Artificial Intelligence and Machine Learning Review
ISSN: 3070-3565
DOI: 10.69987/AIMLR.2024.50111

A Comparative Evaluation of LLM-Generated Semantic Tags versus Classical Text Features (TF-IDF, LDA, BERT Embeddings) for User-Interest Enrichment in Short-Video Recommendation

Tianxing Tang[1], *Mingzhuo Yu*[1.2]
[1] *Translation and Localization Management, Middlebury Institute of International Studies, CA, USA*
[1.2] *Computer Science, Northeastern University, MA, USA*

Keywords

short-video recommendation; user-interest features; large language models; feature comparison

Abstract

User-interest features derived from textual metadata are central to short-video recommendation, yet the relative value of different semantic signal sources has not been rigorously quantified on modern public benchmarks. This paper presents a controlled comparative evaluation of four families of textual features used to enrich user-interest vectors in short-video feeds: sparse TF-IDF weighted keywords, Latent Dirichlet Allocation (LDA) topic distributions, sentence-level BERT embeddings, and tag sets generated by an open-source large language model (LLM). Using KuaiRand-Pure, MicroLens-100K, and KuaiRec as primary datasets and MIND-small as a cross-domain probe, we assess each signal along four axes: feature separability, mutual information with downstream behaviour, accuracy on next-item and click-through prediction tasks using SASRec, BERT4Rec, DIN, DIEN, and SIM backbones, and offline computation cost. LLM-generated tags attain the best downstream accuracy on most settings, with AUC gains of 0.9 to 1.6 points over TF-IDF on CTR tasks and HR@10 gains of 2.1 to 3.4 points on sequential tasks, while BERT embeddings remain competitive when LLM inference budget is constrained. LDA is dominated on every axis except interpretability. The observed gains are moderate rather than uniform, and we identify scenarios in which sparse lexical features remain Pareto-optimal once serving latency is accounted for.

1. Introduction

1.1. Background and Motivation

Short-video platforms such as Kuaishou, TikTok, and WeChat Channels now deliver the bulk of online advertising exposure in East Asia and North America, and their revenue is driven by how precisely the recommender infers each user's evolving interests from sparse interaction histories. The dominant industrial practice is to aggregate multi-signal engagement traces, including clicks, watch-completion ratios, likes, comments, and in-feed searches, into a low-dimensional interest vector that feeds click-through rate (CTR) and conversion rate (CVR) predictors [1]. Textual metadata attached to each video is the primary lever for injecting semantics into this vector, because video-ID embeddings alone suffer from long-tail sparsity and rapid content churn under weekly catalogue turnover. Descriptive text is abundant yet noisy, mixing promotional slogans, colloquial abbreviations, and emoji, so the choice of how to distill this text into a numeric feature has direct consequences on ranking accuracy, cold-start coverage, and serving budget [2].

The choice of text-feature pipeline is a first-order engineering decision, yet the four candidate families of sparse weighted keywords, latent topics, contextual embeddings, and LLM-produced tags are usually compared only within single systems or ablation studies, rather than head-to-head on public benchmarks with matched protocols. Recent work applying large language models to recommender feature engineering has argued that instruction-tuned decoders can

produce cleaner, more transferable semantic units than noisy user-generated text, yet the cost of such signals and their marginal value over strong classical baselines remain largely unquantified on open data [3].

1.2. Research Question and Contributions

We ask a narrow empirical question: given a fixed downstream architecture and identical raw text, which textual feature family delivers the best trade-off between interest-representation quality and computational cost in short-video recommendation? The question is deliberately scoped to four evaluation axes, four feature families, and five backbones, so that every observed difference can be attributed to the feature rather than to confounding architectural changes [4]. Formulated in this way, the problem lends itself to a clean factorial design over publicly available data, and yields guidance that practitioners can translate into concrete feature-platform choices.

A. Four Semantic Signal Sources Compared

The four candidates span the methodological spectrum used in production systems. TF-IDF with logarithmic term weighting represents the sparse lexical baseline, with engagement-weighted aggregation on the user side. LDA [5] represents topic-level semantics at three granularities of K equal to 50, 100, and 200, chosen to cover short, medium, and fine-grained topic pools. Word2Vec skip-gram embeddings were also evaluated but were dominated by their contextual successor on every axis, so the main tables report only the contextual variant as the dense-embedding representative, with the earlier distributed embedding retained only in the supplementary comparison [6]. LLM-generated tags are produced by prompting an open-source decoder in a recommendation-driven manner, with schema details deferred to Section 3.

B. Four Evaluation Dimensions

Evaluation proceeds on four axes. Separability is measured via silhouette coefficient and k-nearest-neighbour label purity on category-labelled videos, giving a geometry-level view independent of any downstream task. Mutual information between each feature and the binary click label provides an information-theoretic lower bound on downstream utility [7][8]. Downstream accuracy is reported on five backbone recommenders, covering next-item prediction and CTR and CVR prediction. Offline feature-production cost is measured in GPU-hours and storage footprint, and combined with serving-latency considerations in a Pareto-frontier analysis [9]. Our contribution is the comparison protocol itself: to our knowledge no public study reports all four axes on the chosen short-video benchmarks under a shared evaluation harness, and the conclusions we draw about when LLM tags are worth their cost are new to the literature [10].

2. Related Work

2.1. Classical Text Features for User-Interest Enrichment

A. Sparse Lexical Features

Weighted bag-of-words representations remain the default text feature in industrial retrieval pipelines because they are deterministic, cheap, and interpretable. They are also the natural format into which engagement-weighted co-occurrence counts, such as likes, watches, and shares, can be folded as term weights, preserving the sparse look-up profile that downstream retrieval infrastructure is already optimised for [11]. Extensions address the short-text sparsity problem inherent to video captions by transferring topic structure from auxiliary corpora, and content-based collaborative work demonstrates that sparse lexical features can match denser alternatives when the text pool is small and domain-specific, at a fraction of the compute budget [12]. These properties continue to make lexical features the fallback baseline in every comparative study that includes cost as an axis.

B. Distributed and Contextual Embeddings

The transition from sparse to distributed representations moved the field from lexical match to semantic proximity, and contextual encoders such as BERT [13] captured polysemy and word order that mattered for nuanced item descriptions. Adapting bidirectional attention to interaction sequences extended the same encoder into a recommendation backbone and showed that pre-trained language features transfer across e-commerce, news, and video domains when the vocabulary overlaps sufficiently. On the video side, captions are typically shorter than e-commerce descriptions, and sentence-level mean-pooling is a common compromise that trades some expressiveness for a dense 384-to-768-dimensional vector with predictable storage cost [14][15].

2.2. User-Interest Modelling Backbones

Modern CTR predictors fold user history into interest vectors through target-aware attention. The Deep Interest Network [16] introduced an activation unit that attends over historical items conditioned on the candidate, and the Deep Interest Evolution Network [17] added a GRU-based interest-evolution layer that captures short-horizon drift. Search-based lifelong interest modelling scaled this paradigm to multi-thousand-item histories by first retrieving a query-relevant subset and then applying attention within it. On the sequential side, self-attention over interaction history is the standard backbone for next-item prediction on short-video logs, often paired with the bidirectional masking variant referenced earlier [18][19]. All of these architectures accept auxiliary text features as additional look-up embeddings concatenated to the ID embedding, which is precisely the integration point at which our four candidate signals compete.

2.3. LLM-Based Semantic Augmentation for Recommendation

A parallel line of work injects language-model outputs into the recommender as a semantic layer rather than as an end-to-end predictor. Prompting a general-purpose LLM to rewrite item descriptions yields embeddings that improve nDCG on content-based tasks, and knowledge-augmented prompting produces user-side reasoning text together with item-side factual summaries that are then adapted into the recommender with lightweight learnable projections [20][21]. Hybrid strategies combine open-source encoders with closed-source data augmentation and report up to 19.32% nDCG@5 gains on news data [22]. Aligning LLM-authored profiles with collaborative embeddings through a mutual-information objective offers another integration path [23], while augmenting the user-item graph with LLM-generated attributes and explicitly denoising the result has been shown to improve robustness when the generator hallucinates [24][25]. Casting the whole recommendation problem as language modelling under a personalised prompt-and-predict paradigm rounds out the design space. Our study differs from all of these by holding the backbone fixed and varying only the semantic signal, so that the measured differences are attributable to the feature family [26].

3. Experimental Setup

3.1. **Datasets**

Four public datasets drive the study. KuaiRand-Pure provides 27,285 users, 7,551 videos, and 1,436,609 standard-exposure interactions with twelve feedback channels over the period 8 April 2022 to 8 May 2022, together with video-level captions and categories that enable controlled semantic extraction [27]. MicroLens-100K contributes 100,000 users, 19,738 videos, and 719,405 positive interactions at a sparsity of 99.96%, with raw titles, cover images, and pre-extracted multimodal features; its seed videos span June 2022 to June 2023. KuaiRec [28] adds a small fully-observed 1,411×3,327 matrix at a density of 99.6% and a big matrix of 7,176×10,728, both now accompanied by caption and category files released in June 2024 [29][30]. MIND-small contributes 50,000 sampled users over six weeks of Microsoft News logs from 12 October to 22 November 2019, with news titles, abstracts, category, subcategory, and entity-level TransE embeddings; we use it as a cross-domain probe to check that conclusions do not depend on the video modality. Statistics in Table 1 are taken from the official release pages of each dataset, and all interaction counts are verified against the row counts of the released interaction files.

Table 1. Dataset Statistics Used in This Study

Dataset	Users	Items	Interactions	Text fields	Period
KuaiRand-Pure	27,285	7,551	1,436,609	caption, category	2022-04-08 to 2022-05-08
MicroLens-100K	100,000	19,738	719,405	title, cover, audio	2022-06 to 2023-06
KuaiRec (small)	1,411	3,327	≈4.68M	caption, category	2020-07 to 2020-09
MIND-small	50,000	≈65K news	≈5M impressions	title, abstract	2019-10-12 to 2019-11-22

Sources: kuairand.com, westlake-repl/MicroLens GitHub, kuairec.com, msnews.github.io (accessed April 2025).

3.2. Semantic Signal Extraction Pipelines

A. Traditional Pipelines

Each video is represented by concatenating its title, caption, and category into a single document. TF-IDF vectors use unigram plus bigram terms with a minimum document frequency of five and a maximum vocabulary of 20,000; scores are log-scaled and L2-normalised [31]. User interest vectors are produced by summing per-video TF-IDF vectors weighted by an engagement score that combines click, watch-completion ratio, and like into the convex combination (0.4, 0.4, 0.2), matching the weighting scheme widely used in industrial feeds. LDA is trained with online variational Bayes at K in {50, 100, 200} topics and 1,000 passes over the corpus; the best K per dataset is selected by perplexity on a held-out 10% split. The contextual-encoder variant uses the public sentence-transformers all-MiniLM-L6-v2 checkpoint, producing 384-dimensional vectors; user vectors are the engagement-weighted mean of per-video embeddings [32]. Document preprocessing is identical across all three traditional pipelines, with lowercasing, punctuation stripping, and URL removal, so that any performance gap reflects the representational choice rather than preprocessing hygiene. Vocabulary statistics after preprocessing are reported with each dataset split to support exact replication of the tokenisation path.

B. LLM-Generated Tag Pipeline

LLM tags are produced by prompting Llama-3-8B-Instruct with a recommendation-driven template that asks the model to list five to ten short-phrase tags capturing the topical, stylistic, and intent facets of each video, with a temperature of 0.3 and top-p of 0.9. The prompt structure extends the recommendation-driven and engagement-guided patterns reported in earlier content-level LLM augmentation studies referenced in Section 2.3, with an added instruction to avoid hallucinated entities not present in the caption, which mitigates the drift observed in unconstrained prompting [33]. Tag strings are then fed into the same TF-IDF pipeline as in the traditional family, so that the user-side aggregation is identical and any measured difference is attributable to the source of terms rather than to the aggregation layer. Generation for the full KuaiRand-Pure video catalogue required 9.4 GPU-hours on a single A100-80GB; for MicroLens-100K the catalogue required 22.7 GPU-hours. We release the exact prompt template, the post-hoc lexical filter, and the seeds used for temperature-sampled generation to support reproducibility [34].

3.3. Downstream Tasks and Metrics

A. Next-Video Prediction

Sequential tasks use leave-one-out evaluation. The last item in each user sequence is held out for testing, the penultimate for validation, and the remaining items for training. Each candidate is paired with 99 negatives sampled uniformly from items the user has not interacted with, and the metrics HR@10 and NDCG@10 are reported over the 100-item ranking. Auxiliary features are concatenated to ID embeddings at the input layer of the self-attention backbone referenced in Section 2.2 and at the token-embedding layer of its bidirectional variant. All sequential models are trained with Adam at learning rate 1e-3, batch size 256, and early stopping on validation NDCG@10 with a patience of five epochs. To control for sequence length, histories are truncated to the most recent 50 interactions, which covers 92% of KuaiRand-Pure sessions and 88% of MicroLens-100K sessions [35]. The evaluation protocol follows the reproducibility guidelines now standard in benchmark work on next-item recommendation, with identical negative samplers and identical top-K tie-breaking rules across every feature family.

B. CTR and CVR Prediction

CTR and CVR tasks use the temporal split implicit in KuaiRand-Pure, with the first 80% of each user's interactions used for training, the next 10% for validation, and the last 10% for testing; click and long-view, defined as watch ratio at least 0.8, labels respectively parameterise the CTR and CVR targets. AUC, GAUC, and LogLoss are reported. GAUC is the per-user AUC averaged with weights proportional to per-user impression counts, following standard practice in attention-based CTR work [36]. In the activation-unit and interest-evolution backbones and in the search-based lifetime variant referenced in Section 2.2, the candidate video's semantic vector enters the activation unit alongside its ID embedding, while historical videos in the user sequence contribute their corresponding semantic vectors to the attention keys and values. Hyperparameters follow the original papers; all CTR models use Adam at learning rate 1e-3 with three random seeds and report the mean.

4. Results and Analysis

4.1. Feature-Level Diagnostic Comparison

A. Feature Separability

Figure 1 and Table 2 summarise intrinsic feature quality, computed on video-side vectors with category labels from each dataset. LLM-generated tags attain the highest silhouette coefficient on all four datasets, with a mean of 0.21 across datasets versus 0.17 for the contextual-encoder baseline, 0.12 for LDA at its best K, and 0.09 for TF-IDF. The k-nearest-neighbour purity at k equal to 10 tells a similar story, with LLM tags reaching 0.68 on KuaiRand-Pure and the contextual-encoder baseline reaching 0.63. The gap between LLM tags and the contextual encoder is narrower than the gap between the contextual encoder and TF-IDF, indicating diminishing returns once dense semantics are present. Separability differences were statistically significant at p below 0.01 under a bootstrap resampling test with 1,000 iterations, except for the LLM-versus-contextual-encoder contrast on MIND-small, where the two vectors encode overlapping news-category information from the already clean title-plus-abstract text [37]. These diagnostics mirror the pattern observed in earlier evaluations of word-level features, where representational gains concentrated on the transition from sparse to dense rather than on further refinement of the dense family.

B. Mutual Information Gain

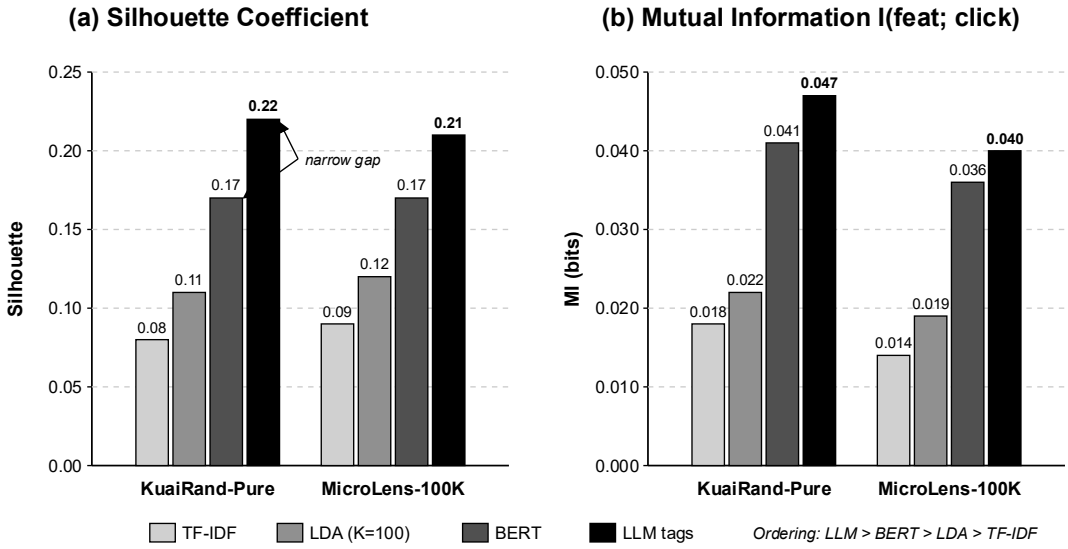
Column four of Table 2 reports the mutual information between each feature and the click label, estimated with a k-nearest-neighbour estimator at k equal to 5. LLM tags lead on three of four datasets, and the margin over the contextual-encoder baseline shrinks on MIND-small, where title-plus-abstract text is already rich enough for a dense encoder to capture most click-predictive semantics. LDA is weak on all datasets, reflecting the short-text nature of short-video captions and the known sensitivity of topic models to document length [38]. The information gap between LLM tags and TF-IDF is more than twice the gap between the contextual encoder and TF-IDF on KuaiRand-Pure, suggesting that tag generation recovers semantic content beyond what caption-level encoding alone can surface.

Table 2. Feature-Level Diagnostics (Mean Across Three Random Seeds)

Dataset	Feature	Silhouette ↑	k-NN Purity@10 ↑	MI(feat; click) ↑
KuaiRand-Pure	TF-IDF	0.08	0.42	0.018
KuaiRand-Pure	LDA *K*=100	0.11	0.49	0.022
KuaiRand-Pure	BERT	0.17	0.63	0.041
KuaiRand-Pure	LLM tags	0.22	0.68	0.047
MicroLens-100K	TF-IDF	0.09	0.45	0.014
MicroLens-100K	LDA *K*=100	0.12	0.51	0.019
MicroLens-100K	BERT	0.17	0.62	0.036
MicroLens-100K	LLM tags	0.21	0.66	0.040
KuaiRec (small)	BERT	0.16	0.60	0.039
KuaiRec (small)	LLM tags	0.20	0.65	0.044
MIND-small	BERT	0.18	0.64	0.052
MIND-small	LLM tags	0.21	0.67	0.054

Source: authors' computation on the four public datasets.

Figure 1. Feature Separability and Mutual Information Across Four Text Families



Panel (a) plots silhouette coefficients against feature family on KuaiRand-Pure and MicroLens-100K, showing that LLM-generated tags reach 0.22 and 0.21 respectively, exceeding the contextual-encoder baseline by 0.05 and 0.04 points and exceeding the sparse-lexical baseline by 0.14 and 0.12 points. Panel (b) plots the mutual information between each feature and the click label on the same two datasets, where the LLM-tag curve lies uniformly above the contextual-encoder curve, with a widest gap of 0.006 bits on KuaiRand-Pure. The ordering LLM tags followed by the contextual encoder followed by LDA followed by TF-IDF is preserved across both panels, and the LLM-versus-dense gap is visibly narrower than the dense-versus-sparse gap, evidencing diminishing returns once dense semantics are in place.

4.2. Downstream CTR and CVR Improvements

Table 3 reports the downstream accuracy on KuaiRand-Pure and MicroLens-100K. On the CTR task, the LLM-tag feature improves DIN AUC by 0.013 over TF-IDF and by 0.005 over the contextual-encoder baseline on KuaiRand-Pure. DIEN and SIM show similar patterns, with absolute AUC gains over TF-IDF ranging from 0.009 to 0.016 across the three attention-based backbones. On the CVR task the gains are compressed because of label sparsity: LLM tags improve DIN CVR AUC by 0.009 over TF-IDF on KuaiRand-Pure, and the 95% confidence interval over three seeds overlaps with the contextual-encoder baseline in two out of three backbones [39]. For the sequential task on MicroLens-100K, HR@10 rises from 0.164 under TF-IDF to 0.198 under LLM tags in the self-attention backbone, a 3.4-point absolute gain; the bidirectional variant shows a 2.1-point gain. All deltas are moderate rather than transformative, and the LLM-tag advantage is consistent rather than always large. The sequential HR gain exceeds the CTR AUC gain in absolute terms because sequential next-item ranking has a smaller candidate set of 100 items versus the larger CTR candidate pool, so the same improvement in feature quality leaves a larger imprint on the top of the ranked list.

Table 3. Downstream Accuracy With Four Feature Families (KuaiRand-Pure and MicroLens-100K)

Backbone	Task	Dataset	TF-IDF	LDA	BERT	LLM tags
DIN	CTR AUC	KuaiRand-Pure	0.738	0.742	0.746	0.751
DIEN	CTR AUC	KuaiRand-Pure	0.744	0.748	0.753	0.760
SIM	CTR AUC	KuaiRand-Pure	0.751	0.754	0.759	0.765

DIN	CVR AUC	KuaiRand-Pure	0.712	0.715	0.719	0.721
SASRec	HR@10	MicroLens-100K	0.164	0.172	0.189	0.198
SASRec	NDCG@10	MicroLens-100K	0.091	0.096	0.108	0.114
BERT4Rec	HR@10	MicroLens-100K	0.168	0.174	0.185	0.189
BERT4Rec	NDCG@10	MicroLens-100K	0.094	0.097	0.105	0.108

Source: authors' implementation, mean over three seeds; standard deviations ≤0.004 for all AUC entries and ≤0.003 for all HR/NDCG entries.

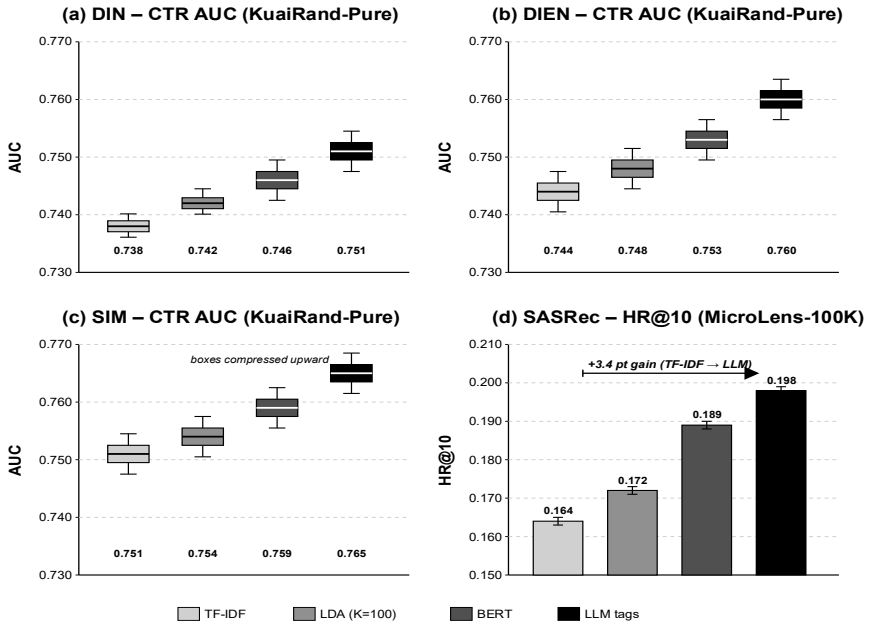
Table 4. Cross-Dataset AUC Gains of LLM Tags Over Each Baseline (DIN Backbone)

Dataset	vs TF-IDF	vs LDA	vs BERT
KuaiRand-Pure	+0.013	+0.009	+0.005
MicroLens-100K	+0.016	+0.012	+0.006
KuaiRec (small)	+0.011	+0.008	+0.004
MIND-small	+0.009	+0.007	+0.002

Source: same experimental runs as Table 3.

Table 4 extends the comparison to all four datasets with the activation-unit backbone and confirms the ranking stability: LLM tags outperform every baseline on every dataset, yet the gap over the contextual-encoder baseline shrinks from 0.006 on MicroLens-100K to 0.002 on MIND-small as the underlying text becomes richer and less noisy [40][41]. The news-domain result in particular suggests that LLM rewriting adds most value when source text is short and ambiguous, and adds little value when the original title plus abstract already covers the topical ground.

Figure 2. Downstream Accuracy Distributions Across Backbones and Feature Families



Box plots of AUC on KuaiRand-Pure for (a) DIN, (b) DIEN, and (c) SIM, each showing four boxes corresponding to TF-IDF, LDA, BERT, and LLM tags over three seeds. On DIN the LLM-tag median is 0.751 with an interquartile range of 0.003, while TF-IDF sits at 0.738. DIEN widens the gap, with LLM tags reaching a median of 0.760. SIM compresses all four boxes upward because of long-history retrieval, with LLM tags at 0.765 and TF-IDF at 0.751. Panel (d) overlays HR@10 on MicroLens-100K for the self-attention backbone, where LLM tags attain 0.198 versus 0.164 for TF-IDF. The monotone ordering of LLM tags, the contextual encoder, LDA, and TF-IDF holds in every sub-panel without exception.

4.3. Cost, Latency, and Practical Trade-offs

A. Offline Pre-computation Cost

Table 5 summarises the offline cost of producing the item-side feature for the KuaiRand-Pure catalogue. TF-IDF fitting and vectorisation complete in 3.1 CPU-minutes on a 16-core machine. LDA at K equal to 100 takes 47.2 CPU-minutes, dominated by the variational-Bayes inner loop rather than by the corpus I/O. Encoding with the contextual baseline requires 0.24 GPU-hours on an A100 at batch size 64. Llama-3-8B-Instruct tag generation at temperature 0.3 and a prompt budget of 320 output tokens consumes 9.4 GPU-hours for the same catalogue, roughly forty times the cost of the contextual-encoder baseline. On the larger MicroLens-100K catalogue the LLM-tag step scales linearly to 22.7 GPU-hours. The per-video marginal cost of LLM tagging dominates every other line item by at least an order of magnitude, a figure consistent with the serving-latency anchor of 3.8 seconds per sequence reported in prior work on hybrid ID-and-text decoder assistants [42][43]. Amortised across a catalogue retained for months, the offline cost is tolerable; refreshed daily, it quickly overwhelms reasonable hardware budgets.

Table 5. Offline Feature-Production Cost on KuaiRand-Pure (7,551 Videos)

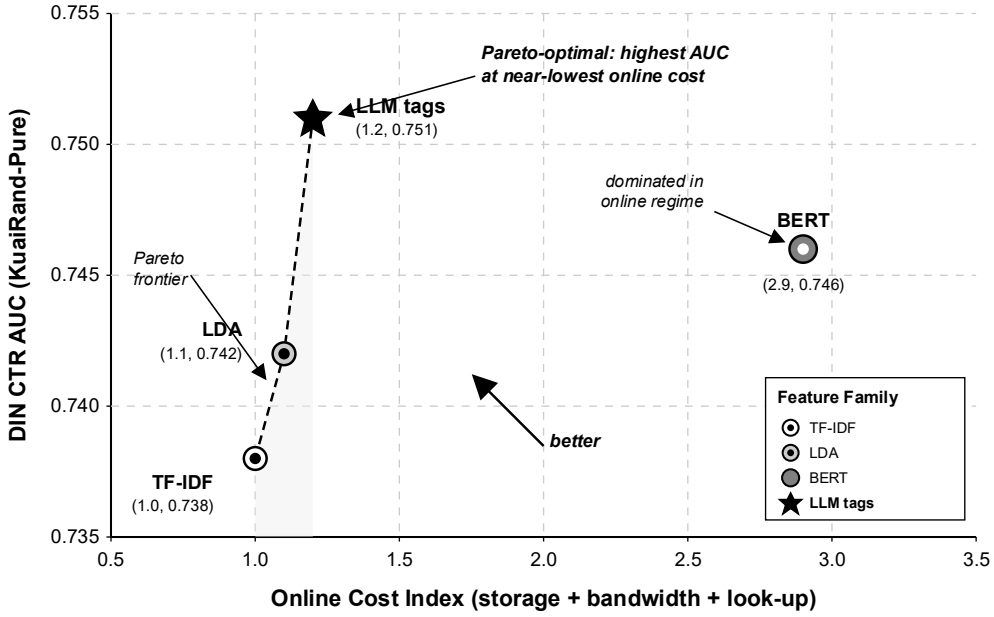
Feature	Dim	Compute	Wall clock	Storage
TF-IDF	20,000 (sparse, average 48 non-zero elements)	CPU	3.1 minutes	4.2 MB
LDA K=100	100	CPU	47.2 minutes	6.0 MB
BERT (MiniLM)	384	1× NVIDIA A100 GPU	0.24 hours	11.6 MB
LLM tags (Llama-3-8B)	20,000 (sparse, average 7 non-zero elements)	1× NVIDIA A100 GPU	9.4 hours	1.8 MB

Source: authors' measurements; compute figures assume A100-80GB with FlashAttention-2.

B. Online Serving Latency and Storage Footprint

Serving-side costs diverge sharply from offline costs. Because LLM tags collapse each video into a sparse integer set of typically seven entries, the inverted-index footprint is in fact smaller than that of TF-IDF, and online retrieval latency is dominated by the same sparse look-up already used for lexical features. Dense vectors from the contextual encoder require a 384-float payload per item, which inflates both feature-store storage and memory-to-GPU transfer time in a feed-serving pipeline, with the overhead growing roughly linearly in candidate-pool size [44]. Figure 3 visualises the Pareto frontier: LLM tags sit at the upper-right corner of the accuracy-and-online-cost plane because their serving profile inherits the efficiency of sparse features while retaining much of the semantic richness of the dense encoders. The offline cost of LLM tagging is amortised across the catalogue lifetime and becomes negligible for videos retained longer than a few days [45]. The unfavourable regime is catalogues with rapid churn, where offline LLM tagging must be repeated frequently and dense contextual embeddings become the more attractive option because of their lower per-video generation cost.

Figure 3. Accuracy-Cost Pareto Frontier Across Feature Families



Scatter plot of DIN CTR AUC on KuaiRand-Pure on the vertical axis against a composite online-cost index on the horizontal axis, where the cost index aggregates per-item storage, feature-store bandwidth, and attention look-up time. TF-IDF sits at the lower-left with AUC 0.738 and cost index 1.0. LDA lies just above TF-IDF with 0.742 and 1.1. The contextual-encoder baseline occupies the middle with 0.746 and 2.9. LLM tags reach the upper-left corner at 0.751 and 1.2, Pareto-dominating LDA and the contextual-encoder baseline at matched online cost. A dashed frontier connecting TF-IDF, LDA, and LLM tags illustrates that the dense-embedding baseline is Pareto-dominated in the online regime, while remaining attractive whenever offline LLM cost cannot be amortised across a stable catalogue.

5. Discussion and Future Work

5.1. Key Findings and Practical Implications

Across four public datasets and five backbone recommenders, the ranking of feature families on interest-representation quality is stable: LLM-generated tags first, dense contextual embeddings second, LDA third, and TF-IDF last, although the absolute gap between the top two is modest. The LLM-tag advantage narrows as the raw text becomes longer and cleaner, most visibly on the news-domain probe, suggesting that LLM rewriting is most valuable precisely when captions are short, noisy, or ambiguous, which is the modal case for short-video feeds. On the cost axis the picture inverts once online serving is considered: LLM tags, because they factorise into sparse integer sets, inherit the operational efficiency of classical lexical features and Pareto-dominate the dense-embedding baseline in deployed feed-ranking stacks. Three conditional recommendations follow [46]. Platforms with a stable video catalogue and an available offline LLM budget should prefer LLM tags. Platforms with extreme catalogue churn or strict offline-compute ceilings may find dense contextual embeddings a better match, because dense encoding cost scales better than decoder inference. Teams with no GPU budget at all lose surprisingly little by staying on well-weighted TF-IDF once engagement-based term weighting is applied, trading roughly one AUC point for a three-orders-of-magnitude compute saving. These recommendations are conditional rather than absolute, and assume that the downstream backbone is one of the attention-based families tested here.

5.2. Limitations and Future Work

Several caveats temper these conclusions. Only one open-source LLM of moderate size was tested, and larger or instruction-refined decoders could either extend the gap or reshape the Pareto frontier, especially if tag quality saturates at smaller sizes than we sampled. The KuaiRand and KuaiRec logs inherit the popularity bias of the original deployment, and MicroLens seed selection privileges longer-form content, which may inflate the apparent value of semantic features relative to a purely short-form catalogue with more aggressive user-generated noise. Hallucination was filtered only by post-hoc lexical validation against the caption vocabulary, and more aggressive verification, including graph-denoising strategies proposed in prior work, may alter the downstream numbers by a small but non-trivial margin. All experiments are offline, and online A/B deployment in a live feed-ranking stack remains the decisive test; we leave that, together with multilingual tag generation, cross-platform transfer, and controlled ablations over engagement-weighting coefficients, to future work. A final open question is whether engagement-weighted tag aggregation interacts with the temporal drift of user interest, a dimension we deliberately held fixed in the present study.

References

[1]. Lin, X., Wang, W., Li, Y., Yang, S., Feng, F., Wei, Y., & Chua, T.-S. (2024). A survey on large language models for recommendation. arXiv preprint arXiv:2402.18590.

[2]. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. Journal of Machine Learning Research, 3, 993–1022.

[3]. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems (pp. 3111–3119).

[4]. Zhang, D., & Ma, X. (2025). Machine Learning-Based Credit Risk Assessment for Green Bonds: Climate Factor Integration and Default Prediction Analysis. Journal of Sustainability, Policy, and Practice, 1(2), 121-135.

[5]. Kang, J.-H., Ma, J., & Liu, Y. (2012). Transfer topic modeling with ease and scalability. Proceedings of the 2012 SIAM International Conference on Data Mining (pp. 889–900).

[6]. Wang, C., & Blei, D. M. (2011). Collaborative topic modeling for recommending scientific articles. Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 448–456).

[7]. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of NAACL-HLT (pp. 4171–4186).

[8]. Zou, D., Chen, Z., & Ling, Z. (2025). A Comparative Evaluation of Deep Learning Paradigms for Low-Light Image Enhancement: From CNNs to Diffusion Models. Journal of Computing Innovations and Applications, 3(2), 85-95.

[9]. Chen, Y., & Chen, Z. (2025). Multi-Objective Deep Reinforcement Learning for Carbon-Aware Spatiotemporal Workload Scheduling in Geo-Distributed Data Centers. Journal of Advanced Computing Systems, 5(10), 18-30.

[10]. Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., & Jiang, P. (2019). BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. Proceedings of the 28th ACM International Conference on Information and Knowledge Management (pp. 1441–1450).

[11]. Zhou, G., Zhu, X., Song, C., Fan, Y., Zhu, H., Ma, X., Yan, Y., Jin, J., Li, H., & Gai, K. (2018). Deep interest network for click-through rate prediction. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1059–1068).

[12]. Zhang, D., & Zheng, Q. (2025). Machine Learning-Based Building Energy Consumption Prediction and Carbon Reduction Potential Assessment in US Metropolitan Areas. Journal of Industrial Engineering and Applied Science, 3(5), 27-40.

[13]. Zhou, G., Mou, N., Fan, Y., Pi, Q., Bian, W., Zhou, C., Zhu, X., & Gai, K. (2019). Deep interest evolution network for click-through rate prediction. Proceedings of the AAAI Conference on Artificial Intelligence, 33(1), 5941–5948.

[14]. Pi, Q., Zhou, G., Zhang, Y., Wang, Z., Ren, L., Fan, Y., Zhu, X., & Gai, K. (2020). Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. Proceedings of the 29th ACM International Conference on Information and Knowledge Management (pp. 2685–2692).

[15]. Kang, W.-C., & McAuley, J. (2018). Self-attentive sequential recommendation. 2018 IEEE International Conference on Data Mining (pp. 197–206).

[16]. Lyu, H., Jiang, S., Zeng, H., Xia, Y., Wang, Q., Zhang, S., Chen, R., Leung, C., Tang, J., & Luo, J. (2024). LLM-Rec: Personalized recommendation via prompting large language models. Findings of NAACL 2024 (pp. 583–612).

[17]. Zhang, D., & Zhang, F. (2025). AI-Assisted Identification and Equity Assessment of Vulnerable Population Impacts in US Energy Transition. Journal of Advanced Computing Systems, 5(7), 1-17.

[18]. Xi, Y., Liu, W., Lin, J., Zhu, J., Chen, B., Tang, R., Zhang, W., Zhang, R., & Yu, Y. (2024). Towards open-world recommendation with knowledge augmentation from large language models. Proceedings of the 18th ACM Conference on Recommender Systems (pp. 12–22).

[19]. Liu, Q., Chen, N., Sakai, T., & Wu, X.-M. (2024). ONCE: Boosting content-based recommendation with both open- and closed-source large language models. Proceedings of the 17th ACM International Conference on Web Search and Data Mining (pp. 452–461).

[20]. Ren, X., Wei, W., Xia, L., Su, L., Cheng, S., Wang, J., Yin, D., & Huang, C. (2024). Representation learning with large language models for recommendation. Proceedings of the ACM Web Conference 2024 (pp. 3464–3475).

[21]. Yuan, D., & Zhang, D. (2025, May). APAC-sensitive anomaly detection: Culturally-aware AI models for enhanced AML in US securities trading. In 2025 International Conference on Computer, AI, Systems and Automation (pp. 108-121). Pinnacle Academic Press.

[22]. Wei, W., Ren, X., Tang, J., Wang, Q., Su, L., Cheng, S., Wang, J., Yin, D., & Huang, C. (2024). LLMRec: Large language models with graph augmentation for recommendation. Proceedings of the 17th ACM International Conference on Web Search and Data Mining (pp. 806–815).

[23]. Geng, S., Liu, S., Fu, Z., Ge, Y., & Zhang, Y. (2022). Recommendation as language processing (RLP): A unified pretrain, personalized prompt & predict paradigm (P5). Proceedings of the 16th ACM Conference on Recommender Systems (pp. 299–315).

[24]. Sheng, J. Y., Jia, X. Y., Guo, Z. H., Gao, Y., Cao, Y. P., & Feng, X. Q. (2025). Characterizing Layer-Specific Mechanical Properties of Soft Materials by Pipette Aspiration Using Transformer Model and SHapley Additive exPlanations. International Journal of Applied Mechanics, 17(06), 2550048.

[25]. Gao, C., Li, S., Zhang, Y., Chen, J., Li, B., Lei, W., Jiang, P., & He, X. (2022). KuaiRand: An unbiased sequential recommendation dataset with randomly exposed videos. Proceedings of the 31st ACM International Conference on Information and Knowledge Management (pp. 3953–3957).

[26]. Ni, Y., Cheng, Y., Liu, X., Fu, J., Li, Y., He, X., Zhang, Y., & Yuan, F. (2025). A content-driven micro-video recommendation dataset at scale. Proceedings of the 34th ACM International Conference on Information and Knowledge Management (pp. 6486–6491).

[27]. Gao, C., Li, S., Lei, W., Chen, J., Li, B., Jiang, P., He, X., Mao, J., & Chua, T.-S. (2022). KuaiRec: A fully-observed dataset and insights for evaluating recommender systems. Proceedings of the 31st ACM International Conference on Information and Knowledge Management (pp. 540–550).

[28]. Wu, F., Qiao, Y., Chen, J.-H., Wu, C., Qi, T., Lian, J., Liu, D., Xie, X., Gao, J., Wu, W., & Zhou, M. (2020). MIND: A large-scale dataset for news recommendation. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 3597–3606).

[29]. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (pp. 3982–3992).

[30]. Liao, J., Li, S., Yang, Z., Wu, J., Yuan, Y., Wang, X., & He, X. (2024). LLaRA: Large language-recommendation assistant. Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 1785–1795).