PLUM: Adapting Pre-trained Language Models for Industrial-scale Generative Recommendations (Google, 2025)

Challenges

- **Scaling limits of embedding-table-based retrieval.** The paper frames industrial recommenders as dominated by large embedding models (LEMs), which scale primarily by enlarging embedding tables and thus face memory/compute bottlenecks and diminishing returns. The authors emphasize that this approach "is in contrast to the inherent sequence modeling capabilities and vast world knowledge" of LLMs (Introduction).
- **Need for compact, semantically meaningful IDs.** Generative retrieval models rely on Semantic IDs (SIDs) that compress items into discrete token sequences. The paper notes that retrieval quality is "fundamentally dependent" on SID quality (Section 2), and SID generation must encode multi-modal content and collaborative signals rather than single-modality embeddings.
- **Aligning LLMs to recommendation signals.** LLMs are not natively aligned to user behavior signals or industrial item vocabularies. PLUM must teach the model a new modality (SIDs) and align it with language, while preserving scaling benefits and deployment constraints.
- **Industrial constraints (latency/throughput).** The system must operate under YouTube-scale retrieval traffic. The paper repeatedly stresses production constraints and reports live experiments that require practical feasibility beyond offline metrics.

Key Initiatives (Core Contributions)

- **PLUM framework for LLM-based generative retrieval.** "We introduce PLUM" as an end-to-end framework to adapt pre-trained LLMs for industrial recommendation retrieval, replacing embedding-table heavy architectures with SID generation and autoregressive retrieval (Abstract + Sec. 2).
- **SIDv2: improved semantic ID generation.** The paper introduces "a set of new techniques (referred to as SID-v2)" that improve SID quality. These include multi-modal fusion, hierarchical codebooks, and co-occurrence regularization, with the explicit goal of producing SIDs that better match user behavior co-occurrence (Sec. 2.1).
- **Continued Pre-Training (CPT).** The model is "further pre-trained on a mixture of domain text data" (Abstract) to align SIDs with language tokens and leverage LLM capabilities in recommendation context. This CPT stage is a core alignment step before fine-tuning.
- **SFT for generative retrieval.** The retrieval model is trained to "autoregressively generate the SIDs of next items" (Sec. 2.3), enabling a direct generative retrieval formulation instead of embedding lookup.
- **Scaling study at iso-FLOPs.** The paper presents "a scaling study for the model's retrieval performance" (Abstract) with MoE variants and synchronized data/model scaling, emphasizing compute-optimal training patterns.

Methodology

Overall Pipeline

1. **SID generation (SIDv2).** Items are compressed into hierarchical SID token sequences using a semantic ID model (RQ-VAE style) with multi-modal fusion and collaborative regularization.
2. **Continued Pre-Training (CPT).** An LLM is further pre-trained on a mixture of user behavior sequences (with SIDs) and domain text, to align the new SID modality with the LLM token space.
3. **Supervised Fine-Tuning (SFT).** The model is fine-tuned to autoregressively predict next-item SIDs from user context and history, yielding a generative retrieval model.

SIDv2 Details (from Sec. 2.1)

- **Multi-modal fusion:** The SID model combines multiple video embeddings (e.g., text, visual, audio) and projects them into a unified representation before quantization. The paper describes a concatenation of modality embeddings and a projection into shared space, explicitly designed to avoid isolated modality tokens.
- **Hierarchical multi-resolution codebooks:** The vocabulary size decreases with hierarchy depth to preserve fine detail early while compressing higher levels. This structure enables semantic hierarchies and improves coverage.
- **Progressive masking:** Higher-level codebooks are masked during training to enforce hierarchical dependence, making SIDs more robust and semantically structured.
- **Co-occurrence contrastive loss:** The model injects collaborative filtering signals directly into SID space, encouraging items frequently co-watched to map to closer SIDs.

Training Objectives (Sec. 2.2–2.3)

- **CPT objective:** Next-token prediction on combined corpora: user watch histories (SIDs) and domain text. The paper notes that this aligns SIDs with text tokens, improving in-context learning and language grounding.
- **SFT objective:** Minimize negative log-likelihood of next SID tokens, i.e., "the model is trained to minimize the following loss" for autoregressive next-item prediction.
- **Decoding:** Retrieval uses beam search over SID token sequences to generate candidate items (Sec. 2.3).

Experiments

- **Datasets:** Large-scale internal YouTube recommendation datasets spanning Long-Form Video (LFV) and Shorts.
- **Baselines:** Heavily-optimized Transformer-based retrieval model using large embedding tables (LEMs), reflecting production practice (Sec. 3).
- **Main retrieval gains:** The paper reports that "PLUM achieves substantial improvements for retrieval compared to a heavily-optimized production model built with large embedding tables" (Abstract). Table 1 shows lift in CTR and watch-time metrics across LFV and Shorts.
- **SIDv2 gains:** Table 4 ablations highlight SIDv2 improvements over SIDv1, including higher SID uniqueness and better retrieval recall.
- **Scaling study:** The paper demonstrates that synchronized scaling of model size and training data yields continued performance gains, with MoE models (>900M activated parameters) showing strong scaling behavior.
- **Live experiments:** The model was tested in live traffic by inserting PLUM recommendations into the candidate pool; results validate production relevance.

References (selected)

- Hoffmann et al. (2022). Training compute-optimal large language models. arXiv:2203.15556.
- Kang & McAuley (2018). Self-Attentive Sequential Recommendation. arXiv:1808.09781.
- Liu et al. (2022). Monolith: Real Time Recommendation System With Collisionless Embedding Table. arXiv:2209.07663.

- Li et al. (2025). BBQRec: Behavior-Bind Quantization for Multi-Modal Sequential Recommendation. arXiv:2504.06636.
- Huang et al. (2025). Towards Large-scale Generative Ranking. arXiv:2505.04180.
- Ju et al. (2025). Generative Recommendation with Semantic IDs: A Practitioner's Handbook. arXiv:2507.22224.

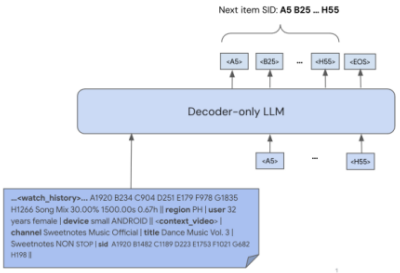

Figure 2: Illustration of Generative Retrieval for next video recommendation. The input prompt is a sequence of interleaved SID tokens, text and custom tokens for numerical features.

This fine-tuning employs a standard autoregressive, maximum-likelihood objective. The model learns to predict the SID tokens of ground-truth videos, which are defined as clicked videos from user logs given user context and history. Concretely, the model is trained to minimize the following loss:

$$\mathcal{L}_{\text{SFT}} = -\sum_{t=1}^{L} r(\text{user}, v_{click}) \cdot \log P(sid_t | \text{Context}_{\text{user}}, \text{History}_{\text{user}}, sid_{<t}), \tag{3}$$

where $[sid_1, .., sid_L]$ represents the SID of clicked video v_{click}, and $r(user, v_{click})$ is a handcrafted reward signal of each click. In practice, given the high cost of training, we sample training examples based on this reward and then weigh the sampled examples equally in the loss. As illustrated in Figure 2, the input prompt contains not only SID tokens and custom tokens for numerical features, but also other text features that can be naturally encoded by pre-trained LLMs.

During inference, we use beam search to decode multiple SID sequences, which serve as the set of retrieved candidates. Each generated SID is then mapped to a real video in our billions-scale corpus. While this generative process can potentially produce invalid SIDs (hallucination) or SID-to-video collisions, we observe that the hallucination rate after SFT is very low (< 5%), and the uniqueness of SID-to-video mapping remains high (see Table 4).

3 Experiments

In this section, we conduct a comprehensive set of experiments to validate the PLUM framework. We first evaluate the performance of a PLUM-based generative retrieval model against a Transformer-based large-embedding retrieval model that contributes to a majority of impressions in production (Section 3.1). Following this, Section 3.2 demonstrates the effectiveness of our proposed enhancements to SIDs. We then conduct ablation studies for two critical components in the PLUM framework: Section 3.3 quantifies the precise impact of the continued pre-training (CPT) stage and the value of initializing from a pre-trained LLM. We finally present a detailed scaling study to analyze the relationship between model size, compute, and retrieval performance (Section 3.4).

3.1 Performance of Generative Retrieval

Traditional recommender systems use large embedding models to recommend candidates. However, generative retrieval has a number of advantages over traditional systems. In this section, we study the effectiveness of generative retrieval on the YouTube production setup.

3.1.1 *Experiment Setup.*

Model. For the following experiments, we trained a 900M activated-param PLUM model from the Gemini-1.5 Mixture-of-Experts (MoE) family on both Long Form Video (LFV) and Shorts. The model is warm started from Gemini and continuously finetuned on new engagement data as described above. The training data is a mixture of the most recent data and historical data. During serving, target sequences are decoded using beam search. In our experiments, beam search performed better than random decoding, although at the cost of some diversity.

Baseline. We compare the performance of the above model to traditional LEM retrieval. This baseline is the top-performing production model, being highly-optimized since early work such as Chen et al. [2]. This model is also based on the Transformer architecture, but most of its parameters are in the embedding layer, with O(10M) vocab sizes for input and output item IDs. Specifically, LEM's neural network comprises only 0.4% of its total parameters, whereas PLUM's neural network accounts for 90%.

3.1.2 *Experiment Results.*

Recommendation Quality. We assess the quality of recommendations on a few different axes. First, we measure the effective vocab size of a recommender as the number of unique videos needed to cover 95% of its impressions. Higher vocab sizes are generally desirable for personalized discovery of niche content, showing the model can generalize better. Next, we compare the effectiveness of recommendations using two metrics: a) The click-through-rate (CTR) and Recommendation Acceptance. For Recommendation Acceptance, we compare both the length of video watched (WT/View) and the fraction of video watched (WF/View). In all cases, we report the ratio of the metric for the 900M MoE to the metric for the LEM model.

Table 2: Comparison of recommendation quality: Each number is a ratio, dividing the metric for PLUM by that of LEM.

Metric	LFV	Shorts
Effective Vocab Size	2.60x	13.24x
CTR	1.42x	1.33x
WT/View	0.72x	1.13x
WF/View	1.32x	1.03x

We observe that the PLUM model achieves much larger effective vocab size, while having competitive performance against LEM based on the metrics related to user reactions.

Figure 1: Generative retrieval overview (from PLUM paper, Figure 2).