GENERATIVE VISUAL RECOMMENDATION THROUGH RETRIEVAL-TO-GENERATION MAPPING

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative recommendation has evolved along two major paradigms: generative retrieval (GenRet) and personalized visual generation (PVGen). Specifically, GenRet captures collaborative signals from user–item interactions but remains restricted to existing catalog items. While PVGen synthesize new items, yet may not fully exploit these signals for user preference modeling. However, integrating the two paradigms presents challenges in computational efficiency and semantic misaligned between their retrieval and visual generation latent spaces. To address these challenges, we propose Retrieval-to-Generation Mapping (R2G-Map), which efficiently and effectively integrates GenRet and PVGen by aligning recommendation-oriented preference representations with the visual generation space. Specifically, R2G-Map uses continuous recommender hidden states before discrete token selection to retain fine-grained collaborative and predictive information. To bridge the two latent spaces, we introduce a Factorized Feature Predictor that predicts content, description, and collaborative features and constructs visual condition for image generation. We further develop a Dual-Branch Visual Generator that combines this condition with preference representations through joint attention between an image branch and a latent branch. The resulting visual-token distributions support catalog ranking, while the corresponding token sequences are decoded into personalized item images. Experiments on bundle and sequential recommendation demonstrate that R2G-Map improves ranking performance while generating visually coherent and compatible item images.

1 INTRODUCTION

Generative recommendation uses generative models to recommend items or create personalized content based on users' preferences and interaction histories (Rajput et al., 2023). Existing work can be organized into two major paradigms: *generative retrieval (GenRet)* and *personalized visual generation (PVGen)*[1]. Specifically, *GenRet* retrieves existing catalog items by representing each item as a token sequence (Rajput et al., 2023). *PVGen* instead synthesizes new images tailored to user preference. (Xu et al., 2025c). Together, these paradigms have made generative recommendation an increasingly important direction in recommender systems.

Despite substantial progress, *GenRet* and *PVGen* face different limitations in catalog coverage and user preference modeling. Regarding to *GenRet*, semantic ID-based methods such as TIGER predict token sequences that encode item semantics while learning collaborative patterns from user interactions (Rajput et al., 2023). LLM-based methods can instead generate textual descriptions and match them to existing catalog entries. However, both semantic ID-based and LLM-based methods ultimately describe or retrieve existing catalog items and cannot create new items when the available catalog does not meet a user's needs. As the second paradigm of generative recommendation, *PVGen* directly synthesizes new items that do not exist in the catalog. For example, in the fashion domain, existing methods generate personalized and compatible item images using GANs (Kang et al., 2017; Moosaei et al., 2022) or diffusion models (Xu et al., 2024; Yu et al., 2024a; 2025; 2026). However, their user preference modeling remains relatively simple, relying on averaged representations of historically interacted items (Xu et al., 2024) or preference keywords and embeddings extracted from

[1]Some studies use the broader term *personalized content generation*. We focus on visual content, specifically images, and leave extensions to other modalities, such as text, audio, and video, for future work.

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.

user histories by an LLM (Shen et al., 2024), without explicitly optimizing item relevance through recommendation objectives. Nevertheless, personalized recommender systems have extensively studied user preference modeling through collaborative filtering and sequential preference modeling (Liang et al., 2018; Zheng et al., 2022; Xu et al., 2025a), yet these capabilities have not been fully integrated into *PVGen* methods. Consequently, *PVGen* may fail to effectively exploit the collaborative or sequential patterns from the user-item interactions or item-item transition sequences. This gap raises a natural question: can we unify *GenRet* and *PVGen* within a single integrated framework?

However, building such a unified framework poses two technical challenges. The first lies in achieving *efficient and effective integration* of *GenRet* and *PVGen*. A straightforward approach would be to integrate the two models for end-to-end joint training. Yet their combined scale and complexity would make joint training computationally expensive and potentially hinder convergence. Moreover, their distinct learning objectives would require careful coordination to balance recommendation accuracy and generation quality. The second challenge is the *semantic misalignment* between the latent spaces of *GenRet* and *PVGen*. Specifically, *GenRet* learns preference representations that capture user preferences and item relationships but may fail to preserve the fine-grained visual semantics required for generation. These representations are not inherently aligned with the visual generation space, and directly using them as conditioning inputs may therefore introduce out-of-distribution signals that the generator cannot effectively interpret or exploit (see the empirical evidence in Section 4.2).

To address these challenges, we propose **R**etrieval-**to**-**G**eneration **Map**ping (**R2G-Map**), which enables efficient and effective integration of *GenRet* and *PVGen* by aligning recommendation-oriented preference representations with the visual generation space. For efficient and effective integration, we inject preference representations learned by *GenRet* into *PVGen* to guide personalized visual synthesis. Specifically, we extract continuous recommender hidden states before discrete token selection and map them into the visual generation space to preserve predictive uncertainty and fine-grained collaborative information. To address semantic misalignment, Factorized Feature Predictor (FFP) that predicts content, description, and collaborative features from the recommender hidden states. The predicted content features are then used to construct a visual condition, providing explicit visual-semantic guidance for subsequent generation. To combine this visual condition with preferences representations, we further introduce a Dual-Branch Visual Generator (DVG), comprising an image branch for modeling item appearance and a latent branch for incorporating recommendation preferences. Within DVG, joint attention enables information exchange between the two branches, allowing visual-token representations to incorporate collaborative preference representations. The resulting visual-token distributions support catalog ranking, while the corresponding token sequences are decoded into item images for personalized visual generation.

Our main contributions are summarized as follows: 1) We propose R2G-Map, which efficiently and effectively integrates *GenRet* and *PVGen* by aligning recommendation-oriented preference representations with the visual generation space. 2) We introduce FFP and DVG module to address semantic misalignment between recommendation and generation. FFP predicts factorized features, while DVG incorporates collaborative preferences into visual-token representations. 3) Experiments on bundle and sequential recommendation demonstrate that R2G-Map improves catalog-ranking performance while generating visually coherent and compatible items, validating its effectiveness in both retrieval and personalized visual generation.

2 Related Work

Generative retrieval-based recommendation formulates item retrieval as sequence generation, with methods differing primarily in their output representations of items. LLM-based methods inherit the semantic priors of pretrained language models and directly express item attributes or descriptions (Xu et al., 2025c). Semantic ID-based methods instead quantize continuous item embeddings into sequences of discrete codes, enabling parameter sharing across semantically related items (Rajput et al., 2023; Ju et al., 2025). Recent work further disentangles semantic and collaborative factors (Liu et al., 2025), extends ID sequences while preserving parallel decoding (Hou et al., 2025), and applies discrete diffusion to bundle generation (Tu et al., 2026). However, these methods describe or retrieve existing catalog items and cannot create new items when the catalog fails to meet a user's needs.

Generative visual recommendation extends visually aware recommendation from selecting existing items to synthesizing personalized and compatible images. Researchers in the fashion domain have

actively adopted these advancements for fashion recommendation. DiFashion generates personalized outfits through denoising conditioned on category prompts, outfit context, and interaction history (Xu et al., 2024), whereas FashionDPO refines pretrained outfit generators using feedback on visual quality, compatibility, and personalization (Yu et al., 2025). PMG combines LLM-derived keywords and implicit preference embeddings to guide multimodal generation (Shen et al., 2024), while Pigeon adapts large multimodal models to personalized image generation from user histories and multimodal instructions (Xu et al., 2025b). And DualFashion applies coupled image and text diffusion to generate fashion items with structured descriptions (Yu et al., 2026). These approaches remain image-centric, incorporating recommendation signals into image generation through visual histories, textual prompts, or auxiliary conditions. In contrast, our R2G-Map allows the semantic and collaborative knowledge learned by different generative recommendation backbones to guide visual synthesis.

Discrete visual tokenization formulates image synthesis as the prediction of discrete token sequences (Esser et al., 2021; Chang et al., 2022; Pang et al., 2025). This representation is also compatible with the token-prediction paradigm of language models, facilitating the unified modeling of images and text within a shared sequence-modeling framework (Wu et al., 2025). VQ-VAE introduced learned vector-quantized latent representations (van den Oord et al., 2017). TiTok reduced sequence length through compact one-dimensional tokenization (Yu et al., 2024b), and TA-TiTok further introduced text-conditioned decoding for both discrete and continuous tokens (Kim et al., 2025). These developments provide compact visual vocabulary and efficient decoder, but do not address their alignment with recommendation-specific hidden states.

3 METHOD

3.1 PROBLEM FORMULATION

Let $\mathcal{I} = \{i_1, \ldots, i_N\}$ denote a catalog of N fashion items, each associated with an image x_i. We consider bundle recommendation and sequential recommendation under two ranking protocols: *Negative-Sampling Ranking* ranks target items against randomly sampled negatives within a retrieval candidate pool $\mathcal{A}_{\text{ret}} \subseteq \mathcal{I}$, while *All Ranking* ranks the entire catalog $\mathcal{I}$.

Bundle recommendation follows the partial-bundle completion setting (Tu et al., 2026). Let $\mathcal{B} = \{b_1, \ldots, b_M\}$ denote a collection of bundles, each containing an unordered set of compatible items from $\mathcal{I}$. For each bundle $b \in \mathcal{B}$, the partial-bundle items b_x serve as the recommendation context, while the missing items $b_y = b \setminus b_x$ form the target, with $m = |b_y|$ denoting the number of items to complete. *GenRet* selects m items that are not in b_x, yielding a completion $\hat{b}_y \subseteq \mathcal{I} \setminus b_x$. *PVGen* synthesizes a set of m item images $\hat{\mathcal{X}}_y = \{\hat{x}_1, \ldots, \hat{x}_m\}$ that jointly complement the items b_x.

Sequential recommendation predicts the next item $i_{t+1} \in \mathcal{I}$ given a sequence prefix $\mathcal{S} = (i_1, \ldots, i_t)$, where $i_\tau \in \mathcal{I}$. This task targets on a sequence continuation rather than completing the partial-bundle. The prefix $\mathcal{S}$ serves as the recommendation context, while i_{t+1} forms the target. *GenRet* recommendation ranks candidate items according to their relevance scores $s(i \mid \mathcal{S})$. *PVGen* synthesizes an item image $\hat{x}_{t+1}$ that visually expresses the next-item characteristics.

3.2 GENERATIVE RETRIEVAL MODEL

Our framework starts from the retrieval space, where item representations are learned for *GenRet*. Given a recommendation context $\mathcal{C}$, this module extracts target-item states from frozen recommender and supplies them to the subsequent mapping module.

Semantic-ID tokenization. Semantic IDs are constructed from continuous catalog-item representations. For each item $i \in \mathcal{I}$, the pretrained item-encoding and fusion pipeline (Ma et al., 2024) produces $\boldsymbol{e}_i = F(\boldsymbol{e}_i^c, \boldsymbol{e}_i^t, \boldsymbol{e}_i^f) \in \mathbb{R}^{d_e}$, where $\boldsymbol{e}_i^c$, $\boldsymbol{e}_i^t$, and $\boldsymbol{e}_i^f$ denote item content, description, and collaborative features, respectively, and d_e denotes the dimensionality of the fused item embedding $\boldsymbol{e}_i$. This embedding integrates multimodal content semantics and collaborative signals into a unified continuous item representation for subsequent tokenization.

We then tokenize the fused item embedding $\boldsymbol{e}_i$ using residual vector quantization (RVQ) (Rajput et al., 2023; Tu et al., 2026). RVQ employs L_q quantization levels, each with a codebook $\mathcal{Q}^\ell = \{\mathbf{q}_k^\ell\}_{k=1}^{K_s} \subseteq \mathbb{R}^{d_e}$ containing K_s codewords. Starting from $\mathbf{r}_i^0 = \boldsymbol{e}_i$, each level $\ell = 1, \ldots, L_q$ selects

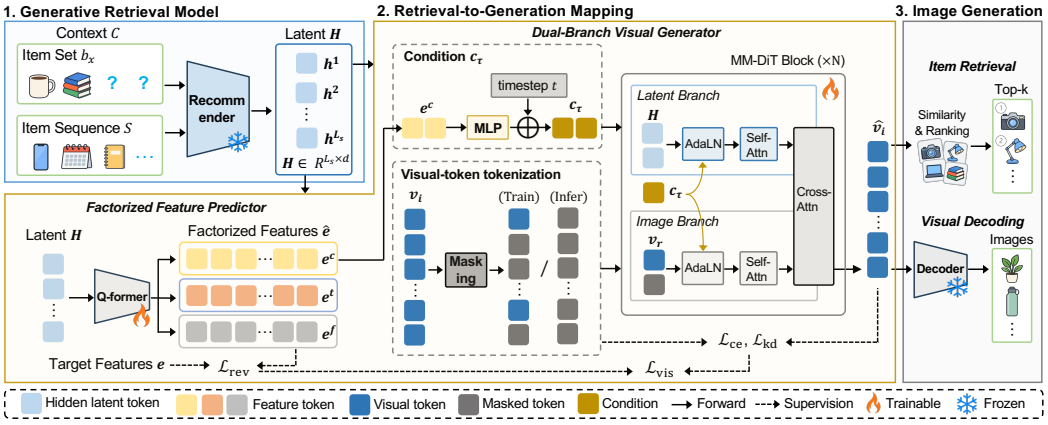


Figure 1: Overview of R2G-Map. The mapping bridges retrieval space and visual generation space, while the predicted feature conditions visual-token generation for ranking and generation.

the codeword closest to the current residual and subtracts it to obtain the residual for the next level:

$$s_i^\ell = \arg\min_{k \in \{1,\dots,K_s\}} \|\mathbf{r}_i^{\ell-1} - \mathbf{q}_k^\ell\|_2^2, \quad \mathbf{r}_i^\ell = \mathbf{r}_i^{\ell-1} - \mathbf{q}_{s_i^\ell}^\ell. \tag{1}$$

Here, s_i^ℓ denotes the index of the codeword selected for item i at quantization level ℓ. The corresponding sequence of codeword indices $\mathbf{s}_i = (s_i^1, \dots, s_i^{L_q})$ serves as the item's Semantic ID and provides a structured discrete representation for *GenRet*.

Latent extraction. Given a recommendation context $\mathcal{C}$, we represent each context item i by its code sequence $\boldsymbol{s}_i$ and feed it into a frozen *GenRet* backbone. Our framework supports both autoregressive and discrete diffusion recommender, which use a shared Semantic-ID tokenizer and predict codeword indices from the same level-specific vocabularies.

During inference, the recommender computes contextual hidden states for the target semantic tokens. Let $\mathbf{h}_{j,\ell}^{(u)} \in \mathbb{R}^{d_r}$ denote the hidden state at token position ℓ of target item slot j, evaluated at decoding or denoising step u, where d_r is the hidden-state dimensionality. The prediction head maps this state to a probability distribution over the K_s valid code tokens at level ℓ:

$$\boldsymbol{\pi}_{j,\ell}^{(u)} = \mathrm{softmax}\Big(g_\omega^\ell(\mathbf{h}_{j,\ell}^{(u)}; \xi_u)\Big) \in \Delta^{K_s-1}, \tag{2}$$

where ω denotes the recommender parameters, ξ_u specifies the decoding information required at step u and Δ^{K_s-1} denotes the probability simplex over the valid code tokens. These distributions guide the decoding or denoising process, ultimately yielding a predicted code sequence $\hat{\boldsymbol{s}}_j = (\hat{s}_j^1, \dots, \hat{s}_j^{L_q})$ for each target item slot j. Alongside the predicted code sequence, we form a continuous recommendation preference representations by stacking the contextual hidden states in Semantic-ID token order as $\boldsymbol{H}_j = [\mathbf{h}_{j,1}^{(u_{j,1})}, \dots, \mathbf{h}_{j,L_q}^{(u_{j,L_q})}]^\top \in \mathbb{R}^{L_q \times d_r}$, where $u_{j,\ell}$ denotes the prediction step for token ℓ of target item slot j along the selected decoding trajectory.

3.3 Retrieval-to-Generation Mapping

3.3.1 Factorized Feature Predictor

We introduce the Factorized Feature Predictor (FFP) module to map the contextual target-item state $\boldsymbol{H} \in \mathbb{R}^{L_q \times d_r}$ to content, description, and collaborative features through $G_\psi(\boldsymbol{H}) = (\hat{\boldsymbol{e}}^c, \hat{\boldsymbol{e}}^t, \hat{\boldsymbol{e}}^f)$. We first normalize and linearly project $\boldsymbol{H}$, then add learned position embeddings to form the memory $\boldsymbol{M} \in \mathbb{R}^{L_q \times d_q}$. Six learnable queries $\boldsymbol{Q}^{(0)} \in \mathbb{R}^{6 \times d_q}$ interact through self-attention, read the memory through cross-attention, and undergo feed-forward transformations across D_q blocks. The output queries are assigned to three pairs indexed by $\mathcal{I}_a$, one for each factor $a \in \mathcal{F} = \{c, t, f\}$. Each pair is averaged and mapped to $\hat{\boldsymbol{e}}^a \in \mathbb{R}^{d_a}$ through a separate affine head h_a:

$$\boldsymbol{Q}^{(D_q)} = \mathrm{QFormer}_\psi(\boldsymbol{Q}^{(0)}, \boldsymbol{M}), \quad \hat{\boldsymbol{e}}^a = h_a\big(\tfrac{1}{2}\textstyle\sum_{j \in \mathcal{I}_a} \boldsymbol{q}_j^{(D_q)}\big). \tag{3}$$

Here, $\boldsymbol{q}_j^{(D_q)}$ denotes the jth output query. The query pairs extract hidden latent derived from the retrieval space and project them into the visual generation space. Shared attention preserves the common recommendation context, while branch-wise supervision aligns each output with its corresponding feature, grounding the latent-space mapping in explicit semantics for visual generation.

Training objective. We begin with a warm-up stage in which the FFP learns to predict target-item features in the visual generation space while preserving candidate discrimination. Accordingly, we combine an MSE loss for feature reconstruction with a ranking loss for recommendation relevance.

For a training target i^+, the MSE loss penalizes the difference between each recovered feature $\hat{\boldsymbol{e}}^a$ and its fixed catalog target $\boldsymbol{e}^a_{i^+}$. The squared error is averaged over the d_a dimensions of each factor. This objective anchors the latent-space mapping to the coordinates and magnitudes of the target item's catalog features, providing feature-level supervision for reversion. The ranking loss complements feature reconstruction by encouraging the positive target to score higher than competing candidates. We augment the upstream retrieval pool with the target, $\mathcal{A} = \mathcal{A}_{\text{ret}} \cup \{i^+\}$. For each factor, temperature-scaled cosine scores $\ell_a(i) = \cos(\hat{\boldsymbol{e}}^a, \boldsymbol{e}^a_i)/\tau_a$, with $\tau_a > 0$, define the candidate distribution $p_a(i) = \exp(\ell_a(i))/\sum_{j\in\mathcal{A}} \exp(\ell_a(j))$. Minimizing cross-entropy for the positive target supervises relative angular similarity within this pool. The two losses are

$$\mathcal{L}_{\text{mse}} = \sum_{a\in\mathcal{F}} \tfrac{1}{d_a} \|\hat{\boldsymbol{e}}^a - \boldsymbol{e}^a_{i^+}\|_2^2, \quad \mathcal{L}_{\text{rank}} = -\sum_{a\in\mathcal{F}} \gamma_a \log p_a(i^+). \tag{4}$$

where γ_a weights factor a. The overall objective is:

$$\mathcal{L}_{\text{rev}}(\psi) = \mathbb{E}_{(\mathcal{C},i^+)}\big[\lambda_{\text{mse}}\mathcal{L}_{\text{mse}} + \lambda_{\text{rank}}\mathcal{L}_{\text{rank}}\big], \tag{5}$$

Given the contextual target-item state $\boldsymbol{H}$, FFP outputs the recovered content, description, and collaborative features $(\hat{\boldsymbol{e}}^c, \hat{\boldsymbol{e}}^t, \hat{\boldsymbol{e}}^f)$. In our final configuration, the recovered content feature $\hat{\boldsymbol{e}}^c$ is passed to the subsequent visual generation module to form its pooled conditioning.

3.3.2 Dual-Branch Visual Generator

We propose Dual-Branch Visual Generator (DVG) to map recommendation representations into visual-token distributions for both catalog ranking and image synthesis.

Visual-token tokenization. We use the TA-TiTok tokenizer (Kim et al., 2025) to encode and quantize each catalog image x_i into a discrete visual-token sequence $\mathbf{v}_i = E_v(x_i) = (v_i^1, \ldots, v_i^{L_v})$, where $v_i^r \in \{1, \ldots, K_v\}$, L_v is the sequence length, and K_v is the visual codebook size. We precompute these sequences as generation targets and candidate representations for visual-token ranking.

Feature conditioning. The recovered content feature $\hat{\boldsymbol{e}}^c$ supplies a global condition describing the target item's inferred visual characteristics. We normalize the content feature and project it through a two-layer MLP ϕ_c, while a separate mapping ϕ_{time} projects the sinusoidal timestep encoding into the same d_v-dimensional space. We then add the two embeddings element-wise to form condition $\mathbf{c}_\tau$:

$$\mathbf{c}_\tau = \phi_c(\hat{\boldsymbol{e}}^c) + \phi_{\text{time}}(\tau) \in \mathbb{R}^{d_v}. \tag{6}$$

Here, τ controls the masking ratio during training and reflects the current masking state during inference. Each Transformer block maps $\mathbf{c}_\tau$ to channel-wise scale and shift parameters for AdaLN. This condition guides visual-token prediction with the recovered item characteristics while adapting the network's transformations to the visual information available at each refinement step.

Masked token denoising. We adopt the joint-attention architecture to progressively recover masked visual tokens through an image branch and a recommendation latent branch. At each denoising step, the image branch receives a partially masked sequence $\tilde{\mathbf{v}}_\tau$, where $\tilde{v}_\tau^r = \texttt{[MASK]}$ for $r \in \mathcal{M}_\tau$ and the remaining positions retain their available visual tokens. The two branches are initialized as

$$\mathbf{V}^{(0)} = \text{Emb}_v(\tilde{\mathbf{v}}_\tau) + \mathbf{P}_v, \qquad \mathbf{U}^{(0)} = \text{Linear}_h(\boldsymbol{H}), \tag{7}$$

where $\mathbf{V}^{(0)} \in \mathbb{R}^{L_v\times d_v}$ contains both visual-token and mask-token embeddings, and $\mathbf{U}^{(0)} \in \mathbb{R}^{L_q\times d_v}$ preserves the contextual recommendation information.

The branches interact across D_g Transformer blocks. Within each block, let $\mathbf{X}_{\text{img}}$ and $\mathbf{X}_{\text{lat}}$ denote their input states. The shared condition $\mathbf{c}_\tau$ produces separate AdaLN parameters and residual gates for each branch. Writing $\mathcal{N}_a^s(\cdot; \mathbf{c}_\tau)$ for the corresponding AdaLN, where $a \in \{\text{img}, \text{lat}\}$ and $s \in \{\text{attn}, \text{ffn}\}$, joint attention computes

$$\bar{\mathbf{X}}_a = \mathcal{N}_a^{\text{attn}}(\mathbf{X}_a; \mathbf{c}_\tau), \qquad (\mathbf{O}_{\text{img}}, \mathbf{O}_{\text{lat}}) = \text{JointAttn}(\bar{\mathbf{X}}_{\text{img}}, \bar{\mathbf{X}}_{\text{lat}}). \tag{8}$$

Each branch uses its own query, key, and value projections, which are concatenated across branches for attention. This allows masked visual positions to attend to both available visual tokens and recommendation latents. The final block updates only the image branch. After the last block, a conditioned normalization and vocabulary head convert the image states into visual-token logits:

$$\mathbf{Z}_\theta = \mathrm{Head}\Big(\mathrm{AdaLN}_{\mathrm{out}}(\mathbf{V}^{(D_g)}; \mathbf{c}_\tau)\Big), \qquad p_{\theta,r} = \mathrm{softmax}(\mathbf{Z}_{\theta,r}), \tag{9}$$

where $\mathbf{Z}_\theta \in \mathbb{R}^{L_v \times K_v}$. Each $p_{\theta,r}$ predicts a distribution over visual codewords, excluding the mask token. During generation, masked positions are sampled from these predictions; higher-confidence tokens are retained, while the remaining positions stay masked for the next step. The condition remains shared across blocks within a step and is updated with the timestep between steps. Repeating this process progressively replaces mask tokens with visual tokens until the sequence is complete.

Training objective. We supervise token prediction with cross-entropy, supplemented by knowledge distillation. To obtain teacher predictions, we run the same predictor without gradients, replacing the recovered feature $\hat{\boldsymbol{e}}^c$ with the ground-truth feature $\boldsymbol{e}^c_{i^+}$. A KL-divergence loss aligns the student's softened predictions with the teacher's at masked positions. The two losses are defined as follows:

$$\mathcal{L}_{\mathrm{ce}} = \frac{\sum_r w_r \, \mathrm{CE}_\epsilon(p_{\theta,r}, v_r)}{\sum_r w_r}, \qquad \mathcal{L}_{\mathrm{kd}} = \frac{T_d^2}{|\mathcal{M}|} \sum_{r \in \mathcal{M}} \mathrm{KL}\big(\mathrm{sg}[p_r^{\mathrm{gt},T_d}] \,\|\, p_r^{\mathrm{pred},T_d}\big). \tag{10}$$

Here, CE_ϵ is label-smoothed cross-entropy, and w_r equals 1 for masked tokens and $\eta < 1$ otherwise. $\mathcal{M}$ indexes masked positions in the minibatch. We soften predictions as $p_r^{a,T_d} = \mathrm{softmax}(\mathbf{Z}_r^a / T_d)$, where T_d is the temperature and sg stops gradients. We jointly train FFP and DVG module by combining $\mathcal{L}_{\mathrm{rev}}$ in Eq. equation 5 with the cross-entropy and distillation losses:

$$\mathcal{L}_{\mathrm{vis}} = \lambda_{\mathrm{ce}}\mathcal{L}_{\mathrm{ce}} + \lambda_{\mathrm{rev}}\mathcal{L}_{\mathrm{rev}} + \lambda_{\mathrm{kd}}\mathcal{L}_{\mathrm{kd}}. \tag{11}$$

Inference. We initialize the image-branch input $\tilde{\mathbf{v}}_\tau$ as a fully masked visual-token sequence. We then apply classifier-free guidance (CFG) (Ho & Salimans, 2022) with the condition $\mathbf{c}_\tau$ to strengthen semantic guidance during visual-token prediction. The resulting token distributions support catalog ranking, while the predicted token sequences are decoded into personalized item images.

3.4 Image Generation.

Item Retrieval. We score each candidate i by the average log probability of its visual tokens, $s(i \mid \mathcal{C}) = L_v^{-1} \sum_{r=1}^{L_v} \log p_r^{\mathrm{rank}}(v_i^r)$, where $p_r^{\mathrm{rank}}(v_i^r)$ is the probability assigned to candidate i's visual token at the corresponding position r. We rank candidates by descending score, prioritizing items that are more visually consistent with the recommendation context.

Visual Decoding. For image synthesis, we decode the visual-token sequence $\hat{\mathbf{v}}$ as $\hat{x} = D_v(\hat{\mathbf{v}})$, where D_v is a frozen decoder. Decoding the token sequence for each target slot yields $\hat{\mathcal{X}}_y = \{\hat{x}_1, \ldots, \hat{x}_m\}$ for bundle completion or $\hat{x}_{t+1}$ for sequential recommendation. The generated sequences can form new code combinations, enabling image synthesis beyond existing catalog items.

4 Experiments

We conduct experiments on bundle recommendation and sequential recommendation, aimming to answer the following research questions (RQs): **RQ1:** Do mapped visual-token representations in the visual generation space improve catalog-ranking performance over preference representations in the retrieval space? **RQ2:** What are the effects of the FFP module, warm-up stage, loss design, and CFG in our proposed latent space mapping architecture? **RQ3:** How do item images generated by R2G-Map compare qualitatively with those produced by state-of-the-art fashion generative visual recommendation models in terms of visual coherence, personalization, and outfit compatibility?

4.1 Experimental settings

Datasets. We construct two task-specific datasets from POG (Chen et al., 2019), a fashion dataset collected from Alibaba iFashion. For bundle recommendation, following DDBC (Tu et al., 2026),

Table 1: Performance under negative sampling and all-item ranking. Improv. denotes relative improvement over the corresponding base model; R2G-Map(DDBC) denotes R2G-Map using DDBC-derived preference representations.

Model	Negative Sampling Ranking							All Ranking						
	Bundle Rec.			Sequential Rec.				Bundle Rec.			Sequential Rec.			
	F1↑	Jacc.↑	OAS↑	R@5↑	R@10↑	N@10↑	N@20↑	F1↑	Jacc.↑	OAS↑	R@5↑	R@10↑	N@10↑	N@20↑
CLHE	0.138	0.085	0.446	0.227	0.287	0.214	0.266	0.023	0.016	0.428	0.025	0.032	0.021	0.024
SASRec	0.178	0.129	0.477	0.234	0.307	0.231	0.337	0.039	0.028	0.442	0.037	0.041	0.032	0.038
TIGER	0.204	0.150	0.442	0.347	0.442	0.338	0.382	0.104	0.071	0.488	0.139	0.147	0.170	0.172
DDBC	0.139	0.098	0.471	0.318	0.436	0.298	0.344	0.072	0.048	0.475	0.101	0.104	0.105	0.107
DiFashion	0.062	0.042	0.342	0.135	0.209	0.124	0.152	0.021	0.014	0.325	0.022	0.023	0.023	0.024
FashionDPO	0.063	0.042	0.343	0.131	0.204	0.121	0.150	0.027	0.018	0.327	0.024	0.025	0.025	0.026
R2G-Map(DDBC)	0.311	0.241	0.527	0.528	0.568	0.502	0.523	0.136	0.095	0.491	0.215	0.241	0.203	0.210
Improv. +	*123.7%*	*145.9%*	*11.9%*	*66.0%*	*30.3%*	*68.5%*	*52.0%*	*88.9%*	*97.9%*	*3.4%*	*112.9%*	*131.7%*	*93.3%*	*96.3%*
R2G-Map(TIGER)	**0.342**	**0.254**	**0.530**	**0.537**	**0.588**	**0.510**	**0.528**	**0.140**	**0.100**	**0.494**	**0.228**	**0.250**	**0.220**	**0.227**
Improv. +	*67.6%*	*69.3%*	*19.9%*	*54.8%*	*33.0%*	*50.9%*	*38.2%*	*34.6%*	*40.8%*	*1.2%*	*64.0%*	*70.1%*	*29.4%*	*32.0%*

we sample four-item subsequences from longer outfits, using two observed items to predict the other two. The training/validation/test splits contain 29,703/1,303/2,521 instances. For sequential recommendation, we retain records containing four or five items and representing them as item sequences. The corresponding splits contain 50,464/1,285/2,483 sequences. The two settings therefore differ in sample construction, context length, and target definition.

Baselines. For generative retrieval recommendation, we compare against CLHE (Ma et al., 2024), SASRec (Kang & McAuley, 2018), TIGER (Rajput et al., 2023), and DDBC (Tu et al., 2026). We follow the settings in their respective open-source implementations. For generative visual recommendation, we compare against DiFashion (Xu et al., 2024) and FashionDPO (Yu et al., 2025). Since the datasets do not contain user interaction histories, we remove user-preference conditioning from both models while retaining category-text conditioning. For FashionDPO, we construct preference pairs using a ranking-based strategy. We exclude DualFashion (Yu et al., 2026) because the atasets lack the structured captions required by its coupled image and text branches.

Evaluation Metrics. For generative retrieval recommendation, we evaluate bundle recommendation following DDBC (Tu et al., 2026), using F1, Jaccard and optimal-assignment similarity (OAS). OAS measures semantic similarity between predicted and held-out items as the mean cosine similarity of their catalog content features. For sequential recommendation, we report Recall@K for $K \in \{5, 10\}$ and NDCG@K for $K \in \{10, 20\}$. For generative visual recommendation, we evaluate image quality using Inception Score (IS) and perceptual similarity using LPIPS. For compatibility (Comp.), we adapt the evaluation prompt template from DualFashion (Yu et al., 2026) and employ Gemini 3.5 Flash as a judge to assess how well the generated items complement the given context.

Implementation Details. The generative retrieval backbones remain frozen throughout training. TA-TiTok encodes each 256×256 image into 128 discrete tokens using an 8,192-entry codebook. During FFP module's training, we set the ranking loss weight to $\lambda_r = 0.1$ and the ranking temperature to 0.1, while assigning equal supervision weights to all feature modalities. For `feature-fused`, we combine the standardized cosine-similarity scores of content, description, and collaborative features with weights $(0.575, 0.2, 0.225)$. During joint training, we set the feature-supervision weight to $\lambda_p = 0.05$ and apply knowledge distillation with weight $\lambda_{\mathrm{KD}} = 0.2$ and temperature 2.0. At inference, we generate images in 16 sampling steps with a classifier-free guidance scale of 6.

4.2 Quantitative Results

Retrieval performance. (RQ1) Table 1 shows that R2G-Map improves both backbones under negative sampling and all ranking. For bundle recommendation, R2G-Map(TIGER) improves F1 over TIGER by 67.6% and 34.6% under the two protocols, respectively, while R2G-Map(DDBC) achieves corresponding gains of 123.7% and 88.9% over DDBC. These gains show that R2G-Map improves recommendation through latent-space mapping, while mapping retrieval representations into the visual representation space further enables visual generation. Compared with generative visual recommendation baselines, both R2G-Map variants outperform DiFashion and FashionDPO

Table 3: Ablation study on retrieval and visual generation metrics across recommendation tasks. Best results are bold, and second-best results are underlined.

Model	Ranking-based Metric							Visual-based Metric					
	Bundle Rec.			Sequential Rec.				Bundle Rec.			Sequential Rec.		
	F1↑	Jacc.↑	OAS↑	R@5↑	R@10↑	N@10↑	N@20↑	IS↑	LPIPS↓	Comp.↑	IS↑	LPIPS↓	Comp.↑
w/o FFP module	0.291	0.210	0.519	0.524	0.561	0.492	0.513	12.015	0.611	4.920	10.935	0.566	5.130
w/o Warm-up	0.319	0.234	0.525	0.533	0.570	0.498	0.510	24.944	0.598	5.450	20.738	0.559	5.600
w/o L_{mse}	0.316	0.228	0.525	0.526	0.568	0.497	0.505	27.431	0.598	5.450	20.674	0.558	5.620
w/o Ranking Loss	0.321	0.221	0.520	0.531	0.578	0.506	0.515	28.141	0.597	5.480	21.651	0.552	5.650
w/o KL	0.301	0.212	0.511	0.517	0.554	0.493	0.503	28.977	0.595	5.490	21.715	0.549	5.650
w/o CE	0.009	0.007	0.249	0.037	0.066	0.033	0.051	5.499	0.920	4.200	5.089	1.015	4.430
w/o CFG	0.339	0.250	0.529	0.527	0.563	0.508	0.518	21.573	0.598	5.510	19.876	0.546	5.600
R2G-Map(TIGER)	**0.342**	**0.254**	**0.530**	**0.537**	**0.588**	**0.510**	**0.528**	**30.399**	**0.594**	**6.220**	**26.154**	**0.541**	**6.710**

across tasks and ranking settings. This suggests that R2G-Map effectively leverages the preference representations learned by *GenRet* to better capture collaborative signals from user–item interactions.

Visual generation performance. Table 2 compares image generation performance across the two recommendation tasks. Both R2G-Map variants outperform DiFashion and FashionDPO on all three metrics. For bundle recommendation, R2G-Map(TIGER) improves IS and compatibility over DiFashion by 7.84% and 22.44%, respectively. For sequential recommendation, it raises IS from 24.970 to 26.154 and compatibility from 6.180 to 6.710. R2G-Map(TIGER) also consistently outperforms R2G-Map(DDBC). Together with the ranking results, these findings show that R2G-Map improves ranking performance while also generating high-quality, compatible item images.

Table 2: Comparison of image generation performance across recommendation tasks.

Model	Bundle Rec.			Sequential Rec.		
	IS↑	LPIPS↓	Comp.↑	IS↑	LPIPS↓	Comp.↑
DiFashion	28.189	0.604	5.080	24.970	0.611	6.180
FashionDPO	25.916	0.603	5.120	25.090	0.623	6.500
R2G-Map(DDBC)	29.637	0.601	6.050	25.109	0.565	6.590
R2G-Map(TIGER)	**30.399**	**0.594**	**6.220**	**26.154**	**0.541**	**6.710**

Ablation Study. (RQ2) Table 3 shows that R2G-Map(TIGER) performs best across all metrics. Removing FFP sharply reduces bundle and sequential IS to 12.015 and 10.935, respectively, and degrades ranking, highlighting the importance of visual conditioning in bridging the two latent spaces. Without FFP warm-up also hurts both ranking and generation. For loss design, CE is essential: removing it reduces bundle F1 to 0.009 and sequential NDCG@10 to 0.033, while severely degrading visual quality. Removing L_{mse}, Ranking Loss, or KL also hurts performance across both tasks. Finally, removing CFG substantially reduces IS and compatibility while producing only minor changes in ranking performance, indicating that CFG primarily benefits visual generation.

4.3 Qualitative Results (RQ3)

Figure 2 presents a qualitative comparison on bundle recommendation, where the context consists of a dark dress and white footwear, and the target items are star-shaped earrings and a handbag. DiFashion and FashionDPO generate earrings and handbags, consistent with their category-text conditioning. Without explicit target-category prompts, R2G-Map(DDBC) generates a watch and a light-colored hat, while R2G-Map(TIGER) produces dangling earrings and a straw hat. These visual results illustrate how R2G-Map maps continuous recommendation states into visual generation space through factorized feature predictor and dual-branch visual generator. Although the generated items differ in category from the targets,



Figure 2: Qualitative comparison of image generation performance on bundle recommendation.

Table 4: Comparison of generation backbones in terms of ranking, visual quality, and efficiency.

Module	Ranking-based Metrics			Visual-based Metrics			Computational Efficiency		
	F1↑	Jacc.↑	OAS↑	IS↑	LPIPS↓	Comp.↑	Train Time↓(min)	Params.(M)	Throughput↑(img/s)
SD	0.050	0.033	0.331	16.254	0.614	5.910	43.70	2119.10	2.349
DVG	0.311	0.241	0.527	29.637	0.601	6.050	8.76	1003.98	17.348

removing category constraints encourages the model to rely less on explicit prompts and make better use of collaborative signals, consistent with the substantial ranking gains reported in Table 1.

4.4 Model Study

Computational efficiency. To evaluate the training and inference efficiency of our generation backbone, we replace the DVG model with Stable Diffusion (SD) (Esser et al., 2024) under the same bundle recommendation setting. Training time is amortized per 1,000 updates with an effective batch size of 128. As shown in Table 4, DVG also consistently outperforms SD on ranking-based metrics, improving ranking-based metrics, while achieving better visual quality with higher IS and compatibility and lower LPIPS. Beyond effectiveness, DVG reduces the loaded parameter count by 52.6% and achieves 17.348 images/s, compared with 2.349 images/s for SD, yielding a 7.39× throughput improvement. Its recorded training time is also lower at 8.76 minutes per 1,000 updates.

Preference representation analysis. Preference representation influences how much recommendation information is mapped to visual space. Semantic-ID tokens compress predictive distributions into discrete codes, losing uncertainty and fine-grained item distinctions. To examine this information loss, we compare Semantic-ID tokens with the continuous hidden latents. As shown in Table 5, hidden latents improve most ranking and visual metrics, indicating that they preserve richer recommendation signals for visual generation.

Table 5: Comparison of preference representations for bundle recommendation.

Representation	Ranking-based Metrics			Visual-based Metrics		
	F1↑	Jacc.↑	OAS↑	IS↑	LPIPS↓	Comp.↑
Semantic-ID token	0.295	0.227	0.540	23.004	0.644	5.140
hidden latent	0.311	0.241	0.527	29.637	0.601	6.050

Prediction preservation and correction. We further examine whether the mapped hidden latent preserves correct predictions from the original *GenRet* model while correcting its retrieval errors. For each target position, we define a *code hit* when *GenRet's* Top-1 semantic-code prediction matches either target item, and a *visual hit* when the Top-1 prediction after latent mapping and visual ranking matches either target item; otherwise, the prediction is treated as a miss. As shown in Figure 3, visual ranking substantially improves Top-1 prediction accuracy after latent mapping. And the mapping largely preserves correct semantic-code predictions while recovering a considerable portion of the original retrieval errors.

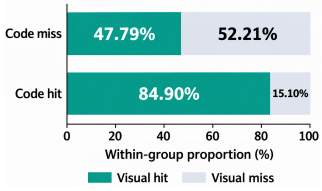


Figure 3: Hit transitions from retrieval to visual ranking.

5 Conclusion And Future Work

We presented R2G-Map, which efficiently and effectively integrating *GenRet* and *PVGen* through latent space mapping. FFP predicts factorized features to provide visual-semantic guidance, while DVG incorporates continuous preference representations into visual-token generation through joint attention. This design supports both catalog ranking and personalized image synthesis. Experiments on bundle and sequential recommendation demonstrate that R2G-Map improves ranking performance while generating visually coherent and compatible item images.

In the future work, we will explore a broader range of retrieval and visual generation backbones and evaluate R2G-Map across more diverse recommendation scenarios. We will also investigate recommendation infrastructure to reduce inference latency and better support real-world deployment. Finally, we plan to extend R2G-Map beyond visual content generation to other modalities, such as text, audio, and video, to further examine its generalizability.

AI USE STATEMENT

In this work, we used generative AI tools for aiding or polishing writing. We have not used generative AI tools for finding related work and research ideas, and generating synthetic datasets are not applicable to this work. Additionally, we used generative AI tools for troubleshooting LaTeX errors. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To support reproducibility, we have publicly released the inference code, evaluation metric implementations, and the test datasets and checkpoints for both bundle recommendation and sequential recommendation: `https://anonymous.4open.science/r/R2G-Map_Anonymous-7311/README.md`. These resources support reproducing inference and evaluation using the released checkpoints. We will extend the release with training and validation code upon publication.

REFERENCES

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative image transformer. In *CVPR*, pp. 11305–11315. IEEE, 2022.

Wen Chen, Pipei Huang, Jiaming Xu, Xin Guo, Cheng Guo, Fei Sun, Chao Li, Andreas Pfadler, Huan Zhao, and Binqiang Zhao. POG: personalized outfit generation for fashion recommendation at alibaba ifashion. In *KDD*, pp. 2662–2670. ACM, 2019.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, pp. 12873–12883. Computer Vision Foundation / IEEE, 2021.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*. OpenReview.net, 2024.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *CoRR*, abs/2207.12598, 2022.

Yupeng Hou, Jiacheng Li, Ashley Shin, Jinsung Jeon, Abhishek Santhanam, Wei Shao, Kaveh Hassani, Ning Yao, and Julian J. McAuley. Generating long semantic ids in parallel for recommendation. In *KDD (2)*, pp. 956–966. ACM, 2025.

Clark Mingxuan Ju, Liam Collins, Leonardo Neves, Bhuvesh Kumar, Louis Yufeng Wang, Tong Zhao, and Neil Shah. Generative recommendation with semantic ids: A practitioner's handbook. In *CIKM*, pp. 6420–6425. ACM, 2025.

Wang-Cheng Kang and Julian J. McAuley. Self-attentive sequential recommendation. In *ICDM*, pp. 197–206. IEEE Computer Society, 2018.

Wang-Cheng Kang, Chen Fang, Zhaowen Wang, and Julian J. McAuley. Visually-aware fashion recommendation and design with generative image models. In *ICDM*, pp. 207–216. IEEE Computer Society, 2017.

Dongwon Kim, Ju He, Qihang Yu, Chenglin Yang, Xiaohui Shen, Suha Kwak, and Liang-Chieh Chen. Democratizing text-to-image masked generative models with compact text-aware one-dimensional tokens. In *ICCV*, pp. 18442–18452. IEEE, 2025.

Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *WWW*, pp. 689–698. ACM, 2018.

Chang Liu, Yimeng Bai, Xiaoyan Zhao, Yang Zhang, Fuli Feng, and Wenge Rong. Discrec: Disentangled semantic-collaborative modeling for generative recommendation. *CoRR*, abs/2506.15576, 2025.

Yunshan Ma, Xiaohao Liu, Yinwei Wei, Zhulin Tao, Xiang Wang, and Tat-Seng Chua. Leveraging multimodal features and item-level user feedback for bundle construction. In *WSDM*, pp. 510–519. ACM, 2024.

Maryam Moosaei, Yusan Lin, Ablaikhan Akhazhanov, Huiyuan Chen, Fei Wang, and Hao Yang. Outfitgan: Learning compatible items for generative fashion outfits. In *CVPR Workshops*, pp. 2272–2276. IEEE, 2022.

Ziqi Pang, Tianyuan Zhang, Fujun Luan, Yunze Man, Hao Tan, Kai Zhang, William T. Freeman, and Yu-Xiong Wang. Randar: Decoder-only autoregressive visual generation in random orders. In *CVPR*, pp. 45–55. Computer Vision Foundation / IEEE, 2025.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Mahesh Sathiamoorthy. Recommender systems with generative retrieval. In *NeurIPS*, 2023.

Xiaoteng Shen, Rui Zhang, Xiaoyan Zhao, Jieming Zhu, and Xi Xiao. PMG : Personalized multimodal generation with large language models. In *WWW*, pp. 3833–3843. ACM, 2024.

Teng Tu, Ai Li, Yunshan Ma, Shuo Xu, Xiaohao Liu, Haokai Ma, Liang Pang, and Tat-Seng Chua. Discrete diffusion for bundle construction. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=dKyhgfe50H.

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NIPS*, pp. 6306–6315, 2017.

Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, and Ping Luo. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *CVPR*, pp. 12966–12977. Computer Vision Foundation / IEEE, 2025.

Xiang Xu, Hao Wang, Wei Guo, Luankang Zhang, Wanshan Yang, Runlong Yu, Yong Liu, Defu Lian, and Enhong Chen. Multi-granularity interest retrieval and refinement network for long-term user behavior modeling in CTR prediction. In *KDD (1)*, pp. 2745–2755. ACM, 2025a.

Yiyan Xu, Wenjie Wang, Fuli Feng, Yunshan Ma, Jizhi Zhang, and Xiangnan He. Diffusion models for generative outfit recommendation. In *SIGIR*, pp. 1350–1359. ACM, 2024.

Yiyan Xu, Wenjie Wang, Yang Zhang, Biao Tang, Peng Yan, Fuli Feng, and Xiangnan He. Personalized image generation with large multimodal models. In *WWW*, pp. 264–274. ACM, 2025b.

Yiyan Xu, Jinghao Zhang, Alireza Salemi, Xinting Hu, Wenjie Wang, Fuli Feng, Hamed Zamani, Xiangnan He, and Tat-Seng Chua. Personalized generation in large model era: A survey. In *ACL (1)*, pp. 24607–24649. Association for Computational Linguistics, 2025c.

Mingzhe Yu, Yunshan Ma, Lei Wu, Kai Cheng, Xue Li, Lei Meng, and Tat-Seng Chua. Smart fitting room: A one-stop framework for matching-aware virtual try-on. In *ICMR*, pp. 184–192. ACM, 2024a.

Mingzhe Yu, Yunshan Ma, Lei Wu, Changshuo Wang, Xue Li, and Lei Meng. Fashiondpo: Fine-tune fashion outfit generation model using direct preference optimization. In *SIGIR*, pp. 212–222. ACM, 2025.

Mingzhe Yu, Lei Wu, Qianru Sun, and Yunshan Ma. Dual-diffusional generative fashion recommendation. In *SIGIR*, pp. 2264–2273. ACM, 2026.

Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. In *NeurIPS*, 2024b.

Yu Zheng, Chen Gao, Jianxin Chang, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. Disentangling long and short-term interests for recommendation. In *WWW*, pp. 2256–2267. ACM, 2022.

A APPENDIX

A.1 DETAILED EXPERIMENT SETTINGS.

Datasets. We construct two task-specific datasets from POG (Chen et al., 2019), a fashion dataset collected from Alibaba iFashion. For bundle recommendation, following DDBC (Tu et al., 2026), we discard outfits with fewer than four items and sample four-item subsequences from longer outfits, using two observed items to predict the remaining two. The training/validation/test splits contain 29,703/1,303/2,521 instances, respectively. For sequential recommendation, we retain records containing four or five items and representing them as item sequences. Each context contains three observed items, while the remaining one or two items form the ground-truth target set. The corresponding splits contain 50,464/1,285/2,483 sequences. The two settings therefore differ in sample construction, context length, and target definition: bundle recommendation completes a partial outfit by predicting the missing items, whereas sequential recommendation predicts the next item from contexts.

Baselines. For generative retrieval recommendation, we compare against CLHE (Ma et al., 2024), SASRec (Kang & McAuley, 2018), TIGER (Rajput et al., 2023), and DDBC (Tu et al., 2026). We use CLHE-derived fused item embeddings for the retrieval backbones and follow the model-specific settings in their respective open-source implementations, retaining the required representation and tokenization procedures. We also instantiate R2G-Map with both TIGER and DDBC to examine its applicability to different generative retrieval backbones. For generative visual recommendation, we compare against DiFashion (Xu et al., 2024) and FashionDPO (Yu et al., 2025). Since the datasets do not contain user interaction histories, we remove user-preference conditioning from both models while retaining category-text conditioning. For FashionDPO, we construct preference pairs using a ranking-based strategy, treating higher-ranked candidates as preferred over lower-ranked alternatives under the same recommendation context. We exclude DualFashion (Yu et al., 2026) because the structured attribute-level captions required by its coupled image and text branches are unavailable in our task datasets. Given the interdependence of its image and text branches, the absence of this information would substantially impair its recommendation performance.

Evaluation Metrics. We evaluate retrieval performance under both negative sampling ranking and all ranking protocols. All ranking evaluation ranks all 31,217 catalog items for each recommendation context. Negative sampling ranking evaluation ranks the supplied candidates for each context, inserting any missing ground-truth items to ensure complete target coverage. The latter therefore measures reranking performance conditional on target inclusion. We report the two protocols separately, using the same metric definitions and retaining all ground-truth targets in both settings. For bundle recommendation, following DDBC (Tu et al., 2026), we report F1, Jaccard, and optimal-assignment similarity (OAS). F1 and Jaccard measure item-level overlap between predicted and held-out sets. OAS measures semantic similarity between predicted and held-out items as the mean cosine similarity of their catalog content features. For sequential recommendation, we report Recall@K for $K \in \{5, 10\}$ and NDCG@K for $K \in \{10, 20\}$. Both metrics use the complete ground-truth set and are averaged over contexts. Thus, retrieving one relevant item contributes $1/2$ to Recall when a context has two ground-truth targets, and NDCG is normalized by the ranking of all available targets.

For generative visual recommendation, we evaluate image quality using Inception Score (IS) and paired perceptual similarity using LPIPS. Our implementation computes IS with a POG-finetuned 50-category Inception classifier and a uniform reference distribution: $\mathrm{IS} = \exp(\mathbb{E}_x\left[D_{\mathrm{KL}}(p(y \mid x) \,\|\, U_{50})\right])$, where U_{50} denotes the uniform distribution over the 50 categories. LPIPS uses the AlexNet-based perceptual network and compares generated images with their corresponding ground-truth images at 256×256 resolution. For bundle recommendation, image evaluation covers both target items. For sequential recommendation, we generate one item per context and use the first ground-truth item as the paired image reference, while retaining all ground-truth items for retrieval evaluation. For compatibility (Comp.), we adapt the evaluation prompt template, as shown in Figure 4, from DualFashion (Yu et al., 2026) and employ Gemini 3.5 Flash to assess the outfit formed by the observed context and generated items. The judge assigns an integer score from 1 to 10 based on style, color, category, material, and occasion compatibility, and Comp. is the mean score over valid responses. Higher values indicate better performance for all reported metrics except LPIPS.

You are evaluating visual compatibility for a fashion bundle.

Images are provided in this order:

1. Fixed input item images already present in the bundle.
2. Generated target item images proposed to complete the same bundle.

If no fixed input item images are provided, judge the mutual compatibility of the generated target item images as the candidate bundle completion.

Task:

Classify the compatibility of the complete bundle formed by all fixed input items plus all generated target items. This is a 10-class classification task. Use exactly one integer class from 1 to 10.

Class meaning:

1. Completely incompatible; severe style/category/color/occasion conflicts.
2. Very poor compatibility; most items clash.
3. Poor compatibility; several major conflicts.
4. Below average; noticeable mismatch but some usable elements.
5. Weakly acceptable; mixed compatibility with clear issues.
6. Moderately compatible; mostly coherent with minor issues.
7. Good compatibility; coherent outfit/bundle with small imperfections.
8. Very good compatibility; strong harmony in style, color, category, and occasion.
9. Excellent compatibility; highly coherent and visually appealing.
10. Near-perfect compatibility; all items look intentionally designed together.

Judging criteria:

- Style and occasion consistency.
- Category complementarity among items.
- Color palette harmony and contrast.
- Material/texture compatibility.
- Whether generated target items visually fit with the fixed input items.
- Ignore small background artifacts unless they prevent judging the item.
- Do not reward similarity to a ground-truth image; only judge the complete input-plus-generated bundle shown here.

Return only valid JSON with this schema:

```
{
  "compatibility_class": 1,
  "confidence": 0.0,
  "rationale": "short reason, one sentence"
}
```

Figure 4: Text prompt and JSON response format for compatibility evaluation using Gemini 3.5 Flash. The judge evaluates the outfit formed by the observed context and generated items on a scale from 1 (completely incompatible) to 10 (near-perfect compatibility).

Implementation Details. The generative retrieval backbones and the TA-TiTok encoder and decoder remain frozen during R2G-Map training. TA-TiTok represents each 256×256 image using 128 discrete tokens from an 8,192-entry codebook. The FFP module uses six learnable queries and a four-layer Transformer with hidden dimension 256 and eight attention heads. It predicts content, description, and collaborative features with dimensions 768, 768, and 64, respectively. The visual-token generator contains 16 Transformer layers with hidden dimension 1,024 and 16 attention heads.

Training proceeds in two stages: In Stage-1, FFP training followed by joint optimization of FFP and the visual-token generator. During FFP training, we set the ranking loss weight to $\lambda_r = 0.1$ and the ranking temperature to 0.1, assigning equal supervision weights to the three feature modalities. For `feature-fused`, we standardize the cosine-similarity scores within each modality and combine content, description, and collaborative scores with weights $(0.575, 0.2, 0.225)$. During the first half of Stage-2 joint training, we set the feature-supervision weight to $\lambda_p = 0.05$ and use knowledge distillation with weight $\lambda_{\mathrm{KD}} = 0.2$ and temperature 2.0. During the second half, we reduce the feature-supervision weight to 0.02 and disable the distillation branch. The learning rate follows a cosine schedule and decays to 10^{-5}. The effective batch sizes per device are 1,280 target-item

Table 6: Comparison of conditioning features for DDBC-based bundle recommendation. Different predicted single feature serve as the visual conditioning input to the generator, and ranking performance is evaluated using the generated visual tokens.

Condition	Ranking-based Metric			Visual-based Metric		
	F1↑	Jacc.↑	OAS↑	IS↑	LPIPS↓	Comp.↑
feature_t	0.291	0.225	0.452	25.588	0.620	5.710
feature_c	0.311	0.241	0.527	29.637	0.601	6.050
feature_cf	0.268	0.204	0.434	21.624	0.632	5.120
feature_fused	0.303	0.226	0.515	26.523	0.619	5.930

Table 7: Bundle recommendation results using the fine-tuned TIGER model under negative sampling and all ranking. Predicted features are used directly to rank candidate items by comparison with their corresponding item features.

		Negative Sampling Ranking			All Ranking		
Setting	Ranking method	F1@1 ↑	Jacc@1 ↑	OAS ↑	F1@1 ↑	Jacc@1 ↑	OAS ↑
Fine-tuned TIGER	Original ranker	0.204	0.150	0.442	0.104	0.071	<u>0.488</u>
R2G-Map / Stage 1	feature_t	0.090	0.063	0.333	0.000	0.000	0.331
	feature_c	0.214	0.153	0.451	0.126	0.091	0.441
	feature_cf	0.160	0.112	0.393	0.090	0.062	0.381
	feature_fused	0.245	0.176	0.471	0.134	0.095	0.465
R2G-Map / Stage 2	feature_t	0.090	0.064	0.332	0.000	0.000	0.335
	feature_c	0.213	0.152	0.454	0.132	0.094	0.432
	feature_cf	0.172	0.122	0.398	0.091	0.064	0.382
	feature_fused	<u>0.256</u>	<u>0.185</u>	<u>0.473</u>	<u>0.135</u>	<u>0.098</u>	0.477
R2G-Map / Stage 2	Visual-token	**0.342**	**0.254**	**0.530**	**0.140**	**0.100**	**0.494**

examples for FFP training and 128 for joint training, using gradient accumulation where necessary. We use FP16 mixed precision, gradient clipping at a norm of 1.0, and an exponential moving average of model parameters with decay 0.999. The image generation process uses classifier-free guidance scale of 6 and a cosine guidance schedule.

Preference representation. To be noted, we use $\boldsymbol{H}_j$ as the continuous recommendation representation supplied to the visual bridge because hard token decisions can discard distinctions encoded in the contextual hidden states. To illustrate this, fix the decoding information ξ and consider the deterministic readout $d^{\ell}_{\omega}(\mathbf{h};\xi) = \arg\max_k \pi^{\ell}_{\omega}(k;\mathbf{h},\xi)$. Two hidden states may yield the same selected token while inducing different predictive distributions:

$$d^{\ell}_{\omega}(\mathbf{h}_1;\xi) = d^{\ell}_{\omega}(\mathbf{h}_2;\xi) \quad \not\Rightarrow \quad \boldsymbol{\pi}^{\ell}_{\omega}(\mathbf{h}_1;\xi) = \boldsymbol{\pi}^{\ell}_{\omega}(\mathbf{h}_2;\xi). \tag{12}$$

A lookup based only on the selected code cannot distinguish such states. Retaining $\boldsymbol{H}_j$ therefore gives the visual bridge access to contextual information beyond the discrete code sequence.

A.2 More results

Predicted features analysis. We use predicted features as conditioning inputs to the Dual-Branch Visual Generator to guide visual token generation. To examine their effects on recommendation and generation performance, we compare content features (feature_c), description features (feature_t), collaborative filtering features (feature_cf), and fused features (feature_fused) on DDBC-based bundle recommendation, as reported in Table 6.

Content conditioning achieves the best results across all six metrics. Compared with description conditioning, it improves F1 from 0.291 to 0.311 and IS from 25.588 to 29.637, while reducing LPIPS from 0.620 to 0.601. Fused features outperform both description and collaborative filtering features across all six metrics, but consistently underperform content features. Collaborative filtering features yield the weakest results, suggesting that they are less effective as the sole conditioning signal in this setting. These results identify content features as the most effective conditioning input

Table 8: Bundle recommendation results using the fine-tuned DDBC model under negative sampling ranking and all ranking. Predicted features are used directly to rank candidate items by comparison with their corresponding item features.

		Negative Sampling Ranking			All Ranking		
Setting	Ranking method	F1@1 ↑	Jacc@1 ↑	OAS ↑	F1@1 ↑	Jacc@1 ↑	OAS ↑
Fine-tuned DDBC	Original ranker	0.139	0.098	<u>0.471</u>	0.072	0.048	<u>0.475</u>
R2G-Map / Stage 1	`feature_t`	0.078	0.054	0.327	0.000	0.000	0.340
	`feature_c`	0.191	0.135	0.437	0.111	0.077	0.408
	`feature_cf`	0.142	0.098	0.386	0.073	0.049	0.373
	`feature_fused`	0.230	0.163	0.458	0.122	0.085	0.449
R2G-Map / Stage 2	`feature_t`	0.084	0.060	0.324	0.000	0.000	0.355
	`feature_c`	0.200	0.143	0.439	0.114	0.082	0.409
	`feature_cf`	0.165	0.116	0.395	0.083	0.057	0.383
	`feature_fused`	<u>0.245</u>	<u>0.176</u>	0.461	<u>0.128</u>	<u>0.093</u>	0.465
R2G-Map / Stage 2	Visual-token	**0.311**	**0.241**	**0.527**	**0.136**	**0.095**	**0.491**

among the evaluated alternatives and show that feature fusion does not improve upon content-only conditioning in this setting.

Ranking Performance of Predicted Features. To investigate the ranking capability of our feature predictor, we examine whether its predicted features improve bundle recommendation over the original fine-tuned TIGER ranker. We compare ranking based on individual and fused predicted features from Stage 1 and Stage 2 under negative sampling and all-item ranking. As shown in Table 7, Stage 1 `feature_fused` achieves 0.245 F1@1 and 0.176 Jacc@1 under negative sampling ranking, outperforming the original ranker by 20.1% and 17.3%, respectively. The gains also hold under all-item ranking, where F1@1 increases from 0.104 to 0.134 and Jacc@1 from 0.071 to 0.095. These improvements indicate that the predicted features capture useful signals for retrieving ground-truth items. Content features provide the strongest individual representation, while feature fusion outperforms every individual feature across both stages and ranking settings. Stage 2 further improves fused-feature ranking, reaching 0.256 F1@1 and 0.185 Jacc@1 under negative sampling ranking, and 0.135 F1@1 and 0.098 Jacc@1 under all-item ranking. However, the advantage in OAS depends on the ranking setting: fused features outperform the original ranker under negative sampling ranking but remain below it under all-item ranking. Finally, visual-embedding ranking achieves the best results across all three metrics under both settings, providing further improvements beyond ranking directly in the predicted feature space.

Bundle recommendation details. Tables 7 and 8 evaluate our framework with fine-tuned TIGER and DDBC under negative sampling and all-item ranking. Both backbones exhibit consistent trends: fused predicted features outperform individual features, Stage 2 further improves fused-feature ranking, and visual-embedding ranking achieves the best results across all three metrics. Compared with the original recommender, visual-embedding ranking improves F1@1 from 0.204 to 0.342 for TIGER and from 0.139 to 0.311 for DDBC under negative sampling ranking. Under all-item ranking, the corresponding scores increase from 0.104 to 0.140 and from 0.072 to 0.136, respectively. These consistent improvements support the applicability and generalizability of our framework across the two evaluated recommendation backbones and both ranking protocols.

Sequential recommendation details. Tables 9, 10, 11, and 12 report sequential recommendation results with DDBC and TIGER under negative sampling ranking and all ranking. Across both backbones and ranking protocols, fused predicted features outperform individual features, and Stage 2 further improves fused-feature ranking across all reported metrics. Compared with the original recommender, visual-embedding ranking increases R@1 from 0.170 to 0.338 for DDBC and from 0.234 to 0.351 for TIGER under candidate-set ranking. Under all ranking, the corresponding scores improve from 0.095 to 0.151 and from 0.134 to 0.179, respectively. Together with the bundle recommendation results, these findings support the applicability of our framework across both recommendation tasks, the two evaluated backbones, and different ranking protocols.

Table 9: Sequential recommendation results on DDBC under negative sampling ranking. Predicted features are used directly to rank candidate items by comparison with their corresponding item features.

Stage / Model	Ranking method	R@1 ↑	R@5 ↑	R@10 ↑	R@20 ↑	N@5 ↑	N@10 ↑	N@20 ↑	N@30 ↑	N@50 ↑
Fine-tuned DDBC	Original ranker	0.170	0.318	0.436	0.607	0.257	0.298	0.344	0.369	0.400
R2G-Map / Stage 1	feature_t	0.111	0.215	0.268	0.356	0.176	0.194	0.217	0.236	0.266
	feature_c	0.191	0.271	0.344	0.466	0.244	0.269	0.302	0.324	0.354
	feature_cf	0.149	0.247	0.342	0.474	0.209	0.241	0.276	0.294	0.316
	feature_fused	0.204	0.297	0.376	0.487	0.264	0.292	0.321	0.345	0.376
R2G-Map / Stage 2	feature_t	0.119	0.244	0.309	0.401	0.196	0.218	0.243	0.262	0.290
	feature_c	0.228	0.336	0.417	0.544	0.298	0.326	0.360	0.381	0.412
	feature_cf	0.186	0.285	0.377	0.494	0.250	0.281	0.313	0.330	0.352
	feature_fused	0.258	0.361	0.448	0.574	0.327	0.357	0.390	0.409	0.439
R2G-Map / Stage 2	Visual Emb.	**0.338**	**0.527**	**0.563**	**0.630**	**0.423**	**0.508**	**0.518**	**0.530**	**0.549**

Table 10: Sequential recommendation results on TIGER under negative sampling ranking. Predicted features are used directly to rank candidate items by comparison with their corresponding item features.

Stage / Model	Ranking method	R@1 ↑	R@5 ↑	R@10 ↑	R@20 ↑	N@5 ↑	N@10 ↑	N@20 ↑	N@30 ↑	N@50 ↑
Fine-tuned TIGER	Original ranker	0.234	0.347	0.442	0.605	0.305	0.338	0.382	0.413	0.444
R2G-Map / Stage 1	feature_t	0.134	0.241	0.285	0.373	0.203	0.218	0.241	0.259	0.288
	feature_c	0.222	0.298	0.364	0.477	0.274	0.296	0.326	0.348	0.378
	feature_cf	0.172	0.266	0.363	0.494	0.229	0.262	0.297	0.312	0.335
	feature_fused	0.240	0.315	0.392	0.501	0.289	0.315	0.344	0.366	0.396
R2G-Map / Stage 2	feature_t	0.139	0.253	0.313	0.400	0.212	0.232	0.255	0.274	0.301
	feature_c	0.252	0.352	0.431	0.552	0.317	0.345	0.377	0.398	0.426
	feature_cf	0.200	0.299	0.393	0.501	0.263	0.294	0.323	0.338	0.360
	feature_fused	0.269	0.377	0.458	0.567	0.340	0.368	0.397	0.417	0.443
R2G-Map / Stage 2	Visual-token	**0.351**	**0.537**	**0.588**	**0.648**	**0.438**	**0.510**	**0.528**	**0.536**	**0.555**

Table 11: Sequential recommendation results on DDBC under all ranking. Predicted features are used directly to rank candidate items by comparison with their corresponding item features.

Model / Stage	Ranking feature	R@1 ↑	R@5 ↑	R@10 ↑	R@20 ↑	N@5 ↑	N@10 ↑	N@20 ↑	N@30 ↑	N@50 ↑
Fine-tuned DDBC	Original ranker	0.095	0.101	0.104	0.112	0.104	0.105	0.107	0.108	0.110
R2G-Map / Stage 1	feature_t	0.000	0.000	0.001	0.051	0.000	0.000	0.014	0.014	0.014
	feature_c	0.125	0.137	0.143	0.149	0.139	0.141	0.142	0.143	0.144
	feature_cf	0.096	0.104	0.108	0.114	0.106	0.108	0.109	0.110	0.111
	feature_fused	0.127	0.154	0.160	0.169	0.150	0.152	0.154	0.155	0.156
R2G-Map / Stage 2	feature_t	0.000	0.000	0.001	0.058	0.000	0.000	0.015	0.016	0.016
	feature_c	0.143	0.161	0.168	0.175	0.161	0.163	0.165	0.166	0.168
	feature_cf	0.109	0.126	0.132	0.137	0.125	0.127	0.129	0.130	0.132
	feature_fused	0.149	0.179	0.187	0.196	0.176	0.179	0.181	0.182	0.184
R2G-Map / Stage 2	Visual-token	**0.151**	**0.215**	**0.241**	**0.266**	**0.194**	**0.203**	**0.210**	**0.214**	**0.218**

Table 12: Sequential recommendation results on TIGER under all ranking. Predicted features are used directly to rank candidate items by comparison with their corresponding item features.

Model / Stage	Ranking feature	R@1 ↑	R@5 ↑	R@10 ↑	R@20 ↑	N@5 ↑	N@10 ↑	N@20 ↑	N@30 ↑	N@50 ↑
Fine-tuned TIGER	Original ranker	0.134	0.139	0.147	0.155	0.168	0.170	0.172	0.174	0.177
R2G-Map / Stage 1	feature_t	0.000	0.001	0.003	0.055	0.001	0.001	0.015	0.015	0.016
	feature_c	0.164	0.172	0.175	0.182	0.176	0.178	0.179	0.180	0.182
	feature_cf	0.124	0.134	0.139	0.143	0.136	0.137	0.139	0.140	0.140
	feature_fused	0.168	0.181	0.187	0.192	0.184	0.186	0.187	0.188	0.189
R2G-Map / Stage 2	feature_t	0.000	0.001	0.002	0.055	0.000	0.001	0.015	0.015	0.015
	feature_c	0.168	0.188	0.195	0.202	0.188	0.191	0.193	0.194	0.196
	feature_cf	0.119	0.141	0.147	0.157	0.138	0.140	0.143	0.144	0.145
	feature_fused	0.170	0.206	0.212	0.217	0.204	0.206	0.207	0.209	0.211
R2G-Map / Stage 2	Visual-token	**0.179**	**0.228**	**0.250**	**0.279**	**0.212**	**0.220**	**0.227**	**0.231**	**0.236**