Learning to Retrieve User History and Generate User Profiles for Personalized Persuasiveness Prediction

Sejun Park [1] **Yoonah Park** [1] **Jongwon Lim** [1] **Yohan Jo** [1]

Abstract

Estimating the persuasiveness of messages is critical in various applications, from recommender systems to safety assessment of LLMs. While it is imperative to consider the target persuadee's characteristics, such as their values, experiences, and reasoning styles, there is currently no established systematic framework to optimize leveraging a persuadee's past activities (e.g., conversations) to the benefit of a persuasiveness prediction model. To address this problem, we propose a context-aware user profiling framework with two trainable components: a *query generator* that generates optimal queries to retrieve persuasion-relevant records from a user's history, and a *profiler* that summarizes these records into a profile to effectively inform the persuasiveness prediction model. Our evaluation on the ChangeMyView Reddit dataset shows consistent improvements over existing methods across multiple predictor models, raising F1 from 33% to 47% on `Llama-3.3-70B-Instruct`. Further analysis shows that effective user profiles are context-dependent and predictor-specific, rather than relying on static attributes or surface-level similarity. Together, these results highlight the importance of task-oriented, context-dependent user profiling for personalized persuasiveness prediction. Our code and data are available at `https://github.com/holi-lab/ReCAP`.

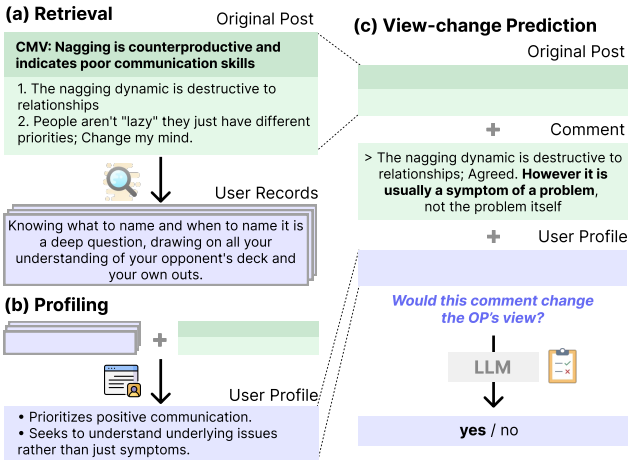

Figure 1. Overview of the view-change prediction task with context-aware user profiling on CMV. Given an original post, the system (a) retrieves relevant user records, (b) constructs a textual user profile, and (c) predicts whether a comment will change the user's view.

1. Introduction

Large language models (LLMs) are increasingly used in decision-support applications that aim to influence human behavior or beliefs, such as health coaching, tutoring, and targeted marketing (Salvi et al., 2024; Hackenburg et al., 2025). In these settings, an LLM may generate or evaluate multiple candidate messages (e.g., campaign messages for marketing companies) to assist a human decision maker, requiring the system to determine which message is most likely to persuade a target user. We refer to this problem as *persuasiveness prediction*, defined as predicting a user's belief or attitude change in response to a given message (Perloff, 2021). The main challenge in persuasiveness prediction stems from the fact that persuasion is inherently personalized: the same argument may be compelling for one user but ineffective for another, depending on factors such as beliefs, values, experiences, and reasoning style (Lukin et al., 2017; Durmus & Cardie, 2018; Al Khatib et al., 2020). As a result, accurate persuasiveness prediction requires inferring how each individual user is likely to interpret and respond to a message. In practice, such inference must rely on signals from a user's past interaction history, as explicit user attributes are often unavailable. This motivates methods that infer user characteristics from historical interactions to enable personalized persuasiveness prediction.

We formulate this problem as a *personalized view-change prediction* task using data from the `ChangeMyView`

[1]Graduate School of Data Science, Seoul National University. Correspondence to: Yohan Jo <yohan.jo@snu.ac.kr>.

Pluralistic Alignment Workshop @ ICML 2026, Seoul, South Korea. Copyright 2026 by the author(s).

(CMV) Reddit forum. In this setting, each instance consists of (i) an original post expressing a user's view on a specific topic, (ii) a comment responding to the post with the intent to change that view, and (iii) the user's historical *records*, consisting of past Reddit posts and comments. The goal is to predict whether the comment will successfully change the user's stated view by leveraging user information inferred from the user's history (Figure 1c; Tan et al., 2016). To make accurate predictions, a system requires to construct user representations from their historical records that capture what matters for the current persuasion context (Figure 1a,b; Li et al., 2016; Xu et al., 2025). However, existing approaches typically rely on heuristic retrieval methods (e.g., selecting recent records or random sampling) to select relevant historical records and generic profiling techniques (e.g., extracting demographic traits) to summarize user characteristics from those records (Hackenburg et al., 2025; Al Khatib et al., 2020; Salvi et al., 2024). We argue that these approaches are insufficient because persuasion is inherently context-dependent: which aspects of a user's history are informative depends on the topic and stance of the original post as well as the argument in the candidate message (Tan et al., 2016; Ji et al., 2018).

To address this limitation, we propose a framework with two learnable modules for generating a user profile in textual form (Figure 1) : (i) a *query generator* that produces retrieval queries to identify persuasion-relevant records from a user's history, and (ii) a *profiler* that summarizes the retrieved records into a textual user profile, conditioned on the original post. This profile, together with the original post, is then used by a predictor model to determine whether the candidate message would change the user's view. Examples of the final user profiles are presented in Appendix E.

We train the components through the joint optimization of a query generator and a profiler in the following three steps. First, we train the *profiler* to leverage user records to generate effective textual profiles for view-change prediction. Second, building on this profiler, we score each user record based on the effectiveness of its resulting profiles for view-change prediction. Third, we train the *query generator* to retrieve high-scoring user records from the user history pool.

Our evaluation on the CMV dataset shows consistent gains over prior approaches, demonstrating the effectiveness of our framework for personalized persuasiveness prediction, with further gains on OpinionQA and PRISM confirming that the benefits extend beyond CMV. Further analyses show that persuasion-relevant user characteristics vary across posts and predictor models, highlighting the need for predictor-specific, context-aware user profiles rather than generic or static attributes. Our retrieval-based design also exhibits computational advantages; compared to many baselines that use a summary of the entire user history, our method keeps per-instance inference cost 6–13× lower. Beyond view-change prediction, these findings suggest that effective personalization requires learning both what to retrieve from a user's history and how to summarize it for the given context.

Our contributions are threefold:

1. **Annotation-free learnable pipeline.** We train retrieval and profiling via view-change prediction as a utility-based signal, without ground-truth annotations.
2. **Persuasion-aware query generation.** We retrieve user records by context-relevant aspects rather than semantic similarity.
3. **Context- and predictor-dependent profiling.** We show effective profiling is dynamic and predictor-specific, not static or predictor-agnostic.

2. Related Work

Early Work on Personalized Persuasion Since Tan et al. (2016) established the view-change prediction task on the CMV dataset, subsequent work has enriched modeling by incorporating richer linguistic features, such as interaction dynamics and discourse relations (Ji et al., 2018; Hidey & McKeown, 2018). More and more research demonstrated the importance of personalization by leveraging persuadee characteristics in persuasion outcome prediction, including ideology, demographic, and personality traits (Lukin et al., 2017; Durmus & Cardie, 2018; 2019a;b; Al Khatib et al., 2020). However, they largely rely on pre-defined, explicit user attributes. Leveraging recent advances in LLMs, we infer richer persuasion-relevant user information from the users' past writings.

User Profiling for LLM Personalization Early work has formulated LLM personalization as making models behave *like* a specific user given their historical writings (Salemi et al., 2024b; Mysore et al., 2024). Building on this, several studies have explored retrieval- and profiling-based approaches (Richardson et al., 2023; Li et al., 2024; Salemi et al., 2024a; Zhang, 2024). These methods focus on linguistic style and topical relevance, remaining limited in capturing user *values* (Qin et al., 2025), which are crucial for personalized persuasion. Studies on personalized dialogue agents construct user profiles via summarization to generate user-aligned responses (Zhong et al., 2024; Wang et al., 2025), but remain limited in dynamically adapting profiles to the current interaction context.

Another line of work fine-tunes the predictor on users' historical data to encode user-specific information (Zhang et al., 2025). While effective, it requires retraining as new user data arrives, limiting scalability in practice. In contrast, we

keep the predictor fixed and focus on constructing context-aware user profiles, and thus do not directly compare against such approaches.

3. User Profiling Framework

3.1. Problem Formulation

We construct a dataset from the `ChangeMyView` (CMV) Reddit forum, where users post opinions and award a *delta* to comments that change their views. The discussions cover diverse topics, including politics, personal values, and everyday issues. To support personalized prediction, we collect each user's historical posts and comments from both CMV and other subreddits. We split this dataset into training, validation, and test sets with an 8:1:1 ratio, partitioning at the original poster (OP) level so that no OP appears in more than one split. This prevents leakage of OP-specific information from training to evaluation. For training and validation, we subsample up to 100 user records per post. This is necessary because user records, which are scored via repeated view-change prediction (Section 3.3.2), are large and highly variable in size (mean 784, max 19K records per post). Concretely, we use the *delta* comment as a retrieval query and build the pool using a hybrid retriever that combines BM25 with BGE-M3 semantic similarity. This substantially reduces computational cost while preserving records that are informative for view-change prediction.

To formalize the task, we represent each data instance as a tuple (u, x_i, c_i, y_i, R_u). Here, u denotes a user; x_i is an original post authored by u in the CMV forum that expresses the user's initial view on a topic; and c_i is a comment written by another user in response to x_i. Each comment is labeled as a *delta* or *non-delta*, with a label $y_i \in \{0, 1\}$ indicating whether it changed the user's view ($y_i = 1$ if a *delta* was awarded). We further provide access to the user's historical records $R_u = \{r^{u,1}, r^{u,2}, \ldots, r^{u,|R_u|}\}$, where each $r^{u,j}$ is a Reddit post or comment written by u prior to x_i. The personalized view-change task is formulated as $\tilde{y}_i = f(u, R_u, x_i, c_i)$, where the goal is to predict whether c_i will change user u's view expressed in x_i, given the user's history R_u.

3.2. Inference

To address this task, we introduce a three-stage inference pipeline comprising *retrieval*, *profiling*, and *view-change prediction* (Figure 2a–c). Since R_u is typically large and noisy, direct conditioning on the entire set is impractical. Instead, we construct a compact user profile P_i that summarizes persuasion-relevant information about u and condition the prediction solely on P_i. The construction of P_i consists of two stages: retrieval and profiling.

The retrieval stage (Figure 2a) selects a subset of k records from R_u that are directly used for profile construction. For this, we first generate a retrieval query q_i using a trainable query generator ϕ^{query}, which takes the original post x_i as input. Using an embedding-based retriever $\mathcal{M}^{\text{ret}}$, we retrieve the top-k records most relevant to q_i:

$$\{r^{u,i_1}, \ldots, r^{u,i_k}\} = \mathcal{M}^{\text{ret}}(q_i, R_u, k) \subseteq R_u.$$

The profiling stage (Figure 2b) constructs a natural-language user profile P_i by summarizing the retrieved records into a textual representation that the predictor model can effectively utilize. We employ a trainable LLM-based profiler ϕ^{prof} that takes the retrieved records and the original post x_i as input, enabling the profile to be conditioned on the persuasion context expressed in x_i:

$$P_i = \phi^{\text{prof}}\big(\mathcal{M}^{\text{ret}}(q_i, R_u, k)\,;\, x_i\big).$$

Finally, at the prediction stage (Figure 2c), an LLM-based predictor $\mathcal{M}^{\text{pred}}$ takes post x_i, comment c_i, and the user profile P_i as input to predict whether c_i will change the user's view expressed in x_i:

$$\tilde{y}_i = \mathcal{M}^{\text{pred}}(x_i, c_i\,;\, P_i).$$

3.3. Training

Our framework consists of two learnable components: the query generation module (ϕ^{query}) and the profiler (ϕ^{prof}). We jointly optimize them in three stages: *profiler training*, *record-level persuasion utility scoring*, and *query generator training* (Figure 2d–f). We first train the profiler to generate effective user profiles from retrieved records for view-change prediction (Section 3.3.1). Next, we train the query generator to produce queries that can retrieve effective user records (Section 3.3.3), where the utility of each user record is scored based on the effectiveness of its resulting profiles for view-change prediction (Section 3.3.2).

3.3.1. Profiler

Our core hypothesis is that an ideal user profile should be optimized for personalized view-change prediction rather than merely summarizing user history. Since ground-truth profiles for this task are unavailable, we adopt a weakly supervised approach and optimize the profiler via DPO (Figure 2d). We use end-task performance as the preference signal: profiles that successfully predict view change are treated as *chosen*, while unsuccessful ones are treated as *rejected*. To construct such preference data, for each post x, we randomly sample multiple groups of historical records and prompt the base profiler to generate candidate profiles. We evaluate each profile by its task performance across all comments associated with x, using the resulting F1 score as a measure of profile quality. Preference pairs are then constructed by pairing higher-scoring profiles with lower-scoring ones separated by a sufficient F1 margin, and the

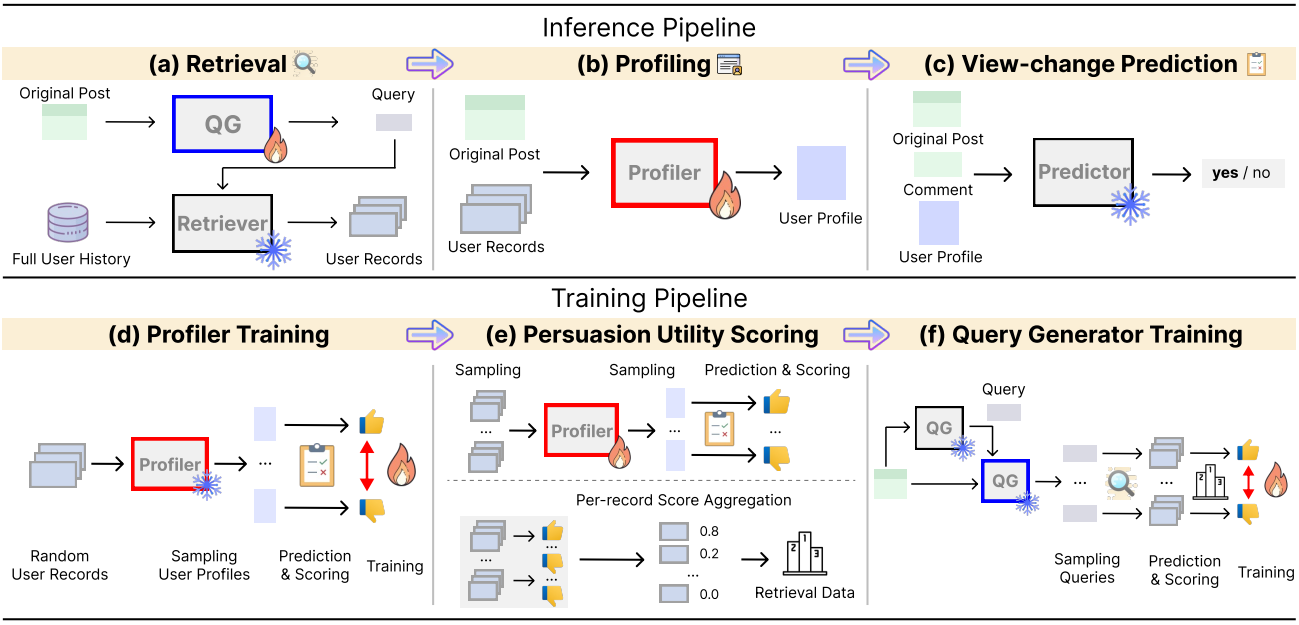

Figure 2. Overview of the proposed framework. *Flame* and *snowflake* icons denote trainable and frozen models, respectively. QG denotes our query generator model. The top illustrates the inference pipeline with three stages: (a) retrieval, (b) profiling, and (c) view-change prediction. The bottom shows the training pipeline, consisting of (d) profiler training, (e) record-level persuasion utility scoring, and (f) query generator training.

profiler is trained to prefer higher-quality profiles. Training details are provided in Appendix C.

3.3.2. Record-Level Persuasion Utility Scoring

After training the profiler, we estimate the *persuasion utility* of individual user records. This step enables learning in the retrieval stage: training a record-selection module requires supervision that reflects how useful each record is for predicting persuasion outcomes. However, existing datasets lack ground-truth annotations identifying which individual records are most informative for persuasion prediction, and collecting such labels from Reddit users is infeasible. We therefore derive record-level supervision by estimating each record's contribution to view-change prediction performance (Figure 2e). For each record $r \in R_u$, we estimate its contribution by evaluating its effect across different record sets that include r. Specifically, we randomly partition the records into groups of five, and repeat this grouping process three times. For each group, we generate three user profiles using the trained profiler with a decoding temperature of 0.7. As a result, each record r is associated with a total of nine generated profiles. We aggregate the F1 scores from all view-change prediction instances performed using profiles that include record r, and use the result as its persuasion utility score.

3.3.3. Query Generator

For effective user profiling, we must retrieve historical user records informative for persuading each user. A naive approach would use the post text directly as a retrieval query. However, CMV posts often lack explicit user-specific attributes critical for persuasion—such as underlying values, relevant experiences, or decision-making styles. Post-only queries thus often fail to retrieve sufficiently useful records for predicting view change. To address this, we train an LLM-based query generator to produce user-focused queries that explicitly target attributes absent from the post but potentially critical for persuasion.

Because training the model to directly infer these implicit attributes from the post alone is difficult, we adopt a two-stage training for the query generator (Figure 2f). First, we prompt the model to generate a user-focused question that targets information not present in the post but likely to influence persuasion (e.g., for a healthcare policy post: "What are the user's core values regarding government intervention in individual choice?"). Second, we train the model to take both the post and the generated user-focused question as input and generate a single retrieval query that contextualizes the user attribute using salient cues from the post (e.g., "Does the user prioritize individual autonomy over collective benefit when it comes to healthcare access?"). By learning to ground user attributes in the context of the post, the model can better identify which user information is likely to affect persuasion outcomes.

Table 1. Retrieval performance of different query formulation strategies. Random reports the average performance over 10 runs. BGE-Post and BGE-Post-Tuned use an embedding-based retriever based on BGE-M3, with and without retriever fine-tuning, respectively.

Query Strategy	Mean NCG@5	Mean NDCG@5
Random	0.6173	0.6080
BGE-Post	0.6267	0.6180
BGE-Post-Tuned	0.6280	0.6162
HyDE	0.6229	0.6126
Ours	**0.6357**	**0.6214**

For each post, we sample multiple candidate queries, retrieve user records for each, and score each query by NDCG@5 based on the persuasion utility of the retrieved records. The model is then trained via DPO to prefer queries that yield higher-quality retrieval. At inference, the trained query generator receives only the post and outputs a user-focused query that effectively surfaces persuasion-relevant user records. Full details of candidate generation, preference construction, and optimization are provided in Appendix D.

4. Experiments

In this section, we evaluate our proposed framework through two complementary analyses: (1) a retrieval-side evaluation of our query generation strategy based on persuasion utility scores (Section 4.1); (2) end-to-end view-change prediction performance, which evaluates the combined effect of all pipeline components (Section 4.2).

Experimental Setup We use data collected from the CMV Reddit forum, as described in Section 3.1. Detailed dataset statistics are provided in Appendix A. For the learnable components, we employ `Llama-3.1-8B-Instruct` as the backbone for both the query generator and the profiler. For embedding-based retrieval, we use the `BGE-M3` embedding model. For view-change predictor models, we use two open-weight models at different scales, `Llama-3.1-8B-Instruct` and `Llama-3.3-70B-Instruct`, and a closed-source model, `GPT-4o-mini`, to assess whether our trainable user profiling framework generalizes across predictors. We evaluate performance using the F1 score, which is well-suited to the inherently imbalanced comment labels (e.g., few delta comments among many non-delta comments).

4.1. Retrieval-side Experiments

Setup We analyze the retrieval component in isolation, focusing on how query generation affects the retrieval of persuasion-relevant user records. Specifically, we compare a random baseline (mean over 10 runs), embedding-based retrieval methods, and our query generation strategies using pre-computed utility scores for individual records (Section 3.3.2). For embedding-based retrieval baselines, we evaluate BGE-POST, which directly use the original post text as the retrieval query, and HYDE (Gao et al., 2023), which generates a hypothetical document that approximates the retrieval target and uses it as the query. Concretely, for HYDE, we prompt `Llama-3.1-8B-Instruct` with the original post to generate a plausible user record that is likely to be relevant in the given persuasion context.

Results Table 1 reports retrieval performance measured by utility-based NCG@5 and NDCG@5. The results indicate that dense retrieval using the post text as the query (BGE-Post) is inherently limited, and that fine-tuning the retriever under the same post-only query formulation (BGE-Post-Tuned) leads to only marginal improvements. This suggests that the bottleneck is not retriever capacity but the incompleteness of the post as a query for eliciting persuasion-relevant user attributes. In contrast, our method improves over BGE-Post and HyDE by transforming the post into a user-focused query that explicitly targets missing attributes conditioned on the post. Consistent with this, end-to-end results (Section 4.2) show that our framework achieves the best view-change prediction performance, highlighting that persuasion-aware query formulation is more beneficial for the full pipeline.

4.2. End-to-End View-Change Prediction

Setup We evaluate the end-to-end view-change prediction performance of our overall pipeline, comparing it against (1) existing personalized profiling frameworks, and (2) different combinations of retrieval and profiling baselines.

We first compare our method with prior **user profiling frameworks** for personalized dialogue and retrieval-augmented generation, including PAG (Richardson et al., 2023), HSUMM (Zhong et al., 2024), and RECURSUMM (Wang et al., 2025). Details of these baselines are provided in Appendix F. We additionally evaluate two ablations: NO PERSONALIZATION, which performs view-change prediction without user profiles or historical records, and RETRIEVAL-ONLY, which conditions the predictor on raw retrieved records without profile construction.

Next, we conduct a more detailed comparison across different retriever-profiler combinations. For **retrieval** variants, we compare embedding-based strategies evaluated in Section 4.1 (BGE-POST and HYDE), a sparse retrieval baseline using the post as the query (BM25-POST), and heuristic baselines (RANDOM and RECENT). For **profiling** variants, we consider three approaches to user profile construction: (i) DEMOGRAPHIC, which extracts demographic

Table 2. End-to-end comparison of our proposed framework with prior user profiling approaches. The table reports F1 and area under the ROC curve (AUC) for view-change prediction across three predictor models.

Method	Llama-3.1-8B-Instruct		Llama-3.3-70B-Instruct		GPT-4o mini	
	F1	AUC	F1	AUC	F1	AUC
No Personalization	0.3457	0.5677	0.3284	0.6538	0.2525	**0.6415**
PAG	0.2571	0.5775	0.3141	0.6346	0.0833	0.6165
Recursumm	0.3133	0.5869	0.4139	0.6571	0.1050	0.6318
Hsumm	0.3244	0.5965	0.4063	0.6615	0.1128	0.6214
Retrieval-only	0.2952	0.5424	0.4177	0.6635	0.1323	0.6306
Ours	**0.4000**	**0.6158**	**0.4661**	**0.6828**	**0.2787**	0.6299

Table 3. Effect of retriever and profiler choices on view-change prediction under different predictors (F1). Random reports the average performance over 10 random runs. Underlined results denote our final proposed method, while **boldface** highlights the best-performing configuration within each column. Column groups correspond to different predictor models, with sub-columns indicating profiler configurations (demographic, base profiler, and our trained profiler). Corresponding results using the AUC metric are reported in Appendix G.6.

	Llama-3.1-8B-Instruct			Llama-3.3-70B-Instruct			GPT-4o mini		
Retrieval	Demograph.	Base	*Ours*	Demograph.	Base	*Ours*	Demograph.	Base	*Ours*
Recent	0.3364	0.3805	0.3951	0.3891	**0.4058**	0.4428	0.0714	0.1629	0.2533
Random	0.3199	0.3758	0.3860	0.4038	0.3979	0.4304	0.0578	0.1516	0.2476
BM25	0.3286	0.3636	0.3742	0.3905	0.3981	0.4218	0.0720	0.1658	0.2754
BGE	0.3286	0.3410	0.3554	**0.3912**	0.3799	0.4454	0.0663	0.1465	0.2441
HyDE	0.3344	0.3701	0.3785	0.3800	0.3917	0.4507	0.0720	**0.1805**	0.2570
Ours	**0.3466**	**0.3893**	**0.4000**	0.3837	0.3929	**0.4661**	**0.0765**	0.1695	**0.2787**

attributes from retrieved records using `GPT-4.1-mini` (Hackenburg et al., 2025; Salvi et al., 2024), (ii) BASE PROFILER, an instruction-tuned LLM without additional training prompted to summarize retrieved records, and (iii) DPO PROFILER, our profiler trained via DPO (Section 3.3.1).

Results Table 2 shows that existing personalization frameworks transfer poorly to view-change prediction. These methods primarily aim to generate user-aligned responses or compress a user's history, and generic profiles can even hurt performance for `Llama-3.1-8B-Instruct` and `GPT-4o-mini` compared to NO PERSONALIZATION. In contrast, our framework yields consistent gains across predictors, raising F1 on `Llama-3.3-70B-Instruct` from 0.3284 (NO PERSONALIZATION) to 0.4661, indicating that task-oriented, trainable profiling is crucial for personalized persuasion prediction. Compared to the retrieval-only baseline, our approach yields consistent gains across all predictors; for example, F1 on `Llama-3.3-70B-Instruct` improves from 0.4177 to 0.4661, highlighting the critical role of the profiler.

Table 3 further decomposes performance by retriever–profiler combinations. Our DPO-trained profiler consistently outperforms demographic and base profiling baselines across all predictors, while demographic profiles perform poorly, suggesting that persuasion-relevant signals are not well captured by demographics alone. On the retrieval side, our query generator delivers the strongest end-to-end performance overall; notably, RECENT is a competitive baseline, which aligns with Zhang et al. (2025). Our query generator shows substantial synergy with the trained profiler, highlighting that record-level scoring using the trained profiler provides a clear learning signal. Together, these results highlight that view-change prediction benefits most from profiles optimized for the task and retrieval queries that expose persuasion-relevant user attributes, rather than from generic personalization pipelines or standard post-only retrieval.

4.3. Generalization to Other Datasets

To test whether our framework generalizes beyond CMV, we evaluate it on two personalization datasets with different characteristics: OpinionQA (Santurkar et al., 2023), a survey-based stance prediction task, and PRISM (Kirk et al., 2024), a real-world multi-session conversational preference dataset.

For OpinionQA, no interaction history is available; we instead construct each user's history from their responses to *other* survey questions, testing whether our framework can leverage non-conversational user records. Prediction accuracy replaces delta labels as the supervision signal

Table 4. Profiling comparison on PRISM and OpinionQA with `Llama-3.1-8B-Instruct` as the predictor.

Profiling	PRISM (F1)	OpinionQA (Acc.)
No Personalization	0.5751	0.4354
RECURSUMM	0.5466	0.4218
HSUMM	0.5051	0.4014
PAG	0.4754	0.5238
Ours	**0.5879**	**0.5306**

Table 5. Retrieval comparison on PRISM and OpinionQA with `Llama-3.1-8B-Instruct` as the predictor.

Retrieval	PRISM (F1)	OpinionQA (Acc.)
Random	0.5686	0.4323
BM25	0.5678	0.4898
BGE	0.5814	0.4966
Ours	**0.5879**	**0.5306**

for utility scoring. For PRISM, we treat each user's top-rated response as a positive instance and responses with preference score gap ≥ 0.3 from the top as negatives, enabling the same pairwise formulation as CMV. We use `Llama-3.1-8B-Instruct` as the predictor.

Tables 4 and 5 show that our framework outperforms all profiling and retrieval baselines on both datasets. The OpinionQA gain is particularly notable: user history here consists of isolated survey responses rather than interaction logs, yet our framework still produces effective profiles. We also observe that full-history summarization baselines (HSUMM, RECURSUMM) fall below the No Personalization baseline on both datasets. This mirrors our CMV findings and suggests that generic summarization fails to capture task-relevant user signals across domains.

5. Analysis

5.1. Profiler Analysis

In this section, we analyze the impact of profiler training by comparing profiles generated by the base profiler (*original profiles*) and the trained profiler (*trained profiles*) on the test set. We present two key analyses below, with results for all predictor models reported in Appendix H.

The effectiveness of profiler training varies by post topic. To analyze how profiler effectiveness varies across post characteristics, we annotate each post by *topic* and *claim type* using `GPT-4.1-mini`. Topics are categorized into *Political* (27.4%), *Sociomoral* (46.4%), and *Others* (26.2%), and claim types into *Interpretation* (39.9%) and *Evaluation* (60.1%), following prior work (Hidey et al., 2017; Priniski & Horne, 2018). Across most topic-claim combinations, trained profiles consistently outperform original profiles in F1 (Figure 3). The only exception is political posts under

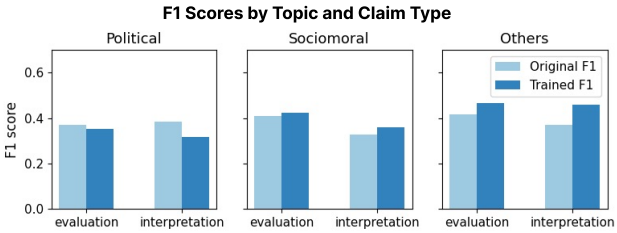

Figure 3. F1 by topic and claim type of the post, comparing the original and trained profilers. `Llama-3.1-8B-Instruct` is used as the predictor model.

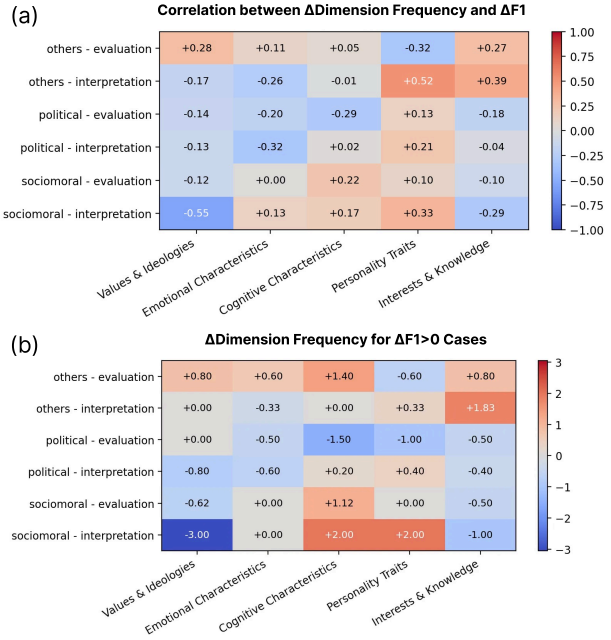

Figure 4. Analysis of profile-dimension frequency shifts (ΔDF) and performance gains (ΔF1) between the original and trained profilers. (a) Correlation between ΔDF and ΔF1. (b) ΔDF for cases with $\Delta F1 > 0$. `Llama-3.1-8B-Instruct` is used as the predictor model.

`Llama-3.1-8B-Instruct`, where profiling benefits sociomoral and other topics but not political posts, likely due to the dominance of group identities in political persuasion. Overall, the results suggest that the trained profiler effectively captures individual-level characteristics relevant to persuasion.

Important profile dimensions vary by post characteristics and the predictor model. To further analyze profile content, we decompose each profile into sentence-level units, referred to as *profile items*. We annotate each item for the presence of five profile dimensions using `GPT-4.1-mini`; the five dimensions—*Values & Ideologies*, *Emotional Characteristics*, *Cognitive Characteristics*, *Personality Traits*, and *Interests & Knowledge*—are constructed from persuasion literature (Fabrigar & Petty, 1999;

Al Khatib et al., 2020). For each profile, we count the frequency of profile items associated with each dimension and compute *Profile-F1*, the F1 score aggregated over all comments associated with the profile. Figure 4a shows the correlation between (i) change in item frequency for each dimension from original to trained profiles and (ii) the change in Profile-F1.

Our analysis yields three key findings: (1) No single profile dimension is consistently beneficial or detrimental across all posts. (2) The effect of each dimension is strongly post-dependent: for example, cognitive traits (e.g., reasoning styles, decision making styles) are positively associated with performance gains for political evaluation posts, but negatively associated with sociomoral evaluation posts. (3) For cases where performance improves, shifts in item frequency across profile dimensions align with these correlation trends (Figure 4b), indicating that different dimensions are emphasized depending on the post. These three patterns remain consistent across different predictor models. However, the specific association patterns between post characteristics and profile dimensions vary substantially with the choice of predictor model. Taken together, these results suggest that persuasion-relevant dimensions differ across posts and the predictor models, and that the trained profiler captures this post-dependent and predictor-specific variation by dynamically adjusting the emphasis on different dimensions. Results for all predictor models are reported in Appendix H.

5.2. User Record Analysis

We conduct an analysis of user records scored via persuasion-utility scoring, using LLAMA-3.1-8B-INSTRUCT as the predictor model, focusing on (1) semantic differences between high- and low-scoring records and (2) cross-model patterns in utility scores. More detailed analyses are provided in Appendix I.

Low-scoring records are not semantically dissimilar to the post. We analyze pairs of top-1 and bottom-1 records for the same post in the validation set, focusing on cases where the bottom-1 record receives an F1 score of zero. Following Section 5.1, we annotate the records along *topic* and *claim type*. Contrary to the hypothesis that low-scoring records fail due to semantic misalignment with the post, we observe the opposite trend: low-scoring records are more likely than high-scoring ones to share the same topic or claim type as the post. This highlights the need for finer-grained contextualization in persuasion.

Different predictor models prefer different user records. We further compare persuasion utility scores across predictor models and find little agreement in their preferred records (Table 6). Pairwise comparisons of the top-5 records show low overlap (0.24–0.28), corresponding to only about 1.25 shared records on average. Similarly, Spearman rank correlations are near zero (-0.005–0.083), indicating weak consistency in relative ordering. These results indicate that record utility for view-change prediction is highly model-dependent, motivating the training of predictor-specific retrieval modules.

Table 6. Pairwise agreement between predictors on record-level utility scores, measured by mean top-5 overlap and Spearman ρ. GPT, Llama8B, and Llama70B correspond to `GPT-4o-mini`, `Llama-3.1-8B-Instruct`, and `Llama-3.3-70B-Instruct`, respectively.

Model Pair	Top-5 Overlap	Spearman ρ
GPT / Llama70B	0.273	0.007
GPT / Llama8B	0.281	0.083
Llama70B / Llama8B	0.245	-0.005

Table 7. Per-instance inference cost across profiling methods.

Method	Tokens	FLOPs
PAG	3,228	4.71×10^{13}
Ours	3,429	5.09×10^{13}
RECURSUMM	21,678	3.58×10^{14}
HSUMM	42,880	7.03×10^{14}

5.3. Efficiency Analysis

For real-time decision-support, inference latency matters. Under a deployment scenario where user-history embeddings are maintained offline, full-history summarization baselines (HSUMM, RECURSUMM) incur 6–13$\times$ our token and FLOP cost, with overhead growing linearly in history size (Table 7). Retrieval-based methods instead scale with the retrieved subset: relative to PAG, our query-generation step adds only $\sim$6–8% overhead while yielding an average $+0.16$ F1 gain across predictors (Table 2). Stage-wise breakdowns are in Appendix J.

6. Conclusion

We introduce a trainable user profiling framework that captures persuasion-relevant user factors. Experiments on the CMV dataset show that our approach consistently outperforms baselines by constructing context-dependent profiles tailored to the downstream predictor, with consistent gains also observed on OpinionQA and PRISM. By learning to retrieve and construct task-oriented user profiles, our framework enables scalable, context-sensitive personalization without retraining predictors or requiring extensive user annotations, at 6–13× lower inference cost than full-history summarization baselines, making it practical for real-world decision-support systems such as conversational agents, recommendation, and coaching.

Impact Statement

Research on predicting view change in online discussions could be related to ethical considerations about user autonomy and the responsible use of predictive insights. In this work, we strictly focus on predicting whether a view change occurs, rather than intervening in user behavior. The framework does not include any mechanisms for generating persuasive content or intervening on individuals, but is designed to enhance understanding of view-change dynamics in natural settings.

All experiments are conducted using publicly available and anonymized data, without any personally identifiable information.

References

Al Khatib, K., Völske, M., Syed, S., Kolyada, N., and Stein, B. Exploiting personal characteristics of debaters for predicting persuasiveness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7067–7072, 2020.

Durmus, E. and Cardie, C. Exploring the role of prior beliefs for argument persuasion. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1035–1045, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1094. URL https://aclanthology.org/N18-1094/.

Durmus, E. and Cardie, C. A corpus for modeling user and language effects in argumentation on online debating. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 602–607, Florence, Italy, 2019a. Association for Computational Linguistics. doi: 10.18653/v1/P19-1057. URL https://aclanthology.org/P19-1057/.

Durmus, E. and Cardie, C. Modeling the factors of user success in online debate. In *The World Wide Web Conference (WWW) 2019*, pp. 2701–2707, 2019b. doi: 10.1145/3308558.3313676. URL https://doi.org/10.1145/3308558.3313676.

Fabrigar, L. R. and Petty, R. E. The role of the affective and cognitive bases of attitudes in susceptibility to affectively and cognitively based persuasion. *Personality and social psychology bulletin*, 25(3):363–381, 1999.

Gao, L., Ma, X., Lin, J., and Callan, J. Precise zero-shot dense retrieval without relevance labels. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1762–1777, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.99. URL https://aclanthology.org/2023.acl-long.99/.

Hackenburg, K., Tappin, B. M., Hewitt, L., Saunders, E., Black, S., Lin, H., Fist, C., Margetts, H., Rand, D. G., and Summerfield, C. The levers of political persuasion with conversational artificial intelligence. *Science*, 390 (6777):eaea3884, 2025.

Hidey, C. and McKeown, K. Persuasive influence detection: The role of argument sequencing. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2018. URL https://ojs.aaai.org/index.php/AAAI/article/view/12003.

Hidey, C., Musi, E., Hwang, A., Muresan, S., and McKeown, K. Analyzing the semantic types of claims and premises in an online persuasive forum. In *Proceedings of the 4th Workshop on Argument Mining*, pp. 11–21, 2017.

Ji, L., Wei, Z., Hu, X., Liu, Y., Zhang, Q., and Huang, X. Incorporating argument-level interactions for persuasion comments evaluation using co-attention model. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 3703–3714, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://aclanthology.org/C18-1314/.

Kirk, H. R., Whitefield, A., Röttger, P., Bean, A., Margatina, K., Ciro, J., Mosquera, R., Bartolo, M., Williams, A., He, H., Vidgen, B., and Hale, S. A. The PRISM alignment dataset: What participatory, representative and individualised human feedback reveals about the subjective and multicultural alignment of large language models. *Advances in Neural Information Processing Systems*, 37: 105236–105344, 2024.

Li, C., Zhang, M., Mei, Q., Kong, W., and Bendersky, M. Learning to rewrite prompts for personalized text generation. In *Proceedings of the ACM Web Conference 2024*, pp. 3367–3378, 2024.

Li, J., Galley, M., Brockett, C., Spithourakis, G., Gao, J., and Dolan, B. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 994–1003, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1094. URL https://aclanthology.org/P16-1094/.

Lukin, S., Anand, P., Walker, M., and Whittaker, S. Argument strength is in the eye of the beholder: Audience

effects in persuasion. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pp. 742–753, 2017.

Mysore, S., Lu, Z., Wan, M., Yang, L., Sarrafzadeh, B., Menezes, S., Baghaee, T., Gonzalez, E. B., Neville, J., and Safavi, T. Pearl: Personalizing large language model writing assistants with generation-calibrated retrievers. In *Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U)*, pp. 198–219, 2024.

Perloff, R. M. *The Dynamics of Persuasion: Communication and Attitudes in the 21st Century*. Routledge, New York, NY, 7 edition, 2021.

Priniski, J. H. and Horne, Z. Attitude change on reddit's change my view. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 40, 2018.

Qin, W., Xu, Y., Yu, W., Shi, T., Shen, C., He, M., Fan, J., Zhang, X., and Xu, J. Similarity = value? consultation value-assessment and alignment for personalized search. In Christodoulopoulos, C., Chakraborty, T., Rose, C., and Peng, V. (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9839–9852, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.498. URL https://aclanthology.org/2025.emnlp-main.498/.

Richardson, C., Zhang, Y., Gillespie, K., Kar, S., Singh, A., Raeesy, Z., Khan, O. Z., and Sethy, A. Integrating summarization and retrieval for enhanced personalization via large language models. *arXiv preprint arXiv:2310.20081*, 2023.

Salemi, A., Kallumadi, S., and Zamani, H. Optimization methods for personalizing large language models through retrieval augmentation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 752–762, 2024a.

Salemi, A., Mysore, S., Bendersky, M., and Zamani, H. Lamp: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7370–7392, 2024b.

Salvi, F., Ribeiro, M. H., Gallotti, R., and West, R. On the conversational persuasiveness of large language models: A randomized controlled trial. *arXiv preprint arXiv:2403.14380*, 2024.

Santurkar, S., Durmus, E., Ladhak, F., Lee, C., Liang, P., and Hashimoto, T. Whose opinions do language models reflect? In *Proceedings of the 40th International Conference on Machine Learning*, pp. 29971–30004. PMLR, 2023.

Tan, C., Niculae, V., Danescu-Niculescu-Mizil, C., and Lee, L. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *Proceedings of the 25th International Conference on World Wide Web (WWW)*, pp. 613–624, 2016. doi: 10.1145/2872427.2883081.

Wang, Q., Fu, Y., Cao, Y., Wang, S., Tian, Z., and Ding, L. Recursively summarizing enables long-term dialogue memory in large language models. *Neurocomputing*, 639: 130193, 2025.

Xu, Y., Zhang, J., Salemi, A., Hu, X., Wang, W., Feng, F., Zamani, H., He, X., and Chua, T.-S. Personalized generation in large model era: A survey. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24607–24649, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.1201. URL https://aclanthology.org/2025.acl-long.1201/.

Zhang, J. Guided profile generation improves personalization with large language models. In *EMNLP (Findings)*, 2024.

Zhang, X. F., Beauchamp, N., and Wang, L. Prime: Large language model personalization with cognitive dual-memory and personalized thought process. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 33695–33724, 2025.

Zhong, W., Guo, L., Gao, Q., Ye, H., and Wang, Y. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19724–19731, 2024.

A. Dataset Collection and Statistics

We construct our dataset from the `ChangeMyView` (CMV) Reddit forum. To ensure the dataset supports the study of personalized persuasion with sufficient historical context, we applied a multi-stage filtering process to the raw data:

1. **Interaction Completeness:** We filtered for original posts (OPs) that contain at least one *delta* comment (successful persuasion) and at least one *non-delta* comment (unsuccessful persuasion). This ensures that each instance allows for the comparison of successful and unsuccessful arguments within the same context.
2. **History Availability:** We restricted the dataset to users who had at least 15 historical records (posts or comments) published prior to the timestamp of the original post. This threshold ensures there is sufficient data to construct a meaningful user profile.

After filtering, the final dataset consists of 1,676 posts from 1,326 unique users. We split this dataset into training, validation, and test sets with an approximate ratio of 8:1:1. Table 8 presents the detailed statistics for each split, including the distribution of user history length and the volume of delta and non-delta comments.

B. Predictor Model Prompts

In this section, we provide the detailed prompts used for the predictor models. We present the System Prompt and User Prompt sequentially for each setting.

B.1. Prediction with User Profile (Ours)

System Prompt

```
You are the author of the post.  The
section labeled "User Profile" is
your profile | it describes who you
are.

Read it carefully and fully adopt
this as your identity and mindset.

You will then be shown a post you
wrote, and a comment written in
response to it.  Based on your
profile, determine whether the
comment would change your mind from
the opinion expressed in the post.

Respond only with one word:  "yes"
if your mind would change after
reading the comment, or "no" if not.
Do not provide any explanation or
reasoning.
```

User Prompt

```
### User Profile
{user_profile}

### Post
{post}

### Comment
{comment}

---
Would this comment change your mind
from the opinion you expressed in
the post?

Respond only with one word:  "yes"
or "no".
```

B.2. Prediction with User History (Retrieval-Only)

System Prompt

```
You are the author of the post.  The
section labeled "User History" is
relevant past history about you.

Read it carefully and incorporate it
into your identity and mindset.

You will then be shown a post you
wrote, and a comment written in
response to it.  Based on your
history, determine whether the
comment would change your mind from
the opinion expressed in the post.

Respond only with one word:  "yes"
if your mind would change after
reading the comment, or "no" if not.
Do not provide any explanation or
reasoning.
```

User Prompt

```
### User History
{user_profile}

### Post
{post}

### Comment
{comment}

---
Would this comment change your mind
from the opinion you expressed in
the post?

Respond only with one word:  "yes"
or "no".
```

Table 8. Detailed statistics of the dataset splits. User History Count refers to the number of historical posts/comments available for the OP prior to the current post.

			User History Count				Delta Comments			Non-Delta Comments		
Split	**# Posts**	**Unique OPs**	**Min**	**Max**	**Mean**	**Median**	**Min**	**Mean**	**Median**	**Min**	**Mean**	**Median**
Train	1,341	1,257	15	11,965	252.40	57	1	1.77	1	1	33.06	20.0
Validation	167	69	16	19,583	956.35	65	1	2.24	1	1	31.81	19.0
Test	168	69	15	19,583	613.36	71	1	2.54	1.5	2	35.30	19.0

B.3. Prediction without Personalization

System Prompt

```
You are the author of the post.
Carefully read your own post and
the comment written in response to
it.

Decide whether you would change your
mind after reading the comment.

Ignore your own beliefs as a
language model and fully adopt the
mindset of the person who wrote the
post.

Respond with only one word:  "yes"
if you think you would change
your mind, or "no" if not.  Do
not provide any explanation or
reasoning.
```

User Prompt

```
[Post]
{post}

[Comment]
{comment}

Would you change your mind after
reading the comment?
```

C. Profiler Training Details

In this section, we provide detailed specifications for the preference construction process and the hyperparameters used for Direct Preference Optimization (DPO).

C.1. Preference Pair Construction

To derive robust training signals from the synthesized candidate profiles, we employ a margin-based stratified sampling strategy. As described in Section 3.3.1, for each input group $\mathcal{G}_i$, we generate a set of 16 candidate profiles $\Pi_{\mathcal{G}_i}$. We rank these profiles based on their utility score $S(\pi)$, which represents the macro-F1 score on the view-change prediction task.

To avoid noisy training signals arising from pairs with negligible performance differences, we enforce a minimum utility margin δ. We construct a dataset of preference pairs $\mathcal{D} = \{(x, \pi_w, \pi_l)\}$ where:

$$S(\pi_w) - S(\pi_l) \geq \delta \tag{1}$$

where x represents the input historical records. Specifically, we select the top-K performing profiles as positive samples and the bottom-K profiles as negative samples from the candidate set $\Pi_{\mathcal{G}_i}$. We then form pairs from the Cartesian product of these two subsets, filtering out any pairs that do not satisfy the margin condition. In our experiments, we set $K = 4$ (top 25% and bottom 25%) and the margin $\delta = 0.05$ to ensure distinct quality separation.

C.2. DPO Training Configuration

We optimize the profiler π_θ using the standard DPO objective, which increases the likelihood of the preferred profile π_w while decreasing that of the dispreferred profile π_l, implicitly optimizing the reward function without a separate reward model training step. The loss function is defined as:

$$\begin{aligned}\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = \\ -\mathbb{E}_{(x,\pi_w,\pi_l)\sim\mathcal{D}}\Big[\log\sigma\Big(\beta\log\frac{\pi_\theta(\pi_w|x)}{\pi_{\text{ref}}(\pi_w|x)} \\ -\beta\log\frac{\pi_\theta(\pi_l|x)}{\pi_{\text{ref}}(\pi_l|x)}\Big)\Big]\end{aligned} \tag{2}$$

where π_{ref} is the frozen reference model (the initial base profiler), σ is the logistic sigmoid function, and β is a hyperparameter controlling the deviation from the reference model.

We initialized the profiler with `Llama-3.1-8B-Instruct`. To ensure training stability and prevent overfitting to the small number of high-utility patterns, we utilized Low-Rank Adaptation (LoRA) for parameter-efficient fine-tuning. The detailed hyperparameters are listed in Table 9.

Table 9. DPO training hyperparameters for the profiler.

Hyperparameter	Value
Base Model	Llama-3.1-8B-Instruct
LoRA Rank (r)	32
LoRA Alpha (α)	64
Optimizer	AdamW
Learning Rate	5e-7
LR Scheduler	Linear
Warmup Ratio	0.05
Batch Size	64
Beta (β)	0.1
Epochs	3
Max Sequence Length	16384

C.3. Profile Generation Prompts

We use the following prompts to generate a context-aware user profile tailored for persuasion.

System Prompt

```
You are an expert assistant
whose task is to extract concise,
high-level information about the
author of a set of passages.

Focus only on traits that would
be most useful for persuading
or changing the author's view in
relation to the current post.

Your goal is to produce a compact,
context-aware user profile optimized
for persuasive messaging toward the
given post.
```

User Prompt

```
You are given a set of passages
written by the same author, along
with the author's current post.

Extract only the most essential
information about the author that
is clearly stated or strongly and
consistently implied across multiple
passages, focusing on traits that
are most relevant for understanding
how to persuade them in the context
of the current post.

Instructions:
  - Consider the current post as
the immediate context in which
persuasion would occur.
  - Identify attitudes, reasoning
patterns, or sensitivities that
could influence how the author might
respond to persuasion regarding the
post.
  - Do not guess or speculate beyond
what is well supported.
  - Exclude personally identifying
or sensitive details unless
explicitly stated.
  - Generalize from specific events
or examples into higher-level
traits; avoid direct quotes or
low-level details.
  - Remove redundancy and keep
bullets concise.
  - Do NOT respond with anything
other than the bullet points.

Current Post:
{post}

Input Passages:
{passages}

Output:
  • ...
  • ...
```

D. Query Generator Training Details

This appendix provides implementation details for training the query generator, including candidate generation, retrieval-based supervision, preference construction, and optimization settings. Across all experiments, the query generator is implemented as a single LLM (Llama-3.1-8B-Instruct) and trained using Direct Preference Optimization (DPO) with a two-stage training strategy, following Section 3.3.3.

D.1. Overview

The query generator is trained to produce *user-focused retrieval queries* that retrieve historical user records informative for personalized persuasion. A post-only query is often insufficient because the post may not explicitly mention persuasion-critical user attributes (e.g., values, experiences, decision-making styles). To make learning easier, we adopt a two-stage training strategy:

1. **Stage 1 (User-Focused Question Generation).** Prompt the model to produce a user-focused *question* that asks for user information *not present in the post* but likely to affect persuasion.
2. **Stage 2 (Post-Contextualized Query Generation).** Train the model to take the post and the Stage-1 question as input and generate a single *retrieval query* that contextualizes the user attribute using salient post cues (topic, stance, constraints).

In second stage, supervision is derived from retrieval quality: we score candidate queries by NDCG@5 based on the persuasion utility of retrieved records and apply DPO to prefer candidates with higher retrieval quality. At inference, the trained model receives only the post and outputs a user-focused retrieval query.

D.2. Candidate Generation Procedure

For each post x_i, we generate candidates as follows.

Stage 1: User-Focused Question. We first generate a single user-focused question $q_i^{(1)}$ from the query generator using the Stage-1 prompt with decoding temperature 0. This question serves as an intermediate representation of the user attribute to seek.

Stage 2: Post-Contextualized Retrieval Query. Conditioned on $(x_i, q_i^{(1)})$, we sample 16 candidate retrieval queries $\{q_{i,j}^{(2)}\}_{j=1}^{16}$ using the Stage-2 prompt with temperature 0.8. Each candidate is a single natural-language sentence that integrates (i) the user attribute targeted by $q_i^{(1)}$ and (ii) salient cues from x_i.

This two-step candidate generation is used to construct DPO training data and is applied consistently across all predictor models.

D.3. Retrieval and Scoring

Each Stage-2 candidate query $q_{i,j}^{(2)}$ is used to retrieve the top-5 user records from the author's historical records using a fixed embedding-based retriever (BGE-M3). We evaluate query quality using NDCG@5, where the graded relevance of each retrieved record is given by its pre-computed *record-level persuasion utility score* (Section 3.3.2).

Both the retriever and the utility scores are kept fixed throughout query generator training. Thus, the query generator is trained solely to improve retrieval quality under a fixed downstream evaluation signal.

D.4. Preference Pair Construction

For each post, we partition the 16 Stage-2 candidates into positive and negative pools based on their NDCG@5 scores, and construct preference pairs by pairing positives with negatives. We additionally enforce a minimum margin of 0.10 between the chosen and rejected query scores, and select up to a fixed maximum number of pairs per post.

Because the NDCG@5 score distributions differ across predictor models (due to predictor-specific persuasion utility scoring), we use predictor-specific thresholds and pair caps to ensure sufficient supervision. Table 10 summarizes the settings.

Table 10. Predictor-specific thresholds for preference pair construction based on NDCG@5.

Predictor	Pos.	Neg.	Max
Llama-3.1-8B-Instruct	≥ 0.65	≤ 0.55	8
Llama-3.3-70B-Instruct	≥ 0.75	≤ 0.65	8
GPT-4o-mini	≥ 0.55	≤ 0.45	10

Table 11. DPO training hyperparameters for the query generator.

Hyperparameter	Value
Base model	Llama-3.1-8B-Instruct
LoRA rank r	16
LoRA scaling α	32
Learning rate	2×10^{-5}
Max epochs	3
DPO β	0.3 (0.1 for GPT-4o-mini)

D.5. Optimization via DPO

We train the query generator using DPO with LoRA fine-tuning. All settings are shared across predictors except the DPO inverse temperature β, which we tune to account for differences in preference sharpness under each predictor's utility signal.

D.6. Query Generator Prompts

User-Focused Question Prompt.

System Prompt

```
You will be given an online post
where a user explains their view on
a specific topic.
Write ONE short question that asks
for information regarding the user
that is NOT explicitly stated in
the post, but would be important for
persuading the user expressed in the
post.
The question should focus on
aspects such as the user's values,
experiences, priorities, or decision
making styles related to the topic.

Instructions:

  • Output MUST be a single question
    sentence ending with ``?''.
  • Do NOT explain your reasoning.
  • Do NOT ask for information
    already provided in the post.
```

User Prompt

```
Post:
---
{post}
---
Respond in ONE question.
```

Post-Contextualized Query Prompt.

System Prompt

```
You will be given two inputs:
(1) an online post where a user
explains their view on a specific
topic.
(2) a question asking for
information that is NOT explicitly
stated in the post, but is important
for persuading the user in this
situation.

Write ONE sentence that
incorporates:

  • what the question is asking
    about the user
  • the most important cues from the
    post

The sentence should clearly reflect
what the question asks about the
user, while also grounding it in the
most important cues from the post.

Instructions:

  • Output MUST be a single
    sentence.
  • Do NOT explain your reasoning.
```

User Prompt

```
Post:
---
{post}
---
Question:
---
{question}
---
Respond in ONE sentence.
```

E. Examples of the Generated User Profiles

To provide qualitative insight into the behavior of our trained profiler, we present examples of generated user profiles at Figure 5.

Figure 6 compares profiles generated for the same user when trained with different predictor models. Despite being grounded in identical user records, the resulting profiles exhibit systematic differences in emphasis and framing, reflecting predictor-specific preferences for information that is most useful for downstream prediction. This comparison highlights that our profiler does learn to adapt profile construction tailored to the target predictor model.

- The author values practicality and efficiency in decision-making, prioritizing the enjoyment and benefit gained from an experience over the cost.
- The author is likely a rational and logical thinker, weighing the pros and cons of a situation before making a decision.
- The author may be skeptical of the idea of "getting one's money's worth" and instead focuses on the value gained from an experience.
- The author values personal enjoyment and satisfaction over material possessions or financial gain.
- The author may be open to re-evaluating past decisions and choices if they no longer align with their current values or interests.
- The author is likely resistant to the idea of doing something solely for the sake of "getting their money's worth" if it doesn't bring them enjoyment or benefit.

- The author values clarity and predictability in laws, seeking a system where laws are clearly defined and easily understandable.
- They believe in the importance of accountability, particularly for those in power, and want to reduce the potential for bias and corruption.
- The author is concerned about the potential for abuse of power and the need for checks and balances.
- They are skeptical of the current system's ability to hold those in power accountable, particularly when it comes to lawmakers and the government.
- The author is interested in exploring alternative solutions to address issues such as corruption and conflict of interest.
- They value the idea of a more autonomous system, where the court system is reserved for complex or unclear cases.
- The author is open to considering the possibility of starting a court case for an unrelated crime, especially if there is strong evidence and the authorities are refusing to take action.
- They are likely to be influenced by logical and evidence-based arguments, particularly those that highlight the potential flaws in the current system

- Values self-restraint and responsibility, particularly in the context of societal well-being.
- Recognizes the importance of considering the potential consequences of individual actions on others.
- Is skeptical of simplistic or absolute advice, such as "be yourself," and prefers nuanced thinking.
- Understands that personal flaws and imperfections can have negative impacts on others.
- Is willing to acknowledge and work on personal limitations, rather than pretending to be perfect.
- May be concerned with the potential negative outcomes of certain actions, such as rape and murder, and sees these as self-damning.
- May be open to considering the potential consequences of individual actions on the greater good.
- May be interested in exploring the complexities of human behavior and the factors that influence it.
- May be critical of absolute or dogmatic thinking.

Figure 5. Examples of user profiles generated by our trained profiler when used `Llama3.1 8B Instruct` as the predictor model for training.

F. Details of User Profiling Baselines

We compare our approach against several representative user profiling frameworks. Specifically, we consider: (i) PAG (Richardson et al., 2023), which retrieves a subset of user records using BM25, independently summarizes each selected record, and concatenates the resulting summaries into a user profile; (ii) HSUMM (Zhong et al., 2024), which applies hierarchical summarization by first generating summaries over subsets of user records and then aggregating these intermediate summaries into a single profile; (iii) RECURSUMM (Wang et al., 2025), which incrementally updates the user profile by recursively integrating each new user record with the existing summary. PAG relies on retrieval to select a subset of user records for profiling, whereas HSUMM, RECURSUMM, and PRIME assume access to and utilize all available user historical records when constructing user profiles. Concretely, HSUMM and RE-

Llama 3.1 8B Instruct

- Values self-restraint and responsible behavior, particularly in the context of societal impact.
- Believes that individual flaws and impulses can have negative consequences for others.
- Prioritizes the greater good and the well-being of society over personal desires or impulses.
- Is skeptical of simplistic or overly idealistic advice, such as "be yourself," when it may lead to negative outcomes.
- Recognizes the importance of personal growth and self-improvement, as evident in the desire to "become a good citizen."
- May be concerned with the potential consequences of unchecked individual behavior on the social fabric.
- May be open to nuanced and context-dependent advice, rather than one-size-fits-all solutions.

Llama 3.3 70B Instruct

- Values self-restraint and responsibility, particularly in the context of societal well-being.
- Recognizes the importance of considering the potential consequences of individual actions on others.
- Is skeptical of simplistic or absolute advice, such as \"be yourself,\" and prefers nuanced thinking.
- Understands that personal flaws and imperfections can have negative impacts on others.
- Is willing to acknowledge and work on personal limitations, rather than pretending to be perfect.
- May be concerned with the potential negative outcomes of certain actions, such as rape and murder, and sees these as self-damning.
- May be open to considering the potential consequences of individual actions on the greater good.
- May be interested in exploring the complexities of human behavior and the factors that influence it.
- May be critical of absolute or dogmatic thinking.

GPT 4o mini

- The author is likely open to critical thinking and nuanced perspectives, as they acknowledge the limitations of a simplistic "be yourself" approach and consider various counterexamples.
- They value pragmatism and the greater good, as they prioritize becoming a "good citizen" over personal flaws and their potential consequences.
- The author is likely skeptical of absolute moral or philosophical principles, as they question the idea that people would behave poorly without a higher power.
- They may be concerned with social responsibility and the impact of individual actions on society, as they consider the potential consequences of unrestrained behavior.
- The author is willing to admit their own flaws and imperfections, indicating a level of self-awareness and humility.
- They may be interested in exploring the complexities of human nature and the role of societal factors in shaping individual behavior.
- The author is likely open to considering multiple perspectives and evaluating evidence, as they engage in a debate and provide counterexamples to a given argument.

Figure 6. Examples of user profiles generated by our trained profilers for the same user under different predictor models used for training.

CURSUMM construct a profile for each user by repeatedly summarizing that user's entire history through multiple summarization steps.

All baselines were minimally adapted with task-specific instructions to align them with the persuasion objective, while keeping their core methods unchanged. Table 12 provides representative examples of the modifications applied to each baseline.

G. Additional Experiment Results

G.1. Comparison with Long-Context Direct Conditioning

To examine whether the proposed retrieval-and-profiling pipeline provides advantages beyond what long-context models can achieve through direct conditioning, we conduct additional experiments feeding raw user history directly to the predictor without any retrieval or profiling. Specifically, the latest k records are included for each user ($k = 10, 50, 100, 500$).

Two observations emerge consistently across models. First, directly feeding raw history performs substantially worse than our profiling-based approach across all settings, confirming that the benefit of our framework lies in the structured, persuasion-aware transformation of user history rather than simply providing more context. Second, performance does not improve monotonically with the number of records and in some cases degrades as more records are added, suggesting that raw user history contains substantial noise and that indiscriminately feeding more records makes it harder for the predictor to identify persuasion-relevant signals.

We attribute this to a fundamental difference in what each approach does with user history. Long-context direct conditioning effectively performs episodic memory retrieval—surfacing specific facts or events from a user's past—and leaves it to the predictor to extract useful signals from noisy, unstructured input. In contrast, our framework goes beyond surface-level fact retrieval to **infer deeper user characteristics** such as values, beliefs, and reasoning styles, constructing a compact representation of the user's underlying disposition toward persuasion. This distinction is particularly important for tasks involving human judgment and attitude change, where responses are driven not by isolated past events but by stable, abstract traits that must be inferred from behavioral traces. The performance degradation observed with increasing history size further supports this view: more raw context does not help if the model cannot effectively abstract the persuasion-relevant signal from it.

Beyond prediction quality, directly conditioning on large user histories is also **highly inefficient** from an inference perspective, as the number of input tokens grows substantially with history size, resulting in significantly higher computational cost. Together, these results demonstrate that long-context direct conditioning is neither an effective nor a practical alternative to our retrieval-and-profiling pipeline.

G.2. Effect of Thread Context

To examine the role of multi-turn discussion dynamics relative to persistent user characteristics, we conduct additional experiments on a test subset consisting of comments embedded in multi-turn exchanges. We compare three conditions

Table 12. Representative prompt modifications for each profiling baseline. Original instructions are adapted to the persuasion prediction task while preserving each method's core mechanism.

Baseline	Original Instruction	Revised Instruction
PAG	Write a summary, in English, of the research interests and topics of a researcher who has published the following papers.	Write a summary of the most essential information about the author that would be useful for persuading or changing the author's view.
HSUMM	System: Below is a transcript of a conversation between a human and an AI assistant that is intelligent and knowledgeable in psychology. User: Hello! Please help me summarize the content of the conversation. System: Sure, I will do my best to assist you.	System: You are an expert assistant whose task is to extract information about the author from a set of passages. Your goal is to produce a compact, context-aware user profile optimized for changing the author's view.
RECURSUMM	Remember, the memory should serve as a reference point to maintain continuity in the dialogue and help you respond accurately to the user based on their personality.	Remember, the memory should serve as a reference point to produce a compact, context-aware user profile optimized for persuasive messaging toward the given post.

Table 13. Prediction performance (F1) when directly conditioning on the latest k user records, compared with our retrieval-and-profiling pipeline (**Ours**).

Model	Setting	F1
`Llama-3.3-70B`	$k = 10$	0.3698
	$k = 50$	0.2920
	$k = 100$	0.2938
	$k = 500$	0.2611
	Ours	**0.4661**
`Llama-3.1-8B`	$k = 10$	0.2411
	$k = 50$	0.2257
	$k = 100$	0.2629
	$k = 500$	0.2621
	Ours	**0.4000**
`GPT-4o-mini`	$k = 10$	0.1344
	$k = 50$	0.1126
	$k = 100$	0.1383
	$k = 500$	0.1038
	Ours	**0.2787**

under which the predictor determines whether the target comment receives a delta: (1) USER PROFILE only, using profiles generated by our proposed framework; (2) THREAD CONTEXT only, where we concatenate the turns prior to the target comment in the thread and provide them as additional context to the predictor alongside the target comment; and (3) PROFILE + THREAD CONTEXT, combining both.

Table 14 reports comment-level view-change prediction F1 across the three predictor models. Two findings emerge consistently. First, user profile alone outperforms thread context alone across all predictors, indicating that modeling the persuadee's persistent characteristics is more informative than thread-level dynamics for this task. Second, combining user profile with thread context improves over thread context alone, suggesting that the two sources of information are complementary, though the profile remains the dominant signal.

Table 14. Comment-level view-change prediction F1 under three conditioning settings: user profile only, thread context only, and their combination. GPT, LLaMA-8B, and LLaMA-70B denote `GPT-4o-mini`, `Llama-3.1-8B-Instruct`, and `Llama-3.3-70B-Instruct`, respectively.

Setting	GPT	LLaMA-8B	LLaMA-70B
User Profile (Ours)	0.2054	0.3612	0.3755
Thread Context	0.1437	0.2388	0.2680
Profile + Thread Context	0.2210	0.3186	0.3612

We note that our main experiments deliberately adopt a single-turn formulation restricted to top-level comments in order to isolate the contribution of user profiling as clearly as possible. By pairing each top-level comment with its corresponding original post, we disentangle the effect of the user's persistent characteristics from the contingent dynamics of a specific thread, providing a controlled setting in which the role of user profiling can be directly evaluated. Specifically, positive instances are top-level comments that directly received a delta, while negative instances are top-level comments from threads in which a delta was never awarded throughout the entire discussion. Extending the framework to incorporate multi-turn dynamics is a promising direction for future work.

G.3. Effect of Stronger Predictor Models

To examine whether the benefit of our framework persists under stronger predictor models, we conduct an additional experiment using GPT-5 as the predictor. Due to the prohibitive cost of retraining all pipeline components from scratch with a stronger model (approximately $10\times$ the cost of GPT-4o-mini), we adopt a transfer setting in which the query generator and profiler trained under GPT-4o-mini are kept fixed, and only the predictor is replaced with GPT-5.

Table 15. View-change prediction F1 with GPT-5 as the predictor. The query generator and profiler are those trained under GPT-4o-mini and kept fixed; only the predictor is replaced.

Method	F1
No Personalization	0.3719
Retrieval-only	0.1684
User Profile (Ours)	**0.3791**

Table 15 reports F1 scores under three conditions: no personalization, retrieval-only, and our full framework. The results show a trend consistent with GPT-4o-mini: no personalization outperforms retrieval-only, yet applying our user profiles provides additional benefit over no personalization. This indicates that our profiler remains effective even under a stronger predictor.

We note that this experiment is not a fully controlled comparison: the profiler and query generator were optimized under GPT-4o-mini's utility signal, and GPT-5 may prefer different records and profile dimensions. Since our framework derives training supervision directly from the target predictor's feedback, retraining the full pipeline with GPT-5 as the predictor would likely yield larger gains. We therefore view the current result as a conservative lower bound, and expect the performance gap to widen when all components are trained end-to-end with a stronger predictor.

G.4. Ablation on Repetition Count

Because the predictor-side computation scales directly with the number of grouping repetitions m, we ablate m to characterize the trade-off between additional compute and the stability of the training signal derived from the utility scores.

Compute scaling. As expected, the overall token usage increases near-linearly with m: $m = 5$ requires approximately 2.3B tokens (2,338,263,616), corresponding to $1.67\times$ the compute of $m = 3$; $m = 10$ requires approximately 4.6B tokens (4,682,146,776), corresponding to $3.34\times$ the compute of $m = 3$.

Compute-stability trade-off. The query generator is trained with DPO using preference pairs of candidate queries, where the "chosen" and "rejected" labels are derived from their NDCG@5 scores (computed by using record utilities as graded relevance). For our analysis, we bucket candidate queries into chosen vs. rejected pools using the same thresholds used in our experiments (chosen if NDCG@5 ≥ 0.55, rejected if ≤ 0.45), and measure stability via a **flip rate**: among all candidates that are labeled chosen or rejected under $m = 3$, the fraction of candidates whose label changes when utility scores are recomputed with larger m. The flip rates are 5.91% for $m = 3 \rightarrow 5$ and 8.05% for $m = 3 \rightarrow 10$.

Table 16. Additional retrieval-side results under different predictor-specific persuasion utility signals. All methods are evaluated on the same record pool as the main experiment. Random reports the average over 10 runs.

	Llama-3.3-70B-Instruct		GPT-4o-mini	
Query Strategy	**NCG@5**	**NDCG@5**	**NCG@5**	**NDCG@5**
Random	0.7461	0.7395	0.4713	0.4647
BGE-Post	0.7461	0.7357	0.4826	0.4736
HyDE	0.7528	**0.7482**	0.4685	0.4562
Ours	**0.7536**	0.7471	**0.4827**	**0.4747**

Considering that two candidates constitute a preference sample, this means that only 4.025% of preference samples (at most) have their original preference direction reversed as repetitions increase from 3 to 10, while $\geq 92\%$ of preference/rejection labels used for DPO are preserved. Given that moving from $m = 3$ to $m = 10$ costs $> 3\times$ more compute, these results indicate that $m = 3$ is a compute-efficient choice that retains the vast majority of the training signal.

G.5. Additional Retrieval Results

Table 16 reports retrieval performance under different predictor-specific persuasion utility signals, using the same candidate record pool and query strategies as in the main experiment (Table 1).

G.6. AUC scores across different retrieval and profiling variants

Table 17 reports the full end-to-end AUC results across different retriever and profiler combinations for each predictor model.

H. Details of Profiler Analysis

H.1. Additional Results

Figure 7 extends the main-text analysis by reporting F1 scores by topic and claim type for `GPT-4o-mini` and `Llama-3.3-70B-Instruct`, comparing the original and trained profilers.

Figures 8 and 9 present the profile-dimension analysis results for `GPT-4o-mini` and `Llama-3.3-70B-Instruct`, respectively.

H.2. Failure Case Analysis

To better understand failure cases, we conduct an additional analysis analogous to Figure 4b, focusing on instances where ΔF1 < 0 (i.e., the trained profile underperforms the original). Using the same procedure as in the profiler analysis (Section 5.1), we compute delta-dimension frequency for performance-degrading profiles,

Table 17. Effect of retriever and profiler choices on view-change prediction under different predictors (AUC). Random reports the average performance over 10 runs. Underlined results denote our final proposed method, while **boldface** highlights the best-performing configuration within each column. Column groups correspond to different predictor models, with sub-columns indicating profiler configurations (demographic, base profiler, and our trained profiler).

	Llama-3.1-8B-Instruct			Llama-3.3-70B-Instruct			GPT-4o-mini		
Retrieval	Demograph.	Base	*Ours*	Demograph.	Base	*Ours*	Demograph.	Base	*Ours*
Recent	0.5828	0.5953	0.6088	0.6577	0.6740	0.6746	0.6214	0.6214	0.6232
Random	**0.5859**	**0.6121**	0.6112	0.6528	0.6669	0.6716	0.6188	0.6121	0.6253
BM25	0.5858	0.5955	0.6082	0.6564	0.6588	0.6697	0.6159	0.6189	0.6365
BGE	0.5851	0.5859	0.6029	**0.6596**	**0.6768**	0.6798	0.6226	0.6305	0.6349
HyDE	0.5845	0.5997	0.6104	0.6569	0.6655	0.6825	0.6216	**0.6311**	**0.6447**
Ours	0.5850	0.6054	**0.6146**	0.6574	0.6736	**0.6828**	**0.6232**	0.6020	0.6299

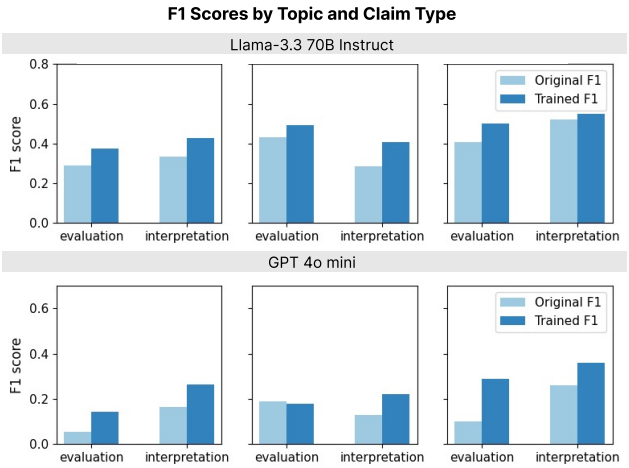

Figure 7. F1 by topic and claim type of the post, comparing the original and trained profilers.

using `Llama-3.1-8B-Instruct` as the predictor. Table 18 reports the delta-dimension frequency for these cases. The results show that the shifts are post-type dependent: **no single dimension is uniformly over- or under-produced across all post types**, indicating that failures are not driven by a global bias toward any particular dimension. To interpret these shifts, Table 19 reproduces the Pearson correlation between delta-dimension frequency and performance change. As is largely expected, the results indicate that failures arise from misweighting profile dimensions: in contrast to performance-improving profiles, they do not sufficiently increase dimensions positively correlated with gains for a given post type and instead overrepresent dimensions negatively correlated with performance. A concrete example appears in Others–Interpretation posts. In this category, *Interest & Knowledge* (+0.39) and *Personality Traits* (+0.52) are strongly positively correlated with performance gains (Table 19). However, performance-degrading profiles contain on average approximately two fewer *Interest & Knowledge* items and one fewer *Personality Traits* item than the original profiles (Table 18), indicating that under-

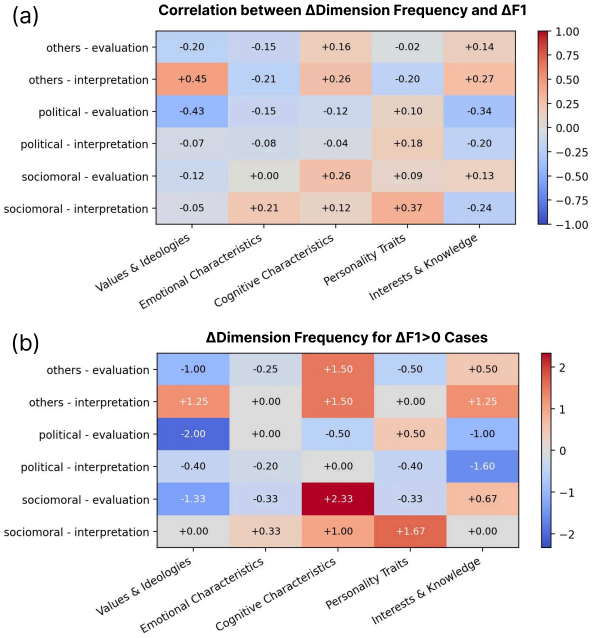

Figure 8. Analysis of profile-dimension frequency shifts (ΔDF) and performance gains (ΔF1) between the original and trained profilers. (a) Correlation between ΔDF and ΔF1. (b) ΔDF for cases with $\Delta F1 > 0$. `GPT-4o-mini` is used as the predictor.

representation of beneficial dimensions contributes to the performance drop.

Table 18. Delta-dimension frequency (ΔDF) for performance-decreasing cases ($\Delta F1 < 0$). Column abbreviations: Val = Values & Ideologies; Emo = Emotional Characteristics; Cog = Cognitive Characteristics; Pers = Personality Traits; Int = Interests & Knowledge.

Task	Val	Emo	Cog	Pers	Int
Others-Eval	+0.14	+0.00	+0.29	+0.71	−0.57
Others-Interp	+1.00	+0.00	+0.00	−1.00	−2.00
Political-Eval	+0.00	+0.33	+1.00	+0.67	+0.67
Political-Interp	+0.14	+0.57	+0.29	−0.29	+1.00
Sociomoral-Eval	+0.50	+0.40	−0.50	−0.70	+0.70
Sociomoral-Interp	+0.33	−1.00	+1.00	−0.67	+0.67

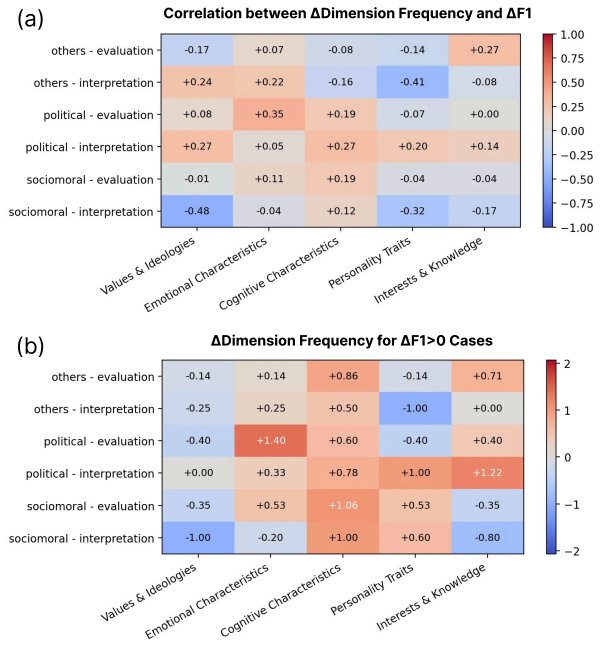

Figure 9. Analysis of profile-dimension frequency shifts (ΔDF) and performance gains (ΔF1) between the original and trained profilers. (a) Correlation between ΔDF and ΔF1. (b) ΔDF for cases with ΔF1 > 0. `Llama-3.3-70B-Instruct` is used as the predictor.

Table 19. Pearson correlation between ΔDF and ΔF1. Abbreviations follow Table 18.

Task	Val	Emo	Cog	Pers	Int
Others-Eval	+0.28	+0.11	+0.05	−0.32	+0.27
Others-Interp	−0.17	−0.26	−0.01	+0.52	+0.39
Political-Eval	−0.14	−0.20	−0.29	+0.13	−0.18
Political-Interp	−0.13	−0.32	+0.02	+0.21	−0.04
Sociomoral-Eval	−0.12	+0.00	+0.22	+0.10	−0.10
Sociomoral-Interp	−0.55	+0.13	+0.17	+0.33	−0.29

I. Details of User Record Analysis

I.1. Additional Results

Figure 10 analyzes the topical and claim type characteristics of user records ranked by utility score. Top-ranked records tend to align more closely with the post in both topic and claim type, whereas bottom-ranked records show weaker alignment.

Table 20 reports mean F1 over all records as well as the top-5 and bottom-5 subsets, showing that while `Llama-3.3-70B-Instruct` assigns higher scores overall, `GPT-4o-mini` exhibits a larger contrast between high- and low-ranked records. This trend is further reflected in Table 21, where `GPT-4o-mini` yields consistently larger margins between top-5 and bottom-5 records.

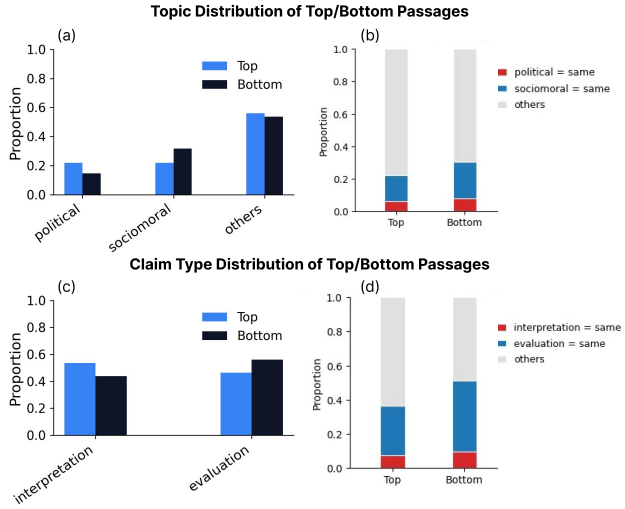

Figure 10. Topic and claim type distributions of top- and bottom-ranked records. (a,c) Marginal distributions by topic and claim type. (b,d) Proportions of records sharing the same topic or claim type with the post.

Table 20. Mean F1 scores across all records, top-5 records, and bottom-5 records. While `Llama-3.3-70B-Instruct` assigns higher absolute scores overall, `GPT-4o-mini` exhibits a larger contrast between top-5 and bottom-5 records.

Model	All	Top-5	Bottom-5
GPT-4o-mini	0.163	0.302	0.092
Llama-3.3-70B-Instruct	**0.375**	**0.468**	**0.301**
Llama-3.1-8B-Instruct	0.248	0.321	0.183

J. Details of Efficiency Analysis

This appendix provides per-stage breakdowns and measurement details for the efficiency analysis reported in Section 5.3.

J.1. Measurement Setup

We measure per-instance inference cost under a realistic deployment scenario in which user history embeddings are maintained offline, so that only newly added records trigger re-encoding at inference time. Tokens are counted directly from the actual prompts and completions consumed by each LLM call in the pipeline. FLOPs are estimated from token counts using the standard approximation $C \approx 2NT$, where N is the number of model parameters and T is the number of tokens processed. All measurements are averaged over the test set with `Llama-3.1-8B-Instruct` as the predictor, profiler, and query generator.

J.2. Per-Stage Token Usage and FLOPs

Tables 22 and 23 decompose per-instance cost into three stages: search (query formulation and retrieval call), pro-

Table 21. Distribution of margin between high- and low-scoring records (min top-5 minus max bottom-5). `GPT-4o-mini` exhibits consistently larger margins, indicating clearer separation between beneficial and non-beneficial records.

Model	Mean	Median (p50)	p90
GPT-4o-mini	**0.180**	**0.133**	**0.500**
Llama3.3-70B-Instruct	0.135	0.090	0.337
Llama-3.1-8B-Instruct	0.112	0.080	0.260

filing (profile construction from retrieved or full history), and prediction (final view-change prediction). Search tokens correspond to the query-generation step in our method and to BM25 query formulation in PAG. HSUMM and RECURSUMM operate directly on full user history without a separate retrieval query, so their search cost is zero. Profiling dominates the cost of full-history baselines, while retrieval-based methods distribute cost more evenly across query generation, profiling, and prediction.

Table 22. Per-instance token usage across stages.

Method	Search	Profile	Predict	Total
PAG	294	1,464	1,470	3,228
Ours	916	1,479	1,034	3,429
RECURSUMM	0	20,750	928	21,678
HSUMM	0	41,816	1,064	42,880

Table 23. Per-instance FLOPs across stages.

Method	Search	Profile	Predict	Total
PAG	1.2×10^{6}	2.30×10^{13}	2.41×10^{13}	4.71×10^{13}
Ours	9.97×10^{12}	2.41×10^{13}	1.68×10^{13}	5.09×10^{13}
RECURSUMM	0	3.43×10^{14}	1.50×10^{13}	3.58×10^{14}
HSUMM	0	6.86×10^{14}	1.73×10^{13}	7.03×10^{14}

J.3. Training Cost of Persuasion Utility Scoring

Record-level persuasion utility scoring (Section 3.3.2) is performed once during training and requires repeated profile generation and prediction over randomly grouped user records. Under our default setting of three repetitions with `GPT-4o-mini` as the predictor, the total token consumption on the training set is approximately 362M profiler tokens and 1.04B predictor tokens, totaling 1.40B tokens. Predictor-side computation dominates (74.2% of total), reflecting that each sampled profile is evaluated against multiple comments for the associated post. This cost is incurred only once during training; at inference time, no utility scoring is needed, and the framework applies to new users without additional training.