

Grounded Token Initialization for New Vocabulary in LMs for Generative Recommendation

Daiwei Chen^{1,2*}, Zhoutong Fu², Chengming Jiang², Haichao Zhang³, Ran Zhou², Tan Wang², Chunnan Yao², Guoyao Li², Rui Cai⁴, Yihan Cao², Ruijie Jiang², Fedor Borisjuk², Jianqiang Shen², Jingwei Wu², Ramya Korlakai Vinayak¹

¹University of Wisconsin-Madison ²LinkedIn Corporation

³Northeastern University ⁴University of California, Davis

Abstract

Language models (LMs) are increasingly extended with new learnable vocabulary tokens for domain-specific tasks, such as Semantic-ID tokens in generative recommendation. The standard practice initializes these new tokens as the mean of existing vocabulary embeddings, then relies on supervised fine-tuning to learn their representations. We present a systematic analysis of this strategy: through spectral and geometric diagnostics, we show that mean initialization collapses all new tokens into a degenerate subspace, erasing inter-token distinctions that subsequent fine-tuning struggles to fully recover. These findings suggest that *token initialization* is a key bottleneck when extending LMs with new vocabularies. Motivated by this diagnosis, we propose the *Grounded Token Initialization Hypothesis*: linguistically grounding novel tokens in the pretrained embedding space before fine-tuning better enables the model to leverage its general-purpose knowledge for novel-token domains. We operationalize this hypothesis as **GTI** (Grounded Token Initialization), a lightweight grounding stage that, prior to fine-tuning, maps new tokens to distinct, semantically meaningful locations in the pretrained embedding space using only paired linguistic supervision. Despite its simplicity, **GTI** outperforms both mean initialization and existing auxiliary-task adaptation methods in the majority of evaluation settings across multiple generative recommendation benchmarks, including industry-scale and public datasets. Further analyses show that grounded embeddings produce richer inter-token structure that persists through fine-tuning, corroborating the hypothesis that initialization quality is a key bottleneck in vocabulary extension.

1 Introduction

Pretrained language models (LMs) are increasingly adapted to specialized domains by extending their

vocabulary with new learnable tokens. A prominent example is generative retrieval, where items (Rajput et al., 2023; Deldjoo et al., 2024) or documents (Tay et al., 2022) are assigned discrete semantic codes and generated autoregressively by the LM; similar challenges arise whenever domain-specific symbols must be integrated into a pretrained vocabulary. These systems introduce thousands of new tokens into the model’s vocabulary, and a fundamental challenge is how to incorporate them into the pretrained embedding space so that the LM can transfer its general-purpose knowledge to the novel-token domain.

The prevailing practice initializes new token embeddings as the mean of the existing vocabulary embeddings (Hewitt, 2021). This heuristic is widely adopted because it is simple, places new tokens on the pretrained embedding manifold, and provides a tighter KL-divergence upper bound on output probabilities. However, it collapses all new tokens into a single point in embedding space, erasing inter-token distinctions and stripping domain-level semantics. An existing alternative (Zheng et al., 2024) employs auxiliary-task adaptation of the full LM to induce linguistic signals for new tokens, but the multi-task training introduces an objective mismatch: the auxiliary losses are not aligned with the target downstream task, resulting in limited and inconsistent gains.

In this paper, we identify token-embedding misalignment as a fundamental limitation when extending pretrained LMs with new vocabulary, and propose the *Grounded Token Initialization Hypothesis*: **linguistically grounding novel tokens in an LM’s pretrained embedding space, before fine-tuning, better enables the model to leverage its general-purpose knowledge for novel-token domains**. The intuition is that pretrained LM embeddings encode rich linguistic structure, semantically related tokens occupy nearby regions (Levy and Goldberg, 2014), and the model’s attention and

*Work done during internship at LinkedIn Corporation.
Correspondence to: dchen365@wisc.edu

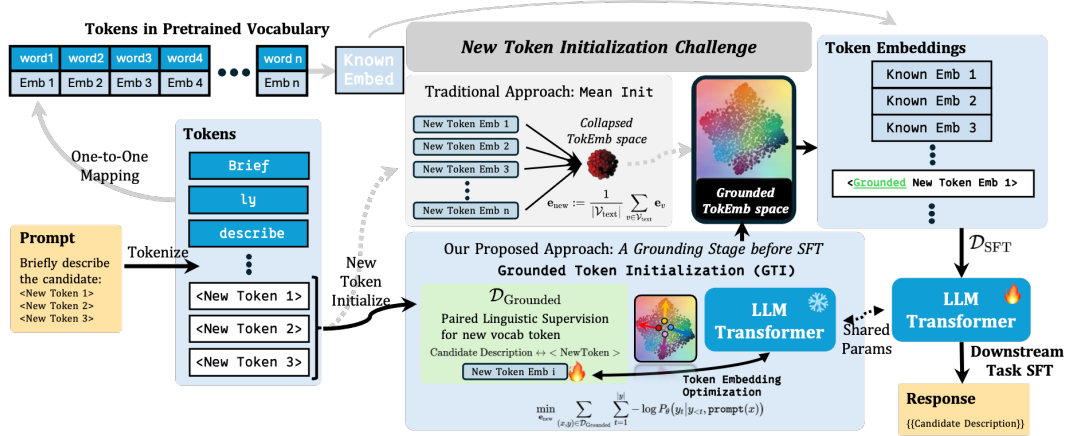


Figure 1: Overview of the **GTI** grounding stage. The LM backbone and original vocabulary embeddings are frozen (snowflake); only the newly introduced Semantic-ID (SID) token embeddings ($|\mathcal{V}_{\text{SID}}| \times D$ parameters, fire) are trained. Paired prompts map between natural language descriptions and SID tokens in both directions, grounding the new tokens in the pretrained embedding space. This stage is inserted before standard end-to-end fine-tuning.

feed-forward layers have learned to exploit this geometry (Gao et al., 2019). If new tokens are placed meaningfully within this structure, the LM can immediately leverage its existing representations to process them in context, rather than relying on fine-tuning alone to recover from a degenerate starting point. This motivates framing vocabulary extension as a token-grounding problem: new token embeddings should be grounded in linguistically meaningful representations while remaining coherent with the pretrained LM’s embedding geometry.

Building on this hypothesis, we introduce **GTI**, a simple and effective grounded token initialization method. Before downstream fine-tuning, **GTI** freezes the LM backbone and grounds newly introduced token using paired supervision between natural language descriptions and the corresponding new tokens (Figure 1). This grounding stage resolves the mismatch between well-trained vocabulary embeddings and newly initialized tokens, providing the LM with a semantically structured starting point for subsequent end-to-end fine-tuning of the full model for target downstream tasks.

We validate **GTI** on Generative Recommendation (GR) (Rajput et al., 2023; Zhai et al., 2024), a challenging and practically important application of vocabulary extension. GRs have attracted growing attention in both academia and industry (Ding et al., 2026; Han et al., 2025; Chen et al., 2025a; Deng et al., 2025), as they dramatically simplify retrieval by autoregressively generating item identifiers token-by-token from user interaction histories, replacing the expensive user-item inner products required by traditional dense-embedding meth-

ods (Koren et al., 2009; He et al., 2017, 2020; Wang et al., 2019). GRs can further exploit scaling-law behavior as model size and data increase (Han et al., 2025), offering a clear path to continued improvement. The GR setting is a particularly demanding testbed for grounded token initialization: large sets of new learnable Semantic-ID (SID) tokens must be incorporated into pretrained LMs, each encoding fine-grained item-level semantics and hierarchical codebook structure that should be properly grounded in the LM’s embedding space to support effective retrieval.

Contributions.

- 1. Diagnosis.** Through spectral and geometric analysis, we characterize the token-embedding misalignment caused by mean initialization: all new learnable tokens collapse into a degenerate, low-rank subspace that does not fully recover under subsequent fine-tuning. This motivates the *Grounded Token Initialization Hypothesis*: linguistically grounding new tokens before fine-tuning better enables the LM to leverage its pretrained knowledge for the new domain.
- 2. Methodology.** We introduce **GTI**, a simple and effective grounding stage that freezes the LM backbone and learns new token embeddings via paired linguistic supervision before standard fine-tuning, providing a semantically structured starting point for downstream adaptation.
- 3. Finding.** On generative recommendation

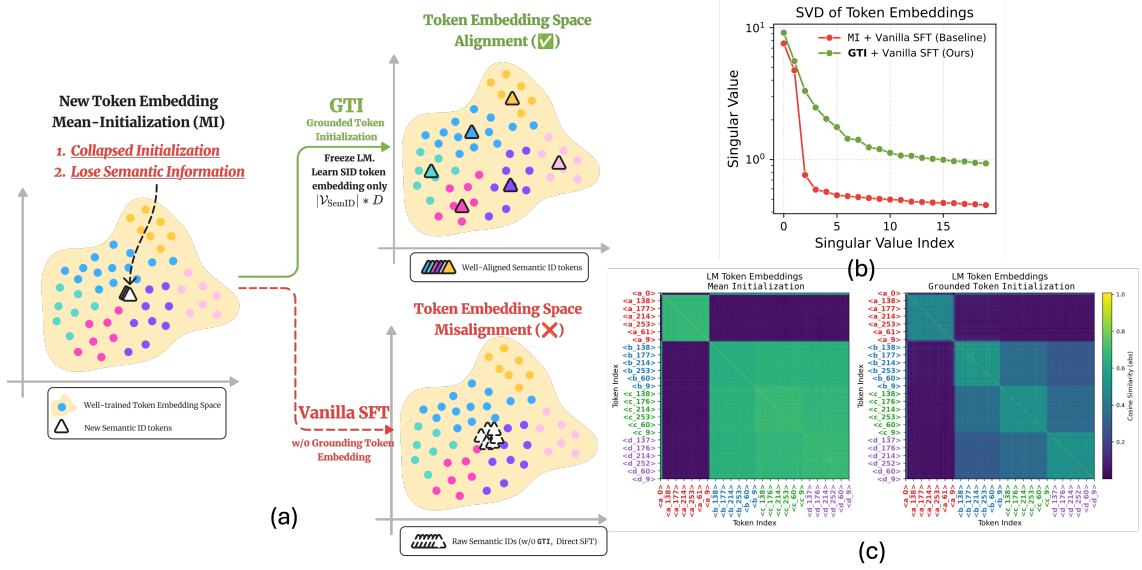


Figure 2: **Token-embedding collapse under mean initialization and the effect of grounding.** (a) *Left*: Mean initialization maps all SID tokens (white triangles) to a single point, collapsing inter-token distinctions. *Top-right*: GTI grounds SID tokens (colored triangles) into distinct regions by training only the $|\mathcal{V}_{\text{SID}}| \times d$ embedding parameters while freezing the backbone. *Bottom-right*: Fine-tuning without grounding does not fully resolve the collapse (see Figure 7). (b) & (c) GTI initialization yields higher effective rank and preserves blockwise hierarchical structure among SID tokens after downstream task supervised finetuning.

benchmarks, spanning industry-scale and public datasets, GTI consistently outperforms both direct supervised fine-tuning and LC-Rec (Zheng et al., 2024), an existing approach that jointly adapts the full model via auxiliary tasks. These results suggest that token initialization is a key bottleneck in vocabulary extension.

2 Token-Embedding Misalignment

We formalize the vocabulary extension problem in the context of generative retrieval, our primary application domain, while recognizing that vocabulary extension more broadly arises whenever pre-trained language models are adapted to newly introduced symbolic or domain-specific tokens. We then use spectral and geometric diagnostics to characterize a systematic token-embedding misalignment caused by standard initialization practices.

Generative Retrieval. We adopt the framework of Rajput et al. (2023). Each item $I_i \in \mathcal{I}$ has content features (title, description, etc.) that a pre-trained text encoder maps to a semantic embedding $\mathbf{z}_i \in \mathbb{R}^d$. An RQ-VAE (Lee et al., 2022) with L codebook levels of K entries discretizes $\mathbf{z}_i$ into a Semantic ID $(c_1, \dots, c_L)$, $c_l \in \{1, \dots, K\}$, via

recursive residual quantization:

$$\begin{aligned} \mathbf{r}_1 &:= \mathbf{z}_i, \quad c_l = \arg \min_k \|\mathbf{r}_l - \mathbf{q}_k^{(l)}\|_2, \\ \mathbf{r}_{l+1} &:= \mathbf{r}_l - \mathbf{q}_{c_l}^{(l)}, \quad l = 1, \dots, L. \end{aligned}$$

where $\{\mathbf{q}_k^{(l)}\}_{k=1}^K \subset \mathbb{R}^d$ is the level- l codebook. The $K \times L$ SID codes¹ are appended to the LM’s original vocabulary $\mathcal{V}_{\text{text}}$ as new tokens $\mathcal{V}_{\text{SID}}$. Given a context $\mathbf{x}$, either a user’s interaction history (retrieval) or a natural language query (search), the LM generates the target SID autoregressively:

$$P_\theta(c_1, \dots, c_L | \mathbf{x}) = \prod_{t=1}^L P_\theta(c_t | c_{<t}, \mathbf{x}).$$

Mean-of-Vocabulary Initialization. Standard practice initializes all novel token embeddings to the mean of the existing vocabulary embeddings (Hewitt, 2021):

$$\mathbf{e}_c := \frac{1}{|\mathcal{V}_{\text{text}}|} \sum_{v \in \mathcal{V}_{\text{text}}} \mathbf{e}_v, \quad \forall c \in \mathcal{V}_{\text{SID}}, \quad (1)$$

where $\mathbf{e}_v$ denotes the input embedding of token v .

¹SID tokens are labeled by level to encode codebook membership, e.g., $\langle \mathbf{a}_1 \rangle$, $\langle \mathbf{b}_1 \rangle$, $\langle \mathbf{c}_1 \rangle$...

Diagnosing the misalignment. Under mean-of-vocabulary initialization (Eq. 1), every new token receives an identical embedding, 1) collapsing all inter-token distinctions and 2) discarding the semantic structure each token should encode (Fig. 2, left). This heuristic is nonetheless widely adopted (Wolf et al., 2020) because it places new tokens on the pretrained manifold and yields a tighter KL-divergence upper bound on output probabilities compared with random initialization (Hewitt, 2021). Random initialization, conversely, assigns distinct vectors to each token but places them without coherent relation to the pretrained manifold, providing no linguistic prior for the model to build on. Pairwise cosine similarities among token embeddings (Fig. 5) confirm that mean initialization produces a near-uniform similarity block across all SID tokens, while random initialization yields unstructured noise.

We examine whether supervised fine-tuning recovers the structure lost under mean initialization. The pairwise similarity among SID embeddings (Fig. 2 (c) and Fig. 6 Left&Mid) and the singular-value decomposition of the SID embedding matrix $E_{\text{SID}} \in \mathbb{R}^{|\mathcal{V}_{\text{SID}}| \times d}$ after supervised fine-tuning from the mean-initialized state (Fig. 2 (b) and Fig. 7) reveals rapid spectral decay and low effective rank, confirming that supervised fine-tuning alone does not recover the inter-token structure lost at mean or random initialization. Taken together, these analyses show that neither strategy provides a suitable starting point: mean initialization places tokens on the pretrained manifold but erases discrimination, while random initialization preserves discrimination but lacks linguistic grounding.

Grounded Token Initialization (GTI) Hypothesis. These observations motivate our central hypothesis: **linguistically grounding novel tokens in an LM’s pretrained embedding space, before downstream fine-tuning, better enables the model to leverage its general-purpose knowledge for novel-token domains.** Rather than relying on fine-tuning alone to recover from a degenerate initialization, we propose inserting a simple and efficient grounding stage that learns new token embeddings via linguistic supervision with the backbone frozen, before proceeding to standard end-to-end fine-tuning. We operationalize this hypothesis in Section 3 and verify its effectiveness in Section 4.

3 GTI: Grounded Token Initialization Stage

The diagnosis in Section 2 motivates a straightforward modification to the standard training pipeline: before downstream fine-tuning, insert a grounding stage that freezes the LM backbone and only learns new token embeddings via paired linguistic supervision. This design builds on the established principle of training new token embeddings within a frozen LM (Hao et al., 2024; Nguyen et al., 2024). We term the resulting procedure **GTI**. Despite its simplicity, we show that this additional stage yields consistent improvements across multiple benchmarks, including both public and industry-scale datasets (Section 4), suggesting that token initialization is a key bottleneck in vocabulary extension.

Algorithm. Let $\mathcal{V} = \mathcal{V}_{\text{text}} \cup \mathcal{V}_{\text{new}}$ denote the extended vocabulary, where $\mathcal{V}_{\text{new}}$ are the newly added domain tokens. Given a pretrained autoregressive LM with input-embedding matrix $E \in \mathbb{R}^{|\mathcal{V}| \times d}$, we partition E into the pretrained rows E_{text} and the new rows $E_{\text{new}} \in \mathbb{R}^{|\mathcal{V}_{\text{new}}| \times d}$. Each domain entity is associated with a natural-language description x_i (e.g., title or definition) and a canonical new-token sequence $y_i = (c_{i,1}, \dots, c_{i,L})$. We instantiate **GTI** in the generative recommendation setting, where $\mathcal{V}_{\text{new}} = \mathcal{V}_{\text{SID}}$, x_i is an item title/description, and y_i is the corresponding SID sequence.

We construct a grounding corpus $\mathcal{D}_{\text{Grounded}} = \{(x_i, y_i)\}_{i=1}^n$ pairing each description with its new token sequence, along with reversed pairs $\{(y_i, x_i)\}$ that require the model to infer descriptions from new tokens². Using an instruction-style prompt template $\text{prompt}(x)$ (Listing as follows), we minimize the negative log-likelihood over E_{new} :

$$\min_{E_{\text{new}}} \sum_{(x,y) \in \mathcal{D}_{\text{Grounded}}} \sum_{t=1}^{|y|} \times [-\log P_{\theta}(y_t | y_{<t}, \text{prompt}(x))] \quad (2)$$

where θ denotes all LM parameters. During this stage, all parameters except E_{new} are held fixed, including E_{text} and the LM head, which shares weights with E via the standard tied-embedding parameterization. This weight tying means the grounding stage simultaneously shapes how the model reads and generates new tokens. After

²Bidirectional training encourages new token embeddings to encode semantics in both the input and output directions; see ablation in Section 4.3 and template details in Appendix A.2.

grounding, we retain the learned E_{new} as initialization and proceed with standard supervised fine-tuning of all model parameters θ . Implementation details are provided in Algorithm 1.

Item Title/Description $\rightarrow$ Semantic IDs
(Text $\rightarrow$ New Vocabulary Tokens)

```
<system>
You are a helpful assistant.
<user>
What item is called {{title}} and described
as {{description}}?
<assistant>
{{ITEM_SEMANTIC_ID}}
```

4 Experiments

We evaluate **GTI** within the highly demanding domain of generative recommendation. This domain serves as an ideal testbed for the initialization bottleneck, as it requires incorporating thousands of new Semantic-ID (SID) tokens into a pretrained language model. To empirically validate whether aligning these tokens with the model’s pre-existing linguistic geometry can prevent semantic collapse, we evaluate across two diverse environments: an industrial-scale candidate retrieval system and the public Vibrent benchmark (Borgersen, 2024).

4.1 Setup

Datasets. We evaluate across two distinct scales and domains: a large-scale internal dataset and a small public benchmark.

(1) Industrial candidate retrieval. This dataset consists of job requirement–candidate pairs from a world-leading recruitment platform. Each pair is categorized into three relevance levels (good, good&maybe, and not match) by an internal LLM evaluator according to how many job requirements a candidate satisfies. Due to data-sharing constraints, we report only relative performance gains over the SFT baseline for this dataset.

(2) Vibrent Clothes Rental. To validate generalizability, we adapt the public Vibrent Clothes Dataset (Borgersen, 2024) into a generative retrieval task, treating users as queries and clothing items as candidates based on historical transactions.

Baselines. To strictly isolate the initialization bottleneck, all methods share an identical Qwen3-0.6B backbone and RQ-VAE tokenization structure, differing only in how they introduce novel tokens.

(1) **Vanilla SFT:** New SID tokens are mean-initialized (Eq. 1), inducing a semantic collapse. The model relies entirely on downstream fine-tuning to disambiguate tokens from this degenerate starting point.

(2) **LC-Rec (Zheng et al., 2024):** A recent multi-task approach that begins from the same collapsed state but attempts to recover semantic structure by applying auxiliary natural language alignment objectives *during* fine-tuning.

(3) **GTI (Ours):** Using the grounding stage described in Section 3, we ground the new SID tokens into distinct, semantically meaningful regions of the frozen LM’s embedding space, providing a structurally rich starting point for the subsequent SFT procedure.

Evaluation Metrics. We measure retrieval accuracy using Top- K Precision, Recall, and NDCG. For the industrial dataset, we sample 200 jobs as evaluation queries (retrieving 200 candidates each). To comply with data-sharing constraints, we isolate the direct performance uplift of our grounding stage by reporting results strictly as a relative percentage gain over the standard SFT baseline, formulated as $(M_{\text{method}} - M_{\text{Baseline}})/M_{\text{Baseline}}$. For the public Vibrent dataset, we adopt the standard leave-one-out sequence splitting strategy (Kang and McAuley, 2018; Geng et al., 2023).

Implementation Details. Across both datasets, we employ Qwen3-0.6B as the backbone language model. Semantic IDs are constructed via RQ-VAE, following the formulation in Rajput et al. (2023). During the grounding stage and the subsequent supervised fine-tuning stage, we adapt Qwen3-0.6B via parameter-efficient finetuning (LoRA). All experiments use four NVIDIA H100 GPUs.

Industrial dataset. Candidate-level semantic representations are obtained by fine-tuning Mistral-E5 in a two-tower architecture with recruiter engagement signals, producing high-dimensional embeddings. The RQ-VAE uses $L = 3$ codebook levels with $K = 8,192$ codes per level.

Public dataset. Item-level semantic representations are derived using the off-the-shelf Qwen3-Embedding-0.6B encoder, producing high-dimensional embeddings. The RQ-VAE uses $L = 4$ codebook levels with $K = 256$ codes per level, and the diversity regularizer of Wang et al. (2024) to encourage balanced codebook utilization.

4.2 Overall Performance Analysis

Tables 1 & 2 and Figure 3 detail the overall performance on the industrial-scale dataset.

The effectiveness of GTI Initialization. Across all cutoffs, evaluation metrics, and relevance thresholds (Good Match and Good & Maybe Match), **GTI** outperforms both baselines. Under the strict Good Match criterion, **GTI** achieves **+21.63%** relative gain at P@5 over vanilla SFT, compared to +6.38% for LC-Rec, yielding an extra gain Δ of 15.25% attributable to the grounding stage. This pattern is consistent across evaluation settings: under Good & Maybe Match, **GTI** maintains a clear advantage

(+15.83% vs. +5.63% at P@5), and NDCG@5 exhibits the same trend (+17.88% vs. +6.94%). Sweeping the candidate pool size from 5 to 200 (Figure 3) further confirms that the improvement is robust across retrieval scales.

Evidence for the GTI hypothesis. The comparison between LC-Rec and **GTI** provides a controlled test of our hypothesis, as both methods introduce linguistic supervision for new tokens but differ in *when* it is applied: LC-Rec incorporates auxiliary language modeling objectives during fine-tuning while retaining mean initialization, whereas **GTI** addresses the initialization directly through a ground-

Table 1: Relative Precision@K gain (%) over SFT baseline on a real-world candidate retrieval dataset. Bold and underline denote the best result.

Methodology	Precision@K (Good Match)					Precision@K (Good & Maybe Match)				
	P@5	P@10	P@20	P@50	P@100	P@5	P@10	P@20	P@50	P@100
MI+Vanilla SFT (Baseline)	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
MI+Multi-task SFT (LC-Rec)	+6.38%	+5.20%	+3.87%	+3.00%	+3.47%	+5.63%	+5.35%	+2.98%	+3.32%	+3.05%
GTI+Multi-task SFT (Ours)	+21.63%	+13.59%	+8.16%	+6.35%	+4.25%	+15.83%	+10.89%	+5.74%	+5.87%	+4.10%
GTI: extra gain over LC-Rec (Δ)	+15.25%	+8.39%	+4.29%	+3.35%	+0.78%	+10.20%	+5.54%	+2.76%	+2.55%	+1.05%

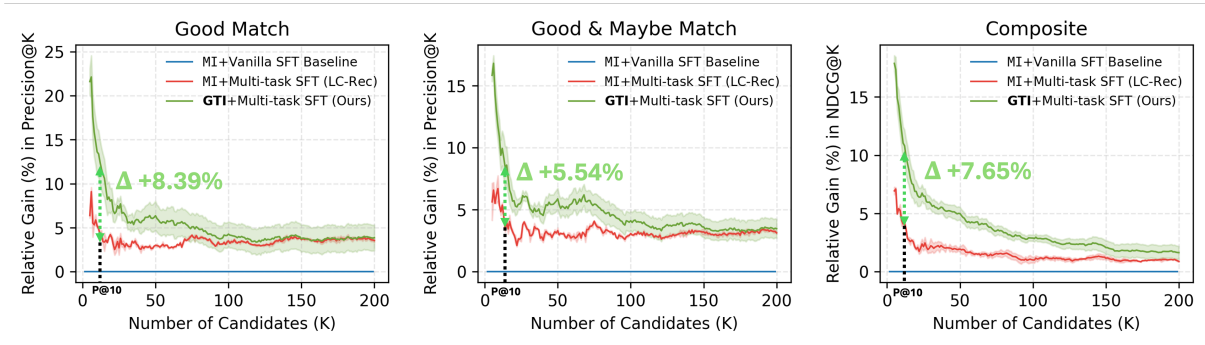


Figure 3: **Relative gain versus candidate pool size.** Left/Middle: Relative Precision@K gain under Good Match and Good & Maybe Match; Right: Relative NDCG@K gain (Composite). **GTI** consistently outperforms both baselines across all pool sizes, with the largest gains at small K . Shaded areas denote variability across runs.

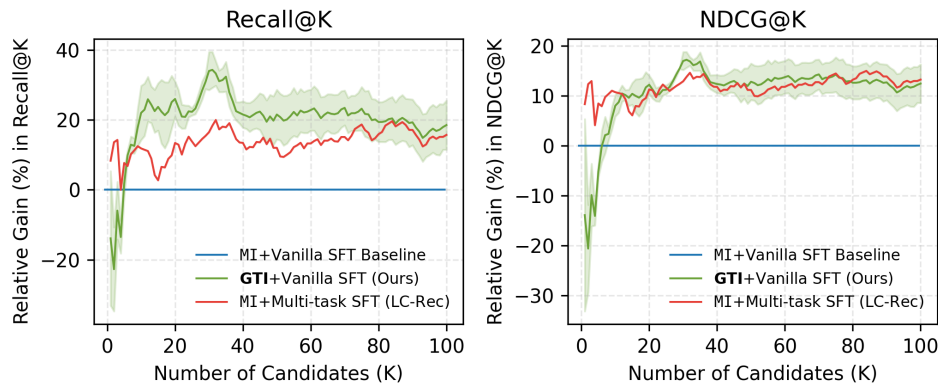


Figure 4: **Relative gain versus candidate pool size.** Left: Relative Recall@K gain; Right: Relative NDCG@K gain. Shaded areas denote variability across runs.

Table 2: Relative NDCG@K (Composite) gain (%) over SFT baseline on a real-world candidate retrieval dataset. Bold and underline denote the best result.

Methodology	NDCG@K (Composite)				
	N@5	N@10	N@20	N@50	N@100
MI+Vanilla SFT (Baseline)	0.00%	0.00%	0.00%	0.00%	0.00%
MI+Multi-task SFT (LC-Rec)	+6.94%	+4.38%	+1.94%	+1.95%	+1.01%
GTI+Multi-task SFT (Ours)	+17.88%	+12.03%	+6.90%	+4.99%	+2.89%
GTI: extra gain over LC-Rec (Δ)	+10.94%	+7.65%	+4.96%	+3.04%	+1.88%

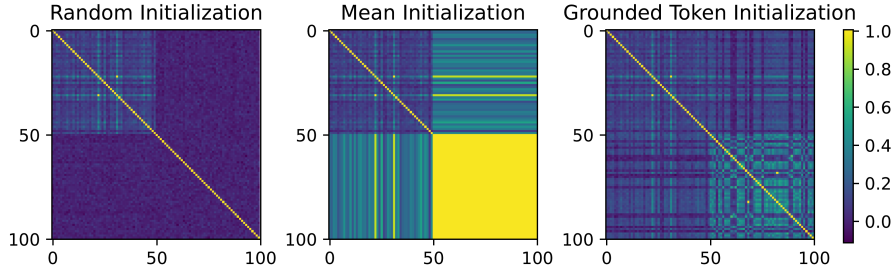


Figure 5: **Pairwise cosine-similarity matrices under three initialization strategies.** Each matrix shows similarities between 50 pretrained tokens (upper-left block) and 50 SID tokens (bottom-right block); 50 tokens are randomly sampled from each vocabulary. Random initialization (left) yields noninformative SID embeddings. Mean initialization (middle) collapses SID tokens into a near-uniform block. **GTI** (right) produces differentiated intra-SID structure with meaningful affinities to pretrained tokens.

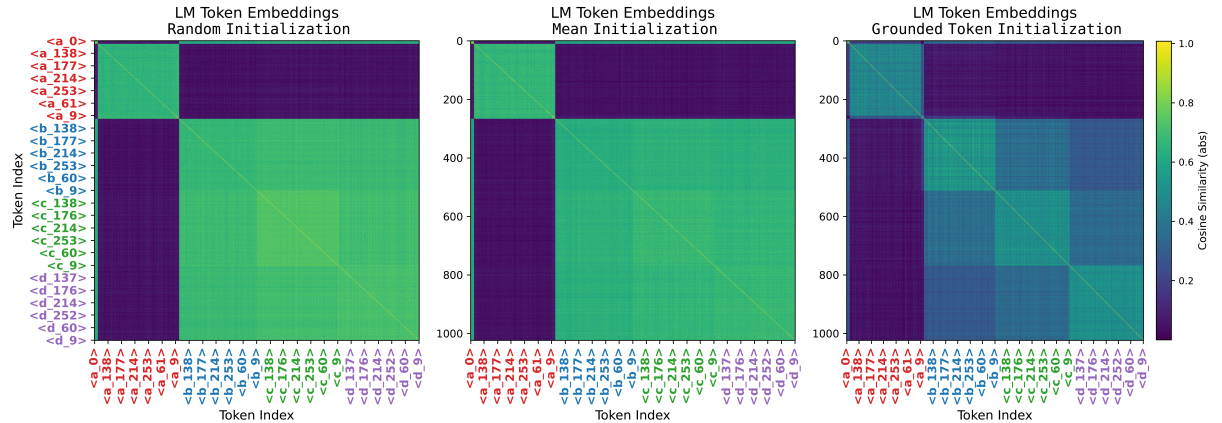


Figure 6: **Pairwise SID similarity after fine-tuning (public dataset).** We visualize the pairwise cosine similarity matrix of SID embeddings at the fine-tuned checkpoint. **GTI** is the only initialization strategy that preserves a clear blockwise hierarchical semantics among SID tokens, suggesting improved preservation of semantic geometry. By contrast, mean and random initialization produce flat or noisy similarity patterns even after SFT stage.

ing stage that precedes fine-tuning. The consistent performance gap (extra gain Δ) between the two methods, despite sharing the same downstream

SFT procedure, suggests that grounding new tokens before fine-tuning provides a more effective starting point than relying on auxiliary objectives

Table 3: Relative Recall@K and NDCG@K (%) over SFT baseline on Vibrent Dataset.

Methodology	Recall@K					NDCG@K				
	R@5	R@10	R@20	R@50	R@100	N@5	N@10	N@20	N@50	N@100
MI+Vanilla SFT (Baseline)	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
MI+Multi-task SFT (LC-Rec)	+7.69%	+11.86%	+13.41%	+12.03%	+15.73%	+8.47%	+10.74%	+11.30%	+11.18%	+13.26%
GTI+Vanilla SFT (Ours)	+1.71%	+22.03%	+26.02%	+21.55%	+18.54%	-5.19%	+8.02%	+12.23%	+12.83%	+12.46%

alone, consistent with the **GTI** hypothesis.

Controlled comparison on public dataset. To disentangle the effect of **GTI** from that of multi-task adaptation and assess the generalization of our method beyond the proprietary dataset, we compare **GTI**+Vanilla SFT against LC-Rec (Multi-task SFT) on the public Vibrent dataset (Table 3 and Figure 4). Even without auxiliary objectives during fine-tuning, **GTI** achieves substantially higher Recall at $K \geq 10$ (e.g., **+26.02%** vs. +13.41% at Recall@20) and comparable NDCG, indicating that the grounding stage alone accounts for a large portion of the downstream improvement.

4.3 Further Analysis

The preceding results establish *that* grounded initialization improves downstream performance; we now investigate *why*. We use spectral and geometric diagnostics on the SID embedding subspace, both at initialization and after fine-tuning. These analyses provide direct evidence to the Grounded Token Initialization Hypothesis (Section 2).

Grounded initialization produces differentiated embedding geometry. Figure 5 visualizes pairwise cosine similarities among pretrained vocabulary tokens and SID tokens under three initialization strategies. Random initialization avoids uniformity but yields unstructured noise with no coherent affinity to the pretrained manifold. Mean initialization produces a uniform SID block, confirming the collapse diagnosed in Section 2. In contrast, **GTI** produces rich, differentiated structure within the SID block together with coherent cross-block affinities to relevant lexical tokens.

Grounded structure persists through fine-tuning. We next examine whether the structure induced by grounding persists through fine-tuning. (1) Pairwise cosine similarities among SID embeddings after fine-tuning on the public dataset (Figure 6) show that only the **GTI**-initialized model preserves the blockwise hierarchical structure encoded by the RQ-VAE; mean and random initialization produce flat or noisy similarity patterns. (2) The singular-value spectrum of $E_{\text{SID}} \in \mathbb{R}^{|\mathcal{V}_{\text{SID}}| \times d}$ after fine-tuning on the industrial dataset (Figure 7a) shows that mean initialization leads to rapid spectral decay and low effective rank, while grounded initialization yields slower decay and higher effective rank, indicating a non-degenerate subspace with multiple active directions along which items

differ (see Appendix A.4 for extended SVD analysis of the industrial dataset). (3) Representational similarity analysis (RSA) between the learned SID embeddings and the ground-truth RQ-VAE codebook vectors (Figure 7b) shows that **GTI**-initialized embeddings better preserve the original semantic structure through training. Taken together, these results suggest that the grounding stage seeds embedding structure that persists through fine-tuning, corroborating the downstream performance gains.

5 Related Work

Vocabulary Extension in Language Models.

Extending a pretrained LM’s vocabulary with new tokens is a recurring challenge. Standard approaches initialize new embeddings at the vocabulary mean (Hewitt, 2021) or randomly, then rely on fine-tuning. ToolkenGPT (Hao et al., 2024) and Yo’LLaVA (Nguyen et al., 2024) show that training only new token embeddings against a frozen LM can be effective for tool invocation and visual concept grounding, respectively. **GTI** reframes this mechanism as an *initialization* strategy: by grounding new tokens before fine-tuning, the learned structure serves as a starting point that benefits arbitrary downstream tasks, rather than being tied to a specific end use.

Generative Recommendation. We adopt generative recommendation as our primary evaluation domain, as it requires injecting thousands of novel tokens into a pretrained LM, making it a demanding testbed for vocabulary extension. This paradigm frames retrieval as autoregressive decoding of Semantic IDs (SIDs) discretized via RQ-VAE (van den Oord et al., 2018; Lee et al., 2022; Rajput et al., 2023; Zheng et al., 2024). We provide an extended discussion in Appendix A.5.

6 Conclusion

Through spectral and geometric diagnostics, we show that mean-of-vocabulary initialization collapses new tokens into a degenerate subspace that fine-tuning does not fully recover. Motivated by this diagnosis, we propose a **simple yet effective** grounding stage, **GTI**, that learns only the new token embeddings via paired linguistic supervision before standard fine-tuning. On generative recommendation benchmarks spanning industrial-scale and public datasets, **GTI** consistently outperforms both mean initialization and auxiliary-task adaptation, with further analyses confirming

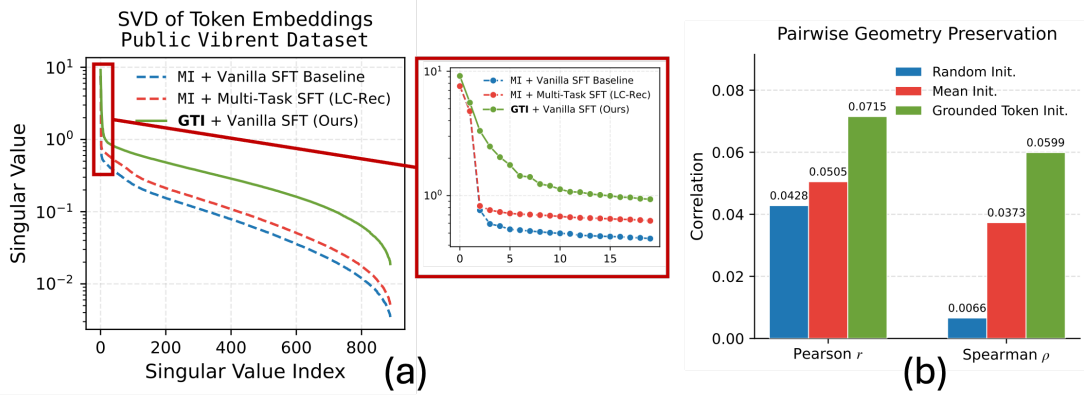


Figure 7: (a) **Singular-Value Spectra** of SID embedding matrix after SFT: **GTI** initialization yields slower spectral decay and higher effective rank than mean initialization. (b) **Representational Similarity Analysis (RSA)** of SID embeddings after SFT: We compare the pairwise geometry of the ground-truth RQ-VAE codebook vectors and the learned SID embeddings using Pearson r and Spearman ρ . **GTI** initialization achieves the highest correlation under both metrics, indicating better preservation of semantic structure among SID embeddings than mean or random init.

that grounded structure persists through fine-tuning. These findings support the Grounded Token Initialization Hypothesis. While our experiments focus on generative recommendation, **GTI** makes no task-specific assumptions: vocabulary extension broadly arises whenever pretrained language models are adapted to newly introduced symbolic or domain-specific tokens. Testing the generality of **GTI** in broader vocabulary-extension settings beyond recommendation is therefore an important direction for future work.

7 Limitations

Despite the empirical effectiveness of **GTI**, several limitations remain.

First, while we provide comprehensive spectral and geometric diagnostics of the token-embedding collapse induced by mean-of-vocabulary initialization, our analysis is primarily empirical. A better theoretical understanding is still needed to characterize when and why grounded initialization improves downstream fine-tuning performance. In particular, an important open question is the apparent "irrecoverability" of the collapsed initialization: why standard supervised fine-tuning does not reliably move newly introduced tokens out of the degenerate subspace induced by mean init.

Second, our experiments do not yet cover a broad range of larger model families. Due to time and computational constraints, we focus on the model scales available in our current experimental setting. Although the observed gains suggest that **GTI** is beneficial under these settings, further evaluation

on larger backbone models and diverse pretrained model families is necessary to establish its scalability and robustness.

Third, although **GTI** is empirically lightweight compared with the subsequent supervised fine-tuning stage, we do not provide a systematic analysis of its computational cost.

References

- Karl Audun Kagnes Borgersen. 2024. Vibrent clothes rental dataset. <https://www.kaggle.com/datasets/kaborg15/vibrent-clothes-rental-dataset>. Kaggle dataset, accessed March 22, 2026.
- Rui Cai, Chao Wang, Qianyi Cai, Dazhong Shen, and Hui Xiong. 2025. Boosting knowledge graph-based recommendations through confidence-aware augmentation with large language models. *arXiv preprint arXiv:2502.03715*.
- Ben Chen, Xian Guo, Siyuan Wang, Zihan Liang, Yue Lv, Yufei Ma, Xinlong Xiao, Bowen Xue, Xuxin Zhang, Ying Yang, Huangyu Dai, Xing Xu, Tong Zhao, Mingcan Peng, Xiaoyang Zheng, Chao Wang, Qihang Zhao, Zhixin Zhai, Yang Zhao, and 9 others. 2025a. *Onesearch: A preliminary exploration of the unified end-to-end generative framework for e-commerce search*. *Preprint*, arXiv:2509.03236.
- Daiwei Chen, Yi Chen, Aniket Rege, Zhi Wang, and Ramya Korlakai Vinayak. 2025b. *Pal: Sample-efficient personalized reward modeling for pluralistic alignment*. In *The Thirteenth International Conference on Learning Representations*.
- Yashar Deldjoo, Zhankui He, Julian McAuley, Anton Korikov, Scott Sanner, Arnau Ramisa, René Vidal, Maheswaran Sathiamoorthy, Atoosa Kasirzadeh,

- and Silvia Milano. 2024. [A review of modern recommender systems using generative models \(genrecsys\)](#). *Preprint*, arXiv:2404.00579.
- Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. 2025. [Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment](#). *arXiv preprint 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. 2026. [How well does generative recommendation generalize?](#) *Preprint*, arXiv:2603.19809.
- Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*.
- Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2023. [Recommendation as language processing \(rlp\): A unified pretrain, personalized prompt & predict paradigm \(p5\)](#). *Preprint*, arXiv:2203.13366.
- Ruidong Han, Bin Yin, Shangyu Chen, He Jiang, Fei Jiang, Xiang Li, Chi Ma, Mincong Huang, Xiaoguang Li, Chunzhen Jing, and 1 others. 2025. [Mtgr: Industrial-scale generative recommendation framework in meituan](#). *arXiv preprint arXiv:2505.18654*.
- Shibo Hao, Tianyang Liu, Zhen Wang, and Zhiting Hu. 2024. [Toolkengpt: Augmenting frozen language models with massive tools via tool embeddings](#). *Preprint*, arXiv:2305.11554.
- Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. [Lightgcn: Simplifying and powering graph convolution network for recommendation](#). *Preprint*, arXiv:2002.02126.
- Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. [Neural collaborative filtering](#). In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*, pages 173–182.
- John Hewitt. 2021. Initializing new word embeddings for pretrained language models. <https://nlp.stanford.edu/~johnhew/vocab-expansion.html>.
- HuggingFace. 2025. Trl documentation: Quickstart. <https://huggingface.co/docs/trl/en/quickstart>. Accessed: 2025-09-23.
- Ruijie Jiang, Thuan Nguyen, Shuchin Aeron, and Prakash Ishwar. 2024. Hard-negative sampling for contrastive learning: Optimal representation geometry and neural-vs dimensional-collapse. *Transactions on Machine Learning Research*.
- Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. 2021. Understanding dimensional collapse in contrastive self-supervised learning. *arXiv preprint arXiv:2110.09348*.
- Wang-Cheng Kang and Julian McAuley. 2018. [Self-attentive sequential recommendation](#). In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 197–206.
- Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. [Matrix factorization techniques for recommender systems](#). *Computer*, 42(8):30–37.
- Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. 2022. [Autoregressive image generation using residual quantization](#). *Preprint*, arXiv:2203.01941.
- Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, volume 27.
- Thao Nguyen, Haotian Liu, Yuheng Li, Mu Cai, Utkarsh Ojha, and Yong Jae Lee. 2024. [Yo'llava: Your personalized language and vision assistant](#). *Preprint*, arXiv:2406.09400.
- Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan H. Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Maheswaran Sathiamoorthy. 2023. [Recommender systems with generative retrieval](#). *Preprint*, arXiv:2305.05065.
- Aniket Rege, Aditya Kusupati, Sharan Ranjit S, Alan Fan, Qingqing Cao, Sham Kakade, Prateek Jain, and Ali Farhadi. 2023. [Adanns: A framework for adaptive semantic search](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 76311–76335. Curran Associates, Inc.
- Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, and 1 others. 2022. Transformer memory as a differentiable search index. In *Advances in Neural Information Processing Systems*.
- Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2018. [Neural discrete representation learning](#). *Preprint*, arXiv:1711.00937.
- Wenjie Wang, Honghui Bao, Xinyu Lin, Jizhi Zhang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. 2024. [Learnable item tokenization for generative recommendation](#). *Preprint*, arXiv:2405.07314.
- Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. [Neural graph collaborative filtering](#). In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19)*, pages 165–174.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Jiayuan He, Yinghai Lu, and Yu Shi. 2024. [Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 58484–58509. PMLR.

Haichao Zhang and Yun Fu. 2025. [VQToken: Neural discrete token representation learning for extreme token reduction in video large language models](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Haichao Zhang, Yao Lu, Lichen Wang, Yunzhe Li, Daiwei Chen, Yunpeng Xu, and Yun Fu. 2025. Linked-out: Linking world knowledge representation out of video llm for next-generation video recommendation. *arXiv preprint arXiv:2512.16891*.

Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. 2024. [Adapting large language models by integrating collaborative semantics for recommendation](#). *Preprint*, arXiv:2311.09049.

Guorui Zhou, Jiaxin Deng, Jinghao Zhang, Kuo Cai, Lejian Ren, Qiang Luo, Qianqian Wang, Qigen Hu, Rui Huang, Shiyao Wang, Weifeng Ding, Wuchao Li, Xinchun Luo, Xingmei Wang, Zexuan Cheng, Zixing Zhang, Bin Zhang, Boxuan Wang, Chaoyi Ma, and 46 others. 2025. [Onerec technical report](#). *Preprint*, arXiv:2506.13695.

A Appendix

A.1 Datasets

A.1.1 Retrieval Dataset

Industrial Candidate Retrieval Dataset. The industrial-scale candidate retrieval dataset³ consists of job requirement–candidate pairs collected in 2025 from a world-leading professional networking platform with global user coverage and evaluated by our internal LLM judge. According to product policy, which measures how many job requirements a candidate satisfies, each pair is assigned to one of three relevance levels: good match, good&maybe match, and not match. We use the good match pairs for supervised fine-tuning (SFT).

The member profile dataset contains profiles of users who provide at least one of the following attributes: geographic location, job positions, education history, or skill information.

Vibrent Dataset. The [Vibrent Clothes Rental Dataset](#) is a publicly available dataset from Kaggle. To complement our industrial dataset with a publicly available benchmark, we also evaluate our method on it. The dataset contains anonymized user–item rental transactions from a clothing rental platform. We construct a candidate retrieval task by treating users as queries and clothing items as candidates, where observed rental interactions are considered positive relevance signals, and non-interacted items are treated as negatives during training.

A.2 Prompt Templates

A.2.1 Prompt Template: Auxiliary Task (Item Title/Description $\leftrightarrow$ New Vocabulary Tokens)

Item Title/Description $\rightarrow$ Semantic IDs⁴ (Title $\rightarrow$ New Vocabulary Tokens)

```
<system>
You are a helpful assistant.
<user>
Which item has the title: {{title}}?
<assistant>
{{ITEM_SEMANTIC_ID}}
```

Item Title/Description $\rightarrow$ Semantic IDs (Description $\rightarrow$ New Vocabulary Tokens)

```
<system>
You are a helpful assistant.
<user>
Can you tell me what item is described as {{description}}?
<assistant>
{{ITEM_SEMANTIC_ID}}
```

Item Title/Description $\rightarrow$ Semantic IDs (Title+Description $\rightarrow$ New Vocabulary Tokens)

```
<system>
You are a helpful assistant.
<user>
What item is called {{title}} and described as {{description}}?
<assistant>
{{ITEM_SEMANTIC_ID}}
```

Semantic IDs $\rightarrow$ Item Title/Description (New Vocabulary Tokens $\rightarrow$ Title)

```
<system>
You are a helpful assistant.
<user>
Could you please tell me what item {{ITEM_SEMANTIC_ID}} is called?
<assistant>
{{title}}
```

Semantic IDs $\rightarrow$ Item Title/Description (New Vocabulary Tokens $\rightarrow$ Description)

```
<system>
You are a helpful assistant.
<user>
Briefly describe item {{ITEM_SEMANTIC_ID}}.
<assistant>
{{description}}
```

Semantic IDs $\rightarrow$ Item Title/Description (New Vocabulary Tokens $\rightarrow$ Title+Description)

```
<system>
You are a helpful assistant.
<user>
What is the title and description of item {{ITEM_SEMANTIC_ID}}?
<assistant>
{{title}}\n\n{{description}}
```

³Data access and use complied with internal privacy and security frameworks. Member profile attributes were processed in accordance with applicable member controls and visibility settings, and analyses were conducted on de-identified datasets with results reported in aggregate only.

⁴Most of Item Title/Description $\leftrightarrow$ Semantic IDs prompts and retrieval prompts are adapted from (Zheng et al., 2024).

A.2.2 Prompt Template: Search Query Task

Candidate Description → Semantic Id Alignment Prompt⁵

```
<user>
Instruction: Predict the candidate SID.
Input:
{{candidate headline}}
{{candidate profile description}}
{{candidate job title}}
{{candidate education}}
{{candidate employment history}}
{{candidate location}}
{{candidate skills}}
{{candidate company}}
<assistant>
{{CANDIDATE SEMANTIC_ID}}
```

Semantic Id → Candidate Description Alignment Prompt

```
<user>
Instruction: Recover the candidate text.
Input: {{CANDIDATE SEMANTIC_ID}}
<assistant>
{{candidate headline}}
{{candidate profile description}}
{{candidate job title}}
{{candidate education}}
{{candidate employment history}}
{{candidate location}}
{{candidate skills}}
{{candidate company}}
```

Search Query → Candidate Semantic Id Alignment Prompt

```
<user>
Instruction: Predict the relevant candidate
SID that matches job requirements as
much as possible. If there are no
candidates that meet the criteria,
return the candidate SID that
prioritizes the matching required
requirements then matching the preferred
requirements. Input:
{{required requirements}}
{{preferred requirements}}
<assistant>
{{ITEM SEMANTIC_ID}}
```

A.2.3 Prompt Template: Retrieval Task

Retrieval Prompt (Template 1)

```
<system>
You are a helpful assistant.
<user>
The user has interacted with items {{inters}}
in chronological order. Can you
predict the next possible item that the
```

```
user may expect?
<assistant>
{{ITEM SEMANTIC_ID}}
```

Retrieval Prompt (Template 2)

```
<system>
You are a helpful assistant.
<user>
Based on the items that the user has
interacted with: {{inters}}, can you
determine what item would be recommended
to the user next?
<assistant>
{{ITEM SEMANTIC_ID}}
```

Retrieval Prompt (Template 3)

```
<system>
You are a helpful assistant.
<user>
Here is the item interaction history of the
user: {{inters}}, what to recommend to
the user next?
<assistant>
{{ITEM SEMANTIC_ID}}
```

A.3 Implementation Details

We utilize the pre-trained Qwen3-Embedding-0.6B encoder to extract semantic representations for items. The encoder processes item metadata including titles and descriptions to generate 1024-dimensional dense vectors that capture semantic similarities between items. We process text features of products by concatenating them as: [TITLE] [DESCRIPTION]. We set the maximum input sequence length as 2048. The final outputs are dense semantic embeddings: $z_i \in \mathbb{R}^{1024}$ for item i .

Our Residual Quantized Variational Autoencoder (RQ-VAE) follows the TIGER (Rajput et al., 2023) framework with carefully designed architectural specifications to ensure effective quantization of semantic representations. The encoder architecture consists of a 3-layer Multi-Layer Perceptron (MLP) with hidden dimensions of [1024, 512, 256], utilizing ReLU activation functions and applying a dropout rate of 0.1 between layers. The residual quantization mechanism employs four codebook layers, each containing 256 entries with 32-dimensional codes. This hierarchical quantization approach enables fine-grained representation of semantic information while maintaining discrete tokenization properties essential for language model integration. We trained the model for 20,000 epochs to achieve a high codebook utilization rate and minimize collision rates. To fur-

⁵For brevity, we illustrate only three representative prompting templates.

ther prevent collisions where multiple items map to identical sequences of semantic IDs, we employed the Sinkhorn-Knopp trick used by LC-Rec (Zheng et al., 2024), which ensures uniform distribution of item semantics across codebook embeddings in the final layer.

The base language model employs Qwen3-0.6B with hidden dimension of 1024. The model architecture comprises 28 transformer layers supporting a maximum context length of 32,768 tokens. This configuration provides sufficient capacity for processing sequential recommendation tasks while maintaining computational efficiency. Parameter-efficient fine-tuning is implemented through Quantized Low-Rank Adaptation (QLoRA) with a rank of 8 and alpha value of 32. The LoRA adaptation applies a dropout rate of 0.05 and targets key projection matrices including q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, and down_proj. We also set LoRA modules to be saved as embed_tokens and lm_head, so that only the embedding layer and the language modeling head are preserved during training while other modules can remain frozen. This configuration enables efficient adaptation while preserving pre-trained knowledge.

We implement the token-embedding grounding stage of GTI by extending the Hugging Face TRL (HuggingFace, 2025) SFTTrainer to update only the Semantic-ID embedding matrix while freezing the LM backbone; the trainer consumes paired (title/description, SemID) examples and optimizes the embeddings as outlined in the pseudo code below. Unless otherwise stated, we train for 10 epochs with a learning rate of 1e-3 and a batch size 16.

A.4 Analysis Details

Representation Similarity Analysis (RSA). To quantitatively measure whether the learned representations preserves the semantic structure of SID new vocabulary tokens, we perform representational similarity analysis. Given the well-trained RQ-VAE codebooks, which encode the compressed representation of SID new vocabulary tokens, we define the oracle semantic embeddings as $X = \{x_1, \dots, x_n\}, x_i \in \mathbf{R}^{32}$. And let the corresponding learned token embeddings from language model as $\hat{X} = \{\hat{x}_1, \dots, \hat{x}_n\}, x_i \in \mathbf{R}^d$, where d depends on the language model dimensionality. We construct pairwise token similarity matrices

$S_X, S_{\hat{X}} \in \mathbf{R}^{n \times n}$, where:

$$(S_X)_{i,j} = \cos(x_i, x_j),$$

$$(S_{\hat{X}})_{i,j} = \cos(\hat{x}_i, \hat{x}_j).$$

We then vectorize the upper-triangular entries of S_X and $S_{\hat{X}}$ and compute their correlation (We implement both Spearman correlation and Pearson correlation to capture complementary aspects of representational alignment). This yields an RSA score that quantifies the extent to which the learned representation space preserves the pairwise semantic relations of the oracle space. Since RSA compares representational geometry rather than coordinates directly, it is well suited to our setting where the oracle and learned embeddings live in different ambient dimensions (32 vs. d).

Extended SVD analysis of the industrial dataset.

The slower spectral decay and higher effective rank observed with GTI initialization suggest that this method preserves a more expressive and diverse feature space throughout the SFT process, preventing the dimensional collapse often associated with mean initialization.

A.5 Full Related Work

RQ-VAE and Semantic IDs. Vector-quantized autoencoders (van den Oord et al., 2018; Zhang and Fu, 2025) learn discrete item representations by mapping continuous embeddings to codebook entries. Residual Quantized VAEs (RQ-VAE) (Lee et al., 2022) extend this with a hierarchy of residual codebooks, producing multi-level Semantic IDs (SIDs) that capture progressively finer semantic distinctions. Unlike conventional item IDs, SIDs carry compositional structure amenable to autoregressive generation, making them a standard component in generative recommendation (Rajput et al., 2023; Zheng et al., 2024; Han et al., 2025). Crucially, each codebook entry becomes a new token in the LM vocabulary, and how these tokens are initialized is precisely the bottleneck our work addresses.

Generative Recommendation. Generative retrieval reframes recommendation as autoregressive decoding of item identifiers rather than nearest-neighbor search in embedding space (Rege et al., 2023; Chen et al., 2025b). TIGER (Rajput et al., 2023) introduced RQ-VAE-learned SIDs as generation targets, and LC-Rec (Zheng et al., 2024) added auxiliary linguistic objectives during fine-tuning to improve SID representations. Several

Algorithm 1: GTI Grounding Stage

Input: Pretrained model $\mathcal{M}$ with embedding matrix $E \in \mathbb{R}^{V \times d}$; new token indices $\mathcal{T} \subseteq \{0, \dots, V-1\}$; paired corpus $\mathcal{D} = \{(\text{text}_j, \text{token}_j)\}$
Output: Model $\mathcal{M}$ with grounded embeddings for tokens in $\mathcal{T}$
// Setup: freeze all parameters except new token embeddings
Freeze all parameters of $\mathcal{M}$
Construct binary mask $\mathbf{m} \in \{0, 1\}^V$ where $m_i = 1$ iff $i \in \mathcal{T}$
 $\mathbf{M} \leftarrow \mathbf{m} \otimes \mathbf{1}_d$ // Broadcast to $\mathbb{R}^{V \times d}$
// Training: update only new token embeddings via masked gradients
for each batch $\mathcal{B} \subset \mathcal{D}$ **do**
 $\mathcal{L} \leftarrow \text{LM_Loss}(\mathcal{M}, \mathcal{B})$ // Forward pass
 $\nabla E \leftarrow \nabla_E \mathcal{L}$ // Compute gradients
 $E \leftarrow E - \eta \cdot (\nabla E \odot \mathbf{M})$ // Update only new token embeddings
end

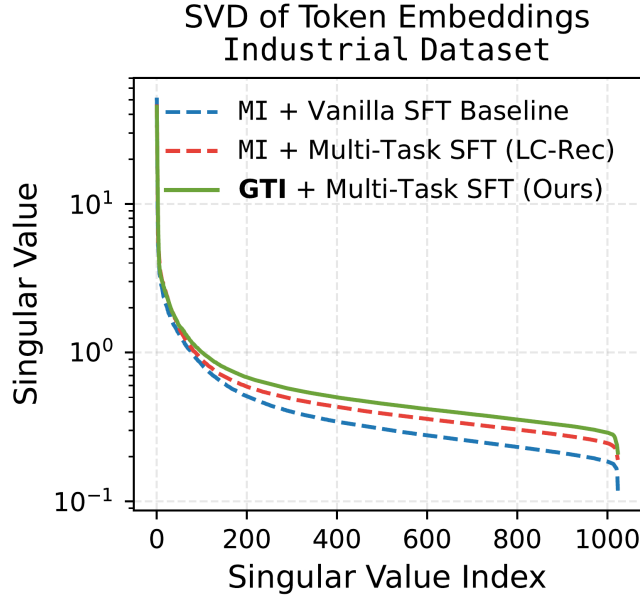


Figure 8: Singular-Value Spectra of SID embedding matrix after SFT for Industrial dataset.

systems have demonstrated industrial-scale deployment: MTGR (Han et al., 2025) integrates generative retrieval with DLRM cross-feature signals; OneSearch (Chen et al., 2025a) combines keyword-

Table 4: **Additional retrieval results on the Vibrent dataset.** We report Recall@K and NDCG@K for Baseline (MI+Vanilla SFT), LC-Rec (MI+Multi-task SFT), and our method GTI+Vanilla SFT. GTI+Vanilla SFT achieves the best performance on most Recall@K metrics and remains competitive on NDCG@K, further supporting the effectiveness of grounded token initialization for generative retrieval.

Methodology	Recall@K					NDCG@K				
	@5	@10	@20	@50	@100	@5	@10	@20	@50	@100
MI+Vanilla SFT (Baseline)	0.0226	0.0342	0.0475	0.0771	0.1031	0.0150	0.0188	0.0222	0.0280	0.0322
MI+Multi-task SFT (LC-Rec)	0.0243	0.0382	0.0539	0.0863	0.1194	0.0163	0.0208	0.0247	0.0311	0.0365
GTI+Vanilla SFT (Ours)	0.0230	0.0417	0.0599	0.0937	0.1222	0.0143	0.0203	0.0249	0.0316	0.0362

enhanced quantization with preference-aware rewards; and OneRec (Deng et al., 2025; Zhou et al., 2025) unifies retrieval and ranking via session-wise generation. Complementary directions include LLM-driven knowledge-graph recommenders (Cai et al., 2025) and MLLM-based world-knowledge integration (Zhang et al., 2025). All of these systems must inject novel tokens into a pretrained LM; our work addresses a step that is upstream of and complementary to their contributions, namely how those tokens should be initialized.

Connection to Dimensional Collapse. The initialization collapse we diagnose is related to *dimensional collapse* in contrastive and self-supervised learning (Jing et al., 2021; Jiang et al., 2024), where learned representations are restricted to a low-dimensional subspace, eliminating fine-grained distinctions (Figure 2). Mean-of-vocabulary initialization induces a similar effect: all new tokens start at the same point, forming a rank-deficient configuration. Jiang et al. (2024) show that appropriate initialization can mitigate dimensional collapse in contrastive learning, which parallels our finding that grounding new tokens before fine-tuning preserves a higher-rank, more differentiated embedding subspace.