Attn-GS: Attention-Guided Context Compression for Efficient Personalized LLMs

Shenglai Zeng[1*], **Tianqi Zheng**[2], **Chuan Tian**[2], **Dante Everaert**[2], **Yau-Shian Wang**[2], **Yupin Huang**[2], **Michael J. Morais**[2], **Rohit Patki**[2], **Jinjin Tian**[2], **Xinnan Dai**[1], **Kai Guo**[1†], **Monica Xiao Cheng**[2], **Hui Liu**[1]

[1]Michigan State University [2]Amazon.com

Abstract

Personalizing large language models (LLMs) to individual users requires incorporating extensive interaction histories and profiles, but input token constraints make this impractical due to high inference latency and API costs. Existing approaches rely on heuristic methods such as selecting recent interactions or prompting summarization models to compress user profiles. However, these methods treat context as a monolithic whole and fail to consider how LLMs internally process and prioritize different profile components. We investigate whether LLMs' attention patterns can effectively identify important personalization signals for intelligent context compression. Through preliminary studies on representative personalization tasks, we discover that (a) LLMs' attention patterns naturally reveal important signals, and (b) fine-tuning enhances LLMs' ability to distinguish between relevant and irrelevant information. Based on these insights, we propose **Attn-GS**, an attention-guided context compression framework that leverages attention feedback from a marking model to mark important personalization sentences, then guides a compression model to generate task-relevant, high-quality compressed user contexts. Extensive experiments demonstrate that Attn-GS significantly outperforms various baselines across different tasks, token limits, and settings, achieving performance close to using full context while reducing token usage by 50 ×.

1 Introduction

Large language models have been widely adopted across various applications, serving billions of users worldwide. In many applications such as query autocompletion (Zhou et al., 2024; Everaert et al., 2024), customer service (Agarwal et al., 2022), and content-based recommendation agents(Yang et al., 2023; Zhang et al., 2024a), it is essential to adapt general LLMs into personalized models that can *generate responses tailored to the unique needs and preferences of individual users*. The general personalization task can be formalized as shown in Figure 1. Based on user profiles and interaction histories, the model is expected to understand user preferences and provide personalized responses to given tasks that align with each user's specific needs. However, a fundamental challenge in personalization tasks is the constraint imposed by input token limits (Zhu et al., 2025; Shi et al., 2025). As users accumulate extensive interaction histories, directly inputting all contextual information into LLMs for personalization (Path 1 in Figure 1) can result in high inference latency and API costs, making it impractical for real-world deployment(Zhu et al., 2025; Everaert et al., 2024; Wang et al., 2025a). Moreover, not all signals are relevant to specific tasks, and naively incorporating more signals into long contexts may not always yield benefits (Lyu et al., 2023). Therefore, as shown in the path 2 in Figure 1, compressing extensive user histories and profiles offline into a condensed context within token limits while maintaining acceptable downstream performance becomes essential. Consequently, it is crucial to identify which personalization signals are truly important for the task and intelligently select or compress these signals to satisfy context length requirements.

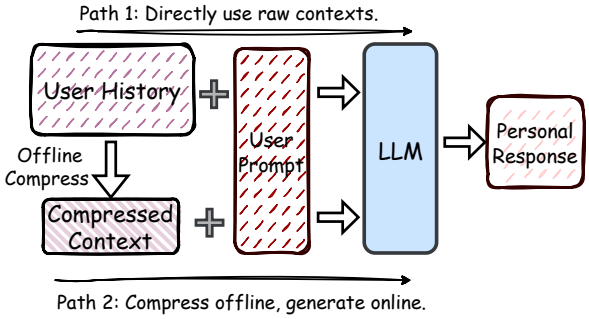


Figure 1: Personalized LLMs.

Current solutions primarily rely on heuristic or

[*]Work done during Shenglai Zeng's <zengshe1@msu.edu> internship at Amazon.

[†]**Correspondence to Kai Guo <guokai1@msu.edu>.**

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 35925–35946

naive methods to address context limitation problems, such as selecting recent interactions (Everaert et al., 2024) or prompting summarization LLMs (Liu et al., 2024; Zhang et al., 2024a) to compress user profiles. However, these approaches treat the context as a monolithic whole and fail to consider how LLMs internally process and prioritize different profile components. Since LLMs naturally exhibit selective attention (He et al., 2025; Olsson et al., 2022; Gould et al., 2023) when processing input sequences—focusing more on relevant information while down-weighting less pertinent details—understanding these internal mechanisms could reveal which personalization signals the model **actually utilizes for generating responses**. Meanwhile, it is evident from other domains, such as evidence-based question answering, has demonstrated the crucial role of LLMs' internal representations (Zeng et al., 2025a) and attention patterns (Liu et al., 2025), successfully leveraging these mechanisms to enhance performance. Inspired from the insights , we investigate whether LLMs' attention behaviors can effectively identify important personalization signals and whether attention patterns can enable novel context compression strategies.

To answer these questions, we first conduct preliminary studies to examine attention weight distributions on representative personalization tasks in Section 3. We find that (a) LLMs' attention patterns naturally unveil important signals, and (b) fine-tuning enables LLMs to better distinguish between important and unimportant signals. Based on these findings, we develop an attention-guided context compression framework, *Attn-GS* (Section 4), to generate high-quality, task-relevant compressed profiles for downstream use. Specifically, we first utilize attention feedback from a small white-box marking model to identify important personalization sentences, then guide a compression model with these attention-marked contexts to generate task-relevant, high-quality compressed user contexts. Experiments (Section 5) demonstrate that Attn-GS generates superior compressed profiles that outperform various baselines across different tasks, token limits and scenarios (inference-only and training-inference settings), achieving performance close to using full context while reducing token usage by 50×.

2 Related Work

2.1 Personalized LLMs

LLM personalization refers to adapting Large Language Models to individual user preferences and contexts to deliver tailored responses. This paradigm has great potential across applications including education (Wang et al., 2024), healthcare (Yu et al., 2024), recommendation systems (Yang et al., 2023), and search (Zhou et al., 2024). Researchers employ various methods such as Retrieval-Augmented Generation (RAG) (Zhao et al., 2024; Rajput et al., 2023; Zeng et al., 2024, 2025a,b), prompting (Jiang et al., 2023; Serapio-García et al., 2023), and fine-tuning (Li and Zhao, 2021; Zeng et al., 2025a). However, a key challenge lies in handling extensive user contexts: directly inputting them incurs prohibitive inference latency and API costs. Current approaches rely on either rule-based heuristics (e.g., using recent interactions or single signal types (Dai et al., 2023; Liu et al., 2023; Everaert et al., 2024)) or LLM-based methods leveraging summarization (Liu et al., 2024; Zhang et al., 2024a) and self-reflection (Zhang et al., 2024b; Wang et al., 2025b) to condense contexts. Yet significant gaps remain in understanding how LLMs internally process and prioritize different profile components.

2.2 Utilization of LLMs' internal signals

Recently, an emerging research direction focuses on utilizing LLMs' internal signals to understand how LLMs process contexts (Halawi et al., 2023; Chen et al., 2024; Li et al., 2024). In question answering, researchers have shown that LLMs' internal representations (Zeng et al., 2025b) can identify high-level concepts in RAG systems, such as context helpfulness, and can be controlled (Zeng et al., 2025a) to enhance RAG robustness. Moreover, Liu et al. (2025) demonstrate that LLMs possess "evidence-seeking layers" that identify evidence in QA tasks, and that attention scores can improve extractive QA performance. However, existing work predominantly focuses on QA tasks that extract explicit answers from context, while personalization requires synthesizing diverse user signals to generate tailored responses—a fundamentally different mechanism. Whether LLMs' internal representations can identify important personalization signals and facilitate context compression remains unexplored. We extend this exploration to the personalization domain.

3 Preliminary Studies

In this section, we conduct preliminary studies on the inherent abilities of LLMs' attention mechanisms to distinguish between important and unimportant personalization signals. Specifically, we examine the distribution of LLMs' attention scores across different types of context signals to explore whether they can effectively differentiate important from unimportant signals. We first introduce the problem description and notations in Section 3.1, followed by our preliminary findings in Section 3.2.

3.1 Problem Description & Notations

In this subsection, we introduce the problem setting and notations used in our paper. Specifically, given a user's personalization context and a task description for the personalized LLM, we examine the model's attention patterns across its context. Suppose that we have a long user history/context H (e.g., the user's movie interaction history or writing history) and a task description $\mathcal{T}$ (e.g., recommend next movie or suggest title). We input these to a marking LLM Φ_{Mark}. Following previous work (), we focus on the attention scores of the last input tokens for the user context, as these directly contribute to the model's answer.

Token-level attention scores. Suppose that there are N input tokens in total. We calculate the attention scores of the last token to the input sequence at the layer d. Denote the token-level attention probability vector across input tokens of layer d and head j as $\mathbf{w}^{(d,j)} := [w_1^{(d,j)}, w_2^{(d,j)}, \ldots, w_N^{(d,j)}] \in \mathbb{R}^N$. The average attention scores across all heads can be defined as:

$$\tilde{\mathbf{w}}^{(d)} = \frac{1}{J} \sum_{j=1}^{J} \mathbf{w}^{(d,j)} \tag{1}$$

where J is the number of attention heads and $\mathbf{w}^{(d,j)} \in \mathbb{R}^N$ is the attention vector for head j. Thus, $\tilde{\mathbf{w}}^{(d)} := [\tilde{w}_1^{(d)}, \tilde{w}_2^{(d)}, \ldots, \tilde{w}_N^{(d)}] \in \mathbb{R}^N$.

Sentence-level attention scores. After obtaining the token-level attention scores $\tilde{\mathbf{w}}^{(d)}$, we aggregate and average these scores across sentences to derive sentence-level attention scores. For sentence u_i spanning tokens from position $p_{start}^{u_i}$ to $p_{end}^{u_i}$, the sentence-level attention score is:

$$\hat{w}_i^{(d)} = \frac{1}{p_{end}^{u_i} - p_{start}^{u_i} + 1} \sum_{m=p_{start}^{u_i}}^{p_{end}^{u_i}} \tilde{w}_m^{(d)} \tag{2}$$

Signal-level attention scores. To systematically analyze which personalization signals matter most, we categorize user behavioral data into distinct signal types $\mathcal{L}_s = \{\tau_1, \tau_2, \ldots, \tau_K\}$ (e.g., titles, user ratings, user reviews), where K is the number of signal types. Each sentence u_i in the user context is assigned to exactly one signal type $\tau_k \in \mathcal{S}$. To reveal which signal types receive the most attention, we compute type-wise average scores:

$$\hat{w}_{\tau_k}^{(d)} = \frac{1}{|\mathcal{U}_{\tau_k}|} \sum_{u_i \in \mathcal{U}_{\tau_k}} \hat{w}_i^{(d)} \tag{3}$$

where $\mathcal{U}_{\tau_k}$ represents all sentences of type τ_k.

3.2 Key Findings

To explore whether LLMs' attention scores reflect personalization signal importance, we conduct a study on 2 representative personalization datasets. Specifically, we examine whether the model assigns higher attention to important signal types and lower attention to unimportant ones.

Datasets. We conduct this study on two datasets: **(a) MovieLens-1M** (Harper and Konstan, 2015) for movie recommendation. Signal types withine contexts include movie titles, user ratings, summaries, genres, watch times, release years, and users' basic information(age, gender, occupation). Following prior content-based recommendation work (Zhang et al., 2024b,a), the task is to select the user's last interaction from among four random negatives, with all previous interactions serving as personalized context. We use 1,000 users for training the Mark model and 1,000 users for testing and attention visualization. **(b) LaMP-5** (Salemi et al., 2024) for personalized title generation. Contexts contain authors' previous papers, including titles, abstracts, and publication dates. The task is to generate personalized paper titles given an abstract and the author's writing history. We use 750 users for training the Mark model and 750 users for testing and attention visualization. Detailed dataset descriptions and examples are provided in Appendix A.1.

Experimental Setup. As introduced in Section 3.1, we input the user personalized context H and task description $\mathcal{T}$ of each dataset into the mark model Φ_{Mark} to obtain attention weights $\hat{w}_{\tau_k}^{(d)}$ for different signal types. For MovieLens, we use both the original Llama-3.1-1B-Instruct and its fine-tuned version as mark models. For LaMP-5, we employ both the original Llama-3.1-8B-Instruct

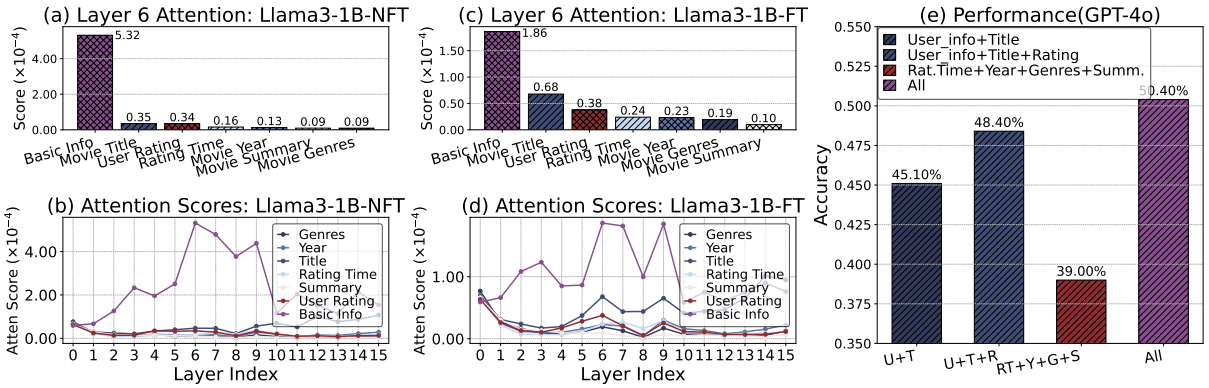


Figure 2: Attention visualization on MovieLens dataset. Layer-6 attention and cross-layer attention for non-fine-tuned Φ_{Mark} (a-b) and fine-tuned Φ_{Mark} (c-d), and performance comparison across signal subsets (e). U: User Basic Info, T: Title, R: User Rating, RT: Rating Time, Y: Movie Year, G: Genre, S: Movie Summary, All: All signals.

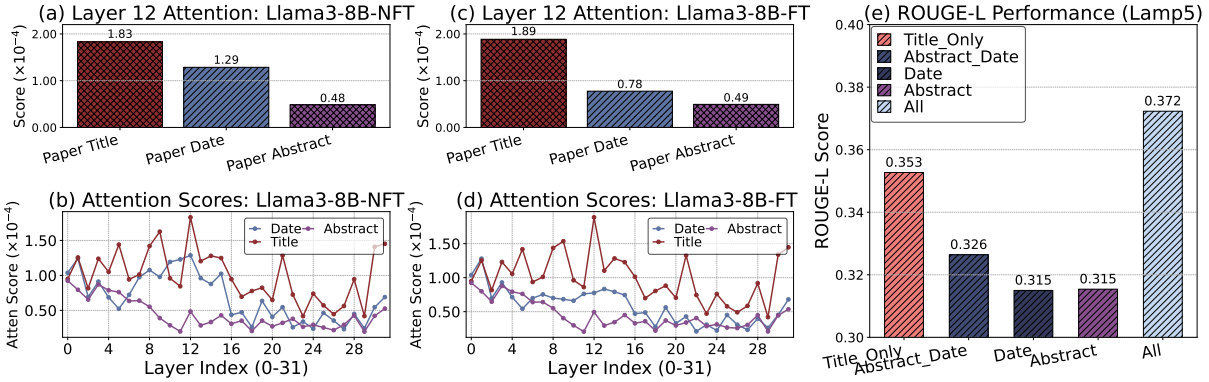


Figure 3: Attention visualization on LaMP-5 dataset. Layer-12 attention and cross-layer attention for non-fine-tuned Φ_{Mark} (a-b) and fine-tuned Φ_{Mark} (c-d), and performance comparison across signal subsets (e).

and its fine-tuned version as mark models. Note that the fine-tuned Φ_{Mark} is obtained by fine-tuning the original model on the training set following standard practices, where the input consists of the concatenation of H, $\mathcal{T}$, and the actual question (i.e., candidate movies for MovieLens or given abstract for LaMP-5), with the ground-truth answer as the output target. To validate the correlation between attention scores and signal importance, we also evaluate performance using only subsets of signals (important signals only vs. unimportant signals only) with GPT-4o as the generator. Performance is measured by accuracy for MovieLens-1M and ROUGE-L for LaMP-5.

Results & findings. The results for the MovieLens dataset are shown in Figure 2. Figures 2(b) and 2(d) show signal-level attention distributions across layers for different signal types, while Figures 2(a) and 2(c) display the attention distributions at layer 6. Attention scores begin to diverge in the first few layers and exhibit relatively large differences in the early-to-middle layers (e.g., layer 6). For the original model (Figures 2(a) and 2(b)), the model primarily attends to users' basic information while assigning low attention to other signal types, with slightly higher attention to movie titles and user ratings. In contrast, the finetuned model(Figures 2(c) and 2(d)) demonstrates clearer discriminative attention across signal types. While user basic information still receives the highest attention (though reduced compared to the original model), among the remaining signals, movie titles receive substantially higher attention, followed by user ratings, with other signals receiving notably less attention. This attention pattern aligns closely with the performance results in Figure 2(e), where using only the important signals identified by attention scores achieves significantly higher accuracy (Basic+Title [U+T]: 45.1%, Basic+Title+Rating [U+T+R]: 48.4%) compared to using all low-attention signals (Rating Time+Year+Genre+Summary [RT+Y+G+S]: 39.0%).

We observe similar patterns on the LaMP-5 dataset in Figure 3. The attention scores show large differences in the early-to-middle layers (e.g., layer

12). The original model assigns high attention to paper titles and publication dates (with minimal distinction between them) and low attention to abstracts. For the finetuned model, title signals (important signals) become dominantly higher than the other two signal types (unimportant signals). This pattern better aligns with the performance results in Figure 3(e), where using only title information yields clearly higher performance than using paper dates, abstracts, or both together.

These results preliminarily demonstrate that *(a) LLMs' attention scores can naturally reveal the importance of personalization signals* and that *(b) finetuning enhances the model's ability to distinguish between important and unimportant signals*.[1] This finding motivates us to utilize attention scores to mark important signals for better summarization.

4 Method

Inspired by the above findings, we propose an attention-guided summarization pipeline *Attn-GS* to utilize LLMs' attention feedback for generating high-quality compressed user contexts. The overall framework is illustrated in Figure 4 and consists of two stages: (1) **Critical Personalization Sentence Marking**, which leverages attention feedback from a white-box marking model Φ_{Mark} to identify and highlight important personalization sentences based on attention scores, and (2) **Summarization Based on Marked Context**, which utilizes a summary model Φ_{Sum} that takes the marked context as input to generate a compressed profile within a predefined token limit. We introduce these two stages in detail in Sections 4.1 and 4.2, respectively, and present the full pipeline in Section 4.3[2].

4.1 Critical Personalization Sentence Marking

To identify which personalization sentences contain critical information, we first input the user's history context concatenated with the task description into the marking model Φ_{Mark} to obtain token-level attention scores $\tilde{\mathbf{w}}^{(d)}$, as shown in Eq.(1):

$$\tilde{\mathbf{w}}^{(d)} \leftarrow \text{AttnScores}(\Phi_{\text{Mark}}(H, \mathcal{T}), d) \quad (4)$$

Next, we aggregate the token-level attention scores into sentence-level attention scores $\hat{w}_i^{(d)}$ for every sentence $u_i \in H$. We then select sentences with relatively high attention scores using a threshold-based filtering mechanism. Specifically, given a threshold factor $\alpha \in (0, 1]$, we identify the set of important sentences as:

$$\mathcal{S}_{\text{Mark}} = \left\{ u_i \in H \mid \hat{w}_i^{(d)} \geq \alpha \cdot \max_{u_j \in H} \hat{w}_j^{(d)} \right\} \quad (5)$$

where sentences whose attention scores exceed α times the maximum attention score are marked as important.

After identifying these important personalization sentences, we explicitly mark them to obtain a modified version of the user context, denoted as H^*. Following previous practices (Liu et al., 2025), we add markers `<start_important>` before each important sentence and `<end_important>` after it. The entire marking process can be expressed as:

$$H^* \leftarrow \text{Mark}(\mathcal{S}_{\text{Mark}}, H) \quad (6)$$

This marking helps the subsequent summary model become aware of and prioritize these personalization signals during summarization.

4.2 Summarization Based on Marked Context

After obtaining the marked version of the user context H^* with highlighted personalization sentences $\mathcal{S}_{\text{Mark}}$, we generate a compressed user profile G within a token limit of m tokens. We input H^* into the summary model Φ_{Sum} with a prompt that explicitly instructs the model to prioritize these important sentences when generating the compressed profile. Example prompt templates can be seen in Appendix A.2.1. The overall summarization process can be expressed as: $\text{G} \leftarrow \Phi_{\text{Sum}}(H^*, m)$.

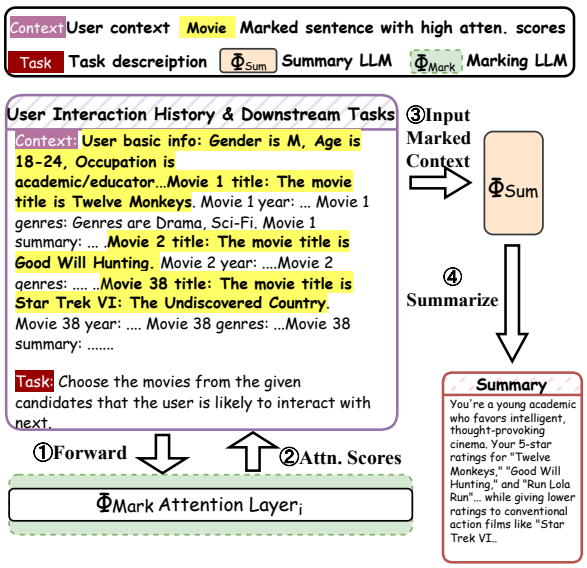


Figure 4: An illustration of Attn-GS framework.

[1] This claim is further validated in Section 5.4.

[2] Examples of Marked Sentence and summarized results can be found in Appendix A.2.2 and A.2.3

Algorithm 1 Attn-GS Algorithm

Input: Language Models Φ_{Mark} and Φ_{Sum}, User History H, Task Description $\mathcal{T}$, Attention Layers $\mathcal{L}_{\text{Attn}}$, Target Layer Index d, Threshold α, Token limits m. **Output:** User Profile Summary G.

// Step 1: Compute Attention Scores
$\mathbf{w}^{(d)} \leftarrow \text{AttnScores}(\Phi_{\text{Mark}}(H, \mathcal{T}), d)$ ▷ Eq.(4)
// Step 2: Select Key Information
$S_{\text{Mark}} \leftarrow \text{SelectSentences}(\mathbf{w}^{(d)}, H, \alpha)$ ▷ Select S_{Mark} with high attn scores (Eq.(5))
// Step 3: Construct Input Context
$H^* \leftarrow \text{Mark}(S_{\text{Mark}}, H)$ ▷ highlighting S_{Mark}
// Step 4: Generate Summary with Sum LLM
$G \leftarrow \Phi_{\text{Sum}}(H^*, m)$
return G

4.3 Attn-GS Algorithm

The framework's process can be summarized in Algorithm 1. First, we input the user interaction history and task description to the marking model Φ_{Mark} to obtain attention scores(step 1) and identify(step 2) and mark(step 3) critical personalization sentences with high attention scores. Next, we input the marked context to the summary model to generate high-quality, task-relevant compressed context(step 4). This entire process can be conducted offline. During inference, the compressed context can be combined with user real-time queries(e.g, movie candidates) as input to generator Φ_{G} to reduce inference latency and costs.

5 Experiment

In this section, we conduct comprehensive experiments to validate the effectiveness of our proposed Attn-GS method. We first introduce our experimental settings in Section 5.1, then compare the performance of our method against traditional baselines in two common personalization settings: (a) **Inference-only**: using the compressed profile for direct inference (Section 5.2), and (b) **Training and inference**: using the compressed profile from a subset of users for model training and inference on another subset (Section 5.3). Finally, we conduct further probing on token efficiency, fine-tuned vs. non-fine-tuned mark models, threshold α, and mark model layer selection in Section 5.4.

5.1 Experiment Settings

Datasets. We evaluate our method on two datasets: MovieLens-1M for context-based movie recommendation and LaMP-5 for personalized title generation. Following the settings in Section 3.2, the MovieLens task selects the user's truly interacted movie from 5 candidates (measured by accuracy), while the LaMP-5 task generates personalized paper titles from abstracts (measured by ROUGE-L scores). For inference-only settings, we evaluate on 1,000 MovieLens users and 750 LaMP-5 users using compressed user contexts. For training and inference settings, we train the generation model Φ_{G} on 500 MovieLens users and 375 LaMP-5 users with compressed contexts, then test on another 500 and 350 users, respectively. Note that all experimental data is excluded from the marking model Φ_{Mark} training process.

Models. We use the fine-tuned models from Section 3.2 as Φ_{Mark} for both datasets. We use Layer 6 (Llama-3.1-1B) for MovieLens marking and Layer 12 (Llama-3.1-8B) for LaMP marking, with the threshold set to 0.2 by default. For inference-only settings, we explore two configurations: (1) Llama-3.1-8B-Instruct as both Φ_{Sum} and Φ_{G}, and (2) GPT-4o-mini as both Φ_{Sum} and Φ_{G}. For training and inference settings, we use Llama-3.1-8B-Instruct as Φ_{Sum} and Llama-3.1-1B-Instruct as Φ_{G}.

Baselines. To validate the effectiveness of our method and the necessity of attention-based marking, we compare against various baselines across four categories: **Truncation-based:** (a) *Truncate* (Everaert et al., 2024) directly uses the most recent user contexts. **Direct summarization:** (b) *Direct Summary* (Zhang et al., 2024a) inputs unmarked contexts into Φ_{Sum} to derive compressed profiles. **Reasoning-enhanced:** (c) *CoT* (Wei et al., 2022) prompts Φ_{Sum} to think step-by-step for summarization; (d) *Self-Reflection* (Ji et al., 2023) asks Φ_{Sum} to self-reflect and refine its summary. **Alternative marking methods:** (e) *Mark All* marks all sentences as important for summarization; (f) *Random Mark* randomly marks the same number of sentences as H^* for summarization; (g) *Prompt-GS* first prompts Φ_{Sum} to identify important sentences, marks them to derive H^*, then summarizes.[3] We compare our Attn-GS method with these baselines under various maximum token settings [50, 100, 150, 200]. The prompt template for inference and training for Φ_{G} is fixed for fair comparison.

[3] More detail of these baselines are shown in Appendix A.3. RAG baselines and discussions are shown in Appendix A.6.

Method	Token Count							
	50		100		150		200	
	Llama-3	GPT-4o-mini	Llama-3	GPT-4o-mini	Llama-3	GPT-4o-mini	Llama-3	GPT-4o-mini
Truncate	40.9	41.0	41.9	42.6	42.3	43.2	42.5	45.8
Direct Summary	41.2	43.2	43.0	45.4	43.5	45.6	44.1	45.7
Random Mark	41.5	42.8	42.6	44.9	43.1	45.2	43.7	45.3
Mark All	41.8	43.5	43.2	45.6	43.7	45.8	44.3	45.9
Prompt-G	41.6	43.4	43.1	45.5	43.8	46.0	44.4	46.1
CoT	41.4	43.1	42.8	45.2	43.6	45.8	44.2	46.0
Self-Reflection	41.3	43.3	43.0	45.3	43.4	45.7	44.0	45.8
Attn-G	**44.7**	**45.3**	**46.0**	**48.0**	**47.1**	**49.6**	**47.7**	**50.6**

Table 1: Zero-Shot Accuracy (%) on MovieLens. Full context: Llama-3 48.7%, GPT-4o-mini 52.4%. No context: Llama-3 19.8%, GPT-4o-mini 21.1%.

Method	Token Count							
	50		100		150		200	
	Llama-3	GPT-4o-mini	Llama-3	GPT-4o-mini	Llama-3	GPT-4o-mini	Llama-3	GPT-4o-mini
Truncated Cont	0.330	0.353	0.343	0.368	0.344	0.370	0.345	0.370
Direct Summary	0.338	0.371	0.352	0.375	0.353	0.377	0.354	0.378
Random Mark	0.334	0.367	0.348	0.371	0.349	0.373	0.350	0.375
Mark All	0.339	0.372	0.352	0.376	0.352	0.376	0.354	0.378
Prompt-G	0.338	0.371	0.352	0.376	0.354	0.378	0.355	0.379
CoT	0.336	0.370	0.349	0.374	0.351	0.377	0.352	0.379
Self-Reflection	0.335	0.370	0.349	0.375	0.350	0.376	0.351	0.379
Attn-G	**0.359**	**0.388**	**0.370**	**0.394**	**0.372**	**0.397**	**0.374**	**0.401**

Table 2: ROUGE-L/Lsu Results. **Baselines:** The upper bound ('Full Context') for Llama-3-8B is 0.3895 and for GPT-4o-mini is 0.4139. The lower bound ('None') for Llama-3-8B is 0.256 and for GPT-4o-mini is 0.316.

5.2 Inference-only Performance

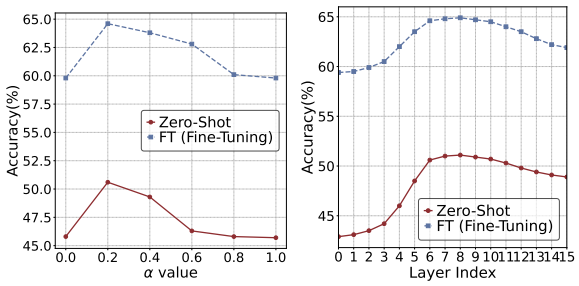
In this subsection, we evaluate the performance of Attn-GS in the inference-only setting, where compressed user contexts are directly used with real-time queries for inference. Results on the MovieLens and LaMP-5 datasets are shown in Tables 1 and 2, respectively. From Table 1, we observe that utilizing personalized contexts is essential for task performance, and performance generally increases with token length. Among all baselines, Truncate and Direct Summary consistently yield unsatisfactory performance, while reasoning-enhanced methods (CoT and Self-Reflection) do not provide clear performance gains, demonstrating that simply relying on LLMs' reasoning ability is insufficient to generate effective compressed profiles. Alternative marking methods such as Mark All, Random Mark, and Prompt-GS do not effectively enhance performance compared with Direct Summary, indicating their inability to identify truly important sentences without utilizing attention signals. In contrast, our Attn-GS consistently achieves the best performance across all settings (models and token lengths) and scales well with increasing tokens. Attn-GS achieves performance within 1.8% of using full context (10,000 tokens) with only 200 tokens, representing a 50× reduction. Similar findings are observed in Table 2, where Attn-GS significantly outperforms all baselines and achieves performance close to using full context. These findings validate the necessity of attention-based marking and the effectiveness of our method in inference-only scenarios.

Method	Token Count			
	50	100	150	200
Truncate	52.60	54.00	56.00	58.10
Direct Summary	56.20	57.10	58.40	59.80
Random Mark	55.80	56.70	58.00	59.40
Mark ALL	56.40	57.30	58.60	59.90
Prompt-G	56.00	56.90	58.20	59.60
CoT	55.90	56.80	58.10	59.50
Self-Reflection	56.10	57.00	58.30	59.60
Attn-G	**58.20**	**59.00**	**61.80**	**64.60**

Table 3: FT performance on MovieLens (accuracy %): Llama-1B-FT. Full context: 67.20%.

5.3 Training and Inference Setting Performance

In this subsection, we investigate the performance of Attn-GS in the training and inference setting. In this setting, compressed user profiles are utilized to train a better Φ_G that can be used for further inference (also taking compressed profiles as input). The results are presented in Table 3 for MovieLens and Table 4 for LaMP-5. From these tables, we observe that while increasing token count enhances performance, all baselines (truncation-based, direct



(a) Ablation study on α (b) Ablation study on layer

Figure 5: Ablation Studies

summarization, reasoning-enhanced, and alternative marking) yield unsatisfactory performance and present a large performance gap compared to using full context. In contrast, our Attn-GS significantly outperforms all baselines across all token settings, with a much smaller gap to using full context. These results demonstrate the superiority of Attn-GS not only in inference-only settings but also for model training, and further validate that the compressed user contexts generated by Attn-GS contain more valuable key personalization information for downstream tasks.

Method	Token Count			
	50	100	150	200
Truncated Cont	0.382	0.384	0.385	0.386
Simple Summary	0.384	0.384	0.386	0.389
Random Mark	0.380	0.381	0.384	0.387
Mark ALL	0.385	0.384	0.386	0.389
Prompt	0.384	0.384	0.386	0.390
CoT	0.383	0.383	0.386	0.388
Self-Reflection	0.383	0.383	0.385	0.389
Attn-GS	**0.396**	**0.402**	**0.404**	**0.405**

Table 4: FT Performance on LaMP-5(Rouge-L) 'Full Context': 0.4147.

5.4 Further Probing

Method	Truncate	Direct Summ.	Prompt-GS	Attn-GS
Tokens	750	750	700	**100**
Percentage	8.0%	8.0%	7.5%	**1.1%**

Table 5: Efficiency Comparison(GPT-4o-mini,48%)

Token Efficiency. To quantify Attn-GS's token efficiency, we compare the number of tokens required to achieve 48% accuracy in the inference-only setting with GPT-4o-mini as Φ_G in Table 5. We observe that Attn-GS reduces token requirements by 7× compared to baselines, using only 1.1% of full context tokens, which validates that Attn-GS significantly reduces inference tokens.

Finetuned mark model vs not finetuned. To validate that a fine-tuned Φ_{Mark} is more suitable for context marking than a non-fine-tuned Φ_{Mark}, we compare their performance in Table 6 in the inference-only setting with GPT-4o-mini as Φ_G and other parameters fixed. We observe that although the non-fine-tuned model still outperforms baselines in Table 1, it clearly underperforms Attn-GS using the fine-tuned model as Φ_{Mark}. This further validates the findings in Section 3.2 that fine-tuning enhances the model's ability to distinguish between important and unimportant signals.

Method	50	100	150	200
Not Fine-tuned Φ_{Mark}	44.1	46.2	47.1	47.9
Fine-tuned Φ_{Mark}	45.3	48.0	49.6	50.6

Table 6: Fine-tuned vs. Not Fine-tuned Φ_{Mark}

Thereshold α. We fix other parameters and vary the marking threshold α for MovieLens compression, reporting the downstream inference-only performance at 200 tokens on GPT-4o-mini as well as training and inference performance on Llama-1B-Instruct in Figure 5a. From the results, we observe that choosing a threshold between 0.2 and 0.4 yields the best overall performance. Setting the threshold too high (e.g., marking only the most highly-attended sentences) or too low (e.g., Mark All) fails to effectively guide the summarization process. We recommend carefully selecting a moderate threshold value based on validation set performance in practice.

Marking layers. Similarly, we keep other parameters fixed and vary the model layer[4] of Φ_{Mark}, reporting both downstream inference-only performance at 200 tokens on GPT-4o-mini (as Φ_{Sum} and Φ_G) and training and inference performance (Llama-3.1-8B-Instruct as Φ_{Sum} and Llama-3.1-1B-Instruct as Φ_G). From the results in Figure 5b, we observe that early layers yield low performance, performance peaks in the middle layers, and slightly decreases in the final layers. This is likely because early layers primarily capture low-level syntactic features, while middle and later layers are more responsible for processing personalized information and allocating attention weights for generation. Based on the results, we recommend using middle layers of LLMs for marking.

6 Conclusions

In this work, we explore the potential of leveraging LLM attention patterns to guide personalization context compression. Through extensive experiments, we discover that LLMs' attention patterns naturally reveal important personalization signals, and fine-tuning enhances their ability to distinguish

[4]Ablation study on marking model size are shown in Appendix A.8

relevant information. Based on these findings, we propose Attn-GS, an attention-guided compression framework that uses fine-tuned attention patterns to intelligently compress personalization contexts. Experiments demonstrate that Attn-GS consistently outperforms all baselines, achieving close performance of using full context while reducing token usage by 50× compared to full context and 7× compared to baseline methods.

Limitations

While our work demonstrates the potential of attention-guided compression for personalization tasks, several aspects warrant further exploration. First, although we reveal that attention patterns can unveil personalization signal importance, a deeper understanding of how LLMs internally process and integrate personalization inputs remains an open question. Future work could investigate the underlying mechanisms in greater detail to develop more sophisticated compression strategies. Second, extending our framework to broader scenarios, including multi-task settings and diverse application domains, would help establish the generalizability and scope of attention-guided compression approaches.

Acknowledgment

Shenglai Zeng, Xinan Dai, Kai Guo, and Hui Liu are supported by the National Science Foundation (NSF) under grant numbers CNS2321416, IIS2212032, IIS2212144, IIS 2504089, DUE2234015, CNS2246050, DRL2405483 and IOS2035472, the Michigan Department of Agriculture and Rural Development, US Dept of Commerce, Gates Foundation, Amazon Faculty Award, Meta, NVIDIA, Microsoft and SNAP.

References

Mansheel Agarwal, Valliappan Raju, Rajesh Dey, and Ipseeta Nanda. 2022. Descriptive research on ai-based tools to aid personalized customer service: Case of chatgpt. *Journal of Reproducible Research*, 1(1):140–146.

Shiqi Chen, Miao Xiong, Junteng Liu, Zhengxuan Wu, Teng Xiao, Siyang Gao, and Junxian He. 2024. In-context sharpness as alerts: An inner representation perspective for hallucination mitigation. *arXiv preprint arXiv:2403.01548*.

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*, pages 1126–1132.

Dante Everaert, Rohit Patki, Tianqi Zheng, and Christopher Potts. 2024. Amazonqac: A large-scale, naturalistic query autocomplete dataset. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1046–1055.

Rhys Gould, Euan Ong, George Ogden, and Arthur Conmy. 2023. Successor heads: Recurring, interpretable attention heads in the wild. *arXiv preprint arXiv:2312.09230*.

Danny Halawi, Jean-Stanislas Denain, and Jacob Steinhardt. 2023. Overthinking the truth: Understanding how language models process false demonstrations. *arXiv preprint arXiv:2307.09476*.

F. Maxwell Harper and Joseph A. Konstan. 2015. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4).

Pengfei He, Zhenwei Dai, Xianfeng Tang, Yue Xing, Hui Liu, Jingying Zeng, Qiankun Peng, Shrivats Agrawal, Samarth Varshney, Suhang Wang, et al. 2025. Attention knows whom to trust: Attention-based trust management for llm multi-agent systems. *arXiv preprint arXiv:2506.02546*.

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023. Towards mitigating llm hallucination via self reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1827–1843.

Guangyuan Jiang, Manjie Xu, Song-Chun Zhu, Wenjuan Han, Chi Zhang, and Yixin Zhu. 2023. Evaluating and inducing personality in pre-trained language models. *Advances in Neural Information Processing Systems*, 36:10622–10643.

Sheng Li and Handong Zhao. 2021. A survey on representation learning for user modeling. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 4997–5003.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. 2024. Snapkv: Llm knows what you are looking for before generation. *Advances in Neural Information Processing Systems*, 37:22947–22970.

Junling Liu, Chao Liu, Peilin Zhou, Renjie Lv, Kang Zhou, and Yan Zhang. 2023. Is chatgpt a good recommender? a preliminary study. *arXiv preprint arXiv:2304.10149*.

Qijiong Liu, Nuo Chen, Tetsuya Sakai, and Xiao-Ming Wu. 2024. Once: Boosting content-based recommendation with both open-and closed-source large

language models. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 452–461.

Zhining Liu, Rana Ali Amjad, Ravinarayana Adkathimar, Tianxin Wei, and Hanghang Tong. 2025. Selfelicit: Your language model secretly knows where is the relevant evidence. *arXiv preprint arXiv:2502.08767*.

Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Si Zhang, Ren Chen, Christopher Leung, Jiajie Tang, and Jiebo Luo. 2023. Llm-rec: Personalized recommendation via prompting large language models. *arXiv preprint arXiv:2307.15780*.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. 2022. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. 2023. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36:10299–10315.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024. Lamp: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7370–7392.

Gregory Serapio-García, Mustafa Safdari, Clément Crepy, Luning Sun, Stephen Fitz, Marwa Abdulhai, Aleksandra Faust, and Maja Matarić. 2023. Personality traits in large language models.

Yunxiao Shi, Wujiang Xu, Zeqi Zhang, Xing Zi, Qiang Wu, and Min Xu. 2025. Personax: A recommendation agent oriented user modeling framework for long behavior sequence. *arXiv preprint arXiv:2503.02398*.

Jiwei Tang, Zhicheng Zhang, Shunlong Wu, Jingheng Ye, Lichen Bai, Zitai Wang, Tingwei Lu, Lin Hai, Yiming Zhao, Hai-Tao Zheng, et al. 2025. Gmsa: Enhancing context compression via group merging and layer semantic alignment. *arXiv preprint arXiv:2505.12215*.

Fali Wang, Minhua Lin, Yao Ma, Hui Liu, Qi He, Xianfeng Tang, Jiliang Tang, Jian Pei, and Suhang Wang. 2025a. A survey on small language models in the era of large language models: Architecture, capabilities, and trustworthiness. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, page 6173–6183, New York, NY, USA. Association for Computing Machinery.

Lei Wang, Jingsen Zhang, Hao Yang, Zhi-Yuan Chen, Jiakai Tang, Zeyu Zhang, Xu Chen, Yankai Lin, Hao Sun, Ruihua Song, et al. 2025b. User behavior simulation with large language model-based agents. *ACM Transactions on Information Systems*, 43(2):1–37.

Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S Yu, and Qingsong Wen. 2024. Large language models for education: A survey and outlook. *arXiv preprint arXiv:2403.18105*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Fan Yang, Zheng Chen, Ziyan Jiang, Eunah Cho, Xiaojiang Huang, and Yanbin Lu. 2023. Palr: Personalization aware llms for recommendation. *arXiv preprint arXiv:2305.07622*.

Qinkai Yu, Mingyu Jin, Dong Shu, Chong Zhang, Lizhou Fan, Wenyue Hua, Suiyuan Zhu, Yanda Meng, Zhenting Wang, Mengnan Du, et al. 2024. Healthllm: Personalized retrieval-augmented disease prediction system. *arXiv preprint arXiv:2402.00746*.

Shenglai Zeng, Pengfei He, Kai Guo, Tianqi Zheng, Hanqing Lu, Yue Xing, and Hui Liu. 2025a. Towards context-robust llms: A gated representation finetuning approach. *arXiv preprint arXiv:2502.14100*.

Shenglai Zeng, Jiankun Zhang, Pengfei He, Yue Xing, Yiding Liu, Han Xu, Jie Ren, Shuaiqiang Wang, Dawei Yin, Yi Chang, et al. 2024. The good and the bad: Exploring privacy issues in retrieval-augmented generation (rag). *arXiv preprint arXiv:2402.16893*.

Shenglai Zeng, Jiankun Zhang, Bingheng Li, Yuping Lin, Tianqi Zheng, Dante Everaert, Hanqing Lu, Hui Liu, Yue Xing, Monica Xiao Cheng, et al. 2025b. Towards knowledge checking in retrieval-augmented generation: A representation perspective. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2952–2969.

An Zhang, Yuxin Chen, Leheng Sheng, Xiang Wang, and Tat-Seng Chua. 2024a. On generative agents in recommendation. In *Proceedings of the 47th international ACM SIGIR conference on research and development in Information Retrieval*, pages 1807–1817.

Junjie Zhang, Yupeng Hou, Ruobing Xie, Wenqi Sun, Julian McAuley, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2024b. Agentcf: Collaborative learning with autonomous language agents for recommender systems. In *Proceedings of the ACM Web Conference 2024*, pages 3679–3689.

Zihuai Zhao, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, et al. 2024. Recommender systems in the era of large language models (llms).

IEEE Transactions on Knowledge and Data Engineering, 36(11):6889–6907.

Yujia Zhou, Qiannan Zhu, Jiajie Jin, and Zhicheng Dou. 2024. Cognitive personalized search integrating large language models with an efficient memory mechanism. In *Proceedings of the ACM Web Conference 2024*, pages 1464–1473.

Xi Zhu, Yu Wang, Hang Gao, Wujiang Xu, Chen Wang, Zhiwei Liu, Kun Wang, Mingyu Jin, Linsey Pang, Qingsong Weng, et al. 2025. Recommender systems meet large language model agents: A survey. *Foundations and Trends® in Privacy and Security*, 7(4):247–396.

A Appendix

A.1 Dataset Details

MovieLens-1M. In our experiment, we utilize a subset of MovieLens-1M dataset (Harper and Konstan, 2015), using 1000 users for training while another 1000 users for testing. For each user, user contexts include Movie Title, User Rating, Movie Summary, Movie Genres, Watch Time, Movie Year, and User Basic Information (age, gender, occupation), providing diverse personalization signals for analysis. Examples of user contexts can be found at Figure 6.

LaMP-5. In our experiment we utilize a subset of LaMP-5: Personalized Scholarly Title Generation dataset (Salemi et al., 2024) . The user contexts include the authors' previous papers, with signals comprising paper title, paper summary, and publication date. Examples can be found in Figure 7. For each user, the task is to generate a personalized paper title based on a given paper summary and the user's writing history. We measure performance using Rouge-L scores between the generated titles and ground truth (actual titles).

A.2 Details of Attn-GS

A.2.1 Prompt Used

The summarization prompts used for MovieLens are shown in Figure 8 , used for Lamp-5 are shown in Figure 9.

A.2.2 Examples of Marked Contexts

Examples of Marked Contexts are shown in Figures 6 and 7. The results demonstrate that LLMs' attention mechanisms can effectively identify important personalization signals within long user contexts, such as movie titles, user ratings, and paper titles.

A.2.3 Examples of Summarization Results

We show some summarized results under different tokens for MovieLens and Lamp-5 dataset in Figure 10 and Figure 11, respectively.

A.2.4 Definition of Sentence-Level Attention

To identify important personalization signals, we formalize the computation of sentence-level attention scores. We define the importance of a sentence by aggregating the attention weights that the model assigns to its constituent tokens during the inference of a personalization task.

Specifically, we first calculate token-level attention scores $\tilde{w}_m^{(d)}$, which represent the attention weights from the last input token (the Query) toward the preceding input tokens (the Keys) within the user history. For a given sentence u_i that spans a sequence of tokens from position $p_{start}^{u_i}$ to $p_{end}^{u_i}$, the sentence-level attention score $\hat{w}_i^{(d)}$ is computed by averaging the scores of all tokens within that span:

$$\hat{w}_i^{(d)} = \frac{1}{p_{end}^{u_i} - p_{start}^{u_i} + 1} \sum_{m=p_{start}^{u_i}}^{p_{end}^{u_i}} \tilde{w}_m^{(d)} \quad (7)$$

This metric provides a normalized measure of the relative importance the marking model Φ_{Mark} attributes to each sentence, allowing for the subsequent hard selection of high-signal content for compression.

A.2.5 Task and Prompt Specifications

To ensure the reproducibility of **Attn-GS**, we define the task descriptions, marking prompts, and generation protocols for the primary evaluation tasks: *MovieLens* (recommendation) and *Title Generation*.

Task Descriptions The task description ($\mathcal{T}$) serves as the contextual anchor for both the marking and generation processes:

- **MovieLens**: "Based on your watch history, please choose ONE movie in the candidate movies that you most likely to watch... Please respond ONLY with the format 'The answer is <single_letter_id>'."
- **Title Generation**: "Based on the user's writing summary, generate a suitable title for the given abstract. The title must be the ONLY output..."

Generation Process (Φ_G) The generation model Φ_G takes the compressed user profile and the task-specific query to produce the final output. The input formats are formalized as follows:

- **MovieLens**: `User profile: {summarized context} \n\n {Task_description} \n\n {candidate_movies}`
- **Title Generation**: `User Writing Summary: {summarized context} \n\n {Task_description} \n\n Abstract: {question} \n\n Title:`

Marking Model (Φ_{Mark}) Protocols The marking model is utilized in two settings to identify critical personalization signals:

- **Fine-tuning Format**: When Φ_{Mark} is fine-tuned, it is trained on a small subset of task-specific input-output pairs using the *full context*. For example, in MovieLens, the model learns to map the `full context + Task_description + candidates` to the ground-truth `Output: "The answer is {B}"`.
- **Marking Input**: During the inference phase of the marking process (to obtain attention scores), the input consists of the task description and the full user history. Notably, the specific query (e.g., candidate movies or abstract) is excluded to ensure the resulting profile is broadly applicable:
  - **MovieLens Marking**: `User profile: {full context} \n\n {Task_description}`
  - **Title Generation Marking**: `User Writing history: {full context} \n\n {Task_description}`

A.3 Baseline Details

We compare against the following baselines: **Truncation-based:** (a) *Truncate* (Everaert et al., 2024) uses the most recent m tokens from user contexts without additional processing. **Direct summarization:** (b) *Direct Summary* (Zhang et al., 2024a) inputs raw contexts into Φ_{Sum} to generate compressed profiles. **Reasoning-enhanced:** (c) *CoT* (Wei et al., 2022) prompts Φ_{Sum} to think step-by-step for summarization; (d) *Self-Reflection* (Ji et al., 2023) prompts Φ_{Sum} to self-reflect and refine its summary. **Alternative marking methods:** (e) *Mark All* marks all sentences as important; (f) *Random Mark* randomly marks the same number of sentences as H^*; (g) *Prompt-GS* first prompts Φ_{Sum} to identify important sentences, marks them to derive H^*, then summarizes based on the marked context.

- *Truncate*: Directly truncates user contexts to the most recent m tokens without additional processing.
- *Direct Summary*: Inputs raw contexts into Φ_{Sum} to generate compressed profiles. We evaluate various prompts and select the best-performing one. The prompts are shown in Figure 12 for MovieLens and Figure 13 for LaMP-5.
- *CoT*: Prompts Φ_{Sum} to think step-by-step for summarization. The prompts are shown in Figure 14 and Figure 15.
- *Self-Reflection*: Prompts Φ_{Sum} to self-reflect and refine its summary. The prompts are shown in Figure 16 and Figure 17.
- *Random Mark* and *Mark All*: Random Mark randomly selects the same number of sentences as H^* for marking, while Mark All marks all sentences as important. Both use the same summarization prompts as Attn-GS, shown in Figure 8 and Figure 9.
- *Prompt-GS*: First prompts Φ_{Sum} to identify important sentences, marks them to derive H^*, then summarizes based on the marked context. The identification prompts are shown in Figure 18 , while the summarization prompts are the same as Attn-GS (Figure 8 and Figure 9).

A.4 Comparison with General-Purpose Context Compression (e.g., GMSA (Tang et al., 2025))

Recent research explores generic context compression methods, primarily focusing on accelerating inference by reducing total input length. A notable soft compression method is GMSA (Tang et al., 2025) (Group Merging and Layer Semantic Alignment), which uses an encoder-decoder framework to convert vast knowledge into highly efficient soft tokens (vector representations). GMSA (Tang et al., 2025) features a Group Merging mechanism for efficient vector extraction and a Layer Semantic Alignment (LSA) module to explicitly bridge the semantic gap between the compressed representation and the decoder's input. Its primary goal is maximizing computational efficiency and knowledge retention for general tasks like QA. In contrast, Attn-GS is explicitly designed for the unique challenges of personalization. Our mechanism fundamentally differs from GMSA's structured vector encoding: Attn-GS uses a hybrid Attention-Guided Marking strategy, leveraging a Marking LLM's attention scores to selectively filter and extract only the most critical personalization signals. This selective process addresses the high noise and task-irrelevance inherent in raw user history. Furthermore, Attn-GS's goal is to output a task-relevant, human-readable text summary (user

profile), rather than GMSA's machine-readable soft tokens, which is key for interpretability in personalized LLM applications. Finally, we note that these two methods are not mutually exclusive, and there is high potential for integrating Attn-GS's high-level task-relevance filtering with GMSA's efficient soft-token encoding in future work.

A.5 Handling Dynamic Updates and Task Shifts

In this section, we discuss the implications of dynamic query environments and evolving user profiles on the **Attn-GS** framework, addressing the trade-offs between offline efficiency and online adaptability.

- **Task and Query Shifts**: The **Attn-GS** pipeline is designed for low-latency inference by decoupling the marking process from real-time queries. While the marking model leverages a general task description ($\mathcal{T}$), it intentionally excludes the specific real-time query ($\mathcal{Q}$) to ensure that the generated profile ($\mathcal{G}$) remains broadly applicable to diverse queries within the same task category. In scenarios where task categories shift frequently, we propose defining importance signals based on a **cross-task attention average**. By aggregating attention scores across multiple critical task types during the offline phase, the model can generate a robust, multi-task generalized profile that minimizes the need for frequent recalculation when task goals shift.

- **Profile Updates**: When new user history is added or existing information is deleted, the compressed profile $\mathcal{G}$ requires periodic recalculation to maintain accuracy. However, because this process is conducted **offline and asynchronously** (e.g., on an hourly or daily batch schedule), the computational cost is amortized. This design choice prioritizes minimal online inference latency and reduced API costs, representing a strategic trade-off where periodic offline updates enable high-speed, real-time personalization.

The decoupling of signal marking from the final query allows **Attn-GS** to maintain a stable and informative user concept that is both computationally efficient and versatile across various downstream interaction patterns.

A.6 RAG Baselines and Design Justification

To evaluate the effectiveness of **Attn-GS** against retrieval-based strategies, we compare it with a **Retrieval-Augmented Generation (RAG)** baseline. While RAG is a prevalent approach for handling long-context scenarios, there are fundamental differences in design philosophy and performance:

- **Offline vs. Online Compression**: Standard RAG typically operates *online* during inference, calculating similarity scores between a real-time query and the user history. This process introduces additional computational overhead and latency, which can be prohibitive for sensitive applications(Everaert et al., 2024). In contrast, **Attn-GS** performs the intensive marking and summarization steps *offline* without requiring the final query, producing a fixed, low-token summary that ensures rapid online deployment.

- **Granularity of Selection**: As shown in Table 7, naive RAG improves upon simple truncation but still underperforms **Attn-GS**, even when the specific query is accessible to the retrieval model. This suggests that RAG's sample-level similarity matching is less effective than **Attn-GS**'s ability to capture fine-grained, signal-level importance within the history.

- **Synergy (RAG+Attn-GS)**: Our results further demonstrate that retrieval and attention-guided compression are not mutually exclusive. By integrating the two–using RAG to retrieve the top-k relevant history items and then applying **Attn-GS** to refine those items into a high-density summary–we achieve the best performance.

Method	Token Count			
	50	**100**	**150**	**200**
Truncate	40.90	41.90	42.30	42.50
RAG	43.10	44.80	46.30	46.90
Attn-GS	44.70	46.00	47.10	47.70
RAG+Attn-GS	**46.30**	**47.20**	**48.50**	**49.70**

Table 7: Comparison with RAG baselines on MovieLens (accuracy %). Attn-GS performs fine-grained signal marking which complements the coarse-grained retrieval of RAG.

A.7 Prompting-based Approximations for Black-Box Models

While Attn-GS fundamentally relies on accessing the internal attention scores of a white-box Marking LLM (Φ_{Mark}), the core principle—identifying task-relevant signals for compression—is feasible even with black-box (API-only) LLMs.We can approximate the attention-guided selection through Prompting-based Importance Estimation. This involves using a carefully designed prompt to instruct the black-box LLM (e.g., GPT-4) to perform the critical sentence identification task. Specifically, the LLM is prompted to read the context and the task description, then output only the sentences it deems most important for the task, effectively simulating the filtering step. This strategy leverages the instruction-following capabilities of powerful black-box models to achieve signal-aware filtering, trading off the precision of direct attention scores for broader applicability and immediate deployment across various commercial LLM APIs.

A.8 Ablation on Marking Model Size

To investigate the impact of the marking model's scale on the accuracy of identified personalization signals, we conduct an ablation study using marking models (Φ_{Mark}) of varying sizes. We evaluate fine-tuned versions of Llama-3 (1B, 3B, and 8B) and Qwen-2.5-14B on the LaMP-5 dataset, while keeping the summary model (Φ_{Sum}) and generation model (Φ_G) fixed as GPT-4o.

As shown in Table 8, increasing the parameter count and capability of Φ_{Mark} generally improves performance, confirming that larger models produce more accurate attention patterns. However, we observe that the performance gain is substantial when scaling from 1B to 3B parameters, but becomes marginal as the size increases further to 14B. These results suggest that while a baseline level of model capability is required for effective marking, extremely large models may yield diminishing returns for this specific task. For practical deployment, we recommend selecting the marking model based on a cost-performance trade-off analysis tailored to the specific application.

Model Size	**Performance (ROUGE-L)**
1B	0.390
3B	0.397
8B	0.401
14B (Qwen-2.5)	**0.403**

Table 8: Ablation study on Φ_{Mark} size (LaMP-5 dataset, 200 token budget). Significant gains are observed up to 3B, with diminishing returns beyond that scale.

Examples of Marked Context (MovieLens)

User basic info: Gender is M, Age is 18-24, Occupation is academic/educator, Total Movies watched is 39, Average Rating is 3.24 out of 5.0.
Movie 1 title: The movie title is Twelve Monkeys. Movie 1 year: Released in 1995. Movie 1 genres: Genres are Drama, Sci-Fi. Movie 1 summary: A time traveler from a post-apocalyptic future is sent back in time to gather information about a deadly virus that has wiped out most of humanity. **Movie 1 rating: User gave it 5 stars. Movie 1 rating time: Rated on 2000-12-06 at 18:21:10.**
Movie 2 title: The movie title is Good Will Hunting. Movie 2 year: Released in 1997. **Movie 2 genres: Genres are Drama.** Movie 2 summary: A brilliant but troubled janitor discovers his mathematical genius with the help of a professor who challenges him to confront his past and embrace his potential. Movie 2 rating: User gave it 5 stars. Movie 2 rating time: Rated on 2000-12-06 at 18:21:03.
Movie 3 title: The movie title is Run Lola Run (Lola rennt). Movie 3 year: Released in 1998. Movie 3 genres: Genres are Action, Crime, Romance. Movie 3 summary: A high-stakes race against time unfolds as Lola must find a way to save her boyfriend's life in just 20 minutes. Movie 3 rating: User gave it 5 stars. Movie 3 rating time: Rated on 2000-12-06 at 18:21:03.
Movie 4 title: The movie title is Trainspotting. Movie 4 year: Released in 1996. Movie 4 genres: Genres are Drama. Movie 4 summary: A group of heroin addicts navigate the gritty streets of Edinburgh, Scotland, in a darkly humorous tale of addiction, friendship, and self-destruction.....Movie 37 rating: User gave it 1 stars. Movie 37 rating time: Rated on 2000-12-06 at 18:11:03.
Movie 38 title: The movie title is Star Trek VI: The Undiscovered Country. Movie 38 year: Released in 1991. Movie 38 genres: Genres are Action, Adventure, Sci-Fi. Movie 38 summary: A thrilling and suspenseful adventure as the crew of the USS Enterprise must prevent an interstellar war and uncover a conspiracy threatening peace in the galaxy. Movie 38 rating: User gave it 1 stars. Movie 38 rating time: Rated on 2000-12-06 at 18:11:03.

Figure 6: Examples of marked context on the MovieLens dataset.The bold texts are marked as important signals (marked between <START_IMPORTANT> and <END_IMPORTANT> tags)

Examples of Marked Context (Lamp5)

Paper 1 title: Visualizing time-oriented data-A systematic view. Paper 1 abstract: The analysis of time-oriented data is an important task in many application scenarios. In recent years, a variety of techniques for visualizing such data have been published. This variety makes it difficult for prospective users to select methods or tools that are useful for their particular task at hand. In this article, we develop and discuss a systematic view on the diversity of methods for visualizing time-oriented data. With the proposed categorization we try to untangle the visualization of time-oriented data, which is such an important concern in Visual Analytics. **The categorization is not only helpful for users, but also for researchers to identify future tasks in Visual Analytics..**
Paper 1 date: Published in 2007. Paper 2 title: Visualizing Statistical Properties of Smoothly Brushed Data Subsets. Paper 2 abstract: In many application fields, the statistical properties of data sets are of great interest for data analysts. Since local variations can occur especially in large datasets, it is useful to visualize not only global values,.....

Figure 7: Examples of marked context on the Lamp-5 dataset.The bold texts are marked as important signals (marked between <START_IMPORTANT> and <END_IMPORTANT> tags)

Summarization Prompt for MovieLens

The information marked between <START_IMPORTANT> and <END_IMPORTANT> tags contains critical personalization signals. Prioritize these details in your summary as they are essential for understanding user preferences and behavior patterns.
Generate a {args.max_tokens}-token first-person summary that:

1. Prioritizes all important-marked information
2. Uses first-person perspective ("I am...", "I like...")
3. Preserves key details when possible (names, dates, ratings, etc.)
4. Output ONLY the summary directly without introductory phrases
5. Be concise to maximize information density

Figure 8: Attention-guided summarization prompt template for MovieLens.

Summarization Prompt for Academic Title Generation

The information marked between <START_IMPORTANT> and <END_IMPORTANT> tags contains critical personalization signals. Prioritize these details in your summary as they are essential for understanding user preferences and behavior patterns.
Generate a {args.max_tokens}-token first-person summary that:

1. Prioritizes all important-marked information
2. About the users' title writting patterns. optimized for the subsequent academic title generation.
3. Preserves key details when possible
4. Output ONLY the summary directly without introductory phrases
5. Be concise to maximize information density

Figure 9: Attention-guided summarization prompt template for academic title generation.

Examples of User Profile and Movie Preferences

50 tokens I am an 18-24 year old female doctor who has watched 25 movies with a 4.15 average rating. I love romantic comedies like Notting Hill, You've Got Mail, and Four Weddings and a Funeral, giving most 4-5 stars. I particularly enjoyed Hope Floats, Bed of Roses, and 101 Dalmatians with perfect 5-star ratings, and rated all my movies on July 2, 2000.

100 tokens I am an 18-24 year old female doctor who has watched 25 movies with a 4.15 average rating. I love romantic comedies and gave 5 stars to classics like Notting Hill, You've Got Mail, Hope Floats, Bed of Roses, E.T., and 101 Dalmatians. I consistently rated popular rom-coms like Forrest Gump, Runaway Bride, Princess Bride, Never Been Kissed, My Best Friend's Wedding, and Four Weddings and a Funeral with 4 stars. I rated all these movies on July 2, 2000, showing my preference for heartwarming stories with romantic elements, comedic adventures, and feel-good narratives from the 1990s era.

150 tokens I am an 18-24 year old female doctor who has watched 25 movies with an average rating of 4.15 stars. I rated most of my movies on July 2, 2000, showing a clear preference for romantic comedies and dramas. I gave 5-star ratings to 101 Dalmatians, E.T., Bed of Roses, Notting Hill, and Hope Floats. My favorites include romantic films like Forrest Gump, Runaway Bride, You've Got Mail, My Best Friend's Wedding, Four Weddings and a Funeral, and Ghost. I particularly enjoy stories about love, relationships, and emotional journeys. I rated Shakespeare in Love and So I Married an Axe Murderer lower at 3 stars. My viewing patterns suggest I gravitate toward heartwarming narratives with romantic elements, comedic situations, and character-driven plots that explore human connections and personal growth.

200 tokens I am a female healthcare professional, aged 18-24, who has watched 25 movies with an average rating of 4.15 out of 5 stars. I have a strong preference for romantic comedies and dramas, consistently rating films like Notting Hill, You've Got Mail, and Bed of Roses with 5 stars. My viewing pattern shows I rated all my movies on July 2, 2000, suggesting I discovered cinema around that time. I particularly enjoy heartwarming stories like Forrest Gump, romantic tales like Four Weddings and a Funeral and My Best Friend's Wedding, and even gave Hope Floats a perfect 5-star rating. I also appreciate family-friendly adventures like Flipper and E.T., rating both highly. My taste spans from classic romances like Ghost and The English Patient to quirky comedies like The Nutty Professor and Never Been Kissed. Even unconventional romantic stories like Chasing Amy earned 4 stars from me. The only film I rated below 4 stars was Shakespeare in Love with 3 stars, along with the thriller-comedy So I Married an Axe Murderer, suggesting I prefer straightforward romantic narratives over complex period pieces or dark humor.

Figure 10: Examples of attention-guided summaries of MovieLens dataset

Examples of Title Writing Patterns

50 tokens: I write technically precise titles with systematic structure: [Technical Qualifier] + [Core Method] + [Application Domain] + [Performance Goal]. I favor compound constructions using hyphens, emphasize optimization language ("efficient," "robust," "minimizing"), and target systems/computing domains. My patterns prioritize immediate domain identification and methodological contribution clarity.

100 tokens: I structure titles with technical qualifiers followed by core methods and application domains. My pattern: [Technical Descriptor] + [System/Algorithm] + [Specific Domain] + [Performance Goal]. I favor compound constructions using hyphens and colons for information density. Semantic clusters include energy efficiency, real-time systems, and distributed computing. I use performance-oriented language ("robust," "efficient," "optimal") with domain-specific acronyms (WCRT, GPU, NFV). Titles progress from methodological contribution to application context, maintaining academic formality while ensuring immediate domain recognition for computer science audiences.

150 tokens: I structure titles with consistent technical qualifiers followed by core methods and specific application domains. My dominant pattern: [Technical Descriptor] + [System/Algorithm] + [Domain Context] + [Performance Objective]. I favor compound constructions using hyphens and colons to maximize information density, as in "Energy-Efficient Resource Utilization for Heterogeneous Embedded Computing Systems." My semantic preferences cluster around systems optimization, particularly energy efficiency, real-time computing, and distributed architectures. I consistently use performance-oriented language ("robust," "efficient," "optimal," "minimizing") paired with specific technical domains (cloud computing, embedded systems, automotive networks). Strategic acronym usage incorporates established abbreviations (WCRT, CAN, GPU, NFV) for precision. Titles typically progress from broad methodological contribution to specific application context, maintaining academic formality while ensuring immediate domain identification. I prioritize scannable hierarchical information that communicates scope, technical contribution, and target application within computer science and engineering contexts.

200 tokens: I structure titles using a systematic formula: [Technical Qualifier] + [Core Method/System] + [Specific Application Domain] + [Performance Objective]. My preference for compound constructions with hyphens and colons creates hierarchical information density, as in "Energy-Efficient Resource Utilization for Heterogeneous Embedded Computing Systems." My semantic patterns cluster around systems optimization, particularly energy efficiency, real-time computing, and distributed architectures. I consistently employ performance-oriented language ("robust," "efficient," "optimal," "minimizing") paired with specific technical domains (cloud computing, embedded systems, automotive networks, GPU computing). Technical descriptors typically open my titles: "Robust," "Adaptive," "Parallel," "Energy-Efficient," immediately establishing domain expertise. I strategically incorporate established acronyms (WCRT, CAN, GPU, NFV, IoT) when they enhance precision without sacrificing clarity. My title progression follows a logical flow from broad methodological contribution to specific application context, ensuring immediate domain identification. I favor multi-layered technical specifications that communicate both theoretical contribution and practical application scope. The structure maintains academic formality while maximizing information density for computer science and engineering conference audiences, emphasizing algorithmic innovation within constrained system environments.

Figure 11: Examples of attention-guided summaries of Lamp-5 dataset

Direct Summarization Prompt for MovieLens

Generate a {args.max_tokens}-token first-person summary of the user's movie preferences that:

1. Captures the user's viewing history and rating patterns
2. Uses first-person perspective (`"I am..."`, `"I like..."`)
3. Preserves key details when possible (names, dates, ratings, etc.)
4. Output ONLY the summary directly without introductory phrases
5. Be concise to maximize information density

Figure 12: Direct summarization prompt template for MovieLens.

Direct Summarization Prompt for Academic Title Generation

Generate a {args.max_tokens}-token first-person summary that:

1. Captures the user's title writing patterns and research interests
2. Optimized for subsequent academic title generation
3. Preserves key details when possible
4. Output ONLY the summary directly without introductory phrases
5. Be concise to maximize information density

Figure 13: Direct summarization prompt template for academic title generation.

CoT Summarization Prompt for MovieLens

Let's think step by step to summarize the user's movie preferences.
Generate a {args.max_tokens}-token first-person summary that:

1. Captures the user's viewing history and rating patterns
2. Uses first-person perspective (`"I am..."`, `"I like..."`)
3. Preserves key details when possible (names, dates, ratings, etc.)
4. Output ONLY the summary directly without introductory phrases
5. Be concise to maximize information density

Figure 14: CoT summarization prompt template for MovieLens.

CoT Summarization Prompt for Academic Title Generation

Let's think step by step to summarize the user's title writing patterns.
Generate a {args.max_tokens}-token first-person summary that:

1. Captures the user's title writing patterns and research interests
2. Optimized for subsequent academic title generation
3. Preserves key details when possible
4. Output ONLY the summary directly without introductory phrases
5. Be concise to maximize information density

Figure 15: CoT summarization prompt template for academic title generation.

Self-Reflection Summarization Prompt for MovieLens (Step 1: Initial Summary)

Generate a {args.max_tokens}-token first-person summary of the user's movie preferences that:

1. Captures the user's viewing history and rating patterns
2. Uses first-person perspective (`"I am..."`, `"I like..."`)
3. Preserves key details when possible (names, dates, ratings, etc.)
4. Be concise to maximize information density

Self-Reflection Summarization Prompt for MovieLens (Step 2: Refinement)

Here is an initial summary of the user's movie preferences: {initial_summary}
Please reflect on this summary and refine it. Consider: - Does it capture the most important viewing patterns? - Are key preferences clearly stated? - Is any critical information missing? - Can it be more concise while retaining essential details?
Generate a refined {args.max_tokens}-token first-person summary that:

1. Uses first-person perspective (`"I am..."`, `"I like..."`)
2. Preserves key details when possible (names, dates, ratings, etc.)
3. Output ONLY the summary directly without introductory phrases
4. Be concise to maximize information density

Figure 16: Self-reflection summarization prompt template for MovieLens (two-step process).

Self-Reflection Summarization Prompt for Academic Title Generation (Step 1: Initial Summary)

Generate a {args.max_tokens}-token first-person summary that:

1. Captures the user's title writing patterns and research interests
2. Optimized for subsequent academic title generation
3. Preserves key details when possible
4. Be concise to maximize information density

Self-Reflection Summarization Prompt for Academic Title Generation (Step 2: Refinement)

Here is an initial summary of the user's title writing patterns: {initial_summary}
Please reflect on this summary and refine it. Consider: - Does it capture key writing patterns and preferences? - Are recurring themes and terminology clearly identified? - Is any critical stylistic information missing? - Can it be more concise while retaining essential details?
Generate a refined {args.max_tokens}-token first-person summary that:

1. Captures the user's title writing patterns optimized for subsequent academic title generation
2. Preserves key details when possible
3. Output ONLY the summary directly without introductory phrases
4. Be concise to maximize information density

Figure 17: Self-reflection summarization prompt template for academic title generation (two-step process).

Prompt-GS Identification Prompt

You are given a user's personalization context and a task description. Your task is to identify and extract the most important sentences that are relevant for the given task.
Please read through the context carefully and identify sentences that contain critical personalization signals. Copy and paste only the important sentences to the output, preserving their original text exactly.
Input format:
Context: {context}
Task Description: {task_description}
Output format: Simply list the important sentences, one per line, without any additional explanation or formatting.

Figure 18: Prompt-GS identification prompt for extracting important sentences.