ChatGPT

Using Frozen CLIP Embeddings in a Two-Stage Lite-LLM4Rec Pipeline

Expected Advantages of Offline CLIP Fusion

Using **frozen CLIP image embeddings** for offline item encoding can yield a richer understanding of items, leading to multiple benefits:

- **Richer Item Representations:** By fusing visual features from CLIP with text metadata, the system captures more item characteristics than text alone. CLIP's aligned image-text space encodes semantic visual cues (color, style, etc.) that complement textual descriptions [1]. This enriched item representation can make recommendations more accurate and nuanced. For example, a recent study showed that integrating CLIP-derived visual and textual embeddings *"enriches item representations"* and *"significantly improves recommendation performance"* compared to text-only models [1].

- **Cold-Start Item Lift:** Incorporating image features is especially valuable for new or rarely-interacted items. In cold-start scenarios (e.g. a new product with only an image and description), a CLIP-based embedding gives the model meaningful signals about the item's category and style, whereas a traditional ID-based model would know nothing. Empirical results confirm a large boost in cold-start recommendation quality when using content embeddings – e.g. a multimodal recommender significantly outperformed a pure ID-based approach for **new items** on multiple datasets [2]. In other words, the frozen CLIP features allow the system to place cold-start items in the right "neighborhood" of the preference space, making them recommendable to the right users from day one.

- **Enhanced Recall@K and Diversity:** A multimodal item representation can improve overall Recall@K by retrieving relevant items that a text-only or ID-only model might miss. Visual features provide alternative similarity signals; items that are visually similar (even if textual features differ) can be surfaced, increasing the diversity of recommendations. Industry evidence from eBay's recommender system shows that a **multimodal embedding** recall can retrieve more relevant and diverse items than relying on a single modality, improving downstream metrics like click-through-rate (+15.9% CTR) and purchase rate (+31.5%) in A/B tests [3]. This indicates that combining image and text features helps **recall a broader set of high-quality items**, making recommendations more varied and engaging.

- **No Latency Penalty:** Crucially, this design does **not increase online latency**. All heavy fusion of image and text occurs offline. Item embeddings (augmented with CLIP image features) are pre-computed and stored. At query time, the recommender needs only to lookup these enriched embeddings and feed them into the Lite-LLM4Rec model or use them in similarity computations – just as it would with standard item vectors. There is no runtime image processing or CLIP inference. Thus, users benefit from better recommendations without slower response times. This aligns with

the goal of Lite-LLM4Rec to maintain efficient inference [4] [5]. In practice, computing and storing CLIP embeddings is feasible at scale: open implementations report processing 100+ million images in hours offline [6], and the added memory footprint (e.g. a 512-dimensional vector per item) is moderate for modern servers.

- **Potential Novel Insights:** With visual signals, the system might learn patterns that pure text or ID models would overlook – for instance, identifying that a user has a preference for products with certain visual styles or colors. This could increase personalization and even help unearth niche items (thus increasing **catalog coverage**). Additionally, the fused representation can be used to detect item anomalies (e.g. image-text mismatches or low-quality content), which can indirectly improve recommendation quality by filtering out bad data [7] [8].

Architectural Feasibility and Key Design Points

The two-stage Lite-LLM4Rec architecture is well-suited to incorporating CLIP-based multimodal fusion. Key architectural considerations include:

- **Two-Stage Pipeline Structure:** The pipeline is split into an **offline item encoding stage** and an **online recommendation stage**, which aligns with Lite-LLM4Rec's design for efficiency [4]. In stage 1 (offline), each item's metadata is processed: the item's image is passed through a **frozen CLIP image encoder**, and its text (title/description) through a text encoder (which could be CLIP's text encoder for convenience or another model). The resulting image and text embeddings (which CLIP produces in the *same vector space* by design) are then **fused into a single item embedding** [1] [9]. Fusion can be as simple as concatenation or weighted averaging followed by a linear projection, or a small multimodal encoder network – but this is all done offline. Stage 2 (online) uses these pre-computed item vectors in the Lite-LLM4Rec model to score or retrieve items for a given user. For example, Lite-LLM4Rec might encode a user's sequence of interactions (using a hierarchical LLM as described in the paper) and then compute scores via a dot-product between the user's latent vector and all item embeddings [4] [5]. The item projection head in Lite-LLM4Rec can directly incorporate the new multimodal item embeddings as its "vocabulary" for recommendation.

- **Embedding Alignment:** One cornerstone is that CLIP provides **pre-aligned multimodal embeddings**. The image and text features from CLIP live in a shared semantic space, which greatly simplifies fusion – the model has, in a sense, already learned how images and text correspond [1]. This means even a simple combination (like averaging the two) yields a meaningful joint representation of the item. Many prior multimodal recommenders had to train complex alignment models or use joint loss (e.g. eBay's team applied a Siamese network with triplet-loss to project image and text features into one space [10] [11]). By using CLIP's frozen encoders, we inherit a strong alignment out-of-the-box. We may still fine-tune a lightweight adapter or linear layer on top of the fused embedding during model training to better fit the recommendation task, but we **avoid training any expensive vision backbone**. This keeps the item tower relatively "lite" as well.

- **Hierarchical LLM and Metadata Fusion:** Lite-LLM4Rec introduced a hierarchical LLM structure to efficiently handle extensive item context [12]. In practice, this could mean the item's content (now a multimodal embedding) is used as part of the input to the LLM or as part of the item representation in the sequence. The fusion being offline means the LLM isn't processing raw images – it just consumes the fused embedding. One can implement this by expanding the item token embedding

matrix in the LLM with the multimodal vectors. For example, each item ID could be associated with a vector initialized (or fixed) as the CLIP-fused embedding instead of a random learned vector. Recent research on LLM-enhanced recommenders follows a similar idea: Wu *et al.* (2025) feed each item's image embedding and title embedding (from CLIP) into their model alongside ID embeddings [13] [14], and use attention mechanisms to integrate these features. This indicates it's feasible to incorporate the multimodal item data within an LLM-based architecture without architectural breakage.

- **Memory and Scalability:** Storing precomputed item embeddings for all items is a consideration. However, the size is manageable in many scenarios. For instance, with an embedding dimension of 512 (typical for CLIP ViT-B/32) and, say, 1 million items, that's ~512 million float parameters (~2 GB). Many industrial recommender systems handle tens of millions of item vectors in memory or via approximate nearest neighbor indices. If needed, dimensionality could be reduced (PCA or a smaller CLIP model) to fit strict memory budgets. The offline computation can be scaled horizontally – CLIP inference is easily parallelizable, and there are existing pipelines (e.g. ClipRetrieval) that process **100M+ images in a few hours** on a single GPU [6]. So even very large catalogs can be processed offline without issue. The two-stage design also naturally supports **incremental updates**: new items can be encoded on the fly (or in scheduled batches) using the frozen CLIP encoder and appended to the item embedding database, keeping the system up-to-date.

- **No Impact on User-Inference Path:** The user's online inference (e.g. sequential LLM forward pass to get a user vector, then a scoring) remains unchanged in complexity. We've essentially turned the item side into a **two-tower model**: one tower is the item encoder (with CLIP and fusion) computed offline, the other is the user's LLM that produces a compatible vector online. This two-tower setup is a proven scalable architecture for recommender retrieval because it allows using efficient vector dot-products for scoring. Our case is analogous, except the user tower is an LLM (possibly compressed or "lite") and the item tower is largely handled pre-runtime. Therefore, the design is feasible and **scalable to production** – it adds some storage and offline compute, but no extra inference latency.

- **Possible Training Approach:** During model training (offline), one could freeze CLIP and just train the fusion layer and the user-LLM parameters on recommendation data (e.g. using a cross-entropy loss to predict next item). Alternatively, one might use a small amount of fine-tuning on the CLIP text encoder to better handle domain-specific item descriptions (if they differ greatly from CLIP's training data). However, freezing CLIP is attractive for efficiency and to avoid overfitting when data is limited. Recent works indicate that **parameter-efficient tuning** can adapt pre-trained vision-language features to recommender domains without full fine-tuning [15] [16]. Our design naturally fits such a paradigm: train a few extra layers (or prompt tokens) on top of frozen features to nudge them towards the rec task, all while keeping the heavy model frozen.

Limitations and Risks of This Approach

While promising, this design has some limitations and potential risks to consider:

- **Visual-Textual Mismatch and Noise:** Not all item images will perfectly align with user preferences or textual attributes. If an item's image is uninformative or misleading (e.g. a poor quality image, or an image showing something not representative of the item), the CLIP embedding might introduce noise. A frozen CLIP model hasn't seen the specific recommendation objective, so it might emphasize

visual features that don't matter for recommendation (for example, background scenery in a product photo). This **modality mismatch** could hurt accuracy in some cases. Techniques to detect and mitigate image-text mismatches (as done by eBay with a mismatch detector [7] [8]) might be necessary. Otherwise, the fused item vector might occasionally be pulled in the wrong semantic direction by irrelevant visual features.

- **Domain Adaptation Gaps:** CLIP was trained on general web images and captions. If our domain is specialized (say fashion, furniture, etc.), the CLIP embeddings, while powerful, may not capture the fine-grained distinctions that drive user choices (e.g. subtle style nuances or brand logos, unless CLIP learned those). Because we keep CLIP frozen and only fuse features, the system might not fully **optimize visual features for recommendation relevance**. Prior research notes that using pre-extracted features without adaptation can limit performance potential – they provide a good initialization but are not *optimal* for the task [17] [18]. In our case, the recommendation model can learn to re-weight the fused features to some extent, but it won't alter the underlying image representation. If certain visual factors are highly predictive of user behavior in our domain, we might eventually need to fine-tune or adjust CLIP's representation (perhaps via a lightweight adapter) to emphasize those.

- **Lack of User-Image Interaction Modeling:** By performing fusion offline, the user's side of the model never directly "sees" images in context – it only deals with the compact item embeddings. In most scenarios this is fine (the item embedding carries the needed info), but it does mean the model cannot do things like attend to specific parts of an image based on a user's current query or preferences. All such interactions are implicit. For example, if a user's interest in an item is specifically tied to its visual aspect (say they only like an outfit if it has a floral pattern visible in the image), the user's preference vector is built from item embeddings that (hopefully) encoded "floral pattern" features. However, a more direct vision-aware model (e.g. one that feeds images into the model at inference) might learn a sharper signal (it could, in theory, highlight *which* visual features the user looked at). Our approach trades that potentially higher fidelity modeling for efficiency. **User preferences towards images are captured only indirectly** through the item embedding. If the fusion is well-designed, this should be sufficient; indeed, sequential models with multimodal item inputs have shown strong performance, implying users' visual tastes are learnable from the item vectors [19] [20]. It's just something to be mindful of: we are assuming a fixed representation of an item's visuals works for all users, which might miss some personalized interpretations of images.

- **Static Representations and Freshness:** The item embeddings are computed once and used for potentially a long time. If item content changes (e.g. a new image is uploaded, or the textual description is updated), the embedding needs to be refreshed offline. This is a general challenge with any content-based system, not specific to CLIP, but it adds maintenance overhead. Also, if user behavior shifts over time in ways that deviate from the static content signals (for instance, a fashion trend emerges that CLIP features alone can't capture because it's a subtle combination of image cues), the model might struggle until retrained or updated. In short, **frozen embeddings could become stale or suboptimal** if not updated periodically or if the domain evolves beyond what CLIP originally captured.

- **Marginal Gains in Data-Rich Cases:** When an item already has abundant interaction data (a "popular" item), collaborative signals (the learned ID embedding or user co-interaction patterns) are usually very strong. In those cases, adding the image/text content might yield only minor

improvement or none at all. There's even a risk of slight performance degradation if the model over-relies on content for popular items where the collaborative signal was actually more accurate. A recent analysis found that for very high-popularity items, a pure ID-based model can still edge out a modality-based model in accuracy [21]. Our design mitigates this by presumably **combining** content features with ID-based learning (e.g. the model can learn a weight to balance the CLIP features vs. collaborative latent features). But if not carefully balanced, the fused representation could dilute the precision for well-known items. Thus, one must ensure the model learns to trust CLIP features mainly when collaborative data is sparse (which can be achieved via training or gating mechanisms).

- **Complexity and Validation:** Introducing a multimodal component adds complexity to the system's design and training. We need to validate that the CLIP embedding dimensions and scales work well with the LLM's embeddings. Since CLIP's space is not originally designed for our recommendation loss, there might be some trial-and-error in how to fuse (simple averaging vs. learned fusion). Also, careful offline evaluation is needed to ensure we truly get the improvements (especially diversity or long-tail metrics) without unexpected side effects (e.g. recommending items that are visually similar but conceptually irrelevant). It's a manageable complexity, but the approach must be thoroughly tested – for instance, verifying that the Recall@K indeed increases and cold-start recommendations are meaningfully improved, as hypothesized.

In summary, the main risks revolve around *alignment and relevance*: ensuring the visual features we add are actually helping the recommendation objective and not introducing noise. Most of these risks can be mitigated by proper training (e.g. using an objective that weighs visual features according to actual clicks/purchases, as in eBay's co-click training with triplet loss [10]) and by monitoring performance on segments (like new items vs. popular items). The use of a **frozen** CLIP is double-edged: it keeps efficiency high, but any domain mismatch has to be addressed by the surrounding network rather than by fine-tuning CLIP itself. This is a reasonable trade-off for an efficient system, as long as we acknowledge the limits.

Performance Potential vs. Baseline and Prior Work

How much improvement can we expect? Based on both literature and practical reports, integrating CLIP embeddings in this two-stage LLM recommender is likely to improve performance on multiple fronts:

- **Overall Recall and Ranking Metrics:** We anticipate a moderate boost in metrics like Recall@K, NDCG, MRR overall, and a larger boost on cold-start subsets. The baseline Lite-LLM4Rec (text-only) already improves efficiency and uses item text to some extent, but adding images should let it make finer distinctions. For instance, Wu *et al.* (2025) report that their multimodal LLM-based model **outperforms state-of-the-art baselines on all evaluation metrics** across four datasets after fusing visual+text features [22] [23]. In their ablation, adding visual embeddings yielded a notable performance jump (MRR improved from 5.88 to 6.08 in one setting just by adding image features) [13]. This suggests that even when text is present, images provide complementary signal that lifts accuracy. We can expect our approach to similarly outperform a text-only Lite-LLM4Rec. In particular, **Recall@K for cold items** or less popular items should increase, since those items become more discoverable via content similarity.

- **Cold-Start and Long-Tail Items:** This is a primary area of gain. Studies consistently show content-based methods shine here. In a comparison of multimodal recommender (MoRec) vs ID-based (IDRec) methods, it was found that *"MoRec significantly improves over IDRec in cold-start scenarios"* for

both text and image modalities [2]. In our context, that means new items with only an image and description can be recommended more effectively – the model can find users who like similar looking or described items. We should see higher hit rates for new items and a faster ramp-up in engagement for them. The ability to recommend *visually similar* items also improves coverage of the long-tail: users might get recommendations of niche products that are visually akin to something they liked, even if those niche items have scant interaction history. This addresses the research goal of improving **Recall@K overall by not overlooking visually-relevant results** that a unimodal model might miss [24].

- **Comparison to Other Multimodal Recommenders:** Our approach is somewhat a middle ground between traditional multimodal recommenders and the latest large-model approaches:

    - Compared to classic multimodal methods (which often use CNN image features or fine-tune them), our use of CLIP offers a stronger pretrained representation and avoids costly fine-tuning. We likely will perform on par or better than those older methods. For example, earlier systems using pre-extracted VGG or simple image features saw only modest gains, but newer ones using CLIP or stronger vision models report significant improvements in recommendation performance [17] [18]. One paper demonstrated a ~10.6% **average increase in Recall@10** by leveraging CLIP features with minimal tuning, versus a model that did not use CLIP [25]. This gives a ballpark figure of the improvement one might see by injecting CLIP-based data.

    - Compared to bleeding-edge approaches like **Rec-GPT4V** (which apparently uses large vision-language models like GPT-4V or similar to directly reason over images), our approach will be more lightweight but possibly not as *theoretically* powerful. Rec-GPT4V is likely extremely computationally heavy – doing on-the-fly image understanding for each recommendation – which is impractical for real-time use. Our two-stage CLIP pipeline sacrifices some of that flexibility (we're not generating novel image captions or doing per-user image analysis) but still gains the bulk of the benefit in terms of accuracy. It's telling that even without fancy reasoning, adding image features in a straightforward way already boosts performance greatly. For instance, Rec-GPT4V's premise is that image content can be turned into text and fed to an LLM; but by using CLIP embeddings we've essentially achieved a similar fusion in vector form. We expect our Recall@K to be **competitive with state-of-the-art** multimodal methods, as evidenced by LLM-EMF (CLIP+LLM model) beating many baselines [26]. Unless the dataset or domain truly requires deep image reasoning (rare in typical recommendations), frozen CLIP features should capture most relevant visual information.

    - One advantage we have is efficiency: our method could be deployed in large-scale settings where something like GPT-4 Vision cannot due to latency. This **practical scalability** means our approach can actually realize the improvements in a production system. A method that scores slightly higher offline but is unusable in production is less valuable. Therefore, in a head-to-head comparison, our CLIP-enhanced Lite-LLM4Rec might slightly underperform a fully fine-tuned multimodal Transformer or a GPT-4V on some benchmarks, but it *will* be deployable. If it achieves, say, 95% of the more complex model's recall gain with 0% of the latency cost, that's a net win for a real-world recommender.

- **Validation from Real Systems:** As a concrete example supporting this approach, eBay's deployment of a multimodal item embedding (with offline image+text encoding and a two-tower retrieval model) led to **substantial gains in recommendation performance**. They reported +15% higher buyer engagement and +31.5% higher purchase-through rate after introducing fused image-text embeddings for items [3]. These are huge lifts in a production environment, underlining that multimodal fusion isn't just academically interesting – it yields tangible improvements in what users click and buy. Our proposed design is very much in line with what eBay did (they used ResNet and BERT; we use CLIP which is effectively a stronger combination of the two). This suggests that, performance-wise, one can expect notable improvements in both precision and recall metrics, especially for visually-driven product domains. Additionally, the eBay system was able to detect and filter out image-title mismatches [7] [8], improving result quality – our approach could inherit similar benefits (since CLIP can flag when image and text embeddings are very divergent, indicating a possibly misleading listing).

In summary, **compared to baseline Lite-LLM4Rec**, a CLIP-enhanced version should deliver higher Recall@K overall, a dramatic improvement for cold-start items, and better capability to recommend based on visual similarity or attributes. It stands up well against other multimodal recommenders: it uses modern pre-trained features (which is a step above early multimodal models) while maintaining the efficiency ethos (unlike some LLM+Vision methods). The performance gains target exactly the areas highlighted in the question – cold-start and recall – and evidence from research and industry validates these gains (e.g. "LLM-EMF consistently outperforms existing methods" with visual/text fusion [27], and multimodal models matching or beating ID-based models in strong sequential setups [19] [20]).

Research Gap and Contributions of This Design

Finally, we consider whether this approach fills a meaningful research gap and is a *practically significant, publishable* innovation. There are strong arguments that it is:

- **Bridging LLM4Rec and Multimodal Research:** Large Language Model for Rec (LLM4Rec) is a new paradigm, and most LLM4Rec work so far has focused on textual or ID inputs (formulating recommendation as a language modeling task, etc.). On the other hand, multimodal recommender research has typically not used LLMs as the core (instead using conventional neural rec models with images). The proposed two-stage pipeline brings these two lines together: it injects rich multimodal content into an LLM-based recommender *without* sacrificing the efficiency that makes LLM4Rec viable. This is relatively novel – only very recent papers (2024–2025) have started exploring LLMs with images for rec, and they either use heavy models (GPT-4V as in Rec-GPT4V) or specific scenarios (cross-domain as in LLM-EMF). Our approach would be among the first to demonstrate a **lightweight multimodal LLM recommender** that could realistically be deployed. This addresses a known gap: how to leverage powerful vision-language models in recommender systems **while keeping inference tractable** [18] [28]. The research community has noted that straightforwardly applying large pre-trained models can hurt efficiency, and calls for methods to *"maintain efficiency while coupling pre-trained models with recommender objectives"* [28]. Our two-stage design is exactly such a solution.

- **Efficiency & Practicality as a Contribution:** From a research perspective, showing that one can achieve near state-of-the-art multimodal recommendation quality *without* fine-tuning massive models or introducing high latency is a valuable insight. It can encourage more adoption of

multimodal approaches in industry. Many papers focus solely on raw accuracy improvements; a paper that demonstrates a **pragmatic trade-off** – e.g. "we got X% recall improvement and solved cold-start, with essentially zero impact on serving latency and only minor offline cost" – would stand out as extremely useful for real-world recommender system design. It would fill a gap between theory and practice. In essence, it could be positioned as *"CLIP meets Lite-LLM4Rec: Multimodal Boost with No Inference Cost"*, which addresses the common criticism that advanced multimodal methods are too slow or complex for production.

- **Cold-Start Emphasis:** The cold-start problem remains a fundamental challenge, and any new successful strategy for it is of interest. By explicitly evaluating the cold-start lift from image/text fusion, the work targets a well-known gap. It contributes to the narrative that leveraging content (especially with modern foundation models) can finally close the cold-start gap that traditional collaborative filtering couldn't. Given that the community is looking for ways to "overturn ID-based recommender reliance" in favor of more content-aware models [29] [30], a positive result here would be impactful. The Medium article we referenced highlights that if modality-based representations can achieve comparable results to ID-based methods across scenarios, it would be a paradigm shift in recommender systems [31] [32]. Our approach is a step in that direction, demonstrating one way to reach parity with ID-based methods (and superiority in cold-start) by using powerful content features.

- **Validation of Foundation Models in RecSys:** There is also a broader research question of how far large pre-trained models (like CLIP, BERT, GPT) can go in improving recommender systems. This work would provide a case study in successfully applying CLIP (a foundation vision-language model) in a recommendation pipeline. As such, it contributes to the ongoing discussion (and enthusiasm) around "Recommendation Foundation Models". The work could identify best practices for fusing embeddings, show where the benefits plateau, and reveal any pitfalls (like the need for slight fine-tuning or not). All of this is publishable material, as the community is actively exploring combining foundation models with recsys (e.g., tutorials are now appearing on *"multimodal pre-training for recommendation, using models like LLM, CLIP, etc."* [33]).

- **Experimental Evidence and Novelty:** While similar ideas have been tried (we cited LLM-EMF and industry solutions), our scenario might differ in the specifics (e.g., focusing on single-domain recommendations with a sequential LLM model, rather than cross-domain). We would bring new experimental evidence by benchmarking Lite-LLM4Rec vs. its CLIP-enhanced version on standard datasets. The novelty is in the **combination and restriction**: using a *frozen* CLIP in a *two-stage LLM rec* setup is a unique combination that hasn't been extensively covered. LLM-EMF, for instance, augmented item text with GPT and then used CLIP, focusing on cross-domain transfer [34]. Our work could simplify that (no GPT textual augmentation, perhaps) and concentrate on efficiency. It fills a gap between heavy multimodal LLM approaches and lightweight content-based recommenders.

- **Publishability:** Given the current trends, a paper on this topic is timely and likely to be well-received. It addresses **practical relevance (latency, scalability)** and **academic interest (LLMs + CLIP for RecSys)**. If our results show significant improvements (especially a nice bump in Recall@K and a strong solution for cold-start), that is a clear contribution. We also tackle some of the challenges (alignment without fine-tuning, etc.), providing insights. For example, we could contribute an analysis of when the CLIP features help most, how to best fuse them, and how much of CLIP's knowledge is actually utilized by the recommender – these are interesting points for the community.

The work would complement recent surveys that ask *“can stronger CV/NLP encoders directly improve recommender effectiveness?”* [31] by answering in the affirmative with empirical proof.

In conclusion, using frozen CLIP embeddings in a two-stage Lite-LLM4Rec pipeline appears to be a **practically effective and research-worthy approach**. It promises the best of both worlds – the power of multimodal content understanding and the efficiency of a lightweight LLM recommender. We expect to see richer item representations translate into higher recall (especially for cold items and diverse recommendations) without incurring the typical latency costs. The design is feasible with current technology and addresses a meaningful gap in recommender system research. Early evidence from literature and industry supports the potential of this approach, and a thorough evaluation would solidify its value. If executed and validated properly, this idea would indeed be scalable and publishable, marking a step forward in recommender systems that leverage the latest advances in vision-language modeling **for tangible gains in recommendation quality** [3] [1].

Sources:

1. Wu *et al.*, *"LLM-Enhanced Multimodal Fusion for Cross-Domain Sequential Recommendation"*, arXiv 2025 – uses frozen CLIP to generate image/text embeddings and finds that integrating visual data *“significantly improves”* cross-domain recommendation performance [1] [9].

2. Medium (Lifei Zhang), *"Multimodal Recommender System vs. ID-based Recommender System Revisited"*, 2023 – analysis showing multimodal methods outperform ID-based in cold-start and can match them in general with strong sequential models [2] [20].

3. Wang *et al.*, *"Rethinking Large Language Model Architectures for Sequential Recommendations (Lite-LLM4Rec)"*, arXiv 2024 – proposes Lite-LLM4Rec, a hierarchical LLM for efficient recommendations, achieving large efficiency gains by removing generative decoding [4] [12].

4. Li *et al.*, *"Bridging Domain Gaps between Pretrained Multimodal Models and Recommendations (PTMRec)"*, arXiv 2025 – shows that using pre-trained vision-language features (CLIP) with parameter-efficient tuning improved Recall@10 by ~10.6% on average, highlighting the benefit of such features for RecSys [15] [16].

5. eBay Tech Blog, *"Beyond Words: How Multimodal Embeddings Elevate eBay's Product Recommendations"*, 2023 – describes eBay’s two-tower model combining BERT text and ResNet-50 image embeddings for item recall, leading to +15% engagement and +31.5% purchase rate in A/B tests [3]. Their model used triplet loss to align modalities and a module to detect image-title mismatches [10] [7], illustrating practical considerations for a multimodal recommender deployment.

---

[1] [27] [2506.17966] LLM-Enhanced Multimodal Fusion for Cross-Domain Sequential Recommendation
https://ar5iv.labs.arxiv.org/html/2506.17966v1

[2] [19] [20] [21] [29] [30] [31] [32] Multimodal Recommender System vs. ID-based Recommender System Revisited | by AI-Advance | Medium
https://medium.com/@lifengyi_6964/multimodal-recommender-system-vs-id-based-recommender-system-revisited-588ca88cd16e

3 Multimodal Search for Enhanced Ecommerce Product Discovery
https://www.getfocal.co/post/multimodal-search-for-enhanced-ecommerce-product-discovery

4 5 12 [2402.09543] Rethinking Large Language Model Architectures for Sequential Recommendations
https://arxiv.org/abs/2402.09543

6 rom1504/clip-retrieval: Easily compute clip embeddings ... - GitHub
https://github.com/rom1504/clip-retrieval

7 8 10 11 24 Beyond Words: How Multimodal Embeddings Elevate eBay's Product Recommendations
https://innovation.ebayinc.com/stories/beyond-words-how-multimodal-embeddings-elevate-ebays-product-recommendations/

9 13 14 22 23 26 34 LLM-Enhanced Multimodal Fusion for Cross-Domain Sequential Recommendation
https://arxiv.org/html/2506.17966

15 16 17 18 25 28 Bridging Domain Gaps between Pretrained Multimodal Models and Recommendations
https://arxiv.org/html/2502.15542v1

33 [PDF] Multimodal Pre-training and Generation for Recommendation
https://www2024.thewebconf.org/docs/tutorial-slides/multimodal-pretraining-and-generation.pdf