp. 585–593). New York, NY: ACM. https://doi.org/10.1145/3534678.3539381 (https://doi.org/10.1145/3534678.3539381)
- Hou, Y., Zhang, J., Lin, Z., Lu, H., Xie, R., McAuley, J., & Zhao, W. X. (2024). Large language models are zero-shot rankers for recommender systems. In European Conference on Information Retrieval, Part II (pp. 364–381). Cham: Springer. https://doi.org/10.1007/978-3-031-56060-6_24 (https://doi.org/10.1007/978-3-031-56060-6_24)
- Kim, J., Kim, E., Yeo, K., Jeon, Y., Kim, C., Lee, S., & Lee, J. (2024). Content-based graph reconstruction for cold-start item recommendation. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 1263–1273). New York, NY: ACM. https://doi.org/10.1145/3626772.3657801 (https://doi.org/10.1145/3626772.3657801)
- Leviathan, Y., Kalman, M., & Matias, Y. (2023). Fast inference from transformers via speculative decoding. In Proceedings of the 40th International Conference on Machine Learning (ICML 2023). PMLR.
- Li, J., Wang, M., Li, J., Fu, J., Shen, X., Shang, J., & McAuley, J. (2023). Text is all you need: Learning language representations for sequential recommendation. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (pp. 1258–1267). New York, NY: ACM. https://doi.org/10.1145/3580305.3599519 (https://doi.org/10.1145/3580305.3599519)
- Liu, Z., Hou, Y., & McAuley, J. (2024). Multi-Behavior generative recommendation. In Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (pp. 1575–1585). New York, NY: ACM. https://doi.org/10.1145/3627673.3679730 (https://doi.org/10.1145/3627673.3679730)
- Rajput, S., Mehta, N., Singh, A., Keshavan, R. H., Vu, T., Heldt, L., Hong, L., Tay, Y., Tran, V. Q., Samost, J., Kula, M., Chi, E. H., & Sathiamoorthy, M. (2023). Recommender systems with generative retrieval. In Advances in Neural Information Processing Systems (NeurIPS 2023).
- Tay, Y., Tran, V. Q., Dehghani, M., Ni, J., Bahri, D., Mehta, H., Qin, Z., Hui, K., Zhao, Z., Gupta, J., Schuster, T., Cohen, W. W., & Metzler, D. (2022). Transformer memory as a differentiable search index. In Advances in Neural Information Processing Systems (NeurIPS 2022).
- Wu, F., Qiao, Y., Chen, J., Wu, C., Qi, T., Lyu, S., & Xie, X. (2020). MIND: A large-scale dataset for news recommendation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 3597–3606). Association for Computational Linguistics.
- Wu, J., Chang, C.-C., Yu, T., He, Z., Wang, J., Hou, Y., & McAuley, J. (2024). CoRAL: Collaborative retrieval-augmented large language models improve long-tail recommendation. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (pp. 3391–3401). New York, NY: ACM.
- Yang, L., Paischer, F., Hassani, K., Li, J., Shao, S., Li, Z. G., He, Y., Feng, X., Noorshams, N., Park, S., et al. (2024). Unifying generative and dense retrieval for sequential recommendations. arXiv preprint arXiv:2411.18814.
- Zhai, J., Liao, L., Liu, X., Wang, Y., Li, R., Cao, X., Gao, L., Gong, Z., Gu, F., He, J., Lu, Y., & Shi, Y. (2024). Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. In Proceedings of the 41st International Conference on Machine Learning (Vol. 235, pp. 58484–58509). PMLR.
- Zhou, Z., Zhang, L., & Yang, N. (2023). Contrastive collaborative filtering for cold-start item recommendation. In Proceedings of the ACM Web Conference 2023 (pp. 928–937). New York, NY: ACM. https://doi.org/10.1145/3543507.3583286 (https://doi.org/10.1145/3543507.3583286)