CARTS: Collaborative Agents for Recommendation Textual Summarization

Jiao Chen [*1] **Kehui Yao** [*1] **Reza Yousefi Maragheh** [*2] **Kai Zhao** [*2]

Abstract

Current recommendation systems often require some form of textual data summarization, such as generating concise and coherent titles for product carousels or other grouped item displays. While large language models have shown promise in NLP domains for textual summarization, these approaches do not directly apply to recommendation systems, where explanations must be highly relevant to the core features of item sets, adhere to strict word limit constraints. In this paper, we propose CARTS (Collaborative Agents for Recommendation Textual Summarization), a multi-agent LLM framework designed for structured summarization in recommendation systems. CARTS decomposes the task into three stages—Generation Augmented Generation (GAG), refinement circle, and arbitration, where successive agent roles are responsible for extracting salient item features, iteratively refining candidate titles based on relevance and length feedback, and selecting the final title through a collaborative arbitration process. Experiments on large-scale e-commerce data and live A/B testing show that CARTS significantly outperforms single-pass and chain-of-thought LLM baselines, delivering higher title relevance and improved user engagement metrics.

1. Introduction

Modern recommendation systems increasingly rely on textual summarization to enhance transparency, usability, and engagement. A common example is the need to generate concise and coherent titles for grouped item displays—such as product carousels or recommendation modules—that communicate the shared theme or purpose of the items shown (Ge et al., 2022). These summary titles serve not only to improve user understanding but also to drive attention and interaction within limited user interface space (Zhang et al., 2020). Figure 1 illustrates such a scenario in e-commerce, where a module displays several sundresses. The accompanying module title—"Versatile Summer Dresses: Stylish and Comfortable"—serves as a human-readable summary of the visual and semantic commonalities across the items.

Automatically generating such titles presents unique challenges in semantic aggregation, coverage, and conciseness. Multi-agent LLM frameworks have shown promise in domains like law, finance, and long-form summarization—where agents collaboratively process lengthy, unstructured documents—their applicability to recommendation systems remains limited and underexplored. In contrast to these domains, recommendation explanation involves summarizing structured item metadata (e.g., titles, categories, reviews) across diverse products into a single, short, behaviorally effective title (Wang et al., 2024a; Liang et al., 2023; Du et al., 2023). This task introduces unique challenges: explanations must be generated under strict length constraints, reflect semantic overlap across multiple items, and align with business objectives such as user engagement and conversion. Existing agentic frameworks are typically optimized for coherence and factuality, not for maximizing item coverage or optimizing for click-through rates in user interfaces.

To fill the research gap above, we propose CARTS (Collaborative Recommendation Agent Framework for Titles), a multi-agent LLM-based framework for generating module-level recommendation explanations. This collaborative process enables more accurate, diverse, and UI-aligned summaries compared to single-agent LLM baselines. CARTS decomposes the task into three stages. First, in the *Generation Augmented Generation (GAG)* stage, a two-step process is used to distill key item features and synthesize initial candidate titles. Next, during the *Refinement Circle*, feedback agents iteratively critique each title with respect to item relevance, stylistic constraints, and length limitations. These critiques are then used to guide the generator in producing progressively improved title versions across multiple iterations. Finally, in the *Arbitration* stage, an Arbitrator selects the final title that best balances semantic relevance

[*]Equal contribution [1]Walmart Global Tech, Bellevue, WA, US [2]Walmart Global Tech, Sunnyvale, US. Correspondence to: Jiao Chen <jiao.chen0@walmart.com>, Kehui Yao <kehui.yao@walmart.com>, Reza Yousefi Maragheh <reza.yousefimaragheh@walmart.com>, Kai Zhao <kai.zhao@walmart.com>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

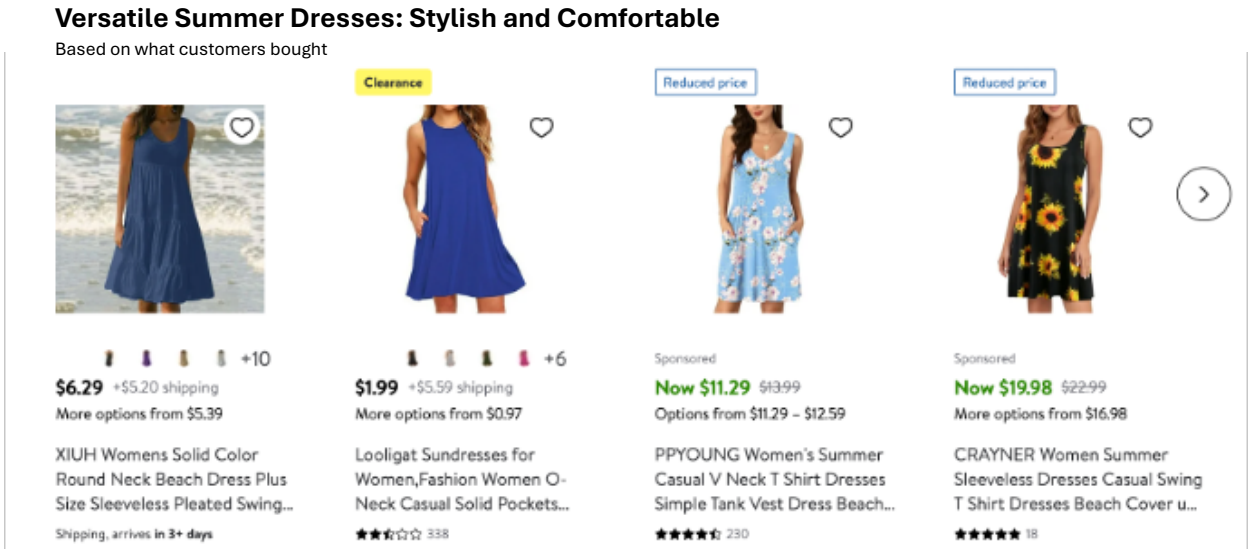

Figure 1. Example of a Product Carousel with Human-Curated Module Title

and constraint satisfaction.

In addition to its architectural novelty, CARTS provides a theoretical foundation for the agent collaboration process by framing title generation as a constrained coverage optimization problem. We theoretically analyze the *Refinement Circle* and derive approximation guarantees on the number of refinement steps required to reach near-optimal coverage under practical character-length constraints. Specifically, under assumptions of reliable feedback and generator agents, we prove that CARTS can achieve a desired fraction of optimal relevance with high probability in a bounded number of iterations. This theoretical insight not only differentiates CARTS from prior heuristic agent workflows, but also grounds its design in convergence-aware optimization.

We evaluate CARTS through extensive offline experiments and online A/B tests on a real-world e-commerce platform. Results demonstrate that CARTS improves both module-level relevance scores and business outcomes such as click-through rate (CTR), add-to-cart rate (ATCR), and gross merchandise value (GMV), highlighting the promise of multi-agent LLM systems in real-world recommendation workflows.

Our contribution are summarized as follows:

- We propose CARTS, a novel multi-agent LLM framework that integrates generation, refinement and arbitration to collaboratively generate concise module titles that maximize item-level relevance coverage under practical constraints.
- We provide a theoretical analysis of CARTS's *Refinement Circle*, proving an approximation guarantee on the number of steps required to achieve near-optimal item coverage under length constraints, based on reliability assumptions of the feedback and generator agents.
- We demonstrate the empirical effectiveness of CARTS through comprehensive offline experiments and online A/B tests, showing consistent improvements in both explanation quality and commercial impact.

2. Related Works

Recommendation systems are crucial in various industries, retrieving relevant documents for users based on context. Recommendation explaination can help reduce customer confusion and abandonment (Ge et al., 2022), making it important to explain recommendations and align them with explanations (Zhang et al., 2020).

Classic recommendation explanation approaches often use surrogate models trained alongside the primary recommendation model (Catherine et al., 2017; Chen et al., 2021; Wang et al., 2018). Recent advancements in Large Language Models (LLMs) have extended their applicability to various domains (Maragheh et al., 2023a;b; Chen et al., 2023; Wei et al., 2024), including recommendation explanation (Lei et al., 2023; Gao et al., 2024). However, default LLM-based frameworks may struggle with complex tasks, potentially leading to hallucinations (Huang et al., 2023). To address this, agentic frameworks have emerged, combining LLMs with planning and memory modules to enhance

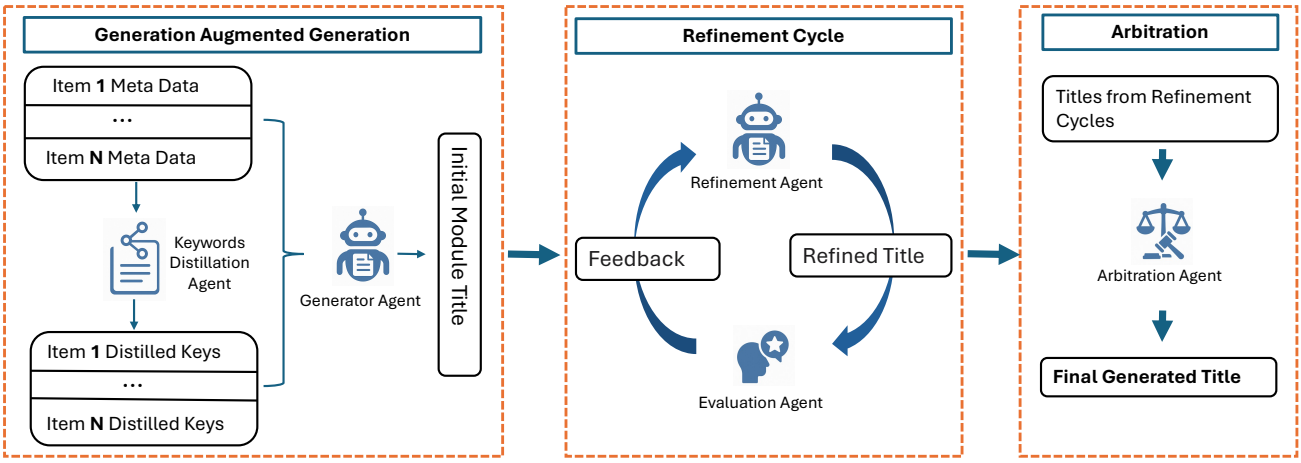

Figure 2. The overview of proposed multi-agent methods for module title generation.

performance and execute complex tasks more effectively (Wang et al., 2024a; Zhang et al., 2024).

The most simple agentic frameworks use a single agent to complete the entire sequence of tasks. While effective in many cases (Wang et al., 2023; Zhang et al., 2023), single-agent frameworks may struggle to handle highly complex, multi-faceted tasks due to their lack of specialization and inability to multitask effectively. Generating recommendation explanations, for instance, requires satisfying multiple, often competing objectives, such as ensuring high relevance to recommendations while being concise, persuasive, and transparent to customers. Single-agent approaches may not be able to sufficiently balance these diverse requirements. Multi-agent frameworks, by contrast, uses the collective intelligence of individual specialized LLM agents, are capable of imitating complex settings where humans engage in collaboration to achieve a common goals, through planning, discussions, decision making, task conduction, and evaluation (Wang et al., 2024b; Liang et al., 2023; Du et al., 2023).

Different from existing recommendation explanation studies, which usually focus on explaining the single recommended item, CARTS introduces a novel multi-agent framework specifically tailored for the task of generating unified and compelling carousel titles for a list of recommended items, aiming to improve module transparency and increase customer engagement. Previous existing multi-agent systems mainly focus on general problem-solving or conversational collaboration, CARTS deploys specialized LLM agents that iteratively refine the generated module titles to address the diverse requirements in effective eCommerce carousel titling, leading to a significant improvement in title quality compared to single-agent or simpler sequential approaches.

3. Methodology

In this section, we formally describe the proposed **CARTS** framework. Our approach consists of three main components: (i) *Generation Augmented Generation (GAG)* (ii) *Refinement Circle* (iii) *Arbitration*. Figure 2 illustrates the entire pipeline.

3.1. Problem Definition

Suppose a recommendation system produces a list of N items: $\mathcal{I} = \{I_1, I_2, \ldots, I_N\}$ to be displayed in a carousel interface. Each item I_i contains textual metadata such as: (i) Catalog Information C_i (e.g., product categories or subcategories), (ii) Title Text T_i, Supplementary Text P_i (e.g., seller descriptions or customer reviews). We write $I_i = (C_i, T_i, P_i)$ for $i = 1, 2, \ldots, N$. Our goal is to generate a succinct and persuasive *module title*, denoted M_{title}, that highlights the shared use cases, advantages, or attributes among the N recommended items. Formally, we seek a function GEN such that $M_{\text{title}} = \text{GEN}(I_1, I_2, \ldots, I_N)$.

In addition to simply generating a title, we further desire that M_{title} to be (i) relevant to as many items in $\{I_1, I_2, \ldots, I_N\}$ as possible, and (ii) satisfy a given set of imposed constraints on the written text (for instance to respects a practical length limit, such as a maximum of K characters–reflecting UI constraints or readability considerations).

Let us define a relevance indicator, $R(M_{\text{title}}, I_i)$, which evaluates how well the title corresponds to item I_i. For each item, R outputs 1 if M_{title} is deemed relevant, and 0

otherwise.[1] By assuming at least one constraint for character length, we pose the selection of M_{title} as the following optimization problem:

$$\max_{M_{\text{title}}} \quad \sum_{i=1}^{N} R\big(M_{\text{title}}, I_i\big) \tag{1}$$

$$\text{subject to} \quad \text{len}\big(M_{\text{title}}\big) \le K, \tag{2}$$

$$M_{\text{title}} \in \mathcal{C} \tag{3}$$

where $\text{len}\big(M_{\text{title}}\big)$ denotes the character length of the proposed title, and K is a threshold specified by the interface constraints. The set $\mathcal{C}$ (constraint (3)) can represent any additional conditions, such as stylistic guidelines or language appropriateness rules.

In summary, we wish to choose a title M_{title} that maximizes the count of items for which the title is relevant, while ensuring the title length remains below K. This setup allows for diverse scoring functions or additional constraints (e.g., word-based or phrasing constraints) as required by the application. In the subsequent sections, we discuss how our LLM-based multi-agent framework (CARTS) is equipped to implicitly balance these objectives and constraints when generating, refining, and selecting the final module title.

3.2. Generation Augmented Generation (GAG)

A naive large language model (LLM) approach is to concatenate $I_1, I_2, \ldots, I_N$ into a single prompt and directly request a module title. As discussed in Section 3.1, our goal is not only to generate a concise title but also to *maximize the number of items* for which this title is relevant, all while satisfying a character-length constraint (see Eq. 1–2). Direct prompting often yields irrelevant or under-specified titles that fail to cover the shared attributes or exceed length limits. Hence, we propose *Generation Augmented Generation* (**GAG**) in two steps, designed to spotlight each item's most salient features and thereby improve coverage within allowable length bounds.

3.2.1. Distillation of Keywords

In the first step, we focus on extracting a small set of keywords for each item to highlight its essential features. Let DISTILL be an LLM-driven function that produces l keywords from the catalog information, title text, and supplementary text of item I_i:

$$\{K_i^1, K_i^2, \ldots, K_i^l\} \;=\; \text{DISTILL}(C_i, T_i, P_i), \tag{4}$$

where K_i^j is the j-th keyword for item I_i. This keyword set allows the subsequent generation stage to more effectively capture overlapping aspects across all items, thus increasing the potential for broad relevance.

3.2.2. Title Generation with Augmented Prompts

Next, we augment the original metadata I_i with the distilled keyword set $\{K_i^1, \ldots, K_i^l\}$. Define G_{title} as an LLM-based function that produces a concise module title from the augmented prompt:

$$M_{\text{title}}^{(0)} \;=\; G_{\text{title}}\Big(\{(I_1, K_1^{1:l}), \ldots, (I_N, K_N^{1:l})\}\Big). \tag{5}$$

Here, $M_{\text{title}}^{(0)}$ is the *initially generated* title, which should ideally cover the shared attributes of $\{I_1, \ldots, I_N\}$ as much as possible while remaining within the length limit K. In practice, the prompt can include explicit reminders (e.g., "The title must not exceed K characters and should be relevant to as many items as possible") to help the LLM internalize these constraints.

3.3. Refinement Circle

To further improve coverage and manage the length constraint, we introduce a refinement loop involving multiple LLM agents:

1. A **generator agent**, which proposes a candidate title based on the item information and the feedback.
2. A **feedback agent**, which evaluates the candidate title and provides natural language feedback to the item list. For example: (i) whether the title adheres to the character limit K, and (ii) whether each item is relevant to the title.

Formally, let EVAL be a function that critiques a candidate title $M_{\text{title}}^{(0)}$ with respect to the item set $\mathcal{I}$ and the constraints in Eqs. 1–2. The feedback is then used to refine the generation:

$$\begin{aligned} \text{Feedback} \;&=\; \text{EVAL}\big(M_{\text{title}}^{(0)},\, I_1, \ldots, I_N\big), \\ M_{\text{title}}^{(1)} \;&=\; \text{GEN}\Big(\text{Feedback},\, M_{\text{title}}^{(0)},\, \{K_i^j\}\Big). \end{aligned} \tag{6}$$

We refer to this iterative process as *Refinement Circle*, where the generator agent refines the title based on feedback to improve coverage while maintaining the length constraint K. While prior work mostly stops after a predefined number of iterations (Wu et al., 2023; Yao et al., 2022), our key contribution is a theoretical bound on the number of iterations T required to approximate the optimal title. Specifically, we analyze the convergence behavior of the refinement circle and formally characterize how quickly it approaches the optimal title defined in Eq.,1. We present the detailed analysis in Section 3.5.

[1] In practice, R could be a more nuanced function (e.g., a continuous measure of semantic overlap), but here we use a binary indicator for simplicity.

3.4. Arbitration Agents

Since LLM sampling can yield diverse outputs, the system generates k candidate titles by executing the *GAG* and *Refinement Circle* stages multiple times:

$$\{M_{\text{title}}^{(1)}, M_{\text{title}}^{(2)}, \ldots, M_{\text{title}}^{(k)}\},$$

for the same set of items $\mathcal{I}$. Each candidate title is accompanied by a corresponding "reasoning trace" (e.g., chain-of-thought or justification), and may differ in both coverage (i.e., relevance to the items) and length.

An arbitrator agent selects the best module title among the k candidates. Let ARBIT be an LLM-driven function that takes as input the k candidate titles and the moderator's summary S_{mod}, and outputs a single best title:

$$M_{\text{final}} \;=\; \text{ARBIT}\Big(\{M_{\text{title}}^{(1)}, \ldots, M_{\text{title}}^{(k)}\},\; S_{\text{mod}}\Big). \quad (7)$$

To approximate the objective in Eqs. 1–2, the arbitrator may internally rank each title based on (i) Coverage: The number of items I_i for which the title is deemed relevant, (ii) Length Compliance: Whether (or how closely) the title adheres to the character limit K, (iii) Stylistic or Additional Constraints: Including any platform or language guidelines.

Thus, M_{final} is the final module title displayed in the carousel, selected to balance item coverage and constraint satisfaction in accordance with our optimization framework. An algorithmic summary of the CARTS framework is provided in Appendix A.

3.5. Approximation Guarantee of CARTS

In this subsection, we derive a theoretical bound on the number of iterations T required to approximate the optimal title during *Refinement Circle*. Let $\mathcal{C}$ denote the set of all feasible titles satisfying the length constraint $\text{len}(M) \le K$. We define the optimal achievable coverage as

$$\text{OPT} := \max_{M \in \mathcal{C}} \sum_{i \in \mathcal{N}} R(M, I_i),$$

where $R(M, I_i) \in \{0, 1\}$ indicates whether the title M covers item I_i.

To enable the convergence analysis, we introduce two assumptions:

Assumption 3.1 (β-reliable feedback agent)**.** At iteration m let $C_m = \sum_{i \in \mathcal{N}} R\big(M_{\text{title}}^{(m)}, I_i\big)$ be the current coverage. If $C_m < \text{OPT}$, the feedback agent outputs a non–empty set $\mathcal{U}_m \subseteq \mathcal{N}$ such that $\exists I_j \in \mathcal{U}_m$ with $R(M_{\text{title}}^{(m)}, I_j) = 0$ (i.e. *at least one* uncovered item is flagged) with probability at least $\beta > 0$.

Assumption 3.2 (γ-reliable generator agent)**.** Conditioned on $\mathcal{U}_m$, the generator produces a refined title $M_{\text{title}}^{(m+1)}$ satisfying

1. $C_{m+1} - C_m \ge 1$ **(covers *at least one* new item);**
2. $\text{len}\big(M_{\text{title}}^{(m+1)}\big) \le K$;
3. $R(M_{\text{title}}^{(m+1)}, I_i) \ge R(M_{\text{title}}^{(m)}, I_i)$ for all i **(no regression).**

The generator succeeds with probability $\gamma > 0$.

Theorem 3.3 (Approximate optimal coverage in T rounds)**.** *Fix $\alpha \in (0, 1]$ and $\varepsilon \in (0, 1)$. Let $p := \beta\gamma$ and define*

$$\Lambda(\alpha, \beta, \gamma, \text{OPT}, \varepsilon) \;:=\; \Big\lceil \frac{\alpha\,\text{OPT} - C_0}{p} \;+\; \frac{2\ln(1/\varepsilon)}{p} \Big\rceil.$$

Under Assumptions 3.1–3.2, running the generator–feedback loop for $T \ge \Lambda(\alpha, \beta, \gamma, \text{OPT}, \varepsilon)$ iterations guarantees

$$\Pr\big[C_T \;\ge\; \alpha\,\text{OPT}\big] \;\ge\; 1 - \varepsilon.$$

Proof. For $m \ge 0$ define

$$Y_{m+1} \;:=\; \mathbf{1}\big[C_{m+1} > C_m\big].$$

By Assumptions 3.1–3.2, whenever $C_m < \text{OPT}$

$$\Pr\big[Y_{m+1} = 1 \;\big|\; \text{history}\big] \;\ge\; p = \beta\gamma,$$

and the Y_{m+1} are conditionally independent across rounds.

Let $S_T := \sum_{m=1}^{T} Y_m$ be the number of rounds that achieve *any* improvement. Since each such round covers at least one additional item and never loses coverage,

$$C_T \;\ge\; C_0 + S_T.$$

Thus $C_T \ge \alpha\,\text{OPT}$ is implied by $S_T \ge \alpha\,\text{OPT} - C_0$.

Each Y_{m+1} stochastically dominates $\text{Bernoulli}(p)$, so S_T dominates $\text{Binomial}(T, p)$. Consequently it suffices to bound a Binomial lower tail.

For $Z \sim \text{Binomial}(T, p)$ with mean $\mu = pT$ and any $\delta \in (0, 1)$, by Chernoff bound,

$$\Pr[Z \le (1-\delta)\mu] \;\le\; \exp\big(-\tfrac{\delta^2\mu}{2}\big).$$

Choose δ so that $(1-\delta)\mu = \alpha\,\text{OPT} - C_0$, i.e. $\delta = 1 - \frac{\alpha\,\text{OPT} - C_0}{pT}$. Requiring the bound to be $\le \varepsilon$ gives

$$pT - (\alpha\,\text{OPT} - C_0) \;\ge\; 2\ln(1/\varepsilon),$$

which is equivalent to $T \ge \Lambda(\alpha, \beta, \gamma, \text{OPT}, \varepsilon)$. At this T $\Pr[S_T < \alpha\,\text{OPT} - C_0] \le \varepsilon$, so $\Pr[C_T \ge \alpha\,\text{OPT}] \ge 1 - \varepsilon$. □

Corollary 3.4 (Expected iterations to achieve OPT)**.** *Let $T_{\text{OPT}} := \min\{m : C_m = \text{OPT}\}$ and set $T^\star = (\text{OPT} - C_0)/(\beta\gamma)$. Then under Assumptions 3.1–3.2*

$$\mathbb{E}[T_{\text{OPT}}] \;\le\; \frac{\text{OPT} - C_0}{\beta\gamma} \;+\; \frac{2}{\beta\gamma}.$$

Proof. Proof details can be seen in Appendix D. □

Table 1. Compare CARTS with benchmarks.

Benchmarks	Beauty		Electronics		Fashion		Home and Kitchen	
	LLM judge	BERT	LLM judge	BERT	LLM judge	BERT	LLM judge	BERT
Vanilla	0.73	0.56	0.852	0.572	0.739	0.591	0.831	0.562
DRE	0.745	0.582	0.851	0.602	0.779	0.6	0.853	0.57
CoT	0.744	0.57	0.866	0.578	0.751	0.6	0.845	0.567
LLM-CoT	0.809	0.57	0.865	0.575	0.835	0.589	0.872	0.57
CARTS	0.891	0.634	0.939	0.655	0.928	0.70	0.953	0.631

4. Experiments

4.1. Datasets

For our experimental evaluation, we utilize the Amazon Review dataset (Hou et al., 2024) across four categories: Beauty, Electronics, Fashion, and Home and Kitchen. We randomly sampled 1,000 anchor items from each category. For each anchor item, we generated 10 similar recommended items, yielding a total of 40,000 recommended items across all categories. This comprehensive sampling strategy ensured a diverse and representative dataset for our study. Similar item recommendations were generated using an approximate nearest neighbor (ANN) model in the item embedding space. Item embedding vectors are derived from item categories, titles, and descriptions with the Universal Sentence Encoder (USE) model (Cer et al., 2018).

4.2. Benchmark Models and Evaluation Metrics

To demonstrate the efficacy of our proposed framework, we compare its performance against four benchmark models: Vanilla GPT (Vanilla), which utilizes a single vanilla LLM call to generate the module title in a single step; Chain of Thought (CoT), which enhances the generation process by instructing the model to think step-by-step, following the method proposed by Kojima et al. (2022); LLM-guided Chain of Thought (LLM-CoT), which first prompts the LLM to explicitly outline the reasoning steps required for the title generation task before executing the task, as described in Zhang et al. (2022); and the Data-level Recommendation Explanation (DRE) framework, which generates keywords for individual items first and then creates the title based on these keywords (Gao et al., 2024).

We evaluate the quality and relevance of the generated titles using two metrics.

- **BERT Score:** Computes semantic similarity between the generated title and each item's description using token-level cosine similarity of BERT embeddings. We calculate the score for each item-title pair and report the module-level average. Higher scores indicate stronger relevance.
- **LLM Judge Score:** Uses GPT-4o (OpenAI, 2023), guided by the Chain of Steps prompting strategy (Zheng et al., 2024), to assess if a title accurately represents each item. The model outputs 1 (relevant) or 0 (not), and the module-level score is the average across items.

4.3. Experiment Results with Benchmarks

Table 1 presents the performance comparison of four methods—Vanilla, CoT, LLM-CoT, and our proposed method CARTS—on the Amazon Review dataset across four product categories. We report BERT scores and LLM judge scores using GPT-4o for the generated title results.

The Vanilla method, a single LLM call without explicit guidance, exhibits the lowest performance across all categories on both metrics. It often produces generic titles that lack comprehensive item coverage, leading to lower semantic alignment (BERT Score) and reduced LLM-judged relevance.

The DRE method, which distills keywords from item information to guide title generation, consistently improves upon Vanilla. It shows an LLM judge score improvement of 2-5% and a BERT score improvement of 1.4-5.3%, demonstrating the benefit of providing structured input for the LLM (except Electronics LLM judge score).

The CoT approach enhances performance by prompting the model for step-by-step reasoning, improving LLM judge scores by 1-2% and BERT scores by 1-1.6% over Vanilla. This guided reasoning helps identify more salient features. However, CoT's single-agent, single-pass nature still limits its ability to fully capture diverse attributes or correct misalignments.

LLM-CoT explicitly separates reasoning and generation, leading to significant gains in LLM judge scores compared to CoT (e.g., Beauty: 0.744 to 0.809; Fashion: 0.751 to 0.835). However, its BERT scores do not show a corresponding improvement, and its effectiveness remains bounded by internal planning without external feedback for iterative refinement.

Our proposed framework, CARTS, achieves the highest performance across all categories and metrics. It significantly

Table 2. Ablation studies of CARTS over four categories.

Benchmarks	Beauty		Electronics		Fashion		Home and Kitchen	
	LLM judge	BERT	LLM judge	BERT	LLM judge	BERT	LLM judge	BERT
CARTS w.t. refinement	0.827	0.633	0.9215	0.652	0.876	0.702	0.927	0.627
CARTS w.t. arbitrator	0.845	0.633	0.93	0.653	0.889	0.701	0.9345	0.628
CARTS	0.891	0.634	0.939	0.655	0.928	0.70	0.953	0.631

improves both LLM judge (12-26%) and BERT (10-18%) scores compared to Vanilla.

In summary, the offline results demonstrate a consistent and substantial performance gap, validating CARTS's design choices and highlighting the importance of agent collaboration and feedback-driven optimization for effective recommendation explanation.

4.4. Ablation Study

We conducted an ablation study to systematically evaluate the individual contributions of the critical components within CARTS. Specifically, we compared three variants: (i) CARTS without refinement cycle, testing the impact of the iterative evaluation-refinement loop; (ii) CARTS without arbitrator, examining the effect of removing the final decision-making agent; and (iii) the complete CARTS framework.

Results demonstrate that removing the refinement cycle significantly reduces LLM Judge scores across all categories. LLM judge scores reduces between 1.9% to 7.2% for four categories, underscoring the importance of iterative refinement in enhancing the relevance. Similarly, removing the arbitrator agent caused a noticeable decline in module title relevance performance, LLM judge score reduces between 1% to 5.4% for four categories, highlighting that explicitly selecting the best candidate title among multiple refined options is crucial for achieving optimal coverage. The complete CARTS framework consistently delivered the highest scores, confirming the indispensable role of each component.

4.5. Comparision with Different LLM on Summarazation

Table 3 compares five LLMs using BERT Score and LLM Judge Score across four categories. GPT-4o achieves the highest performance on both metrics, with BERT Scores ranging from 0.631 to 0.700 and LLM Judge Scores consistently above 0.89. This indicates strong semantic coverage and item-level relevance, making GPT-4o the most reliable model for our task.

Gemini-2.0-Flash ranks second, outperforming Gemini-1.5-Flash across all categories. LLaMA-3 follows with moderate performance. In contrast, GPT-3.5 shows the weakest results, with LLM Judge Scores around 0.60 and BERT Scores between 0.565 and 0.695, reflecting poor semantic alignment and relevance.

Moreover, we observe that all models except GPT-3.5 consistently respect the character-length constraint, which is critical in real-world UI applications. Based on these results, we adopt GPT-4o as the final model in our framework due to its superior alignment, relevance, and constraint adherence.

4.6. Case studies

In Figure 3, we show two examples of module titles generated with CARTS. In Figure 3(A), this carousel shows a list of Apple MacBook laptops, including Pro and Air two models. The generated title "**Sleek and high-performance MacBook Pro and Air**" successfully captures the two models and highlights the two key advantages of them ("sleek" and "high performance"). In Figure 3(B), this carousel presents a list of portable speakers from various brands. Instead of a simple summarization "Portable Speakers", CARTS provides a carousel title showing the use scenarios of these speakers: "Outdoor and Party Music", making it more engaging and informative.

5. Online A/B Test Results

To evaluate the impact of generated module titles for customer engagement and business metrics, we also conducted an A/B test experiment for generated in-house module title results and report the business related metrics, including lifts on click-through rate (CTR), add-to-cart rate (ATCR), and gross merchandise value (GMV). Figure 3 shows two module title examples launched in the experiment.

In the A/B testing, the control is a black-box recommendation explanation model, while the variant has the module specific titles generated by CARTS. The traffic was split as 50-50 between control and the variant. The A/B testing results are shown in Table 4.

The results demonstrate statistically significant increase in CTR, ATCR, and GMV metrics. Specifically, the CTR has an uplift of 0.8% and ATCR has an uplift of 6.28%, indicating higher customer engagement with our generated module titles. In addition, the GMV, reflecting the total sales value, showed a remarkable increase of 3.78%. These

Table 3. Comparison of different LLMs on LLM Judge Score and BERT Score across four categories

LLM	Beauty		Electronics		Fashion		Home	
	LLM Judge	BERT	LLM Judge	BERT	LLM Judge	BERT	LLM Judge	BERT
GPT-4o	0.891	0.634	0.939	0.655	0.928	0.700	0.953	0.631
Gemini-2.0-Flash	0.850	0.581	0.870	0.566	0.840	0.640	0.860	0.617
Gemini-1.5-Flash	0.820	0.630	0.840	0.566	0.810	0.656	0.830	0.567
LLaMA-3	0.790	0.611	0.810	0.556	0.780	0.553	0.800	0.565
GPT-3.5	0.600	0.600	0.620	0.641	0.580	0.695	0.610	0.565

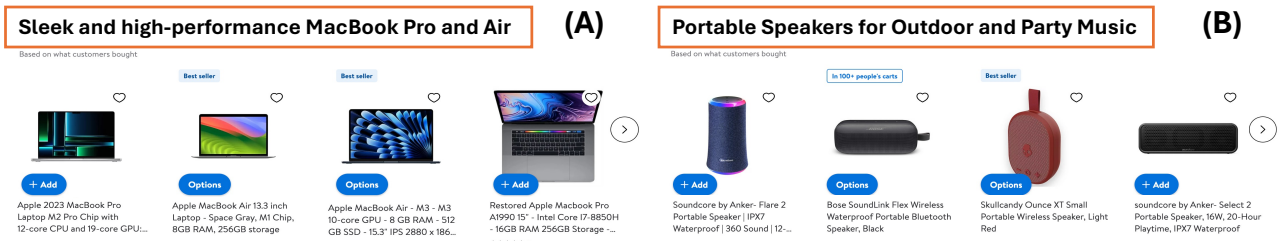

Figure 3. Two module title examples.

results confirm that the generated module titles not only enhance customer-product interactions but also drive more sales for the e-commerce platform. Please note that due to proprietary data protection agreement, details about actual dollar value implementation costs and business metrics cannot be published in the paper, however, increased business metrics far exceed the cost of running our LLM-baed multi-agent pipeline to generate the eCommerce module titles.

Table 4. A/B testing evaluation results. The lift results are percentages with 95% confidence interval.

	CTR	ATCR	GMV
Lift	0.8 ± 0.46	6.28 ± 2.07	3.78 ± 0.19

6. Conclusion

We presented CARTS, a multi-agent LLM framework for generating concise and relevant module-level titles in recommendation systems. Unlike existing multi-agent approaches developed for unstructured domains like law or finance, CARTS addresses the unique challenges of structured input, length constraints, and engagement-driven objectives in recommender settings. By decomposing the task into generation, feedback and arbitration agents, and introducing a Generation-Augmented Generation (GAG) strategy, CARTS achieves high semantic coverage under practical constraints. We further provide theoretical guarantees on convergence to near-optimal coverage. Extensive offline experiments and real-world A/B testing confirm CARTS's effectiveness in improving title relevance, CTR, and GMV, demonstrating the practical value of collaborative LLM agents in recommendation workflows.

Impact Statement

The module title generation pipeline proposed in this study can be applied to different eCommerce platforms with carousels showing a group of recommended items. In addition, the proposed multi-agent framework with refinement cycles can be applied to other complex text summarization tasks in diverse domains, such as distilling key insights from lengthy legal documents, or generating concise abstracts or keywords for scientific literatures.

References

Catherine, R., Mazaitis, K., Eskenazi, M., and Cohen, W. Explainable entity-based recommendations with knowledge graphs. *arXiv preprint arXiv:1707.05254*, 2017.

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

Chen, H., Chen, X., Shi, S., and Zhang, Y. Generate natural language explanations for recommendation. *arXiv preprint arXiv:2101.03392*, 2021.

Chen, J., Ma, L., Li, X., Thakurdesai, N., Xu, J., Cho, J. H., Nag, K., Korpeoglu, E., Kumar, S., and Achan, K. Knowledge graph completion models are few-shot learners: An empirical study of relation labeling in e-commerce with llms. *arXiv preprint arXiv:2305.09858*, 2023.

Du, Y., Li, S., Torralba, A., Tenenbaum, J. B., and Mordatch, I. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

Gao, S., Wang, Y., Fang, J., Chen, L., Han, P., and Shang, S. Dre: Generating recommendation explanations by aligning large language models at data-level. *arXiv preprint arXiv:2404.06311*, 2024.

Ge, Y., Liu, S., Fu, Z., Tan, J., Li, Z., Xu, S., Li, Y., Xian, Y., and Zhang, Y. A survey on trustworthy recommender systems. *ACM Transactions on Recommender Systems*, 2022.

Hou, Y., Li, J., He, Z., Yan, A., Chen, X., and McAuley, J. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*, 2024.

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*, 2023.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

Lei, Y., Lian, J., Yao, J., Huang, X., Lian, D., and Xie, X. Recexplainer: Aligning large language models for recommendation model interpretability. *arXiv preprint arXiv:2311.10947*, 2023.

Liang, T., He, Z., Jiao, W., Wang, X., Wang, Y., Wang, R., Yang, Y., Tu, Z., and Shi, S. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*, 2023.

Maragheh, R. Y., Fang, C., Irugu, C. C., Parikh, P., Cho, J., Xu, J., Sukumar, S., Patel, M., Korpeoglu, E., Kumar, S., et al. Llm-take: Theme-aware keyword extraction using large language models. In *2023 IEEE International Conference on Big Data (BigData)*, pp. 4318–4324. IEEE, 2023a.

Maragheh, R. Y., Morishetti, L., Giahi, R., Nag, K., Xu, J., Cho, J., Korpeoglu, E., Kumar, S., and Achan, K. Llm-based aspect augmentations for recommendation systems. 2023b.

OpenAI. Chatgpt, 2023. URL https://openai.com/blog/chatgpt.

Wang, L., Zhang, J., Yang, H., Chen, Z., Tang, J., Zhang, Z., Chen, X., Lin, Y., Song, R., Zhao, W. X., et al. When large language model based agent meets user behavior analysis: A novel user simulation paradigm. *arXiv preprint ArXiv:2306.02552*, 2023.

Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024a.

Wang, X., Chen, Y., Yang, J., Wu, L., Wu, Z., and Xie, X. A reinforcement learning framework for explainable recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pp. 587–596. IEEE, 2018.

Wang, Z., Yu, Y., Zheng, W., Ma, W., and Zhang, M. Multi-agent collaboration framework for recommender systems. *arXiv preprint arXiv:2402.15235*, 2024b.

Wei, W., Ren, X., Tang, J., Wang, Q., Su, L., Cheng, S., Wang, J., Yin, D., and Huang, C. Llmrec: Large language models with graph augmentation for recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pp. 806–815, 2024.

Wu, Q., Bansal, G., Zhang, J., Wu, Y., Zhang, S., Zhu, E., Li, B., Jiang, L., Zhang, X., and Wang, C. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*, 2023.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Zhang, A., Sheng, L., Chen, Y., Li, H., Deng, Y., Wang, X., and Chua, T.-S. On generative agents in recommendation. *arXiv preprint arXiv:2310.10108*, 2023.

Zhang, Y., Chen, X., et al. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval*, 14(1):1–101, 2020.

Zhang, Z., Zhang, A., Li, M., and Smola, A. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*, 2022.

Zhang, Z., Bo, X., Ma, C., Li, R., Chen, X., Dai, Q., Zhu, J., Dong, Z., and Wen, J.-R. A survey on the memory mechanism of large language model based agents. *arXiv preprint arXiv:2404.13501*, 2024.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 2024.

Appendix A: Algorithmic Summary

Algorithm 1 summarizes the entire **CARTS** procedure in pseudocode. This procedure approximates the objective of maximizing relevance coverage (i.e., number of items matched) while respecting a character limit K and any additional constraints (see Eqs. 1–3).

Note that under CARTS by prompting an LLM to extract only l keywords, CARTS highlights each item's essential attributes, increasing the chance of capturing shared properties across items. Multiple language agents collaborate iteratively. A `feedback` agent critiques each intermediate title for coverage (number of relevant items) and length adherence ($\leq K$), while the `generator` agent refines accordingly. In addition, generating k distinct titles enables diversity. The `moderator` summarizes each candidate's coverage, length compliance, and reasoning. Finally, the `arbitrator` selects the single best title that best satisfies the optimization criteria.

In Section 4, we show that CARTS significantly improves the relevance and transparency of generated module titles for carousel recommendations, enhancing both offline coverage metrics and online user engagement.

Algorithm 1 CARTS: Multi-Agent Generation Augmented Generation

Input: A set of items $\mathcal{I} = \{I_1, \ldots, I_N\}$, number of keywords l, number of title samples k, length limit K
Output: Final module title M_{final}
1. Keyword Distillation
for $i = 1$ to N **do**
  $\{K_i^1, \ldots, K_i^l\} \leftarrow \text{DISTILL}(C_i, T_i, P_i)$
end for
2. Initial Title Generation
$M_{\text{title}}^{(0)} \leftarrow G_{\text{title}}\big(\{(I_i, K_i^{1:l})\}_{i=1}^{N}\big)$ {Prompt includes coverage and length reminders.}
3. Refinement Circle
for $m = 1$ to k **do**
  $\text{Feedback}^{(m)} \leftarrow \text{EVAL}\Big(M_{\text{title}}^{(m-1)},\, \mathcal{I},\, K\Big)$ {Checks coverage & length.}
  $M_{\text{title}}^{(m)} \leftarrow \text{GEN}\Big(\text{Feedback}^{(m)},\, M_{\text{title}}^{(m-1)},\, \{K_i^j\}\Big)$
end for
4. Arbitration
$S_{\text{mod}} \leftarrow \text{MOD}\Big(\{(M_{\text{title}}^{(m)}, R^{(m)})\}_{m=1}^{k}\Big)$
$M_{\text{final}} \leftarrow \text{ARBIT}\Big(\{M_{\text{title}}^{(m)}\}_{m=1}^{k},\, S_{\text{mod}},\, K\Big)$ {Selects title maximizing coverage while respecting K.}
return M_{final}

Appendix B: Sample Generation Results for CARTS

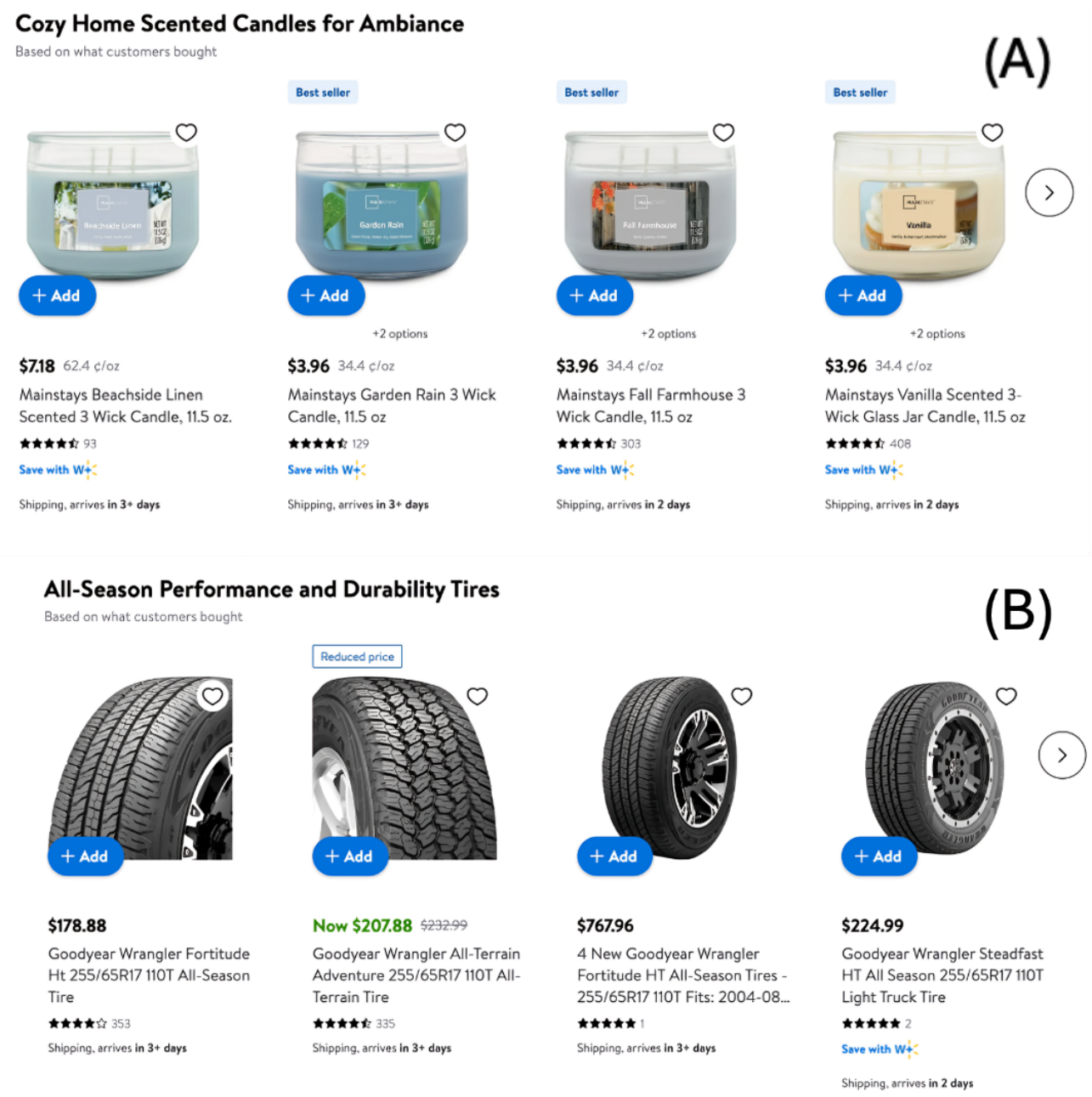

Figure 4. Sample Generation Results for CARTS. Examples from (A) Candles, (B) Tires

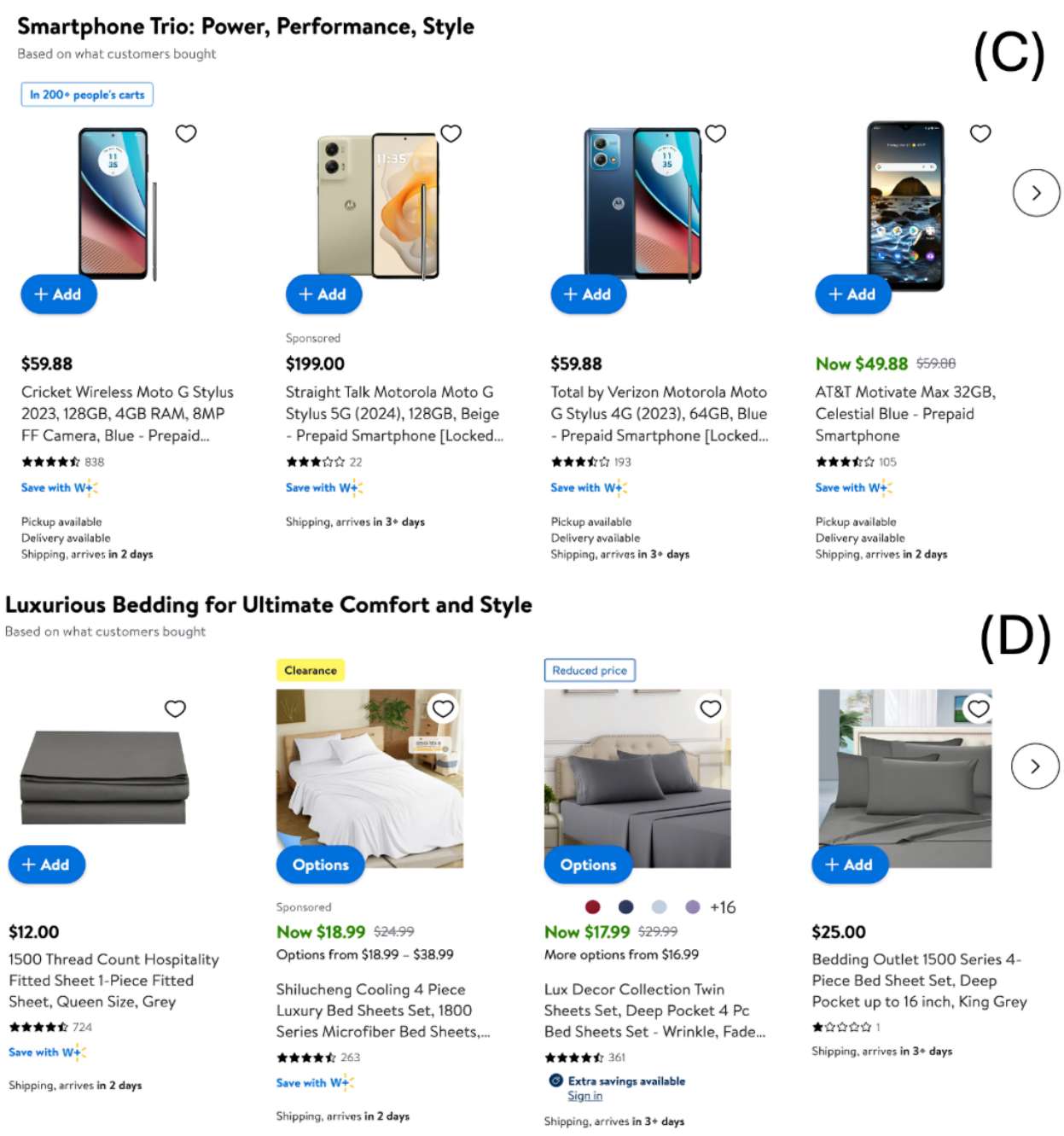

Figure 5. Sample Generation Results for CARTS. Examples from (C) Smart Phones, and (D) Bedding Products

Appendix C: prompts for CARTS

Keywords generation prompt

You are an English speaking eCommerce catalog specialist. You are an expert in generating keywords for a given product.

Consider the following product:
{prod_info}

Output at most 5 short keywords relevant to the product.

Please just output the keywords and separate them with commas.
Do not add any other text.

GAG prompt

You are an eCommerce specialist. Your expertise is in generating a title for a group of items presented in an eCommerce module.
Your task is to name this eCommerce module.

The list of products and their associated keywords are:
{prod_info_and_keys}

Generate a module title for the list of items to explain them, aiming to increase customer engagement and improve eCommerce module transparency.

Restrict the response to the following format strictly:

"title: A maximum of 10 word title that is relevant to the list of items."

Feedback Prompt

You are an evaluator that evaluates the generated titles for a group of items.

Here is the generated title:
{title}
Here is the set of items and their associated keywords that the title is generated for them:
{prod_info_and_keys}

Determine if the title is relevant enough to some or all of the items and can increase customer engagement or improve ecommerece module transparency.
If so, provide concise feedback pointing to at least one such uncovered item that the generator could improve on.
Keep the feedback within 30 words.

Regeneration Prompt

You are an eCommerce title generation specialist working in a refinement circle. Your task is to improve a previously generated title based on feedback from an evaluator.

The list of items and their associated keywords are:
{prod_info_and_keys}

The original title was:
{title}

The evaluator provided the following feedback:
{feedback}

Generate a refined title that
(1) addresses at least one uncovered or weakly covered item indicated in the feedback
(2) preserves all existing item coverage in the original title.
(3) Ensure the revised title is no more than 10 words and formatted exactly as follows:
title: ¡your refined title¿

Arbitrator Prompt

You are an eCommerece specialist tasked with selecting the best title for an eCommerce module after refinement. You are given two titles:

1. The original title: title
2. The refined title: title_2

These titles are generated based on the following list of products and their associated keywords:
{prod_info_and_keys}

Select the title that is more relevant and likely to increase customer engagement and improve module transparency. Output only the selected title.

Appendix D: proof of collary 3.4

Proof. Apply Theorem 3.3 with $\alpha = 1$. For any integer $k \geq 0$ set $\varepsilon_k := e^{-k}$. Then with $T_k := \Lambda(1, \beta, \gamma, \mathrm{OPT}, \varepsilon_k) = T^\star + \frac{2k}{\beta\gamma}$ we have $\Pr[T_{\mathrm{OPT}} > T_k] \leq e^{-k}$. Using the tail–sum representation $\mathbb{E}[T_{\mathrm{OPT}}] = \sum_{t\geq 0} \Pr[T_{\mathrm{OPT}} > t]$ and splitting the sum into blocks of length $2/(\beta\gamma)$ gives

$$\mathbb{E}[T_{\mathrm{OPT}}] \leq T^\star + \frac{2}{\beta\gamma} \sum_{k\geq 0} e^{-k} \leq T^\star + \frac{2}{\beta\gamma}.$$

□