Personalized Text Generation Using a Mixture of LLM Experts

Shenova Davis (ssd2184)
Shachi Hardi (sh4615)
Ashley Noh (cn2671)
Jeongyong Yang (jy3464)

Abstract

Personalized text generation is often treated as a single-agent problem, even though personalization spans multiple competing objectives—capturing stylistic voice, selecting content aspects, reflecting preferences, and maintaining coherent long-form structure. A single LLM struggles to optimize all of these dimensions simultaneously. We introduce Mixture-of-Personalized-Agents (MoPA), a multi-agent framework that decomposes personalization into specialized proposers guided by a personalized aspect planner, and a final aggregator to generate a final coherent output that better matches a user's historical writing patterns. Across three personalization benchmarks (product reviews, research abstracts, Reddit posts) and six settings (Top vs cold-start users), all MoPA variants outperform single-agent, RAG-only, and planning-only baselines on standard generation metrics. We find that structured planning delivers the strongest improvements, and that diversity among agents, which stems through planning, retrieval, or model variation, further enhances personalization quality. Scaling analyses show that performance increases up to a moderate number of agents before degrading, indicating that planning, diversity, and controlled scale are jointly critical for multi-agent personalization. These findings highlight that structured agent collaboration and targeted diversity drive meaningful gains in personalized long-form generation.

1 Introduction

Personalized text generation is increasingly important as large language models (LLMs) are deployed in real-world applications such as content creation, email drafting, and document writing. In these settings, users expect generated text to reflect their individual writing styles, preferences, and recurring themes (Salemi et al., 2024; Li et al., 2024b). Despite advances in language modeling, most LLMs are trained on generic data and struggle to adapt their outputs to user-specific contexts, especially in long-form generation scenarios (Salemi et al., 2024; Kumar et al., 2024).

Existing personalization approaches primarily fall into two frameworks: personalized Retrieval-Augmented Generation (RAG) and plan-based personalization (PlanPers). Personalized RAG retrieves user-specific documents or examples and appends them to the model's input prompt at inference time, offering scalability and avoiding retraining (Salemi et al., 2024). However, this framework often treats retrieved content as unstructured context, limiting the model's ability to reason deeply over user preferences and leading to inconsistent personalization across long outputs (Kumar et al., 2024).

Plan-based personalization methods address some of these limitations by introducing an explicit planning stage that summarizes user-specific stylistic tendencies, writing preferences, and thematic patterns before generation (Salemi et al., 2025). While this structured guidance improves personalization, prior work suggests that maintaining consistent and rich personalization across long-form generation remains challenging (Salemi et al., 2025; Kumar et al., 2024).

These challenges are especially pronounced for long-form tasks, where models must maintain coherent personalization across extended outputs. As input context grows, models frequently degrade in performance due to ineffective prioritization of relevant user information (Kumar et al., 2024). Although prior work has focused largely on short-text personalization, most practical applications require long-form outputs. The LongLaMP benchmark was introduced to evaluate this setting, yet developing models that maintain high-quality personalization throughout long documents re-

mains an open problem (Kumar et al., 2024).

To address these limitations, we propose a framework that integrates personalized RAG and plan-based personalization within a multi-layer Mixture-of-Personalized-Agents (MoPA) architecture, inspired by recent mixture-of-agents approaches for collaborative LLM reasoning (Chen et al., 2024; Wang et al., 2024; Li et al., 2024a), and combining structured personalization planning with diversified generation and aggregation.

Contributions. Our work makes the following contributions:

- **A new multi-agent architecture for personalization.** We propose *Mixture-of-Personalized-Agents (MoPA)*, a collaborative multi-agent framework that decomposes personalization into specialized proposers guided by an aspect planner, with a final aggregator that fuses diverse drafts into a coherent output.
- **A principled integration of planning and personalized retrieval.** MoPA combines structured aspect planning and profile-aware retrieval to guide generation toward user-specific content, enabling complementary stylistic and thematic exploration rather than collapsing to a single template.
- **Evaluation against State-of-the-art personalization methods for long-form generation.** Across product reviews, research abstracts, and Reddit posts—and across both top-history and cold-start users, MoPA consistently outperforms SOTA single-agent baselines, personalized RAG, and Planpers.
- **Ablation studies on planning, diversity, and scale.** We show that structured planning delivers the strongest improvements, diversity across agents further enhances personalization quality, and performance increases up to a moderate number of agents before degrading—indicating that controlled scale and targeted collaboration, not model size alone, drive personalization gains.

2 Problem Description

We consider the problem of personalized long-form text generation. Given a user prompt and a set of the user's past writings, the task is to generate a long-form document that reflects the user's writing style and preferences, without performing user-specific fine-tuning. Because fine-tuning on user data is costly, difficult to scale, and raises privacy concerns, prior work typically incorporates user history at inference time, for example through Retrieval-Augmented Generation (RAG) or plan-based personalization. However, existing approaches often apply personalization signals inconsistently across long outputs, resulting in degraded stylistic coherence as text length increases. In practice, models must determine which parts of a user's history are relevant and apply them consistently across long documents.

To address these challenges, we proposed an approach that integrates plan-based personalization (PlanPers) (Salemi and Zamani, 2025) with a Mixture-of-Personalized-Agents (MoPA) architecture (Wang et al., 2024). PlanPers introduces an explicit planning stage that summarizes user-specific characteristics—such as stylistic tendencies, writing preferences, and recurring themes—from prior user writings. This plan provides structured guidance for generation by indicating which aspects of personalization should be emphasized.

Building on this planning stage, we adopt a multi-layer MoPA framework in which multiple LLM agents independently generate personalized candidates, and a higher-level model aggregates these outputs into a single response. This design allows personalization signals to be applied more consistently across long-form generation than retrieval or plan-based approaches alone. Through this layered and diversified generation process, we aim to demonstrate that multi-layer MopA with diverse plan-based personalization methods produces superior personalized text compared to both standard RAG and Planpers approaches.

3 Data Description

We use the **LongLaMP** benchmark (Kumar et al., 2024), which is designed to evaluate personalized long-form text generation with large language models. LongLaMP consists of four tasks: Personalized Email Completion, Abstract Generation, Product Review Generation, and Topic Writing. Each task is provided in two settings: *User*, where the model personalizes for unseen users, and *Temporal*, where the model must adapt to an individ-

ual's evolving writing style over time.

Each task is divided into train, validation, and test splits, with no overlap in users or time periods across settings. Data instances are stored in JSONL format and include a user identifier, a task prompt, the ground-truth text written by the original author, and a user profile composed of prior writings. These profiles serve as the personalization context for retrieval-based methods.

In this work, we focus on the *temporal* setting of three tasks: Product Review Generation, Topic Writing, and Abstract Generation. Product Review Generation involves writing reviews for consumer products on a five-star scale. Topic Writing consists of Reddit-style responses reflecting individual opinions and writing styles. Abstract Generation requires producing academic abstracts conditioned on scientific papers.

For all experiments, user profiles are constructed from historical texts written by the same author and retrieved using a consistent BM25-based procedure. Using identical prompts and retrieved profiles across all configurations enables a controlled comparison between single-model approaches and Mixture-of-Personalized-Agents (MoPA) architectures.

4 Experiments

Experimental Setup

In this study, we investigated two distinct personalization methods: RAG Personalization and Plan-RAG Personalization (PlanPers). RAG-Personalization focuses on generating an output by adapting the writing style and preferences of the author solely based on historical writings selected through retrieval methods. The PlanPers setting incorporates an additional planning layer prior to output generation. Given the original prompt and the top K retrieved writings from the user profile, the planner model generates guidance on the aspects the generated text should emphasize.

For each personalization type, we implemented and compared two configurations: a MoPA and an individual LLM. The MoPA architecture, as seen in Figure 1, comprises four models that independently generate candidate responses, which are then combined into a single final output by passing them to the aggregator model along with the original prompt and a secondary prompt. This multi-agent generation is applied iteratively across five layers, with each layer refining the outputs of the previous one. This setup allows us to combine diverse responses from multiple models and synthesize the most relevant personalized elements into a single output.

By evaluating both personalization methods under the two architectural settings, our objective is to determine whether MoPA improves the quality of personalized responses compared to single models.

All experiments were conducted using 1000 data points from the temporal subset of the personalized product review generation, topic writing, and abstract generation in the LongLaMP benchmark. For all setups, we employed RAG using the BM25 retriever to select the top K = 4 relevant profile entries for each prompt. The plan step also utilized the identical BM25 retrieval procedure.

4.1 Standard RAG-Personalization

In the RAG-Personalization setup, the GPT 4o model was used to generate personalized responses by conditioning on both the current prompt and a user's previous writing history. For each individual, the top four most relevant historical texts were retrieved from their profile using the BM25 algorithm, which identifies and ranks prior writings based on their textual similarity to the current description. These retrieved examples were then provided as additional context for the model, allowing it to align its generation with the writer's established tone, phrasing, and stylistic preferences. This baseline setup demonstrates the ability of a single LLM to adapt an individual's writing syntax and patterns over time using only past context.

4.2 Standard Planpers-Personalization

The PlanPers configuration builds on a RAG framework by adding a planning component that determines what information should be included before generation. Instead of directly producing text from the retrieved examples, the model first creates a plan that highlights the key aspects, themes, and stylistic elements relevant to the response. It allows the system to use the user's past history to infer which aspects they are most likely to include in their new text.

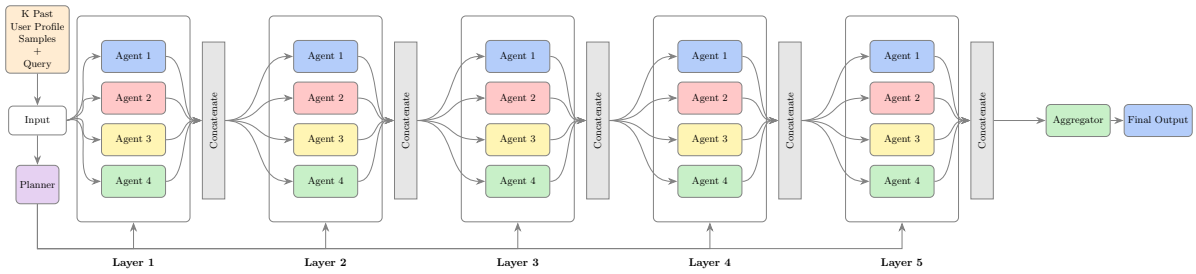

Figure 1: Overview of the PlanPers Mixture-of-Personalized-Agents architecture.

In our setup, we use the GPT 4o model to combine the plan text, the retrieved profile texts that reflect the user's previous writing style, and the original prompt. This integration allows the model to generate responses that are consistent with the user's tone and preferences while maintaining focus on the most important content.

Overall, PlanPers supports both content-level and style-level personalization. It guides the model to write in a way that aligns with the user's typical style and emphasizes the aspects that are most important, resulting in outputs that are both coherent and personally tailored.

4.3 MoPA Experiments

The MoPA PlanPers configuration integrates the planning component of the standard PlanPers framework with the MoPA architecture. An initial planning stage summarizes stylistic and semantic features from the historical writings of the user. Following this stage, four independent LLM agents generate candidate responses. Each agent generates text using both the retrieved user profile and the shared personalization plan, allowing it to draw on the user's past writing while focusing on the aspects most relevant to the current task.

This architecture allows multiple agents to interpret and implement the same plan in distinct ways, encouraging diversity in generation while maintaining consistency with the established style of the user. Additionally, we experimented utilizing different setups to test various methods of Planpers MoPA.

4.3.1 RAG MoPA

To establish a baseline for multi-agent personalization for RAG MoPA, we first evaluated a homogeneous MoA architecture in which all proposer agents are GPT-4o. Each proposer receives identical inputs, which are the user profile and the query from the LongLaMP benchmark, to generate the candidate responses independently. A GPT-4o aggregator then synthesizes these candidates into a final response. We systematically varied layer depth and agent count per layer to separately measure the contributions of layer depth and agent count to personalization quality. This design allows us to determine whether scaling homogeneous agents alone improves output quality, providing a reference point for the cross-model experiments.

4.3.2 Planner MoPA

To establish a baseline for multi-agent personalization for planner MoPA, we then evaluated a homogeneous MoPA architecture in which all proposer agents use the same underlying LLM, GPT-4o. On top of the user profile and query like the RAG MoPA above, each proposer is also introduced with the planned output from PlanPers. The rest of the setting remains the same as the RAG MoPA architecture.

4.3.3 RAG MoPA-MD

To examine the effect of cross-model diversity (MD) in RAG MoPA, we constructed an architecture in which the proposer agents consist of a mixture of different LLMs, while the aggregator remains fixed with GPT-4o. Specifically, two Chat-5-GPT and DeepSeek agents were used as proposers, each conditioning on the same user profile and the original query from LongLaMP dataset. The aggregator synthesized these heterogeneous candidate generations into a final response. This experiment evaluated whether diversity across models in proposers improves personalization quality beyond the homogeneous MoPA setup.

4.3.4 Planner MoPA-MD

To examine the effect of cross-model diversity in planner MoPA, we constructed an architecture similar to the RAG MoPA-MD in which the proposer agents consist of a mixture of different LLMs, while the aggregator remains fixed with GPT-4o. The setting of two Chat-5-GPT and DeepSeek agents as proposals is identical to the RAG MoPA-MD, but each model was also introduced with the planned output. The rest of the architecture remains the same as the RAG MoPA-MD structure.

4.3.5 MoPA-AD

Additionally, we propose Cross-Agent diversity (AD) MoPA, a hybrid MoPA configuration that combines RAG-Personalization and PlanPers agents within the same layer. Each MoPA layer contains four proposer agents: two RAG-Personalization agents and two PlanPers agents. All agents receive the same query and top-K user profile entries retrieved using BM25; however, only the PlanPers agents are additionally conditioned on a personalization plan, while the RAG agents generate responses directly from the retrieved context. All proposer agents, the planner, and the aggregator are implemented using GPT-4o. This design introduces heterogeneity in personalization by combining retrieval-based and plan-guided agents. The candidate responses are aggregated using a GPT-4o model to form the final output. This experiment examines whether such hybrid agent configurations outperform homogeneous MoPA architectures.

4.3.6 Multi-Planner MoPA

To evaluate the effect of planner diversity within a single model setting, we constructed a multi-planner MoPA architecture in which each proposer agent independently generates its own high-level plan using GPT-4o. All agents use GPT-4o and are conditioned on the same user profile, query, and task specification from the LongLaMP dataset, but produce plans and candidate responses independently. The aggregator model, also with GPT-4o, then synthesizes these candidate generations into a final response. This experiment assesses whether implicit plan diversity arising from independent agent planning improves personalization quality compared to MoPA architectures with a shared or fixed planning process.

4.3.7 Role-Planner MoPA

To examine the effect of functional role specialization, we constructed a role-based MoPA architecture in which proposer agents are assigned distinct roles during generation, while the planner, all agents, and the aggregator consistently use GPT-4o. Each agent is conditioned on the same user profile, query, and task specification from the LongLaMP dataset, but is prompted to focus on a specific aspect of the response. In particular, agents are designed to emphasize complementary objectives such as content coverage, analytical reasoning, practical usefulness, and stylistic or tonal alignment with the user profile. The aggregator then fuses these role-specific candidate generations into a single final output. This experiment evaluates whether explicit role differentiation among agents leads to improved personalization and response quality compared to role-agnostic MoPA setups.

4.4 Prompt Templates

The prompt templates used in the individual LLM and MoPA configurations are shown in the figures below. These prompt examples are specific for the product review temporal dataset. They include:

1. **Proposal Generation Prompt:** The prompt passed to the individual LLMs. It takes the query provided by the LongLaMP benchmark for each specific reviewer, retrieved context, and plan aspects (if necessary) to output a final result for evaluation or to be passed to the MoPA aggregation prompt. The PlanPers model takes an additional plan block to generate the response. See Figure 1 below.

Figure 1: Proposal Prompt

You are a helpful assistant designed to generate a personalized product review that adapts to the author's evolving writing style over time. You are given several of the user's past reviews, ordered from most recent to oldest.

Use the most recent reviews at the top as the strongest signal of the author's tone, phrasing, and focus.

Past Reviews (most recent to oldest):

[profile_block]

High-level plan describing which aspects to focus on in the new review (if applicable): [plan_block]

Now write a new review for the following product that matches the user's style above.

Product information: [query_text]

Formatting and policy:

1. Plain text only (no HTML, Markdown, or code blocks).
2. Do not repeat sentences or phrases.
3. Do not add summaries, disclaimers, or extra notes.

2. **Planner Prompt:** The prompt for the planner model that takes the original query provided by the LongLaMP benchmark for each specific individual and the past texts written by the individual to output three to six specific aspects to provide additional information about the writing style of the user. See Figure 2 below.

Figure 2: Planner Prompt

You are a helpful assistant that extracts the main aspects a reviewer is likely to discuss in a product review. Given a product description and the user's previous reviews, identify 3–6 concise, specific aspects that this reviewer would probably comment on.

Past User Profiles:
[profile_block]

Product Description and rating: [query_text]

List 3–6 concise aspects the reviewer is likely to mention (2–5 words each)

3. **MoPA Aggregation Prompt:** The prompt for the aggregator model that takes the original query provided by the LongLaMP benchmark for each specific individual and four independent responses from the LLMs to output a final personalized text for evaluation. See Figure 3 below.

Figure 3: MoPA Aggregation Prompt

You are a helpful assistant that produces a single personalized product review by integrating multiple candidate drafts into one coherent response.

Task: [query_text]

Candidate drafts:

- Draft 1: [cand A]
- Draft 2: [cand B]
- Draft 3: [cand C]
- Draft 4: [cand D]

Preserve the user's current tone and focus reflected in the drafts.

Formatting and policy:

1. Do not include commentary or comparisons between drafts.
2. Do not simply rewrite one draft; merge them coherently.
3. Use plain text only (no Markdown or code blocks).
4. Do not add summaries, disclaimers, or extra notes.

Produce the final response only.

5 Results

All experimental configurations were evaluated utilizing the ground-truth texts generated by the author. For MoPA-based models, we report results from the fifth layer, shown in Table 1, 2 and 3.

To ensure the quality of all the generated texts, they underwent a post-processing step to remove unrelated content. Specifically, HTML tags, repetitive texts, and special notes explaining the reasoning were filtered out through rule-based cleaning methods. This normalization confirmed that evaluations reflected only the quality of the style, syntax, and content of the generated response.

The results for all models on the Product Review Temporal test dataset are shown in Table 1.

For the Product Review Temporal dataset (Table 1), Cross-Agent MoPA PlanPers achieves the strongest overall performance, particularly on METEOR and ROUGE-1, while Role-Planner MoPA variant performs best on ROUGE-L. These results are highlighted in bold to indicate the

Table 1: Model Performance on Product Review Temporal (Test Dataset). K = 4, Retrieval = BM25, Number of Samples = 1000

Approach	R-1	R-L	METEOR
Standard RAG-Personalized	0.2270	0.1237	0.1167
Standard PlanPers-Personalized	0.2433	0.1287	0.1325
RAG MoPA	0.2911	0.1382	0.1823
Planner MoPA	0.2962	0.1384	0.2028
RAG MoPA-MD	0.2810	0.1349	0.1685
Planner MoPA-MD	0.2884	0.1356	0.1846
MoPA-AD	**0.3008**	0.1362	**0.2286**
Multi-Planner MoPA	0.2934	0.1372	0.1962
Role-Planner MoPA	0.3005	**0.1395**	0.1987

Table 2: Model Performance on Abstract Generation Temporal (Test Dataset). K = 4, Retrieval = BM25, Number of Samples = 1000

Approach	R-1	R-L	METEOR
Standard RAG-Personalized	0.3475	0.1983	0.2260
Standard PlanPers-Personalized	0.3448	0.1932	0.2280
RAG MoPA	0.3461	0.1887	0.2450
Planner MoPA	0.3402	0.1840	0.2421
RAG MoPA-MD	0.3449	0.1870	0.2515
Planner MoPA-MD	0.3372	0.1804	0.2493
MoPA-AD	0.3396	0.1791	**0.2519**
Multi-Planner MoPA	0.3423	0.1855	0.2490
Role-Planner MoPA	0.3427	0.1867	0.2504

best-performing configuration. Comparing the baseline approaches, PlanPers outperforms RAG-Personalized on most metrics, suggesting that the planning-based personalization strategy captures user style more effectively than retrieval alone. When the Mixture-of-Personalized-Agents framework is applied, both approaches show notable improvements. MoPA RAG-Personalized improves over its baseline across all metrics with ROUGE-1 increasing from 0.2270 to 0.2911 and METEOR rising from 0.1167 to 0.1823. Similarly, Homogeneous MoPA PlanPers outperforms the standard PlanPers baseline, demonstrating that aggregating outputs from multiple agents enhances generation quality. Among the MoPA variants, Cross-Agent MoPA PlanPers consistently outperforms all other configurations, which suggests that combining diverse agent outputs leads to text that better matches the original author's style.

The results for all the models on the Abstract Generation Temporal test dataset are shown in Table 2. For the Abstract Generation Temporal dataset (Table 2), MoPA-based approaches outperform standard personalization baselines on METEOR. Applying the Mixture-of-Personalized-Agents framework leads to consistent METEOR improvements for both retrieval and planner-based approaches; for example, RAG MoPA improves METEOR from 0.2260 to 0.2450 and MoPA-AD achieves the highest METEOR score (0.2519). Among the MoPA variants, approaches that introduce agent diversity, such as MoPA-AD and Role-Planner MoPA, tend to yield higher METEOR scores, indicating that combining multiple personalized outputs helps better match the author's style in abstract generation.

Table 3: Model Performance on Topic Writing Temporal (Test Dataset). K = 4, Retrieval = BM25, Number of Samples = 1000

Approach	R-1	R-L	METEOR
Standard RAG-Personalized	0.2433	0.1192	0.1357
Standard PlanPers-Personalized	0.2674	0.1242	0.1607
RAG MoPA	**0.2910**	**0.1258**	0.2079
Planner MoPA	0.2870	0.1216	0.2204
RAG MoPA-MD	0.2827	0.1255	0.1847
Planner MoPA-MD	0.2884	0.1229	0.2015
MoPA-AD	0.2690	0.1134	**0.2304**
Multi-Planner MoPA	0.2438	0.1078	0.2118
Role-Planner MoPA	0.2778	0.1134	0.2155

The results for all the models on the Topic Writing Temporal test dataset are shown in Table 3. On the Topic Writing Temporal dataset (Table 3), MoPA-based methods again demonstrate clear advantages over standard baselines. RAG MoPA achieves the best ROUGE-1 (0.2910) and ROUGE-L (0.1258) scores, while MoPA-AD attains the highest METEOR score (0.2304). Compared to RAG-Personalized, the standard PlanPers baseline performs better across all metrics, reinforcing the importance of structured planning for topic-focused writing. Incorporating the Mixture-of-Personalized-Agents framework further improves performance for both retrieval- and planning-based approaches, with Planner MoPA and Planner MoPA-MD consistently outperforming the standard PlanPers baseline. Similar to the abstract generation setting, MoPA variants that emphasize agent diversity tend to achieve higher METEOR scores, suggesting that combining multiple personalized perspectives improves semantic alignment and stylistic consistency in long-form topic writing.

6 Ablation Studies

To examine the effects of agent scaling and layer depth in homogeneous MoPA configurations, we varied the number of agents per layer (4-10) and measured performance across layer depths (through 5 layers). As shown in Figure 2, both MoPA and Planner MoPA maintain stable performance from 4-9 agents before collapsing at 10 agents. Figure 3 illustrates diminishing returns beyond layer 3, with the largest gains occurring between the Individual and layer 1 configurations.

7 Conclusion

In this work, we presented Mixture-of-Personalized-Agents (MoPA), a multi-agent framework for personalized long-form text generation that combines structured planning, personalized retrieval, and collaborative aggregation. Across three LongLaMP personalization tasks, MoPA consistently outperformed single-agent baselines, personalized RAG, and planning-only methods.

Our results show that structured planning provides the largest gains, while controlled agent diversity further improves stylistic consistency and semantic alignment. We also find that performance improves only up to a moderate number of agents and layers, highlighting the importance of balanced collaboration rather than scale alone.

Overall, MoPA demonstrates that structured multi-agent collaboration is an effective approach for scalable and consistent personalization without user-specific fine-tuning.

8 Future Scope

This research can be extended in several meaningful directions. Future studies may explore additional datasets beyond the three LongLaMP temporal benchmarks used in this experiment. Applying the MoPA PlanPers framework to a broader range of personalization tasks would further validate the generalizability and strengthen the evidence for its effectiveness.

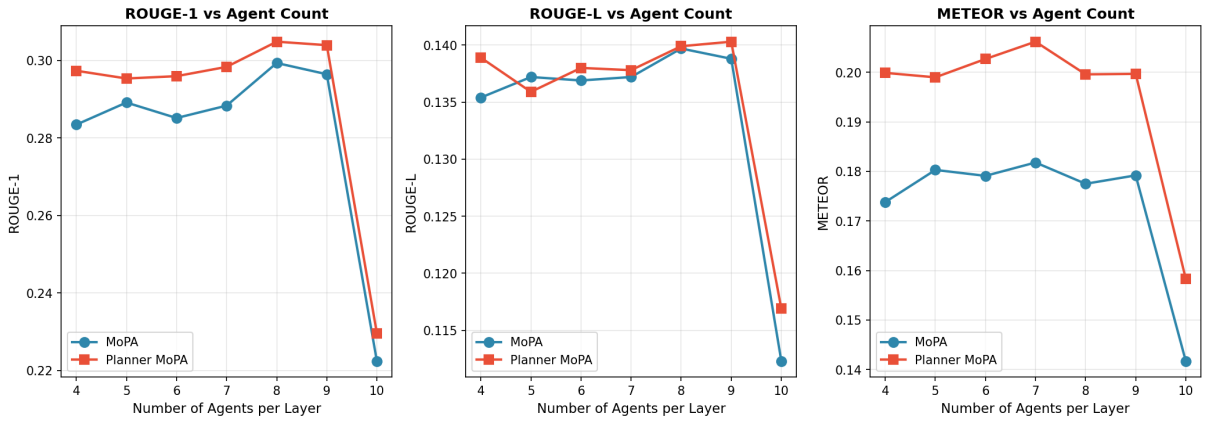

Figure 2: MoPA and Planner MoPA: Layer 5 Performance by Agent Count (GPT-4o, Product Review, Number of Samples = 200)

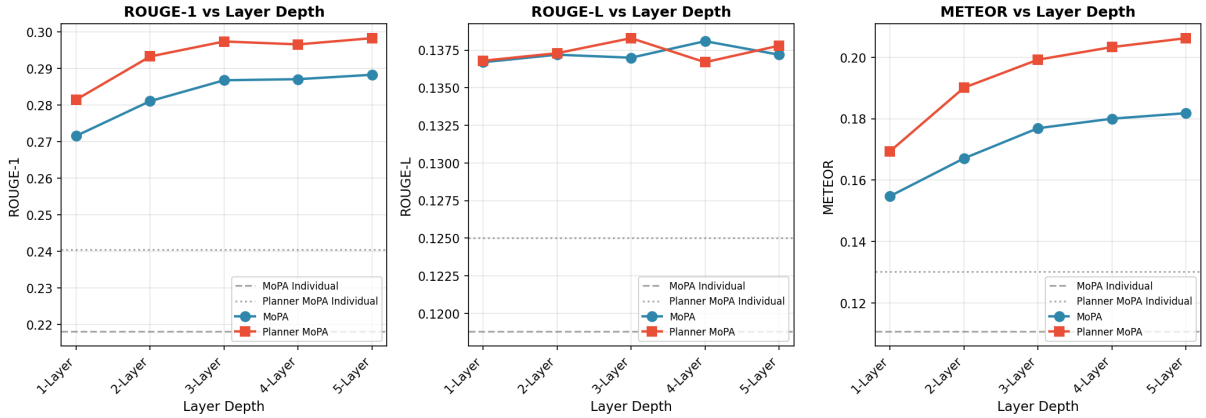

Figure 3: MoPA and Planner MoPA: Performance by Layer Depth (7Agents, GPT-4o, Product Review, Number of Samples = 200)

While our experiments focused on GPT-4o, DeepSeek, and Chat-5-GPT, future work could incorporate a wider variety and combination of LLMs, including Qwen, GPT-5 variants, Claude, and Gemini. Evaluating a more diverse set of models may yield additional performance gains per dataset and provide a deeper insight into how different model architectures interact within MoPA frameworks.

Moreover, the evaluation method can be updated beyond traditional metrics commonly used in personalization tasks. Incorporating alternative evaluation strategies, such as LLM-as-a-judge or human-in-the-loop, may offer more sophisticated perspectives on evaluating personalization quality.

Finally, our experiments were limited to four agents per layer due to the computational and time constraints. Future works may investigate a larger number of agents per layer, as well as adaptive agent selection, to further enhance the performance. Scaling the number of agents and model types for each specific dataset through tuning could lead to an additional improvement in personalized text generation.

9 Related Work

9.1 Personalized LLMs

Recent work has explored how LLMs can be tailored to individual users, rather than producing generic responses. Salemi et al. (2024) introduced the LaMP benchmark, with seven generations and classification tasks to determine whether models can utilize user profiles to generate personalized text. They proposed retrieval-based methods to incorporate relevant items from a user's history into the prompt, and showed strong improvements over standard non-personalized baselines. Follow-up studies extended this line of work to long-form

generation (Kumar et al., 2024) and personalized question answering with LaMP-QA (Salemi and Zamani, 2025). These works highlight that including user history and context leads to a more tailored model output, making personalization an important area of LLM research.

To better match the outputs with individual user needs, the field has increasingly emphasized explicit reasoning as a means of enhancing personalized text generation. Reasoning Enhanced Self Training for Personalized Generation (REST-PG) trains language models to map out reasoning steps based on the user's history and preferences before generating long-form personalized content (Salemi et al., 2025). By using self-training and expectation-maximization, REST-PG strengthens the connection between reasoning and output, resulting in strong personalization. However, placing the entire reasoning process within a single model may result in flexibility and scalability constraints as personalization complexity increases.

While REST-PG emphasizes explicit reasoning processes, another line of research focuses on preference learning and representation editing for more flexible adaptation. For example, Personalized Language Modeling from Personalized Human Feedback (P-RLHF) (Li et al., 2024b) introduces a lightweight user model that is jointly learned with the LLM, allowing the system to capture both explicit and implicit user preferences and scale across many users without fine-tuning separate models for each one. Furthermore, CHAMELEON (Zhang et al., 2025b) adopts a data and compute-efficient approach. It generates synthetic user preference data, identifies the embedding subspaces corresponding to personalized versus non-personalized traits, and edits model embeddings to nudge the model toward personalized behavior.

Beyond reasoning-based and embedding-editing methods, researchers have also proposed hierarchical and causal personalization strategies. *Progressive Personalization with Group-level Adaptation* (PROPER) (Zhang et al., 2025a) introduces a two-stage process that first learns shared preferences among user groups and then refines the model for individual users, allowing scalability to be balanced with fine-grained personalization. NextQuill (Zhao et al., 2025) applies causal analysis to locate the text segments that encode user preferences and align them, reducing unnecessary changes and better preserving the intended meaning. Together, these works complement reasoning-enhanced and embedding-based approaches by offering scalable personalization strategies and causal insight into preference modeling, which are considerations that contribute to the development of our mixture-of-agents model.

9.2 Mixture of LLMs

Another research direction focuses on combining multiple models instead of relying on a single system. Wang et al. (2024) proposed the Mixture-of-Agents (MoA) framework, where some models are proposers that generate user responses, and others are aggregators that merge these responses across layers. MoA coordinates multiple full models using only prompting, unlike Mixture-of-Experts, which splits work inside one model. Experiments on AlpacaEval 2.0, MT-Bench, and FLASK showed that MoA outperforms even GPT4 Omni while being cost-efficient. Their analysis also showed that models often improve when they can build on each other's outputs, suggesting that mixtures of LLMs can be a powerful way to improve overall performance.

Findings by Li et al. (2024a) show that simply increasing the number of independent LLM agents often outperforms more complex coordination approaches. The performance consistently improves with agent count before leveling off, which highlights the effectiveness of agent diversity but also the computational trade-offs of larger systems. These results indicate that the mixture-of-agents methods benefit most from scale and agent specialization rather than complex coordination.

Chen et al. (2024) demonstrates that the MoA framework improves text tasks by combining several smaller, specialized models instead of relying on a single large one. In their experiments, a simple MoA setup with two agents captured nearly all of the key details in long documents, outperforming GPT-4 and Claude 3 in surfacing context around Apple's Q1 2023 reports. Importantly, each agent processes different parts of the text, and then their outputs are combined, so the system can handle more information overall than a single model with a fixed context window. They show that MoA can triple the effective context window while keeping costs similar to single-model systems, making it both higher quality and practical

to scale.

References

Sandy Chen, Leqi Zeng, Abhinav Raghunathan, Flora Huang, and Terrence C. Kim. 2024. Moa is all you need: Building llm research team using mixture of agents. *Preprint*, arXiv:2409.07487.

Ishita Kumar, Snigdha Viswanathan, Sushrita Yerra, Alireza Salemi, Ryan A. Rossi, Franck Dernoncourt, Hanieh Deilamsalehy, Xiang Chen, Ruiyi Zhang, Shubham Agarwal, Nedim Lipka, Chien Van Nguyen, Thien Huu Nguyen, and Hamed Zamani. 2024. Longlamp: A benchmark for personalized long-form text generation. *Preprint*, arXiv:2407.11016.

Junyou Li, Qin Zhang, Yangbin Yu, Qiang Fu, and Deheng Ye. 2024a. More agents is all you need. *Preprint*, arXiv:2402.05120.

Xinyu Li, Ruiyang Zhou, Zachary C. Lipton, and Liu Leqi. 2024b. Personalized language modeling from personalized human feedback. *Preprint*, arXiv:2402.05133.

Alireza Salemi, Cheng Li, Mingyang Zhang, Qiaozhu Mei, Weize Kong, Tao Chen, Zhuowan Li, Michael Bendersky, and Hamed Zamani. 2025. Reasoning-enhanced self-training for long-form personalized text generation. *Preprint*, arXiv:2501.04167.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024. Lamp: When large language models meet personalization. *Preprint*, arXiv:2304.11406.

Alireza Salemi and Hamed Zamani. 2025. Lamp-qa: A benchmark for personalized long-form question answering. *Preprint*, arXiv:2506.00137.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. 2024. Mixture-of-agents enhances large language model capabilities. *Preprint*, arXiv:2406.04692.

Linhai Zhang, Jialong Wu, Deyu Zhou, and Yulan He. 2025a. Progressive personalization with group-level adaptation (proper). *Preprint*, arXiv:2503.01303.

Yijing Zhang, Dyah Adila, Changho Shin, and Frederic Sala. 2025b. Chameleon: Efficient representation editing for personalized language models. *Preprint*, arXiv:2503.01048.

Xiaoyan Zhao, Juntao You, Wenjie Wang, Hong Cheng, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. 2025. Nextquill: Causal preference modeling for personalized language generation. *Preprint*, arXiv:2506.02368.