

000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 REASONING-BASED PERSONALIZED GENERATION FOR USERS WITH SPARSE DATA

Anonymous authors

Paper under double-blind review

ABSTRACT

Large Language Model (LLM) personalization holds great promise for tailoring responses by leveraging personal context and history. However, real-world users usually possess sparse interaction histories with limited personal context, such as cold-start users in social platforms and newly registered customers in online E-commerce platforms, compromising the LLM-based personalized generation. To address this challenge, we introduce GRASPER (**G**raph-based **S**parse **P**ersonalized **R**easoning), a novel framework for enhancing personalized text generation under sparse context. GRASPER first augments user context by predicting items that the user would likely interact with in the future. With reasoning alignment, it then generates texts for these interactions to enrich the augmented context. In the end, it generates personalized outputs conditioned on both the real and synthetic histories, ensuring alignment with user style and preferences. Extensive experiments on three benchmark personalized generation datasets show that GRASPER achieves significant performance gain, substantially improving personalization in sparse user context settings.

1 INTRODUCTION

Personalized Large Language Models (LLMs) have recently garnered significant attention (Salemi et al., 2023; Tsai et al., 2024) due to their various downstream applications in search, recommendation, and conversational agents (Yoganarasimhan, 2019; Qian et al., 2014; Shumanov & Johnson, 2021). By retrieving relevant personal context from user history, LLMs can produce outputs that are tailored to the given user’s personal preferences and enhance overall satisfaction and quality.

The core of LLM personalization lies in retrieving personal context, typically derived from a user’s history. However, most existing approaches emphasize textual histories (Salemi et al., 2023; 2024a). While useful, these histories are often sparse and limited, which severely constrains personalization for long-tail users due to insufficient context. For example, in e-commerce platforms and social

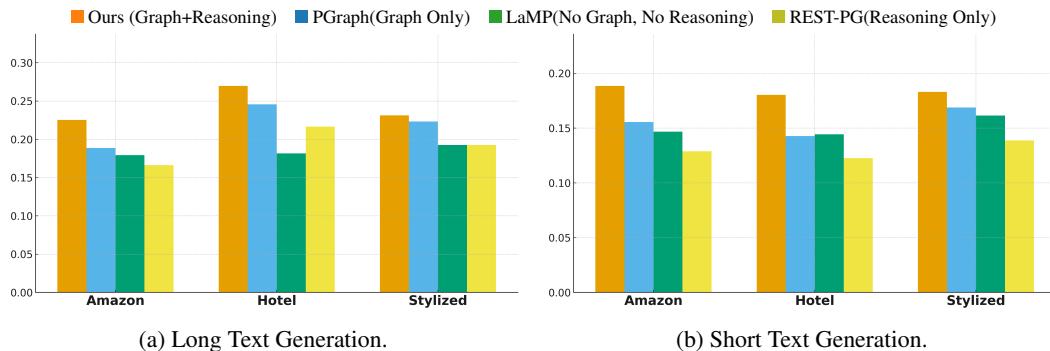


Figure 1: Results comparing our approach across two fundamental tasks and across datasets. Legends use descriptive labels: Ours (Graph + Reasoning), PGraph (Graph Only), LaMP (No Graph, No Reasoning), and REST-PG (Reasoning Only). GRASPER achieves over 10% gains, on average, across datasets.

054 networks, more than 95% of users tend to be cold-start (Au et al., 2025; Ni & McAuley, 2018). Recent
 055 studies show that incorporating auxiliary product reviews from other users—naturally represented as
 056 user-item graphs that connect users, items, and their interactions—can effectively improve model
 057 performance in settings such as recommendation and text generation (Au et al., 2025; Wang et al.,
 058 2025). By leveraging such graph-based histories, language models are able to generate more accurate
 059 and contextually rich personalized responses.

060 Despite these advancements, how to effectively leverage diverse data sources, such as graphs, to
 061 improve data sparsity in personalized LLMs remains underexplored. While prior work (Au et al.,
 062 2025) focused on retrieving existing textual histories for target items, it ignores the potential of
 063 more complex, structural information in the user-item interaction graphs. Such rich structural data
 064 could provide crucial signals for users with otherwise sparse textual histories, as shown in recent
 065 works (Wang et al., 2025). Furthermore, the integration of varied contextual information often
 066 lacks a dedicated reasoning phase before generation. We argue that this reasoning step is crucial as
 067 the diversity and volume of retrieved information requires the LLM to strategically synthesize and
 068 generate coherent and personalized outputs (Salemi et al., 2025).

069 To address these challenges, we propose GRASPER, a framework with two key stages: graph-based
 070 augmentation and reasoning-aligned generation. In the augmentation stage, we integrate a pretrained
 071 link predictor to enrich the sparse user context by simulating potential future interactions. In the
 072 generation stage, the personalized text generation model leverages both the augmented history of the
 073 user and existing texts of the target item to craft tailored responses. Crucially, reasoning is explicitly
 074 incorporated into both stages: during augmentation, reasoning guides the generation of features for the
 075 simulated edges to ensure alignment with user preferences; during text generation, reasoning enforces
 076 consistency between the synthesized history and the final personalized response (Salemi et al.,
 077 2025). Extensive experiments demonstrate that GRASPER substantially outperforms state-of-the-art
 078 baselines.

079 Our contribution can be summarized as follows:

- 080 • We introduce GRASPER, a novel framework that tackles sparsity in personalized text
 081 generation by combining graph learning and reasoning. The graph-based augmentation
 082 expands limited user histories with simulated texts of predicted interactions, while reasoning
 083 ensures coherence between augmented and original data.
- 084 • To the best of our knowledge, this is the first work to explicitly integrate reasoning into
 085 sparse personalization, enabling LLMs to cohesively synthesize augmented context and
 086 generate faithful, user-aligned outputs.
- 087 • We conduct extensive experiments on real-world datasets (Amazon (Ni & McAuley, 2018),
 088 Hotel (Kanouchi et al., 2020), and Stylized Feedback (Alhafni et al., 2024)), showing that
 089 GRASPER achieves, on average, over 10% improvement on state-of-the-art baselines for
 090 text generation, and 15% for rating prediction.

092 2 PROBLEM DEFINITION

093 Most existing works on personalized text generation assume that users have access to rich personal
 094 contexts, typically requiring more than ten historical text entries (Au et al., 2025; Salemi et al., 2023).
 095 However, this assumption does not align with real-world usage patterns, where user data often follows
 096 a long-tail distribution—over 95% of users have written fewer than two text entries (Ni & McAuley,
 097 2018; Au et al., 2025). This data sparsity poses a significant challenge for personalization systems
 098 that rely heavily on individual user histories.

099 We argue that the naturally occurring bipartite user-item interaction graph, constructed from collective
 100 user-item histories, offers a valuable source of auxiliary context (Zhao et al., 2021b;a). It encodes
 101 implicit relationships between users and items through shared interactions and textual feedback,
 102 providing a structural foundation to infer user preferences even in low-resource settings.

103 Thus, to address the sparsity challenge and enrich the user context in such cases, we follow the setting
 104 in prior work (Au et al., 2025) and define the task of *graph-based personalized text generation*. In
 105 this task, the goal is to leverage both the sparse personal history of a user and the broader structural
 106 context encoded in the user-item graph to generate personalized text (e.g., reviews) for a target item.

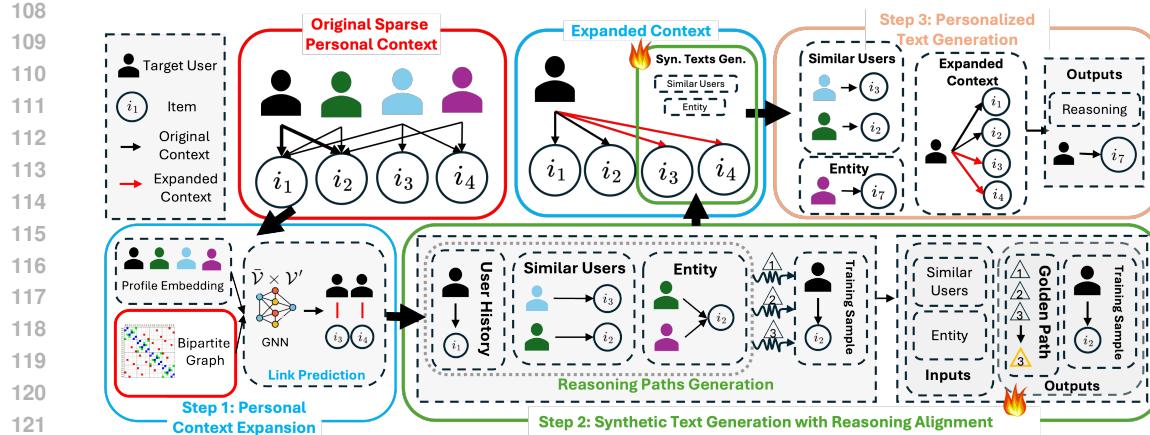


Figure 2: Overview of the proposed GRASPER. In **Step 1**, the personal context is enriched by leveraging the underlying graph structure to predict potential interactions. In **Step 2**, we generate synthetic reviews for the predicted interactions with aligned reasoning. In **Step 3**, the enhanced personal context that contains both the observed and simulated interactions enables the generation of more accurate and personalized text.

Personalized Text Generation with Graph. We formally define the task in this section. Let $\mathcal{U} = \{u_1, u_2, \dots, u_n\}$ denote the set of n users, and $\mathcal{I} = \{i_1, i_2, \dots, i_m\}$ denote the set of m target items (e.g., products, hotels). For each user $u \in \mathcal{U}$, we define their interaction history as a sequence

$$H_u = [(i_{u,1}, t_{u,1}), \dots, (i_{u,k_u}, t_{u,k_u})],$$

where k_u is the number of interactions of user u (which may vary across users), $i_{u,\ell} \in \mathcal{I}$ is the ℓ -th item interacted with by u , $t_{u,\ell} \in \mathcal{T}$ is the associated text, such as the review written by a customer or a comment written by a social media user, and \mathcal{T} is the space of all possible text.

Following previous work (Au et al., 2025), we model the interactions as a bipartite graph $G = (\mathcal{U} \cup \mathcal{I}, \mathcal{E})$, where each edge $(u, i) \in \mathcal{E}$ exists if $(i, t) \in H_u$ for some text t and user u .

The goal of personalized text generation is to learn a function:

$$f : \mathcal{U} \times \mathcal{H} \times \mathcal{I} \rightarrow \mathcal{T}$$

that generates a personalized text t_{u,i^*} for a given user $u \in \mathcal{U}$, their history $H_u \in \mathcal{H}$, a new target item $i^* \in \mathcal{I}$ and the bipartite graph G :

$$t_{u,i^*} = f(u, H_u, i^*),$$

where \mathcal{H} is the space of all possible user histories.

3 METHOD

As illustrated in Figure 2, GRASPER operates in three steps to address the challenge of personalization under sparse context: **(Step 1) Personal Context Expansion via Link Prediction**, which augments sparse histories with potential user–item interactions predicted from a graph model; **(Step 2) Synthetic Review Generation with Reasoning Alignment**, which integrates intermediate reasoning paths into the text generation process to ensure personalization alignment; and **(Step 3) Personalized Text Generation**, which produces personalized outputs conditioned on the augmented personal context.

3.1 STEP 1: PERSONAL CONTEXT EXPANSION

Personalization often suffers in the long-tail setting where users have limited history, as limited contexts constrain downstream LLM generation quality and further compromise personalization. To mitigate this limitation, we propose enriching sparse histories by predicting additional user–item interactions through graph-based link prediction. These predicted edges allow us to construct a more comprehensive profile for each user, which can then be leveraged in later reasoning alignment and generation stages. To simulate additional user–item interactions, we perform link prediction over the bipartite user–item interaction graph $G = (\mathcal{U} \cup \mathcal{I}, \mathcal{E})$ defined in Section 2.

162 **Training Graph-based Link Predictor.** Each node $v \in \mathcal{U} \cup \mathcal{I}$ is initialized with an embedding
 163 $\mathbf{h}_v^{(0)} \in \mathbb{R}^d$ derived from textual features using a pretrained encoder:
 164

$$\mathbf{h}_v^{(0)} = \begin{cases} \text{Enc}(T_u), & v = u \in \mathcal{U}, \\ \text{Enc}(R_i), & v = i \in \mathcal{I}, \end{cases} \quad (1)$$

165 where $\text{Enc}(\cdot)$ denotes a sentence encoder mapping text to a d -dimensional embedding, T_u is the
 166 concatenated string of the sequence of text in user u 's profile $T_u = \text{Concat}[t_{u,1}, \dots, t_{u,k_u}]$, and
 167 $R_i = \{t_{u,i} \mid (u, i) \in E\}$ is the set of texts aggregated for item i . Enc can be text embedding
 168 model (Reimers & Gurevych, 2019). Since style is a crucial element for personalization compared
 169 to content itself (Zhang et al., 2025), we instantiate Enc as a style-aware encoder that disentangles
 170 stylistic elements from content following Wegmann et al. (2022).
 171

172 With the initialized node features, we adopt GraphSAGE (Hamilton et al., 2018) for inductive graph
 173 encoding. At each layer l , a node v updates its representation by first aggregating its neighbors $\mathcal{N}(v)$
 174 and then combining this aggregation with its own embedding:
 175

$$\mathbf{m}_v^{(l)} = \text{AGG}^{(l)} \left(\{ \mathbf{h}_u^{(l-1)} : u \in \mathcal{N}(v) \} \right), \quad \mathbf{h}_v^{(l)} = \sigma \left(\mathbf{W}^{(l)} [\mathbf{h}_v^{(l-1)} \parallel \mathbf{m}_v^{(l)}] \right) \quad (2)$$

176 where $\text{AGG}^{(l)}$ is a permutation-invariant mean aggregator, $\mathbf{W}^{(l)}$ is a learnable projection matrix,
 177 \parallel denotes concatenation, and σ is a non-linear activation function (ReLU in our implementation).
 178 After L layers, we obtain the final node embedding $\mathbf{z}_v = \mathbf{h}_v^{(L)}$. In conjunction with the style-aware
 179 initialization, \mathbf{z}_v captures both preference and stylistic elements of the personalization target.
 180

181 To estimate the likelihood of a new interaction between a user u and an item i , we apply a decoder
 182 over their embeddings. Specifically, we compute a score with a multi-layer perception (MLP) and
 183 turn it into a probability as:
 184

$$s_{u,i*} = \text{MLP}([\mathbf{z}_u \parallel \mathbf{z}_{i*}]), \quad \hat{y}_{u,i*} = \text{Sigmoid}(s_{u,i*}). \quad (3)$$

185 The link predictor is trained with binary cross-entropy (BCE) loss, where observed interactions are
 186 positives and uniformly sampled non-interacted pairs are negatives:
 187

$$\mathcal{L}_{\text{link}} = - \sum_{(u,i) \in E^+} \log \sigma(s_{u,i}) - \sum_{(u,i) \in E^-} \log(1 - \sigma(s_{u,i})), \quad (4)$$

188 with $E^+ = E$ and E^- constructed via negative sampling (Zhang & Chen, 2018).
 189

190 **Inference and Profile Augmentation.** At inference, for each sparse user u , we score all candidate
 191 items $i \in \mathcal{I} \setminus \{i : (u, i) \in \mathcal{E}\}$, rank them by $s_{u,i*}$, and select the top- K predictions. Let $\mathcal{I}_u^K =$
 192 $\{i_{u,1}, \dots, i_{u,K}\}$ denote this top- K set. For each $i \in \mathcal{I}_u^K$, we generate a synthetic text (as detailed in
 193 Step 2) to approximate how u might interact with it, and these texts are appended to u 's history. The
 194 result is an augmented user profile that incorporates both observed and predicted interactions:
 195

$$\tilde{H}_u = H_u \cup \{ \tilde{t}_{u,i_{u,k}} : k = 1, \dots, K \}. \quad (5)$$

196 where $\tilde{t}_{u,i}$ explicitly denotes the synthetically generated text for user u on item i . Importantly, these
 197 predicted edges are used only locally for the given user u . For example, an edge (u, i) generated for
 198 user u does not affect another user u' through the shared item i . Thus, we are not reconstructing a full
 199 new bipartite graph but enriching each user profile independently for downstream personalization.
 200

211 3.2 STEP 2: SYNTHETIC TEXT GENERATION WITH REASONING ALIGNMENT

212 The augmented profiles from Step 1 yield predicted candidate items, but directly generating texts
 213 from these signals risks propagating noise or stylistic mismatch. To address this, we design a
 214 reasoning-based synthetic text generation process that integrates explicit reasoning before producing
 215 final outputs.

216 **Synthetic Text Generation Setup.** For a target user u , let \mathcal{S}_u denote the set of similar users
 217 identified from the user-item graph G , and let \mathcal{I}_u be the set of items inferred in Step 1 that u is likely
 218 to interact with. \mathcal{S}_u is obtained by calculating the cosine similarity between the node embeddings
 219 \mathbf{z}_v . We select the top 3 similar users to construct $H_{\mathcal{S}_u}$. For each item $i \in \mathcal{I}_u$, we aim to generate a
 220 synthetic text $\tilde{y}_{u,i}$ that reflects u 's style and preferences. The generation process conditions on three
 221 sources of input:

$$222 \quad x = \{H_u, H_{\mathcal{S}_u}, P_{u,i}\}, \quad (6)$$

223 where H_u are past texts of u , $H_{\mathcal{S}_u}$ are texts written by similar users, and $P_{u,i}$ are peer texts associated
 224 with item i . Following Au et al. (2025), $P_{u,i}$ is constructed by ranking all texts associated with item i
 225 using the BM25 retrieval model (Robertson et al., 1995) and selecting the top 4 most relevant entries,
 226 where relevance is measured by semantic similarity to the input query.

227 **Reasoning Path Generation.** We use a language model \mathcal{M} to produce intermediate rationales that
 228 explain why u might write for item i in a certain way. Formally, a reasoning path \mathcal{Z} is an intermediate
 229 textual explanation conditioned on x , such as “*User u tends to prefer lightweight laptops; similar
 230 users highlighted battery life for item i ; hence the review should emphasize portability and battery.*”
 231 During training, for each text entry $t_{u,j}$, we let $x = \{H_u \setminus t_{u,j}, H_{\mathcal{S}_u}, P_{u,j}\}$. We then obtain a set of
 232 candidate reasoning paths by sampling with a formatted prompt $\phi(x, t_{u,j})$ (The prompt formulation is
 233 supplied in Appendix G) that includes the input and the expected output

$$234 \quad \mathcal{Z}^{(r)} = \mathcal{M}(\phi(x, t_{u,j})), \quad r = 1, 2, \dots, R. \quad (7)$$

235 We sample the candidate reasoning paths because not all candidate reasoning paths are equally reliable.
 236 We select a *golden* reasoning path \mathcal{Z}^* that best aligns with ground-truth outputs by maximizing task
 237 performance under an evaluation metric Ω , which we take the average of the ROUGE and METEOR
 238 scores (see Appendix C.4 for more details on the evaluation metrics)

$$239 \quad \mathcal{Z}^* = \arg \max_{\mathcal{Z}} \Omega(t'_{u,j}, t_{u,j}), \quad \text{where } t'_{u,j} = \mathcal{M}(\xi(x, \mathcal{Z}^{(r)})), \quad (8)$$

240 with $t'_{u,j}^{(k)}$ denoting the generated synthetic text, and ξ a prompt-construction function combining x
 241 and \mathcal{Z} . The specification of ξ is supplied in Appendix G.

242 **Reasoning Alignment.** Finally, the model is fine-tuned to jointly generate the selected reasoning
 243 path \mathcal{Z}^* and text $t_{u,j}$:

$$244 \quad \mathcal{L}_{\text{gen}} = \text{CE}(\mathcal{M}(\rho(x)), t_{u,j}), \quad (9)$$

245 where ρ is the prompt formatting function that is designed to generate both the reasoning and the
 246 output (The prompt formulation is supplied in Appendix G). $\mathcal{M}(\rho(x))$ is trained to output \mathcal{Z}^*
 247 followed by the text. It enables the model to leverage noisy augmentation while staying faithful to
 248 user-specific style and preferences. Let the fine-tuned model be \mathcal{M}' , for a predicted item $i \in \mathcal{I}_u^K$, the
 249 final synthetic text $\hat{t}_{u,i}$ is then generated with $\mathcal{M}'(x) \setminus \mathcal{Z}'$ where \mathcal{Z}' is the generated reasoning.

250 3.3 STEP 3: PERSONALIZED TEXT GENERATION

251 The final stage reuses the reasoning aligned language model from Step 2, which defines a mapping
 252 from user histories, similar-user signals, and candidate items to reasoning paths and synthetic texts.
 253 In Step 3, we employ the same function for the personalization task.

254 Let \mathcal{I}_u be the set of candidate items inferred in Step 1. For each $i \in \mathcal{I}_u$, Step 2 already learns to
 255 map from the user's profile history, texts written by similar users, and peer texts associated with the
 256 predicted target item, as shown in Equation (6). Note that for personalized text generation for a
 257 specific target item i^* , we have the same input formulation $x^* = [\tilde{H}_u, H_{\mathcal{S}_u}, P_{u,i^*}]$ where instead of
 258 profile history, \tilde{H}_u is the augmented history obtained from Step 1. Since the reasoning-generation
 259 function is shared, we reuse the fine-tuned model \mathcal{M}' in Step 2 for the target text generation. Given
 260 the input x^* , the fine-tuned model \mathcal{M}' produces both a reasoning path z^* and a personalized text
 261 \hat{t}_{u,i^*} :

$$262 \quad \mathcal{M}'(x^*) = [z^* \parallel \hat{t}_{u,i^*}]. \quad (10)$$

263 The final prediction strips away the reasoning tokens:

$$264 \quad \hat{t}_{u,i^*} = \mathcal{M}'(x^*) \setminus z^*. \quad (11)$$

270 Table 1: Results comparing the proposed approach called GRASPER to state-of-the-art methods
 271 across 3 different tasks on the Amazon Review benchmark.

273	Task	Metric	LLM GRASPER (ours)	PGraph	LaMP	REST-PG
274	LONG TEXT GEN.	ROUGE-1 \uparrow	<i>4o-mini LLaMA3</i> 0.219 0.215	0.189 0.178	0.171 0.173	N/A 0.165
275		ROUGE-L \uparrow	<i>4o-mini LLaMA3</i> 0.170 0.171	0.130 0.129	0.117 0.129	N/A 0.109
276		METEOR \uparrow	<i>4o-mini LLaMA3</i> 0.182 0.178	0.196 0.151	0.176 0.138	N/A 0.122
277		LLM-as-a-Judge \uparrow	<i>4o-mini LLaMA3</i> 0.421 0.337	0.389 0.297	0.328 0.277	N/A 0.269
278	SHORT TEXT GEN.	ROUGE-1 \uparrow	<i>4o-mini LLaMA</i> 0.178 0.155	0.115 0.131	0.108 0.124	N/A 0.089
279		ROUGE-L \uparrow	<i>4o-mini LLaMA3</i> 0.174 0.153	0.112 0.125	0.105 0.118	N/A 0.081
280		METEOR \uparrow	<i>4o-mini LLaMA3</i> 0.162 0.142	0.099 0.125	0.091 0.117	N/A 0.113
281		LLM-as-a-Judge \uparrow	<i>4o-mini LLaMA3</i> 0.406 0.304	0.353 0.241	0.334 0.228	N/A 0.232
282	RATING PRED. (Recommendation)	RMSE \downarrow	<i>4o-mini LLaMA3</i> 0.33 0.32	0.38 0.76	0.34 0.72	N/A 0.65
283		MAE \downarrow	<i>4o-mini LLaMA3</i> 0.34 0.31	0.73 0.34	0.70 0.31	N/A 0.46

290 4 EXPERIMENTS

291 4.1 DATASETS AND METRICS

292 We follow the experiment setup in prior works (Au et al., 2025), which consists of three datasets:
 293 Amazon Review (Ni & McAuley, 2018), Hotel Review (Kanouchi et al., 2020), and Stylized Feed-
 294 backs (Alhafni et al., 2024). We cover three personalization tasks: long text generation, short text
 295 generation, and rating prediction. For long text generation, a title will be given to guide the generation,
 296 and for short text generation, a paragraph will be given for summarization.

297 For both long and short text generation tasks, we adopt widely used lexical overlap metrics, including
 298 ROUGE-1, ROUGE-L, and METEOR, following prior work (Au et al., 2025; Kumar et al., 2024;
 299 Salemi et al., 2023). To complement these surface-level metrics, we further incorporate LLM-as-
 300 a-Judge evaluation (Salemi et al., 2025; Liu et al., 2023), where a strong language model provides
 301 comparative assessments of personalization. Additional details on LLM-as-a-Judge can be found in
 302 Appendix B. For rating prediction, we use Root Mean Squared Error (RMSE) and Mean Average
 303 Error (MAE). Further details on the metrics and experimental setup can be found in Appendix C.

304 4.2 BASELINES

305 We benchmark against three state-of-the-art personalization baselines. **LaMP** (Salemi et al.,
 306 2023) conditions on a user’s past writing via prompts but uses no graph learning or reasoning.
 307 **PGraphRAG** (Au et al., 2025) (i.e., PGraph) employs graph-based retrieval augmented generation
 308 with BM25 for personalization but lacks reasoning or fine-tuning. **REST-PG** (Salemi et al., 2025)
 309 models user preferences through reasoning paths with iterative fine-tuning, but it does not employ
 310 any context expansion. Expanded baseline descriptions can be found in Appendix C. We evaluate
 311 LaMP and PGraph with LLaMA-3-8b-instruct and GPT-4o mini; REST-PG is implemented with
 312 LLaMA-3-8b-instruct only due to its fine-tuning requirement.

313 4.3 MAIN RESULTS

314 We report the experimental results for the Amazon Reviews, Hotel Experience, and Stylized Feedback
 315 datasets in Tables 1 to 3, respectively. These results evaluate our approach across three tasks—long
 316 text generation, short text generation, and ordinal classification—and compare its performance to the
 317 state-of-the-art personalization baselines. Additional results are provided in Appendix E.

324
325
326 Table 2: Performance Metrics for Hotel Experience Generation.
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342

Task	Metric	LLM	GRASPER (ours)	PGraph	LaMP	REST-PG
LONG TEXT GEN.	ROUGE-1 \uparrow	<i>4o-mini</i> <i>LlaMA3</i>	0.257 0.258	0.263 0.263	0.221 0.199	N/A 0.221
	ROUGE-L \uparrow	<i>4o-mini</i> <i>LlaMA3</i>	0.163 0.168	0.152 0.157	0.135 0.129	N/A 0.131
	METEOR \uparrow	<i>4o-mini</i> <i>LlaMA3</i>	0.165 0.165	0.184 0.191	0.164 0.152	N/A 0.145
	LLM-as-a-Judge \uparrow	<i>4o-mini</i> <i>LlaMA3</i>	0.500 0.488	0.414 0.372	0.300 0.246	N/A 0.369
SHORT TEXT GEN.	ROUGE-1 \uparrow	<i>4o-mini</i> <i>LlaMA3</i>	0.135 0.147	0.112 0.127	0.108 0.126	N/A 0.100
	ROUGE-L \uparrow	<i>4o-mini</i> <i>LlaMA3</i>	0.128 0.140	0.111 0.118	0.104 0.117	N/A 0.091
	METEOR \uparrow	<i>4o-mini</i> <i>LlaMA3</i>	0.120 0.130	0.081 0.102	0.075 0.106	N/A 0.084
	LLM-as-a-Judge \uparrow	<i>4o-mini</i> <i>LlaMA3</i>	0.469 0.304	0.360 0.224	0.346 0.228	N/A 0.215
RATING PRED.	RMSE \downarrow	<i>4o-mini</i> <i>LlaMA3</i>	0.660 0.322	0.328 0.347	0.340 0.326	N/A 0.335
	MAE \downarrow	<i>4o-mini</i> <i>LlaMA3</i>	0.356 0.520	0.336 0.724	0.700 0.680	N/A 0.642

343
344 Table 3: Performance Metrics for Stylized Feedback Generation.
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362

Task	Metric	LLM	GRASPER (ours)	PGraph	LaMP	REST-PG
LONG TEXT GEN.	ROUGE-1 \uparrow	<i>4o-mini</i> <i>LlaMA3</i>	0.214 0.235	0.185 0.217	0.187 0.186	N/A 0.189
	ROUGE-L \uparrow	<i>4o-mini</i> <i>LlaMA3</i>	0.147 0.175	0.123 0.158	0.123 0.134	N/A 0.127
	METEOR \uparrow	<i>4o-mini</i> <i>LlaMA3</i>	0.178 0.175	0.183 0.178	0.189 0.177	N/A 0.173
	LLM-as-a-Judge \uparrow	<i>4o-mini</i> <i>LlaMA3</i>	0.423 0.340	0.399 0.340	0.318 0.273	N/A 0.281
SHORT TEXT GEN.	ROUGE-1 \uparrow	<i>4o-mini</i> <i>LlaMA3</i>	0.137 0.160	0.122 0.149	0.113 0.140	N/A 0.097
	ROUGE-L \uparrow	<i>4o-mini</i> <i>LlaMA3</i>	0.133 0.157	0.118 0.142	0.109 0.134	N/A 0.091
	METEOR \uparrow	<i>4o-mini</i> <i>LlaMA3</i>	0.144 0.131	0.104 0.142	0.096 0.136	N/A 0.112
	LLM-as-a-Judge \uparrow	<i>4o-mini</i> <i>LlaMA3</i>	0.395 0.284	0.343 0.242	0.331 0.236	N/A 0.255
RATING PRED.	RMSE \downarrow	<i>4o-mini</i> <i>LlaMA3</i>	0.637 0.684	0.673 0.724	0.667 0.680	N/A 0.678
	MAE \downarrow	<i>4o-mini</i> <i>LlaMA3</i>	0.332 0.337	0.347 0.347	0.344 0.327	N/A 0.326

363 Overall, our method consistently outperforms the baselines across the datasets and tasks. For long text
364 generation, we observe significant improvements in ROUGE-1, ROUGE-L, and METEOR scores,
365 demonstrating the model’s ability to generate more accurate and contextually relevant outputs through
366 reasoning enhanced retrieval and generation. Notably, the Amazon Reviews dataset shows the largest
367 performance gains. This can be attributed to it having the fewest average degree of 1.68 comparing to
368 2.12 and 2.42 for the Hotel Experience dataset and Stylized Feedback dataset (see Appendix C.1).
369 The increased sparsity leads to better performance gain with the proposed method. For ordinal
370 classification, our approach achieves lower RMSE and MAE compared to the baselines, indicating
371 better alignment with user rating tendencies. Furthermore, we find that the relative advantage of our
372 method becomes more pronounced under LLM-as-a-Judge evaluation. Compared with conventional
373 textual similarity metrics, which primarily capture surface-level overlap, LLM-as-a-Judge better
374 aligns with human preference when assessing personalization (Salemi et al., 2025; Liu et al., 2023).
375 This is because personalization often extends beyond literal similarity to reflect nuanced aspects
376 such as style, tone, and contextual coherence. The gains observed in this setting demonstrate that
377 GRASPER, by jointly leveraging context expansion and reasoning alignment, is able to generalize
378 user-specific stylistic patterns more effectively, leading to outputs that are not only accurate but also
379 more faithful to individual user preferences.

378 Table 4: Module ablation studies for Amazon Reviews dataset with the Llama3 backbone. GRASPER-
 379 ft is an ablation without fine-tuning, and GRASPER-r-ft is without reasoning and fine-tuning.
 380

Task	Metric	GRASPER	GRASPER-ft	GRASPER-r-ft	PGraph	LaMP	REST-PG
Text Generation	ROUGE-1	0.215	0.175	0.182	0.178	0.173	0.165
	ROUGE-L	0.171	0.121	0.125	0.129	0.129	0.009
	METEOR	0.178	0.188	0.200	0.151	0.138	0.122
Title Generation	ROUGE-1	0.155	0.100	0.118	0.131	0.124	0.112
	ROUGE-L	0.153	0.098	0.117	0.125	0.118	0.077
	METEOR	0.142	0.085	0.111	0.125	0.117	0.113
Rating Prediction	RMSE	0.32	0.53	0.52	0.52	0.72	0.65
	MAE	0.31	0.39	0.36	0.34	0.31	0.46

388 389 4.4 ABLATION STUDIES 390

391 We conduct extensive ablations on GRASPER, examining its components to validate the framework’s
 392 effectiveness. We further analyze the key hyperparameter K (number of predicted items) along with
 393 supporting theory. Results using additional language models are presented in Appendix E.
 394

395 4.4.1 MODEL VARIANTS 396

397 In Table 4, we ablate GRASPER with additional variants to demonstrate the effectiveness of its two
 398 main contributions: personal context expansion and reasoning alignment. GRASPER-ft removes the
 399 fine-tuning for reasoning, which means it will only include the reasoning prompt. GRASPER-r-ft
 400 further removes the reasoning process and lets the model directly generate the final output with the
 401 input as specified in Equation (6). The full model (GRASPER) consistently outperforms its reduced
 402 counterparts, confirming that each component is indispensable to the framework.
 403

404 **Effect of Context Expansion.** Personal context expansion provides additional evidence for personal-
 405 alization, but without proper reasoning, the augmented context can introduce noise. This is reflected in
 406 GRASPER-ft-r, which relies solely on link prediction without reasoning or fine-tuning. These results
 407 indicate that context expansion alone is insufficient and may even hurt performance if not paired with
 408 reasoning alignment. The effect of noise-induced bias is extensively discussed in Appendix F.
 409

410 **Effect of Reasoning Alignment.** Reasoning alignment ensures that the augmented context contributes
 411 in a way that matches user preferences and task requirements. Comparing GRASPER-ft (with
 412 reasoning but no fine-tuning) to GRASPER shows that reasoning alignment improves performance
 413 across metrics. Additionally, reasoning alignment without additional context (as in REST-PG) also
 414 underperforms, since the model lacks sufficient personalized evidence to reason over.
 415

416 The results demonstrate that context expansion and reasoning alignment are complementary. With
 417 only context expansion (GRASPER-r-ft), the model introduces noise and degrades performance.
 418 With only reasoning, even when aligned with finetuning (e.g., REST-PG), the model has nothing
 419 substantial to reason over. Only by combining both can GRASPER achieve better personalization
 420 across text generation and rating prediction.
 421

422 4.4.2 HYPERPARAMETER ANALYSIS 423

424 The hyperparameter K as defined in Equation (5) controls how many candidate items we add via the
 425 link predictor when augmenting a user’s personal context. In our main experiments, we fix $K = 2$ for
 426 efficiency, but the design of our method allows more effective and robust use with larger values of K .
 427 By comparison, PGraph expands context by retrieving K nearest reviews to the query embedding.
 428 However, because it lacks explicit reasoning alignment, its performance drops as K grows — the
 429 extra retrieved context introduces more noise than benefit (as seen in Table 5).
 430

431 Our theoretical analysis (detailed in Appendix F) shows that this phenomenon follows a bias–variance
 432 trade-off. Adding more synthetic context reduces variance (helping especially sparse users with few
 433 real samples), but also introduces bias from preference mismatch. Without alignment, the bias term
 434 grows with K , limiting the benefit. However, with reasoning alignment, the mismatch is effectively
 435 shrunk: the bias is reduced, so larger K values become safe and beneficial. This explains why in
 436 experiments our method continues to improve as K increases, while PGraph plateaus or even degrades.
 437 Full mathematical details of this trade-off are provided in the Proposition 1.
 438

432 Table 5: Sensitivity of GRASPER and PGraph to the neighborhood size k on the Amazon Reviews
 433 dataset with the Llama3 backbone. Best in **bold**, second-best underlined.

Task	Metric	GRASPER				PGraph			
		$K=1$	$K=2$	$K=3$	$K=4$	$K=1$	$K=2$	$K=3$	$K=4$
Long Text Gen.	ROUGE-1	0.226	0.215	0.222	0.225	0.160	0.169	0.169	0.173
	ROUGE-L	0.172	0.171	0.173	0.177	0.121	0.125	0.124	0.124
	METEOR	0.184	0.178	<u>0.189</u>	0.196	0.125	0.138	0.149	0.150
Short Text Gen.	ROUGE-1	0.168	0.155	0.168	0.195	0.128	0.123	0.112	0.125
	ROUGE-L	<u>0.173</u>	0.153	0.165	0.192	0.121	0.118	0.107	0.119
	METEOR	0.160	0.142	<u>0.169</u>	0.185	0.123	0.128	0.116	0.117

Input	GraSPeR	PGraph	Ground Truth
Title: Disappointing. Profile: {}	The spray bottles broke in the first use. They were not worth the money.	Disappointment.	Didn't hold up. One broke apart first time used. Ended up buying different spray bottles.
Title: Great Option! Profile: {}	This is a great option! Love this product. It's gentle and leaves my skin feeling clean and soft. I would definitely recommend this to anyone who wants a gentle facial scrub that smells great.	This product is an excellent choice, it's really effective and leaves my skin feeling so smooth	This is a great option for an exfoliating cleanser! Love that there is an approachable option that is of a decent quality and isn't excessively costly.
Title: Nice. Profile: {}	I like it. It's comfortable and easy to use.	Looks good, nice and soft.	I like this - it does seem well made and it's just easy to wear.

451 Figure 3: Case study with three examples. The matching green, blue, and yellow boxes show
 452 matching semantics or expression. The red box shows misalignment against the ground truth.
 453

4.5 CASE STUDY

457 To better illustrate how GRASPER generates more faithful and personalized outputs, we present a
 458 case study in Figure 3. We compare outputs from GRASPER and PGraph against the ground truth
 459 under different input titles. As highlighted by the colored boxes, GRASPER consistently captures key
 460 semantics and stylistic expressions that align with the reference. For instance, in the first example,
 461 GRASPER generates “spray bottles broke in the first use,” which mirrors both the semantics (“first
 462 use”) and specific product mentioned (“spray bottles”) in the ground truth. In contrast, PGraph only
 463 outputs a vague summary (“Disappointment”) without grounding in the product context.

464 In the second example, GRASPER reproduces stylistic markers such as “great option” and “Love”.
 465 Although PGraph’s output “excellent choice” also captures the semantic meaning, it diverges from
 466 the ground truth in word choice. Finally, in the third example, both GRASPER and the ground truth
 467 emphasize short, colloquial phrasing (“I like it / I like this”). GRASPER also correctly matches
 468 the emphasis on usability (“easy to use / easy to wear”), while PGraph generates more general,
 469 less faithful wording (“Looks good, nice and soft”), deviating from the intended expression. These
 470 qualitative examples support our quantitative findings: context expansion and reasoning alignment
 471 together enable GRASPER to preserve fine-grained semantics and stylistic fidelity, while methods
 472 that rely only on raw augmentation often produce generic or misaligned outputs.

5 CONCLUSION

476 In this work, we proposed GRASPER, a reasoning-based framework for personalized text generation
 477 under sparse user contexts. By combining graph-based context expansion with explicit reasoning
 478 alignment, our method effectively enriches limited personal histories while ensuring generated outputs
 479 remain faithful to user style and preferences. Extensive experiments across datasets in different
 480 domains demonstrate that GRASPER significantly outperforms strong baselines.

481 Our findings highlight that context expansion and reasoning alignment are complementary: expansion
 482 alone risks introducing noise, while reasoning without sufficient context lacks grounding. Together,
 483 they enable models to better capture fine-grained semantics and stylistic fidelity, even for long-
 484 tail users with minimal histories. We believe this reasoning-enhanced paradigm opens promising
 485 directions for future research in large language model personalization, especially in real-world
 486 scenarios where data sparsity is the norm.

486 REFERENCES
487

488 Bashar Alhafni, Vivek Kulkarni, Dhruv Kumar, and Vipul Raheja. Personalized text generation
489 with fine-grained linguistic control. In Ameet Deshpande, EunJeong Hwang, Vishvak Murahari,
490 Joon Sung Park, Diyi Yang, Ashish Sabharwal, Karthik Narasimhan, and Ashwin Kalyan (eds.),
491 *Proceedings of the 1st Workshop on Personalization of Generative AI Systems (PERSONALIZE*
492 *2024)*, pp. 88–101, St. Julians, Malta, March 2024. Association for Computational Linguistics.
493 URL <https://aclanthology.org/2024.personalize-1.8>.

494 Steven Au, Cameron J. Dimacali, Ojasmitra Pedirappagari, Namyong Park, Franck Dernoncourt,
495 Yu Wang, Nikos Kanakaris, Hanieh Deilamsalehy, Ryan A. Rossi, and Nesreen K. Ahmed.
496 Personalized graph-based retrieval for large language models, 2025. URL <http://arxiv.org/abs/2501.02157>.

497 Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi,
498 Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczek, and Torsten Hoefer. Graph
499 of thoughts: Solving elaborate problems with large language models. *Proceedings of the AAAI*
500 *Conference on Artificial Intelligence*, 38(16):17682–17690, March 2024. ISSN 2159-5399. doi: 10.
501 1609/aaai.v38i16.29720. URL <http://dx.doi.org/10.1609/aaai.v38i16.29720>.

502 Millennium Bismay, Xiangjue Dong, and James Caverlee. Reasoningrec: Bridging personalized
503 recommendations and human-interpretable explanations through llm reasoning, 2024. URL
504 <https://arxiv.org/abs/2410.23180>.

505 William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs,
506 2018. URL <https://arxiv.org/abs/1706.02216>.

507 Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language
508 and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*, 2024.

509 Shin Kanouchi, Masato Neishi, Yuta Hayashibe, Hiroki Ouchi, and Naoaki Okazaki. You may
510 like this hotel because: Identifying evidence for explainable recommendations. In Kam-
511 Fai Wong, Kevin Knight, and Hua Wu (eds.), *Proceedings of the 1st Conference of the Asia-
512 Pacific Chapter of the Association for Computational Linguistics and the 10th International
513 Joint Conference on Natural Language Processing*, pp. 890–899, Suzhou, China, December
514 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.aacl-main.89. URL
515 <https://aclanthology.org/2020.aacl-main.89>.

516 Jieyong Kim, Tongyoung Kim, Soojin Yoon, Jaehyung Kim, and Dongha Lee. Llms think, but not
517 in your flow: Reasoning-level personalization for black-box large language models, 2025. URL
518 <https://arxiv.org/abs/2505.21082>.

519 Ishita Kumar, Snigdha Viswanathan, Sushrita Yerra, Alireza Salemi, Ryan A. Rossi, Franck Der-
520 noncourt, Hanieh Deilamsalehy, Xiang Chen, Ruiyi Zhang, Shubham Agarwal, Nedim Lipka,
521 Chien Van Nguyen, Thien Huu Nguyen, and Hamed Zamani. Longlamp: A benchmark for person-
522 alized long-form text generation, 2024. URL <https://arxiv.org/abs/2407.11016>.

523 Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian
524 Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, Yingying Zhang, Fei Yin, Jiahua Dong, Zhiwei Li,
525 Bao-Long Bi, Ling-Rui Mei, Junfeng Fang, Zhijiang Guo, Le Song, and Cheng-Lin Liu. From
526 system 1 to system 2: A survey of reasoning large language models. URL <http://arxiv.org/abs/2502.17419>.

527 Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization
528 Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.
529 URL <https://aclanthology.org/W04-1013/>.

530 Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: NLG
531 evaluation using gpt-4 with better human alignment. In Houda Bouamor, Juan Pino, and Kalika
532 Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language
533 Processing*, pp. 2511–2522, Singapore, December 2023. Association for Computational Linguistics.
534 doi: 10.18653/v1/2023.emnlp-main.153. URL [https://aclanthology.org/2023.emnlp-main.153/](https://aclanthology.org/2023.emnlp-main.153).

540 Sichun Luo, Guanzhi Deng, Jian Xu, Xiaojie Zhang, Hanxu Hou, and Linqi Song. Reasoning meets
 541 personalization: Unleashing the potential of large reasoning model for personalized generation,
 542 2025. URL <https://arxiv.org/abs/2505.17571>.

543 Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Si Zhang, Ren Chen, Christopher
 544 Leung, Jiajie Tang, and Jiebo Luo. Llm-rec: Personalized recommendation via prompting large
 545 language models, 2024. URL <https://arxiv.org/abs/2307.15780>.

546 Jianmo Ni and Julian McAuley. Personalized review generation by expanding phrases and attending
 547 on aspect-aware representations. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the*
 548 *56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*,
 549 pp. 706–711, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi:
 550 10.18653/v1/P18-2112. URL <https://aclanthology.org/P18-2112/>.

551 Xueming Qian, He Feng, Guoshuai Zhao, and Tao Mei. Personalized recommendation combining
 552 user interest and social circle. *IEEE Transactions on Knowledge and Data Engineering*, 26(7):
 553 1763–1777, 2014. doi: 10.1109/TKDE.2013.168.

554 Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks,
 555 2019. URL <https://arxiv.org/abs/1908.10084>.

556 Stephen E. Robertson, Steve Walker, Susan Jones, Micheline M. Hancock-Beaulieu, and Mike
 557 Gatford. Okapi at trec-3. In *Proceedings of the Third Text REtrieval Conference (TREC-3)*, pp.
 558 109–126, 1995.

559 Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. LaMP: When large
 560 language models meet personalization, 2023.

561 Alireza Salemi, Surya Kallumadi, and Hamed Zamani. Optimization methods for personalizing large
 562 language models through retrieval augmentation, 2024a.

563 Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. Lamp: When large lan-
 564 guage models meet personalization, 2024b. URL <https://arxiv.org/abs/2304.11406>.

565 Alireza Salemi, Cheng Li, Mingyang Zhang, Qiaozhu Mei, Weize Kong, Tao Chen, Zhuowan
 566 Li, Michael Bendersky, and Hamed Zamani. Reasoning-enhanced self-training for long-form
 567 personalized text generation, 2025. URL <https://arxiv.org/abs/2501.04167>.

568 Michael Shumanov and Lester Johnson. Making conversations with chatbots more personalized. *Com-
 569 puters in Human Behavior*, 117:106627, 2021. ISSN 0747-5632. doi: <https://doi.org/10.1016/j.chb.2020.106627>. URL <https://www.sciencedirect.com/science/article/pii/S0747563220303745>.

570 Alicia Tsai, Adam Kraft, Long Jin, Chenwei Cai, Anahita Hosseini, Taibai Xu, Zemin Zhang,
 571 Lichan Hong, Ed H. Chi, and Xinyang Yi. Leveraging LLM reasoning enhances personalized
 572 recommender systems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of*
 573 *the Association for Computational Linguistics: ACL 2024*, pp. 13176–13188, Bangkok, Thailand,
 574 August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.780.
 575 URL <https://aclanthology.org/2024.findings-acl.780/>.

576 Leyao Wang, Xutao Mao, Xuhui Zhan, Yuying Zhao, Bo Ni, Ryan A. Rossi, Nesreen K. Ahmed,
 577 and Tyler Derr. Towards bridging review sparsity in recommendation with textual edge graph
 578 representation, 2025. URL <https://arxiv.org/abs/2508.01128>.

579 Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdh-
 580 ery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models,
 581 2023. URL <https://arxiv.org/abs/2203.11171>.

582 Anna Wegmann, Marijn Schraagen, and Dong Nguyen. Same author or just same topic? to-
 583 wards content-independent style representations. In Spandana Gella, He He, Bodhisattwa Prasad
 584 Majumder, Burcu Can, Eleonora Giunchiglia, Samuel Cahyawijaya, Sewon Min, Maximilian
 585 Mozes, Xiang Lorraine Li, Isabelle Augenstein, Anna Rogers, Kyunghyun Cho, Edward Grefen-
 586 stette, Laura Rimell, and Chris Dyer (eds.), *Proceedings of the 7th Workshop on Representation*

594 *Learning for NLP*, pp. 249–268, Dublin, Ireland, May 2022. Association for Computational Lin-
 595 guistics. doi: 10.18653/v1/2022.repl4nlp-1.26. URL [https://aclanthology.org/2022.
 596 repl4nlp-1.26/](https://aclanthology.org/2022.repl4nlp-1.26/).

597 Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le,
 598 and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.
 599 URL <https://arxiv.org/abs/2201.11903>.

600 Fan Yang, Zheng Chen, Ziyuan Jiang, Eunah Cho, Xiaojiang Huang, and Yanbin Lu. Palr: Personaliza-
 601 tion aware llms for recommendation, 2023. URL <https://arxiv.org/abs/2305.07622>.

602 Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik
 603 Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023.
 604 URL <https://arxiv.org/abs/2305.10601>.

605 Hema Yoganarasimhan. Search personalization using machine learning. *Management Science*, 66(3):
 606 1045–1070, 2019. doi: 10.1287/mnsc.2018.3255.

607 Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks, 2018. URL
 608 <https://arxiv.org/abs/1802.09691>.

609 Zhehao Zhang, Ryan A. Rossi, Branislav Kveton, Yijia Shao, Diyi Yang, Hamed Zamani, Franck
 610 Dernoncourt, Joe Barrow, Tong Yu, Sungchul Kim, Ruiyi Zhang, Jiuxiang Gu, Tyler Derr, Hongjie
 611 Chen, Junda Wu, Xiang Chen, Zichao Wang, Subrata Mitra, Nedim Lipka, Nesreen Ahmed, and
 612 Yu Wang. Personalization of large language models: A survey, 2025. URL [https://arxiv.
 613 org/abs/2411.00027](https://arxiv.org/abs/2411.00027).

614 Tong Zhao, Yozhen Liu, Leonardo Neves, Oliver Woodford, Meng Jiang, and Neil Shah. Data
 615 augmentation for graph neural networks. *Proceedings of the AAAI Conference on Artificial
 616 Intelligence*, 35(12):11015–11023, May 2021a. doi: 10.1609/aaai.v35i12.17315. URL <https://ojs.aaai.org/index.php/AAAI/article/view/17315>.

617 Tong Zhao, Bo Ni, Wenhao Yu, Zhichun Guo, Neil Shah, and Meng Jiang. Action sequence
 618 augmentation for early graph-based anomaly detection. In *Proceedings of the 30th ACM Interna-
 619 tional Conference on Information & Knowledge Management*, CIKM ’21, pp. 2668–2678, New
 620 York, NY, USA, 2021b. Association for Computing Machinery. ISBN 9781450384469. doi:
 621 10.1145/3459637.3482313. URL <https://doi.org/10.1145/3459637.3482313>.

622
 623
 624
 625
 626
 627
 628
 629
 630
 631
 632
 633
 634
 635
 636
 637
 638
 639
 640
 641
 642
 643
 644
 645
 646
 647

648 A RELATED WORK
649650 **LLM Personalization.** Personalization in LLMs has recently garnered significant attention Salemi
651 et al. (2024b); Zhang et al. (2025) due to its potential to improve various downstream applications,
652 including search, recommendation, and conversational agents Yoganarasimhan (2019); Qian et al.
653 (2014); Shumanov & Johnson (2021). Salemi et al. (2023) introduced the LaMP benchmark, which
654 comprises seven datasets designed to evaluate personalization in language models by incorporating
655 personal context into downstream generation and classification tasks. Building on this, Salemi et al.
656 (2024a) explores optimization strategies for personalization by improving the retriever’s ability to
657 select relevant personal context. Furthermore, Kumar et al. (2024) extends personalization research
658 to the domain of long-form text generation. In addition, to address the issue of long-tail data sparsity
659 in the personal context histories, Au et al. (2025) proposes to augment the personal context with
660 user-centric graphs, retrieving relevant histories from other users.
661662 **LLM Reasoning and Planning.** Reasoning in LLM encourages the language model to think
663 and plan before generation, leading to more coherent and accurate outputs Li et al.. Chain-of-
664 Thought (CoT) prompting Wei et al. (2023) was first proposed to elicit reasoning capabilities of
665 language models by prompting the model to generate a series of intermediate steps that lead to a
666 final answer. Various methods have extended the CoT prompting to address its deficiencies. For
667 example, Self-Consistency Wang et al. (2023) samples multiple reasoning paths and selects the
668 most consistent answer, mitigating the impact of occasional reasoning errors. Additionally, Tree-
669 of-Thoughts(ToT) Yao et al. (2023) allow LLMs to explore multiple reasoning paths in a tree-like
670 structure, performing deliberate lookahead and backtracking to make more informed decisions.
671 Graph-of-Thoughts(GoT) Besta et al. (2024) further generalizes the ToT by modeling the reasoning
672 processes as arbitrary graphs.
673674 Recently, several works have explored reasoning in personalization. REST-PG Salemi et al. (2025)
675 employs self-training on LLM reasoning paths to improve the personalization. Luo et al. (2025)
676 explored reinforced reasoning for personalization by incorporating and refining a hierarchical rea-
677 soning thought template to guide the reasoning process. Additionally, Kim et al. (2025) explored
678 reasoning-level personalization that aligns model’s reasoning process with a user’s personalized logic.
679 Several works have also explored reasoning to enhance personalized recommendations Lyu et al.
680 (2024); Bismay et al. (2024); Yang et al. (2023) beyond traditional item-based recommendations.
681682 B LLM-AS-A-JUDGE
683684 Traditionally, in the prior personalization benchmarks (Au et al., 2025; Kumar et al., 2024; Salemi
685 et al., 2023), personalized text generation has been evaluated with lexical overlap metrics such as
686 ROUGE (Lin, 2004). However, it has been shown that such metrics may fail to capture the semantic
687 nuances and stylistic alignment in personalization. Thus, we adopt the LLM-as-a-Judge prompt from
688 prior works on personalized text generation Salemi et al. (2025), which is designed based on the
689 evaluation paradigm introduced in Liu et al. (2023). Our prompt is introduced as follows.
690691
692
693
694
695
696
697
698
699
700
701

702 **LLM-as-a-Judge**
 703
 704 Please compare the generated text to the reference text based on how well they match and/or
 705 are similar.
 706
 707 **Scoring Scale:**
 708 1 – Strongly disagree
 709 2 – Disagree
 710 3 – Somewhat disagree
 711 4 – Neither agree nor disagree
 712 5 – Somewhat agree
 713 6 – Agree
 714 7 – Strongly agree
 715
 716 **Content to Evaluate:**
 717 Reference Text (Ground Truth): {target_text}
 718 Generated Text: {generated_text}
 719
 720 Provide only the numeric score (1–7).
 721
 722

723 We use GPT-4 as the judge LLM, and report the normalized score (0.1–0.7) in our main experiment
 724 table. Salemi et al. (2025) designed additional experiments to validate the effectiveness of the
 725 LLM-as-a-Judge evaluation. First, they conduct a human evaluation comparing 100 model outputs
 726 and find that the LLM-as-a-Judge scores agree with human preference in 73% of cases, with a Pearson
 727 correlation of 0.46. Second, they design a controlled perturbation study by randomly replacing a
 728 portion of the personalized contexts with unrelated ones. The LLM-as-a-Judge scores decrease
 729 linearly as the perturbation rate increases, showing that the evaluator is sensitive to mismatched
 730 personalization.

731 While no automatic metric can fully replicate human evaluation for personalization—since the
 732 “true” judge of style and preference is the original user—LLM-as-a-Judge provides a scalable and
 733 semantically meaningful proxy. In our setting, it enables consistent evaluation across sparse and
 734 noisy contexts, capturing personalization quality beyond what lexical metrics can measure.

736 C EXPERIMENTAL SETUP

739 C.1 DATASET STATISTICS

741 Dataset	742 Train Size	743 Validation Size	744 Test Size
User-Product Review	20,000	2,500	2,500
Stylized Feedback	20,000	2,500	2,500
Hotel Experiences	9,000	2,500	2,500

745 Table 6: Dataset split sizes across training, validation, and test sets for the four domains.

748 We provide the dataset statistics in this section. In Table 6, we give the train/validation/test split
 749 statistics for the datasets. It is worth noting that the Hotel Experience dataset is a smaller dataset
 750 with a smaller training set, leading to the more inconsistent performance that we presented in the
 751 Experiment section. In Table 7, we introduce the task statistics for Long Text Generation, Short
 752 Text Generation, and Ordinal Classification. The datasets are constructed to reflect the real-world
 753 distribution (Au et al., 2025), which results in the sparse profiles as shown in the Average Profile Size.
 754 GRASPER achieves more consistent and significant performance gain in scenarios where the output
 755 length is shorter, such as Short Text Generation and the User-Product Review (Amazon dataset), as
 longer text implicitly gives more context for text generation.

756	Task	Type	Avg. Input Length	Avg. Output Length	Avg. Profile Size	# Classes
757	User-Product Review Generation	Long Text Generation	3.754 ± 2.71	47.90 ± 19.28	1.05 ± 0.31	-
758	Hotel Experiences Generation	Long Text Generation	4.29 ± 2.57	76.26 ± 22.39	1.14 ± 0.61	-
759	Stylized Feedback Generation	Long Text Generation	3.35 ± 2.02	51.80 ± 20.07	1.09 ± 0.47	-
760	User-Product Review Title Generation	Short Text Generation	30.34 ± 37.95	7.02 ± 1.14	1.05 ± 0.31	-
761	Hotel Experiences Summary Generation	Short Text Generation	90.40 ± 99.17	7.64 ± 0.92	1.14 ± 0.61	-
762	Stylized Feedback Title Generation	Short Text Generation	37.42 ± 38.17	7.16 ± 1.11	1.09 ± 0.47	-
763	User-Product Review Ratings	Ordinal Classification	34.10 ± 38.66	-	1.05 ± 0.31	5
764	Hotel Experiences Ratings	Ordinal Classification	94.69 ± 99.62	-	1.14 ± 0.61	5
765	Stylized Feedback Ratings	Ordinal Classification	40.77 ± 38.69	-	1.09 ± 0.47	5

Table 7: Data statistics for the PGraphRAG Benchmark across the four datasets. For each task, we report the average input and output lengths (in words), measured on the test set using BM25-based retrieval with GPT. The average profile size indicates the number of reviews per user used for personalization.

770 C.2 DATASETS

771 We evaluate our approach on three benchmark datasets introduced in prior research Au et al. (2025).
 772 These datasets cover diverse domains and graph structures, enabling us to assess the effectiveness of
 773 our method.

774 **Amazon Review.** The Amazon Review dataset is constructed from the Amazon Review 2023
 775 corpus (Hou et al., 2024). We build a user-item interaction graph where nodes represent users and
 776 products, and edges indicate review interactions between them.

777 **Hotel Experience.** The Hotel Experience dataset is collected from the Datafiniti Hotel Reviews
 778 dataset (Au et al., 2025). It contains user-hotel interaction data, where edges denote users’ stays at
 779 hotels and are annotated with textual reviews.

780 **Stylized Feedback Review.** The Stylized Feedback Review dataset is derived from the Datafiniti
 781 Grammar and Online Product dataset (Au et al., 2025). It focuses on generating stylistic and domain-
 782 specific feedback from user-product interactions. This dataset emphasizes linguistic diversity and
 783 style adaptation.

784 C.3 TASKS

785 Here we present an extended discussion on the tasks that we used to evaluate GRASPER: Long Text
 786 Generation, Short Text Generation, and Ordinal Classification.

787 **Long Text Generation.** The long text generation task focuses on producing detailed user reviews
 788 given a review title and the user’s profile. The objective is to generate coherent and contextually
 789 relevant review text that aligns with the user’s preferences. This task evaluates the model’s capability
 790 for generating high-quality, personalized text.

791 **Short Text Generation.** The short text generation task involves generating concise product titles or
 792 summaries given a user review. The challenge lies in distilling a longer text into a shorter title. This
 793 task assesses the model’s ability to distill information from highly personalized user context.

794 **Ordinal Classification.** The ordinal classification task aims to predict the rating score a user would
 795 assign to a product based on the title and review text. This task is particularly challenging because
 796 of varying rating behaviors; for example, some users might write critical reviews yet still assign
 797 high scores. This task is designed to evaluate the model’s ability to capture subtle patterns in user
 798 preferences and rating tendencies.

810 C.4 METRICS
811

812 **Text Generation.** For both long and short text generation tasks, we adopt widely used lexical
813 overlap metrics, including ROUGE-1 and ROUGE-L, following prior work Au et al. (2025). These
814 metrics capture n-gram and subsequence overlaps between the generated output and ground-truth
815 references. To complement these surface-level measures, we further incorporate LLM-as-a-Judge
816 evaluation, where a strong language model provides comparative assessments of personalization
817 and accuracy. We design the prompt based on prior studies which has been validated with human
818 evaluators on the task of personalization (Salemi et al., 2025). The prompt for LLM-as-a-Judge
819 evaluation is provided in Appendix B.

820 **Ordinal Classification.** For the ordinal classification task, we evaluate rating prediction using Root
821 Mean Squared Error (RMSE) and Mean Absolute Error (MAE). RMSE penalizes large deviations
822 more heavily, highlighting extreme mispredictions, while MAE measures the average magnitude of
823 prediction errors.

825 D PSEUDO CODE
826828 **Algorithm 1** GRASPER — Training

830 **Require:** Bipartite graph $G = (U \cup I, E)$; user histories $\{H_u\}$; item reviews $\{R_i\}$; encoder $\text{Enc}(\cdot)$;
831 base LLM M ; hyperparameters: K (items to augment), k_{sim} (similar users), k_{peer} (peer texts)
832 **Ensure:** Trained link predictor (GraphSAGE + MLP), fine-tuned LLM M'

833
834 1: // Step 1: Personal Context Expansion
835 2: **for** each node $v \in U \cup I$ **do**
836 3: **if** v is user u **then**
837 4: $h_v^{(0)} \leftarrow \text{Enc}(\text{concat}(H_u))$
838 5: **end if**
839 6: **if** v is item i **then**
840 7: $h_v^{(0)} \leftarrow \text{Enc}(R_i)$
841 8: **end if**
842 9: **end for**
843 10: **for** $\ell = 1$ to L **do** ▷ GraphSAGE layers
844 11: $m_v^{(\ell)} \leftarrow \text{AGG}_\ell(\{h_u^{(\ell-1)} : u \in \mathcal{N}(v)\})$
845 12: $h_v^{(\ell)} \leftarrow \text{ReLU}(W_\ell[h_v^{(\ell-1)} \| m_v^{(\ell)}])$
846 13: **end for**
847 14: $z_v \leftarrow h_v^{(L)}$ for all v
848 15: Score edges with $s(u, i) = \text{MLP}([z_u \| z_i])$, $\hat{y}(u, i) = \sigma(s(u, i))$
849 16: Optimize BCE with negative sampling to train (GraphSAGE+MLP) \rightarrow Link Predictor
850
851 17: // Step 2: Synthetic Review Generation with Reasoning Alignment
852 18: **for** each training user u **do**
853 19: $\mathcal{S}_u \leftarrow \text{TopK}_{k_{\text{sim}}}(\cos(z_u, z_\cdot))$
854 20: $H_{\mathcal{S}_u} \leftarrow \{\text{reviews from users in } \mathcal{S}_u\}$
855 21: **for** each observed pair (u, j) **do**
856 22: $P_{u,j} \leftarrow \text{BM25_TopK}_{k_{\text{peer}}}(\text{reviews of } j)$
857 23: $x \leftarrow \{H_u \setminus \{t_{u,j}\}, H_{\mathcal{S}_u}, P_{u,j}\}$
858 24: Sample candidate reasoning paths $\{Z^{(1)}, \dots, Z^{(K)}\} \sim M$ with prompt $\phi(x)$
859 25: $Z^* \leftarrow \arg \max_Z \Omega(M(\xi(x, Z)), t_{u,j})$ ▷ Ω : dev metric (e.g., ROUGE/METEOR)
860 26: Update M by minimizing $\text{CrossEntropy}(M(\rho(x)), [Z^* \| t_{u,j}])$
861 27: **end for**
862 28: **end for**
863 29: $M' \leftarrow M$
30: **return** (Link Predictor), M'

Table 8: Backbone ablation on Amazon Reviews dataset for GRASPER using open-source (Llama 3, Gemma 2) and proprietary (GPT-4o mini, GPT-4.1) backbones. Metrics for text/title generation are higher-is-better; for rating prediction, lower-is-better.

Category	Backbone	Method	Text Generation				Title Generation				Rating Prediction	
			R-1	R-L	MET	LJ	R-1	R-L	MET	LJ	RMSE	MAE
Open-Source	Llama 3	GRASPER	0.215	0.171	0.178	0.337	0.155	0.153	0.142	0.304	0.32	0.31
		PGraph	0.178	0.129	0.151	0.297	0.178	0.129	0.151	0.241	0.76	0.34
	Gemma 2	GRASPER	0.160	0.119	0.121	0.326	0.127	0.123	0.122	0.329	0.52	0.53
		PGraph	0.155	0.119	0.117	0.316	0.098	0.093	0.124	0.297	0.88	0.42
Proprietary	GPT-4o mini	GRASPER	0.219	0.170	0.182	0.421	0.178	0.174	0.162	0.406	0.33	0.34
		PGraph	0.189	0.130	0.196	0.389	0.115	0.112	0.099	0.353	0.38	0.73
	GPT-4.1	GRASPER	0.221	0.176	0.181	0.433	0.151	0.150	0.166	0.401	0.31	0.32
		PGraph	0.185	0.128	0.191	0.403	0.107	0.103	0.122	0.346	0.38	0.70

E ADDITIONAL EXPERIMENT RESULTS

E.1 LANGUAGE MODEL VARIANTS

In Table 8, we compare GRASPER across different backbone models, covering both open-source (Llama 3, Gemma 2) and proprietary (GPT-4o mini, GPT-4.1) variants. This setup allows us to test whether the improvements of GRASPER depend on a particular language model family or extend across architectures with varying sizes and training pipelines.

For open-source models, GRASPER consistently improves over the baseline PGraph across all metrics. With Llama 3, GRASPER achieves a clear gain in text generation and rating prediction. Gemma-2, though smaller in scale, still benefits from our framework, showing improved semantic quality on LLM-as-a-Judge metric. These results suggest that GRASPER effectively enhances smaller open-source models, making them more competitive for personalization tasks.

When applied to proprietary models, the improvements remain consistent. On GPT-4.0 mini, GRASPER outperforms the baseline in text generation and especially in LLM-as-a-Judge, demonstrating better alignment with human preferences. GPT-4.1 mini, the more advanced backbone, also benefits: GRASPER achieves the highest score across metrics, indicating strong personalization quality even when starting from a more powerful model.

918
919
920 Table 9: Link Prediction Performance across Different Datasets
921
922
923
924
925
926

Dataset	MRR	Hits@1	Hits@5	Hits@10
Amazon	0.531	0.415	0.659	0.760
Hotel	0.324	0.210	0.446	0.546
Feedbacks	0.275	0.178	0.394	0.477

927 Overall, the results confirm that GRASPER is robust to the choice of language model backbone.
928 Gains are observed consistently across both open-source and proprietary families. Importantly,
929 improvements in LLM-as-a-Judge are more significant, underscoring that our framework aligns better
930 with human preference. This robustness highlights GRASPER’s practicality, as it can be flexibly
931 deployed in various settings with different backbone models.

932 E.2 LINK PREDICTION NOISE AND ROBUSTNESS

933 Although GRASPER achieves strong improvements on personalized text generation—particularly
934 in sparse-user settings—the link prediction module inevitably introduces a degree of noise due to
935 imperfect edge predictions. It is therefore important to assess both the quality of the predicted
936 user-item links and the robustness of the downstream reasoning-based personalization to such noise.

937 Table 9 reports the standalone performance of the link prediction module across all datasets. The
938 module demonstrates consistently strong ranking metrics, indicating its ability to recover meaningful
939 user-item affinities even under sparse supervision. Nonetheless, some level of incorrect or low-
940 confidence predictions is unavoidable. To study whether such noise impacts the final generation
941 quality, we further conduct an analysis on the Amazon test set by partitioning examples into two
942 groups: the top 50% and bottom 50% based on their link-prediction confidence scores.

943
944 Table 10: Comparison of Text Generation Scores by Link Prediction Performance
945

Group	ROUGE-L Mean	METEOR Mean
Bottom 50% link pred scores	0.587982	0.555729
Top 50% link pred scores	0.605974	0.569973

946 As shown in Table 10, the generation quality of the low-confidence group remains comparable
947 to that of the high-confidence group across ROUGE and METEOR metrics. This suggests that
948 even when some retrieved neighbors originate from noisy edges, the reasoning module is able to
949 filter, contextualize, and extract stylistically relevant information from the neighborhood. Overall,
950 these results indicate that GRASPER is robust to moderate imperfections in link prediction and can
951 effectively leverage the noisy-but-useful relational signals present in sparse user-item graphs.

952 E.3 PERSONALIZATION SPARSITY ROBUSTNESS

953 To further examine how GRASPER behaves under different levels of personalization sparsity, we
954 partition users in the test split by the number of real historical reviews available: users with 0 reviews
955 (cold-start), 1 review, and 2+ reviews. This allows us to isolate how much GRASPER depends
956 on explicit user history versus the contextual and relational reasoning signals introduced by our
957 framework. As shown in Table 11, GRASPER demonstrates strong robustness across all sparsity
958 levels and consistently outperforms the PGraph baseline. Notably, our model achieves meaningful
959 improvements even in the cold-start setting. This behavior arises because, even when a user has no
960 prior reviews, GRASPER can still leverage contextual cues provided at inference time, including
961 the review title, partial user-written text, or product description, to retrieve relevant neighbors and
962 construct a personalized reasoning path. By contrast, prior personalization methods such as PGraph
963 depend primarily on embedding-based retrieval, which is significantly less effective when a user
964 lacks a profile or has only one review.

Sparsity Level	GRASPER ROUGE-L	PGraph ROUGE-L	GRASPER ROUGE-1	PGraph ROUGE-1
0 historical reviews (cold-start)	0.160	0.125	0.204	0.183
1 historical review	0.170	0.137	0.225	0.212
2+ historical reviews	0.186	0.149	0.296	0.262

Table 11: Performance under different sparsity levels of user history.

E.4 UTILITY OF REASONING PATH SELECTION

In this section, we further explore the utility of the reasoning path selection as introduced in Eq. 7. We analyze the ranked reasoning paths produced by the scoring metric Ω . The distribution of candidate scores, shown in Table 12, reveals clear separation among candidate paths, indicating that their quality varies and that the proposed selection mechanism is necessary for GRASPER to identify the most coherent and stylistically aligned reasoning trace.

Rank	1 (lowest)	2	3	4	5 (highest)
Score	0.3943	0.4465	0.4709	0.4964	0.5137

Table 12: Score distribution of ranked reasoning paths produced by Ω (Eq. 8).

These results indicate that while reasoning traces cannot be directly evaluated in isolation, the model benefits substantially from the ranked reasoning guidance.

F THEORETICAL ANALYSIS OF THE BIAS-VARIANCE TRADE-OFF IN GRASPER

As detailed in Section 4.4.2, the hyperparameter K determines the number of predicted items. In Table 5, we demonstrate that GRASPER, with the reasoning alignment, can more reliably utilize the additional retrieved context compared to other baselines with retrieval. We hypothesize the behavior corresponds to the bias-variance trade-off theory, where the reasoning serves as a regularization trick that can offset the trade-off and allow the variance reduction without bias increase. Note that in the below we use k instead of K to represent the number of predicted/synthetic.

Proposition 1 (Bias–Variance trade-off). *Let $\theta_u \in \mathbb{R}^d$ denote the user’s latent style vector. We observe n real samples $x_j = \theta_u + \varepsilon_j$ with $\mathbb{E}[\varepsilon_j] = 0$, $\text{Var}(\varepsilon_j) = \sigma^2 I$, and k synthetic samples $\tilde{x}_\ell = \theta_u + \Delta + \tilde{\varepsilon}_\ell$ with $\mathbb{E}[\tilde{\varepsilon}_\ell] = 0$, $\text{Var}(\tilde{\varepsilon}_\ell) = \tilde{\sigma}^2 I$, where $\Delta \in \mathbb{R}^d$ is a fixed (unknown) bias. Consider the pooled estimator*

$$\hat{\theta}_u = \frac{1}{n+k} \left(\sum_{j=1}^n x_j + \sum_{\ell=1}^k \tilde{x}_\ell \right).$$

Then the (per-coordinate) mean squared error is

$$\text{MSE}(k) = \underbrace{\frac{n\sigma^2 + k\tilde{\sigma}^2}{(n+k)^2}}_{\text{variance}} + \underbrace{\left(\frac{k}{n+k}\right)^2 \|\Delta\|^2}_{\text{bias}^2/d \text{ (per-dim)}}$$

In the equal-noise case $\tilde{\sigma}^2 = \sigma^2$, this simplifies to

$$\text{MSE}(k) = \frac{\sigma^2}{n+k} + \left(\frac{k}{n+k}\right)^2 \|\Delta\|^2.$$

Sketch. $\mathbb{E}[\hat{\theta}_u] = \theta_u + \frac{k}{n+k} \Delta$, so the squared bias per dimension is $\left(\frac{k}{n+k}\right)^2 \|\Delta\|^2$ (treating σ^2 as per-dimension noise). Since samples are independent with isotropic noise, $\text{Var}(\hat{\theta}_u) = \frac{n\sigma^2 + k\tilde{\sigma}^2}{(n+k)^2} I$. Add variance and bias² to obtain the expression. For $\tilde{\sigma}^2 = \sigma^2$, write $\text{MSE}(t) = \frac{\sigma^2}{n} (1-t) + \|\Delta\|^2 t^2$, differentiate w.r.t. t , set to zero, and solve. ■

1026
1027 *Remark 1 (Effect of Reasoning Alignment).* If reasoning alignment attenuates the preference mis-
1028 match to $\Delta_{\text{RA}} = \beta\Delta$ with $\beta \in (0, 1)$, then

1029
$$\text{MSE}_{\text{RA}}(k) = \frac{\sigma^2}{n+k} + \left(\frac{k}{n+k}\right)^2 \beta^2 \|\Delta\|^2,$$

1030

1031 so the minimum achievable error decreases and the optimal fraction $t_{\text{RA}}^* = \frac{\sigma^2}{2n\beta^2\|\Delta\|^2}$ increases, i.e.,
1032 *alignment lets you safely use larger k.*

1033
1034 *Remark 2 (Sparse Users Benefit More).* t^* scales as $1/n$: when n is small (sparse users), the variance
1035 term dominates and the optimal augmentation fraction is larger. Thus augmentation disproportionately
1036 helps sparse users by reducing variance, while reasoning alignment curbs the bias induced by Δ .

1037 G GRASPER PROMPTS

1039
1040 In this section, we supply the prompts we used in GRASPER. ϕ is used in Equation (7) where the
1041 prompt is used to elicit candidate reasoning paths. ξ is used in Equation (8), where the prompt is used
1042 to generate the final answer given the input and reasoning to evaluate the candidate paths. Lastly, ρ is
1043 used in Equation (9) where it structures the final input for personalized text generation.

1044 Reasoning Paths Generation (ϕ)

1045
1046 System: You are a personalized review generation assistant that generates high-quality
1047 reviews based on user history and context.

1048
1049 Given profile which contains past documents written by the same person (might be
1050 empty), documents written by users that have similar writing style, reviews on the target
1051 product, and reasoning.

1052 User's own profile: {history_reviews_str}
1053 Similar profiles: {neighbor_reviews_str}
1054 Product Reviews: {product_reviews_str})

1055
1056 Based on the above information, provide a detailed reasoning path that explains how
1057 we can arrive at the expected output. Consider:

1058 1. User's Writing Style: Analyze their typical review length, tone, and language patterns.
1059 2. User's Preferences: What aspects of products do they typically focus on or value?
1060 3. Product Information: What are the commonly mentioned features, pros, and cons from
1061 other reviews?

1062 Do not limit the reasoning to the above points. You can use your own knowledge to reason
1063 about the user's review. It is important to make sure that you only talk about information
1064 from the profile while considering the expected output in the reasoning process. You cannot
1065 directly copy or mention anything about the expected output. The expected output is only
1066 used to determine the reasoning process and how profile can affect the expected output.

1067
1068 Provide your reasoning that leads to the following expected review on the target
1069 product from the user:

1070 Expected Output:
1071 Title: "target_review['title']"
1072 Text: "target_review['text']"
1073 Rating: target_review['rating']

1074 As mentioned before, you cannot directly copy or mention anything about the ex-
1075 pected output. The expected output is only used to determine the reasoning process. Do not
1076 mention the expected output in your reasoning. Your reasoning should only analyze the
1077 profile and the other reviews.

1078
1079 Output your reasoning in a single paragraph. Do not output anything else.

1080

1081

1082

1083

1084

1085

1086

1087

1088

1089

1090

1091

1092

1093

1094

1095

1096

1097

1098

1099

1100

1101

1102

1103

1104

1105

1106

1107

1108

1109

1110

1111

1112

1113

1114

1115

1116

1117

1118

1119

1120

1121

1122

1123

1124

1125

1126

1127

1128

1129

1130

1131

1132

1133

Your reasoning:

1134
1135Reasoning Paths Evaluation (ξ)

1136

1137

1138

1139

1140

1141

1142

1143

1144

1145

1146

1147

1148

1149

1150

1151

1152

1153

1154

1155

1156

1157

1158

1159

1160

1161

1162

1163

1164

1165

1166

1167

System: You are a personalized review evaluation assistant that judges whether the generated reasoning and review are consistent with the user's style and product context.

Given a profile containing past documents written by the same person (may be empty), documents from users with similar writing style, reviews on the target product, and a reasoning trace, you will evaluate and refine the review text.

User's own profile: {history_reviews_str}

Similar profiles: {neighbor_reviews_str}

Product Reviews: {product_reviews_str}

Reasoning: {reasoning_str}

Based on the above information, evaluate how well the provided review text follows the reasoning and user profile. Consider:

1. Faithfulness to the reasoning: Does the review follow the logical path outlined in the reasoning?
2. Stylistic alignment: Does the review reflect the user's writing style and preferences?
3. Product grounding: Is the review consistent with the product reviews and features mentioned?

Do not copy directly from the reasoning or profiles. Your task is to provide a short evaluation and, if needed, produce a refined review text.

Provide your output strictly in the format:

Evaluation: <evaluation>. Review text: <Review text>

Do not output anything else.

Review text: {review_text}

Text Generation (ρ)

System: You are a personalized review generation assistant that generates high-quality reviews based on user history and context.

Given a profile containing past documents written by the same person (may be empty), documents written by users with similar writing style, and reviews on the target product.

User's own profile: {history_reviews_str}

Similar profiles: {neighbor_reviews_str}

Product Reviews: {product_reviews_str}

Reason and generate a review title/review text based on the following review text/review title. Use the format:

Reasoning: <reasoning>. Review title/text: <Review title/text>.

Do not output anything else.

Review text/Review title: {review_text} / {review_title}

1188

H NOTATIONS

1189
 1190 To facilitate readability, we summarize the main mathematical symbols and notations used throughout
 1191 the paper in Table 13, which serves as a quick reference to clarify definitions of variables, functions,
 1192 and operators appearing in the main text.
 1193

1194 Table 13: Summary of key notations used throughout the paper.
 1195

Symbol	Type / Shape	Description
$G = (\mathcal{U} \cup \mathcal{I}, \mathcal{E})$	graph	Graph with user set \mathcal{U} , items \mathcal{I} , edges \mathcal{E}
H_u	set of texts	Observed history (reviews) of user u
R_i	set of texts	Reviews associated with item i
$\text{Enc}(\cdot)$	text $\rightarrow \mathbb{R}^d$	Text encoder (e.g. SentenceTransformers)
$h_v^{(\ell)}$	\mathbb{R}^d	Node representation at layer ℓ (GraphSAGE)
$z_v = h_v^{(L)}$	\mathbb{R}^d	Final node embedding for node v
$\mathcal{N}(v)$	set of nodes	Neighborhood of node v
$\text{AGG}_\ell(\cdot)$	operator	Neighborhood aggregator at layer ℓ
\mathcal{M}	LLM	Base language model
\mathcal{M}'	LLM	Fine-tuned LLM used for inference
$s_{u,i}$	\mathbb{R}	Link score from decoder for user u and item i
K	integer	# predicted items to augment per user
k_{sim}	integer	# similar users retrieved for u
k_{peer}	integer	# peer texts (BM25) per item
\mathcal{S}_u	set of users	Top- k_{sim} similar users to u
$P_{u,i}$	set of texts	Top- k_{peer} peer reviews for item i
\mathcal{I}_u^K	set of items	Top- K predicted items for user u by $s(u, i)$
$t_{u,i}$	text	Ground-truth review by u for item i (observed)
$\tilde{t}_{u,i}$	text	Synthetic review for (u, i) during expansion
$\hat{t}_{u,i*}$	text	Final predicted personalized review for user u on target item i^*
\mathcal{Z}	text	A reasoning path
$\{Z^{(k)}\}_{k=1}^K$	list of texts	K candidate reasoning paths
$\phi(x, t_{u,j})$	prompt	Prompt to elicit candidate reasoning paths given input and expected output
$\xi(x, \mathcal{Z})$	prompt	Prompt that conditions generation on input x and rationale \mathcal{Z} to evaluate \mathcal{Z}
$\rho(x)$	prompt	Prompt that instructs model to output the reasoning path and the review
$\Omega(\cdot, \cdot)$	metric	Evaluation metric for reasoning paths (e.g., ROUGE/METEOR)

1221
 1222
 1223
 1224
 1225
 1226
 1227
 1228
 1229
 1230
 1231
 1232
 1233
 1234
 1235
 1236
 1237
 1238
 1239
 1240
 1241