FLUID: From Ephemeral IDs to Multimodal Semantic Codes for Industrial-Scale Livestreaming Recommendation

Xinhang Yuan*
rose.yuan@tiktok.com
TikTok
San Jose, California, USA

Zexi Huang*†
zexi.huang@tiktok.com
TikTok
San Jose, California, USA

Anjia Cao*
caoanjia@bytedance.com
ByteDance
Shanghai, China

Xudong Lu*
luka.30@bytedance.com
ByteDance
Shanghai, China

Zikai Wang
wangzikai.kevin@bytedance.com
ByteDance
San Jose, California, USA

Penghao Zhou
zhoupenghao.leon@bytedance.com
ByteDance
Shanghai, China

Chang Liu
chang.liu.8@bytedance.com
ByteDance
Singapore

Wentao Guo
wentao.guo@bytedance.com
ByteDance
San Jose, California, USA

Qinglei Wang
wangqinglei@bytedance.com
ByteDance
Singapore

arXiv:2605.21832v2 [cs.AI] 26 May 2026

Abstract

Modern recommender systems rely heavily on ID-based collaborative filtering: each item is represented by a unique ID embedding that accumulates collaborative signals from user interactions. Livestreaming recommendation, however, faces a unique challenge in this paradigm: a live room typically broadcasts for only tens of minutes, so its item ID remains poorly learned in a persistent cold-start state and ID-centric ranking models fail to generalize. We present FLUID, the first framework to fully retire the candidate-side item ID from a production-scale livestreaming ranker. FLUID introduces a cross-domain multimodal encoder, jointly trained on short videos and livestreams, to produce discrete hierarchical semantic codes, called LUCID, for content-based item characterization. To adapt the ranker to LUCID, FLUID further employs a staged warmup scheme: it first incorporates cold, slice-level LUCID as an independent token alongside the ID embedding, and then replaces the ID embedding with warm, room-level LUCID before online incremental training. Deployed on our industrial livestreaming recommenders with a cross-platform combined user base of over one billion globally, FLUID delivers significant online gains of +0.55% Quality Watch Duration, +2.05% Cold-Start Room Views, and +0.05% Active Hours.

*Core contributors.
†Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference acronym 'XX, Woodstock, NY

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

CCS Concepts

• **Information systems** → **Recommender systems**; **Learning to rank**; Multimedia streaming; • **Computing methodologies** → *Neural networks.*

Keywords

Livestreaming Recommendation, Multimodal Representation, Large Recommendation Models

ACM Reference Format:
Xinhang Yuan, Zexi Huang, Anjia Cao, Xudong Lu, Zikai Wang, Penghao Zhou, Chang Liu, Wentao Guo, and Qinglei Wang. 2018. FLUID: From Ephemeral IDs to Multimodal Semantic Codes for Industrial-Scale Livestreaming Recommendation. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX).* ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Livestreaming has emerged as one of the fastest-growing online content ecosystems [18, 35, 44], where creators broadcast in real time and users interact through viewing, commenting, and gifting. Unlike videos (typical item lifetime measured in days to months) and e-commerce (months to years), livestreams are only relevant when on air. On our leading livestreaming platforms with a combined user base of over one billion, a live room typically broadcasts for tens of minutes (median ∼40 min). This poses a fundamental challenge for ID-centric recommender systems, which rely on accumulating collaborative signals on item ID embeddings to power personalization. As shown in Figure 1, the embedding norm fails to converge within the median 40-minute room lifetime. Most items therefore spend their entire lifetime with undertrained embeddings. Meanwhile, modern large recommendation models (LRMs) [9, 39, 41, 44] still place most of their capacity in the ID embedding tables, which only memorize short-lived exposure signals and struggle to generalize in the livestreaming setting.

A promising solution to reduce ranker reliance on memorization-based ID embeddings is to leverage content-derived signals. One

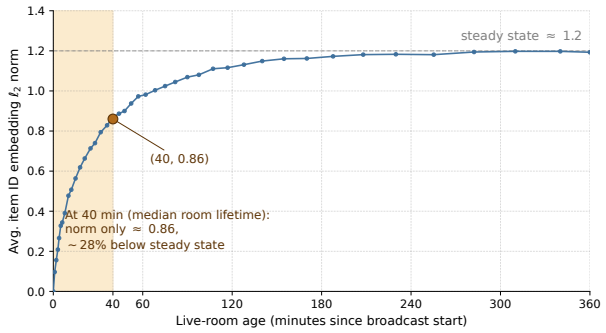

Figure 1: Item ID embedding ℓ_2 norm vs. live-room age, aggregated over one day of production traffic. The norm fails to converge within the median ~40-minute room lifetime.

line of work appends frozen multimodal embeddings to items as auxiliary features [20, 28]. Another aligns or discretizes multimodal representations into compact semantic codes with learnable embedding tables similar to ID features [7, 19, 20, 25, 36, 47, 48]. Further, end-to-end multimodal recommenders [18, 35, 43] train the multimodal encoder jointly with downstream ranking tasks. Despite their diverse forms, all these systems treat multimodal features as *complementary* signals *alongside* the dominant item IDs, including those specifically designed for livestreaming [18]. Yet, two livestreaming-specific challenges remain unsolved.

First, producing high-quality semantic representations for livestreaming is nontrivial. Live content is inherently multimodal and fast-changing, with visuals, speech, on-screen text, streamer metadata, and audience signal all evolve within minutes [18, 35]. Content representations must capture both transient dynamics and persistent room characteristics. In addition, unlike videos and e-commerce where contents are themed and user feedback signals are dense, which majority of multimodal encoders are designed for [7, 20, 25, 28], livestreaming can be more challenging to clearly characterize with less supervised signals for encoders to train on.

Second, incorporating semantic representations into ID-centric rankers is difficult. When adding multimodal features on top of ID-based signals, rankers often take the "shortcut" of ID-embeddings and underutilize the multimodal input, usually requiring explicit alignment, optimization rebalancing, or contrastive regularization to strengthen multimodal contribution [7, 20, 47]. This ID-dominant effect is especially critical to livestreaming as item IDs are ephemeral and do not carry as much collaborative information as in other content recommendation scenarios.

To address these two challenges, we propose **FLUID** (**F**ramework for **L**ive **U**niversal **ID**-free Recommendation, Figure 2), the first framework to *fully replace* item ID with content-based semantic codes in industrial-scale livestreaming rankers. Specifically, FLUID first introduces a *cross-domain* multimodal encoder jointly trained on short videos and livestreams to leverage clean and dense supervision signals from the short-video domain, producing discrete semantic codes we call **LUCID** (**L**ive **U**niversal **C**ontent **ID**entifier). Second, a *staged warmup* training scheme adapts the ranker to LUCID by first leveraging the ranker backbone for a late fusion of cold, slice-level LUCID embedding and the existing item ID embedding and then replacing the item ID embedding with warm, room-level LUCID embedding before the final online incremental training.

In summary, our contributions are as follows:

- **Problem framing.** We argue that when the item ID is ephemeral, the ID-dominance effect becomes a fundamental bottleneck rather than a tolerable nuisance, motivating full retirement of the item ID for optimized ranking performance.
- **Cross-domain semantic representations.** A multimodal encoder jointly trained on short-video and livestream supervision produces discrete hierarchical codes that converts to embeddings via the prefix-n-gram scheme.
- **Late-fusion ID-free ranker.** We deploy the first production-scale livestreaming ranker without item IDs thanks to our staged warmup scheme, which adds LUCID and replaces the item ID with a late fusion strategy.

Deployed on our industrial livestreaming recommenders with a combined user base of over one billion globally, FLUID delivers significant online gains of +0.55% Quality Watch Duration, +2.05% Cold-Start Room Views, +2.87% Niche Room Views, +1.63% Unique Watched Tags, and +0.05% Active Hours.

2 Related Work

2.1 Multimodal Embedding

Multimodal embedding learning aligns heterogeneous data—such as images, text, and videos—into a unified semantic space [12, 24], enabling diverse downstream applications including industrial recommendation [18, 20]. Early work pioneered by CLIP [24] adopts a *dual-tower* paradigm with independent unimodal encoders aligned via contrastive learning, and has been extensively refined in specific aspects such as training objectives, architecture and applications [2, 7, 11, 17, 31, 32, 37, 40, 42]. More recent works repurpose *single-tower* paradigms for deeper cross-modal fusion, leveraging the rich world knowledge and reasoning capacity of pre-trained (multimodal) large language models [3, 13, 18, 19, 21, 34, 43, 45]. Our multimodal encoder follows this *single-tower* line.

Prior multimodal encoders for recommendation are also typically trained on data from a *single domain*, such as live streaming, short videos, or e-commerce [7, 18, 20, 25, 28]. In contrast, our multimodal encoder is trained on *cross-domain* data, enabling it to capture more robust and transferable semantic signals across diverse scenarios.

2.2 ID-based Recommendation Models

The dominant paradigm in modern recommender systems represents users and items as ID embeddings that carry collaborative signals. From early collaborative filtering [27] and matrix factorization [8, 16] to Factorization Machines and their variants [14, 26], ID embeddings have served as the primary carrier of collaborative signal across heterogeneous sparse features.

Deep learning has further scaled this paradigm. Embedding-and-MLP architectures [4, 5, 10, 33] replaced hand-crafted cross features with learnable interactions; sequential models [15, 30, 49, 50] brought attention and recurrence to user-behavior sequences; and large recommendation models (LRMs) [9, 22, 39, 41, 44] pushed model capacity onto massive ID-embedding tables. The bulk of

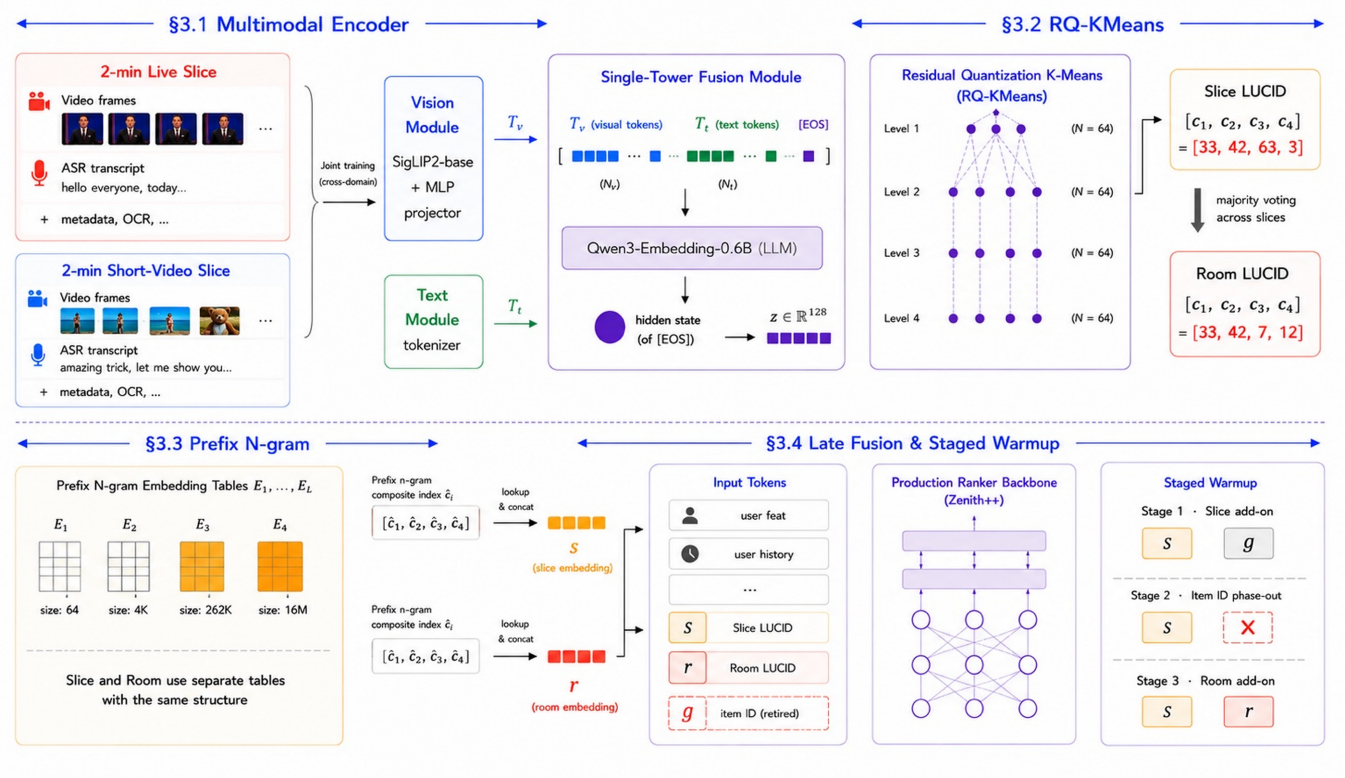

Figure 2: Overview of FLUID. ***Top:*** **a cross-domain multimodal encoder (SigLIP2 ViT + Qwen3-Embedding) jointly trained on livestreams and short videos produces a 128-d slice embedding z, which RQ-KMeans discretizes into a 4-level codeword tuple—*LUCID*—with room-level LUCID obtained by per-level majority voting over slices in a session.** ***Bottom:*** **slice-level and room-level LUCID enter the token-based ranker backbone as independent tokens (late fusion) via prefix n-gram embeddings, replacing the item id embedding with a staged warmup scheme.**

model capacity thus resides in ID-embedding tables, and performance degrades sharply when the ID signal is insufficient. This limitation is particularly acute in livestreaming, where item lifetimes are extremely short.

2.3 Multimodal Information in Ranking and Recommendation

While effective for long-lived items, the ID-centric paradigm degrades under cold-start or short-lived conditions, where insufficient interactions prevent meaningful ID embeddings from forming—an effect especially pronounced in livestreaming, where room identifiers are ephemeral by nature. To address this, recent work has explored compact content-derived semantic codes such as YouTube's Semantic IDs [29] and TIGER [25], and, more broadly, has argued that sufficiently powerful modality encoders can match or surpass pure ID-based models in cold-start regimes [38]. These developments point toward a broader trend of combining the memorization strength of IDs with the generalization capacity of content-derived representations.

Efforts to inject multimodal signals into industrial ranking systems can be broadly grouped into two families, distinguished by how tightly multimodal features are coupled with the ID embedding space: (i) *dense multimodal features*, where frozen or lightly-trained embeddings are attached to the item as auxiliary inputs; and (ii) *alignment and discretization*, where multimodal representations are explicitly aligned to, or compressed into, discrete codes more compatible with downstream ID-based interaction.

Dense multimodal features. A common industrial recipe attaches frozen multimodal embeddings to items as auxiliary dense inputs to downstream rankers [20, 28]. The approach is simple but suffers from two well-documented limitations [7, 20]: *representation mismatch* with the user–item interaction signal, and *representation unlearning*, as frozen features cannot adapt to drifting preferences or business semantics.

Alignment and discretization. To close this gap, recent work either aligns multimodal embeddings to the interaction space [7, 19, 47], or discretizes them into ID-like semantic codes via quantization [18, 20, 36, 43, 48]; we adopt the prefix n-gram parameterization of Zheng et al. [48] in our embedding lookup (Section 3.3). Despite their diversity, all these systems share a common pattern: the multimodal signal serves as a *complementary* feature alongside the dominant item ID in the ranker.

Our positioning. In contrast, FLUID *fully retires* the candidate-side item ID and lets LUCID serve as the sole content identifier on that

side. We further train a dedicated multimodal encoder with cross-domain transfer from short videos to live streams, and target the extreme cold-start regime of live streaming (median room lifetime ~40 min). Table 1 places FLUID against representative industrial and academic multimodal recommendation systems along four design axes. Two of these axes are uniquely satisfied by FLUID: it is the only method that (a) *retires* the candidate-side item ID rather than letting the multimodal signal coexist with it, and (b) trains a multimodal encoder *jointly across content domains* (short videos and live streams). Combined with *Late* fusion and coverage of *ephemeral* items, FLUID occupies a design cell that no prior industrial system has jointly occupied.

3 Method

Figure 2 gives an overview of FLUID. The FLUID pipeline replaces the candidate-side item ID with content-derived multimodal codes in four stages: the *cross-domain multimodal encoder* (§3.1) produces a content embedding for each live-stream slice; *RQ-KMeans* (§3.2) discretizes the embedding into a hierarchical code we call **LUCID**; the *prefix n-gram* scheme (§3.3) maps each LUCID tuple into a learnable embedding; and the *late-fusion ID-free ranker* (§3.4) introduces LUCID as independent candidate-side tokens and retires the item ID via a staged warmup.

3.1 Multimodal Encoder

Live rooms are too short-lived for the ranker to learn useful per-item ID embeddings. FLUID therefore introduces a multimodal encoder as its first stage, producing a content-derived signal in place of the item ID.

3.1.1 Cross-domain Training Data. Training the encoder requires pairs of queries and content slices labeled by user engagement—likes, shares, and watch-through. Many live rooms, however, end before accumulating enough engagement to form useful pairs. FLUID therefore trains a single encoder jointly on *livestreams and short videos* in a shared embedding space, where the denser engagement signal from short videos improves generalization.

3.1.2 Architecture. The multimodal encoder (Figure 2 top) consists of three components: a vision module, a text module, and a single-tower fusion module. The **Vision Module**, based on SigLIP2-base (native-resolution ViT) [32] with a two-layer MLP projector, produces visual tokens $T_v \in \mathbb{R}^{N_v \times D_h}$. The **Text Module** tokenizes rich metadata—titles, OCR, ASR transcripts, author bios, audience comments, and sticker tags—into $T_t \in \mathbb{R}^{N_t \times D_h}$. The **Single-Tower Fusion Module**, built on Qwen3-Embedding-0.6B [46], processes the concatenated sequence $T = [T_v, T_t, \texttt{[EOS]}]$. Finally, the `[EOS]` hidden state is linearly projected to a 128-d embedding z. We adopt a single-tower architecture so that vision and text interact across the LLM's full depth, rather than only meeting at the output layer as in dual-tower designs. This single-tower advantage holds consistently across retrieval and classification benchmarks (Section 4.3.1).

3.1.3 Training Recipe. The encoder is trained on a query-to-item (Q2I) contrastive task with InfoNCE loss [23] and false-negative masking. Queries are user search terms or MLLM-synthesized keywords; items are 2-minute content slices drawn from both livestreams and short videos, with positive pairs constructed from user-behavior signals (likes, shares, and watch-through). To preserve the pre-trained semantics in the LLM, we train in two stages: (1) *Alignment*—only the MLP projector and output projection are trained; (2) *Joint fine-tuning*—the full model is unfrozen and fine-tuned end-to-end at a reduced learning rate. Within each batch, we discard entire negative pairs whose query embeddings have pairwise similarity above a predefined threshold to reduce spurious negatives.

3.2 Discrete Representation via RQ-KMeans

Using the multimodal embedding z corresponding to the 2-min content slice directly as a ranker input is suboptimal: embeddings from a generic vision–language objective are *misaligned* with the user–item interaction signal, and a shared MLP over frozen multimodal embeddings lacks the expressiveness of embedding lookup tables of the ranker [20]. We therefore discretize z via Residual Quantization K-Means (RQ-KMeans) [6, 20] into an L-level codeword tuple we call **LUCID** (**L**ive **U**niversal **C**ontent **ID**entifier), which would later enter the ranker through a learnable embedding table co-trained with other sparse-ID features. We use RQ-KMeans rather than RQ-VAE because RQ-VAE collapses codebook entries under our online streaming retraining cadence, whereas K-means gives stable partitions once fit. We set L=4 and N=64. The resulting LUCID is a tuple $[c_1, \ldots, c_4]$, e.g., $[33, 42, 63, 3]$.

Slice-level LUCID encodes 2-minute content dynamics—a streamer briefly switching from chatting to singing. A live room, however, has a persistent identity: the streamer's style, audience, and topical focus remain consistent throughout the broadcast, motivating a stable room-level identifier. We obtain this identifier by *majority voting at each level*: at each quantization depth l, we take the most frequent codeword across all cumulative slices. Because residual quantization decouples coarse-to-fine semantics, per-level voting distills dominant content without mixing levels.

3.3 Prefix N-gram LUCID Embedding

Each LUCID code—whether slice-level or room-level—is a tuple $[c_1, \ldots, c_L]$ with $c_l \in \{0, 1, \ldots, N-1\}$. We now describe the way to map the tuple to a learnable embedding $\mathbf{e}_{\text{LUCID}}$ consumed by the ranker backbone.

Earlier approaches [20, 25] use *level-wise decoding*: Create L independent embedding lookup tables $\mathbf{E}_1, \ldots, \mathbf{E}_L$ with $\mathbf{E}_l \in \mathbb{R}^{N \times d}$, and let $\mathbf{e}_{\text{LUCID}}$ as the concatenation of $\mathbf{E}_l(c_l)$. However, in residual quantization, deeper levels encode refinements *relative to the prefix path*: the same c_l under different $[c_1, \ldots, c_{l-1}]$ indexes entirely different semantic regions. For example, two slices with c_1=0 and c_1=3 that happen to share c_2=2 would both look up the same entry $\mathbf{E}_2(2)$ under level-wise decoding embedding, even though residual geometry indicates that those two sub-regions are unrelated.

We therefore adopt a *prefix n-gram* embedding scheme [48], which conditions each level's embedding on the full prefix path—analogous to the classical n-gram neural language model [1], but with the context made explicit as a composite key:

$$\bar{c}_l = \sum_{k=1}^{l} c_k \cdot N^{l-k}, \quad l = 1, \ldots, L, \tag{1}$$

Table 1: Positioning of FLUID against representative industrial and academic multimodal recommendation systems.

	Sheng et al. [28]	AlignRec [19]	QARM [20]	EM3 [7]	AB-Rec [47]	DAS [36]	LARM [18]	NoteLLM-2 [43]	Zheng et al. [48]	**Ours**
Platform	Taobao	Academic	Kuaishou	Kuaishou	Academic	Kuaishou	Kuaishou	Xiaohongshu	Meta	**Industry**
Year	2024	2024	2025	2024	2025	2025	2025	2025	2025	**2026**
Scenario	Ads	Gen.	SV	EC	SV	Ads	Live	I2I	Ads	**Live**
Fusion	Early	Early	Early	Early	Early	Early	Early	Late	Late	**Late**
Retires item ID	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Ephemeral items	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓
Cross-domain encoder	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓

where $\bar{c}_l$ uniquely identifies the path from the root to level l in the quantization tree. With N=4, the two cases above yield different composite indices: [c_1=0, c_2=2] gives $\bar{c}_2 = 2$ while [c_1=3, c_2=2] gives $\bar{c}_2 = 14$—different slots in $\mathbf{E}_2$. Each level-l table is expanded from N to at most N^l rows, and

$$\mathbf{e}_{\text{LUCID}} = \mathbf{E}_1(\bar{c}_1) \,\|\, \mathbf{E}_2(\bar{c}_2) \,\|\, \cdots \,\|\, \mathbf{E}_L(\bar{c}_L). \tag{2}$$

The embedding at level l is now a refinement of its own parent path, not a feature shared across unrelated sub-trees.

3.4 Late Fusion and Staged Warmup

The final step is to incorporate LUCID embeddings to the production ranker. Our ranker backbone is a transformer-like architecture [44] based on token-level feature interaction, where different groups of feature embeddings are represented by different tokens. We now provide details on the incorporation of LUCID embeddings as separate feature tokens into our ranker.

3.4.1 Room-level and slice-level integration. Slice-level LUCID describes the current 2-minute content segment and captures short-term changes in the livestream, such as a streamer switching from chatting to singing. Room-level LUCID is obtained by the majority-voting procedure in Section 3.2 and provides a more stable descriptor of the live room. The two granularities are complementary to each other: the slice token provides transient multimodal evidence, while the room token provides persistent candidate identity.

Both embeddings use the prefix n-gram embedding scheme in Eq. 1, but they are parameterized by separate embedding tables. Sharing the same embedding tables leaves stable room semantics and fast-changing slice semantics into the same parameter space, which empirically weakens their expressiveness. We report the shared-table ablation in Section 4.3.3.

3.4.2 Fusion with the existing item ID feature. Let $\mathbf{g}$ denote the existing item ID embedding of the candidate item, and let $\mathbf{s}$ and $\mathbf{r}$ denote the prefix-n-gram embeddings of slice-level and room-level LUCID, respectively. The fusion strategies between content-based embeddings and ID-based embeddings can be broadly divided into two categories: early fusion and late fusion.

Early fusion first maps the item ID and LUCID embeddings into a single candidate representation $\mathbf{h}$. For example:

$$\mathbf{h}_{\text{replace}} = \mathbf{s}, \tag{3}$$

$$\mathbf{h}_{\text{concat}} = \mathbf{W}[\mathbf{g} \,\|\, \mathbf{s}], \quad \mathbf{W} \text{ a learnable projection}, \tag{4}$$

$$\mathbf{h}_{\text{gate}} = \alpha\mathbf{g} + (1-\alpha)\mathbf{s}, \quad \alpha = \sigma(f(\mathbf{u})). \tag{5}$$

where $\mathbf{u}$ can be the item ID embedding itself or side information such as exposure counters and item ID embedding norm. These formulations cover direct replacement, concatenation, rule-based gating, and learnable LARM-style gates [18]. And the backbone treats the final representation as a single token:

$$\mathcal{T}_{\text{early}} = \{\mathbf{h}\}. \tag{6}$$

Even though the early fusion strategy sounds intuitive in that it combines two item-side information directly into a single token, as we shall demonstrate later in our experiments, it fails to extract the useful information from LUCID due to the strong memorization effect of the ID embedding.

Late fusion instead leaves each signal as an independent token and lets the backbone learn their interactions:

$$\mathcal{T}_{\text{late}} = \{\mathbf{g}, \mathbf{s}\}. \tag{7}$$

This late fusion configuration allows us to fully leverage the strong interaction capacity of the ranker backbone to learn the incremental multimodal information from LUCID on top of the item ID.

However, keeping the candidate-side item ID after the introduction of LUCID limits the generalizability ceiling of the ranker model, especially for livestreaming ranking where item ID embeddings are often under-trained because rooms are short-lived. This limitation is empirically validated in our later analysis in Section 4.3.4. The final FLUID architecture therefore retires the candidate-side item ID completely and uses the room-level LUCID in its place, in addition to the slice-level LUCID.

$$\mathcal{T}_{\text{FLUID}} = \{\mathbf{r}, \mathbf{s}\}, \quad \mathbf{g} \notin \mathcal{T}_{\text{FLUID}}. \tag{8}$$

This design relieves the model reliance on short-lived item ID embeddings while preserving two distinct content signals to characterize the candidate item: stable room identity and transient slice dynamics.

3.4.3 Staged warmup in production. Directly switching from the item-ID-based representation to $\mathcal{T}_{\text{FLUID}}$ is unstable because the LUCID embedding tables are newly introduced and the downstream

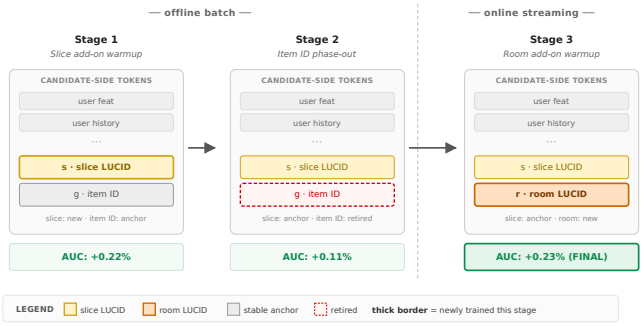

Figure 3: Three-stage warmup procedure for FLUID.

interaction layers have been trained around an item-ID-bearing candidate representation. The deeper reason this transition cannot be done in a single retrain is an *optimization asymmetry* between the item ID and LUCID. The ephemeral item ID embedding, although poorly generalizing, carries a strong item-level memorization signal that the backbone absorbs early in joint training, suppressing the slower but more generalizable LUCID signal; the resulting collapse is measured directly in Section 4.3.4. We therefore use a staged warmup procedure (Figure 3) that changes only one item-side component per stage, leaving the rest as a stable anchor.

(1) **Slice add-on.** Initialize from a converged item-ID checkpoint and add slice LUCID as an independent token.
(2) **Item ID phase-out.** Progressively zero out the candidate-side item ID, transitioning the model to rely on slice LUCID alone.
(3) **Room add-on.** Initialize room-level LUCID tables from the warm slice-level LUCID embedding tables and train them independently to specialize on persistent room identity.

After the third stage, we continue the production streaming online training schedule on the item-ID-free configuration with both slice-level and room-level LUCID.

4 Experiments

4.1 Experiment Setup

FLUID comprises two trained models that we evaluate separately: the multimodal encoder of Section 3.1, which produces LUCID codes, and the ranking model of Section 3.4, into which the LUCID codes are injected.

Ranking model. All ranking experiments are conducted on the production ranking dataset of our industrial live-streaming platforms, which collectively serve a combined user base of over one billion globally. We adopt the full backbone, dataset, and training configuration of the production Zenith++ pipeline [44] (dataset statistics in Table 2), and on top of the Zenith++ backbone we plug in slice- and room-level LUCID codes produced by RQ-KMeans with $L = 4$ residual levels and $N = 64$ clusters per level.

Multimodal encoder. The encoder is trained on a query-to-item contrastive objective jointly over short videos and live streams from our industrial platforms, following the recipe in Section 3.1. We evaluate it on an internal benchmark suite covering retrieval (live and video keyphrase retrieval, live taxonomy classification, video e-commerce retrieval, live I2I retrieval) and semantic classification, under both linear-probing and full fine-tuning settings.

Table 2: Statistics of the training dataset for the ranking model, shared with the production Zenith++ pipeline.

	# Instances	# Features	# Targets
Industrial Live Ranking	168B	4,552	98

4.2 Model Performance

Performance results. Table 3 compares four ranker configurations on the CTR target. Adding slice LUCID as an independent token alongside the item ID improves AUC by +0.22%, showing that multimodal semantics carry incremental signal. Directly removing the item ID (w/o item ID) is sharply negative, confirming the ranker's reliance on item ID memorization. Our FLUID design—retiring the item ID and replacing it with slice + room LUCID under late fusion and staged warmup (Section 3.4)—improves over the baseline on both metrics. It also surpasses the +Slice LUCID configuration by an additional +0.01% AUC and −0.11% logloss, showing that LUCID realizes its value as the item ID's successor rather than a supplement.

Table 3: Model performance across four configurations on the CTR target.

Configuration	AUC	Logloss
baseline (with item ID)	0.7784	0.1264
baseline + Slice LUCID	0.7801 (+0.22%)	0.1263 (−0.08%)
baseline w/o item ID	0.7748 (−0.47%)	0.1273 (+0.65%)
baseline w/o item ID + Slice & Room LUCID **(FLUID)**	**0.7802** (**+0.23**%)	**0.1262** (**−0.19**%)

Table 4: Online A/B test results. All values are relative changes vs. the production baseline. "n.s." denotes changes not significant at $p < 0.05$. Arms: +Slice = baseline + Slice LUCID; −ID = baseline w/o item ID; FLUID = baseline w/o item ID + Slice & Room LUCID (FLUID).

Metric	+Slice	−ID	FLUID
Engagement quality			
Quality Watch Duration	+0.43%	−0.02%	**+0.55%**
Quality Watch Session	+0.39%	−0.10%	**+0.51%**
Cold-start, niche content, and diversity			
Cold-Start Room Views	+1.15%	+1.58%	**+2.05%**
Niche Room Views	+0.69%	+2.23%	**+2.87%**
Unique Watched Tags	+0.55%	+0.20%	**+1.63%**
Retention			
Stay Duration	n.s.	−0.05%	+0.07%
Active Hours	n.s.	n.s.	+0.05%

Online A/B test. To validate these offline gains in production, we further A/B-test the three arms (+Slice, −ID, FLUID) against the

production baseline. Table 4 reports *engagement quality*, *cold-start, niche, and diversity*, and *retention* metrics. *Baseline + Slice LUCID* improves engagement and diversity but leaves retention unchanged, indicating that LUCID functions here as a secondary signal rather than a content identifier. *Baseline w/o item ID* boosts cold-start and niche exposure—releasing long-tail traffic that the item ID had previously suppressed—but at a −0.05% cost in Stay Duration, showing that broader exposure trades off against ranking efficiency rather than coexisting with it. *FLUID* is the only configuration delivering consistent gains across all three groups—engagement (Quality Watch Duration +0.55%), diversity (Cold-Start Room Views +2.05%), and retention (Stay Duration +0.07%)—confirming that room-level LUCID truly *replaces* the item ID on the candidate side, rather than merely *supplementing* it.

4.3 Ablation Studies

We conduct ablation experiments to validate the key design decisions in FLUID. Ablations on the multimodal encoder are reported on retrieval and classification benchmarks, while those on ranking components use CTR AUC unless otherwise stated.

4.3.1 Multimodal Model. Table 6 ablates the three design choices behind this encoder: backbone (Qwen3-Embedding + SigLip2 vs. CLIP/Albert), fusion paradigm (single-tower vs. dual-tower with shallow fusion), and training schedule (two-stage alignment + joint fine-tuning vs. single-stage end-to-end). The backbone change contributes the largest gain (Live Q2I R@10: 43.96 → 47.44), with single-tower fusion and two-stage training each adding further improvements on every retrieval column.

4.3.2 Cross-domain Case Review. To illustrate the performance of our cross-domain encoder, we conduct a case review with two representative LUCID clusters in Figure 4: (39, 41) for swimming, (17, 26) for dance. Livestream slices and short videos remain semantically consistent under the same LUCID code, despite their genre differences.

(a) LUCID (39, 41)**: "swimming".** **(b) LUCID** (17, 26)**: "dancing".**

Figure 4: Live slices and short videos grouped by LUCID: items sharing the same code remain semantically coherent.

4.3.3 LUCID Embedding Design. We ablate two design choices for mapping LUCID codes to embeddings on the candidate side: (i) the *embedding scheme*—Prefix n-gram (subsection 3.3) vs. Level-wise decoding—and (ii) whether slice- and room-level LUCID codes share a single embedding table or use separate ones (subsubsection 3.4.1). As summarized in Table 5, both prefix n-gram embedding (+0.11% AUC over level-wise decoding) and independent slice/room tables (+0.05% AUC over the shared table) contribute positively, with prefix n-gram providing the larger gain.

Table 5: Ablation on LUCID embedding design.

Variant	AUC
Baseline (item ID only)	0.7784
Embedding scheme	
Level-wise decoding	0.7793 (+0.12%)
Prefix n-gram (ours)	**0.7802** (**+0.23**%)
Embedding tables	
Shared table	0.7798 (+0.18%)
Independent tables (ours)	**0.7802** (**+0.23**%)

4.3.4 Candidate-side LUCID Integration: Fusion and Training Recipe. This section ablates how LUCID is integrated into our token-based ranker backbone. The ablation spans two dimensions: (i) the *fusion mechanism*—whether slice LUCID and the item ID are merged before the backbone (early fusion) or enter as independent tokens (late fusion); and (ii) the *training recipe* under late fusion—how to train slice LUCID, progressively retire the item ID, and finally add room LUCID. Table 7 organizes the resulting experiments.

Fusion mechanism. We first ablate four early-fusion methods—parameter-free combinations (Replace, Concat) and learnable gates (LARM learnable, LARM feature) [18]—and none of them yields meaningful improvement (max +0.01%, min −0.13%). Figure 5 explains why for the LARM learnable gate: the converged gate weight collapses to the item ID, leaving slice LUCID essentially unused. We therefore turn to late fusion. EM3 CIC alignment [7] adds an auxiliary loss that pulls the slice LUCID embedding toward the item ID embedding, but yields only +0.01%. Removing the alignment loss and keeping slice LUCID and the item ID as plain independent tokens lifts AUC to +0.22%. This configuration also serves as Stage 1 of our staged warmup below.

Training recipe. Naive joint training—where all embedding tables and the ranker backbone are reinitialized and trained from scratch—yields no improvement (+0.00%): the ranker collapses to item-ID-only behavior, with slice and room LUCID failing to contribute. In our staged warmup (§3.4), Stage 1 adds slice LUCID (+0.22%), demonstrating that slice LUCID provides incremental signal beyond the item ID. Stage 2 removes the item ID (+0.11%), revealing that slice LUCID alone cannot fully replace it. Stage 3 further adds room LUCID (+0.23% AUC), which fully compensates for the item-ID removal. After that, we try adding the item ID back, but it yields no additional AUC gain, confirming that LUCID has fully absorbed the candidate-side information previously provided by the item ID.

5 Conclusion

In this work, we present **FLUID**, the first framework to fully retire the candidate-side item ID from a production-scale livestreaming ranker. FLUID introduces a *cross-domain* multimodal encoder that is jointly trained on short videos and livestreams to produce discrete semantic codes called **LUCID**. To adapt the ranker to LUCID, FLUID applies a staged warmup training scheme: it first leverages the ranker backbone for a late fusion of cold, slice-level LUCID embedding and the existing item ID embedding, and then

Table 6: Multimodal encoder ablation results on backbone, fusion paradigm, and training schedule.

Method	Live Keyphrase RET		Video Keyphrase RET	
	Q2I R@10/50	I2Q R@10/50	Q2I R@10/50	I2Q R@10/50
Backbone				
Baseline (CLIP-B/32 + Albert-V2)	43.96/59.95	45.57/60.87	73.40/84.36	72.86/83.46
+ Qwen3-Embedding	46.15/60.43	47.18/61.32	84.74/91.97	85.30/91.40
+ SigLip2 ViT	**47.44/61.62**	**48.67/62.71**	**87.10/93.35**	**87.20/93.39**
Fusion paradigm				
Dual tower (shallow fusion)	45.37/59.90	47.00/61.42	83.68/90.91	84.59/91.52
Single tower (ours)	**47.44/61.62**	**48.67/62.71**	**87.10/93.35**	**87.20/93.39**
Training schedule				
Single stage	45.01/59.83	45.78/60.53	84.13/91.47	84.10/91.43
Two stage (ours)	**47.44/61.62**	**48.67/62.71**	**87.10/93.35**	**87.20/93.39**

Table 7: Ablation on candidate-side integration of LUCID with the existing item ID. Δ AUC is relative to the baseline (item ID only, AUC = 0.7784).

Category	Method	AUC	Δ AUC
Fusion mechanism	Early: Replace item ID with LUCID	0.7774	−0.13%
	Early: Concat item ID + LUCID	0.7785	+0.01%
	Early: LARM learnable gate [18]	0.7783	−0.01%
	Early: LARM feature gate [18]	0.7785	+0.01%
	Late: EM3 CIC alignment loss [7]	0.7785	+0.01%
	Late: Independent token (= Stage 1 below)	0.7801	+0.22%
Training recipe	Naive: joint training from scratch	0.7784	+0.00%
	Staged warmup (ours, §3.4):		
	Stage 1: Slice add-on (item ID + slice)	0.7801	+0.22%
	Stage 2: Item ID phase-out (slice only)	0.7793	+0.11%
	Stage 3: Room add-on (slice + room, FLUID)	**0.7802**	**+0.23%**

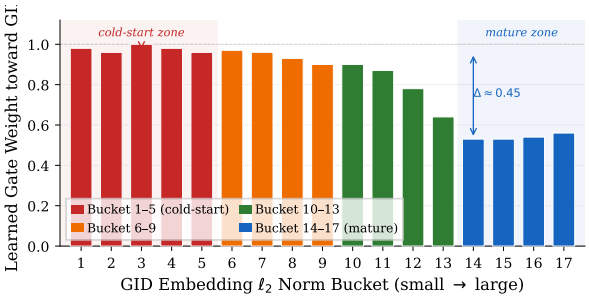

Figure 5: Gate inversion under the LARM learnable gate (Table 7). Converged gate weight on the item ID is plotted against item-ID embedding ℓ_2 norm buckets (small → large).

replaces the item ID embedding with warm, room-level LUCID embedding before the final online incremental training. Deployed on our industrial livestreaming platforms with a combined user base of over one billion globally, FLUID delivers significant online gains of +0.55% Quality Watch Duration, +2.05% Cold-Start Room Views, and +0.05% Active Hours, covering engagement quality, cold-start exposure, content diversity, and user retention. This suggests a design lesson: when items are inherently short-lived, retiring the item ID is a more effective remedy than further fusion tricks for keeping the multimodal signal alive. We hope this work serves as a step toward content-grounded ranking in short-lived recommendation domains.

References

[1] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A Neural Probabilistic Language Model. *Journal of Machine Learning Research* 3 (2003), 1137–1155.

[2] Anjia Cao, Xing Wei, and Zhiheng Ma. 2025. FLAME: Frozen Large Language Models Enable Data-Efficient Language-Image Pre-training. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 4080–4090. doi:10.1109/cvpr52734.2025.00386

[3] Haonan Chen, Hong Liu, Yuping Luo, Liang Wang, Nan Yang, Furu Wei, and Zhicheng Dou. 2025. Moca: Modality-aware continual pre-training makes better bidirectional multimodal embeddings. *arXiv preprint arXiv:2506.23115* (2025).

[4] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & Deep Learning for Recommender Systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS 2016)*. ACM, 7–10. doi:10.1145/2988450.2988454

[5] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for YouTube Recommendations. In *Proceedings of the 10th ACM Conference on*

Recommender Systems (RecSys '16). ACM, 191–198. doi:10.1145/2959100.2959190

[6] Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. 2025. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *arXiv preprint arXiv:2502.18965* (2025).

[7] Xiuqi Deng, Lu Xu, Xiyao Li, Jinkai Yu, Erpeng Xue, Zhongyuan Wang, Di Zhang, Zhaojie Liu, Yang Song, Guorui Zhou, Na Mou, and Shen Jiang. 2024. EM3: End-to-End Training of Multimodal Model and Ranking Model. *arXiv preprint arXiv:2404.06078* (2024).

[8] Simon Funk. 2006. Netflix Update: Try This at Home. Blog post. https://sifter.org/~simon/journal/20061211.html

[9] Huan Gui, Ruoxi Wang, Ke Yin, Long Jin, Maciej Kula, Taibai Xu, Lichan Hong, and Ed H. Chi. 2023. HiFormer: Heterogeneous Feature Interactions Learning with Transformers for Recommender Systems. *arXiv preprint arXiv:2311.05884* (2023).

[10] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-2017)*. International Joint Conferences on Artificial Intelligence Organization, 1725–1731. doi:10.24963/ijcai.2017/239

[11] Weiquan Huang, Aoqi Wu, Yifan Yang, Xufang Luo, Yuqing Yang, Liang Hu, Qi Dai, Chunyu Wang, Xiyang Dai, Dongdong Chen, et al. 2024. Llm2clip: Powerful language model unlocks richer visual representation. *arXiv preprint arXiv:2411.04997* (2024).

[12] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*.

[13] Ziyan Jiang, Rui Meng, Xinyi Yang, Semih Yavuz, Yingbo Zhou, and Wenhu Chen. 2024. Vlm2vec: Training vision-language models for massive multimodal embedding tasks. *arXiv preprint arXiv:2410.05160* (2024).

[14] Yuchin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. 2016. Field-aware Factorization Machines for CTR Prediction. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, 43–50. doi:10.1145/2959100.2959134

[15] Wang-Cheng Kang and Julian McAuley. 2018. Self-Attentive Sequential Recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 197–206. doi:10.1109/icdm.2018.00035

[16] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (Aug. 2009), 30–37. doi:10.1109/mc.2009.263

[17] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*. PMLR, 12888–12900.

[18] Yueyang Liu, Jiangxia Cao, Shen Wang, Shuang Wen, Xiang Chen, Xiangyu Wu, Shuang Yang, Zhaojie Liu, Kun Gai, and Guorui Zhou. 2025. LLM-Alignment Live-Streaming Recommendation. *arXiv preprint arXiv:2504.05217* (2025).

[19] Yifan Liu, Kangning Zhang, Xiangyuan Ren, Yanhua Huang, Jiarui Jin, Yingjie Qin, Ruilong Su, Ruiwen Xu, Yong Yu, and Weinan Zhang. 2024. AlignRec: Aligning and Training in Multimodal Recommendations. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24)*. ACM, 1503–1512. doi:10.1145/3627673.3679626

[20] Xinchen Luo, Jiangxia Cao, Tianyu Sun, Jinkai Yu, Rui Huang, Wei Yuan, Hezheng Lin, Yichen Zheng, Shiyao Wang, Qigen Hu, et al. 2025. Qarm: Quantitative alignment multi-modal recommendation at kuaishou. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*. 5915–5922.

[21] Rui Meng, Ziyan Jiang, Ye Liu, Mingyi Su, Xinyi Yang, Yuepeng Fu, Can Qin, Zeyuan Chen, Ran Xu, Caiming Xiong, et al. 2025. Vlm2vec-v2: Advancing multimodal embedding for videos, images, and visual documents. *arXiv preprint arXiv:2507.04590* (2025).

[22] Maxim Naumov, Dheevatsa Mudigere, Hao-Jun Michael Shi, Jianyu Huang, Narayanan Sundaraman, Jongsoo Park, Xiaodong Wang, Udit Gupta, Carole-Jean Wu, Alisson G. Azzolini, Dmytro Dzhulgakov, Andrey Mallevich, Ilia Cherniavskii, Yinghai Lu, Raghuraman Krishnamoorthi, Ansha Yu, Volodymyr Kondratenko, Stephanie Pereira, Xianjie Chen, Wenlin Chen, Vijay Rao, Bill Jia, Liang Xiong, and Misha Smelyanskiy. 2019. Deep Learning Recommendation Model for Personalization and Recommendation Systems. *arXiv preprint arXiv:1906.00091* (2019).

[23] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*.

[25] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan H. Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Saber, Maheswaran Sathiamoorthy, Ed H. Chi, and Jonathon Shlens. 2023. Recommender Systems with Generative Retrieval. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*.

[26] Steffen Rendle. 2010. Factorization Machines. In *2010 IEEE International Conference on Data Mining*. IEEE, 995–1000. doi:10.1109/icdm.2010.127

[27] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web (WWW01)*. ACM, 285–295. doi:10.1145/371920.372071

[28] Xiang-Rong Sheng, Feifan Yang, Litong Gong, Biao Wang, Zhangming Chan, Yujing Zhang, Yueyao Cheng, Yong-Nan Zhu, Tiezheng Ge, Han Zhu, Yuning Jiang, Jian Xu, and Bo Zheng. 2024. Enhancing Taobao Display Advertising with Multimodal Representations: Challenges, Approaches and Insights. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24)*. ACM, 4858–4865. doi:10.1145/3627673.3680068

[29] Anima Singh, Trung Vu, Nikhil Mehta, Raghunandan Keshavan, Maheswaran Sathiamoorthy, Yilin Zheng, Lichan Hong, Lukasz Heldt, Li Wei, Devansh Tandon, et al. 2024. Better generalization with semantic ids: A case study in ranking for recommendations. In *Proceedings of the 18th ACM Conference on Recommender Systems*. 1039–1044.

[30] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*. ACM, 1441–1450. doi:10.1145/3357384.3357895

[31] Zhulin Tao, Xiaohao Liu, Yewei Xia, Xiang Wang, Lifang Yang, Xianglin Huang, and Tat-Seng Chua. 2023. Self-Supervised Learning for Multimedia Recommendation. *IEEE Transactions on Multimedia* 25 (2023), 5107–5116. doi:10.1109/tmm.2022.3187556

[32] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. 2025. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786* (2025).

[33] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & Cross Network for Ad Click Predictions. In *Proceedings of the ADKDD'17 (KDD '17)*. ACM, 1–7. doi:10.1145/3124749.3124754

[34] Zilin Xiao, Qi Ma, Mengting Gu, Chun-cheng Jason Chen, Xintao Chen, Vicente Ordonez, and Vijai Mohan. 2025. Metaembed: Scaling multimodal retrieval at test-time with flexible late interaction. *arXiv preprint arXiv:2509.18095* (2025).

[35] Ruochen Yang, Yueyang Liu, Zijie Zhuang, Changxin Lao, Yuhui Zhang, Jiangxia Cao, Jia Xu, Xiang Chen, Haoke Xiao, Xiangyu Wu, et al. 2026. SARM: LLM-Augmented Semantic Anchor for End-to-End Live-Streaming Ranking. *arXiv preprint arXiv:2602.09401* (2026).

[36] Wencai Ye, Mingjie Sun, Shaoyun Shi, Peng Wang, Wenjin Wu, and Peng Jiang. 2025. DAS: Dual-Aligned Semantic IDs Empowered Industrial Recommender System. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*. 6217–6224.

[37] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917* (2022).

[38] Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni. 2023. Where to Go Next for Recommender Systems? ID- vs. Modality-based Recommender Models Revisited. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*. ACM, 2639–2649. doi:10.1145/3539618.3591932

[39] Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Gong, Fangda Gu, Jiayuan He, Yinghai Lu, and Yu Shi. 2024. Actions Speak Louder than Words: Trillion-Parameter Sequential Transducers for Generative Recommendations. In *Proceedings of the 41st International Conference on Machine Learning (ICML) (Proceedings of Machine Learning Research, Vol. 235)*. PMLR, 58484–58509.

[40] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *ICCV*.

[41] Buyun Zhang, Liang Luo, Yuxin Chen, Jade Nie, Xi Liu, Daifeng Guo, Yanli Zhao, Shen Li, Yuchen Hao, Yantao Yao, et al. 2024. Wukong: Towards a scaling law for large-scale recommendation. *arXiv preprint arXiv:2403.02545* (2024).

[42] Beichen Zhang, Pan Zhang, Xiaoyi Dong, Yuhang Zang, and Jiaqi Wang. 2024. *Long-CLIP: Unlocking the Long-Text Capability of CLIP*. Springer Nature Switzerland, 310–325. doi:10.1007/978-3-031-72983-6_18

[43] Chao Zhang, Haoxin Zhang, Shiwei Wu, Di Wu, Tong Xu, Xiangyu Zhao, Yan Gao, Yao Hu, and Enhong Chen. 2025. NoteLLM-2: Multimodal Large Representation Models for Recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1 (KDD '25)*. ACM, 2815–2826. doi:10.1145/3690624.3709440

[44] Ruifeng Zhang, Zexi Huang, Zikai Wang, Ke Sun, Bohang Zheng, Yuchen Jiang, Zhe Chen, Zhen Ouyang, Huimin Xie, Phil Shen, et al. 2026. Zenith: Scaling up Ranking Models for Billion-scale Livestreaming Recommendation. *arXiv preprint arXiv:2601.21285* (2026).

[45] Xin Zhang, Yanzhao Zhang, Wen Xie, Mingxin Li, Ziqi Dai, Dingkun Long, Pengjun Xie, Meishan Zhang, Wenjie Li, and Min Zhang. 2024. GME: improving universal multimodal retrieval by multimodal LLMs. *arXiv preprint arXiv:2412.16855* (2024).

[46] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176* (2025).

[47] Yuhan Zhao, Rui Chen, Qilong Han, Hongjun Li, and Li Chen. 2025. Aligning and Balancing ID and Multimodal Representations for Recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. ACM. doi:10.1145/3711896.3737275

[48] Carolina Zheng, Minhui Huang, Dmitrii Pedchenko, Kaushik Rangadurai, Siyu Wang, Fan Xia, Gaby Nahum, Jie Lei, Yang Yang, Tao Liu, Zutian Luo, Xiaohan Wei, Dinesh Ramasamy, Jiyan Yang, Yiping Han, Lin Yang, Hangjun Xu, Rong Jin, and Shuang Yang. 2025. Enhancing Embedding Representation Stability in Recommendation Systems with Semantic ID. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems (RecSys '25)*. ACM, 954–957. doi:10.1145/3705328.3748123

[49] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep Interest Evolution Network for Click-Through Rate Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 01 (2019), 5941–5948. doi:10.1609/aaai.v33i01.33015941

[50] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep Interest Network for Click-Through Rate Prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. ACM, 1059–1068. doi:10.1145/3219819.3219823