

VILLA-MMBENCH: A Unified Benchmark Suite for LLM-Augmented Multimodal Movie Recommendation

Fatemeh Nazary¹, Ali Tourani², Yashar Deldjoo^{1,*}, Tommaso Di Noia¹

¹Department of Electrical and Information Engineering, Polytechnic University of Bari, Bari, Italy

²Interdisciplinary Centre for Security, Reliability, and Trust (SnT), University of Luxembourg, Luxembourg

{fatemeh.nazary, tommaso.dinoia}@poliba.it, deldjooy@acm.org, ali.tourani@uni.lu

*Corresponding author

August 7, 2025

Abstract

Recommending long-form video content requires an integrated treatment of visual, audio and textual modalities, yet most benchmarks focus on either raw item features or narrow fusion pipelines. We introduce **VILLA-MMBENCH**, a fully reproducible, extensible benchmark suite for next-generation LLM-augmented multimodal movie recommendation research. The toolkit leverages the widely-used MovieLens and MMTF-14K datasets, integrating and aligning item-level dense embeddings from three modalities: audio (block-level features and i-vector), visual (CNN and AVF), and text. Notably, it automatically augments missing or sparse item metadata using state-of-the-art Large Language Models (LLMs), such as OpenAI GPT (via the Ada model), generating high-quality synopses for thousands of movies. All text, whether raw or LLM-augmented, is embedded using configurable dense encoders, producing multiple ready-to-use sets (OpenAI Ada, LLaMA-2, Sentence-T5).

Furthermore, the pipeline in **VILLA-MMBENCH** supports interchangeable **early**-, **mid**-, and **late**-fusion operators (concatenation, PCA, CCA, and rank-aggregation), and exposes a variety of backbone recommenders (MF, VAECF, VBPR, AMR, VMF) for ablation studies. All experimental parameters—including dataset splits, modality variants, fusion strategy, and LLM type—are declaratively specified via a single YAML file for transparent, versioned experimentation. Evaluation is comprehensive, covering not only accuracy (Recall, nDCG), but also beyond-accuracy axes: cold-start rate, coverage, novelty, diversity, and fairness, supporting rigorous, multi-metric benchmarking.

Experiments demonstrate that LLM-based text augmentation and dense embedding extraction directly benefit cold-start and coverage performance, especially when strong textual representations are fused with audio-visual descriptors. Systematic benchmarking reveals which embedding and fusion combinations are universal (strong across models) versus backbone- or metric-specific. Overall, the open-source code, embeddings, and configuration templates make it a robust foundation for reproducible, extensible, and fair comparison in multimodal recommender systems, and offer a clear step forward toward principled integration of generative AI in large-scale movie recommendation. All resources are publicly available at <https://recsys-lab.github.io/VILLA-MMBench>.

1 Introduction

Recommending long-form video content remains a challenging task, despite recent advances in computer vision, audio processing, and large language models (LLMs). Movies and series deliver rich visual, auditory, and textual cues that need to be appropriately aligned and thereby integrated into a coherent representation before relevant recommendations can be produced. Traditional collaborative filtering completely ignores item content, while many multimodal recommender systems rely on a single fusion strategy (typically simple feature concatenation) and offer limited transparency or reproducibility. In addition, video-oriented datasets are difficult to share because of copyright restrictions, and widely used benchmarks such as MovieLens and MMTF-14K provide only raw features or partial multimodal alignment [1, 2, 3].

The motivation for this work arises from several persistent obstacles:

- ▣ **Scarcity of shareable content.** Full-length, high-quality movies span hours and are subject to strict copyright restrictions, severely limiting the availability of large-scale, publicly shareable datasets for reproducible research [1, 4, 5].
- ⌚ **Temporal complexity.** Videos are inherently temporal, consisting of thousands of frames and variable-length audio, requiring summarization into compact representations that preserve semantics, narrative, and affect [6, 7, 4, 8]. Yet, temporal aggregation remains underexplored in multimodal fusion pipelines.
- ▣ **Fusion strategy uncertainty.** There is no consensus on optimal *fusion* strategies, as highlighted in recent surveys and benchmarks [9, 6, 4]. Pipelines often default to early-fusion (feature concatenation, as in MMRec-LLM [10], or Ducho [11]), mid-fusion with learned projections (PCA, CCA [6]), often with less attention to system-level fusion. The reproducibility and interpretability of these choices remain open issues.
- ⌚ **The rise of LLM-driven augmentation.** Recent LLMs—such as GPT-4, LLaMA, and LVLMs—have dramatically improved the generation and embedding of textual side information [12, 10, 13]. LLMs can synthesize fluent synopses for items with sparse or missing metadata (addressing the long-tail problem in MovieLens [2, 10]), and their embeddings encode broad world knowledge [12]. However, principled strategies for fusing LLM-generated signals with audio-visual descriptors and for evaluating their impact on user-facing and beyond-accuracy metrics are still lacking [10, 6].

Recent research has nevertheless opened two significant opportunities. First, compact visual and audio embeddings can be extracted from trailers via pre-trained convolutional and audio models, enabling efficient content representation. Second, LLMs can fill metadata gaps, thereby enriching textual features. Yet, existing benchmarks either focus on general multimodal frameworks without LLM integration (e.g., Ducho, MMRec) or propose LLM-augmented synopses without a unified evaluation pipeline (§1.1). Consequently, the community still lacks an open benchmark that simultaneously:

1. integrates dense audio, visual, and LLM-generated textual descriptors,
2. exposes interchangeable early-, mid-, and late-fusion operators,
3. supports diverse recommendation backbones, and
4. reports a comprehensive suite of accuracy and beyond-accuracy metrics.

Contributions. This paper introduces *ViLLA-MMBench*, a unified benchmark suite for *LLM-augmented multimodal movie recommendation*. In contrast to earlier prototypes, we provide a complete Python package that can be installed via pip, configured through a YAML file and executed on local machines or cloud servers. Our contributions are fourfold:

- **Unified multimodal pipeline.** ViLLA-MMBench aligns audio, visual and textual embeddings for MovieLens-1M and MMTF-14K, augments missing synopses with a variety of LLMs, and supplies ready-to-use dense text embeddings produced by OpenAI-Ada, Sentence-T5 and LLaMA-2. This results in a coherent tri-modal representation for roughly 1000 movies after modality filtering.
- **Configurable fusion strategies.** The toolkit supports early-fusion methods (concatenation, PCA, CCA), mid-fusion (projected representations), and late-fusion (ensemble ranking). Because each modality is loaded through a dedicated module, new embeddings or fusion techniques can be incorporated with minimal engineering effort.
- **Diverse recommendation backbones and beyond-accuracy metrics.** We implement matrix factorization, variational autoencoder collaborative filtering, and content-aware models such as VBPR, VMF and AMR, and we expose a simple interface to add more algorithms. A grid-search module performs GPU-aware hyper-parameter optimisation and evaluates models on an extensive suite of metrics, including Recall, nDCG, coverage, cold-start rate, novelty, intra-list diversity, and calibration bias.
- **Reproducibility and extensibility.** All experimental parameters (dataset, split strategy, fusion operator, LLM choice, modality variant) are specified in a YAML configuration file. The codebase contains modular loaders for data and embeddings, model training and evaluation, and utility functions, thereby facilitating the integration of new datasets or modalities without altering the core pipeline.

Table 1: Multimodal movie/video recommendation systems and resources. **Video Type:** Tr = Trailer, μ V = Micro-video. **Modalities:** icons for Visual (☒), Audio (🎵), Text (Ⓐ). **Fusion:** both timing stage and technique. **LLM:** ✓= LLM augmentation. **RS Model:** core recommender family; Model Fusion indicates late fusion of models or rank aggregation.

Class	Type	System	Modalities		Fusion		RS Model Family Model Fusion	Key Insight	Link
			☒	Ⓐ	Stage	Type			
General-Purpose Multi-Modal Recommenders									
-	Ducho 2.0 [11]	☒ Ⓐ	☒ Ⓐ	✓, type	–	Early	Basic (Concat, sum, mean etc.)	VBPR, BM3, FREEDOM –	Turns raw V/A/T into embeddings; user plugs any RS model. [code]
	Ducho×Elliot [9]	☒ Ⓐ	☒ Ⓐ	✓	–	Early	Basic (Concat, sum, mean etc.)	12 models –	Combines features in Ducho with evaluation in Elliot for reproducible benchmarks. [code]
	Rec-GPT4V [12]	☒ Ⓐ	☒ Ⓐ	✓ LVLM	Late	LLM-based (prompt level)	LLM –	LVLM “see-and-chat” recommendations ¹ —no additional training needed. –	–
	MMSSL [7]	☒ Ⓐ	☒ Ⓐ	✓	–	Hybrid	Attention	Deep (GNN-based CF) –	Claims SSL “alleviates label sparsity” and “integrates unlabeled data” at the cost of heavier training. Training complexity is not directly measured. [code]
	MMRec [6]	☒ Ⓐ	☒ Ⓐ	✓	–	Early	Concat, PCA, Attn	CF, Deep (NGCF, VBPR, etc.); –	Toolbox unifying many multimodal RS; ideal for surveys and ablations. [code]
	MMRec-LLM [10]	☒ Ⓐ	☒ Ⓐ	✓ LLM	Early	Basic (Concat)	CF, Deep –	Shows that LLM-generated synopses (fusing visual + text cues) yield improved recommendation performance. –	–
Video Recommender	μ V	MicroLens [4]	☒ Ⓐ	✓	–	Early	Basic (Concat)	– –	Billion-interaction micro-video corpus—great for deep seq-RS research. [data]
	Tr	MMTF-14K [1]	☒ Ⓐ	✓	–	–	–	Late (Rank Aggregation)	Staple trailer-based multimodal dataset aligned with MovieLens. [data]
	Tr	Ours	☒ Ⓐ	✓ LLM	Hybrid	Early (Concat, PCA, CCA)	VBPR, AMR, VMF Late (Rank Aggregation)	Fully reproducible, plug-and-play platform for LLM-enriched, multimodal recommendations aligned with MovieLens. Multiple fusion techniques are employed both at the feature level and system level.	[code]

¹ Rec-GPT4V: “See-and-chat” refers to prompting an LVLM with both image (video frames, posters) and text (user history, metadata, title, etc.), and receiving a text-based answer (“chat”)—here, a recommendation list or justification.

² Rec-GPT4V: **Late fusion** (“prompt-level” after all modalities are presented).

³ MMSSL: **Hybrid fusion:** Features are fused at both early (representation) and late (joint learning) stages. It uses cross-modal self-attention and MM graph attention.

⁴ MMRec-LLM: Uses GPT-3.5 to generate “synopsis” (synthetic) text for items, combining image tags + text.

By providing code, documentation and pre-processed embeddings, ViLLA-MMBENCH serves as a plug-and-play platform for systematic research into LLM-augmented multimodal recommendation, addressing limitations in existing work and enabling controlled ablation studies across fusion operators, modalities and recommendation models.

- **Unified multimodal pipeline:** ViLLA-MMBENCH natively ingests and aligns audio, visual, and textual embeddings, integrating metadata and dense content features from MovieLens 1M [2], MMTF-14K [1], and in-house LLM-augmented synopses [10] produced in this work.
- **Configurable fusion strategies:** The toolkit provides interchangeable early (concatenation, PCA, CCA), mid, and late/system-level (ensemble ranking) fusion methods, enabling controlled ablation studies and benchmarking to advance research in this area with respect to recent advances [6, 10, 11, 9].
- **LLM-based augmentation:** We auto-generate rich, human-readable synopses for MovieLens movies lacking metadata, and provide multiple ready-to-use embedding sets (Sentence-T5, LLaMA-2, OpenAI Ada, etc.)—each on both raw and LLM-augmented synopses.
- **Systematic evaluation:** The framework benchmarks state-of-the-art recommenders—including MF, VAECF, VBPR, AMR, hybrids, and recent GNNs—under GPU-aware hyperparameter grids, reporting more than ten metrics covering accuracy (Recall, nDCG), coverage, cold-start, fairness, novelty, and diversity.
- **Reproducibility and extensibility:** Every configuration, split, and metric is declarative and versioned; new

modalities or models can be incorporated via drop-in loaders or subclassing, following best practices for transparent, reproducible research [9, 4, 10].

By making all code and resources publicly available², **ViLLA-MMBENCH** provides a robust, plug-and-play platform for systematic research on multimodal, LLM-enriched video recommendation. Our toolkit lays the groundwork for reproducible, extensible benchmarking and fair comparison across fusion operators, model classes, and evaluation criteria—addressing the key open questions identified in prior surveys and recent benchmarks [6, 10, 12, 9], and highlighted in Table 1 (§1.1).

In summary, this work contributes the first unified, fully reproducible resource for exploring how LLM-augmented text, audio, and vision interact in large-scale movie recommendation—a crucial step toward the next generation of multimodal recommender systems.

1.1 Related Multimodal Frameworks and Gaps

Over the past decade, a variety of multimodal recommender systems have emerged, each supporting different modalities and fusion strategies at varying levels of scale and reproducibility. Table 1 organizes these systems and resources by their *modality support*, fusion strategies, use of large language models (LLMs), system-level fusion, and recommendation model families.

General-purpose frameworks. Systems such as Ducho 2.0 [11], and MMRec [6] primarily leverage early fusion through concatenation of audio, visual, and textual features, positioning themselves as flexible frameworks or toolkits for rapid benchmarking and ablation studies. More specialized systems like Rec-GPT4V [12] use LLM-based visual-to-text capabilities, enabling “see-and-chat” style recommendations without retraining, whereas MMRec-LLM [10] integrates synopsis generation via LLMs to significantly enhance side-information quality and recommendation accuracy. All of them accept video, audio, and text, yet they differ markedly in how they marry these signals: early concatenation is still the dominant strategy (Cornac, Ducho), but attention mechanisms (MMSSL [7]) and configurable PCA/Attn hybrids (MMRec) appear when scalability or interpretability becomes an issue.

Video-centered resources. Datasets such as MicroLens [4], MMTF-14K [1] have become indispensable for developing scalable and realistic benchmarks, supporting research on sequence modelling, temporal aggregation, and modality alignment in video recommendation. However, they typically offer only raw interaction logs or extracted features and lack built-in pipelines for systematic fusion, evaluation, or LLM-augmented descriptors.

Gaps and limitations. Despite these advances, several gaps remain: most systems still default to basic early fusion (e.g., concatenation), and only a handful support hybrid or late/system-level fusion through modular pipelines [6, 7, 10]. The integration of LLM-based features is not yet standardized; benchmarks often lack fair, flexible support for modality selection, multi-fusion strategies, and beyond-accuracy evaluation criteria (e.g., fairness, diversity, cold-start, coverage) [4, 6, 9, 10, 8]. Temporal aspects—so critical in video—are often only superficially addressed or handled outside the main fusion framework.

ViLLA-MMBENCH. To address these limitations, **ViLLA-MMBENCH** provides the first open-source, fully reproducible pipeline for audio-visual-textual video recommendation with native LLM support. Compared to prior work, our proposed systems puts forward the following novel steps: (i) unifying feature extraction and LLM-driven augmentation for all MovieLens/MMTF-14K items; (ii) offering fully configurable early, mid, and late fusion operators; (iii) exposing a plug-and-play layer for integrating new models or evaluation metrics; and (iv) supporting comprehensive benchmarking across accuracy and beyond-accuracy axes. Our pipeline is designed for extensibility, declarative experimentation, and fair, apples-to-apples comparison—closing the reproducibility, flexibility, and LLM-integration gaps identified in Table 1.

²<https://recsys-lab.github.io/ViLLA-MMBench>

Table 2: Deterministic fusion operators used in this work.

Tag	Operator $f(\mathbf{e}_i^{(\text{aud})}, \mathbf{e}_i^{(\text{vis})}, \mathbf{e}_i^{(\text{txt})})$	Output Dim. d_f
concat	$\mathbf{e}_i = [\mathbf{e}_i^{(\text{aud})}; \mathbf{e}_i^{(\text{vis})}; \mathbf{e}_i^{(\text{txt})}]$	$d_{\text{aud}} + d_{\text{vis}} + d_{\text{txt}}$
pca_ρ	$\tilde{\mathbf{e}}_i = \mathbf{P}^\top \text{zscore}(\mathbf{e}_i^{\text{concat}})$, retain d_ρ s.t. $\sum_{j=1}^{d_\rho} \lambda_j / \sum_j \lambda_j \geq \rho$	d_ρ
cca_k	$\tilde{\mathbf{e}}_i = [\mathbf{W}_1^\top \mathbf{e}_i^{(1)}]_{1:k}, \mathbf{W}_1, \mathbf{W}_2$ maximize $\text{corr}(\mathbf{W}_1^\top \mathbf{e}_i^{(1)}, \mathbf{W}_2^\top \mathbf{e}_i^{(2)})$	k

2 Technical Background

We organize the technical background and related models into three principal categories: (i) *interaction-only collaborative filtering*, (ii) collaborative filtering models that incorporate side information, and (iii) system-level fusion strategies that aggregate outputs from independently trained recommenders. Throughout this section, we consistently define all variables and formal notation to ensure clarity and coherence.

2.1 Compared Recommendation Models

Notation and Problem Setting. Let $\mathcal{U} = \{1, \dots, |\mathcal{U}|\}$ denote the set of users, and $\mathcal{I} = \{1, \dots, |\mathcal{I}|\}$ the set of items (e.g., movies). In the implicit feedback scenario, interactions are captured by the set

$$\mathcal{R} = \{(u, i) \mid u \in \mathcal{U}, i \in \mathcal{I}, r_{ui} = 1\}, \quad (1)$$

where $r_{ui} \in \{0, 1\}$ indicates whether user u has interacted positively with item i . Unobserved pairs $(u, i) \notin \mathcal{R}$ may correspond to either uninterest or lack of exposure.

The goal of a recommender system is to learn a scoring function $\hat{r} : \mathcal{U} \times \mathcal{I} \rightarrow \mathbb{R}$ that estimates the affinity of user u for the item i . For each user u , items are ranked in descending order of \hat{r}_{ui} , producing personalized top- N recommendations.

Interaction-Only Baselines (Pure CF). As baselines for our multimodal recommender models and as building blocks for ensemble-based fusion, we employ the following two models

Matrix Factorization (MF) [14]. Matrix Factorization represents each user u and item i by latent vectors $\mathbf{p}_u, \mathbf{q}_i \in \mathbb{R}^d$ in a shared d -dimensional space, with global and individual bias terms:

$$\hat{r}_{ui} = \mu + b_u + b_i + \mathbf{p}_u^\top \mathbf{q}_i, \quad (2)$$

where μ is the global bias, b_u, b_i are user and item biases, respectively. The model parameters are learned by minimizing the regularized squared error:

$$\min_{\mathbf{P}, \mathbf{Q}, b, \mu} \sum_{(u, i) \in \mathcal{R}} (r_{ui} - \hat{r}_{ui})^2 + \lambda (\|\mathbf{p}_u\|_2^2 + \|\mathbf{q}_i\|_2^2), \quad (3)$$

where $\lambda > 0$ is the regularization coefficient.

Variational Autoencoder for Collaborative Filtering (VAECF) [15]. VAECF encodes each user’s interaction vector $\mathbf{x}_u \in \{0, 1\}^{|\mathcal{I}|}$ into a Gaussian latent code $q_\phi(\mathbf{z}_u | \mathbf{x}_u)$ and reconstructs it via a decoder $p_\theta(\mathbf{x}_u | \mathbf{z}_u)$. Learning maximizes the evidence lower bound (ELBO):

$$\text{ELBO}(\theta, \phi) = \sum_{u \in \mathcal{U}} \left[\mathbb{E}_{q_\phi} [\ln p_\theta(\mathbf{x}_u | \mathbf{z}_u)] - \beta \text{KL}(q_\phi \| p) \right], \quad (4)$$

where β controls the regularization strength and $\text{KL}(\cdot \| \cdot)$ denotes the Kullback–Leibler divergence.

Collaborative Filtering with Side Information. Many recent models incorporate item side information (e.g., text, image, audio) to address cold-start and improve generalization. We distinguish two principal approaches to model textual and multimodal signals:

- (a) **Raw Text Models:** These methods (e.g., HFT [16], CDL [17]) operate directly on raw text using topic models or neural networks to extract interpretable item representations.
- (b) **Dense Embedding Models:** These approaches leverage precomputed dense vectors derived from deep neural encoders, applicable to text, audio, and visual modalities. They enable efficient and flexible integration of rich semantic cues into collaborative filtering.

Given our goal of systematically evaluating the effect of aligned dense embeddings for audio, visual, and textual modalities, we focus on models—VBPR, VMF,³ and AMR—designed for direct embedding input, and previously validated for visual, multimedia, and textual recommendation tasks. This ensures fair and balanced comparison across modalities and models.

Formally, for each item $i \in \mathcal{I}$, let:

- $\mathbf{e}_i^{(\text{txt})} \in \mathbb{R}^{d_{\text{txt}}}$: ℓ_2 -normalized text embedding
- $\mathbf{e}_i^{(\text{vis})} \in \mathbb{R}^{d_{\text{vis}}}$: ℓ_2 -normalized visual embedding
- $\mathbf{e}_i^{(\text{aud})} \in \mathbb{R}^{d_{\text{aud}}}$: ℓ_2 -normalized audio embedding

These can be concatenated as $\mathbf{e}_i = [\mathbf{e}_i^{(\text{txt})} \parallel \mathbf{e}_i^{(\text{vis})} \parallel \mathbf{e}_i^{(\text{aud})}]$.

VBPR [18]. VBPR extends Bayesian Personalized Ranking (BPR) to incorporate visual (or general content) features:

$$\hat{r}_{ui} = \mathbf{p}_u^\top \mathbf{q}_i + \mathbf{w}_u^\top \mathbf{e}_i, \quad (5)$$

where \mathbf{w}_u captures user-specific preferences for side features.

VMF [19]. VMF is a multi-modal extension of MF, projecting the fused side information to the collaborative latent space:

$$\mathbf{q}_i = \mathbf{H} \mathbf{e}_i, \quad \hat{r}_{ui} = \mu + b_u + b_i + \mathbf{p}_u^\top \mathbf{q}_i, \quad (6)$$

with $\mathbf{H} \in \mathbb{R}^{d \times d_e}$ a learned projection matrix.

AMR [20]. AMR uses a gating (attention) network $g(\cdot)$ to assign weights to each modality:

$$\hat{r}_{ui} = \mathbf{p}_u^\top \mathbf{q}_i + g(\mathbf{e}_i^{(\text{txt})}, \mathbf{e}_i^{(\text{vis})}, \mathbf{e}_i^{(\text{aud})}). \quad (7)$$

2.2 Multi-Modal Fusion of Embeddings

For every item $i \in \mathcal{I}$ we pre-compute three *modality-specific* embeddings, all ℓ_2 -normalised:

$$\mathbf{e}_i^{(\text{aud})} \in \mathbb{R}^{d_{\text{aud}}}, \quad \mathbf{e}_i^{(\text{vis})} \in \mathbb{R}^{d_{\text{vis}}}, \quad \mathbf{e}_i^{(\text{txt})} \in \mathbb{R}^{d_{\text{txt}}}.$$

A deterministic operator $f : \mathbb{R}^{d_{\text{aud}}} \times \mathbb{R}^{d_{\text{vis}}} \times \mathbb{R}^{d_{\text{txt}}} \rightarrow \mathbb{R}^{d_f}$ maps the triplet $(\mathbf{e}_i^{(\text{aud})}, \mathbf{e}_i^{(\text{vis})}, \mathbf{e}_i^{(\text{txt})})$ to a single multimodal descriptor $\mathbf{e}_i = f(\mathbf{e}_i^{(\text{aud})}, \mathbf{e}_i^{(\text{vis})}, \mathbf{e}_i^{(\text{txt})})$.

We evaluate three early-fusion rules:

- **Concatenation:** $\mathbf{e}_i = [\mathbf{e}_i^{(\text{aud})}; \mathbf{e}_i^{(\text{vis})}; \mathbf{e}_i^{(\text{txt})}]$, giving dimensionality $d_f = d_{\text{aud}} + d_{\text{vis}} + d_{\text{txt}}$.
- **Principal Component Analysis (PCA):** form the concatenated vector above, standardise it (z-score), then project onto the first d_ρ principal components such that the retained cumulative variance satisfies $\sum_{j=1}^{d_\rho} \lambda_j / \sum_j \lambda_j \geq \rho$.
- **Canonical Correlation Analysis (CCA):** Split the concatenated vector in two equal halves, learn linear maps that maximise the correlation between the projected halves, and keep the first k canonical dimensions.

The resulting vector \mathbf{e}_i is passed to the downstream recommender either as an **ImageModality** or a **FeatureModality** (Cornac API), depending on the model. The three operators used in this work are summarised in Table 2.

³Due to the extensive experiments and the superior performance of VBPR and AMR, for space limitations, we focus our report on these two models; complete results—including VMF and concatenation-based fusion which were omitted here (see §2.2)—are available on our GitHub.

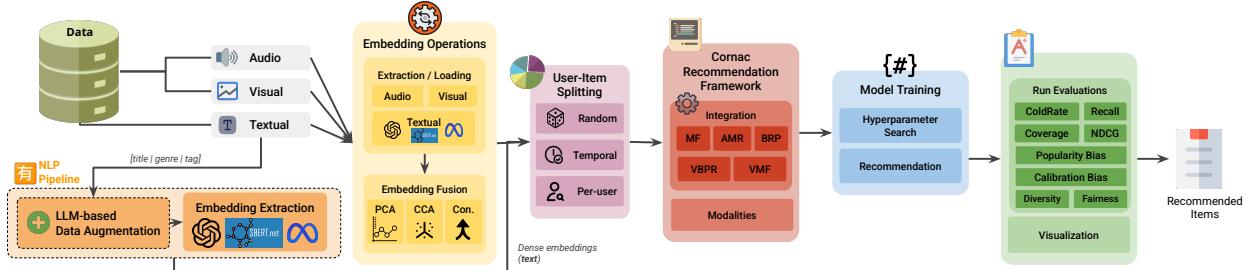


Figure 1: The architecture of the proposed toolkit for movie recommendation.

2.3 System-Level Fusion of Recommender Outputs

Assume M independently trained recommenders indexed by $\mathcal{S} = \{1, \dots, M\}$. For user u and system $m \in \mathcal{S}$, let $L_u^{(m)} = (i_{u,1}^{(m)}, i_{u,2}^{(m)}, \dots)$ be its ranked list. The aim is to merge $\{L_u^{(m)}\}_{m \in \mathcal{S}}$ into a single meta-ranking $F_u = (i_{u,1}, i_{u,2}, \dots)$. We apply four classical, parameter-free aggregation rules:

Borda count. Each system assigns $s_{u,i}^{(m)} = |\mathcal{I}| - \text{rank}_{L_u^{(m)}}(i) + 1$; fused scores are $s_{u,i} = \sum_m s_{u,i}^{(m)}$.

Weighted Borda. Extends the above with weights w_m s.t. $\sum_m w_m = 1$: $s_{u,i} = \sum_m w_m s_{u,i}^{(m)}$.

Average rank. Computes $\bar{r}_{u,i} = M^{-1} \sum_m \text{rank}_{L_u^{(m)}}(i)$ and sorts by ascending $\bar{r}_{u,i}$.

Reciprocal-rank fusion (RRF) [21]. Uses $s_{u,i} = \sum_m (k + \text{rank}_{L_u^{(m)}}(i))^{-1}$ with $k = 60$ (fixed).

The meta-rankings F_u are evaluated using the same top- k accuracy metrics (Recall@10, nDCG@10, Hit Rate@10) and beyond-accuracy metrics (catalogue coverage, cold-start rate) as those used for individual recommenders.

3 ViLLA-MMBench Design and Data Pipeline

In this section, we provide a detailed technical description of the ViLLA-MMBench implementation, covering the system overview (§3.1), configuration and customization options (§3.2), and the suite of evaluation metrics (§3.4). Each subsection includes comprehensive explanations and implementation details.

3.1 System Overview

Figure 1 illustrates the architecture of ViLLA-MMBench, which is divided into four stages: data preparation, textual enrichment and embedding, multimodal alignment and fusion, and training and evaluation. After configuring the framework via the `config.yml` file, the entire data preparation and training pipeline can be executed sequentially by running the `main.py` script, which orchestrates all necessary procedures based on the specified settings.

① Data preparation and ingestion. The framework loads datasets through a uniform pandas interface. The `prepareML` function downloads and reads the MovieLens dataset (100K or 1M variants, based on the given configuration), applies k-core filtering (if set), and performs train/test splitting according to the selected strategy (`random`, `temporal`, or `per_user`). The `prepareModalities` function loads and preprocesses textual data from our in-house dataset, as well as visual and audio embeddings from the MMTF-14K dataset. The variants of these modalities to be loaded are also adjustable from the configuration file. While we provide loaders for MovieLens-1M and MMTF-14K, any dataset with user-item-rating triples can be ingested by implementing a similar loader. Any contribution for adding other modalities or datasets requires implementing simple loader functions by extending the modular structure in the `data` directory of the framework, similar to the existing `loadText`, `loadAudio`, or `loadVisual` functions within the respective `text.py`, `audio.py`, or `visual.py` files.

- ❷ **Textual enrichment and embedding.** For each item, a textual description is created by concatenating the title, genres, and tags or by prompting an LLM to produce a 100–150-word synopsis. The prompt and output are logged for transparency. The resulting text is embedded using the specified model (OpenAI-Ada, Sentence-T5, LLaMA-2), yielding a dense vector. Since the embedding code resides in `villa/mmbench/data/text.py`, adding new LLMs or embedding models only requires registering a function in this module.
- ❸ **Multimodal alignment and fusion.** Audio, visual, and textual embeddings are aligned via item identifiers. Three deterministic early-fusion operators are supplied (`CONCAT`, PCA_ρ , or CCA_k), representing concatenation, PCA (retaining a fixed proportion of variance) and CCA (projecting halves to maximise correlation). Mid- and late-fusion strategies are available via configuration. The `prepareModalities` function merges modalities, handles missing values and wraps the resulting features into the appropriate Cornac objects for downstream recommendation.
- ❹ **Training, evaluation and logging.** The `gridSearch` module in the framework’s `grid.py` file performs hyper-parameter optimisation for the chosen model class, optionally using GPU resources. It evaluates each candidate on recall and nDCG and records the best configuration. Finally, the `generateLists` function in `processes.py` trains the selected model on the full training data and produces recommendation lists for each test user, computing metrics such as recall, nDCG, coverage, cold-start rate, novelty, intra-list diversity, popularity bias, and fairness. Results are saved as CSV files (by default in the `outputs` folder) for subsequent analysis.

3.2 Configuration and Customization

Experiments are specified entirely through a YAML file (`config.yml`). A rich suite of parameters fully specifies an experiment.

- **General:** `root_path`
- **Dataset & Split:**
 - **MovieLens:** `100k` | `1m`
 - **Split:** `random` | `temporal` | `per_user`
 - **Cold-start:** `k_core`, `simulate_cold_start`
- **Modality:**
 - **LLM:** `openai` | `st` | `llama`
 - **Augmentation:** `true` | `false`
 - **Audio-variant:** `blf` | `i_vec`
 - **Visual-variant:** `cnn` | `avf`
 - **Fusion:** `concat` | pca_ρ | cca_k (§2.2)
- **Experiment:** `seed`, `epochs`, `use_gpu`, `fast_prototype`, `parallel_hpo`, etc.
- **Recommendation and Experiments:**
 - **Model:** `cf` | `vbpr` | `amr` | `vmf`
 - **Runtime:** `seed`, `epochs`, `gpu_id`, `fast_prototype`

Default values reproduce all results reported in §4 and more information is provided below:

Dataset and split: Select a predefined `movielens` version, choose a splitting mode (`random`, `temporal`, `per_user`), and set the test ratio and `k_core`. Researchers can thus replicate experiments across different scenarios or apply the framework to new datasets.

Modality variants: Choose audio embeddings (`blf` or `i_vec`), visual features (`cnn` or `avf`) and decide whether to use LLM-generated text. Adding additional modalities, such as user demographics or interaction contexts, is straightforward through the data modules.

Fusion operator: Specify `concat`, `pca_ρ`, or `cca_k` for early fusion, or enable mid- or late-fusion. More sophisticated operators (e.g., attention-based fusion) can be integrated as future work. Currently, our framework supports all these variants. Users can also specify the number of principal components for PCA and the number of canonical variables for CCA.

Model and hyper-parameters: Select the recommendation backbone (`cf`, `vbpr`, `amr`, `vmaf`, `vaecf`) and optionally provide model-specific hyper-parameters. The modular design allows researchers to introduce graph-based or transformer-based recommenders with minimal changes.

Runtime options: Toggle fast prototypes (a single training epoch) for quick testing, specify a GPU for hyper-parameter search and set random seeds to ensure reproducibility. Logging options can be extended to record additional metadata or integrate with experiment tracking tools.

This declarative configuration approach reduces boilerplate code, ensures that experiments are reproducible, and eases the extension of the benchmark to new domains.

3.3 Textual Data Enrichment and Embedding.

Given an item i described by title, genre list, and user tags, we generate a canonical text view $\mathcal{T}(i)$ in one of two mutually exclusive modes:

(1) **No Augmentation (NA):** After lower-casing and removing structural delimiters, we concatenate the following:

$$\mathcal{T}_{\text{NA}}(i) = \text{title}_i + " " + \text{genres}_i["\text{---}" \mapsto "] + " " + (\text{tags}_i \text{ space-joined}).$$

(2) **LLM-based Augmentation (A):** If synopses are missing, sparse, or inconsistent, a large language model (LLM) is prompted once per item as follows:

Synopsis Generation Template

Role: You are a helpful assistant.
Task: Write a vivid, engaging **100–150-word** synopsis for a movie or artist.
Inputs:

- **Title:** *[Movie/Artist Title]*
- **Genre List:** *[List of genres, e.g., drama, mystery, thriller]*
- **Tags:** *[Comma-separated, free-form tags, e.g., coming-of-age, family, 1980s]*

Passing title_i , genres_i , and tags_i as the user message yields the enriched synopsis $\mathcal{T}_{\text{A}}(i)$, which is stored verbatim.

Regardless of the mode, the resulting text is embedded using a configurable model (*OpenAI-Ada*, *Sentence-T5*, or fine-tuned *LLaMA-2*):

$$\mathbf{e}_i^{(\text{txt})} = \Phi_{\text{mdl}}(\mathcal{T}_{\text{NA/A}}(i)) \in \mathbb{R}^{d_{\text{txt}}}.$$

The process—augmentation, tokenisation, batching, and embedding—is fully automated, and documented in `data_augment_llm.ipynb`.

Multimodal Alignment and Fusion. Audio, visual, and text keys are intersected to ensure every item has complete features. Three deterministic operators are provided:

$$\begin{aligned} \text{CONCAT} : \quad & \mathbf{e}_i = [\mathbf{e}_i^{(\text{aud})}; \mathbf{e}_i^{(\text{vis})}; \mathbf{e}_i^{(\text{txt})}] \\ \text{PCA}_\rho : \quad & \mathbf{e}_i = \mathbf{P}_\rho^\top \text{zscore}(\mathbf{e}_i^{\text{CONCAT}}) \\ \text{CCA}_k : \quad & \mathbf{e}_i = [\mathbf{W}_1^\top \mathbf{e}_i^{(1)}]_{1:k} \end{aligned}$$

Fused vectors are transparently wrapped for use with Cornac.

Data Splitting, Training, and Evaluation. Splitting strategies (RANDOM, TEMPORAL, PER_USER) preserve chronology in the test fold. Each recommender is trained using fused features and hyperparameters, with performance tracked across a suite of metrics: recall, nDCG, cold-rate, coverage, novelty, diversity, and calibration bias (see below). Ensemble methods (Borda, RRF, etc.) can be enabled post-hoc without retraining.

Table 3: Final dataset characteristics after merging with MOVIELENS-1M and MMTF-4K. Although approximately 3,000 movies were initially augmented, the final item count is lower due to overlap with these datasets.

Metric	Value
Total Interactions ($ R $)	632,397
Number of Users ($ U $)	6,040
Number of Items ($ I $)	992
Avg. Ratings per User ($ R / U $)	104.70
Avg. Ratings per Item ($ R / I $)	637.50
Sparsity ($ R /(U \cdot I)$)	0.1055%

3.4 Evaluauion in ViLLA-MMBench

ViLLA-MMBench evaluates recommendation quality along multiple dimensions. Besides Recall@ K and nDCG@ K , the framework computes:

- **Cold-start rate**—the proportion of users or items in the top- K that were unseen during training.
- **Coverage**—the fraction of the item catalogue that appears in at least one user’s top- K list.
- **Novelty**—the mean negative log-popularity of recommended items, thereby encouraging less popular content.
- **Intra-list diversity**—the mean cosine distance between pairs of recommended items for each user.
- **Calibration bias**—the difference between attribute distributions in recommendations and those in the user’s historical interactions.

These metrics can be extended to cover fairness or serendipity in subsequent work. Full experiments typically take two to six hours on a free Colab GPU; local execution is supported through the Python package to avoid the version conflicts previously noted by reviewers.

3.5 Implementation

ViLLA-MMBench is implemented in **Python 3.10** and is designed to provide flexibility, modularity, and reproducibility in one framework. It supports both CPU and GPU execution scenarios, which can be easily toggled via the configuration file, along with options for CPU parallelization to accelerate training or evaluation. We provide a complete `setup.py`-based installation, making it straightforward to install all dependencies locally. For recommendation tasks, the framework integrates with Cornac [22], offering a robust backend for collaborative filtering and multimodal models. While the codebase is structured to run locally, either directly or through containerized environments, we also offer a dedicated Google Colab implementation that mirrors the full pipeline, allowing users to benefit from Colab’s GPU resources without setup overhead. Additionally, we provide a secondary Colab file that demonstrates how to load and call the local Python modules and GitHub repository of the framework directly from within a Google Colab environment, further simplifying reproducibility and experimentation.

4 Evaluation and Benchmarking

In this section, we present the experimental results obtained using the proposed framework, benchmarking movie recommendation performance. Our results provide a comprehensive view of the impact of incorporating both visual and audio features from movie trailers, together with LLM-augmented textual data, on downstream movie recommendation tasks. The experiments are designed to address the following research questions.

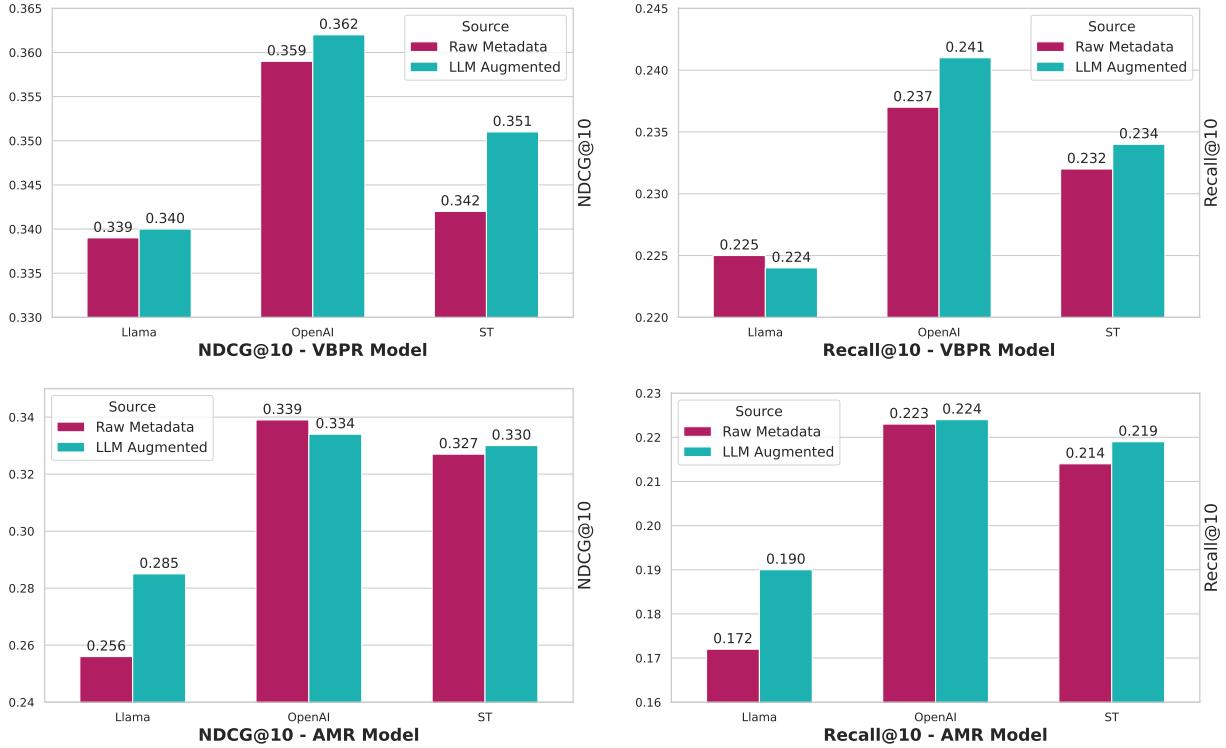


Figure 2: Effect of LLM-based text augmentation on NDCG@10 and Recall@10 for unimodal textual sources.

RQ1. Impact of text augmentation with LLMs.

What is the impact of **text augmentation** with Large Language Models (LLMs) on video recommendation performance, particularly measured by Recall@10 and NDCG@10, across selected recommendation models (AMR and VBPR)?

RQ2. Modality impact.

2.a. Which individual modalities (text, vision, audio) or multimodal combinations are consistently beneficial—or detrimental—for overall performance, as measured by the AUC metrics corresponding to NDCG@10 versus ColdRate@10, and Coverage@10?

2.b. Universality of features.

Do any features behave in a “universal” manner consistently benefiting or harming both AMR and VBPR backbones, irrespective of the evaluation metric used?

2.c. Projection schemes comparison (CCA vs PCA).

Between the two experimented projection (dimensionality-reduction) schemes—95% PCA and 40-dimensional CCA—does one clearly dominate the other across evaluation metrics and recommendation backbones, or should the projection choice remain a tunable hyperparameter?

RQ3. Model-based fusion impact.

What is the impact of **model-based fusion approaches** when combining collaborative filtering with multimodal models?

4.1 RQ1. (LLM-based text augmentation Impact)

Textual metadata for video items is often sparse or noisy, limiting the effectiveness of text-aware recommenders. Augmenting this metadata with Large Language Models (LLMs) can enrich semantic representations, potentially

improving recommendation quality despite risks of added noise or fairness issues. We present an initial analysis within our framework, encoding augmented text using dense embeddings (LLaMA, OpenAI, & SentenceTransformer).

A key question we address is *how sentence-level embeddings interact differently with the two recommender backbones?* Results summarizing our observations are shown in Fig. 2 and detailed below:

- **SentenceTransformer (ST)** embeddings consistently yield modest but reliable improvements for both backbones and metrics: Recall increases by **+0.9%** on VBPR ($0.232 \rightarrow 0.234$) and by **+2.3%** on AMR ($0.214 \rightarrow 0.219$); NDCG rises by **+2.6%** on VBPR ($0.342 \rightarrow 0.351$) and by **+0.9%** on AMR ($0.327 \rightarrow 0.330$). This indicates that ST effectively captures general semantic cues beneficial across models without specific tuning.
- **OpenAI** embeddings exhibit conservative improvements for VBPR—Recall **+1.7%**, NDCG **+0.8%**—but produce mixed results with AMR: minimal Recall improvement (**+0.4%**) accompanied by a slight NDCG decrease (**-1.5%**, $0.339 \rightarrow 0.334$). Hence, OpenAI embeddings remain a reliable choice for VBPR but might slightly degrade re-ranking effectiveness by AMR.
- **LLaMA** embeddings show strong model-specific effects: neutral on VBPR (Recall **-0.4%**, NDCG **+0.3%**), yet highly beneficial for AMR with pronounced gains in Recall (**+10.5%**, $0.172 \rightarrow 0.190$) and NDCG (**+11.3%**, $0.256 \rightarrow 0.285$). This suggests that embedding by LLaMA structure aligns particularly well with AMR’s item-aware layers, providing limited added value to VBPR.

In summary, ST embeddings serve as a robust, universal baseline; OpenAI embeddings provide consistent performance for VBPR but might require further tuning for AMR; LLaMA embeddings offer significant benefits for AMR, yet minimal advantage or slight detriment for VBPR. Summaries of these insights can be found in top part of Table 4.

Table 4: Top: % change for each embedding baseline. Bottom: Recommended configs by KPI/backbone.

Emb.	VB R	VB N	AMR R	AMR N	Avg
ST	+0.9	+2.6	+2.3	+0.9	1.7
OpenAI	+1.7	+0.8	+0.4	-1.5	0.4
Llama	-0.4	+0.3	+10.5	+11.3	5.4
Avg	0.7	1.2	4.4	3.6	3.0

KPI	Recommended Configs (AMR / VBPR)	
Cold-start	Raw OpenAI + CNN + BLF	/ Aug ST (text-only)
Coverage	Raw OpenAI + AVF + i-vec	/ Aug OpenAI + AVF + i-vec

Tips: Strong text emb. matter most; **AVF+i-vec** boost Coverage.

R: Recall, **N:** NDCG, **VB:** VBPR. ST is consistently positive; OpenAI excels on VBPR; Llama excels for AMR.

4.2 RQ2-a. (Modality Impact)

Here we aim to analyze the impact of multi-modal features across two recommender backbones, **AMR** and **VBPR**, focusing on cold start performance (NDCG-ColdRate@10 AUC) and catalog coverage (NDCG-Coverage@10 AUC). As these metrics are intended to reflect aspects beyond pure accuracy, we ensure our analysis is grounded in systems that already perform well in terms of ranking quality—measured by NDCG. Therefore, we base the following discussion on AUC values derived from the NDCG-ColdRate and NDCG-Coverage curves (after min-max normalizing the values per model VBPR and AMR). The raw values are reported in Figure 3. For brevity, we refer to the AUC of NDCG-ColdRate@10 as *ColdRate@10 AUC*, and the AUC of NDCG-Coverage@10 simply as *Coverage@10 AUC* in the following discussion.

Note. We use *Raw* for original text and *Aug* for LLM-generated text. Since RQ1 shows the advantage of augmentation, most multimodal combinations use the augmented version for the effort consideration.

4.2.1 AMR Backbone

For **AMR**, distinct modality combinations optimize each KPI differently. For cold-start, the combination of textual (**Raw OpenAI**), visual (**CNN**), and audio (**BLF**) modalities clearly outperforms other configurations, achieving the highest observed ColdRate@10 AUC of **0.8576** (NDCG@10 = 0.314, ColdRate@10 = 0.019). Interestingly, using only audio (**i-vec**) also offers good cold-start performance (AUC **0.8333**, NDCG@10 = 0.356, ColdRate@10 = 0.016), but trails the multimodal **CNN+BLF** combo by approximately 2.4 percentage points. Pure-text configurations, while foundational, remain significantly below multimodal runs in cold-start effectiveness—for example, **Raw OpenAI** alone achieves an AUC of only **0.7330**, demonstrating a substantial multimodal advantage (+0.12).

In terms of catalog coverage, the optimal AMR configuration shifts distinctly. Here, the combination of textual (**Raw OpenAI**), aesthetic visual features (**AVF**), and audio (**i-vec**) using a CCA projection yields the highest AUC (**0.7452**, with NDCG@10 = 0.321 and Coverage@10 = 0.926). Notably, this clearly surpasses pure-text solutions like **Aug ST** (AUC **0.7044**, NDCG@10 = 0.33, Coverage@10 = 0.909), highlighting the importance of multimodality for effectively exploring the long tail issue of the catalog.

4.2.2 VBPR Backbone

For **VBPR**, however, the modality effects differ substantially. Cold-start performance strongly favors textual modalities alone. Specifically, **Augmented SentenceTransformer (Aug ST)** achieves the highest ColdRate@10 AUC (**0.8620**), closely followed by other text-only methods (**Aug OpenAI**, **0.8333**, and **Raw Llama**, **0.8175**). Incorporating visual or audio modalities significantly deteriorates cold-start effectiveness, exemplified by the visual modality alone (**CNN**), which yields an AUC of only **0.6671** (NDCG@10 = 0.325, ColdRate@10 = 0.025). Thus, cold-start effectiveness by VBPR hinges exclusively on textual embeddings.

For catalog coverage, however, VBPR mirrors AMR in preferring multimodal approaches. The best-performing VBPR coverage configuration again includes textual (**Aug OpenAI**), visual (**AVF**), and audio (**i-vec**) modalities, achieving the highest Coverage@10 AUC of **0.7051**. The NDCG@10 and Coverage@10 values by this configuration are 0.336 and 0.96, respectively. Pure text modalities remain substantially behind (AUC \leq **0.58**), underscoring the critical role of multimodal fusion to maximize catalog exploration.

4.3 RQ2-b. (Universality)

We examined whether specific features consistently enhance or impair performance across both backbones and metrics as shown in Table 4:

Universally beneficial: Strong textual embeddings, particularly from large language models like **OpenAI** or **Aug ST**, consistently form the foundation of high-performing configurations across both backbones and metrics.

Nearly universally beneficial (for Coverage): The fusion of visual aesthetics (**AVF**) with audio i-vectors (**i-vec**) consistently enhances catalog coverage performance, as demonstrated by their presence in top-ranking configurations for both AMR (AUC = **0.7452**) and VBPR (AUC = **0.7051**).

Model-specific modality effects: The combination of visual CNN features (**CNN**) with audio block-level features (**BLF**) significantly benefits AMR cold-start performance but consistently harms VBPR cold-start performance. This highlights that certain modality combinations should be specifically tailored to the recommender backbone used.

4.4 RQ2-c. (Projection Schemes)

We analyzed two projection schemes—Principal Component Analysis (**PCA-95**) and Canonical Correlation Analysis (**CCA-40**)—to understand their relative benefits for multimodal recommender performance across AMR and VBPR backbones, considering AUC of both ColdRate@10 and Coverage@10 metrics.

4.4.1 AMR Backbone

For the AMR backbone, CCA substantially outperformed PCA in both evaluation metrics, clearly dominating the performance landscape. In the cold-start scenario, the top-performing CCA-based configuration (**Raw OpenAI + CNN**

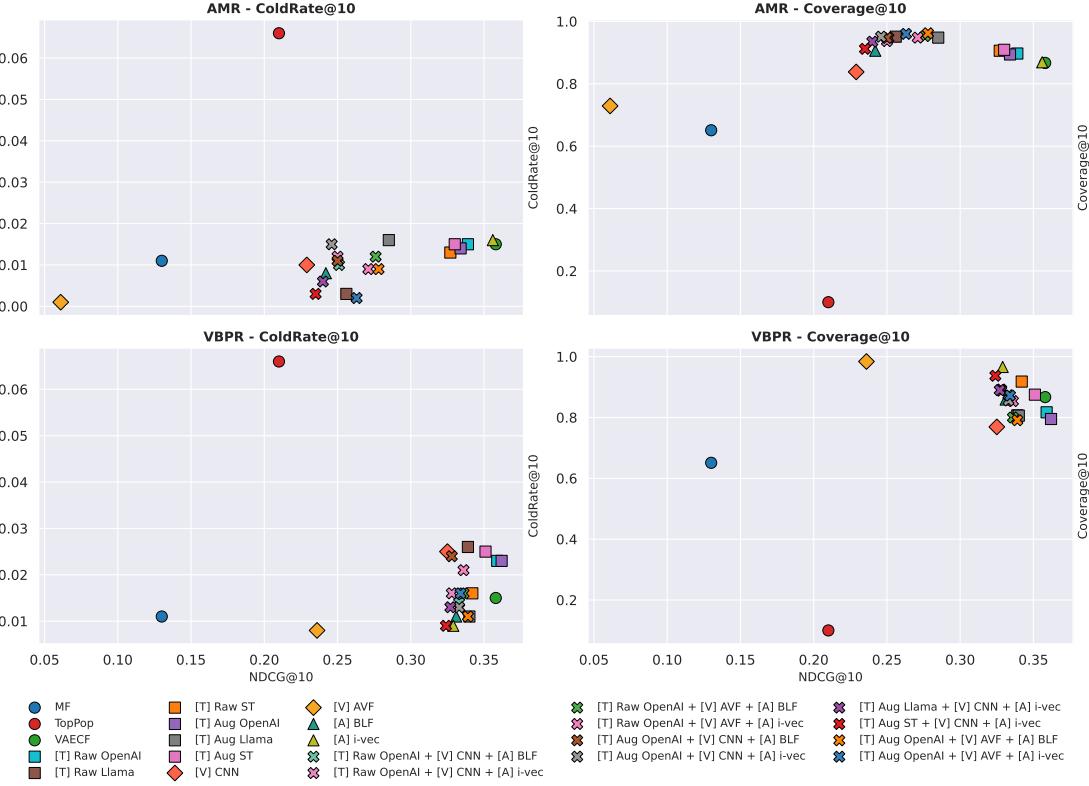


Figure 3: Evaluation results across various model and modality combinations. Multimodal data are fused using Principal Component Analysis (PCA) to combine modalities. We provide this visualization for clarity. However, the results using Canonical Correlation Analysis (CCA) are also available at the provided link.

+ **BLF**) achieved an exceptionally high AUC of **0.8576**, vastly exceeding the best PCA-based configuration (**Aug OpenAI + CNN + i-vec**) at only **0.4878**—a remarkable absolute improvement. Such a substantial gap indicates that the cross-modal alignment captured by CCA is essential to effectively handle cold-start challenges posed by AMR.

Regarding Coverage@10, CCA again provided the superior choice, albeit with a smaller but still meaningful margin. The top CCA-based model (**Raw OpenAI + AVF + i-vec**) reached an AUC of **0.7452**, surpassing PCA’s best configuration (**Aug OpenAI + AVF + BLF**) with an AUC of **0.7356**. Although the margin is smaller (+0.0096), the consistent advantage reinforces CCA’s suitability as the default projection method for AMR.

4.4.2 VBPR Backbone

The VBPR backbone, however, exhibited a mixed picture. For the ColdRate@10 metric, PCA showed a clear advantage. Specifically, leading configuration in PCA (**Aug OpenAI + CNN + BLF**) attained an AUC of **0.6490**, significantly above the best-performing CCA configuration (**Aug OpenAI + CNN + i-vec**) which only reached **0.3968**. This indicates the ability of PCA to preserve the distinctive modality-specific variance essential for VBPR’s cold-start performance.

In contrast, when assessing Coverage@10, CCA regained dominance. The highest performing configuration under CCA (**Aug OpenAI + AVF + i-vec**) scored **0.7051**, substantially outperforming leading configuration offered by PCA, (**Aug ST + CNN + i-vec**) at **0.5457**. This difference of approximately 16 percentage points emphasizes the strength of CCA in distributing recommendations more broadly across the catalog.

These results provide insightful guidance for future modeling. For AMR-based recommenders, CCA is the preferred projection method, consistently improving cold-start performance and coverage. For VBPR, PCA is optimal when prioritizing cold-start novelty, while CCA is better for catalog coverage or fairness.

Table 5: Comparison of model-based fusion (VAECF + AMR variants). Best per block in **bold**.

Fusion	Model	Rec@10	N@10	HR@10
<i>Audio (AMR Audio)</i>				
RRF	VAECF + AMR Audio	.2521	.3739	.8610
Borda	VAECF + AMR Audio	.2517	.3732	.8615
Avg Rank	VAECF + AMR Audio	.2520	.3738	.8606
<i>Text (AMR Text, OpenAI)</i>				
RRF	VAECF + AMR Text (OpenAI)	.2496	.3666	.8605
Borda	VAECF + AMR Text (OpenAI)	.2496	.3647	.8605
Avg Rank	VAECF + AMR Text (OpenAI)	.2486	.3661	.8593
<i>Text (AMR Text, ST)</i>				
RRF	VAECF + AMR Text (ST)	.2490	.3698	.8564
Borda	VAECF + AMR Text (ST)	.2490	.3682	.8564
Avg Rank	VAECF + AMR Text (ST)	.2486	.3690	.8568
<i>No Fusion</i>				
—	VAECF Only	.2492	.3584	.8581

Rec: Recall, N: NDCG, HR: HitRate, ST: SentenceTransformer.

4.5 RQ3. (Model-based Fusion Impact)

Table 5 summarizes the comparative results. Note that our primary goal here is to combine the best-performing CF model (VACEF) with the best multimodal model to further enhance performance. Rank-based fusion (esp. RRF and Borda) consistently improves performance over single-model VAECF, across both audio and text modalities. Gains are most visible in NDCG and Recall, and Hit Rate. This confirms the value of model-based fusion and rank aggregation methods for leveraging diverse modality-specific signals in video recommendation.

5 Conclusion

The proposed toolkit, **ViLLA-MMBench**, offers a lightweight, reproducible framework that elevates the classic MovieLens benchmark into a comprehensive multi-modal testbed for recommendation research. By combining visual features from trailers, audio embeddings, and LLM-generated text, the toolkit systematically evaluates both individual and fused modalities across a broad spectrum of metrics, including not only accuracy but also beyond-accuracy criteria such as cold-start handling, fairness, novelty, diversity, and catalog coverage.

A distinguishing contribution of **ViLLA-MMBench** is the automated augmentation of sparse or missing item metadata using state-of-the-art Large Language Models (LLMs), specifically OpenAI’s GPT, which enables the generation of high-quality synopses and consistent textual signals for every movie. Multiple dense embedding types—including OpenAI Ada, LLaMA-2, Sentence-T5, CNN, AVF, BLF, and i-vector—are aligned and made available, supporting interchangeable early-, mid-, and late-fusion strategies and facilitating principled ablation studies.

The fully scripted and logged pipeline, driven by declarative YAML configuration, ensures transparency, repeatability, and ease of extension to new modalities, recommendation backbones, or evaluation protocols. With robust support for MovieLens (100K/1M), MMTF-14K, and a custom LLM-augmented review dataset, as well as modular interfaces for integrating additional data sources, **ViLLA-MMBench** provides a solid foundation for rigorous benchmarking and fair comparison in multimodal recommendation.

Empirical results demonstrate clear improvements in cold-start and catalog coverage, particularly in scenarios where LLM-augmented text is fused with audio-visual descriptors. In summary, via making all code, embeddings, and configuration templates openly available, this toolkit aims to foster reproducible, extensible, and responsible research, paving the way for principled integration of generative AI in recommender systems. Future work will explore further modalities, additional domains, and more advanced evaluation criteria, continuing to advance the state of the art in trustworthy, multi-modal recommendations.

References

- [1] Y. Deldjoo, M. G. Constantin, B. Ionescu, M. Schedl, and P. Cremonesi, “Mmtf-14k: a multifaceted movie trailer feature dataset for recommendation and retrieval,” in *Proceedings of the 9th ACM Multimedia Systems Conference*, 2018, pp. 450–455.
- [2] F. M. Harper and J. A. Konstan, “The movielens datasets: History and context,” *Acm transactions on interactive intelligent systems (tiis)*, vol. 5, no. 4, pp. 1–19, 2015.
- [3] Y. Fan, Y. Wang, H. Yu, and B. Liu, “Movie recommendation based on visual features of trailers,” in *Innovative Mobile and Internet Services in Ubiquitous Computing: Proceedings of the 11th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2017)*. Springer, 2018, pp. 242–253.
- [4] Y. Ni, Y. Cheng, X. Liu, J. Fu, Y. Li, X. He, Y. Zhang, and F. Yuan, “A content-driven micro-video recommendation dataset at scale,” *arXiv preprint arXiv:2309.15379*, 2023.
- [5] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, “Youtube-8m: A large-scale video classification benchmark,” *arXiv preprint arXiv:1609.08675*, 2016.
- [6] X. Zhou, “Mmrec: Simplifying multimodal recommendation,” in *Proceedings of the 5th ACM International Conference on Multimedia in Asia Workshops*, 2023, pp. 1–2.
- [7] W. Wei, C. Huang, L. Xia, and C. Zhang, “Multi-modal self-supervised learning for recommendation,” in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 790–800.
- [8] K. Zhou, H. Wang, W. X. Zhao, Y. Zhu, S. Wang, F. Zhang, Z. Wang, and J.-R. Wen, “S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization,” in *Proceedings of the 29th ACM international conference on information & knowledge management*, 2020, pp. 1893–1902.
- [9] M. Attimonelli, D. Danese, A. Di Fazio, D. Malitesta, C. Pomo, and T. Di Noia, “Ducho meets elliot: Large-scale benchmarks for multimodal recommendation,” *arXiv preprint arXiv:2409.15857*, 2024.
- [10] J. Tian, Z. Wang, J. Zhao, and Z. Ding, “Mmrec: Llm based multi-modal recommender system,” in *2024 19th International Workshop on Semantic and Social Media Adaptation & Personalization (SMAP)*. IEEE, 2024, pp. 105–110.
- [11] M. Attimonelli, D. Danese, D. Malitesta, C. Pomo, G. Gassi, and T. Di Noia, “Ducho 2.0: Towards a more up-to-date unified framework for the extraction of multimodal features in recommendation,” in *Companion Proceedings of the ACM on Web Conference 2024*, 2024, pp. 1075–1078.
- [12] Y. Liu, Y. Wang, L. Sun, and P. S. Yu, “Rec-gpt4v: Multimodal recommendation with large vision-language models,” *arXiv preprint arXiv:2402.08670*, 2024.
- [13] W. Wei, X. Ren, J. Tang, Q. Wang, L. Su, S. Cheng, J. Wang, D. Yin, and C. Huang, “Llmrec: Large language models with graph augmentation for recommendation,” in *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, 2024, pp. 806–815.
- [14] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
- [15] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, “Variational autoencoders for collaborative filtering,” in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 689–698.
- [16] J. McAuley and J. Leskovec, “Hidden factors and hidden topics: understanding rating dimensions with review text,” in *Proceedings of the 7th ACM conference on Recommender systems*, 2013, pp. 165–172.
- [17] H. Wang, N. Wang, and D.-Y. Yeung, “Collaborative deep learning for recommender systems,” in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 1235–1244.
- [18] R. He and J. McAuley, “Vbpr: visual bayesian personalized ranking from implicit feedback,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.

- [19] C. Park, D. Kim, J. Oh, and H. Yu, “Do” also-viewed” products help user rating prediction?” in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 1113–1122.
- [20] J. Tang, X. Du, X. He, F. Yuan, Q. Tian, and T.-S. Chua, “Adversarial training towards robust multimedia recommender system,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 5, pp. 855–867, 2019.
- [21] G. V. Cormack, C. L. Clarke, and S. Buettcher, “Reciprocal rank fusion outperforms condorcet and individual rank learning methods,” in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 2009, pp. 758–759.
- [22] A. Salah, Q.-T. Truong, and H. W. Lauw, “Cornac: A comparative framework for multimodal recommender systems,” *Journal of Machine Learning Research*, vol. 21, no. 95, pp. 1–5, 2020.