CAN GENERATIVE RETRIEVERS LEARN SEMANTIC IDS WITHOUT FORGETTING HOW TO SPEAK?

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative retrieval (GR) enables end-to-end retrieval by generating document semantic identifiers (SIDs). However, retrieval-only fine-tuning can over-specialize pretrained language models to SID prediction, substantially distorting their natural-language distribution and limiting their suitability for interactive systems that must both retrieve documents and generate natural-language responses. We introduce SpeakGR, a dual-objective framework that learns SIDs while preserving language generation. It combines supervised SID learning with speak-preserving regularization: an on-policy distillation objective that aligns the current model with a frozen copy of the original model on student-generated prefixes using forward KL over the original text vocabulary. We further propose Adaptive SpeakGR, which dynamically adjusts the preservation strength based on observed language drift. Compared with SFT-only, SpeakGR reduces WikiText-2 forward KL by 81.3–93.8% on MS MARCO and 81.2–85.2% on Natural Questions (NQ) while retaining effective retrieval across three different LLMs. Adaptive SpeakGR further improves retrieval over SpeakGR in most settings while maintaining substantially lower language drift than SFT-only. Code is available at `https://anonymous.4open.science/r/SpeakGR-D75E`.

1 INTRODUCTION

Generative retrieval (GR), in which a neural model generates document identifiers directly from a query without relying on an explicit search index, has become an active research direction in information retrieval (Tay et al., 2022; Rajput et al., 2023; Wang et al., 2022). Its central idea is to reformulate retrieval as an *autoregressive generation* problem, replacing conventional candidate scoring with direct prediction of the identifier of a relevant document (Tay et al., 2022; Wang et al., 2022; Bevilacqua et al., 2022). A common design identifies each document with a short sequence of semantic identifiers (SIDs), extends the vocabulary of a pretrained language model with the SID tokens, and fine-tunes the model to generate these identifiers autoregressively. This framing turns document retrieval into a native sequence-generation task and has motivated substantial progress in document representations, identifier construction, training objectives, and decoding strategies (Lee et al., 2023). However, standard GR objectives optimize the model for SID generation without explicitly preserving its pretrained language-modeling behavior.

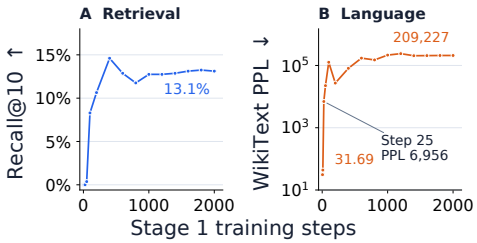

Figure 1: **Language forgetting during SID indexing.** Qwen3-0.6B+MS MARCO over 2,000 updates. (A) Recall@10; (B) WikiText-2 PPL (log scale). PPL rises from 31.69 to 6,956 within 25 updates. Stage 1 consists of document-to-SID indexing.

Why does this matter? Generative retrieval is attractive when a single language model serves the user end-to-end. The model first retrieves relevant documents by generating their identifiers, then uses the retrieved content to answer, summarize, cite, or continue the interaction (Radlinski and Craswell, 2017; Aliannejadi et al., 2019; Lewis et al., 2020). In ambiguous cases, the model may also ask a clarifying question, receive the user's response, and use that clarification to refine retrieval or generate a better answer. This is precisely why retrieval is built on top of a pretrained language

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.

model rather than a dedicated encoder. If retrieval fine-tuning substantially degrades the model's language behavior, it can no longer support the broader interaction beyond retrieval, regardless of its Recall@10. Preserving language generation is therefore important for interactive search, where retrieval quality alone does not capture the model's ability to answer follow-up questions, explain retrieved information, or conduct clarification turns with the user.[1]

Learning SIDs while preserving the pretrained language policy therefore provides three concrete benefits. *(i) Retrieval as an added capability, not a replacement:* the model learns to emit SIDs while retaining its ability to generate text, allowing retrieval specialization without sacrificing language generation. *(ii) A single interface for retrieval and for language:* because both SIDs and text are generated autoregressively, a model that keeps both can support a unified autoregressive interface for retrieval and language generation, with SID resolution connecting retrieval to subsequent generation. *(iii) Lower system cost:* existing GR systems often pair the retriever with a separate generator or reader for downstream knowledge-intensive tasks (Bevilacqua et al., 2022; Song et al., 2024). In contrast, a retriever that retains strong language capabilities could reuse the same model for both retrieval and generation, reducing the need to maintain a separate generation model.

However, none of these benefits and capabilities can be realized, or even studied, until a more basic question is settled: *is the pretrained language behavior retained during SID-based retrieval specialization?* Standard GR training gives little reason to expect that it does. SID specialization repeatedly trains the same parameters to map documents, pseudo-queries, and queries into short sequences over an identifier vocabulary (Zhou et al., 2022; Mekonnen et al., 2025; Wang et al., 2022), with nothing in the objective to constrain the text-token policy. Also, most prior work on GR evaluates only retrieval effectiveness (Wang et al., 2022; Tay et al., 2022; Rajput et al., 2023), leaving the effect on pretrained language behavior largely unknown. In this work, we therefore first measure how SID-based retrieval specialization changes the pretrained language behavior of the model. Across three LLM backbones and two retrieval datasets (six backbone–dataset settings), our experiments reveal a consistent tension: standard SID fine-tuning learns effective retrieval but can rapidly distort the pretrained language distribution, with substantial degradation emerging even at an early stage of training (Figure 1). These results further motivate us to ask: **Can a generative retriever learn to retrieve with semantic IDs without forgetting the pretrained language behavior of its backbone?**

To mitigate forgetting of pretrained speaking abilities, we introduce *SpeakGR*, a dual-objective framework combining supervised SID learning with *speak-preserving regularization*. While SID learning specializes the model for GR, the regularizer preserves its pretrained language behavior. Inspired by on-policy distillation (OPD) (Agarwal et al., 2024), the current model generates continuations, while a frozen original model provides next-token distributions at the same prefixes. A forward-KL objective over the original text vocabulary constrains language drift during SID learning. We further introduce *Adaptive SpeakGR*, which adjusts preservation strength according to observed drift. Across matched model–dataset settings (see Table 1), SpeakGR substantially reduces language drift while retaining effective retrieval. Compared with SpeakGR, Adaptive SpeakGR improves retrieval in five of the six settings and maintains substantially lower language drift than SFT-only.

Our contributions are threefold:

- We systematically study forgetting during SID-based generative retrieval training and show that standard retrieval specialization can substantially distort a pretrained model's natural-language behavior despite achieving effective retrieval.
- We introduce *SpeakGR*, a dual-objective training framework that couples supervised SID learning with *speak-preserving regularization*, allowing retrieval specialization and language preservation to be optimized jointly within the same model. Building on this, we further introduce *Adaptive SpeakGR*, which adjusts the preservation strength in response to the observed language drift.
- We evaluate SpeakGR and Adaptive SpeakGR across a variety of model–dataset settings and show that they substantially reduce language drift while retaining effective retrieval performance.

[1] In this work, "language preservation" specifically refers to preserving the pretrained model's natural-language predictive behavior, rather than guaranteeing preservation of every specialized downstream capability such as mathematical reasoning or code generation.

2 Related Work

Generative retrieval. Generative retrieval (GR) defines document retrieval as autoregressive generation of document identifiers. Representative methods include DSI (Tay et al., 2022), NCI (Wang et al., 2022), SEAL (Bevilacqua et al., 2022), and GLEN (Lee et al., 2023), which improve identifier construction, generation, and constrained decoding. More recent work further explores alternative identifier designs and retrieval objectives, including term-set generation in TSGen (Zhang et al., 2024), simultaneous decoding with hybrid identifiers in PAG (Zeng et al., 2024), and relevance-aware identifier learning in MERGE (Zhang et al., 2025a). Related semantic-ID paradigms have also been adopted in generative recommendation, from learning semantic item identifiers for autoregressive recommendation (Rajput et al., 2023) to recent differentiable SID learning such as DIGER (Fu et al., 2026). While these works primarily focus on improving retrieval or recommendation effectiveness and identifier learning, we study how SID specialization affects the pretrained natural-language behavior of the underlying language model.

Preserving language during specialization. Fine-tuning on narrow objectives can degrade previously acquired language capabilities (Luo et al., 2025). Existing approaches mitigate such forgetting through parameter constraints, replay, or distillation (Kirkpatrick et al., 2017; Sun et al., 2020). A concurrent work, ORBIT (Verma et al., 2026), studies forgetting in sequential product recommendation, whereas we focus on query-to-document generative retrieval over MS MARCO (Nguyen et al., 2016) and NQ (Kwiatkowski et al., 2019). The two methods also differ in preservation strategy: ORBIT monitors parameter drift and triggers origin-model weight merging, while SpeakGR directly regularizes the pretrained language distribution during SID learning. Adaptive SpeakGR further adjusts this preservation strength online according to observed language drift, without weight merging.

More broadly, language preservation during GR specialization remains underexplored, and, to the best of our knowledge, no prior work has explicitly formulated SID learning and language preservation as coupled training objectives. This motivates a unified training framework that learns to retrieve while preserving the model's natural-language behavior.

3 Problem Formulation

We consider a pretrained language model that is adapted to generative retrieval by extending its vocabulary with semantic identifier (SID) tokens. Let π_0 denote the frozen pretrained language policy over its original text vocabulary $\mathcal{V}_{\text{text}}$.

For each retrieval setting, let L denote the fixed identifier length and let $\mathcal{V}_\ell$ denote the SID token set available at position $\ell \in \{1, \dots, L\}$. We define the complete SID vocabulary as

$$\mathcal{V}_{\text{SID}} = \bigsqcup_{\ell=1}^{L} \mathcal{V}_\ell, \qquad \mathcal{V}_{\text{SID}} \cap \mathcal{V}_{\text{text}} = \varnothing, \tag{1}$$

where the identifier length and the size of each level-specific vocabulary are specified by the SID construction used in each experimental setting.

Let π_θ denote the *current trainable student policy* with parameters θ. The student is initialized from π_0, which has its vocabulary extended from $\mathcal{V}_{\text{text}}$ to $\mathcal{V}_{\text{text}} \cup \mathcal{V}_{\text{SID}}$. During retrieval specialization, θ is updated while the origin policy π_0 remains frozen.

Each indexed document d is assigned an SID using RQ-VAE or another quantization approach based on its text embedding (Tay et al., 2022; Rajput et al., 2023):

$$z(d) = (z_1, \dots, z_L), \qquad z_\ell \in \mathcal{V}_\ell. \tag{2}$$

For a query q, the generative retriever models the probability of an identifier autoregressively as

$$\pi_\theta(z \mid q) = \prod_{\ell=1}^{L} \pi_\theta\left(z_\ell \mid q, z_{<\ell}; \mathcal{V}_\ell\right), \tag{3}$$

where $\pi_\theta(\cdot \mid q, z_{<\ell}; \mathcal{V}_\ell)$ denotes the student's next-token distribution after restricting and renormalizing the logits to the valid SID tokens in $\mathcal{V}_\ell$. Thus, although the student is defined over the enlarged vocabulary $\mathcal{V}_{\text{text}} \cup \mathcal{V}_{\text{SID}}$, retrieval operates over a level-specific SID action space.

3.1 RETRIEVAL SPECIALIZATION

Let $\mathcal{Y}$ denote the set of supervised query–identifier pairs (q, z). If a query is associated with multiple relevant documents, each query–document association contributes a separate pair with the corresponding document identifier.

Following Wang et al. (2022), we optimize a level-restricted cross-entropy objective over the supervised SID positions:

$$\mathcal{L}_{\text{ret}}(\theta) = -\frac{1}{N_{\text{ret}}} \sum_{(q,z)\in\mathcal{Y}} \sum_{\ell=1}^{L} \log \pi_\theta \left(z_\ell \mid q, z_{<\ell}; \mathcal{V}_\ell\right), \qquad N_{\text{ret}} = \sum_{(q,z)\in\mathcal{Y}} |z| = L|\mathcal{Y}|. \tag{4}$$

The loss is normalized by the number of supervised SID tokens. Each training sequence has the form $(q, z_1, \ldots, z_L, \texttt{EOS})$. The `EOS` token is teacher-forced for sequence-format consistency but is excluded from Equation (4), since the identifier length is fixed within each retrieval setting.

At inference time, SID generation is constrained by the level-specific action spaces together with a trie constructed from the identifiers of the indexed documents. The trie ensures that decoding produces a valid and complete document identifier. Each valid decoded identifier maps to a document bucket; documents sharing an SID are ranked within that bucket.

3.2 MEASURING LANGUAGE DRIFT

Retrieval specialization updates a model that was originally pretrained to predict natural-language tokens. To quantify the resulting language forgetting (a form of catastrophic forgetting), we measure how far the specialized model's next-token distribution deviates from that of the frozen pretrained model on natural text.

For any natural-language prefix s, let $\pi_\theta^{\mathcal{V}_{\text{text}}}(\cdot \mid s)$ denote the student's next-token distribution restricted to the original text vocabulary $\mathcal{V}_{\text{text}}$ and renormalized:

$$\pi_\theta^{\mathcal{V}_{\text{text}}}(v \mid s) = \frac{\pi_\theta(v \mid s)}{\sum_{u\in\mathcal{V}_{\text{text}}} \pi_\theta(u \mid s)}, \qquad v \in \mathcal{V}_{\text{text}}. \tag{5}$$

This restriction isolates changes within the original text-token distribution from probability mass assigned directly to the newly introduced SID tokens.

Let μ_{text} denote a distribution over natural-language prefixes. We quantify language drift as follows:

$$\Delta_{\text{lang}}(\pi_\theta, \pi_0) = \mathbb{E}_{s\sim\mu_{\text{text}}} \left[D_{\text{KL}}\Big(\pi_0(\cdot \mid s) \,\Big\|\, \pi_\theta^{\mathcal{V}_{\text{text}}}(\cdot \mid s)\Big)\right], \tag{6}$$

where KL is the Kullback–Leibler divergence. A smaller Δ_{lang} indicates that the specialized model remains closer to the pretrained model's next-token behavior on natural text. In our experiments, we instantiate μ_{text} with fixed prefixes from the WikiText-2 test split (Merity et al., 2016) and use Δ_{lang} as our primary measure of distributional preservation.

4 SPEAKGR: JOINT SID LEARNING AND SPEAK-PRESERVING REGULARIZATION

One model, two objectives. The goal of *SpeakGR* is to jointly optimize two objectives over the same student parameters θ: (i) an SID-learning objective for effective generative retrieval and (ii) a *speak-preserving regularization* objective used to retain the model's pretrained natural-language behavior. The decoding space is specified by the application: retrieval restricts generation to the level-specific SID vocabulary and the document trie, whereas natural-language generation uses the original text vocabulary $\mathcal{V}_{\text{text}}$. This, in short, means that both objectives train the same model parameters for their respective behaviors.

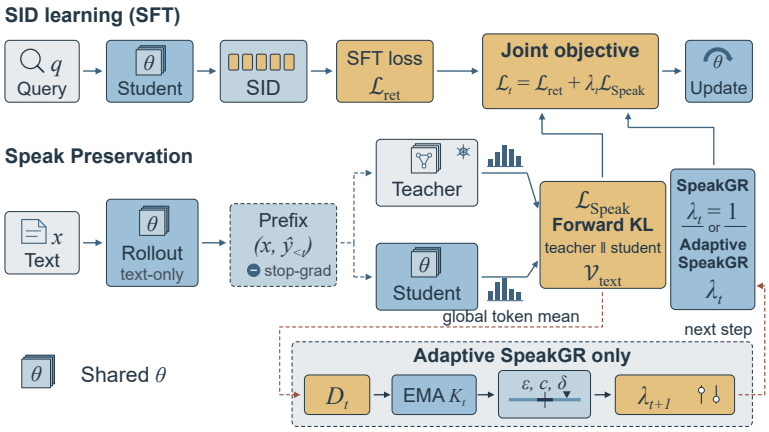

Figure 2: **Overview of SpeakGR and its adaptive variant.** A shared student learns SID retrieval while preserving language behavior through forward KL from a frozen teacher at student-sampled text prefixes. SpeakGR fixes the preservation weight at $\lambda_t = 1$; Adaptive SpeakGR uses the dashed controller to update λ_{t+1} from observed preservation KL for the next step.

4.1 Speak-preserving regularization

Inspired by on-policy distillation (Agarwal et al., 2024), we regularize the current model on natural-language states that it visits during generation. Let $\mathcal{D}_{\text{lang}}$ denote the fixed pool of natural-language prompts used for preservation, and let $\widetilde{\pi}_\theta$ denote the rollout distribution obtained from the current student after masking all SID logits and sampling only from $\mathcal{V}_{\text{text}}$.

For a prompt $x \in \mathcal{D}_{\text{lang}}$, the current student samples a text-only continuation:

$$\hat{y} \sim \widetilde{\pi}_\theta(\cdot \mid x; \mathcal{V}_{\text{text}}). \tag{7}$$

The sampled tokens are treated as constants, so no gradient passes through the discrete rollout. The frozen pretrained model and the trainable student then receive the same concatenated sequence $(x, \hat{y})$, each producing a next-token distribution at every response position.

For a preservation minibatch $\mathcal{B}_{\text{lang}} \subset \mathcal{D}_{\text{lang}}$, we match the two models at every sampled response position using forward KL over the original text vocabulary (where T_i is the length of the continuation the student sampled for prompt i):

$$\mathcal{L}_{\text{Speak}}(\theta) = \frac{1}{N_{\text{lang}}} \sum_{i \in \mathcal{B}_{\text{lang}}} \sum_{t=1}^{T_i} \sum_{v \in \mathcal{V}_{\text{text}}} \pi_0(v \mid s_{i,t}) \log \frac{\pi_0(v \mid s_{i,t})}{\pi_\theta^{\mathcal{V}_{\text{text}}}(v \mid s_{i,t})}, \qquad s_{i,t} = (x_i, \hat{y}_{i,<t}), \qquad N_{\text{lang}} = \sum_{i \in \mathcal{B}_{\text{lang}}} T_i. \tag{8}$$

The current student determines the prefixes $s_{i,t}$ on which preservation is performed. The frozen model does not generate target sequences; instead, it provides its full next-token distribution at the same model-generated prefixes. This allows *speak-preserving regularization* to regularize the model on the natural-language states it encounters as SID specialization progresses.

We use forward KL because, at a fixed prefix, minimizing $D_{\text{KL}}[\pi_0 \| \pi_\theta^{\mathcal{V}_{\text{text}}}]$ is equivalent, up to the entropy of the frozen model, to minimizing cross-entropy under its token distribution. Forward KL therefore penalizes the student for assigning insufficient probability to tokens favored by the pretrained model. We use forward KL to preserve the pretrained text distribution; we do not imply that it is universally preferable to other divergence measures.

$\mathcal{L}_{\text{Speak}}$ constrains the student's natural language next-token distribution, but does not directly optimize any specific downstream task capability. As we will see in Section 5.2, it reverses most of the distributional damage caused by SID specialization.

4.2 SPEAKGR TRAINING

SpeakGR combines supervised SID learning with *speak-preserving regularization* (see Section 4.1) through a dual-objective training loss:

$$\mathcal{L}_{\text{SpeakGR}} = \mathcal{L}_{\text{ret}} + \mathcal{L}_{\text{Speak}}. \tag{9}$$

The two losses are separately token-normalized. The retrieval objective drives specialization toward SID generation, while the preservation objective regularizes natural-language behavior during the same optimization process.

4.2.1 ADAPTIVE SPEAKGR

A static preservation weight applies the same pressure throughout training and cannot adapt to changes in language drift. An intuitive alternative is to strengthen preservation when the model drifts further from its pretrained behavior and relax it when the drift is small. Adaptive SpeakGR follows this principle as it dynamically adjusts only the preservation coefficient λ_t, while leaving the rest of SpeakGR unchanged.

At optimizer step t, let $\mathcal{B}_{\text{lang},t}$ denote the current preservation minibatch and define

$$D_t := \mathcal{L}_{\text{Speak}}(\theta_t; \mathcal{B}_{\text{lang},t}) \tag{10}$$

as the globally token-averaged preservation KL observed at that step. We maintain its exponential moving average (EMA) (Brown, 1959), and update the coefficient multiplicatively:

$$K_t = \alpha K_{t-1} + (1-\alpha) D_t, \tag{11}$$

$$e_t = \text{clip}\left(K_t/\epsilon - 1, -c, c\right), \tag{12}$$

$$\lambda_{t+1} = \text{clip}_{[\lambda_{\min}, \lambda_{\max}]}\left(\lambda_t \exp(\eta\, \tilde{e}_t)\right), \tag{13}$$

where

$$\tilde{e}_t = \begin{cases} 0, & |e_t| \le \delta, \\ e_t, & \text{otherwise.} \end{cases} \tag{14}$$

Here, $\alpha \in [0, 1)$ is the EMA smoothing factor, $\epsilon > 0$ is the reference preservation-KL level, c bounds the normalized deviation e_t, and δ defines a deadband around zero. The adaptation rate η controls how quickly λ_t changes, while $\lambda_{\min}$ and $\lambda_{\max}$ bound its allowable range. The current optimizer step uses λ_t, while the observed preservation KL determines the coefficient for the next step. The resulting objective is

$$\mathcal{L}_t = \mathcal{L}_{\text{ret}} + \lambda_t \mathcal{L}_{\text{Speak}} \tag{15}$$

Adaptive SpeakGR therefore changes only the strength of speak-preserving regularization over training, leaving the underlying dual-objective formulation unchanged. Details of the controller hyperparameters are given in Appendix A.

5 EXPERIMENTS

Our experiments aim to answer three questions:

1. How effectively can SpeakGR mitigate the language drift caused by SID specialization while retaining effective SID retrieval?
2. How does SpeakGR compare with other language-preservation strategies for generative retrieval?
3. How does adaptive preservation affect the retrieval–preservation trade-off, and how sensitive is it to the reference level ϵ, which directly sets the controller's target preservation-KL level?

We compare retrieval-only SFT, SpeakGR, and Adaptive SpeakGR across two retrieval corpora and three backbones, measuring both retrieval quality and language drift. We evaluate offline replay and

Table 1: Retrieval and language-preservation results across three backbones on MS MARCO and NQ. The best and second-best results among adapted models within each dataset–backbone setting are highlighted in **green** and light green, respectively. AdaSpeakGR denotes Adaptive SpeakGR.

Dataset	Backbone	Method	R@1 ↑	R@10 ↑	MRR@10 ↑	FKL ↓	PPL ↓	Top-1 ↑	SID mass ↓
MS MARCO	Qwen3-0.6B	Base	–	–	–	0.000	31.69	100.00%	≈ 0
		SFT-only	0.2426	0.6386	0.3649	8.103	53,966.98	0.13%	4.16%
		SpeakGR	0.2364	0.6349	0.3571	**1.519**	**88.48**	**46.15%**	0.62%
		AdaSpeakGR	**0.2537**	**0.6498**	**0.3725**	1.728	105.86	43.24%	**0.55%**
	Qwen3-1.7B	Base	–	–	–	0.000	23.07	100.00%	≈ 0
		SFT-only	**0.2760**	0.6696	**0.3977**	11.222	648,108.29	0.54%	4.18%
		SpeakGR	0.2562	0.6324	0.3764	**1.761**	**66.72**	**48.05%**	**0.15%**
		AdaSpeakGR	0.2649	**0.6745**	0.3927	1.841	70.66	46.88%	0.17%
	Gemma-3-1B-IT	Base	–	–	–	0.000	102.38	100.00%	≈ 0
		SFT-only	0.2525	0.6337	0.3808	13.005	13,615,975.95	1.20%	26.37%
		SpeakGR	0.2624	0.6361	0.3813	0.809	88.15	56.40%	0.20%
		AdaSpeakGR	**0.2686**	**0.6399**	**0.3864**	**0.775**	**86.10**	**57.87%**	**0.19%**
NQ	Qwen3-0.6B	Base	–	–	–	0.000	31.69	100.00%	≈ 0
		SFT-only	**0.4713**	**0.7397**	**0.5618**	6.379	10,251.46	6.07%	7.00%
		SpeakGR	0.3960	0.6582	0.4786	**1.044**	**56.30**	**52.40%**	1.91%
		AdaSpeakGR	0.4706	0.7285	0.5543	1.116	66.51	52.33%	**1.51%**
	Qwen3-1.7B	Base	–	–	–	0.000	23.07	100.00%	≈ 0
		SFT-only	**0.4808**	**0.7507**	**0.5701**	5.545	2,452.24	16.53%	8.60%
		SpeakGR	0.4056	0.7172	0.5084	1.042	35.74	**58.11%**	0.23%
		AdaSpeakGR	0.4387	0.7280	0.5324	**1.030**	**34.66**	57.84%	**0.20%**
	Gemma-3-1B-IT	Base	–	–	–	0.000	102.38	100.00%	≈ 0
		SFT-only	0.3925	0.7138	0.4961	5.631	7,791.61	6.70%	18.32%
		SpeakGR	**0.4126**	**0.7324**	**0.5169**	0.831	91.58	55.96%	0.34%
		AdaSpeakGR	0.3932	0.7189	0.4956	**0.743**	**86.04**	**58.04%**	**0.30%**

ORBIT-style weight merging under a matched protocol. Throughout, retrieval and language metrics are computed on fixed held-out sets that play no role in training or checkpoint selection. We first describe the datasets and evaluation protocol, then present the results in Section 5.2. Additional experimental details are provided in Appendix A, and the evaluation metrics in Appendix E. We further discuss our results in Appendix B.

5.1 Datasets

Retrieval datasets and training. Following Mekonnen et al. (2025) and Sun et al. (2023), we use retrieval subsets derived from MS MARCO (Nguyen et al., 2016) and NQ (Kwiatkowski et al., 2019). Our processed corpora contain 319,872 and 109,739 documents, respectively; we refer to them as MS MARCO and NQ throughout. Following Zhou et al. (2022) and Mekonnen et al. (2025), training proceeds in three stages: document-to-SID indexing, synthetic query-to-SID training, and supervised query-to-SID fine-tuning (see Table 4 for detailed configurations). These stages learn the document catalog, broaden query coverage, and adapt to human queries. For synthetic queries, we use ten released docT5query generations per document on MS MARCO (Nogueira and Lin, 2019) and the released synthetic queries on NQ (Sun et al., 2023).

Language-preservation prompts. Our pool contains 2,048 continuation prefixes from WikiText-2 training text (Merity et al., 2016) and 2,048 instruction-style prompts derived from training passages, MS MARCO's training questions, and task templates. Motivated by the importance of instruction diversity (Wang et al., 2023), this mixture provides varied contexts for preserving the base model's language behavior, covering both free-text continuation and instruction-following. SpeakGR and Adaptive SpeakGR use only prompts: the student generates continuations, and the frozen base model provides next-token distributions on the same prefixes. WikiText-2 test is reserved exclusively for evaluation. Pool composition, construction details, and examples appear in Appendix D (Table 5 and Table 6).

5.2 Results

5.2.1 Retrieval and language preservation

To address the first question, we compare retrieval-only SFT with SpeakGR and Adaptive SpeakGR across six matched dataset–backbone settings: MS MARCO and NQ with Qwen3-0.6B, Qwen3-1.7B, and Gemma-3-1B-IT. For each setting, all methods start from the same pretrained model, use the same

Table 2: Comparison with language-preservation baselines on Qwen3-0.6B across MS MARCO and NQ. In ORBIT, the fraction of original parameters whose signs differ from the pretrained model is used to trigger back-merging when the threshold τ is exceeded. LP, MP, and HP denote low, medium, and high preservation constraint strength, respectively; smaller τ imposes a stronger constraint. The best and second-best results among adapted models within each dataset are highlighted in **green** and light green, respectively. AdaSpeakGR denotes Adaptive SpeakGR.

Dataset	Method	R@1 ↑	R@10 ↑	MRR@10 ↑	FKL ↓	PPL ↓	Top-1 ↑	SID mass ↓
MS MARCO	Base	–	–	–	0.000	31.69	100.00%	≈ 0
	SFT-only	0.2426	0.6386	0.3649	8.103	53,966.98	0.13%	4.16%
	SpeakGR	0.2364	0.6349	0.3571	1.519	88.48	46.15%	0.62%
	AdaSpeakGR	0.2537	0.6498	0.3725	1.728	105.86	43.24%	0.55%
	Offline Replay	0.2475	0.5903	0.3531	8.672	32,933.41	28.58%	0.116%
	ORBIT (LP, $\tau = 0.10$)	**0.2946**	**0.6584**	**0.4119**	7.391	35,443.46	3.57%	7.576%
	ORBIT (MP, $\tau = 0.05$)	0.2611	0.5817	0.3657	3.927	1,262.86	17.55%	0.931%
	ORBIT (HP, $\tau = 0.007$)	0.0309	0.1980	0.0731	**0.273**	**42.84**	**76.33%**	**0.013%**
NQ	Base	–	–	–	0.000	31.69	100.00%	≈ 0
	SFT-only	**0.4713**	**0.7397**	**0.5618**	6.379	10,251.46	6.07%	7.00%
	SpeakGR	0.3960	0.6582	0.4786	1.044	56.30	52.40%	1.91%
	AdaSpeakGR	0.4706	0.7285	0.5543	1.116	66.51	52.33%	1.51%
	Offline Replay	0.3075	0.5811	0.3906	4.445	2,420.93	26.06%	5.681%
	ORBIT (LP, $\tau = 0.10$)	0.4406	0.7236	0.5327	6.257	9,208.21	6.73%	8.337%
	ORBIT (MP, $\tau = 0.05$)	0.3799	0.6656	0.4713	4.110	1,251.65	21.36%	6.570%
	ORBIT (HP, $\tau = 0.007$)	0.1775	0.3849	0.2394	**0.500**	**43.73**	**67.52%**	**0.272%**

SID assignments and retrieval supervision, and are evaluated with the same retrieval and WikiText-2 protocols. Table 1 reports both retrieval metrics and measures of language preservation.

Retrieval-only SFT consistently causes substantial language drift. Across the six settings, its WikiText forward KL (FKL) ranges from 5.545 to 13.005. SpeakGR reduces this to 0.809–1.761, corresponding to relative reductions of 81.3–93.8% on MS MARCO and 81.2–85.2% on NQ. The same trend is reflected in perplexity (PPL) and top-1 agreement. Retrieval effects depend on the setting: SpeakGR remains close to SFT on MS MARCO with Qwen3-0.6B, loses more recall on the Qwen NQ settings, and improves recall for both Gemma settings. Adaptive SpeakGR often recovers additional retrieval performance: for example, on MS MARCO with Qwen3-1.7B it reaches an R@10 of 0.6745 versus 0.6696 for SFT-only while reducing FKL from 11.222 to 1.841.

What this tells us. SpeakGR consistently preserves the pretrained natural-language distribution far better than retrieval-only SFT while retaining effective SID retrieval. The exact retrieval–preservation balance varies by dataset and backbone, and the adaptive variant can recover additional retrieval performance in several settings without giving up the preservation benefit.[2] Further analysis shows that the degradation is driven primarily by changes in the model's relative probabilities over ordinary text tokens, rather than by probability mass leaking from text tokens to the newly introduced SID vocabulary (Appendix C). Representation analysis further shows that speak-preserving regularization substantially reduces directional drift in middle and later layers while still allowing internal representations to adapt for retrieval.

5.2.2 Comparison with language-preservation baselines

To address the second question, we compare SpeakGR with two representative alternatives for retaining previous behavior during specialization: offline replay and ORBIT. Replay adds a response-level language CE objective from a fixed buffer, whereas ORBIT monitors parameter drift and conditionally merges the student weights with the frozen origin model. We evaluate all methods on Qwen3-0.6B for both NQ and MS MARCO in Table 2.

The baselines expose different retrieval–preservation trade-offs. At the evaluated setting, offline replay provides limited protection of the held-out language distribution: on MS MARCO, it reaches an FKL of 8.672 and an R@10 of 0.5903. ORBIT is strongly threshold-dependent: a loose threshold retains retrieval performance but leaves substantial language drift (R@10/FKL $0.6584/7.391$ on MS MARCO), whereas a tight threshold reduces FKL to 0.273 but also reduces R@10 to 0.1980.

[2] Achieving state-of-the-art retrieval performance is beyond the scope of this paper; stronger retrieval methods are complementary and could be combined with speak-preserving regularization (Sun et al., 2023; Mekonnen et al., 2025).

SpeakGR occupies a different region of this trade-off, with Adaptive SpeakGR retaining an R@10 of 0.6498 and an FKL of 1.728 on MS MARCO. The same qualitative pattern appears on NQ: Adaptive SpeakGR achieves an R@10 of 0.7285 with an FKL of 1.116, compared with $0.5811/4.445$ for replay and $0.7236/6.257$ for ORBIT with a loose threshold, reported as R@10/FKL.

What this tells us. No preservation method dominates across all operating points. At its evaluated setting, replay reduces language drift relative to retrieval-only SFT on NQ but does not match SpeakGR's joint retrieval–preservation balance. ORBIT can move between retrieval and preservation through its threshold, but incurs a non-trivial retrieval cost under strong correction. SpeakGR instead provides a direct objective-level alternative that achieves substantial language preservation while maintaining strong retrieval performance, with Adaptive SpeakGR offering a stronger retrieval-oriented operating point.

5.3 Ablation and Adaptive Sensitivity

To address the third question, we vary the adaptive controller's reference preservation level $\epsilon \in \{0.25, 0.50, 0.75, 1.00, 1.25, 1.50, 2.00\}$ on Qwen3-0.6B with MS MARCO. Every point is a three-stage run and uses the same checkpoint-selection rule based on a separate Stage-3 validation set. Table 3 reports the numerical results, while Figure 3 shows the retrieval–preservation trade-off.

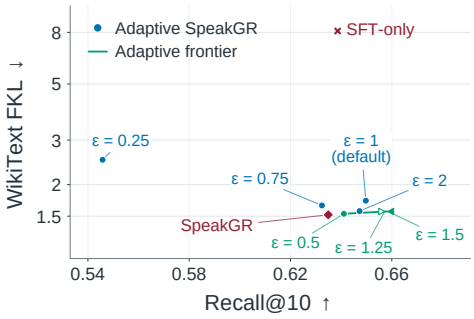

Figure 3: **Retrieval–preservation trade-off on Qwen3-0.6B+MS MARCO.** Lower right is better. Labels indicate ϵ; the star marks the default $\epsilon = 1$, and the diamond denotes SpeakGR. The green line is the Pareto frontier, which connects Adaptive SpeakGR runs not outperformed by another ϵ in both objectives. FKL uses a log scale; triangle tips locate near-coincident points.

The response to ϵ is non-monotonic. The strictest setting ($\epsilon = 0.25$) hurts both retrieval and held-out preservation, reaching R@10 0.5458 and FKL 2.507. In contrast, moderate values between 0.5 and 2.0 produce a substantially better region of the trade-off; the best R@10 is 0.6584 at $\epsilon = 1.5$ with FKL 1.566. Importantly, a smaller training reference level does not necessarily yield a smaller held-out WikiText FKL because the controller observes student-generated training states rather than the held-out evaluation distribution. We additionally study sensitivity to a fixed preservation coefficient in Appendix F. The results show that SpeakGR remains effective across a broad range of fixed weights, while Adaptive SpeakGR removes the need to commit to a single fixed coefficient by adjusting its preservation strength online.

What this tells us. Adaptive SpeakGR does not require a narrowly tuned reference level in a moderate range, but an overly strict target can be counterproductive. The adaptive mechanism therefore provides a practical way to move along the retrieval–preservation trade-off, while ϵ should be interpreted as a training reference rather than a direct bound on held-out language drift.

6 Conclusion

We ask a simple question: can a language model learn to retrieve with semantic IDs without forgetting how to generate natural language? Our experiments show that retrieval-only fine-tuning can substantially distort the pretrained language distribution, but this degradation can be substantially mitigated. SpeakGR jointly optimizes SID learning and speak-preserving regularization, reducing WikiText forward-KL drift while retaining effective retrieval performance, with trade-offs that vary across datasets and backbones. Adaptive SpeakGR adjusts preservation strength in response to the observed drift, offering an alternative to fixed weighting. Together, our results show that SID-based generative retrieval can be learned while substantially limiting drift from the model's pretrained natural-language distribution.

Reproducibility Statement

Code is available at the anonymous repository linked in the abstract. The experimental section and appendices provide the training schedule, model and optimization settings, Adaptive SpeakGR hyperparameters, preservation-prompt construction, and retrieval and language evaluation protocols needed to reproduce the reported experiments.

AI Assistance

The core research idea was developed by the authors. Generative AI tools assisted with identifying relevant literature, generating code for experimental implementation, curating part of the language-preservation prompt pool, assisting figures preparation, and drafting and polishing manuscript text. The authors reviewed the AI-assisted materials and take full responsibility for the methods, results, and final manuscript.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.13649.

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*, SIGIR'19, page 475–484, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361729. doi: 10.1145/3331184.3331265. URL https://doi.org/10.1145/3331184.3331265.

Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Wen-tau Yih, Sebastian Riedel, and Fabio Petroni. Autoregressive search engines: Generating substrings as document identifiers. In *Advances in Neural Information Processing Systems*, volume 35, pages 31668–31683, 2022.

Robert Goodell Brown. *Statistical Forecasting for Inventory Control*. McGraw-Hill, New York, 1959.

Clément L Canonne. A short note on an inequality between kl and tv. *arXiv preprint arXiv:2202.07198*, 2022.

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, page 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.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114. URL https://www.pnas.org/doi/abs/10.1073/pnas.1611835114.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR, 2019.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Sunkyung Lee, Minjin Choi, and Jongwuk Lee. GLEN: Generative retrieval via lexical index learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7693–7704. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.477.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 33:3776–3786, 2025. doi: 10.1109/TASLPRO.2025.3606231.

Kidist Amde Mekonnen, Yubao Tang, and Maarten de Rijke. Lightweight and direct document relevance optimization for generative information retrieval. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1327–1338, 2025.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A human generated machine reading comprehension dataset. In *Workshop on Cognitive Computation at NeurIPS*, 2016.

Rodrigo Nogueira and Jimmy Lin. From doc2query to doctttttquery. *Online preprint*, 6(2):1–3, 2019.

Filip Radlinski and Nick Craswell. A theoretical framework for conversational search. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*, CHIIR '17, page 117–126, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450346771. doi: 10.1145/3020165.3020183. URL https://doi.org/10.1145/3020165.3020183.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36:10299–10315, 2023.

EuiYul Song, Sangryul Kim, Haeju Lee, Joonkee Kim, and James Thorne. Re3val: Reinforced and reranked generative retrieval. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 393–409, 2024.

Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A Alemi, and Andrew G Wilson. Does knowledge distillation really work? *Advances in neural information processing systems*, 34:6906–6919, 2021.

Jingyuan Sun, Shaonan Wang, Jiajun Zhang, and Chengqing Zong. Distill and replay for continual language learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3569–3579. International Committee on Computational Linguistics, 2020. doi: 10.18653/v1/2020.coling-main.318. URL https://aclanthology.org/2020.coling-main.318/.

Weiwei Sun, Lingyong Yan, Zheng Chen, Shuaiqiang Wang, Haichao Zhu, Pengjie Ren, Zhumin Chen, Dawei Yin, Maarten de Rijke, and Zhaochun Ren. Learning to tokenize for generative retrieval. *Advances in Neural Information Processing Systems*, 36:46345–46361, 2023.

Yi Tay, Vinh Q. Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, Tal Schuster, William W. Cohen, and Donald Metzler. Transformer memory as a differentiable search index. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Neha Verma, Nikhil Mehta, Shao-Chuan Wang, Naijing Zhang, Alicia Tsai, Li Wei, Lukasz Heldt, Lichan Hong, Ed Chi, and Xinyang Yi. ORBIT: Preserving foundational language capabilities in genretrieval via origin-regulated merging. *arXiv preprint arXiv:2605.12419*, 2026. URL https://arxiv.org/abs/2605.12419.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 13484–13508, 2023.

Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Hao Sun, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, Xing Xie, Hao Allen Sun, Weiwei Deng, Qi Zhang, and Mao Yang. A neural corpus indexer for document retrieval. In *Advances in Neural Information Processing Systems*, volume 35, pages 25600–25614, 2022.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Hansi Zeng, Chen Luo, and Hamed Zamani. Planning ahead in generative retrieval: Guiding autoregressive generation through simultaneous decoding. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 469–480, 2024.

Fuwei Zhang, Xiaoyu Liu, Xinyu Jia, Yingfei Zhang, Shuai Zhang, Xiang Li, Fuzhen Zhuang, Wei Lin, and Zhao Zhang. Multi-level relevance document identifier learning for generative retrieval. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10066–10080, 2025a.

Peitian Zhang, Zheng Liu, Yujia Zhou, Zhicheng Dou, Fangchao Liu, and Zhao Cao. Generative retrieval via term set generation. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 458–468, 2024.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025b.

Yujia Zhou, Jing Yao, Zhicheng Dou, Ledell Wu, Peitian Zhang, and Ji-Rong Wen. Ultron: An ultimate retriever on corpus with a model-based indexer. *arXiv preprint arXiv:2208.09257*, 2022.

Table 3: Sensitivity of Adaptive SpeakGR to the reference preservation level ϵ on Qwen3-0.6B with MS MARCO. The $\epsilon = 1$ result is reused from the main table as the default setup.

Reference KL ϵ	R@10 ↑	Wiki FKL ↓
0.25	0.5458	2.507
0.50	0.6411	1.533
0.75	0.6324	1.654
1.00	0.6498	1.728
1.25	0.6572	1.565
1.50	0.6584	1.566
2.00	0.6473	1.572

A Experimental Setup

Table 4: The three-stage retrieval schedule. Effective global batches and data order are held fixed across GPU counts through gradient accumulation.

Stage	Mapping	Steps	LR	Global batch	Max length
1	document → SID	2,000	2×10^{-4}	1,536	320
2	pseudo-query → SID	8,000	5×10^{-4}	3,072	144
3	query → SID	2,000	1×10^{-4}	1,536	144

A.1 Models and training

We use Qwen3-0.6B/1.7B (Yang et al., 2025) and Gemma-3-1B-IT (Team et al., 2025), fully fine-tuning all student parameters, including embeddings, while keeping the base teacher frozen. We tune the number and size of the RQ-VAE codebooks, selecting five 256-entry codebooks to quantize Qwen3-Embedding document representations (Zhang et al., 2025b; Rajput et al., 2023). SIDs contain five tokens. Within each dataset–backbone comparison, all methods use identical SID assignments and student initialization. Only 1.94% of MS MARCO SID buckets contain multiple documents; we use a frozen Qwen3-Embedding-0.6B encoder to rerank documents within colliding SID buckets on both datasets while preserving the generated bucket order.

Table 4 summarizes the primary retrieval schedule, with the longest stage devoted to pseudo-query coverage. We use AdamW (Loshchilov and Hutter, 2017), 3% linear warmup followed by linear decay, and six language prompts per optimizer update. We generate student continuations with SID tokens masked and a temperature of 0.8, top-$p = 0.9$, top-$k = 20$, and at most 32 tokens. SpeakGR minimizes $\mathcal{L}_{\text{ret}} + \mathcal{L}_{\text{Speak}}$. In all main settings, Adaptive SpeakGR uses $\epsilon = 1$ nat/token, $\alpha = 0.95$, $c = 0.2$, $\delta = 0.05$, $\eta = 0.05$, and $\lambda_{\max} = 1$. Stage 1 starts with $\lambda_0 = 0.1$ and $\lambda_{\min} = 0.01$. Qwen3-0.6B retains this lower bound and carries the controller state across stages. For Qwen3-1.7B and Gemma-3-1B-IT, Stage 2 resets the controller to $\lambda = 0.25$ with a fresh KL EMA; Stages 2–3 use $\lambda_{\min} = 0.25$, and Stage 3 inherits the Stage 2 controller state. The controller observes training-stream KL.

Baselines. Following prior work on replay for continual language learning (Sun et al., 2020), we use an offline language-replay baseline. Its numerical settings are chosen to match SpeakGR's language-data and sequence-length budget: replay uses six examples per optimizer update, retains at most 640 prompt tokens, and supervises at most 32 response tokens per example. The replay buffer uses the same prompt sources and fixed offline responses described in Appendix Appendix D. Its response-only cross-entropy is computed over the original text vocabulary, excluding added SID tokens from the softmax normalization and masking prompt and padding positions. To match SpeakGR's auxiliary-loss weighting, the response-only replay loss is added to the retrieval loss with coefficient $\lambda_{\text{replay}} = 1.0$, the same coefficient used for SpeakGR's language-preservation loss. We fix this configuration as a matched reference and do not claim that it is optimal. We reimplement ORBIT following its original paper (Verma et al., 2026), because no official implementation was publicly available at the time of our experiments. Following the original algorithm, whenever sign

dissimilarity exceeds its threshold, the student weights are averaged 1:1 with the corresponding base-model weights, excluding SID rows and leaving the optimizer states unchanged. In Table 2, we denote the ORBIT threshold by τ to distinguish it from Adaptive SpeakGR's reference level ϵ.

A.2 Evaluation

We report Recall@1/10 and MRR@10 using trie-constrained beams of 10 for MS MARCO and 100 for NQ following Mekonnen et al. (2025) and Sun et al. (2023). Language evaluation scores 16,384 next-token predictions in 64 non-overlapping, 256-token WikiText-2 test input blocks, with shifted targets and reset context. Every causal position is scored. We measure fidelity to the base model using forward KL and top-1 agreement (Stanton et al., 2021), and additionally report perplexity. These metrics are computed after renormalizing over the base-model vocabulary. We also report SID probability mass before masking. Comparisons are made within each dataset–backbone setting.

B Discussion

Why WikiText? We use WikiText-2 as the primary language-preservation evaluation because it provides a fixed held-out distribution of natural text on which the specialized model can be compared directly with its pretrained counterpart token by token. This gives a controlled measure of distributional drift, but should not be interpreted as a complete measure of language capability. We use this as our primary measure of distributional preservation; broader capability preservation is outside the scope of our evaluation.

Can one model both retrieve and generate? Preserving natural-language behavior is particularly relevant when the same model is expected to support both retrieval and generation. In such a setting, the model can first generate an SID under retrieval constraints, resolve it to a document, and then use the retrieved content to produce a natural-language response. Our current experiments establish preservation of the pretrained text distribution during retrieval specialization, but do not yet constitute a full end-to-end retrieve–resolve–generate evaluation. We therefore view this unified use case as a motivating application rather than a demonstrated capability.

Hyperparameter sensitivity. Our study does not exhaustively tune the regularization strength, rollout sampling parameters, replay settings, or the adaptive controller. Accordingly, the replay result represents one fixed operating point rather than a tuned replay frontier. SpeakGR uses a simple static preservation weight, while Adaptive SpeakGR adjusts this weight online according to the observed preservation KL. Although this design provides a useful adaptive mechanism, improvements cannot be attributed solely to adaptation without comparing against a static SpeakGR run using a matched average preservation weight. We therefore treat the adaptive variant as an additional operating strategy rather than claiming that adaptive weighting is universally superior.

Limitations. Our main evaluation covers three language-model backbones and two retrieval corpora, but most cross-setting comparisons are based on a single main seed and one SID construction family. Moreover, strong recovery of the WikiText-2 distribution should not be interpreted as complete preservation of broader capabilities. Our evaluation is intentionally scoped to pretrained natural-language distribution preservation during retrieval specialization. Specialized capabilities such as mathematical reasoning, coding, and task-specific problem solving are outside the scope of this study. The displayed SpeakGR anchors in Table 2 match those in Table 1. Finally, broader multi-seed evaluation and stronger matched controls for the adaptive coefficient remain necessary before making claims of universal capability preservation or Pareto superiority.

C Mechanism analysis

Preserving the language distribution. For a natural-language prefix distribution μ, the preservation penalty is

$$R_\mu(\theta) = \mathbb{E}_{s\sim\mu} D_{\mathrm{KL}}\Big[\pi_0(\cdot \mid s) \,\|\, \pi_\theta^{\mathcal{V}_{\mathrm{text}}}(\cdot \mid s)\Big]. \tag{16}$$

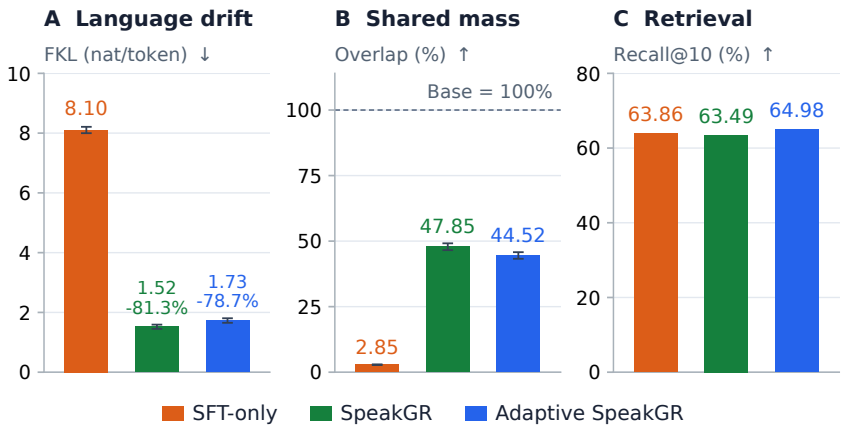

Figure 4: **Language preservation with SpeakGR and Adaptive SpeakGR.** Selected Qwen3-0.6B checkpoints on MS MARCO. (A) Forward KL from the frozen base model to the student; percentages indicate reductions relative to SFT-only. (B) Shared predictive probability mass with the base model. (C) Recall@10 on 808 retrieval queries. A/B use the same 64×256 WikiText-2 test positions; 95% confidence intervals use 10,000 block-bootstrap resamples conditional on the selected checkpoints.

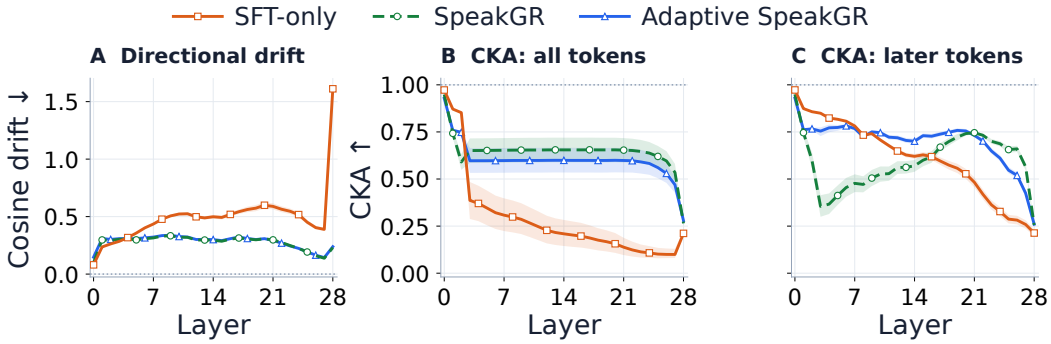

Figure 5: **Layer-wise residual-stream comparison on Qwen3-0.6B+MS MARCO.** (A) Cosine drift from the base model; (B) centered linear CKA; (C) CKA excluding the first 16 positions per block, with context unchanged. Layer 0 is the embedding output; layers 1–28 are block outputs before the final RMS normalization. A/B use the same 16,384 WikiText positions; C uses 15,360 of these positions. Bands show pointwise 95% block-bootstrap intervals.

At a fixed prefix, its gradient with respect to the student text logits is $\pi_\theta^{\mathcal{V}_{\text{text}}}(\cdot \mid s) - \pi_0(\cdot \mid s)$, correcting probability differences from the frozen base model. Define the shared predictive mass as $M(s) = \sum_{v \in \mathcal{V}_{\text{text}}} \min\{\pi_0(v \mid s), \pi_\theta^{\mathcal{V}_{\text{text}}}(v \mid s)\}$. Since $M(s) = 1 - \text{TV}(\pi_0, \pi_\theta^{\mathcal{V}_{\text{text}}})$, Pinsker's inequality (Canonne, 2022) and Jensen's inequality yield

$$\mathbb{E}_{s\sim\mu} M(s) \geq \max\left\{0,\, 1 - \sqrt{R_\mu(\theta)/2}\right\}. \tag{17}$$

Reducing KL therefore raises a lower bound on shared probability mass on the same prefix distribution; transfer to held-out text is measured empirically.

On the WikiText evaluation in Figure 4, SpeakGR and Adaptive SpeakGR reduce FKL from SFT's 8.10 to 1.52 and 1.73, while increasing shared mass from 2.85% to 47.85% and 44.52%. SID leakage accounts for only 0.58% of SFT's excess full-vocabulary NLL; the main degradation is within the normalized text distribution.

Where do the representations change? On the same checkpoints and prefixes, we compare base and student residuals $h_0^{(j)}(s)$ and $h_\theta^{(j)}(s)$ using mean cosine drift $1 - \cos(h_0^{(j)}(s), h_\theta^{(j)}(s))$

and centered linear CKA (Kornblith et al., 2019). CKA compares representation geometry up to orthogonal rotations and uniform rescaling, computed within each text block and then averaged.

Figure 5A shows lower directional drift through the middle and later layers with preservation. At the final block, drift is 1.61 for SFT, 0.23 for SpeakGR, and 0.24 for Adaptive SpeakGR; these values are nearly unchanged when the first 16 positions of each block are excluded. The geometry comparison is more sensitive to position selection (B/C): at layer 14, SpeakGR versus SFT CKA changes from 0.65 versus 0.21 to 0.57 versus 0.62 after this exclusion. Early layers are also not uniformly closer to the base. Thus, the evidence supports directional alignment in later layers, rather than a claim that all representation geometry is preserved.

These observations are consistent with function-space regularization: preserving language predictions permits internal adaptation for retrieval. This endpoint comparison does not identify a causal layer of forgetting or establish complete downstream capability preservation.

D LANGUAGE-PRESERVATION PROMPT POOL

The complete pool combines natural-text continuation prefixes with instruction-style prompts. This mixture provides contexts for preserving both ordinary text generation and responses to instructions and questions. In Table 5, we show the full composition, and in Table 6, we illustrate the inputs. These categories describe training prompts, not independently evaluated capabilities. They are included to diversify natural-language contexts for preservation.

Table 5: **Composition of the complete 4,096-entry preservation-prompt pool.** Counts refer to stored prompt instances, not unique templates. Source groups are shown in italics.

Prompt category	Count
A. Continuation prefixes: WikiText-2 training split	
Natural-text continuation	2,048
B. Instruction-style prompts	
B.1. WikiText-2 training text + task instructions (1,208)	
Summarization	218
Explanation	218
Paraphrasing	212
Key-point extraction	218
Title generation	217
Translation	125
B.2. MS MARCO training-query QA (217)	
Direct QA	111
Document-conditioned QA	106
B.3. Programmatically instantiated task templates (623)	
Writing (email)	120
Arithmetic	184
Logic	97
Coding	110
Advice/list generation	112
Instruction subtotal	**2,048**
Complete pool	**4,096**

Construction and supervision. Passage-based prompts attach fixed instructions to WikiText-2 training text. QA prompts use MS MARCO training questions, either alone or with retrieved evidence. Other prompts instantiate task templates with varying inputs, such as numbers for arithmetic and entities for logic. Candidates are screened using offline responses and are selected by task category; arithmetic and logic use rule-based reference responses during preparation. SpeakGR and Adaptive

SpeakGR consume only the selected prompts, not their stored answers: the student generates continuations and the frozen base model scores the same prefixes.

Table 6: **Representative examples from the preservation-prompt pool.** Examples are abbreviated from actual prompts and lightly formatted for readability. Bracketed fields replace passages or evidence; ellipses mark omitted continuation text. Continuation inputs are raw text prefixes, not explicit instructions to continue writing.

Category	Example prompt (abbreviated)
Continuation	Virginia Tech running back Cyrus Lawrence finished the game…
Summarization	Summarize the passage in two concise sentences. Passage: *[WikiText-2 training passage]*
Translation	Translate the following English sentence into Chinese. English: The church's north and south walls are supported by a series of buttresses.
Direct QA	Question: what is the arctic desert. Answer the question in one or two clear sentences.
Document-conditioned QA	Question: define statistical inference. Evidence: *[retrieved document]*. Use only the provided evidence.
Writing	Write exactly three short sentences to Sam, politely asking to reschedule a meeting because of a seminar.
Arithmetic	A store has 80 items and packs them equally into 4 boxes. How many full items go in each box and how many remain?
Logic	Premise: All salmon are fish. Some fish are red. Can we conclude that some salmon are red? Explain in two sentences.
Coding	Write a Python function that counts word frequencies in a string. Return only the code, without explanation.
Advice/list generation	List three practical ways of saving energy at home. Keep each item concise.

E Evaluation Metrics

Retrieval. Recall@1 and Recall@10 measure the fraction of queries whose relevant document appears in the top-1 and top-10 results. Each evaluation query is scored against a single designated relevant document. MRR@10 averages the reciprocal rank of the relevant document, assigning zero when it falls outside the top 10. Higher values indicate better retrieval.

Language preservation. All language metrics are averaged over the same WikiText-2 evaluation positions. Using π_0 and the text-normalized student $\pi_\theta^{\mathcal{V}_{\text{text}}}$ from Equation (5), we report: **FKL**, the base-to-student forward KL defined in Equation (6); **PPL**, the exponential of the student's mean negative log-likelihood of the observed next tokens; and **Top-1 agreement**, the percentage of positions where the two models predict the same highest-probability token. Lower FKL/PPL and higher agreement are better.

SID mass. We report the average percentage of probability assigned to SID tokens, $100 \sum_{v \in \mathcal{V}_{\text{SID}}} \pi_\theta(v \mid s)$, before text-vocabulary masking. Lower values indicate less probability allocated to identifiers in language contexts. Language metrics are compared within each dataset–backbone setting.

F Fixed-Weight Sensitivity

Table 7 examines whether SpeakGR depends on a carefully chosen static preservation coefficient. Across $\lambda \in \{0.1, 0.5, 1.0, 1.5\}$, all non-zero settings substantially reduce WikiText FKL, from 8.103 for retrieval-only SFT to 1.435–1.532, while retaining effective retrieval performance. This indicates that the preservation effect is robust to the choice of a fixed coefficient rather than depending on a

single carefully tuned value. Different coefficients nevertheless lead to different retrieval–preservation operating points, showing that the choice of preservation strength still affects the balance between the two objectives.

Notably, appropriately chosen fixed coefficients achieve some of the strongest results in this sensitivity study. We therefore do not claim that adaptive weighting is intrinsically superior to a well-chosen fixed weight. Instead, Adaptive SpeakGR is motivated by a practical difference in how the preservation strength is controlled. Static SpeakGR requires a single coefficient to be selected before training and uses the same preservation coefficient throughout training. Adaptive SpeakGR instead adjusts λ_t online according to the observed preservation KL, allowing the regularization pressure to change as language drift evolves. Its intended advantage is therefore not a guaranteed improvement over the best fixed operating point, but reduced reliance on committing to one manually selected preservation coefficient for the entire training process.

What this tells us. Together with the sensitivity analysis in Table 3, these results show that speak-preserving regularization is effective and robust under fixed weighting, while Adaptive SpeakGR provides a self-adjusting alternative for controlling the retrieval–preservation trade-off. When a suitable fixed coefficient is known, static weighting can be a strong choice; when the appropriate preservation strength is less clear or may vary over training, the adaptive formulation offers a practical mechanism for adjusting that strength online in response to the observed language drift.

Table 7: Sensitivity to the fixed SpeakGR preservation coefficient on Qwen3-0.6B with MS MARCO. Best results are highlighted in **green**.

λ	R@1 ↑	R@10 ↑	MRR@10 ↑	FKL ↓	PPL ↓	Top-1 ↑	SID mass ↓
0 (SFT-only)	0.2426	0.6386	0.3649	8.103	53,966.98	0.13%	4.16%
0.1	0.2512	**0.6621**	0.3804	**1.435**	82.02	**48.11%**	0.66%
0.5	**0.2723**	0.6535	**0.3904**	1.474	**80.21**	47.31%	**0.54%**
1.0 (SpeakGR)	0.2364	0.6349	0.3571	1.519	88.48	46.15%	0.62%
1.5	0.2364	0.6262	0.3493	1.532	89.31	46.55%	0.63%