Elevating Style with MLLM-Enhanced, Outfit Recommendation System

Sangeet Jaiswal[1,†], Gaurav Parashar[2,‡], Sreekanth Vempati[3] and Vijay Kumar[4]

Myntra Designs Pvt Ltd, Bangalore, India

Abstract

Fashion e-commerce platforms aim to replicate the curated experience of traditional retail by recommending items that complement a given product. In this paper, we present a framework for outfit recommendation for Myntra, a leading fashion e-commerce platform in India. We propose the use of Multimodal Large language models (MLLMs) to generate descriptive text for complementary fashion categories based on the visual and textual attributes of the primary product. These descriptions are converted into vector representations using a fine-tuned CLIP model, used to retrieve relevant products from our catalog via category-specific Approximate Nearest Neighbors (ANNs). To address the inherent combinatorial explosion when assembling outfits, potentially thousands of combinations, we introduce a ML ranking model trained on expert-annotated data, ensuring that only the most compatible combinations are surfaced to users.

To evaluate our approach, we propose an evaluation framework using Multimodal Large Language Models (MLLMs) as judge to compare outfit pairs generated by our proposed model and a Bi-LSTM-based baseline. For each input product, outfits generated by both models were evaluated across various categories. Our method achieves a win rate of 75.3%, indicating strong preference for outfits produced by the new model. This trend was further validated by expert human annotators, with high alignment between human judgments and LLM-generated scores, demonstrating the reliability of the automated evaluation and the superiority of our approach across categories. Additionally, we perform an ablation study comparing LLM judgments under two input conditions: outfit descriptions alone vs. descriptions with corresponding product images. Results demonstrate a significant improvement in win rate when images are included from 64.03% to 75.3% , highlighting the critical role of visual cues in assessing outfit compatibility.

Keywords

Outfit Recommendations, LLM, Multimodal Embeddings, Transformer

1. Introduction

Fashion e-commerce increasingly demands recommender systems that emulate the nuanced guidance of an in-store stylist, delivering outfits that not only appear visually coherent but also align with current fashion trends. In online shopping environments, where customers must navigate vast product catalogs without human assistance, the need for automated outfit recommendations becomes even more critical. To surface relevant combinations from millions of items, e-commerce platforms must rely on scalable recommendation systems. In our formulation, an outfit is defined as a set of 3–4 fashion items, typically comprising top-wear, bottom-wear, footwear, and accessories, that together form a stylistically coherent and complete ensemble without redundancy. The goal of an outfit recommendation system is to assign a compatibility score to a given outfit, reflecting how well the constituent items complement each other in terms of aesthetics and overall fashion sensibility.

Designing such systems hinges on two core challenges: (1) learning effective representations for various fashion items, and (2) modeling the intricate fashion compatibility relationships among them. Early approaches to outfit compatibility prediction primarily focused on pairwise compatibility learning[1]. For instance, Tan et al.[2] propose a weakly supervised framework that captures multifaceted and implicit similarities between item pairs. Similarly, Yang et al.[3] introduce a method that jointly learns

ECOM'25: SIGIR Workshop on eCommerce, Jul 17, 2025, Padua, Italy

[†]These authors contributed equally to this work.

[‡]Work done while at Myntra.

sangeet.jaiswal@myntra.com (S. Jaiswal); pgaurav@iisc.ac.in (G. Parashar); sreekanth.vempati@myntra.com (S. Vempati); vijay.kumar3@myntra.com (V. Kumar)

pairwise item embeddings and category-specific complementary relationships in a unified embedding space, optimized via end-to-end training. They all have achieved state of the art results but doesn't explicitly model outfit-level compatibility. Some of the earlier approaches to outfit modeling treat an outfit as an ordered sequence of fashion items [4, 5, 6], employing variants of Recurrent Neural Networks (RNNs) to encode the sequence and derive a global outfit representation. However, this formulation imposes a strong assumption on item order, which is not well-aligned with the nature of fashion compatibility, shuffling the items in an outfit should not affect its overall coherence or compatibility.

While prior approaches thoroughly describe their modeling pipelines and report competitive results on public datasets such as Polyvore[4] and IQON3000[7], they often overlook the practical challenges involved in generating outfit recommendations at scale. In particular, existing literature seldom addresses how to construct candidate sets for individual products—an essential step in large-scale outfit generation for real-world applications. In this work, we bridge this gap by proposing a novel candidate generation strategy that leverages Multimodal Large Language Models (MLLMs). Given a query product, we prompt the MLLM to first infer relevant usage occasions such as *brunch with friends*, *office wear*, or *festive gatherings* and then generate natural language descriptions of complementary items tailored for each occasion. These descriptions are subsequently used to retrieve compatible products from our catalog via multi modal retrieval, enabling more contextual and diverse outfit generation.

In summary, the main contributions of this work are as follows:

- We propose a novel use of MLLMs to generate complementary product descriptions conditioned on both the image and the text of a seed item.
- We leverage a fine-tuned CLIP[8] model, trained on our proprietary in-house fashion dataset, to embed these descriptions and retrieve compatible products from a multimodal ANN index.
- We introduce a Transformer-based outfit compatibility model trained on expert-labeled and augmented data generated from our catalogue and studio posts.
- We design a hybrid evaluation strategy that combines human assessment with Multimodal LLM-as-a-judge scoring to measure comparative outfit quality.
- We demonstrate how this can be deployed at large scale.

2. Related Work

The increasing capabilities of deep learning neural networks, as notably demonstrated by[9], have garnered considerable interest and application across diverse tasks within the fashion domain in recent years. Fashion outfit recommendation task in particular is of considerable interest. Initial attempts to address the outfit compatibility treated it as pairwise comparison among all the products in an outfit. These network uses Siamese network trained based on triplet loss[10, 11]. In contrast some researchers captures the holistic outfit-level representation through Bi-LSTM[4, 5] based methods. Some studies also have shifted to attention-based methods, including the Transformer architecture for personalized outfit recommendations and complementary item retrieval[12, 13, 14]. Personalized Outfit Generation (POG), proposed in [14], addresses the outfit generation problem by using transformer blocks on embeddings of all items in an outfit along with a masked embedding for the missing item. The output embedding on the masked slot is used to find the best matching item out of the alternatives. Contrary to our work, personalised outfit generations is achieved by encoding the entire purchase history of a shopper as one outfit with no masks. This embedding is then used in the masked transformer decoder blocks to generate outfit items, one at a time. HAT (History-Aware-Transformer)[13] personalises outfit recommendations by stacking two transformers: one encodes candidate-outfit compatibility, the other summarises a shopper's purchase history. The model is trained jointly with focal, contrastive, and adaptive-margin losses—bringing compatible outfits closer to a user's history while pushing random (weak-negative) outfits apart—enabling it to score new outfits in the context of each shopper's learned style.

Recent advancements have explored the integration of large language models (LLMs) into the fashion recommendation pipeline. Shi and Yang [15] proposed enhancing LLM-based outfit generation by explicitly incorporating fashion domain knowledge. They construct a structured knowledge graph capturing style rules, silhouette compatibilities, and colour coordination guidelines, and condition the LLM's generation through retrieval-augmented prompts. This approach significantly improves stylistic coherence, achieving notable gains in outfit completion and compatibility tasks over vanilla LLM baselines. Separately, Chen et al. [16] introduced a fine-tuning strategy that adapts a pre-trained LLM to image-guided outfit recommendation using preference feedback. Their method fuses visual features with textual prompts and employs efficient low-rank adaptation (LoRA) techniques to minimize computational overhead during personalization. Together, these works demonstrate that adapting LLMs with structured external signals—whether via explicit domain knowledge or implicit preference tuning—substantially advances the quality and personalization of generated outfits, setting a strong precedent for LLM-driven stylistic reasoning in fashion.

Our system adopts a two-stage strategy where an LLM is used to guide complementary product retrieval through textual conditioning, and a downstream ranking model—trained on expert-curated outfits—refines the selection.

3. Methodology

Our outfit recommendation pipeline is designed as a multi-stage system that combines the generative reasoning capabilities of large language models (LLMs) with retrieval-based candidate generation and learning-based scoring. The process begins with a multimodal LLM, which is used to generate high-quality, occasion-aware outfit descriptions for individual catalog items (Section 3.1). By conditioning on both textual metadata and product images, the MLLM mimics the role of a fashion stylist—suggesting complementary outfit items that are both visually and contextually aligned with how the product might be worn in real-world scenarios.

In the next stage (Section 3.2), these generated descriptions are mapped into an embedding space using an in-house CLIP model trained on Myntra's catalog of product images and descriptions. This embedding serves as a query to perform nearest-neighbor search over the catalog's visual space, enabling retrieval of suitable candidates for each outfit component based on stylistic compatibility.

Finally, in Section 3.3 we refine the recommendations by scoring each candidate outfit using a transformer-based model trained on fashion expert annotations. This model evaluates the visual and semantic compatibility among outfit items, assigning a relevance score that reflects the overall cohesiveness of the ensemble.

Together, these stages form a robust pipeline that generates, retrieves, and ranks outfit recommendations.

3.1. LLM based candidate generation

To kickstart the outfit recommendation pipeline, we leverage a state-of-the-art multimodal Large Language Model (MLLM), as a fashion stylist to generate context-aware complementary items descriptions for catalog products. The MLLM interprets both the visual and textual attributes of a given seed product and synthesizes descriptive prompts for complementary pieces, serving as the foundation for constructing high-quality outfit candidates.

This process unfolds in two stages: (1) predicting suitable occasions for a given product, and (2) conditioned on each occasion, it generates natural-language descriptions of complementary items that would complete the outfit.

3.1.1. Stage 1: Inferring Occasions for Catalog Items

For each product in our catalog, we collect its metadata—including product image, article type, gender, color, and long-form textual description—and query MLLM to identify appropriate usage contexts. The

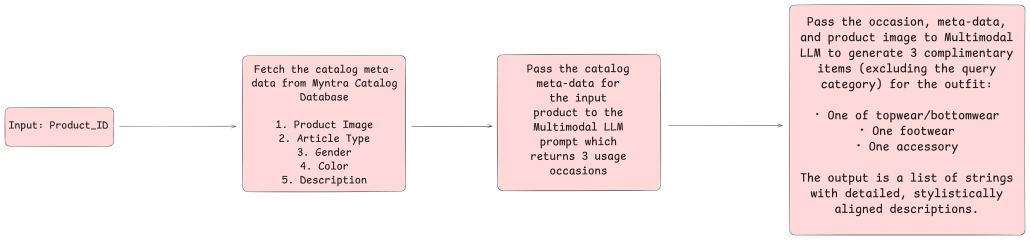

Figure 1: Diagram for generating the occasion and outfit description

model receives this structured input and is prompted to return **three specific occasions** (e.g., *brunch with friends*, *office wear*, *evening cocktail event*). This serves two key purposes:

1. **Semantic grounding:** It anchors the outfit generation process around real-world events such as *casual brunch*, *evening party*, or *office wear*, thereby enhancing the contextual relevance of the recommendations.

2. **User intent alignment:** It aligns with how users typically browse fashion platforms—driven by occasion-specific needs—which improves the utility of the generated outfits.

3.1.2. Stage 2: Outfit Composition

Once the occasion is determined, we make a second call to MLLM to generate a complete outfit centered around the query item. The model is instructed to recommend complementary items across three categories—*topwear or bottomwear* (whichever is not the query product), *footwear*, and one *accessory*. The category of the query product is explicitly excluded from the generation to ensure meaningful complementarity rather than redundancy.

The generation is conditioned on several inputs: the product metadata (including *gender*, *base color*, *article type*, and *textual description*), the inferred occasion, and the product image. The resulting outfit aims to be stylistically cohesive and appropriate for the specified context.

The prompt enforces the following constraints:

- Recommendations must be **stylistically aligned** with the input item, considering visual signals such as texture, pattern, and silhouette.

- Each complementary item must be **descriptive**, specifying color(s), material, design elements, and use case.

- Overly generic or low-utility accessories (e.g., cufflinks, pocket squares) are discouraged.

- The model should return an **unordered collection** of three textual descriptions—one each for an apparel item, suitable footwear, and an accessory.

By conditioning on both **visual elements** and **textual metadata**, MLLM generates high-quality, coherent outfit descriptions suitable for downstream retrieval tasks. Figure (1) illustrates the complete pipeline for obtaining the occasion context and generating the corresponding outfit description. Figure (2) illustrates an example of outfit generation for one Women Kurta style.

3.1.3. Why MLLM?

Unlike traditional text-only language models, the Multimodal LLM we employ supports image inputs alongside text, allowing it to reason jointly over visual and semantic attributes of fashion items—crucial for styling tasks in which subtle visual cues determine compatibility and aesthetic appeal.

Leveraging this multimodal "style engine" enables us to:

- Create high-quality outfit templates before retrieval (§3.2),
- Offer human-readable rationales for outfit coherence,
- Generate context-aware outfits by conditioning on the inferred usage occasion.

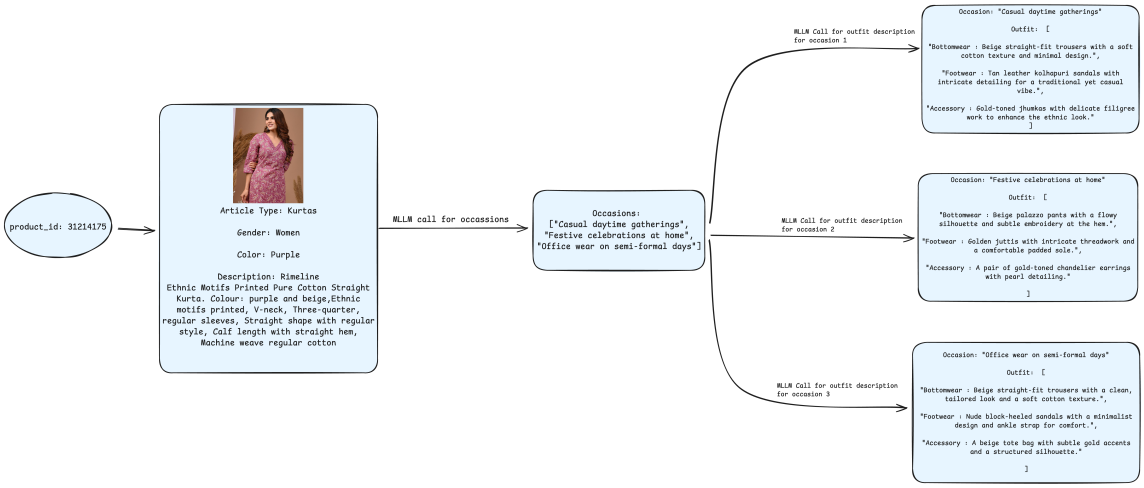

Figure 2: Example for generating the occasion and outfit description for a Women Kurta

3.2. Complementary-Item Retrieval Using CLIP Embeddings

To retrieve candidate complementary products, we first use Multimodal LLM to generate category-specific textual descriptions conditioned on the primary product's image and description as described in the section above.

These generated descriptions are then encoded using our fine-tuned CLIP text encoder to obtain 512-dimensional embeddings. Our in-house CLIP model was fine-tuned on a large fashion-specific dataset to better align visual and textual representations within our product catalog. To enable retrieval, we construct category-wise Approximate Nearest Neighbor(ANN) indices over our product catalog. Each product is represented as the weighted average of its CLIP-derived image and text embeddings, ensuring that the index captures both visual style and semantic content.

We empirically found that using the averaged text-image embeddings yields significantly better alignment with the GPT-generated descriptions compared to image-only indices, which tend to overfit to surface-level visual similarities. This design ensures that retrieval favors semantically complementary items, enhancing the relevance of the candidate set.

For each complementary category (e.g., bottomwear, footwear, accesories), we query the ANN with the corresponding GPT description embedding to retrieve the top-K most compatible products. While this ensures high-quality candidates per category, the number of possible outfit combinations grows combinatorially across categories and usage occasions. For example, selecting K items from each of three categories results in K^3 potential outfits per occasion. To effectively score and rank these combinatorial generated candidates, we employ a learned ranker in the final stage.

3.3. Transformer based model for outfits ranking

Our previous work [5] utilized a Bi-LSTM architecture for outfit compatibility prediction, modeling outfits as strictly ordered sequences of fashion items—top-wear followed by bottom-wear, footwear, and accessories. However, this approach exhibited several limitations. First, the imposed sequence ordering is problematic, as real-world outfits are inherently unordered. Second, training separate models for each gender-category combination restricts cross-category knowledge sharing, limiting generalization.

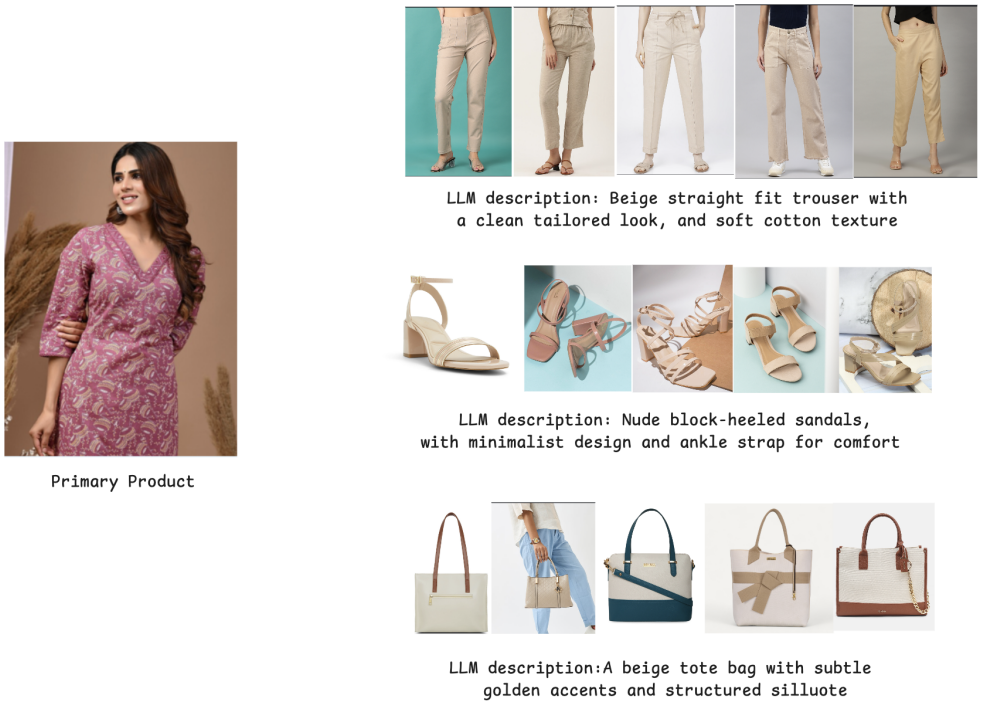

Figure 3: Top-5 Retrieved Complementary Products for a Given Kurti. The primary product (left) is used by the Multimodal LLM to generate category-specific descriptions, which are then used to retrieve relevant items (top-5 per category shown here: trousers, sandals, tote bags) from category-specific ANN indices.

Third, relying exclusively on Bayesian Personalized Ranking (BPR)[17] embeddings derived solely from user-item interactions neglected rich visual and textual product features. Additionally, BPR embeddings were continuously updated based on user interactions, leading to representational drift over time. This posed significant challenges for model stability and necessitated frequent retraining to maintain ranking quality—particularly for new or sparsely interacted products where interaction signals were weak or inconsistent.

Motivated by these shortcomings, we train a transformer-based model inspired by OutfitTransformer [12], designed to overcome the identified limitations. Our model integrates fixed multimodal embeddings derived from a fine-tuned CLIP model [8], capturing rich visual and textual signals. This approach provides stable and reusable representations that generalize well to cold-start scenarios, removes sequence order dependencies, enables cross-category learning, and produces content-informed product representations. The outfit compatibility score prediction task can be defined as follows: Given a set of N fashion products that form an outfit, each represented by a pair consisting of an image I_i and a textual description T_i, our objective is to learn a model f_θ that predicts an outfit-level compatibility score $y \in [0, 1]$. Mathematically, the task is expressed as:

$$y = f_\theta(\{(I_1, T_1), (I_2, T_2), \ldots, (I_N, T_N)\}) \tag{1}$$

Here, each (I_i, T_i) denotes the i^{th} product's image and corresponding textual description. The proposed model architecture, illustrated in Figure 5, consists of four primary components: We utilize an fine-tuned CLIP model consisting of separate image and text encoders, denoted as E_{img} and E_{text} respectively. Given product images and textual descriptions, these encoders produce embeddings:

$$\mathbf{v}_i^{img} = E_{\text{img}}(I_i) \in \mathbb{R}^d, \tag{2}$$

$$\mathbf{v}_i^{text} = E_{\text{text}}(T_i) \in \mathbb{R}^d, \tag{3}$$

where d is the embedding dimension. The parameters of these CLIP encoders remain fixed throughout training. For each product, we fuse its image and text embeddings into a single unified representation using multi-head attention[18]. Specifically, we compute:

$$\mathbf{v}_i^{fused} = \text{MultiHeadAttention}\left(Q = \mathbf{v}_i^{text},\ K = \mathbf{v}_i^{img},\ V = \mathbf{v}_i^{img}\right), \tag{4}$$

This formulation allows the model to selectively attend to image features conditioned on textual descriptions, enhancing the fused representation's contextual richness.

The fused embeddings for all products in the outfit are processed collectively by a Transformer encoder[19]. We prepend a learnable outfit token embedding $\mathbf{v}_{outfit}$ to the sequence of fused product embeddings. Formally:

$$[\mathbf{h}_{outfit}, \mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_N] = \text{TransformerEncoder}\left([\mathbf{v}_{outfit}, \mathbf{v}_1^{fused}, \mathbf{v}_2^{fused}, \ldots, \mathbf{v}_N^{fused}]\right), \tag{5}$$

where $\mathbf{h}_{outfit}$ is the learned contextual representation of the entire outfit token.

Finally, we pass the outfit-level embedding $\mathbf{h}_{outfit}$ directly through a Multilayer Perceptron (MLP) to produce the outfit compatibility score:

$$y = \sigma\left(\text{MLP}(\mathbf{h}_{outfit})\right), \tag{6}$$

where σ denotes the sigmoid activation to ensure the predicted score falls within the range $[0, 1]$.

To train the model, we curated a dataset of 78K outfits annotated for compatibility by domain experts, covering over 200K unique fashion products across multiple categories. Each outfit consists of 3 to 5 items, spanning top-wear, bottom-wear, footwear, and accessories. To augment the training data, we also generated an additional 100K outfits using full-shot studio and catalog images. These images were segmented into category-specific regions (e.g., topwear, footwear), and we performed nearest neighbor retrieval using CLIP embeddings to find visually similar items, forming plausible complementary product sets.

For negative sample construction, we adopted a semi-hard negative mining strategy[5]. From each positive outfit, we randomly replaced one or two items with products from the same category that exhibited the lowest similarity scores (farther neighbors) in the CLIP embedding space. This ensured the negative outfits were structurally plausible but semantically less compatible. This strategy helps the model better learn subtle compatibility cues. Additionally we constructed negatives by randomly replacing all items,resulting in trivially incompatible combinations, we incorporated such easy negatives early in training and gradually introduced harder ones. This curriculum-based strategy improved model robustness by guiding it from coarse discrimination to more nuanced compatibility learning.

During training, we froze the weights of the CLIP image and text encoders and extracted 512-dimensional embeddings for each modality. These were fused using a multi-head attention mechanism with 4 heads, enabling fine-grained cross-modal alignment. The resulting fused embeddings were processed by a transformer encoder with 2 layers (4 heads each), where a learnable outfit token was prepended and used to derive the global outfit representation. This representation was passed through a two-layer MLP (hidden size 256) to predict the final compatibility score.

The model was trained for up to 40 epochs using the Adam optimizer with a learning rate of 1×10^{-4}, a batch size of 128, and early stopping based on validation AUC. We used binary cross-entropy loss with logits. All experiments were conducted on NVIDIA A100 GPU.

We compare our proposed approach with our self attention based Bi-LSTM baseline *expert-annotated, disjoint* test set containing 15k outfits drawn from five Article-Type-Gender(ATG) categories. For every test sample the primary style (seed product) is strictly excluded from the training data, guaranteeing that the models must infer compatibility for previously unseen items. We compare our proposed Transformer-based ranker against the production Self-Attention Bi-LSTM baseline (window = 4). Table 1 reports AUC (area under the ROC curve) scores: the Transformer attains **0.85**, a substantial gain over the Bi-LSTM's 0.69.

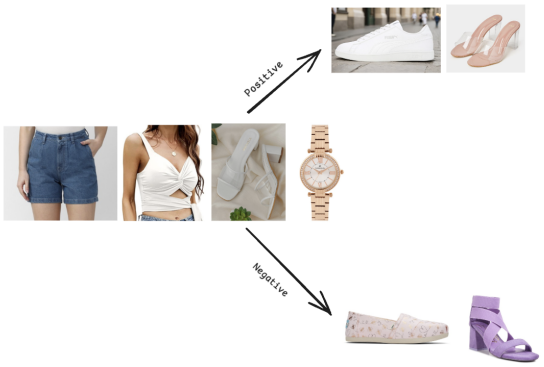

Figure 4: Example of the labelled training instances used for the outfit compatibility model. The four centre-left items constitute a *positive outfit* curated by human stylists. From the same primary product: (i) a **positive variant** (upper-right) obtained by substituting the footwear with an alternative selected by the stylists; and (ii) a **negative variant** (lower-right) generated by replacing the footwear with a distant item from the same category.

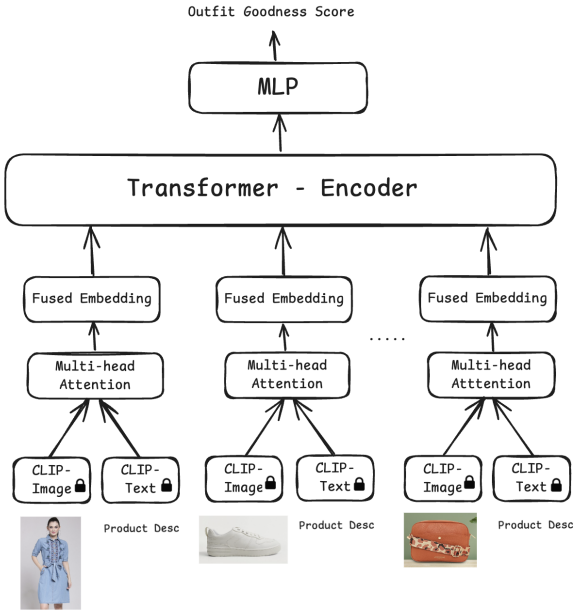

Figure 5: Architecture of the transformer-based outfit compatibility model.

Table 1
Test-set AUC on 15K expert-labelled outfits spanning five ATG categories.

Model	AUC
Self-Attention Bi-LSTM (w=4)	0.69
Transformer (ours)	**0.85**

4. Evaluation and Results

To evaluate the visual and stylistic quality of generated outfits, we did a benchmark of **5,390 outfit pairs** that enables direct, pairwise comparison between our proposed pipeline and Bi-LSTM–based approach [5]. Pairs were generated by selecting the top-k outfits from each method for the same **1,254** seed products drawn from five Article-Type-Gender (ATG) categories—*T-shirts (Men), Shirts (Men), Tops*

(Women), *Kurtas (Women)*, and *Dresses (Women)*. Altogether the pairs cover **42 410 unique fashion items**, providing a diverse test bed for both automated and human evaluation.

Our evaluation proceeds along three complementary dimensions. First, an independent Multimodal LLM rates each outfit pair on four criteria—visual coherence, occasion suitability, trend alignment, and overall appeal (Section 4.1). Second, the same set of pairs is reviewed by professional fashion stylists, yielding a human gold standard (Section 4.2). Finally, we quantify reliability through two agreement analyses: (i) inter-annotator agreement between the three human judges and (ii) cross-modal agreement between the LLM scores and the aggregated human ratings. Consistent gains across all three assessment streams confirm both the robustness of the evaluation protocol and the superiority of our multimodal generation-and-ranking framework over the Bi-LSTM baseline.

4.1. LLM as a Judge

To assess the stylistic coherence and overall appeal of the generated outfits, we employ a *separate* multimodal large language model (MLLM) as an automated judge. This evaluation model is used **only** for scoring and is distinct from the MLLM that produces the outfit descriptions in our generation pipeline. Leveraging an independent MLLM for evaluation allows us to obtain consistent, scalable, and bias-reduced comparisons between our proposed LLM+vector-search framework and previously deployed Bi-LSTM baseline [5]. For each primary product in our held-out set we first let our *MLLM + vector-search* pipeline and the baseline independently generate their respective top-k outfits. We then create k comparison pairs by aligning the i^{th} outfit from our approach with the i^{th} outfit from the baseline, ensuring that both outfits share the same seed item and Article–Type–Gender (ATG) category (e.g. *Shirts Men*, *Tops Women*). The descriptions for individual styles within an outfit were derived from the product catalog, ensuring consistency and realism in representation. Structured prompts were then constructed for evaluation MLLM, focusing on four key evaluation criteria:

- Visual and stylistic coherence across items
- Suitability for casual or occasion-specific wear
- Fashion sensibility and trend alignment
- Overall appeal and creativity

To ensure unbiased and independent evaluations, each outfit pair was passed to the LLM via isolated, stateless API calls—avoiding any potential influence from conversational history or prior context. Additionally, the ordering of outfits within each pair was randomized to prevent position-based bias in the model's decision-making. The MLLM was instructed to select **one winner or declare a tie** per pair and provide a concise rationale. Figure 6 illustrates an example of the system and user prompt passed to the MLLM judge and the decision made by it.

Moreover, these relative judgments provide a valuable signal that can be incorporated as weak supervision to further fine-tune our transformer-based outfit relevance ranking model (see Section 3.3)—enabling it to better capture nuanced preferences that align with both human and LLM-based evaluations.

After extracting the model's winner declarations, we compute per-ATG win rates for our approach. Table 2 reports results when the judging MLLM sees only the textual outfit descriptions, whereas Table 3 shows results when both text and outfit images are provided. Supplying the visual cues boosts our method's overall win rate from 64.08% to 75.30%, confirming that the additional image information substantially improves the perceived quality of the generated outfits.

4.2. Human Evaluators

To assess the visual and stylistic appeal of generated outfits, we conducted human evaluation. A total of **5,390 outfit pairs** were created, each containing one outfit generated using our proposed method and another generated using Myntra's prior approach [5]. These pairs spanned **1,254 unique primary**

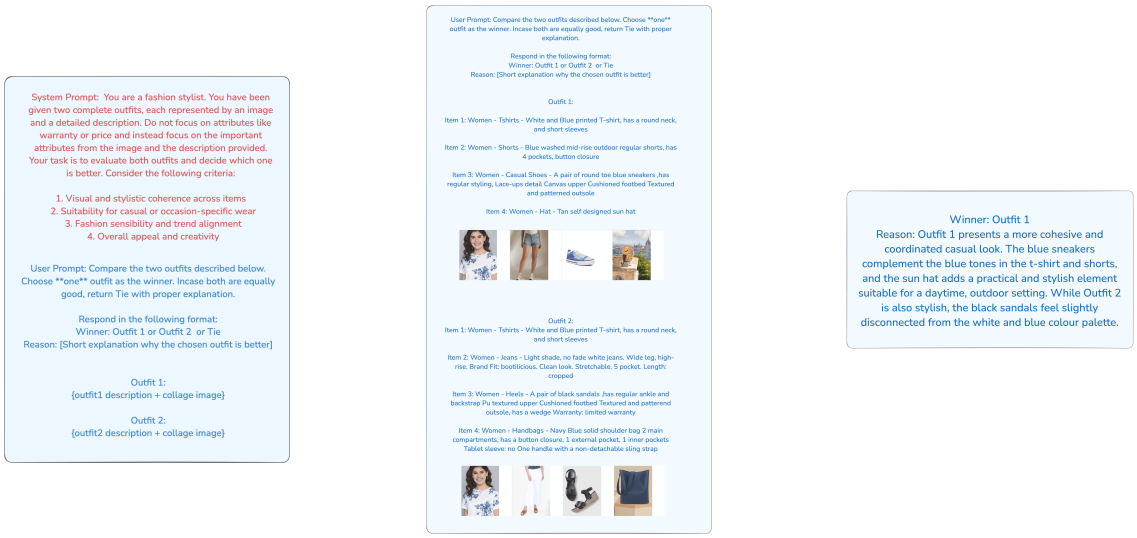

Figure 6: MLLM-judging prompts: (a) generic template, (b) user prompt example for style comparison, and (c) MLLM output result for the style judgment.

Table 2
MLLM-judge results when the evaluator sees **text-only** outfit descriptions. Win rates are reported for the proposed method versus the baseline.

ATG	# Pairs	MLLM Wins	Baseline Wins	Invalid/Ties	MLLM Win %
Tshirts Men	1002	520	469	13	52.6%
Shirts Men	1140	570	543	27	51.2%
Tops Women	1050	618	426	6	59.8%
Kurtas Women	1022	809	207	6	79.6%
Dresses Women	1176	903	242	31	78.9%
Overall	**5390**	**3420**	**1887**	**83**	**64.08%**

Table 3
MLLM-judge results when the evaluator receives both the textual descriptions and the outfit images. Win rates are reported for the proposed method versus the baseline.

ATG	# Pairs	MLLM Wins	Baseline Wins	Invalid/Ties	MLLM Win %
Tshirts Men	1002	621	381	0	62.0%
Shirts Men	1140	777	363	0	68.2%
Tops Women	1050	779	271	0	74.2%
Kurtas Women	1022	873	149	0	85.4%
Dresses Women	1176	1008	168	0	85.7%
Overall	**5390**	**4058**	**1332**	**0**	**75.3%**

styles across **5 distinct Article Type Gender (ATG)** categories: *Men Tshirts*, *Men Shirts*, *Women Tops*, *Women Kurtas*, and *Women Dresses*. In total, the outfit pairs encompassed **42,410 unique fashion items**.

To ensure fairness and eliminate potential position bias, the order of outfits within each pair was randomized before annotation. For each pair, we displayed the catalog image and the associated textual description of the outfit items. These descriptions were extracted from Myntra's product catalog and enabled the annotators to form a holistic understanding of each outfit's style and coherence.

We engaged **trained human annotators** to independently evaluate all pairs and select the better-looking outfit or mark a tie if both were equally good. The annotators were familiar with fashion merchandising and styling, ensuring that judgments reflected relevant fashion sensibilities.

Annotation Statistics. The responses were aggregated in two ways: (1) individual judgments across all annotators, and (2) majority voting to establish a final label per outfit pair. The detailed annotation breakdown is presented in Table 4.

Table 4
Human annotator preferences across 5,390 outfit pairs. "Ours" refers to the MLLM + vector search approach; "Baseline" refers to the Bi-LSTM baseline [5].

Vote Type	Ours Wins	Baseline Wins	Tie
Individual Votes (3 annotators × 5,390 pairs)	9,702 (60.0%)	5,756 (35.6%)	712 (4.4%)
Majority Vote (per pair)	3,220 (59.7%)	1,847 (34.2%)	323 (6.0%)

LLM-Human Alignment. We also evaluated the agreement between human annotators and the Multimodal LLM judge (see Section 4.1). We computed alignment at the majority-vote level by comparing the final MLLM judgment for each pair with the majority human judgment. Table 5 summarizes the alignment results.

Table 5
Agreement between MLLM and majority human vote over 5,390 outfit pairs.

Agreement Type	Count	Percentage
Exact Match (MLLM == Human Vote)	3870	71.8%
Disagreement	1520	28.2%

These results indicate a high degree of alignment between human aesthetic judgment and the LLM-based evaluations, supporting the reliability of using large language models as weak judges in relative outfit ranking.

5. Deployment and Serving

To enable scalable outfit recommendation, the system is designed around a hybrid architecture that can combine a daily offline pipeline for batch outfit generation with an online service for real-time serving. The intended overall flow is illustrated in Figure 7.

5.1. Eligible Styles Detection

A daily offline pipeline is initiated over the full fashion catalog to identify eligible styles for outfit generation. The resulting set, referred to as `eligible_styles`, forms the input for subsequent stages.

5.2. Style Description Generation (MLLM Layer)

For each style in `eligible_styles`, complementary style descriptions can be generated using a Multimodal Large Language Model (MLLM), conditioned on both the textual and visual attributes of the seed product. These descriptions can be cached in persistent storage to avoid redundant generation and ensure consistency. This step would yield `product_id`, `outfit_description` pairs that semantically characterize each style.

5.3. Semantic Embedding and Retrieval (CLIP + ANN Layer)

The generated textual descriptions can be embedded into dense vectors using a fine-tuned CLIP text encoder. Approximate Nearest Neighbor (ANN) retrieval, based on the HNSW algorithm [20], can then be performed over a filtered search space restricted by category and gender. This stage would retrieve a candidate pool of items that exhibit strong semantic and visual similarity to the seed style.

5.4. Outfit Ranking (ML Relevance Layer)

Candidate outfit combinations, typically comprising 3–4 items, can be scored using a Transformer-based ranking model that computes inter-item compatibility through multimodal embeddings. This approach would produce a ranked list of outfits for each seed style, optimized for stylistic coherence and fashion relevance, though it is not currently in place.

5.5. Persistence and Online Serving

The ranked outfits can be stored in a low-latency key-value store in the format `product_id`: [`outfit_1`, `outfit_2`, ...]. At inference time, recommendations on product detail pages could then be fetched directly from this store and served to users. While this approach is feasible, it is not currently implemented.

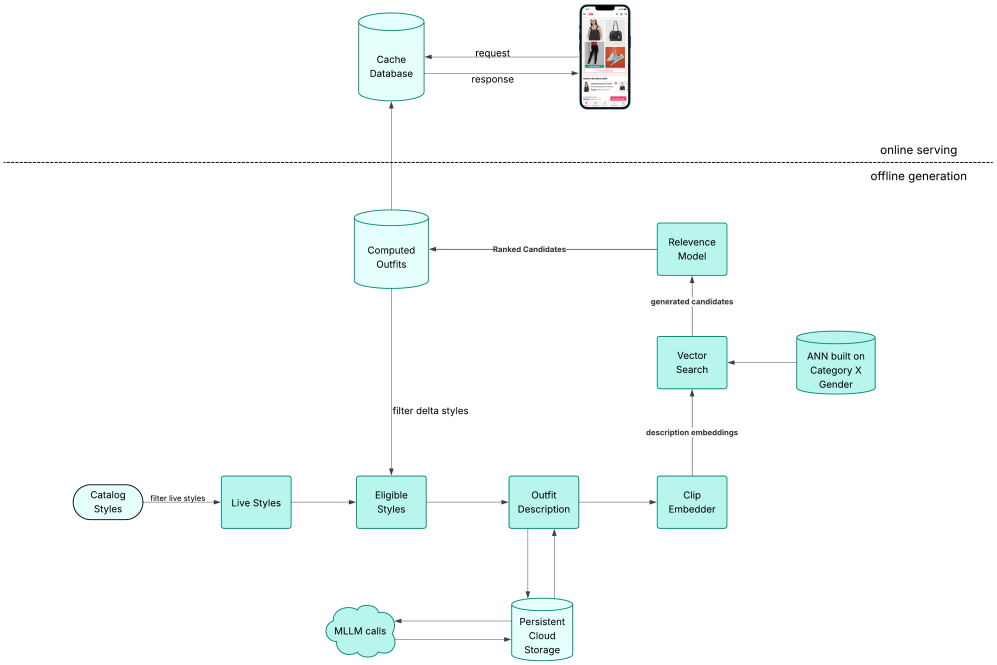

Figure 7: High-level system architecture for large-scale outfit generation and online serving.

6. Conclusion

In conclusion, we introduced a novel outfit recommendation framework. Our method uses LLMs to generate descriptive text for complementary categories based on a primary product's visual and textual features. These descriptions, encoded by a fine-tuned CLIP model, enable efficient retrieval of relevant items. A learned ranking model, trained on expert data, addresses combinatorial challenges by surfacing only the most compatible outfits. Our evaluation framework, employing MLLMs as judge, demonstrated a significant preference for our LLM-driven outfits over a Bi-LSTM baseline, a finding validated by expert human annotators. Notably, an ablation study highlighted the critical role of visual cues, showing a substantial improvement in evaluation accuracy when images were included. While our current pipeline focuses on generating stylistically coherent and context-aware outfit recommendations, future work can explore integrating user-level personalization signals—such as historical preferences, interaction data, or cohort behavior—to tailor outfit generation to individual tastes. Additionally, incorporating feedback loops (e.g., clicks, saves, or purchases) could enable adaptive learning and continuous refinement of recommendations.

Declaration on Generative AI

During the preparation of this work, the author(s) used GPT-4o and Grammarly in order to: Grammar and spelling check. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] X. Song, F. Feng, X. Han, X. Yang, W. Liu, L. Nie, Neural compatibility modeling with attentive knowledge distillation, in: The 41st International ACM SIGIR conference on research & development in information retrieval, 2018, pp. 5–14.

[2] R. Tan, M. I. Vasileva, K. Saenko, B. A. Plummer, Learning similarity conditions without explicit supervision, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 10373–10382.

[3] X. Yang, Y. Ma, L. Liao, M. Wang, T.-S. Chua, Transnfcm: Translation-based neural fashion compatibility modeling, in: Proceedings of the AAAI conference on artificial intelligence, volume 33, 2019, pp. 403–410.

[4] X. Han, Z. Wu, Y.-G. Jiang, L. S. Davis, Learning fashion compatibility with bidirectional lstms, in: Proceedings of the 25th ACM international conference on Multimedia, 2017, pp. 1078–1086.

[5] M. Madan, A. Chouragade, S. Vempati, The joy of dressing is an art: Outfit generation using self-attention bi-lstm, in: Y. Dong, N. Kourtellis, B. Hammer, J. A. Lozano (Eds.), Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track, Springer International Publishing, Cham, 2021, pp. 218–233.

[6] T. Nakamura, R. Goto, Outfit generation and style extraction via bidirectional lstm and autoencoder, arXiv preprint arXiv:1807.03133 (2018).

[7] X. Song, X. Han, Y. Li, J. Chen, X.-S. Xu, L. Nie, Gp-bpr: Personalized compatibility modeling for clothing matching, in: Proceedings of the 27th ACM International Conference on Multimedia, MM '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 320–328. URL: https://doi.org/10.1145/3343031.3350956. doi:10.1145/3343031.3350956.

[8] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: International conference on machine learning, PmLR, 2021, pp. 8748–8763.

[9] W.-H. Cheng, S. Song, C.-Y. Chen, S. C. Hidayati, J. Liu, Fashion meets computer vision: A survey, 2021. URL: https://arxiv.org/abs/2003.13988. arXiv:2003.13988.

[10] M. I. Vasileva, B. A. Plummer, K. Dusad, S. Rajpal, R. Kumar, D. Forsyth, Learning type-aware embeddings for fashion compatibility, 2018. URL: https://arxiv.org/abs/1803.09196. arXiv:1803.09196.

[11] A. Veit, B. Kovacs, S. Bell, J. McAuley, K. Bala, S. Belongie, Learning visual clothing style with heterogeneous dyadic co-occurrences, 2015. URL: https://arxiv.org/abs/1509.07473. arXiv:1509.07473.

[12] R. Sarkar, N. Bodla, M. Vasileva, Y.-L. Lin, A. Beniwal, A. Lu, G. Medioni, Outfittransformer: Outfit representations for fashion recommendation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2022, pp. 2263–2267.

[13] M. C. Jung, J. Monteil, P. Schulz, V. Vaskovych, Personalised outfit recommendation via history-aware transformers, 2024. URL: https://arxiv.org/abs/2407.00289. arXiv:2407.00289.

[14] W. Chen, P. Huang, J. Xu, X. Guo, C. Guo, F. Sun, C. Li, A. Pfadler, H. Zhao, B. Zhao, Pog: Personalized outfit generation for fashion recommendation at alibaba ifashion, 2019. URL: https://arxiv.org/abs/1905.01866. arXiv:1905.01866.

[15] Z. Shi, S. Yang, Integrating domain knowledge into large language models for enhanced fashion recommendations, 2025. URL: https://arxiv.org/abs/2502.15696. arXiv:2502.15696.

[16] N. Forouzandehmehr, N. Farrokhsiar, R. Giahi, E. Korpeoglu, K. Achan, Decoding style: Efficient fine-tuning of llms for image-guided outfit recommendation with preference, 2024. URL: https://arxiv.org/abs/2409.12150. arXiv:2409.12150.

[17] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, Bpr: Bayesian personalized ranking from implicit feedback, arXiv preprint arXiv:1205.2618 (2012).

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, 2023. URL: https://arxiv.org/abs/1706.03762. arXiv:1706.03762.

[19] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL: https://arxiv.org/abs/1810.04805. arXiv:1810.04805.

[20] Y. A. Malkov, D. A. Yashunin, Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs, 2018. URL: https://arxiv.org/abs/1603.09320. arXiv:1603.09320.