Does Knowledge Distillation Matter for Large Language Model-Based Bundle Generation?

KAIDONG FENG, Yanshan University, Qinhuangdao, China
ZHU SUN, Information Systems Technology and Design, Singapore University of Technology and Design, Singapore, Singapore
JIE YANG, Delft University of Technology, Delft, Netherlands
HUI FANG, Key Laboratory of Interdisciplinary Research of Computation and Economics and School of Computing and Artificial Intelligence, Shanghai University of Finance and Economics, Shanghai, China
XINGHUA QU, Bytedance (Seed), Singapore, Singapore
WENYUAN LIU, Yanshan University, Qinhuangdao, China

Large Language Models (LLMs) have been extensively applied in various recommendation scenarios, including bundle generation, thanks to their exceptional reasoning capabilities and comprehensive knowledge. However, exploiting large-scale LLMs for bundle generation introduces significant efficiency challenges—primarily high computational costs during fine-tuning and inference due to their massive parameterization. Knowledge Distillation (KD) offers a promising solution by transferring expertise from large teacher models to more compact student models. This study systematically investigates KD approaches for bundle generation with the goal of minimizing computational demands while preserving performance. Specifically, we explore three critical research questions: (1) how does the *format of distilled knowledge* impact bundle generation performance? (2) to what extent does the *quantity of distilled knowledge* influence the performance? and (3) how do different *ways of utilizing the distilled knowledge* affect the performance? To support this investigation, we propose a comprehensive KD framework that (i) progressively extracts knowledge from raw data in increasingly complex forms, i.e., frequent patterns → formalized rules → deep thoughts; (ii) captures varying quantities of distilled knowledge through different sampling strategies, multi-domain accumulation, and multi-format aggregation; and (iii) exploits complementary LLM adaptation techniques—in-context learning, supervised fine-tuning, and their combination—to leverage the distilled knowledge for domain-specific adaptation and enhanced efficiency in small student models. Through extensive experiments on multiple real-world datasets, we provide valuable insights into how knowledge format, quantity, and utilization methods collectively shape

This article is supported by Open Foundation of Key Laboratory of Interdisciplinary Research of Computation and Economics (Shanghai University of Finance and Economics), Ministry of Education, China. It is partially supported by the Ministry of Education, Singapore, under its MOE AcRF Tier 1, SUTD Kickstarter Initiative (SKI 2021_06_12). We also greatly acknowledge the support of the National Natural Science Foundation of China (Grant No. 72371148 and 72192832), the Shanghai Rising-Star Program (Grant No. 23QA1403100), and the Program for Innovative Research Team of Shanghai University of Finance and Economics. Additionally, this article has been refined by Claude and ChatGPT to enhance readability.

Authors' Contact Information: Kaidong Feng, Yanshan University, Qinhuangdao, China; e-mail: fengkaidong@stumail.ysu.edu.cn; Zhu Sun (corresponding author), Information Systems Technology and Design, Singapore University of Technology and Design, Singapore, Singapore; e-mail: zhu_sun@sutd.edu.sg; Jie Yang, Delft University of Technology, Delft, Netherlands; e-mail: j.yang-3@tudelft.nl; Hui Fang, Key Laboratory of Interdisciplinary Research of Computation and Economics and School of Computing and Artificial Intelligence, Shanghai University of Finance and Economics, Shanghai, China; e-mail: fang.hui@mail.shufe.edu.cn; Xinghua Qu, Bytedance (Seed), Singapore, Singapore; e-mail: quxinghua17@gmail.com; Wenyuan Liu, Yanshan University, Qinhuangdao, China; e-mail: wyliu@ysu.edu.cn.
*For correspondence, please contact Zhu Sun and Wenyuan Liu.

ACM 1558-2868/2026/5-ART99
https://doi.org/10.1145/3808223

the performance of LLM-based bundle generation, which exhibits the significant potential of KD for more efficient yet effective LLM-based bundle generation.

CCS Concepts: • **Information systems** → **Recommender systems**; • **Computing methodologies** → *Information extraction*;

Additional Key Words and Phrases: Recommender Systems, Bundle Generation, Large Language Models, Knowledge Distillation, Efficiency

ACM Reference format:
Kaidong Feng, Zhu Sun, Jie Yang, Hui Fang, Xinghua Qu, and Wenyuan Liu. 2026. Does Knowledge Distillation Matter for Large Language Model-Based Bundle Generation? *ACM Trans. Inf. Syst.* 44, 4, Article 99 (May 2026), 40 pages.
https://doi.org/10.1145/3808223

1 Introduction

Product bundling represents a cornerstone marketing strategy that combines multiple products or services into a single package, typically offered at a discounted price [45, 61], as illustrated in Figure 1. This strategy has demonstrated remarkable effectiveness across diverse domains like telecommunications, retail, and e-commerce [16, 50]. It creates a dual benefit ecosystem: consumers gain enhanced value and convenience, while businesses experience increased sales volume and improved transaction efficiency [20]. Given these compelling advantages, product bundling has garnered significant research attention, particularly in the area of bundle recommendation—a field focused on suggesting curated item sets (i.e., bundles) to users based on their preferences, assuming the pre-existence of bundles. In particular, they treat either co-purchased products [26, 32] or user-generated lists [7, 23, 24] as bundles, or directly use pre-defined bundles by retailers [10]. However, the quality of bundles used in these methods is often suboptimal due to limitations such as noise in co-purchased data, lack of diversity in user-generated lists, and the high cost and expertise required for creating pre-defined bundles.

These limitations have led to the emergence of the bundle generation task, which is specifically dedicated to constructing high-quality bundles. Early studies utilized constraint-based methods, specifying conditions like budget limits or maximizing metrics such as savings, customer adoption, or expected revenue [5, 18, 61, 62, 64, 70]. Subsequent methods leverage neural network models (e.g., attention mechanism) to learn relationships (e.g., alternative or complementary) among items to form bundles [3, 8, 56]. Nevertheless, significant challenges persist as existing methods demonstrate critical limitations in generating bundles with necessary diversity and flexible sizing while simultaneously aligning them with the nuanced spectrum of consumer intentions and contextual needs. Recent advancements in **Large Language Models (LLMs)** have demonstrated transformative capabilities within **Recommender Systems (RSs)**, excelling at preference understanding, recommendation reasoning, and personalized content generation [51, 54]. These approaches fundamentally reconceptualize recommendation tasks as natural language problems, enabling LLMs to produce sophisticated recommendations in textual formats. This paradigm shift has naturally extended to bundle generation [48], where LLMs' reasoning capabilities and contextual understanding offer promising solutions to longstanding challenges in creating coherent, diverse, and intent-aligned bundles.

While LLMs demonstrate exceptional performance in reasoning and context understanding for bundle generation, their massive parameterization creates a critical bottleneck for practical deployment. Generating personalized bundles dynamically based on a user's current session (e.g., clicked items) requires real-time responsiveness. Standard large-scale LLMs suffer from high inference

Fig. 1. Example bundles for (1) a camera and its accessories; and (2) mystery, thriller, and historical fiction [48].

latency, rendering them impractical for such online interaction scenarios where recommendations must be delivered instantly. To mitigate these issues, **Knowledge Distillation (KD)** has emerged as a prominent technique by transferring knowledge from a larger, more complex teacher model to a smaller, more efficient student model without significantly sacrificing performance. Inspired by this, our study systematically investigates KD approaches for LLM-driven bundle generation with the dual objectives of substantially reducing computational demands while maintaining generation quality. We explore three fundamental **Research Questions (RQs)**:

- *RQ1: How does the format of distilled knowledge affect the performance on bundle generation?* We examine whether different types of knowledge yield varying effectiveness when transferred to student LLMs.
- *RQ2: To what extent does the quantity of distilled knowledge affect the performance on bundle generation?* We investigate the relationship between knowledge volume and bundle generation quality to identify optimal efficiency-effectiveness tradeoffs for student LLMs.
- *RQ3: How do different ways of utilizing the distilled knowledge affect the performance on bundle generation?* We compare various knowledge integration approaches to determine their relative effectiveness in preserving critical capabilities of student LLMs.

To answer these RQs, we design a comprehensive KD framework to investigate how knowledge format, quantity, and utilization methods collectively shape the performance of LLM-based bundle generation, establishing a foundation for more efficient bundle generation architectures without sacrificing quality. First, it progressively extracts knowledge from the raw data in increasingly sophisticated forms, ranging from frequent patterns to formalized rules and deep thoughts specifically optimized for bundle generation tasks. Next, it captures varying quantities of distilled knowledge through various sampling (i.e., random-, length-, diversity-, and difficulty-based) strategies, multi-domain accumulation, and multi-format aggregation, enabling systematic evaluation of knowledge volume–performance relationships. Finally, it exploits complementary LLM adaptation techniques, i.e., **In-Context Learning (ICL)**, **Supervised Fine-Tuning (SFT)**, and their combination, to leverage the distilled knowledge for domain-specific adaptation in lightweight, small student LLMs (e.g., Llama3.1-8B), significantly reducing computational requirements while maintaining generation quality comparable to larger language models.

Several major findings emerge from our extensive experiments on three real-world datasets. Regarding *RQ1*, different formats of distilled knowledge consistently improve bundle generation, with SFT yielding stronger and more stable gains than ICL and sometimes enabling larger student models to surpass their teachers, although performance still depends on teacher–student compatibility and the chosen knowledge format. For *RQ2*, increasing the quantity of distilled knowledge further boosts performance: ICL benefits more from higher sampling ratios due to its reliance on a larger knowledge pool for retrieval, while SFT peaks at a certain sampling ratio and then

gains substantial additional improvements by aggregating knowledge across different formats and domains, with these trends remaining consistent across teacher–student pairs. For *RQ3*, bundle generation is highly sensitive to the knowledge utilization method, where combining SFT and ICL with distilled knowledge generally performs best and SFT alone is more consistently effective than ICL alone, and these advantages remain stable across different teacher–student pairs. Overall, among the three factors, the utilization method investigated in RQ3 has the largest impact on student model performance, whereas the knowledge format examined in RQ1 contributes the least.

Overall, our contributions are highlighted as follows:

- We, for the first time, conduct a systematic exploration of KD techniques for the bundle generation task, addressing the critical challenge of computational overhead in LLM-based bundle generation while maintaining generation quality.
- We formulate three RQs to guide our exploration. To address these questions, we propose a comprehensive KD framework that (1) progressively extracts knowledge in increasingly sophisticated forms, (2) designs various strategies to capture varying quantities, and (3) employs various LLM adaptation techniques, to investigate how knowledge format, quantity, and utilization methods influence the performance of bundle generation.
- We conduct extensive experiments across three real-world datasets, whereby we identify solutions that leverage small student models to generate bundles of comparable quality to larger language models while significantly reducing computational requirements. Additionally, our empirical exploration yields valuable insights into how different characteristics of distilled knowledge collectively shape the performance of bundle generation, offering essential guidance for future research.

2 Related Work

This section briefly reviews the related literature for our work, consisting of bundle recommendation and generation, LLMs for recommendation, and KD in recommendation.

2.1 Bundle Recommendation

Early research on bundle recommendation mainly exploits *conventional methods*. For instance, constrained-based methods take into account different practical constraints like cost, revenue across different scenarios such as e-commerce [70] and travel package recommendation [33, 61]. Data-mining approaches leverage association rules [17] or probabilistic models [32] to discover bundle patterns. Preference elicitation methods focus on learning utility functions [16, 62] to capture user preferences across various features, while factorization-based approaches like LIRE [35], BBPR [41], and EFM-Joint [6] decompose user–item and user–bundle matrices to learn underlying preferences. Subsequently, *deep learning-based methods* show their stronger capabilities in bundle recommendation. Accordingly, different neural network architectures have been proposed. Specifically, sequence-based methods like BGN [3] and ComEmb [29] utilize LSTM to build bundles dynamically and then recommend these bundles to users. Attention-based approaches like DAM [9], AttList [23], CAR [24], and BRUCE [2] learn item affinity or user preference toward bundles using the attention mechanism. Graph-based methods, such as BundleNet [14] and BGCN [7], capture complex user–item–bundle relationships through graph convolutional networks. Recent advances include contrastive learning methods (e.g., MIDGN [69], HIDGN [71], CrossCBR [37], BundleGT [57], and MultiCBR [36]) for more effective representation learning and conversational approaches (e.g., BUNT [26]) that enable dynamic recommendation through multi-round interactions. Besides, CoHEAT [28] employs curriculum learning to dynamically balance user–bundle and user–item interactions based on bundle popularity to address the cold-start problem in bundle

recommendation. Despite their effectiveness, most of these methods directly regard co-purchased products [32, 70] or user-generated lists [7, 9, 16, 23, 24, 35] as bundles. However, many products are co-purchased with no common underlying intents, so there is no quality control to ensure bundle coherence. Besides, the user-generated lists are only available in limited domains, e.g., books and music. Although some studies utilize pre-defined bundles by retailers [3, 14, 17, 19, 33], the data size is limited due to the high cost of collecting.

2.2 Bundle Generation

The demand for high-quality bundles has driven significant research in bundle generation, evolving from simple pattern mining to modern multimodal approaches. Early studies [50] focus on mining co-occurrence patterns at the item category level. The field then advanced to include user preferences, marking a shift toward more personalized solutions. Specifically, POG [10] employs the Transformer architecture to capture item compatibility relationships based on the image and textual information of items and learn from user interaction history to generate personalized outfits that match user preferences. BYOB [13] treats bundle generation as a combinatorial optimization problem and adopt reinforcement learning techniques to improve personalization. Recent approaches have increasingly leveraged multimodal information and user feedback. For instance, CLHE [39] uses self-attention mechanisms to fuse multiple data modalities, and Conna [56] incorporates a contrastive non-autoregressive decoding strategy to improve both creative quality and generation efficiency. Cross-item relationships have been explored by CIRP [38], which integrates item semantics and relations into multimodal representations. DiFashion [63] utilizes the diffusion model to synthesize fashion item images that compose a visually compatible outfit. By incorporating a trainable fusion module, BundleLLM [34] aligns multimodal features in the LLM semantic space, allowing it to process multiple data formats and complete items in the partial bundles. However, these methods still face critical limitations: (1) most methods can only generate fixed-size bundles, lacking the flexibility to create bundles of varying sizes based on different scenarios, and (2) some approaches require a partial bundle as context, limiting their practical applications. A recent approach, AICL [48], leverages LLMs with ICL techniques to generate bundles of dynamic lengths while inferring user intent. Although this method addresses the aforementioned limitations, the utilization of LLMs brings new challenges related to computational resource demands and response latency.

2.3 LLMs for Recommendation

Recently, LLMs have been widely applied into RSs thanks to their superior reasoning capabilities and extensive knowledge [21, 49, 60, 68], which can be divided into two paradigms: *prompting-based* and *tuning-based methods*. In particular, prompting-based methods keep the original LLM parameters while designing effective demonstrations through ICL to improve performance on specific tasks. For example, KP4SR [67] transforms the structured knowledge graph into knowledge prompts for sequential recommendation. Other methods [12, 25] adopt LLMs for different recommendation tasks (e.g., ranking and conversation) by reformulating them into prompt formats. In contrast, the tuning-based methods utilize the parameter-efficient fine-tuning to update a small number of parameters of LLMs, injecting collaborative signals or other side information. These methods transform traditional user–item interaction data into textual prompts and then fine-tune LLMs to enhance performance across various downstream recommendation tasks, such as TallRec [4], GLRec [59], and Rella [31]. Although they have demonstrated the potential of LLMs in various recommendation scenarios, the integration of LLMs into bundle generation has been under-explored. To the best of our knowledge, AICL [48] is among the few methods that have applied LLMs to bundle generation, which utilizes retrieval-augmented generation and **Chain-of-Thought (CoT)** techniques to generate effective

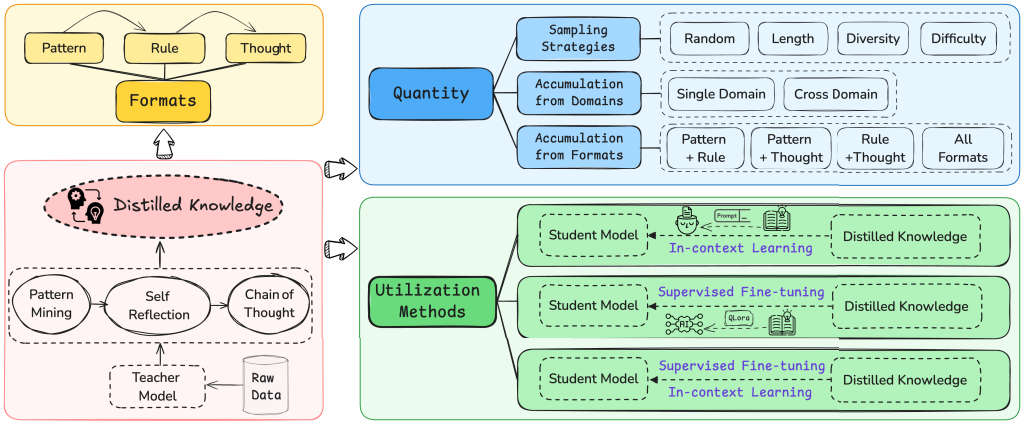


Fig. 2. The overview of our proposed KD framework.

demonstrations, enabling user intent inference and personalized bundle generation. However, this method faces challenges in computational efficiency due to its reliance on resource-intensive LLMs.

2.4 KD in Recommendation

KD is first proposed in [27], which trains a compact student model to mimic a larger teacher model's soft probability outputs on MNIST digit classification tasks. This approach aims to transfer knowledge from an ensemble or complex model into a more lightweight and efficient model. Following its success in computer vision, KD has been widely adopted into RSs in two distinct ways: *explicit* and *implicit* knowledge transfer. The former methods take the final outputs of the teacher model as supervisory labels, training student models to produce similar results. For example, SLIM [55] and RDRec [52] distill the capability of generating rationales behind users' behaviors from LLMs to smaller student models using CoT [53]. Similarly, DLLM2Rec [11] transfers ranking and collaborative knowledge from LLM-based teacher models to conventional sequential-based student recommendation models. Conversely, the latter methods enhance the feature representations of student models by aligning outputs between teacher and student models at hidden layers or output layers. For instance, LaMP [44] minimizes the KL divergence between the probability distribution of student model and a target distribution computed from the teacher model for personalized text generation. PRM-KD [46] distills knowledge from multiple teacher models to a student model by minimizing the KL divergence between probability distributions in scoring and ranking items, focusing on in-batch negative samples while dynamically weighing each teacher's contribution based on confidence and mutual agreement. Additionally, some methods, like NewsBERT [58], Tiny-Newsrec [65], and SSI [66], aim to align with not only the teacher models' probability distribution but also with the intermediate layers' representations. Inspired by the remarkable success of KD methods in various recommendation tasks, our study explores how to leverage KD to achieve both effective and efficient bundle generation using LLMs.[1]

3 The Proposed KD Framework

As illustrated in Figure 2, we propose a comprehensive KD framework to systematically investigate the impacts of the format, quantity, and utilization methods of distilled knowledge (corresponding to RQ1–RQ3) in optimizing lightweight LLMs for efficient yet effective bundle generation.

[1]We focus solely on explicit KD because most state-of-the-art LLMs (e.g., ChatGPT) are accessible only through APIs, which do not provide access to logits or intermediate representations. Additionally, open source LLMs require substantial computational resources to achieve comparable performance. Therefore, we do not consider implicit KD in our current work.

Table 1. Comparison of Existing Bundle Datasets with Relevant Attributes

Dataset	Title	Image	Category	Session
Steam	No	No	No	No
Netease	No	No	No	No
Youshu	No	No	No	No
Goodreads	No	No	No	No
iFashion	Yes	Yes	Yes (ID)	No
Our Datasets	Yes	Yes	Yes (Text)	Yes

Table 2. The Statistics of the Three Bundle Datasets

	Electronic	Clothing	Food
#Users	888	965	879
#Items	3,499	4,487	3,767
#Sessions	1,145	1,181	1,161
#Bundles	1,750	1,910	1,784
#Intents	1,537	1,590	1,323
#Categories	454	377	652
Average Bundle Size	3.52	3.31	3.58
#User–Item Interactions	6,165	6,326	6,395
#User–Bundle Interactions	1,753	1,912	1,785
Density of User–Item Interactions	0.20%	0.15%	0.19%
Density of User–Bundle Interactions	0.11%	0.10%	0.11%

3.1 Preliminaries

3.1.1 Bundle Definition. Let $\mathcal{I} = \{i_1, i_2, \ldots, i_{|\mathcal{I}|}\}$ denote the set of all items in the dataset, and $\mathcal{C} = \{c_1, c_2, \ldots, c_{|C|}\}$ denote the set of item categories. Each item i belongs to a specific category c. A bundle b is defined as a subset of items $b = \{i_1, i_2, \ldots, i_{|b|}\} \subseteq \mathcal{I}$, where $|b|$ denotes the bundle size, i.e., the number of items contained in the bundle. In our setting, bundles are required to contain at least two items, $|b| \geq 2$. Items within a bundle typically share a coherent relationship, such as being complementary (e.g., a camera and a lens) or alternative (e.g., different flavors of snacks), and satisfy a specific user intent (e.g., assembling a desktop).

3.1.2 Raw Data. We utilize three public bundle datasets released in the SIGIR 2022 resource paper [47, 50], covering three domains: Electronic, Clothing, and Food. These datasets are selected because: (1) they are derived from real-world user–item interactions within the Amazon datasets [22]. Additionally, to ensure data quality, both the bundles and their corresponding user intents were annotated through a carefully designed crowdsourcing task, as described in the SIGIR 2022 papers [47, 50]; (2) compared to other public bundle datasets, such as Steam, Netease, Youshu [2], Goodreads [24], and iFashion [42], our selected datasets provide rich side information, including user sessions (where the bundles are identified), well-labeled user intents, item titles, images, and categories, which can better support our study, as summarized in Table 1.

Note that a user session s is represented as a set of items: $s = \{i_1, i_2, \ldots, i_{|s|}\}$, collected from a specific interaction period [49]. Within a session, there may exist multiple bundles: $\mathcal{B}_s = \{b_1, b_2, \ldots, b_n\}$, where n represents the number of bundles under session s. Each bundle is associated with exactly one user intent, while a single intent can correspond to multiple bundles (e.g., different item combinations reflecting a similar need), so the numbers of bundles and intents are not necessarily identical. The statistics of the datasets are summarized in Table 2.

3.1.3 Problem Formulation. The bundle generation task is formulated as identifying potential bundles within user sessions. Formally, given a target session s containing a sequence of items, the goal is to generate a set of bundles $\hat{\mathcal{B}}_s = \{\hat{b}_1, \hat{b}_2, \ldots, \hat{b}_k\}$, where each predicted bundle $\hat{b}_j$ is a subset of the items in s and satisfies the size constraint $|\hat{b}_j| \geq 2$. The objective is to maximize the alignment between the predicted bundles $\hat{\mathcal{B}}_s$ and the ground truth bundles $\mathcal{B}_s$ in terms of both the number of bundles and the coverage of items within each bundle. This requires the model to learn how many bundles to generate for each session and how to determine the item composition of each bundle.

3.1.4 Motivation of Using KD. The adoption of KD is driven by the necessity to reconcile the tradeoffs between large-scale LLMs (Teacher LLMs) and their smaller counterparts (Student LLMs). Teacher LLMs possess exceptional reasoning capabilities and comprehensive knowledge, enabling them to effectively identify complex item relationships (e.g., complementary or alternative products)

and infer user intent. However, their massive parameterization results in prohibitive computational overhead and high inference latency. In our specific problem setting—where personalized bundles can be detected dynamically based on a user's ongoing session—this high latency makes Teacher LLMs impractical for real-time deployment. Conversely, Student LLMs are lightweight and efficient for rapid inference but typically fall short in effectiveness on high-quality bundle generation. To bridge this gap, we employ KD to transfer the teacher's capabilities to the student, offering three critical advantages. First, it achieves a balance between efficiency and effectiveness; KD enables the student model to approximate the teacher's reasoning while retaining the inference speed required for real-time, session-based bundle generation. Second, it facilitates private deployment. Unlike massive teacher LLMs that often require external API calls, a distilled student model allows for local deployment, ensuring that sensitive user session data are processed securely within the system. Third, it enables high specialization. While teacher models are general-purpose reasoners, KD allows us to fine-tune the student model specifically for the bundle generation task, focusing its capacity on domain-specific adaptation to outperform generalist models in niche scenarios.

3.2 Formats of Distilled Knowledge (RQ1)

To investigate the impact of different types of knowledge on bundle generation, we first introduce a progressive KD module that systematically extracts knowledge from the raw data in increasingly sophisticated forms, evolving from frequent patterns to formalized rules and deep thoughts.

3.2.1 Frequent Patterns. Frequent patterns represent the first level of our distilled knowledge, marking a significant advancement in abstraction from raw data. While raw data merely provide isolated information about individual items within bundles, frequent patterns uncover meaningful relationships that link these items within coherent bundles, transforming our understanding from merely "what items exist" to "what items belong together." Inspired by prior works [50], we attempt to identify frequent patterns at the item category level across different bundles by employing the Apriori algorithm [1]. Note that we do not use teacher LLMs to distill the frequent patterns due to the content length constraint. Specifically, we transform all bundles (e.g., $b_1 = \{i_1, i_2, i_3\}$) into their categorical representations (e.g., $b_1 = \{c_1, c_2, c_3\}$) and apply the Apriori algorithm to mine frequent patterns (i.e., frequently co-occurred categories) across different bundles (e.g., $p_1 = \{c_1, c_2\}$ or $p_2 = \{c_1, c_2, c_3\}$), where b, i, c, p denote a bundle, an item, an item category, and a pattern, respectively. These extracted frequent patterns thus serve as pre-processed insights and by explicitly providing such patterns to the student LLM, we enable it to recognize that specific category combinations (complementary or alternative items) consistently appear in high-quality bundles. In summary, the advantage of leveraging pattern knowledge is twofold: it reduces the cognitive load on LLMs by providing distilled category-level heuristics, and it acts as a structural guide for bundle generation. When prompted or fine-tuned with pattern knowledge, LLMs can more effectively generate coherent bundles by first selecting appropriate category combinations and then populating those categories with contextually relevant items in user sessions, resulting in bundles that better mirror real-world user preferences and behaviors. Table 3 demonstrates the examples of frequent patterns mined from the three domains.

3.2.2 Formalized Rules. Rules represent the second level of our distilled knowledge, transforming frequent patterns into explicit and actionable principles. While pattern knowledge identifies relationships between item categories, rule knowledge establishes clear conditions and guidelines for effective bundle generation. This shift advances understanding from simply recognizing "what items belong together" to reasoning about "why and when items should be bundled."

Table 3. Examples of Frequent Patterns Mined on the Three Domains

Domains	Electronic	Clothing	Food
Frequent Patterns	[Camera, Camera Batteries, Camera Lenses, Micro SD Cards]	[Sandals, Cover-Ups, Handbags, Hats and Caps]	[Baking Cups, Crackers, Peanut Butter, Toaster Pastries]
Corresponding Bundles			

To distill rule knowledge, we employ the self-reflection method [40] with teacher LLMs. For each session, we first design a prompt to ask LLMs to generate initial bundles as—*A bundle can be a set of alternative or complementary products that are purchased with a certain intent. Given the list of products with their descriptions: {product X: category, title}, identify bundles where: - Each bundle must contain at least two products; - Products must serve a common user intent; - Products must be either complementary (work together) or alternative (substitutes). Answer format in JSON: {'bundle number':['product number']}. Please do not provide any explanations for the results.* Then, we provide the ground truth bundles and ask LLMs to compare their predictions with the correct bundles, analyzing the reasons behind any discrepancies using the prompt: *Compare the correct bundles and your answers: Correct bundles: {correct bundles}. Your answers: {detected bundles}. Identify which detected bundles in your answers are incorrect and explain why they are incorrect based on the product categories and descriptions. Answer format in JSON: {'incorrect bundle number':['reason for incorrect detection']}*. Next, we ask LLMs to review the entire process, identifying the underlying causes of incorrect bundle predictions with the prompt: *Review your bundle detection process considering the following: 1. User Intent Analysis: How well did you identify the primary intent? 2. Product Relationships: The products within a bundle should have a certain relationship (e.g., alternatively or complementarily). 3. Bundle Logic: Each bundle should contain at least two products; Practical usage scenarios that reflect certain topics or user intents. Answer format in JSON: {'issue number':['specific aspect', 'detailed reasoning with example']}*. Finally, based on this analysis, we prompt LLMs to summarize explicit rules for correct bundle generation: *Based on your analysis of correct and incorrect bundles, formulate the rules that should be followed to improve the accuracy of bundle detection. Each rule should: 1. Define specific criteria for grouping (e.g., complementary relationships or alternative options). 2. Avoid common mistakes (e.g., unrelated products in a bundle). 3. Align with purchase intent. Answer format in JSON: {'rule number':['rule description']}*. Unlike patterns that merely reflect statistical co-occurrence, rules capture causal relationships and conditional logic, which explain why certain item combinations succeed while others fail, thus enabling LLMs to make more natural bundling decisions for more principled and effective bundle generation. Table 4 showcases the examples of rules derived from the three domains.

3.2.3 Deep Thoughts. Thoughts are the highest level of our distilled knowledge, which introduce a more flexible, context-sensitive approach to bundle generation. While rule knowledge provides guidelines and principles for bundle generation, thought knowledge enables dynamic reasoning that adapts to specific scenarios. This transition shifts the focus from "why items should be bundled" to "how to optimize bundle generation for unique situations."

To distill thought knowledge, we utilize CoT [53] reasoning with teacher LLMs. For each session, given its item information (e.g., title, category), all possible bundles, and the corresponding user intents, we prompt LLMs to explain the reasons why a particular bundle is formed, following a complete reasoning path—from meta-information (categories) to user intent and bundle composition.

Table 4. Examples of Formalized Rules Derived from the Three Domains

Domain	Rules
Electronic	1. Group products together if they are complementary and serve a common user intent, such as protective cases and batteries for electronic devices. 2. Aim to create bundles with at least two products that reflect practical usage scenarios aligned with specific topics or user intents.
Clothing	1. Avoid combining products in a bundle if they are unrelated in terms of user intent or practical usage scenarios (e.g., Belts and Earmuffs) 2. Group products based on shared use or user intent, such as creating a coordinated jewelry accessories set for a specific occasion.
Food	1. Avoid grouping products that are unrelated or do not align with a common user intent, such as mixing nutrition bars with mints. 2. Ensure that products in a bundle belong to related or overlapping categories to avoid mismatched pairings like coffee capsules with coffee pod holders.

— *You are provided with: 1. A product list containing products with their categories: {'product id':{'title': '$title', 'category': '$category'}} 2. A list of bundles containing groups of products commonly purchased together with the user intents, detected from the product list: {'bundle id':{'group': ['$product id'], 'intent': '$intent'}}.*
— *Your Task: The products in a bundle are commonly purchased together based on a specific user intent, and all the bundles are detected from the product list. Generate natural language insights for each bundle explaining why certain product categories are bundled together based on user intent.*
— *Output Format: {'bundle1': 'Customers buying [category a] and [category b] are typically looking to [intent]', 'bundle2': 'The combination of [category x] and [category y] suggests [intent]'.}*
— *Example Output: {'bundle1': 'Gaming enthusiasts purchase GPU, CPU, and Motherboard together for building high-performance gaming setups.', 'bundle2': 'Customers combining Office Chairs and Monitors are focused on creating an ergonomic and efficient workspace.'}*
— *Requirements: 1. Insightful Explanations: Derive meaningful connections between product categories and user intents. 2. Clarity and Natural Tone: Write easy-to-understand, conversational explanations. 3. Category-Intent Pairing: Each explanation must include references to product categories and intents. 4. Deliver only JSON output with explanations. 5. Avoid unnecessary details or supplementary text.*

This step-by-step process articulates the complex considerations behind effective bundling, capturing subtle nuances often missed by simpler knowledge types. Thought knowledge equips LLMs to move beyond fixed patterns and rules, integrating contextual intelligence into bundle generation. It enables student LLMs to reason dynamically, evaluate multiple factors concurrently, and adapt their strategies based on specific user needs and contexts, leading to more intuitive and effective bundling decisions. Examples of derived thoughts from the three domains are shown in Table 5.

3.2.4 Differences and Connections among Knowledge Types. Figure 3 illustrates how each format of knowledge is generated. Here, we analyze their differences and intrinsic connections to justify the necessity of extracting them.

Differences. The distinction among the three types of knowledge lies in their level of abstraction and reasoning depth. Frequent patterns represent the most fundamental level, capturing statistical co-occurrences of item categories based on historical data. They simply reflect what items typically

Table 5. Examples of Deep Thoughts Derived on the Three Domains

Domain	Thoughts
Electronic	1. Customers purchasing *Screen Protectors* and *Folio Cases* together are likely looking to protect and accessorize their MacBook Pro 13.3 with Retina Display. 2. Customers buying *Film, Camera Batteries*, and *Skylight and UV Filters* are typically looking to enhance their photography equipment with essential accessories for better image quality.
Clothing	1. Customers purchasing *Hats* and *Winter Accessories* are likely preparing for cold weather and looking to stay warm and stylish. 2. The combination of *Dresses, Pumps*, and *Costumes* suggests customers are preparing for various special events or themed parties.
Food	1. Customers buying *Cakes* and *Gluten Free* products are likely looking for baking options. 2. Customers buying *Fruit Snacks* and *Oatmeal* together are likely looking for convenient and healthy snack options for on-the-go consumption.

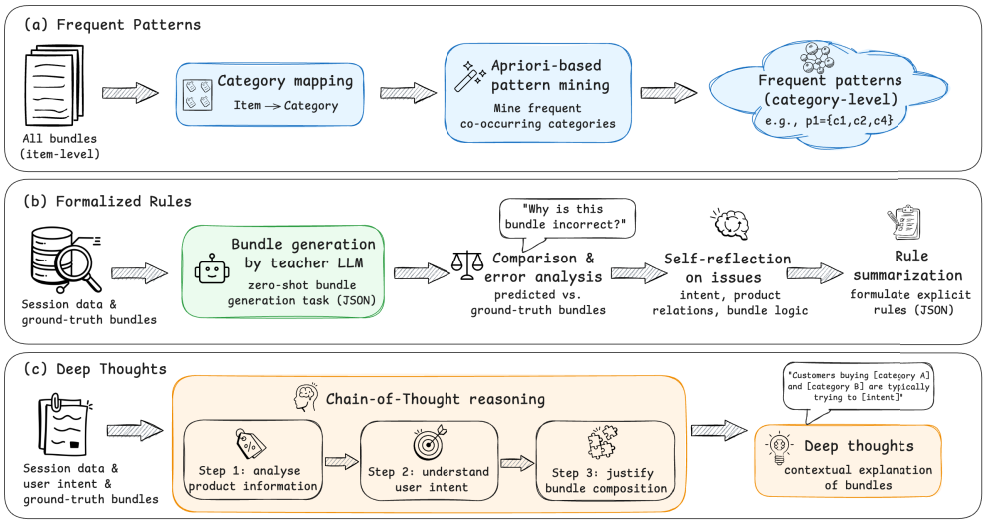


Fig. 3. Generation pipeline of three types of knowledge (Frequent Patterns, Formalized Rules, and Deep Thoughts).

appear together without explaining the underlying rationale. Formalized rules elevate this by establishing static, general principles derived from the domain. They provide the validity constraints answering why items generally belong together. Finally, deep thoughts represent dynamic, session-specific reasoning. They synthesize specific product semantics within a unique session to infer a concrete user intent, answering how these specific items fit the current user context.

Connections. These knowledge types are not redundant but rather form a progressive hierarchy that mirrors human cognitive processes in recommendation. Frequent patterns provide the data-driven foundation by identifying statistical correlations. Formalized rules abstract these statistics into logical principles, filtering out noise and establishing criteria for valid combinations. Deep thoughts apply these principles through a reasoning process to handle complex or novel scenarios that rigid rules cannot cover. Thus, utilizing all three formats provides a complementary framework:

patterns provide the skeleton, rules provide the logical constraints, and thoughts provide the contextual reasoning required for high-quality bundle generation.

3.3 Quantity of Distilled Knowledge (RQ2)

To explore how the quantity of distilled knowledge impacts bundle generation performance, we design four distinct sampling strategies (i.e., random-, length-, diversity-, and difficulty-based) to select specific portions of the entire raw data. These strategies aim to capture varying knowledge quantities while ensuring representative data coverage for effective and efficient KD. Note that this process assumes that varying data portions will yield different amounts of distilled knowledge.

3.3.1 Different Sampling Strategies. *Random-based strategy* randomly samples a portion of raw data to distill different formats of knowledge. *Length-based strategy* first divides the entire raw data into three groups based on session length, with defined ranges of [2–4], [5–7], and [8–10], and then samples a portion of raw data from each group. The rationale behind this is that sessions of varying lengths may capture different levels of user interaction complexity and encompass diverse numbers of bundles. *Diversity-based strategy* first divides the entire raw data into three groups (i.e., low, medium, and high) based on session diversity levels, which are measured by the ratio of the number of categories to the number of items within each session. It then samples a portion of raw data from each group. *Difficulty-based strategy* first divides the entire raw data into three groups (i.e., easy, medium, and hard) based on the difficulty levels for bundle generation. Specifically, we leverage teacher LLMs with the zero-shot setting to perform the bundle generation task (i.e., identify bundles from each session) and define the difficulty based on the accuracy of the identified bundles. Then, it samples a portion of raw data from each group. To investigate the impact of varying amounts of distilled knowledge, we apply these sampling strategies with different sample ratios, specifically in the range of {10%, 30%, 50%, 70%}.

Assumption Verification. As emphasized, our approach is based on the assumption that varying the data portions will result in different amounts of distilled knowledge. For verification, we apply the four sampling strategies with sample ratios in the range of {10%, 30%, 50%, 70%} on the raw data. Then, we distill knowledge from the sampled data and calculate the amounts of the distilled knowledge. Figure 4 shows the results, where the x-axis represents the sampling ratio and the y-axis means the amount of distilled knowledge. Specifically, for pattern knowledge, we merge the same patterns (e.g., $[c_1, c_2]$ and $[c_2, c_1]$) to eliminate duplicates. For rule and thought knowledge expressed as natural language, we use a pre-trained model (i.e., BERT) to compute the semantic similarity and filter out redundant rules or thoughts based on a similarity threshold of 0.8. The results show that as the sample ratios increase, the amount of distilled knowledge also increases across the four strategies and three domains, validating our assumption.

3.3.2 Accumulation of Knowledge from Different Domains and Formats. In addition to employing different sampling strategies to vary the quantity of distilled knowledge, we explore two additional approaches. The first method focuses on accumulating a single type of knowledge from multiple domains, enabling a comparison between using a specific type from a single domain (e.g., rules from Electronic) and combining such type of knowledge from multiple domains (e.g., rules from Electronic + Clothing + Food). The second method involves accumulating diverse types of knowledge, allowing us to compare the performance of using a single knowledge type (e.g., frequent patterns) with that of multiple types (e.g., frequent patterns + formalized rules + deep thoughts). In summary, the first method incorporates cross-domain knowledge to enhance the student LLM's generalizability, whereas the second method integrates diverse and rich knowledge from a single domain into the student LLM, enabling it to acquire more comprehensive and in-depth domain-specific knowledge.

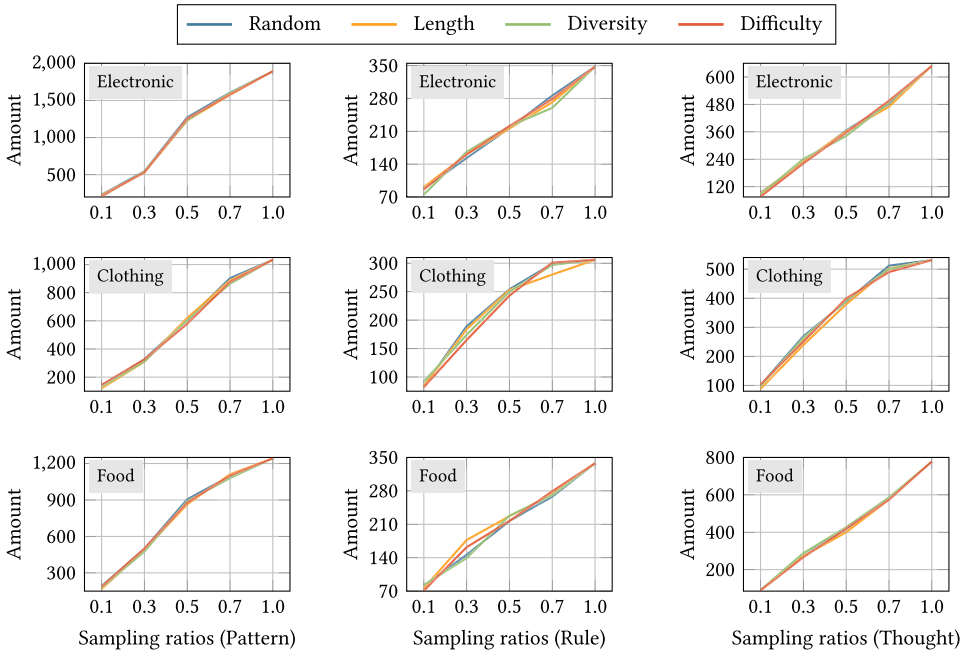


Fig. 4. Variation in knowledge quantity as sampling ratio increases (Pattern: 1st Column; Rule: 2nd Column; Thought: 3rd Column).

3.4 Utilization Methods of Distilled Knowledge (RQ3)

Given the distilled knowledge in various formats and quantities, we further explore the impacts of different methods for utilizing distilled knowledge on bundle generation performance. To this end, we consider two complementary LLM adaptation techniques: ICL and SFT. Using these techniques, we examine three distinct utilization methods on the student LLMs: (1) incorporating distilled knowledge into ICL for inference, (2) integrating distilled knowledge through SFT for training, and (3) injecting distilled knowledge into both SFT for training and ICL for inference simultaneously.

3.4.1 ICL. We propose various knowledge retrieval methods based on the features of the target session and incorporate the retrieved knowledge into the prompt as context to enhance the student LLM's task comprehension. For pattern knowledge, we employ a fine-grained category matching strategy. Given a session, we retrieve all patterns that form a subset of the categories involved in the target session. The retrieved patterns are then integrated into the prompt as context. For rules and thoughts, we utilize a pre-trained language model (i.e., BERT) to generate semantic embeddings to identify relevant contexts. Specifically, we first obtain the embedding of each item's title by performing mean pooling on its word embeddings. Then, we average the embeddings of all item titles within a session to derive the final session embedding. Next, we retrieve the session most similar to the target session based on the cosine similarity of their session embeddings. The rules and thoughts associated with the retrieved session are incorporated into the prompt as context. The final prompt combines the target session, its product information, and the retrieved knowledge as context, given below:

—You will be given a list of products, each with a category and description. You will also receive a single piece of overarching guidance in the variable *KNOWLEDGE*.

—Your task is to: Analyze the products and identify relationships between them, referencing *KNOWLEDGE* to determine their connections. Group the products into bundles based on these criteria: (1) Each bundle must contain at least two products; (2) Products in a bundle should be either alternative options for the same purpose or complementary items typically bought together; (3) The products should fit a specific customer intent or use case indicated by *KNOWLEDGE*. Present the bundles in JSON format as follows: {'bundle1': ['product1', 'product2', ...], 'bundle2': ['product3', 'product4', ...], ...}
—Important notes: (1) Only provide the JSON output; (2) Do not include any explanations or additional text; (3) Use 'bundle1', 'bundle2', etc. as keys in the JSON; (4) Use the product IDs as values in the arrays; (5) Leverage the guidance from *KNOWLEDGE* to remain consistent with the bundling logic.

3.4.2 SFT. This method enables student LLMs to acquire task-specific knowledge, thereby mitigating the hallucination problem. To implement this method, we construct training data using the same textual prompt in Section 3.4.1 (i.e., including session information and retrieved knowledge) as input, and the ground truth bundles from the session as output. To further enhance the robustness of student LLMs, we introduce a permutation augmentation strategy during training data construction. Since a session may involve multiple bundles, this strategy perturbs the bundle order, allowing the student LLM to learn order-invariant properties among bundles. Specifically, for each training sample, we generate all possible bundle permutations to form an equivalent sample set. For example, given a session s_1, if the ground truth bundles (label) are $\{b_1 : [i_1, i_2], b_2 : [i_3, i_4]\}$, applying our permutation strategy will also include $\{b_2 : [i_3, i_4], b_1 : [i_1, i_2]\}$.

4 Experiments, Results, and Analysis

4.1 Experimental Settings

Datasets. We conduct experiments with three public bundle datasets, as introduced in Section 3.1.2. Specifically, we split all session data into training, validation, and test sets with a ratio of 7:1:2.

Teacher and Student Models. We use GPT-3.5-turbo and GPT-4.1 as the teacher LLMs to investigate the impact of different teacher capabilities. For the student LLMs, we employ Llama3.1-8B and Llama3.2-3B to examine the influence of student model capacity. Based on this setup, we construct four teacher–student pairs (including GPT-3.5-turbo+Llama3.1-8B, GPT-3.5-turbo+Llama3.2-3B, GPT-4.1+Llama3.1-8B, and GPT-4.1+Llama3.2-3B) by integrating distilled knowledge from the teacher LLMs through ICL for inference and SFT for training. To demonstrate the *effectiveness and efficiency* of our student models, we compare them against models from two categories: conventional bundle generation methods and LLM-based methods. For conventional methods, **Freq** [50] employs the Apriori algorithm to identify frequent patterns at the item category level. **BBPR** [41] is a greedy algorithm that dynamically generates bundles by computing the similarity between users, items, and bundles representation learned by BPRMF [43]. **POG** [10] utilizes an encoder–decoder model based on Transformer architecture to generate personalized outfits by using multi-modal data. For LLM-based methods, **Zero-shot** method prompts the teacher LLMs (GPT-3.5-turbo and GPT-4.1) to generate bundles directly. **AICL** [48] enhances Zero-shot by incorporating dynamic demonstration generation with retrieval augmentation to infer user intents and generate bundles simultaneously, establishing it as the state-of-the-art method. Specifically, comparisons with conventional methods aim to validate the superior effectiveness of our student models in generating high-quality bundles, while comparisons with LLM-based methods aim to assess whether they can deliver comparable performance with significantly enhanced efficiency.

Evaluation Metrics. We adopt the same evaluation metrics as in prior work [47, 50] to evaluate the quality of generated bundles. Specifically, at the session level, we adopt *Precision* and *Recall* to

Table 6. Summary of Experimental Figures and Tables Mapping to RQs and Analysis Sections

Figure/Table	Description	Related RQ(s)	Analysis Location
Tables 6 and 7	Formats of distilled knowledge	RQ1	Section 4.2
Figures 5–12	Impact of sampling ratios/strategies	RQ2	Section 4.3.1
Figures 13–14	Accumulating distilled knowledge from different domains	RQ2	Section 4.3.2
Figures 15–18	Accumulating distilled knowledge from different formats	RQ2	Section 4.3.3
Figures 19–20	Utilization methods of distilled knowledge	RQ3	Section 4.4
Tables 8–10	Effectiveness analysis of conventional and LLM-based models	RQ1–RQ3	Section 4.5.1
Table 11	Efficiency analysis of teacher and student models	-	Section 4.5.2

measure the quantity of correctly predicted bundles within each session. *Precision* indicates how many of the generated bundles are correct, while *Recall* measures how many of the ground truth bundles are identified. Note that we treat a generated bundle as a hit bundle if it either completely matches or is a subset of a ground truth bundle. At the bundle level, we use *Coverage* to evaluate the item-wise accuracy of hit bundles by calculating the average proportion of correctly predicted items within the ground truth bundles.

Implement Details. For conventional bundle generation methods, we adopt the same parameter configurations suggested in [48]. For LLM-based methods, we leverage the OpenAI API with GPT-3.5-turbo and GPT-4.1 to implement Zero-shot and to distill domain knowledge (rule and thought), while GPT-3.5-turbo is specifically employed for AICL. Our student models, Llama3.2-3B-SFT and Llama3.1-8B-SFT, are fine-tuned with QLoRA [15] based on the framework Unsloth.[2] We conduct systematic hyperparameter tuning via grid search for learning rates in the range of {2e-5, 8e-5, 2e-4}, epochs in the range of {3, 4, 5}, QLoRA rank in the range of {8, 16, 32}, and QLoRA alpha values in the range of {8, 16, 32}. All experiments are conducted on 4× NVIDIA A40 GPUs with 48GB memory, with a batch size of 4 and gradient accumulation steps of 4.[3]

Roadmap of Experiments. Since Sections 4.2–4.5 present the experimental results and analysis for three RQs, we provide a summary of how each figure and table is linked to the corresponding RQ and analysis subsection in Table 6. This summary is intended to facilitate navigation in the subsequent result and analysis sections.

4.2 Investigation on Formats of Distilled Knowledge (RQ1)

This section explores how different formats of distilled knowledge impact the bundle generation performance of student models, directly addressing RQ1. We present the experimental results in Tables 7 and 8, where the best performance achieved by each student model is in bold; and "Improve" (last row) indicates the relative improvements achieved by the best student model (marked by "*") compared with the teacher model.[4] For ease of analysis, we first analyze the results on the GPT-3.5-turbo+Llama3.1-8B pair and then discuss the rest teacher–student pairs to check whether the observed trends generalize and remain consistent across all of them.

4.2.1 Impact of Knowledge Formats on the GPT-3.5-Turbo+Llama3.1-8B Pair. We first focus on the GPT-3.5-turbo+Llama3.1-8B pair in Table 7 and analyze how different knowledge formats influence the ICL and SFT variants of the same student. Several observations can be noted.

First, *incorporating explicit knowledge leads to performance improvements to some extent. However, the most effective type of knowledge varies between student models with ICL and SFT.* Specifically, for

[2] https://unsloth.ai/.

[3] Our code is available at https://github.com/KaiDF/KD4BG.

[4] Unless otherwise specified, each type of knowledge (e.g., Pattern) used in the reported results refers to the accumulated knowledge from all three domains (e.g., Pattern mined from the three domains), ensuring a fair and consistent comparison.

Table 7. Performance Comparison Regarding Different Formats of Distilled Knowledge with GPT-3.5-Turbo as the Teacher LLM (RQ1, Section 4.2)

Method	Knowledge	Electronic			Clothing			Food		
		Precision	Recall	Coverage	Precision	Recall	Coverage	Precision	Recall	Coverage
Zero-shot (GPT-3.5-turbo)		0.580	0.820	0.720	0.603	0.752	0.788	0.604	0.815	0.748
Llama3.1-8B -ICL	Raw Data	0.564	0.582	0.685	0.582	**0.644***	0.655	0.571	0.591	0.633
	Pattern	0.574	0.514	0.636	0.577	0.529	0.717	0.602	0.533	0.656
	Rule	**0.611**	**0.615**	**0.693**	**0.621***	0.633	**0.768**	**0.621**	**0.601**	**0.773**
	Thought	0.585	0.529	0.685	0.603	0.611	0.757	0.608	0.589	0.751
Llama3.1-8B -SFT	Raw Data	0.618	0.621	0.857	0.608	0.607	0.901	0.615	0.597	0.841
	Pattern	0.626	0.615	0.818	0.611	**0.623**	0.860	**0.668***	**0.649***	0.842
	Rule	0.610	0.594	**0.858***	0.609	0.598	**0.915***	0.623	0.600	**0.862***
	Thought	**0.633***	**0.644***	0.725	**0.614**	0.621	0.828	0.635	0.640	0.748
Llama3.2-3B -ICL	Raw Data	0.458	0.567	0.578	0.525	0.603	0.697	0.547	0.612	0.616
	Pattern	0.478	0.503	0.634	**0.601**	**0.614**	**0.749**	**0.562**	0.600	**0.673**
	Rule	**0.510**	**0.615**	0.636	0.511	0.606	0.693	0.535	**0.625**	0.617
	Thought	0.538	0.549	**0.648**	0.557	0.551	0.722	0.560	0.571	0.671
Llama3.2-3B -SFT	Raw Data	0.583	0.582	**0.858**	**0.611**	0.605	**0.902**	0.560	0.571	**0.855**
	Pattern	**0.586**	0.583	0.826	0.601	0.594	0.880	**0.652**	0.632	0.824
	Rule	0.541	0.513	0.850	0.562	0.527	0.896	0.551	0.533	0.844
	Thought	0.561	**0.614**	0.705	0.557	**0.617**	0.838	0.569	**0.645**	0.730
Improve		9.14%	−21.5%	19.17%	2.99%	−14.36%	16.12%	10.60%	−20.37%	15.24%

The gray-shaded row denotes the zero-shot teacher LLM baseline; bold indicates the best performance achieved by each student model; and * marks the best student result used to compute "Improve".

Table 8. Performance Comparison Regarding Different Formats of Distilled Knowledge with GPT-4.1 as the Teacher LLM (RQ1, Section 4.2)

Method	Knowledge	Electronic			Clothing			Food		
		Precision	Recall	Coverage	Precision	Recall	Coverage	Precision	Recall	Coverage
Zero-shot (GPT-4.1)		0.644	0.856	0.719	0.609	0.809	0.831	0.652	0.820	0.752
Llama3.1-8B -ICL	Raw Data	0.564	0.582	0.685	0.582	**0.644**	0.655	0.571	0.591	0.633
	Pattern	0.574	0.514	0.636	0.577	0.529	0.717	0.602	0.533	0.656
	Rule	**0.628**	**0.619**	0.649	**0.617**	0.610	0.736	**0.611**	**0.622**	0.660
	Thought	0.602	0.547	**0.711**	0.593	0.598	**0.737**	0.604	0.592	**0.710**
Llama3.1-8B -SFT	Raw Data	0.618	0.621	**0.857**	0.608	0.607	0.901	0.615	0.597	0.841
	Pattern	0.626	0.615	0.818	0.611	0.623	0.860	**0.668***	**0.649***	**0.842**
	Rule	0.582	**0.649***	0.815	0.595	0.647	**0.906***	0.618	0.648	0.835
	Thought	**0.631***	0.631	0.775	**0.641***	**0.677***	0.804	0.624	0.629	0.800
Llama3.2-3B -ICL	Raw Data	0.458	**0.567**	0.578	0.525	0.603	0.697	0.547	**0.612**	0.616
	Pattern	0.478	0.503	0.634	**0.601**	**0.614**	**0.749**	**0.562**	0.600	**0.671**
	Rule	0.444	0.545	0.571	0.473	0.572	0.655	0.493	0.580	0.600
	Thought	**0.532**	0.550	**0.644**	0.581	0.562	0.731	0.554	0.568	0.667
Llama3.2-3B -SFT	Raw Data	0.583	0.582	**0.858***	0.611	**0.605**	**0.902**	0.560	0.571	**0.855***
	Pattern	0.586	0.583	0.826	0.601	0.594	0.880	**0.652**	**0.632**	0.824
	Rule	0.542	**0.620**	0.786	0.577	0.598	0.878	0.565	0.606	0.829
	Thought	**0.588**	0.543	0.770	**0.618**	0.556	0.800	0.615	0.563	0.750
Improve		−2.02%	−24.18%	19.33%	5.25%	−16.32%	9.03%	2.45%	−20.85%	13.69%

the gray-shaded row denotes the zero-shot teacher LLM baseline; bold indicates the best performance achieved by each student model; and * marks the best student result used to compute "Improve".

Llama3.1-8B-ICL, compared to using raw data (no explicit knowledge), Rule almost achieves the best performance across all domains and metrics, while Pattern and Thought yield better results in some cases. This suggests that rule knowledge is more acceptable for student LLM with ICL due to its ease of understanding and versatility. As for pattern knowledge, while it identifies frequent patterns in bundles, the category-level patterns tend to make the student LLM focus too much on

relationships between categories while overlooking fine-grained item information. For example, if a session includes two cases for different tablet types and one screen protector, the LLM might incorrectly pair the protector with the wrong case, resulting in an inaccurate bundle. Although thought knowledge illustrates the reasoning process, in the ICL setting, relevant thoughts are retrieved from the training data based on semantic similarity to the target session, which may not perfectly align with the current session, thus leading to a performance drop. For Llama3.1-8B-SFT, Thought achieves the highest Precision in the Electronic and Clothing domains, while Pattern yields the best Recall in both the Clothing and Food domains. In contrast, Rule demonstrates superior performance only in terms of Coverage. These results may be attributed to the following reasons: (1) injecting thought knowledge enhances the student LLM's reasoning ability, thus identifying more precise bundles from the target session; and (2) injecting pattern knowledge helps the student LLM better identify and capture global trends, thereby improving Recall.

Second, *SFT outperforms ICL in most cases across all knowledge formats and metrics in the three domains.* This highlights the advantage of SFT, which explicitly trains the student model using distilled knowledge, enabling it to better internalize the task in the latent space and generate more accurate bundles. In contrast, ICL operates solely at inference time, leveraging distilled knowledge presented in the prompt without updating the model's internal parameters. While ICL can dynamically retrieve relevant context, it lacks the capacity to fully internalize the task structure. Consequently, SFT's ability to align the model's internal representations with the bundle generation task results in more robust and reliable performance compared to the purely prompt-based approach of ICL. Interestingly, in the Clothing domain, Llama3.1-8B-ICL surpasses its Llama3.1-8B-SFT on Precision and Recall, suggesting that with the right type of knowledge, ICL can still offer promising performance.

Last, *by integrating different formats of knowledge with various utilization methods, the student model can surpass the teacher model in terms of both Precision and Coverage; however, there remains a substantial gap in Recall (see row "Improve").* This suggests that the student model fails to generate many of the relevant bundles that the teacher model identifies. The possible reasons are two-fold. (1) **Limited coverage of distilled knowledge**: The distilled knowledge often captures only the most common or easily formalizable aspects of bundle generation, potentially missing rare or complex cases that the teacher model is capable of recognizing. Consequently, guiding or training the student model with such generalized knowledge tends to improve the quality of typical bundles, resulting in higher Precision and Coverage, but fails to account for less frequent cases, thereby lowering Recall. (2) **Limited expressiveness of distilled knowledge**: When the teacher's knowledge is distilled into explicit formats such as patterns, rules, or thoughts, the student model receives only high-level summaries or derivations rather than the full spectrum of the teacher's latent understanding. The teacher model's performance may hinge on complex and implicit reasoning paths that are difficult to capture through structured representations, which in turn limits the student model's ability to recover all relevant bundles, further contributing to lower Recall.

4.2.2 Impact of Knowledge Formats on Other Teacher–Student Pairs. We now extend the analysis to the remaining pairs in Tables 7 and 8, including GPT-3.5-turbo+Llama3.2-3B, GPT-4.1+Llama3.2-3B, and GPT-4.1+Llama3.1-8B. Several insights are observed:

(1) *For ICL, Rule is the most reliable format for larger students, while smaller students show mixed preferences.* Under both GPT-3.5-turbo and GPT-4.1 as teachers, Llama3.1-8B-ICL with Rule tends to achieve the best or near-best performance across domains and is especially strong in terms of Coverage. This confirms the observation in previous part that compact rule knowledge is easy for the 8B student to follow at inference time. For Llama3.2-3B-ICL, however, there is no single universally best format. Rule still provides competitive performance and often leads to the highest Coverage, but Pattern or Thought can outperform

Rule on Precision or Recall in some domains and metrics. This indicates that limited-capacity students may benefit more from simpler co-occurrence patterns or shorter reasoning snippets, whereas long or complex rules can exceed their representational capacity and introduce noise. Overall, Rule is a strong and robust choice for ICL on larger students, while for smaller students the optimal format is more task- and metric-dependent.

(2) *The superiority of SFT over ICL is consistent across models.* Across all teacher–student pairs, SFT-based students consistently outperform their ICL counterparts for the same knowledge format and student size. This holds for both GPT-3.5-turbo and GPT-4.1 as teachers and for both Llama3.2-3B and Llama3.1-8B as students. The performance gaps between formats also become smaller under SFT than under ICL, suggesting that the fine-tuning process helps the student smooth out noisy or overly specific knowledge and retain the most useful patterns.

(3) *Strong teachers do not necessarily yield strong students.* Across Tables 7 and 8, we observe that a stronger teacher (GPT-4.1) does not automatically translate into a stronger distilled student. Although GPT-4.1 clearly outperforms GPT-3.5-turbo in zero-shot bundle generation, the gains of the distilled students are sometimes smaller or even negative in Precision and Recall, especially for some ICL configurations. A possible explanation is that GPT-4.1 generates more complex and fine-grained patterns, rules, and reasoning traces, which may exceed the representational capacity of smaller students (particularly Llama3.2-3B) and introduce additional noise when distilled in limited quantities. At the same time, student capacity plays a crucial role. For the same teacher and knowledge format, Llama3.1-8B generally outperforms Llama3.2-3B, confirming that larger students can better internalize and exploit the distilled knowledge. However, a smaller student with SFT on well-chosen formats can sometimes outperform a larger student with ICL: for example, in Table 8, Llama3.2-3B-SFT with Pattern in the Food domain achieves Precision and Recall comparable to or higher than the Llama3.1-8B-ICL variants. This again demonstrates that what and how the knowledge is injected (format, SFT or ICL) can be as important as, or even more important than, simply increasing model size.

Overall, the core answer to RQ1 is:

> *Answer to RQ1:* Different formats of distilled knowledge consistently improve bundle generation, with SFT yielding stronger and more stable gains than ICL and enabling student models to sometimes surpass their teachers in Precision and Coverage. Larger students benefit more from distilled knowledge, but a stronger teacher does not always yield a better student, as performance also depends on knowledge format and teacher–student compatibility.

4.3 Investigation on Quantity of Distilled Knowledge (RQ2)

This section conducts extensive experiments, aiming to address RQ2: To what extent does the quantity of distilled knowledge influence bundle generation performance? To explore this, we employ the three approaches introduced in Section 3.3 to obtain varying amounts of distilled knowledge. In particular, we first apply different data sampling strategies (i.e., random-based, length-based, diversity-based, and difficulty-based) with different sample ratios to capture varying distilled knowledge quantities. Second, we aggregate the same type of knowledge from a single domain to multiple domains and evaluate the corresponding performance (e.g., Pattern from one domain vs. Pattern mined from three domains). Third, we accumulate knowledge in different

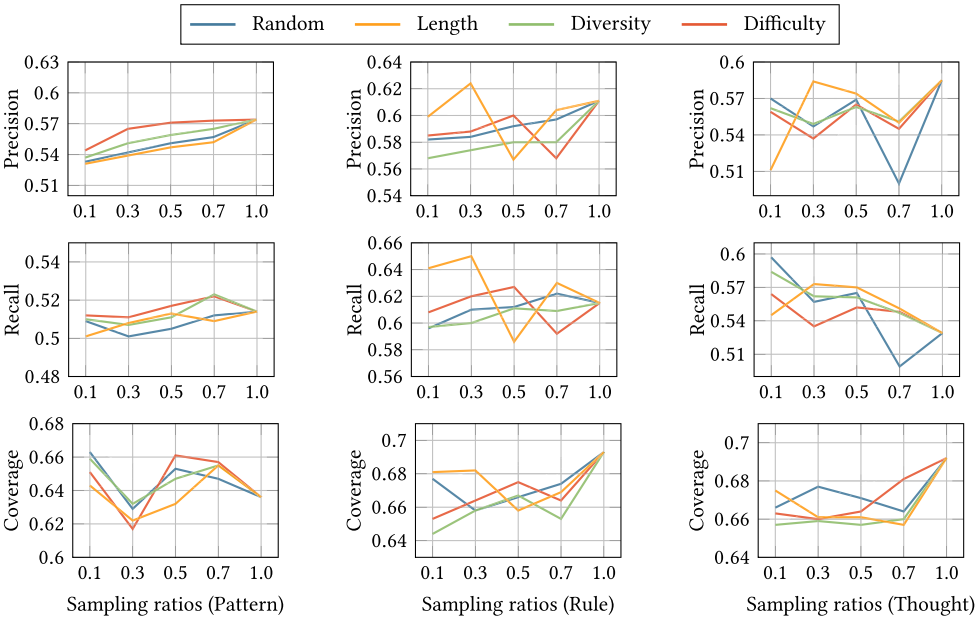


Fig. 5. Performance comparison of Llama3.1-8B-ICL with GPT-3.5-turbo as the teacher under different sampling strategies and ratios in the Electronic domain (RQ2, Section 4.3).

formats and compare the performance of using a single format versus combining multiple formats (e.g., Pattern vs. Pattern + Rule).

4.3.1 Impact of Sampling Strategies on the GPT-3.5-Turbo+Llama3.1-8B Pair. Figures 5–7 present the results on Llama3.1-8B-ICL with GPT-3.5-turbo as the teacher across the three domains on the three metrics. Some observations can be found. (1) *More is better.* In most cases, the performance consistently scales with the sampling ratio across different sampling strategies. Higher ratio (e.g., 0.7 or 1.0) usually yield better results than lower ratios. This indicates that ICL's effectiveness relies heavily on access to a large volume of knowledge. A richer knowledge pool provides more relevant examples for the retrieval mechanism to construct effective prompts. (2) *The impact of different sampling strategies is limited.* The performance gap between different sampling strategies at the same ratio is relatively minor, suggesting that the quantity of retrievable data is more critical than the specific method used to select it. However, a counter-intuitive observation arises when examining Recall with Rule and Thought: increasing the sample ratio does not necessarily lead to improved Recall in the Electronic and Food domains. A possible explanation is the high bundle diversity in these domains compared to Clothing. For example, Electronics bundles can range from computer-related items to camera gear, while Food bundles vary from snacks to full meals. In contrast, the Clothing domain exhibits more homogeneity, making it easier for fine-grained knowledge (i.e., Rule and Thought) to help the student model group similar items effectively. In the case of Electronics and Food, this detailed knowledge may cause the model to overfit to specific bundle types, narrowing its focus and reducing its ability to generate diverse valid combinations. As a result, the total number of generated bundles decreases, leading to lower overall Recall. Besides, for Pattern knowledge, its trend in Coverage differs from other types of knowledge. This is mainly because frequent patterns typically involve only two to four categories, which encourages the student model to generate shorter bundles, ultimately lowering Coverage.

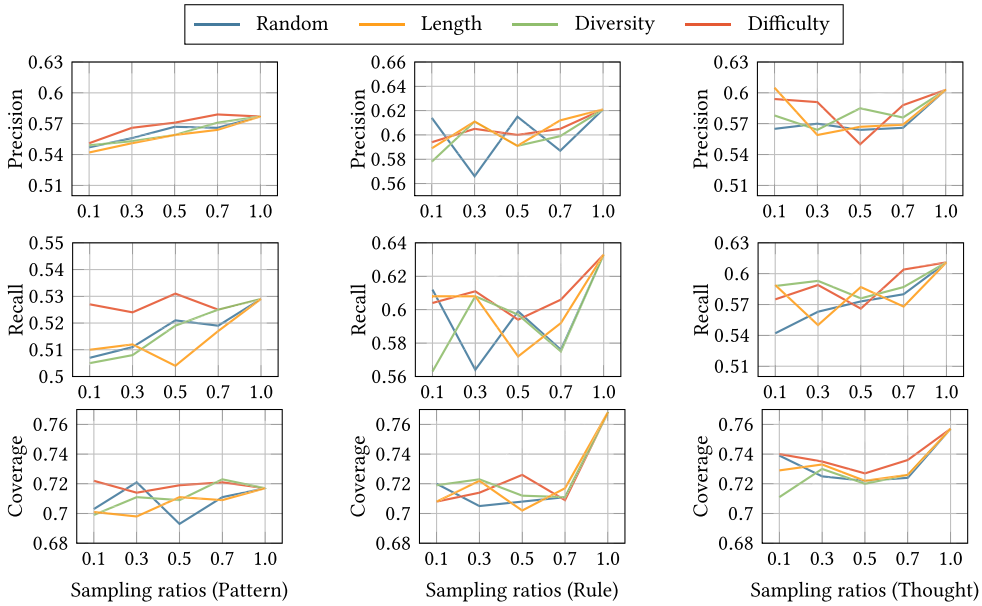


Fig. 6. Performance comparison of Llama3.1-8B-ICL with GPT-3.5-turbo as the teacher under different sampling strategies and ratios in the Clothing domain (RQ2, Section 4.3).

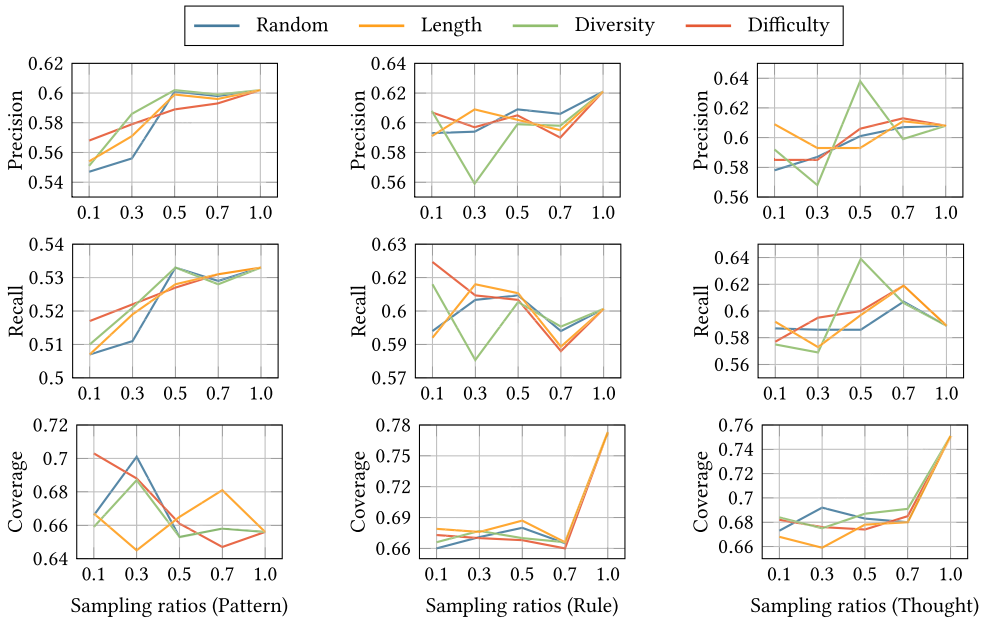


Fig. 7. Performance comparison of Llama3.1-8B-ICL with GPT-3.5-turbo as the teacher under different sampling strategies and ratios in the Food domain (RQ2, Section 4.3).

Figures 8–10 illustrate the performance of Llama3.1-8B-SFT with GPT-3.5-turbo as the teacher across three domains, revealing different observations compared to Llama3.1-8B-ICL. (1) *The performance improves as the sampling ratio (i.e., knowledge) increases, reaching its peak with a certain ratio*

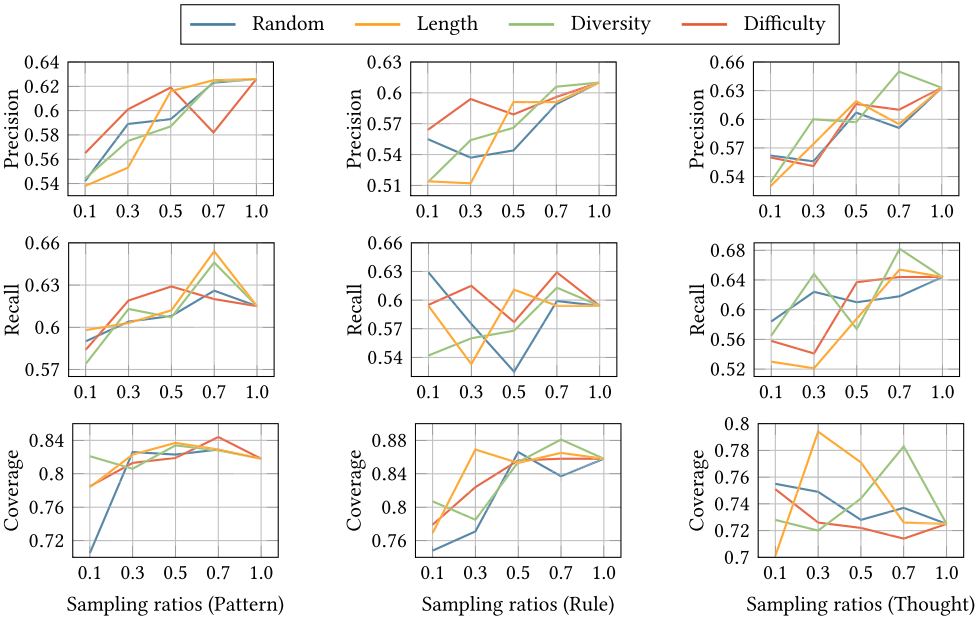


Fig. 8. Performance comparison of Llama3.1-8B-SFT with GPT-3.5-turbo as the teacher under different sampling strategies and ratios in the Electronic domain (RQ2, Section 4.3).

(e.g., 0.7) and then shows a slight drop. This suggests that for Llama3.1-8B-SFT, sampling data at an appropriate quantity range can achieve optimal performance, while adding more data yields limited benefits even degradation. (2) *Sampling strategy has a greater impact on performance.* Different sampling strategies lead to relatively noticeable performance difference at the same sampling ratio. Specifically, for Llama3.1-8B-ICL, the average performance standard deviation of the four sampling strategies across different sample ratios, knowledge formats, and domains for Precision, Recall, and Coverage are 0.013, 0.013, 0.01, respectively. In contrast, the corresponding average performance standard deviation for Llama3.1-8B-SFT are 0.025, 0.029, 0.022. The difference is statistically significant, as confirmed by a paired t-test with a p-value < 0.01. These results suggest that Llama3.1-8B-SFT is more sensitive to the choice of sampling strategy compared to Llama3.1-8B-ICL.

Impact of Sampling Strategies on Other Teacher–Student Pairs. We further investigate whether the impact of sampling strategies varies with different teacher–student combinations. Other teacher–student pairs and domains exhibit similar trends, so, due to page limits, we report detailed curves only for the GPT-4.1+Llama3.2-3B pair on the Electronic domain. Figures 11 and 12 illustrate the performance of Llama3.2-3B distilled from GPT4.1 under the ICL and SFT settings in the Electronic domain. Regarding sample ratios, we observe trends consistent with the GPT-3.5-turbo and Llama3.1-8B pair. Specifically, Llama3.2-3B-ICL exhibits a "more is better" trend where performance scales with the ratio. In contrast, Llama3.2-3B-SFT shows that performance peaks at a specific ratio (e.g., 0.7) before saturating or degrading, indicating that for SFT, an excessive amount of distilled knowledge does not guarantee better internalization. Regarding sampling strategies, the results reveal that model capacity influences sensitivity to data selection. For ICL, similar to the 8B model, the small gaps between sampling strategies indicate that increasing the sampling ratio to obtain more high-quality references is more important than how those samples are selected. However, for SFT, the choice of sampling strategy has a significant impact. Notably, Llama3.2-3B-SFT shows

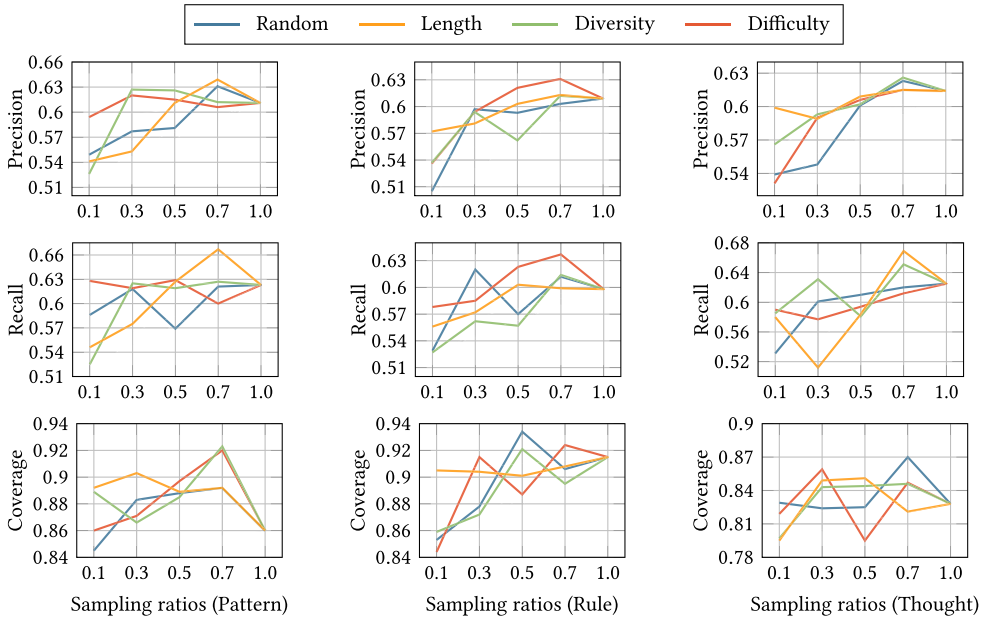


Fig. 9. Performance comparison of Llama3.1-8B-SFT with GPT-3.5-turbo as the teacher under different sampling strategies and ratios in the Clothing domain (RQ2, Section 4.3).

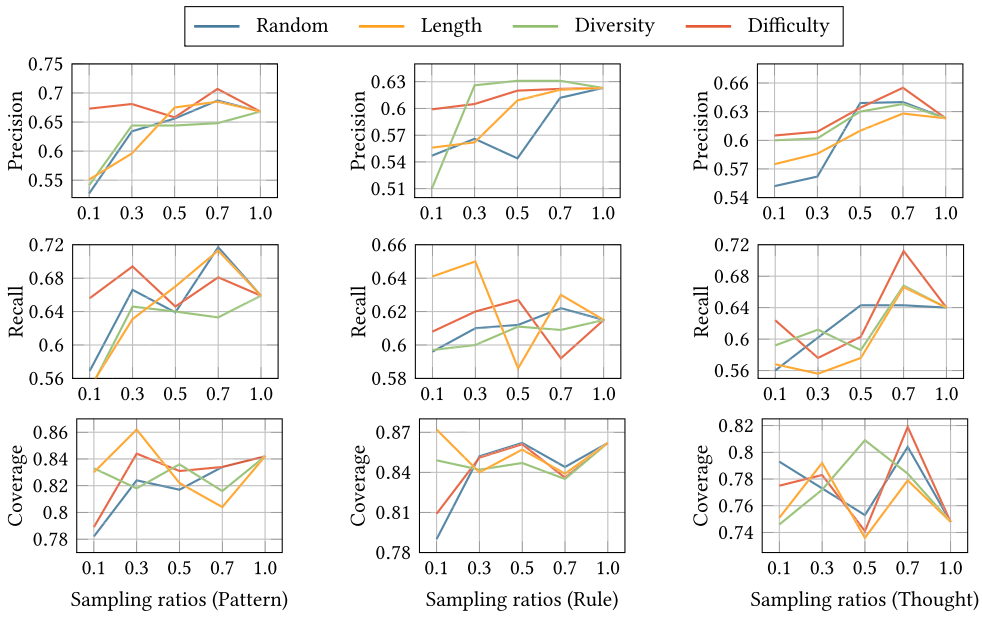


Fig. 10. Performance comparison of Llama3.1-8B-SFT with GPT-3.5-turbo as the teacher under different sampling strategies and ratios in the Food domain (RQ2, Section 4.3).

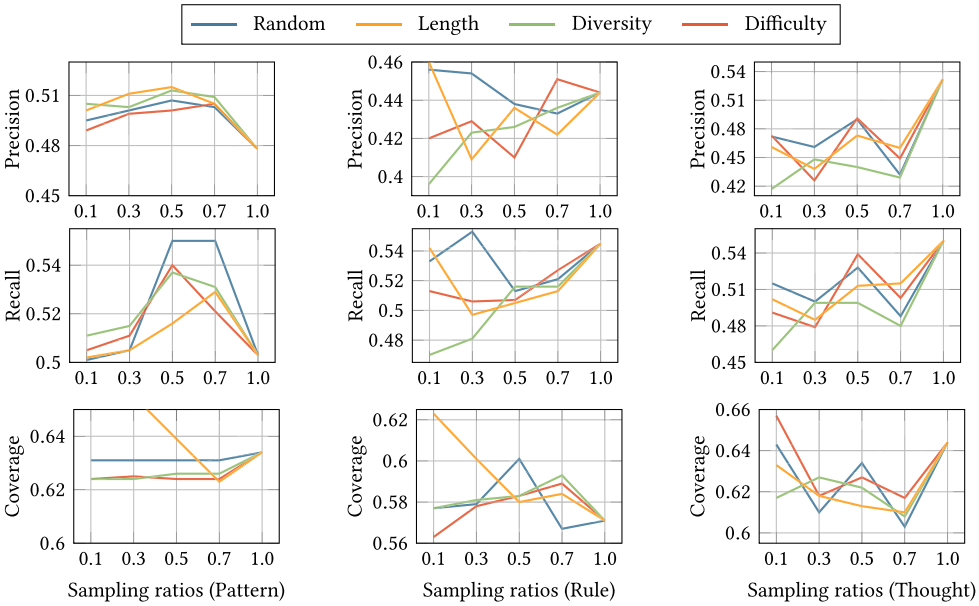


Fig. 11. Performance comparison of Llama3.2-3B-ICL with GPT-4.1 as the teacher under different sampling strategies and ratios in the Electronic domain (RQ2, Section 4.3).

greater performance variance across different strategies compared to Llama3.1-8B-SFT. This suggests that smaller student models are more sensitive to the quality and diversity of training data, making the design of effective sampling strategies even more critical when fine-tuning compact models.

4.3.2 Impact of Knowledge Accumulation from Different Domains on the GPT-3.5-Turbo+Llama3.1-8B Pair. First, we find that *accumulation from different domains does not necessarily impact the performance of Llama3.1-8B-ICL.* This is because ICL typically retrieves the session most similar to the target session and utilizes its associated knowledge to enhance bundle generation performance. However, sessions from other domains are often dissimilar to the target session, making their accumulated knowledge less likely to be leveraged. Second, we observe that *aggregating knowledge from multiple domains consistently outperforms using knowledge from a single domain, regardless of the knowledge type across the three domains when using Llama3.1-8B-SFT.* In particular, Figure 13 illustrates the performance comparison between accumulating different knowledge types from single (black line) to multiple domains (red line) using Llama3.1-8B-SFT distilled from GPT-3.5-turbo, where the x-axis represents different types of knowledge, and "Raw" indicates that no explicit knowledge is considered. This improvement can be attributed to the nature of SFT: unlike ICL, SFT leverages the accumulated knowledge to fine-tune the student model. Aggregated knowledge from multiple domains (1) brings diversity, preventing the model from over-fitting; and (2) provides complementary insights, allowing the model to learn more generalizable patterns for bundle generation. For instance, understanding the concept of "complementary accessories" from Electronic might help generalize to "coordinating items" in Clothing, even if the specific items differ.

Impact of Knowledge Accumulation from Different Domains on Other Teacher–Student Pairs. To verify the universality of knowledge accumulation across domains, we conducted experiments on the Electronic domain with the rest three teacher–student pairs. As shown in Figure 14, aggregating knowledge from multiple domains consistently outperforms single-domain knowledge across all

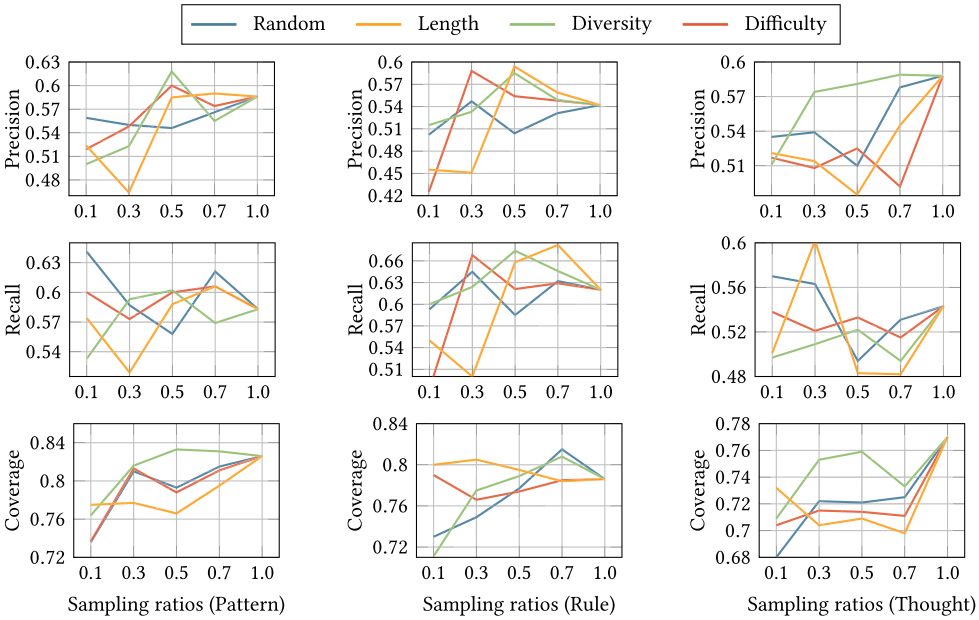


Fig. 12. Performance comparison of Llama3.2-3B-SFT with GPT-4.1 as the teacher under different sampling strategies and ratios in the Electronic domain (RQ2, Section 4.3).

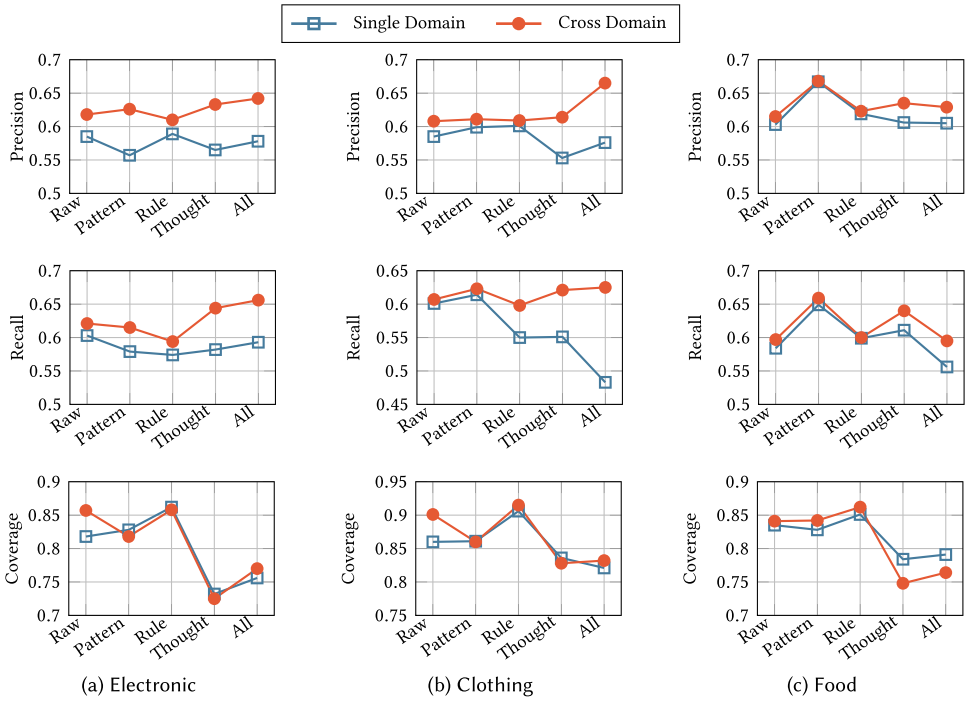
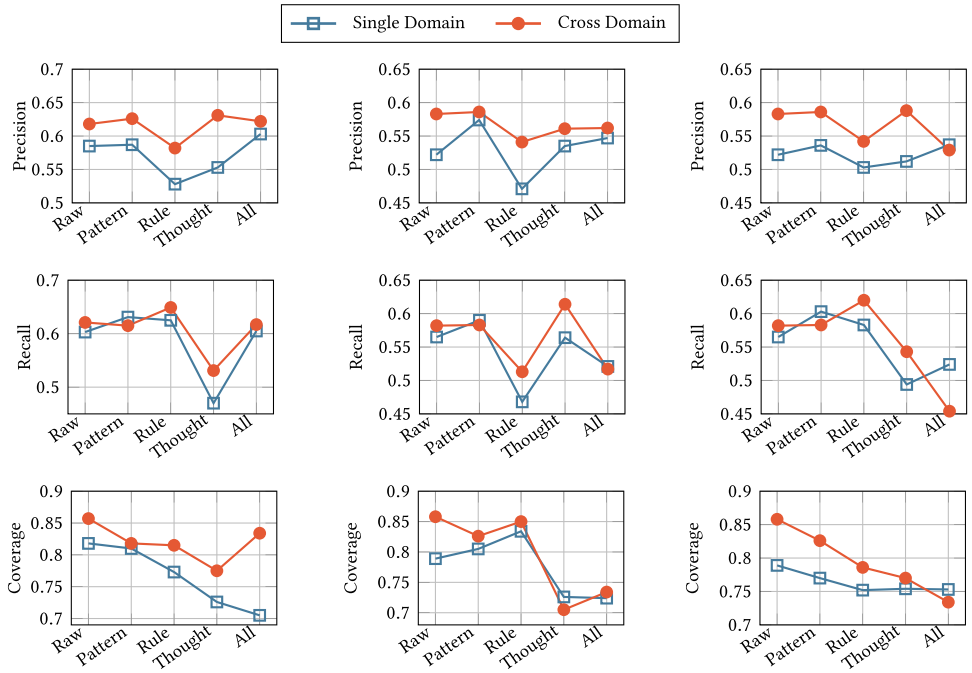


Fig. 13. The results of knowledge accumulation from different domains in the Electronic, Clothing, and Food domains using Llama3.1-8B-SFT with GPT-3.5-turbo as the teacher (RQ2, Section 4.3).



(a) GPT-4.1 and Llama-3.1-8B-SFT (b) GPT-3.5-Turbo and Llama-3.2-3B-SFT (c) GPT-4.1 and Llama-3.2-3B-SFT

Fig. 14. Results of knowledge accumulation from different domains across teacher–student pairs in the Electronic domain (RQ2, Section 4.3).

combinations using SFT. This confirms that the benefit of cross-domain accumulation is robust and not specific to a single model. However, we observe a difference in the magnitude of improvement: pairs involving the larger student model (Llama3.1-8B, Figure 14(a)) tend to exhibit a more distinct performance gap between single and cross-domain settings compared to the smaller student model (Llama3.2-3B, Figure 14(b) and (c)). This indicates that while cross-domain knowledge is universally beneficial, models with larger parameter capacities possess a stronger ability to synthesize and generalize diverse patterns from varying domains.

4.3.3 Impact of Knowledge Accumulation from Different Formats on the GPT-3.5-Turbo+Llama3.1-8B Pair. Figures 15 and 16 present the results of accumulating different types of knowledge in the three domains using Llama3.1- 8B-SFT with GPT-3.5-turbo as the teacher, where y-axis represents different combinations of knowledge (from single type to two and three types). Some observations can be obtained.

(1) *For Llama3.1-8B-ICL, combining two types of knowledge (e.g., Pattern + Rule or Rule + Thought) generally improves Precision and Coverage across all three domains compared to using a single knowledge type, but does not improve Recall.* The gain in Precision and Coverage may stem from the model receiving more comprehensive guidance through multiple knowledge types, thus generating more precise bundles. However, using Rule alone yields the highest Recall in the Electronic and Clothing domains, and a relatively high Recall in the Food domain. This suggests that, in the ICL setting, adding more knowledge types may increase the model's confidence in identifying the most relevant bundle within a session. As a result, the model becomes more conservative, focusing on a small number of high-confidence predictions while overlooking other potential candidates. Such

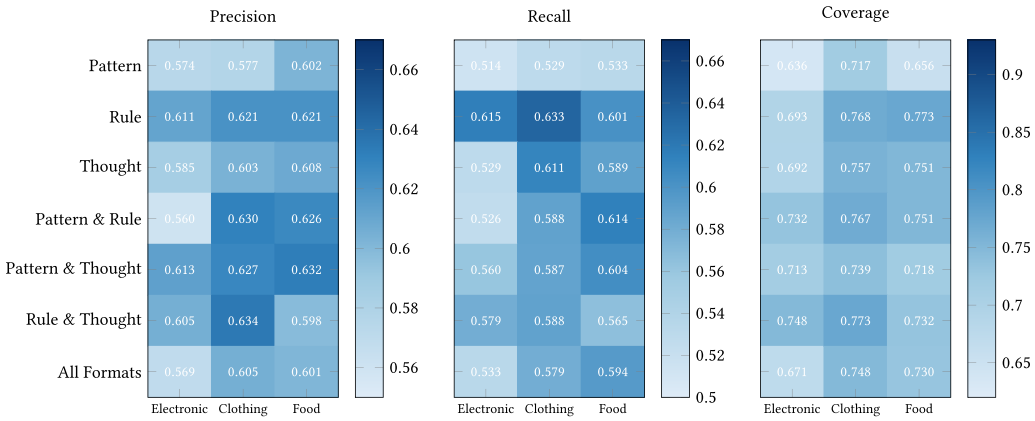


Fig. 15. Results of knowledge accumulation from different formats in the Electronic, Clothing, and Food domains using Llama3.1-8B-ICL with GPT-3.5-turbo as the teacher (RQ2, Section 4.3).

Fig. 16. Results of knowledge accumulation from different formats in the Electronic, Clothing, and Food domains using Llama3.1-8B-SFT with GPT-3.5-turbo as the teacher (RQ2, Section 4.3).

behavior increases Precision at the expense of Recall. Moreover, we notice that integrating all types of knowledge does not lead to further improvements. This is likely due to two main challenges: (i) the longer context increases the model's processing burden, and (ii) potential inconsistencies among different knowledge formats may introduce confusion. Compared to combining two knowledge types, merging all three increases the risk of conflicting signals, making it harder for the model to determine which guidance to prioritize, ultimately leading to diminished performance.

(2) *For Llama3.1-8B-SFT, combining all three types of knowledge yields the highest Precision and Recall in the Electronic and Clothing domains. Unlike Llama3.1-ICL, the performance trends for Precision and Recall with Llama3.1-8B-SFT are more consistent.* This can be attributed to the nature of SFT, which updates the model's internal parameters through paired input–output examples. This fine-tuning process enables the model to effectively integrate diverse signals and understand how different types of knowledge interact to support bundle generation. As a result, the model can leverage the complementary strengths of various knowledge formats, leading to improvements in both Precision and Recall, as well as greater stability in performance. However, the model's ability to balance diverse knowledge sources can still be improved. Notably, the best Coverage is achieved

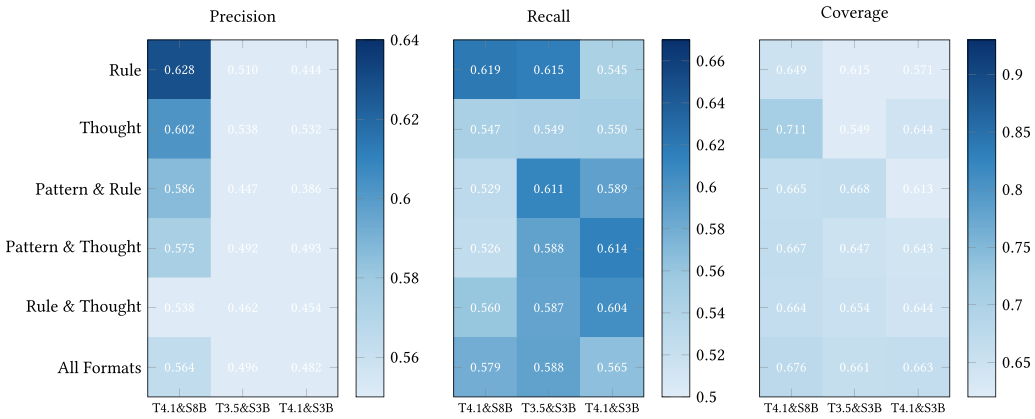


Fig. 17. Results of knowledge accumulation from different formats across teacher–student pairs in the Electronic domain using ICL (RQ2, Section 4.3). Due to space limitations, we abbreviate the teacher–student pairs. For example, T4.1&S8B denotes the pair of GPT-4.1 teacher and Llama3.1-8B student.

by combining only two knowledge types—Pattern and Rule—suggesting that the integration of all three knowledge types may still introduce redundant or conflicting signals, thereby limiting the model's capacity to generate bundles of greater size.

Impact of Knowledge Accumulation from Different Formats on Other Teacher–Student Pairs. We further investigate the impact of accumulating knowledge formats on different teacher–student pairs in the Electronic domain. Figures 17 and 18 show that the distinct behaviors between ICL and SFT remain consistent across different model capabilities. For ICL, smaller student models (Llama3.2-3B) show a more significant performance drop when integrating all three types of knowledge compared to Llama3.1-8B. This exacerbates the "less is more" observation, likely because the limited context window and reasoning capacity of smaller models make them more susceptible to information overload and conflicting signals from heterogeneous knowledge formats. In contrast, for SFT, all student models consistently achieve optimal or near-optimal performance when combining multiple knowledge formats. This reinforces the conclusion that SFT is a more robust mechanism for integrating multiple types of distilled knowledge, effectively smoothing out the inconsistencies that hinder prompt-based inference.

According to the analysis in Sections 4.3.1–4.3.3, the answer to RQ2 is:

> *Answer to RQ2:* Overall, increasing the quantity of distilled knowledge positively impacts bundle generation performance. Specifically, ICL benefits more from higher sampling ratios, as it relies on a larger knowledge pool for effective retrieval. In contrast, SFT tends to peak at a certain sampling ratio and gains substantial improvements from incorporating knowledge across different formats and domains. These trends remain consistent across different teacher–student pairs: different students mainly affect how much performance improves, rather than which settings work best.

4.4 Investigation on Utilization Method of Distilled Knowledge (RQ3)

In this section, we study how different utilization methods for distilled knowledge affect bundle generation to answer RQ3. We first conduct a detailed analysis on the GPT-3.5-turbo+Llama3.1-8B

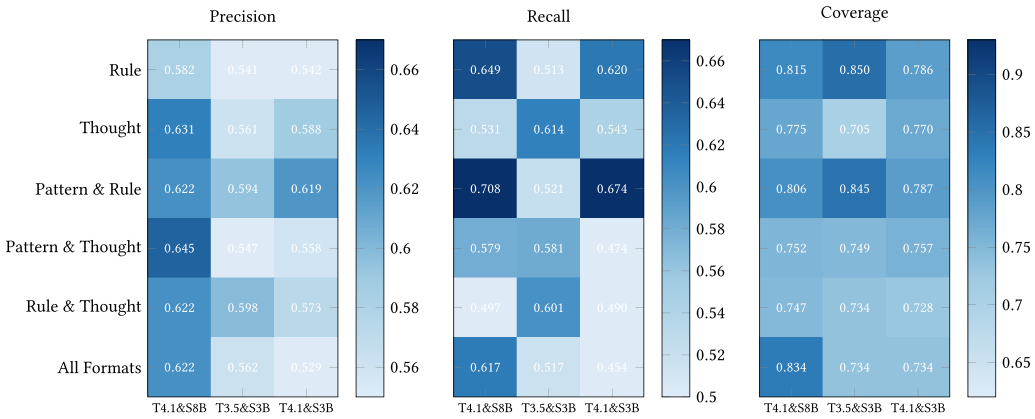


Fig. 18. Results of knowledge accumulation from different formats across teacher–student pairs in the Electronic domain using SFT (RQ2, Section 4.3). Due to space limitations, we abbreviate the teacher–student pairs. For example, T4.1&S8B denotes the pair of GPT-4.1 teacher and Llama3.1-8B student.

pair and then extend the comparison to other teacher–student pairs. For each setting, we consider three utilization methods—ICL, SFT, and their combination ICL+SFT—as introduced in Section 3.4.

4.4.1 Impact of Utilization Method on the GPT-3.5-Turbo+Llama3.1-8B Pair. Figure 19 shows the results of applying the three methods to the student LLM (Llama3.1-8B) with different knowledge distilled from GPT-3.5-turbo. In these figures, the colored bars (SFT+K) represent SFT with different types of distilled knowledge K for fine-tuning (e.g., the black bar represents using Pattern knowledge in the SFT); the x-axis (ICL+K) represents ICL with different types of distilled knowledge K for inference (e.g., ICL+Pattern means Pattern is used in ICL); and the y-axis represents the performance differences (gaps) between (ICL+K) and different combinations of (ICL+K) and (SFT+K).[5] For instance, the black bar under ICL+Pattern represents the performance difference between ICL+Pattern and (ICL+Pattern, SFT+Pattern). Particularly, the purple bars show the performance of integrating knowledge only in the SFT, so the corresponding y-value indicates the performance difference of using the same type of distilled knowledge in ICL for inference and in SFT for fine-tuning (i.e., ICL+K vs. SFT+K).

From the results, four major observations can be noted. (1) *The combination (ICL+K, SFT+K) generally yields better performance than either ICL+K or SFT+K alone, indicating that leveraging knowledge in both ICL and SFT is more effective than using it in only one of them.* As observed, in the majority of cases, the colored bars correspond to positive y-values, and bars in black, red, green, and yellow exhibit higher y-values compared to those in purple. This suggests that the student model benefits from using the distilled knowledge both for fine-tuning its parameters and as guidance during inference. Alternatively stated, the student model fine-tuned with distilled knowledge can continue to benefit from domain-specific guidance during inference, further boosting task performance. (2) *The combination (ICL+K, SFT+K) does not always achieve positive gains, which is highly dependent on the specific knowledge K used.* For instance, SFT+Pattern (black bar) usually shows positive y-values, regardless of the type of knowledge used in ICL. Similarly, ICL+Pattern generally gains positive y-values regardless of the knowledge utilized in SFT. In contrast, the combination (ICL+Thought/All, SFT+Thought/All) tends to result in negative y-values in most

[5]We compute the y-value by subtracting the performance of (ICL+K) from that of each combination (ICL+K, SFT+K).

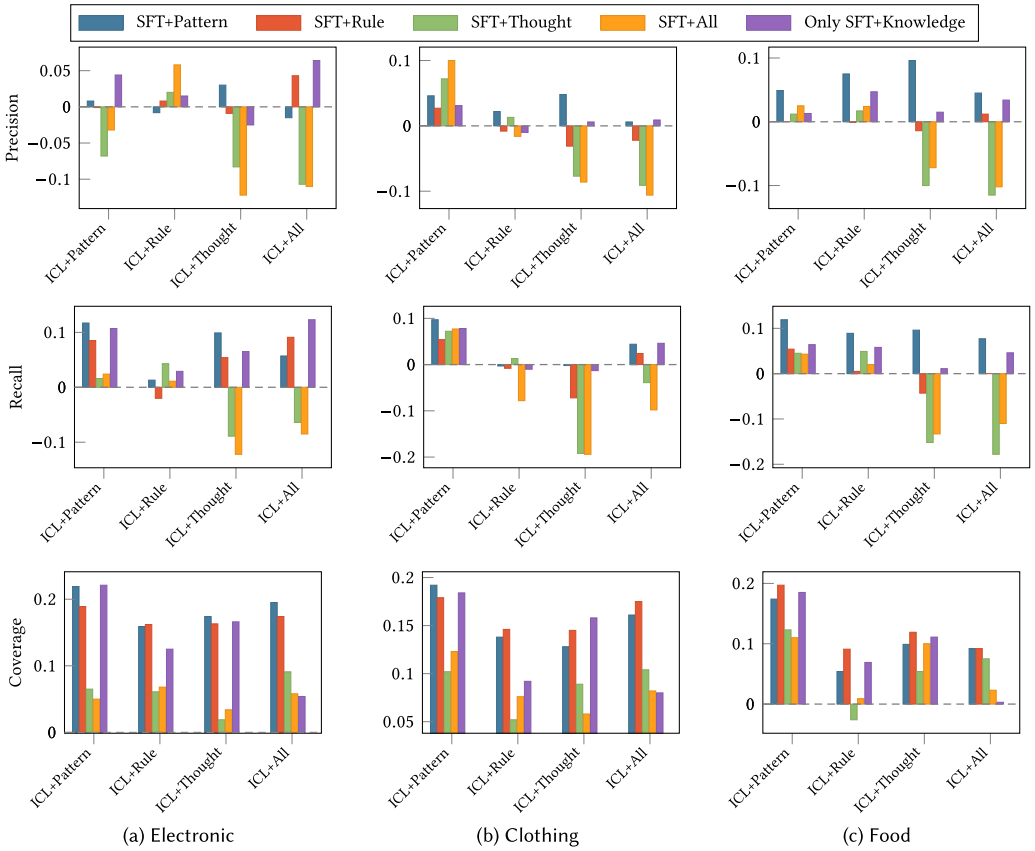
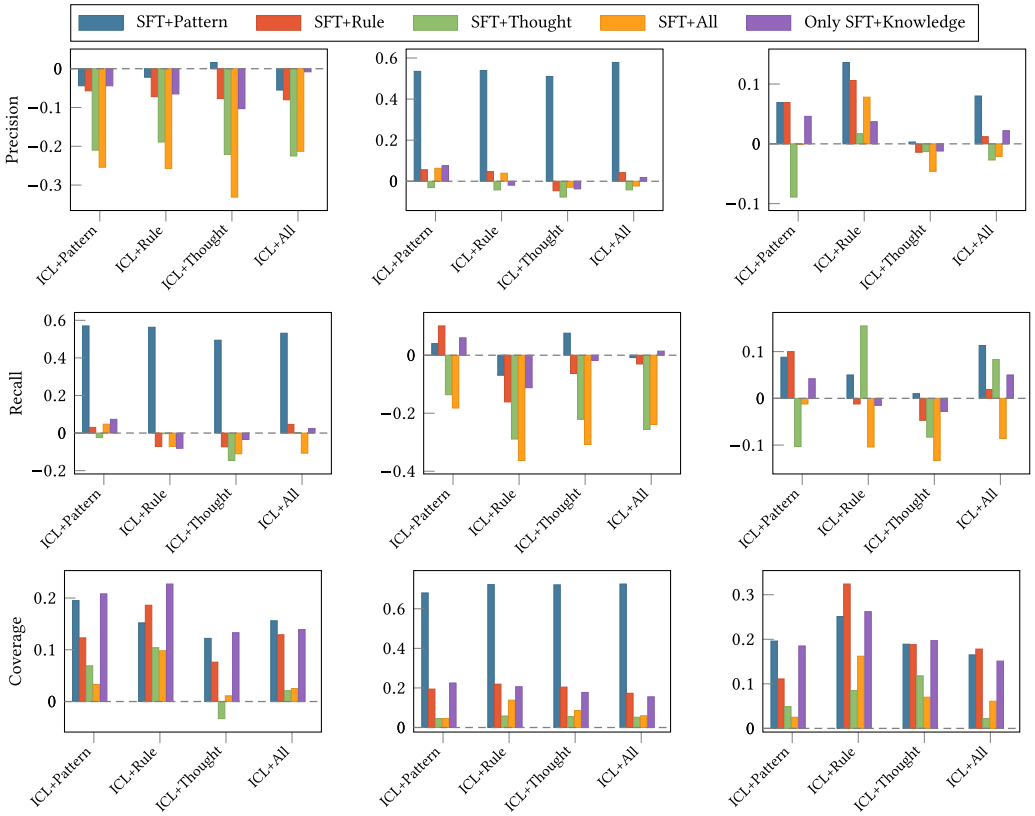


Fig. 19. Relative performance differences of various knowledge utilization strategies on the GPT-3.5-turbo+Llama3.1-8B pair across the three domains (RQ3, Section 4.4).

cases. This indicates a potential risk of combining divergent knowledge types across SFT and ICL: when the information introduced at each stage is not well aligned, it can impair the model's ability to focus and generalize effectively. (3) *SFT+K alone yields positive y-values in most cases, suggesting that incorporating distilled knowledge into SFT is more effective than doing so in ICL (can also refer to* Table 7*).* Moreover, SFT can even outperform the combination (ICL+K, SFT+K) in certain cases—for example, in terms of Precision and Recall in the Electronic domain. This suggests a stronger ability of SFT in injecting knowledge into the student model.

4.4.2 Impact of Utilization Method on Other Teacher–Student Pairs. We further investigate whether the effect of different utilization methods of distilled knowledge is consistent in the Electronic domain across the rest three teacher–student pairs. As shown in Figure 20, we observe that the qualitative conclusions in this section are robust across all these pairs. First, the hybrid strategy that leverages knowledge in both ICL and SFT (ICL+K, SFT+K) generally yields the best or near-best performance compared with using ICL+K or SFT+K alone, indicating that training-time and inference-time utilization of distilled knowledge are largely complementary rather than mutually exclusive. Second, for every teacher–student pair, SFT+K alone tends to outperform ICL+K alone, especially in terms of Recall and Coverage, which confirms that injecting distilled knowledge into the student's parameters provides a more stable and effective way of exploiting



(a) GPT-4.1 and Llama3.1-8B (b) GPT-3.5-turbo and Llama3.2-3B (c) GPT-4.1 and Llama3.2-3B

Fig. 20. Relative performance differences of various knowledge utilization strategies across different teacher–student pairs in the Electronic domain (RQ3, Section 4.4).

it than using it only as retrieved context during inference. Third, the performance gains from combining ICL and SFT are more pronounced for the smaller student (Llama3.2-3B), while the larger student (Llama3.1-8B) already achieves strong performance with SFT+K and thus benefits less from additional ICL guidance. These observations suggest that although the absolute improvements vary with model capacity and teacher strength, the relative advantages of different utilization methods identified in RQ3 remain consistent across teacher–student pairs.

Overall, the main answer to RQ3 is:

> *Answer to RQ3:* Bundle generation performance significantly depends on the knowledge utilization method. Using distilled knowledge in both SFT and ICL generally yields the best performance, but its effectiveness depends on the specific knowledge used. Notably, SFT with distilled knowledge alone is more consistently effective than ICL alone. These advantages remain stable across teacher–student pairs, indicating that the utilization method matters more than which teacher–student pair is used.

Table 9. Effectiveness Comparison across Different Models in the Three Domains

Type	Method	Electronic			Clothing			Food		
		Precision	Recall	Coverage	Precision	Recall	Coverage	Precision	Recall	Coverage
Conventional Models	Freq	0.423	0.597	0.701	0.532	0.566	0.698	0.491	0.525	0.684
	BBPR	0.260	0.122	0.433	0.239	0.211	0.449	0.210	0.183	0.416
	POG	0.339	0.250	0.412	0.312	0.221	0.399	0.365	0.266	0.393
	BYOB	0.340	0.294	0.361	0.311	0.273	0.457	0.304	0.253	0.427
LLM-based Models	AICL	0.769	0.859	0.741	0.677	0.788	0.839	0.698	0.851	0.755
	Teacher Model	0.580	0.820	0.720	0.603	0.752	0.788	0.604	0.815	0.748
	Student_RAG	0.434	0.432	0.609	0.463	0.479	0.697	0.442	0.485	0.654
	Student_RQ1	0.633	0.644	0.725	0.614	0.621	0.828	0.668	0.649	0.842
	Student_RQ2	0.650	**0.682**	**0.783**	0.665	**0.625**	0.832	**0.707**	0.681	0.834
	Student_RQ3	**0.669**	0.626	0.761	**0.677**	0.606	**0.840**	0.704	**0.685**	**0.850**
Improve		15.35%	−16.83%	8.75%	12.27%	−16.89%	6.60%	17.05%	−15.95%	13.64%

4.5 Comparison on Effectiveness and Efficiency

The previous analysis in Sections 4.2–4.4 demonstrates how the format, quantity, and utilization method of distilled knowledge affect the performance of the student model on the bundle generation task. Now, we further analyze the effectiveness and efficiency of the student model in comparison with both conventional models and its teacher models.

4.5.1 Analysis on Effectiveness. For ease of analysis, we use GPT-3.5-turbo+Llama3.1-8B as our primary teacher–student pair. Generally, using a stronger teacher (i.e., GPT-4.1) leads to similar performance trends but slightly reduces the student's relative improvements. For example, as shown in Tables 7 and 8, when comparing Llama3.1-8B students distilled from GPT-4.1 and GPT-3.5-turbo, the " *Improve* " values on most metrics decrease slightly, while Precision and Coverage may marginally increase in some cases (e.g., Precision in the Clothing domain increases from 2.99% to 5.25%, whereas Coverage in the Food domain decreases from 15.24% to 13.69%). Given these consistent trends, Table 9 reports the results under GPT-3.5-turbo+Llama3.1-8B, summarizing the effectiveness of conventional models, GPT-3.5-turbo as the teacher model, and Llama3.1-8B as the student model across all metrics and domains. For student models, we report the best-performing result under each RQ (denoted as Student_RQX).

Specifically, for RQ1, we select Llama3.1-8B-SFT with Thought for the Electronic and Clothing domains, and Llama3.1-8B-SFT with Pattern for the Food domain. For RQ2, the best results come from Llama3.1-8B-SFT with Thought using diversity-based sampling (sampling ratio = 0.7) for Electronic, Llama3.1-8B-SFT with All formats of knowledge using full raw data for Clothing, and Llama3.1-8B-SFT with Pattern using difficulty-based sampling (sampling ratio = 0.7) for Food. For RQ3, we choose Llama3.1-8B-SFT with All formats of knowledge in fine-tuning and ICL with Rule in inference for Electronic; Llama3.1-8B-SFT with All formats of knowledge and ICL with Pattern for Clothing; and Llama3.1-8B-SFT with Pattern and ICL with Thought for Food. Moreover, we further include a retrieval-augmented student model [30], denoted as Student_RAG. Specifically, we adopt a dense-retrieval pipeline that shares the same session representation and top-k selection strategy as the ICL setting: for each session in training set, we encode the titles of all interacted items using a BERT encoder and aggregate them into a single session embedding. At inference time, given a target session, we retrieve the top-k most similar historical session based on cosine similarity and use its associated bundle as an example. The retrieved bundle is concatenated with the current session description as additional context for the student model to generate the target bundle. The performance of the teacher model (GPT-3.5-turbo) is highlighted in gray; the best performance achieved by the student models is in bold; and "Improve" indicates the relative improvements achieved by the best-performing student model against the teacher model.

We make the following observations: (1) *LLMs-based models outperform the conventional models across all metrics.*[6] Both the teacher and student models, based on LLMs, consistently achieve better performance across all metrics and domains compared to conventional models. This advantage stems from their extensive pre-training, inherent reasoning capabilities, and effective adaptation even with limited task-specific data—a scenario where conventional models may be underfitting. (2) *Directly using RAG on the student model is not effective.* Although Student_RAG augments the input with retrieved sessions, it still underperforms the distilled students across domains. This suggests that simply attaching a dense-retrieval module to a smaller LLM is insufficient for the bundle generation task. A likely reason is that the retrieved sessions often contain only a subset of bundles that truly provide useful guidance for the target session, while other bundles introduce ambiguous or even conflicting signals with respect to the current session, such conflicts are hard for a small student model to resolve, and the noisy references can ultimately hurt its generation quality. (3) *KD enables student models to outperform the teacher model on certain metrics.* While the teacher model achieves the best Recall, KD successfully empowers the smaller student model to achieve highly competitive, and in some aspects superior, results. In particular, the best-performing student models consistently outperform the teacher model in Precision and Coverage across all domains. Additionally, when compared to AICL, the advanced LLM-based method for bundle generation, the student models achieve higher Coverage across the three domains, and notably higher Precision in the Food domain. These results strongly confirm the effectiveness of KD for more effective bundle generation. (4) *Optimizing knowledge utilization method (RQ3) is crucial for maximizing the performance of student models.* As shown in Table 10, we calculate the relative performance gap under each RQ across the three domains, defined as $(best\_result - worst\_result)/worst\_result$.[7] Intuitively, larger performance gaps reflect stronger impacts. Based on the results, we find the following observations: (i) Among the three factors (format, quantity, and utilization method), the utilization method generally has the greatest overall impact on model performance, while the knowledge format contributes the least. This trend is evident in the results with black background—RQ1: 11.33%, RQ2: 23.21%, RQ3: 39.62%. (ii) This pattern is consistent across individual domains (highlighted with red backgrounds), with the exception of the Food domain. In that case, knowledge quantity shows a slightly higher impact than the utilization method (RQ2: 24.84% vs. RQ3: 24.58%), although the difference is marginal. (iii) For both ICL and SFT, knowledge quantity consistently has a stronger influence than knowledge format. This effect is more pronounced in SFT. Specifically, as shown in the orange background for ICL: RQ1: 13.90% vs. RQ2: 18.93%; and in the green background for SFT: RQ1: 8.75% vs. RQ2: 27.48%. (iv) As shown in Table 11, when we further decompose results from the Electronic domain in Table 10 by different teacher–student pairs, the same patterns observed in (i)–(iii) still hold. In each pair, the utilization method (RQ3) induces the largest relative performance gap, and for both ICL and SFT, the gaps associated with knowledge quantity (RQ2) are consistently larger than those of knowledge format (RQ1), with this effect again more pronounced on the SFT side. This confirms that our conclusions about the dominant role of utilization method and the stronger influence of knowledge quantity over format are not tied to a specific teacher–student choice, but remain stable across different model pairings.

[6]For conventional models, training on multiple domains did not improve performance and, in some cases, even led to degradation. Therefore, we used domain-specific training for these models. This may be due to the fact that cross-domain data introduce noise and increase training difficulty for conventional models, whereas large-scale LLMs benefit from diverse data and exhibit better generalization.

[7]We report the result for each domain by averaging the relative performance gaps across the three metrics (Precision, Recall, and Coverage). For RQ1 and RQ2, we also compute the gaps separately for each utilization method (i.e., ICL and SFT) and then average the gaps of the two methods to obtain an overall result for each RQ.

Table 10. The Relative Performance Gap between the Best and Worst Result under Each Research Question across the Three Domains

	RQ1: Format of Knowledge			RQ2: Quantity of Knowledge			RQ3: Utilization Methods of Knowledge
	ICL	SFT	Avg.	ICL	SFT	Avg.	Avg.
Electronic	12.31%	10.18%	11.25%	22.54%	28.11%	25.33%	44.56%
Clothing	14.85%	5.22%	10.04%	15.00%	23.91%	19.46%	49.72%
Food	14.54%	10.86%	12.70%	19.25%	30.42%	24.84%	24.58%
Avg.	13.90%	8.75%	11.33%	18.93%	27.48%	23.21%	39.62%

The different background colors represent the influence of three factors—knowledge format, quantity, and utilization method—from various perspectives. Specifically, red, orange, and green indicate their impacts on different domains, ICL and SFT, respectively, while blue reflects their overall impact. A darker shade corresponds to a larger performance gap, indicating a stronger influence.

Table 11. The Relative Performance Gap between the Best and Worst Result under Each Research Question for Different Teacher–Student Pairs in the Electronic Domain

	RQ1: Format of Knowledge			RQ2: Quantity of Knowledge			RQ3: Utilization Methods of Knowledge
	ICL	SFT	Avg.	ICL	SFT	Avg.	Avg.
GPT4.1&Llama3.1	18.41%	17.34%	17.88%	15.01%	15.41%	15.21%	53.13%
GPT3.5&Llama3.2	15.38%	16.57%	15.98%	19.98%	32.22%	26.10%	37.75%
GPT4.1&Llama3.2	16.67%	21.53%	19.10%	15.80%	22.84%	19.32%	24.58%
Avg.	16.82%	18.48%	17.65%	16.93%	23.49%	20.21%	46.40%

The different background colors represent the influence of three factors (knowledge format, quantity, and utilization method) from various perspectives.

User Study. To further assess the practical quality of the student model in bundle generation, we conduct a user study along three aspects followed by [47]. Specifically, *Relevance* measures whether a bundle is reasonable as a whole and whether its items are well coordinated and its scale range from 1 to 3 (1—the bundle is unreasonable or items are not coordinated; 2—the bundle is somewhat reasonable; 3—the bundle is reasonable and items are well coordinated). *Diversity* captures the variety among items in a bundle, on a 3-point scale (1—the items are very similar or lack variety; 2—the items exhibit some variety; 3—the items are diverse). *Attractiveness* reflects how appealing the bundle is to consumers, with scale ranging from 1 to 3 (1—the bundle is not attractive; 2—bundle is somewhat attractive; 3—the bundle is very attractive). For each domain, we randomly sample 10 sessions from the test set (30 sessions in total) and evaluate three models: BYOB (the best-performing conventional method), GPT-4.1 (the strongest teacher LLM), and the best-performing student model in Table 9. We include GPT-4.1 because it serves as a strong reference (upper bound) to assess whether this advantage remains perceptible in real user interactions and to quantify the practical gap. To avoid bias, we anonymize the model name and randomly shuffle the order of their bundles for every case. We then recruit 10 volunteers with online shopping experience to independently rate each bundle based on the above three metrics. The average scores across annotators and evaluation cases are reported in Table 12. We observe that both LLM-based methods outperform the conventional model across all criteria and domains. The teacher model achieves the highest scores in most settings, especially in terms of attractiveness, confirming its strong ability to generate appealing bundles. Nevertheless, the student model remains highly competitive: the score gaps between the student and teacher models are small, and the student even slightly surpasses the teacher in relevance for *Clothing* and diversity for *Food*. Considering that the student model

Table 12. Human Evaluation on Bundle Generation across Three Domains

	Electronic			Clothing			Food		
	Relevance	Diversity	Attractiveness	Relevance	Diversity	Attractiveness	Relevance	Diversity	Attractiveness
BYOB	2.17	2.38	1.97	2.09	1.99	2.09	2.06	2.07	1.99
Student Model	2.58	2.59	2.32	**2.42**	2.33	2.28	2.64	**2.39**	2.31
Teacher Model	**2.77**	**2.82**	**2.70**	2.41	**2.50**	**2.56**	**2.74**	2.32	**2.54**

The statistical significance of pairwise differences of Student vs. Teacher is determined by a paired t-test with a p-value < 0.05. The values in bold indicate the best results in each column.

Table 13. Efficiency Comparison between the Student Model (8B) and the Teacher Proxy (70B) on RTX PRO 6000 with 4-Bit Quantization

Model Size	Inference Time (ms/bundle)↓	Throughput (bundles/ms)↑	VRAM Cost (GB)↓	Speedup↑
70B (Teacher Proxy)	757.55	0.0013	45	1.0×
8B (Student)	**162.56**	**0.0062**	**8**	**4.66×**

The values in bold indicate the best results in each column.

is much more efficient at inference time, these results indicate that our distillation framework preserves most of the teacher's user-perceived quality while reducing computational cost.

4.5.2 Analysis on Efficiency. To quantify the efficiency gains, we conduct a comparative analysis between the student model (Llama3.1-8B) and a teacher-level proxy (Llama3.1-70B). Due to the deployment constraints of teacher models like GPT-3.5-turbo, we selected Llama3.1-70B as the baseline for the teacher model, given its comparable performance in the electronic domain (Zero-shot Precision 0.654, Recall 0.787). Both models were deployed using 4-bit quantization on the Ollama[8] framework, running on a consistent hardware setup equipped with an NVIDIA RTX PRO 6000 (96GB VRAM). As shown in Table 13, the student model demonstrates significant advantages in both inference latency and resource consumption. Specifically, the student model achieves an average inference speed of 162.56 ms per bundle, which is approximately 4.66× faster than the teacher proxy's 757.55 ms. In terms of memory footprint, the student model requires only 8GB of VRAM, representing a 82% reduction compared to the 45GB required by the 70B model. These results confirm that our student model offers a highly cost-effective solution suitable for deployment on consumer-grade hardware without compromising efficiency.

4.6 Case Study

To illustrate the impact of different distilled knowledge types on bundle generation, we conducted a case study in the electronic domain (Figure 21). We employed ICL with various distilled knowledge formats on Llama-3.1-8B.[9] In the test session, the user interacts with items including Micro SD Cards, a Carrying Case, an External Hard Drive, a Tablet, Tablet Cases, and a Monitor. The latent user intents are likely twofold: (1) equipping a tablet with protection and (2) organizing digital storage solutions.

For Raw Data, the model correctly identifies the simple Tablet-Cases bundle but misses the storage-related bundle, confirming that raw data lacks the ability to capture complex item relationships. *For*

[8] https://ollama.com/.

[9] We did not use the same knowledge type in both SFT and ICL, as observed in the RQ3 experiment, where using the same knowledge type in both formats did not necessarily lead to positive gains and could interfere with the case study's observations.

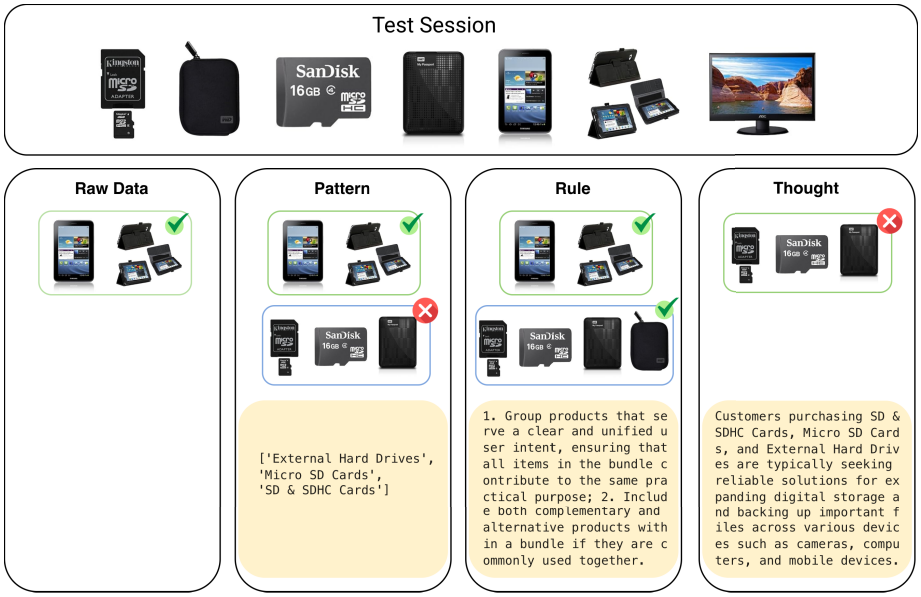


Fig. 21. The generated bundles with different types of distilled knowledge on the Electronic domain.

Pattern Knowledge, the student model recognizes frequent co-occurrences among storage categories [‘External Hard Drives’, ‘Micro SD Cards’, ‘SD & SDHC Cards’] and attempts a storage bundle. However, it incorrectly excludes the Carrying Case, indicating that while Pattern knowledge improves performance through category associations, it lacks the logic to identify complementary items in specific contexts. *For Rule Knowledge*, it performs best by correctly identifying both bundles. Guided by distilled rules, it successfully groups the Carrying Case with storage devices, recognizing their functional complementarity. *For Thought Knowledge*, the model generates a semantically rich explanation about “expanding digital storage” that appears relevant to the user’s intent. However, since the thought is retrieved based on semantic similarity rather than exact session alignment, it does not perfectly match the current items. Consequently, the model fails to produce the correct storage bundle, missing the Carrying Case’s physical compatibility with storage devices. This reveals a key limitation: while Thought knowledge provides useful high-level context, retrieved thoughts from training data may not map precisely to the current session’s specifics.

In summary, this case study highlights the distinct impacts of different types of distilled knowledge on bundle generation. Raw Data provides high precision in detecting simple bundles, but it struggles to uncover more complex item relationships due to the lack of additional guidance. Pattern knowledge enhances the model’s ability to discover item associations based on category co-occurrence, but it may lead to incorrect bundle formations when finer details are necessary. Rule knowledge strikes the best balance by providing explicit, actionable constraints, guiding the model to generate more coherent and contextually appropriate bundles. Thought knowledge, while useful for providing high-level contextual reasoning, may not always align perfectly with the specific session at hand, as relevant thoughts retrieved from the training data based on semantic similarity may not perfectly match the current user intent, leading to incomplete bundle generation.

5 Conclusion and Future Work

In this work, we systematically explore KD techniques for LLM-based bundle generation task, aiming to reduce the significant computational costs associated with large models while preserving

their effectiveness. We propose a comprehensive KD framework featuring progressive knowledge extraction (frequent patterns, formalized rules, deep thoughts), diverse strategies to vary knowledge quantity (sampling, domain/format accumulation), and complementary LLM adaptation techniques (ICL, SFT and SFT+ICL) along with their combination for knowledge utilization. Our extensive experiments across the three real-world bundle datasets demonstrate that KD is indeed a viable and potent strategy. The effectiveness is nuanced, depending significantly on the interplay between the format of distilled knowledge, its quantity, and the method used to utilize it, as well as the specific characteristics of the dataset domain. To be specific, SFT with distilled knowledge consistently emerged as a strong approach, enabling smaller student models to achieve performance comparable, and on certain metrics (Precision, Coverage) even superior, to the large teacher model. The combined SFT+ICL approach often yields the best results, depending on careful selection of knowledge for each stage. Importantly, these results are obtained with significantly lower computational cost compared to the teacher LLM. Overall, our findings highlight the potential of efficient LLM-based solutions for complex tasks like bundle generation and offer practical insights for optimizing KD strategies in future work.

Limitations and Future Work. While this work demonstrates the efficiency and effectiveness of explicit KD, it primarily focuses on transferring knowledge expressible in textual formats (patterns, rules, thoughts). This explicit approach may not fully capture the rich, implicit knowledge residing within the internal states (e.g., hidden representations, attention weights) of the teacher LLM. To address these limitations and further improve the robustness of knowledge-distilled students, we highlight three promising directions for future work.

First, beyond purely explicit KD, it is natural to investigate hybrid pipelines that combine explicit and implicit signals when teacher internals are partially accessible. With open source teacher LLMs, one could perform joint or two-stage training in which the student first learns from teacher-generated knowledge bundles (explicit KD) and is then further aligned by matching teacher logits or intermediate representations (implicit KD) on the same queries. In more realistic deployment scenarios where commercial teacher LLMs only expose API-level outputs but open source teachers are available locally, commercial models could provide high-quality explicit distillation while open source teachers supply complementary logit- or hidden-state-level supervision, effectively acting as an implicit regularizer. We expect explicit KD to remain advantageous for its flexibility in encoding diverse knowledge formats and compatibility with black-box APIs, whereas implicit KD is better suited for transferring fine-grained inductive biases and calibration; designing principled combinations of these two paradigms is thus an important avenue for future work.

Second, our experiments reveal that the selection and combination of different knowledge formats and quantities significantly impact performance, especially for the combined SFT+ICL utilization method. Simply using all available knowledge is not always optimal. This highlights the need for more sophisticated knowledge selection and fusion mechanisms. Future research could explore adaptive or model-driven approaches that automatically determine the most beneficial subset, weighting, or composition of knowledge bundles for a given task, domain, or utilization strategy, thereby improving robustness and maximizing the potential of the student model.

Last, we currently restrict ourselves to textual knowledge distilled from LLMs, whereas many practical bundle generation scenarios involve multi-modal information such as item images. Extending these KD techniques to incorporate multi-modal data and to distill from vision–language or other multi-modal teachers is an important direction for enhancing the effectiveness and applicability of the proposed framework in real-world recommendation and content generation tasks.

References

[1] Rakesh Agrawal and Ramakrishnan Srikant. 1994. Fast algorithms for mining association rules. In *Proceedings of 20th International Conference on Very Large Scale Data Bases (VLDB)*, 487–499.

[2] Tzoof Avny Brosh, Amit Livne, Oren Sar Shalom, Bracha Shapira, and Mark Last. 2022. BRUCE: Bundle recommendation using contextualized item embeddings. In *Proceedings of the 16th ACM Conference on Recommender Systems (RecSys)*, 237–245.

[3] Jinze Bai, Chang Zhou, Junshuai Song, Xiaoru Qu, Weiting An, Zhao Li, and Jun Gao. 2019. Personalized bundle list recommendation. In *The Web Conference (WWW)*, 60–71.

[4] Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems (RecSys)*, 1007–1014.

[5] Moran Beladev, Lior Rokach, and Bracha Shapira. 2016. Recommender systems for product bundling. *Knowledge-Based Systems* 111 (2016), 193–206.

[6] Da Cao, Liqiang Nie, Xiangnan He, Xiaochi Wei, Shunzhi Zhu, and Tat-Seng Chua. 2017. Embedding factorization models for jointly recommending items and user generated lists. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 585–594.

[7] Jianxin Chang, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. 2020. Bundle recommendation with graph convolutional networks. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 1673–1676.

[8] Jianxin Chang, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. 2021. Bundle recommendation and generation with graph neural networks. *IEEE Transactions on Knowledge and Data Engineering* 35, 3 (2021), 2326–2340.

[9] Liang Chen, Yang Liu, Xiangnan He, Lianli Gao, and Zibin Zheng. 2019. Matching user with item set: Collaborative bundle recommendation with deep attention network. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2095–2101.

[10] Wen Chen, Pipei Huang, Jiaming Xu, Xin Guo, Cheng Guo, Fei Sun, Chao Li, Andreas Pfadler, Huan Zhao, and Binqiang Zhao. 2019. POG: Personalized outfit generation for fashion recommendation at Alibaba iFashion. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2662–2670.

[11] Yu Cui, Feng Liu, Pengbo Wang, Bohao Wang, Heng Tang, Yi Wan, Jun Wang, and Jiawei Chen. 2024. Distillation matters: Empowering sequential recommenders to match the performance of large language models. In *Proceedings of the 18th ACM Conference on Recommender Systems (RecSys)*, 507–517.

[12] Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. 2023. Uncovering chatgpt's capabilities in recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems (RecSys)*, 1126–1132.

[13] Qilin Deng, Kai Wang, Minghao Zhao, Runze Wu, Yu Ding, Zhene Zou, Yue Shang, Jianrong Tao, and Changjie Fan. 2021. Build your own bundle—A neural combinatorial optimization method. In *Proceedings of the 29th ACM International Conference on Multimedia (MM)*, 2625–2633.

[14] Qilin Deng, Kai Wang, Minghao Zhao, Zhene Zou, Runze Wu, Jianrong Tao, Changjie Fan, and Liang Chen. 2020. Personalized bundle recommendation in online games. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM)*, 2381–2388.

[15] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLORA: Efficient finetuning of quantized llms. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (NeurIPS '23)*, 10088–10115.

[16] Paolo Dragone, Giovanni Pellegrini, Michele Vescovi, Katya Tentori, and Andrea Passerini. 2018. No more ready-made deals: Constructive recommendation for telco service bundling. In *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys)*, 163–171.

[17] Yan Fang, Xinyue Xiao, Xiaoyu Wang, and Huiqing Lan. 2018. Customized bundle recommendation by association rules of product categories for online supermarkets. In *Proceedings of the 3rd International Conference on Data Science in Cyberspace (DSC)*, 472–475.

[18] Robert Garfinkel, Ram Gopal, Arvind Tripathi, and Fang Yin. 2006. Design of a shopbot and recommender system for bundle purchases. *Decision Support Systems* 42, 3 (2006), 1974–1986.

[19] Yong Ge, Hui Xiong, Alexander Tuzhilin, and Qi Liu. 2014. Cost-aware collaborative filtering for travel tour recommendations. *ACM Transactions on Information Systems* 32, 1 (2014), 1–31.

[20] Judy Harris and Edward A. Blair. 2006. Consumer preference for product bundles: The role of reduced search costs. *Journal of the Academy of Marketing Science* 34, 4 (2006), 506–513.

[21] Jesse Harte, Wouter Zorgdrager, Panos Louridas, Asterios Katsifodimos, Dietmar Jannach, and Marios Fragkoulis. 2023. Leveraging large language models for sequential recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems (RecSys)*, 1096–1102.

[22] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web (WWW)*, 507–517.

[23] Yun He, Jianling Wang, Wei Niu, and James Caverlee. 2019. A hierarchical self-attentive model for recommending user-generated item lists. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*, 1481–1490.

[24] Yun He, Yin Zhang, Weiwen Liu, and James Caverlee. 2020. Consistency-aware recommendation for user-generated item list continuation. In *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM)*, 250–258.

[25] Zhankui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Majumder, Nathan Kallus, and Julian Mcauley. 2023. Large language models as zero-shot conversational recommenders. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM)*, 720–730.

[26] Zhankui He, Handong Zhao, Tong Yu, Sungchul Kim, Fan Du, and Julian McAuley. 2022. Bundle mcr: Towards conversational bundle recommendation. In *Proceedings of the 16th ACM Conference on Recommender Systems (RecSys)*, 288–298.

[27] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. arXiv:1503.02531. Retrieved from https://arxiv.org/abs/1503.02531

[28] Hyunsik Jeon, Jong-Eun Lee, Jeongin Yun, and U. Kang. 2024. Cold-start bundle recommendation via popularity-based coalescence and curriculum heating. In *Proceedings of the ACM Web Conference 2024 (WWW)*, 3277–3286.

[29] Pigi Kouki, Ilias Fountalis, Nikolaos Vasiloglou, Nian Yan, Unaiza Ahsan, Khalifeh Al Jadda, and Huiming Qu. 2019. Product collection recommendation in online retail. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys)*, 486–490.

[30] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-Tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.

[31] Jianghao Lin, Rong Shan, Chenxu Zhu, Kounianhua Du, Bo Chen, Shigang Quan, Ruiming Tang, Yong Yu, and Weinan Zhang. 2024. Rella: Retrieval-enhanced large language models for lifelong sequential behavior comprehension in recommendation. In *Proceedings of the ACM Web Conference 2024 (WWW)*, 3497–3508.

[32] Guannan Liu, Yanjie Fu, Guoqing Chen, Hui Xiong, and Can Chen. 2017. Modeling buying motives for personalized product bundle recommendation. *ACM Transactions on Knowledge Discovery from Data* 11, 3 (2017), 1–26.

[33] Qi Liu, Yong Ge, Zhongmou Li, Enhong Chen, and Hui Xiong. 2011. Personalized travel package recommendation. In *IEEE 11th International Conference on Data Mining (ICDM)*, 407–416.

[34] Xiaohao Liu, Jie Wu, Zhulin Tao, Yunshan Ma, Yinwei Wei, and Tat-Seng Chua. 2025. Fine-tuning multimodal large language models for product bundling. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 848–858.

[35] Yidan Liu, Min Xie, and Laks V. S. Lakshmanan. 2014. Recommending user generated item lists. In *Proceedings of the 8th ACM Conference on Recommender Systems (RecSys)*, 185–192.

[36] Yunshan Ma, Yingzhi He, Xiang Wang, Yinwei Wei, Xiaoyu Du, Yuyangzi Fu, and Tat-Seng Chua. 2024. Multicbr: Multi-view contrastive learning for bundle recommendation. *ACM Transactions on Information Systems* 42, 4 (2024), 1–23.

[37] Yunshan Ma, Yingzhi He, An Zhang, Xiang Wang, and Tat-Seng Chua. 2022. CrossCBR: Cross-view contrastive learning for bundle recommendation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 1233–1241.

[38] Yunshan Ma, Yingzhi He, Wenjun Zhong, Xiang Wang, Roger Zimmermann, and Tat-Seng Chua. 2024. CIRP: Cross-item relational pre-training for multimodal product bundling. In *Proceedings of the 32nd ACM International Conference on Multimedia (MM)*, 9641–9649.

[39] Yunshan Ma, Xiaohao Liu, Yinwei Wei, Zhulin Tao, Xiang Wang, and Tat-Seng Chua. 2024. Leveraging multimodal features and item-level user feedback for bundle construction. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining (WSDM)*, 510–519.

[40] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 46534–46594.

[41] Apurva Pathak, Kshitiz Gupta, and Julian McAuley. 2017. Generating and personalizing bundle recommendations on steam. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 1073–1076.

[42] Yuyang Ren, Zhang Haonan, Luoyi Fu, Xinbing Wang, and Chenghu Zhou. 2023. Distillation-enhanced graph masked autoencoders for bundle recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 1660–1669.

[43] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI)*, 452–461.

[44] Alireza Salemi, Surya Kallumadi, and Hamed Zamani. 2024. Optimization methods for personalizing large language models through retrieval augmentation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 752–762.

[45] Oren Sar Shalom, Noam Koenigstein, Ulrich Paquet, and Hastagiri P. Vanchinathan. 2016. Beyond collaborative filtering: The list recommendation problem. In *Proceedings of the 25th International Conference on World Wide Web (WWW)*, 63–72.

[46] Wenqi Sun, Ruobing Xie, Junjie Zhang, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2024. Distillation is all you need for practically using different pre-trained recommendation models. arXiv:2401.00797v1. Retrieved from https://arxiv.org/html/2401.00797v1

[47] Zhu Sun, Kaidong Feng, Jie Yang, Hui Fang, Xinghua Qu, Yew-Soon Ong, and Wenyuan Liu. 2024. Revisiting bundle recommendation for intent-aware product bundling. *ACM Transactions on Recommender Systems* 2, 3, Article 24 (2024), 24:1–24:34.

[48] Zhu Sun, Kaidong Feng, Jie Yang, Xinghua Qu, Hui Fang, Yew-Soon Ong, and Wenyuan Liu. 2024. Adaptive in-context learning with large language models for bundle generation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 966–976.

[49] Zhu Sun, Hongyang Liu, Xinghua Qu, Kaidong Feng, Yan Wang, and Yew Soon Ong. 2024. Large language models for intent-driven session recommendations. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 324–334.

[50] Zhu Sun, Jie Yang, Kaidong Feng, Hui Fang, Xinghua Qu, and Yew Soon Ong. 2022. Revisiting bundle recommendation: Datasets, tasks, challenges and opportunities for intent-aware product bundling. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2900–2911.

[51] Wenjie Wang, Xinyu Lin, Fuli Feng, Xiangnan He, and Tat-Seng Chua. 2023. Generative recommendation: Towards next-generation recommender paradigm. arXiv:2304.03516. Retrieved from https://arxiv.org/abs/2304.03516

[52] Xinfeng Wang, Jin Cui, Yoshimi Suzuki, and Fumiyo Fukumoto. 2024. RDRec: Rationale distillation for LLM-based recommendation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 65–74.

[53] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 12648–12671.

[54] Yan Wang, Zhixuan Chu, Xin Ouyang, Simeng Wang, Hongyan Hao, Yue Shen, Jinjie Gu, Siqiao Xue, James Y. Zhang, Qing Cui, et al. 2023. Enhancing recommender systems with large language model reasoning graphs. arXiv:2308.10835. Retrieved from https://arxiv.org/abs/2308.10835

[55] Yuling Wang, Changxin Tian, Binbin Hu, Yanhua Yu, Ziqi Liu, Zhiqiang Zhang, Jun Zhou, Liang Pang, and Xiao Wang. 2024. Can small language models be good reasoners for sequential recommendation? In *Proceedings of the ACM Web Conference 2024 (WWW)*, 3876–3887.

[56] Penghui Wei, Shaoguo Liu, Xuanhua Yang, Liang Wang, and Bo Zheng. 2022. Towards personalized bundle creative generation with contrastive non-autoregressive decoding. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2634–2638.

[57] Yinwei Wei, Xiaohao Liu, Yunshan Ma, Xiang Wang, Liqiang Nie, and Tat-Seng Chua. 2023. Strategy-aware bundle recommender system. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 1198–1207.

[58] Chuhan Wu, Fangzhao Wu, Yang Yu, Tao Qi, Yongfeng Huang, and Qi Liu. 2021. NewsBERT: Distilling pre-trained language model for intelligent news application. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 3285–3295.

[59] Likang Wu, Zhaopeng Qiu, Zhi Zheng, Hengshu Zhu, and Enhong Chen. 2024. Exploring large language model for graph data understanding in online job recommendations. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*, 9178–9186.

[60] Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, et al. 2023. A survey on large language models for recommendation. arXiv:2305.19860. Retrieved from https://arxiv.org/abs/2305.19860

[61] Min Xie, Laks V. S. Lakshmanan, and Peter T. Wood. 2010. Breaking out of the box of recommendations: From items to packages. In *Proceedings of the 4th ACM Conference on Recommender Systems (RecSys)*, 151–158.

[62] Min Xie, Laks V. S. Lakshmanan, and Peter T. Wood. 2014. Generating top-k packages via preference elicitation. *Proceedings of the VLDB Endowment* 7, 14 (2014), 1941–1952.

[63] Yiyan Xu, Wenjie Wang, Fuli Feng, Yunshan Ma, Jizhi Zhang, and Xiangnan He. 2024. Diffusion models for generative outfit recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 1350–1359.

[64] De-Nian Yang, Wang-Chien Lee, Nai-Hui Chia, Mao Ye, and Hui-Ju Hung. 2012. On bundle configuration for viral marketing in social networks. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM)*, 2234–2238.

[65] Yang Yu, Fangzhao Wu, Chuhan Wu, Jingwei Yi, and Qi Liu. 2021. Tiny-newsrec: Effective and efficient plm-based news recommendation. arXiv:2112.00944. Retrieved from https://arxiv.org/abs/2112.00944

[66] Xu Yuan, Hongshen Chen, Yonghao Song, Xiaofang Zhao, Zhuoye Ding, Zhen He, and Bo Long. 2021. Improving sequential recommendation consistency with self-supervised imitation. arXiv:2106.14031. Retrieved from https://arxiv.org/abs/2106.14031

[67] Jianyang Zhai, Xiawu Zheng, Chang-Dong Wang, Hui Li, and Yonghong Tian. 2023. Knowledge prompt-tuning for sequential recommendation. In *Proceedings of the 29th ACM International Conference on Multimedia (MM)*, 6451–6461.

[68] Jizhi Zhang, Keqin Bao, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Is chatgpt fair for recommendation? Evaluating fairness in large language model recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems (RecSys)*, 993–999.

[69] Sen Zhao, Wei Wei, Ding Zou, and Xianling Mao. 2022. Multi-view intent disentangle graph networks for bundle recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 4379–4387.

[70] Tao Zhu, Patrick Harrington, Junjun Li, and Lei Tang. 2014. Bundle recommendation in ecommerce. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 657–666.

[71] Ding Zou, Sen Zhao, Wei Wei, Xian-Ling Mao, Ruixuan Li, Dangyang Chen, Rui Fang, and Yuanyuan Fu. 2023. Towards hierarchical intent disentanglement for bundle recommendation. *IEEE Transactions on Knowledge and Data Engineering* 36, 7 (2023), 3556–3567.

Received 23 April 2025; revised 18 February 2026; accepted 4 April 2026