Unsupervised Domain Specialization for Multimodal Embeddings

Anonymous ACL submission

Abstract

Recent advances in multimodal foundation models have enabled powerful generic embeddings. However, real-world applications—such as e-commerce or healthcare—rely on domain-specific patterns that these generic models often fail to capture precisely. Adapting them without readily available labeled data remains a critical challenge. In this paper, we propose MM-UDA, a robust unsupervised domain adaptation framework designed to systematically convert a generic multimodal embedding model into a domain expert using only unlabeled target data. Our framework employs a two-stage strategy. First, we introduce a within-modality pseudo-labeling task refined by Gaussian Mixture Model filtering, which rapidly adapts the model to domain-specific features while suppressing label noise. Then, we align modalities through a score-guided learning scheme to refine cross-modal pairing. Extensive evaluations on four real-world datasets demonstrate that MM-UDA both significantly enhances various backbone models and consistently outperforms competitive baselines, confirming its effectiveness for cross-modal retrieval in specialized domains. The source code is publicly available at https://anonymous.4open.science/r/UDSME.

1 Introduction

Multimodal retrieval, the task of matching a user query to the most relevant item across different data types (e.g., image, text), is a foundational component for applications like cross-modal recommendation(Yang et al., 2026; Zhuang et al., 2025) and retrieval-augmented generation (RAG)(Mortaheb et al., 2025; Li et al., 2025). Recently, Multimodal Large Language Models (MLLMs) have emerged as powerful general-purpose encoders, offering significantly improved universal cross-modal representations. While they achieve strong performance on broad benchmarks, they are inherently designed to capture common, domain-agnostic patterns. Consequently, they often fail to adequately represent the nuanced, domain-specific semantics and relationships that characterize specialized real-world environments such as e-commerce platforms, medical archives, or technical databases.

Applying generic MLLMs directly to such target domains—where data distributions and terminology diverge substantially from their pre-training corpus—thus presents significant practical hurdles, especially in the realistic scenario where **no manually labeled data is available for adaptation**. On the one hand, the distribution shift undermines the model's semantic alignment, causing it to misinterpret domain-specific features. On the other hand, the key challenge of unsupervised adaptation is the inherent noise in self-supervised signals. Directly generating pseudo-labels from the unaligned model on novel domain data typically yields low-quality, contradictory supervision. This noise can misguide the learning process, leading to incorrect cross-modal associations or even model collapse, rather than producing a reliable domain expert.

To address these issues, we propose **MM-UDA**, a systematic framework that progressively adapts MLLM embeddings. Our method is built on the insight that *effective cross-modal adaptation requires a solid intra-modal foundation followed by precise inter-modal alignment*. Therefore, we prioritize a robust unsupervised strategy consisting of two stages. In Stage 1, we perform intra-modal pseudo-label fine-tuning to adapt to domain-specific features. Crucially, we incorporate a Gaussian Mixture Model (GMM)-based filtering mechanism to dynamically identify and discard noisy pseudo-labels, ensuring stable optimization. In Stage 2, building on the refined domain features, we focus on cross-modal alignment. We introduce a novel score scheme combining local similarity and global ranking to select high-confidence cross-modal pairs, explicitly bridging the modality gap

and reducing separation. To demonstrate its effectiveness, we evaluate MM-UDA through comprehensive experiments on four real-world datasets across diverse intra-modal and cross-modal retrieval tasks. Our framework is validated using three distinct MLLM foundation models, consistently and significantly outperforming competitive baselines and substantially enhancing the base models via unsupervised adaptation.

Our main contributions are threefold:

- We propose **MM-UDA**, a progressive unsupervised domain adaptation framework that systematically overcomes domain shift by first learning intra-modal features and then aligning cross-modal representations using only unlabeled data.
- We design two novel mechanisms: a GMM-based pseudo-label filter for robust intra-modal adaptation, and a dual-stream confidence-invariance score for precise cross-modal alignment.
- Comprehensive experiments on four real-world datasets, conducted with three different foundation models, demonstrate that MM-UDA consistently outperforms baselines and significantly improves retrieval accuracy.

2 Related Work

MLLMs and Embedding-based Retrieval
The evolution of LLMs (Brown et al., 2020; Achiam et al., 2023) has directly enabled powerful MLLMs (Liu et al., 2023, 2024; Wang et al., 2024; Bai et al., 2025; Yang et al., 2025; Xu et al., 2025a,b; Chen et al., 2024; Yao et al., 2024), which integrate and reason over multiple modalities like text and images. This capability has established MLLMs as a promising foundation for generating universal, joint representations. Consequently, recent work has begun to leverage MLLMs as general-purpose encoders for *embedding-based multimodal retrieval*. Methods such as GME (Zhang et al., 2025), BGE (Zhou et al., 2025), and Tevatron (Ma et al., 2025) fine-tune the embedding spaces of models like LLaVA (Liu et al., 2023) and Qwen-VL (Bai et al., 2025) to extend their semantic matching capabilities from text to cross-modal tasks. While these approaches benefit from the strong general-world knowledge of foundation models, they typically assume the training and target data are from the same distribution. As a result, their performance degrades when applied to specialized domains (e.g., e-commerce, healthcare) with unique terminology and visual features, as the generic pretraining fails to capture these fine-grained, domain-specific patterns.

Unsupervised Domain Adaptation
Unsupervised Domain Adaptation (UDA) aims to transfer a model from a labeled source domain to an unlabeled target domain by mitigating distribution shift. In visual recognition, a dominant paradigm uses pseudo-labeling, where classifier predictions or feature clustering on target data generate surrogate labels for self-training (Zou et al., 2018; Caron et al., 2018). To improve reliability, these methods often incorporate confidence thresholds (Sohn et al., 2020) or uncertainty estimation (Wang et al., 2023; Rizve et al.) to filter noisy labels. Adaptation methods for CLIP-like vision-language models, such as TPT (Shu et al., 2022) and LatteCLIP (Cao et al., 2025), have also been explored to improve model transferability. While effective for closed-set classification, these strategies are fundamentally misaligned with the objectives of representation learning for *retrieval*. Classification UDA optimizes for discrete decision boundaries, whereas retrieval requires modeling fine-grained, continuous similarity structures within and across modalities. Directly applying pseudo-label filtering designed for classification leads to poor supervision for aligning embedding spaces, as it fails to account for the relational geometry and relative ranking that define retrieval quality. This gap necessitates a tailored, retrieval-centric adaptation framework.

3 Methodology

3.1 Framework Overview

We propose **MM-UDA**, a two-stage framework for unsupervised domain adaptation of a pretrained multimodal embedding model M. Given only unlabeled target domain data, the framework first adapts the model to capture domain-specific patterns within each modality (*Stage 1: Intra-modal Adaptation*), then refines it to align representations across modalities (*Stage 2: Cross-modal Alignment*). Both stages follow an iterative refinement process but employ distinct strategies for generating supervisory signals.

Formally, let $M^{(t)}$ denote the model parameters after the t-th adaptation iteration, initialized with the pretrained model $M^{(0)}$. Each iteration t consists of two core steps:

1. **Pseudo-Label Generation.** The current

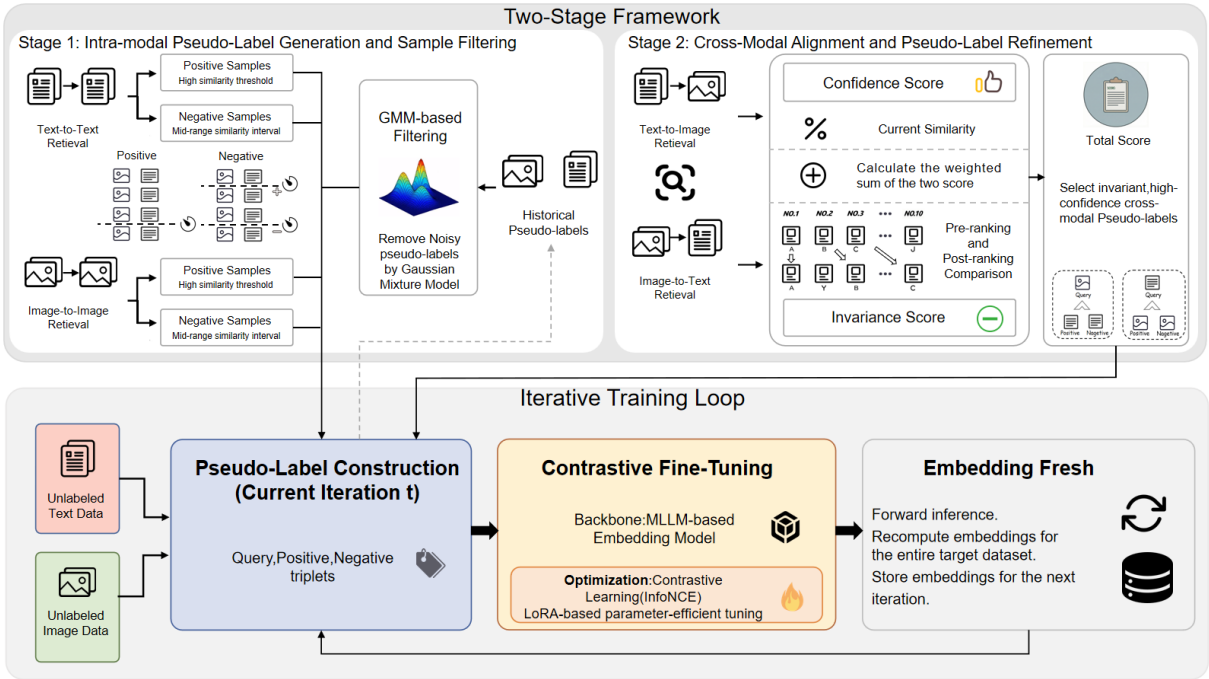


Figure 1: Illustration of proposed framework

model $M^{(t-1)}$ is used to construct a pseudo-label set $S^{(t)}$. For each query q, a retrieval operation in the model's embedding space identifies a positive sample set $P^+(q)$ and a negative sample set $P^-(q)$, forming a pseudo-labeled tuple $(q, P^+(q), P^-(q))$. The mechanisms for building P^+ and P^- differ between Stage 1 and Stage 2, targeting intra-modal and cross-modal relationships respectively.

2. Contrastive Fine-tuning. The model parameters are updated to $M^{(t)}$ by minimizing a contrastive loss over $S^{(t)}$. For a query q with positive $p \in P^+(q)$ and negative $a \subseteq P^-(q)$, we use the InfoNCE objective:

$$\mathcal{L}(q) = -\log \frac{\exp\left(\mathrm{sim}(z_q, z_p)/\tau\right)}{\sum\limits_{a \in P^+(q) \cup P^-(q)} \exp\left(\mathrm{sim}(z_q, z_a)/\tau\right)} \tag{1}$$

where $z_x = f_M(x)$ is the embedding of input x, $\mathrm{sim}(\cdot)$ denotes cosine similarity, and τ is a temperature hyperparameter.

3. Embedding Refresh and Iteration. After the contrastive fine-tuning step, $M^{(t)}$ is used to recompute the embeddings for all target domain data. The process of pseudo-label generation, contrastive fine-tuning, and embedding refresh forms a complete adaptation cycle. By repeating this cycle for multiple iterations, the model progressively refines its understanding of the target domain. The iterative nature of the framework allows it to gradually correct noisy pseudo-labels and converge to more robust target-domain representations. The following sections detail the specialized strategies employed in each stage to ensure robust adaptation.

3.2 Stage 1: Intra-modal Pseudo-Label Generation and Sample Filtering

Prior work shows that single-modality fine-tuning can still transfer to cross-modal tasks (Jiang et al., 2024). Motivated by this observation, Stage 1 first mines high-confidence pseudo-labels within each modality. This avoids unreliable early cross-modal mining under unknown target-domain shift, and instead uses more stable same-modality neighborhoods to rapidly adapt to domain-specific features and improve retrieval ability before cross-modal alignment.

1) Intra-modal Pseudo-Label Generation For each iteration, current model $M^{(t-1)}$ computes embeddings for all samples in the target domain, treating the text and image subsets as independent retrieval galleries. We randomly sample a query q (which can be either text or image) from the target dataset. Then, we perform dense retrieval within the same modality to construct pseudo-labeled triplets. For the query q, we select positive samples P^+ from the corresponding uni-modal candidates that satisfy a high similarity threshold:

$$P^+(q) = \{x \in \mathcal{D}_q \mid \mathrm{sim}(z_q, z_x) \geq \tau_{pos}\} \tag{2}$$

where $\mathcal{D}_q$ denotes the data subset of the same modality as q, and we retain the top-k candidates. Conversely, to form the negative set P^-, we select intra-modal samples falling within a specific similarity interval $[\tau_{neg}^{min}, \tau_{neg}^{max}]$. This strategy ensures that the model learns local manifold structures while avoiding easy negatives and minimizing false negatives.

2) GMM-based Sample Filtering Unsupervised pseudo-labels inevitably contain noise. We observe that noisy samples typically exhibit higher and more fluctuating loss values during training compared to clean samples. To exploit this pattern, we introduce a loss-based sample filtering mechanism.

At the end of each epoch, we collect the training loss ℓ_i for every sample i. We fit a two-component Gaussian Mixture Model (GMM) to the loss distribution:

$$p(\ell) = \sum_{k=1}^{2} \pi_k \mathcal{N}(\ell \mid \mu_k, \sigma_k^2) \tag{3}$$

where π_k, μ_k, and σ_k^2 represent the mixing probability, mean, and variance of the k-th component, respectively. The component with the smaller mean is treated as the clean set, and only samples with high posterior probability under this component are retained for the next round. Initial fixed scalar thresholds provide a coarse cross-domain boundary without manual tuning. Then GMM adaptively fits each dataset's loss distribution and filters samples by training dynamics, alleviating the rigidity of fixed thresholds and progressively purifying the data.

3.3 Stage 2: Cross-Modal Alignment and Pseudo-Label Refinement

Since Stage 1 have learned robust domain-specific features and improve retrieval ability significantly within each modality. Based on this,Stage 2 then target on mines reliable cross-modal pairs to explicitly align the representations across different modalities. Our strategy is driven by a critical empirical observation regarding prediction invariance. We observe that although the model evolves during Stage 1, a large majority of correctly retrieved items remain stable in their top-rank position after fine-tuning. This suggests that samples maintaining their top-rank position before and after fine-tuning represent the model's highest-confidence predictions. These invariant pairs constitute the most reliable anchors for bridging the modality gap.

Based on this insight, we design a score scheme to identify and prioritize these invariant samples.

1) Invariance Score The core of our strategy is to quantify the stability of high-ranked items between the pre-fine-tuning model (M_{pre}) and the post-fine-tuning model (M_{post}). We introduce a Invariant Score $S_{\text{invariance}}$ composed of two metrics:

Weighted Overlap Ratio (O_{weighted}): This metric measures the presence of important items in the top retrieval results of both models. For a randomly sampled query q, let R_{pre} and R_{post} be the top-10 retrieval lists obtained from the complementary modality gallery using M_{pre} and M_{post}, respectively. Let $S = R_{pre} \cap R_{post}$ be the set of overlapping items. Since we are particularly interested in the invariance of the very top results (e.g., Top-1), We use $w(k) = 1/k^2$ to reflect the head-heavy confidence distribution in retrieval: top-rank changes are penalized sharply, while lower-ranked neighbors still contribute non-zero context. The overlap score is:

$$O_{\text{weighted}} = \frac{\sum_{i \in S} w(\text{rank}_{pre}(i))}{\sum_{k=1}^{10} w(k)} \tag{4}$$

Weighted Average Displacement Score (D_{weighted}): Beyond mere overlap, we assess the strictness of rank preservation. Even if an item remains in the top-10, a shift in position indicates reduced confidence. For each overlapping item $i \in S$, we compute a displacement score based on its rank change $\Delta\text{rank}(i)$:

$$\text{score}(i) = 1 - \frac{\Delta\text{rank}(i)}{9} \tag{5}$$

The weighted average displacement is:

$$D_{\text{weighted}} = \frac{\sum_{i \in S} w_i \cdot \text{score}(i)}{\sum_{i \in S} w_i} \tag{6}$$

where $w_i = w(\text{rank}_{pre}(i))$. The final ranking score combines these factors to explicitly favor samples that remain "invariant" at the top ranks:

$$S_{\text{invariance}} = O_{\text{weighted}} \times D_{\text{weighted}} \tag{7}$$

2) Confidence Score To complement the ranking structure, we also consider local similarity confidence. We compute a normalized similarity R_{sim} to mitigate batch-wise distribution shifts. For the top-1 candidate in R_{post} with similarity s normalized within a batch range $[\text{min}_{\text{sim}}, \text{max}_{\text{sim}}]$:

$$S_{\text{con}} = \begin{cases} \dfrac{s - \text{min}_{\text{sim}}}{\text{max}_{\text{sim}} - \text{min}_{\text{sim}}} & \text{if } \text{max}_{\text{sim}} > \text{min}_{\text{sim}} \\ 0.5 & \text{otherwise} \end{cases} \tag{8}$$

3) Sample Selection The total reward is a weighted sum:

$$S_{\text{total}} = w_{\text{invariance}} \cdot S_{\text{invariance}} + w_{\text{con}} \cdot S_{\text{con}} \quad (9)$$

By selecting pairs with the highest S_{total}, we effectively filter for samples that are both locally similar and structurally invariant. This allows us to construct a high-quality cross-modal training set that aligns the modalities using the model's own most confident predictions.Further,we employ a Hard Negative Mining strategy. To avoid false negatives among high-confidence predictions, we sample n negatives strictly from the rank interval $[k, k+n-1]$. This ensures the model learns from challenging distractors while excluding potential unlabeled positives in the top-k ranks.

4 Experiments

4.1 Datasets

We conducted experimental evaluations on four real-world datasets, covering diverse domains, including movies, books, games, and long videos.

Letterboxd (gsimonx37, 2024): We use the movie *synopsis* as the text modality and the movie *poster* as the image modality.

Goodreads (Wan and McAuley, 2018; Wan et al., 2019): We use the book *summary* as the text modality and the book *cover* as the image modality.

Steam (Artermiloff, 2025): We use the game *description* as the text modality and the game *header image* as the image modality.

AgentMem-VQA: We construct AgentMem-VQA to evaluate multimodal memory retrieval for video QA agents over three long-form videos from different categories.

Data Protocol: The detailed statistics of the datasets are shown in Appendix C and Appendix D.4. To ensure a rigorous evaluation of our unsupervised framework, we adopt a strict data protocol. During the unsupervised phases, we treat the training partitions solely as unlabeled data, hiding the ground-truth text-image correspondences from the model. This simulates a realistic scenario where only raw corpora are available.

4.2 Implementation Details

We selected three multimodal embedding models as the backbone for our study: Qwen2.5-OmniEmbed-v0.1 (Ma et al., 2025), BGE-VL-MLLM-S2 (Zhou et al., 2025), and GME-Qwen2-VL-7B-Instruct (Zhang et al., 2025). All models possess cross-modal alignment capabilities, enabling them to generate semantically consistent embeddings for both text and image inputs.

Training Setup. We adopt the LoRA technique for parameter-efficient fine-tuning, where trainable parameters account for approximately 0.17% of the total. We use the AdamW optimizer with a learning rate of 2×10^{-5}. The LoRA rank is set to 8, and the scaling factor α is set to 32. The training objective is the InfoNCE contrastive loss with a temperature coefficient of 0.07. The detailed training settings, together with the hyperparameter sensitivity analysis, are provided in Appendix F.1 and Appendix F.2.

Baselines. Since no prior UDA work specifically targets MLLM-based embedding models, we adapt classical methods into generating pseudo-label for comparison. We select three representative baselines—SimCSE (Gao et al., 2021), FixMatch (Sohn et al., 2020), and UPS (Rizve et al.)—implemented with the same backbone and hyperparameters for fair evaluation. Detailed settings are provided in Appendix G.

Hybrid Evaluation Data Generation. To comprehensively assess the model's performance on hybrid embedding tasks, we constructed a evaluation set that mimics real-world user behavior. We employed Qwen3-32B to generate natural language queries for items in the test set by simulating user search intent based on item's content. These queries capture the semantic essence of the target items and serve as inputs for hybrid retrieval evaluation, while the original text descriptions and images are used to construct multimodal representations.

Evaluation Metrics. Retrieval accuracy and ranking quality serve as the core metrics for evaluating embedding models. To quantitatively assess performance, we employ Hit@1, Hit@5, and Mean Reciprocal Rank (MRR) as primary metrics.

4.3 Main Results on Item Retrieval

As shown in Table 1, the results validate the effectiveness of our proposed framework across two dimensions. First, the substantial gains in Intrinsic Evaluation ($T \leftrightarrow I$) demonstrate that our unsupervised strategy bridges the semantic gap, unifying previously disjoint textual and visual modalities into a cohesive representation space. Second, and perhaps more critically, these internal alignment improvements translate into superior performance on Extrinsic Evaluation. The pronounced uplift in query-based retrieval (T $\rightarrow$ T, T $\rightarrow$ I and T $\rightarrow$

Dataset	Method	Intrinsic Evaluation			Extrinsic Evaluation								
		T ↔ I			T → T			T → I			T → T&I		
		Hit@1	Hit@5	MRR	Hit@1	Hit@5	MRR	Hit@1	Hit@5	MRR	Hit@1	Hit@5	MRR
Letterboxd	OpenAI/CLIP	8.18	15.46	0.120	62.72	75.88	0.684	28.54	43.90	0.361	-	-	-
	Backbone: *Qwen2.5-OmniEmbed-v0.1*												
	Qwen2.5-OmniEmbed-v0.1	3.74	6.48	0.060	49.84	61.14	0.555	6.12	10.38	0.091	29.42	39.29	0.346
	SimCSE	7.38	11.28	0.094	58.68	71.62	0.648	19.54	34.76	0.268	50.42	59.54	0.553
	Fixmatch	7.18	10.84	0.091	57.94	69.86	0.637	19.02	33.80	0.262	49.80	58.62	0.539
	UPS	5.76	9.72	0.079	53.40	65.76	0.583	15.44	24.54	0.206	44.36	52.44	0.489
	Ours	**15.06**	**23.50**	**0.197**	**76.18**	**90.12**	**0.824**	**38.76**	**55.32**	**0.466**	**76.28**	**90.10**	**0.824**
	Backbone: *GME-Qwen2-VL-7B-Instruct*												
	GME-Qwen2-VL-7B-Instruct	10.08	16.80	0.142	80.28	92.30	0.867	40.84	54.92	0.476	78.88	91.64	0.846
	SimCSE	11.12	19.82	0.154	81.26	92.78	0.872	43.78	57.90	0.503	80.42	92.06	0.853
	Fixmatch	12.06	20.46	0.167	83.10	93.24	0.880	47.78	60.00	0.532	82.86	93.04	0.879
	UPS	12.48	21.14	0.172	84.32	93.68	0.899	49.34	63.52	0.547	84.50	93.56	0.894
	Ours	**15.76**	**24.28**	**0.203**	**88.32**	**95.26**	**0.920**	**60.70**	**73.34**	**0.662**	**88.94**	**95.42**	**0.925**
	Backbone: *BGE-VL-MLLM-S2*												
	BGE-MLLM-S2	6.92	13.20	0.108	58.50	73.42	0.654	19.22	30.30	0.251	52.98	70.52	0.611
	SimCSE	8.92	14.98	0.112	61.44	75.28	0.689	23.70	36.04	0.308	60.20	73.68	0.677
	Fixmatch	10.54	17.86	0.140	63.66	77.10	0.703	25.32	40.74	0.329	64.52	79.54	0.718
	UPS	11.06	18.64	0.143	64.18	77.48	0.711	25.58	40.46	0.327	64.76	77.56	0.713
	Ours	**13.26**	**20.44**	**0.168**	**69.78**	**85.50**	**0.763**	**31.58**	**47.62**	**0.390**	**70.32**	**85.36**	**0.771**
Goodreads	OpenAI/CLIP	7.82	14.60	0.112	32.68	46.72	0.405	20.34	32.82	0.276	-	-	-
	Backbone: *Qwen2.5-OmniEmbed-v0.1*												
	Qwen2.5-OmniEmbed-v0.1	12.78	19.92	0.170	29.80	42.40	0.357	9.52	17.20	0.139	30.50	46.04	0.380
	SimCSE	16.80	23.56	0.204	37.78	49.82	0.433	16.90	27.92	0.224	38.64	53.48	0.456
	Fixmatch	18.54	28.74	0.230	41.52	51.72	0.479	19.32	32.06	0.251	42.66	56.88	0.505
	UPS	21.76	34.82	0.266	44.08	56.10	0.517	24.64	38.00	0.319	46.20	64.82	0.550
	Ours	**30.40**	**42.78**	**0.361**	**67.16**	**84.48**	**0.750**	**48.24**	**69.68**	**0.580**	**67.64**	**87.30**	**0.765**
	Backbone: *GME-Qwen2-VL-7B-Instruct*												
	GME-Qwen2-VL-7B-Instruct	23.74	37.04	0.304	65.22	84.52	0.740	43.78	66.26	0.542	65.24	85.00	0.742
	SimCSE	25.46	37.80	0.313	67.24	86.38	0.752	47.56	70.64	0.558	67.46	86.80	0.763
	Fixmatch	28.12	41.08	0.346	69.52	87.00	0.763	50.58	72.80	0.595	70.12	87.88	0.778
	UPS	27.98	40.60	0.341	69.76	87.24	0.764	50.36	72.72	0.588	70.04	87.80	0.776
	Ours	**35.52**	**48.08**	**0.427**	**77.86**	**90.90**	**0.833**	**62.82**	**75.54**	**0.687**	**79.26**	**91.48**	**0.846**
	Backbone: *BGE-VL-MLLM-S2*												
	BGE-MLLM-S2	11.76	20.14	0.166	49.30	70.08	0.590	19.66	35.44	0.274	53.88	76.16	0.640
	SimCSE	12.92	21.34	0.180	53.34	72.24	0.627	27.76	41.70	0.348	55.94	78.56	0.667
	Fixmatch	14.74	23.84	0.203	55.48	74.46	0.642	30.60	44.12	0.382	57.26	80.34	0.688
	UPS	15.06	24.42	0.207	56.30	74.68	0.650	29.98	43.68	0.380	57.36	80.18	0.694
	Ours	**21.56**	**32.44**	**0.276**	**60.70**	**79.50**	**0.684**	**38.72**	**56.58**	**0.484**	**62.52**	**84.40**	**0.729**
Steam	OpenAI/CLIP	23.22	36.02	0.301	62.46	76.58	0.699	43.40	56.56	0.501	-	-	-
	Backbone: *Qwen2.5-OmniEmbed-v0.1*												
	Qwen2.5-OmniEmbed-v0.1	14.88	27.16	0.215	44.84	55.06	0.500	5.82	11.38	0.097	48.66	60.52	0.544
	SimCSE	29.84	43.70	0.367	58.80	70.46	0.641	32.44	49.60	0.382	62.34	72.34	0.679
	Fixmatch	35.62	46.08	0.433	65.58	75.86	0.694	42.50	54.82	0.492	67.74	75.68	0.717
	UPS	40.82	52.12	0.487	69.84	79.28	0.749	48.96	59.92	0.553	72.38	78.84	0.752
	Ours	**64.22**	**74.46**	**0.692**	**92.76**	**97.88**	**0.956**	**77.98**	**89.10**	**0.830**	**94.08**	**98.62**	**0.961**
	Backbone: *GME-Qwen2-VL-7B-Instruct*												
	GME-Qwen2-VL-7B-Instruct	46.28	62.06	0.537	87.46	94.84	0.909	58.14	74.80	0.658	87.70	95.56	0.913
	SimCSE	52.22	66.68	0.593	88.58	95.40	0.920	64.78	79.86	0.707	88.36	96.10	0.926
	Fixmatch	55.64	68.06	0.618	89.36	96.42	0.933	67.22	82.30	0.724	89.86	96.98	0.933
	UPS	57.72	69.30	0.625	90.96	97.18	0.941	70.56	83.90	0.755	90.32	97.44	0.937
	Ours	**67.82**	**74.58**	**0.707**	**93.88**	**98.56**	**0.960**	**81.34**	**92.46**	**0.866**	**94.30**	**98.94**	**0.964**
	Backbone: *BGE-VL-MLLM-S2*												
	BGE-MLLM-S2	20.82	35.66	0.282	71.58	88.22	0.791	39.82	57.40	0.483	83.38	94.30	0.883
	SimCSE	25.58	39.58	0.326	75.08	89.90	0.824	48.60	67.34	0.585	84.88	95.08	0.895
	Fixmatch	30.56	45.02	0.370	79.42	91.16	0.849	54.38	73.12	0.651	87.42	96.22	0.913
	UPS	27.50	42.48	0.359	77.58	90.44	0.835	51.66	70.64	0.620	86.80	96.14	0.904
	Ours	**40.22**	**60.62**	**0.517**	**83.60**	**93.12**	**0.878**	**62.50**	**80.62**	**0.724**	**89.20**	**97.56**	**0.936**

Table 1: Performance of our method versus baselines on three datasets using three model backbones. Intrinsic Evaluation measures the semantic alignment between the ground-truth item description and its corresponding image, whereas Extrinsic Evaluation assesses the model's practical utility in real-world search scenarios using LLM-generated user queries to retrieve target items. The best result is highlighted in **boldface** and the runner-up is denoted with underline.

Bench.	Qwen2.5-OmniEmbed-v0.1					GME-Qwen2-VL-7B-Instruct					BGE-VL-MLLM-S2				
	Base	SimCSE	Fix	UPS	Ours	Base	SimCSE	Fix	UPS	Ours	Base	SimCSE	Fix	UPS	Ours
RE9	54.2	57.6	55.0	55.4	**60.8**	58.4	60.8	61.4	61.6	**63.0**	53.4	55.8	55.4	55.2	**56.6**
Pilot	69.4	72.2	71.2	71.8	**74.2**	71.6	73.0	72.6	71.8	**75.0**	70.0	71.2	70.2	69.8	**72.8**
BMW	62.0	64.6	64.0	62.8	**66.4**	64.8	65.8	66.8	66.0	**68.2**	61.6	62.8	62.0	61.8	**64.0**

Table 2: MCQ accuracy (%) on MM-RAG. Fix: FixMatch. The best result is highlighted in **boldface** and the runner-up is denoted with underline.

T&I) indicates that the model has not only learned feature correspondence but enhanced its utility as a robust retrieval engine for real-world user intent. Furthermore, when compared with baselines, MM-UDA establishes a clear advantage. While general methods struggle with modality separation, our approach demonstrates superior adaptability and robustness across the diverse domains.

4.4 Application to Multimodal Memory Retrieval for Agents

To evaluate the practical utility of MM-UDA in agentic retrieval scenarios, we construct AgentMem-VQA, a benchmark for multimodal memory retrieval in memory-augmented video question answering. It consists of three long-form videos from different categories. We construct a chunk-level multimodal memory, where each memory contains a scene-level summary, transcript, and representative key frames.

During inference, a fixed answering agent retrieves the relevant memory entries conditioned on the question and reasons over the retrieved content to select the answer. Since both the memory construction procedure and the answering agent remain unchanged across all settings, performance differences primarily reflect the quality of the embeddings used for multimodal memory retrieval. We report MCQ accuracy over the three long-form video settings in Table 2. Detailed implementation and evaluation settings are provided in Appendix D.

As shown, MM-UDA-enhanced embeddings consistently achieve the highest accuracy across all three backbones and video settings, outperforming both the original embeddings and all baselines. On the most challenging setting, RE9, MM-UDA improves over the original backbone by 6.6, 4.6, and 3.2 points for Omni, GME, and BGE, respectively. These results demonstrate that MM-UDA enables more accurate retrieval of relevant multimodal memories, thereby improving the downstream reasoning performance of memory-augmented agents.

4.5 Ablation Study

To further validate the effectiveness of each module in the proposed framework, we conducted a series of ablation studies in Table 3.For conciseness, Table 3 reports the results on representative Steam dataset, while the complete ablation results on all datasets are provided in Appendix A. Specifically, *w/o(GMM,Stage 2)* denotes the model trained without Stage 2 and also with GMM sample filtering disabled in Stage 1, *w/o(Stage 2)* represents the model trained using only Stage 1, *w/o($S_{\text{invariance}}$)* represents the full model but remove Invariance Score in Stage 2, *w/o(S_{con})* represents the full model but disable the Confidence Score in stage 2, *Joint-training* denotes a variant that jointly optimizes the objectives of Stage 1 and Stage 2, and *full* corresponds to the complete proposed framework.

Experimental results show that models fine-tuned using only Stage 1 already achieve significant performance improvements compared with their pretrained counterparts. This confirms the necessity and effectiveness of Stage 1 in rapidly adapting the model to the feature of the target domain. Furthermore, when the sample filtering process is removed from Stage 1, the model performance exhibits noticeable degradation.

Regarding the ablation study of the scoring scheme in Stage 2, results shows that removing either $S_{\text{invariance}}$ or S_{con} leads to a noticeable performance degradation compared to our full framework. The *Joint-training* variant highlights the advantage of our decoupled design. The full pipeline consistently outperforms joint training across all, while some joint training even underperforms Stage 1 alone,as premature cross-modal pseudo-labels inject substantial noise into the training process. This suggests that premature cross-modal alignment may introduce noisy pseudo-labels, validating the necessity of first performing intra-modal adaptation before cross-modal alignment.

In summary, these findings corroborate the hierarchical design of our framework.

Method	Intrinsic Evaluation T ↔ I			Extrinsic Evaluation T → T&I		
	Hit@1	Hit@5	MRR	Hit@1	Hit@5	MRR
Backbone: *Qwen2.5-OmniEmbed-v0.1*						
w/o(GMM,Stage 2)	55.48	66.80	0.601	84.66	89.32	0.875
w/o(Stage 2)	58.64	69.72	0.633	86.34	90.78	0.880
w/o($S_{\text{invariance}}$)	62.92	73.06	0.676	92.68	97.54	0.945
w/o(S_{con})	63.50	73.64	0.684	93.12	98.08	0.950
Joint-training	51.64	60.98	0.558	77.40	83.56	0.802
full	**64.22**	**74.46**	**0.692**	**94.08**	**98.62**	**0.961**
Backbone: *GME-Qwen2-VL-7B-Instruct*						
w/o(GMM,Stage 2)	59.68	70.56	0.650	88.18	96.20	0.921
w/o(Stage 2)	61.70	71.64	0.668	88.58	96.42	0.923
w/o($S_{\text{invariance}}$)	67.60	**74.72**	0.715	94.14	98.54	0.958
w/o(S_{con})	66.84	73.76	0.706	93.56	97.88	0.952
Joint-training	62.18	72.48	0.687	90.42	97.10	0.935
full	**67.82**	74.58	**0.717**	**94.30**	**98.94**	**0.964**
Backbone: *BGE-VL-MLLM-S2*						
w/o(GMM,Stage 2)	31.76	44.48	0.382	87.60	96.42	0.918
w/o(Stage 2)	35.46	51.38	0.426	88.10	96.80	0.923
w/o($S_{\text{invariance}}$)	39.68	60.00	0.509	88.52	96.90	0.927
w/o(S_{con})	39.06	59.48	0.503	88.66	96.94	0.930
Joint-training	32.48	46.42	0.394	87.36	96.30	0.915
full	**40.22**	**60.62**	**0.517**	**89.20**	**97.56**	**0.936**

Table 3: Result of ablation study on Steam dataset.

Method	Letterboxd		Goodreads		Steam	
	Hit@1	MRR	Hit@1	MRR	Hit@1	MRR
Backbone: *Qwen2.5-OmniEmbed-v0.1*						
Base-SFT	71.82	0.780	62.70	0.703	81.56	0.865
Ours-SFT	**78.02**	**0.843**	**69.40**	**0.787**	**94.76**	**0.967**
Backbone: *GME-Qwen2-VL-7B-Instruct*						
Base-SFT	86.06	0.902	75.58	0.811	93.88	0.959
Ours-SFT	**90.18**	**0.930**	**82.00**	**0.861**	**94.90**	**0.969**
Backbone: *BGE-VL-MLLM-S2*						
Base-SFT	68.52	0.759	56.94	0.661	89.52	0.933
Ours-SFT	**74.86**	**0.822**	**64.40**	**0.755**	**90.34**	**0.940**

Table 4: Comparison in low-resource learning setting.

4.6 Further Discussions

Effectiveness for Low-Resource Learning: To further evaluate the practical utility of our framework in real-world settings where high-quality annotations are scarce, we simulated a low-resource scenario. We randomly sampled 1,500 labeled pairs from the training set to represent a limited labeling budget. We designed a comparative experiment with two setups: (1) Base-SFT: directly performing Supervised Fine-Tuning on the pre-trained base model using the 1,500 labeled pairs; (2) Our-SFT: performing the same fine-tuning but initializing from our Stage 2-aligned model. Evaluation focuses on the Hybrid Retrieval task.

Results in Table 4 show that our method substantially outperforms the Base-SFT baseline. The scarcity of labeled data makes it insufficient to adapt the generic pre-trained model to complex domain distribution. In contrast, initializing from our Stage 2-aligned model provides a domain-adapted warm start. This allows the same limited labeled data to refine decision boundaries rather than learn alignment from scratch, leading to significantly better performance with minimal supervision.

t-SNE Visualization: To gain deeper insights into how different methods influence the structure of the embedding space, we conducted a t-SNE visualization analysis. The experiment used Qwen2.5-OmniEmbed-v0.1 on the Letterboxd dataset. We randomly sampled 500 query–item pairs from the test set. The visualization results are provided in Appendix B.

For the *base model*, text and image embeddings are clearly distributed on opposite sides of the plot, showing a pronounced modality separation. This indicates that the original model has limited cross-modal alignment capability. After Stage 1 fine-tuning, the distributions of embeddings become noticeably closer, and the previously empty region shrinks significantly. This suggests that the model begins to capture domain-specific semantic correspondences, although two modalities still form relatively independent clusters.

After Stage 2 fine-tuning, the embedding space undergoes a substantial transformation: embeddings from both modalities become highly interwoven, and the modality boundaries almost disappear. This demonstrates that the model achieves substantial improvement in cross-modal semantic alignment. In contrast, other unsupervised domain adaptation baselines also reduce modality separation to some extent, but their embedding spaces still show visible clustering boundaries and fail to achieve complete modality fusion.

5 Conclusion

We present MM-UDA, a 2-stage framework for unsupervised domain adaptation of multimodal embedding models. By first stabilizing intra-modal learning with GMM-based noise filtering and then aligning modalities via a score-guided scheme, MM-UDA effectively overcomes domain shift without labeled data. Experiments cross four datasets demonstrate consistent improvements over base models and baselines, validating its utility for domain-specialized multimodal retrieval. Future work will extend the framework to additional modalities such as audio, and explore parameter-efficient adaptation for streaming environments.

6 Limitations

Our framework presents several limitations that point to future research directions. First, it assumes the base model retains a minimal level of initial cross-modal alignment; performance may degrade under extreme domain shifts where this invariant signal is too weak. Second, the method requires a sufficiently large candidate pool within the target domain to construct reliable contrastive pairs, which may not be available in extremely small-scale applications. Third, our current study is limited to the text-image modality pair due to dataset availability; extending our approach to other modalities such as audio or tabular data remains to be explored.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Artermiloff. 2025. Steam games dataset. https://www.kaggle.com/datasets/artermiloff/steam-games-dataset.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Anh-Quan Cao, Maximilian Jaritz, Matthieu Guillaumin, Raoul de Charette, and Loris Bazzani. 2025. Latteclip: Unsupervised clip fine-tuning via lmm-synthetic texts. In *WACV 2025-IEEE/CVF Winter Conference on Applications of Computer Vision*.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, and 1 others. 2024. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *Science China Information Sciences*, 67(12):220101.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 6894–6910.

gsimonx37. 2024. Letterboxd dataset. https://www.kaggle.com/datasets/gsimonx37/letterboxd.

Ting Jiang, Minghui Song, Zihan Zhang, Haizhen Huang, Weiwei Deng, Feng Sun, Qi Zhang, Deqing Wang, and Fuzhen Zhuang. 2024. E5-v: Universal embeddings with multimodal large language models. *arXiv preprint arXiv:2407.12580*.

Yangning Li, Yinghui Li, Xinyu Wang, Yong Jiang, Zhen Zhang, Xinran Zheng, Hui Wang, Hai-Tao Zheng, Fei Huang, Jingren Zhou, and 1 others. 2025. Benchmarking multimodal retrieval augmented generation with dynamic vqa dataset and self-adaptive planning agent. In *International Conference on Learning Representations*, volume 2025, pages 95582–95604.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024. Llava-next: Improved reasoning, ocr, and world knowledge.

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

Xueguang Ma, Luyu Gao, Shengyao Zhuang, Jiaqi Samantha Zhan, Jamie Callan, and Jimmy Lin. 2025. Tevatron 2.0: Unified document retrieval toolkit across scale, language, and modality. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4061–4065.

Matin Mortaheb, Mohammad A Amir Khojastepour, Srimat T Chakradhar, and Sennur Ulukus. 2025. Re-ranking the context for multimodal retrieval augmented generation. *arXiv preprint arXiv:2501.04695*.

Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In *International Conference on Learning Representations*.

Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. 2022. Test-time prompt tuning for zero-shot generalization in vision-language models. *Advances in Neural Information Processing Systems*, 35:14274–14289.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608.

Mengting Wan and Julian J. McAuley. 2018. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018*, pages 86–94. ACM.

Mengting Wan, Rishabh Misra, Ndapa Nakashole, and Julian J. McAuley. 2019. Fine-grained spoiler detection from large-scale review corpora. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2605–2610. Association for Computational Linguistics.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Yue Fan, Zhen Wu, Jindong Wang, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, and 1 others. 2023. Freematch: Self-adaptive thresholding for semi-supervised learning. In *Eleventh International Conference on Learning Representations*. OpenReview. net.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, and 1 others. 2025a. Qwen2. 5-omni technical report. *arXiv preprint arXiv:2503.20215*.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, and 1 others. 2025b. Qwen3-omni technical report. *arXiv preprint arXiv:2509.17765*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Wei Yang, Rui Zhong, Yiqun Chen, Chi Lu, and Peng Jiang. 2026. Structured spectral reasoning for frequency-adaptive multimodal recommendation. *Advances in Neural Information Processing Systems*, 38:28122–28143.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, and 1 others. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*.

Xin Zhang, Yanzhao Zhang, Wen Xie, Mingxin Li, Ziqi Dai, Dingkun Long, Pengjun Xie, Meishan Zhang, Wenjie Li, and Min Zhang. 2025. Bridging modalities: Improving universal multimodal retrieval by multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Junjie Zhou, Yongping Xiong, Zheng Liu, Ze Liu, Shitao Xiao, Yueze Wang, Bo Zhao, Chen Jason Zhang, and Defu Lian. 2025. Megapairs: Massive data synthesis for universal multimodal retrieval. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19076–19095.

Ziyi Zhuang, Hanwen Du, Hui Han, Youhua Li, Junchen Fu, Joemon M Jose, and Yongxin Ni. 2025. Bridging the gap: Teacher-assisted wasserstein knowledge distillation for efficient multi-modal recommendation. In *Proceedings of the ACM on Web Conference 2025*, pages 2464–2475.

Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. 2018. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 289–305.

A Full Ablation Study

To provide a more comprehensive evaluation, we report the complete ablation study results on all datasets in Table 5.

B t-SNE Visualization

Figure 2 presents the t-SNE visualization results discussed in Section 4.6.

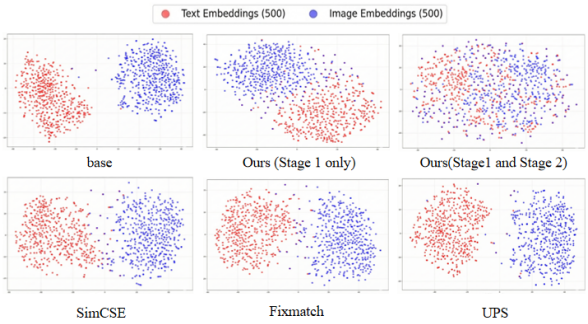


Figure 2: t-SNE visualization.

C Detail of Data Statistics.

The data statistics are summarized in Table 6.For each dataset, we randomly sampled 1,000 labeled pairs to constitute the validation set and 5,000 pairs for the test set, while the remainder of the dataset serves as the training set. The data do not contain any information that names or uniquely identifies individual people or offensive content.

License and usage terms: The Letterboxd, Goodreads, and Steam datasets are collected from public sources and used for research purposes only; their original licenses (if any) follow the

	Method	Intrinsic Evaluation T ↔ I			Extrinsic Evaluation T → T&I		
		Hit@1	Hit@5	MRR	Hit@1	Hit@5	MRR
Letterboxd	Backbone: *Qwen2.5-OmniEmbed-v0.1*						
	w/o(GMM,Stage 2)	10.58	17.38	0.157	65.46	78.66	0.711
	w/o(Stage 2)	11.80	19.52	0.173	69.80	83.48	0.757
	w/o($S_{\text{invariance}}$)	14.80	23.02	0.191	75.46	89.30	0.813
	w/o(S_{con})	14.42	22.34	0.185	74.78	87.96	0.802
	Joint-training	10.92	18.04	0.164	67.44	80.34	0.739
	full	**15.06**	**23.50**	**0.197**	**76.28**	**90.10**	**0.824**
	Backbone: *GME-Qwen2-VL-7B-Instruct*						
	w/o(GMM,Stage 2)	11.86	19.10	0.172	81.36	92.32	0.862
	w/o(Stage 2)	12.32	21.44	0.178	83.82	93.24	0.881
	w/o($S_{\text{invariance}}$)	15.32	23.48	0.196	88.30	94.98	0.919
	w/o(S_{con})	15.10	23.04	0.194	88.14	94.64	0.917
	Joint-training	13.68	21.30	0.185	85.62	93.88	0.893
	full	**15.76**	**24.28**	**0.203**	**88.94**	**95.42**	**0.925**
	Backbone: *BGE-VL-MLLM-S2*						
	w/o(GMM,Stage 2)	9.98	16.56	0.133	63.78	77.60	0.700
	w/o(Stage 2)	10.62	17.58	0.140	65.10	80.54	0.722
	w/o($S_{\text{invariance}}$)	12.48	19.08	0.159	69.20	84.28	0.763
	w/o(S_{con})	12.70	19.16	0.162	69.34	84.44	0.762
	Joint-training	10.84	17.92	0.143	65.78	81.20	0.726
	full	**13.26**	**20.44**	**0.168**	**70.32**	**85.36**	**0.771**
Goodreads	Backbone: *Qwen2.5-OmniEmbed-v0.1*						
	w/o(GMM,Stage 2)	24.76	38.38	0.314	58.96	79.60	0.675
	w/o(Stage 2)	27.64	36.46	0.340	61.52	83.88	0.708
	w/o($S_{\text{invariance}}$)	30.02	42.28	0.358	66.88	86.30	0.757
	w/o(S_{con})	29.56	41.44	0.352	66.26	86.00	0.751
	Joint-training	28.32	37.18	0.346	63.54	85.00	0.726
	full	**30.40**	**42.78**	**0.361**	**67.64**	**87.30**	**0.765**
	Backbone: *GME-Qwen2-VL-7B-Instruct*						
	w/o(GMM,Stage 2)	28.40	39.48	0.358	70.88	88.12	0.784
	w/o(Stage 2)	30.32	42.05	0.376	72.68	89.36	0.789
	w/o($S_{\text{invariance}}$)	34.56	47.18	0.417	77.82	89.70	0.831
	w/o(S_{con})	34.92	47.60	0.421	78.44	90.48	0.839
	Joint-training	31.44	43.64	0.390	72.80	89.18	0.792
	full	**35.52**	**48.08**	**0.427**	**79.26**	**91.48**	**0.846**
	Backbone: *BGE-VL-MLLM-S2*						
	w/o(GMM,Stage 2)	13.56	20.98	0.185	55.62	78.84	0.661
	w/o(Stage 2)	14.80	22.40	0.197	57.02	79.88	0.682
	w/o($S_{\text{invariance}}$)	21.18	31.46	0.270	61.90	83.76	0.720
	w/o(S_{con})	20.60	30.84	0.264	60.82	82.58	0.711
	Joint-training	14.98	22.92	0.202	57.20	79.80	0.683
	full	**21.56**	**32.44**	**0.276**	**62.52**	**84.40**	**0.729**
Steam	Backbone: *Qwen2.5-OmniEmbed-v0.1*						
	w/o(GMM,Stage 2)	55.48	66.80	0.601	84.66	89.32	0.875
	w/o(Stage 2)	58.64	69.72	0.633	86.34	90.78	0.880
	w/o($S_{\text{invariance}}$)	62.92	73.06	0.676	92.68	97.54	0.945
	w/o(S_{con})	63.50	73.64	0.684	93.12	98.08	0.950
	Joint-training	51.64	60.98	0.558	77.40	83.56	0.802
	full	**64.22**	**74.46**	**0.692**	**94.08**	**98.62**	**0.961**
	Backbone: *GME-Qwen2-VL-7B-Instruct*						
	w/o(GMM,Stage 2)	59.68	70.56	0.650	88.18	96.20	0.921
	w/o(Stage 2)	61.70	71.64	0.668	88.58	96.42	0.923
	w/o($S_{\text{invariance}}$)	67.60	**74.72**	0.715	94.14	98.54	0.958
	w/o(S_{con})	66.84	73.76	0.706	93.56	97.88	0.952
	Joint-training	62.18	72.48	0.687	90.42	97.10	0.935
	full	**67.82**	74.58	**0.717**	**94.30**	**98.94**	**0.964**
	Backbone: *BGE-VL-MLLM-S2*						
	w/o(GMM,Stage 2)	31.76	44.48	0.382	87.60	96.42	0.918
	w/o(Stage 2)	35.46	51.38	0.426	88.10	96.80	0.923
	w/o($S_{\text{invariance}}$)	39.68	60.00	0.509	88.52	96.90	0.927
	w/o(S_{con})	39.06	59.48	0.503	88.66	96.94	0.930
	Joint-training	32.48	46.42	0.394	87.36	96.30	0.915
	full	**40.22**	**60.62**	**0.517**	**89.20**	**97.56**	**0.936**

Table 5: Full result of ablation study.

respective platform terms. All pre-trained models (Qwen2.5-OmniEmbed, GME, BGE) are used under their official non-commercial research licenses.The AgentMem-VQA videos were collected from public online sources and are used solely for non-commercial research; the original copyright belongs to their respective creators.

Dataset	Texts	Images
Letterboxd	523965	523965
Goodreads	500000	500000
Steam	89599	89599

Table 6: Basic statistics of the three datasets.

C.1 Detail of Query Generation.

C.1.1 Prompt Design for Query Generation

To generate realistic user-style query texts corresponding to each target movie description, we employed the Qwen-32B model with the prompt in Listing 1.

C.1.2 Generated Query examples

To illustrate the process of query generation used in constructing the text-to-text evaluation dataset, we provide several representative examples in Listing 2.

C.2 Qualitative Check of Generated Query Quality

We use Qwen3-32B to generate natural-language queries from item descriptions. To demonstrate that Qwen3-32B provides sufficient query quality, we compare its outputs with those from GPT-5 on representative examples. Table 7 shows several paired generations; both models produce queries that preserve the main search intent and sound natural, indicating that Qwen3-32B is adequate for this controlled query-generation step.

D Memory-Augmented Video QA: Implementation Details

D.1 Chunk-Level Memory System

We implement a chunk-level multimodal memory system that segments long videos into time-bounded memory chunks and encodes each chunk as a multimodal embedding for retrieval-augmented question answering.

Memory Unit Design. Each memory chunk contains: (1) a unique `chunk_id` and the source `video_id`; (2) start and end timestamps; (3) aggregated transcript text from the corresponding time

```
# Role
You are a user trying to find a specific {category} using a search engine.

# Input Data
- **Name:** {name}
- **Description:** {description}

# Task
Based on the `Description`, identify the **most obvious and distinct feature** (such as the
    core plot, unique gameplay mechanism, visual style, or main character) and formulate **one
    ** search query.

# Constraints
1. **No Exact Title:** Do not simply output the Name. Simulate a user who remembers the content
     but not the title.
2. **Focus on Content:** The query should describe *what it is* or *what happens* (e.g., "game
    where you hunt robot dinosaurs" instead of "Horizon Zero Dawn").
3. **Natural Style:** Use casual, short search language suitable for a search bar.
4. **Output Format:** Output **only** the query text. No quotes or explanations.

# Output
```

Listing 1: Prompt used for query generation.

```
{
  "id": 1000756,
  "name": "Synecdoche, New York",
  "description": "A theater director struggles with his work, and the women in his life, as he
    attempts to create a life-size replica of New York inside a warehouse as part of his new
    play.",
  "query": "What happens in the ending of Synecdoche New York?"
}

{
  "id": 289650,
  "name": "Assassin's Creed Unity",
 "description": "Assassin's Creed Unity is an action/adventure game set in the city of Paris
    during one of its darkest hours, the French Revolution. Take ownership of the story by
    customizing Arno's equipment to make the experience unique to you, both visually and
    mechanically. In addition to an epic single-player experience, Assassin's Creed Unity
    delivers the excitement of playing with up to three friends through online cooperative
    gameplay in specific missions. Throughout the game, take part in one of the most pivotal
    moments of French history in a compelling storyline and a breath-taking playground that
    brought you the city of lights of today.",
  "query": "how to play Assassin's Creed Unity with friends online"
}
```

Listing 2: Examples of queries generated by LLM.

Item	GPT-5 query	Qwen3-32B query
Assassin's Creed Unity	Does Assassin's Creed Unity have co-op missions?	How to play Assassin's Creed Unity with friends online?
American Gigolo	Who is framed for murder in American Gigolo?	Who is the real killer in American Gigolo?

Table 7: Representative examples of generated queries. Both generators preserve the main search intent, although surface wording may vary.

span; (4) a scene description, a scene summary, and scene anchors (key phrases for retrieval); and (5) references to selected key frames (typically 6, up to 8). This design ensures that each memory unit is a semantically enriched representation combining textual and visual evidence.

Indexing and Retrieval. We encode each chunk into text and image embeddings, and construct fused embeddings by combining both modalities. A FAISS index is built over the normalized fused embeddings using inner-product search.The final top-5 chunks are organized into an explicit memory context for the answer model.

D.2 Memory Construction Pipeline

Backbone Model. Throughout the entire pipeline—including memory enrichment, QA generation, and MCQ evaluation—we use Qwen3.5-27B as the unified backbone LLM.

Audio Extraction and ASR. We extract mono 16kHz audio from each source video, then generate word-level timestamped transcripts using Qwen3-ASR-1.7B and Qwen3-ForcedAligner-0.6B.

Chunk Segmentation. Chunks are constructed using a 20-second sliding window with 5-second overlap. Each window boundary is aligned to the nearest transcript boundary within a configurable tolerance to reduce mid-sentence splits. All transcript segments overlapping with the window are concatenated into the chunk's transcript text.

Memory Enrichment. Each chunk undergoes an enrichment stage where an LLM receives the chunk's video clip, transcript, and a prompt profile, and generates the scene description, scene summary, scene anchors, and key frame timestamps. This stage transforms raw time-windowed segments into retrieval-friendly memory units.

D.3 QA Data Generation

For each video, we generate 500 question-answer pairs from the video chunks. Each question is generated from one or multiple evidence chunks, covering fact-based questions (targeting objects, states, and events) and guidance-based questions (targeting procedural steps and operational judgments).

D.4 Dataset Statistics

We use three video sources spanning different domains. Table 8 summarizes the key statistics.

	RE9	BMW	Pilot
Domain	Game	Automotive	Tutorial
Duration	4h 52m	3h 13m	2h 42m
# Chunks	1168	774	649
Chunk window	20s	20s	20s
Overlap	5s	5s	5s
# QA	500	500	500

Table 8: Dataset statistics for the three video sources. RE9: Resident Evil Requiem (game walkthrough); BMW: BMW M3 crash rebuild (automotive repair); Pilot: ArduPilot Pixhawk multirotor tutorial (drone setup guide).

E Scalability and Convergence Analysis

In this section, we report the computational cost per iteration round and the number of rounds required for convergence across all backbone–dataset combinations.

Computational Cost per Round. Table 9 reports the wall-clock GPU hours consumed per round of MM-UDA. All experiments are conducted on a single NVIDIA A100 40GB GPU. The cost per round encompasses embedding computation over the entire target dataset, pseudo-label generation (including GMM filtering or cross-modal scoring), and contrastive fine-tuning.

Dataset	Omni	GME	BGE
Letterboxd	12	16	16
Goodreads	15	18	19
Steam	7	10	10

Table 9: GPU hours per round on a single NVIDIA A100. Omni: Qwen2.5-OmniEmbed-v0.1; GME: GME-Qwen2-VL-7B-Instruct; BGE: BGE-VL-MLLM-S2.

Number of Rounds and Convergence. Table 10 details the number of rounds executed for Stage 1 (intra-modal adaptation) and Stage 2 (cross-modal alignment) across all settings. We monitor the validation performance after every round and terminate training when no further improvement is observed, following a patience-based early stopping criterion. Stage 1 converges within 2–3 rounds and Stage 2 converges within 2–4 rounds across all nine backbone–dataset combinations, yielding a total of 4–7 rounds per full run.

Dataset	Omni	GME	BGE
Letterboxd	3 / 4	3 / 2	3 / 2
Goodreads	3 / 3	3 / 3	2 / 3
Steam	3 / 3	3 / 3	3 / 2

Table 10: Number of rounds for Stage 1 / Stage 2.

Table 11 further illustrates the per-round performance evolution across all datasets. Hit@1 improves monotonically throughout both stages, with diminishing gains in later rounds, confirming stable and consistent convergence behavior.

F Hyperparameters

F.1 Hyperparameter Settings

This section provides the detailed hyperparameter settings used in our two-stage training frame-

Dataset	Model	Base	S1-R1	S1-R2	S1-R3	S2-R1	S2-R2	S2-R3	S2-R4
Letterboxd	Omni	3.74	8.36	10.58	11.80	12.94	14.12	14.62	15.06
	GME	10.08	11.10	11.92	12.32	14.66	15.76	–	–
	BGE	6.92	8.44	10.04	10.62	12.32	13.26	–	–
Goodreads	Omni	12.78	19.36	24.08	27.64	28.96	29.70	30.40	–
	GME	23.74	26.70	28.82	30.32	32.14	34.38	35.52	–
	BGE	11.76	12.98	14.80	–	18.54	20.32	21.56	–
Steam	Omni	14.88	42.54	54.88	58.64	61.50	63.12	64.22	–
	GME	46.28	55.26	59.04	61.70	64.82	66.78	67.82	–
	BGE	20.82	27.30	31.94	35.46	38.74	40.22	–	–

Table 11: Evolution of Hit@1 across iterations on all datasets. S1-Rn and S2-Rn denote the n-th round of Stage 1 and Stage 2, respectively. – indicates that the stage has converged and no further round is executed. Omni: Qwen2.5-OmniEmbed-v0.1; GME: GME-Qwen2-VL-7B-Instruct; BGE: BGE-VL-MLLM-S2.

work.We use the same set of hyperparameters across all experiments.

Stage 1 Settings. In Stage 1, we generate 2,000 pseudo-labeled samples at each iteration for intra-modal adaptation. For pseudo-label selection, we use a similarity threshold of 0.85 to identify positive samples, with at most 3 positive samples selected for each query. Negative samples are selected from candidates whose similarity scores fall within the interval of $[0.6, 0.7]$, with at most 5 negative samples selected for each query.

Stage 2 Settings. In Stage 2, we generate cross-modal pseudo-labels by processing 50,000 queries. Among these candidates, 2,000 high-quality samples are selected according to the proposed scoring scheme. The weights of the two scoring terms are set to $w_{\text{invariance}} = 0.5$ and $w_{\text{con}} = 0.5$, giving equal importance to the invariance score and the confidence score. The k for hard negative mining strategy is set to 15.

F.2 Hyperparameter Sensitivity Analysis

MM-UDA introduces several hyperparameters across its two stages. In this section, we conduct a systematic sensitivity analysis on two key hyperparameters in Stage 1: the positive threshold for pseudo-label generation and the hard-negative starting rank k. All experiments are conducted on the Steam dataset and evaluated with Hit@1.

Positive Threshold. The positive threshold controls the minimum similarity required for a retrieved sample to be considered a positive pair in intra-modal contrastive learning. Table 12 reports the results across six threshold values ranging from 0.5 to 0.9. Performance is consistently strong in the range of 0.85–0.9 across all three backbones, and degrades gradually as the threshold decreases. This is expected: a lower threshold admits noisier positives into the training set, which undermines the quality of contrastive learning. The difference within the 0.85–0.9 range is marginal, indicating that MM-UDA is not sensitive to the exact threshold value within this range. In practice, we set this threshold to 0.85 across all settings without per-dataset tuning.

Model	0.9	0.85	0.8	0.7	0.6	0.5
Omni	58.32	58.64	57.88	53.10	48.46	44.52
GME	61.82	61.70	60.26	58.38	57.72	51.90
BGE	35.42	35.46	35.08	33.64	30.00	28.64

Table 12: Sensitivity analysis of the Stage 1 positive threshold on the Steam dataset (Hit@1). Omni: Qwen2.5-OmniEmbed-v0.1; GME: GME-Qwen2-VL-7B-Instruct; BGE: BGE-VL-MLLM-S2.

Hard-Negative Starting Rank k. The hard-negative starting rank k determines the position in the retrieval ranking from which hard negatives are sampled. A smaller k yields harder negatives that are closer to the query in the embedding space, while a larger k produces easier negatives. Table 13 reports the results across six values of k ranging from 10 to 1000. Performance remains stable across $k \in \{10, 15, 20\}$, with only minor degradation as k increases to 100 or beyond. This suggests that the model benefits from moderately hard negatives but is robust to the exact choice of k within a reasonable range. Importantly, the selection of k does not require any labeled data: we simply choose a small value (15 in our experiments) based on the intuition that top-ranked but non-positive items serve as informative negatives. This choice is applied uniformly across all datasets and backbones, preserving the fully unsupervised nature of our framework.

Model	10	15	20	100	500	1000
Omni	63.86	64.22	63.70	63.58	62.96	62.24
GME	67.80	67.82	67.76	67.02	65.66	64.78
BGE	40.22	40.22	39.46	38.70	38.04	37.36

Table 13: Sensitivity analysis of the hard-negative starting rank k on the Steam dataset (Hit@1). Omni: Qwen2.5-OmniEmbed-v0.1; GME: GME-Qwen2-VL-7B-Instruct; BGE: BGE-VL-MLLM-S2.

G Detailed baseline settings

Augmentation Strategies.

(1) For Image Queries, weak augmentation applies standard resizing and normalization to the raw pixels, while strong augmentation employs RandAugment, including color jittering, Gaussian blur, and random rotation. (2) For Text Queries, weak augmentation involves *Random Word Deletion*, whereas strong augmentation involves stochastic *Random Word Deletion* and *Word Reordering* applied directly to the input sentence structure to perturb semantic coherence.

SimCSE-based Adaptation. We adapt the SimCSE framework for pseudo-label generation.

- **Positive Pair Construction:** For a given query q (text or image), we apply the strong augmentation strategy $\mathcal{A}(\cdot)$ (as defined above) to generate an augmented view $q' = \mathcal{A}(q)$. The original query q and its augmented version q' are treated as the positive pair.
- **Negative Sampling:** We employ **random sampling** from the dataset to form the negative set $\mathcal{N}_{neg}$ for each query.
- **Optimization:** The model is trained using the same contrastive learning settings as our framework.

FixMatch-based Adaptation. We adapt the semi-supervised learning framework FixMatch for pseudo-label generation. We migrate the core concept of consistency regularization to the multimodal retrieval task. Specifically, for an unlabeled query q (which can be either an image or a text), we generate two views: a weakly augmented view q_{weak} and a *strongly augmented* view q_{strong}.

Label Generation and Training. The training process follows a three-step pipeline:

- **Pseudo-Labeling:** The model encodes q_{weak} to retrieve the Top-1 k^+ from the gallery. We employ a fixed confidence threshold τ (e.g., 0.35) based on cosine similarity. If $\text{sim}(q_{\text{weak}}, k^+) > \tau$, k^+ is assigned as the positive pseudo-label.Strongly augmented query q_{strong} treated as query.
- **Negative Sampling:** This baseline employs the same negative sampling as our Stage 2.
- **Optimization:** The model is trained using the same contrastive learning settings as our framework.

UPS-based Adaptation. We adapt the uncertainty-aware pseudo-label selection framework UPS to pseudo-labeling.

- **Uncertainty Estimation via Augmentation:** For a given query q, we generate N strongly augmented views $\{q'_1, q'_2, \ldots, q'_N\}$ (e.g., $N = 5$) using the stochastic augmentation strategies defined above. The model encodes the original query q to retrieve the Top-1 nearest neighbor k^+. Subsequently, we compute the cosine similarities between k^+ and each of the augmented views $\{q'_i\}_{i=1}^N$.
- **Pseudo-Label Selection:** We calculate the mean μ and variance σ^2 of these similarity scores. Here, μ represents the overall prediction confidence, while σ^2 quantifies the sensitivity of the prediction to input perturbations. A candidate pair (q, k^+) is selected as a valid pseudo-label only if it demonstrates both high confidence ($\mu > \tau_{conf}$) and high stability ($\sigma^2 < \kappa_{unc}$). Empirically, we set $\tau_{conf} = 0.35$ and $\kappa_{unc} = 0.008$ to strictly filter out unreliable pairs.
- **Negative Sampling:** This baseline employs the same negative sampling as our Stage 2.
- **Optimization:** The model is trained using the same contrastive learning settings as our framework.

H Experimental Environments

All experiments were conducted on a Linux server running Ubuntu 20.04 LTS. The framework was implemented in PyTorch 2.6.0 with CUDA 12.9. Hardware configurations included an Intel Xeon Gold 6248R CPU (3.00 GHz) with 1512 GB of RAM, and NVIDIA A100 GPU (40GB VRAM) for model training.

I Case Study

To further provide an intuitive demonstration of the performance differences before and after unsupervised domain adaptation fine-tuning, we present two representative case analyses.

(1) Text-to-Image Retrieval

In the first case shown in Table 14, the query corresponds to the movie synopsis of *The Wolf of Wall Street*. Before fine-tuning, the retrieval results were largely constrained by the model's reliance on keywords. The retrieved items mainly focused on samples containing frequent terms such as "Wall Street" failing to accurately capture the overall semantics of the text or its semantic alignment with images. After fine-tuning, the model exhibited a significantly improved understanding of textual semantics, performing cross-modal matching based on the holistic meaning of the sentences rather than keyword overlap. The correct matching image, which was absent from the pre-finetuning retrieval list, rose to the top rank after fine-tuning, indicating a substantial enhancement in the model's cross-modal semantic alignment ability.

(2) Image-to-Text Retrieval

In the second case shown in Table 15, the query is the movie poster of Barbie. Before fine-tuning, the model tended to focus on visual elements such as color tones and character appearance, resulting in retrievals biased toward visually similar but semantically irrelevant text descriptions. After fine-tuning, the model demonstrated a better understanding of the semantic correspondence between image and text, producing retrieval results that are semantically more coherent. Empirically, the correct textual match improved its ranking from a lower position to the second place after fine-tuning, showing clear progress in the model's ability to model cross-modal associations.

(3) Hybrid Retrieval ($T \rightarrow T\&I$) In the hybrid retrieval scenario presented in Table 16, the query "Did truman ever find out his life was a show" is designed to retrieve the specific plot details of *The Truman Show*. Before fine-tuning, the retrieval results were largely constrained by the model's reliance on surface-level keyword matching. As observed, the base model failed to disambiguate the entity "Truman," prioritizing historically prominent figures such as President Harry Truman (No.1) and writer Truman Capote (No.3) due to their high frequency in the corpus. Although the correct item contains the keyword, it was suppressed to the 5th rank, overwhelmed by these semantic distractors. After fine-tuning, the model exhibited a significantly improved ability to interpret complex semantic context. It successfully aligned the query's narrative intent with the target description's core concept, rather than getting distracted by the name "Truman." Consequently, the correct item rose to the Top-1 position, demonstrating the framework's effectiveness in locking onto the precise hybrid semantics.

(4) Open-ended Hybrid Retrieval ($T \rightarrow T\&I$)

In this case (Table 17), the user explicitly expresses an intent to "do farmwork" in a game. The Base Model captures the general agricultural theme but suffers from semantic drift. In contrast, the Finetuned Model demonstrates superior intent alignment. It strictly retrieves titles centered on agricultural mechanics, such as *Farming World* and *Farm Manager World*. Crucially, it successfully identifies *Stardew Valley* (No.5)—a premier title in the genre—demonstrating the model's capability to associate the abstract concept of "farmwork" with specific, high-quality game instances that may not strictly rely on the word "work" in their titles.

From the cases above, it is evident that our framework leads to a comprehensive performance improvement in retrieval tasks. This not only enhances retrieval accuracy but also validates the practical effectiveness and robustness of the proposed two-stage unsupervised training strategy for multimodal representation learning in target domains.

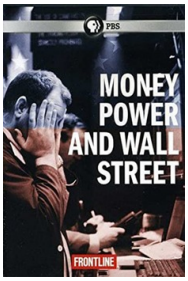
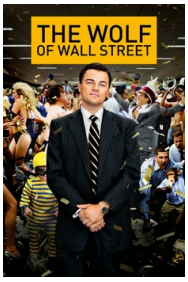
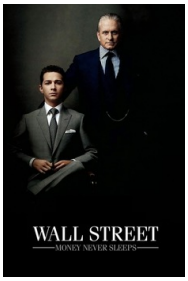
Query: "A New York stockbroker refuses to cooperate in a large securities fraud case involving corruption on Wall Street, corporate banking world and mob infiltration. Based on Jordan Belfort's autobiography."					
Model	**No.1**	**No.2**	**No.3**	**No.4**	**No.5**
Base	 *Frontline: Money Power & Wall Street*	*Wall Street(1987)*	*Wolves of Wall Street*	*Goldman Sachs: The Bank That Runs the World*	*Wall Street(1929)*
Fintuned	 ***The Wolf of Wall Street***	 *Wall Street: Money Never Sleeps*	*Madoff: The Monster of Wall Street*	*Wolves of Wall Street*	*The Wizard of Lies*

Table 14: Case study showing image retrieval results for an text query from two models.The movie titles displayed below each image correspond to the posters shown. Entries in **bold** indicate the ground truth.

	Model	**No.**	**Result**
Query:	**Base**	1	A young man and a woman on a boat.
		2	A portrait of a young couple.
		3	Two young men and a sultry divorcee flee their drab hometown existence, taking to the road in a bubblegum pink chevrolet.
		4	Popeye drives up to take Olive for a ride, but Bluto in his much fancier car does what he can to spoil their jaunt.
		5	Various glamorous fashions are modeled by two "friends" including hat with large feather trim.
	Finetuned	1	When Barbie's estranged stepfather Joe tries to quash her romance with young beau Ken, the fur flies.
		2	**Barbie and Ken are having the time of their lives in the colorful and seemingly perfect world of Barbie Land. However, when they get a chance to go to the real world, they soon discover the joys and perils of living among humans.**
		3	After a long and failed marriage, Barbie wants to get far away from her husband.
		4	The toys throw Ken and Barbie a Hawaiian vacation in Bonnie's room.
		5	Barbie and her sisters take off on another exciting, global adventure to visit their friend Ken at his summer internship at a beautiful and exotic coral reef.

Table 15: Case study showing text retrieval results for an image query from two models. Entries in **bold** indicate the ground truth.

Query: "Did truman ever find out his life was a show"

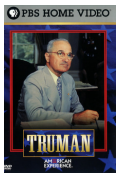
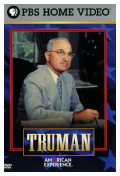
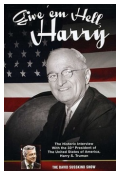
Rank	Base Model		Finetuned (Ours)	
	Text	Image	Text	Image
No.1	He was a farmer, a businessman, an unknown politician who suddenly found himself president. Of all the men who had held the highest office, Harry Truman was the least prepared, but would prove to be a surprise.		**Truman Burbank is the star of The Truman Show, a 24-hour-a-day reality TV show that broadcasts every aspect of his life without his knowledge. His entire life has been an unending soap opera for consumption by the rest of the world. And everyone he knows, including his wife and his best friend is really an actor, paid to be part of his life.**	
No.2	The making of 'The Truman Show'		Biographical account of America's President for the latter part of WWII. Shows Truman's rise from small-town nobody to leader of the USA, his decision to use the Atomic Bomb against Japan, and subsequent election as the US' post-war President.	
No.3	At his Long Island beach house, and on the occasion of the publication of his masterful nonfiction novel In Cold Blood, reporter Karen Dennison interviews celebrated writer Truman Capote, who displays his exuberant personality, makes witty jokes, shares his thoughts on writing, reflects on various aspects of the book and, in a sweet and endearing voice, reads and explains some of its highlights.		At his Long Island beach house, and on the occasion of the publication of his masterful nonfiction novel In Cold Blood, reporter Karen Dennison interviews celebrated writer Truman Capote, who displays his exuberant personality, makes witty jokes, shares his thoughts on writing, reflects on various aspects of the book and, in a sweet and endearing voice, reads and explains some of its highlights.	
No.4	Biographical account of America's President for the latter part of WWII. Shows Truman's rise from small-town nobody to leader of the USA, his decision to use the Atomic Bomb against Japan, and subsequent election as the US' post-war President.		He was a farmer, a businessman, an unknown politician who suddenly found himself president. Of all the men who had held the highest office, Harry Truman was the least prepared, but would prove to be a surprise.	
No.5	**Truman Burbank is the star of The Truman Show, a 24-hour-a-day reality TV show that broadcasts every aspect of his life without his knowledge. His entire life has been an unending soap opera for consumption by the rest of the world. And everyone he knows, including his wife and his best friend is really an actor, paid to be part of his life.**		In 1961, David Susskind conducted a series of interviews with former President Harry Truman in Truman's hometown of Independence, Missouri. After picking Truman up at his home to take him to the Truman Presidential Library for the interviews over a number of days.	

Table 16: Comparative case study of Hybrid Retrieval ($T \rightarrow T\&I$).Entries in **bold** indicate the ground truth.

Query: I want to do farmwork in the game , can you recommend one.				
Rank	**Base Model**		**Finetuned (Ours)**	
	Text	**Image**	**Text**	**Image**
No.1	The ultimate farming experience! From the creators of Avatar Farm comes Farm Together, the ultimate farming experience! Start from scratch, with a small plot, and end with a huge farm that extends further than the eye can see! Grow your farm Grow crops, plant trees, take care of the animals, and much more.....		Whether you specialise in grain, fruits or even maintaining your own dairy farm, in Farming World you have the ultimate choice on how you shape your business. With dozens of seed types to choose from and an extensive stock market listing, you'll have to adapt to become profitable. With plenty of seeds growing at different seasons you'll need to stock up so you can sell when produce is at a high demand.....	
No.2	Welcome to AnimalFarmland Official QQ chat group. About the Game Leisure Farming Click to help plants grow and earn money Unlock Buildings with rich effects Cute and bouncy animal villagers helps work.		Immerse yourself in the farming world of Farm Manager World! Build your agricultural empire in exciting new locations around the world, cultivate exotic plants, breed animals, utilize crop rotation and fertilizers to care for the soil, and trade resources with friends.....	
No.3	You can manage crops on your farm, sell them, and talk to people. And you can meet various creatures through fishing. And you can explore all the places in the world.		CHECK OUR NEW GAME Discord About the Game Start with a small plot of land and sell your crops at your market stall. Over time, you will acquire vast expanses of farmland, allowing you to reap bountiful harvests that you can then sell to purchase anything you desire.....	
No.4	Casual games! Take care of plants and animals, craft and sell them, add some fun elements, and grow your ranch! Free Resources - Growth Click on the blocks to get food and water for free, drag to give them to animals and plants, and let them grow! Growth - Production You can also use crops to feed animals and obtain by-products, which can be directly dragged to the left to sell. Sort out - Processing There are tools that can help you quickly sort out produce or process animals into meat.		Welcome to Family Farm 2023, the ultimate farming simulation game! As a player, you will become a true farmer and own an entire island, where you are the boss. You can crop, breed, build and more, using your imagination to create a thriving farm on a beautiful island. This game offers a wide range of features, including over 22 crops and flowers such as rice, corn, potatoes, sugar cane, roses, lavender, strawberries, mulberries, watermelons, carrots, purple cabbage, cucumbers, and peanuts.....	
No.5	In Bakery Simulator every recipe is inspired by the real ones! Check the cookbook, use a database of original recipes, and learn to bake several dozen types of bread, buns, and more. Be precise - otherwise, your goods will not be properly baked. Choose your target. Stores need different kinds of baked goods. Bread and buns are just not enough – some expect croissants, bagels, or muffins. Browse daily lists of orders, choose them and commit to regular deliveries. Raise your reputation and trust with the stores by making the deliveries on time. Find the best ingredients.....		Stardew Valley is an open-ended country-life RPG! You've inherited your grandfather's old farm plot in Stardew Valley. Armed with hand-me-down tools and a few coins, you set out to begin your new life. Can you learn to live off the land and turn these overgrown fields into a thriving home? It won't be easy. Ever since Joja Corporation came to town, the old ways of life have all but disappeared. The community center, once the town's most vibrant hub of activity, now lies in shambles. But the valley seems full of opportunity.....	

Table 17: Comparative case study of Hybrid Retrieval ($T \rightarrow T\&I$).Note that due to space constraints, only excerpts of the lengthy textual descriptions are displayed in the table.