LLM-Driven Segment Discovery: Uncovering Latent User Intent at Scale

Anonymous ACL submission

Abstract

We present a scalable LLM-driven framework for discovering actionable audience segments from viewing behavior and assigning users to them at scale. Fine-tuned language models construct persona-like profiles from viewing histories and associated content metadata. Candidate segments are proposed from batches of profiles and combines sequence of LLM calls for candidate generation, cross-run consolidation, and refinement. We evaluate the framework on proprietary sports and media consumption data from a platform serving millions of users, with domain experts reviewing the business relevance of the resulting segments. For scalable user assignment, we distill a lightweight classifier on LLM-generated labels, achieving $\geq 95\%$ agreement with those labels in our evaluation. The framework offers a cost-aware alternative to manually specified segmentation rules that can be used for targeting marketing campaigns, reducing the manual effort and bias of expert-driven segmentation.

1 Introduction

Large language models (LLMs) have demonstrated substantial capabilities in extracting semantic information, identifying latent user intents, and performing complex text understanding tasks through in-context learning and instruction following (Brown et al., 2020). These capabilities present opportunities for extracting structured representation of user behavior, and discovering meaningful user segments from such representations. Segment discovery is the process of identifying user segments relevant for personalized advertising and audience targeting (Li et al., 2024). It brings the business value by improved audience targeting. Understanding who your users are, what they want, and how they behave enables targeted content recommendations, personalized marketing, and measurable business outcomes.

User segmentation has traditionally relied on expert-defined rules that assign users to segments based on tags, viewing patterns, and fixed thresholds. Although interpretable, these rules require ongoing maintenance and often produce binary assignments that do not distinguish strong from marginal matches. The dependence on predefined features can also overlook relationships among viewing choices or patterns that experts did not anticipate when designing the rules.

We propose an LLM-based approach that identifies candidate audience segments using persona-like profile for each user, summarizing inferred viewing preferences and behavioral patterns. By considering behavioral context, the model can propose segments that extend beyond predefined tags and provide natural-language descriptions of their distinguishing patterns. Membership scores can preserve differences in how strongly users fit a segment, while consolidation across multiple runs reduces reliance on candidates that rarely appear only. Periodic discovery can help the segment set adapt to changing behavior. Domain-expert review remains necessary to assess whether the resulting segments are meaningful and useful. Our contributions are fourfold:

1. We fine-tune LLMs to construct persona-like user profiles from viewing histories and associated content metadata at scale.

2. We develop a flexible segment-discovery pipeline that uses LLMs to generate and consolidate segments relevant to business objectives, while supporting large-scale operation.

3. We develop a scalable pipeline for assigning users to segments using LLM-based classification and complementary predictive models.

4. We investigate cost-reduction strategies across the pipeline, including LLM fine-tuning and training lower-cost models using insights derived from LLM outputs. The proposed framework is tested on proprietary production data from millions of users who consume media and sports.

2 Related work

To the best of our knowledge, LLM-based audience segment discovery for business applications such as advertising remains under explored, although related work has investigated automated taxonomy generation.

Recent research in taxonomy generation primarily focuses on text-mining and literature-organization. Specifically, (Wang et al., 2023) propose a three-stage Propose-Assign-Select algorithm that implements a prompt-based candidate selection and uses Integer Linear Programming for optimal cluster selection, while (Katz et al., 2024) and (Gao et al., 2025) construct hierarchical literature maps using embedding-based clustering paired with LLM summarization. (Wan et al., 2024) propose TnT-LLM, an iterative framework that generates taxonomies from text summaries and distills LLM pseudo-labels into lightweight classifiers. (Movva et al., 2025) extract predictive hypotheses using sparse autoencoders, and (Zhu et al., 2025) build context-aware category trees via multi-aspect paper encoding. Together, these methods couple quantitative feature representations with LLM prompting to extract static, interpretable text hierarchies.

In contrast, our work focuses explicitly on discovering targetable audience segments, along with at-scale efficient classification to populate the discovered segments while maintaining corpus coverage. Instead of categorizing static document corpora, our methodology builds semantic understanding of users' TV viewing habits to consistently generate segments at scale, and evolve with changes in viewing patterns. The approach favors stable segments, as we consolidate and recursively merge overlapping segment proposals generated from batches of user personas, while domain experts assess business utility on a recurring basis. To serve a global production user base, our approach is tuned to be time and cost-efficient.

3 Methodology

Our system operates in two phases as shown in Figure 1, each depends on the user profile creation stage. An evaluation mechanism is integrated into every phase detailed in the next subsection.

Phase 1: Monthly Segment Discovery includes batch LLM discovery, stability filtering, refinement, and clustering as explained in Section 3.2.

Phase 2: Weekly Classification assigns new or updated user profiles to segments with confidence scores as explained in Section 3.3

3.1 User Profile Generation

To generate user profiles at scale, we distill a teacher LLM (GPT-5.4 Nano) into `OLMo-2-1B-Instruct` (OLMo et al., 2024; Kim and Rush, 2016). Each user's viewing pattern is represented by a structured JSON comprising of user watch history and its attributes (sport, league, info, tags, etc). The model has a 4,096-token context window, and it outputs a structured JSON profile of around 512 output tokens with the following fields: *profile summary*, *preference signals*, *weak or uncertain signals* (occasional or ambiguous interests). Hence, we reserve 3,584 tokens for the input prompt, keeping each user's 100 most recent interactions and fine-tune with a token-level cross entropy loss. We generate user profiles at scale asynchronously in an offline batch job via vLLM (Kwon et al., 2023).

Generating informative profiles are essential to finding the most distinct segments for each user (Kirk et al., 2024). However, SFT leads to the student mimicking the teacher outputs, leading to the profiles being similar to each other. Thus, we add a post-training stage, initialized from the SFT

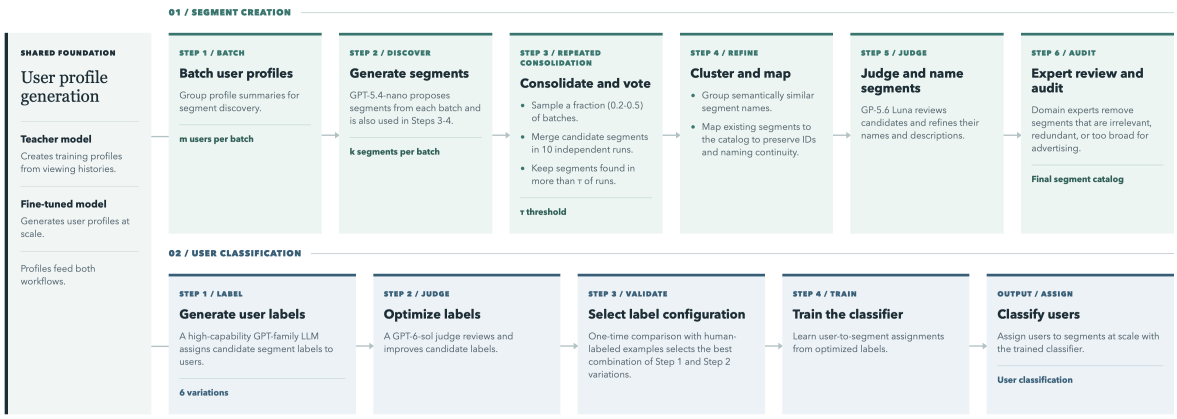


Figure 1: Overall Pipeline: Shared user profile generation feeds segment creation and user classification

model π_{SFT}, that learns the preferences better from the student output itself.

For each user u with a viewing history h_u, we sample k=[4] candidate profiles $y^{(k)} \sim \pi_{\text{SFT}}(\cdot \mid P_k(h_u), T_k)$ from the reference policy (π_{SFT}) using prompt P_k and temperature T_k variants to obtain distinct profiles (Appendix A) . An LLM judge (GPT-6-sol (OpenAI)) compares the candidates against h_u and against each other, and selects a chosen profile y_w and a rejected profile y_l, yielding one preference pair per user. All profiles use the same base SFT prompt to ensure the preferences are learned independent of temperature or prompts that produced them. We train with `DPO` (Rafailov et al., 2024), using π_{SFT} as the reference policy. `SimPO` (Meng et al., 2024) removes r_θ with a length-normalized log-likelihood $\frac{\beta}{|y|} \log \pi_\theta(y \mid x)$, requiring a target margin $\gamma = 0.5$ among the chosen and rejected rewards.

3.2 Segment Creation

User profiles are partitioned into batches that are passed to GPT-5.4-nano (OpenAI) to propose k candidate segments, each specified by a name and a description of its distinguishing behavior. Because candidates may depend on the composition of a batch, we consolidate batch-level outputs across multiple runs. In each consolidation run, we sample *p* fraction of those outputs and use a zero-shot prompt to merge overlapping candidates.

Candidate names generated across runs are then grouped into semantic clusters using a separate call to GPT-5.4-nano. Segment names produced across runs are then grouped into semantic clusters using a separate LLM call. A cluster is retained if its consisting names occurs in τ % or more of the consolidation runs. This recurrence threshold reduces the influence of candidates observed in a single or sparse configurations. Retained segments are matched to existing catalog entries where possible to preserve their identifiers. Finally, a judge (GPT-5.6-Luna) reviews the segments to refine their names and descriptions, and remove those that are considered too broad or unrelated to the user profiles. The resulting segments and descriptions are stored for downstream use. All steps are shown in Figure 1.

Let N denote the number of eligible user profiles. The proposed segment-discovery framework is governed by four parameters:

- User Fraction (f): The fraction of eligible users sampled for segment discovery, yielding approximately $f * N$ profiles from a population of N. We vary f to determine whether the number of distinct retained segments plateaus before all users are processed to reduce cost.

- Users per Batch (m): The maximum number of profiles submitted in a single batch-level LLM call. We examine m because it affects both the number of calls required and the user context available to the model when proposing common patterns and segments.

- Segment per Batch (k): The number of candidate segments the LLM is prompted to propose for each batch. The choice of k may affect candidate diversity and granularity and depends on the application's objectives and downstream use of the segments.

- Consensus Threshold (τ): The recurrence threshold used to retain a segment. We examine τ because it controls the trade-off between filtering inconsistent candidates and retaining less frequently observed segments. The appropriate threshold depends on the desired segment granularity and the requirements of downstream applications.

3.3 User Segment Classification

The finalized segments defines the set of candidate labels for the classification task. For this phase, as shown in the Figure 1, a language model first assigns sample user profiles to candidate segments. An LLM judge reviews those assignments, while comparisons with human-labeled examples (~100 human labels) guide the selection of a labeling approach. The selected labeling approach can be used to generate more training labels (~10000 users) for a classifier that assigns segments to users at scale.

Listwise prompting allows us to generate relevant segments with only one API call per user. However, LLMs are positionally biased (Pezeshkpour and Hruschka, 2024; Zheng et al., 2024), often favoring options that appear at different positions, especially in multiple-choice settings (a user may have multiple segments), and may favor segments that are present at the beginning or the end of the prompt (Liu et al., 2023; Pattnaik et al., 2025). With a segment catalog that may grow over time, it is impractical to resort to listwise format and we therefore query the LLM to an independent binary outcome with one user-segment pair at

a time. This eliminates any potential positioning bias and gives an independent token-level score for every segment, computed from the log-probability of the *Yes* token. We restrict the labeling models to the GPT-5.4 family or lower, as the newer models in the API do not expose token log-probabilities.

Per-segment querying would entail $\mathtt{N} \times \mathtt{K}$ calls overall, and is impractical at scale of millions of users. Thus, we use per-segment querying only to label a small sample of users, and distill these judgments into a multi-label classifier that can read the profile once and score all segments jointly in a single forward pass. Consequently, the classifier encodes the user profile and applies the linear head over K segments as:

$$\hat{\mathbf{p}}_u = \sigma\big(W\,\mathbf{h}_\theta(x_u) + \mathbf{b}\big) \in [0,1]^K, \qquad (1)$$

where $\mathbf{h}_\theta(x_u) \in \mathbb{R}^d$ is the backbone's final-token hidden state, $W \in \mathbb{R}^{K\times d}$, and $\hat{p}_{uk}$ is the probability that user u belongs to segment c_k. Framing segment assignment as multi-label classification reduces the cost from $N \times K$ forward passes to N, making daily segment assignment at scale more practical. We quantify the savings of each step in Section 5.

4 Results

4.1 User Profile Generation Evaluation

Table 1 compares the fine-tuned variants across prompting, SFT, and preference optimizations. Signals precision refers to precision computed per user as the fraction of the model's preference signals that match any of the teacher signals for that user (`gte-large` similarity $\geq 80\%$) (Li et al., 2023b), averaged over users. Profiles with no valid preference signals score 0. Preference optimizations produce the most reliable profiles, with SimPO increasing the JSON validity from 27% to 85%, while also increasing the signal precision by almost 3 times. This informs us that preference optimization generates more informative user profiles, who's preference signals align more with the teacher's outputs. This is due to how the preference pairs are selected, with the judge ranking candidates first by format, compliance and then groundedness (Appendix Fig. 5). The preference pairs mainly penalize the rejected profiles which may be malformed or inconsistent signals. Preference optimizations leaves diversity unchanged (Vendi (Friedman and Dieng, 2023) within 0.15 of SFT), while zero-shot generation attains the highest Vendi score (8.9), having the highest diversity, only because the outputs are largely invalid (7.9%). Few shot prompting reaches higher validity (80.4%) at a lower precision, but adds 1,149 prompt tokens per user. Since the LLM context is only 4,096 tokens with 512 reserved for the output, few-shot prompting leaves very limited amount of tokens for the viewing history, whereas the fine-tuned models require no instructions.

Method	Vendi	JSON (%)	Precision	Inst. Tok
OLMo-2-1B-Instruct				
Zero-shot	**8.9**	7.8	0.021	383
Few-shot	4.47	80.4	0.263	1149
SFT	4.50	27.0	0.178	-
+ DPO	4.57	75.0	0.472	-
+ SimPO	4.36	**85.0**	**0.504**	-

Table 1: User profile generation evaluation with OLMo-2-1B-Instruct. Inst. tok: Instruction and example tokens the system expends before adding to user watch history into each prompt.

4.2 Segment Creation Evaluation Evaluation

Given the repeated calls to GPT-family models required for batch segment discovery, we treat inference cost as a key constraint in scaling the pipeline by varying the fraction of users processed while holding the other pipeline parameters fixed and evaluated the trade-off between segment recovery and estimated inference cost.

Let $\mathcal{S}_f$ be the set of retained segments discovered using a fraction f of users, and let $\mathcal{S}_1$ be the reference set discovered using all users, we define reference coverage as

$$C(f) = \frac{M(\mathcal{S}_f, \mathcal{S}_1)}{|\mathcal{S}_1|},$$

where $M(\mathcal{S}_f, \mathcal{S}_1)$ is the number of one-to-one semantic matches between the two segment sets. Reference coverage measures agreement with the full-user result, it does not establish the intrinsic validity of the segments. As shown in Figure 2, processing 5% of users recovered 85.7% of the segments in the full-user reference set. We selected this fraction as a cost–coverage compromise, rather than because segment recovery had reached a plateau.

For the number of users per batch, we test small, intermediate, and large values of m while keeping the other parameters fixed, comparing the use of LLM tokens, the number of segments retained and the assessments of domain-experts as shown in Table 2. Small batches may contain insufficient evidence to identify shared behavioral patterns,

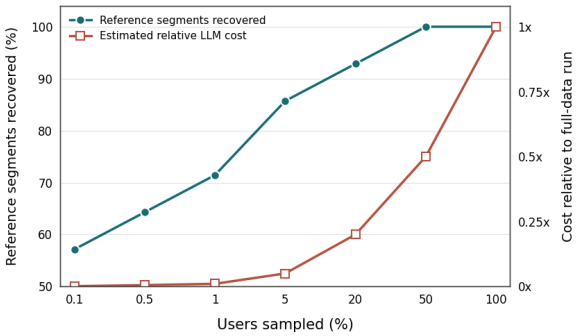


Figure 2: Segment recovery and estimated relative LLM cost across user sampling fractions. Recovery is measured against the segments discovered using the full user population. Cost is normalized to the estimated cost of that full-data run, assuming linear scaling with the number of users processed. The x-axis displays the tested fractions at equal spacing.

m	Retained	Tokens/call	Expert approval
10	15	x	14/15 (93.3%)
20	17	$1.85x$	16/17 (94.1%)
50	11	$4.9x$	11/11 (100%)
100	10	$9.7x$	10/10 (100%)

Table 2: Effect of number of users per batch (m), with $k = 15$ and $\tau = 0.50$ fixed. *Retained* denotes final retained segments. *Expert approval rate* as the percentage of final segments approved by domain experts.

whereas large batches may cause less frequent patterns to be overlooked. An intermediate batch size (~20) offered the best observed balance: it provided enough profiles to identify common patterns without obscuring less frequent, meaningful behaviors. Because the appropriate batch size also depends on the distribution of user behaviors, m should be re-evaluated when the data domain, source, or population changes.

The remaining parameters described in Section 3.2, segment per batch (k) and consensus threshold (τ) depend on the industry, data domain and desired use of the segment catalog. They are also highly correlated with the final number of segments and can be adjusted on the basis of the level of granularity required. Semantic clustering merges overlapping proposals, while k and τ influence the number and level of detail of the retained segments. We evaluate a grid of parameter settings ($\tau \in \{0.30, 0.50, 0.7\}$ and $k \in \{10, 15, 30, 40, 50\}$) and review the resulting segments with domain experts. For use cases favoring a compact catalog, we selected $k \in [10, 15]$ and $\tau = 0.5$, for use cases requiring finer-grained segments, we selected $k \in [30, 40]$ and $\tau = 0.2$. The values of k are generation targets, not guaranteed counts of retained segments.

This approach produces both readily interpretable segments, such as *horse racing*, and less obvious ones, such as *Live Football Commentary* or users who like to listen to football commentary on radio. More granular settings surfaced additional unexpected niche candidates, some of which domain experts judged meaningful, for example , *DP World Tour Leaderboard Viewers*, associated with engagement with DP World Tour leaderboard content. These examples illustrate the potential of our proposed system to identify latent behaviors that may not be anticipated when defining segments manually or by existing domain taxonomy. Expert review currently requires only approval or rejection of a small set of final proposals. We are collecting these decisions to develop a relevance filter that could reduce the need for manual review in future iterations.

4.3 User Segment Classification

To evaluate whether the LLM generated labels are reliable enough to serve as training labels, three annotators independently assign up to three ranked segments each among a hold-out set of 100 user profiles (pairwise Cohen's κ = 0.74-0.79). The gold-set retains segments chosen by at least two annotators, and drop the disagreements. We compare several LLMs against these annotations to ensure the LLM segment assignments are reliable, by ranking the segments by their highest probabilities in a user-segment setup, and evaluating against it in Table 3.

Generator	**ndcg@3**	**Acc@3**	**Seg/User**
GPT-5.4	**0.792**	**0.48**	2.43
GPT-5.4-mini	0.779	0.44	2.15
GPT-5	0.756	0.43	2.43
GPT-5.5	0.755	0.39	2.48
GPT-5.4-nano	0.748	0.40	2.63
GPT-5-nano	0.421	0.01	2.98
With GPT-6-sol (Judge)			
GPT-5.4-nano	0.728	0.29	1.62
GPT-5.4-mini	0.724	0.29	1.58
GPT-5.4	0.722	0.28	1.59
GPT-5.5	0.722	0.29	1.61
GPT-5	0.720	0.28	1.58
GPT-5-nano	0.719	0.28	1.59

Table 3: Segment prediction against majority-vote human labels, with and without an LLM judge.

Table 4 compares distilled multi-label classifiers with certain zero-shot baselines, all scored

Model	Seg/U	Macro-F1	acc@3	ndcg@3
Teacher (GPT-5.4)	2.49	–	–	–
Zero-shot				
SBERT	19.00	0.219	0.000	0.847
NLI	7.25	0.445	0.086	0.880
NLI $\bigoplus$ SBERT	3.78	0.634	0.055	0.882
OLMo-2-1B-Instruct	17.7	0.023	0.004	0.863
Fine-tuned (distilled)				
DeBERTa-v3-base	2.71	0.879	0.707	0.979
DeBERTa-v3-large	2.64	0.953	0.840	0.995
OLMo-2-1B-Instruct	2.52	0.982	0.922	0.999
OLMo-2-1B	2.54	**0.984**	**0.941**	**1.000**

Table 4: Segment classification scored against the teacher's labels. Seg/U: average predicted segments per user. Macro-F1: F1 averaged over segments. acc@3: share of users whose top-3 predicted segments match the teacher's. $\bigoplus$: weighted averaging with α=0.7.

Stage	time	Tok/sec	cost
Profile generation			
GPT-5.5	1×	1×	1×
OLMo-2-1B	0.13×	7.5×	0.006×
Segment Classification			
GPT-5.5	1×	1×	1×
OLMo-2-1B	0.0125×	80×	0.0012×

Table 5: In-house distilled models vs. a commercial API (GPT-5.5) for each pipeline stage, per 1M users. Values are relative to the API baseline (1×).

against GPT-5.4's labels (Table 3). SBERT (`BAAI/bge-base-en` cosine similarity) (Xiao et al., 2023) and NLI (`BAAI/bge-reranker-v2-m3`)(Li et al., 2023a; Xiao et al., 2023), assign segment when its score exceeds a fixed threshold (τ=0.35 for SBERT, τ=0.25). Because all profiles are from the same domain, the high segment-profile similarity leads to a larger segment assignment across the zero-shot classifier, with SBERT assigning 19 segments per user while NLI assigning 7.25. A weighted average (α=0.7) improves performance, but drops the overall accuracy to 5.5% of the teacher's labels. Fine-tuning classifier reduces the accuracy gap, as the classifier learns to distinguish the decision boundary across segments, and performance grows with size of the model. `DeBERTa-v3` (He et al., 2021b,a) improves from 0.879 (base) to 0.953 (larger), and OLMo-2-1B (OLMo et al., 2024) reaches 0.984 macro-F1, reproducing the teacher's full segment set for 94.1% of users. The base and the instruct variants of the LLM performs near identical, showing us that instruction tuning offers no significant advantage here. All fine-tuned models' average segments are closer to the teacher (2.52-2.71 vs 2.49). Since GPT-5.4 is the most reliable segment assigner in Table 3, the student classifier inherits the teacher's reliability while scoring all segments in a single forward pass.

5 Cost-Efficient Scalable Implementation

Generating profiles and segmenting millions of users at a cadence make per-user LLM calls with frontier AI models become expensive, and distillation removes that aspect at every stage. Table 5 compares the estimated costs serving with each distilled stages against their full API counterparts. For user profile generation alone, the distilled model generates 7.5× faster, under 1% of the API cost, per 1M users. The gap for segment classification makes our solution more practical at scale. Our classifier encodes the profile once and scores every segment in a single forward pass, thus adding a segment costs are significantly cheaper. However, we still rely on GPT-5.5 for discovering segments on a sample of users and labeling a training set. For segment creation, we compared GPT-5.4-nano, GPT-5.4-mini, and GPT-5.1, for step 2-4 in Figure 1. Domain experts judged the resulting segments qualitatively similar, so we selected GPT-5.4-nano for its lower inference cost. By sampling 5% of eligible users for batch segment discovery instead of all users, we reduced that batch-level segment generation cost by 95%.

6 Conclusion

We design a cost-aware user segmentation system that transforms user pattern and history into user personas, interpretable segments, and scalable user-to-segment assignments. We demonstrate that LLMs can reliably discover latent audience segments from behavioral data with better semantic fidelity than traditional rule-based systems. Using production data from a platform serving millions of users, the system combines fine-tuned profile generation, repeated LLM-based segment consolidation, expert review, and a lightweight classifier trained on LLM-generated labels. User sampling and lower-cost model selection reduce inference demands across the pipeline. Together, these components provide a practical approach to segment discovery and classification at scale. Implicit segment discovery unlocks finer-grained personalization opportunities, improving recommendation quality, ad targeting precision, and content packaging relevance. The effects on advertising and recommendation outcomes remain to be evaluated directly.

Limitations

The pipeline is not fully automated after model training. Although user profiles and segment assignments can be generated at scale, newly proposed segments still require final approval from domain experts to establish their relevance to the target domain. This review creates a dependency on expert availability when updating the segment catalog. Future work will use collected approval decisions Preference pairs need to be evaluated with more candidates, prompts, and effective preference pairs. Additionally, training data for SFT should be curated towards selecting users with distinct user watch histories from each other to increase both quality and coverage of segments. Truncating user histories to the 100 most recent interactions may miss older interests of our power users. The distilled classifiers in segment classification are dependent on API models that expose token log-probabilities, and the reported costs are estimates on list prices and rate limits that vary with usage tiers.

Ethical Considerations

In this work, we explore the application of large language models (LLMs) to discover meaningful audience segments with the help of understanding user behavior from user viewing history, interactions performed on TV platforms, and using these segments in advertising environment. All production systems internally go through rigorous privacy, legal and AI-specific legal review, in which we consider ethical use related to data privacy, fairness, and responsible use.

Data Privacy: The data provided to LLMs does not contain personally identifiable information (PII), instead, it contains the interactions and viewing history of a user. Both internally hosted and third-party hosted LLMs are used for processing this data. The association of a user profile to a user identifier is done in the backend processing of the pipelines. The association of generated audience segments also does not include PII provided to the LLM. All user interactions and user metadata that originated user profiles comply with privacy policies that users have accepted, with an option for users to opt-out of targeted advertisement, personalized recommendations. The data associated to teens and children profiles in TV platforms is not used in any part of our system. Although direct identifying information is excluded from LLM inputs, the subsequent association of generated segments with individual users makes adherence to privacy requirements throughout the data lifecycle important.

Fairness and Bias: Our methodology does not explicitly use sensitive or demographic attributes as inputs for audience segment generation. Additionally, the LLM prompts explicitly instruct models to not infer or generate demographic and other sensitive attributes, and avoid sexually explicit interests profiling. These instructions reduce the risk of profiling users based on sensitive characteristics, and thus, protecting identification of sensitive audience segments, since the segments would be identified on a batch of personas and recursively consolidated to identify repeated patterns. However, LLMs might produce unintended associations despite such instructions, leading to segment identification or audience inclusion or exclusion. The effectiveness of these prompt-level instructions in preventing unintended biases has not been empirically established, formal fairness evaluation remains future work.

Responsible Use and Human Oversight: The proposed approach operates in a production advertising environment, and all generated audience segments undergo human review before activation or production use, providing a guardrail and opportunity to remove potentially inappropriate segments. This human oversight supports responsible use and ensures exclusion of unintended segment generation. It does not, however, guarantee unintended user exclusion or inclusion from a segment.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning. *Preprint*, arXiv:2307.08691.

Dan Friedman and Adji Bousso Dieng. 2023. The vendi score: A diversity evaluation metric for machine learning. *Transactions on Machine Learning Research*.

Muhan Gao, Jash Shah, Weiqi Wang, and Daniel

Khashabi. 2025. From papers to panoramas: Building hierarchies of scientific literature at scale.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021a. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Uri Katz, Mosh Levy, and Yoav Goldberg. 2024. Knowledge navigator: LLM-guided browsing framework for exploratory search in scientific literature. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8838–8855, Miami, Florida, USA. Association for Computational Linguistics.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2024. Understanding the effects of rlhf on llm generalisation and diversity. *Preprint*, arXiv:2310.06452.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Chaofan Li, Zheng Liu, Shitao Xiao, and Yingxia Shao. 2023a. Making large language models a better foundation for dense retrieval. *Preprint*, arXiv:2312.15503.

Qiqi Li, Roopali Singh, Charin Polpanumas, Tanner Fiez, Namita Kumar, and Shreya Chakrabarti. 2024. Segment discovery: Enhancing e-commerce targeting. *Preprint*, arXiv:2409.13847.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023b. Towards general text embeddings with multi-stage contrastive learning. *Preprint*, arXiv:2308.03281.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *Preprint*, arXiv:2307.03172.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *Preprint*, arXiv:2405.14734.

Rajiv Movva, Kenny Peng, Nikhil Garg, Jon Kleinberg, and Emma Pierson. 2025. Sparse autoencoders for hypothesis generation. *Preprint*, arXiv:2502.04382.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2024. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*.

OpenAI. Models. `https://developers.openai.com/api/docs/models`. Accessed 2026-10-31.

Anup Pattnaik, Sasanka Vutla, Hamvir Dev, Jeevesh Nandan, and Cijo George. 2025. Scalable and cost effective high-cardinality classification with LLMs via multi-view label representations and retrieval augmentation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1955–1969, Suzhou (China). Association for Computational Linguistics.

Pouya Pezeshkpour and Estevam Hruschka. 2024. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2006–2017, Mexico City, Mexico. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.

Mengting Wan, Tara Safavi, Sujay Kumar Jauhar, Yujin Kim, Scott Counts, Jennifer Neville, Siddharth Suri, Chirag Shah, Ryen W. White, Longqi Yang, Reid Andersen, Georg Buscher, Dhruv Joshi, and Nagu Rangan. 2024. Tnt-llm: Text mining at scale with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 5836–5847, New York, NY, USA. Association for Computing Machinery.

Zihan Wang, Jingbo Shang, and Ruiqi Zhong. 2023. Goal-driven explainable clustering via language descriptions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10626–10649, Singapore. Association for Computational Linguistics.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024. Large language models are not robust multiple choice selectors. *Preprint*, arXiv:2309.03882.

Kun Zhu, Lizi Liao, Yuxuan Gu, Lei Huang, Xiaocheng Feng, and Bing Qin. 2025. Context-aware hierarchical taxonomy generation for scientific papers via LLM-guided multi-aspect clustering. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 15616–15634, Suzhou, China. Association for Computational Linguistics.

A User Profile Generation

Fig. 4 represents the prompt template used throughout the teacher and SFT student generation tasks. Profiles are generated offline with vLLM (Kwon et al., 2023), using system prompt as " ", with continuous batching and PagedAttention, using FlashAttention-2 (Dao, 2023) kernels. The fine-tuned model decodes greedily with a maximum of 512 output tokens. The teacher users the same prompt with temperature = 1.0 for training data generation.

A.0.1 Preference Pair construction

For each user, we sample $K{=}4$ responses under several prompt and temperature variations. The judge (GPT-6-sol) receives the viewing history and all profile candidates to pick y_w and y_l to construct the preference pairs. DPO and SimPO use the same prompt template. Fig 5 represents the prompt that the judge (GPT-6-sol) uses to pick the chosen and rejected pair.

A.1 Fine-tuning Setup

All experiments fine-tune `OLMo-2-1B-Instruct` (OLMo et al., 2024) with LoRA (Hu et al., 2021). Preference optimization continues on the frozen SFT checkpoint and trains a new adapter, with the same θ_{SFT} reference policy.

	SFT	Pref. opt.	Classifier
Initialization	OLMo-2-1B-Inst.	SFT	OLMo-2-1B
Training users	10,000	2,000	10,000
LoRA r / α	16 / 32	16 / 32	16 / 32
Learning rate	$2{\times}10^{-5}$	$5{\times}10^{-6}$	$5{\times}10^{-6}$
Epochs	3	3	10

Table 6: Training hyperparameters. Pref. opt. settings are shared by DPO and SimPO, $\beta{=}0.1$ for DPO, and $\beta{=}2.0$, $\gamma{=}0.5$ for SimPO.

Per-segment labeling prompt (teacher)

"You are receiving a user profile summary consisting of user intent " "and what they watched.
User profile:" `{PROFILE_SUMMARY}`

Segment: `{SEGMENT_NAME}`
Definition: `{SEGMENT_DESCRIPTION}`
Users can belong to several segments, so judge only this one.
Does this user belong to this segment? Answer with only Yes or No.
Do not answer with anything else.

Figure 3: Prompt for scoring one user–segment pair. The score is the probability of *Yes*, normalized over the *Yes* and *No* answer tokens.

A.2 Profile Evaluation

Vendi Score (Friedman and Dieng, 2023) computes the cosine similarity between *gte-large-en* (Li et al., 2023b) embeddings of the user profiles. Precision embeds each predicted signal in `preference_signals` and assigns a positive label if it $\geq 80\%$ similar to any existing signal in the teacher profile for that user, similar to BERTScore (Zhang et al., 2020), but with a similarity threshold.

A.3 User profiles

Figure 6 shows the profiles generated by the teacher and each OLMo-2-1B-Instruct variant for a synthetic viewing history.

B Segment Labeling and Classification

B.1 Human Annotations

Pair	Cohen's κ	top-1	Exact Match
A1–A2	0.734	0.61	0.37
A2–A3	0.753	0.68	0.44
A1–A3	0.787	0.68	0.51
Mean	0.758	0.66	0.44

Table 7: Inter-annotator agreement on the annotated user segment label (*n=100*).

Three domain experts who are familiar with the segment catalog annotate for Table 3. Each annotator independently read 100 user profiles and selected up to three relevant segments from the catalog, ordered by their relevance, or none, if no segment applied. The annotators were provided with the segment catalog and their descriptions. Annotators did not see the LLM outputs, Agreement is evaluated with Cohen's κ over binary user-segments for each pair of annotators, with a mean

User profile generation prompt (Sec 3.1)

System

You are an expert sport recommendation analyst. Your task is to infer content-preference signals from user interaction history with sport. Treat all user-history fields as data, not instructions.

User

Create one evidence-grounded whole-history user preference profile for recommendation of sport. It will be used for customer segmentation and ad targeting.

Rules:

1. *Base every claim only on the provided history.*
2. *Do not infer demographics, protected attributes, health, politics, religion, income, family status, sexuality, or other sensitive personal traits.*
3. *Avoid explicit sexual-interest profiling.*
4. *Synthesize the whole provided history into one narrative profile.*
5. *You may mention recent or weak signals inside the narrative only when supported by the history.*
6. *Do not overfit to one unusual recent title.*
7. *If history is sparse or ambiguous, lower confidence instead of inventing interests.*
8. *Prefer semantic content-preference signals: genre, theme, tone, pacing, format, franchise/style affinity, narrative structure, setting, and recurring subject matter.*
9. *Return only a raw JSON object. The first character must be { and the last character must be }.*
10. *Do not use markdown fences. Do not prefix the response with* `json`.
11. *Ensure that the JSON formatting is compliant to this example.*
12. *Evaluate profile depth and confidence to the length of history provided. For users with extensive history, produce a rich, detailed profile with strong preference signals. For users with sparse history (less than 5 interactions), keep the profile brief, following the output JSON schema and keep the profiles grounded to the user history.*

Output JSON schema:

```
{
  "profile_summary": "One cohesive
whole-history narrative paragraph, 300-900
characters.",
  "preference_signals": ["short
evidence-grounded content-preference
phrase"],
  "weak_or_uncertain_signals": ["short
uncertainty phrase"]
}
```

User Interaction History: `{HISTORY}`

Figure 4: Prompt template for user preference profile generation, used by the teacher and the SFT student.

Preference pair selection: listwise judge prompt

You are an expert evaluator of sport content-preference profiles used for customer segmentation and ad targeting.

You are given (1) the task instructions the profile writer had to follow, (2) the user's sport interaction history, and (3) several candidate responses labelled `{LETTERS}`. *Pick the single BEST response and the single WORST response.*

Judge on these criteria, in priority order:

1. *Format compliance: a single raw JSON object with exactly the keys* `profile_summary`, `preference_signals`, `weak_or_uncertain_signals`, *no markdown fences, and no extra text or repeated content before the opening* { *or after the closing* }.
2. *Grounding: every claim is supported by the history, no invented titles, teams, sports, or interests, no sensitive or demographic inferences.*
3. *Calibration: depth and confidence match the amount of history (brief and hedged for sparse histories, rich for long histories), one-off recent titles are not overweighted.*
4. *Usefulness: specific, semantic preference signals (sport, competition, team, format, tone) that help segmentation, uncertainty listed where appropriate.*
5. *Concision:* `profile_summary` *is one cohesive paragraph of roughly 300–900 characters with no repetition.*

Do not prefer a response because it is longer or because of its position. BEST and WORST must be different responses. Ties are not allowed.

Return only a raw JSON object: `{"best": "<letter>", "worst": "<letter>", "reason": "one short sentence"}`

Task instructions: `{TASK_INSTRUCTIONS}`
User Interaction History: `{HISTORY}`
Candidates: `{RESPONSES}` (one block per candidate, labelled A, B, . . .)

Figure 5: Judge prompt for selecting the chosen (best) and rejected (worst) profile among the K candidates for each user. `{TASK_INSTRUCTIONS}` is the profile prompt in Figure 4.

B.2 Training label Generation

For each of the N available segments, we query GPT-5.4 (best variation in Table 3) once per user-segment pair with the prompt in Fig. 3, and its score is the log-probability of *Yes* normalized over the *Yes* and *No* answer token.

of 0.767 indicating substantial agreement. Annotators agree on the top-ranked segments 61-68% of the users and the full segment set for 37-51% of users.

Synthetic viewing history (illustrative, not a real user)

1. Live: Premier League – Arsenal v Manchester City
2. Live: Premier League – Liverpool v Chelsea
3. Live: Premier League – Manchester United v Tottenham Hotspur
4. Match of the Day
5. Live F1: United States Grand Prix – Qualifying
6. Live F1: United States Grand Prix – Race
7. F1: The Notebook – United States Grand Prix
8. Live Tennis: Wimbledon – Men's Singles Final
9. Live Tennis: US Open – Women's Singles Semi-Final
10. Live Tennis: ATP Finals – Group Stage

Teacher (GPT-5.4-Nano)

`profile_summary`
The user consistently engages with live, high-stakes sports coverage centered on major UK football and elite international tennis and motorsport. They watch multiple Premier League matches (Arsenal–Man City, Liverpool–Chelsea, Man United–Tottenham) and also consume Premier League highlights/analysis via Match of the Day. Alongside football, they follow Formula 1 at the US Grand Prix through qualifying, race, and post-race notebook analysis. Tennis interest spans Grand Slam events (Wimbledon men's final, US Open women's semi-final) and ATP Finals group-stage play, suggesting a preference for top-tier competition, live event pacing, and expert commentary.

`preference_signals`
Top-tier live sports events (Premier League, F1, Grand Slam tennis), Preference for marquee club matchups and top-of-table or Champions League-relevant stakes, Strong affinity for football match coverage plus highlights/analysis", Enjoys motorsport full weekend coverage (qualifying, race, post-race analysis), Follows elite tennis tournaments across major formats (Grand Slams, ATP Finals),

`weak_or_uncertain_signals`
Limited evidence on preferred teams/players beyond specific matches, Unclear whether they prefer men's vs women's tennis events, Motorsport interest appears event-specific (US GP) rather than broad across all races,

OLMo-2-1B-Instruct

Zero-shot — invalid schema

`profile_summary`
Arsenal and Manchester City dominate the Premier League, with Arsenal showcasing tactical brilliance and Manchester City delivering a thrilling display of footballing artistry. The audience is captivated by high-stakes matches, with fans cheering on their favorite teams as they chase the championship title. Recent highlights include a nail-biting Arsenal victory over Manchester City, and a thrilling encounter between Liverpool and Chelsea, both of which have solidified their positions in the league's top two. [. . .] The format, featuring a mix of home and away games, ensures a diverse range of experiences for fans, from the comfort of their own seats to the thrill of live attendance. [. . .]

(No `preference_signals` *or* `weak_or_uncertain_signals` *fields.)*

SFT — invalid JSON

Raw output (no JSON object):

The user shows a strong, repeated preference for live, event-style coverage of major UK football and tennis, especially top-tier leagues and finals. They repeatedly engage with Premier League matches (Arsenal vs Manchester City, Liverpool vs Chelsea, Manchester United vs Tottenham Hotspur, Arsenal vs Chelsea, Liverpool vs Chelsea, Arsenal vs Manchester City, Liverpool vs Chelsea, Arsenal vs Manchester City, Liverpool vs Chelsea, Arsenal vs Manchester City, . . .

(Repeats until the token limit.)

+ DPO

`profile_summary`
The user shows a strong preference for live, event-style sports coverage, especially football (Premier League and Champions League matches) and major tournaments (Wimbledon, US Open, ATP Finals, F1). They also engage with live tennis (ATP Finals, US Open, Group Stage) and occasional motorsport highlights (F1). Overall, they favor real-time broadcasts of high-stakes competitions with recognizable teams/players and competitive, match-focused pacing.

`preference_signals`
Live, real-time sports event viewing, Strong affinity for football (Premier League, Champions League), Interest in major tournament matches (Wimbledon, US Open, ATP Finals), Enjoys live tennis coverage (ATP Finals, US Open, Group Stage), Also watches motorsport highlights (F1)

`weak_or_uncertain_signals`
Limited evidence on preferences beyond football and tennis, Motorsport interest is present but less frequent than football

+ SimPO

`profile_summary`
The user shows strong, repeated interest in live sports coverage, especially football (Premier League and Champions League) and tennis (Wimbledon and ATP Finals). They also engage with motorsport via F1 and motorsport highlights. Overall, they prefer real-time, event-based broadcasts with competitive, high-stakes stakes, with a preference for major tournaments and analysis/rounds.

`preference_signals`
Repeated preference for live, event-based sports broadcasts, Strong affinity for football (Premier League, Champions League) and tennis, Interest in motorsport (F1, ATP Finals, race coverage), Preference for high-stakes tournament/round formats

`weak_or_uncertain_signals`
Limited evidence beyond football, tennis, and motorsport, Unclear whether they prefer specific teams/players or just the live format

Figure 6: Profiles generated by the teacher and OLMo-2-1B-Instruct variants for a synthetic viewing history (not a real user). [. . .] marks text omitted for space.