

SAND: Subjective-Anticipation-Augmented News Recommendation via Dual-Pathway Distillation

Hong-Kyun Bae^{*,1} Zhenping Jin^{*,2} Sang-Wook Kim²

¹Kookmin University ²Hanyang University

hongkyun@kookmin.ac.kr

{jzp0306, wook}@hanyang.ac.kr

Correspondence: wook@hanyang.ac.kr

Abstract

On news platforms, a user clicks on an article based on her *subjective anticipation* of its content, inferred from the title using her background knowledge. We introduce SAND (Subjective Anticipation-Augmented News Recommendation via Dual-pathway Distillation), the first framework to explicitly incorporate this user-specific anticipation that drives click behavior, formalized as the *anticipated body*, into news modeling. SAND (i) generates personalized anticipated bodies via an *agent-based collaborative system*, and (ii) learns preferences from anticipated bodies using a *candidate-aware attention network* with *dual-pathway knowledge distillation*. Experiments on real-world news datasets (*i.e.*, MIND and Adressa) demonstrate that SAND significantly outperforms eleven state-of-the-art baselines.

1 Introduction

Online news platforms (*e.g.*, Google News, Bing News) have become a major channel for accessing *up-to-the-minute* news. To accommodate the diverse news preferences of individual users on such platforms, *personalized news recommender systems* have been actively studied (An et al., 2019; Wu et al., 2019c,a; Hu et al., 2020; Mao et al., 2022; Yang et al., 2023; Bae et al., 2023; Yang et al., 2025; Ryu et al., 2025). In this line of research, news recommendation has primarily been formulated as a *click prediction* task: a user’s preference is inferred from her click history (*i.e.*, *user modeling*), while the characteristics of each news article are inferred from its content such as title or body text (*i.e.*, *news modeling*); then, the news whose characteristics best align with the user’s preference is recommended to her. Throughout this paper, we designate the set of news articles considered for click prediction as *candidate news articles*.

Typically, users rely solely on the title and category of a news article to decide whether to click it. The news title is a *highly condensed expression* that highlights just a few aspects of the full article; while the body provides comprehensive information, the title offers only *limited insight*. To address this gap, a user forms an *expectation* of the content she is likely to encounter upon clicking, based on *cues* in the title, before making a click decision. For instance, a title such as “Apple Unveils New Product” may lead technology enthusiasts to expect details on chipsets or cameras, while investors might anticipate stock price movements or revenue forecasts. Consequently, a user’s click decision is made by *filling* in the title’s *sparse* information with her own *background knowledge or interests*, *i.e.*, what she already knows about the topic. We refer to this *subjective anticipation* for the news as her “*anticipated body*” for that news article, *i.e.*, the content expected by her to be read *after a click*.

As a first attempt to fully capture the click mechanism described above, we propose a novel SAND (Subjective-Anticipation-Augmented News Recommendation via Dual-pathway Distillation) framework that incorporates the notion of *anticipated body* into news modeling, rather than relying solely on *objective* news content (*i.e.*, title and body text) as previous methods (An et al., 2019; Wu et al., 2019c,b; Ge et al., 2020; Mao et al., 2022; Yang et al., 2025). Specifically, our SAND consists of two modules: (i) generating *user-specific* anticipated bodies for a candidate news article; and (ii) learning user preferences from the result of news modeling (*i.e.*, *news embedding*) incorporated with the anticipated body.

Module (i) generates a user’s subjective anticipation, reflecting her prior knowledge and interests, by addressing two challenges:

- (C1) Because the quality of an anticipated body is a *qualitative* attribute *lacking* explicit ground

* Equal contribution.

truth, relying on a single *Large Language Model* (LLM) for generation does not reliably ensure its quality.

- (C2) For the same candidate news article, different subjective anticipations should be generated depending on each user’s prior knowledge and interests.

(Idea 1) Agentic Collaborative Generation. For (C1), we design an *agentic system*, which consists of the *Summarizer*, *Generator*, and *Controller* LLMs; it determines the *generation policy* through *collaboration* among these LLMs. For (C2), we make users grouped by their characteristics, *e.g.*, interests, with each group receiving a *distinct generation policy*.

Furthermore, module (ii) captures user preferences through candidate news embeddings built from the *anticipated body*—*i.e.*, rather than the actual body—since the click decision is made before the actual content is observed. It addresses the following two challenges:

- (C3) When modeling a user’s preference for a candidate news, the clicked articles in her history whose *actual bodies* align with the anticipated body should carry a *greater weight* in forming her user embedding: they provide the past evidence supporting her current anticipation for the candidate.
- (C4) LLM-generated text generally exhibits *less lexical diversity* compared to human-written text (Reviriego et al., 2024). Consequently, news embeddings based on the anticipated body tend to be *misaligned* with those based on the actual body in the embedding space, thereby weakening the *alignment assessment* described in (C3).

(Idea 2) Anticipated-body-aware Preference Learning. For (C3), we employ a *candidate-aware attention network* (Bahdanau, 2014; Ryu et al., 2025) that derives the user embedding by using the candidate news as a query, thereby producing a user representation conditioned on the candidate. For (C4), we propose a *Knowledge Distillation* (KD) (Hinton et al., 2015; Kang et al., 2020; Bae et al., 2025b) scheme in which a model pre-trained on actual bodies serves as the *teacher* and a model trained on anticipated bodies serves as the *student*. Notably, our KD distills alignment knowledge in *both* user and news embedding

spaces, forming a *dual-pathway* structure that sets it apart from conventional single-pathway KD.

The main contributions of this paper are summarized as follows:

- **Observation:** We define the notion of *anticipated body* as a user’s subjective expectation of a candidate news before clicking, serving as a key input for news modeling.
- **Methodology:** We introduce SAND, a framework that (i) creates user-specific anticipated bodies via an *agentic collaborative system* and learns preferences through (ii) a *candidate-aware attention network* and (iii) a *dual-pathway knowledge distillation* scheme.
- **Evaluation:** We test SAND on real-world datasets, comparing it against eleven baselines spanning DL-, GCN-, PLM-, and agent-based news recommendation.

2 SAND: The Proposed Framework

2.1 Problem Definition

We formulate the news recommendation as a click prediction task, following prior work (An et al., 2019; Wu et al., 2019c,a; Hu et al., 2020; Mao et al., 2022; Yang et al., 2023). Let $\mathcal{U}$ denote the set of users and $\mathcal{D}$ the set of news articles. Each news article $d \in \mathcal{D}$ is associated with a title t_d , a category c_d , and a body b_d .

For a user $u \in \mathcal{U}$, we denote her click history as the chronologically ordered sequence $\mathcal{H}_u = [d_1^u, d_2^u, \dots, d_{|\mathcal{H}_u|}^u]$ of articles she has previously clicked. The recommender $\mathcal{F}(\cdot)$ is presented with a candidate news set $\mathcal{C}_u \subseteq \mathcal{D}$ that the user has not yet clicked, and predicts the preference score $\hat{p}(u, d_c)$ for each candidate $d_c \in \mathcal{C}_u$.

In this work, we depart from existing approaches to candidate representation. Rather than encoding d_c solely from its *objective* textual content (*i.e.*, t_{d_c} , c_{d_c} , or b_{d_c}), we introduce the notion of an *anticipated body*. Given u and d_c , we define the anticipated body as $\tilde{b}_{u,d_c}$; for the same candidate, different users may obtain different anticipated bodies, reflecting their *distinct* background knowledge and interests. Then, our SAND framework produces the preference score as $\hat{p}(u, d_c) = \mathcal{F}(\mathcal{H}_u, \{t_{d_c}, \tilde{b}_{u,d_c}\})$, so that the top- N articles with highest predicted scores are recommended to u . Notations are summarized in Appendix A.

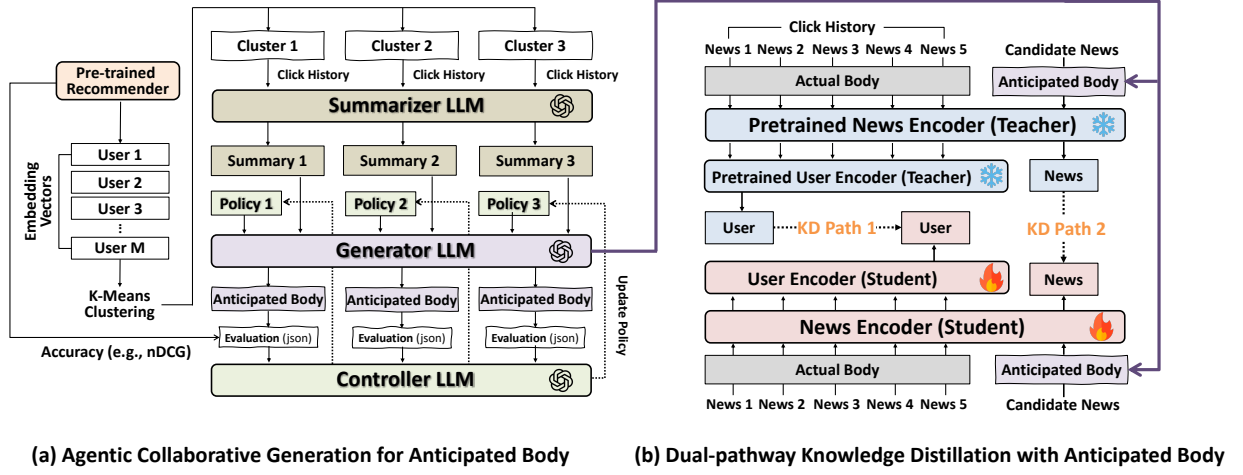


Figure 1: Overview of our SAND framework. (a) For each user cluster, a Summarizer LLM extracts users’ interests from their click histories, a Generator LLM produces anticipated bodies under a group-wise policy, and a Controller LLM iteratively refines the policy. (b) A student model trained on anticipated bodies is aligned with a frozen teacher pre-trained on actual bodies, via KD on both user (Path 1) and news (Path 2) embeddings. The candidate-aware attention network is omitted for clarity.

2.2 Overview

Figure 1 illustrates the overall architecture of SAND: Figure 1-(a) shows **(Idea 1)**, which generates user-specific anticipated bodies via an *agentic collaborative system* of three LLMs; and Figure 1-(b) shows **(Idea 2)**, which trains a *student model* (i.e., the recommender $\mathcal{F}(\cdot)$ in SAND) on the generated anticipated bodies. A *dual-pathway knowledge distillation* scheme aligns the *student’s* user and news embeddings with those of a *teacher* pre-trained on actual bodies. We detail **(Idea 1)**, **(Idea 2)**, and the Training & Inference procedure in Sections 2.3, 2.4, and 2.5, respectively.

2.3 (Idea 1) Agentic Collaborative Generation

To produce the anticipated body $\tilde{b}_{u,d_c}$ for a user u and a candidate d_c , we design an *agentic system* of three collaborating LLMs: a *Summarizer* f_{sum} , a *Generator* f_{gen} , and a *Controller* f_{ctrl} . The Summarizer f_{sum} distills the user u ’s preferences from her click history, f_{gen} produces the anticipated body conditioned on this summary under a *generation policy*, and f_{ctrl} iteratively refines the policy.

2.3.1 User Grouping and Preference Summarization

User grouping. To generate *distinct* anticipated bodies for users with *varying* preferences and backgrounds, we assign a dedicated *generation policy* to each *group* of similar users. First, we extract user

embeddings from a news recommender $\mathcal{R}^1$ pre-trained on actual bodies. These embeddings are then clustered into K groups via k-means clustering (Hartigan and Wong, 1979); we denote the cluster assignment as $\phi(u) = k$, with the k -th group represented by $\mathcal{U}_k$. Subsequent steps for **(Idea 1)** are performed *independently* within each $\mathcal{U}_k$.

Summarizer LLM. Before generating a user u ’s anticipated body for a specific candidate news d_c , we first understand her *overall background knowledge and interests* from her click history $\mathcal{H}_u$. For $u \in \mathcal{U}_k$, the Summarizer f_{sum} produces a textual summary s_u from her click history:

$$s_u = f_{\text{sum}}(\{t_d : d \in \mathcal{H}_u\}), \quad (1)$$

where s_u encapsulates u ’s background knowledge, interests, and likely perspective on news content. The full prompt for f_{sum} is provided in Appendix F.

2.3.2 Policy-controlled Anticipated Body Generation

Generation policy. The Generator f_{gen} produces $\tilde{b}_{u,d_c}$ according to a generation policy π , a structured specification that governs the stylistic axes of the generated text. We define π as a 5-tuple $(\pi_{\text{tone}}, \pi_{\text{abs}}, \pi_{\text{spec}}, \pi_{\text{len}}, \pi_{\text{fmt}})$, which controls the tone, abstraction, speculation, length, and narrative format of the generated text, respectively. Each

¹Any existing news recommender based on *actual* bodies can serve as $\mathcal{R}$. In our experiments, we used NAML (Wu et al., 2019a); see Section 3 for details.

component adopts a value from a predefined discrete set, grounded in the classical model of news objectivity (Westerståhl, 1983) and recent computational studies of news language (Chuey et al., 2024); leaving no axis unspecified, π thus limits the *unpredictable* variations that free-form generation would admit. We do not claim these five axes are exhaustive; we chose them for their grounding in the literature and to keep the Controller’s search space manageable. Appendix E shows the full value sets.

Generator LLM. Given a candidate $d_c \in \mathcal{C}_u$ and the current policy π_k^2 for group $k = \phi(u)$, the Generator f_{gen} produces:

$$\tilde{b}_{u,d_c} = f_{\text{gen}}(s_u, t_{d_c}, \pi_k). \quad (2)$$

Conditioned on s_u , f_{gen} produces the *subjective anticipation* that u is likely to form upon reading the title t_{d_c} ; the policy π_k then shapes this anticipation into a *news-like writing style*, obtaining the anticipated body $\tilde{b}_{u,d_c}$. The full prompt for f_{gen} is provided in Appendix F.

Controller LLM. The Controller f_{ctrl} refines the policy π_k through a sequence of user batches sampled from $\mathcal{U}_k$. At refinement step t , f_{ctrl} analyzes the current batch $\mathcal{B}^{(t)} \subseteq \mathcal{U}_k$, drawing on two sources of *evidence*: (i) a *batch-level loss* $\Delta^{(t)}$ and (ii) *diagnostic samples* $\mathcal{S}^{(t)}$, along with a *memory* $\mathcal{M}_k^{(t)}$ that accumulates persistent observations from prior iterations $1, \dots, t-1$. Based on these inputs, f_{ctrl} updates the policy:

$$\pi_k^{(t+1)} = f_{\text{ctrl}}(\pi_k^{(t)}, \Delta^{(t)}, \mathcal{S}^{(t)}, \mathcal{M}_k^{(t)}). \quad (3)$$

To compute $\Delta^{(t)}$, we examine each pair (u, d_c) where $u \in \mathcal{B}^{(t)}$ and $d_c \in \mathcal{C}_u$.³ The frozen recommender $\mathcal{R}$ infers two preference scores by only varying the candidate body: $\hat{p}^b(u, d_c)$ with the *actual* body b_{d_c} , and $\hat{p}^{\tilde{b}}(u, d_c)$ with the *anticipated* body $\tilde{b}_{u,d_c}$. The difference $\delta_{u,d_c} = \hat{p}^b(u, d_c) - \hat{p}^{\tilde{b}}(u, d_c)$ quantifies the user-relevant signal *lost* in the anticipated body. Taking the average over the batch yields $\Delta^{(t)} = \frac{1}{|\mathcal{B}^{(t)}|} \sum_{(u,d_c)} \delta_{u,d_c}$.

The diagnostic samples $\mathcal{S}^{(t)}$ gather the anticipated bodies for the pairs exhibiting both the smallest and largest values of δ_{u,d_c} . It provides concrete

examples of *success* and *failure* that $\Delta^{(t)}$ alone fails to convey. The resulting policy, denoted as $\pi_k^{(t+1)}$, guides the generation process by f_{gen} in the subsequent batch, while $\mathcal{M}_k^{(t)}$ preserves *cross-iteration trends* to avoid short-sighted adjustments.

Note that this feedback loop is strictly *one-directional*: $\mathcal{R}$ is pre-trained on actual bodies and remains frozen throughout, and $\tilde{b}_{u,d_c}$ enters $\mathcal{R}$ only as a scoring input at inference time.

Final policy determination. The refinement proceeds over a *fixed number* of batches, and the policy from the last batch is designated as the final policy π_k^* , which will then be utilized by f_{gen} throughout (Idea 2).

2.4 (Idea 2) Anticipated-body-aware Preference Learning

To capture user preferences from candidate news embeddings built from the *anticipated bodies*, we propose two components: a *candidate-aware attention network* and a *dual-pathway knowledge distillation* scheme.

2.4.1 Candidate-aware Attention Network

We follow the standard news recommender architecture (Wu et al., 2019a,c; An et al., 2019; Wu et al., 2019b) for the news encoder E_{news} , which maps an article’s input tokens to pre-trained word embeddings (Pennington et al., 2014) and aggregates them via a *Convolutional Neural Network* (CNN) followed by a word-level attention layer, yielding an article embedding $\mathbf{e}_d$. For the user encoder, however, the standard *additive attention* (Bahdanau, 2014)—which assigns each clicked article a weight from a learnable query vector—produces a *candidate-independent* user embedding $\mathbf{u}_u$, failing to reflect that the user’s relevant interests shift with the candidate she is currently facing.

We thus replace the learnable query vector with the *candidate news embedding* itself. Then, the candidate-aware user embedding $\mathbf{u}_{u,d_c}$ is obtained by a weighted sum of clicked-article embeddings $\mathbf{e}_d$, with weights $\alpha_{u,d_c,d}$ given by their dot products with the candidate $\mathbf{e}_{d_c}$:

$$\mathbf{u}_{u,d_c} = \sum_{d \in \mathcal{H}_u} \alpha_{u,d_c,d} \cdot \mathbf{e}_d, \quad (4)$$

$$\text{where } \alpha_{u,d_c,d} = \frac{\exp(\mathbf{e}_{d_c}^\top \mathbf{e}_d)}{\sum_{d' \in \mathcal{H}_u} \exp(\mathbf{e}_{d_c}^\top \mathbf{e}_{d'})}.$$

²The policy π_k is iteratively refined by the Controller LLM f_{ctrl} ; $\pi_k^{(0)}$ is initialized with random values.

³Each d_c at this stage is selected from the clicked news of u in the training set utilized to pre-train $\mathcal{R}$.

Importantly, the candidate news embedding is computed from its title and *anticipated body* as $\mathbf{e}_{d_c} = E_{\text{news}}(t_{d_c}, \tilde{b}_{u,d_c})$, whereas each clicked article uses its *actual body* as $\mathbf{e}_d = E_{\text{news}}(t_d, b_d)$. Through this mechanism, clicked articles whose actual bodies are more aligned with $\tilde{b}_{u,d_c}$ contribute more to $\mathbf{u}_{u,d_c}$, capturing the *candidate-specific* portion of the user’s preferences.

2.4.2 Dual-pathway Knowledge Distillation

The dot-product comparison in Eq. (4) implicitly assumes that $\mathbf{e}_d$ and $\mathbf{e}_{d_c}$ reside in a *coherent embedding space*. However, the LLM-generated $\tilde{b}_{u,d_c}$ tends to be inherently *less lexically diverse* than the human-written b_{d_c} , according to prior studies (Reviriego et al., 2024; Shaib et al., 2025). Thus, a single news encoder E_{news} trained from scratch on both text types is prone to a *restricted* embedding space biased toward the anticipated bodies, weakening the very alignment that Eq. (4) relies on.

To resolve this issue, we regard the news encoder and the user encoder constructed above as the *student*, and introduce a *teacher* that shares the same architecture but is pre-trained on the standard click prediction task using actual bodies for both the click history and the candidate news. Then, the teacher remains *fully frozen*, whereas the student is trained *from scratch* with *actual bodies* for the click history and *anticipated bodies* for candidates. During the distillation, the teacher is fed the *same inputs* as the student, so that the alignment isolates differences in their encoders rather than in inputs. We denote their news encoders as E_{news}^t (teacher) and E_{news}^s (student), and align the two embedding spaces (*i.e.*, user and news embedding spaces) via *two pathways*.

KD Path 1: User-embedding alignment. This pathway regularizes how the student aggregates clicked articles into a user representation. Each model computes its candidate-aware user embedding via Eq. (4), yielding the student’s $\mathbf{u}_{u,d_c}^s$ and the teacher’s $\mathbf{u}_{u,d_c}^t$, and the KD loss is: $\mathcal{L}_{\text{KD}}^{\text{user}} = 1 - \cos(\mathbf{u}_{u,d_c}^s, \mathbf{u}_{u,d_c}^t)$.

KD Path 2: News-embedding alignment. This pathway directly aligns the candidate news embeddings produced by the two encoders; we denote the student’s by $\mathbf{e}_{d_c}^s = E_{\text{news}}^s(t_{d_c}, \tilde{b}_{u,d_c})$ and the teacher’s by $\mathbf{e}_{d_c}^t = E_{\text{news}}^t(t_{d_c}, \tilde{b}_{u,d_c})$. The KD loss is: $\mathcal{L}_{\text{KD}}^{\text{news}} = 1 - \cos(\mathbf{e}_{d_c}^s, \mathbf{e}_{d_c}^t)$.

Training objective. The student is trained jointly with the click prediction loss $\mathcal{L}_{\text{rec}}$ (*i.e.*, negative log-likelihood) and the two KD losses:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rec}} + \lambda_1 \mathcal{L}_{\text{KD}}^{\text{user}} + \lambda_2 \mathcal{L}_{\text{KD}}^{\text{news}}, \quad (5)$$

where $\lambda_1, \lambda_2 \geq 0$ balance the two distillation pathways against $\mathcal{L}_{\text{rec}}$.

2.5 Training and Inference

For each user u in the training set, we treat her clicked articles as *positive candidates* and sample J unclicked articles per positive as *negative candidates*. Before computing the loss, we produce the anticipated body $\tilde{b}_{u,d_c} = f_{\text{gen}}(s_u, t_{d_c}, \pi_{\phi(u)}^*)$ for every (user, candidate) pair. The student then encodes each of the $(1 + J)$ candidates from its title and anticipated body, and is optimized so that the positive receives a higher preference score than the J negatives under $\mathcal{L}_{\text{total}}$ in Eq. (5). At recommendation time, the recommender $\mathcal{F}$, defined in Section 2.1, operates as the trained student: for each candidate d_c presented to u , it computes $\hat{p}(u, d_c) = \mathcal{F}(\mathcal{H}_u, \{t_{d_c}, \tilde{b}_{u,d_c}\}) = (\mathbf{u}_{u,d_c}^s)^\top \mathbf{e}_{d_c}^s$, making the candidates with the highest scores recommended.

3 Experiments

We conduct extensive experiments to address the following *evaluation questions* (EQs):

- (EQ1) How accurate is SAND compared to state-of-the-art baselines?
- (EQ2) How effective are the design choices to implement (Idea 1)?
- (EQ3) How effective are the design choices to implement (Idea 2)?
- (EQ4) How sensitive is SAND to the distillation weights λ_1 and λ_2 ?
- (EQ5) Does the Controller LLM effectively refine the policy over batches?
- (EQ6) How efficient is SAND at training and inference time?

3.1 Experimental Setup

Datasets. We use two widely adopted benchmarks: MIND-1W (Wu et al., 2020) and Adressa-1W (Gulla et al., 2017). Since SAND requires LLM API calls to generate anticipated bodies for

Table 1: Overall accuracy comparison of SAND against eleven state-of-the-art baselines on MIND-1W and Adressa-1W. The best result in each column is in **bold**, and the best baseline is underlined. Gain (%) reports the relative improvement of SAND over the best-performing baseline.

Category	Method	MIND-1W			Adressa-1W		
		MRR	nDCG@5	HIT@1	MRR	nDCG@5	HIT@1
DL-based	LSTUR (An et al., 2019)	0.5742	0.6800	0.3338	0.4753	0.6015	0.2702
	NAML (Wu et al., 2019a)	0.5807	0.6845	0.3544	0.5571	0.6662	0.3314
	FIM (Wang et al., 2020)	0.5348	0.6496	0.2932	0.4539	0.5854	0.2338
GCN-based	CNE-SUE (Mao et al., 2021)	0.5767	0.6815	0.3490	0.5384	0.6501	0.3389
	CROWN (Ko et al., 2025)	0.5762	0.6813	0.3428	0.5294	0.6432	0.3276
	LIME (Ryu et al., 2025)	0.5561	0.6661	0.3118	0.5587	0.6665	0.3462
PLM-based	NAML-BERT (Wu et al., 2021)	<u>0.5907</u>	<u>0.6920</u>	<u>0.3673</u>	0.4956	0.6171	0.2907
Agent-based	AgentCF (Zhang et al., 2024)	0.5323	0.6479	0.2791	0.4787	0.6047	0.2376
	ToolRec (Zhao et al., 2024)	0.5205	0.6387	0.2701	0.4887	0.6137	0.2482
	MACRec (Wang et al., 2024)	0.5705	0.6768	0.3376	<u>0.5602</u>	<u>0.6671</u>	<u>0.3691</u>
	AFL (Cai et al., 2025)	0.5858	0.6882	0.3635	0.5495	0.6601	0.3291
Ours	SAND	0.5981	0.6978	0.3790	0.5826	0.6854	0.3692
Gain (%) over best baseline		+1.25	+0.84	+3.18	+4.00	+2.74	+0.03

every (user, candidate) pair, we randomly sample 2,000 users from each dataset to keep API costs tractable, while retaining all of their click interactions. This protocol follows the prior practice of agentic LLM-based recommender systems (Zhang et al., 2024; Zhao et al., 2024; Cai et al., 2025), while operating at a *substantially larger scale* than is typical in this line of work. We follow the chronological train/validation/test split adopted in prior work (Ko et al., 2025; Ryu et al., 2025), with the click logs partitioned in a 5:1:1 ratio. Detailed dataset statistics are provided in Appendix B.

Baselines. We compare SAND against eleven state-of-the-art baselines, spanning four categories: (i) *DL-based*—LSTUR (An et al., 2019), NAML (Wu et al., 2019a), and FIM (Wang et al., 2020); (ii) *GCN-based*—CNE-SUE (Mao et al., 2021), CROWN (Ko et al., 2025) and LIME (Ryu et al., 2025); (iii) *PLM-based*—NAML-BERT (Wu et al., 2021), which replaces NAML’s text encoders with BERT (Devlin et al., 2019); and (iv) *agent-based*—AgentCF (Zhang et al., 2024), ToolRec (Zhao et al., 2024), MACRec (Wang et al., 2024), and AFL (Cai et al., 2025). We use **GPT-4o-mini** as the backbone LLM for *all* agent-based baselines, as well as for SAND, to ensure a *fair comparison* among methods. Detailed descriptions of each baseline are provided in Appendix C.

Evaluation protocol. For both datasets, we randomly sample four negatives from the pool of unclicked articles (*i.e.*, $J = 4$) (Kim et al., 2022; Ahn et al., 2023). All hyperparameters are tuned

on the validation set, and we report the corresponding results on the test set. Further implementation details, such as batch size, are provided in Appendix D.

3.2 Main Results (EQ1)

Table 1 reports the overall accuracy of SAND against eleven state-of-the-art baselines, which makes the following observations.

(O1) SAND achieves the highest accuracy. SAND outperforms *all* eleven baselines on five of the six evaluation settings, with relative gains of up to 3.18% in HIT@1 on MIND and 4.00% in MRR on Adressa over the best competitor, confirming the effectiveness of modeling a user’s subjective anticipation as the *anticipated body*.

(O2) A language model brings consistent benefits in news recommendation. The PLM-based (NAML-BERT) and agent-based (AFL, MACRec) baselines outperform most DL- and GCN-based methods. Nonetheless, SAND outperforms the best agent-based competitors by significant margins (*e.g.*, +4.27% in HIT@1 on MIND over AFL), demonstrating that it is more effective than directly employing LLMs to rank candidates via prompts.

(O3) SAND surpasses its NAML backbone by a large margin. SAND improves over NAML by up to 6.94% (HIT@1) on MIND and 11.41% (HIT@1) on Adressa, quantifying the net benefit of the proposed two modules.

Table 2: Ablation study on **(Idea 1)**, evaluated on MIND. Each row *cumulatively* adds one component on top of the previous. All variants are trained without dual-pathway knowledge distillation to isolate the contribution of **(Idea 1)**.

Variant	MRR	nDCG@5	HIT@1
Base (One-Shot)	0.5645	0.6725	0.3276
(+) Controller	0.5675	0.6747	0.3331
(+) Clustering	0.5743	0.6798	0.3420
(+) Summarizer	0.5789	0.6832	0.3523

3.3 Ablation Study on (Idea 1) (EQ2)

On MIND, we start from a base variant that generates $\tilde{b}_{u,d_c}$ via a single-shot prompt *without* any of the proposed components.

(O1) All three components contribute positively to recommendation accuracy. Each incremental addition consistently improves accuracy in all three metrics. The full configuration achieves a cumulative gain of +7.54% in HIT@1 over the base, indicating that the three components are complementary.

(O2) User clustering shows the largest gain. Adding clustering on top of policy refinement yields the largest improvement in MRR, confirming the value of tailoring the generation policy to group-specific patterns.

(O3) Summarization sharpens top-rank predictions. While adding the Summarizer (*i.e.*, $f_{\text{sum}}(\cdot)$) yields moderate gains in MRR and nDCG@5, its impact on HIT@1 is the largest (+3.01%), suggesting that a candidate-agnostic preference profile helps the Generator (*i.e.*, $f_{\text{gen}}(\cdot)$) better pinpoint each user’s most preferred article.

3.4 Ablation Study on (Idea 2) (EQ3)

We evaluate seven variants of SAND on MIND, where all variants use the full **(Idea 1)** configuration. Table 3 reports the results.

(O1) Candidate-aware attention is essential. Removing it (*w/o* CAA) drops MRR, showing that conditioning the user representation on each candidate is important when it is represented by a user’s anticipation.

(O2) Dual-pathway distillation is essential, and single-pathway variants are insufficient. Removing both KD pathways (*w/o* KD) significantly decreases MRR. Interestingly, relying on only one pathway—as in conventional single-pathway

Table 3: Ablation study on **(Idea 2)**, evaluated on MIND, where **Teacher** indicates NAML pre-trained on actual bodies with candidate-aware attention and **Fine-tuning** indicates the teacher fine-tuned on anticipated bodies. **SAND w/ Actual body** denotes the full SAND configuration in which the candidate is encoded from its actual body instead of the anticipated body.

Variant	MRR	nDCG@5	HIT@1
Teacher	0.5805	0.6845	0.3489
Fine-tuning	0.5782	0.6828	0.3441
SAND w/o KD	0.5789	0.6832	0.3523
SAND w/o CAA	0.5895	0.6911	0.3674
SAND w/ Path 1	0.5607	0.6694	0.3215
SAND w/ Path 2	0.5714	0.6777	0.3352
SAND w/ Actual body	0.5875	0.6899	0.3571
SAND(full)	0.5981	0.6978	0.3790

KD—performs worse than the *w/o* KD baseline. This is due to the *structural coupling* that the user embedding is an aggregation of news embeddings (Eq. (4)), preventing independent drift. It validates our dual-pathway design as an effective adaptation of KD to the structural characteristics of news recommendation.

(O3) The gain originates from the anticipated body itself, not from the distillation scheme. Replacing the candidate’s anticipated body with its actual body, while keeping the candidate-aware attention and the dual-pathway KD intact, drops HIT@1 from 0.3790 to 0.3571. Notably, this variant still improves over the Teacher (0.3489), confirming that dual-pathway KD is beneficial on its own; yet the remaining 6.13% margin is attributable solely to representing the candidate by what the user anticipates rather than by what the article actually contains. This isolates the contribution of **(Idea 1)** under an identical architecture and training setting.

(O4) Fine-tuning is not a viable alternative to distillation. Fine-tuning the teacher on anticipated bodies underperforms even the *w/o* KD variant, indicating that the distributional gap *cannot be reconciled* by gradient-based adaptation alone.

3.5 Hyperparameter Sensitivity (EQ4)

We analyze the sensitivity of SAND to the two distillation weights λ_1 (user-side KD) and λ_2 (news-side KD) on the validation set of MIND. In Figure 2, SAND achieves its best validation accuracy at $(\lambda_1, \lambda_2) = (0.5, 0.5)$ across all three metrics,

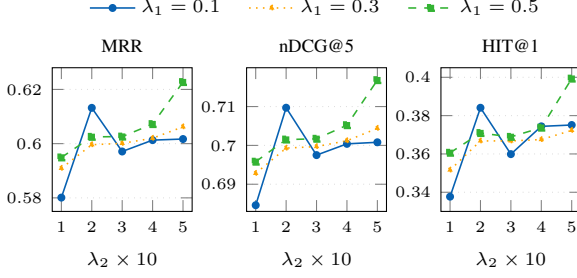


Figure 2: Sensitivity of SAND to the weights λ_1 and λ_2 . The full 5×5 grid is reported in Appendix G.

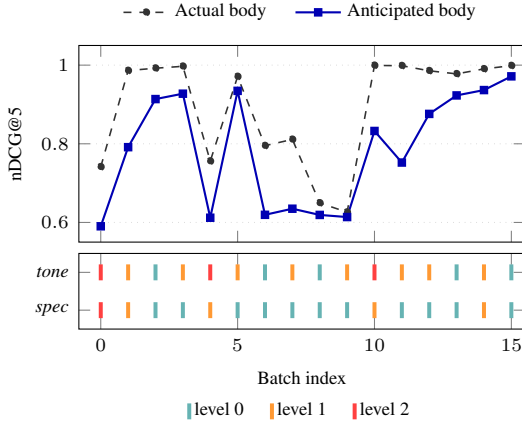


Figure 3: Diagnostic signal used by the Controller, measured on Cluster 0 of Adressa-1W. Top: per-batch nDCG@5 of the frozen pre-trained recommender $\mathcal{R}$ when the candidate is encoded from its actual versus anticipated body. Bottom: the corresponding tone and speculation_count policy parameters along the same batch axis. Note that these curves are *not* final recommendation accuracy: $\mathcal{R}$ is pre-trained solely on actual bodies and therefore inherently favors actual-body inputs. The figure shows how the Controller narrows this diagnostic gap while refining the policy.

which we adopt as the default configuration.

3.6 Case Study on Policy (EQ5)

We track the diagnostic signal that drives policy refinement, *i.e.*, the per-batch nDCG@5 of the frozen $\mathcal{R}$ under actual versus anticipated candidate bodies, along with the current policy parameters, on Cluster 0 of Adressa-1W. Since $\mathcal{R}$ is pre-trained on actual bodies, the absolute level of these curves reflects $\mathcal{R}$ ’s own bias rather than the accuracy of SAND; what matters here is the trend of their gap.

The gap between the nDCG@5 of actual and anticipated bodies steadily narrows over batches, indicating that the Controller (*i.e.*, $f_{\text{ctrl}}(\cdot)$) effectively refines the generation policy based on the

Table 4: Efficiency comparison on MIND (wall-clock time on a single RTX PRO 6000; LLM cost over the full test set). MACRec is prompting-based and has no training phase. SAND’s training time covers both teacher pre-training and student training.

	NAML	AFL	MACRec	SAND
Training time (min)	9.4	10.5	—	23.3
Inference time (min)	1.2	61.0	611.2	37.7
Inference LLM cost (\$)	—	0.75	6.54	9.86

per-batch feedback. The bottom strips show the evolution of the *tone* and *speculation_count* parameters: starting from expressive and speculative settings (*tone* = mildly_expressive, *speculation_count* = 2 at batch 0), f_{ctrl} progressively converges to a more conservative regime (*tone* = mostly neutral or very_neutral, *speculation_count* = 0).

Appendix I provides qualitative examples of the resulting anticipated bodies across different user clusters.

3.7 Efficiency Analysis (EQ6)

We measure SAND’s wall-clock time and monetary cost on MIND against its backbone (NAML) and the two strongest agent-based competitors. MACRec is prompting-based and thus has no training phase (Wang et al., 2024).

(O1) SAND is substantially faster than agent-based competitors at inference.

It reduces inference time by 38% relative to AFL and by more than $16\times$ relative to MACRec, since the LLM is invoked once per candidate to generate its anticipated body, rather than repeatedly to rank candidates. Inference latency directly determines user-perceived responsiveness in deployment, which makes this margin practically meaningful. SAND incurs a higher LLM cost than the baselines (\$9.86) because it generates a full anticipated body for every (user, candidate) pair; we discuss amortization strategies for this overhead in the Limitations section.

(O2) Training overhead is bounded by a constant factor.

Let n denote the number of training impressions, J the negatives per positive, h the history length, L the maximum text length, and d the embedding dimension. NAML’s training cost is $T_{\text{NAML}} = O(E \cdot n \cdot (h+1+J) \cdot L \cdot d)$, dominated by encoding $(h+1+J)$ articles per impression over E epochs. SAND’s model training consists of (i) teacher pre-training, identical to T_{NAML} , and (ii) student training, whose per-step cost adds one

frozen teacher forward pass and $O(d)$ KD losses, *i.e.*, at most $2 \times$ NAML’s per-step cost. The total is thus bounded by $3 \cdot T_{\text{NAML}}$, and the measured ratio ($2.5 \times$; 23.3 vs. 9.4 min) falls within this bound. Beyond model training, determining the final policy π_k^* requires 12,254 LLM calls (\$9.91, 299.6 min) on MIND, but this procedure runs offline exactly once and is never repeated during training or inference. Neither term grows super-linearly beyond NAML’s own complexity.

4 Related Work

4.1 News Recommendation

Early DL-based methods focus on richer text and user encoding: LSTUR (An et al., 2019) captures long- and short-term interests using a GRU; NAML (Wu et al., 2019a) jointly encodes title, body, and category through multi-view attention; and FIM (Wang et al., 2020) performs fine-grained matching via hierarchical dilated convolutions. More recent work introduces structural and semantic signals: GCN-based methods such as CNE-SUE (Mao et al., 2021), CROWN (Ko et al., 2025) represent user-news correlations through graph structures, while PLM-based variants (Wu et al., 2021) replace shallow text encoders with pre-trained language models. Despite these advances, all of the above methods represent each candidate by its *objective content* (title and/or body), overlooking the role of the user’s subjective anticipation.

4.2 Agent-based Recommendation

With the rise of LLMs, agent-based recommenders have emerged as a new paradigm in which LLMs participate in the recommendation process beyond text encoding. One line of work casts the LLM itself as a recommender or ranker through prompting: AgentCF (Zhang et al., 2024) models users and items as interacting agents, ToolRec (Zhao et al., 2024) treats the LLM as a tool-using agent over candidate items, MACRec (Wang et al., 2024) coordinates multiple specialized LLM agents, and AFL (Cai et al., 2025) augments traditional recommenders with LLM-derived semantic features. Despite these advances, they typically use LLMs to *rank or summarize* existing content. In contrast, SAND uses LLMs to *generate* user-specific content as a new input to a downstream recommender, repositioning the role of the LLM in news recommendation.

5 Conclusion

We presented SAND, the first news recommendation framework that explicitly models the user’s subjective anticipation about a candidate article as the *anticipated body*. SAND consists of two modules: (i) an agentic collaborative generation module, in which a Summarizer, Generator, and Controller LLM jointly determine a group-wise generation policy offline; and (ii) a candidate-aware attention network coupled with a dual-pathway knowledge distillation scheme that aligns the anticipated-body representation with an actual-body teacher. Extensive experiments on MIND and Adressa demonstrate that SAND consistently outperforms eleven state-of-the-art baselines spanning DL-based, GCN-based, PLM-based, and agent-based methods, and our ablation confirms that the gain originates from the anticipated body itself rather than from the distillation scheme.

We believe the notion of anticipated content is not specific to news. The premise underlying SAND is general: whenever a user decides to engage with an item from a condensed cue, her decision rests on what she anticipates rather than on the item’s full content, which she has not yet observed. This condition holds in *point-of-interest* (POI) recommendation (Bae et al., 2024), where a user judges a venue from its name and category before visiting, and in e-commerce (Bae et al., 2025a), where a product is often selected from a title and a thumbnail alone. Applying SAND to such domains would require adapting the generation policy to their respective conventions, which we leave to future work.

Limitations

We discuss several limitations of this work that suggest directions for future research.

Dataset scale. Since SAND relies on LLM API calls to generate anticipated bodies for each (user, candidate) pair, we evaluate it on 2,000 randomly selected users per dataset rather than the entire benchmark. Although this sample size is *significantly larger* than previous agentic recommender studies, testing SAND on the full datasets would offer more robust proof of its scalability.

Dependence on a single LLM. All anticipated bodies in our experiments are generated using GPT-4o-mini. The behavior of SAND under other LLMs—open-source models, larger proprietary

models, or domain-adapted variants—remains to be examined, and the optimal generation policy may differ across LLMs.

Inference-time cost. During inference, SAND invokes the Generator LLM to produce an anticipated body for each candidate, thereby requiring higher latency and monetary cost compared with conventional news recommenders. Reducing this overhead, for instance, through caching, distillation into a lightweight generator, or batched generation, is an important direction for practical deployment.

Users without click history. Our study does not address users with no interaction record, since SAND infers a user’s background knowledge from her click history. This reflects the nature of public news benchmarks, which provide click logs and news metadata but no user attributes, and it is a limitation shared by *all* baselines we compare against. If user attributes such as demographics become available, our clustering-based design offers a natural remedy: we could assign a new user to an appropriate cluster based on those attributes and serve the corresponding policy π_k^* without any interaction history. We leave this extension to future work.

Reproducibility under LLM stochasticity. LLM generations are non-deterministic, and the anticipated bodies produced by GPT-4o-mini may vary across runs even with identical inputs. To mitigate this, we set the decoding temperature low and report results averaged over multiple generations; nonetheless, exact reproducibility of the anticipated body corpus is inherently limited.

Ethical Considerations

Use of AI tools. We disclose the use of generative AI tools in the preparation of this work, in accordance with the ACL Policy on AI Writing Assistance. We used Claude (Anthropic) for two purposes: (i) language polishing of the manuscript text—paraphrasing, proofreading, and improving readability of drafts written by the authors; and (ii) assistance with generating L^AT_EX code for figures and tables, including the TikZ/pgfplots code used to render Figures 2, 3, 4, and 5. All research ideas, experimental design, model architecture, and results analysis are the authors’ own. The authors take full responsibility for the correctness of the methods, results, and writing.

Use of LLM-generated content. SAND relies on GPT-4o-mini to generate anticipated bodies, user preference summaries, and policy updates. As LLM-generated text may inherit biases from the model’s pre-training data, the anticipated bodies could reflect such biases in the content users are exposed to. In deployment scenarios, this concern would warrant additional content moderation and bias auditing beyond the scope of this paper.

Privacy and user data. Our experiments use the publicly released MIND (Wu et al., 2020) and Adressa (Gulla et al., 2017) datasets, which contain anonymized user click histories. We do not collect new user data, and the user IDs in these datasets are not linkable to real-world individuals. The anticipated bodies generated by SAND are based solely on titles of clicked articles and do not expose user-identifiable information.

Reproducibility. We commit to releasing the source code, prompt templates (Appendix F), and processed datasets upon publication to support reproducibility.

Acknowledgements

Hong-Kyun Bae’s work has been supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2026-25499637) and the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.RS-2026-25527987, D-PRISM: Disaster Prediction via Reconstruction and Inferential Scenario Modeling) and the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant (No. RS-2025-02219317, AI Star Fellowship (Kookmin University)) funded by the Korean government (MSIT).

Sang-Wook Kim’s work has been supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2020-II201373, Artificial Intelligence Graduate School Program(Hanyang University)) and under the Leading Generative AI Human Resources Development(IITP-2026-RS-2026-25544375) grant funded by the Korea government(MSIT).

References

- Jeewon Ahn, Hong-Kyun Bae, and Sang-Wook Kim. 2023. Is the impression log beneficial to effective model training in news recommender systems? No, it's not. In *Companion Proceedings of the ACM Web Conference 2023*, pages 61–64.
- Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. 2019. Neural news recommendation with long-and short-term user representations. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 336–345.
- Hong-Kyun Bae, Jeewon Ahn, Dongwon Lee, and Sang-Wook Kim. 2023. Lancer: A lifetime-aware news recommender system. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4141–4148.
- Hong-Kyun Bae, Hae-Ri Jang, Won-Yong Shin, and Sang-Wook Kim. 2025a. Ranking items by the current-preferences and profits: A list-wise learning-to-rank approach to profit maximization. In *Proceedings of the ACM on Web Conference 2025*, pages 5010–5021.
- Hong-Kyun Bae, Jiyeon Kim, Jongwuk Lee, and Sang-Wook Kim. 2025b. Duet: Dually guided knowledge distillation from explicit feedback. *Information Fusion*, 120:103098.
- Hong-Kyun Bae, Yebeen Kim, Hyunjoon Kim, and Sang-Wook Kim. 2024. Negative sampling in next-POI recommendations: Observation, approach, and evaluation. In *Proceedings of the ACM Web Conference 2024*, pages 3888–3899.
- Dzmitry Bahdanau. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Shihao Cai, Jizhi Zhang, Keqin Bao, Chongming Gao, Qifan Wang, Fuli Feng, and Xiangnan He. 2025. Agentic feedback loop modeling improves recommendation and user simulation. In *Proceedings of the 48th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 2235–2244.
- Aaron Chuey, Yiwei Luo, and Ellen M Markman. 2024. Epistemic language in news headlines shapes readers' perceptions of objectivity. *Proceedings of the National Academy of Sciences*, 121(20):e2314091121.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.
- Suyu Ge, Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020. Graph enhanced representation learning for news recommendation. In *Proceedings of the web conference 2020*, pages 2863–2869.
- Jon Atle Gulla, Lemei Zhang, Peng Liu, Özlem Özgöbek, and Xiaomeng Su. 2017. The Adressa dataset for news recommendation. In *Proceedings of the international conference on web intelligence*, pages 1042–1048.
- John A Hartigan and Manchek A Wong. 1979. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
- Linmei Hu, Chen Li, Chuan Shi, Cheng Yang, and Chao Shao. 2020. Graph neural news recommendation with long-term and short-term interest modeling. *Information Processing & Management*, 57(2):102142.
- SeongKu Kang, Junyoung Hwang, Wonbin Kweon, and Hwanjo Yu. 2020. DE-RRD: A knowledge distillation framework for recommender system. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 605–614.
- Taeho Kim, Yungi Kim, Yeon-Chang Lee, Won-Yong Shin, and Sang-Wook Kim. 2022. Is it enough just looking at the title? leveraging body text to enrich title words towards accurate news recommendation. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pages 4138–4142.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Yunyong Ko, Seongeun Ryu, and Sang-Wook Kim. 2025. CROWN: A novel approach to comprehending users' preferences for accurate personalized news recommendation. In *Proceedings of the ACM on Web Conference 2025*, pages 1911–1921.
- Dairui Liu, Boming Yang, Honghui Du, Derek Greene, Neil Hurley, Aonghus Lawlor, Ruihai Dong, and Irene Li. 2024. RecPrompt: A self-tuning prompting framework for news recommendation using large language models. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 3902–3906.
- Zhiming Mao, Jian Li, Hongru Wang, Xingshan Zeng, and Kam-Fai Wong. 2022. DIGAT: modeling news recommendation with dual-graph interaction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6595–6607.

- Zhiming Mao, Xingshan Zeng, and Kam-Fai Wong. 2021. Neural news recommendation with collaborative news encoding and structural user encoding. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 46–55.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Pedro Reviriego, Javier Conde, Elena Merino-Gómez, Gonzalo Martínez, and José Alberto Hernández. 2024. Playing with words: Comparing the vocabulary and lexical diversity of ChatGPT and humans. *Machine Learning with Applications*, 18:100602.
- Seongeun Ryu, Yunyong Ko, and Sang-Wook Kim. 2025. Is this news still interesting to you?: Lifetime-aware interest matching for news recommendation. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 2515–2524.
- Chantal Shaib, Venkata S Govindarajan, Joe Barrow, Jiuding Sun, Alexa Siu, Byron C Wallace, and Ani Nenkova. 2025. Standardizing the measurement of text diversity: A tool and comparative analysis. In *Proceedings of The 14th International Joint Conference on Natural Language Processing and The 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 36–46.
- Heyuan Wang, Fangzhao Wu, Zheng Liu, and Xing Xie. 2020. Fine-grained interest matching for neural news recommendation. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 836–845.
- Zhefan Wang, Yuanqing Yu, Wendi Zheng, Weizhi Ma, and Min Zhang. 2024. MACRec: A multi-agent collaboration framework for recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2760–2764.
- Jörgen Westerstähl. 1983. Objective news reporting: General premises. *Communication research*, 10(3):403–424.
- Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019a. Neural news recommendation with attentive multi-view learning. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 3863–3869.
- Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019b. NPA: neural news recommendation with personalized attention. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2576–2584.
- Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019c. 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)*, pages 6389–6394.
- Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Empowering news recommendation with pre-trained language models. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 1652–1656.
- Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and 1 others. 2020. MIND: A large-scale dataset for news recommendation. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 3597–3606.
- Boming Yang, Dairui Liu, Toyotaro Suzumura, Ruihai Dong, and Irene Li. 2023. Going beyond local: Global graph-enhanced personalized news recommendations. In *Proceedings of the 17th ACM conference on recommender systems*, pages 24–34.
- Zhen Yang, Wenhui Wang, Tao Qi, Peng Zhang, Tianyun Zhang, Ru Zhang, Jianyi Liu, and Yongfeng Huang. 2025. GLoCIM: Global-view long chain interest modeling for news recommendation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6855–6865.
- Junjie Zhang, Yupeng Hou, Ruobing Xie, Wenqi Sun, Julian McAuley, Wayne Xin Zhao, Leyu Lin, and Jirong Wen. 2024. AgentCF: Collaborative learning with autonomous language agents for recommender systems. In *Proceedings of the ACM Web Conference 2024*, pages 3679–3689.
- Yuyue Zhao, Jiancan Wu, Xiang Wang, Wei Tang, Dingxian Wang, and Maarten De Rijke. 2024. Let me do it for you: Towards llm empowered recommendation via tool learning. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1796–1806.

A Notations

Table 5 summarizes the notations used throughout this paper.

B Dataset Statistics

Table 6 reports the statistics of the sampled MIND-1W and Adressa-1W datasets used in our experiments. From each dataset, we randomly sample 2,000 users and retain all of their click interactions, then split the resulting click logs chronologically in a 5:1:1 ratio for training, validation, and testing.

Table 5: Summary of notations used throughout this paper.

Symbol	Description
$\mathcal{U}, \mathcal{D}$	Sets of users and news articles
u, d	A user and a news article
t_d, c_d, b_d	Title, category, actual body of d
$\mathcal{H}_u$	Click history of u
$\mathcal{C}_u$	Candidate news set for u
d_c	A candidate news ($d_c \in \mathcal{C}_u$)
$\tilde{b}_{u,d_c}$	Anticipated body of d_c for u
$\mathcal{F}(\cdot)$	Recommender (the student)
$\hat{p}(u, d_c)$	Predicted preference of u for d_c
$\mathcal{R}$	Pre-trained recommender for (Idea 1)
$f_{\text{sum}}, f_{\text{gen}}, f_{\text{ctrl}}$	Summarizer, Generator, Controller
s_u	Preference summary of u from f_{sum}
$K, \phi(u), \mathcal{U}_k$	# clusters, assignment of u , k -th cluster
π_k, π_k^*	Policy for cluster k and its final form
$\mathcal{B}^{(t)}$	User batch at step t
$\Delta^{(t)}, \mathcal{S}^{(t)}$	Batch-level loss and diagnostic samples
$\mathcal{M}_k^{(t)}$	Controller’s memory at step t
δ_{u,d_c}	Per-sample gap, $\hat{p}^b - \hat{p}^{\tilde{b}}$
E_{news}	News encoder (shared architecture)
$E_{\text{news}}^s, E_{\text{news}}^t$	Student and teacher encoders
$\mathbf{e}_d, \mathbf{e}_{d_c}$	News embeddings of d and d_c
$\mathbf{u}_{u,d_c}$	Candidate-aware user embedding
$\alpha_{u,d_c,d}$	Attention weight for $d \in \mathcal{H}_u$
$\mathcal{L}_{\text{rec}}$	Click-prediction loss
$\mathcal{L}_{\text{KD}}^{\text{user}}, \mathcal{L}_{\text{KD}}^{\text{news}}$	KD losses for Paths 1 and 2
λ_1, λ_2	Weights for Paths 1 and 2
J	Negatives sampled per positive

Specifically, we order all users’ click logs on a single global timeline and partition them chronologically, so every interaction seen during training strictly precedes every validation and test interaction. This rules out temporal leakage across users, which a user-disjoint split would otherwise admit.

Table 6: Statistics of the sampled datasets. Each contains 2,000 randomly sampled users with all click interactions retained.

Statistic	MIND-1W	Adressa-1W
# users	2,000	2,000
# news articles	24,060	17,983
# click interactions	78,294	95,809
Sparsity	99.73%	99.83%
Avg. history length	38.7	38.0
Avg. title #tokens	11.0	6.6
Avg. body #tokens	29.6	472.4

Moreover, the sampling scale of SAND (2,000 users per dataset) is substantially larger than that adopted in prior agentic LLM-based recommender studies (e.g., 100 users (Zhang et al., 2024)) and 400 users (Liu et al., 2024).

C Baseline Descriptions

We provide brief descriptions of the eleven baselines compared with SAND.

DL-based methods. **LSTUR** (An et al., 2019) captures long-term user interests via ID embeddings and short-term interests via a GRU over recent clicks. **NAML** (Wu et al., 2019a) encodes a news article from multiple views (title, body, category) and aggregates them through additive attention; the user encoder applies additive attention over clicked-article embeddings. **FIM** (Wang et al., 2020) performs fine-grained matching between candidate and clicked news through hierarchical dilated convolutions, capturing multi-granularity semantic interactions.

GCN-based methods. **CNE-SUE** (Mao et al., 2021) introduces collaborative news encoding and structural user encoding, modeling correlations among news articles and across users via graph structures. **CROWN** (Ko et al., 2025) addresses cold-start users by augmenting the user representation with collaborative signals through a GNN-enhanced hybrid encoder. **LIME** (Ryu et al., 2025) incorporates the lifetime of news topics into news and user modeling via candidate-aware lifetime attention.

PLM-based method. **NAML-BERT** (Wu et al., 2021) replaces NAML’s CNN-based text encoders with BERT (Devlin et al., 2019), leveraging contextualized representations from a pre-trained language model.

Agent-based methods. **AgentCF** (Zhang et al., 2024) models users and items as autonomous LLM agents that mutually refine their preferences through simulated interactions. **ToolRec** (Zhao et al., 2024) treats the LLM as a tool-using agent that invokes external retrieval tools to find candidate items aligned with user preferences. **MACRec** (Wang et al., 2024) coordinates multiple specialized LLM agents (e.g., user analyst, item analyst, reflector) for recommendation. **AFL** (Cai et al., 2025) leverages an LLM as a feature learner that augments traditional recommenders with semantic features distilled from user histories.

For all agent-based baselines, we use GPT-4o-mini as the backbone LLM to ensure a fair comparison with SAND, which also uses GPT-4o-mini for anticipated body generation.

D Implementation Details

NAML backbone. We adopt NAML (Wu et al., 2019a) as the news encoder for both the teacher and the student. Following the original NAML configuration, we use 300-dimensional GloVe (Pennington et al., 2014) word embeddings, a CNN with 300 filters and a kernel size of 2, and additive attention to aggregate word-level representations. The maximum title length is set to 30, the maximum body length to 300, and the history length to 50.

Training. The teacher is pre-trained on actual bodies with the standard click-prediction objective until convergence. The student is then trained from scratch with the loss $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rec}} + \lambda_1 \mathcal{L}_{\text{KD}}^{\text{user}} + \lambda_2 \mathcal{L}_{\text{KD}}^{\text{news}}$. Following the sensitivity analysis in Section 3.5, we set $\lambda_1 = 0.5$ and $\lambda_2 = 0.5$ as the default. We use Adam (Kingma and Ba, 2014) with a learning rate of 0.001 and a batch size of 32, training for at most 10 epochs with early stopping on the validation set. The negative sampling ratio is set to $J = 4$.

Agent-based generation (Idea 1). We use GPT-4o-mini as the backbone LLM for f_{sum} , f_{gen} , and f_{ctrl} . The number of user clusters is set to $K = 3$, determined on the validation set. The Controller refines the policy over batches of size 300. The full set of policy values is reported in Appendix E, and the prompt templates are provided in Appendix F.

Hardware. All experiments are conducted on a single NVIDIA RTX PRO 6000 Blackwell GPU with 96 GB of memory.

E Generation Policy: Value Sets

The generation policy π is a 5-tuple $(\pi_{\text{tone}}, \pi_{\text{abs}}, \pi_{\text{spec}}, \pi_{\text{len}}, \pi_{\text{fmt}})$. Table 7 lists the possible values of each dimension, and we describe their semantics below.

Table 7: Value sets for each dimension of the generation policy π .

Dimension	Possible values
π_{tone} (tone)	very_neutral, neutral, mildly_expressive
π_{abs} (abstraction)	high_level, mixed, specific
π_{spec} (speculation)	0, 1, 2
π_{len} (length)	short, medium, long
π_{fmt} (format)	narrative, bullet

Tone (π_{tone}). This dimension controls how evaluatively the anticipated body is written. `very_neutral` enforces a strictly analytical tone with no evaluative adjectives or subjective expressions. `neutral` allows mild emphasis for clarifying relevance, while disallowing emotional or persuasive phrasing. `mildly_expressive` permits limited expressions of interest or anticipation, while sensational language remains prohibited.

Abstraction (π_{abs}). This dimension determines the granularity of the generated content. `high_level` focuses on overall themes or general directions, avoiding concrete numbers or fine-grained facts. `mixed` balances high-level themes with a small number of concrete details on aspects likely of interest. `specific` permits relatively concrete details (e.g., product categories, features, metrics), while still using speculative phrasing.

Speculation (π_{spec}). This dimension specifies how many speculative hypotheses are included in the anticipated body. A value of 0 excludes any speculation, describing only what the user broadly expects. 1 requires exactly one hypothesis, framed with uncertainty markers (e.g., “may,” “could,” “is likely to”). 2 requires two distinct hypotheses representing alternative expectations of the article’s content.

Length (π_{len}). This dimension controls the word count of the anticipated body: `short` for approximately 40–60 words, `medium` for 70–90 words, and `long` for 110–140 words.

Format (π_{fmt}). This dimension determines the structural form of the anticipated body. `narrative` produces a coherent paragraph in natural prose, while `bullet` produces a list of bullet points, each covering a distinct aspect of the expected content.

The initial policy $\pi_k^{(0)}$ is randomly initialized over the Cartesian product of these value sets and iteratively refined by the Controller LLM f_{ctrl} over user batches.

F Prompt Templates

We provide the full prompt templates used for the three LLMs (f_{sum} , f_{gen} , f_{ctrl}) of SAND. Placeholders enclosed in curly braces (e.g.,

{history_titles}) are replaced with the corresponding inputs at generation time. All three LLMs use GPT-4o-mini as the backbone.

F.1 Summarizer LLM (f_{sum})

The Summarizer LLM takes a user’s click history (titles only) and produces a textual profile of her interests, information-seeking pattern, and likely perspective. The output is used as the user summary s_u in Eq. (1) of the main text.

Given the following news titles that a user has clicked, infer the user’s primary interests, information-seeking patterns, and likely perspective when reading news.

Output format:

Primary interests: (2-3 specific domains)
Information-seeking pattern: (e.g., “focuses on financial impact”, “interested in technical details”, “follows political implications”)
Likely perspective: (e.g., “reads news as an investor”, “reads news as a tech enthusiast”)

[User Click History]
{history_titles}

F.2 Generator LLM (f_{gen})

The Generator LLM takes the user summary s_u , candidate news title t_{dc} , and the current group-wise policy $\pi_k = (\pi_{\text{tone}}, \pi_{\text{abs}}, \pi_{\text{spec}}, \pi_{\text{len}}, \pi_{\text{fmt}})$, and produces the anticipated body $b_{u,dc}$ in Eq. (2) of the main text. The prompt includes a one-shot example to anchor the news-like writing style; the example user summary s_u is itself an output of the Summarizer LLM for a representative user.

You are a user-centric expectation generator for news recommendation. Your task is to generate an expected news body that reflects what a specific user is likely expecting to read when seeing a news title.

[One-shot Example: User-Conditioned Expected News Body]

[User Preference Profile]
{example_s_u}

[Candidate News Title]
“Apple hints at a surprise announcement next week”

Expected Body (news-style, user-conditioned):

```
...
[End of Example]
-----

[User Click History - Titles Only]
{history_titles}

[Candidate News Title]
{candidate_title}

[OUTPUT]
Write the expected body in the SAME
STYLE as the example above, as if it
were the body of a news article (not a
meta description).

Use the following generation settings:

Tone: {pi_tone}
(Instruction corresponding to
{pi_tone}; see Appendix E.)
Abstraction: {pi_abs}
(Instruction corresponding to
{pi_abs}.)
SpeculationCount: {pi_spec}
(Instruction corresponding to
{pi_spec}.)
LengthBucket: {pi_len}
(Instruction corresponding to
{pi_len}.)
Format: {pi_fmt}
(Instruction corresponding to
{pi_fmt}.)

Generation rules:
Describe what this user is likely
expecting to read if they click the
article.
Write directly as article content; do
NOT use phrases such as “The article
is likely to...” or “Readers may
expect...”.
Do not assume access to the real
article body.
Avoid emotional or sensational
expressions.

Now generate the expected body.
```

F.3 Controller LLM (f_{ctrl})

The Controller LLM analyzes the batch-level loss $\Delta^{(t)}$, diagnostic samples $\mathcal{S}^{(t)}$, and accumulated memory $\mathcal{M}_k^{(t)}$, and produces an updated policy $\pi_k^{(t+1)}$ along with an updated memory in Eq. (3) of the main text. All inputs and outputs are exchanged as JSON. The batch size for policy refinement is set to 300 interactions.

You are a coordinator LLM that updates the generation policy used by an executor LLM.

You do NOT generate any expected-body text.
You ONLY (1) update the policy and (2) update the running policy summary.

You will receive a single JSON payload containing:

- allowed_policy_space
- current_policy
- performance_feedback
- diagnostic_samples
- running_policy_summary

Output ONLY a valid JSON object with exactly these two top-level keys:

- "updated_policy": an object with keys {tone, abstraction_level, speculation_count, length_bucket, format}
- "updated_running_policy_summary": an array of short bullet strings

Constraints:

- Each policy value must be chosen from allowed_policy_space.
- The summary must be concise (5-10 bullets), reflect persistent trends, and avoid batch-specific details.
- Do not include any extra keys, comments, or explanations.
- Do not infer or reconstruct the real article content.

The JSON payload supplied to the Controller LLM at each refinement step is structured as follows.

```
{
  "allowed_policy_space": {
    "tone": ["very_neutral",
"neutral", "mildly_expressive"],
    "abstraction_level":
["high_level", "mixed", "specific"],
    "speculation_count": [0, 1, 2],
    "length_bucket": ["short",
"medium", "long"],
    "format": ["narrative", "bullet"]
  },
  "current_policy": { ... },
  "performance_feedback": {
    "loss_expected": <float>,
    "loss_real": <float>,
    "ndcg5_expected": <float>,
    "ndcg5_real": <float>
  },
  "diagnostic_samples": [
    {"type": "failure",
"user_click_history_titles": [...],
"candidate_news_title": "...",
"generated_expected_body":
"..."},
    {"type": "success", ...}
  ],
  "running_policy_summary": [...]
}
```

At the first refinement step, the running_policy_summary field is initialized with a brief seed instruction (“Maintain 5–7 concise bullets summarizing persistent policy trends across itera-

tions.”). From the second step onward, the updated_running_policy_summary from the previous step is fed in as the running_policy_summary, serving as a persistent memory $\mathcal{M}_k^{(t)}$.

G Full Hyperparameter Sensitivity Analysis

Figure 4 reports the validation accuracy of SAND on MIND across the full 5×5 grid of distillation weights, $\lambda_1 \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ and $\lambda_2 \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. The main paper (Section 3.5) presents a subset of these results for readability; we provide the complete grid here.

The best validation accuracy across all three metrics is achieved at $(\lambda_1, \lambda_2) = (0.5, 0.5)$, which we adopt as the default configuration of SAND. The accuracy generally increases as both λ_1 and λ_2 become larger, with a notable jump at the top-right corner of the grid. A secondary peak is also visible at $(\lambda_1, \lambda_2) = (0.1, 0.2)$, where the news-side distillation appears to be particularly informative for an otherwise weak user-side alignment. Across the full grid, the variation in nDCG@5 stays within [0.6846, 0.7168], confirming that SAND is robust to the choice of distillation weights within the explored range.

H Policy Refinement Trajectories: Additional Clusters

We provide the per-batch nDCG@5 and policy trajectories for Clusters 1 and 2 of Adressa-1W in Figure 5, complementing the Cluster 0 result reported in Section 3.6. The number of batches differs across clusters, reflecting differences in cluster size. Across both clusters, the gap between actual and anticipated bodies generally narrows over the early batches, consistent with the pattern observed in Cluster 0. The specific refinement trajectories, however, differ from cluster to cluster—suggesting that f_{ctrl} adapts its policy updates to the characteristics of each user group.

I Qualitative Examples of Anticipated Bodies

To illustrate how SAND generates user-specific anticipated bodies, we present a case study on a single candidate news article shared by users from three different clusters of MIND-1W. The candidate is N84137: “Why Do Cats Meow?” (category: lifestyle/pets). For each cluster

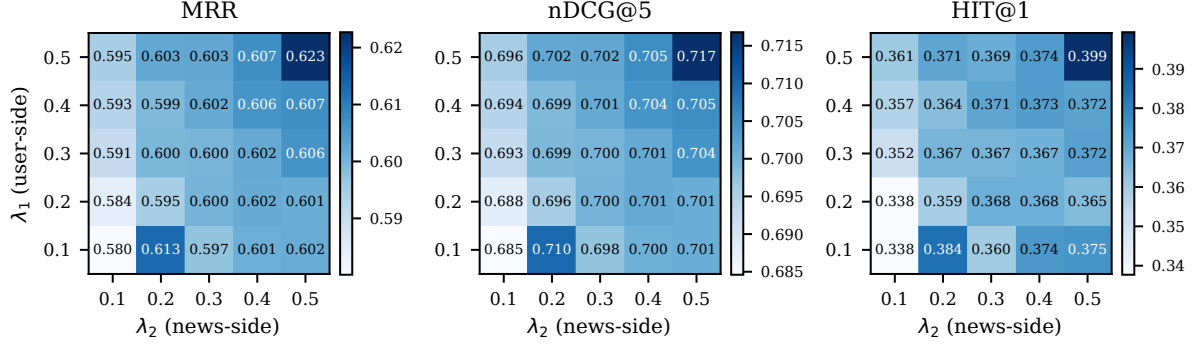


Figure 4: Validation accuracy of SAND on MIND across the full 5×5 grid of λ_1 (user-side KD weight) and λ_2 (news-side KD weight). The best configuration (λ_1, λ_2) = (0.5, 0.5) is selected as the default.

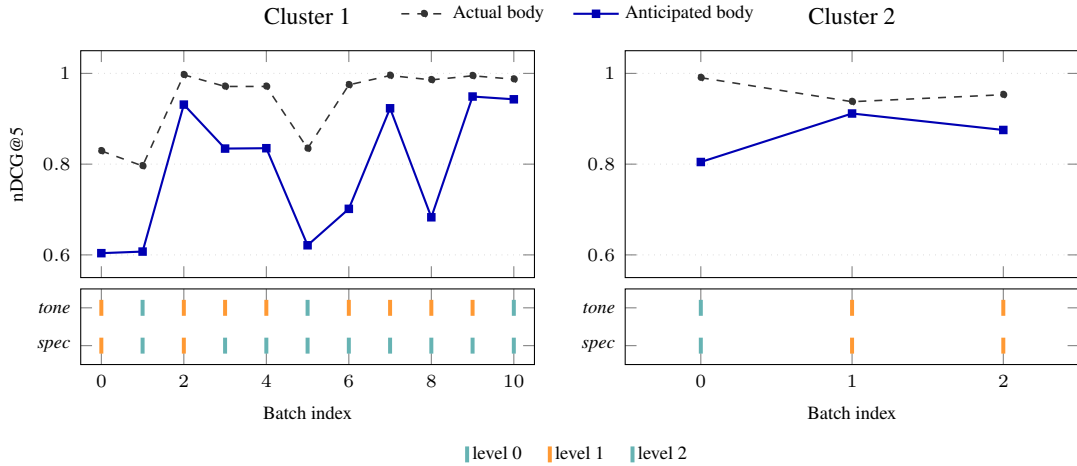


Figure 5: Per-batch nDCG@5 of the pre-trained NAML on Clusters 1 and 2 of Adressa-1W (top), and the corresponding evolution of the *tone* and *speculation_count* policy parameters (bottom). Each policy parameter has three ordinal levels (level 0/1/2), where higher levels indicate more expressive tone (*very_neutral*, *neutral*, *mildly_expressive*) or more speculation. As in Figure 3, these curves report the diagnostic signal measured by $\mathcal{R}$, not final recommendation accuracy.

$k \in \{0, 1, 2\}$, we pick two users ($u, u' \in \mathcal{U}_k$) and contrast their anticipated bodies $\tilde{b}_{u,d_c}$ and $\tilde{b}_{u',d_c}$ for the same d_c . Tables 8–10 report the preference summaries s_u (paraphrased for space; original phrasing preserved on our public repository) and the corresponding anticipated bodies. We highlight two observations.

Inter-cluster variation. The three clusters yield distinctly different framings of the same article. Cluster 0 users (politics- and social-issue-oriented) receive anticipated bodies that foreground *communication and human–animal interaction*, Cluster 1 users (legal/celebrity vs. automotive/sports interests) receive bodies emphasizing *behavioral interpretation and breed-level variation*, and Cluster 2 users (personal-narrative- and wellness-oriented) receive bodies that incorporate *emotional bonding*

and *practical owner advice*. This pattern supports (C2): the group-wise policy π_k effectively steers f_{gen} toward cluster-specific stylistic regimes even when the candidate topic lies outside the users’ primary interests.

Intra-cluster variation. Within each cluster, the two users still receive subtly different anticipated bodies, reflecting the per-user summary s_u that f_{sum} produces from $\mathcal{H}_u$. For instance, in Cluster 2 the wellness-oriented User 173 receives an anticipated body that explicitly mentions “strengthening bonds” between cats and owners, whereas User 27 (animal/family narratives) receives one centered on “communicative functions” and breed-specific patterns. This validates that s_u provides meaningful per-user conditioning on top of the group-wise policy.

Table 8: Anticipated bodies for two users in Cluster 0 on the candidate *N84137: “Why Do Cats Meow?”*.

	User 58	User 542
s_u	<i>Current events and political news, social justice and human-interest stories, health and public-safety issues; focuses on social implications and the impact of political decisions on communities; reads news with empathy and concern for broader societal implications.</i>	<i>Politics, environmental issues, cooking and food; follows public-opinion trends and political implications, while also seeking practical cooking tips and food trends; reads news as a politically engaged citizen and home cook.</i>
$\tilde{b}_{u,d_c}$	The article will explore the reasons behind cats’ vocalizations, focusing on the various types of meows and their meanings in feline communication. It will likely examine the evolutionary context of meowing, considering how domestic cats have adapted their vocalizations to interact with humans. The piece may also include insights from animal behaviorists or veterinarians on how different meows can signify specific needs or emotions, such as hunger, distress, or attention-seeking. ...	The article will explore the various reasons why cats meow, focusing on communication methods between cats and humans, as well as among cats themselves. It is expected to include insights into the different types of meows and what they may signify in terms of a cat’s needs or emotional state. Practical tips for cat owners on how to interpret their cat’s meowing and respond appropriately could be included, enhancing understanding and interaction. ...

Table 9: Anticipated bodies for two users in Cluster 1 on the same candidate as in Table 8.

	User 433	User 763
s_u	<i>Celebrity news, legal issues, social justice; interested in the societal implications of legal cases, celebrity culture, and human-interest stories; reads news as a socially conscious individual engaged with the impact of legal and ethical matters.</i>	<i>Automotive industry and vehicle performance, sports (esp. NFL), social issues and crime; follows product launches, team-management dynamics, and the community impact of crime; reads news as an automotive enthusiast, NFL fan, and concerned citizen.</i>
$\tilde{b}_{u,d_c}$	The article explores the various reasons behind a cat’s meowing, detailing how this vocalization differs from other forms of communication in feline behavior. It examines the context in which cats typically meow, such as during interactions with humans or in response to hunger and attention-seeking behaviors. The discussion may also touch on variations in meowing patterns among different breeds and individual personalities. ...	Cats meow primarily as a form of communication with humans, differing from their vocalizations with other cats, which are often more varied and complex. The article outlines the various reasons behind a cat’s meowing, including demands for attention, food, or companionship, as well as expressing discomfort or distress. It may also delve into the different types of meows and their meanings, highlighting how owners can interpret their pets’ needs. ...

Table 10: Anticipated bodies for two users in Cluster 2 on the same candidate as in Table 8.

	User 27	User 173
s_u	<i>Human–animal relationships (esp. dogs), personal and emotional narratives (family, health struggles), lifestyle and wellness (health tips, food, fashion); drawn to heartwarming stories involving animals and family; reads news as a compassionate individual valuing family and well-being.</i>	<i>Health and nutrition, celebrity news and pop culture, cooking and recipes; follows health-related recipes, celebrity lifestyles, and practical cooking tips; reads news as a health-conscious individual curious about celebrity influence on culture and trends.</i>
$\tilde{b}_{u,d_c}$	The article will provide an overview of the various reasons why cats meow, detailing the different contexts in which this vocalization occurs. It will explain the communicative functions of meowing, including how cats use this sound to interact with humans and other animals. The piece may also touch on the variations in meowing behavior among different breeds or individual cats, highlighting any notable patterns. ...	An exploration of the reasons behind the various vocalizations that cats produce, particularly the meow, as a form of communication. An overview of how different meows may indicate specific needs or emotions, such as hunger, discomfort, or a desire for attention. A look at the relationship between cats and their human companions, emphasizing how meowing may serve to strengthen bonds and facilitate mutual understanding. ...