LinkedOut: Linking World Knowledge Representation Out of Video LLM for Next-Generation Video Recommendation

Haichao Zhang[1,†,‡] Yao Lu[2,†] Lichen Wang[2,†] Yunzhe Li[2]
Daiwei Chen[3] Yunpeng Xu[2] Yun Fu[1]
[1]Northeastern University [2]LinkedIn [3]University of Wisconsin–Madison
zhang.haich@northeastern.edu daiwei.chen@wisc.edu yunfu@ece.neu.edu
{ylu1, licwang, yunzli, yunxu}@linkedin.com

Abstract

Video Large Language Models (VLLMs) unlock world-knowledge-aware video understanding through pretraining on internet-scale data and have already shown promise on tasks such as movie analysis and video question answering. However, deploying VLLMs for downstream tasks such as video recommendation remains challenging, since real systems require multi-video inputs, lightweight backbones, low-latency sequential inference, and rapid response. In practice, (1) decode-only generation yields high latency for sequential inference, (2) typical interfaces do not support multi-video inputs, and (3) constraining outputs to language discards fine-grained visual details that matter for downstream vision tasks. We argue that these limitations stem from the absence of a representation that preserves pixel-level detail while leveraging world knowledge. We present LinkedOut, a representation that extracts VLLM world knowledge directly from video to enable fast inference, supports multi-video histories, and removes the language bottleneck. LinkedOut extracts semantically grounded, knowledge-aware tokens from raw frames using VLLMs, guided by promptable queries and optional auxiliary modalities. We introduce a cross-layer knowledge fusion MoE that selects the appropriate level of abstraction from the rich VLLM features, enabling personalized, interpretable, and low-latency recommendation. To our knowledge, LinkedOut is the first VLLM-based video recommendation method that operates on raw frames without handcrafted labels, achieving state-of-the-art results on standard benchmarks. Interpretability studies and ablations confirm the benefits of layer diversity and layer-wise fusion, pointing a practical way to better exploit VLLM world knowledge and visual reasoning for downstream tasks.

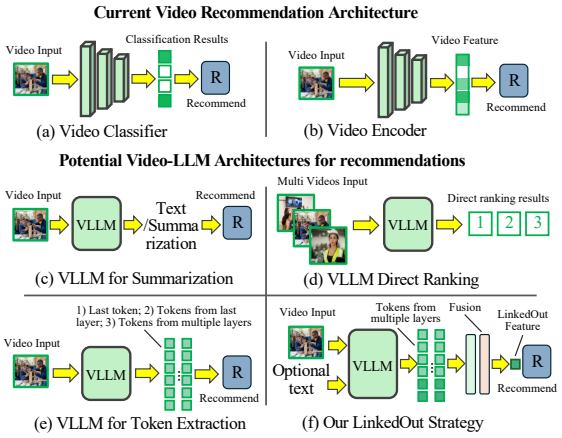

Figure 1. Conceptual differences of our LinkedOut and other methods. Existing approaches using video models either as a classifier or a feature extractor (i.e., (a) and (b)) which are trained on task specific data without LLM/world knowledge. Other solutions such as (c) VLLM for content summarization, (d) VLLM direct video ranking, or (e) VLLM for token level feature extraction, they can neither fully utilize the world knowledge from VLLM nor are computationally feasible for recommendation online serving. Our LinkedOut approach is able to fully utilize both pixel level visual information with VLLM knowledge. Videos are eventually transferred to feature vectors for fast online recommendation serving.

1. Introduction

Video recommendation is the primary interface through which viewers discover content at scale [6]. Its inherent need for lightweight backbones, low-latency sequential inference, and rapid response makes it a representative downstream setting for studying foundation-model integration. Despite remarkable advances in representation learning and ranking architectures [16, 27, 33, 35, 48], production systems still depend heavily on hand-crafted or pre-extracted labels (e.g., categories, tags, reactions) distilled from raw videos. This label-centric design reduces input dimension-

[†]Equal Contribution.
[‡]Work done as an intern at LinkedIn (a Microsoft company).

ality and eases serving, but it also discards the majority of the information present in pixels, limits semantic coverage to what was annotated in the training set, and makes personalization brittle in cold-start and long-tail regimes. Moreover, pipelines that summarize videos into text before recommendation inherit a language bottleneck: they constrain the system to preselected features and textual abstractions rather than the full visual token space. In particular, such pipelines struggle to recognize nuanced attributes (e.g., visual humor and narrative pacing) or to leverage broader factual or commonsense context and to adapt quickly to fast-evolving trends without expensive retraining.

Multimodal LLMs have recently demonstrated strong transferability from internet-scale pretraining, aligning visual tokens with natural language and encoding broad world knowledge [1, 19, 21, 31]. In video, joint training on raw frames and transcripts unlocks temporally grounded semantics and cross-modal reasoning [2, 26]. These models are promptable, so natural language can steer them to extract task-specific attributes and abstractions on demand without modifying the backbone. Transformer layers also specialize by depth: later layers aggregate global context and higher-level semantics, while earlier layers emphasize local patterns. This suggests an opportunity to link world knowledge and pixel-level cues from a video LLM directly to a recommender, without first reducing videos to text summaries, while adaptively selecting the semantic granularity that best serves each ranking decision. Adapting VLLMs to video recommendation is therefore a promising way to move beyond hand-crafted or pre-extracted label-centric pipelines and to bring world-knowledge-aware visual understanding and possible reasoning into recommendation.

Although many recent efforts have built video large language models (VLLMs) with world knowledge and reasoning that benefit tasks such as video content understanding, visual question answering, and video world modeling, the scope of these downstream uses remains narrow. They mainly target settings that can tolerate the current limitations of VLLMs. When a task requires multi-video inputs, lightweight backbones, low-latency sequential inference, and rapid response, these limitations become impractical for straightforward pipelines like Fig. 1(c–e).

Video recommendation is therefore a representative task for analyzing this problem. It must keep the pretrained world knowledge inside the VLLM, but its outputs are item indices from a database rather than natural language, so they do not follow the language distribution. Current VLLMs typically rely on text outputs due to the language interface. Some tasks, such as robotics planning, fine-tune VLLMs to produce structured actions, but this often narrows the output distribution and can lead to catastrophic forgetting of the original knowledge. In addition, current VLLMs (and many LLMs) are slow at inference time because of the decode-only transformer architecture: every newly generated token must be fed back as input for the next step, which lengthens the reasoning process and is unsuitable for real-time video recommendation. Typical VLLM interfaces also lack native support for multi-video inputs. A single video can already occupy tens of thousands of tokens, and large token budgets further increase latency and computation, making it difficult to design or post-train VLLMs that accept multiple videos at once. However, video recommendation usually needs to consume a sequence of historical videos per user and to encode a large candidate pool. These challenges have so far limited the use of VLLMs in tasks that simultaneously require multi-video inputs, lightweight models, low-latency sequential inference, and rapid response, while still needing world knowledge for visual reasoning, content understanding, and generalization.

To this end, we introduce ***LinkedOut***, a knowledge-aware, modality-extendable video recommendation framework that links world knowledge out of video pixels through a video LLM. The general structure is shown in Figure 2. To address the language-output constraint, we do not fine-tune the VLLM into the recommendation space, which would risk domain shift and catastrophic forgetting. Instead, we directly extract embeddings from intermediate token representations of the VLLM, since world knowledge is distributed across transformer layers and is computed jointly with the input tokens. To reconcile the different abstraction levels encoded across the backbone, we propose a cross-layer Knowledge-fusion Mixture-of-Experts that fuses representations from multiple transformer blocks. Concretely, *LinkedOut* combines both existing tokens and generated tokens within each layer, then applies cross-layer gating to produce a unified recommendation embedding that blends fine-grained visual cues with conceptual knowledge [8, 34].

To address slow inference and enable scalable deployment, *LinkedOut* adopts a store-and-retrieve architecture. An offline pipeline first precomputes compact *LinkedOut* features with the Cross-layer Knowledge-fusion MoE and stores them in an embedding database. At serving time, a lightweight recommendation module retrieves candidate embeddings and produces fast rankings conditioned on user context and prompts. This design preserves the benefits of content-aware modeling while keeping latency low, and it allows new modalities to be added offline without retraining the online ranker. It also strengthens support for multi-video inputs, since feature extraction and heavy reasoning are isolated from the online VLLM path, avoiding the quadratic token cost of feeding multiple videos into the VLLM at serving time. To move beyond traditional video recommendation that relies on tags or preselected features, our approach replaces fragile label taxonomies with dense, semantically grounded token embeddings extracted directly from raw frames. A promptable feature-focus module steers

the backbone toward user-relevant attributes or auxiliary modalities (for example, transcripts, audio, metadata), or toward dynamic context (for example, trending topics) using text prompts, which enables rapid adaptation without retraining. Empirically, *LinkedOut* achieves state-of-the-art performance on public video recommendation benchmarks, benefiting from VLLM world knowledge aggregated across layers. Our main contributions are as follows:

- We propose *LinkedOut*, a knowledge-aware, modality-extendable video recommendation framework that links world-knowledge features out of video pixels via a video LLM, thereby removing the language bottleneck. To the best of our knowledge, this is the first VLLM-based recommendation system that integrates a video LLM and consumes raw video frames directly, while avoiding text-only summarization bottlenecks and fully leveraging web-scale factual and commonsense knowledge from pretrained VLLMs.
- We design a *Cross-layer Knowledge-fusion MoE* that selects and concentrates the appropriate abstraction from different depths of intermediate VLLM tokens, producing a unified embedding that mixes fine-grained visual cues with high-level semantics under strict latency constraints, outperforming last-layer or last-token baselines and improving robustness across genres and user intents.
- For fast inference, we use a store-and-retrieve pipeline that precomputes compact *LinkedOut* features offline and supports low-latency online ranking with multi-video histories. On public video recommendation benchmarks, *LinkedOut* achieves state-of-the-art results, benefiting from world-knowledge priors concentrated across multiple layers of the VLLM.

2. Related Work

2.1. Multimodal LLMs for Video Understanding

Web-scale foundation models have substantially advanced multimodal representation learning by aligning visual inputs [23, 24] with natural language supervision. Early work such as CLIP learns image–text alignment via contrastive pretraining and enables strong zero-shot transfer across recognition tasks [31]. BLIP-2 introduces a lightweight query transformer to connect pretrained visual encoders with LLMs, improving data efficiency and generalization [19], while LLaVA scales visual instruction tuning to align LLMs with image-grounded instructions and open-ended dialogues [21].

Extending multimodal LLMs from images to video requires modeling temporal dynamics and long-range dependencies. Large-scale narrated video corpora such as HowTo100M provide web-scale paired video–text supervision that has catalyzed progress in video–language representation learning [26]. Architectures such as Frozen-in-Time jointly encode frames and text to support retrieval and transfer, offering an effective foundation for downstream video understanding [2]. More recent video LLM systems further combine frame-wise attention with instruction tuning to elicit temporally grounded semantics and world knowledge from long videos [1, 19]. In parallel, efficiency and evaluation for video understanding remain central challenges: VQToken proposes extreme token reduction to enable lightweight video-LLM inference with a minimal token budget [44], and Dense Video Understanding introduces dense, fine-grained evaluation settings that stress temporally detailed multimodal reasoning [45].

A consistent empirical trend in transformer models is that deeper layers aggregate broader context and encode higher-level abstractions, while earlier layers preserve more local and fine-grained patterns. This motivates layer-aware representations that can match the abstraction level required by a downstream task. Mixture-of-Experts (MoE) provides a scalable mechanism to route inputs to specialized experts and increase model capacity without a proportional increase in compute [34], complementing MoE advances in language models such as Switch Transformers [8]. Our work leverages these insights by extracting layer-wise token representations from a video LLM and fusing them via a cross-layer MoE, producing a unified embedding that integrates fine-grained visual cues with higher-level, knowledge-grounded semantics. Unlike prior MoE approaches that route across separate modules or models, we explicitly fuse cross-layer token features within a single VLLM, as different layers encode distinct semantic depths.

2.2. Multimodal LLMs in Recommendation

Traditional recommender systems rely on collaborative filtering [9, 12, 33] and sequential modeling [4, 7, 16, 43], which operate primarily on item IDs. To address cold-start and long-tail limitations, content-aware recommendation leverages auxiliary modalities. Early approaches extract fixed labels or shallow features from pretrained networks [6]. Recent multimodal recommenders integrate richer features through graph convolution (MMGCN [38]) or graph-based representations (GRCN [39]). Recent foundation models offer a transformative alternative: CLIP-like features [31] enhance retrieval via zero-shot transfer, while multimodal LLMs such as Flamingo [1], BLIP-2 [19], and LLaVA [21] unlock flexible conditioning via natural-language prompts and provide web-scale world knowledge. For video recommendation specifically, three paradigms have been explored: first, using frozen pretrained video models to extract fixed features [28]; second, jointly training video encoders end-to-end with recommendation objectives [28]; and third, summarizing videos into text via captioning or ASR, then applying text-based LLMs.

However, existing approaches typically suffer from three

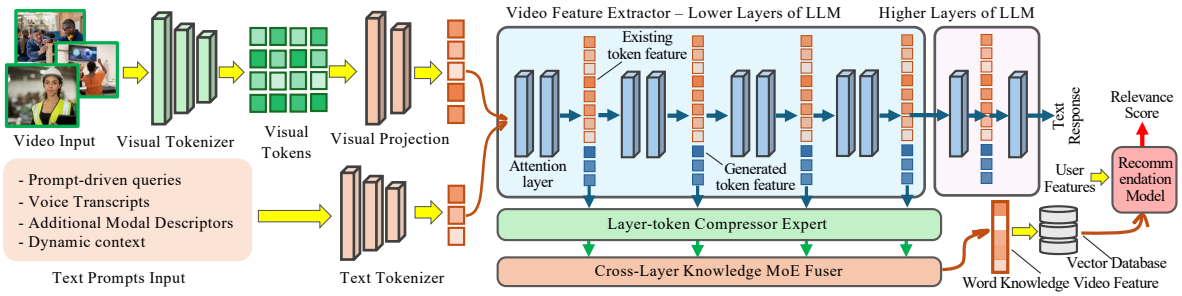

Figure 2. LinkedOut framework. Raw video frames are tokenized and projected into a pretrained video LLM. The optional text queries (e.g. transcripts, metadata, and dynamic context) are co-tokenized and injected to guide extraction. As tokens traverse lower to higher transformer layers, cross-modal attention enriches the sequence with both existing visual/text tokens and LLM-generated knowledge tokens. A Layer-token Compressor Expert condenses token sequences, and a Layer Knowledge Fuser MoE gates across layers to adaptively select the appropriate semantic depth, yielding a unified, world-knowledge-aware video embedding. At serving time, user features and video embeddings are passed to a lightweight recommendation model to produce a relevance score. The promptable design supports flexible modality fusion and selective fine-tuning, enabling personalized, interpretable, and cold-start-robust video recommendation.

limitations: (1) frozen encoders treat multimodal features as static descriptors, limiting adaptability to recommendation-specific semantics and user context; (2) end-to-end training achieves better performance but incurs prohibitive computational costs; and (3) text-based summarization introduces a language bottleneck that discards pixel-level visual information while relying on hand-crafted prompts or fixed layers that may provide either insufficient or excessive abstraction. LinkedOut addresses these gaps by combining knowledge-aware extraction directly from video pixels, adaptive layer-wise fusion that selects appropriate semantic granularity per instance, and efficient decoupling of offline embedding computation from online inference, thereby bridging foundation model capabilities with production-grade recommender constraints.

3. LinkedOut Approach

3.1. Preliminaries and Interpretability Analysis

Preliminaries on VLLMs. A typical video LLM (VLLM) first tokenizes each input modality (for example, video frames, text prompts, audio) with its corresponding tokenizer. The resulting tokens are projected to a common embedding space and concatenated to form a single token sequence, which is then fed into the LLM to perform reasoning with the world knowledge stored in the model [15, 30, 36, 40]. The LLM itself is composed of multiple transformer layers. World knowledge is distributed across these pretrained transformer layers. At each layer, token representations are updated through self-attention and feed-forward blocks, producing new token embeddings that are passed to the next layer in an autoregressive next-token prediction manner.

Old and new tokens. During decoding, each transformer layer first applies (multi-head) self-attention or cross-attention over all input tokens from all modalities. After this first pass, the model starts to generate tokens autoregressively: each newly predicted token is appended to the previous tokens and used as input for the next step. KV cache [29] can speed up this process, but the overall decoding is still sequential. For clarity, at a given layer we refer to tokens whose indices are within the original input length as *old* tokens, and to tokens whose indices exceed the original input length as *new* tokens.

Layer-wise token representation and interpretability. Recent studies suggest that different transformer layers tend to capture different levels of knowledge [5, 17, 22, 32]. For example, in a 24-layer LLM, we can loosely group layers into early, middle, and late stages: early layers typically encode local visual patterns (such as edges and textures) and basic token embeddings [37, 46]; middle layers often capture cross-modal alignment [14, 41] (for example, region-to-phrase), object parts, relations, and shallow semantic groupings; and late layers are more associated with global scene understanding, abstract concept grounding, and high-level multimodal reasoning [5, 20, 47]. This motivates extracting and fusing representations from multiple depths rather than relying only on the last layer.

LinkedOut representation definition. Since different layers encode different levels of visual and semantic knowledge, and the final text output is only a projection of the internal reasoning of a VLLM, our goal is to extract both old and new tokens across multiple layers so that we preserve and concentrate the world-knowledge embeddings inside the model [5, 15, 36]. We formalize *LinkedOut* as a content-aware, knowledge-grounded representation that reuses a pretrained video LLM as a feature extractor and exposes world knowledge through prompts. Let a video v be sampled into frames $\{x_t\}_{t=1}^{T}$, and let optional side channels (for example, ASR transcripts, metadata, tags) be s. A natural-language prompt p steers the backbone toward

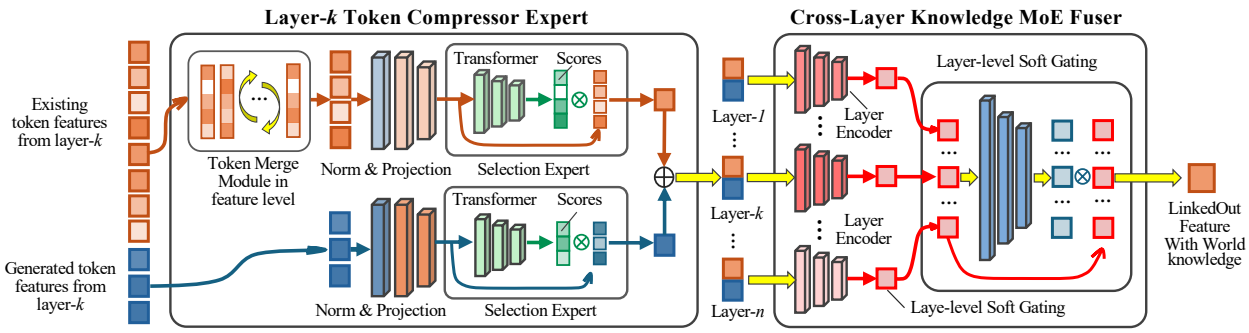

Figure 3. Overview of the Cross-Layer Knowledge-Fusion MoE. *Token Compressor Expert* receives existing and generated token features. A token-merge module aggregates redundant signals at the feature level, followed by normalization and projection. A lightweight selection expert predicts scores to reweight tokens; the two streams are then merged to yield a compact, semantically faithful representation for layer k. *Cross-Layer Knowledge MoE Fuser* treats each layer as an expert. Layer encoders summarize the compressed tokens, and layer-level soft-gating produces context-conditioned mixture weights that adaptively combine information across layers. The resulting LinkedOut feature fuses fine-grained visual details with world knowledge, providing a single, retrieval-ready embedding that improves efficiency while preserving recommendation-relevant semantics.

task-relevant attributes. The video LLM $\mathcal{L}$ has L transformer blocks and produces hidden states $\{H^{(\ell)}\}_{\ell=1}^{L}$ for the joint sequence of visual and textual tokens. Below we describe how these layer-wise token features are collected; the Cross-Layer Knowledge MoE Fuser is presented in Section 3.3.2. Figure 3 illustrates the two core modules: the Token Compressor Expert and the Cross-Layer MoE.

3.2. Raw World-Knowledge Token Extraction

Because world knowledge is encoded in the pretrained transformer weights and applied to token embeddings at every layer, we directly extract token representations from intermediate layers of the LLM. Our Raw World-Knowledge Token Extraction module replaces predefined or handcrafted labels with dense intermediate token features obtained from a frozen (or selectively tuned) video LLM [1, 19, 21]. Concretely, each frame x_t is tokenized into patches by a vision tokenizer $g(\cdot)$, projected to the LLM embedding space by a lightweight adaptor ϕ (initialized from the vision–text bridge of the backbone), and concatenated with text tokens:

$$Z = \big[\phi(g(x_1)), \ldots, \phi(g(x_T)),\ \mathrm{Tok}(p, s)\big]. \qquad (1)$$

The backbone $\mathcal{L}$ then processes Z and produces layer-wise hidden states $H^{(\ell)} \in \mathbb{R}^{N_\ell \times d}$, where N_ℓ is the token length at layer ℓ and d is the model width.

3.3. Cross-Layer Knowledge-Fusion MoE

The Cross-Layer Knowledge-Fusion MoE forms the *LinkedOut* representation from a VLLM in two steps. First, the *Layer Token Compressor Expert* condenses old and new tokens within each layer into compact features. Second, the *Cross-Layer Knowledge MoE Fuser* assigns data-dependent weights across layers and combines their features to produce a unified, knowledge-aware item embedding. These two components are tightly coupled by design; removing either breaks the construction of *LinkedOut*.

3.3.1. Layer Token Compressor Expert

Tokens at each layer can be numerous, creating a heavy computational burden and making it hard for downstream modules to determine each layer's contribution. To make layer-wise features compact and comparable, we assign a token-compression expert to every layer, denoted $C^{(\ell)}$ (see the left panel of Figure 3). Each expert processes old and new tokens with separate branches, since old tokens mainly encode accumulated context while new tokens capture freshly generated knowledge. Concretely, the old branch first performs token merging, then compression; the new branch compresses tokens directly. We finally concatenate the two branch outputs to form the layer representation.

$$\mathbf{e}^{(\ell)} = \mathrm{concat}\Big(C^{(\ell)}_{\mathrm{old}}\big(\mathrm{Merge}^{(\ell)}_{\mathrm{old}}(H^{(\ell)}_{\mathrm{old}})\big),\ C^{(\ell)}_{\mathrm{new}}\big(H^{(\ell)}_{\mathrm{new}}\big)\Big).$$

Here $C^{(\ell)}_{\mathrm{old}}$ and $C^{(\ell)}_{\mathrm{new}}$ are lightweight attention-pooling modules with learnable queries; the old branch uses more aggressive merging [3] to reduce redundancy. We apply this to every N layers (for example, every 2 or 4 layers), which empirically preserves the progression of world knowledge across depth while keeping the representation concentrated.

3.3.2. Cross-Layer Knowledge MoE Fuser

Transformers at different depths in a VLLM organize information from local patterns to global abstractions. We have already concentrated knowledge in the old and new tokens of each layer, but we still need to learn which layers are most useful for a given downstream instance.

To select the right semantic granularity per video item, we fuse layer-wise features with a Mixture-of-Experts

Table 1. Comparison on MicroLens-50K and MicroLens-100K, the only publicly available raw-video datasets for video recommendation to our knowledge.

Model	MicroLens-50K				MicroLens-100K			
	HR @10	HR @20	NDCG @10	NDCG @20	HR @10	HR @20	NDCG @10	NDCG @20
YouTube [6]	0.0375	0.0632	0.0178	0.0245	0.0461	0.0747	0.0229	0.0301
VBPR [10]	0.0544	0.0888	0.0273	0.0361	0.0624	0.1002	0.0314	0.0410
MMGCN [38]	0.0403	0.0670	0.0197	0.0264	0.0214	0.0374	0.0103	0.0143
LightGCN [12]	0.0365	0.0534	0.0284	0.0345	0.0372	0.0618	0.0177	0.0239
GRCN [39]	0.0631	0.0982	0.0328	0.0415	0.0282	0.0497	0.0131	0.0185
LayerGCN [50]	0.0627	0.0994	0.0320	0.0414	0.0730	0.1120	0.0382	0.0480
BM3 [51]	0.0565	0.0918	0.0281	0.0372	0.0601	0.0975	0.0305	0.0401
Freedom [49]	0.0656	0.1028	0.0334	0.0429	0.0654	0.1016	0.0337	0.0431
MGCN [42]	0.0708	0.1089	0.0363	0.0459	0.0717	0.1096	0.0371	0.0467
MHCR [25]	0.0736	0.1102	0.0383	0.0477	0.0798	0.1187	0.0420	0.0519
LinkedOut (Ours)	**0.0761**	**0.1127**	**0.0399**	**0.0491**	**0.1015**	**0.1477**	**0.0548**	**0.0664**

(MoE) [8, 34] (see the right panel of Figure 3). Each of the last N tapped layers provides a compressed vector $\tilde{\mathbf{e}}^{(\ell)} \in \mathbb{R}^{d_c}$ (after token compression and prompt mixing). A per-layer expert then maps this vector to a common space:

$$\mathbf{h}^{(\ell)} = E^{(\ell)}\big(\tilde{\mathbf{e}}^{(\ell)}\big) \in \mathbb{R}^{d_z}, \tag{2}$$

where $E^{(\ell)}$ is a lightweight MLP with residual connections. An item-conditioned gate then assigns soft weights over layers without circular dependence:

$$\pi = \mathrm{Softmax}\big(G(\mathbf{z}_v)\big) \in \mathbb{R}^N, \quad \mathbf{z}_v = \sum_{\ell=1}^{N} \pi_\ell \, \mathbf{h}^{(\ell)}, \tag{3}$$

where $G(\cdot)$ is a gating MLP. The resulting embedding $\mathbf{z}_v$ is the unified, world-knowledge-aware item representation used by the ranker. We use dense soft weights over all tapped layers. A sparse top-k variant with load balancing can be adopted if desired, but is unnecessary here given the modest number of experts and layers.

3.4. Store-and-Retrieve Architecture

We adopt a store-and-retrieve architecture for fast VLLM-based inference. As shown in Fig. 1(c–f), applying a VLLM directly at serving time is too slow. In practical recommendation systems, however, video feature extraction runs offline and is updated periodically, separate from the online inference path. Since $\tilde{\mathbf{e}}^{(\ell)}$ are precomputed offline for each item (and for each prompt bank), the online stage only evaluates the gating network and a small set of selected experts. The gate weights π indicate which abstraction levels contribute to the final score, and the prompt mixture reveals which attributes were emphasized, yielding human-readable explanations.

Offline item embeddings are obtained by running the extractor on all catalog videos with a small set of canonical prompts. The resulting per-layer vectors are stored in a feature store keyed by item IDs, which enables low-latency retrieval and online ranking without re-encoding videos. The extractor scales linearly with the number of frames T and the number of selected layers N. Token compression reduces memory from $O(TNd)$ to $O(d_c)$ per item, where d is the original token dimension and d_c is the compressed world-knowledge dimension, which accelerates encoding and makes the store-and-retrieve pipeline practical at scale.

Table 2. Performance Comparison by Video-Recommendation Model Types on MicroLens-100k benchmark.

Model	Type	HR @10	NDCG @10	HR @20	NDCG @20
DSSM [13]	IDRec (CF)	0.0394	0.0193	0.0654	0.0258
LightGCN [12]		0.0372	0.0177	0.0618	0.0239
NFM [11]		0.0313	0.0159	0.0480	0.0201
DeepFM [9]		0.0350	0.0170	0.0571	0.0225
NextItNet [43]	IDRec (SR)	0.0805	0.0442	0.1175	0.0535
GRU4Rec [7]		0.0782	0.0423	0.1147	0.0515
SASRec [16]		0.0909	0.0517	0.1278	0.0610
YouTubeID [6]	VID Rec	0.0461	0.0229	0.0747	0.0301
YouTubeID+V [6]		0.0392	0.0188	0.0648	0.0252
MMGCNID [38]		0.0141	0.0065	0.0247	0.0092
MMGCNID+V [38]		0.0214	0.0103	0.0374	0.0143
GRCNID [39]		0.0282	0.0131	0.0497	0.0185
GRCNID+V [39]		0.0306	0.0144	0.0547	0.0204
DSSMID+V [13]		0.0279	0.0137	0.0461	0.0183
SASRecID+V [16]		0.0799	0.0415	0.1217	0.0520
NextItNetV [43]	Video Rec	0.0862	0.0468	0.1246	0.0562
GRU4RecV [7]		0.0954	0.0517	0.1377	0.0623
SASRecV [16]		0.0948	0.0515	0.1364	0.0619
LinkedOut (Ours)	Video LLMRec	**0.1015**	**0.0548**	**0.1477**	**0.0664**

4. Experiments

4.1. Datasets

We evaluate on MicroLens-50K and MicroLens-100K [28], two large-scale, content-driven micro-video recommendation benchmarks that, to the best of our knowledge, are the *only* public datasets providing raw videos for content-aware modeling. Each video includes rich multimodal content, such as raw frames (mostly under 400 seconds), audio tracks, titles, cover images, and user comments, which makes these datasets suitable for evaluating foundation-model-driven recommendation. Interactions are timestamped comments spanning 15,580 fine-grained tags (for example, food, sports, travel, entertainment), enabling sequential modeling. We follow the official training and testing protocol in [28].

4.2. Benchmarks

We compare LinkedOut with state-of-the-art video recommendation baselines, which can be broadly categorized into three groups: (1) **IDRec** [28] methods rely solely on item IDs without leveraging video content. They include collaborative filtering (CF) approaches: **DSSM** [13] learns user/item embeddings via dual-tower matching; **LightGCN**

Table 3. Ablation study of component effectiveness analysis.

Variant	HR@10	NDCG@10	HR@20	NDCG@20
Last layer last token	0.0763	0.0404	0.1121	0.0494
Mean pooling+MoE	0.0888	0.0471	0.1291	0.0572
Last token+MoE	0.0958	0.0518	0.1394	0.0628
LinkedOut (full)	**0.1015**	**0.0548**	**0.1477**	**0.0664**

[12] applies graph convolution on the user-item bipartite graph; **NFM** [11] models second-order feature interactions with neural bi-interaction; **DeepFM** [9] combines factorization machines with deep networks. Sequential recommendation (SR) models include **GRU4Rec** [7] with recurrent units, **SASRec** [16] with self-attention, and **NextItNet** [43] with dilated convolutions. (2) **VIDRec** methods augment IDRec with frozen, pre-extracted video features as side information, denoted by the "+V" suffix (e.g., **YouTubeID+V** [6], **MMGCNID+V** [38], **GRCNID+V** [39], **DSSMID+V**, **SASRecID+V**). These methods concatenate video embeddings with ID embeddings while keeping the video encoder frozen. **YouTubeID** [6] is a two-stage system with candidate generation and ranking; **MMGCN** [38] propagates user-item signals separately per modality; **GRCN** [39] refines collaborative filtering with graph-based representations. (3) **VideoRec** [28] methods jointly train a video encoder with the recommender via end-to-end (E2E) learning, replacing ID embeddings with learnable video representations (**GRU4RecV** [7], **SASRecV** [16], **NextItNetV** [43]). These E2E models are computationally expensive but achieve superior performance by optimizing video understanding and recommendation simultaneously [28].

4.3. Implementation Details

VLLM extraction. We use LLaVA-OneVision [18] 7B as the VLLM backbone and freeze its parameters during feature extraction. We attach PyTorch hooks to capture token embeddings during generation: as the model decodes, previously generated tokens are first processed by the decoder and new tokens are produced one by one. To reduce computation, we record token representations every four transformer layers inside the VLLM.

Recommendation model training. We train on 8× NVIDIA H100 GPUs with mixed precision and a batch size of 256 sequences. Each user history contains at most 10 videos. Unless noted otherwise, we train for 100 epochs with AdamW and parameter-group-specific learning rates: 0.0001 for the video encoder (when unfrozen), 0.0001 for the text and image encoders (with pooler and classifier heads separated), and 0.00001 for the recommendation head. We set weight decay to 0.1 and clip the global gradient norm at 5. Images are resized to 224×224. We use a `step_schedule_with_warmup` scheduler at the epoch granularity (gap $=$ 1 epoch, $\alpha = 1.0$) with zero warmup steps by default. The total loss combines an alignment term and a uniformity term, reported per batch, and is optimized jointly with the recommendation objective.

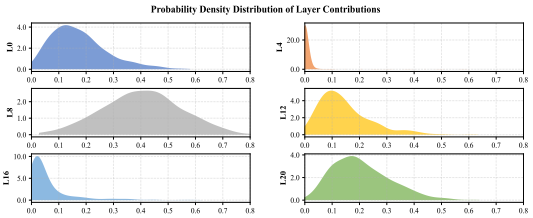

Figure 4. Probability density distribution of layer-wise MoE gate weights via kernel density estimation (KDE). Each subplot shows the distribution of normalized contribution values for a transformer layer (L0, L4, L8, L12, L16, L20). Distribution width and peak location reveal utilization patterns, where L8 shows the broadest distribution centered at 0.4, while L4 and L16 exhibit more concentrated distributions with relatively smaller contributions.

4.4. Performance Analysis

Performance Comparison by Video-Recommendation Model Class. Table 2 compares LinkedOut with state-of-the-art baselines on MicroLens-100K across four categories. LinkedOut achieves the best performance with HR@10 of 0.1015 and NDCG@10 of 0.0548, outperforming all baseline approaches by substantial margins. Compared to the strongest ID-based sequential recommender (SASRec, HR@10: 0.0909), LinkedOut achieves an 11.7% relative improvement, demonstrating that foundation-model-driven video understanding provides richer semantics beyond collaborative filtering patterns. Against the best end-to-end VideoRec model (GRU4RecV, HR@10: 0.0954), LinkedOut achieves a 6.4% relative gain, validating that our layer-wise MoE fusion and token compression effectively extract more discriminative features than standard video encoders. Notably, most VIDRec methods (e.g., MMGCNID+V, GRCNID+V) underperform their ID-only counterparts, suggesting that naively concatenating frozen video features with ID embeddings introduces noise rather than useful signals. In contrast, LinkedOut's adaptive layer selection and token-level aggregation enable fine-grained control over which semantic levels contribute to recommendation, avoiding the feature degradation observed in VIDRec approaches. The consistent improvements across all metrics (HR@10, NDCG@10, HR@20, NDCG@20) confirm that LinkedOut captures both relevance and ranking quality more effectively.

Comparison across datasets. Table 1 compares *LinkedOut* with recent baselines on the two public raw-video datasets, MicroLens-50K and MicroLens-100K. *LinkedOut* improves all four metrics (HR@10, HR@20, NDCG@10, NDCG@20) on both datasets, indicating consistent performance across different dataset sizes and content complexity.

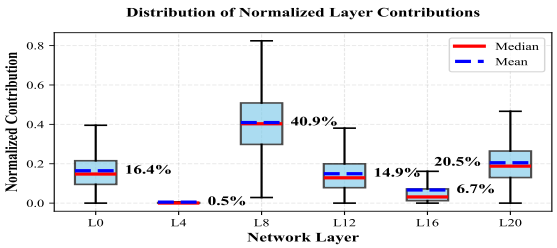

Figure 5. Statistical summary of layer-wise MoE gate contributions. Box plots show distribution of normalized contribution values per layer, with red lines (median) and blue dashed lines (mean). Annotated percentages are average contributions, where L8 dominates at 40.9%, followed by L20 (20.5%) and L0 (16.4%), while L4 (0.5%) and L16 (6.7%) contribute relatively less.

4.5. Ablation Study

Component effectiveness analysis. To validate the importance of our design choices, we conduct ablation studies on the components. Table 3 reports results on MicroLens-100K. First, we evaluate a baseline that uses only the last layer last token without MoE fusion. This simpler approach mirrors standard LLM extraction practices, resulting in significant performance drops (HR@10: 0.0763 vs. 0.1015), demonstrating that both layer-wise fusion and token-level aggregation are crucial. Second, we examine mean pooling with MoE, which applies layer-wise fusion but uses simple average pooling across all tokens at each layer. While this variant benefits from multi-layer knowledge (HR@10: 0.0888), it treats all tokens equally and fails to emphasize recommendation-relevant features. Third, we test last token with MoE, where we apply layer-wise fusion but only use the final token from each layer. This variant achieves better performance (HR@10: 0.0958), suggesting that the last token contains more task-relevant information, yet it still underperforms our complete model by ignoring rich intermediate token representations. These ablations confirm that LinkedOut's complete design is essential for achieving SOTA performance. Each component contributes meaningfully to the final results.

Layer-wise contribution analysis. To understand which semantic levels are most valuable for video recommendation, we analyze the learned MoE gate weights across all validation samples. We visualize layer contributions through two complementary perspectives: Figure 4 shows probability density distributions via kernel density estimation (KDE) for each tapped layer (L0, L4, L8, L12, L16, L20); Figure 5 provides box plot statistics of median, mean, and distribution spread, annotated with average contribution percentages. Our key finding is that intermediate layers, particularly L8, dominate with 40.9% contribution, significantly outperforming early layers (L0: 16.4%) and later layers (L20: 20.5%), while L4 and L16 contribute minimally (0.5% and 6.7%). The KDE distributions further show that L8 exhibits broad, stable patterns across diverse videos, whereas L4 and L16 are more concentrated. This confirms that video recommendation benefits from adaptive multi-level semantic features, with mid-level representations balancing visual details and abstract concepts rather than relying solely on the deepest layers as commonly assumed in vision-language models.

Table 4. Inference-time comparison for potential design of *VLLM-based* recommendation. We report latency for direct VLLM serving and for our Store-and-Retrieve design; a traditional non-VLLM module is shown only as a reference for systems-level latency and is not a semantic baseline. The goal is to bring VLLM-based approaches to practical serving latency.

VLLM-based?	Description	Type	Time
✗	Traditional video recommendation *[reference only]*	Fig.1 (a,b)	0.864 ms
✓	Direct VLLM-based	Fig.1 (c,d)	5510 ms
✓	Store-and-Retrieve VLLM + Video recommendation module (LinkedOut)	Fig.1 (f)	5.964 ms

Efficiency Analysis. To demonstrate the computational advantages of our Store-and-Retrieve architecture, we compare inference latency across three approaches on a single NVIDIA H100 GPU (Table4). Traditional video recommendation without VLLM (Fig.1 a,b) achieves 0.864ms per inference, while direct real-time VLLM inference (Fig.1 c,d) incurs a prohibitive 5.51s latency, making it infeasible for deployment. LinkedOut's Store-and-Retrieve paradigm (Fig.1 f) decouples offline feature extraction (5.02s per video when batch-processing 100 videos, paid once) from online inference, where MoE fusion (5.1ms) and recommendation ranking (0.846ms) total just 5.964ms per query nearly 1000× faster than direct VLLM while preserving rich semantic understanding. This efficiency gain validates that VLLM-driven video recommendation is practical at scale.

5. Conclusion

We presented LinkedOut, a knowledge-aware video recommendation framework that links world knowledge out of raw pixels via pretrained video LLMs. By extracting semantically grounded token representations directly from frames, LinkedOut eliminates the language bottleneck and hand-crafted-label constraints of conventional approaches. Our promptable feature focus module enables flexible steering without retraining, while the cross-layer MoE adaptively fuses representations across transformer depths to select the right semantic granularity per instance. Experiments on MicroLens-50K and MicroLens-100K demonstrate state-of-the-art performance, with consistent gains in personalization, cold-start, and long-tail scenarios. Ablation studies confirm the importance of layer-wise fusion, promptable extraction, and adaptive semantic depth selection. These results highlight a practical path toward next-generation recommendation systems that fully leverage video LLM world knowledge and pixel-level visual semantics for robust, scalable video recommendation.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning, 2022. 2, 3, 5

[2] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1728–1738, 2021. 2, 3

[3] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. 2022. 5

[4] Daiwei Chen, Zhoutong Fu, Chengming Jiang, Haichao Zhang, Ran Zhou, Tan Wang, Chunnan Yao, Guoyao Li, Rui Cai, Yihan Cao, et al. Grounded token initialization for new vocabulary in lms for generative recommendation, 2026. 3

[5] Haoran Chen, Junyan Lin, Xinghao Chen, Yue Fan, Jianfeng Dong, Xin Jin, Hui Su, Jinlan Fu, and Xiaoyu Shen. Multimodal language models see better when they look shallower. pages 6688–6706, 2025. 4

[6] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198, 2016. 1, 3, 6, 7

[7] Michal Dobrovolny, Ali Selamat, and Ondrej Krejcar. Session based recommendations using recurrent neural networks - long short-term memory. In *Intelligent Information and Database Systems: 13th Asian Conference, ACIIDS 2021, Phuket, Thailand, April 7–10, 2021, Proceedings*, page 53–65, Berlin, Heidelberg, 2021. Springer-Verlag. 3, 6, 7

[8] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, 2022. 2, 3, 6

[9] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: a factorization-machine based neural network for ctr prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 1725–1731, 2017. 3, 6, 7

[10] Ruining He and Julian McAuley. Vbpr: visual bayesian personalized ranking from implicit feedback. In *Proceedings of the AAAI conference on artificial intelligence*, 2016. 6

[11] Xiangnan He and Tat-Seng Chua. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 355–364, 2017. 6, 7

[12] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648, 2020. 3, 6, 7

[13] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, page 2333–2338, New York, NY, USA, 2013. Association for Computing Machinery. 6

[14] Qidong Huang, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Jiaqi Wang, Weiming Zhang, and Nenghai Yu. Deciphering cross-modal alignment in large vision-language models via modality integration rate. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 218–227, 2025. 4

[15] Jindong Jiang, Xiuyu Li, Zhijian Liu, Muyang Li, Guo Chen, Zhiqi Li, De-An Huang, Guilin Liu, Zhiding Yu, Kurt Keutzer, et al. Storm: Token-efficient long video understanding for multimodal llms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5830–5841, 2025. 4

[16] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE, 2018. 1, 3, 6, 7

[17] Michael A Lepori, Alexa R Tartaglini, Wai K Vong, Thomas Serre, Brenden M Lake, and Ellie Pavlick. Beyond the doors of perception: Vision transformers represent relations between objects. *Advances in Neural Information Processing Systems*, 37:131503–131544, 2024. 4

[18] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-onevision: Easy visual task transfer. *Transactions on Machine Learning Research*, 2025. 7

[19] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. 2, 3, 5

[20] Zhihang Lin, Mingbao Lin, Luxi Lin, and Rongrong Ji. Boosting multimodal large language models with visual tokens withdrawal for rapid inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5334–5342, 2025. 4

[21] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023. 2, 3, 5

[22] Nelson F Liu, Matt Gardner, Yonatan Belinkov, Matthew E Peters, and Noah A Smith. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, 2019. 4

[23] Jianglin Lu, Hailing Wang, Yi Xu, Yizhou Wang, Kuo Yang, and Yun Fu. Representation potentials of foundation models for multimodal alignment: A survey. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 16680–16695, 2025. 3

[24] Jianglin Lu, Hailing Wang, Kuo Yang, Yitian Zhang, Simon Jenni, and Yun Fu. The indra representation hypothesis. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 3

[25] Sisuo Lyu, Xiuze Zhou, and Xuming Hu. Multi-view hypergraph-based contrastive learning model for cold-start micro-video recommendation. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025. 6

[26] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2630–2640, 2019. 2, 3

[27] Maxim Naumov, Dheevatsa Mudigere, Hao-Jun Michael Shi, Jianyu Huang, Narayanan Sundaraman, Jongsoo Park, Xiaodong Wang, Udit Gupta, Carole-Jean Wu, Alisson G Azzolini, et al. Deep learning recommendation model for personalization and recommendation systems. *arXiv preprint arXiv:1906.00091*, 2019. 1

[28] Yongxin Ni, Yu Cheng, Xiangyan Liu, Junchen Fu, Youhua Li, Xiangnan He, Yongfeng Zhang, and Fajie Yuan. A content-driven micro-video recommendation dataset at scale. pages 6486–6491, 2025. 3, 6, 7

[29] Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. *Proceedings of machine learning and systems*, 5: 606–624, 2023. 4

[30] Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models. *Advances in Neural Information Processing Systems*, 37:119336–119360, 2024. 4

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 2, 3

[32] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in neural information processing systems*, 34:12116–12128, 2021. 4

[33] Steffen Rendle. Factorization machines. In *2010 IEEE International conference on data mining*, pages 995–1000. IEEE, 2010. 1, 3

[34] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34:8583–8595, 2021. 2, 3, 6

[35] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450, 2019. 1

[36] Yunlong Tang, Jing Bi, Siting Xu, Luchuan Song, Susan Liang, Teng Wang, Daoan Zhang, Jie An, Jingyang Lin, Rongyi Zhu, Ali Vosoughi, Chao Huang, Zeliang Zhang, Pinxin Liu, Mingqian Feng, Feng Zheng, Jianguo Zhang, Ping Luo, Jiebo Luo, and Chenliang Xu. Video understanding with large language models: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 36(2): 1355–1376, 2026. 4

[37] Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. pages 4593–4601, 2019. 4

[38] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. Mmgcn: Multi-modal graph convolution network for personalized recommendation of micro-video. In *Proceedings of the 27th ACM international conference on multimedia*, pages 1437–1445, 2019. 3, 6, 7

[39] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, and Tat-Seng Chua. Graph-refined convolutional network for multimedia recommendation with implicit feedback. In *Proceedings of the 28th ACM international conference on multimedia*, pages 3541–3549, 2020. 3, 6, 7

[40] Jiayang Wu, Wensheng Gan, Zefeng Chen, Shicheng Wan, and Philip S Yu. Multimodal large language models: A survey. In *2023 IEEE International Conference on Big Data (BigData)*, pages 2247–2256. IEEE, 2023. 4

[41] Heeji Yoon, Jaewoo Jung, Junwan Kim, Hyungyu Choi, Heeseong Shin, Sangbeom Lim, Honggyu An, Chaehyun Kim, Jisang Han, Donghyun Kim, et al. Visual representation alignment for multimodal large language models. *arXiv preprint arXiv:2509.07979*, 2025. 4

[42] Penghang Yu, Zhiyi Tan, Guanming Lu, and Bing-Kun Bao. Multi-view graph convolutional network for multimedia recommendation. In *Proceedings of the 31st ACM international conference on multimedia*, pages 6576–6585, 2023. 6

[43] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. A simple convolutional generative network for next item recommendation, 2019. 3, 6, 7

[44] Haichao Zhang and Yun Fu. Vqtoken: Neural discrete token representation learning for extreme token reduction in video large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 3

[45] Haichao Zhang, Wenhao Chai, Shwai He, Ang Li, and Yun Fu. Dense video understanding with gated residual tokenization. *arXiv preprint arXiv:2509.14199*, 2025. 3

[46] Zhi Zhang, Srishti Yadav, Fengze Han, and Ekaterina Shutova. Cross-modal information flow in multimodal large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19781–19791, 2025. 4

[47] Shiyu Zhao, Zhenting Wang, Felix Juefei-Xu, Xide Xia, Miao Liu, Xiaofang Wang, Mingfu Liang, Ning Zhang, Dimitris N Metaxas, and Licheng Yu. Accelerating multimodal large language models by searching optimal vision token reduction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29869–29879, 2025. 4

[48] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction.

In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1059–1068, 2018. 1

[49] Xin Zhou and Zhiqi Shen. A tale of two graphs: Freezing and denoising graph structures for multimodal recommendation. In *Proceedings of the 31st ACM international conference on multimedia*, pages 935–943, 2023. 6

[50] Xin Zhou, Donghui Lin, Yong Liu, and Chunyan Miao. Layer-refined graph convolutional networks for recommendation. In *2023 IEEE 39th international conference on data engineering (ICDE)*, pages 1247–1259. IEEE, 2023. 6

[51] Xin Zhou, Hongyu Zhou, Yong Liu, Zhiwei Zeng, Chunyan Miao, Pengwei Wang, Yuan You, and Feijun Jiang. Bootstrap latent representations for multi-modal recommendation. In *Proceedings of the ACM web conference 2023*, pages 845–854, 2023. 6