Diversity Augmentation of Dynamic User Preference Data for Boosting Personalized Text Summarizers

Anonymous authors
Paper under double-blind review

Abstract

Document summarization facilitates efficient identification and assimilation of user-relevant content, a process inherently influenced by individual subjectivity. Discerning *subjective* salient information within a document, particularly when it has multiple facets, poses significant challenges. This complexity underscores the necessity for *personalized summarization*. However, training models for personalized summarization has so far been challenging, particularly because diverse training data containing both user preference history (i.e., *click-skip* trajectory) and expected (gold-reference) summaries are scarce. The MS/CAS PENS dataset is a rare resource in this direction. However, the training data only contains preference history *without any target summaries, thereby blocking end-to-end supervised learning*. Also, the diversity in terms of topic transitions along the trajectory is relatively low, thereby leaving scope for better generalization. To address this, we first introduce a novel user preference data diversity evaluation metric, called `DegreeD`. We then propose `PerAugy`, a novel cross-trajectory shuffling and summary-content perturbation-based data augmentation technique that increases the `DegreeD`-score and thereby, significantly boosts the accuracy of four state-of-the-art (SOTA) baseline user-encoders commonly used in personalized summarization frameworks (best result: 0.132↑ w.r.t AUC). We select two such SOTA summarizer frameworks as baselines and observe that when augmented with their corresponding improved user-encoders, they consistently show an increase in personalization (avg. boost: 61.2% ↑ w.r.t. PSE-SU4 metric). This further establishes the efficacy of `PerAugy` as an augmentation method to boost personalized summarizers.

1 Introduction

The rapid increase in information requires efficient summarizers for fast comprehension and prioritization Ter Hoeve et al. (2022). However, identifying "salient" information is subjective, particularly in multi-aspect documents, which makes *personalized summarization* critical. Training such models requires datasets with diverse user histories and subjective summaries. However, such datasets are scarce due to privacy concerns Kirk et al. (2024); Liu et al. (2024).

The MS/CAS PENS dataset Ao et al. (2021), sourced from the MIND dataset Wu et al. (2020), is a valuable and the only resource suitable for training and evaluating state-of-the-art (SOTA) personalized summarization models. The training dataset consists of user preference histories (i.e., temporal sequences of *click* and *skip* interactions with news articles), while the test dataset comprises preference histories with user-specific gold-reference (i.e., expected) summaries. The dataset has been widely adopted as the standard benchmark for SOTA personalised summarizers Ao et al. (2021); Song et al. (2023). However, the evaluation focus was solely on their accuracy and, more recently, on their *degree-of-personalization* w.r.t the novel (and, to the best of our knowledge, the only known) personalization metric, PerSEval Dasgupta et al. (2024), overlooking rigorous analysis of the effectiveness of PENS-like datasets as training data. In this paper, we propose `DegreeD` (**Degree**-of-**D**iversity) – a novel metric for quantitatively measuring the dataset diversity in terms of thematic evolution of user's interest and expectations over time[1]. Low `DegreeD` score would suggest that the

[1] Training dataset diversity behaves as regularizer during training/finetuning Hu et al. (2019); Fabbri et al. (2021).

train dataset will not support model generalization very well. This emphasizes the need for diversity-boosting data augmentation of PENS-like datasets.

To address the diversity challenge, we propose **PerAugy** – a perturbation-based augmentation of historical user preference data for personalized summarization. A seed preference dataset is first modeled as a **User Interaction Graph** (**UIG**). The nodes of a UIG represent users' clicked and skipped documents (**d-nodes**) as well as their generated/expected summaries (**s-nodes**). A path, called **trajectory**, of the UIG starts with a specific user node (**u-nodes**) and terminates in d/s-nodes. The edges denote *click*, *skip*, *generate-summary*, and *click-summary* actions. The UIG, hence, can be seen as a pool of user trajectories representing dynamic user behavior histories. We apply PerAugy on this UIG.

PerAugy is a cross-trajectory augmentation technique in which we propose a novel controlled both-ways exchange of user trajectory segments (termed **Double Shuffling** (**DS**)) between sampled copies of the original trajectories to create a new pool of diverse synthetic user trajectories. There are two primary controls in DS – (i) the **gap-length** that determines how much the new trajectory should resemble the original seed, and (ii) the **trajectory-length** that determines the number of diverse profiles to be synthesized. The DS operation mimics the stochastic diffusion of a user's interest into diverse themes, reflecting naturalistic behavior. A perturbation operation is then applied to the exchanged s-nodes to eliminate unnatural thematic jitters at the boundaries of two segments on the new synthetic trajectory, enhancing realistic nature of the synthetic trajectories. The content of every exchanged s-node is replaced by its corresponding d-node content that most closely matches the nearest prior nodes of the new synthetic trajectory. The substitution's influence diminishes over a k-step context window that represents what is to be considered within proximity. Since this perturbation follows a k-order Markov Chain, we term this as **Stochastic Markovian Perturbation** (**SMP**). Through the DS and SMP operations, PerAugy increases the diversity (w.r.t DegreeD) of preference datasets. **Upon acceptance, we will release GitHub link for the codebase and datasets.**

As mentioned earlier, most user preference datasets do not contain historical preference trajectories of users. However, several of them, such as the OpenAI (Reddit) and OpenAI (CNN-DailyMail) datasets Völske et al. (2017), contain human-annotated ratings (i.e., evaluation) of model-generated summaries w.r.t. varying granularity of quality (e.g., faithfulness, conciseness, completeness, fluency, etc.). In this paper, we propose a novel methodology to appropriate such OpenAI (Reddit)-like datasets into base synthetic UIGs that form a pool of base trajectories. Base trajectories consist of *clicks* (documents rated with high confidence), *skips* (documents rated with low confidence), and <*generate-summary*, *click-summary*> pairs where the highest-rated summaries are selected as gold references. PerAugy (DS + SMP) is then subsequently applied on this base UIG. This way, PerAugy proves to be a reliable and versatile tool for synthetic datasets generation in domains like OpenAI (Reddit) Völske et al. (2017), which lack PENS-like resources.

Summary of the Results. We evaluate PerAugy across three core dimensions: (i) its impact on user representation diversity (DegreeD) in the original datasets post-augmentation, (ii) improvements in accuracy for SOTA user-encoder models trained on PerAugy-augmented data, and (iii) downstream gains in personalized summarization frameworks. We find a strong correlation between enhanced DegreeD and user-encoder performance, with Pearson r: 0.68, Spearman's ρ: 0.73, and Kendall's τ: 0.57. Augmented data consistently improves models like NAML, EBNR, and NRMS, showing average gains of 24%, 25%, and 18% across AUC, MRR, and nDCG@5&10, respectively. In downstream task of personalized summarization, GTP and PENS frameworks show an average improvement of 61.2%↑ in PSE-SU4, with setups like PENS+NRMS+T2 reaching up to 75% in PSE-RG-SU4. Additionally, PerAugy generalizes effectively to low-resource domains (e.g., OpenAI Reddit), yielding consistent encoder gains of 19%, 25%, and 17% across accuracy metrics. These results collectively demonstrate PerAugy's effectiveness in enhancing data diversity, improving user modeling, and boosting downstream personalization quality.

2 Background

2.1 Dynamic User Preference (vs. Static User Persona)

It is crucial to distinguish between user persona and user-preference history in the context of preference datasets. Persona information, such as address, nationality, or broad interests like genres, tends to remain *relatively static over time.* In contrast, preference histories are *highly dynamic*, since they constitute interaction (or reading) behavior as a temporal sequence that is complex, and spans across multiple subtopics and discourses. A user is unlikely to display consistent behavioral repetition; for instance, it is improbable that Alice's weekly reading consistently centers only on European soccer highlights while predictably skipping U.S. politics or film updates. This distinction reinforces the need for training datasets that consist of dynamic user preferences rather than static persona features.

2.2 Personalized Summarizers

Most research on personalized summarization assumes a static user persona (i.e., user profile information that is relatively time-invariant). These works leverage the simplicity of guided (or controlled) summarization. In this direction, Dou et al. (2021) proposed *GSUM*, where the goal was to inject a *generic guidance* in terms of *explicit* user-provided key-phrases that are restricted to the query-document only and do not account for the dynamic shift in user preference. *CTRLSum* and *TMWIN* were also proposed on similar lines, where either static control signals were given explicitly or extracted from dialogue sessions He et al. (2022); Kirstein et al. (2024). Xiao et al. (2024b) proposed the *Tri-Agent* personalized summarizer that was iteratively trained under an RL setup using an *oracle-as-an-instructor* that knows historical user-edits of previous summaries. However, the user-edit-preference does not entail subjectivity and is also static. Static user preference is unrealistic in most situations, while a shift in topics of interest is the norm.

In the more realistic context of dynamic user preference, *personalized summarization* refers to the extent to which a summarization model aligns its outputs with a reader's subjective expectations. The subjectivity is a function of the user's characteristic shift in preference as reflected through the *reading history* – a temporal trajectory of the reading and skipping actions of the user on a sequence of documents. It is important to note that this trajectory may occasionally be interleaved by the actions of generating and reading summaries instead of the full-length documents. The PENS framework (with external user-encoders such as NRMS, NAML, EBNR Wu et al. (2019b;a); Okura et al. (2017)) is an early example that attempts to address this Ao et al. (2021). The plugged user-encoders embed the user behavior trajectories from the PENS dataset. However, the encoders do not capture the dynamic temporal behavioral trend and are also tightly-coupled with the three injection techniques (T-1/2/3) of the encoder-decoder-based pointer-generator summarizer. Song et al. (2023) proposed the *GTP* framework that follows a similar summary-editing approach as *Tri-Agent*, except there is no explicit static guidance but rather the editing (latent) control is generated from the user trajectory. However, the internal user encoder, TrRMIo does not encode the dynamically shifting user trajectory without differentiating short vs. long-term influences. Also, so far we have not found any work that explicitly differentiates the various semantics of the user actions – *click, skip, read-summary.*

2.3 Personalized Summarization Datasets

A key challenge in the task of personalized summarization is the lack of suitable training data across varied domains that covers the three key conditions: (i) chronological ordering of *evolving* user actions (clicks, skips, etc.) – i.e., historical trajectories, (ii) subjective summary expectations (i.e. gold references/ratings) for same document by multiple users, and (iii) diversity and dynamicity w.r.t topics and topic transition. Standard summarization datasets like CNN/DM, Multinews Hermann et al. (2015); Fabbri et al. (2019) do not qualify. Only a few real-world datasets, notably PENS Ao et al. (2021) and PersonalSum Zhang et al. (2024), meet these criteria[2].

2.4 Personalized Summarization Evaluation

Vansh et al. (2023) introduced the notion of *degree-of-personalization* as a measure of the *subjective* user experience (UX), which is inversely related to both information overload and lack of expected information. They proved, arguably for the first time, that *accuracy metrics are unsuitable for measuring UX*, i.e., there are real-world cases where we find low UX even with high accuracy. As a solution, they proposed EGISES as

[2] PersonalSum was skipped because of insufficient samples and it being in Norwegian, makes it infeasible to test performance boost of summarizers that are not pre-trained in Norwegian.

UIG Symbols	
$\text{UIG} = \langle N, E \rangle$	Directed acyclic graph of user-document-summary interactions.
$u_j^{(t_0)}$	User node j at time t_0.
$d^{(t_p)}$	Document node at time t_p.
$s_j^{(t_q)}$	Summary node for user j at time t_q.
τ_{u_j}	User j's interaction trajectory.
$\mathcal{T}$	Trajectory pool dataset.
DegreeD Symbols	
$\delta[X]_d$	Divergence between consecutive documents.
$\delta[X]_s^j$	Divergence between consecutive summaries for user j.
$\text{DePS}^j_{\Delta(t_i, t_{i+1})}$	Preference shift ratio over time interval $\Delta(t_i, t_{i+1})$.
$E_j[\text{DePS}^{\mathcal{P}}]$	Penalized average preference shift for user j.
$\text{DegreeD}(D)$	Overall diversity score for dataset D: $\frac{\alpha}{\lvert U \rvert} \sum_j \delta[X]_s^j \cdot E_j[\text{DePS}^{\mathcal{P}}]$.
PerAugy Symbols	
$\mathcal{T}^{\text{syn}}$	Synthetically created trajectory pool by incorporating s-nodes.
$\tau_{u_j,\,\text{target}}$	Trajectory selected for augmentation.
$\tau_{u_i,\,\text{seg}}$	Segment extracted from a source trajectory.
g_l	Gap length between inserted segments.
$\tau^{u_j}_{\text{DS}}$	Final trajectory after double shuffling.
$\tau^{u_j}_{\text{SMP}}$	Final trajectory after SMP.
k	Context window size for computing influence.
λ	Exponential decay constant for context weighting.
p_{SMP}	Probability of perturbing a summary node.

Table 1: Notations and denotations used in the paper.

a metric to quantify the degree-of-personalization. EGISES was further modified and completed by Dasgupta et al. (2024) to incorporate a penalty due to accuracy drop and the PerSEval metric was proposed (detailed exposition in Appendix A). In this paper, we adopt PerSEval to demonstrate that SOTA user-encoders trained on a PerAugy-generated augmented dataset enhance the performance of SOTA personalized summarizers.

3 Modeling User Preference Datasets

In the following section, we first introduce User-Interaction-Graph (UIG) – a temporal knowledge graph-inspired data model for capturing dynamic user behavior trajectories. PerAugy operates on UIG to generate diverse synthetic trajectories, as discussed in depth in Section 5.

3.1 User-Interaction Graph (UIG)

A UIG is a Directed Acyclic Graph UIG:$<N, E>$ where N(nodes) $\equiv \{u_j^{(t_0)}\} \cup \{d^{(t_p)}\} \cup \{s^{(t_q)}\}$ and E (edges) $\equiv \{a_d^{(t_p)}\} \cup \{a_s^{(t_q)}\}$, where:

- $u_j^{(t_0)}$: j-th user (termed *u-node*) at time-step t_0 (i.e., source/start node of UIG).

- $d^{(t_p)}$: document (termed *d-node*) interacted at time-step t_p by one or more $u_j \in U$ ($U \equiv \{u_j^{(t_0)}\}$); *NT: $d^{(t_p)}$ can recur at different time-steps.*

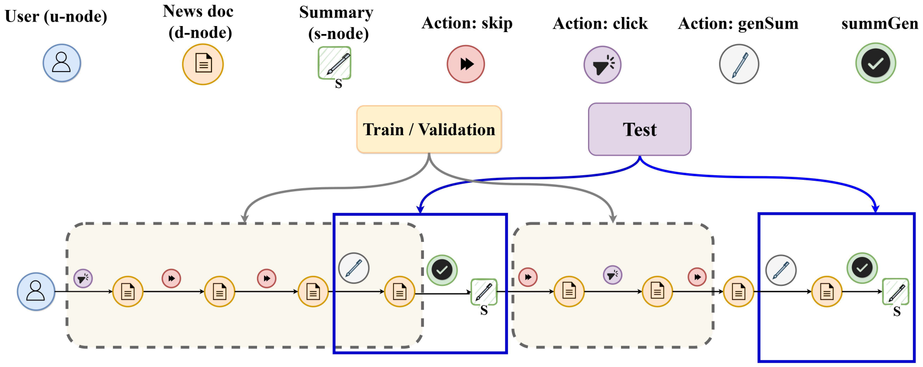


Figure 1: **UIG construction pipeline for PENS-styled datasets: Step 1**: *Documents from train/valid data are sequenced as d-nodes*; **Step 2**: *Reference personalized headlines for an intersecting d-node from test data are interleaved as s-nodes based on time-step*; **Step 3**: *If no intersecting d-node is found, the s-node along with corresponding d-node from test data are simply appended at their respective time-step.*

- $s_j^{(t_q)}$: the ***expected* user-specific** summary (termed *s-node*) at time-step t_q for the corresponding d-node $d^{(t_{q-1})}$ which needs to be summarized for $u_j \in U$; NT: $d^{(t_{q-1})}$ may have *multiple subjective* $s^{(t_q)}$ corresponding different users.
- $a_d^{(t_p)}$: a user interaction (incoming edge) on d-node $d^{(t_p)}$ at time-step t_p s.t. $a_d \in \{$*click, skip, generate-summary*$\}$, where *click* denotes interest (positive engagement), *skip* denotes non-engagement, and *generate-summary* (termed *genSumm*) denotes intent of reading the summary of the document instead.
- $a_s^{(t_q)}$: the follow-up edge of *genSumm* (acting on $d^{(t_{q-1})}$), between $d^{(t_{q-1})}$ and $s^{(t_q)}$, denoting either a write action by the user (in the case of UIG as training dataset) or a read action of summary generated by a summarizer model (during inference), and is termed *summGen*.

Trajectory: Given a UIG, the preference history (termed *trajectory*) of u_j is a sequence of interactions, denoted τ^{u_j}, starting at t_0 and ending at a d-node or s-node at t_{l-1}, where l is the trajectory length. Hence, a UIG is a pool of trajectories $\mathcal{T}$.

A UIG can hence be seen as a dynamic temporal knowledge graph (TKG) of user behavior. We now formally define Personalized Summarization as follows:

Definition 1. ***Personalized Summarization*** *Given a user trajectory τ^{u_j} of length l, a personalized parameterized (θ) summarizer model $\mathcal{M}_\theta$ takes a query document node ($d_q^{(t_l)}$) and generates a corresponding user (u_j) specific summary $s_{(q,u_j)}^{(t_{l+1})}$, where $\mathcal{M}_\theta$ is trained on $\mathcal{T}_{train}$ to approach the upper-bound of a chosen personalization metric (in our case, we use PerSEval (PSE)).*

3.2 UIG Construction from Preference Data

In the parlance of UIG, preference datasets suitable for personalized summarization training and evaluation are of two categories – (i) those which can be directly modeled into a trajectory pool $\mathcal{T}$ (e.g., the PENS dataset Ao et al. (2021)), dataset statistics are in Table 6, and (ii) those which lack user trajectories but contain discrete d-nodes, *model-generated* s-nodes (in contrast to user-generated s-nodes as per UIG definition), and *subjective* user feedback in the form of rating and the associated confidence score for that rating (e.g. OpenAI-Reddit dataset Völske et al. (2017)), statistics in Table 7. [3]. In our experiments, we select the OpenAI-Reddit dataset for establishing cross-domain generalizability of PerAugy and its broader applicability in more widely available datasets. We describe the UIG construction method for both types as follows:

[3] For detailed exposition of datasets, see Appendix B

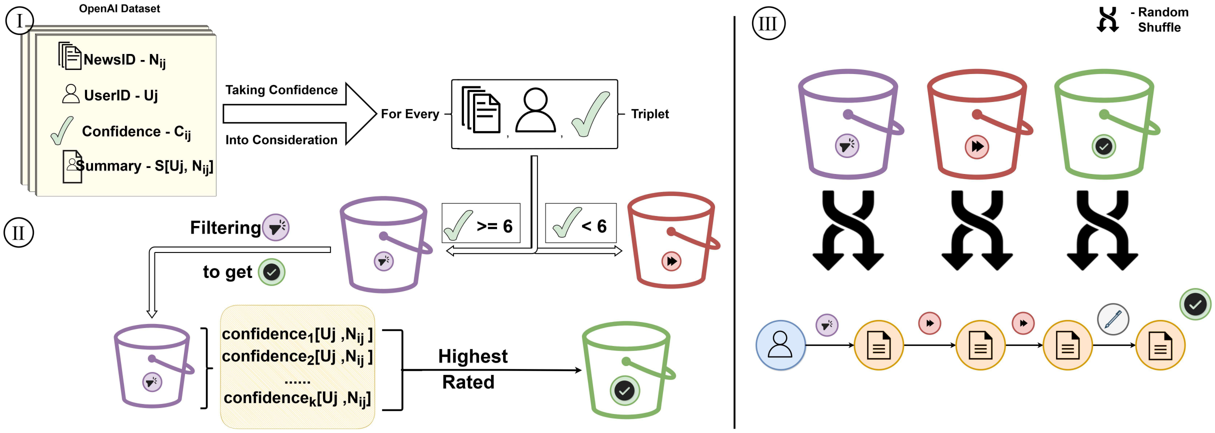


Figure 2: **UIG construction pipeline for OpenAI-styled datasets: Step 1:** *Extract NewsID, UserID, confidence, and summary*; **Step 2:** *Select top-rating $< U_j, N_{ij} >$ click pairs from filtered confidences*; **Step 3:** *Shuffle clicks, skips, and summaries to form trajectories.*

PENS-styled Datasets The construction of UIG is straightforward in the first case and is done in two steps. In the first step, *click* and *skip* interactions in the train dataset are mapped to document nodes (d-nodes) as incoming edges, forming the corresponding u-tier pool $\mathcal{T}$. As an example, for the PENS dataset, the *clkNews* interaction corresponds to a *click* edge and *uclkNews* to a *skip* edge, forming $\mathcal{T}^{\text{P}}_{\text{base}}$. However, the PENS dataset lacks user-specific s-nodes (i.e., true interest evolution over time), rendering $\mathcal{T}^{\text{P}}_{\text{base}}$ an *incomplete representation of the user dynamic preference*. It is important to note that despite this, most recent frameworks train on $\mathcal{T}^{\text{PENS}}$ using history or document titles as "pseudo-targets" or via unsupervised learning Ao et al. (2021); Song et al. (2023); Yang et al. (2023); Lian et al. (2025). We address this issue in the second step, where we incorporate the s-nodes from the test dataset ($\mathcal{T}_{\text{test}}$) at their associated time-steps into $\mathcal{T}$ with the addition of *genSumm* and *summGen* edges, forming a derived (and more diverse) user-profile pool $\mathcal{T}^{\text{syn-P}}_{\text{base}}$ (see Figure 1).

OpenAI-styled Datasets For the second category of datasets, we first do a pre-construction classification of clicked and skipped d-nodes for every human rater u_j. This is done based on a simple heuristic of selecting those d-nodes as clicked which has at least one corresponding model-generated summary (note that there can be multiple models) that received a confidence score above a chosen threshold. In the case of OpenAI-Reddit, we chose the threshold for clicked d-nodes to be 6 out of 9 (see Figure 2-II), forming $\mathcal{T}^{\text{OAI}}_{\text{base}}$. We then select the best model-generated summary (i.e., one with the highest rating given by u_j) as the surrogate expected s-node for u_j (Figure 2-II). We then randomly sequence all such $(d-s)$-node pairs along with the skipped d-nodes to form τ^{u_j} (thereby $\mathcal{T}^{\text{syn-OAI}}_{\text{base}}$; Figure 2-III). This method makes UIG-modeling *compatible with most summarization datasets that are not PENS-styled*. A detailed pseudo-code of UIG construction is given Algorithm 1.

4 Measuring Preference Data Diversity

To quantitatively measure the efficacy w.r.t diversity of any preference dataset (that can be modeled as a UIG), we propose a novel metric called DegreeD (***Degree**-of-**D**iversity*) and then briefly discuss the method to compute it given any UIG.

4.1 Degree-of-Diversity (DegreeD)

As a building block of the proposed metric DegreeD, we first define ***D**egree-of-**P**reference-**S**hift* (DePS) in a given UIG trajectory τ^{u_j} corresponding user u_j, where DePS quantifies the shift in a user's interest across τ^{u_j}.

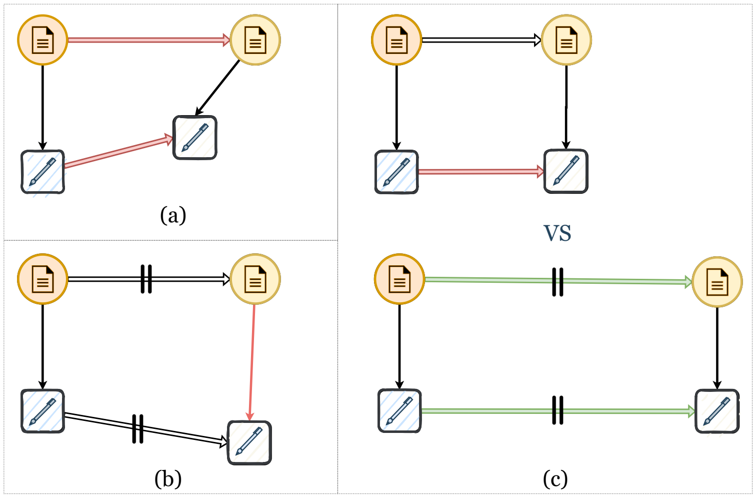


Figure 3: **Cases of `DegreeD` penalization**: (a) ***Disproportionality*** *between two successive expected s-nodes* $\delta(X_s)$ *and d-nodes divergence* $\delta(X_d)$, (b) *Proportionate but* ***unfaithful*** *in terms of alignment of s-node to core topic centrality of the corresponding d-node i.e. negative shift of s-node from corresponding d-node*, (c) ***Absolute Thematic Divergence*** *between two consecutive (d-s) pairs in terms of diversity of interactions in between.*

Definition 2 (**Degree of Preference Shift (`DePS`)**). `DePS` *is the ratio of the thematic divergence between consecutive d-nodes* $d^{(t_i)}$ *and* $d^{(t_{i+1})}$ *(denoted as* $\delta[X]_d$*) over a time span* $\Delta_{(t_i,t_{i+1})} = t_{i+1} - t_i$ *and that between the corresponding s-nodes (user's subjective expected summaries)* $s_j^{(t_i)}$ *and* $s_j^{(t_{i+1})}$ *(denoted as* $\delta[X]_s$*).*

The thematic divergence $\delta[X]_\bullet$ ($\bullet \in \{d, s\}$) is calculated as $\sigma(\bullet_j^{t_i}, \bullet_j^{t_{i+1}})$ where σ is a distance measure on a chosen metric space. As per the definition, `DePS` for the j-th user at any unit interval $\Delta_{(t_i,t_{i+1})}$ is formulated as follows:

$$\texttt{DePS}_j^{\Delta_{(t_i,t_{i+1})}} = \frac{min(\delta[X]_d, \delta[X]_{s_j}) + \epsilon}{max(\delta[X]_d, \delta[X]_{s_j}) + \epsilon} \tag{1}$$

The Expected `DePS`, $\mathbb{E}_j[\texttt{DePS}]$, for j-th user over the trajectory τ^{u_j} having length l is:

$$\mathbb{E}_j[\texttt{DePS}] = \frac{1}{l-1}\sum_{i=1}^{l-1} \texttt{DePS}_j^{\Delta_{(t_i,t_{i+1})}} \tag{2}$$

$\mathbb{E}_j$[DePS] penalizes the disproportionate alignment between the expected summaries ($\delta[X]_{s_j}$) and the document divergence ($\delta[X]_d$) for τ^{u_j} (Figure 3 (a)). However, it fails to penalize the case when any expected summary (s-node) in τ^{u_j} is not consistently faithful (in the sense of centrality to the core topic) to the corresponding document (d-node) (Figure 3 (b)). It also fails to penalize the case when the absolute thematic divergence is small (Figure 3 (c)). Both these cases can happen even with a high $\mathbb{E}_j[\texttt{DePS}]$. To address the first issue, we modify $\mathbb{E}_j[\texttt{DePS}]$ to incorporate the necessary penalties (i.e, penalized $\mathbb{E}_j[\texttt{DePS}]$ or $\mathbb{E}_j[\texttt{DePS}^{\mathcal{P}}]$) as follows:

$$\mathbb{E}_j[\texttt{DePS}^{\mathcal{P}}] = \frac{\sum_{i=1}^{l-1}\left(\texttt{DePS}_j^{\Delta_{(t_i,t_{i+1})}} \cdot \frac{\sigma(d^{(t_i)}, s_j^{(t_i)})}{\sigma(d^{(t_{i+1})}, s_j^{(t_{i+1})})}\right)}{l-1} \tag{3}$$

In the above equation, the second factor penalizes *negative shift* in faithfulness where the s-node starts deviating away from the corresponding d-node as compared to the deviation at previous time-steps. Note that the **second factor rewards *positive shift* where s-node comes closer to its corresponding**

d-node. To address the second issue, we inject an additional factor $\delta[X]_{s_j}$ that penalizes lack of thematic divergence in terms of user's actual interest/focus (hence, $\delta[X]_{s_j}$ instead of $\delta[X]_{d_j}$). $\delta[X]_{s_j}$ is regulated by the hyper-parameter $\alpha = (0, 1]$. The final DegreeD formulation for a UIG representing dataset $\mathcal{D}$ containing $|\mathbf{U}|$ unique user trajectories is:

$$\texttt{DegreeD}(\mathcal{D}) = \frac{\alpha}{|\mathbf{U}|} \cdot \sum_{j=1}^{|\mathbf{U}|} \delta[X]_{s_j} \cdot \mathbb{E}_j[\texttt{DePS}^{\mathcal{P}}] \tag{4}$$

A dataset $\mathcal{D}$ is suitable for personalization training if it has a high DegreeD score.

4.2 DegreeD Computation of UIGs

In section 4, we described a special case where for every d-node in an UIG we have a corresponding s-node in every trajectory τ, which evidently is unrealistic. In reality, many of the d-nodes will not have a corresponding s-node and hence, the calculation of DePS cannot be done at every unit time-interval $\Delta_{(t_i,t_{i+1})}$. This requires modification in the computing procedure so as to account for the missing s-nodes. To address this, the first "*surrogate*" s-node $s_j^{(t_1)}$ for the initial d-node $d^{(t_1)}$ at time-step t_1 is assumed to be the same as the document's title, as there is no prior preference history of a user at time-step t_1 and hence, subjectivity as a function of preference history does not arise yet. Let the first s-node $s_j^{(t_k)}$ occur at time-step t_k (i.e., the first valid interval is $\Delta_{(t_1,t_k)}$). Therefore, $\texttt{DePS}^{\Delta_{(t_1,t_k)}}$ for j-th user is calculated as:

$$\begin{aligned} \texttt{DePS}_j^{\Delta_{(t_1,t_k)}} &= \frac{min(\delta[X^{\Delta_{(t_1,t_k)}}]_d, \delta[X^{\Delta_{(t_1,t_k)}}]_{s_j}) + \epsilon}{max(\delta[X^{\Delta_{(t_1,t_k)}}]_d, \delta[X^{\Delta_{(t_1,t_k)}}]_{s_j}) + \epsilon}; \\ \text{where: } \delta[X^{\Delta_{(t_1,t_k)}}]_d &= \frac{1}{k-1} \cdot \sum_{i=1}^{k-1} \sigma(d^{(t_1)}, d^{(t_{1+i})}); \delta[X^{\Delta_{(t_1,t_k)}}]_{s_j} = \sigma(s_j^{(t_1)}, s_j^{(t_k)}) \end{aligned} \tag{5}$$

DegreeD is then computed over all valid intervals as per equation 4. In this paper, we represent d-nodes and s-nodes with their embeddings generated from a lightweight Sentence-BERT model (all-MiniLM-L6-v2) Reimers & Gurevych (2019) and use Manhattan Distance as the distance metric σ. In section 7, we empirically provide strong evidence that higher UIG DegreeD has strong correlation with user-encoder model accuracy when trained on such UIGs (DegreeD computation is in Algorithm 2). We compute the DegreeD of the PENS synthetic base pool $\mathcal{T}_{\text{base}}^{\text{syn-P}}$ and find a very low DegreeD score of 0.009. The OpenAI (Reddit) synthetic base pool $\mathcal{T}_{\text{base}}^{\text{syn-OAI}}$ also shows low DegreeD of 0.0079. In Section 7, we empirically show that higher UIG DegreeD has a strong correlation with user-encoder model accuracy when trained on such UIGs[4].

5 PerAugy: Augmentation of Base UIG

In this section, we introduce PerAugy, a novel data augmentation method aimed at improving personalization in summarizers. It enhances the accuracy of user-encoders via increasing diversity (i.e., w.r.t DegreeD) of the training datasets, thereby acting as a regularizer for better generalization.

5.1 PerAugy Pipeline: Overview

The PerAugy (**Per**turbation-based **Aug**mentation) pipeline has four steps. In the first step, we randomly sample (without replacement) m seed trajectories ($\mathcal{T}_{\text{sample}}^{m}$) from a given UIG (i.e., trajectory pool $\mathcal{T}_{\text{base}}^{syn}$). In the next step, we perform the "*Double Shuffling* (DS)" operation that selects each of the sampled trajectories as the target and **substitutes** trajectory-segments at different time-steps of the target with that of segments from the other $m-1$ trajectories. This leads to a modified "shuffled" sample $\mathcal{T}_{\text{DS}}^{m}$. We describe the DS operation in details in section 5.2. We then select the s-nodes of each trajectory $\tau \in \mathcal{T}_{\text{DS}}^{m}$ via a Bernoulli Trial and **perturb** the summary content on the basis of the corresponding d-node's similarity with preceding d-nodes. The details of the perturbation method, called "*Stochastic Markovian Perturbation* (SMP)", is given

[4] d/s-node embedding via SBERT model (all-MiniLM-L6-v2) Reimers & Gurevych (2019); σ : Manhattan distance

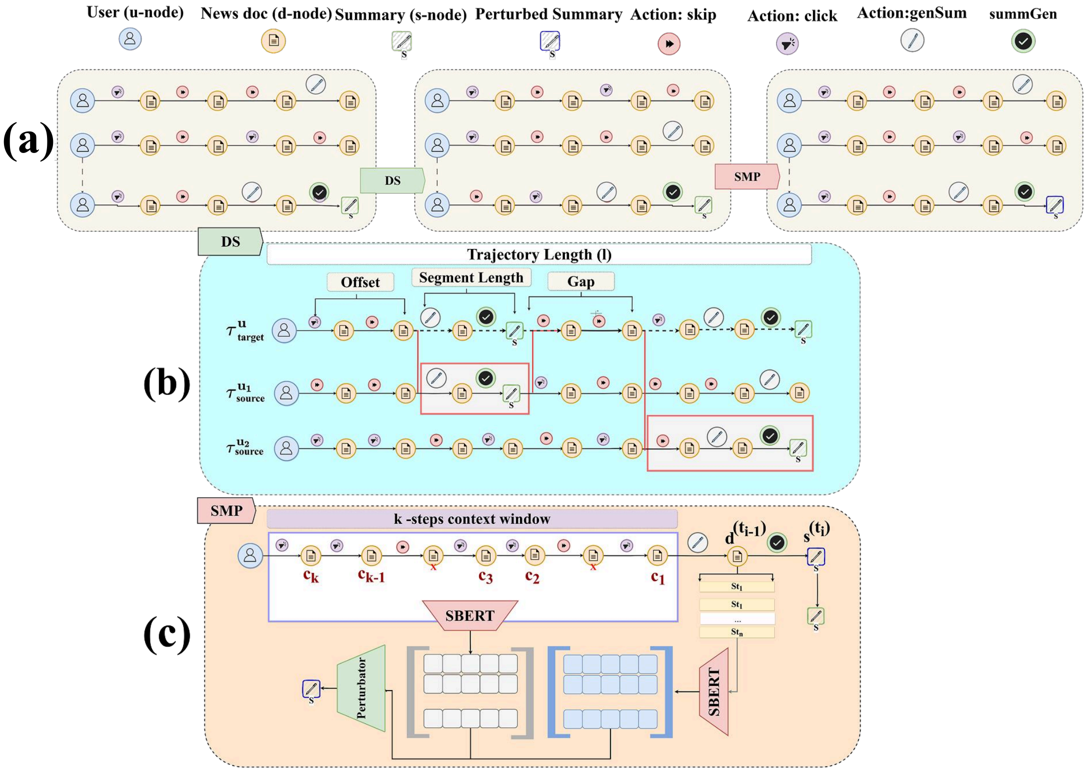


Figure 4: **PerAugy:** our proposed framework – (a) *Pipeline overview depicting the two-step augmentation*, (b) *Double Shuffling (DS) to ensure cross-trajectory augmentation and induce diffusion*, (c) *Stochastic Markovian Perturbation (SMP) to smoothen the s-nodes and modulate random diffusion incorporated in DS stage.*

in section 5.3. The resulting perturbed trajectories form $\mathcal{T}^{m}_{\text{SMP}}$ which represents a *new set of synthetic user preference history data*. Note that $\mathcal{T}^{m}_{\text{DS}}$ is added back into the trajectory pool ($\mathcal{T}^{syn}_{base} - \mathcal{T}^{m}_{\text{sample}}$) before the next iteration of sampling is done. PerAugy terminates after $|\mathcal{T}^{syn}_{\text{base}}|/m$ sampling iterations. We depict the pipeline in Figure 4(a).

5.2 Double Shuffling

In this section, we detail the Double Shuffling (DS) operation. The key design heuristics behind DS is that behavioral subsequences within real-life trajectories can be stitched together to form more diverse realistic synthetic trajectories. That is, a sequence of interactions of Alice can be stitched with that of Bob and Joe to result in a realistic synthetic profile that behaves like Alice during the first session, then Bob in the second, and then finally Joe. To simulate this, following the sampling method described in the previous section, given a target trajectory $\tau^{u_j}_{\text{target}} \in \mathcal{T}^{m}_{\text{sample}}$ corresponding to a user u_j (i.e., an existing ***real*** user), we first randomly select an "*Offset*" **O**. This offset determines the ***early-stage behavior sequence*** (behavior sequence is termed "*trajectory segment*" τ_{seg}) that ***should remain the same*** as u_j. **O** helps to make sure that a new trajectory does not start with an unrealistic initial segment. The random selection helps to generate early-stage trajectory segments of varying lengths in the augmented dataset. In the next step, we select $m-1$ trajectory segments ($\tau_{seg}^{u_{i=1:m-1}}$) from each of the remaining $m-1$ "*source*" trajectories $\tau^{u_{i=1:m-1}}_{\text{source}} \in \mathcal{T}^{m}_{\text{sample}}$ corresponding to $m-1$ users and substitute corresponding target segments having same length as that of these source segments. The $m-1$ time-steps on $\tau^{u_j}_{\text{target}}$ where the substitutions occur are controlled by "*Gap*" (**G**) which determines the number of time-steps that should be kept intact (i.e., same intermediate behavior sequences as the original u_j). Gap length is a hyper-parameter that we ablate on in our experiments (see section G.2). Longer gaps would signify that the new synthetic trajectory $\tau^{u_j}_{\text{DS}}$ is more similar to the corresponding original $\tau^{u_j}_{\text{sample}}$, while longer source segments and higher m value would lead to longer $\tau^{u_j}_{\text{DS}}$.

Note that each of the m trajectories in $\mathcal{T}^m_{\text{sample}}$ becomes a target in turn. Since the shuffling of segments between source and target trajectories is two-way, we call it "*double shuffling*". Although DS introduces diversity by aggregating trajectory segments from different users and threads them up at different time steps thereby altering their original positions, it *fails to ensure that the s-nodes that come intact along with the source segments have realistic coherence with the preceding nodes.* This is because the source s-node has been influenced by the **preceding source nodes** that **form its "*history*"**, and hence, may not be compatible with the new history in the target trajectory. To address this, we introduce a subsequent operation on each double-shuffled trajectory in $\mathcal{T}^m_{\text{DS}}$ called "*Stochastic Markovian Perturbation*" that we describe in the next section.

5.3 Stochastic Markovian Perturbation (SMP)

SMP smoothens the newly substituted incompatible s-node $s^{(t_i)}$ at time-step t_i by operating over a backward sliding *context window* $\tau^{u_{\text{target}}}c_k$ of k time-steps, derived from the corresponding d-node at $ti-1$. This process refines $s^{(t_i)}$ by perturbing it—replacing it with a sentence that better aligns with the temporal context of the target user. Specifically, SMP selects the top-p sentences from $d^{(t_{i-1})}$ that are most *influenced* by the context nodes $c_{q=1:k}$ within the window. In our experiments, we use top-1 selection, since the s-node represents a summary-level headline, making a single representative sentence sufficient. The notion of "influence" is quantified through the Root Mean Square Distance (RMSD) between each candidate sentence st_p in $d^{(t_{i-1})}$ and each context node c_q in $\tau^{u_{\text{target}}}c_k$. Sentence and context representations are computed using SBERT embeddings (as detailed in Section 4.2). To reflect temporal relevance, this influence is weighted by an exponential decay factor $e^{-\lambda \cdot pos(c_q)}$, where $pos_{c_1} = 0$ denotes the most recent context and thus receives maximum weight. As a result, earlier context nodes contribute less to the influence score. This formulation characterizes SMP as a k-order Markov process, where the prediction at t_i depends on a weighted combination of the preceding k time steps. A compact representation of SMP follows:

$$\begin{aligned}
&\text{SMP}(s^{(t_i)} \in \tau^{u_{\text{target}}}) = \hat{s}^{(t_i)} = \arg\min([\mathbf{\Sigma}]_{(n_{d^{(t_{i-1})}} \times k)} \cdot [e^{-\lambda \cdot pos_{(c_q)}}]_{(k \times 1)}); \\
&\text{where: } \mathbf{\Sigma}_{(p,q)} = \sigma(\mathbf{e}_{st_p}, \mathbf{e}_{c_q}); n_{d^{(t_{i-1})}} : \text{No. of sentences in } d^{(t_{i-1})}; \sigma : \text{RMSD}
\end{aligned} \tag{6}$$

Once the minimum-scoring sentence $\hat{s}^{(t_i)}$ is identified via this influence-weighted RMSD computation, the original s-node $s^{(t_i)}$ is replaced with $\hat{s}^{(t_i)}$. The DS operation mimics the stochastic diffusion of a user's interest into diverse themes, reflecting natural behavior; hence, **smoothing every s-node may inhibit the intended thematic diffusion**. Consequently, an s-node $s^{(t_i)}$ is selected for perturbation via a Bernoulli trial with perturbation probability p_{SMP} [5]. The DS and SMP operations generate synthetic user profiles $\hat{u}$ and trajectories $\tau^{\hat{u}}_{\text{SMP}} \in \mathcal{T}^m_{\text{SMP}}$. We ablate on the context window size k, decay constant λ, and p_{SMP} in Section G.2. The pseudo-code for DS is provided in Algorithm 3 and that of SMP is provided in Algorithm 4.

6 Evaluation

To evaluate the overall effectiveness and utility of `PerAugy`, we frame our investigation around three central research questions: **RQ-1**: Does `PerAugy` enhance `DegreeD`? **RQ-2**: Does an improvement in `DegreeD` lead to higher accuracy in SOTA user-encoder models? **RQ-3**: Do enhanced user-encoders, in turn, improve SOTA personalized summarization frameworks?

6.1 Experiment Setup

In this section, we describe our detailed experimental setup: The training and testing datasets, Training procedure and Baselines.

6.1.1 Augmented Synthetic Datasets

Training Data. We create two training datasets using `PerAugy`: $\mathcal{T}^{\mathcal{E}}_{\text{DS}}$ and $\mathcal{T}^{\mathcal{E}}_{\text{DS+SMP}}$. The dataset $\mathcal{T}^{\mathcal{E}}_{\text{DS}}$ is a *mixed bag* of trajectories sampled from ten different augmented datasets generated with varying <Gap-length

[5] All symbols used are described in Table 1.

g_l, Trajectory-length l> configurations *without SMP operations*. Among them, five datasets $\{\mathcal{T}_{\text{DS}}^{i=1:5}\}$ use a fixed $l = 150$ and vary $g_l \in \{10, 15, 20, 25, 40\}$, while the remaining five $\{\mathcal{T}_{\text{DS}}^{i=6:10}\}$ use a fixed $g_l = 25$ and vary $l \in \{50, 100, 125, 175, 200\}$. Each of the ten datasets contains 400K trajectories. From each, we sample 10% to construct the final $\mathcal{T}_{\text{DS}}^{\mathcal{E}}$.

Similarly, $\mathcal{T}_{\text{DS+SMP}}^{\mathcal{E}}$ is constructed via the same mixed-bag sampling strategy, but using SMP operations applied to each of the ten augmented datasets with a decay constant $\lambda = 0.3$, perturbation probability $p_{SMP} = 0.8$, and context length $k = 10$. Proportional sampling from diverse configurations helps mitigate overfitting and increases the generality of preference histories. We generate four variants in total: (i) $\mathcal{T}_{\text{DS}}^{\mathcal{E}-P}$, (ii) $\mathcal{T}_{\text{DS}}^{\mathcal{E}-OAI}$, (iii) $\mathcal{T}_{\text{DS+SMP}}^{\mathcal{E}-P}$, and (iv) $\mathcal{T}_{\text{DS+SMP}}^{\mathcal{E}-OAI}$, derived respectively from $\mathcal{T}_{\text{base}}^{\text{syn-P}}$ and $\mathcal{T}_{\text{base}}^{\text{syn-OAI}}$.

Test Data: User-Encoder Evaluation. We construct the test dataset $\mathcal{T}_{\text{test}}^{P}$ to accurately evaluate user encoders under realistic interaction conditions. Since the PENS validation set lacks skipped d-nodes and the PENS test set lacks negative d-nodes in the target bin, we merge Phase-1 clicks ($\tau_{seg_h}^{u_j}$) with Phase-2 (d, s) pairs ($\tau_s^{u_j}$) in sequential order. We split $\tau_s^{u_j}$ in half: the first half is appended to $\tau_{seg_h}^{u_j}$ to form the user history $\tau_{h_{\text{test}}}^{u_j}$, while the second half serves as the candidate set for next-click prediction. To ensure balance, we augment positive samples with negative samples—documents not clicked by user u_j in either Phase-1 or 2—randomly drawn from the index range $[50 : n_s^{u_j}]$, where $n_s^{u_j}$ is the total number of s-nodes in $\tau_s^{u_j}$. This process yields a realistic distribution of clicked and non-clicked items, enabling a fair **"next click prediction"** evaluation. The final test set contains 103 user trajectories (Table 6).

Test Data: Personalized Summarizer Evaluation. We evaluate personalized summarizers using the original PENS test set $\mathcal{T}_{\text{test}}$. For each user u_j, we retain Stage-1 click history $\tau_{seg_h}^{u_j}$ and use the 200 (d, s) pairs from Stage-2 as summarization queries. The model generates personalized summaries $\hat{s}^{u_j}$ conditioned on the query document d_{query} and $\tau_{seg_h}^{u_j}$ alone; the intermediate (d, s) pairs are not appended, resulting to infer summaries $\hat{s}_{1:200}^{u_j}$ based solely on $\tau_{seg_h}^{u_j}$.

6.1.2 User-Encoder Training

We train the user-encoder models *from scratch* on each of the mixed sets $\mathcal{T}_{\text{DS}}^{\mathcal{E}}$ and $\mathcal{T}_{\text{DS+SMP}}^{\mathcal{E}}$ (with batch size: 128 trajectories; epochs 2; Adam Optimizer ($\alpha : 1e-4$; $\beta_1 = 0.9$; $\beta_2 : 0.999$;$\epsilon = 1e-8$)), in contrast to standard fine-tuning, to analyze: (a) the extent to which synthetic datasets can replace real datasets, especially when such datasets are extremely scarce, and (b) to clearly understand the effect of the hyperparameters of PerAugy under ablation. During training, we split a user's trajectory $\tau_{\text{DS/DS+SMP}}^{u_j}$ into an input segment, also termed **train history-segment** ($\tau_{h_{\text{train}}}^{u_j}$), and a target segment $\tau_{\text{target}}^{u_j}$ at random time-step within the interval $[l^{u_j}/2, l^{u_j} - 3]$. The nodes of $\tau_{\text{target}}^{u_j}$ form a target candidate bin for the next node prediction task. We ablate on $\tau_{h_{\text{train}}}^{u_j}$ length in section 7.2. We further fine-tune TrRMIo (with epoch 1) on the top of $\mathcal{T}_{\text{base}}^{\text{syn-P}}$ to ascertain the impact of fine-tuning by each baseline augmentations (as discussed in Section 6.1.3).

6.1.3 Baseline Augmentation Methods

We evaluate PerAugy against three state-of-the-art (SOTA) algorithmic augmentation methods. We select PENS-SH as a strong baseline, as it is specifically designed for personalized summarization. We also choose S3-Aug as intrea-trajectory baseline, and SDAInter as inter-trajectory baseline, along with three LLM-based augmentation setups. PENS-SH merges multiple user interaction trajectories from $\mathcal{T}_{\text{train}}^{P}$ into synthetic ones by aligning common d-nodes to create diverse pseudo-users. S3-Aug applies intra-trajectory segment-shuffle-stitch operations to perturb temporal structure while preserving local coherence in $\mathcal{T}_{\text{train}}^{\text{S3}}$. SDAInter swaps interchangeable sub-sequences between user trajectories based on anchor overlaps and IoU (Interaction over Union) confidence, producing cross-user hybrids in $\mathcal{T}_{\text{train}}^{\text{SDA}}$. We convert each of these into UIG-compatible datasets, $\mathcal{T}^{\text{syn-PSH}}$, $\mathcal{T}^{\text{syn-S3}}$, and $\mathcal{T}^{\text{syn-SDA}}$, by injecting s-nodes from test dataset summaries to evaluate diversity. In the LLM-as-augmentor setup, we use LLaMA-2-13B, Mistral-v2-Instruct, and DeepSeek-7b-chat with two prompt strategies. **Chain-of-Thought (CoT)** prompts guide LLMs to reason step-by-step through user preferences to generate personalized summaries. **Prompt-Chaining (PC)** splits the task: first generating user behavior, then using that to generate personalized summaries. All augmented datasets are used to train user encoders, and we evaluate them to assess their diversity and impact on user-encoders.[6] The baseline augmentations are discussed in details in Appendix C.1, and prompt details are depicted in Figure 9 (Chain-of-thoughts) and Figure 10 (Prompt Chaining).

6.1.4 User-Encoder Baselines

To study RQ-2, we evaluate four SOTA user-encoder models originally trained on PENS or its source dataset MIND Wu et al. (2020): **NAML** Wu et al. (2019a), **NRMS** Wu et al. (2019b), **EBNR** Okura et al. (2017), and **TrRMIo** Song et al. (2023). **NAML** employs an attentive multi-view approach to integrate news features (e.g., titles, categories) and models user interests via attention over browsing history. **NRMS** applies multi-head self-attention in both news and user encoders to capture semantic relations in titles and personalize based on attended browsed content. **EBNR** incorporates dwell-time-based implicit negatives, combining Transformers and attention to model user preferences from both positive and negative signals. **TrRMIo** leverages pre-trained transformers with Attention Pooling, defining user interest through CTR-based filtering that emphasizes low-CTR news as indicators of core user preference. All baseline encoders are described in Appendix C.2.

6.1.5 Personalized Summarization Baselines

Most recent frameworks for personalized summarization are trained on the trajectory dataset $\mathcal{T}^{P}_{\text{base}}$ using either of two methodological paradigms. The first involves reinforcement learning setups, where models are optimized using a **"pseudo-target"** such as user history Ao et al. (2021) or document title Song et al. (2023) to approximate personalized summaries. The second involves unsupervised setups, where the training objective is not based on explicit summaries but instead aims to reduce the *surprise* in the generated summary Yang et al. (2023) or to align user representations with the style-preference centroid of similar users in their neighborhood Lian et al. (2025). We could not experiment with models of the second paradigm since they are closed[7]. To systematically investigate RQ-3, we adopt two state-of-the-art personalized summarization frameworks. The first is **PENS** Ao et al. (2021), a pointer-network-based model trained on the PENS dataset using policy gradient–based reinforcement learning. PENS utilizes user embeddings derived from third-party user-encoders such as NAML, NRMS, and EBNR, injecting them into the generation process to personalize summaries. The second is **GTP** Song et al. (2023), which follows a two-stage late-fusion approach trained on PENS-SH. In this framework, a general headline is first generated using a transformer-based encoder–decoder model, and then personalized in a second stage using TrRMIo-generated user embeddings to control stylistic and semantic refinements. Both of the baseline personalized summarization frameworks are detailed in Appendix C.3.

6.2 Evaluation Metrics

Encoder Evaluation. To evaluate user-encoders on the task of *Next-item Prediction*, we use standard metrics: AUC, MRR, and nDCG@5&10. AUC (Area Under the ROC Curve) measures the model's ability to rank a positive item higher than negative ones, indicating overall ranking quality. MRR (Mean Reciprocal Rank) evaluates the position of the first relevant item in the ranked list, giving higher scores when relevant items appear earlier. nDCG (normalized Discounted Cumulative Gain) at cutoff positions 5 and 10 assesses both the relevance and position of items, rewarding models that rank relevant items higher in the top-K predictions.

Personalization Evaluation. We adopt PSE-SU4 as the PerSEval (only existing personalized summarization evaluation metric known so far) variant to measure the boost in *degree-of-personalization* for both frameworks, owing to its high human-judgment correlation (Pearson's r: 0.6; Spearman's ρ: 0.6; Kendall's τ: 0.51) and computational efficiency Dasgupta et al. (2024). PerSEVal measures how well a summarization model personalizes its outputs to individual user preferences (responsiveness) while also penalizing it for poor or inconsistent accuracy across users. It balances the trade-off between generating diverse, user-specific

[6] UIG statistics are detailed in Table 8.

[7] We are yet to receive the codebase from the authors.

Augmentation Baselines	Avg Unique Topics/Trajectory	Rate of Change of Topics (RCT)	DegreeD
PENS	7.3	0.56	0.009
OpenAI (Reddit) (OAI)	8.5	0.42	0.008
PENS-SH†	7.4	0.54	0.067
S3†	7.3	0.56	0.019
SDAInter†	7.8	0.63	0.083
LLaMA-2-13B†	8.8	0.61	0.113
Mistral-7B†	8.6	0.63	0.144
DeepSeek-R1†	9.4	0.68	0.219
PerAugy†	13.6	0.77	**0.289**
PerAugy-OAI	11.7	0.72	0.121

Table 2: **DegreeD analysis:** Comparison of DegreeD scores, unique topics, and topic transition rates across datasets. † indicates augmentation followed by UIG abstraction on PENS. **Observation-1:** PerAugy *shows higher* DegreeD *than its seeds and augmentation baselines.* **Observation-2:** *While topic count and transition rate relate to diversity, they fall short in capturing preference shifts effectively (e.g. PENS-SH has lower RCT than PENS, although it leads to higher user-encoder accuracy; see Table 3).*

summaries and maintaining relevance to the expected content. The detailed formulation is described in Appendix A.

Human-Judgment based Evaluation. Direct human evaluation of personalized summarization faces fundamental feasibility issues. A third-party annotator cannot reasonably adopt the evolving preferences of a target user after parsing through extensive and often noisy interaction histories—ranging from raw click headlines and skipped articles to long Reddit threads. Personalization hinges on nuanced, longitudinal signals such as shifting stances, sub-topic interests, and stylistic inclinations, which are often subtle and subjective. Any attempt to reduce this complexity into a simplified abstraction risks erasing the contributions of more expressive personalization models that can capture temporal shifts in preference history, collapsing the goal of personalized summarization to persona-centric (static) summarization (as described in Section 2.1). Thus, to assess the cognitive validity of PerSEval, Dasgupta et al. (2024) designed a survey-based meta-evaluation simulating how human evaluators perceive personalization. Participants rated the similarity of summary pairs (model-generated and gold-reference) without knowing their source. From these ratings, they constructed DEGRESS-HJ (a human-judged version of DEGRESS) using normalized similarity as divergence and compared it against DEGRESS using correlation metrics (Pearson's r, Spearman's ρ, Kendall's τ), and further evaluated whether applying standard accuracy metrics as discounting factors over DEGRESS-HJ (mimicking EDP) aligns with PerSEval scores. Strong correlations in both stages confirm that *human evaluators intuitively align with PerSEval's ratio-based responsiveness and factor-based accuracy penalty*—indicating that PerSEval has strong human-judgment validity and does not require further human evaluation in this setup.

7 Observations and Insights

In this section, we discuss the results of each of the research questions outlined in Section 6.

7.1 RQ-1: Effect on Dataset Diversity

We study the boost in diversity w.r.t DegreeD[8] in the augmented mix-bag datasets $\mathcal{T}_{\text{DS}}^{\mathcal{E}-\text{P/OAI}}$ and $\mathcal{T}_{\text{DS+SMP}}^{\mathcal{E}-\text{P/OAI}}$ in comparison to that of PENS and its corresponding augmented datasets ($\mathcal{T}_{\text{base}}^{\text{syn-P}}$, $\mathcal{T}^{\text{syn-PSH}}$, $\mathcal{T}^{\text{syn-S3}}$, and $\mathcal{T}^{\text{syn-SDA}}$), OpenAI (Reddit) ($\mathcal{T}_{\text{base}}^{\text{syn-OAI}}$), and LLM-generated UIGs ($\mathcal{T}^{\text{LLaMA-2}}$, $\mathcal{T}^{\text{Mistral}}$, & $\mathcal{T}^{\text{DeepSeek-R1}}$). We see significant[9] boost in DegreeD with $\mathcal{T}_{\text{DS+SMP}}^{\mathcal{E}-P}$ performing the best, with a boost of 0.28↑ w.r.t $\mathcal{T}_{\text{base}}^{\text{syn-P}}$. It outperforms all other baseline *heuristic-based* augmentations (boost 0.23↑ w.r.t best performing $\mathcal{T}^{\text{syn-SDA}}$) and also LLM-generated baselines (boost 0.07↑ w.r.t best performing $\mathcal{T}^{\text{DeepSeek-R1}}$). We also find $\mathcal{T}_{\text{DS+SMP}}^{\mathcal{E}-\text{OAI}}$

[8] $\alpha = 1$ for analyzing the effect of 100% penalty.

[9] Significance: $\rho < 0.01$ (sample size: $400K$).

Method	**NAML**				**EBNR**				**NRMS**				**TrRMIo (ft)**			
	AUC	MRR	nDCG@5	nDCG@10	AUC	MRR	nDCG@5	nDCG@10	AUC	MRR	nDCG@5	nDCG@10	AUC	MRR	nDCG@5	nDCG@10
PENS(Original)	0.48	0.74	0.81	0.81	0.45	0.72	0.77	0.77	0.47	0.73	0.80	0.80	0.47	0.7	0.78	0.78
PENS-SH	0.48	0.67	0.79	0.79	0.46	0.68	0.79	0.79	0.48	0.72	0.81	0.81	0.62	0.89	0.94	0.94
S3	0.47	0.71	0.79	0.79	0.46	0.69	0.78	0.78	0.47	0.70	0.81	0.81	0.51	0.75	0.8	0.8
SDAInter	0.53	0.78	0.83	0.83	0.51	0.75	0.79	0.79	0.52	0.77	0.83	0.83	0.56	0.87	0.95	0.95
LLaMA2(13B)	0.43	0.62	0.68	0.68	0.46	0.68	0.74	0.74	0.41	0.53	0.58	0.58	0.42	0.67	0.73	0.73
Mistral(7B)	0.45	0.64	0.71	0.71	0.48	0.70	0.74	0.74	0.45	0.59	0.68	0.68	0.56	0.73	0.76	0.76
DeepSeek-R1	0.43	0.61	0.65	0.65	0.45	0.64	0.72	0.72	0.44	0.54	0.65	0.65	0.47	0.69	0.77	0.77
PerAugy (ours)	**0.59**	**0.79**	**0.87**	**0.87**	**0.59**	**0.81**	**0.88**	**0.88**	**0.59**	**0.83**	**0.86**	**0.86**	**0.76**	**0.91**	**0.97**	**0.97**

Table 3: **User encoder performance (*trained-from-scratch*):** Models trained on PENS and its augmented variants, including PerAugy (DS+SMP). **Observation-1:** PerAugy *outperforms all baselines across models (NAML, EBNR, NRMS) and metrics (AUC, MRR, nDCG@5/10), when trained-from-scratch.* **Observation-2:** *When finetuned on TrRMIo*PerAugy *consistently outperforms all baseline augmentation strategies, as compared to their fine-tuned versions.* **Observation-3:** *While some methods (e.g., SDAInter) help, others (e.g., S3, LLaMA2) degrade performance, showing the impact of augmentation and UIG quality.*

outperforms its seed UIG $\mathcal{T}^{\text{syn-OAI}}_{\text{base}}$ with a boost of 0.113↑. We observe a linear relationship between the number of topics, topic shifts, and DegreeD scores, as detailed in Table 2. We ablate various hyperparameters of PerAugy, including gap length g_l and trajectory length l for $\mathcal{T}^{\mathcal{E}-\text{P}}_{\text{DS}}$, as well as context length k, decay constant λ, and perturbation probability p_{SMP} for $\mathcal{T}^{\mathcal{E}-\text{P}}_{\text{DS+SMP}}$, and the results are detailed in Appendix G.1.

7.2 RQ-2: Effect on User-Encoder Accuracy

Comparison with SOTA augmentation strategy. We observe a *significant improvement in accuracy* across all the trained-from-scratch baseline encoders[10] when trained on our proposed dataset $\mathcal{T}^{\mathcal{E}-\text{P}}_{\text{DS+SMP}}$, compared to their original performance using the standard preference training set $\mathcal{T}^{P}_{\text{train}}$. The best results obtained using PerAugy show notable relative gains of 0.139 ↑ in AUC, 0.108 ↑ in nDCG@5/10 (on the EBNR encoder), and 0.096 ↑ in MRR (on the NRMS encoder), clearly demonstrating that the PerAugy augmented dataset can effectively *substitute for scarce preference training data.* We also find that PerAugy significantly outperforms all baseline augmentation methods (S3, PENS-SH, and SDAInter) when the user encoders (NAML, EBNR, and NRMS) are trained-from-scratch. Specifically, we observe average gains of 0.127 ↑, 0.143 ↑, and 0.09 ↑ over S3, 0.117 ↑, 0.12 ↑, and 0.07 ↑ over PENS-SH, and 0.103 ↑, 0.09 ↑, and 0.067 ↑ over SDAInter in terms of AUC, MRR, and nDCG@5/10 respectively (Detailed results in Table 3).

Comparison with LLM-generated Train sets. Furthermore, to assess the effectiveness in comparison to LLM-generated training data, we train the same user encoders from scratch using $\mathcal{T}^{\text{LLaMA-2}}_{\text{train}}$, $\mathcal{T}^{\text{Mistral}}_{\text{train}}$, and $\mathcal{T}^{\text{DeepSeek-R1}}_{\text{train}}$. In comparison to these LLM-generated baselines, our PerAugy training set $\mathcal{T}^{\mathcal{E}-\text{P}}_{\text{DS+SMP}}$ yields average performance gains of 0.157 ↑, 0.2 ↑, and 0.203 ↑ over LLaMA2; 0.13 ↑, 0.167 ↑, and 0.16 ↑ over Mistral; and 0.15 ↑, 0.213 ↑, and 0.197 ↑ over DeepSeek-R1 – again reported in terms of AUC, MRR, and nDCG@5/10 respectively. These consistent improvements across all metrics and models further validate the effectiveness of the PerAugy as an augmentation strategy (For details, see Table 3).

Effect of Finetuning. We reserve TrRMIo to analyze the fine-tuning performance of PerAugy and find it to outperform the best performing PENS-SH-based fine-tuning (0.14 ↑ w.r.t AUC, 0.034 ↑ w.r.t MRR, & 0.026 ↑ w.r.t nDCG@5/10) (Results in Table 3).

Cross-domain Study. When trained on $\mathcal{T}^{\mathcal{E}-\text{OAI}}_{\text{DS/DS+SMP}}$ and tested on the same PENS test data, we see that PerAugy can be reliably applied to OpenAI (Reddit) like datasets that do not contain preference histories (best (NRMS): 0.163 ↑ w.r.t AUC, 0.112 ↑ w.r.t MRR, 0.09 ↑ w.r.t nDCG@5/10). This also indicates cross-domain transferability. The results are detailed in Figure 7.

[10] All results are statistically significant with $p < 0.01$ (test size: 18.1K positive and negative decision nodes).

Metric	Correlation with `DegreeD`		
	Pearson r	**Spearman ρ**	**Kendall's τ**
AUC	0.682	**0.737**	0.578
MRR	0.672	0.725	0.552
nDCG@5	**0.687**	0.731	**0.58**
nDCG@10	**0.687**	0.731	**0.58**

Table 4: **Meta-evaluation of `DegreeD` via correlation analysis:** Correlation (Pearson r, Spearman ρ, Kendall's τ) between **DegreeD** and evaluation metrics. **Observation-1:** *AUC, MRR, nDCG@5, and nDCG@10 show strong positive correlations with* `DegreeD`*, supporting it as a reliable generalizability indicator;* **Observation-2:** *AUC and nDCG yield the highest correlations, showing greater sensitivity to data diversity.*

Personalized Summarizer	User-Encoder (w/ Injection Type)	PENS (Original)	`PerAugy` (DS+SMP)		
			Trained from-scratch	**Finetuned**	**End-to-End Finetuned**
PENS	**NAML (T1)**	0.013	**0.014**	**0.015**	**0.017**
	EBNR (T1)	0.008	0.010	0.010	*NA*
	EBNR (T2)	0.005	0.008	0.010	0.009
	NRMS (T1)	0.011	0.010	0.011	*NA*
	NRMS (T2)	0.004	0.007	0.005	*NA*
GTP	**TrRMIo (w/ title)**	0.006	**0.017**	**0.020**	**0.022**

Table 5: **Performance of Personalized Summarizers.** PENS and GTP frameworks are evaluated using PSE-SU4. All models are injected with the same user-encoder used during their original training. **Observation-1:** *GTP utilizes improved user embeddings* **best** *in both encoder finetuning and end-to-end tuning.* **Observation-2:** *In PENS, NAML-T1 shows a clear boost, while other variants fail to capitalize on the finetuned encoder.*

Diversity effect on Encoder Performance (and Meta-evaluation of `DegreeD`). We find a strong positive correlation between `DegreeD` and encoder accuracy metrics (see Table 4), which establishes that (i) diversity in training dataset acts a regularizer and boosts the performance of user encoders, and (ii) `DegreeD` can be reliably trusted as an indicator to the generalizability of the training dataset. We ablate on the mix training data $\mathcal{T}_{\text{DS}}^{\mathcal{E}-\text{P}}$ to analyze the effect of DS hyper-parameters – gap-length g_l (Section 6.1.1) and train history-segment length $\tau_{h_{\text{train}}}$: $\{l/2, 5l/8, 3l/4, 7l/8, l-3\}$ (l: trajectory length). For SMP hyper-parameters (k: $\{10, 15, 20\}, \lambda$: $\{0.3, 0.8, 1\}, p_{\text{SMP}}$: $\{0.5, 0.8, 1\}$), we also ablate on $\mathcal{T}_{\text{DS+SMP}}^{\mathcal{E}-\text{P}}$, and ablation results are detailed in Appendix G.2.

7.3 RQ-3: Effect on Personalization

We examine how effectively the baseline personalized summarization frameworks—GTP and PENS—leverage their corresponding improved user encoders, focusing on the frameworks' sensitivity to enhanced user history embeddings. To isolate the contribution of user encoder improvements, we first replace the fine-tuned encoders with their *train-from-scratch* counterparts. This setup helps us assess the raw effect of our augmentation strategy, `PerAugy`. With DS, the highest improvement in the PSE-SU4 score is for the GTP framework (+TrRMIo), achieving a gain of 0.012 ↑ over its baseline (original PSE-SU4: 0.006). On the PENS framework, the PENS(+EBNR+T2) variant shows a notable performance increase of 0.003↑ under DS. PENS variants that utilize NAML and NRMS encoders with T1 injection appear to benefit more from the SMP augmentation, yielding additional gains of 0.006↑ and 0.002↑ respectively—on top of what DS alone provides. When we switch to using the fine-tuned versions of the user encoders, the improvements in PSE-SU4 become more pronounced across models. The best results with these encoders include a boost of 0.014 ↑ for GTP+TrRMIo, 0.005 ↑ for PENS+EBNR-T2, and 0.002 ↑ for PENS+NAML-T1. Finally, to evaluate the full effect of end-to-end fine-tuning of the summarization frameworks with `PerAugy`, we utilize NAML(T1), EBNR(T2), and TrRMIo (since these encoders had shown relatively greater improvements in their fine-tuned versions). This end-to-end training leads to further performance boosts for NAML(T1) and GTP, both gaining 0.002 ↑, whereas EBNR(T2) shows a slight performance decline of 0.002 ↓. For detailed results, refer to Table 5.

This indicates that the architecture and the injection method play a major role. This is evident from the best-performing GTP, which is trained with the *additional loss on aligning generated summary embedding with the history embedding* by the TrRMIo user-encoder. This results in generated summaries being a superior reflection of user preference, while the PENS framework falls behind due to a lack of alignment with the user's preference history.

8 Related Work

Prior works on data augmentation have largely focused on generative approaches for dialogue summarization Liu et al. (2022); Ouyang et al. (2023); Park et al. (2024), and document-level augmentation for generalized summarization tasks Fabbri et al. (2021); Chen et al. (2023b); Sahu et al. (2025). However, to the best of our knowledge, **preference-oriented data augmentation** in the *context of personalized summarization* remains significantly underexplored. The most relevant effort in this domain is PENS-SH Song et al. (2023), which constructs synthetic user trajectories by identifying and merging common d-nodes across multiple user interaction graphs (UIGs). While effective at preserving shared preferences, PENS-SH fails to retain temporal order information due to the loss of time-step data, and entirely lacks intermediate summary nodes (s-nodes), leading to an incomplete representation of user intent and preference evolution. Broadly, preference data augmentation methods for sequential recommendation can be categorized into two classes: *intra-trajectory* and *cross-trajectory* augmentation.

8.1 Intra-Trajectory Augmentation

Most intra-trajectory methods perturb each user's history by locally manipulating nodes or segments, but within the same trajectory. For example, MBASR Xiao et al. (2024a) swaps adjacent subsequences to introduce diversity, but on highly monotonous trajectories this yields minimal change and may even inject unrealistic temporal transitions. STEAM Lin et al. (2023) drops or inserts individual nodes, yet on long sessions such edits either become computationally prohibitive or break the natural flow of interactions. L2Aug Wang et al. (2022) learns a policy to delete nodes, but deletions disrupt continuity and remove potentially informative context. The Heuristic SAMPLER model Chen et al. (2023c) replaces single nodes via popularity- or co-occurrence-based heuristics, but such isolated swaps fail to shift the sequence's overall engagement degree. ASReP Liu et al. (2021) extends trajectories by reverse-pretrained pseudo-nodes, yet inserted nodes often lack genuine temporal relevance. Finally, BTBR's masking and swapping Li et al. (2023) targets basket-level prediction objectives rather than enriching input diversity for downstream encoders.

PerAugy departs from these schemes by applying controlled perturbations that operate at both micro- and macro-scales while preserving strict temporal coherence since intra-trajectory augmentations does not solve the problem of monotonous user history. Rather than swapping or deleting isolated nodes, we perform shuffling that incorporates controlled shifts and apply perturbation to diffusion. This approach polishes the sequence—enriching diversity—without sacrificing realistic time-step information or overburdening the augmentation pipeline, making it scalable to long trajectories.

8.2 Cross-Trajectory Augmentation

Cross-trajectory techniques draw patterns across users to forge new sequences and augments multiple trajectories. DR4SR Yin et al. (2024) uses a transformer to learn global sequence regeneration, but applying interchangeable patterns across subjective "s-node" summaries risks injecting generic behaviors that dilute personalization. TiCoSeRec Dang et al. (2024) enforces uniform time-interval distributions via operations like crop, mask, and substitute, yet our trajectories inherently assume consistent unit-step timing, making such time-aware edits redundant. FDA Chen et al. (2023a) synthetically balances user profiles for fairness but generates monotonic "complemented" sequences that mirror existing repetition rather than diversify it.

divSPA-style methods Liu et al. (2023) swap segments between similar users, yet similarity-based exchanges often leave the overall interaction degree unchanged and introduce context mismatches.

Instead of wholesale regeneration or arbitrary segment swaps, PerAugy leverages cross-trajectory statistics to adapt its perturbation parameters dynamically. By analyzing inter-user variance in s-node distributions, our method determines optimal perturbation scales and target positions, ensuring that borrowed structure enhances diversity without compromising each trajectory's unique summarization points w.r.t. realistic user-behaviors. This yields augmented sequences that are both personalized and information-rich, overcoming the homogenization pitfalls of existing cross-trajectory approaches.

9 Discussions & Limitations

PerAugy enhances the diversity and expressiveness of the original user history data while preserving personalization fidelity. This approach is particularly beneficial for improving model robustness and generalization in sparse or skewed datasets. Looking forward, we see promising opportunities in extending this augmentation framework using large language models (LLMs). In particular, we are investigating prompt-tuning based augmentation techniques that could generate more semantically rich and user-aligned variations of preference histories. Such methods hold the potential to be especially impactful in low-resource or non-PENS-like domains, where user signals are limited or noisy. Additionally, we aim to ground perturbation modeling in more principled stochastic processes. One such candidate is the Itô process, which incorporates both deterministic trends (drift) and random fluctuations (diffusion). Modeling user preference evolution through such continuous-time stochastic frameworks may offer a more realistic approximation of human behavior, allowing for fine-grained control over the intensity and direction of perturbations. This could open avenues for theoretically grounded, temporally aware augmentation strategies that better reflect user dynamics in real-world settings.

10 Conclusion

In this paper, we introduced PerAugy, a novel data augmentation technique designed to enhance the personalization capabilities of summarization models. By addressing limitations in current personalized datasets like PENS, PerAugy generates synthetic, diverse user interaction trajectories, reducing overfitting and improving generalization across domains. Techniques like Double Shuffling (DS) and Stochastic Markovian Perturbation (SMP) ensure that the augmented data remains realistic and coherent, enabling models to better align with individual user preferences. Our evaluation demonstrated significant improvements in the personalization metric PSE (an average of 61.2% boost), particularly in models like PENS+NRMS+T2, which achieved a 75% performance increase (PSE-RG-SU4). PerAugy also improved user-encoders such as NAML, EBNR, and NRMS, enhancing their ability to capture user preferences with average boosts of 24%, 25%, and 18% over baseline augmentations (w.r.t AUC, MRR, and nDCG@5&10). We further demonstrated its potential as a reliable generator of synthetic datasets in low-resource domains like OpenAI (Reddit), with encoder boosts of 19%, 25%, and 17% in the same metrics, broadening its applicability. While PerAugy is a critical advancement in addressing data scarcity and generalization in personalized summarization, future work will refine its techniques and explore adaptability to more models and architectures.

References

Xiang Ao, Xiting Wang, Ling Luo, Ying Qiao, Qing He, and Xing Xie. PENS: A dataset and generic framework for personalized news headline generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 82–92, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.7. URL https://aclanthology.org/2021.acl-long.7/.

Pengshan Cai, Kaiqiang Song, Sangwoo Cho, Hongwei Wang, Xiaoyang Wang, Hong Yu, Fei Liu, and Dong Yu. Generating user-engaging news headlines. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 3265–3280, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.183. URL https://aclanthology.org/2023.acl-long.183/.

Lei Chen, Le Wu, Kun Zhang, Richang Hong, Defu Lian, Zhiqiang Zhang, Jun Zhou, and Meng Wang. Improving recommendation fairness via data augmentation. In Proceedings of the ACM Web Conference 2023, WWW '23, pp. 1012–1020, New York, NY, USA, 2023a. Association for Computing Machinery. ISBN 9781450394161. doi: 10.1145/3543507.3583341. URL https://doi.org/10.1145/3543507.3583341.

Xiuying Chen, Guodong Long, Chongyang Tao, Mingzhe Li, Xin Gao, Chengqi Zhang, and Xiangliang Zhang. Improving the robustness of summarization systems with dual augmentation. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 6846–6857, Toronto, Canada, July 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.378. URL https://aclanthology.org/2023.acl-long.378/.

Xu Chen, Zhenlei Wang, Hongteng Xu, Jingsen Zhang, Yongfeng Zhang, Wayne Xin Zhao, and Ji-Rong Wen. Data augmented sequential recommendation based on counterfactual thinking. IEEE Transactions on Knowledge and Data Engineering, 35(9):9181–9194, 2023c. doi: 10.1109/TKDE.2022.3222070.

Yizhou Dang, Enneng Yang, Guibing Guo, Linying Jiang, Xingwei Wang, Xiaoxiao Xu, Qinghui Sun, and Hong Liu. Ticoserec: Augmenting data to uniform sequences by time intervals for effective recommendation. IEEE Transactions on Knowledge and Data Engineering, 36(6):2686–2700, 2024. doi: 10.1109/TKDE.2023.3324312.

Sourish Dasgupta, Ankush Chander, Tanmoy Chakraborty, Parth Borad, and Isha Motiyani. PerSEval: Assessing personalization in text summarizers. Transactions on Machine Learning Research, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=yqT7eBz1VJ.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, and Chengqi et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. GSum: A general framework for guided neural abstractive summarization. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4830–4842, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.384. URL https://aclanthology.org/2021.naacl-main.384/.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 1074–1084, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1102. URL https://aclanthology.org/P19-1102.

Alexander Fabbri, Simeng Han, Haoyuan Li, Haoran Li, Marjan Ghazvininejad, Shafiq Joty, Dragomir Radev, and Yashar Mehdad. Improving zero and few-shot abstractive summarization with intermediate fine-tuning and data augmentation. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 704–717, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.57. URL https://aclanthology.org/2021.naacl-main.57/.

Shivam Grover, Amin Jalali, and Ali Etemad. Segment, shuffle, and stitch: A simple layer for improving time-series representations. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), Advances in Neural Information Processing Systems, volume 37, pp. 4878–4905. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/0898f05f6c1d247be3eab8da93d33da1-Paper-Conference.pdf.

Junxian He, Wojciech Kryscinski, Bryan McCann, Nazneen Rajani, and Caiming Xiong. CTRLsum: Towards generic controllable text summarization. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 5879–5915, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.396. URL https://aclanthology.org/2022.emnlp-main.396/.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS'15, pp. 1693–1701, Cambridge, MA, USA, 2015. MIT Press.

Zhiting Hu, Bowen Tan, Ruslan Salakhutdinov, Tom Mitchell, and Eric P. Xing. Learning data manipulation for augmentation and weighting. Curran Associates Inc., Red Hook, NY, USA, 2019.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. arXiv e-prints, pp. arXiv–2310, 2023.

Yang Jiao, Fan Yang, Yetian Chen, Yan Gao, Jia Liu, and Yi Sun. Rethinking sequential relationships: Improving sequential recommenders with inter-sequence data augmentation. In Companion Proceedings of the ACM Web Conference 2024, WWW '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400701726.

Hannah Rose Kirk, Bertie Vidgen, Paul Röttger, and Scott A Hale. The benefits, risks and bounds of personalizing the alignment of large language models to individuals. Nature Machine Intelligence, pp. 1–10, 2024.

Frederic Kirstein, Terry Ruas, Robert Kratel, and Bela Gipp. Tell me what I need to know: Exploring LLM-based (personalized) abstractive multi-source meeting summarization. In Franck Dernoncourt, Daniel Preoţiuc-Pietro, and Anastasia Shimorina (eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, pp. 920–939, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-industry.69. URL https://aclanthology.org/2024.emnlp-industry.69/.

Suchetha N. Kunnath, David Pride, and Petr Knoth. Prompting strategies for citation classification. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM '23, pp. 1127–1137, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701245. doi: 10.1145/3583780.3615018. URL https://doi.org/10.1145/3583780.3615018.

Ming Li, Mozhdeh Ariannezhad, Andrew Yates, and Maarten de Rijke. Masked and swapped sequence modeling for next novel basket recommendation in grocery shopping. In Proceedings of the 17th ACM Conference on Recommender Systems, RecSys '23, pp. 35–46, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702419. doi: 10.1145/3604915.3608803. URL https://doi.org/10.1145/3604915.3608803.

Junhong Lian, Xiang Ao, Xinyu Liu, Yang Liu, and Qing He. Panoramic interests: Stylistic-content aware personalized headline generation. In Companion Proceedings of the ACM on Web Conference 2025, 2025.

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

Yujie Lin, Chenyang Wang, Zhumin Chen, Zhaochun Ren, Xin Xin, Qiang Yan, Maarten de Rijke, Xiuzhen Cheng, and Pengjie Ren. A self-correcting sequential recommender, 2023. URL https://arxiv.org/abs/2303.02297.

Chong Liu, Xiaoyang Liu, Ruobing Xie, Lixin Zhang, Feng Xia, and Leyu Lin. Learning from all sides: Diversified positive augmentation via self-distillation in recommendation. CoRR, abs/2308.07629, 2023. URL https://doi.org/10.48550/arXiv.2308.07629.

Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, and Andrew M. Dai. Best practices and lessons learned on synthetic data. In First Conference on Language Modeling, 2024. URL https://openreview.net/forum?id=OJaWBhh61C.

Yongtai Liu, Joshua Maynez, Gonçalo Simões, and Shashi Narayan. Data augmentation for low-resource dialogue summarization. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), Findings of the Association for Computational Linguistics: NAACL 2022, pp. 703–710, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-naacl.53. URL https://aclanthology.org/2022.findings-naacl.53/.

Zhiwei Liu, Ziwei Fan, Yu Wang, and Philip S. Yu. Augmenting sequential recommendation with pseudo-prior items via reversely pre-training transformer. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '21. ACM, July 2021. doi: 10.1145/3404835.3463036. URL http://dx.doi.org/10.1145/3404835.3463036.

Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. Embedding-based news recommendation for millions of users. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '17, pp. 1933–1942, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450348874. doi: 10.1145/3097983.3098108. URL https://doi.org/10.1145/3097983.3098108.

Siru Ouyang, Jiaao Chen, Jiawei Han, and Diyi Yang. Compositional data augmentation for abstractive conversation summarization. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1471–1488, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.82. URL https://aclanthology.org/2023.acl-long.82/.

Sangwon Park, Hongseok Choi, Dongha Choi, and Hyunju Lee. GENDEX: Generative data augmentation strategy leveraging external data for abstractive dialogue summarization. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), Findings of the Association for Computational Linguistics: ACL 2024, pp. 3171–3185, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.188. URL https://aclanthology.org/2024.findings-acl.188/.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. A systematic survey of prompt engineering in large language models: Techniques and applications, 2024. URL https://arxiv.org/abs/2402.07927.

Gaurav Sahu, Olga Vechtomova, and Issam H. Laradji. A guide to effectively leveraging llms for low-resource text summarization: Data augmentation and semi-supervised approaches, 2025. URL https://arxiv.org/abs/2407.07341.

Yun-Zhu Song, Yi-Syuan Chen, Lu Wang, and Hong-Han Shuai. General then personal: Decoupling and pre-training for personalized headline generation. Transactions of the Association for Computational Linguistics, 11:1588–1607, 2023. doi: 10.1162/tacl_a_00621. URL https://aclanthology.org/2023.tacl-1.90/.

Md Arafat Sultan, Jatin Ganhotra, and Ramón Fernandez Astudillo. Structured chain-of-thought prompting for few-shot generation of content-grounded QA conversations. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), Findings of the Association for Computational Linguistics: EMNLP 2024, pp. 16172–16187, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.948. URL https://aclanthology.org/2024.findings-emnlp.948/.

Shichao Sun, Ruifeng Yuan, Ziqiang Cao, Wenjie Li, and Pengfei Liu. Prompt chaining or stepwise prompt? refinement in text summarization. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), Findings of the Association for Computational Linguistics: ACL 2024, pp. 7551–7558, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.449. URL https://aclanthology.org/2024.findings-acl.449/.

Maartje Ter Hoeve, Julia Kiseleva, and Maarten Rijke. What makes a good and useful summary? Incorporating users in automatic summarization research. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 46–75, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.4. URL https://aclanthology.org/2022.naacl-main.4.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. arXiv e-prints, pp. arXiv–2307, 2023.

Rahul Vansh, Darsh Rank, Sourish Dasgupta, and Tanmoy Chakraborty. Accuracy is not enough: Evaluating personalization in summarizers. In Findings of the Association for Computational Linguistics: EMNLP 2023, pp. 2582–2595, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.169. URL https://aclanthology.org/2023.findings-emnlp.169.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. TL;DR: Mining Reddit to learn automatic summarization. In Lu Wang, Jackie Chi Kit Cheung, Giuseppe Carenini, and Fei Liu (eds.), Proceedings of the Workshop on New Frontiers in Summarization, pp. 59–63, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4508. URL https://aclanthology.org/W17-4508.

Jianing Wang, Qiushi Sun, Xiang Li, and Ming Gao. Boosting language models reasoning with chain-of-knowledge prompting, 2024. URL https://arxiv.org/abs/2306.06427.

Jianling Wang, Ya Le, Bo Chang, Yuyan Wang, Ed H. Chi, and Minmin Chen. Learning to augment for casual user recommendation. In Proceedings of the ACM Web Conference 2022, WWW '22, pp. 2183–2194, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450390965. doi: 10.1145/3485447.3512147. URL https://doi.org/10.1145/3485447.3512147.

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

Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. Neural news recommendation with attentive multi-view learning. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence. International Joint Conferences on Artificial Intelligence Organization, 2019a.

Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. Neural news recommendation with multi-head self-attention. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing

(EMNLP-IJCNLP), pp. 6389–6394, Hong Kong, China, November 2019b. Association for Computational Linguistics. doi: 10.18653/v1/D19-1671. URL https://aclanthology.org/D19-1671.

Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. MIND: A large-scale dataset for news recommendation. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 3597–3606, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.331. URL https://aclanthology.org/2020.acl-main.331.

Jing Xiao, Weike Pan, and Zhong Ming. A generic behavior-aware data augmentation framework for sequential recommendation. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '24, pp. 1578–1588, New York, NY, USA, 2024a. Association for Computing Machinery. ISBN 9798400704314. doi: 10.1145/3626772.3657682. URL https://doi.org/10.1145/3626772.3657682.

Wen Xiao, Yujia Xie, Giuseppe Carenini, and Pengcheng He. Personalized abstractive summarization by tri-agent generation pipeline. In Yvette Graham and Matthew Purver (eds.), Findings of the Association for Computational Linguistics: EACL 2024, pp. 570–581, St. Julian's, Malta, March 2024b. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-eacl.39/.

Zhao Yang, Junhong Lian, and Xiang Ao. Fact-preserved personalized news headline generation. 2023 IEEE International Conference on Data Mining (ICDM), 2023.

Jiacheng Ye, Shansan Gong, Liheng Chen, Lin Zheng, Jiahui Gao, Han Shi, Chuan Wu, Xin Jiang, Zhenguo Li, Wei Bi, and Lingpeng Kong. Diffusion of thought: Chain-of-thought reasoning in diffusion language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), Advances in Neural Information Processing Systems, volume 37, pp. 105345–105374. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/be30024e7fa2c29cac7a6dafcbb8571f-Paper-Conference.pdf.

Mingjia Yin, Hao Wang, Wei Guo, Yong Liu, Suojuan Zhang, Sirui Zhao, Defu Lian, and Enhong Chen. Dataset regeneration for sequential recommendation. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '24, pp. 3954–3965. ACM, August 2024. doi: 10.1145/3637528.3671841. URL http://dx.doi.org/10.1145/3637528.3671841.

Lemei Zhang, Peng Liu, Marcus Tiedemann Oekland Henriksboe, Even W. Lauvrak, Jon Atle Gulla, and Heri Ramampiaro. Personalsum: A user-subjective guided personalized summarization dataset for large language models. In The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2024. URL https://openreview.net/forum?id=ETZk7lqyaF.

A Measuring Degree-of-Personalization

A.1 Motivation

Vansh et al. (2023) proposed EGISES– a metric to measure the degree of **in**sensitivity-to-subjectivity for relative benchmarking of how much models *lack personalization* (i.e., a lower score is better within the range $[0, 1]$) instead of assigning an absolute goodness score. Based on this notion, they defined (summary-level) "**deviation**" of a model $M_{\boldsymbol{\theta},u}$(later termed as ***Deg****ree-of-****R****esponsiven****ess*** (DEGRESS) by Dasgupta et al. (2024)) as follows:

Definition 3. ***Summary-level*** **DEGRESS.** *Given a document d_i and a user-profile u_{ij} (user j's expected summary), the summary-level responsiveness of a personalized model $M_{\boldsymbol{\theta},u}$, (i.e.,* DEGRESS$(s_{u_{ij}}|(d_i, u_{ij}))$*), is defined as the proportional divergence between model-generated summary $s_{u_{ij}}$ of d_i for j-th user from other user-specific summary versions w.r.t a corresponding divergence of u_{ij} from the other user-profiles.*

$\texttt{DEGRESS}(s_{u_{ij}}|(d_i, u_{ij}))$ is formulated as:

$$\begin{aligned}
&\texttt{DEGRESS}(s_{u_{ij}}|(d_i, u_{ij})) = \frac{1}{|\mathbf{U}_{d_i}|}\sum_{k=1}^{|\mathbf{U}_{d_i}|}\frac{min(X_{ijk}, Y_{ijk}) + \epsilon}{max(X_{ijk}, Y_{ijk}) + \epsilon}\\
&X_{ijk} = \frac{\exp(w(u_{ij}|u_{ik}))}{\sum_{l=1}^{|\mathbf{U}_{d_i}|}\exp(w(u_{ij}|u_{il}))}\cdot\sigma(u_{ij}, u_{ik});\ Y_{ijk} = \frac{\exp(w(s_{u_{ij}}|s_{u_{ik}}))}{\sum_{l=1}^{|\mathbf{U}_{d_i}|}\exp(w(s_{u_{ij}}|s_{u_{il}}))}\cdot\sigma(s_{u_{ij}}, s_{u_{ik}})\\
&w(u_{ij}|u_{ik}) = \frac{\sigma(u_{ij}, u_{ik})}{\sigma(u_{ij}, d_i)};\ \ w(s_{u_{ij}}|s_{u_{ik}}) = \frac{\sigma(s_{u_{ij}}, s_{u_{ik}})}{\sigma(s_{u_{ij}}, d_i)}
\end{aligned} \tag{7}$$

Here, $|\mathbf{D}|$ is the total number of documents in the evaluation dataset, $|\mathbf{U}|$ is the total number of users who created gold-reference summaries that reflect their expected summaries (and thereby, their subjective preferences), and $|\mathbf{U}_{d_i}|$ ($= |\mathbf{S}_{d_i}|$) is the number of users who created gold-references for document d_i. w is the divergence of the model-generated summary $s_{u_{ij}}$ (and the corresponding expected summary u_{ij}) from document d_i itself in comparison to all the other versions. It helps to determine how much percentage (therefore, the softmax function) of the divergence (i.e., $\sigma(s_{u_{ij}}, s_{u_{ik}})$ should be considered for the calculation of DEGRESS. If $s_{u_{ij}}$ is farther than $s_{u_{ik}}$ w.r.t d_i then $\texttt{DEGRESS}(s_{u_{ij}}|(d_i, u_{ij})) < \texttt{DEGRESS}(s_{u_{ik}}|(d_i, u_{ik}))$, implying that $M_{\boldsymbol{\theta},u}$ is more responsive to the k-th reader. A lower value of $\texttt{DEGRESS}(s_{u_{ij}}|(d_i, u_{ij}))$ indicates that while reader-profiles are different, the generated summary $s_{u_{ij}}$ is very similar to other reader-specific summaries (or vice versa), and hence, is not responsive at the summary-level. The system-level DEGRESS and EGISES have been formulated as follows:

$$\texttt{DEGRESS}(M_{\boldsymbol{\theta},u}) = \frac{\sum_{i=1}^{|\mathbf{D}|}\frac{\sum_{j=1}^{|\mathbf{U}_{d_i}|}\texttt{DEGRESS}(s_{u_{ij}}|(d_i, u_{ij}))}{|\mathbf{U}_{d_i}|}}{|\mathbf{D}|} \tag{8}$$

A.2 PerSEval: Formulation

As can be noted, the **DEGRESS formualtion does not enforce any penalty on accuracy drop**. To rectify this Dasgupta et al. (2024) proposed PerSEval. The design of PerSEval had two key goals: (i) to penalize models for poor accuracy, while simultaneously (ii) ensuring that the evaluation of responsiveness (i.e., DEGRESS) is not overshadowed by high accuracy. This penalty is referred to as the ***Effective DEGRESS Penalty Factor*** (EDP). If a model achieves 100% accuracy, no EDP will be applied, and the PerSEval score will equal the DEGRESS score. The following formulatiown of PerSEval guarantees these properties:

$$\begin{aligned}
&\texttt{PerSEval}(s_{u_{ij}}|(d_i, u_{ij})) = \texttt{DEGRESS}(s_{u_{ij}}|(d_i, u_{ij})) \times \texttt{EDP}(s_{u_{ij}}|(d_i, u_{ij}))\\
&\text{where, } \texttt{EDP}(s_{u_{ij}}|(d_i, u_{ij})) = 1 - \frac{1}{1 + 10^{\alpha\geq 3}\cdot\exp\left(-(10^{\beta\geq 1}\cdot(s_{u_{ij}}|(d_i, u_{ij})))\right)},\\
&(s_{u_{ij}}|(d_i, u_{ij})) = \texttt{ADP}(s_{u_{i*}}|(d_i, u_i*)) + \texttt{ACP}(s_{u_{ij}}|(d_i, u_{ij}))
\end{aligned} \tag{9}$$

Here, ADP is a document-level penalty due to a drop in accuracy for the best-performance of the model (i.e., the model-generated summary of document d_i ($s_{u_{ij}}$) is closest to the corresponding reader's expected summary u_{ij}). ADP is formulated as follows:

$$\begin{aligned}
&\texttt{ADP}(s_{u_{i*}}|(d_i, u_i*)) = \frac{1}{1 + 10^{\gamma\geq 4}\cdot\exp\left(-10\cdot\frac{\sigma^*(s_{u_{i\bullet}}, u_{i\bullet})|d_i - \mathbf{0}}{(\mathbf{1} - \sigma^*(s_{u_{i\bullet}}, u_{i\bullet})|d_i) + \epsilon}\right)}\\
&\text{where, } \sigma^*(s_{u_{i\bullet}}, u_{i\bullet})|d_i = \min_{j=1}^{|\mathbf{U}_{d_i}|}\sigma(s_{u_{ij}}, u_{ij})|d_i\\
&\text{and } \{\epsilon : \text{An infinitesimally small number} \in (0, 1)\}
\end{aligned} \tag{10}$$

ADP ensures that even if the DEGRESS score is acceptable, a penalty due to accuracy drop can still be imposed as a part of EDP. ADP, however, fails to address the scenario where the best-case scenario is acceptable (i.e., accuracy is fairly high) but is rather an outlier case – i.e., for most of the other model-generated summary versions, there is a considerable accuracy drop. To address this issue, the second penalty component within EDP called ***A**ccuracy-in**c**onsistency **P**enalty* (ACP) was introduced which evaluates whether a model consistently performs w.r.t accuracy for a specific generated summary compared to its average performance. ACPis formulated as:

$$\begin{aligned}\texttt{ACP}(s_{u_{ij}}|(d_i,u_{ij})) &= \frac{1}{1+10^{\gamma\geq 4}\cdot\exp\left(-10\cdot\frac{\sigma(s_{u_{ij}},u_{ij})|d_i-\sigma^*(s_{u_{i\bullet}},u_{i\bullet})|d_i}{(\overline{\sigma}(s_{u_{i\bullet}},u_{i\bullet})|d_i-\sigma^*(s_{u_{i\bullet}},u_{i\bullet})|d_i)+\epsilon}\right)} \\ \text{where, } \overline{\sigma}(s_{u_{i\bullet}},u_{i\bullet})|d_i &= \frac{1}{|\mathbf{U}_{d_i}|}\sum_{j=1}^{|\mathbf{U}_{d_i}|}\sigma(s_{u_{ij}},u_{ij})|d_i\end{aligned} \tag{11}$$

The system-level PerSEval score is as follows:

$$\texttt{PerSEval}(M_{\boldsymbol{\theta},u}) = \frac{\sum_{i=1}^{|\mathbf{D}|}\frac{\sum_{j=1}^{|\mathbf{U}_{d_i}|}\texttt{PerSEval}(s_{u_{ij}}|(d_i,u_{ij}))}{|\mathbf{U}_{d_i}|}}{|\mathbf{D}|} \tag{12}$$

The system-level PerSEval $\in [0,1]$ and is bounded by the system-level DEGRESS score.

PerSEval-RG-SU4. (or PSE-SU4) is the PerSEval variant that uses ROUGE-SU4 (Lin, 2004) as a distance metric (i.e., σ) in the PerSEval formula. PSE-SU4 has been reported to have high human-judgment correlation (Pearson's r: 0.6; Spearman's ρ: 0.6; Kendall's τ: 0.51) Dasgupta et al. (2024). The **ROUGE-SU4** score is based on *skip-bigrams*, which are pairs of words that appear in the same order within a sentence but can have up to four other words between them. The formula is as follows:

For a given generated summary G and reference summary R, the ROUGE-SU4 score is calculated as:

Skip-Bigram Recall (R_{SU4}):

$$R_{\text{SU4}} = \frac{\text{Count of matching skip-bigrams between } G \text{ and } R}{\text{Total skip-bigrams in } R}$$

Skip-Bigram Precision (P_{SU4}):

$$P_{\text{SU4}} = \frac{\text{Count of matching skip-bigrams between } G \text{ and } R}{\text{Total skip-bigrams in } G}$$

F1 Score ($F1_{SU4}$): The F1 score is the harmonic mean of precision and recall:

$$F1_{\text{SU4}} = \frac{2\times P_{\text{SU4}}\times R_{\text{SU4}}}{P_{\text{SU4}}+R_{\text{SU4}}}$$

Where:

- A **skip-bigram** consists of two words in the correct order but with zero to four words skipped in between.
- Matching skip-bigrams are counted between the generated summary and the reference summary.

The final **ROUGE-SU4** score is typically reported as the F1 measure, balancing precision and recall.

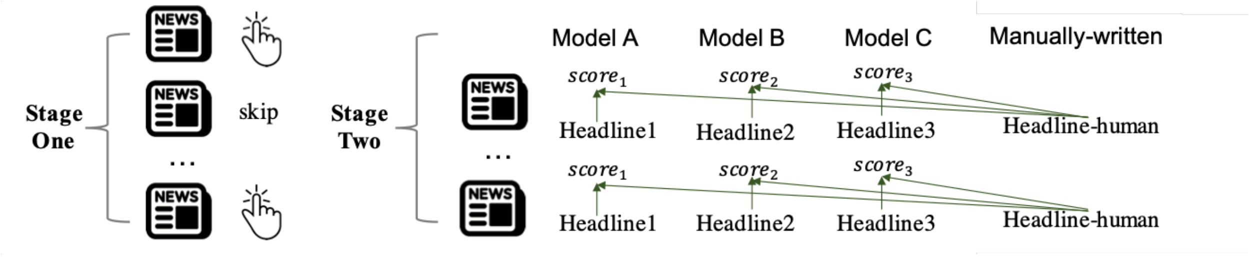


Figure 5: **Two stage PENS test data (original) creation** Ao et al. (2021): Stage 1 - Participants selected 50+ preferred headlines from 1,000 shown titles; Stage 2 - They rewrote headlines for 200 unseen articles using only news bodies, without seeing original titles.

B Dataset and Statistics

PENS The PENS dataset Ao et al. (2021) includes 113,762 news articles across 15 topics. Each article contains an ID, title (avg. 10.5 words), body (avg. 549 words), and category, with titles linked to the WikiData entities. The dataset also includes user interaction data, such as impressions and click behaviors, combined with news bodies and headlines from the MIND dataset Wu et al. (2020). For training, 500k user-news impressions were sampled from June 13 to July 3, 2019. Each log records user interaction as [uID, tmp, clkNews, uclkNews, clkedHis], where 'clkNews' and 'uclkNews' represent clicked and unclicked news, and 'clkedHis' refers to the user's prior clicked articles, sorted by click time. To create an offline testbed, 103 English-speaking students reviewed 1,000 headlines in stage-1, and then selected 50 articles, and created preferred headlines (i.e., expected gold-reference summaries) for 200 unseen articles in stage-2 (see Figure 5). Each article was reviewed by four participants. Editors checked for factual accuracy, discarding incorrect headlines. The high-quality remaining headlines serve as personalized gold-standard references in the PENS dataset. The PENS dataset has become the standard benchmark for personalized summarization task Ao et al. (2021); Song et al. (2023); Yang et al. (2023); Cai et al. (2023); Lian et al. (2025).The statistics of PENS dataset are given in Table 6.

OpenAI (Reddit). The OpenAI (Reddit) dataset Völske et al. (2017) comprises 123,169 Reddit posts collected from 29 distinct subreddits. This dataset provides both OpenAI-generated and human-written summaries and is organized into two splits: Comparisons, used for training and validation, and Axis, designated for validation and testing. A curated subset of 1,038 posts was processed by 13 different summarization policies, resulting in the generation of 7,713 summaries. These summaries underwent evaluation by 64 annotators who rated paired summaries based on selection preferences, confidence in their ratings, and dimensions such as accuracy, coherence, coverage, and overall quality (see Table 7 for details). Notably, unlike datasets like PENS, these summaries are not linked to individual annotators or their reading histories, which means they lack elements of personalization and contextual user information. The detailed statistics are given in Table 7.

C Baselines

Here, we discuss in details the baseline augmentation strategies, user-encoders and summarization frameworks.

C.1 Baseline Augmentations

We compare `PerAugy` with three SOTA algorithmic augmentation methods and three LLM-as-augmentors, and create corresponding UIGs for each of the augmentation methods. The statistical analysis of UIGs are given in Table 8. We describe each method as follows:

Characteristic	**Dimension**	**Value**
Article Stats		
General Stats	# Topics	15
	# Articles	113,762
	Avg. Title Length	10.5 words
	Avg. Body Length	549 words
Train Dataset Statistics		
Interaction Data	# User–News Impressions (anon.)	500,000
	# Users (anon.)	445,000
	Time Period	June 13–July 3, 2019
	User Interaction Fields	[uID, tmp, clkNews, uclkNews, clkedHis]
Test Dataset Statistics		
Participant Stats	# Participants	103
	Participant Category	English-speaking college students
	# Articles	3,940
	Browsed Headlines (Click + Skip)	1,000 per participant
	Min. Interested (Click) Headlines	50 per participant
Gold Reference (Participant-written Headlines)	Summarized Article Bodies	200 per participant
	Avg. Summaries per Article	4

Table 6: **PENS dataset (original) statistics**: Here the '*clkNews*' and '*uclkNews*' indicate clicked and un-clicked (i.e., skipped) news; '*clkedHis*' refers to the user's prior clicked articles, all sorted by click time; news bodies and headlines sourced from the MIND dataset Wu et al. (2020); Test dataset is created in two stages (see Figure 5).

PENS-SH. We choose PENS-SH Song et al. (2023), a SOTA PENS-based synthetic data generator for personalized summarization, as a comparative baseline w.r.t user-encoder accuracy boost. PENS-SH merges multiple (say, m) seed UIG trajectories $\{\tau^{u_{j=1:m}}\}$ from the PENS train dataset ($\mathcal{T}^{P}_{\text{train}}$) into a single synthetic trajectory $\tau^{\hat{u}_{1:m}}_{\text{P-SH}}$ such that all the common d-nodes on $\tau^{\hat{u}_{1:m}}_{\text{P-SH}}$ are unique to it, thereby forming the pool $\mathcal{T}^{\text{PSH}}_{\text{train}}$. We analyze the diversity of the PENS-SH trajectories in $\mathcal{T}^{\text{PSH}}_{\text{train}}$ when injected with s-nodes from the PENS test dataset, denoted $\mathcal{T}^{\text{syn-PSH}}$ as UIG of PENS-SH, following Section 3.2.

S3-Aug. We choose S3 (Segment-Shuffle-Stitch) Grover et al. (2024), a modular neural *intra-trajectory* augmentation mechanism designed for sequential data, as a comparative baseline. S3 restructures user interaction sequences τ^{u} by dividing them into n non-overlapping segments, followed by a differentiable shuffling and stitching operation on $\mathcal{T}^{P}_{\text{train}}$ to yield augmented trajectories $\tau^{\hat{u}_{1:m}}_{\text{S3}}$ that preserve local coherence while introducing temporal perturbations. The resulting S3-augmented trajectories $\mathcal{T}^{\text{S3}}_{\text{train}}$ are used to train user-encoders, and we evaluate its diversity after incorporating s-nodes from PENS testbed (denoting the UIG of it as $\mathcal{T}^{\text{syn-S3}}$) and further effect on user encoders as compared to `PerAugy`.

SDAInter. We include SDAInter Jiao et al. (2024) as a *cross-trajectory* augmentation baseline that generates pseudo user sequences by identifying interchangeable subsequences in $\mathcal{T}^{P}_{\text{train}}$ between different user trajectories based on shared anchor items. If the subsequences between two users meet a minimum IoU-based interchangeability confidence $C \geq T_c$, they are swapped to create new synthetic trajectories $\tau^{\hat{u}_{1:m}}_{\text{SDA}}$. The resulting SDAInter-augmented pool $\mathcal{T}^{\text{SDA}}_{\text{train}}$ is evaluated for its effect on user-encoder performance and compared against `PerAugy`, along with evaluation of $\mathcal{T}^{\text{syn-SDA}}$ (synthetic UIG version of SDAInter augmented trajectories by incorporating intermediate s-nodes) for `DegreeD`.

LLM-as-Augmentor. We also compare our method against three popular LLMs – Llama-2-13B Touvron et al. (2023), Mistral-v2-Instruct Jiang et al. (2023), and DeepSeek-7b-chat DeepSeek-AI et al. (2025) in two prompt-based settings - *1) Chain-of-Thoughts* Wei et al. (2023) which involves guiding an LLM through a series of logical reasoning steps to solve a task Wei et al. (2023). Using LLaMA-2-13B as the base model, we design a CoT prompt with detailed step-by-step instructions to logically generate a personalized summary

Characteristic	Dimension	Value
Dataset Overview		
General Stats	# Reddit Posts	123,169
	# Subreddits (Domains)	29
	Policy-Generated Summaries	115,579
	Human-Written Summaries	Available
Train + Validation Dataset Statistics		
Article Stats	# Reddit Posts	21,111
	# Policies	81
	# Generated Summaries	107,866
	# Annotators	76
	# Summary-Pairs Rated	64,832
Validation Subset Statistics		
Subset Details	# Reddit Posts	1,038
	# Policies	13
	# Generated Summaries	7,713
	# Annotators	32
Test Dataset (RLHF-Tuned Policies) Statistics		
Evaluation Stats	# Evaluated Policies	4
	# Evaluated Reddit Posts	57 (out of 1,038)
	Evaluation Method	Indirect Benchmarking
Annotation and Feedback		
Feedback Collection	Rating Scale	1–7
	Confidence Scale	1–9
	Avg. Ratings per Annotator	1,176
	Annotation Format	Summary-Pairs Selection

Table 7: **OpenAI TL;DR (Reddit) dataset statistics**: The dataset includes 123,169 Reddit posts across 29 subreddits, with policy-generated and human-written summaries. Evaluation involves summary-pair ratings and RLHF-tuned policy benchmarking.

based on a user's interaction history (see Figure 9), and *2) Prompt-Chaining* Sahoo et al. (2024) which is a technique that involves using a series of prompts to deconstruct a task into sequential steps. Our prompt-chaining setup consists of two sequential tasks. In the first task (step-1), the LLM generates user interactions in the form of (document, action) pairs, simulating user behavior (e.g., clicks, skips). In step-2, using few-shot prompting, the LLM generates a personalized summary, conditioned on both the input document and the user interactions generated in step-1. Each few-shot prompt contains four in-context examples (see Figure 10). These generated trajectories are then subsequently used to train user-encoder models (as described in Section 6.1.2).

C.2 Baseline User-Encoders

In this section We discuss SOTA news headline recommendation models which were used as baseline user-encoders to understand the effect of `PerAugy` generated training data.

NAML. Neural News Recommendation with Attentive Multi-View Learning (NAML) Wu et al. (2019a) is a neural news recommendation approach that learns informative representations of users and news by exploiting different kinds of news information. The core of this approach is a user-encoder and a news encoder where the news encoder learns unified news representations from titles, bodies, and topic categories by regarding them as different perspectives of the news, through an attentive multi-view learning model, and the user-encoder learns the representations of users based on their browsing history and applies attention mechanism to select informative news for user representation learning.

NRMS. Neural News Recommendation with Multi-Head Self-Attention (NRMS) Wu et al. (2019b) is a neural news recommendation where the core of the encoder is a news encoder that uses multi-head self-attention to learn news representations from news titles by modeling the interactions between words and a user-encoder which learns user representations from their browsed news and use multi-head self-attention to capture relatedness between the news. Additive attention to learn more informative news and user representations is used, by selecting important words and news.

EBNR. Embedding-based News Recommendation (EBNR) Okura et al. (2017) is an RNN-styled news recommendation approach that incorporates implicit negative user feedback by distinguishing positive and negative news clicks based on the reading dwell time of the news by the user, and learning the user representations from positive and negative samples via a combination of Transformer and additive attention network. It computes a final click score as a combination of positive click scores and negative click scores.

TrRMIo. Transfomer-based Recommendation Model Song et al. (2023) utilizes a personalized news recommendation model to represent users' preferences derived from clicked records. The pre-trained transformer models are used for both recommendation and headline generation tasks, where a news encoder and a user-encoder is adopted for content-based recommendation. The textual information from the news encoder is aggregated via Attention Pooling, which is then further integrated by user representation. The user interest is defined on the basis of Click Through Rate (CTR) by examining the frequency of news articles in users' click histories and positive samples. The assumption is that user history consists of both popular news and interested news, where popular news has a higher CTR ranking across the users, while interested news has lower CTR ranks, depicting the personalized choice of that user to generate the user representation.This laid the foundation for Transformer-based Recommendation Model Interest-Only(TrRMIo), where news histories with lower CTR for a particular user is treated as his 'Interest-Only' features.

C.3 Baseline Personalized Summarizers

To determine whether `PerAugy`-generated training data enhances the regularization of specialized Pretrained Language Models, improving `PerSEval` performance in personalized summarization tasks, we benchmark the PENS personalized summarization framework Ao et al. (2021) and the recent GTP personalized summarization framework Song et al. (2023). We describe each of the frameworks as follows:

PENS. The PENS framework employs a transformer-based encoder to process the news body and a pointer network–based decoder to generate headlines. The pointer mechanism is used to dynamically choose between generating words from a vocabulary and copying words directly from the news text, which helps in handling out-of-vocabulary words and maintaining factual consistency. To personalize the headline generator, three distinct injection strategies for incorporating user embeddings (*learned from user behavior data using state-of-the-art news recommendation models as user-encoders*) is proposed: (i) **Decoder Initialization** where the user embedding is used to initialize the decoder's hidden state, so the generation process is conditioned from the very start on the user's interests, (ii) **Attention Perturbation** where the user embedding is injected into the attention mechanism. This modulates the attention distribution over the news body words, effectively guiding the model to focus on parts of the text that align with the user's preferences, and (iii) **Generation-Copy Switch Adjustment** where the user embedding is also used to perturb the probability (or "switch") that determines whether the decoder generates a word from the vocabulary or copies a word from the news body. This helps ensure that the generated headline reflects personalized nuances rather than just summarizing the article content.

GTP. General Then Personal (GTP) is a framework that tackles personalized headline generation by decoupling the task into two sequential stages. In stage 1, a Transformer-based encoder–decoder model (e.g., BART) is pre-trained on large-scale news article–headline pairs to learn robust, content-focused headline generation without any personalization. In stage 2, a separate "headline customizer" takes the general headline and refines it by incorporating user-specific preferences. These preferences are encoded (control code) by the user-encoder TrRMIo. To bridge the gap between the general generation and personalized refinement, the authors introduce two key mechanisms: (i) **Information Self-Boosting (ISB)** that enhances the

UIG	**# u-nodes**	**# d-nodes**	**# s-nodes**	**Average**	**Maximum**	**DegreeD**
	(trajectories)	**per trajectory**	**per trajectory**	**trajectory length**	**trajectory length**	
PENS	360K+	83.78	1.94	85.72	2433	3
PENS-SH†	197K	310.56	1.01	321.58	5106	3
S3†	360K+	79.4	2.2	83.2	2167	3
SDAInter†	360K+	87.6	2.8	123.6	1742	3
LLaMA-2-13B†	2176	29	2	13.6	17	3
Mistral-7B†	5711	31.25	2.87	34.13	58	14
DeepSeek-R1†	813	37	4.23	32.7	35	4
OpenAI (Reddit) (OAI)	126K	25.19	4.82	30.02	54	4
PerAugy†∗	360K+	123.7	5.1	129.8	200	5
PerAugy-OAI∗	360K+	36.92	11.44	48.37	50	25

Table 8: **User-interaction graph statistics:** Two seed datasets chosen– PENS train dataset (Table 6) and OpenAI (Reddit) train dataset (Table 7); Baseline augmentation methods– (i) PENS-synthetic-base (ours; as *), (ii) PENS-SH, (iii) LLaMA-2-13B, (iv) Mistral-7B, (v) DeepSeek-R1, and (vi) PerAugy-PENS/OAI (ours); †augmentation followed by UIG abstraction on the PENS dataset.

customization by reintroducing relevant content details from the news article to ensure that personalization does not lead to information loss, and (ii) **Masked User Modeling (MUM)** that helps the model learn to recognize and utilize the user control code by randomly masking parts of the user embedding during training and then reconstructing them, thereby reducing over-reliance on the general model parameters.

D Encoder/Decoder Accuracy Metrics

In this section, we provide a detailed formulation of the user-encoder accuracy metrics in the context of the next d-node prediction task.

AUC. The Area Under the Curve (AUC) measures the probability that a randomly chosen positive d-node is ranked higher than a randomly chosen negative d-node in the test dataset $\mathcal{T}^{\mathrm{P}}_{\mathrm{test}}$. The formula is given as:

$$AUC = \frac{1}{|P| \cdot |N|} \sum_{p \in P} \sum_{n \in N} \mathbb{1}(s_p > s_n) \tag{13}$$

where:

- P is the set of positive interactions (set of clicked d-nodes of all users in test data).
- N is the set of negative items (set of skipped d-nodes of all users in test data).
- s_p is the predicted score for a positive d-node.
- s_n is the predicted score for a negative d-node.
- $\mathbb{1}(s_p > s_n)$ is an indicator function that equals 1 if $s_p > s_n$, otherwise 0.
- $|P|$ and $|N|$ are the number of positive and negative d-nodes, respectively.

MRR. Mean Reciprocal Rank (MRR) evaluates how early the ground-truth target d-node appears in the ranking. It is defined as:

$$MRR = \frac{1}{|U|} \sum_{u_j \in U} \frac{1}{\mathrm{rank}_{u_j}} \tag{14}$$

where:

- U is the set of users.
- rank_u is the position of the first relevant d-node for user u_j in the ranked recommendation list.

- $|U|$ is the total number of users.

A higher MRR indicates that the target d-node is ranked closer to the top of the prediction list, improving user experience.

nDCG@k. Normalized Discounted Cumulative Gain at rank k (nDCG@k) evaluates the ranking quality by considering both the prediction score and the position of ground-truth target d-node. It is defined as:

$$nDCG@k = \frac{DCG@k}{IDCG@k} \tag{15}$$

where:

$$DCG@k = \sum_{i=1}^{k} \frac{s_i}{\log_2(i+1)} \tag{16}$$

$$IDCG@k = \sum_{i=1}^{k} \frac{s_i^*}{\log_2(i+1)} \tag{17}$$

Here:

- s_i is the prediction score of the target d-node at rank i in the recommended list.
- s_i^* is the actual score of the target d-node at rank i in the ideal ranking (sorted by prediction score).
- $DCG@k$ is the Discounted Cumulative Gain up to rank k.
- $IDCG@k$ is the Ideal Discounted Cumulative Gain, representing the best possible ranking.

A higher nDCG@k indicates that the target d-node is ranked higher, improving prediction effectiveness.

E Algorithms

In this section, we discuss the details of the algorithms used in our paper.

UIG Construction We construct the User Interaction Graph (UIG) by parsing interaction logs from two types of seed datasets: (i) PENS-styled, where explicit click and skip behaviors are available, and (ii) OpenAI(Reddit)-styled, where user preferences are inferred from model confidence and summary quality ratings. The algorithm maps user interactions to document (d-node) and summary (s-node) nodes, while assigning appropriate behavioral edges such as `click`, `skip`, `gensum`, and `sumgen` to encode both explicit and inferred preferences.

Algorithm 1 UIG Construction

```
Require: train data and test data, dataset_type
  Initialize $\mathcal{T} \leftarrow \emptyset$
  for each user $u$ in train_data do
    Initialize $\tau^u \leftarrow \emptyset$
    for each interaction in $u$'s data do
      if dataset_type = PENS then
        Map interaction to d-node with click/skip edge
      else if dataset_type = OPENAI then
        if any model-generated summary for d-node has confidence score $\geq$ threshold then
          Label as click
          Select best-rated summary by $u$ as surrogate s-node
          Map to d-node and s-node with gensum and sumgen edges
        else
          Label as skip
        end if
      else
        Map rating to d-node with click/skip edge
        if rating is max then
          Map to d-node and s-node with gensum and sumgen edges
        end if
      end if
      Append d-node to $\tau^u$
    end for
    Add $\tau^u$ to $\mathcal{T}$
  end for
  if dataset_type = PENS then
    for each $\tau^u$ in $\mathcal{T}$ do
      if d $\in$ train data AND d $\in$ test data then
        Insert (d-s)-nodes from test_data as genSumm/sumGen edges
      end if
    end for
    return $\mathcal{T}^{\text{PENS-D}} \leftarrow \mathcal{T}$
  else
    return $\mathcal{T}$
  end if=0
```

DegreeD Computation To quantify the *degree of diversity* in user interaction logs, we introduce the DegreeD score—a metric that captures the evolving nature of user preferences over time and evaluates how personalized content (such as summaries) aligns with prior behaviors. The algorithm (Algorithm 2) operates over user trajectories, each consisting of a sequence of actions (click, skip, gen_summ) over documents and associated summaries. As the trajectory is processed, document-level similarity scores (σ) are tracked across interactions to compute a moving average of document distances (D_{MA}), which reflects a user's short-term memory of consumed content. When a generation action occurs, the algorithm compares the newly generated document embedding (D_{t_1}) with the recent average (D_{t_2}), computing a diversity coefficient $\delta = \frac{\min(D_{t_1}, D_{t_2})+\epsilon}{\max(D_{t_1}, D_{t_2})+\epsilon}$ based on their normalized overlap. This coefficient is further modulated by a personalization factor $P = \frac{\sigma(D_{t_1}, U_{t_1})}{\sigma(D_{t_2}, U_{t_2})+\epsilon}$, which assesses how well the current document aligns with the user embedding compared to the historical context, and is weighted by a temporal user similarity $\sigma(U_{t_2}, U_{t_1})$ to reflect behavioral consistency over time. The product $\delta \cdot P \cdot \sigma(U_{t_2}, U_{t_1})$ contributes to an accumulated diversity score for each user trajectory. Finally, the total DegreeD score is averaged across all users, offering a global measure of behavioral richness and personalization diversity in the training dataset.

Algorithm 2 Computing DegreeD

```
Input: Users $U$, Actions $A$, Summaries $S$, Documents $D$, window size $w$
for each trajectory $(U, A, D, S)$ do
    $\mathcal{D}_{\text{total}} = 0$, $D_{\text{dist}} = []$, $D_{\text{MA}} = 0$
    for $t_1 = 1$ to $|A| - 1$ do
        Retrieve $D_{t_1}, U_{t_1}$
        if $A_{t_1}$ is click/skip then
            $D_{\text{dist}} \leftarrow \sigma(D_{t_1}, D_{\text{prev}})$, update $D_{\text{MA}}$
        else if $A_{t_1}$ = gen_summ then
            $D_{t_2} = D_{\text{MA}}$, $\delta = \frac{\min(D_{t_1},D_{t_2})+\epsilon}{\max(D_{t_1},D_{t_2})+\epsilon}$
            $P = \frac{\sigma(D_{t_1},U_{t_1})}{\sigma(D_{t_2},U_{t_2})+\epsilon}$
            $\mathcal{D}_{\text{total}} + = \delta \cdot P \cdot \sigma(U_{t_2}, U_{t_1})$
        end if
    end for
    $\mathcal{D} + = \mathcal{D}_{\text{total}}/(|A| - 1)$
end for
$\mathcal{D} \leftarrow \mathcal{D}/|U|$ =0
```

`PerAugy` Framework The `PerAugy` framework enhances user interaction generalization by introducing two complementary augmentation strategies: Double Shuffling (DS) and Stochastic Markovian Perturbation (SMP). In DS (Algorithm 3), target user trajectories are systematically altered by substituting randomly selected segments from other users' interaction histories. These substitutions occur at randomized offsets and are spaced by controlled gap lengths to maintain temporal realism and simulate cross-user behavioral blending. This process generates diversified yet structurally plausible trajectories, expanding the training distribution without deviating from feasible user behavior patterns. Following DS, the SMP stage (Algorithm 4) further refines the augmented trajectories by focusing on semantic consistency at the summary level. Specifically, newly introduced summaries are evaluated within a local Markovian window of recent document interactions, and replaced with top-ranked candidates based on a relevance score computed via RMSD (Root Mean Square Distance) similarity. These candidates are weighted by an exponential temporal decay factor to prioritize more recent contextual nodes, ensuring that substituted summaries align with short-term user interest profiles. Together, DS and SMP act in synergy to produce coherent, high-quality augmented data that preserves personalization cues while introducing controlled variability—a crucial property for training robust and generalizable user models in recommendation and summarization tasks.

F Implementation Details

Computing Resources. The creation of User Interaction Graphs (UIGs) and computation of `DegreeD` are performed on a standard 4-core CPU with 16GB of RAM. For Stochastic Markovian Perturbation (SMP), we use the SBERT all-MiniLM-L6-v2 model Reimers & Gurevych (2019) to generate embeddings, and the SMP process takes approximately 16 hours to complete on an NVIDIA A-100 GPU. LLM-based experiments are conducted using 3 NVIDIA A-100 GPUs.

`PerAugy` Settings. We generate the embeddings of the d-nodes and s-nodes using the SBERT Reimers & Gurevych (2019) 'all-MiniLM-L6-v2 model', which has 22.7M parameters and an embedding size of 384. Manhattan distance is used to compute embedding divergence during the `DegreeD` calculation, chosen for its linear scalability and efficiency. For SMP, embeddings of sentences and d-nodes within the context window are generated using the same SBERT model, followed by the use of RMSD to compute similarity scores for perturbation.

Model Settings. Three user encoders (NAML Wu et al. (2019a), EBNR Okura et al. (2017), NRMS Wu et al. (2019b)) are trained on `PerAugy` datasets for 2 epochs, with a learning rate of 0.0001 and batch size 128 using the Adam optimizer. The models are finetuned on the $\mathcal{T}^{\mathcal{E}}_{\text{DS/DS+SMP}}$ datasets in the TrRMIo model Song

Algorithm 3 Double Shuffling (DS)

```
Require: A UIG trajectory pool $\mathcal{T}_{\text{base}}^{syn}$, sample size $m$and gap-length $g_l$
Ensure: Modified trajectory set $\mathcal{T}_{\text{DS}}^{m}$
$\mathcal{T}_{\text{sample}}^{m} \leftarrow \text{SampleWithoutReplacement}(\mathcal{T}_{\text{base}}^{syn}, m)$
$\mathcal{T}_{\text{DS}}^{m} \leftarrow \emptyset$
for each target trajectory $\tau_{\text{target}}^{u_j} \in \mathcal{T}_{\text{sample}}^{m}$ do
    $O \leftarrow \textbf{RandomOffset}()$
    $I_{\text{subs}} \leftarrow O$
    for each source trajectory $\tau_{\text{source}}^{u_i} \in \mathcal{T}_{\text{sample}}^{m}$, where $i \neq j$ do
        $\tau_{seg}^{u_i} \leftarrow \textbf{RandomSegment}(\tau_{\text{source}}^{u_i})$ {Select a trajectory segment of random length at random time-steps.}
        $\tau_{\text{target}}^{u_j} \leftarrow \textbf{Substitute}(\tau_{\text{target}}^{u_j}, \tau_{seg}^{u_i}, I_{\text{subs}})$
        $I_{\text{subs}} \leftarrow O + \textit{length}(\tau_{seg}^{u_i}) + g_l$ {Determine substitution indices in $\tau_{\text{target}}^{u_j}$, ensuring that two source segments are separated by gap-length $g_l$.}
    end for
    $\tau_{\text{DS}}^{u_j} \leftarrow \tau_{\text{target}}^{u_j}$
    $\mathcal{T}_{\text{DS}}^{m} \leftarrow \mathcal{T}_{\text{DS}}^{m} \cup \{\tau_{\text{DS}}^{u_j}\}$
end for
return $\mathcal{T}_{\text{DS}}^{m}$ =0
```

Algorithm 4 Stochastic Markovian Perturbation (SMP)

```
Require: DS trajectories $\mathcal{T}_{\text{DS}}^{m}$, window $k$, decay $\lambda$, top-$p$
Ensure: perturbed set $\mathcal{T}_{\text{SMP}}^{m}$
$\mathcal{T}_{\text{SMP}} \leftarrow \emptyset$
for each $\tau \in \mathcal{T}_{\text{DS}}^{m}$ do
    for each step $t$ in $\tau$ do
        if $s^{(t)}$ newly substituted then
            Retrieve $d^{(t-1)}$, extract $\{st\}$, define window $\{c\}$
            for each $st \in d^{(t-1)}$ do
                $I(st) \leftarrow \sum_c \text{RMSD}(st, c)\, e^{-\lambda\, pos(c)}$
            end for
            Rank $\{st\}$, pick top-$p$ $\hat{s}^{(t)}$, replace $s^{(t)}$
        end if
    end for
    $\mathcal{T}_{\text{SMP}} \leftarrow \mathcal{T}_{\text{SMP}} \cup \{\tau\}$
end for
return $\mathcal{T}_{\text{SMP}}$ =0
```

et al. (2023) for one epoch after training from scratch. During training, intermediate s-nodes are modeled as d-nodes to integrate them into the user encoders.

LLM Settings for Prompts. Prompting experiments are conducted using two setups: (1) Chain-of-Thoughts with LLaMa2-13B and (2) Prompt-Chaining with Mistral-Instruct-v2 and DeepSeek-7B-Chat. For LLaMa2-13B, we perform inference using sampling with temperature set to 0.75, top-p to 0.9, and top-k to 50. For Mistral-Instruct-v2 and DeepSeek-7B-Chat, we use a deterministic sampling strategy (temperature = 0.0, top-p = 1.0) for controlled generations. Max_tokens are set to 1024 for both setups.

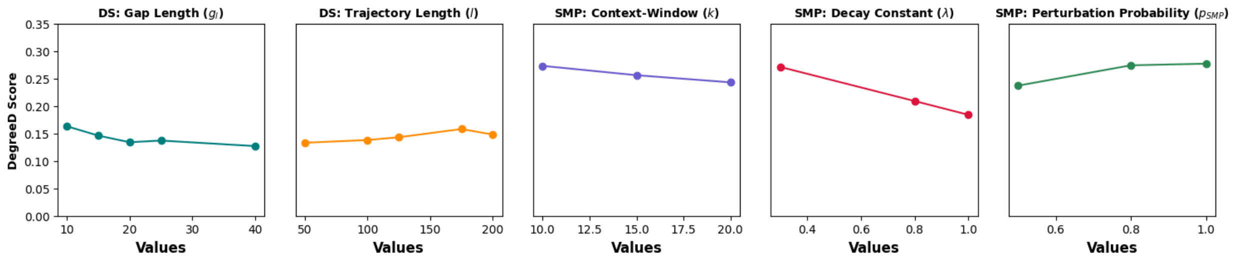


Figure 6: **Ablation (RQ-1) effect of hyper-parameters on DegreeD:** Diversity analysis for all hyper-parameters of PerAugy; **Observation-1**: *lower gap length increases diversity due to diffusion into different topics across a trajectory*; **Observation-2**: *Longer context window may not lead to sufficient perturbation*; **Observation-3**: *stricter Markovian does not yield higher diversity*; & **Observation 4**: *frequent SMP on s-nodes lead to higher diversity.*

G Detailed Results

G.1 DegreeD Ablations (RQ-1)

We ablate various hyperparameters of PerAugy, including gap length g_l and trajectory length l for $\mathcal{T}_{\mathrm{DS}}^{\mathcal{E}-\mathrm{P}}$, as well as context length k, decay constant λ, and perturbation probability p_{SMP} for $\mathcal{T}_{\mathrm{DS+SMP}}^{\mathcal{E}-\mathrm{P}}$. A summary of our findings is given in Figure 6.

Gap Length g_l and Trajectory Length l. Smaller values of g_l generally lead to higher DegreeD, with $g_l = 10$ yielding the best results (DegreeD of 0.163). This suggests that *frequent substitutions in 'source' segments boost thematic divergence.* Similarly, increasing l results in higher DegreeD, with the highest score (0.158) observed at $l = 175$. This indicates that *the length of 'source' segments plays a crucial role in promoting diversity.*

SMP Parameters. We observe that smaller values of the context window k and the decay constant λ lead to higher DegreeD, while higher p_{SMP} improves DegreeD, with the optimal setup being $k = 10$, $\lambda = 0.3$, and $p_{SMP} = 1$ (with a score of 0.278). This suggests that, while there is a *Markovian effect* (as a lower k results in higher diversity), the role of *higher-order influence* should not be overlooked. In other words, user-generated subjective summaries are not solely governed by a Markovian process (as evident from the fact that lower λ corresponds to higher DegreeD), and might have long-term dependencies. We conclude that user behavior exhibits a tendency toward *diffusion* (random or exploratory variation in a user's reading behavior); however, an abrupt diffusion does not necessarily lead to higher diversity. This underscores the importance of *SMP as a smoothing mechanism to regulate diffusion.* The comparative analysis of the ablations on different hyperparameters w.r.t DegreeD are in Table 9.

G.2 Ablation Studies (RQ-2)

We ablate on the mix training data $\mathcal{T}_{\mathrm{DS}}^{\mathcal{E}-\mathrm{P}}$ to analyze the effect of DS hyper-parameters– gap-length g_l (section 6.1.1) and train history-segment length $\tau_{h_{\mathrm{train}}}$: $\{l/2, 5l/8, 3l/4, 7l/8, l-3\}$ (l: trajectory length). For SMP hyper-parameters (k: $\{10, 15, 20\}$, λ: $\{0.3, 0.8, 1\}$, p_{SMP}: $\{0.5, 0.8, 1\}$), we ablate on $\mathcal{T}_{\mathrm{DS+SMP}}^{\mathcal{E}-\mathrm{P}}$. Results are in Figure 8.

Effect of $\tau_{h_{\mathbf{train}}}$ & g_l We fix g_l to 25 and observe that $\tau_{h_{\mathrm{train}}}$ has a major impact across all user-encoders with the longest $\tau_{h_{\mathrm{train}}}$ ($l-3$) having the highest mean boost (0.064 ↑ w.r.t AUC, 0.035 ↑ w.r.t MRR, 0.011 ↑ w.r.t nDCG@5/10) against the least scores, thereby confirming that longer preference history in train is better. g_l (l fixed at 150) also matters particularly w.r.t AUC with best at 40. This shows that synthetic profiles having longer original user segments are better.

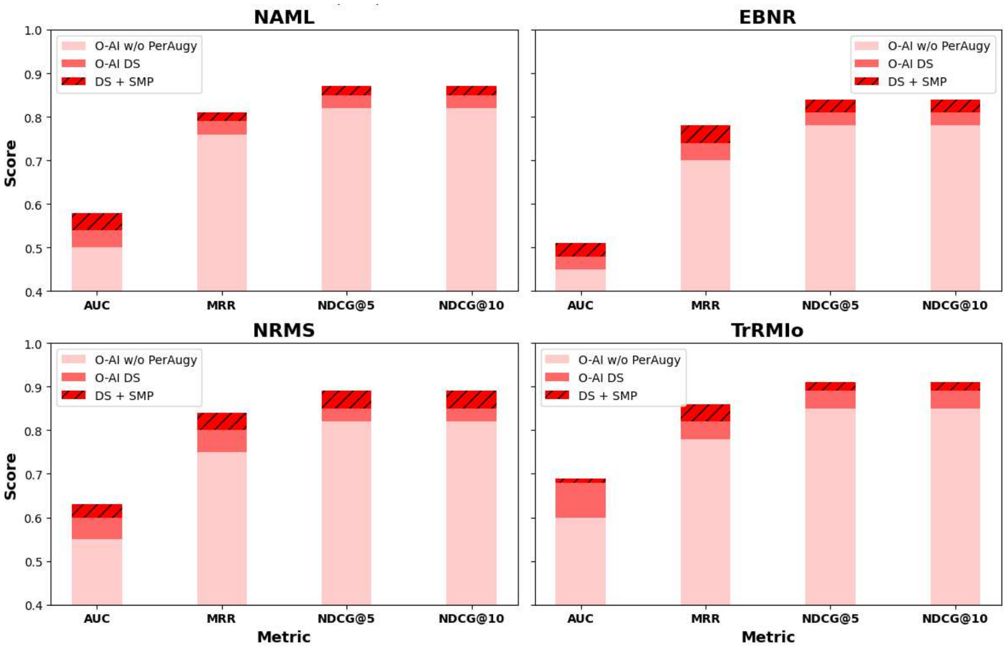


Figure 7: **User-encoder performance (OpenAI (Reddit)):** Impact of Double Shuffling (DS) and DS+SMP on OpenAI seed base $\mathcal{T}^{\text{OAI}}_{\text{base}}$ (SMP hyper-parameters: k=10, λ=0.3, p_{SMP}=0.8). TrRMIo is finetuned; others are trained from scratch. **Observation-1:** *Vanilla OpenAI lags behind DS and DS+SMP, indicating that random augmentation is less effective than* `PerAugy`*;* **Observation-2:** *DS achieves performance boosts as in* `PerAugy`*-PENS;* **Observation-3:** *DS+SMP further boosts performance, demonstrating the cross-domain strength of* `PerAugy`*.*

Hyper-parameter	AUC		MRR		nDCG@5		nDCG@10	
	RWM	AWM	RWM	AWM	RWM	AWM	RWM	AWM
g_l	0.018	0.043	0.005	0.021	0.007	0.023	0.007	0.018
$\tau_{h_{train}}$	0.024	**0.064**	0.013	0.035	0.011	0.026	0.011	0.028
k	**0.028**	0.049	0.016	0.027	0.016	0.032	0.016	0.032
λ	0.019	0.034	**0.023**	**0.047**	0.01	0.029	0.01	0.029
p_{SMP}	0.019	0.04	0.018	0.039	**0.023**	**0.035**	**0.018**	**0.04**

Table 9: **Comparative impact of hyper-parameters.** Metrics shown are Relative Win Margin (RWM) and Absolute Win Margin (AWM). **Observation:** *Shorter gap-length leads to consistent wins across encoders w.r.t AUC, but for prediction ranking, higher perturbation probability and context-window length matter more.*

Effect of $k, \lambda,$ & p_{SMP} We observe that p_{SMP} has the maximum impact (fixing $k = 10; \lambda = 0.5$), particularly for ranking metrics (MRR, nDCG@5/10). We find that $p_{\text{SMP}} = 0.8/1$ have highest boost (0.04 ↑ w.r.t nDCG@10). This shows that SMP smoothing is mostly required during augmentation. We also observe that the length of the context window ($\tau^{u_{\text{target}}}_{c_k}; \lambda = 0.5; p_{\text{SMP}} = 0.8$) also has a significant effect on the overall AUC (0.05 ↑) and nDCG@5/10 (0.032 ↑) with the best at $k = 10$. With the best k and p_{SMP} (0.8), we find the best λ to be 0.3, particularly for MRR (0.047 ↑) and nDCG@5 (0.032 ↑). This shows that (a) long context window is not useful for SMP smoothing and (b) smoothing cannot be strictly Markovian.

Comparative Impact of Hyperparameters. We conduct a detailed ablation to understand the influence of each hyperparameter of `PerAugy` on user-encoder model performance across AUC, MRR, nDCG@5 & 10 using *Relative Win Margin* (the difference between the best and second-best performance for a hyperparameter) and *Absolute Win Margin* (difference between best and worst). History Length exhibits the strongest effect on AUC with an Absolute Win Margin of 0.064 and a Relative Win Margin of 0.024, indicating that *longer*

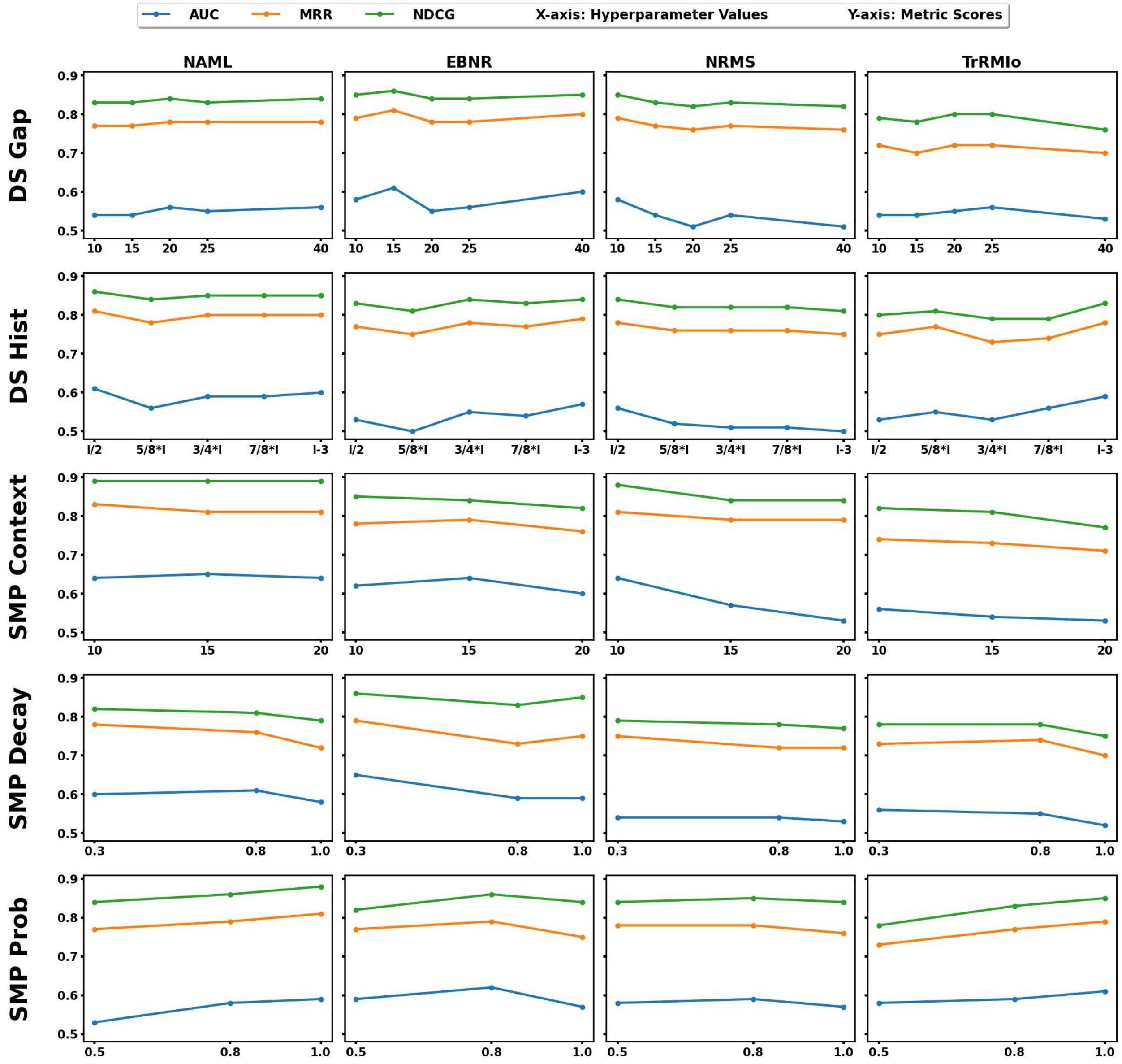


Figure 8: **Effect of `PerAugy` hyper-parameters on User-Encoder Accuracy:** All encoder models are *trained-from-scratch*; results summarized in Table 9. **Observation-1:** *Best hyper-parameter values perform consistently across models;* **Observation-2:** *For DS,* $g_l = 40$ *and* $\tau_{h_{train}} = l - 3$ *favor longer profile/history retention;* **Observation-3:** *For SMP,* $k = 10$, $p_{SMP} = 0.8$, $\lambda = 0.3$ *control abrupt diffusion best, and non-Markovian smoothing is preferred.*

historical context is crucial for general user modeling. Context-Length k provides the highest Relative Win Margin of 0.028 for AUC and consistent gains across all ranking metrics, showing the *importance of larger context windows.* Decay Constant λ achieves the highest MRR Absolute Win Margin of 0.047 and a strong Relative Win Margin of 0.023, highlighting the *impact of temporal recency weighting.* Perturbation Probability p_{SMP} leads in ranking metrics, with nDCG@5 Relative=0.023, Absolute=0.035 and nDCG@10 Absolute=0.04, suggesting *higher perturbation improves top-k relevance.* Finally, Gap Length contributes stable gains to AUC (Abs.=0.043), indicating that *shorter gaps between actions lead to consistent encoder performance.* Overall, each hyperparameter uniquely benefits different objectives, and *careful tuning is vital for optimal performance.* The detailed results are in Table 9.

H Comparative Study

In this section, we provide a comparative analysis of SOTA relatable data augmentation techniques based on the operations they perform on trajectory-like datasets. In terms of operations, the methods are divided into two main categories: Intra-Trajectory Augmentation and Cross-Trajectory Augmentation, serving the purpose of sequential recommendation tasks.

H.1 Intra-Trajectory Augmentation

S3 Grover et al. (2024): Segment-Shuffle-and-Stitch is an intra-trajectory augmentation where non-overlapping segments within same trajectory sequences are segmented, shuffled and finally concatenated (stitched) to form a new optimal trajectory. Since our goal is to generate diverse synthetic trajectories, shuffling among the segments within same sequence does not guarantee a smooth thematic transition of s-nodes w.r.t the historical interactions. Also, in our case, Shuffling does not enhance diversity w.r.t. `DegreeD`.

MBASR Xiao et al. (2024a): Multi-behavior Augmentation for Sequential Recommendation method employs an intra-trajectory augmentation technique by performing pairwise swapping of segments to generate diversity. However, in our case, the base historical dataset has highly monotonous trajectories. Therefore swapping nearby subsequences fails to produce sufficient diversity, and additional operations like order perturbation or redundancy reduction do not effectively smoothen the s-node content in line with historical preferences and might inject (or remove) unrealistic time-step information thereby disrupting the flow of the trajectory. For these reasons, we do not adopt MBASR.

STEAM Lin et al. (2023): STEAM operates in an intra-trajectory manner by deciding whether to drop or insert nodes within a trajectory to create augmented data. However, the method is not scalable to longer trajectories, and the insertion or deletion of nodes can disrupt the historical sequence, ultimately undermining the realistic flow of the synthetic user profiles. Hence, we do not use STEAM.

L2Aug Wang et al. (2022): Learning-to-Augment is an intra-trajectory augmentation method where a node is deleted from the sequence of core users to generate sequence of synthetic casual users through a reinforcement learning-based policy mechanism. Node deletion is irrelevant in our case as it can disrupt the sequential flow of the trajectories.

BTBR Li et al. (2023): Bi-directional Transformer Basket Recommendation model incorporates masking strategies and swapping operation to train the model for 'Next Novel Basket Recommendation'. Despite some similarity in the purpose at broader level, our goal is not to create a model/encoder that encodes the input sequence but to generate a diverse input sequence to make the existing encoders learn the representations.

H.2 Cross-Trajectory Augmentation

SDAinter Jiao et al. (2024): SDAinter is a cross-trajectory technique that matches anchor items (e.g., identical start and end d-nodes or s-nodes) across trajectories to facilitate segment exchange. However, the reliance on anchor-based matching does not effectively capture the subjective nuances of individual user interests, limiting its applicability in personalized summarization. For this reason, we do not consider SDAinter suitable for historical interaction sequence-based tasks.

DR4SR Yin et al. (2024): Data-Regeneration-for-Sequential-Recommendation is a transformer-styled cross-trajectory sequence regeneration model where the pertaining task is constructed for to extract patterns from given set of sequences and feed the patterns to the model to regenerate other set of possible sequences. However, patterns in our case would mean reading and summarizing habits of two different users, where the s-nodes are subjective. Therefore, this technique might incorporate redundant s-nodes, defeating the goal of personalized summarization.

TiCoSeRec Dang et al. (2024): Time Interval Aware Augmentation technique ensures uniform time-interval distribution in the sequence based on the time-aware traditional operations like Crop, Mask, Insert, Reorder and Substitute. However, our trajectories are primarily assumed to be uniform in terms of time-steps (unit time between two successive interactions).

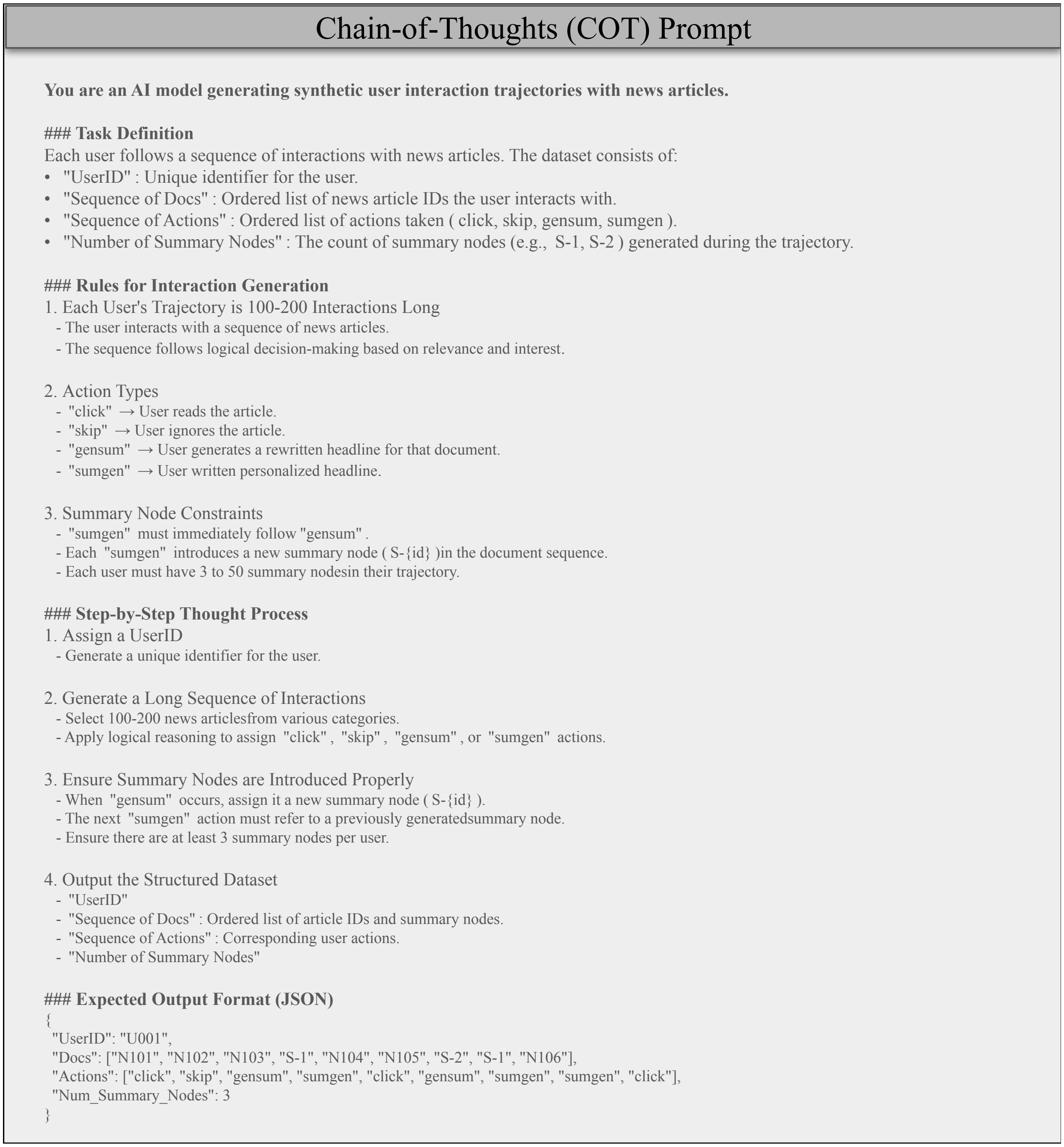
FDA Chen et al. (2023a): Fairness-oriented Data Augmentation is used to generate synthetic user profiles from the realistic profiles to balance between realistic data and pseudo pseudo data. However, modeling historical preference trajectories by generating fake interaction sequences will not lead to diversified trajectories as the 'complemented' sequence will also remain monotonous. Also, ideal datasets for personalized summarization tasks must have intermediate summary nodes for supervised learning setup, which makes the generation of fake interactions challenging.

divSPA-styled methods Liu et al. (2023): These methods use a cross-trajectory augmentation strategy by exchanging segments between trajectories based on similarity metrics. Despite this, the exchanged segments often lack sufficient variation with respect to the overall degree (`DegreeD`), resulting in minimal diversity gains. This limitation makes the approach less effective for our needs.

I Prompt Details

Chain-of-Thoughts. Chain-of-thought (CoT) prompting is a powerful technique that guides large language models to decompose complex problems into a series of intermediate reasoning steps before emitting their final answer. CoT prompting was shown to significantly improve multi-step arithmetic, commonsense, and symbolic reasoning tasks by eliciting explicit rationale chains that mirror human logic Ye et al. (2024). Subsequent work in ACL demonstrated that CoT can be extended to address hallucination and faithfulness issues by injecting structured knowledge during rationale generation Wang et al. (2024). EMNLP findings further confirmed that structured CoT variations, such as state-based prompting, yield substantial gains in content-grounded dialogue systems by promoting intermediate subtask decomposition Sultan et al. (2024). Another EMNLP study introduced prompt tuning of masked language models to generate both intermediate and final reasoning steps jointly, striking a balance between interpretability and performance without full fine-tuning Kunnath et al. (2023). Across these efforts, a consistent insight is that CoT acts as a bridge, enabling LLMs to expose latent reasoning processes. The method is particularly effective for tasks demanding logical coherence and multi-hop inference, such as math word problems and question answering. Although CoT relies on large model scale to be effective, research shows that even generated exemplars (e.g. "Let's think step by step") can approximate few-shot behavior. In the context of our work, CoT prompting with LLaMA-2-13B is used to craft personalized user summaries by breaking down interactions step by step, so as to enhance transparency and accuracy, making LLM reasoning more interpretable and reliable.

Prompt Chaining. Prompt chaining is an effective prompting strategy where a complex task is decomposed into a sequence of smaller, well-defined prompts, with the output of one prompt becoming the input to the next, thereby guiding the model through a structured reasoning pipeline. It improves performance on multi-step tasks by reducing cognitive load on the model and increasing transparency at each stage—developers can verify and debug intermediate outputs, enhancing controllability and reliability. Academic research, such as , shows prompt chaining excels in iterative summarization by orchestrating drafting, critiquing, and refining phases via discrete prompts, outperforming one-shot or stepwise alternatives Sun et al. (2024).Across these efforts, the core insight is that chaining leverages the model's strengths at each subtask rather than relying on single-shot reasoning, leading to superior performance, especially when task complexity or input length is high. In our context, prompt chaining is implemented via two steps—user behavior simulation followed by summary generation—mirroring the alidated pattern of decomposition for enhanced LLM task performance.



Chain-of-Thoughts (COT) Prompt

You are an AI model generating synthetic user interaction trajectories with news articles.

Task Definition

Each user follows a sequence of interactions with news articles. The dataset consists of:
- "UserID" : Unique identifier for the user.
- "Sequence of Docs" : Ordered list of news article IDs the user interacts with.
- "Sequence of Actions" : Ordered list of actions taken (click, skip, gensum, sumgen).
- "Number of Summary Nodes" : The count of summary nodes (e.g., S-1, S-2) generated during the trajectory.

Rules for Interaction Generation

1. Each User's Trajectory is 100-200 Interactions Long
   - The user interacts with a sequence of news articles.
   - The sequence follows logical decision-making based on relevance and interest.

2. Action Types
   - "click" → User reads the article.
   - "skip" → User ignores the article.
   - "gensum" → User generates a rewritten headline for that document.
   - "sumgen" → User written personalized headline.

3. Summary Node Constraints
   - "sumgen" must immediately follow "gensum" .
   - Each "sumgen" introduces a new summary node (S-{id})in the document sequence.
   - Each user must have 3 to 50 summary nodesin their trajectory.

Step-by-Step Thought Process

1. Assign a UserID
   - Generate a unique identifier for the user.

2. Generate a Long Sequence of Interactions
   - Select 100-200 news articlesfrom various categories.
   - Apply logical reasoning to assign "click" , "skip" , "gensum" , or "sumgen" actions.

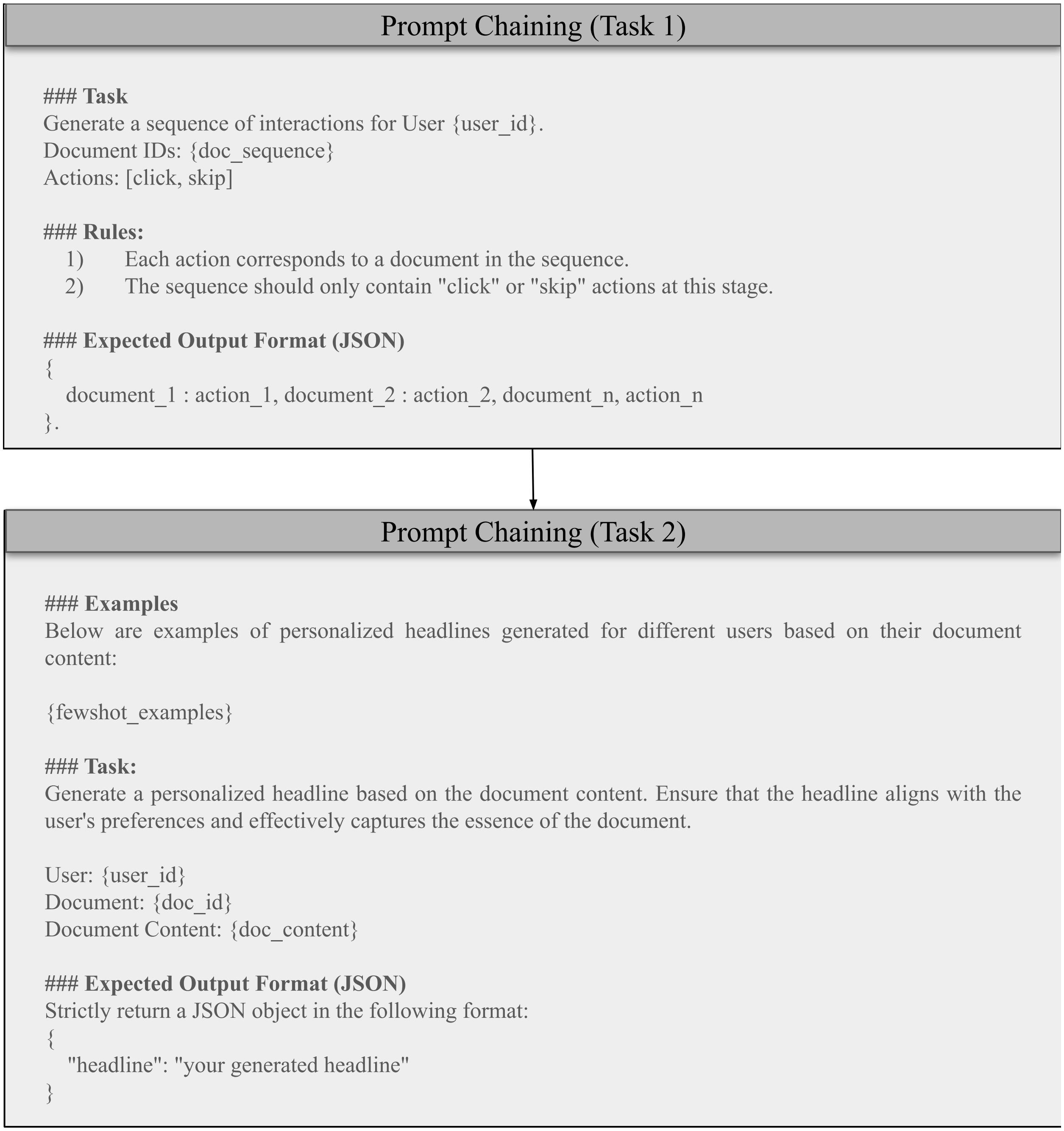
3. Ensure Summary Nodes are Introduced Properly
   - When "gensum" occurs, assign it a new summary node (S-{id}).
   - The next "sumgen" action must refer to a previously generatedsummary node.
   - Ensure there are at least 3 summary nodes per user.

4. Output the Structured Dataset
   - "UserID"
   - "Sequence of Docs" : Ordered list of article IDs and summary nodes.
   - "Sequence of Actions" : Corresponding user actions.
   - "Number of Summary Nodes"

Expected Output Format (JSON)

```
{
 "UserID": "U001",
 "Docs": ["N101", "N102", "N103", "S-1", "N104", "N105", "S-2", "S-1", "N106"],
 "Actions": ["click", "skip", "gensum", "sumgen", "click", "gensum", "sumgen", "sumgen", "click"],
 "Num_Summary_Nodes": 3
}
```

Figure 9: Chain-of-Thoughts (CoT) prompt template used in LLM-based experiments.



Prompt Chaining (Task 1)

```
### Task
Generate a sequence of interactions for User {user_id}.
Document IDs: {doc_sequence}
Actions: [click, skip]

### Rules:
  1)    Each action corresponds to a document in the sequence.
  2)    The sequence should only contain "click" or "skip" actions at this stage.

### Expected Output Format (JSON)
{
  document_1 : action_1, document_2 : action_2, document_n, action_n
}.
```

Prompt Chaining (Task 2)

```
### Examples
Below are examples of personalized headlines generated for different users based on their document
content:

{fewshot_examples}

### Task:
Generate a personalized headline based on the document content. Ensure that the headline aligns with the
user's preferences and effectively captures the essence of the document.

User: {user_id}
Document: {doc_id}
Document Content: {doc_content}

### Expected Output Format (JSON)
Strictly return a JSON object in the following format:
{
  "headline": "your generated headline"
}
```

Figure 10: Prompt-Chaining template used in LLM-based experiments.