INTERNALIZING DENSE RECOMMENDATION KNOWLEDGE FOR GENERATIVE RECOMMENDATION VIA PROGRESSIVE TRANSFER

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative recommendation (GR) has emerged as a promising paradigm for end-to-end item retrieval, directly generating semantic identifiers (SIDs) of target items from user interaction histories. Recent studies have highlighted the value of dense recommendation (DR) knowledge for improving performance in GR. However, existing approaches primarily exploit DR knowledge for SID learning or auxiliary dense retrieval, leaving its potential for training the generative model itself underexplored. We propose **DEGET**, a progressive **De**nse-to-**Ge**nerative **T**ransfer framework that integrates DR knowledge of dense matching into GR's native encoder-decoder architecture. DEGET first jointly optimizes the semantic projector and encoder through full-catalog dense matching, enabling them to capture user preference knowledge from DR. The learned projector is then used to quantize items into SIDs. With the entire encoder and dense item representations frozen, which preserve user preference knowledge learned from the full-catalog, DEGET trains the decoder to express this dense knowledge through SID generation, thereby grounding the shared SID embeddings in DR knowledge. These SID embeddings are subsequently combined with the frozen item representations for SID-based history encoding, and joint encoder-decoder adaptation internalizes the transferred knowledge within GR. Extensive experiments on real-world datasets demonstrate substantial improvements over strong recommendation baselines. By preserving baseline SID tokenizers and applying DEGET's progressive transfer strategy, existing GR methods also achieve consistent performance gains. Our code is avaliable at https://anonymous.4open.science/r/DEGET.

1 INTRODUCTION

Sequential Dense Recommendation (DR) has long relied on dense retrieval (Zhao et al., 2024), where an encoder learns dense representations from interaction histories to capture user preferences and match against candidate items (Kang & McAuley, 2018; Qiu et al., 2022). Recently, generative recommendation (GR) has emerged as a promising alternative that reformulates recommendation as generative retrieval (Rajput et al., 2023; Li et al., 2025), autoregressively generating the next item's semantic identifier (SID) tokens from a compact vocabulary that does not grow with the catalog size. While promising, existing GR methods often remain less accurate than strong DR methods (Yang et al., 2025b), even when the two use the same encoder architecture (Yang et al., 2025a). However, recent analyzes further reveal a connection between the two paradigms: GR can be interpreted as a form of multi-vector DR (Wu et al., 2024), while exhibiting complementary strengths to DR in memorization and generalization (Ding et al., 2026). Taken together, these findings suggest that the performance gap may be related to how recommendation capability is learned within GR, raising a natural question: **can DR knowledge be effectively transferred to GR?**

Since DR essentially encodes recommendation knowledge in both the item representations fed into the encoder and the user preference representations it produces, we first identify three types of representations involved in GR's encoding and generation: semantic item representations quantized into SIDs, user preference representations encoded from interaction histories, and learnable SID embeddings used for history encoding and SID generation. However, some existing methods

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.

primarily focus on incorporating collaborative information into item representations and SIDs (Liang et al., 2026), while others preserve dense semantic representations or parallel dense retrieval pathways alongside generation (Yang et al., 2025b; Ding et al., 2026). Despite these advances, how to transfer DR knowledge across these representations, accounting for their distinct roles in SID construction, history encoding, and generation, remains underexplored. To investigate this question, we identify three interrelated representation gaps to address in dense-to-generative transfer, as illustrated in Figure 1. (1) **Semantic-Collaborative Representation Gap.** Adapting semantic item representations to collaborative preferences for SID construction does not by itself establish their alignment with user preference representations within the generative model itself. (2) **Dense-Generative Capability Gap.** The recommendation capability learned through dense matching is not explicitly transferred from user preference representations to the SID token space used for autoregressive generation. (3) **SID Grounding Gap.** The learnable SID token embeddings are initialized from scratch and are not explicitly grounded in the semantic item representations from which their corresponding SIDs are quantized.

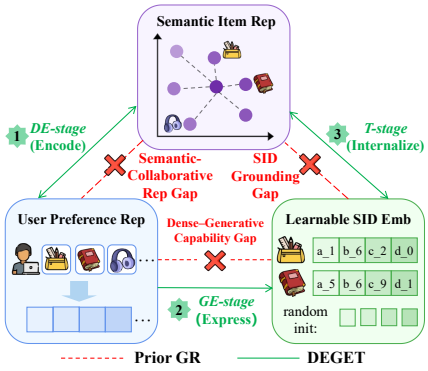


Figure 1: DEGET progressively bridges three representation gaps in prior GR.

Building on these observations, we move beyond improving isolated GR components or preserving dense signals through auxiliary pathways, and instead seek to unlock the intrinsic potential of GR by progressively transferring the recommendation knowledge learned through dense matching into its native encoder–decoder architecture. To bridge the optimization gap between dense matching and GR, we propose a progressive **De**nse-to-**Ge**nerative **T**ransfer framework for GR, dubbed **DEGET**. DEGET establishes the three missing representation-level connections through a progressive **Encode–Express–Internalize** process: (1) the **De**nse recommendation stage (**DE-stage**) connects semantic item representations with user preference representations, **encoding** DR knowledge into the representations underlying GR; (2) the **Ge**nerative recommendation stage (**GE-stage**) connects user preference representations with learnable SID embeddings, allowing the acquired DR knowledge to be **expressed** through autoregressive SID generation; and (3) the **T**ransfer stage (**T-stage**) further connects learnable SID embeddings with semantic-collaborativeitem representations, **internalizing** the transferred knowledge within the native encoder-decoder GR model.

Specifically, in the *DE-stage*, we jointly adapt semantic item embeddings and the GR encoder through full-catalog dense matching. A trainable semantic projector is optimized together with the bidirectional encoder, yielding a semantic-collaborative dense item space. We then freeze the projector to obtain dense item representations directly shaped by DR supervision, which serve as both the basis for subsequent quantization and item-level semantic anchors for SID grounding. In the *GE*-stage, we transfer the DR capability learned in the *DE*-stage to SID generation by freezing the encoder while retaining its dense item inputs, and optimizing only the decoder to express this knowledge through autoregressive SID prediction. Finally, in the *T*-stage, each historical item is expanded into its SID tokens, whose embeddings are combined with the corresponding frozen semantic–collaborative item representation to form SID-based encoder inputs. Joint encoder–decoder optimization then completes the transfer of DR knowledge into GR, thereby internalizing DR knowledge within a purely generative recommender. Our main contributions are as follows:

• We identify a representation-level disconnect between DR and GR: the recommendation knowledge learned through dense matching is not systematically transferred across three key representations involved in GR. We characterize this disconnect as three interrelated gaps, providing a unified view of dense-to-generative transfer in GR.

• We propose **DEGET**, a progressive dense-to-generative transfer framework that encodes, expresses, and internalizes DR knowledge within a purely generative recommender, without introducing an additional DR model.

• Extensive experiments on real-world datasets demonstrate consistent gains from transferring DR knowledge to GR. Further results show that DEGET complements existing SID tokenizers and transfers capability from matching to generation, rather than merely changing the prediction paradigm.

2 Related Work

Generative Recommendation. GR formulates sequential recommendation as autoregressive generation over discrete item identifiers using custom-designed models trained from scratch, typically encoder-decoder architectures, as pioneered by TIGER (Rajput et al., 2023). Subsequent work increasingly leverages DR knowledge for item tokenization by incorporating collaborative information into quantization (Wang et al., 2024a; Liang et al., 2026), introducing contextualized or personalized tokenization (Hou et al., 2025b; Zhong et al., 2025), and more tightly coupling tokenization with recommendation optimization (Liu et al., 2025; Fu et al., 2026). Recent studies further extend beyond tokenization to improve the generative process (Hou et al., 2026a). Despite these advances, DR knowledge is still mainly conveyed to the downstream encoder-decoder through SID construction, while the generative model learns recommendation behavior over SID token embeddings initialized from scratch. Accordingly, DEGET explores how dense recommendation knowledge can be carried beyond SID construction and utilized by the generative model itself.

Combining GR with DR. Recent studies combine GR with dense retrieval to exploit continuous representations alongside generation. LIGER (Yang et al., 2025a) complements GR with semantic dense retrieval over item-content representations, while COBRA (Yang et al., 2025b) cascades SIDs with semantic item representations for coarse-to-fine retrieval. More recently, OMEGA (Liu et al., 2026a) introduces lightweight dense retrieval over collaborative memories to augment existing GR backbones, while MemGen-GR (Ding et al., 2026) adaptively combines GR with a dense recommender based on their complementary memorization and generalization behaviors. These approaches retain a dense retrieval or scoring pathway alongside generation. DEGET takes a different route: rather than introducing a separate DR component at inference, it internalizes knowledge acquired through DR into the native GR encoder-decoder while preserving end-to-end inference.

3 Methodology: progressive dense-to-generative transfer

3.1 Preliminaries

Dense Recommendation. Let $\mathcal{I}$ denote the item catalog, $\mathcal{H} = (i_1, i_2, \ldots, i_T)$ an interaction history, and $y = i_{T+1}$ the next-item target. DR represents each item $i \in \mathcal{I}$ by a dense representation $\mathbf{v}_i$ and employs an encoder to obtain contextualized history representations $\mathbf{h}_{DR} = \mathrm{Encoder}(\mathbf{v}_1, \mathbf{v}_2 \ldots, \mathbf{v}_T)$. To optimize the model, each candidate item $j \in \mathcal{I}$ is first scored by $s(\mathbf{h}_{DR}, \mathbf{v}_j)$, where $s(\mathbf{a}, \mathbf{b}) = \mathbf{a}^\top \mathbf{b}$ denotes the dot-product similarity. The model is then optimized using the InfoNCE loss:

$$\mathcal{L}_{\mathrm{DR}} = -\log \frac{\exp\left(s(\mathbf{h}_{DR}, \mathbf{v}_+)/\tau\right)}{\sum_{j \in \mathcal{I}} \exp\left(s(\mathbf{h}_{DR}, \mathbf{v}_j)/\tau\right)}. \tag{1}$$

Here, $\mathbf{v}_+ = \mathbf{v}_y$ denotes the dense representation of the target item, while items in $\mathcal{I} \setminus \{y\}$ are treated as negative samples. The temperature $\tau > 0$ controls the sharpness of the score distribution.

Generative Recommendation. GR instead associates each item $i \in \mathcal{I}$ with a semantic identifier (SID) $C_i = (c_{i,1}, \ldots, c_{i,L})$, typically constructed by quantizing a semantic item representation with RQ-VAE (Lee et al., 2022). In this paper, we mainly focus on the encoder-decoder architecture for our design. Following TIGER (Rajput et al., 2023), a bidirectional encoder learns the flattened SID tokens of historical items into contextualized hidden states $\mathbf{H}$. Conditioned on $\mathbf{H}$, a decoder autoregressively generates the target SID C_y by minimizing the negative log-likelihood:

$$\mathcal{L}_{\mathrm{GR}} = -\sum_{\ell=1}^{L} \log p\left(c_{y,\ell} \mid c_{y,<\ell}, \mathbf{H}\right). \tag{2}$$

In GR, these SID token embeddings are initialized independently of the semantic item representations used for SID construction and learned through $\mathcal{L}_{\mathrm{GR}}$.

Connection between DR and GR. DR and encoder-decoder GR are connected through their prediction targets and history representations

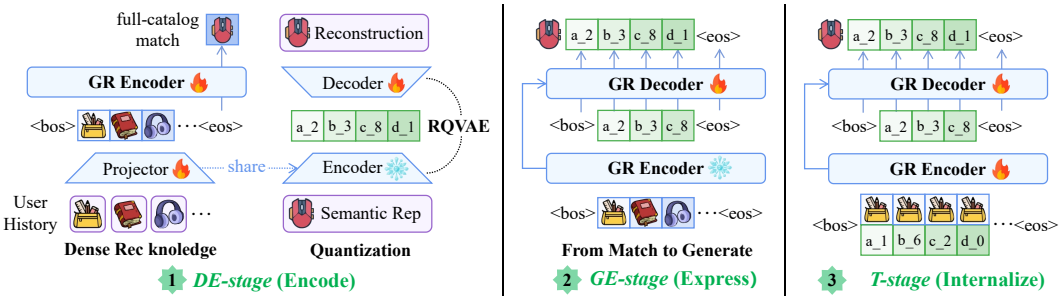


Figure 2: The overall framework of DEGET.

First, an item and its complete SID represent the same recommendation target. Assume a deterministic one-to-one mapping f from catalog items to complete SIDs of length L, with $C_y = f(y)$ (Rajput et al., 2023). Under this mapping, the event that the target item is y is identical to the event that its SID is C_y. Thus, given the same interaction history $\mathcal{H}$, their true conditional probabilities are equal. Let $p^*_{\mathcal{I}}$ denote the conditional target-item distribution and p^*_C its induced distribution over SIDs. The target correspondence and the probability chain rule give:

$$p^*_{\mathcal{I}}(y \mid \mathcal{H}) = p^*_C(C_y \mid \mathcal{H}) = \prod_{\ell=1}^{L} p^*_C(c_{y,\ell} \mid c_{y,<\ell}, \mathcal{H})\,. \tag{3}$$

DR estimates the target probability through dense item matching, whereas GR factorizes it into conditional SID token probabilities. This correspondence concerns the prediction target; it does not imply identical probabilities or rankings from separately trained DR and GR models.

Second, the same history encoder architecture can support both DR and GR (Yang et al., 2025a). Given a common history input representation, the encoder produces contextualized hidden states $\mathbf{H}$. A history vector $\mathbf{h}_{\mathrm{DR}}$ obtained from $\mathbf{H}$ supports dense item matching, while the full sequence of hidden states can condition SID generation:

$$\mathrm{DR} \underset{\text{Dense matching}}{\xleftarrow{\mathbf{h}_{\mathrm{DR}}}} \mathrm{Encoder}(\mathcal{H}) \underset{\text{SID generation}}{\xrightarrow{\mathbf{H}}} \mathrm{GR}. \tag{4}$$

These connections suggest that history representations learned through DR can also support SID generation through cross-attention. At each decoding step, queries derived from decoder states conditioned on preceding SID tokens attend to keys and values projected from $\mathbf{H}$. Retaining the dense history inputs and pretrained encoder therefore allows the decoder to learn SID prediction from DR representations, motivating progressive transfer of recommendation knowledge learned through dense matching into the native encoder-decoder architecture of GR.

3.2 Our Proposed Methods: DEGET

In this section, we present **DEGET**, which transfers DR knowledge into GR through a progressive *Encode-Express-Internalize* process. As illustrated in Figure 2, we first introduce the *DE-stage*, where the encoder learns user preference representations from dense item representations through full-catalog matching, jointly shaping the item representations with semantic and collaborative information before quantizing them into SIDs using an RQ-VAE (Lee et al., 2022) (Section 3.2). We then describe the *GE-stage*, which trains the decoder for autoregressive SID generation with the entire encoder frozen to ground SID embeddings in the semantic information captured through dense history modeling (Section 3.2). Finally, the *T-stage* combines SID token embeddings with the corresponding semantic–collaborative item representations for history encoding and jointly fine-tunes the encoder-decoder to internalize DR knowledge in GR (Section 3.2).

DE-stage (Encode). The *DE-stage* first *encodes* DR knowledge into semantic item representations and user preference representations, and then *encodes* the learned item representations into discrete semantic-collaborative SIDs through quantization. To this end, DEGET uses a semantic projector to map pretrained semantic item representations into recommendation-oriented item representations with a dual role: first providing item semantics for user history modeling, and subsequently

serving as continuous inputs for SID quantization. Let $\mathbf{s}_i$ denote the pretrained semantic representation of item i, obtained by encoding its textual attributes with a pretrained Sentence-T5 model, followed by principal component analysis (PCA) (Zheng et al., 2025). A trainable semantic projector maps it into a dense item representation $\mathbf{v}_i = \mathrm{Proj}(\mathbf{s}_i)$. Therefore, given the interaction history $\mathcal{H} = (i_1, i_2, \ldots, i_T)$, the bidirectional encoder $\mathrm{Encoder}$ produces contextualized hidden states $\mathbf{H} = \mathrm{Encoder}(\mathbf{v}_1, \mathbf{v}_2 \ldots, \mathbf{v}_T)$. We use the hidden state at the final history position, $\mathbf{h}_{\mathrm{DR}} = \mathbf{H}[T]$, as the user preference representation for dense item matching.

Token-level likelihood training does not explicitly compare the target item against all competing items in the catalog. Related studies on LLM-based GR identify limitations in negative modeling (Tan et al., 2025) and in aligning token-level optimization with item-level collaborative preferences (Lin et al., 2026). We therefore use full-catalog dense matching before decoder training to adapt semantic item representations to collaborative interaction patterns and train the encoder to discriminate user preferences among items. Specifically, we jointly optimize the Proj and $\mathrm{Encoder}$ using InfoNCE like Eq.1 as follows:

$$\mathcal{L}_{\mathrm{DE}} = -\log \frac{\exp\left(s(\mathbf{h}_{DR}, \mathbf{v}_+)/\tau + b_y\right)}{\sum_{j \in \mathcal{I}} \exp\left(s(\mathbf{h}_{DR}, \mathbf{v}_j)/\tau + b_j\right)}. \tag{5}$$

Here, τ is the temperature, and b_j is a learnable item-specific bias that accounts for item-dependent preference priors (Tang & Wang, 2018; Sun et al., 2019; Ma et al., 2023). Through this objective, the encoder captures user preferences from full-catalog information enriched with collaborative item signals, while the projector also learns collaborative representations through backpropagated DR supervision. After training, we freeze the learned projector and history encoder, and use them for SID construction and the subsequent GE stage.

We next exploit the projector's second role for SID construction. Prior methods incorporate collaborative representations or supervision from auxiliary recommenders into tokenization (Wei et al., 2025; Li et al., 2026). DEGET instead reuses the learned projector Proj as the frozen RQ-VAE encoder, so the residual codebooks directly quantize the semantic–collaborative item space learned through DR. During tokenizer training, the codebooks are updated while the RQ-VAE decoder reconstructs the input semantic representations $\mathbf{s}_i$. Residual quantization with K codebooks produces a code sequence, which is augmented through collision resolution to obtain the final SID $C_i = (c_{i,1}, \ldots, c_{i,L})$. Details of SID construction are provided in Appendix B.

GE-stage (Express). Prior studies on LLM-based GR show that randomly initialized SID embeddings lack a semantic connection to the pretrained language space (Feng et al., 2026). A generative backbone trained from scratch does not initially provide such a pretrained space for SID alignment. For recommendation, the one-to-one correspondence in Eq.3 allows the same target item to be represented by either its item identity or its complete SID. This motivates grounding SID embeddings in recommendation knowledge learned through DR, using the pretrained preference representations as conditioning. The *GE-stage* therefore trains the decoder and shared SID embeddings to *express* the knowledge acquired in the *DE-stage* through autoregressive SID generation.

Specifically, we freeze the projector and encoder to preserve the semantic–collaborative knowledge learned from DR in *DE-stage*. Given the item representations $\mathbf{v}_i$ produced by the frozen projector, the frozen encoder generates contextualized history representations $\mathbf{H} = \mathrm{Encoder}(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_T)$, which capture user preferences and condition SID generation token by token through cross-attention in the decoder. Therefore, given the target SID $C_y = (c_{y,1}, \ldots, c_{y,L})$, we keep the dense item representations and history encoder frozen and optimize only the decoder via teacher forcing under the generative objective as follows:

$$\mathcal{L}_{\mathrm{GE}} = -\sum_{\ell=1}^{L} \log p\left(c_{y,\ell} \mid c_{y,<\ell}, \mathbf{H}\right). \tag{6}$$

By frozen projector and encoder, the decoder can learn to express user preferences captured through information from dense matching as conditional probabilities over SID tokens.

DEGET employs a single learnable embedding table for SID tokens, shared between the encoder and decoder and tied to the output prediction head. Although the encoder does not consume SID tokens at this stage, the SID embeddings are still optimized through decoder-side token prediction,

enabling the decoder to learn to generate target SIDs conditioned on dense representations. The learned decoder and shared SID embeddings will be further used to initialize their counterparts in the *T-stage*. We further show that the one-to-one SID mapping preserves the target information encoded in these representations and that the GE objective is equivalent to learning the corresponding conditional distribution over SIDs, which are detailed in Appendix C.1.

T-stage (Internalize). The *T-stage* completes dense-to-generative transfer by incorporating the grounded SID embeddings into history encoding. The key challenge is the change in encoder input granularity. In the *GE-stage*, each historical item is represented by a single semantic-collaborative item representation $\mathbf{v}_i$, whereas SID-based history encoding expands each item into L SID tokens, turning a history of T items into TL token positions (Rajput et al., 2023). Recent studies identify SID input construction and preservation of item structure as important challenges in GR (Liu et al., 2026c; Wang et al., 2026; Zheng et al., 2026). Directly replacing $\mathbf{v}_i$ with SID token embeddings may therefore disrupt the history modeling learned by the encoder in the *DE-stage*.

To facilitate this transition, DEGET combines each SID token embedding with the corresponding frozen semantic–collaborative item representation. For a historical item i with SID $C_i = (c_{i,1}, \ldots, c_{i,L})$, the encoder input at each SID position is constructed as $\mathbf{x}_{i,\ell} = \mathrm{LayerNorm}\big(\mathbf{e}_{c_{i,\ell}} + \mathbf{v}_i\big)$ for $\ell = 1, \ldots, L$. The SID token embedding $\mathbf{e}_{c_{i,\ell}}$ is carried over from the *GE-stage* and further adapted during joint optimization, while $\mathbf{v}_i$ remains frozen. The SID embeddings encode token-level distinctions, while $\mathbf{v}_i$ provides all SID positions of the same item with a common item-level anchor. When a SID token is shared by multiple items, the added dense representation provides item-specific semantic-collaborative context beyond the shared token embedding. We further analyze the structure and item-specific information preserved by the additive composition before layer normalization in Appendix C.2.

The *T-stage* retains the history encoder trained in the *DE-stage* and preserved throughout the *GE-stage*, together with the decoder trained in the *GE-stage*. We unfreeze the history encoder and jointly fine-tune the encoder and decoder with a reduced learning rate to adapt the pretrained encoder to SID-based history modeling. The expanded history is encoded as $\mathbf{H} = \mathrm{Encoder}(\mathbf{x}_{i_1,1}, \ldots, \mathbf{x}_{i_1,L}, \ldots, \mathbf{x}_{i_T,1}, \ldots, \mathbf{x}_{i_T,L})$. We use the same autoregressive objective as in the *GE-stage* (Eq.6). Since SID embeddings are shared across the encoder, decoder, and output head, those learned in the *GE-stage* now receive supervision through both history encoding and SID generation. Together with the frozen item representations, joint optimization further grounds these shared embeddings in semantic–collaborative item information and adapts the encoder's acquired recommendation capability to SID-based histories, thereby *internalizing* DR knowledge within the generative encoder-decoder architecture.

4 Experiment

4.1 Experiment Setup

Datasets & Evaluation. We evaluate DEGET on Beauty, Sports, and Toys from Amazon Reviews 2014 (He & McAuley, 2016), and further validate it on three Amazon Reviews 2023 benchmarks (Hou et al., 2026b) in Appendix E.2. Following standard GR protocols (Rajput et al., 2023; Zhong et al., 2025), we use chronological leave-last-out evaluation with a maximum history length of 20 and report Recall@K (R@K) and NDCG@K (N@K) for $K \in \{5, 10\}$. Dataset statistics and full evaluation details are provided in Appendix D.1.

Baselines. For the Amazon Reviews 2014 benchmarks, we compare DEGET against four groups of recommendation methods: (1) **Dense Recommendation**: SASRec (Kang & McAuley, 2018), FDSA (Zhang et al., 2019), BERT4Rec (Sun et al., 2019), and S3-Rec (Zhou et al., 2020); (2) **Generative Recommendation**: TIGER (Rajput et al., 2023), LETTER (Wang et al., 2024a), EAGER (Wang et al., 2024b), UNGER (Xiao et al., 2025), ReSID (Liang et al., 2026), ETEGRec (Liu et al., 2025), DIGER (Fu et al., 2026), and ActionPiece (Hou et al., 2025b); (3) **Hybrid Recommendation**: LIGER (Yang et al., 2025a) and MemGen-GR (Ding et al., 2026). Detailed descriptions and implementation of these baselines, together with additional comparisons including LLM-based GR approaches, are provided in Appendix D.2.

Table 1: Performance comparison on the Amazon Reviews 2014 benchmarks, averaged over three random seeds. Percentage rows report relative improvements over the corresponding baseline. "Baseline + DEGET" applies DEGET's progressive transfer stages while retaining the original SID tokenizer of the baseline methods, such as "TIGER + DEGET". Standard deviations and results at $K = 20$ are provided in Appendix E.1.

Methods	Beauty				Sports				Toys			
	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10
Dense Recommendation												
SASRec	0.0411	0.0641	0.0260	0.0333	0.0224	0.0354	0.0146	0.0188	0.0497	0.0727	0.0333	0.0408
FDSA	0.0240	0.0397	0.0152	0.0202	0.0153	0.0255	0.0101	0.0133	0.0321	0.0479	0.0217	0.0268
BERT4Rec	0.0272	0.0429	0.0176	0.0226	0.0118	0.0202	0.0073	0.0101	0.0224	0.0360	0.0147	0.0191
S^3-Rec	0.0378	0.0634	0.0236	0.0319	0.0235	0.0379	0.0155	0.0201	0.0362	0.0584	0.0229	0.0300
LIGER (Dense)	0.0546	0.0861	0.0354	0.0455	0.0303	0.0485	0.0196	0.0254	0.0526	0.0876	0.0322	0.0434
Generative Recommendation												
TIGER	0.0435	0.0673	0.0291	0.0367	0.0283	0.0432	0.0183	0.0231	0.0422	0.0659	0.0279	0.0355
LETTER	0.0476	0.0723	0.0329	0.0409	0.0264	0.0409	0.0170	0.0216	0.0424	0.0641	0.0279	0.0348
EAGER	0.0448	0.0665	0.0304	0.0373	0.0240	0.0377	0.0154	0.0198	0.0462	0.0656	0.0323	0.0385
UNGER	0.0433	0.0677	0.0283	0.0362	0.0259	0.0408	0.0166	0.0214	0.0416	0.0665	0.0268	0.0348
ReSID	0.0475	0.0739	0.0321	0.0405	0.0273	0.0432	0.0178	0.0229	0.0407	0.0635	0.0267	0.0341
ETEGRec	0.0497	0.0759	0.0334	0.0419	0.0287	0.0455	0.0183	0.0237	0.0438	0.0682	0.0283	0.0362
DIGER	0.0460	0.0694	0.0302	0.0377	0.0263	0.0409	0.0169	0.0216	0.0418	0.0660	0.0271	0.0349
LIGER (GR)	0.0499	0.0775	0.0324	0.0413	0.0290	0.0457	0.0184	0.0238	0.0464	0.0751	0.0295	0.0387
ActionPiece	0.0507	0.0772	0.0339	0.0424	0.0304	0.0473	0.0200	0.0254	0.0462	0.0729	0.0303	0.0389
Hybrid Recommendation												
LIGER	0.0532	0.0830	0.0348	0.0444	0.0304	0.0480	0.0197	0.0253	0.0510	0.0838	0.0314	0.0420
MemGen-GR	0.0502	0.0748	0.0342	0.0421	0.0300	0.0462	0.0201	0.0253	0.0546	0.0796	**0.0383**	0.0464
Ours (GR)												
DEGET	**0.0577**	**0.0907**	**0.0377**	**0.0483**	**0.0344**	**0.0528**	**0.0222**	**0.0281**	**0.0601**	**0.0937**	0.0378	**0.0486**
vs. LIGER	↑ 8.46%	↑ 9.28%	↑ 8.33%	↑ 8.78%	↑ 13.16%	↑ 10.00%	↑ 12.69%	↑ 11.07%	↑ 17.84%	↑ 11.81%	↑ 20.38%	↑ 15.71%
TIGER + **DEGET**	0.0524	0.0824	0.0340	0.0436	0.0323	0.0506	0.0206	0.0265	0.0545	0.0869	0.0355	0.0460
vs. TIGER	↑ 20.46%	↑ 22.44%	↑ 16.84%	↑ 18.80%	↑ 14.13%	↑ 17.13%	↑ 12.57%	↑ 14.72%	↑ 29.15%	↑ 31.87%	↑ 27.24%	↑ 29.58%
ActionPiece + **DEGET**	0.0549	0.0874	0.0363	0.0468	0.0345	0.0532	0.0227	0.0287	0.0556	0.0867	0.0351	0.0451
vs. ActionPiece	↑ 8.28%	↑ 13.21%	↑ 7.08%	↑ 10.38%	↑ 13.49%	↑ 12.47%	↑ 13.50%	↑ 12.99%	↑ 20.35%	↑ 18.93%	↑ 15.84%	↑ 15.94%

Implementation Details. We run each method on the Amazon Reviews 2014 benchmarks with three random seeds, i.e., $\{2024, 2025, 2026\}$. Method-specific configurations and further implementation details are provided in Appendix D.3.

4.2 Main Results

As shown in Table 1, DEGET achieves better overall performance than dense, hybrid, and generative recommendation baselines across the three datasets. Notably, LIGER (Dense) generally outperforms existing GR baselines, whereas DEGET consistently surpasses this dense baseline, highlighting the value of DR pretraining for improving generative recommendation. DEGET also outperforms LIGER and MemGen-GR overall, with an average relative NDCG@10 improvement of 11.86% over LIGER. These results show that progressive transfer enables GR to exploit DR knowledge internally, outperforming hybrid methods that use semantic item representations with dense reranking or directly ensemble DR and GR predictions. Its gains over GR methods that incorporate collaborative signals into SID learning further highlight the benefit of extending DR knowledge beyond tokenization to user preference modeling and SID generation. Moreover, applying DEGET to TIGER and ActionPiece consistently improves both methods while retaining their original tokenizers, with average relative NDCG@10 gains of 21.03% and 13.10%, respectively. These improvements demonstrate that progressive transfer complements different tokenization strategies.

4.3 Ablation Study

Effectiveness of Progressive Transfer. We assess the contribution of each transfer stage in Table 2. Full DEGET outperforms all generative ablations and achieves performance comparable to DE. Without T, *DE+GE* remains substantially below DE, while skipping GE (*DE+T*) consistently reduces performance relative to full DEGET, highlighting the complementary roles of GE and T in progressively connecting semantic–collaborative item representations, user preference representations, and SID token embeddings. Notably, *DE+T* consistently outperforms *T*, which adds projected semantic representations to SID inputs without dense pretraining, showing that learning DR knowledge provides benefits beyond incorporating item semantics into history encoding.

DR Knowledge in SID Construction and History Encoding. Table 3 evaluates the contributions of DR knowledge to SID construction and history encoding. Combining DEGET-SID with DEGET-Rep achieves the best performance across all datasets and metrics, demonstrating the complementary

Table 2: Ablation study of progressive dense-to-generative transfer on the Amazon Reviews 2014 benchmarks, averaged over three random seeds. DE provides the pretrained DR foundation.

Variants	Beauty				Sports				Toys			
	N@5	N@10	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10	R@5	R@10
DE	0.0359	0.0469	0.0561	0.0905	0.0224	0.0286	0.0342	0.0537	0.0374	0.0486	0.0602	0.0950
DE + GE	0.0283	0.0374	0.0445	0.0728	0.0161	0.0209	0.0253	0.0404	0.0278	0.0363	0.0455	0.0720
DE + T	0.0363	0.0462	0.0557	0.0865	0.0214	0.0274	0.0326	0.0512	0.0345	0.0442	0.0543	0.0846
T	0.0312	0.0387	0.0459	0.0693	0.0187	0.0237	0.0280	0.0437	0.0301	0.0376	0.0451	0.0684
DE + GE + T	**0.0377**	**0.0483**	**0.0577**	**0.0907**	**0.0222**	**0.0281**	**0.0344**	**0.0528**	**0.0378**	**0.0486**	**0.0601**	**0.0937**

Table 3: Effects of SID construction and item representations in DEGET on the 2014 benchmarks, averaged over three random seeds. We compare the SID construction between TIGER and our proposed DE-stage. *Item Rep.* denotes the item representations added to the SID embeddings for prediction, including the original semantic representations (*Semantic*), our proposed semantic–collaborative representations (*DEGET-Rep*), or no additional representations ("–").

Input		Beauty				Sports				Toys			
SID	Item Rep.	N@5	N@10	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10	R@5	R@10
TIGER-SID	–	0.0282	0.0360	0.0427	0.0667	0.0161	0.0212	0.0255	0.0415	0.0272	0.0351	0.0422	0.0669
TIGER-SID	Semantic	0.0314	0.0405	0.0483	0.0768	0.0187	0.0243	0.0290	0.0463	0.0306	0.0397	0.0477	0.0759
TIGER-SID	DEGET-Rep	0.0340	0.0436	0.0524	0.0824	0.0206	0.0266	0.0319	0.0507	0.0355	0.0460	0.0545	0.0869
DEGET-SID	–	0.0343	0.0442	0.0528	0.0839	0.0210	0.0269	0.0325	0.0507	0.0334	0.0431	0.0537	0.0837
DEGET-SID	Semantic	0.0323	0.0416	0.0494	0.0786	0.0202	0.0259	0.0307	0.0487	0.0323	0.0415	0.0512	0.0798
DEGET-SID	**DEGET-Rep**	**0.0377**	**0.0483**	**0.0577**	**0.0907**	**0.0222**	**0.0281**	**0.0344**	**0.0528**	**0.0378**	**0.0486**	**0.0601**	**0.0937**

benefits of transferring DR knowledge through SIDs and dense item anchors. DEGET-SID consistently outperforms TIGER-SID for each choice of item representations, while adding DEGET-Rep improves performance with SIDs held fixed. These results show that DR knowledge benefits both SID construction and history encoding. In contrast, adding the original semantic representations reduces performance with DEGET-SID, highlighting the importance of adapting semantic item representations through DR supervision before using them as item-level anchors in history encoding.

4.4 Understanding Progressive Dense-to-Generative Transfer

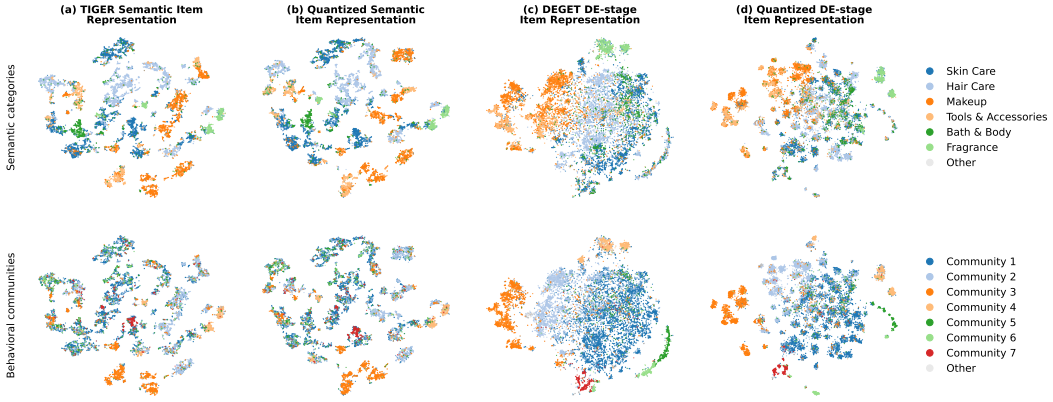


Figure 3: t-SNE visualization of item representations before and after residual quantization on Beauty. (a)–(b) TIGER's RQ-VAE encoder outputs and their quantized counterparts; (c)–(d) DEGET's DE-stage item representations and their quantized counterparts. The same items are colored by metadata categories (top) and behavioral communities (bottom).

Semantic and Collaborative Structure in SID Construction. To examine whether DR knowledge is transferred into SIDs, we visualize item representations before and after residual quantization. Figure 3 suggests that DEGET's SIDs capture collaborative item relationships alongside item semantics. Both methods use the same RQ-VAE architecture and semantic reconstruction target, while DEGET quantizes item representations previously optimized through DR supervision.

Specifically, the bottom row colors items by behavioral communities identified through Louvain community detection on a weighted item co-occurrence graph constructed from training histories. Compared with TIGER, DEGET exhibits more coherent grouping under both behavioral and semantic coloring before and after quantization, with clearer behavioral communities and recognizable category structure. These observations suggest that collaborative signals and item semantics coexist in the learned item representations. Since each quantized representation is the sum of the RQ codewords selected by an item's SID before decoding, these results provide qualitative evidence that collaborative relationships learned through dense matching are incorporated into SID construction.

SID Grounding across Transfer Stages. To examine SID grounding through DR knowledge, we visualize shared SID embeddings and history inputs alongside DE-stage item representations in Figure 4. In GE, SID embeddings learned through generation conditioned on history representations from the frozen DE-stage encoder are concentrated within the projected item distribution. Individual tokens are shared across items, so the vocabulary need not reproduce the full distribution of item representations. From GE to T, the projected vocabulary distribution remains broadly similar, while SID + Item Rep inputs extend across a broader region overlapping the item distribution. For items sharing a token, its embedding is combined with their respective $\mathbf{v}_i$, introducing item-specific semantic–collaborative context into history encoding. This explains the broader input distribution despite the continued concentration of the shared vocabulary. These patterns suggest that the SID embeddings learned in GE provide a consistent representational basis for T, while DR item representations contribute complementary item-level distinctions.

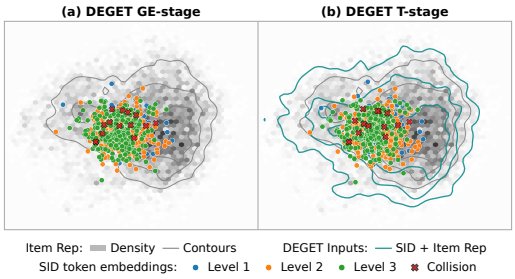


Figure 4: PCA of SID embeddings and history inputs across the GE and T stages on Beauty. All vectors are L2-normalized and projected using the same PCA fitted to DE-stage item representations.

Transfer of DR Capability to GR. To assess how effectively DR capability transfers to GR, we compare GE and T with DE in ranking and retrieval, using beam size 20 for SID generation. Figure 5(a) shows that, averaged across the three datasets, T achieves a relative NDCG@20 of approximately 97.6%, compared with 76.0% for GE. Figure 5(b) shows that top-20 overlap with DE increases from 37.5% in GE to 48.3% in T. These results demonstrate that GE expresses the DR knowledge learned in DE and highlight the role of the T-stage in narrowing the performance gap between dense matching and SID generation, hilighting that DR knowledge is progressively internalized during transfer. Notably, T achieves ranking performance close to DE while sharing only about half of its top-20 items. These results further indicate that DEGET internalizes the full-catalog discriminative capability of DR within autoregressive generation, enabling a substantive transition from dense matching to generation. Addtional analyses are provided in Appendix F.

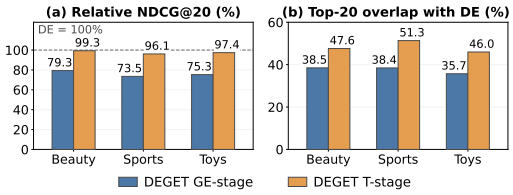


Figure 5: Ranking and retrieval: DR vs. GR. (a) Relative NDCG@20: NDCG@20 of GE and T stage divided by that of DE. (b) Top-20 overlap with DE: rate of retrieved items shared.

5 Conclusion

We proposed **DEGET**, a progressive dense-to-generative transfer framework for GR. Through an Encode-Express-Internalize process, DEGET transfers DR knowledge into GR's native encoder-decoder architecture while preserving purely generative inference. Experiments on real-world datasets demonstrate its effectiveness. Our analyses further show that DR knowledge remains usable within the adapted encoder and is expressed in generative predictions, supporting DR as a foundation for learning stronger generative recommenders.

AI USE STATEMENT

In this work, we used generative AI tools solely to assist with writing and polishing the manuscript. Although ChatGPT assisted with proofreading and language refinement, the authors independently developed the research ideas and conceptual design, conducted all data analyses, and made all final decisions. We have reviewed all AI-assisted content and take responsibility for the final content of this work.

ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. Our research does not involve human subjects or sensitive personal data. Although we use some user interaction data, it is anonymized, publicly available, and does not contain any personally identifiable information. Moreover, we strictly follow the usage guidelines set by the dataset providers.

REFERENCES

Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *CoRR*, abs/2502.18965, 2025. doi: 10.48550/ARXIV.2502.18965. URL https://doi.org/10.48550/arXiv.2502.18965.

Yijie Ding, Zitian Guo, Jiacheng Li, Letian Peng, Shuai Shao, Wei Shao, Xiaoqiang Luo, Luke Simon, Jingbo Shang, Julian McAuley, and Yupeng Hou. On the memorization and generalization of generative recommendation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '26, pp. 921–932, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400722592. doi: 10.1145/3770855.3818148. URL https://doi.org/10.1145/3770855.3818148.

Jiawei Feng, Xiaoyu Kong, Leheng Sheng, Bin Wu, Chao Yi, Feifang Yang, Xiang-Rong Sheng, Han Zhu, Xiang Wang, Jiancan Wu, and Xiangnan He. Fine-grained semantics integration for large language model-based recommendation. *CoRR*, abs/2602.22632, 2026. doi: 10.48550/ARXIV.2602.22632. URL https://doi.org/10.48550/arXiv.2602.22632.

Junchen Fu, Xuri Ge, Alexandros Karatzoglou, Ioannis Arapakis, Suzan Verberne, Joemon M. Jose, and Zhaochun Ren. Differentiable semantic id for generative recommendation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '26, pp. 369–379, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400725999. doi: 10.1145/3805712.3809641. URL https://doi.org/10.1145/3805712.3809641.

Ruining He and Julian J. McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In Jacqueline Bourdeau, Jim Hendler, Roger Nkambou, Ian Horrocks, and Ben Y. Zhao (eds.), *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pp. 507–517. ACM, 2016. doi: 10.1145/2872427.2883037. URL https://doi.org/10.1145/2872427.2883037.

Ruining He, Lukasz Heldt, Lichan Hong, Raghunandan H. Keshavan, Shifan Mao, Nikhil Mehta, Zhengyang Su, Alicia Tsai, Yueqi Wang, Shao-Chuan Wang, Xinyang Yi, Lexi Baugher, Baykal Cakici, Ed H. Chi, Cristos Goodrow, Ningren Han, He Ma, Rómer Rosales, Abby Van Soest, Devansh Tandon, Su-Lin Wu, Weilong Yang, and Yilin Zheng. PLUM: adapting pre-trained language models for industrial-scale generative recommendations. In Hakim Hacid, Yoelle Maarek, Francesco Bonchi, Ido Guy, and Emine Yilmaz (eds.), *Proceedings of the ACM Web Conference 2026, WWW 2026, Dubai, United Arab Emirates, originally scheduled for April 13-17, 2026, rescheduled for June 29 - July 3, 2026*, pp. 8093–8104. ACM, 2026a. doi: 10.1145/3774904.3792802. URL https://doi.org/10.1145/3774904.3792802.

Yinhan He, Liam Collins, Bhuvesh Kumar, Jundong Li, Neil Shah, and Donald Loveland. Implicit reasoning for large language model-based generative recommendation. *CoRR*, abs/2606.14142, 2026b. doi: 10.48550/ARXIV.2606.14142. URL https://doi.org/10.48550/arXiv.2606.14142.

Minjie Hong, Yan Xia, Zehan Wang, Jieming Zhu, Ye Wang, Sihang Cai, Xiaoda Yang, Quanyu Dai, Zhenhua Dong, Zhimeng Zhang, and Zhou Zhao. EAGER-LLM: enhancing large language models as recommenders through exogenous behavior-semantic integration. In Guodong Long, Michale Blumestein, Yi Chang, Liane Lewin-Eytan, Zi Helen Huang, and Elad Yom-Tov (eds.), *Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025- 2 May 2025*, pp. 2754–2762. ACM, 2025. doi: 10.1145/3696410.3714933. URL https://doi.org/10.1145/3696410.3714933.

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 Luiza Antonie, Jian Pei, Xiaohui Yu, Flavio Chierichetti, Hady W. Lauw, Yizhou Sun, and Srinivasan Parthasarathy (eds.), *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, V.2, KDD 2025, Toronto ON, Canada, August 3-7, 2025*, pp. 956–966. ACM, 2025a. doi: 10.1145/3711896.3736979. URL https://doi.org/10.1145/3711896.3736979.

Yupeng Hou, Jianmo Ni, Zhankui He, Noveen Sachdeva, Wang-Cheng Kang, Ed H. Chi, Julian McAuley, and Derek Zhiyuan Cheng. Actionpiece: contextually tokenizing action sequences for generative recommendation. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML'25. JMLR.org, 2025b.

Yupeng Hou, Haven Kim, Clark Mingxuan Ju, Eduardo Escoto, Neil Shah, and Julian J. McAuley. Expressiveness limits of autoregressive semantic ID generation in generative recommendation. *CoRR*, abs/2605.06331, 2026a. doi: 10.48550/ARXIV.2605.06331. URL https://doi.org/10.48550/arXiv.2605.06331.

Yupeng Hou, Jiacheng Li, Xiangjun Fu, Zhankui He, An Yan, Xiusi Chen, and Julian J. McAuley. Bridging language and items for retrieval and recommendation: Benchmarking llms as semantic encoders. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 3251–3265. Association for Computational Linguistics, 2026b. doi: 10.18653/V1/2026.ACL-LONG.147. URL https://doi.org/10.18653/v1/2026.acl-long.147.

Wang-Cheng Kang and Julian J. McAuley. Self-attentive sequential recommendation. In *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*, pp. 197–206. IEEE Computer Society, 2018. doi: 10.1109/ICDM.2018.00035. URL https://doi.org/10.1109/ICDM.2018.00035.

Sunwoo Kim, Sunkyung Lee, Clark Mingxuan Ju, Donald Loveland, Bhuvesh Kumar, Kijung Shin, Neil Shah, and Liam Collins. On the memorization behavior of llms in generative recommendation: Observations, implications, and training strategies. *CoRR*, abs/2606.17276, 2026. doi: 10.48550/ARXIV.2606.17276. URL https://doi.org/10.48550/arXiv.2606.17276.

Xiaoyu Kong, Leheng Sheng, Junfei Tan, Yuxin Chen, Jiancan Wu, An Zhang, Xiang Wang, and Xiangnan He. Minionerec: An open-source framework for scaling generative recommendation. *CoRR*, abs/2510.24431, 2025. doi: 10.48550/ARXIV.2510.24431. URL https://doi.org/10.48550/arXiv.2510.24431.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 11513–11522. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01123. URL https://doi.org/10.1109/CVPR52688.2022.01123.

Jialei Li, Yang Zhang, Yimeng Bai, Shuai Zhu, Ziqi Xue, Xiaoyan Zhao, Dingxian Wang, Frank Yang, Andrew Rabinovich, and Xiangnan He. Unigrec: Unified generative recommendation with soft identifiers for end-to-end optimization. *CoRR*, abs/2601.17438, 2026. doi: 10.48550/ARXIV.2601.17438. URL https://doi.org/10.48550/arXiv.2601.17438.

Xiaoxi Li, Jiajie Jin, Yujia Zhou, Yuyao Zhang, Peitian Zhang, Yutao Zhu, and Zhicheng Dou. From matching to generation: A survey on generative information retrieval. *ACM Trans. Inf.*

Syst., 43(3):83:1–83:62, 2025. doi: 10.1145/3722552. URL https://doi.org/10.1145/3722552.

Yu Liang, Zhongjin Zhang, Yuxuan Zhu, Kerui Zhang, Zhiluohan Guo, Wenhang Zhou, Zonqi Yang, Kangle Wu, Yabo Ni, Anxiang Zeng, Cong Fu, Jianxin Wang, and Jiazhi Xia. Rethinking generative recommender tokenizer: Recsys-native encoding and semantic quantization beyond llms. *CoRR*, abs/2602.02338, 2026. doi: 10.48550/ARXIV.2602.02338. URL https://doi.org/10.48550/arXiv.2602.02338.

Fake Lin, Binbin Hu, Zhi Zheng, Xi Zhu, Ziqi Liu, Zhiqiang Zhang, Jun Zhou, and Tong Xu. Token-level collaborative alignment for llm-based generative recommendation. In Hakim Hacid, Yoelle Maarek, Francesco Bonchi, Ido Guy, and Emine Yilmaz (eds.), *Proceedings of the ACM Web Conference 2026, WWW 2026, Dubai, United Arab Emirates, originally scheduled for April 13-17, 2026, rescheduled for June 29 - July 3, 2026*, pp. 6909–6919. ACM, 2026. doi: 10.1145/3774904.3792727. URL https://doi.org/10.1145/3774904.3792727.

Enze Liu, Bowen Zheng, Cheng Ling, Lantao Hu, Han Li, and Wayne Xin Zhao. Generative recommender with end-to-end learnable item tokenization. In Nicola Ferro, Maria Maistro, Gabriella Pasi, Omar Alonso, Andrew Trotman, and Suzan Verberne (eds.), *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2025, Padua, Italy, July 13-18, 2025*, pp. 729–739. ACM, 2025. doi: 10.1145/3726302.3729989. URL https://doi.org/10.1145/3726302.3729989.

Enze Liu, Zhen Tian, and Wayne Xin Zhao. Collaborative memory augmentation for generative recommendation. In Won Kim, Jae-Gil Lee, Kyuseok Shim, James Caverlee, David F. Gleich, Xiangliang Zhang, Qi He, Faisal Farooq, Rómer Rosales, Sayan Ranu, Haixun Wang, Yan Liu, Carl Yang, Leman Akoglu, Nitesh V. Chawla, and Vagelis Papalexakis (eds.), *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2, KDD 2026, Jeju Island, Korea, August 9-13, 2026*, pp. 3164–3175. ACM, 2026a. doi: 10.1145/3770855.3818179. URL https://doi.org/10.1145/3770855.3818179.

Zhanyu Liu, Shiyao Wang, Xingmei Wang, Rongzhou Zhang, Jiaxin Deng, Honghui Bao, Jinghao Zhang, Wuchao Li, Penggei Zheng, Xiangyu Wu, Yifei Hu, Qigen Hu, Xinchen Luo, Lejian Ren, Zixing Zhang, Qianqian Wang, Kuo Cai, Yunfan Wu, Hongtao Cheng, Zexuan Cheng, Lu Ren, Huanjie Wang, Yi Su, Ruiming Tang, Kun Gai, and Guorui Zhou. Onerec-think: In-text reasoning for generative recommendation. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 2664–2681. Association for Computational Linguistics, 2026b. doi: 10.18653/V1/2026.ACL-LONG.123. URL https://doi.org/10.18653/v1/2026.acl-long.123.

Ziwei Liu, Yejing Wang, Shengyu Zhou, Xinhang Li, and Xiangyu Zhao. Conditional memory enhanced item representation for generative recommendation. *CoRR*, abs/2605.11447, 2026c. doi: 10.48550/ARXIV.2605.11447. URL https://doi.org/10.48550/arXiv.2605.11447.

Muyang Ma, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Huasheng Liang, Jun Ma, and Maarten De Rijke. Improving transformer-based sequential recommenders through preference editing. *ACM Trans. Inf. Syst.*, 41(3), April 2023. ISSN 1046-8188. doi: 10.1145/3564282. URL https://doi.org/10.1145/3564282.

Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Cer, and Yinfei Yang. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, volume ACL 2022 of *Findings of ACL*, pp. 1864–1874. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.FINDINGS-ACL.146. URL https://doi.org/10.18653/v1/2022.findings-acl.146.

Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. Contrastive learning for representation degeneration problem in sequential recommendation. In K. Selcuk Candan, Huan Liu, Leman Akoglu, Xin Luna Dong, and Jiliang Tang (eds.), *WSDM '22: The Fifteenth ACM International*

Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022, pp. 813–823. ACM, 2022. doi: 10.1145/3488560.3498433. URL https://doi.org/10.1145/3488560.3498433.

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 Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/20dcab0f14046a5c6b02b61da9f13229-Abstract-Conference.html.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In Wenwu Zhu, Dacheng Tao, Xueqi Cheng, Peng Cui, Elke A. Rundensteiner, David Carmel, Qi He, and Jeffrey Xu Yu (eds.), *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pp. 1441–1450. ACM, 2019. doi: 10.1145/3357384.3357895. URL https://doi.org/10.1145/3357384.3357895.

Junfei Tan, Yuxin Chen, An Zhang, Junguang Jiang, Bin Liu, Ziru Xu, Zhu Han, Jian Xu, Bo Zheng, and Xiang Wang. Reinforced preference optimization for recommendation. *CoRR*, abs/2510.12211, 2025. doi: 10.48550/ARXIV.2510.12211. URL https://doi.org/10.48550/arXiv.2510.12211.

Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In Yi Chang, Chengxiang Zhai, Yan Liu, and Yoelle Maarek (eds.), *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, pp. 565–573. ACM, 2018. doi: 10.1145/3159652.3159656. URL https://doi.org/10.1145/3159652.3159656.

OneRec Team, Biao Yang, Boyang Ding, Chenglong Chu, Dunju Zang, Fei Pan, Han Li, Hao Jiang, Honghui Bao, Huanjie Wang, et al. Onereason technical report. *arXiv preprint arXiv:2606.06260*, 2026.

Dongsheng Wang, Yuxi Huang, Shen Gao, Yifan Wang, Chengrui Huang, and Shuo Shang. Generative next POI recommendation with semantic ID. In Luiza Antonie, Jian Pei, Xiaohui Yu, Flavio Chierichetti, Hady W. Lauw, Yizhou Sun, and Srinivasan Parthasarathy (eds.), *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, V.2, KDD 2025, Toronto ON, Canada, August 3-7, 2025*, pp. 2904–2914. ACM, 2025. doi: 10.1145/3711896.3736981. URL https://doi.org/10.1145/3711896.3736981.

Junting Wang, Xinrui He, Yunzhe Li, and Hari Sundaram. Understanding semantic ids: From item representation to item selection in generative recommendation. *CoRR*, abs/2607.24995, 2026. doi: 10.48550/ARXIV.2607.24995. URL https://doi.org/10.48550/arXiv.2607.24995.

Wenjie Wang, Honghui Bao, Xinyu Lin, Jizhi Zhang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. Learnable item tokenization for generative recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, CIKM '24, pp. 2400–2409, New York, NY, USA, 2024a. Association for Computing Machinery. ISBN 9798400704369. doi: 10.1145/3627673.3679569. URL https://doi.org/10.1145/3627673.3679569.

Ye Wang, Jiahao Xun, Minjie Hong, Jieming Zhu, Tao Jin, Wang Lin, Haoyuan Li, Linjun Li, Yan Xia, Zhou Zhao, and Zhenhua Dong. EAGER: two-stream generative recommender with behavior-semantic collaboration. In Ricardo Baeza-Yates and Francesco Bonchi (eds.), *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pp. 3245–3254. ACM, 2024b. doi: 10.1145/3637528.3671775. URL https://doi.org/10.1145/3637528.3671775.

Kaiwen Wei, Kejun He, Xiaomian Kang, Jie Zhang, Ymyang, Li Jin, Zhenyang Li, Jiang Zhong, Richard He Bai, and Junnan Zhu. From past to path: Masked history learning for next-item prediction in generative recommendation. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 10421–10441. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.ACL-LONG.475. URL https://doi.org/10.18653/v1/2026.acl-long.475.

Tianxin Wei, Xuying Ning, Xuxing Chen, Ruizhong Qiu, Yupeng Hou, Yan Xie, Shuang Yang, Zhigang Hua, and Jingrui He. Cofirec: Coarse-to-fine tokenization for generative recommendation. *CoRR*, abs/2511.22707, 2025. doi: 10.48550/ARXIV.2511.22707. URL https://doi.org/10.48550/arXiv.2511.22707.

Shiguang Wu, Wenda Wei, Mengqi Zhang, Zhumin Chen, Jun Ma, Zhaochun Ren, Maarten de Rijke, and Pengjie Ren. Generative retrieval as multi-vector dense retrieval. In Grace Hui Yang, Hongning Wang, Sam Han, Claudia Hauff, Guido Zuccon, and Yi Zhang (eds.), *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, pp. 1828–1838. ACM, 2024. doi: 10.1145/3626772.3657697. URL https://doi.org/10.1145/3626772.3657697.

Longtao Xiao, Haozhao Wang, Cheng Wang, Linfei Ji, Yifan Wang, Jieming Zhu, Zhenhua Dong, Rui Zhang, and Ruixuan Li. Unger: Generative recommendation with a unified code via semantic and collaborative integration. *ACM Trans. Inf. Syst.*, 44(2), December 2025. ISSN 1046-8188. doi: 10.1145/3773771. URL https://doi.org/10.1145/3773771.

Liu Yang, Fabian Paischer, Kaveh Hassani, Jiacheng Li, Shuai Shao, Zhang Gabriel Li, Yun He, Xue Feng, Nima Noorshams, Sem Park, Bo Long, Robert D. Nowak, Xiaoli Gao, and Hamid Eghbalzadeh. Unifying generative and dense retrieval for sequential recommendation. *Trans. Mach. Learn. Res.*, 2025, 2025a. URL https://openreview.net/forum?id=jxdnFIsjCb.

Yuhao Yang, Zhi Ji, Zhaopeng Li, Yi Li, Zhonglin Mo, Yue Ding, Kai Chen, Zijian Zhang, Jie Li, Shuanglong Li, and Liu Lin. Sparse meets dense: Unified generative recommendations with cascaded sparse-dense representations. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025b. URL http://papers.nips.cc/paper_files/paper/2025/hash/86ba836d4c5dd859d795a172911745e2-Abstract-Conference.html.

Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, and Xiaofang Zhou. Feature-level deeper self-attention network for sequential recommendation. In Sarit Kraus (ed.), *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pp. 4320–4326. ijcai.org, 2019. doi: 10.24963/IJCAI.2019/600. URL https://doi.org/10.24963/ijcai.2019/600.

Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong (eds.), *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pp. 4653–4664. ACM, 2021. doi: 10.1145/3459637.3482016. URL https://doi.org/10.1145/3459637.3482016.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. Dense text retrieval based on pretrained language models: A survey. *ACM Trans. Inf. Syst.*, 42(4):89:1–89:60, 2024. doi: 10.1145/3637870. URL https://doi.org/10.1145/3637870.

Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. Adapting large language models by integrating collaborative semantics for recommendation. In *40th IEEE International Conference on Data Engineering, ICDE 2024, Utrecht, The Netherlands, May 13-16, 2024*, pp. 1435–1448. IEEE, 2024. doi: 10.1109/ICDE60146.2024.00118. URL https://doi.org/10.1109/ICDE60146.2024.00118.

Bowen Zheng, Enze Liu, Zhongfu Chen, Zhongrui Ma, Yue Wang, Wayne Xin Zhao, and Ji-Rong Wen. Pre-training generative recommender with multi-identifier item tokenization. *CoRR*, abs/2504.04400, 2025. doi: 10.48550/ARXIV.2504.04400. URL https://doi.org/10.48550/arXiv.2504.04400.

Ziyu Zheng, Zhengshun Du, Yaming Yang, Bin Tong, Guan Wang, Meng Yan, Ziyu Guan, and Wei Zhao. Topogr: Revealing and preserving latent structure of semantic ID in generative recommendation. *CoRR*, abs/2607.25216, 2026. doi: 10.48550/ARXIV.2607.25216. URL https://doi.org/10.48550/arXiv.2607.25216.

Qiyong Zhong, Jiajie Su, Yunshan Ma, Julian J. McAuley, and Yupeng Hou. Pctx: Tokenizing personalized context for generative recommendation. *CoRR*, abs/2510.21276, 2025. doi: 10.48550/ARXIV.2510.21276. URL https://doi.org/10.48550/arXiv.2510.21276.

Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux (eds.), *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pp. 1893–1902. ACM, 2020. doi: 10.1145/3340531.3411954. URL https://doi.org/10.1145/3340531.3411954.

A Additional Related Work

LLM-based GR. A parallel line of work builds GR upon pretrained language models and aligns newly introduced item identifiers with the pretrained representation space. Existing methods associate item identifiers with item semantics, collaborative relations, and user contexts, and jointly optimize such alignment objectives with recommendation (Zheng et al., 2024; Liu et al., 2026b; Kong et al., 2025). More recent studies further recognize the representation gap between randomly initialized identifier embeddings and pretrained language representations, motivating finer-grained grounding and richer semantic or collaborative alignment (Feng et al., 2026; Kim et al., 2026; He et al., 2026b). These studies highlight a general grounding principle: discrete recommendation tokens can better exploit previously acquired knowledge when they are explicitly connected to the representation space in which that knowledge is encoded. Whereas LLM-based GR aligns SID embeddings with pretrained language representations to access textual knowledge, the encoder-decoder GR models considered here are trained from scratch and lack such a pretrained representation space. DEGET instead grounds SID embeddings by learning to express DR knowledge acquired through dense pretraining, with SID generation conditioned on the learned user preference representations.

B SID Tokenization

Following the standard item-tokenization paradigm adopted in GR (Rajput et al., 2023; Wang et al., 2025), we construct each item's SID through residual quantization within an RQ-VAE. DEGET retains residual quantization and subsequent collision handling, while the item space used for quantization is learned in the *DE-stage* and remains frozen during tokenizer training.

B.1 RQ-VAE Details

As described in Section 3.2, the learned semantic projector Proj serves as the frozen RQ-VAE encoder, producing the semantic–collaborative item representation $\mathbf{v}_i = \mathrm{Proj}(\mathbf{s}_i)$. Starting from $\mathbf{r}_i^{(0)} = \mathbf{v}_i$, residual quantization assigns a code and updates the residual at each level $\ell = 1, \ldots, K$:

$$c_{i,\ell} = \arg\min_k \left\| \mathbf{r}_i^{(\ell-1)} - \mathbf{q}_k^{(\ell)} \right\|_2^2, \qquad \mathbf{r}_i^{(\ell)} = \mathbf{r}_i^{(\ell-1)} - \mathbf{q}_{c_{i,\ell}}^{(\ell)}. \tag{7}$$

Here, $\mathbf{q}_k^{(\ell)}$ denotes the k-th codeword in the ℓ-th residual codebook, and K is the number of codebooks. This yields the RQ code sequence $C_i^{\mathrm{RQ}} = (c_{i,1}, \ldots, c_{i,K})$. The quantized item representation is the sum of the selected codewords, $\widetilde{\mathbf{v}}_i = \sum_{\ell=1}^{K} \mathbf{q}_{c_{i,\ell}}^{(\ell)}$.

We use the standard straight-through estimator $\widetilde{\mathbf{v}}_i^{\mathrm{st}} = \mathbf{v}_i + \mathrm{sg}[\widetilde{\mathbf{v}}_i - \mathbf{v}_i]$ and reconstruct the input semantic representation as $\widehat{\mathbf{s}}_i = D(\widetilde{\mathbf{v}}_i^{\mathrm{st}})$. Here, D denotes the RQ-VAE decoder, and $\mathrm{sg}[\cdot]$ denotes the stop-gradient operator. The residual codebooks are updated using exponential moving averages (EMA), without a separate gradient-based codebook loss. The RQ-VAE objective consists of the reconstruction loss and a residual commitment term:

$$\mathcal{L}_{\text{RQ-VAE}} = \mathrm{MSE}(\mathbf{s}_i, \widehat{\mathbf{s}}_i) + \beta \sum_{\ell=1}^{K} \mathrm{MSE}\left(\mathbf{r}_i^{(\ell-1)}, \mathrm{sg}[\mathbf{q}_{c_{i,\ell}}^{(\ell)}]\right). \tag{8}$$

Here, MSE denotes the mean squared error over vector dimensions, and β weights the residual commitment term. The loss is averaged over items in each mini-batch. Because the projector is frozen and the codebooks are updated through EMA, the commitment term contributes no parameter gradients in this setting. Gradient-based optimization therefore updates only the RQ-VAE decoder through the reconstruction loss, while EMA updates the codebooks using their assigned residual vectors. The quantization input space remains unchanged throughout training.

B.2 Collision Handling

Residual quantization may assign the same RQ code sequence C_i to multiple items. To ensure that each item has a unique SID, we append a collision token $c_{i,K+1}$ to distinguish items sharing the same RQ codes (Rajput et al., 2023). The final SID is $C_i = (c_{i,1}, \ldots, c_{i,K}, c_{i,K+1})$, with length $L = K + 1$. Items sharing the same RQ code sequence receive distinct collision tokens. The appended token is used only to distinguish items and does not constitute an additional residual quantization level. In our implementation, $K = 3$ and $L = 4$, yielding three RQ tokens followed by one collision token.

C Formal Analysis of Progressive Transfer

We analyze how DEGET transfers DR knowledge through autoregressive SID generation in the *GE-stage* and SID-based history encoding in the *T-stage*. The first analysis connects the pretrained history representations to the GE training objective. The second examines the item-specific information retained by combining SID embeddings with dense item representations before layer normalization.

C.1 Expressing Dense Recommendation through Autoregressive SID Generation

For an interaction history $\mathcal{H} = (i_1, \ldots, i_T)$, the *GE-stage* retains the dense item representations and pretrained history encoder, producing $\mathbf{H} = (\mathbf{h}_1, \ldots, \mathbf{h}_T) = \mathrm{Encoder}(\mathbf{v}_{i_1}, \ldots, \mathbf{v}_{i_T})$. The item representations and encoder remain frozen throughout this stage. Let Y denote the target-item random variable and $C_Y = f(Y) = (c_{Y,1}, \ldots, c_{Y,L})$ its complete SID. Following Eq. equation 3, f is a deterministic one-to-one mapping after collision resolution, and each complete SID contains L tokens. All information-theoretic quantities are evaluated under the joint distribution of target items, pretrained history representations, and corresponding SIDs. We use $H(\cdot)$, $H(\cdot \mid \cdot)$, $I(\cdot;\cdot)$, and $I(\cdot;\cdot \mid \cdot)$ to denote entropy, conditional entropy, mutual information, and conditional mutual information, respectively, with natural logarithms throughout.

DR uses the final history hidden state $\mathbf{h}_{\mathrm{DR}} = \mathbf{h}_T = \mathbf{H}[T]$ for dense matching, whereas the GR decoder receives the complete sequence $\mathbf{H}$ through cross-attention. Since $\mathbf{h}_{\mathrm{DR}}$ is a component of $\mathbf{H}$, $H(Y \mid \mathbf{H}) \le H(Y \mid \mathbf{h}_{\mathrm{DR}})$. Thus, conditioning on the complete history representations preserves access to the target information available in the vector used for dense matching. The one-to-one item-to-SID mapping further yields

$$H(Y \mid \mathbf{H}) = H(C_Y \mid \mathbf{H}) = \sum_{\ell=1}^{L} H(c_{Y,\ell} \mid c_{Y,<\ell}, \mathbf{H}). \tag{9}$$

The first equality follows because the target item and its complete SID uniquely determine each other; the second follows from the conditional entropy chain rule. Together with $H(Y) = H(C_Y)$, this gives $I(Y; \mathbf{H}) = I(C_Y; \mathbf{H})$. Consequently, representing the target item by its complete SID preserves its mutual information with the pretrained history representations. Autoregressive factorization then decomposes the remaining target uncertainty into conditional token uncertainties.

To connect this correspondence to the GE objective, let $p^*_C(C \mid \mathbf{H})$ denote the conditional target SID distribution induced by the data, pretrained history representations, and item-to-SID mapping. Consistent with Eq. equation 6, we use $p(C \mid \mathbf{H})$ to denote the decoder distribution. For a sequence $C = (c_1, \ldots, c_L)$, it factorizes as $p(C \mid \mathbf{H}) = \prod_{\ell=1}^{L} p(c_\ell \mid c_{<\ell}, \mathbf{H})$. Both distributions are defined over the same length-L token-sequence space, with p^*_C assigning zero probability to sequences that do not correspond to catalog items. The expected autoregressive negative log-likelihood corresponding to Eq. equation 6, $\mathcal{R}_{\mathrm{GE}} = \mathbb{E}[-\log p(C_Y \mid \mathbf{H})]$, satisfies

$$\mathcal{R}_{\mathrm{GE}} = H(Y \mid \mathbf{H}) + \mathbb{E}_{\mathbf{H}}\left[D_{\mathrm{KL}}(p^*_C(\cdot \mid \mathbf{H}) \,\|\, p(\cdot \mid \mathbf{H}))\right]. \tag{10}$$

This follows from the cross-entropy decomposition and Eq. equation 9. Because the joint distribution of target items and pretrained history representations remains unchanged during decoder optimization, $H(Y \mid \mathbf{H})$ is constant. Minimizing the expected GE loss therefore minimizes the expected conditional KL divergence between the target and decoder SID distributions. Over unrestricted conditional distributions, the minimum is $H(Y \mid \mathbf{H})$, attained when the two distributions agree for almost every history representation.

These results characterize the role of *Express*: the *GE-stage* trains the decoder to predict the target SID from history representations learned through dense matching, while the one-to-one SID mapping preserves the target information available in those representations. This correspondence does not imply identical predictions from the DE-stage and the generative decoder. The extent to which the decoder realizes the available recommendation capability depends on its capacity, optimization, and generalization, as examined in our transfer experiments.

C.2 Internalizing Dense Recommendation through SID-based History Encoding

The input construction in Section 3.2 combines each SID token embedding with the corresponding frozen semantic–collaborative item representation. To analyze this construction, define the representation before layer normalization as $\mathbf{a}_{i,\ell} = \mathbf{e}_{c_{i,\ell}} + \mathbf{v}_i$, so that $\mathbf{x}_{i,\ell} = \mathrm{LayerNorm}(\mathbf{a}_{i,\ell})$. We characterize the structure and item-specific information preserved by this additive composition. The analysis fixes the model at an arbitrary point during the *T-stage* and uses the SID embeddings at that point.

For any two SID token occurrences (i, ℓ) and (j, r), we have

$$\mathbf{a}_{i,\ell} - \mathbf{a}_{j,r} = \underbrace{(\mathbf{v}_i - \mathbf{v}_j)}_{\text{dense item difference}} + \underbrace{(\mathbf{e}_{c_{i,\ell}} - \mathbf{e}_{c_{j,r}})}_{\text{SID embedding difference}}. \tag{11}$$

This decomposition gives two complementary properties. For two SID positions of the same item, $\mathbf{a}_{i,\ell} - \mathbf{a}_{i,r} = \mathbf{e}_{c_{i,\ell}} - \mathbf{e}_{c_{i,r}}$, so adding the common item representation leaves the differences between their SID embeddings unchanged before normalization. Conversely, whenever $c_{i,\ell} = c_{j,r}$, the shared token embedding cancels and $\mathbf{a}_{i,\ell} - \mathbf{a}_{j,r} = \mathbf{v}_i - \mathbf{v}_j$. Thus, the additive composition retains differences between SID embeddings within an item and preserves dense item differences between occurrences of the same SID token.

The latter property also admits an information-theoretic interpretation. Let J denote a historical item random variable and fix a SID position ℓ. Its SID token is $c_{J,\ell}$, and its frozen dense item representation learned in the *DE-stage* is $\mathbf{v}_J$. Since the shared embedding is determined by the token under the fixed model, it contains no additional information about J once the token is given: $I(J; \mathbf{e}_{c_{J,\ell}} \mid c_{J,\ell}) = 0$. For the additive representation $\mathbf{a}_{J,\ell} = \mathbf{e}_{c_{J,\ell}} + \mathbf{v}_J$, conditioning on $c_{J,\ell}$ determines the token embedding and gives the inverse relation $\mathbf{v}_J = \mathbf{a}_{J,\ell} - \mathbf{e}_{c_{J,\ell}}$. Hence, conditioned on the SID token, the additive representation and the frozen dense item representation determine each other. It follows that

$$H(J \mid \mathbf{a}_{J,\ell}, c_{J,\ell}) = H(J \mid \mathbf{v}_J, c_{J,\ell}), \qquad I(J; \mathbf{a}_{J,\ell} \mid c_{J,\ell}) = I(J; \mathbf{v}_J \mid c_{J,\ell}). \tag{12}$$

Table 4: Statistics of the evaluation datasets. Avg. Len. denotes the average length of user interaction sequences.

Dataset	#Users	#Items	#Interactions	Sparsity	Avg. Len.
			Amazon Reviews 2014		
Beauty	22,363	12,101	198,502	99.927%	8.88
Sports	35,598	18,357	296,337	99.955%	8.32
Toys	19,412	11,924	167,597	99.928%	8.63
			Amazon Reviews 2023		
Instrument	57,439	24,587	511,836	99.964%	8.91
Scientific	50,985	25,848	412,947	99.969%	8.10
Game	94,762	25,612	814,586	99.966%	8.60

Thus, given the SID token, the additive composition preserves exactly the additional item-specific information carried by the frozen dense representation. This preservation does not require the dense representation to uniquely identify the item.

These properties characterize the input construction used in *Internalize*: before normalization, the additive composition retains differences between SID embeddings while preserving dense item information for SID-based history encoding. The information-preservation result is established conditional on the SID token. Since layer normalization is generally not invertible, these results do not establish exact information preservation in $\mathbf{x}_{i,\ell}$ or subsequent encoder hidden states. The effectiveness of this input construction and joint encoder–decoder adaptation is evaluated empirically.

D Experiment Setup

D.1 Datasets & Evaluation

Datasets. We conduct experiments on six real-world benchmarks from Amazon Reviews 2014 (He & McAuley, 2016) and 2023 (Hou et al., 2026b). Following prior work (Rajput et al., 2023; Zhong et al., 2025), we use Beauty, Sports and Outdoors (Sports), and Toys and Games (Toys) from the 2014 version, and Musical Instruments (Instrument), Industrial and Scientific (Scientific), and Video Games (Game) from the 2023 version. We follow the released data preprocessing pipelines of ActionPiece (Hou et al., 2025b)[1] and Pctx (Zhong et al., 2025)[2] for the two versions, respectively. Following standard practice (Zhou et al., 2020; Rajput et al., 2023), we apply 5-core filtering to retain users and items with at least five interactions. For each user, interactions are ordered chronologically, and input sequences are truncated to the 20 most recent interactions for all methods. Table 4 summarizes the processed dataset statistics.

Evaluation. Following the standard leave-last-out protocol (Kang & McAuley, 2018; Rajput et al., 2023), for each user interaction sequence, the most recent item is reserved for testing, the penultimate item is used for validation, and all preceding interactions are used for training. We evaluate all methods without sampled negatives. Dense recommenders rank items over the full candidate catalog, whereas generative recommenders decode valid item SIDs using beam search and rank the generated candidates according to their generation scores. Following TIGER (Rajput et al., 2023), we set the beam size to 20 on the Amazon Reviews 2014 benchmarks. Following Pctx (Zhong et al., 2025)[2], we set the beam size to 50 on the Amazon Reviews 2023 benchmarks. We report Recall@K and Normalized Discounted Cumulative Gain (NDCG)@K to measure whether the ground-truth item is retrieved within the top-K predictions and its ranking position, respectively. The main results use $K \in \{5, 10\}$. For the 2014 benchmarks, we additionally report results at $K = 20$ in Appendix E.1, thereby evaluating recommendation quality over the complete generated beam.

[1] https://github.com/google-deepmind/action_piece

[2] https://github.com/YoungZ365/Pctx

D.2 The Details of Baselines

Dense Recommendation Methods

- **SASRec** (Kang & McAuley, 2018): Encodes item-ID sequences with unidirectional self-attention to capture sequential dependencies for next-item prediction.
- **FDSA** (Zhang et al., 2019): Applies separate self-attention blocks to item-level and feature-level sequences to capture their respective transition patterns, and combines the resulting representations for next-item prediction.
- **BERT4Rec** (Sun et al., 2019): Encodes item-ID sequences with a bidirectional Transformer and learns sequential representations through a Cloze-style masked item prediction objective.
- **S^3-Rec** (Zhou et al., 2020): Uses self-supervised pre-training based on mutual information maximization to learn correlations among attributes, items, subsequences, and sequences, followed by fine-tuning for next-item prediction.
- **LIGER (Dense)** (Yang et al., 2025a)[3]: LIGER integrates sequential dense retrieval into generative retrieval for recommendation. We report its dense retrieval results.

Generative Recommendation

- **TIGER** (Rajput et al., 2023): Quantizes semantic item representations into hierarchical SIDs using RQ-VAE and trains a Transformer encoder-decoder to autoregressively predict the next item's SID.
- **LETTER** (Wang et al., 2024a): Enhances RQ-VAE-based item tokenization with collaborative regularization and diversity regularization to incorporate collaborative signals and reduce code assignment bias.
- **EAGER** (Wang et al., 2024b)[4]: Uses a shared encoder and separate decoders to generate behavior and semantic codes, with a global contrastive task and a semantic-guided transfer task to support behavior-semantic collaboration.
- **UNGER** (Xiao et al., 2025): Integrates semantic and collaborative representations through cross-modality knowledge alignment and next-item prediction, and quantizes the resulting representations into unified item codes. It further applies intra-modality knowledge distillation to mitigate information loss from quantization.
- **ReSID** (Liang et al., 2026)[5]: Learns item representations from structured features through Field-Aware Masked Auto-Encoding and constructs SIDs using Globally Aligned Orthogonal Quantization (GAOQ) to reduce semantic ambiguity and prefix-conditional uncertainty.
- **ETEGRec** (Liu et al., 2025): Couples item tokenization and generative recommendation through sequence–item and preference–semantic alignment, and trains the two components with alternating optimization.
- **DIGER** (Fu et al., 2026): Makes SID learning differentiable so that recommendation gradients can update item identifiers, and uses Gumbel noise with uncertainty decay to mitigate codebook collapse during joint optimization.
- **LIGER (GR)** (Yang et al., 2025a)[3]: LIGER integrates sequential dense retrieval into generative retrieval for recommendation. We report its generative retrieval results.
- **ActionPiece** (Hou et al., 2025b)[1]: Tokenizes action sequences by merging frequently co-occurring feature patterns within and across adjacent actions, with set permutation regularization for multiple segmentations.

Hybrid Recommendation

- **LIGER** (Yang et al., 2025a)[3]: Combines generative retrieval with dense retrieval over item text representations, using generative retrieval to generate candidates and dense retrieval scores to rerank them.

[3] https://github.com/facebookresearch/liger
[4] https://github.com/yewzz/EAGER
[5] https://github.com/FuCongResearchSquad/ReSID

- **MemGen-GR** (Ding et al., 2026)[6]: Exploits the complementary memorization and generalization capabilities of SASRec and TIGER through an adaptive ensemble, with instance-specific weights determined by SASRec's prediction confidence. We reproduce MemGen-GR using our reproduced SASRec and TIGER models as the two component recommenders.

Additional Generative Recommendation Baselines

For the following baselines, we report results from their original papers on the Amazon Reviews 2023 benchmarks.

- **Pctx** (Zhong et al., 2025): Conditions item tokenization on user interaction histories, allowing the same item to have different SIDs under different user contexts. It uses alternative SIDs of the same item for training augmentation and aggregates their generation probabilities during inference.
- **Latte** (Hou et al., 2026a): Prepends a latent token to each SID to form multiple decoding trees conditioned on latent tokens, alleviating tree-induced probability coupling among items. It aggregates generation scores across latent paths corresponding to the same item during inference.

LLM-based Generative Recommendation

For the following baselines, we report results on the Amazon Reviews 2014 benchmarks. The results are taken from IIRG (Kim et al., 2026), and PauseRec (He et al., 2026b).

- **EAGER-LLM** (Hong et al., 2025): Integrates external semantic and behavioral representations into LLM-based recommendation through dual-source item indices and multiscale alignment-reconstruction tasks.
- **PLUM** (He et al., 2026a): Constructs SIDs using RQ-VAE with multimodal item representations and co-occurrence contrastive regularization, and adapts pretrained LLMs through vocabulary expansion, continued pretraining, and recommendation fine-tuning.
- **OneRec-Think** (Liu et al., 2026b): Enables explicit reasoning for generative recommendation through item–text alignment, reasoning scaffolding, and reinforcement learning with recommendation-specific rewards.
- **PauseRec** (He et al., 2026b): Uses pause-token pretraining and implicit-reasoning fine-tuning to provide latent computation before SID prediction, without explicit rationale generation or reinforcement learning.

D.3 Implementation Details

Default Semantic Item Representations. Following MTGRec (Zheng et al., 2025)[7] and Pctx[2], we encode item text using Sentence-T5 (Ni et al., 2022) and apply PCA with whitening to reduce the embeddings from 768 to 128 dimensions. For Amazon Reviews 2014 and 2023, item text includes titles, prices, brands, and categories; and titles, features, categories, and descriptions, respectively.

Default RQ-VAE. We follow the neural RQ-VAE configuration provided by MTGRec[7] and Pctx[2]. The encoder has layer dimensions $[2048, 1024, 512, 256, 128]$, with a mirrored decoder. We use three codebooks of size 256 and dimension 128, initialized with K-means and updated using exponential moving averages (EMA) with a decay of 0.99. The tokenizer is trained using Adagrad with a learning rate of 10^{-3} and a batch size of 2048 for up to 20,000 epochs, with evaluation every 100 epochs. Following ActionPiece[1], we train the tokenizer once per dataset with seed 2024 and keep the SID mapping fixed across recommender seeds. Collision handling is detailed in Appendix B.2.

Default Backbone and Baseline Reproduction. For the GR backbone, we follow ActionPiece[1] and use a T5 encoder–decoder with 4 encoder layers and 4 decoder layers, a hidden size of 128, a feed-forward dimension of 1024, 6 attention heads with a key/value dimension of 64 per head, ReLU activation, and a dropout rate of 0.1. The default batch size is 256, and models are trained for up to 200 epochs with checkpoints selected by validation NDCG@10. For Amazon Reviews 2014, we use

[6] https://github.com/Jamesding000/MemGen-GR
[7] https://github.com/RUCAIBox/MTGRec

Table 5: Standard deviations over three random seeds for the results in Table 1 on Amazon Reviews 2014. All values are multiplied by 100.

Methods	Beauty				Sports				Toys			
	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10
Dense Recommendation												
SASRec	0.24	0.27	0.16	0.17	0.09	0.08	0.07	0.08	0.11	0.21	0.11	0.13
FDSA	0.17	0.26	0.08	0.11	0.18	0.27	0.13	0.15	0.05	0.18	0.09	0.13
BERT4Rec	0.06	0.03	0.06	0.05	0.01	0.07	0.01	0.03	0.13	0.19	0.09	0.11
S^3-Rec	0.09	0.19	0.08	0.11	0.12	0.21	0.12	0.15	0.26	0.34	0.18	0.19
LIGER (Dense)	0.12	0.09	0.04	0.03	0.02	0.10	0.02	0.04	0.07	0.10	0.04	0.07
Generative Recommendation												
TIGER	0.09	0.16	0.09	0.11	0.17	0.26	0.12	0.15	0.45	0.57	0.32	0.35
LETTER	0.13	0.11	0.11	0.08	0.09	0.15	0.08	0.10	0.26	0.24	0.16	0.16
EAGER	0.13	0.06	0.12	0.10	0.02	0.02	0.02	0.02	0.14	0.34	0.02	0.08
UNGER	0.11	0.01	0.06	0.03	0.04	0.11	0.04	0.06	0.09	0.10	0.06	0.04
ReSID	0.08	0.08	0.02	0.03	0.06	0.03	0.07	0.06	0.14	0.26	0.12	0.16
ETEGRec	0.14	0.23	0.07	0.10	0.07	0.05	0.03	0.02	0.10	0.13	0.09	0.08
DIGER	0.13	0.10	0.11	0.10	0.15	0.11	0.10	0.09	0.18	0.13	0.12	0.10
LIGER (GR)	0.04	0.03	0.05	0.04	0.10	0.08	0.05	0.05	0.16	0.18	0.11	0.12
ActionPiece	0.07	0.03	0.04	0.02	0.02	0.05	0.01	0.02	0.12	0.11	0.06	0.06
Hybrid Recommendation												
LIGER	0.06	0.10	0.02	0.03	0.04	0.04	0.02	0.02	0.08	0.15	0.07	0.11
MemGen-GR	0.20	0.20	0.11	0.11	0.17	0.24	0.12	0.14	0.34	0.33	0.25	0.25
Ours (GR)												
DEGET	0.06	0.13	0.04	0.01	0.07	0.11	0.03	0.03	0.08	0.02	0.04	0.01
TIGER + **DEGET**	0.21	0.42	0.19	0.25	0.07	0.07	0.04	0.04	0.01	0.16	0.03	0.02
ActionPiece + **DEGET**	0.06	0.12	0.02	0.02	0.04	0.03	0.02	0.02	0.04	0.13	0.05	0.06

a learning rate of 3×10^{-3}, weight decay of 0.05, and a cosine learning rate schedule with 10,000 warm-up steps. For TIGER, validation begins after 50 epochs, with an early stopping patience of 25 epochs; other methods follow the validation and early stopping settings in the corresponding implementations. For baselines with publicly available code, we evaluate both the original and unified configurations and report the better results. The default beam sizes are 20 and 50 for Amazon Reviews 2014 and 2023, respectively. Under our unified configuration, all methods use the default semantic item representations, while those using RQ-VAE also follow the default configuration. For EAGER[4], ReSID[5], and ActionPiece[1], we use the larger beam sizes specified in the corresponding official implementations to accommodate their method designs. For DR baselines, we implement the models using RecBole (Zhao et al., 2021) or the official implementations.

DEGET. We train each stage for up to 200 epochs with early stopping. On Amazon Reviews 2014, the learning rates for the *DE-stage*, *GE-stage*, and *T-stage* are 3×10^{-3}, 3×10^{-3}, and 3×10^{-4}, with early stopping patience of 10, 10, and 15 epochs, respectively. On Amazon Reviews 2023, the corresponding learning rates are 3×10^{-3}, 3×10^{-3}, and 10^{-4}, with a patience of 20 epochs for all stages. In the *DE-stage*, item biases are initialized to zero and the temperature τ is fixed at 1.0.

Reproducibility. All experiments are implemented in Python 3.9.25 with PyTorch 2.8.0, Transformers 4.41.2, and Sentence-Transformers 3.0.1. Each training and evaluation run is conducted on a single NVIDIA L40 GPU with 48 GB of memory.

E Extended Experimental Evaluation

We further examine DR knowledge as a pretraining foundation for GR across datasets and generation paradigms. Specifically, we report standard deviations and additional results at $K = 20$, evaluate DEGET on Amazon Reviews 2023, and compare with LLM-based GR methods.

E.1 Additional Results on Amazon Reviews 2014

Stability Analysis. Table 5 reports standard deviations over three random seeds to assess the stability of DEGET. DEGET exhibits consistently lower variability than TIGER across datasets and metrics, reinforcing the reliability of the gains from progressive dense-to-generative transfer.

Table 6: Performance comparison at $K = 20$ on Amazon Reviews 2014, averaged over three random seeds. Subscripts report sample standard deviations multiplied by 100. Percentage rows report relative improvements over the corresponding baseline.

Methods	Beauty		Sports		Toys	
	R@20	N@20	R@20	N@20	R@20	N@20
Dense Recommendation						
SASRec	$0.0944_{\pm 0.40}$	$0.0409_{\pm 0.20}$	$0.0544_{\pm 0.17}$	$0.0235_{\pm 0.10}$	$0.1008_{\pm 0.27}$	$0.0478_{\pm 0.15}$
FDSA	$0.0634_{\pm 0.43}$	$0.0262_{\pm 0.15}$	$0.0409_{\pm 0.25}$	$0.0172_{\pm 0.14}$	$0.0726_{\pm 0.22}$	$0.0329_{\pm 0.14}$
BERT4Rec	$0.0659_{\pm 0.06}$	$0.0284_{\pm 0.05}$	$0.0337_{\pm 0.15}$	$0.0134_{\pm 0.05}$	$0.0540_{\pm 0.27}$	$0.0236_{\pm 0.12}$
S^3-Rec	$0.1000_{\pm 0.24}$	$0.0410_{\pm 0.12}$	$0.0588_{\pm 0.25}$	$0.0253_{\pm 0.15}$	$0.0907_{\pm 0.37}$	$0.0381_{\pm 0.21}$
LIGER (Dense)	$\underline{0.1294}_{\pm 0.12}$	$\underline{0.0564}_{\pm 0.03}$	$\underline{0.0751}_{\pm 0.08}$	$\underline{0.0321}_{\pm 0.02}$	$\underline{0.1326}_{\pm 0.02}$	$\underline{0.0547}_{\pm 0.07}$
Generative Recommendation						
TIGER	$0.0989_{\pm 0.26}$	$0.0447_{\pm 0.14}$	$0.0643_{\pm 0.26}$	$0.0284_{\pm 0.15}$	$0.0973_{\pm 0.61}$	$0.0434_{\pm 0.34}$
LETTER	$0.1042_{\pm 0.14}$	$0.0489_{\pm 0.08}$	$0.0611_{\pm 0.23}$	$0.0267_{\pm 0.12}$	$0.0942_{\pm 0.24}$	$0.0424_{\pm 0.14}$
EAGER	$0.0892_{\pm 0.23}$	$0.0431_{\pm 0.08}$	$0.0538_{\pm 0.14}$	$0.0239_{\pm 0.05}$	$0.0837_{\pm 0.58}$	$0.0432_{\pm 0.14}$
UNGER	$0.1042_{\pm 0.07}$	$0.0454_{\pm 0.03}$	$0.0623_{\pm 0.12}$	$0.0268_{\pm 0.06}$	$0.0989_{\pm 0.13}$	$0.0429_{\pm 0.07}$
ReSID	$0.1064_{\pm 0.25}$	$0.0487_{\pm 0.04}$	$0.0659_{\pm 0.09}$	$0.0286_{\pm 0.08}$	$0.0917_{\pm 0.34}$	$0.0411_{\pm 0.17}$
ETEGRec	$0.1110_{\pm 0.08}$	$0.0507_{\pm 0.06}$	$0.0671_{\pm 0.08}$	$0.0292_{\pm 0.03}$	$0.1012_{\pm 0.25}$	$0.0445_{\pm 0.11}$
DIGER	$0.1018_{\pm 0.15}$	$0.0458_{\pm 0.10}$	$0.0615_{\pm 0.23}$	$0.0268_{\pm 0.12}$	$0.0975_{\pm 0.22}$	$0.0428_{\pm 0.13}$
LIGER (GR)	$0.1167_{\pm 0.10}$	$0.0511_{\pm 0.05}$	$0.0696_{\pm 0.07}$	$0.0298_{\pm 0.04}$	$0.1135_{\pm 0.09}$	$0.0484_{\pm 0.07}$
ActionPiece	$0.1123_{\pm 0.17}$	$0.0513_{\pm 0.05}$	$0.0705_{\pm 0.02}$	$0.0313_{\pm 0.01}$	$0.1080_{\pm 0.12}$	$0.0477_{\pm 0.03}$
Hybrid Recommendation						
LIGER	$0.1174_{\pm 0.08}$	$0.0531_{\pm 0.02}$	$0.0698_{\pm 0.06}$	$0.0308_{\pm 0.03}$	$0.1159_{\pm 0.05}$	$0.0501_{\pm 0.08}$
MemGen-GR	$0.1078_{\pm 0.28}$	$0.0504_{\pm 0.13}$	$0.0677_{\pm 0.34}$	$0.0307_{\pm 0.16}$	$0.1126_{\pm 0.34}$	$0.0546_{\pm 0.23}$
Ours (GR)						
DEGET	$\mathbf{0.1308}_{\pm 0.20}$	$\mathbf{0.0584}_{\pm 0.03}$	$\mathbf{0.0782}_{\pm 0.13}$	$\mathbf{0.0345}_{\pm 0.04}$	$\mathbf{0.1334}_{\pm 0.08}$	$\mathbf{0.0587}_{\pm 0.02}$
vs. LIGER	↑ 11.41%	↑ 9.98%	↑ 12.03%	↑ 12.01%	↑ 15.10%	↑ 17.17%
TIGER + **DEGET**	$0.1213_{\pm 0.45}$	$0.0535_{\pm 0.26}$	$0.0743_{\pm 0.17}$	$0.0325_{\pm 0.07}$	$0.1245_{\pm 0.08}$	$0.0555_{\pm 0.04}$
vs. TIGER	↑ 22.65%	↑ 19.69%	↑ 15.55%	↑ 14.44%	↑ 27.95%	↑ 27.88%
ActionPiece + **DEGET**	$0.1294_{\pm 0.11}$	$0.0574_{\pm 0.02}$	$0.0795_{\pm 0.05}$	$0.0353_{\pm 0.02}$	$0.1264_{\pm 0.02}$	$0.0551_{\pm 0.04}$
vs. ActionPiece	↑ 15.23%	↑ 11.89%	↑ 12.77%	↑ 12.78%	↑ 17.04%	↑ 15.51%

Top-20 Recommendation. To examine whether the gains persist for longer recommendation lists, we evaluate Recall@20 and NDCG@20 with the decoding configurations unchanged. Table 6 shows that DEGET maintains its performance advantage, outperforming the strongest baseline, LIGER (Dense), by 4.02% on average across datasets and metrics. TIGER and ActionPiece also consistently benefit from DEGET while retaining their original tokenizers and SIDs. We attribute these gains to the item-level discriminative knowledge acquired through full-catalog DR pretraining, which learns user–item relevance by contrasting each target item against all other items in the catalog. These findings suggest that progressively transferring this knowledge into history encoding and SID generation strengthens GR's ranking ability by complementing token-level generation supervision with item-level discrimination learned over the entire catalog.

E.2 Results on Amazon Reviews 2023

We extend the evaluation to three larger datasets from Amazon Reviews 2023 to assess the effectiveness of progressive transfer beyond the 2014 benchmarks. Table 7 shows that DEGET achieves better overall performance than dense, hybrid, and generative baselines. It consistently outperforms TIGER across all datasets and metrics, with an average relative improvement of 15.46%, and improves over LIGER by 5.09% on average while retaining purely generative inference. These results demonstrate that the benefits of DR pretraining extend to larger recommendation datasets.

Notably, Pctx incorporates personalized context into SID construction, while Latte introduces latent tokens to enable multiple generation paths; both aggregate predictions across paths for the same item. DEGET achieves better overall performance using a single fixed SID per item and standard autoregressive SID generation. These comparisons highlight the value of DR pretraining in strengthening user preference modeling within GR. As user intents continually evolve in real-world settings, DEGET offers practical value by modeling personalized preferences internally without requiring corresponding changes to item identifiers or their decoding paths.

Table 7: Performance comparison on Amazon Reviews 2023. * indicates statistical significance at the $p < 0.01$ level, as determined by a paired t-test against TIGER. R@K and N@K denote Recall@K and NDCG@K, respectively.

Methods	Instrument				Scientific				Game			
	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10
Dense Recommendation												
SASRec	0.0333	0.0523	0.0213	0.0274	0.0259	0.0412	0.0150	0.0199	0.0535	0.0847	0.0331	0.0438
FDSA	0.0347	0.0545	0.0230	0.0293	0.0262	0.0421	0.0169	0.0213	0.0544	0.0852	0.0361	0.0448
BERT4Rec	0.0307	0.0485	0.0195	0.0252	0.0186	0.0296	0.0119	0.0155	0.0460	0.0735	0.0298	0.0386
S^3-Rec	0.0317	0.0496	0.0199	0.0257	0.0263	0.0418	0.0171	0.0219	0.0485	0.0769	0.0315	0.0406
Generative Recommendation												
TIGER	0.0376	0.0570	0.0248	0.0310	0.0279	0.0428	0.0184	0.0232	0.0582	0.0901	0.0380	0.0482
LETTER	0.0372	0.0580	0.0246	0.0313	0.0279	0.0435	0.0182	0.0232	0.0563	0.0877	0.0372	0.0473
ETEGRec	0.0395	0.0625	0.0257	0.0328	0.0288	0.0448	0.0189	0.0238	0.0589	0.0912	0.0385	0.0498
ActionPiece	0.0383	0.0615	0.0243	0.0318	0.0284	0.0452	0.0182	0.0236	0.0591	0.0927	0.0382	0.0490
Pctx	0.0419	0.0655	**0.0275**	0.0350	0.0323	0.0504	0.0205	0.0263	0.0638	0.0981	0.0416	0.0527
Latte	0.0401	0.0618	0.0261	0.0331	0.0304	0.0470	0.0196	0.0249	0.0618	0.0958	0.0406	0.0515
Hybrid Recommendation												
LIGER	0.0408	0.0630	0.0264	0.0334	0.0327	**0.0528**	0.0203	0.0265	0.0619	0.0969	0.0391	0.0503
vs. TIGER	↑ 8.51%	↑ 10.53%	↑ 6.45%	↑ 7.74%	↑ 17.20%	↑ 23.36%	↑ 10.33%	↑ 14.22%	↑ 6.36%	↑ 7.55%	↑ 2.89%	↑ 4.36%
Ours (GR)												
DEGET	**0.0431***	**0.0672***	**0.0275***	**0.0352***	**0.0334***	0.0526*	**0.0218***	**0.0280***	**0.0648***	**0.1021***	**0.0419***	**0.0539***
vs. TIGER	↑ 14.63%	↑ 17.89%	↑ 10.89%	↑ 13.55%	↑ 19.71%	↑ 22.94%	↑ 18.37%	↑ 20.52%	↑ 11.34%	↑ 13.32%	↑ 10.26%	↑ 11.87%

Table 8: Performance comparison with LLM-based GR on Amazon Reviews 2014. DEGET results are averaged over three random seeds.

Methods	Beauty				Sports				Toys			
	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10
LLM-based GR												
EAGER-LLM	0.0548	0.0830	0.0369	0.0459	**0.0373**	**0.0569**	**0.0251**	**0.0315**	0.0546	0.0788	0.0359	0.0437
PLUM	0.0522	0.0745	0.0360	0.0433	0.0297	0.0436	0.0199	0.0245	0.0549	0.0737	0.0389	0.0449
OneRec-Think	0.0563	0.0791	0.0398	0.0471	0.0288	0.0412	0.0199	0.0239	0.0579	0.0797	0.0412	0.0482
PauseRec	0.0568	0.0746	**0.0401**	0.0467	0.0294	0.0422	0.0203	0.0245	**0.0615**	0.0838	**0.0434**	**0.0509**
Ours (GR)												
DEGET	**0.0577**	**0.0907**	0.0377	**0.0483**	0.0344	0.0528	0.0222	0.0281	0.0601	**0.0937**	0.0378	0.0486

E.3 Comparison with LLM-based GR

We compare DEGET with LLM-based GR to assess the effectiveness of DR knowledge transfer. Overall, Table 8 shows that DEGET achieves competitive performance with the evaluated LLM-based GR methods using a compact encoder-decoder with substantially fewer backbone parameters. These comparisons highlight DR pretraining as an effective foundation for user preference modeling in GR. By internalizing this recommendation knowledge through progressive transfer, DEGET achieves competitive performance through direct SID generation without additional reasoning or pause tokens. The comparison also highlights the importance of adapting previously acquired knowledge to recommendation. EAGER-LLM aligns external semantic and behavioral representations with the LLM (Hong et al., 2025), while PLUM and PauseRec use continued pretraining to connect SIDs with pretrained language knowledge (He et al., 2026a;b). DEGET instead acquires recommendation knowledge through dense pretraining and progressively transfers it into generative history modeling and SID prediction. Its competitive performance suggests that recommendation knowledge learned through user–item matching can provide a strong foundation for GR without requiring a pretrained language model as the recommendation backbone.

Notably, beyond next-item recommendation, LLM-based GR can draw on broader pretrained knowledge and reasoning capabilities. For example, OneReason (Team et al., 2026) largely retains its backbone's general reasoning and instruction-following abilities after recommendation-oriented training on the evaluated benchmarks. It also improves online recommendation by distilling knowledge into OneRec (Deng et al., 2025). Together with our findings, these results motivate future work on bidirectional knowledge transfer between train-from-scratch and LLM-based GR to combine preference modeling with language understanding and reasoning.

F Understanding Progressive Dense-to-Generative Transfer

In this section, we further analyze the DR capability retained by the T-stage encoder and changes in prediction at different SID levels from the GE-stage to the T-stage.

F.1 Retention of DR Capability in the Encoder

To further assess the internalization of DR knowledge in the encoder after transfer, we evaluate the T-stage encoder using the original DE-stage prediction head without retraining. Figure 6 shows that SID + Item Rep retains approximately 81.8% of DE's NDCG@20 while sharing 40.1% of its top-20 items, averaged across the three datasets. Thus, substantial dense ranking capability persists despite changes in the retrieved item sets. Item Rep only closely matches SID + Item Rep on both metrics, whereas SID only achieves an average relative NDCG@20 of 6.3% and a top-20 overlap of 4.3%. This contrast suggests that the frozen semantic–collaborative item representations help maintain the adapted encoder's compatibility with dense item scoring, supporting the retention of DR capability within GR. We further observe that this capability remains accessible even though subsequent training optimizes only SID generation, suggesting that DR provides GR with item-level discriminative knowledge. Together with the generative retrieval results in Figure 5, these findings demonstrate that DR can serve as a strong pretraining foundation for GR, enabling the generative model to build on item-level preference knowledge when learning autoregressive SID prediction.

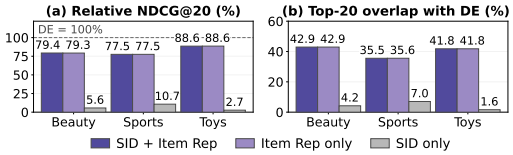


Figure 6: DR capability retained by the T-stage encoder. (a) Relative NDCG@20: NDCG@20 of T divided by that of DE. (b) Top-20 Overlap with DE: rate of top-20 retrieved items shared.

F.2 SID Prediction across Transfer Stages

To identify which parts of SID generation benefit from GE-stage to T-stage, we compare GE and T using target-token negative log-likelihood (NLL) under teacher forcing. Figure 7 shows that the largest NLL reduction consistently occurs at the third residual quantization level across all three datasets. Changes at the first two levels are much smaller and vary in sign, while the collision token exhibits only a small reduction. Residual quantization represents items at progressively finer granularity (Rajput et al., 2023), with later codes refining the information captured by earlier codes. Since both stages receive the same ground-truth SID prefixes, the third-level improvement reflects better conditional prediction after the first two codes are specified. Moreover, complete SID log-likelihood is the sum of conditional token log-likelihoods, so the dominant third-level reduction accounts for most of the overall target-SID likelihood improvement. These findings suggest that the T-stage further refines the expression of transferred DR knowledge through fine-grained SID prediction.

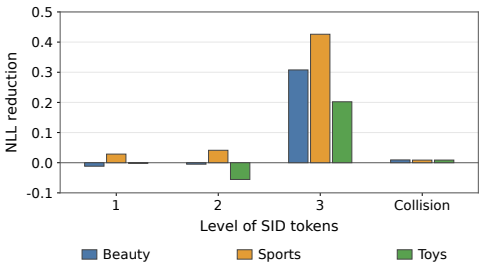


Figure 7: Negative log-likelihood (NLL) reduction from the GE-stage to the T-stage at each SID level. Values denote $\mathrm{NLL}_{\mathrm{GE}} - \mathrm{NLL}_{\mathrm{T}}$. Positive values indicate improvement.

G From Pretrained DR to GR

From the perspective of DR, DEGET provides a pathway for adapting pretrained dense recommenders into generative recommenders by augmenting them with a decoder and applying progressive knowledge transfer. We also investigate the effect of the added decoder scale under both bidirectional and causal encoder architectures.

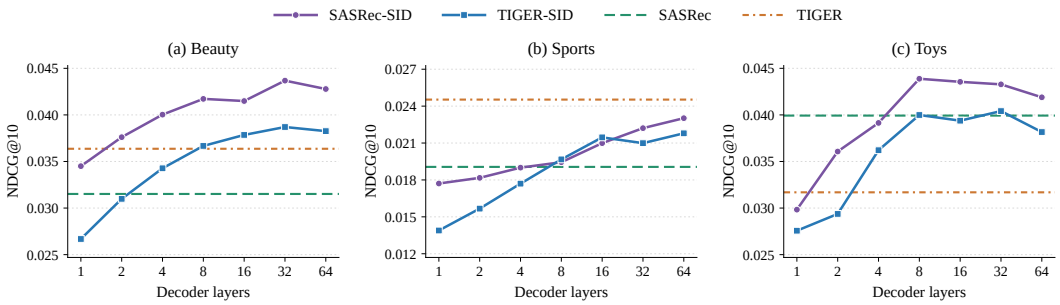


Figure 8: NDCG@10 of SASRec with the DEGET transfer strategy at different decoder depths. SASRec-SID and TIGER-SID are constructed by applying RQ-VAE to pretrained SASRec item embeddings and semantic item embeddings, respectively.

G.1 Adapting Pretrained SASRec

Figure 8 compares SASRec-SID and TIGER-SID with decoder depths of $\{1, 2, 4, 8, 16, 32, 64\}$, using the original SASRec and TIGER as reference baselines. First, SASRec-SID generally outperforms TIGER-SID, and increasing decoder depth alone does not close the performance gap between the two SID configurations. This highlights the importance of constructing SIDs from item representations learned through DR to exploit pretrained DR knowledge during transfer. Second, transfer achieves a balance between generative and dense recommendation performance, but further increasing decoder depth does not yield consistent improvements. Effective transfer therefore requires more than enlarging the decoder. These findings point to a future research direction: how to better adapt well-trained dense recommenders to generative retrieval. This direction holds promise for industrial systems seeking to reuse existing recommendation models.

G.2 Effect of Decoder Depth in DEGET

We investigate whether increasing decoder depth improves recommendation performance in DEGET, which adopts the bidirectional history encoding commonly used in encoder-decoder GR. We evaluate decoder depths of $\{1, 2, 4, 6, 8, 16\}$. Figure 9 shows that the highest NDCG@10 is obtained with 4 layers on Beauty and Toys and 6 layers on Sports, with no further improvementat greater depths. We therefore adopt a four-layer decoder for DEGET, balancing recommendation performance across datasets with model complexity in dense-to-generative transfer. These results suggest that increasing decoder depth alone does not yield consistent performance gains across datasets under our transfer strategy. Additional layers increase the computation required at each autoregressive decoding step; we therefore adopt 4 decoder layers in DEGET to balance recommendation performance and computational cost. Future work could explore additional training and alignment strategies to better transfer DR knowledge into GR.

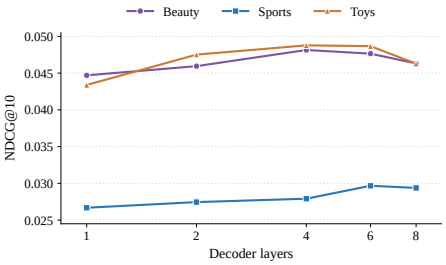


Figure 9: Effect of decoder depth on DEGET.

H Efficiency Analysis

Table 9 shows that DEGET retains inference efficiency close to TIGER, with comparable throughput and GPU memory usage. All methods use aligned backbone configurations on an NVIDIA L40 GPU in FP32, with batch size 256, beam size 20, and KV caching disabled. ActionPiece uses a single deterministic history tokenization without inference-time ensembling. Compared with TIGER, DEGET requires only 0.16 seconds of additional test time and 0.02 GiB of additional peak GPU memory, including its frozen item representations. These representations provide DR knowledge

during history encoding, while recommendations are generated directly as SIDs without dense candidate scoring. Thus, progressive transfer enables GR to exploit DR knowledge while preserving the inference efficiency of its generative backbone.

Table 9: Inference efficiency and recommendation performance on 2014 Beauty. Time denotes total test time, throughput measures user histories processed per second, and memory is peak allocated GPU tensor memory.

Method	Time (s)	Throughput (users/s)	Memory (GiB)
TIGER	15.80	1415.57	2.32
ActionPiece	108.24	206.60	15.13
LIGER	17.12	1306.30	2.33
DEGET	15.96	1401.20	2.34

I Applicability of our methods to Decoder-only model

In this section, we investigate the applicability of our methods to decoder-only GR. We apply DEGET to RPG (Hou et al., 2025a), a decoder-only generative recommender trained from scratch that predicts long SIDs in parallel and uses graph-constrained decoding, yielding **DEGET-RPG**. The main adaptations are to perform the *DE-stage* on RPG's Transformer decoder and quantize the learned semantic-collaborative item representations using optimized product quantization (OPQ). To assess the value of pretrained DR knowledge, we further compare against MHL (Wei et al., 2026), which augments RPG with entropy-guided masked history reconstruction and curriculum learning. For a fair comparison, the shared experimental settings and hyperparameters follow MHL's official implementation[8].

Table 10 shows that DEGET-RPG achieves an average relative improvement of 24.43% over RPG across all reported metrics on Beauty and Sports, with better overall performance than MHL. This comparison suggests that transferring DR knowledge offers benefits beyond optimizing SID sequence modeling alone. Together with the encoder-decoder results, these findings indicate that transferring item representations and user preference knowledge learned through DR enables GR models to internalize pretrained recommendation knowledge. DR knowledge can therefore serve as a common pretraining foundation for GR with different SID structures and model architectures.

Table 10: Performance comparison on the Amazon Reviews 2014 benchmarks, averaged over three random seeds.

Methods	Beauty						Sports					
	R@5	R@10	R@20	N@5	N@10	N@20	R@5	R@10	R@20	N@5	N@10	N@20
RPG	0.0516	0.0734	0.1034	0.0370	0.0440	0.0515	0.0287	0.0422	0.0620	0.0196	0.0239	0.0289
MHL	0.0555	0.0775	0.1076	**0.0410**	0.0481	0.0557	0.0346	0.0488	0.0665	0.0248	0.0294	0.0339
DEGET-RPG	**0.0603**	**0.0889**	**0.1262**	0.0408	**0.0500**	**0.0594**	**0.0377**	**0.0572**	**0.0830**	**0.0253**	**0.0316**	**0.0381**
vs. RPG	↑ 16.86%	↑ 21.12%	↑ 22.05%	↑ 10.27%	↑ 13.64%	↑ 15.34%	↑ 31.36%	↑ 35.55%	↑ 33.87%	↑ 29.08%	↑ 32.22%	↑ 31.83%

[8] https://github.com/CQU-MM-Intelligent-Lab/MHL