VLoRA: Adapting Vision-Language Models for Multimodal Recommendation via Two-Stage Low-Rank Adaptation

Anonymous authors
Paper under double-blind review

Abstract

Learning item representations that jointly capture content semantics and collaborative-filtering (CF) signals is central to sequential recommendation. Existing text-based encoders ignore visual information, while generic vision-language models (VLMs) are not explicitly aligned with user interaction patterns. We present VLoRA, a parameter-efficient framework that adapts a decoder-only VLM into a CF-aware multimodal item encoder while keeping the backbone frozen. VLoRA follows a two-stage design. Collaborative Alignment Pre-training (CAP) first injects CF signals through causal next-item generation over multimodal interaction sequences. Recommendation Task Fine-tuning (RTF) then switches to full-visibility attention and refines item-level representations using multimodal masked token prediction, dual-view self-contrastive learning, content-token pooling, and representation preservation regularization. The resulting catalogue embeddings are context-independent, can be extracted offline, and can be reused by different downstream sequential recommenders without online VLM inference. Only 0.043% of the 2B backbone parameters are trained. Across six downstream datasets, two sequential recommenders, and four ranking metrics, VLoRA ranks first in 41 of 48 settings and second in the remaining seven. Same-backbone controls further show that multimodal-aware adaptation improves recommendation quality, while the benefits of visual input and sequential over interleaved optimization remain dataset-dependent. These results support VLoRA as an efficient and reusable multimodal representation framework for sequential recommendation.

1 Introduction

Sequential recommendation predicts a user's future interactions by learning from their historical behavior sequences (Kang & McAuley, 2018; Hidasi et al., 2016; Chen et al., 2026). Central to this task is building high-quality item representations that capture both content semantics and collaborative filtering (CF) signals (He et al., 2025; Ren & Huang, 2025). Traditional sequential recommenders assign unique identifiers to items and learn representations solely from co-occurrence patterns in interaction trajectories (Kang & McAuley, 2018; Hidasi et al., 2016; Yuan et al., 2020). While effective in data-rich settings, these ID-based representations are inherently domain-specific and fail to generalize to unseen items or new domains (Hou et al., 2024; Chen et al., 2026). Moreover, the reliance on interaction frequency makes ID embeddings susceptible to the cold-start problem, where newly introduced items lack sufficient co-occurrence statistics to form meaningful representations.

Recent advances in large language models (LLMs) (Tie et al., 2025) have motivated a growing body of work on learning item embeddings from textual metadata (Devlin et al., 2019; Li et al., 2024; Yang et al., 2025; Liu et al., 2025), and recommendation-oriented methods further fine-tune LLMs on interaction data to bridge general semantics with user behavior (Ren & Huang, 2025; He et al., 2025). By encoding titles, descriptions, and attributes into dense vectors, text-based approaches alleviate cold-start issues and enable cross-domain transfer through shared linguistic semantics. Despite these successes, text-only representations remain fundamentally limited: in e-commerce, fashion, and media domains, visual appearance plays a decisive role in user decisions, yet it is entirely absent from textual embeddings.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

A natural remedy is to incorporate visual information via vision-language models (VLMs). General-purpose multimodal encoders such as CLIP (Radford et al., 2021) and VLM2Vec (Jiang et al., 2024) produce joint vision-language representations but are oblivious to CF dynamics, and recent recommendation-oriented adaptations (Zhang et al., 2025; Kim et al., 2026; Wu et al., 2025) mostly rely on single-stage supervised fine-tuning (SFT). Adapting a VLM for recommendation introduces three design requirements. First, the encoder must acquire collaborative signals while retaining an attention regime suitable for item-level embeddings. Second, multimodal masking must account for the very different roles of visual and textual content; our asymmetric choice is an empirical design and is not claimed to isolate token-count effects. Third, adaptation should remain close enough to the pretrained semantic space to limit harmful representation drift under sparse CF supervision.

To address these challenges, we propose **VLoRA** (**V**ision-**L**anguage **Lo**w-**R**ank **A**daptation), a two-stage post-training framework that converts a frozen decoder-only VLM into a CF-aware multimodal embedding model. In the first stage, *Collaborative Alignment Pre-training* (CAP), VLoRA fine-tunes the VLM via autoregressive next-item prediction over user interaction sequences, injecting CF signals into the representation space. In the second stage, *Recommendation Task Fine-tuning* (RTF), VLoRA switches to bidirectional attention and refines item-level embeddings through three complementary objectives: *Multimodal Masked Next Token Prediction* (MMNTP) selectively masks text tokens while keeping visual tokens visible to strengthen cross-modal fusion; *Dual-View Self-Contrastive Learning* constructs two views of the same item via independent masking patterns and optimizes an InfoNCE loss (Oord et al., 2018) for view-invariant representations; and *Representation Preservation Regularization* (RPR) imposes a decaying L2 penalty against a fixed frozen-backbone reference to retain pretrained semantics. Both stages train LoRA adapters (Hu et al., 2022); Stage 1 adapts the last four decoder layers and Stage 2 expands the LoRA scope to the last eight while retaining the overlapping adapters. We treat this 4-to-8-layer schedule as an empirical adaptation choice rather than a separately novel mechanism, and analyze alternative layer schedules in Appendix G.3. Our contributions are:

- A VLM-specific two-stage framework that converts a decoder-only VLM into a CF-aware, *context-independent item encoder*, decoupling multimodal representation learning from downstream sequence modeling so that catalogue embeddings can be extracted once and reused across recommenders;
- Controlled evidence on multimodal-aware masking, pooling, contrastive refinement, and semantic-retention regularization, including the settings where visual input or sequential ordering does not help; and
- Evaluation on six downstream datasets and two recommenders, with same-backbone controls, statistical tests, efficiency accounting, and explicit scope boundaries.

2 Related Work

Large models as embedding models. Pre-trained language models have advanced text embedding from task-specific encoders to general-purpose backbones. Bidirectional encoders such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), together with LLM-based embedders including BGE (Li et al., 2024), GTE (Li et al., 2023), LLM2Vec (BehnamGhader et al., 2024), and NV-Embed (Lee et al., 2024), show that large language models can be repurposed as competitive embedders through bidirectional attention, masked next-token prediction (MNTP), and contrastive post-training. On the multimodal side, CLIP (Radford et al., 2021) established contrastive image-text pre-training, while recent VLMs (Bai et al., 2025; Wang et al., 2025; Liu et al., 2023) enable deeper cross-modal fusion. VLM2Vec (Jiang et al., 2024), VLM2Vec-V2 (Meng et al., 2025), and Qwen3-VL-Embedding (Li et al., 2026) further adapt VLMs into general multimodal embedders. These models primarily target semantic retrieval and are not explicitly aligned with CF dynamics, whereas VLoRA aligns multimodal representations with user behavior for recommendation.

Large models for recommendation. Recent work also adapts large models as recommendation-specific encoders. Text-based methods such as EasyRec (Ren & Huang, 2025), LLMEmb (Liu et al., 2025), and SLIM (Wang et al., 2024) incorporate collaborative information through contrastive tuning, attribute augmentation, or generative distillation. LLM2Rec (He et al., 2025) further introduces a two-stage framework that injects CF signals autoregressively before converting the LLM into a bidirectional item embedder, but these methods do not directly model visual content.

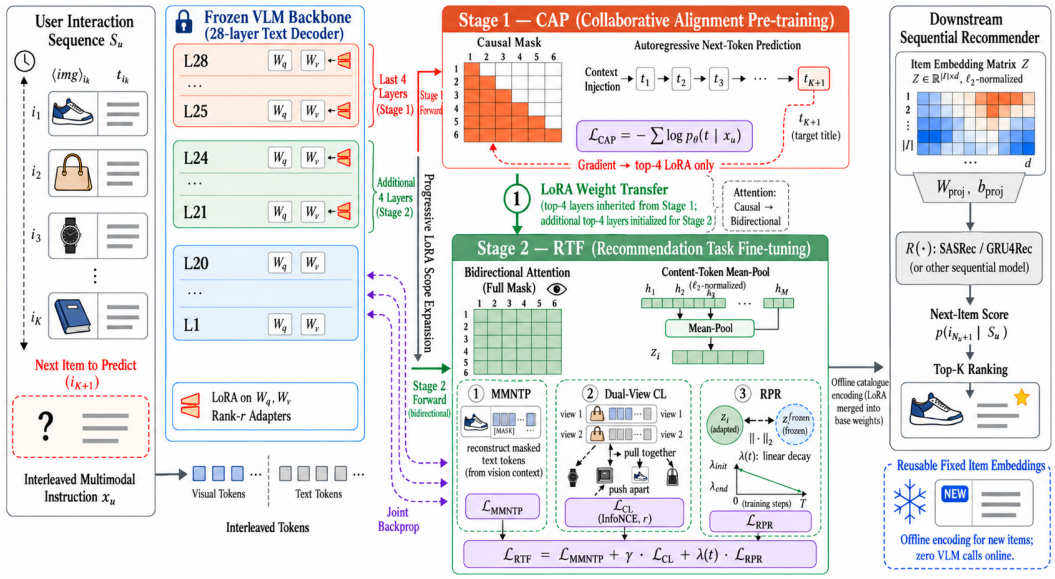


Figure 1: The VLoRA framework. **(Left)** Interleaved inputs $\mathbf{x}_u$ feed a frozen VLM with LoRA on $\mathbf{W}_q, \mathbf{W}_v$ under progressive LoRA scope expansion. **(Stage 1: CAP)** Top-layer LoRAs are trained autoregressively with $\mathcal{L}_{\text{CAP}}$. **(Stage 2: RTF)** LoRA extends to deeper layers, attention turns bidirectional, and ℓ_2-normalized mean-pooled embeddings $\mathbf{z}_i$ are optimized via $\mathcal{L}_{\text{MMNTP}}$, $\mathcal{L}_{\text{CL}}$, $\mathcal{L}_{\text{RPR}}$. **(Right)** A merged-LoRA offline pass produces $\mathbf{Z}$ for downstream recommenders.

Multimodal approaches include VLM$_{\text{Prompt}}$ (Pomo et al., 2025), NoteLLM-2 (Zhang et al., 2025), and VLM2Rec (Kim et al., 2026). VLM2Rec uses sequence-target contrastive learning with WPCL and CRTR to mitigate modality collapse and can also extract item embeddings for downstream recommendation. VLoRA instead differs in its adaptation recipe: causal generative CAP injects collaborative signals from multimodal interaction sequences, followed by bidirectional item-level RTF with modality-asymmetric MMNTP, dual-view self-contrastive learning, content pooling, and frozen-reference regularization. An extended discussion is provided in Appendix A.

3 Methodology

We present **VLoRA**, a parameter-efficient post-training framework that transforms a general-purpose decoder-only VLM into a collaborative filtering (CF)-aware multimodal embedding model. The overall architecture is illustrated in Figure 1.

3.1 Problem Formulation

Let $\mathcal{U}$ and $\mathcal{I}$ denote the sets of users and items. Each item $i \in \mathcal{I}$ carries a textual description $\mathbf{t}_i$ and visual content $\mathbf{v}_i$; user u's chronologically ordered interaction sequence is $S_u = [i_1, \ldots, i_{N_u}]$. *Sequential recommendation* predicts the next item i_{N_u+1} given S_u. Following recent paradigms (He et al., 2025; Hou et al., 2024), we replace learnable ID embeddings with content-derived representations from an embedding model $E(\cdot)$, i.e., $\mathbf{z}_i = E(\mathbf{t}_i, \mathbf{v}_i) \in \mathbb{R}^d$, and feed them into a downstream sequential recommender $R(\cdot)$ that scores candidate items as $p(i_{N_u+1} \mid S_u) = R(\mathbf{z}_{i_1}, \ldots, \mathbf{z}_{i_{N_u}}; \mathbf{z}_i)$. VLoRA targets the embedding model $E(\cdot)$, which must jointly encode semantic understanding and collaborative signals.

3.2 Motivation and Progressive LoRA Adaptation

Converting an autoregressive VLM into an item encoder requires both collaborative context injection and bidirectional representation refinement. CAP and RTF use different objective families and attention regimes, so VLoRA assigns each its natural regime in two sequential stages. Our budget-

matched interleaved control shows that both objective families contribute, but the benefit of sequential ordering is dataset-dependent (Appendix G.2).

We attach LoRA (Hu et al., 2022) modules to the query and value projections ($\mathbf{W}_q, \mathbf{W}_v$) of the frozen VLM. Stage 1 adapts the last 4 of 28 decoder layers; Stage 2 extends adaptation to the last 8, retaining overlapping weights and initializing newly added adapters. For rank 16 and Qwen3-VL-2B dimensions, this trains 917,504 parameters, or 0.043% of the backbone. The schedule is motivated by stability and efficiency; the complete schedule grid in Appendix G.3 does not show that 4-to-8 is uniformly optimal.

3.3 Stage 1: Collaborative Alignment Pre-training (CAP)

Stage 1 injects CF signals into the VLM by modeling multimodal interaction sequences. We format the user's recent K interactions into an interleaved multimodal instruction:

$$\mathbf{x}_u = \big[\langle \texttt{img} \rangle_{i_1}, \mathbf{t}_{i_1}, \ldots, \langle \texttt{img} \rangle_{i_K}, \mathbf{t}_{i_K}\big], \tag{1}$$

where $\langle \texttt{img} \rangle_{i_k}$ and $\mathbf{t}_{i_k}$ denote the visual tokens and text tokens of item i_k, respectively. We train the model via autoregressive next-token prediction to generate the target item's title $\mathbf{t}_{i_{K+1}}$. The loss is computed solely on the target text tokens:

$$\mathcal{L}_{\text{CAP}} = -\sum_{(u,\, i_{K+1}) \in \mathcal{D}} \sum_{s=1}^{\ell_{i_{K+1}}} \log\, p_\theta\big(t_{i_{K+1},s} \mid \mathbf{x}_u,\, t_{i_{K+1},<s}\big), \tag{2}$$

where $\mathcal{D}$ contains only (user, target-item) pairs constructed from each in-domain training prefix. Validation and test interactions are never CAP targets or recommendation-training labels. At evaluation, titles and images are catalogue-side content; no held-out interaction label is exposed to the encoder. The three out-of-domain datasets are disjoint from CAP training.

3.4 Stage 2: Recommendation Task Fine-tuning (RTF)

Stage 2 refines the CF-enriched generative model into a bidirectional embedding model using a self-contained optimization strategy, leveraging three complementary objectives. RTF is trained only on item content from the same six source domains used by CAP; Sports, Baby, and Goodreads are excluded from both VLM-adaptation stages. Unlike CAP, RTF is item-level and does not use interaction targets or target-domain labels.

Bidirectional Transition and Content Pooling. We first replace the causal attention mask with a full-visibility mask ($\mathbf{M}_{ij} = 1$ for all valid tokens i, j) to allow holistic contextualization. To extract the unified item embedding $\mathbf{z}_i$, we perform ℓ_2-normalized mean-pooling over the final-layer hidden states of all *content tokens* (visual patches and text tokens, excluding padding):

$$\mathbf{z}_i = \frac{1}{\|\bar{\mathbf{h}}_i\|_2}\bar{\mathbf{h}}_i, \quad \text{where } \bar{\mathbf{h}}_i = \frac{1}{m_i}\sum_{k=1}^{m_i} \mathbf{h}_{i,k}, \tag{3}$$

where m_i is the total number of content tokens.

Multimodal Masked Next Token Prediction. To bridge the causal-to-full-visibility transition, we extend MNTP (BehnamGhader et al., 2024; He et al., 2025) to multimodal inputs with *modality-asymmetric corruption*. For item i, a set of textual positions $\mathcal{M}_i$ is sampled and replaced by the masking placeholder to form the corrupted text $\widetilde{\mathbf{t}}_i$, while all visual patches remain visible. Under bidirectional attention, we retain the shifted language-model prediction rule of MNTP: the hidden state immediately preceding a masked textual target predicts its original token. The loss is averaged over masked textual targets:

$$\mathcal{L}_{\text{MMNTP}} = -\frac{1}{\sum_{i \in \mathcal{B}} |\mathcal{M}_i|} \sum_{i \in \mathcal{B}} \sum_{s \in \mathcal{M}_i} \log p_\theta\Big(t_{i,s} \mid \widetilde{\mathbf{t}}_i,\, \mathbf{v}_i\Big), \tag{4}$$

where the conditioning context contains the *entire jointly corrupted text* rather than removing only the current target position. This keeps the image intact as a cross-modal anchor while forcing reconstruction of missing textual semantics.

Dual-View Self-Contrastive Learning. We construct two stochastically augmented views of each item, $\mathbf{x}_i^{(1)}$ and $\mathbf{x}_i^{(2)}$, via independent text masking patterns (visual tokens unmasked), and optimize the InfoNCE loss over a mini-batch $\mathcal{B}$:

$$\mathcal{L}_{\text{CL}} = -\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \log \frac{\exp\big(\text{sim}(\mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)})/\tau\big)}{\sum_{j\in\mathcal{B}} \exp\big(\text{sim}(\mathbf{z}_i^{(1)}, \mathbf{z}_j^{(2)})/\tau\big)}, \tag{5}$$

where $\text{sim}(\cdot,\cdot)$ is cosine similarity and τ is the temperature. This dual-view design avoids false-positive noise and enforces robustness to incomplete metadata. A formal information-theoretic justification is provided in Appendix E.

Representation Preservation Regularization (RPR). To limit harmful drift from pretrained semantics, we penalize the ℓ_2 distance between the adapted embedding and a fixed, precomputed, L2-normalized frozen-backbone reference $\mathbf{z}_i^{\text{frozen}}$:

$$\mathcal{L}_{\text{RPR}} = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \big\|\bar{\mathbf{z}}_i - \mathbf{z}_i^{\text{frozen}}\big\|_2^2. \tag{6}$$

The regularization weight follows a linear decay schedule $\lambda(t) = \lambda_{\text{init}} - (\lambda_{\text{init}} - \lambda_{\text{end}})\frac{t}{T}$, prioritizing semantic retention in early training while permitting structural adaptation in later steps.

Both independently masked views share the same two forward passes. MMNTP is evaluated on each view and averaged. The contrastive loss is one-way: view 1 is the query and all view-2 embeddings in the batch form the candidate set. RPR uses $\bar{\mathbf{z}}_i = \text{Normalize}((\mathbf{z}_i^{(1)} + \mathbf{z}_i^{(2)})/2)$; it does not use an unmasked third view. The frozen reference is computed once before adaptation with the frozen backbone Φ_0 using the same full-visibility content-pooling rule in Eq. 3, then ℓ_2-normalized and kept fixed throughout both stages. Thus, RPR anchors the adapted multimodal geometry to a stationary pretrained semantic reference rather than to another trainable view. The complete Stage 2 objective is:

$$\mathcal{L}_{\text{RTF}} = \mathcal{L}_{\text{MMNTP}} + \gamma\mathcal{L}_{\text{CL}} + \lambda(t)\mathcal{L}_{\text{RPR}}, \tag{7}$$

3.5 Downstream Utilization

After RTF, LoRA matrices are merged and the catalogue is encoded offline by a bidirectional forward pass, producing $\mathbf{Z} \in \mathbb{R}^{|\mathcal{I}|\times 2048}$. Newly added items are likewise encoded offline; the reported 106.1 ms batch-size-1 latency is not serving latency. Following prior work (He et al., 2025; Kim et al., 2026), a linear adapter maps item vectors to the recommender hidden space. The offline VLoRA vectors $\mathbf{z}_i$ remain frozen; only the linear adapter and downstream recommender are optimized on the target recommendation task. After downstream training, the projected vectors can be cached, and online recommendation uses embedding lookup with zero VLM calls per request. Because the VLoRA vector for each item is fixed before downstream training, it is independent of user identity and sequence position.

4 Experiments

We conduct extensive experiments to answer the following research questions:

- **RQ1**: How does VLoRA perform compared to state-of-the-art text-based and multimodal baselines on in-domain recommendation tasks?
- **RQ2**: Can VLoRA generalize to unseen domains and even cross-platform datasets without further VLM adaptation?
- **RQ3**: How does each component of VLoRA contribute to the overall performance?
- **RQ4**: How does the choice of VLM backbone affect recommendation quality?

4.1 Experimental Setup

Datasets. We adapt VLoRA on six widely used Amazon Review source domains (Hou et al., 2024): **Games**, **Arts**, **Movies**, **Home**, **Electronics**, and **Tools**, with each item represented only by its product

Table 1: Statistics of the six source domains used for VLM adaptation in both CAP and RTF. #Interactions denotes retained interactions after preprocessing.

Dataset	#Items	#Interactions
Games	9,517	153,221
Arts	12,454	132,566
Movies	13,190	136,471
Home	33,478	256,001
Electronics	20,150	197,984
Tools	19,964	159,969
Total	108,753	1,036,212

Table 2: The statistics of the in-domain and out-of-domain datasets used in downstream sequential recommendation.

Dataset	#Users	#Items	#Interactions
Games	15,322	9,517	153,221
Arts	13,257	12,454	132,566
Movies	13,647	13,190	136,471
Sports	13,674	13,952	136,740
Baby	9,790	6,837	97,899
Goodreads	10,639	4,550	158,347

title and first image. Both CAP and RTF use these six source domains. For downstream evaluation, **Games**, **Arts**, and **Movies** serve as in-domain benchmarks using the same splits as VLM adaptation, while **Sports** and **Baby** (Hou et al., 2024) and the cross-platform **Goodreads** (Wan & McAuley, 2018; Wan et al., 2019) are used to evaluate out-of-domain generalization. No item content, interaction, or label from Sports, Baby, or Goodreads is used for VLM adaptation. We apply 5-core filtering, cap sequence length at 10, and use leave-one-out splitting. Dataset statistics are reported in Table 1 and Table 2, with full preprocessing details in Appendix C.

Evaluation Metrics. We use NDCG@K and Recall@K, $K \in \{10, 20\}$, under full-catalogue ranking. Tables 3 and 4 report means over three downstream-recommender seeds. Aggregate and representative paired tests using these runs appear in Appendix G.7.

Baselines. We compare VLoRA against 12 baselines spanning two broad categories. **(1) General embedding models** include three text-based methods: BERT (Devlin et al., 2019), BGE (Li et al., 2024), and LLM2Vec (BehnamGhader et al., 2024), as well as three multimodal methods: CLIP (Radford et al., 2021), VLM2Vec (Jiang et al., 2024), and VLM_{Emb} (Qwen3-VL-Embedding-2B) (Li et al., 2026). **(2) Recommendation-specific models** include four text-based methods: EasyRec (Ren & Huang, 2025), LLMEmb (Liu et al., 2025), LLM2Rec (He et al., 2025), and SLIM (Wang et al., 2024), as well as two multimodal methods: $\text{VLM}_{\text{Prompt}}$ (Pomo et al., 2025) and NoteLLM-2 (Zhang et al., 2025).

Downstream Recommenders. We evaluate the generated embeddings by plugging them into two downstream sequential recommenders: **SASRec** (Kang & McAuley, 2018) and **GRU4Rec** (Hidasi et al., 2016). The VLoRA embeddings are frozen; only the recommender parameters and the linear adapter ($d \to 256$) are trained. This plug-and-play protocol evaluates whether the same frozen item embeddings remain effective across two distinct downstream recommender architectures. We leave detailed descriptions of baselines and downstream recommenders in Appendix F.

Implementation Details. VLoRA uses Qwen3-VL-2B-Instruct (Bai et al., 2025): 28 text-side layers, 16 attention heads, 8 KV heads, head dimension 128, and embedding dimension d=2,048. LoRA (Hu et al., 2022) (r=16, α=32) is applied to $\mathbf{W}_q$ and $\mathbf{W}_v$ in the selected layers. The last-eight-layer configuration contains exactly 917,504 trainable parameters (0.043% of the backbone). Each product image is resized to 448×448 pixels, and the interaction history is truncated to the most recent K=10 items. In Stage 1 (CAP), LoRA modules are attached to the last 4 decoder layers. The model is trained for 5,000 steps with AdamW (Loshchilov & Hutter, 2017) (learning rate 2×10^{-4}, cosine decay, effective batch size 128). In Stage 2 (RTF), LoRA is progressively extended to the last 8 layers, with the overlapping modules initialized from the Stage 1 checkpoint. RTF samples individual items only from the same six source-domain catalogues used by CAP; it uses no target-domain items or interaction labels. Training runs for 500 steps with learning rate 5×10^{-5}, cosine decay, and effective batch size 128 via GradCache (Gao et al., 2021). The contrastive temperature is τ=0.05, the contrastive loss weight is γ=1, and the RPR weight λ decays linearly from λ_{init}=0.1 to λ_{end}=0.01. The MMNTP masking ratio is set to 15% for text tokens. Unless otherwise stated, model-training experiments are conducted in bf16 mixed precision on 8×NVIDIA H100 GPUs; the dedicated efficiency profiling in Appendix G.8 uses 2×H100 as reported there.

Table 3: Performance comparison of different embedding methods under in-domain datasets. R is short for Recall, N is short for NDCG, and %Improv. indicates the relative improvement compared to the strongest baselines.

Backbone: GRU4Rec

Method	Games				Arts				Movies			
	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20
(1) Text-based General Embedding Models												
BERT	0.0365	0.0184	0.0573	0.0236	0.0363	0.0191	0.0559	0.0240	0.0243	0.0126	0.0383	0.0160
BGE	0.0491	0.0261	0.0760	0.0329	0.0413	0.0221	0.0632	0.0276	0.0379	0.0187	0.0587	0.0239
LLM2Vec	0.0583	0.0306	0.0846	0.0374	0.0509	0.0289	0.0729	0.0350	0.0394	0.0201	0.0601	0.0252
(2) Multimodal-based General Embedding Models												
CLIP	0.0633	0.0357	0.0909	0.0422	0.0390	0.0208	0.0583	0.0257	0.0536	**0.0304**	0.0732	**0.0354**
VLM2Vec	0.0636	0.0349	0.0924	0.0420	0.0439	0.0234	0.0648	0.0287	0.0498	0.0276	0.0705	0.0328
VLM_{Emb}	0.0654	0.0364	0.0915	0.0429	0.0404	0.0212	0.0623	0.0267	0.0506	0.0285	0.0714	0.0337
(3) Text-based Recommendation-specific Models												
EasyRec	0.0450	0.0235	0.0700	0.0298	0.0436	0.0232	0.0643	0.0284	0.0356	0.0180	0.0551	0.0229
LLMEmb	0.0587	0.0321	0.0837	0.0387	0.0516	0.0297	0.0726	0.0350	0.0403	0.0211	0.0582	0.0253
LLM2Rec	0.0674	0.0372	0.0944	0.0440	0.0637	0.0396	0.0856	0.0452	0.0453	0.0234	0.0641	0.0277
SLIM	0.0632	0.0349	0.0916	0.0417	0.0458	0.0295	0.0865	0.0388	0.0504	0.0266	0.0698	0.0313
(4) Multimodal-based Recommendation-specific Models												
$\text{VLM}_{\text{Prompt}}$	0.0633	0.0350	0.0910	0.0419	0.0548	0.0292	0.0821	0.0361	0.0515	0.0288	0.0697	0.0338
NoteLLM-2	0.0632	0.0342	0.0919	0.0413	0.0568	0.0295	0.0870	0.0371	0.0472	0.0254	0.0668	0.0304
VLoRA	**0.0723**	**0.0398**	**0.1007**	**0.0464**	**0.0724**	**0.0418**	**0.1002**	**0.0482**	**0.0542**	0.0295	**0.0778**	0.0349
%Improv.	7.27%	6.99%	6.67%	5.45%	13.66%	5.56%	15.17%	6.64%	1.12%	-2.96%	6.28%	-1.41%

Backbone: SASRec

Method	Games				Arts				Movies			
	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20
(1) Text-based General Embedding Models												
BERT	0.0585	0.0311	0.0863	0.0381	0.0650	0.0405	0.0869	0.0460	0.0447	0.0240	0.0646	0.0290
BGE	0.0733	0.0410	0.1022	0.0483	0.0748	0.0475	0.1006	0.0540	0.0626	0.0350	0.0847	0.0406
LLM2Vec	0.0799	0.0438	0.1112	0.0518	0.0830	0.0546	0.1085	0.0611	0.0714	0.0414	0.0944	0.0473
(2) Multimodal-based General Embedding Models												
CLIP	0.0697	0.0399	0.0991	0.0473	0.0468	0.0256	0.0691	0.0312	0.0634	0.0373	0.0822	0.0421
VLM2Vec	0.0730	0.0418	0.1038	0.0495	0.0520	0.0287	0.0769	0.0349	0.0634	0.0383	0.0853	0.0438
VLM_{Emb}	0.0731	0.0415	0.1058	0.0497	0.0505	0.0279	0.0733	0.0337	0.0606	0.0363	0.0812	0.0414
(3) Text-based Recommendation-specific Models												
EasyRec	0.0647	0.0357	0.0926	0.0428	0.0658	0.0395	0.0929	0.0463	0.0528	0.0278	0.0739	0.0331
LLMEmb	0.0878	0.0526	0.1165	0.0599	0.0865	0.0626	0.1166	0.0709	0.0710	0.0420	0.0904	0.0466
LLM2Rec	0.0934	0.0562	0.1250	0.0641	0.0997	0.0687	0.1232	0.0747	0.0760	0.0463	0.0966	0.0517
SLIM	0.0715	0.0408	0.1027	0.0483	0.0730	0.0403	0.1069	0.0489	0.0642	0.0378	0.0842	0.0428
(4) Multimodal-based Recommendation-specific Models												
$\text{VLM}_{\text{Prompt}}$	0.0710	0.0403	0.1011	0.0482	0.0718	0.0391	0.1047	0.0480	0.0637	0.0377	0.0833	0.0425
NoteLLM-2	0.0732	0.0414	0.1041	0.0492	0.0756	0.0414	0.1095	0.0499	0.0646	0.0383	0.0849	0.0434
VLoRA	**0.0999**	**0.0570**	**0.1471**	**0.0678**	**0.1180**	**0.0784**	**0.1559**	**0.0875**	**0.0860**	**0.0484**	**0.1136**	**0.0559**
%Improv.	6.96%	1.42%	17.68%	5.77%	18.36%	14.12%	26.54%	17.14%	13.16%	4.54%	17.60%	8.12%

4.2 Main Results (RQ1-RQ2)

Tables 3 and 4 contain 48 dataset-recommender-metric cells. VLoRA ranks first in 41 and second in seven; the trailing cells are concentrated in Movies with GRU4Rec, Baby with SASRec, and Goodreads with GRU4Rec, with relative gaps from -1.08% to -4.58%. The Baby-GRU4Rec N@10 difference of $+0.5\%$ is the only case described as an approximate tie. Against NoteLLM-2 specifically, VLoRA is higher in all 48 cells, with a median relative gain of 28.5%.

In-domain, the largest margins occur for Arts with SASRec, including R@20 of 0.1559 versus 0.1232 for LLM2Rec. On Movies with GRU4Rec, CLIP remains higher on both NDCG metrics; these are retained as negative results rather than attributed to an unmeasured property of the dataset. Out-of-domain, VLoRA is strongest on Sports with SASRec and transfers to Goodreads, while Baby has narrow or negative multimodal margins. The controlled modality study in Appendix G.5 therefore supports a dataset-dependent conclusion: images help when they provide complementary item semantics, but gains are not guaranteed.

Table 4: Performance comparison of different embedding methods under out-of-domain datasets. R is short for Recall, N is short for NDCG, and %Improv. indicates the relative improvement compared to the strongest baselines.

Backbone: GRU4Rec

Method	Sports				Baby				Goodreads			
	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20
(1) Text-based General Embedding Models												
BERT	0.0335	0.0183	0.0499	0.0224	0.0111	0.0050	0.0252	0.0086	0.0851	0.0412	0.1226	0.0506
BGE	0.0489	0.0281	0.0685	0.0330	0.0252	0.0131	0.0364	0.0158	0.1072	0.0585	0.1517	0.0697
LLM2Vec	0.0716	0.0501	0.0874	0.0541	0.0273	0.0149	0.0391	0.0175	0.1268	0.0707	0.1774	0.0834
(2) Multimodal-based General Embedding Models												
CLIP	0.0262	0.0133	0.0405	0.0168	0.0272	0.0140	0.0435	0.0180	0.1446	0.0925	0.1863	0.1031
VLM2Vec	0.0266	0.0136	0.0408	0.0171	0.0332	0.0177	0.0496	0.0219	0.1612	**0.1025**	**0.2122**	**0.1154**
VLM_{Emb}	0.0251	0.0128	0.0398	0.0164	0.0304	0.0160	0.0470	0.0201	0.1551	0.0982	0.2056	0.1109
(3) Text-based Recommendation-specific Models												
EasyRec	0.0492	0.0270	0.0674	0.0315	0.0207	0.0105	0.0275	0.0122	0.0951	0.0477	0.1364	0.0581
LLMEmb	0.0764	0.0520	0.0930	0.0558	0.0270	0.0145	0.0408	0.0181	0.1308	0.0756	0.1798	0.0868
LLM2Rec	0.0894	0.0684	0.1024	0.0718	0.0353	0.0196	0.0502	0.0229	0.1399	0.0814	0.1875	0.0938
SLIM	0.0697	0.0366	0.1117	0.0467	0.0282	0.0142	0.0431	0.0180	0.1451	0.0868	0.1884	0.0985
(4) Multimodal-based Recommendation-specific Models												
$\text{VLM}_{\text{Prompt}}$	0.0716	0.0369	0.1125	0.0470	0.0274	0.0141	0.0432	0.0180	0.1447	0.0915	0.1883	0.1023
NoteLLM-2	0.0688	0.0361	0.1106	0.0464	0.0283	0.0144	0.0429	0.0181	0.1453	0.0861	0.1901	0.0974
VLoRA	**0.0961**	**0.0699**	**0.1174**	**0.0747**	**0.0362**	**0.0197**	**0.0512**	**0.0232**	**0.1615**	0.1008	0.2099	0.1131
%Improv.	7.49%	2.19%	4.36%	4.04%	2.55%	0.51%	1.99%	1.31%	0.19%	-1.66%	-1.08%	-1.99%

Backbone: SASRec

Method	Sports				Baby				Goodreads			
	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20
(1) Text-based General Embedding Models												
BERT	0.0860	0.0649	0.1017	0.0689	0.0114	0.0050	0.0232	0.0080	0.1479	0.0858	0.1929	0.0972
BGE	0.0974	0.0736	0.1141	0.0778	0.0428	0.0250	0.0569	0.0286	0.1445	0.0813	0.1972	0.0945
LLM2Vec	0.1158	0.0920	0.1310	0.0955	0.0601	0.0362	0.0772	0.0408	0.1549	0.0846	0.2034	0.0974
(2) Multimodal-based General Embedding Models												
CLIP	0.0323	0.0172	0.0478	0.0211	0.0345	0.0179	0.0512	0.0222	0.1617	0.1020	0.2099	0.1141
VLM2Vec	0.0352	0.0187	0.0525	0.0230	0.0382	0.0204	0.0552	0.0246	0.1866	0.1174	0.2383	0.1305
VLM_{Emb}	0.0337	0.0181	0.0484	0.0218	0.0302	0.0158	0.0462	0.0198	0.1728	0.1085	0.2241	0.1213
(3) Text-based Recommendation-specific Models												
EasyRec	0.0887	0.0627	0.1061	0.0671	0.0271	0.0154	0.0381	0.0182	0.1445	0.0825	0.1908	0.0941
LLMEmb	0.1200	0.1004	0.1340	0.1037	0.0717	0.0473	0.0863	0.0508	0.1475	0.0830	0.1958	0.0964
LLM2Rec	0.1275	0.1059	0.1398	0.1086	0.0754	**0.0546**	0.0925	**0.0586**	0.1642	0.0963	0.2141	0.1102
SLIM	0.1091	0.0610	0.1636	0.0755	0.0337	0.0171	0.0506	0.0214	0.1705	0.1047	0.2176	0.1169
(4) Multimodal-based Recommendation-specific Models												
$\text{VLM}_{\text{Prompt}}$	0.1098	0.0581	0.1626	0.0719	0.0339	0.0175	0.0510	0.0221	0.1649	0.1026	0.2166	0.1164
NoteLLM-2	0.1181	0.0635	0.1686	0.0775	0.0327	0.0167	0.0504	0.0211	0.1733	0.1056	0.2230	0.1182
VLoRA	**0.1564**	**0.1238**	**0.1790**	**0.1297**	**0.0779**	0.0521	**0.1012**	0.0575	**0.1983**	**0.1288**	**0.2527**	**0.1425**
%Improv.	22.67%	16.90%	6.17%	19.43%	3.32%	-4.58%	9.41%	-1.88%	6.27%	9.71%	6.04%	9.20%

4.3 Ablation Study (RQ3)

We ablate each component on Arts and Goodreads with SASRec as the recommender (Table 5).

Stage-wise. Under the current training configuration, removing Stage 1 (CAP) produces the largest observed degradation, showing that CAP contributes substantially to the final representation. Removing Stage 2 (RTF) also degrades performance on both datasets. Because RTF jointly changes the attention regime, pooling rule, and optimization objectives, this ablation supports the complete RTF stage rather than isolating bidirectional attention as the sole cause. The two stages therefore provide complementary empirical gains under the reported training budgets.

Component-wise. Within Stage 2, removing $\mathcal{L}_{\text{CL}}$ causes the largest component-level decrease (N@20 drops by 17.7% on Arts and 12.6% on Goodreads), supporting the contribution of dual-view self-contrastive learning. Removing $\mathcal{L}_{\text{RPR}}$ also degrades performance. The diagnostics in Appendix G.4 show greater cosine retention to the frozen reference together with better recommendation scores under RPR, supporting a semantic-drift interpretation rather than literal rank collapse. Replacing asymmetric with symmetric masking also hurts, although the current evidence does not isolate

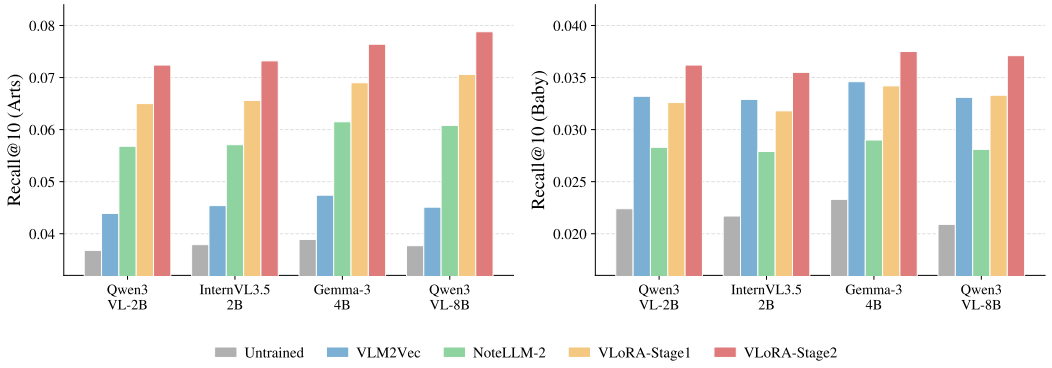


Figure 2: Backbone sensitivity on Arts (left) and Baby (right) with GRU4Rec (R@10). VLM2Vec and NoteLLM-2 are reimplemented on each corresponding backbone and compared with the untrained model and VLoRA after both stages.

modality length as the cause. Appendix G.3 further compares fixed, progressive, reverse, and unrestricted LoRA schedules.

4.4 Evaluation of Different VLM Backbones (RQ4)

To examine backbone effects, we evaluate VLoRA on four pre-trained VLMs spanning three families (Qwen3-VL, InternVL3.5, Gemma-3) and three scales (2B, 4B, 8B). For this study, VLM2Vec and NoteLLM-2 are reimplemented on each corresponding backbone rather than reusing the fixed checkpoints from the main benchmark. We compare the untrained model, the backbone-matched VLM2Vec and NoteLLM-2 variants, and VLoRA after Stage 1 and Stage 2, all evaluated with GRU4Rec (Figure 2).

Table 5: Ablation study of VLoRA with SASRec on Arts and Goodreads.

Variants	Arts		Goodreads	
	R@20	N@20	R@20	N@20
w/o Stage 1 CAP	0.1162	0.0620	0.2254	0.1218
w/o Stage 2 RTF	0.1400	0.0769	0.2381	0.1322
VLoRA	**0.1559**	**0.0875**	**0.2527**	**0.1425**
w/o $\mathcal{L}_{\text{CL}}$	0.1291	0.0720	0.2309	0.1246
w/o $\mathcal{L}_{\text{RPR}}$	0.1423	0.0798	0.2403	0.1338
w/o Asym. Masking	0.1462	0.0809	0.2418	0.1351

Across all four backbones, VLoRA after Stage 2 attains the highest reported result within each backbone setting on both datasets. On Arts, Stage 1 already exceeds the two plotted baselines and Stage 2 further improves performance. On Baby, Stage 1 can trail VLM2Vec, whereas Stage 2 improves over Stage 1 across the plotted settings. Performance is not monotonic with nominal parameter count across model families, indicating that both backbone choice and dataset matter. These results support the portability of the two-stage adaptation recipe across the tested VLM families rather than a general claim that larger VLMs are always better.

5 Conclusion and Limitations

VLoRA adapts a decoder-only VLM into a CF-aware multimodal item encoder through causal collaborative alignment followed by full-visibility embedding refinement. Experiments support the integrated design: multimodal-aware adaptation improves results on Arts and Goodreads, while RPR yields higher cosine retention to the frozen reference and better recommendation performance. Across six downstream datasets, VLoRA ranks first in 41 of 48 settings and second in seven; aggregate exact signed-rank tests are significant for three of four metrics, with the same pattern after Holm correction.

Several limitations remain. Visual gains depend on the dataset and item group, and sequential optimization is not uniformly better than budget-matched interleaving. Catalogue extraction is offline but scales with catalogue size, image resolution, and refresh frequency. The study considers one title and image per item, sequences of at most 10 items, 2B-8B backbones, and two downstream recommenders; item embeddings remain static rather than user- or position-conditioned. VLM2Rec can also produce downstream item embeddings, but a direct controlled comparison is not included; empirical superiority claims are therefore restricted to the evaluated baselines.

AI USE STATEMENT

In this work, we used generative AI tools for language editing, wording refinement, and translation assistance. We have not used generative AI tools for generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implementing methods, cleaning or reformatting datasets, or interpreting results, and qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for manuscript formatting and consistency checks. We have reviewed all AI-assisted work. In particular, the authors manually reviewed the AI-assisted language and formatting edits and checked them against the original manuscript, experimental records, reported tables, and cited references to ensure that the technical meaning and reported results were not altered. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the information needed to reproduce VLoRA and the reported experiments throughout the paper and appendices. The model formulation, training objectives, and two-stage adaptation procedure are described in Section 3, with the complete training pipeline summarized in Algorithm 1. Dataset construction, preprocessing, chronological splitting, and evaluation protocols are documented in Appendix C. Baseline implementations, checkpoints, and downstream recommender settings are detailed in Appendix F, while additional controlled experiments report the LoRA schedule, representation diagnostics, statistical tests, and efficiency measurements. The assumptions and derivations underlying the theoretical analysis are provided in the corresponding appendix sections. Together, these materials specify the data processing, model configuration, optimization procedure, evaluation protocol, and analysis used to obtain the reported results.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. LLM2Vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961*, 2024.

Xiaoqing Chen, Zhitao Li, Weike Pan, Qiang Yang, and Zhong Ming. A survey on multi-behavior sequential recommendation. *Science China Information Sciences*, 69(3):131101, 2026.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Luyu Gao, Yunyi Zhang, Jiawei Han, and Jamie Callan. Scaling deep contrastive learning batch size under memory limited setup. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pp. 316–321, 2021.

Yingzhi He, Xiaohao Liu, An Zhang, Yunshan Ma, and Tat-Seng Chua. LLM2Rec: Large language models are powerful embedding models for sequential recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 896–907, 2025.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *International Conference on Learning Representations (ICLR)*, 2016.

Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*, 2024.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. *International Conference on Learning Representations (ICLR)*, 2022.

Ziyan Jiang, Rui Meng, Xinyi Yang, Semih Yavuz, Yingbo Zhou, and Wenhu Chen. VLM2Vec: Training vision-language models for massive multimodal embedding tasks. *arXiv preprint arXiv:2410.05160*, 2024.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pp. 197–206. IEEE, 2018.

Junyoung Kim, Woojoo Kim, Jaehyung Lim, Dongha Kim, and Hwanjo Yu. Vlm2rec: Resolving modality collapse in vision-language model embedders for multimodal sequential recommendation. *arXiv preprint arXiv:2603.17450*, 2026.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. NV-Embed: Improved techniques for training LLMs as generalist embedding models. *arXiv preprint arXiv:2405.17428*, 2024.

Chaofan Li, MingHao Qin, Shitao Xiao, Jianlyu Chen, Kun Luo, Yingxia Shao, Defu Lian, and Zheng Liu. Making text embedders few-shot learners. *arXiv preprint arXiv:2409.15700*, 2024.

Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, et al. Qwen3-VL-Embedding and Qwen3-VL-Reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv preprint arXiv:2601.04720*, 2026.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023.

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

Qidong Liu, Xian Wu, Wanyu Wang, Yejing Wang, Yuanshao Zhu, Xiangyu Zhao, Feng Tian, and Yefeng Zheng. Llmemb: Large language model can be a good embedding generator for sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 12183–12191, 2025.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Rui Meng, Ziyan Jiang, Ye Liu, Mingyi Su, Xinyi Yang, Yuepeng Fu, Can Qin, Zeyuan Chen, Ran Xu, Caiming Xiong, et al. VLM2Vec-V2: Advancing multimodal embedding for videos, images, and visual documents. *arXiv preprint arXiv:2507.04590*, 2025.

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

Claudio Pomo, Matteo Attimonelli, Danilo Danese, Fedelucio Narducci, and Tommaso Di Noia. Do recommender systems really leverage multimodal content? a comprehensive analysis on multimodal representations for recommendation. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 2377–2387, 2025.

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International conference on machine learning*, pp. 5171–5180. PMLR, 2019.

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

Xubin Ren and Chao Huang. EasyRec: Simple yet effective language models for recommendation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 17728–17743, 2025.

Guiyao Tie, Zeli Zhao, Dingjie Song, Fuyang Wei, Rong Zhou, Yurou Dai, Wen Yin, Zhejian Yang, Jiangyue Yan, Yao Su, et al. A survey on post-training of large language models. *arXiv preprint arXiv:2503.06072*, 2025.

Mengting Wan and Julian McAuley. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM conference on recommender systems*, pp. 86–94, 2018.

Mengting Wan, Rishabh Misra, Ndapandula Nakashole, and Julian McAuley. Fine-grained spoiler detection from large-scale review corpora. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2605–2610, 2019.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

Yuling Wang, Changxin Tian, Binbin Hu, Yanhua Yu, Ziqi Liu, Zhiqiang Zhang, Jun Zhou, Liang Pang, and Xiao Wang. Can small language models be good reasoners for sequential recommendation? In *Proceedings of the ACM Web Conference 2024*, pp. 3876–3887, 2024.

Binrui Wu, Shisong Tang, Fan Li, Bing Han, Chang Meng, Jingyu Xiao, and Jiechao Gao. Aligning and balancing id and multimodal representations for recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 5029–5038, 2025.

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

Fajie Yuan, Xiangnan He, Haochuan Jiang, Guibing Guo, Jian Xiong, Zhezhao Xu, and Yilin Xiong. Future data helps training: Modeling future contexts for session-based recommendation. In *Proceedings of the web conference 2020*, pp. 303–313, 2020.

Chao Zhang, Haoxin Zhang, Shiwei Wu, Di Wu, Tong Xu, Xiangyu Zhao, Yan Gao, Yao Hu, and Enhong Chen. NoteLLM-2: Multimodal large representation models for recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 2815–2826, 2025.

Appendix of paper "VLoRA: Adapting Vision-Language Models for Multimodal Recommendation via Two-Stage Low-Rank Adaptation"

A Extended Related Work

A.1 Large Models as Embedding Models

Text embedding models. Bidirectional encoders such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) have long served as standard text embedding backbones. More recently, the BGE family (Li et al., 2024) shows that LLM-based embedders can be strengthened through in-context learning, while GTE (Li et al., 2023) develops a generalist text embedding model effective across retrieval, classification, and clustering. LLM2Vec (BehnamGhader et al., 2024) demonstrates that decoder-only LLMs can be converted into strong text encoders through bidirectional attention, Masked Next Token Prediction (MNTP), and unsupervised contrastive learning. NV-Embed (Lee et al., 2024) further improves LLM embeddings through latent attention pooling and two-stage contrastive training.

Multimodal embedding models. CLIP (Radford et al., 2021) established contrastive image-text pre-training and a shared space for visual and textual representations, but its dual-encoder design limits cross-modal interaction during encoding. Recent VLMs (Bai et al., 2025; Wang et al., 2025; Liu et al., 2023) enable deeper multimodal fusion. Built on this trend, VLM2Vec (Jiang et al., 2024) adapts VLMs into universal multimodal embedding models through instruction-guided contrastive training, and VLM2Vec-V2 (Meng et al., 2025) further validates LoRA-based adaptation with last-token pooling. Qwen3-VL-Embedding (Li et al., 2026) adopts a multi-stage pipeline, from large-scale contrastive pre-training to reranker distillation, to learn unified embeddings for text, images, and video. These models target semantic retrieval rather than recommendations from user behavior.

A.2 Large Models for Recommendation

LLM-based recommendation embeddings. Recent work increasingly adapts large language models from general-purpose semantic encoders into recommendation-aware representation models. EasyRec (Ren & Huang, 2025) combines contrastive learning with collaborative language-model tuning to inject behavioral information into text representations and support text-based zero-shot recommendation. LLMEmb (Liu et al., 2025) strengthens LLM-generated item embeddings through attribute-level augmentation and contrastive alignment, with particular attention to items that have relatively sparse interaction histories. SLIM (Wang et al., 2024) uses teacher-generated preference reasoning and step-by-step distillation to transfer recommendation knowledge into a smaller text-based model, whose learned representations encode item-attribute and preference information. LLM2Rec (He et al., 2025) proposes a two-stage adaptation framework: Collaborative Supervised Fine-Tuning first injects CF signals through autoregressive sequence modeling, after which bidirectional attention, Masked Next Token Prediction, and item-level contrastive learning are used to obtain recommendation-oriented item embeddings. These methods demonstrate that collaborative signals can be transferred into language-model representations, but their item content remains text-only.

VLM-based recommendation. Multimodal recommendation methods extend this line by incorporating visual content into large-model representations. $\text{VLM}_{\text{Prompt}}$ (Pomo et al., 2025) studies the zero-shot use of VLMs for multimodal recommendation through structured prompting without recommendation-specific parameter updates. NoteLLM-2 (Zhang et al., 2025) focuses on multimodal note recommendation and enhances visual information through compressed visual tokens and multimodal representation learning. VLM2Rec (Kim et al., 2026) adapts VLMs for multimodal sequential recommendation through sequence-target contrastive learning and introduces Weak-modality Penalized Contrastive Learning (WPCL) and Cross-modal Relational Topology Regularization (CRTR) to mitigate modality collapse; its learned model can also be used to extract item embeddings for downstream recommendation. VLoRA is therefore distinguished not by the mere ability to produce reusable item vectors, but by its adaptation procedure and objective design. It first injects collaborative information through causal generative CAP over multimodal interaction sequences, and then performs bidirectional item-level RTF with modality-asymmetric MMNTP, dual-view self-contrastive learning, content-token pooling, and frozen-reference representation preservation regularization. This design

explicitly separates collaborative signal acquisition from item-level multimodal refinement while retaining a catalogue-side embedding interface for downstream sequential recommenders.

B Training Pipeline Summary

The comprehensive training procedure for VLoRA is detailed in Algorithm 1.

Algorithm 1 VLoRA Training Pipeline

Require: Pre-trained VLM Φ_0, interaction dataset $\mathcal{D}$, multimodal content $\{(\mathbf{t}_i, \mathbf{v}_i)\}_{i\in\mathcal{I}}$
Ensure: CF-aware multimodal item embedding matrix $\mathbf{Z}$

% Pre-processing
Compute and cache frozen references $\{\mathbf{z}_i^{\text{frozen}}\}$ once with Φ_0 under full-visibility attention and content pooling (Eq. 3); keep them fixed thereafter.
% Stage 1: Collaborative Alignment Pre-training (CAP)
Inject LoRA matrices to the top L_1 self-attention layers of Φ_0.
for each global step from 1 to T_1 **do**
    Sample a mini-batch of user sequences from $\mathcal{D}$.
    Construct multimodal instructions $\mathbf{x}_u$ (Eq. 1).
    Update LoRA parameters by minimizing $\mathcal{L}_{\text{CAP}}$ (Eq. 2).
end for
% Stage 2: Recommendation Task Fine-tuning (RTF)
Expand LoRA to the top L_2 layers ($L_2 > L_1$); retain weights for overlapping layers.
Transition attention from causal to full bidirectional visibility.
for each global step t from 1 to T_2 **do**
    Construct dual augmented views $\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}$ with text masking.
    Calculate $\mathcal{L}_{\text{MMNTP}}$ using modality-asymmetric masking (Eq. 4).
    Calculate $\mathcal{L}_{\text{CL}}$ via self-contrastive InfoNCE loss (Eq. 5).
    Calculate regularizer $\mathcal{L}_{\text{RPR}}$ against $\{\mathbf{z}_i^{\text{frozen}}\}$ (Eq. 6).
    Update LoRA parameters by minimizing $\mathcal{L}_{\text{RTF}}$ (Eq. 7).
end for
% Downstream Extraction
Merge learned LoRA weights: $\mathbf{W} \leftarrow \mathbf{W}_0 + \frac{\alpha}{r}\mathbf{BA}$.
Extract and return $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_{|\mathcal{I}|}]^\top$ via bidirectional forward pass.

C Dataset Preprocessing Details

This appendix provides the full dataset description and preprocessing pipeline summarized in Section 4.1.

C.1 Pre-training Datasets

We pre-train VLoRA on six widely-used Amazon Review datasets released by Hou et al. (2024), spanning a broad range of product categories: *Video Games* (**Games**), *Arts, Crafts, and Sewing* (**Arts**), *Movies and TV* (**Movies**), *Home and Kitchen* (**Home**), *Electronics* (**Electronics**), and *Tools and Home Improvement* (**Tools**). Each item in these datasets is associated with a product title (textual modality) and one or more product images (visual modality). Following prior work (He et al., 2025; Kim et al., 2026), we use the *first* product image of each item as the visual input to ensure a deterministic and reproducible pipeline. Per-dataset statistics (#items, #interactions) after preprocessing are reported in Table 1.

C.2 Downstream Sequential Recommendation Datasets

To rigorously evaluate both in-domain fit and out-of-domain generalization, we use two groups of downstream datasets.

In-domain datasets. We train and test the downstream recommender on three of the pre-training datasets: **Games**, **Arts**, and **Movies**, using the *same* data splits as in pre-training. This setting directly measures the quality of embeddings on domains to which the VLM has been adapted.

Out-of-domain datasets. To further probe generalization to unseen domains and cross-platform scenarios, we additionally include three datasets that are *disjoint* from the pre-training data:

- **Sports** and **Baby** (Hou et al., 2024): two additional Amazon categories (*Sports and Outdoors*, *Baby Products*) whose items never appear in any pre-training split. These datasets probe intra-platform, cross-category transfer.
- **Goodreads** (Wan & McAuley, 2018; Wan et al., 2019): a cross-platform book recommendation corpus collected from the Goodreads website. Each book provides a title and a cover image, allowing us to evaluate whether VLoRA's multimodal representations generalize beyond the Amazon e-commerce ecosystem to a different platform with fundamentally different visual content distribution (book covers vs. product photographs).

Per-dataset statistics (#users, #items, #interactions) are reported in Table 2.

C.3 Preprocessing Pipeline

We apply an identical preprocessing pipeline to every dataset described above to ensure fair cross-method comparison.

5-core filtering. We iteratively remove users and items with fewer than 5 interactions until both the user and item frequency distributions stabilize above the 5-core threshold. This mitigates extreme sparsity on the cold-user / cold-item tails while preserving long-tail coverage above the minimum reliability bar.

Sequence construction and length truncation. For each remaining user u, we sort the retained interactions chronologically and first assign the leave-one-out targets. Model inputs are then constructed only from interactions that occur before the corresponding target. If the available prefix contains more than 10 interactions, we retain its most recent 10 as input context; later validation/test interactions are never exposed in an earlier context. The full retained chronology is used only to determine target assignment. This $K{=}10$ cap matches the interaction window used in Equation (1) of Section 3.3 and controls the multimodal compute and memory cost of interleaving multiple 448×448 images.

Modality fields. For every retained item i, we extract exactly two fields: the product title as textual content $\mathbf{t}_i$, and the first product image (or book cover, on Goodreads) as visual content $\mathbf{v}_i$. Category, description, and other structured metadata are not used by VLoRA or supplied to multimodal baselines in our benchmark. Items whose title is empty or whose image cannot be downloaded are discarded before constructing the interaction sequences.

D Prompt Template Figure

E Design Rationale for Dual-View Self-Contrastive Learning

We restate two standard information-theoretic facts that motivate the design without treating them as a separate theoretical contribution: the InfoNCE lower bound and a data-processing relation between view agreement and item identity. Keeping visual tokens visible is a construction choice that preserves the observed image in both views; it does not by itself prove better downstream ranking or identify token length as the causal factor.

E.1 Notation and Setup

Let $\mathcal{I}$ denote the item set. For each item $i \in \mathcal{I}$, the multimodal content is $\mathbf{x}_i = (v_i, t_i)$ where v_i and t_i are the visual and textual tokens. The two views are generated by stochastic text masking:

$$\mathbf{x}_i^{(1)} = (v_i, \tilde{t}_i^{(1)}), \quad \mathbf{x}_i^{(2)} = (v_i, \tilde{t}_i^{(2)}), \tag{8}$$

Stage 1: Collaborative Alignment Pre-training (CAP)

User:
User interaction history:
<image_1> Title$_1$
<image_2> Title$_2$
⋮
<image_K> Title$_K$
Next item title:
Assistant:
Title$_{K+1}$

Stage 2: Recommendation Task Fine-tuning (RTF)

User:
Represent this item for recommendation:
<image> Title

Figure 3: Prompt templates used in VLoRA. Stage 1 (CAP) constructs multimodal interaction histories for autoregressive next-item prediction. Stage 2 (RTF) uses a single-item prompt for contrastive embedding learning.

where $\tilde{t}_i^{(k)} \sim \mathcal{M}(t_i, p)$ is obtained by independently masking each text token with probability p (the visual tokens v_i remain intact in both views). Let f_θ denote the encoder, and $\mathbf{Z}^{(k)} = f_\theta(\mathbf{X}^{(k)})$ be the ℓ_2-normalized embedding of the random augmented view. We use I to denote a random variable over item identities, and $\mathbf{X}$, $\mathbf{X}^{(k)}$, $\mathbf{Z}^{(k)}$ to denote the corresponding random variables (with lowercase $\mathbf{x}_i$, $\mathbf{x}_i^{(k)}$, $\mathbf{z}_i^{(k)}$ for realizations).

E.2 Mutual Information Lower Bound

Proposition 1 (InfoNCE as MI Lower Bound)**.** *Let $\mathbf{Z}^{(1)}$ and $\mathbf{Z}^{(2)}$ be the random embedding variables of the two views. For a mini-batch $\mathcal{B}$ of $|\mathcal{B}|$ items drawn i.i.d., the InfoNCE loss in Eq. (5) satisfies:*

$$I(\mathbf{Z}^{(1)}; \mathbf{Z}^{(2)}) \geq \log |\mathcal{B}| - \mathbb{E}[\mathcal{L}_{\mathrm{CL}}], \tag{9}$$

where $I(\cdot;\cdot)$ denotes mutual information and the expectation is taken over the data distribution and stochastic masking.

Proof. This follows directly from the standard InfoNCE bound (Oord et al., 2018; Poole et al., 2019). The contrastive loss can be written as:

$$\mathcal{L}_{\mathrm{CL}} = -\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \log \frac{\exp(\mathrm{sim}(\mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)})/\tau)}{\sum_{j \in \mathcal{B}} \exp(\mathrm{sim}(\mathbf{z}_i^{(1)}, \mathbf{z}_j^{(2)})/\tau)}. \tag{10}$$

Define $h(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}) = \exp(\mathrm{sim}(\mathbf{z}^{(1)}, \mathbf{z}^{(2)})/\tau)$ as the critic function. By Theorem 1 of (Oord et al., 2018), for any critic function h:

$$I(\mathbf{Z}^{(1)}; \mathbf{Z}^{(2)}) \geq \log |\mathcal{B}| - \mathbb{E}[\mathcal{L}_{\mathrm{CL}}]. \tag{11}$$

Thus, minimizing $\mathbb{E}[\mathcal{L}_{\mathrm{CL}}]$ raises the InfoNCE lower bound on the mutual information $I(\mathbf{Z}^{(1)}; \mathbf{Z}^{(2)})$. □

Proposition 1 motivates the dual-view objective by encouraging agreement between independently masked views. A lower contrastive loss raises a lower bound on shared-view mutual information, but does not by itself prove that the true mutual information increases or that all item-relevant information is preserved.

E.3 From View Agreement to Identity Preservation

We now relate the mutual information between views to the information about item identity.

Proposition 2 (Identity Preservation via Data Processing)**.** *Under the dual-view generation process, the two views are conditionally independent given the item identity, forming the Markov chain:*

$$\mathbf{Z}^{(1)} \leftarrow \mathbf{X}^{(1)} \leftarrow I \rightarrow \mathbf{X}^{(2)} \rightarrow \mathbf{Z}^{(2)}. \tag{12}$$

By the data processing inequality (DPI):

$$I(\mathbf{Z}^{(1)}; \mathbf{Z}^{(2)}) \leq I(\mathbf{Z}^{(1)}; I) \leq I(\mathbf{X}^{(1)}; I) \leq H(I), \tag{13}$$

where $H(I) = \log |\mathcal{I}|$ under a uniform prior.

Proof. Because the two masking patterns are drawn independently, $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ are conditionally independent given I, i.e., $\mathbf{X}^{(1)} \perp \mathbf{X}^{(2)} \mid I$. This conditional independence extends to the embeddings: $\mathbf{Z}^{(1)} \perp \mathbf{Z}^{(2)} \mid I$ (since $\mathbf{Z}^{(k)}$ is a deterministic function of $\mathbf{X}^{(k)}$). Therefore, $\mathbf{Z}^{(1)} - I - \mathbf{Z}^{(2)}$ forms a Markov chain. Applying the DPI directly:

$$I(\mathbf{Z}^{(1)}; \mathbf{Z}^{(2)}) \leq I(\mathbf{Z}^{(1)}; I). \tag{14}$$

The second inequality $I(\mathbf{Z}^{(1)}; I) \leq I(\mathbf{X}^{(1)}; I)$ follows from the DPI applied to the Markov chain $I \rightarrow \mathbf{X}^{(1)} \rightarrow \mathbf{Z}^{(1)}$. The final bound $I(\mathbf{X}^{(1)}; I) \leq H(I)$ holds by the property of mutual information. By symmetry, the same inequalities hold for $\mathbf{Z}^{(2)}$. □

Combining Propositions 1 and 2, we obtain:

Corollary 1 (Identity-related Interpretation of the Contrastive Objective)**.** *Combining Propositions 1 and 2 gives:*

$$\log |\mathcal{B}| - \mathbb{E}[\mathcal{L}_{\mathrm{CL}}] \leq I(\mathbf{Z}^{(1)}; \mathbf{Z}^{(2)}) \leq \min_{k} I(I; \mathbf{Z}^{(k)}) \leq H(I). \tag{15}$$

Accordingly, lowering $\mathbb{E}[\mathcal{L}_{\mathrm{CL}}]$ raises a lower bound on shared-view information, while the data-processing inequalities relate that shared information to item identity. This provides a design motivation for identity-preserving representations, but it is not a guarantee that the true identity mutual information, discriminability, or downstream recommendation performance must increase monotonically.

E.4 Asymmetric-Masking Rationale

Remark 1. *Each augmented view is $\mathbf{X}^{(k)} = (V, \widetilde{T}^{(k)})$, so the complete observed image V is shared by construction while textual masks vary. This preserves visual content across the positive pair and supplies textual augmentation. The equal-rate perturbation diagnostic in Appendix G.6 shows larger embedding changes under visual masking, but because equal-rate masking changes both the number and modality of perturbed tokens, it does not uniquely attribute the effect to sequence-length asymmetry or predict recommendation accuracy.*

The InfoNCE bound and data-processing argument explain why view agreement can promote identity-discriminative embeddings. RPR separately constrains drift toward a fixed pretrained reference. Both are design rationales; downstream support comes from the controlled ablations.

F Baselines and Downstream Sequential Recommenders

This section provides detailed descriptions of all baseline embedding models and downstream sequential recommenders used in our experiments.

(1) Text-based General Embedding Models.

- **BERT** (Devlin et al., 2019) is a bidirectional Transformer encoder pre-trained with masked language modeling and next sentence prediction. It captures rich contextual dependencies in text, producing general-purpose semantic representations. We use the pre-trained `bert-large-uncased` model.

- **BGE** (Li et al., 2024) is a state-of-the-art text embedding model built on a bidirectional Transformer architecture. Pre-trained on diverse datasets with multi-task learning, it excels at retrieval and semantic similarity tasks. We use the pre-trained `BAAI/bge-large-en-v1.5` model.
- **LLM2Vec** (BehnamGhader et al., 2024) adapts pre-trained decoder-only LLMs into effective text embedding models through three lightweight modifications: enabling bidirectional attention, masked next token prediction, and unsupervised contrastive learning. We implement it with Qwen3-1.7B.

(2) Multimodal-based General Embedding Models.

- **CLIP** (Radford et al., 2021) is a contrastive vision-language model pre-trained on 400M web-collected image-text pairs. It aligns visual and textual representations in a shared embedding space via a dual-encoder architecture. We use the pre-trained `openai/clip-vit-large-patch14` model and concatenate the image and text embeddings as the item representation.
- **VLM2Vec** (Jiang et al., 2024) converts a pre-trained vision-language model into a universal multimodal embedder through instruction-guided contrastive training with last-token pooling, enabling a single model to handle diverse embedding tasks across text, images, and interleaved multimodal inputs. We use the publicly released `TIGER-Lab/VLM2Vec-Full` checkpoint built on Phi-3.5-V.
- **VLM_{Emb}** (Li et al., 2026) (Qwen3-VL-Embedding-2B) is a multimodal embedding model built on Qwen3-VL through a multi-stage training pipeline combining large-scale contrastive pre-training with reranker distillation, achieving state-of-the-art performance on multimodal retrieval benchmarks. We use the official `Qwen/Qwen3-VL-Embedding-2B` checkpoint.

(3) Text-based Recommendation-specific Models.

- **EasyRec** (Ren & Huang, 2025) is a recommendation-specific text embedding model built on RoBERTa (Liu et al., 2019). It leverages contrastive learning combined with collaborative language model tuning to capture CF signals by aligning representations of user profiles and item descriptions, enabling text-based zero-shot recommendation. We use the pre-trained `hkuds/easyrec-roberta-large` checkpoint.
- **LLMEmb** (Liu et al., 2025) improves LLM-generated item embeddings through attribute-level augmentation and contrastive learning, enhancing semantic understanding, particularly for long-tail items with sparse interaction histories. We implement LLMEmb using Qwen3-1.7B as the backbone for consistency.
- **LLM2Rec** (He et al., 2025) is a two-stage recommendation embedding framework. In Stage 1, Collaborative Supervised Fine-Tuning (CSFT) trains the LLM autoregressively on mixed-domain user interaction sequences to inject CF signals. In Stage 2, the model switches to bidirectional attention with Masked Next Token Prediction and item-level contrastive learning to produce discriminative item embeddings. We use Qwen3-1.7B as its backbone for consistency with the other LLM-based baselines.
- **SLIM** (Wang et al., 2024) is a text-based recommendation-specific baseline that distills teacher-generated reasoning about user preferences into a smaller student model. Its representation pathway uses textual/item-attribute information rather than visual input, so we group it with the text-based recommendation methods in all result tables.

(4) Multimodal-based Recommendation-specific Models.

- **$\text{VLM}_{\text{Prompt}}$** (Pomo et al., 2025) investigates the zero-shot capability of large vision-language models for multimodal recommendation by extracting item embeddings through structured prompts without recommendation-specific fine-tuning. The original method can incorporate categorical attributes; in our controlled benchmark, we provide only the same title and first image available to the other multimodal methods, and use Qwen3-VL-2B-Instruct as the VLM backbone.
- **NoteLLM-2** (Zhang et al., 2025) is a multimodal large representation model designed for item-to-item recommendation. It proposes a multimodal In-Content Learning method to integrate visual and textual information within a unified LLM, combined with a late fusion mechanism

that separately produces visual and multimodal representations for contrastive training. We use Qwen3-VL-2B-Instruct as the VLM backbone.

Downstream Sequential Recommenders.

- **GRU4Rec** (Hidasi et al., 2016) utilizes GRU modules to capture sequential dependencies within user interaction sequences. It is trained to predict the next item in a user's sequence based on the preceding interactions, using cross-entropy loss as the optimization objective.
- **SASRec** (Kang & McAuley, 2018) is a Transformer-based sequential recommender that leverages causal self-attention to model long-range dependencies in user interaction histories, enhancing the accuracy of future interaction predictions. Cross-entropy loss is used as the optimization objective.

G Additional Controlled Experiments

This section reports controlled comparisons that isolate the adaptation recipe, stage ordering, LoRA-layer schedule, representation retention, visual input, and masking perturbations under the settings described in the main paper.

G.1 Same-backbone Multimodal-aware Adaptation

This control asks whether supplying image content is sufficient under a fixed VLM backbone, and whether subsequent multimodal-aware adaptation further improves the resulting item representations. All configurations use Qwen3-VL-2B-Instruct, the same recommendation-adaptation data, LoRA rank and layer schedule, update counts, and SASRec evaluation protocol. The factors changed cumulatively across rows are the input modality, masking rule, pooling rule, and RPR. The text-only recipe uses the same two-stage VLM adaptation scaffold but removes visual input; it is used as a text-oriented control rather than treated as an exact reproduction of LLM2Rec. In the "Uniform" condition, text and visual content use the same masking probability p, but not necessarily the same perturbation magnitude. Images are first processed normally by the vision encoder and visual merge/projection modules. Immediately before the merged visual content tokens enter the language model, eligible visual-token positions are sampled independently with probability p and the selected visual vectors are replaced by zeros. Their token slots, positional information, and attention visibility are retained, and image-boundary or other special visual tokens are never masked. Text tokens follow the MMNTP masking operation described in Section 3.4. The asymmetric condition keeps all visual content tokens intact and masks only text tokens.

Table 6: Stepwise control on the same Qwen3-VL-2B-Instruct backbone and training budget with SASRec.

Configuration	Input	Masking	Pooling	Arts		Goodreads	
				R@20	N@20	R@20	N@20
Text-only recipe	Text	Uniform	Last	.1398	.0775	.2407	.1342
+ image input	Text+image	Uniform	Last	.1456	.0803	.2449	.1361
+ asymmetric masking	Text+image	Text-only	Last	.1494	.0830	.2485	.1397
+ content mean pooling (w/o RPR)	Text+image	Text-only	Mean	.1423	.0798	.2403	.1338
VLoRA with RPR	Text+image	Text-only	Mean	**.1559**	**.0875**	**.2527**	**.1425**

Adding image input alone improves R@20 from .1398 to .1456 on Arts and from .2407 to .2449 on Goodreads, but the cumulative sequence is not monotonic. In particular, switching from last-token to mean pooling without RPR lowers R@20 from .1494 to .1423 on Arts and from .2485 to .2403 on Goodreads; adding RPR then yields the highest reported values, .1559 and .2527. Thus, this control supports the complete configuration but does not establish that mean pooling is generally superior to last-token pooling or that an isolated pooling-RPR synergy has been causally identified.

G.2 Budget-matched Interleaving and Extraction Controls

The interleaved control contains 5,000 CAP updates and 500 RTF updates within a single training run, executed as a fixed small cycle of ten CAP updates followed by one RTF update, repeated 500

times. This preserves the exact 10:1 update ratio while mixing both objective families throughout training. The effective batch size is 128 for both update types; CAP updates use causal attention and RTF updates use full visibility, with eight LoRA-equipped layers active throughout the interleaved run. Each objective maintains its own optimizer-step counter and follows the corresponding VLoRA cosine learning-rate schedule over 5,000 CAP steps or 500 RTF steps, respectively. The RTF loss coefficients γ and λ are kept the same as in the two-stage model rather than separately tuned. We use "budget-matched" here specifically to mean matched numbers of CAP/RTF updates and matched effective batch size; because the interleaved control uses eight LoRA layers throughout whereas VLoRA expands from four to eight, this term does not imply identical FLOPs or wall-clock cost. The full 4→8 VLoRA configuration achieves the strongest overall results in this table; the fixed-eight-layer control is retained to compare ordering under a common LoRA width. Both objective families

Table 7: Stage and interleaving controls with SASRec. The first block reports the original configurations; the second isolates ordering under a fixed-eight-layer schedule.

Variant	Arts R@20	Arts N@20	Goodreads R@20	Goodreads N@20
RTF only	.1162	.0620	.2254	.1218
CAP only	.1400	.0769	.2381	.1322
Budget-matched interleaved	.1485	.0826	.2441	.1367
VLoRA (4→8)	**.1559**	**.0875**	**.2527**	**.1425**
Interleaved, fixed 8	.1485	–	**.2441**	–
Two-stage, fixed 8	**.1533**	–	.2430	–

contribute. The full sequential 4→8 configuration is strongest overall across the reported Arts and Goodreads results. Under a fixed-eight-layer schedule, however, sequential ordering helps on Arts but not Goodreads, so the ordering effect is dataset-dependent and should not be interpreted as uniformly superior.

To test whether changing the extraction rule alone can reproduce the full RTF gain, we evaluate CAP-only weights under both their native causal/last-token extraction and the full-visibility/mean-pooling protocol used after RTF. This control does not by itself isolate the Stage-2 objective, because the complete model also differs in its learned weights and LoRA adaptation.

Table 8: Extraction-protocol control on Arts with SASRec.

Variant	Weights	Extraction	R@20	N@20
CAP only	CAP	causal, last token	.1400	.0769
CAP only	CAP	full visibility, mean	.1376	.0752
RTF only	RTF from base	full visibility, mean	.1162	.0620
Interleaved	all objectives	full visibility, mean	.1485	.0826
VLoRA	CAP then RTF	full visibility, mean	**.1559**	**.0875**

Applying full-visibility mean pooling to CAP-only weights changes Arts R@20 from .1400 to .1376 rather than reproducing the full VLoRA score of .1559. Thus, the full gain cannot be explained by switching the extraction rule alone.

G.3 LoRA Layer Schedule

We vary the number of decoder layers equipped with LoRA while keeping the LoRA rank, objectives, update budgets, and SASRec evaluation protocol fixed. The notation $a \to b$ denotes adapting the last a text-decoder layers in CAP and the last b text-decoder layers in RTF; the backbone itself remains frozen and LoRA is applied only to $\mathbf{W}_q$ and $\mathbf{W}_v$. "all" denotes all 28 text-decoder layers. For the reverse 8→4 schedule, the four LoRA modules removed from the Stage-2 adaptation set are merged after Stage 1 and then kept frozen, while Stage 2 continues adapting only the retained four layers.

The reported sensitivity is dataset-dependent: 4→4 is marginally highest on Arts, whereas 4→8 is highest on Goodreads and has the strongest average over the four reported metrics. The main experiments use the 4→8 configuration; this grid is reported to characterize sensitivity and is not used to claim optimization stability or universal optimality.

Table 9: LoRA schedule grid with SASRec.

Stage 1	Stage 2	Schedule	Arts R@20	Arts N@20	Goodreads R@20	Goodreads N@20
4	4	fixed narrow	**.1564**	**.0878**	.2476	.1389
8	8	fixed wide	.1533	.0855	.2430	.1317
4	8	progressive	.1559	.0875	**.2527**	**.1425**
8	4	reverse	.1516	.0849	.2424	.1338
all	all	unrestricted	.1531	.0863	.2462	.1367

G.4 Reference-retention and Geometry Diagnostics

We compare the frozen backbone, CAP-initialized RTF without RPR, and the full VLoRA configuration on Arts. Effective rank summarizes the spread of the singular-value spectrum of the item-embedding matrix, anisotropy summarizes concentration along its dominant singular direction, and "Cos. to frozen" is the mean item-wise cosine similarity to the fixed, precomputed, ℓ_2-normalized frozen reference. These are geometric diagnostics: because RPR directly penalizes deviation from the frozen reference, higher reference similarity is an expected consequence of the objective and is not, by itself, independent evidence of preserved semantic knowledge.

Table 10: Representation diagnostics on Arts. Geometry is reported together with downstream R@20; these quantities are not treated as independent semantic-quality guarantees.

Variant	Eff. rank	Anisotropy	Cos. to frozen	R@20
Frozen backbone	1040	.246	1.000	.1368
CAP + RTF without RPR	1180	.206	.563	.1423
VLoRA with RPR	**1235**	**.187**	**.708**	**.1559**

RPR is associated with higher cosine similarity to the frozen reference (.708 versus .563) and a higher recommendation score in this comparison. We therefore interpret RPR as constraining movement away from the pretrained reference while retaining useful task adaptation, rather than as independently proving that more semantic knowledge is preserved. Configurations that simultaneously change the RPR objective and learning rate are not used for causal interpretation of RPR, because those factors cannot be isolated.

G.5 Controlled Visual-input Comparison

We compare separately trained text-only and text+image configurations while keeping the backbone, recommendation-adaptation data, LoRA configuration, update schedule, pooling rule, and SASRec evaluator fixed. In the text+image condition, images are supplied throughout the VLM-adaptation and embedding-extraction pipeline; in the text-only condition, the corresponding image input is removed throughout. Subgroup membership is defined using training-side statistics only. We define *warm items* as items with at least 10 interactions in the training split. For title-length analysis, titles are tokenized with the model tokenizer using the title string only: prompt tokens, special tokens, and image tokens are excluded from the length count. *Short-title* items contain at most 10 title tokens, whereas *long-title* items contain at least 20 title tokens; items with 11-19 title tokens are excluded from the short/long comparison. Frequency and title-length groups are defined independently and may overlap. Here "warm" denotes relatively well-observed items rather than a complement to a strict cold-start set. Evaluation for a subgroup is restricted to test cases whose target item belongs to that subgroup, while ranking is still performed against the full catalogue. The reported quantity is

$$\Delta R@20 = \frac{R@20_{\text{text+image}} - R@20_{\text{text-only}}}{R@20_{\text{text-only}}} \times 100\%. \tag{16}$$

For Goodreads, the all-item row is the isolated image-input contrast from .2407 to .2449 ($+1.7\%$), matching the first two configurations in Table 6. Images help in some datasets and subgroups and hurt in others, so these analyses do not support a uniform catalogue-wide visual benefit.

Table 11: Relative R@20 change from text-only to text+image with SASRec. The controlled runs record relative changes.

Dataset or group	Relative R@20 change
Goodreads, all items	$+1.7\%$
Baby, all items	-1.5%
Arts, warm items	$+1.4\%$
Arts, short-title items	-4.0%
Arts, long-title items	$+5.6\%$

G.6 Masking Perturbation Diagnostic

To characterize sensitivity to modality corruption without retraining the recommender, we use the full Goodreads catalogue and the final VLoRA checkpoint. For each item and each nonzero masking condition, we independently sample 10 random masks, encode the resulting corrupted views, and compute cosine similarity to that item's unperturbed full-content embedding. The unperturbed reference embedding is computed once per item. We first average the 10 cosine values for each item and condition and then average these item-level means uniformly over the complete Goodreads catalogue. Text masking is applied only to title tokens. Visual masking uses the same merged-visual-token operation as the uniform control in Appendix G.1: selected merged visual content vectors are zeroed before entering the language model while token positions and attention visibility are retained. This is a representation diagnostic rather than a downstream recommendation experiment.

Table 12: Mean embedding cosine to the unperturbed full-content representation over the Goodreads catalogue. Each nonzero masking condition is averaged over 10 independently sampled masks per item.

Perturbation	Mask rate	Cosine
None	.00	1.000
Text only	.15	.978
Text only	.30	.960
Visual only	.15	.947
Visual only	.30	.916
Uniform over both	.15	.946

At matched nominal rates, visual masking produces the larger embedding change, while uniform masking is close to the visual-masking result. Because equal-rate masking changes both the number and type of perturbed tokens, this diagnostic does not identify token-count asymmetry as the unique cause; the downstream value of asymmetric masking is evaluated separately in the component ablation.

G.7 Statistical Analysis

Across 48 cells, VLoRA is first in 41 and second in seven. Two-sided exact Wilcoxon signed-rank tests across the 12 dataset-recommender pairs against the strongest baseline give $p = .0005$ (R@10), .0015 (R@20), .077 (N@10), and .016 (N@20). Applying Holm correction across the four primary metrics gives adjusted $p = .0020$, .0044, .077, and .032, respectively, preserving the same significance pattern. The statistical unit in this aggregate analysis is a dataset-recommender pair; the two recommenders evaluated on the same dataset are therefore related settings rather than independent datasets, so we treat the aggregate test as cross-setting evidence rather than a complete characterization of training variability. The representative per-setting tests in Table 13 are paired two-sided t-tests over the three matched downstream-recommender seeds. The table intentionally spans clear improvements, narrow differences, and negative differences; the complete result pattern is reported in Tables 3-4. With $n = 3$, a non-significant test establishes neither improvement nor equivalence. For all methods in these tests, the embedding checkpoint is held fixed across the three runs, so the reported seed variation quantifies downstream-training variability rather than variability from independently retraining the embedding backbone.

Table 13: Representative paired two-sided t-tests over three matched downstream-recommender seeds. For every row, the item-embedding checkpoint is fixed across the three runs; only downstream recommender training varies.

Setting	Metric	Comparator	Baseline	VLoRA	Rel. change	p
Arts, SASRec	R@20	LLM2Rec	.1232	.1559	$+26.5\%$	.018
Sports, SASRec	N@10	LLM2Rec	.1059	.1238	$+16.9\%$	.031
Baby, SASRec	N@10	LLM2Rec	.0546	.0521	-4.6%	.112
Movies, GRU4Rec	N@10	CLIP†	.0304	.0295	-3.0%	.168
Baby, GRU4Rec	N@10	LLM2Rec	.0196	.0197	$+0.5\%$	.721
Goodreads, GRU4Rec	R@20	VLM2Vec†	.2122	.2099	-1.1%	.392
Goodreads, GRU4Rec	N@10	VLM2Vec†	.1025	.1008	-1.7%	.284

G.8 Efficiency

Table 14: Efficiency profiling on 2×H100 in bf16 at 448×448. GPU-h denotes wall-clock hours multiplied by the number of GPUs. VLoRA peak memory is the measured Stage-2 peak; offline latency is measured at batch size 1 and is distinct from batched catalogue throughput.

Method	Training GPU-h		Peak/GPU	Catalogue GPU-h	Offline per-item encode latency
	Stage 1	Stage 2			
NoteLLM-2	1.70	–	∼43.5 GiB	.108	84.9 ms
VLM2Vec released	N/A	N/A	–	.108	84.9 ms
VLoRA	1.83	.096 (w/ GC)	11.1 GiB w/ GC / 44.7 GiB w/o GC	.134	106.1 ms

For VLoRA, Stage 2 uses GradCache with effective batch size 128; the reported .096 GPU-h corresponds to this GradCache-enabled configuration and therefore includes its recomputation overhead. Catalogue extraction covers 108,753 items at roughly 225 items/s/GPU; this batched throughput is reported separately from the batch-size-1 latency in the table and does not use GradCache. After extraction, all methods use frozen catalogue vectors and make zero VLM calls per online request. Relative to NoteLLM-2 in this profiling, VLoRA incurs moderately higher offline adaptation and catalogue-extraction cost. The VLM2Vec row evaluates a released checkpoint and therefore does not include the original cost of training that checkpoint.

H Baseline and Training-source Audit

This audit separates released checkpoints from methods trained or instantiated in our pipeline. For the main Tables 3-4, trainable implementations use the same six recommendation-adaptation datasets and preprocessing splits as VLoRA. Released checkpoints are evaluated as released and are not additionally adapted on these six datasets in our experiments; this statement concerns our experimental procedure and does not make claims about the checkpoints' full pretraining histories. All downstream comparisons use the same split, trainable projection, recommender configuration, and full-catalogue ranking protocol.

The self-trained text-only LLM baselines, including SLIM, use Qwen3-1.7B, whereas the Qwen-based VLM methods use Qwen3-VL-2B-Instruct. The nominal parameter counts therefore differ because the VLM additionally contains a vision encoder; we keep the model names and parameter counts explicit rather than treating the two backbones as parameter-matched. The checkpoint/backbone entries in this audit refer to the main benchmark tables. Figure 2 is a separate backbone-sensitivity study: VLM2Vec and NoteLLM-2 are reimplemented from their respective recipes on each of the four tested VLM backbones, so those variants should not be interpreted as the fixed released-checkpoint baselines listed above.

Table 15: Backbones and training sources. Trainable implementations use the same six CAP datasets and preprocessing splits; released checkpoints are evaluated as released. All downstream comparisons share splits, projection, recommender, and full-ranking evaluation.

Method	Checkpoint/backbone	Scale	Source	Adaptation/data
BERT	bert-large-uncased	.34B	released	none
BGE	bge-large-en-v1.5	.34B	released	none
EasyRec	easyrec-roberta-large	.36B	released	published
CLIP	clip-vit-large-patch14	.43B	released	none
LLM2Vec	Qwen3-1.7B	1.7B	self-trained	same six datasets
LLMEmb	Qwen3-1.7B	1.7B	self-trained	same six datasets
LLM2Rec	Qwen3-1.7B	1.7B	self-trained	same six datasets
SLIM	Qwen3-1.7B	1.7B	self-trained	same six datasets
VLMEmb	Qwen3-VL-Embedding-2B	2B	released	published
VLMPrompt	Qwen3-VL-2B-Instruct	2B	evaluated	prompting
NoteLLM-2	Qwen3-VL-2B-Instruct	2B	self-trained	full-parameter adaptation
VLM2Vec	Phi-3.5-V checkpoint	$\sim$4.2B	released	published
VLoRA	Qwen3-VL-2B-Instruct	2B	self-trained	LoRA, same six datasets