Lost in *Decoding*? Reproducing and Stress-Testing the Look-Ahead Prior in Generative Retrieval

Kidist Amde Mekonnen
University of Amsterdam
Amsterdam, The Netherlands
k.a.mekonnen@uva.nl

Yongkang Li
University of Amsterdam
Amsterdam, The Netherlands
y.li7@uva.nl

Yubao Tang
University of Amsterdam
Amsterdam, The Netherlands
y.tang3@uva.nl

Simon Lupart
University of Amsterdam
Amsterdam, The Netherlands
s.c.lupart@uva.nl

Maarten de Rijke
University of Amsterdam
Amsterdam, The Netherlands
m.derijke@uva.nl

Abstract

Generative retrieval (GR) ranks documents by autoregressively generating document identifiers. Because many GR methods rely on trie-constrained beam search, they are vulnerable to early pruning of relevant prefixes under finite-beam decoding. *Planning Ahead in Generative Retrieval* (PAG) mitigates this failure mode by using simultaneous decoding to compute a document-level look-ahead prior that guides subsequent sequential decoding. We reproduce PAG at inference time and stress-test its decoding behavior. Using the authors' released checkpoint and identifier/trie artifacts under the reported decoding setup, we reproduce the main effectiveness results on MS MARCO Dev and TREC-DL 2019/2020, and corroborate the reported beam-size–latency trade-off in our hardware setting. Beyond reproduction, we introduce *plan drift* diagnostics that quantify how intent-preserving query variations, including misspellings, reordering, synonym substitutions, paraphrases, and naturality shifts, alter the planner's top-n candidate set and highest-weight planner tokens, and how these changes affect guided decoding. We find that PAG's planning signal is brittle under lexical surface-form variation: intent-preserving typos can trigger *plan collapse*, where the planned candidate pool shifts enough that the look-ahead bonus provides little useful guidance, effectively reverting decoding toward weaker unguided search. We further evaluate fixed-index cross-lingual robustness using non-English mMARCO queries against an English index, and assess query-side mitigation strategies that require no re-indexing; query translation provides the strongest recovery in our setting. Overall, our results confirm PAG's reported effectiveness and the benefit of planning-guided decoding under the released inference setup, while showing that these gains depend on the stability of the planning signal under realistic query variation and query–document mismatch. Code available at https://github.com/kidist-amde/lost-in-decoding.

CCS Concepts

• **Information systems** → **Query intent**; **Information retrieval**; **Retrieval models and ranking**; **Evaluation of retrieval results**.

SIGIR '26, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-2599-9/2026/07
https://doi.org/10.1145/3805712.3808567

Keywords

Generative retrieval, Trie-constrained decoding, Prefix pruning, Robustness, Cross-lingual query shift

ACM Reference Format:
Kidist Amde Mekonnen, Yongkang Li, Yubao Tang, Simon Lupart, and Maarten de Rijke. 2026. Lost in *Decoding*? Reproducing and Stress-Testing the Look-Ahead Prior in Generative Retrieval. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '26), July 20–24, 2026, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3805712.3808567

1 Introduction

Generative retrieval and prefix pruning. Generative retrieval (GR) reframes search as sequence generation: given a query, a model retrieves by autoregressively generating a document identifier (docid) [32]. At inference time, decoding is constrained to valid docids (e.g., via a trie), making retrieval sensitive to search-time errors. In particular, beam search is myopic: it can discard globally relevant documents when their docid prefixes receive low probability early in generation. This failure mode, *prefix pruning*, occurs when a relevant document's prefix falls outside a beam of width k and is no longer explored under finite-beam decoding [7, 27, 36, 37]. Because decoding choices can dominate retrieval outcomes in GR, it is essential to verify whether reported gains persist under released artifacts and reported inference settings. Moreover, for methods that rely on intermediate guidance signals, a further question arises: *is the guidance itself stable under realistic query variation, or can it become a bottleneck?*

Intermediate signals for reliable decoding. A growing line of GR work targets decoding reliability, motivated by the observation that end-to-end retrieval can be limited by *search errors* rather than raw model capacity. Prior work bridges generation and ranking by designing rank-aware identifiers, learning from relevance feedback (including reinforcement-style objectives), or directly optimizing document-level utility to better align token-level probabilities with retrieval quality [11, 15, 20, 36, 39, 40]. Several approaches in GR and constrained decoding compute a discrete, cheaper intermediate signal (or a look-ahead estimate) to guide constrained search, and then refine predictions in a subsequent stage [8, 10, 11, 18, 22, 26, 33, 37]. The robustness of such intermediate signals under realistic input variation remains under-examined, even though instability can become a single point of failure in the retrieval pipeline [11, 12, 14, 16].

Since real retrieval environments are noisy due to misspellings, paraphrases, segmentation ambiguity, and productive morphology, we argue for evaluations that go beyond end metrics to *instrument* the intermediate signal itself, separating cases where it provides consistent look-ahead guidance from cases where it destabilizes decoding and amplifies search errors.

Planning Ahead in Generative Retrieval (PAG). Planning Ahead in Generative Retrieval (PAG) [37] proposes a two-stage decoding strategy to mitigate *prefix pruning* in trie-constrained beam search. It first derives a fast, query-dependent planning signal via *simultaneous decoding* and uses it to score and shortlist candidate documents. It then uses these planning scores as a *look-ahead bonus* during constrained decoding, favoring prefixes supported by high-scoring planned documents and reducing harmful early pruning.

Why reproduce and stress-test PAG?. Reproduction is particularly valuable for PAG because it targets a central GR failure mode: prefix pruning under trie-constrained beam search. (i) PAG reports improved effectiveness–efficiency trade-offs, achieving strong retrieval with smaller beams by using a single-pass planning signal as a look-ahead bonus during constrained decoding; verifying these gains using the released artifacts and the reported decoding configuration is important for assessing practical utility. (ii) Because the look-ahead bonus is computed from a top-n planning set, guided decoding depends on that set's coverage: prefixes unsupported by planned documents receive no bonus, making the method directly vulnerable to query variation and distribution shift. (iii) GR pipelines involve expensive corpus-side construction (e.g., identifier assignment and trie indexing), motivating tests of whether query-side shift that harms planning coverage or alignment can be mitigated without rebuilding the index. Finally, aggregate metrics can mask *tail risk*: when the planning set omits relevant documents, the look-ahead bonus cannot favor their docid prefixes, making recovery under finite-beam decoding less likely. We therefore instrument the planner with overlap and drift-based diagnostics to characterize when planning provides reliable guidance and how reliability degrades under shift.

Stress tests and scope. We conduct an inference-time reproduction and two stress tests of PAG. First, we evaluate robustness under intent-preserving query variations. Second, we evaluate a stricter *query–document language mismatch* setting by issuing non-English mMARCO queries [2] against the fixed English MS MARCO collection and released identifier trie, without re-indexing. This mismatch setting is a direct test of whether PAG's planning mechanism remains useful when query-side surface form diverges from the evidence space on which the planner and identifier trie were built. For this setting, we evaluate two query-side mitigations with corpus-side artifacts fixed: (i) query-only translation and (ii) trained query-side adaptation via planner-token distillation from aligned English queries.

Our study is organized around three research questions:

RQ1 **Inference-time validation:** To what extent can we reproduce PAG's reported effectiveness and inference-time analyses measurable *without retraining* under the released artifacts and decoding configuration, and what effectiveness–efficiency trends emerge across beam sizes?

RQ2 **Robustness & plan drift:** How does intent-preserving query variation affect planning stability, planned-set overlap, and downstream ranking, relative to strong dense and GR baselines?

RQ3 **Cross-lingual shift with a fixed English index:** Under language mismatch with a fixed English identifier trie, how much performance can be recovered without re-indexing via (a) zero-shot use, (b) query-only translation, and (c) planner-token distillation?

Contributions. (i) We reproduce PAG's inference-time results on MS MARCO Dev and TREC-DL 2019/2020 under the released checkpoint, identifiers, trie, and reported decoding configuration, validating the reported effectiveness and characterizing inference–time behavior under our hardware setup. (ii) We introduce *plan drift* as instability in the planner's top-n candidate set and high--weight planner tokens under intent-preserving query variation, and show how this instability weakens, misaligns, or in some cases collapses planning-guided decoding. (iii) We stress-test PAG under query variation and fixed-index cross-lingual query shift, showing that restoring query-side compatibility through translation is substantially more effective than lightweight planner-token alignment without re-indexing.[1]

2 Related Work

Generative retrieval, constrained decoding, and guidance. Generative retrieval (GR) ranks documents by generating corpus-specific identifiers (docids) rather than scoring documents directly in an embedding space [1, 28, 31, 32, 35]. Because decoding is constrained to valid identifiers (e.g., via a trie), retrieval can be limited by search errors, including *prefix pruning* under finite-beam decoding [7, 25, 27, 36]. A broad response is to augment constrained decoding with auxiliary guidance or look-ahead estimates [8, 10, 18, 22, 26, 33, 37]. Within GR, guidance often combines multiple signals (e.g., lexical/semantic hybrids, ranking-oriented objectives, or alternative generation procedures) to improve decoding reliability [3, 9, 11, 15, 20, 29, 30, 39–41]. PAG instantiates this line of work by using a fast planning-derived look-ahead bonus to bias trie-constrained decoding [37]. In contrast to proposing a new guidance mechanism, we study whether PAG's released planning signal remains reliable under realistic query variation and fixed-index query–document mismatch.

Robustness to query variation and intermediate-signal stability. Intent-preserving query variation (e.g., typos, paraphrases, and reordering) is a standard retrieval stress test and has been operationalized through query-variation generators and UQV-style taxonomies [5, 17, 24]. Such variation can be challenging for pipelines that rely on discrete intermediate predictions or guidance signals, since small surface changes may shift early-stage outputs and propagate downstream [14, 16]. Most robustness evaluations emphasize end-to-end effectiveness [13, 17, 19], with less attention to the stability of intermediate components that steer search. Our *plan drift* analysis addresses this gap for PAG by quantifying changes in

[1] Following ACM's terminology, our study is a combination of an artifact-based *reproducibility study* (for RQ1, we follow the "different team, same setup" mode, as we re-execute released artifacts under the reported setup, without retraining) and a *replicability study* (for RQ2 and RQ3, we follow the "different team, different setup" mode as we conduct stress-tests with query variations and cross-lingual query shifts).

Table 1: Key notation used throughout the reproduced PAG pipeline and our robustness diagnostics.

Symbol	Meaning
c_d	Sequential docid of document d (length L)
t_d	Set-based planning identifier of document d (m tokens)
$s(c_{\le i}; q)$	Sequential prefix score (Eq. 2)
$s_{\text{simul}}(q, d)$	Simultaneous planning score (Eq. 3)
$b(c_{\le i})$	Look-ahead bonus from planned documents
D / $D_n(q)$	Top-n planning set (default n=1000)
k	Beam size (default 100)
m	Planning tokens per document (default 64)
K, ℓ	Truncation depths for overlap diagnostics (default 100)
τ, δ	Stability and drop thresholds for collapse

the planner's candidate set and high-weight planner tokens, and relating these changes to downstream ranking behavior.

Cross-lingual retrieval and query–document mismatch. Cross-lingual retrieval increases query–document mismatch and can weaken methods that rely on surface-form overlap; for example, Bonifacio et al. [2] show that translation artifacts can substantially degrade lexical baselines relative to dense models. While cross-lingual dense retrieval is well studied [21], GR is still commonly evaluated in monolingual settings [11], and multilingual GR efforts have mainly focused on multilingual identifier learning and compression [6]. We instead study PAG under fixed-index query–document language mismatch, where the English identifier space and document-side artifacts remain unchanged, and evaluate query-side mitigations that do not require re-indexing.

3 Reproducibility Methodology

This section formalizes the reproduced PAG inference pipeline and introduces the diagnostics used in our robustness analyses. Table 1 summarizes the main notation used throughout the section.

3.1 Problem Formulation

A GR model ranks documents by autoregressively decoding length-L docids $c_d = [c_{d,1}, \ldots, c_{d,L}]$ conditioned on a query q. Inference uses trie-constrained beam search over valid prefixes. Decoding is scored additively using decoder states $h_i(c_{<i}, q)$ and step-specific docid-token embeddings $E_i[\cdot]$. A complete identifier is scored by

$$s(c_d; q) = \sum_{i=1}^{L} E_i[c_{d,i}] \cdot h_i(c_{d,<i}, q) \tag{1}$$

For a prefix $c_{\le i}$, the corresponding prefix score can be written recursively as

$$s(c_{\le i}; q) = s(c_{<i}; q) + E_i[c_i] \cdot h_i(c_{<i}, q), \qquad s(c_{\le 0}; q) = 0, \tag{2}$$

which is equivalent to $s(c_{\le i}; q) = \sum_{j=1}^{i} E_j[c_j] \cdot h_j(c_{<j}, q)$. Trie-constrained beam search retains the top-k *valid* prefixes at each step and expands them only along trie edges.[2] A key failure mode is *prefix pruning*: once a relevant docid prefix drops out of the beam, trie constraints prevent it from being revisited, even if completing it would yield a highly relevant document. This motivates decoding strategies that incorporate document-level look-ahead guidance.

[2]Equivalently, validity can be enforced by adding a mask $g(c_{\le i})$, with $g = 0$ for valid prefixes and $g = -\infty$ otherwise.

3.2 Planning Ahead in Generative Retrieval

PAG [37] reduces prefix pruning by pairing each document's sequential docid c_d with a set-based docid $t_d = \{t_{d,1}, \ldots, t_{d,m}\}$, an unordered set of m planning tokens drawn from a planning vocabulary. At inference time, PAG combines a fast planning step with planning-guided constrained decoding. Figure 1 summarizes the reproduced PAG pipeline and our probes.

Simultaneous decoding (planning). Given q, one-step simultaneous decoding produces query-dependent token weights $h_q[\cdot]$ over the planning vocabulary. Documents are scored by aggregating weights over t_d:

$$s_{\text{simul}}(q, d) = \sum_{j=1}^{m} h_q[t_{d,j}], \tag{3}$$

and the top-n documents under s_{simul} form the planning set D.

Planning-guided constrained decoding. In trie-constrained beam search over c_d, each valid prefix $c_{\le i}$ receives a look-ahead bonus from compatible planned documents. Let

$$D_{c_{\le i}} = \{d \in D : c_{d,\le i} = c_{\le i}\}, \tag{4}$$

and define

$$b(c_{\le i}) = \begin{cases} \max\limits_{d \in D_{c_{\le i}}} s_{\text{simul}}(q, d) & \text{if } D_{c_{\le i}} \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

PAG scores prefixes for pruning as

$$s'(c_{\le i}; q) = \underbrace{s(c_{\le i}; q)}_{\text{sequential prefix score}} + \underbrace{b(c_{\le i})}_{\text{planning look-ahead bonus}} \tag{6}$$

Beam pruning ranks prefixes using $s'(c_{\le i}; q)$, while expansions remain trie-constrained and add the next-token sequential contribution in Eq. (2). This promotes prefixes that can still complete to highly scored planned documents, reducing early pruning under finite beams.

3.3 PAG Optimization Pipeline

PAG trains a single backbone to support (i) set-based planning scores and (ii) sequential docid decoding, via three stages (see [37] for full objectives and sampling).

- **Stage 1 (set-based planning).** Learn a sparse lexical planner in two steps: (i) train a sparse encoder M_{sp} with a MarginMSE objective and a FLOPs regularizer to produce document token weights $w_d \in \mathbb{R}^{V_T}$; (ii) define set-docids by top-m selection $t_d \leftarrow \text{Top-}m(w_d)$, then fine-tune a set model M_{set} with L_{set} so that query weights $h_q[\cdot]$ yield high $s_{\text{simul}}(q, d) = \sum_{t \in t_d} h_q[t]$ for relevant documents.
- **Stage 2 (sequential decoding).** Build semantic sequential docids via residual quantization: $c_d \leftarrow \text{RQ}(d) \in \mathcal{V}^L$. Train a sequential model M_{seq} as a generative retriever over c_d under trie constraints, using the prefix-oriented loss L_{seq} (supervising intermediate prefixes $c_{d,\le i}$ for $i = 1, \ldots, L$) to reduce prefix pruning during constrained decoding.
- **Stage 3 (unified model).** Combine both capabilities in one model by initializing $M \leftarrow \text{Avg}(M_{\text{set}}, M_{\text{seq}})$ (parameter averaging), then fine-tuning with the joint objective $L_{\text{set}} + L_{\text{seq}}$. To keep simultaneous decoding compatible with the unified backbone, the decoder

Table 2: Plan-drift diagnostics at a glance.

Diagnostic	What it measures
CandOverlap@K	Stability of the top-K planned candidate set
TokJaccard@ℓ	Stability of the top-ℓ planner-token set
$\Delta M_{\text{SimulOnly}}$	Change in planning-only effectiveness under variation
PlanSwapDrop	Effect of using the perturbed plan rather than the clean plan
SeqGain	Gain of full PAG over planning-only retrieval

is additionally conditioned on query tokens while preserving trie-constrained decoding for sequential docids.

3.4 Plan Stability, Plan Sensitivity, and Plan Collapse

For each query q, PAG's planning stage produces a top-n candidate set $D_n(q)$ (top-n documents under $s_{\text{simul}}(q,d)$) and a planner token set $P_\ell(q)$, the top-ℓ planning-vocabulary tokens under query weights $h_q[\cdot]$. Given an original query q and an intent-preserving variation $\tilde{q}$, we quantify plan stability and sensitivity via overlap between $D_K(q)$ and $D_K(\tilde{q})$ and between $P_\ell(q)$ and $P_\ell(\tilde{q})$, and report plan collapse rates under a thresholded criterion. Unless otherwise stated, we use $K = 100$ and $\ell = 100$ for overlap diagnostics, and set $n = 1000$ as the default Stage-1 candidate-pool size used during inference, following PAG's reported setting. Table 2 summarizes the plan-drift diagnostics defined in this subsection.

Candidate-set stability. Let $D_K(q)$ denote the top-K truncation of the Stage-1 candidate pool $D_n(q)$ (with $K \ll n$), and let $I_K(q,\tilde{q}) = |D_K(q) \cap D_K(\tilde{q})|$. We report:

$$\text{CandOverlap@}K(q,\tilde{q}) = \frac{I_K(q,\tilde{q})}{K} \tag{7}$$

Planner-token stability. Let $J_\ell(q,\tilde{q}) = |P_\ell(q) \cap P_\ell(\tilde{q})|$. We report token-set Jaccard similarity:

$$\text{TokJaccard@}\ell(q,\tilde{q}) = \frac{J_\ell(q,\tilde{q})}{|P_\ell(q) \cup P_\ell(\tilde{q})|} \tag{8}$$

We compute these per query and summarize them using mean, median, and tail quantiles (p10, p25, p75, p90).

Planning-only retrieval (SimulOnly). Let $M(\cdot)$ denote the evaluation metric (MRR@10 for MS MARCO Dev, NDCG@10 for TREC-DL). SimulOnly ranks documents by $s_{\text{simul}}(q,d)$; we report the metric change under variation:

$$\Delta M_{\text{SimulOnly}}(q,\tilde{q}) = M_{\text{SimulOnly}}(\tilde{q}) - M_{\text{SimulOnly}}(q) \tag{9}$$

Plan sensitivity (counterfactual plan swap). To isolate sensitivity to look-ahead guidance from the intrinsic difficulty of $\tilde{q}$, we decode $\tilde{q}$ twice under identical sequential decoding and trie constraints: (i) with its own plan (*normal*) and (ii) using the clean-query plan computed for q (*swapped*). This is a counterfactual diagnostic rather than a retrieval setting: it reuses a planning signal from a different query to isolate the effect of guidance quality.

$$\text{PlanSwapDrop}(q,\tilde{q}) = M_{\text{PAG}}(\tilde{q};\text{normal}) - M_{\text{PAG}}(\tilde{q};\text{swapped}) \tag{10}$$

Negative values indicate that the clean-query plan improves effectiveness on $\tilde{q}$ (i.e., the perturbed plan is harmful).

Guided decoding gain over planning. We report the marginal gain of full PAG over planning-only retrieval on $\tilde{q}$:

$$\text{SeqGain}(\tilde{q}) = M_{\text{PAG}}(\tilde{q}) - M_{\text{SimulOnly}}(\tilde{q}) \tag{11}$$

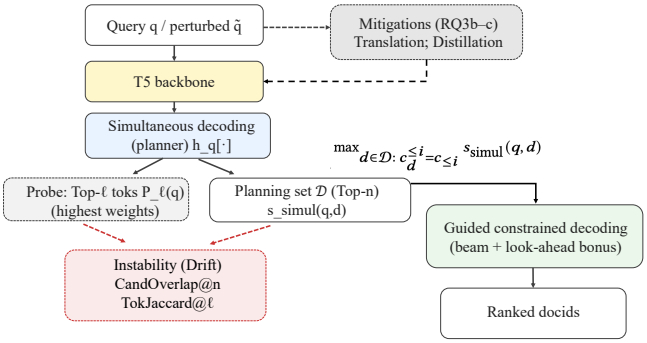

Figure 1: Compact PAG pipeline with probes. Simultaneous decoding yields planner weights $h_q[\cdot]$, inducing a top-n candidate pool used to compute a look-ahead bonus during trie-constrained decoding. Red marks our diagnostics: candidate drift (CandOverlap@100) and token drift (TokJaccard@100). The top-right dashed box summarizes RQ3 query-side mitigations (translation and planner alignment) with corpus-side artifacts fixed.

Low-stability tail events ("plan collapse"). We define *plan collapse* as a query-level tail event where planner stability is low and planning-only effectiveness drops sharply. Using a percentile-based stability threshold τ and an effectiveness-drop threshold δ, a query is flagged as collapsed if

$$\begin{aligned}\big(\text{CandOverlap@}K < \tau \;\vee\; \text{TokJaccard@}\ell < \tau\big) \\ \wedge\; \Delta M_{\text{SimulOnly}}(q,\tilde{q}) \le -\delta\end{aligned} \tag{12}$$

Unless otherwise stated, τ is computed per condition (split × variation × seed) as the 10th percentile of the CandOverlap@K distribution. We report sensitivity by sweeping the τ-percentile and the absolute-drop threshold δ, and we additionally report lower-tail quantiles (e.g., p10/p25) of the stability metrics.

4 Experimental Setup

Reproducibility scope. Our experiments rely on the authors' released checkpoint and corpus-side artifacts (docids and trie). When intermediate artifacts needed to reconstruct a training stage are unavailable, we treat the corresponding components as fixed and explicitly mark results that would require retraining to reproduce. Unless stated otherwise, RQ1–RQ2 use the released PAG artifacts under the reported inference-time decoding configuration, while RQ3 evaluates fixed-index cross-lingual query shift and is reported separately from artifact-level reproduction claims.

4.1 Released artifacts and experimental scopes

Inference-time reproducibility (RQ1, RQ2). We evaluate PAG using the released T5-base checkpoint, the trie built over 8.8M sequential passage identifiers ($L = 8$, $V = 2048$), and the stored set-based identifiers used for simultaneous planning scores $s_{\text{simul}}(q,d)$. We follow the reported decoding procedure without modifying document identifiers.

Cross-lingual query shift (RQ3). We issue non-English mMARCO queries against the fixed English MS MARCO passage collection using the same released docids and trie. This setting evaluates

query–corpus language mismatch without re-indexing. As query-side mitigations, we evaluate translation into English before planning and decoding, and a learned planner-alignment model. For translation, we use M2M100 [4].[3]

4.2 Datasets and metrics

We use the MS MARCO passage retrieval benchmark (8.8M passages) and report results on MS MARCO Dev (6,980 queries) and TREC-DL 2019/2020. Following the original paper, we report MRR@10 on MS MARCO Dev, NDCG@10 on TREC-DL, and Recall@10 on all datasets. For RQ3, we use non-English mMARCO query sets while keeping the document corpus and relevance judgments fixed to the English MS MARCO passage collection.

4.3 Model and identifier configuration

Experiments use the released T5-base checkpoint and fixed corpus-side artifacts. Each document d has (i) a sequential docid $c_d = [c_{d,1}, \ldots, c_{d,L}]$ constructed via residual quantization (RQ) with $L = 8$ and docid-token vocabulary size $V = 2048$, and (ii) a set-based identifier $t_d = \{t_{d,1}, \ldots, t_{d,m}\}$ with $m = 64$ planning tokens.

4.4 Inference and decoding settings

Retrieval uses trie-constrained beam search with the planning-guided prefix score in Eq. 6. Unless otherwise stated, inference follows the paper's default configuration: beam size $k = 100$ and a planning set formed by the top $n = 1000$ documents under $s_{\text{simul}}(q, d)$.

4.5 Query variations

For RQ2, we generate perturbed queries offline and keep them fixed across runs. Variations follow the UQV taxonomy [13, 24]: *misspelling*, *reordering*, *synonym*, *paraphrase*, and *naturality*. We instantiate variations with five seeds (1999, 5, 27, 2016, 2026) and report mean±std across the resulting variation sets.

4.6 Cross-lingual query shift and adaptations

For RQ3, we evaluate four languages (Chinese, Dutch, French, and German) using released mMARCO query sets.[4] These languages induce different sources of mismatch with the English planning vocabulary: Chinese introduces a script mismatch; German and Dutch exhibit richer morphology; French shares script but differs lexically.

RQ3 baselines (inference-only). We compare:

(1) **Naive cross-lingual PAG:** apply the released English PAG pipeline directly to non-English queries.
(2) **Sequential-only:** disable the planning bonus and decode using only the autoregressive score $s(c_{\le i}; q)$.
(3) **Translate-at-inference:** translate non-English queries into English, then run the unmodified English PAG pipeline.

Query-side adaptation: planner alignment (no re-indexing). As a lightweight learned mitigation, we align non-English queries to the released English planner using paired mMARCO queries with the same query id. Let q^{en} be an English query and q^{ℓ} its paired non-English query for $\ell \in \{\texttt{nl}, \texttt{fr}, \texttt{de}, \texttt{zh}\}$. We use the released English PAG checkpoint as a frozen teacher and initialize a student from the same checkpoint; training qids are disjoint from evaluation, which uses the same dev qids as our inference-only baselines. All document-side artifacts remain fixed (t_d, c_d, trie, and released index files), i.e., no re-indexing.

For each pair, the teacher is scored on q^{en} and the student on q^{ℓ} over the *English planning vocabulary*. Concretely, planner scores for vocabulary token v are

$$z[v] = \max_t \Big(\log(1 + \text{ReLU}(\texttt{lexical\_logit}_t[v])) \cdot m_t \Big),$$

where m_t is the attention mask. We update only query-side student parameters (encoder and `lm_head`), while keeping the decoder frozen.

We optimize temperature-scaled distillation:

$$\mathcal{L}_{\text{align}} = \tau^2 \, \text{KL}\big(\text{softmax}\big(\tfrac{z_{\text{teach}}}{\tau}\big) \,\|\, \text{softmax}\big(\tfrac{z_{\text{stud}}}{\tau}\big)\big), \qquad (13)$$

with $\tau = 2.0$. For efficiency, we compute KL on $U = \text{TopK}_{\text{teach}} \cup \text{TopK}_{\text{stud}}$ with $K = 100$ and renormalize on U. We train with AdamW ($\text{lr} = 10^{-5}$) for 5 epochs (effective batch size 32) and select checkpoints by dev TokJaccard@100. Intuitively, alignment encourages non-English queries to activate the same English planning evidence as their English counterparts while keeping the retrieval index unchanged.

5 Results and Analysis

We organize the empirical results around the three research questions introduced in Section 1. We first assess whether PAG's released artifacts suffice to reproduce its reported inference-time effectiveness and efficiency trends (RQ1). We then analyze how intent-preserving query variation affects planning stability and downstream ranking using the plan-drift diagnostics from Section 3.4 (RQ2). Finally, we evaluate fixed-index cross-lingual query shift and compare query-side mitigation strategies that preserve the released English index and document-side artifacts (RQ3).

5.1 RQ1: Inference-time reproducibility

RQ1 evaluates inference-time reproducibility of PAG using released artifacts under the reported decoding configuration, focusing on effectiveness, decoding-time trade-offs, and inference-time ablations.

Effectiveness. Table 3 compares reported effectiveness with results obtained using the released PAG checkpoint and corpus-side artifacts at the default setting ($k = 100$, $m = 64$; cf. original Table 1). Across MS MARCO Dev and TREC-DL 2019/2020, reproduced PAG matches the reported values within 0.002 absolute difference (three-decimal precision), indicating that the released artifacts suffice to reproduce the headline effectiveness in this setting. For context only (i.e., neither part of the original PAG evaluation nor a reproduction target), we report results for three recent dense retrievers spanning model scales (Nomic-v2 [23], EmbeddingGemma [34], and Qwen3-8B [38]), next to the dense references from the original paper, using a uniform brute-force scoring pipeline over the full corpus. These contextual baselines calibrate PAG's effectiveness and index-footprint trade-off relative to dense retrieval across model scales.

[3] Implementation via the Transformers M2M100 model documentation: https://huggingface.co/docs/transformers/en/model_doc/m2m_100.
[4] https://huggingface.co/datasets/unicamp-dl/mmarco.

Table 3: PAG effectiveness reproduction (cf. original Table 1) with dense-retrieval baselines for context. $^{\uparrow/\downarrow}$ indicate significantly higher/lower than reproduced PAG (t-test with Bonferroni correction, $p < 0.01$). Dense baselines are scored by brute-force over the full corpus; we separate baselines reported in [37] from our additional runs.

Method	KD	Index Mem. (GB)	MS MARCO Dev		TREC-DL 2019		TREC-DL 2020	
			MRR@10	Recall@10	NDCG@10	Recall@10	NDCG@10	Recall@10
Dense retrieval baselines reported in [37] (for reference)								
TCT-ColBERT	✓	25.30	$0.335^{\downarrow}$	$0.596^{\downarrow}$	$0.670^{\downarrow}$	$0.240^{\downarrow}$	$0.668^{\downarrow}$	$0.218^{\downarrow}$
MarginMSE	✓	25.30	$0.325^{\downarrow}$	$0.581^{\downarrow}$	$0.699^{\downarrow}$	$0.250^{\downarrow}$	$0.645^{\downarrow}$	$0.203^{\downarrow}$
TAS-B	✓	25.30	$0.344^{\downarrow}$	$0.622^{\downarrow}$	$0.717^{\uparrow}$	$0.255^{\downarrow}$	$0.685^{\downarrow}$	$0.230^{\downarrow}$
CL-DRD	✓	25.30	$0.382^{\downarrow}$	$0.651^{\downarrow}$	$0.725^{\uparrow}$	0.266	$0.687^{\downarrow}$	$0.216^{\downarrow}$
Additional dense retrievers (our runs; contextual reference only)								
Nomic-v2	✗	25.30	$0.341^{\downarrow}$	$0.616^{\downarrow}$	0.705	$0.154^{\downarrow}$	$0.684^{\downarrow}$	$0.227^{\downarrow}$
EmbeddingGemma	✗	25.30	$0.325^{\downarrow}$	$0.591^{\downarrow}$	$0.730^{\uparrow}$	$0.168^{\downarrow}$	$0.721^{\uparrow}$	$0.231^{\downarrow}$
Qwen3-8B	✗	135.10	$0.368^{\downarrow}$	0.670	$0.763^{\uparrow}$	$0.178^{\downarrow}$	$0.743^{\uparrow}$	$0.249^{\uparrow}$
Generative retrieval baselines								
NOVO	✗	0.80	0.126	0.242	0.258	0.112	0.310	0.140
MINDER	✗	12.16	0.186	0.383	0.506	0.201	0.392	0.144
LTRGR	✗	12.16	0.255	0.531	0.598	0.238	0.553	0.182
RIPOR	✓	1.06	0.333	0.562	0.628	0.205	0.631	0.191
PAG (reported)	✓	3.27	0.385	0.670	0.705	0.267	0.700	0.236
PAG (reproduced)	✓	3.27	0.386	0.671	0.703	0.265	0.701	0.236

Table 4: RQ1: MS MARCO Dev effectiveness–efficiency trade-offs (cf. original Table 3) across planner set size m (tokens/doc) and beam size k. Latency values are hardware-specific (paper: A100 80GB; ours: H100 96GB). – denotes settings requiring unreleased artifacts; † marks reproduced effectiveness below the paper.

		MRR@10		Recall@10		Index mem. (GB)		Simul. QL (ms)		Seq. QL (ms)	
m	k	Reported	Reproduced	Reported	Reproduced	Reported	Computed	Reported	Computed	Reported	Computed
16	10	0.342	$0.330^{\dagger}$	0.577	$0.556^{\dagger}$	1.30	1.13	20	4	44	42
32	10	0.367	$0.360^{\dagger}$	0.626	$0.608^{\dagger}$	1.94	2.26	22	6	44	43
64	10	0.379	0.380	0.641	0.644	3.27	4.53	25	8	44	46
128	10	0.386	–	0.645	–	5.96	–	31	–	44	–
16	100	0.355	$0.341^{\dagger}$	0.620	$0.606^{\dagger}$	1.30	1.13	20	4	250	262
32	100	0.372	$0.368^{\dagger}$	0.652	$0.644^{\dagger}$	1.94	2.26	22	6	250	263
64	100	0.385	0.386	0.670	0.671	3.27	4.53	25	8	250	266
128	100	0.390	–	0.664	–	5.96	–	31	–	250	–

Efficiency and beam–latency trade-offs. Table 4 characterizes PAG's effectiveness–efficiency trends on MS MARCO Dev (cf. original Table 3) as a function of beam size k and planner set size m. At the artifact-supported setting $m = 64$, reproduced effectiveness matches the reported values at both $k = 10$ and $k = 100$ (MRR@10 within 0.001; Recall@10 within 0.003). Increasing beam size k consistently improves effectiveness while substantially increasing sequential-decoding latency (Seq. QL 46.8 $\rightarrow$ 268.6 ms for $k = 10 \rightarrow 100$), matching the original qualitative trade-off. However, absolute efficiency values for simultaneous decoding and index memory differ from the paper: at $m = 64$ we measure Simul. QL = 8.3 ms (vs. 25 ms reported) and Index Mem. = 4.53 GB (vs. 3.27 GB). We therefore emphasize *within-table trends* over absolute cross-hardware comparisons. The release provides only the top-64 planner tokens per document, enabling direct evaluation at $m = 64$; $m = 128$ is marked –. For $m \in \{16, 32\}$, we report a pragmatic approximation by truncating the top-64 token lists. The released lists are ordered by decreasing planning weight, so truncation retains the highest-weight planning tokens and provides a reasonable proxy for smaller-m settings. Under truncation, effectiveness at $m \in \{16, 32\}$ is lower than reported, but qualitative trends in m and k are preserved.

Ablations Table 5 reproduces inference-time ablations achievable with the released checkpoint and fixed identifier/trie artifacts (others are –). Reproduced values closely match the reported ones and isolate the role of planning-ahead guidance: *relative to our reproduced PAG scores*, removing the look-ahead term (w/o adding s_{simul}) reduces effectiveness by 0.036 MRR@10 and 0.058 Recall@10, while planning-only retrieval (Only s_{simul}) drops further (0.083 MRR@10, 0.102 Recall@10). Overall, these ablations are consistent with PAG's intended use of the planner as look-ahead guidance for finite-beam sequential decoding rather than as a standalone retriever.

Table 5: Ablation on MS MARCO Dev (cf. Table 2 in the original paper). – denotes variants requiring retraining or unreleased checkpoints.

	MRR@10		Recall@10		Index mem. (GB)	
Variant	Reported	Reproduced	Reported	Reproduced	Reported	Computed
PAG	0.385	0.386	0.670	0.671	3.27	4.53
1. w/o adding $s_{\text{simul}}(\cdot)$	0.349	0.350	0.614	0.613	0.50	0.50
2. Only $s_{\text{simul}}(\cdot)$ for retrieval	0.303	0.303	0.569	0.569	2.77	2.77
3. w/o seq2seq pre-training	0.381	–	0.660	–	3.27	–
4. w/o multi-obj. learning	0.380	–	0.663	–	3.27	–
5. Only M^{set}	0.322	–	0.606	–	2.77	–
6. Only M^{seq}	0.339	–	0.566	–	0.50	–
7. Linear interp. of M^{set} and M^{seq}	0.360	–	0.593	–	3.27	–
8. Only M^{sp}	0.378	–	0.667	–	35.28	–
9. Only M^{ds}	0.365	–	0.641	–	25.30	–

Table 6: RQ2: Retrieval under query variations. Stage 1 ranks by s_{simul}; Stage 2 is full PAG. Values are $\mu \pm \sigma$ across five seeds, with degradation $\Delta = M(q) - M(\tilde{q})$ in parentheses. Primary metrics: MRR@10 (Dev) and NDCG@10 (DL19/20).

		Stage 1: SimulOnly (Planner)		Stage 2: PAG (End-to-end)	
Split	**variation**	**NDCG@10 (Δ)**	**MRR@10 (Δ)**	**NDCG@10 (Δ)**	**MRR@10 (Δ)**
DL19	Clean	0.643 ± 0.000 (–)	0.898 ± 0.000 (–)	0.669 ± 0.000 (–)	0.915 ± 0.000 (–)
	Misspelling	0.407 ± 0.020 (0.236 ± 0.020)	0.637 ± 0.059 (0.260 ± 0.059)	0.452 ± 0.012 (0.217 ± 0.012)	0.726 ± 0.034 (0.189 ± 0.034)
	Reordering	0.628 ± 0.013 (0.015 ± 0.013)	0.899 ± 0.015 (-0.002 ± 0.015)	0.654 ± 0.009 (0.014 ± 0.009)	0.899 ± 0.007 (0.016 ± 0.007)
	Synonymizing	0.454 ± 0.037 (0.189 ± 0.037)	0.657 ± 0.040 (0.241 ± 0.040)	0.526 ± 0.018 (0.143 ± 0.018)	0.786 ± 0.023 (0.129 ± 0.023)
	Paraphrasing	0.557 ± 0.030 (0.086 ± 0.030)	0.798 ± 0.048 (0.099 ± 0.048)	0.596 ± 0.012 (0.073 ± 0.012)	0.871 ± 0.014 (0.044 ± 0.014)
	Naturalizing	0.638 ± 0.000 (0.005 ± 0.000)	0.932 ± 0.000 (-0.034 ± 0.000)	0.626 ± 0.000 (0.043 ± 0.000)	0.869 ± 0.000 (0.046 ± 0.000)
DL20	Clean	0.638 ± 0.000 (–)	0.930 ± 0.000 (–)	0.621 ± 0.000 (–)	0.869 ± 0.000 (–)
	Misspelling	0.420 ± 0.015 (0.217 ± 0.015)	0.678 ± 0.029 (0.252 ± 0.029)	0.461 ± 0.019 (0.161 ± 0.019)	0.703 ± 0.040 (0.166 ± 0.040)
	Reordering	0.627 ± 0.004 (0.010 ± 0.004)	0.919 ± 0.012 (0.010 ± 0.012)	0.607 ± 0.009 (0.014 ± 0.009)	0.848 ± 0.012 (0.021 ± 0.012)
	Synonymizing	0.480 ± 0.042 (0.158 ± 0.042)	0.711 ± 0.066 (0.219 ± 0.066)	0.508 ± 0.007 (0.114 ± 0.007)	0.741 ± 0.034 (0.129 ± 0.034)
	Paraphrasing	0.515 ± 0.017 (0.123 ± 0.017)	0.777 ± 0.014 (0.152 ± 0.014)	0.512 ± 0.019 (0.109 ± 0.019)	0.738 ± 0.039 (0.131 ± 0.039)
	Naturalizing	0.615 ± 0.000 (0.023 ± 0.000)	0.945 ± 0.000 (-0.016 ± 0.000)	0.598 ± 0.000 (0.023 ± 0.000)	0.869 ± 0.000 (0.000 ± 0.000)
Dev	Clean	0.364 ± 0.000 (–)	0.315 ± 0.000 (–)	0.410 ± 0.000 (–)	0.362 ± 0.000 (–)
	Misspelling	0.220 ± 0.002 (0.145 ± 0.002)	0.190 ± 0.002 (0.125 ± 0.002)	0.245 ± 0.003 (0.165 ± 0.003)	0.215 ± 0.003 (0.147 ± 0.003)
	Reordering	0.355 ± 0.001 (0.009 ± 0.001)	0.307 ± 0.001 (0.008 ± 0.001)	0.399 ± 0.001 (0.011 ± 0.001)	0.350 ± 0.001 (0.012 ± 0.001)
	Synonymizing	0.260 ± 0.001 (0.104 ± 0.001)	0.225 ± 0.001 (0.090 ± 0.001)	0.305 ± 0.002 (0.105 ± 0.002)	0.268 ± 0.002 (0.094 ± 0.002)
	Paraphrasing	0.297 ± 0.001 (0.068 ± 0.001)	0.257 ± 0.001 (0.059 ± 0.001)	0.342 ± 0.003 (0.069 ± 0.003)	0.300 ± 0.002 (0.062 ± 0.002)
	Naturalizing	0.341 ± 0.000 (0.023 ± 0.000)	0.294 ± 0.000 (0.021 ± 0.000)	0.388 ± 0.000 (0.022 ± 0.000)	0.342 ± 0.000 (0.020 ± 0.000)

5.2 RQ2: Robustness stress test

RQ2 stress-tests PAG under intent-preserving query variation while holding the released checkpoint, docids, trie, and decoding configuration fixed. This isolates query-side sensitivity in Stage 1 planning (*SimulOnly*, ranking by s_{simul}) and in Stage 2 planning-guided decoding.

Query-variation evaluation. We evaluate five intent-preserving variations of each clean query q. Table 6 shows that lexical variations cause large end-to-end degradation, while reordering is near-invariant. On DL19, Stage 2 drops by 0.217 NDCG (misspelling), 0.143 (synonym), and 0.073 (paraphrase), versus 0.014 for reordering; DL20 shows the same ranking (0.161/0.114/0.109 vs. 0.014). These effectiveness gaps align with plan stability in Table 7: reordering has the highest overlap (CandOverlap@100 ≈0.80; TokJaccard@100 ≈0.68 on DL19/20), whereas misspelling and synonym sharply reduce overlap (e.g., DL19 CandOverlap 0.374/0.439; TokJaccard 0.352/0.422; DL20 CandOverlap 0.348/0.498; TokJaccard 0.363/0.485), and the boxplots (Fig. 2–3) show pronounced lower tails under lexical corruption, indicating that a non-trivial subset of queries undergoes near-replacement of the planned set. A plausible contributor to plan collapse under typos is *subword fragmentation*: small edits can change SentencePiece segmentation, producing rare/mismatched units that fail to fire the lexical planner's sparse triggers and sharply reduce planned-set overlap. Consistent with PAG's mechanism, this instability limits Stage 2 because the look-ahead bonus is computed from a shifted candidate pool, reducing coverage of relevant docid prefixes and increasing pruning risk. Stage 2 improvements are therefore conditional: SeqGain is positive for most Dev conditions (0.043–0.048) and for DL19 under misspelling/synonym/paraphrase (0.045/0.071/0.039 NDCG), but becomes negative in several DL20 settings despite relatively high

Table 7: Planner stability and plan swapping (defined in §3.4). Mean±std over five seeds; PlanSwapDrop < 0 implies the clean plan helps.

		Plan Stability (↑)		Plan Sensitivity / Gain	
Split	**Perturbation**	**CandOverlap@100**	**TokJaccard@100**	**SeqGain (MRR/NDCG)**	**PlanSwapDrop (MRR/NDCG)**
DL19	Misspelling	0.374 ± 0.018	0.352 ± 0.015	0.089 ± 0.035 / 0.045 ± 0.014	-0.070 ± 0.028 / -0.058 ± 0.011
	Ordering	0.801 ± 0.015	0.683 ± 0.019	-0.000 ± 0.014 / 0.026 ± 0.011	0.000 ± 0.000 / -0.004 ± 0.004
	Synonym	0.439 ± 0.021	0.422 ± 0.023	0.130 ± 0.019 / 0.071 ± 0.023	-0.033 ± 0.014 / -0.026 ± 0.008
	Paraphrase	0.625 ± 0.018	0.551 ± 0.009	0.073 ± 0.040 / 0.039 ± 0.027	-0.015 ± 0.010 / -0.016 ± 0.010
	Naturality	0.738 ± 0.000	0.636 ± 0.000	-0.063 ± 0.000 / -0.012 ± 0.000	0.000 ± 0.000 / -0.004 ± 0.000
DL20	Misspelling	0.348 ± 0.018	0.363 ± 0.014	0.025 ± 0.054 / 0.041 ± 0.019	-0.013 ± 0.022 / -0.017 ± 0.016
	Ordering	0.801 ± 0.004	0.684 ± 0.008	-0.071 ± 0.018 / -0.020 ± 0.011	0.006 ± 0.005 / 0.004 ± 0.004
	Synonym	0.498 ± 0.053	0.485 ± 0.032	0.030 ± 0.080 / 0.028 ± 0.045	-0.024 ± 0.017 / -0.018 ± 0.007
	Paraphrase	0.569 ± 0.021	0.522 ± 0.014	-0.039 ± 0.032 / -0.003 ± 0.010	-0.025 ± 0.008 / -0.011 ± 0.003
	Naturality	0.708 ± 0.000	0.577 ± 0.000	-0.076 ± 0.000 / -0.017 ± 0.000	-0.000 ± 0.000 / -0.008 ± 0.000
Dev	Misspelling	0.312 ± 0.002	0.336 ± 0.002	0.025 ± 0.003 / 0.026 ± 0.003	-0.015 ± 0.001 / -0.016 ± 0.001
	Ordering	0.767 ± 0.002	0.661 ± 0.002	0.043 ± 0.001 / 0.044 ± 0.001	-0.001 ± 0.000 / -0.001 ± 0.000
	Synonym	0.460 ± 0.002	0.475 ± 0.001	0.042 ± 0.002 / 0.045 ± 0.002	-0.014 ± 0.000 / -0.015 ± 0.000
	Paraphrase	0.573 ± 0.002	0.536 ± 0.002	0.043 ± 0.002 / 0.045 ± 0.003	-0.007 ± 0.000 / -0.007 ± 0.001
	Naturality	0.680 ± 0.000	0.584 ± 0.000	0.048 ± 0.000 / 0.048 ± 0.000	-0.001 ± 0.000 / -0.001 ± 0.000

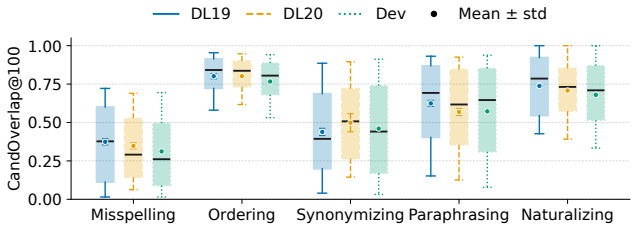

Figure 2: Candidate-set stability (CandOverlap@100). CandOverlap@100 compares the planner top-100 candidate sets for clean q vs. perturbed $\tilde{q}$. Line styles denote splits (DL19 solid, DL20 dashed, Dev dotted). Across five seeds: whiskers $p10$–$p90$, boxes $p25$–$p75$, median line, mean ± std marker (Naturality: std = 0).

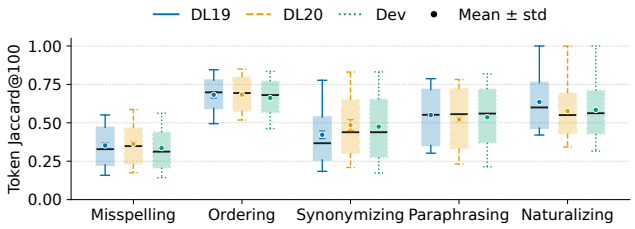

Figure 3: Planner-token stability (TokJaccard@100). TokJaccard@100 compares top-100 planner tokens for clean q vs. perturbed $\tilde{q}$; same plotting convention as Fig. 2.

overlap (e.g., Reordering CandOverlap 0.801 and TokJaccard 0.684, yet SeqGain = −0.020 NDCG), indicating residual sensitivity in the sequential component beyond plan membership.

Finally, plan swapping isolates causality: PlanSwapDrop is typically negative under harder variations (e.g., DL19 misspelling −0.058 NDCG and −0.070 MRR), showing that reusing the clean bonus partially recovers performance, but the magnitude is smaller than the total losses in Table 6, implying that robustness is bounded by both planning instability (especially tail events) and sequential-decoding sensitivity under perturbed inputs.

Table 8: Plan collapse analysis. Collapse rate (mean±std over five seeds) under the criterion from §3.4 ($\delta = 0.05$, $\tau = \text{p10}$ of CandOverlap@100 per seed-condition). We also report the seed-mean threshold $\mathbb{E}[\tau]$ (std. omitted for brevity).

Split	Variation	Collapse Rate (%)	Threshold (τ)
MS MARCO Dev	Misspelling	5.2 ± 0.1	0.015
	Ordering	3.2 ± 0.8	0.531
	Synonym	5.7 ± 0.1	0.032
	Paraphrase	6.2 ± 0.1	0.078
	Naturality	4.5 ± 0.0	0.333
TREC-DL 2019	Misspelling	11.2 ± 1.0	0.015
	Ordering	7.4 ± 3.0	0.580
	Synonym	11.6 ± 0.0	0.040
	Paraphrase	10.7 ± 1.3	0.152
	Naturality	2.3 ± 0.0	0.427
TREC-DL 2020	Misspelling	9.6 ± 1.5	0.063
	Ordering	10.0 ± 1.7	0.617
	Synonym	10.4 ± 2.5	0.144
	Paraphrase	11.1 ± 0.0	0.126
	Naturality	11.1 ± 0.0	0.392

Plan-collapse. Table 8 reports seed-aggregated collapse rates over 75 (split, variation, seed) conditions (176,925 query instances). Collapse is infrequent on MS MARCO Dev (3.2–6.2%) but higher on TREC-DL (2.3–11.6% on DL19; 9.6–11.1% on DL20), indicating a non-trivial subset of queries with both low stability and a planning-only drop ($\Delta M_{\text{SimulOnly}} \leq -0.05$). Ordering exhibits a large low-stability tail (19.0–33.3%; Table 7) yet comparatively lower collapse, suggesting that overlap deviations under ordering often do not coincide with large planning-only drops at $\delta = 0.05$. By contrast, misspelling/synonym/paraphrase yield 9.6–11.6% collapse on TREC-DL, indicating tail events where planning becomes both unstable and planning-only effectiveness degrades.

Comparison with dense and GR baselines. Fig. 4 shows that robustness is primarily limited by *lexical disruption* rather than benign rewrites: misspellings and synonym substitutions strongly affect both PAG/RIPOR and the dense baseline TAS-B, while recent

dense retrievers are typically less impacted across MS MARCO Dev and TREC-DL 2019/2020. Under misspellings, RIPOR/TAS-B/PAG drop by ∼48%/44%/41% on MS MARCO Dev, versus ∼18–26% for recent dense models, and the same ordering holds on DL19/20. Reordering is near-invariant for all methods (single-digit drops), indicating brittleness is driven by surface-form mismatch rather than word order. Synonym substitution shows a similar pattern: recent dense models drop ∼16–22%, while RIPOR/PAG typically drop ∼18–30%, placing PAG closer to RIPOR than to recent dense retrievers under lexical perturbations. Overall, this comparison aligns with RQ2–RQ3: robustness failures concentrate on lexical surface-form shifts, which in RQ2 coincide with planner drift and reduced candidate stability that can limit Stage 2 gains under finite-beam decoding. RQ3 confirms the same sensitivity under stronger surface-form mismatch (cross-lingual queries) with a fixed index and evaluates query-side mitigations.

5.3 RQ3: Cross-lingual query shift

RQ2 showed that lexical surface-form variation can destabilize the planning signal and weaken downstream decoding. RQ3 tests this failure mode under a more extreme setting: *cross-lingual query shift*, where query surface form diverges sharply from the English planning vocabulary on which PAG was trained, while the document collection and all document-side artifacts remain fixed to the released English index (set-based identifiers t_d, sequential docids c_d, and the trie). This setting is *not* studied in the original paper, so we report it as a stress test rather than a reproduction claim. It is also a direct test of PAG's original motivation: if the planner cannot recover a useful candidate pool under fixed-index mismatch, then the look-ahead bonus cannot protect relevant prefixes from early pruning.

Setting. We issue mMARCO queries in four languages (nl, fr, de, zh) against the fixed English MS MARCO passage collection (Section 4.6), keeping decoding and all document-side artifacts unchanged. For zh, the released English tokenizer/vocabulary can tokenize characters lossily, so *Naive* and *Aligned* reflect both cross-lingual shift and vocabulary-coverage failure, while *Translate* mitigates this by converting queries to English before tokenization.

We compare four query-side settings: (i) *Naive*, which applies the released English PAG pipeline directly to non-English queries; (ii) *Seq-only*, which disables planning guidance and decodes using only $s(c_{\leq i}; q)$; (iii) *Translate*, which translates queries into English at inference time and then runs the unchanged English PAG pipeline; and (iv) *Aligned*, which fine-tunes only query-side parameters to match the released English planner-token distribution on paired $(q^{\text{en}}, q^{\ell})$ queries, without re-indexing or modifying t_d, c_d, or the trie.

Headline result. Under a fixed English index, naive cross-lingual transfer substantially degrades planning-guided decoding because non-English queries often fail to activate the English planner evidence that supplies the look-ahead bonus. Query translation provides the strongest recovery across languages, while planner-token alignment yields only partial gains because improvements in token overlap do not consistently translate into candidate-set alignment.

Result analysis. Naive transfer. *Naive* transfer yields low Stage 2 MRR@10 across languages (nl 0.090, fr 0.097, de 0.102, zh 0.027; Fig. 5) and coincides with weak agreement to the English reference (TokJaccard@100 0.072–0.102; CandOverlap@100 0.124–0.176; Table 9). This indicates that non-English queries often fail to activate the English planner-token evidence and top-100 planned candidates used to supply the look-ahead bonus.

Table 9: *RQ3: Planner overlap diagnostics under cross-lingual query shift*. Overlap between each q^{ℓ} and its English reference q^{en} (same query id). p10(TokJaccard@100) reports the 10th-percentile (lower-tail) token overlap.

Language	Diagnostic	Naive	Aligned	Translate
Dutch (nl)	TokJaccard@100	0.101	0.134	0.430
	CandOverlap@100	0.170	0.121	0.563
	p10(TokJaccard@100)	0.020	0.042	0.124
French (fr)	TokJaccard@100	0.102	0.182	0.402
	CandOverlap@100	0.176	0.176	0.537
	p10(TokJaccard@100)	0.031	0.070	0.117
German (de)	TokJaccard@100	0.090	0.194	0.408
	CandOverlap@100	0.154	0.171	0.543
	p10(TokJaccard@100)	0.026	0.070	0.124
Chinese (zh)	TokJaccard@100	0.072	0.082	0.317
	CandOverlap@100	0.124	0.029	0.449
	p10(TokJaccard@100)	0.020	0.020	0.087

Translation. *Translate* provides the strongest recovery (Stage 2 MRR@10: nl 0.230, fr 0.221, de 0.224, zh 0.160), improving over *Naive* by +0.140/+0.124/+0.122/+0.133 MRR@10, respectively (Fig. 5). It also substantially increases overlap with the English reference (e.g., TokJaccard@100 nl 0.101→0.430 and zh 0.072→0.317; CandOverlap@100 nl 0.170→0.563 and zh 0.124→0.449; Table 9), consistent with translation restoring compatibility with the fixed English planning vocabulary and candidate coverage.

Sequential-only ablation. Removing the planner prior is consistently harmful: *Seq-only* is uniformly poor (Stage 2 MRR@10 0.013–0.046) and falls below end-to-end *Naive* for every language (Fig. 5). This shows that trie-constrained decoding without planning guidance does not reliably traverse the fixed English docid space under cross-lingual inputs.

Planner-token alignment. *Aligned* yields partial gains for nl/fr/de in Stage 2 MRR@10 (0.090→0.107, 0.097→0.156, 0.102→0.151), but is minimal for zh (0.027→0.030) and remains far below *Translate*. The diagnostics explain this ceiling: TokJaccard@100 and its lower tail improve for nl/fr/de, while CandOverlap@100 does not consistently increase (nl 0.170→0.121, fr 0.176→0.176, de 0.154→0.171, zh 0.124→0.029; Table 9). Thus, token-level planner alignment does not reliably translate into planned-set alignment.

Residual mismatch. Residual gaps persist even under *Translate*: zh remains below nl/fr/de in Stage 2 MRR@10 (0.160 vs. 0.221–0.230) and retains lower overlap (TokJaccard@100 0.317 vs. 0.402–0.430; CandOverlap@100 0.449 vs. 0.537–0.563; Table 9), consistent with remaining mismatch under a fixed English index.

Overall, under a fixed English index, Stage 2 degrades when planning fails to recover English planner evidence, causing the look-ahead bonus to rely on an irrelevant candidate pool. Translation restores overlap and recovers most performance without re-indexing, whereas token-level alignment yields only partial gains because candidate-set overlap does not consistently improve.

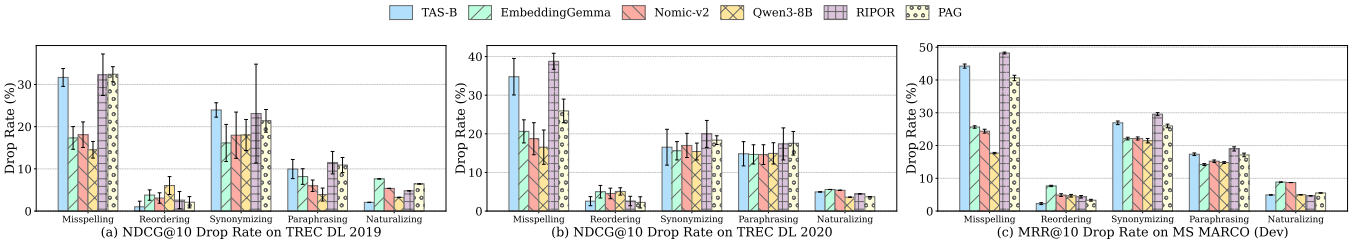

Figure 4: Robustness under query variations: PAG vs. dense and GR baselines. Mean relative drop rate (%) under five intent-preserving query variations, computed relative to the original query. We compare PAG and a strong GR baseline (RIPOR) against recent strong dense retrievers on MS MARCO Dev (MRR@10) and TREC-DL 2019/2020 (NDCG@10). Error bars indicate variability across queries.

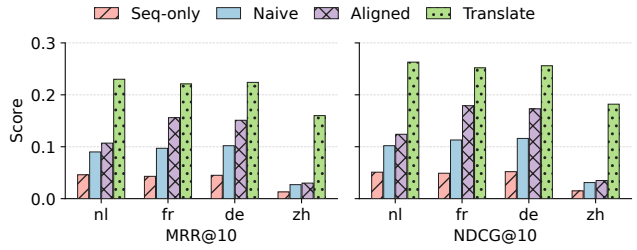

Figure 5: RQ3: Cross-lingual query shift. Stage 2 effectiveness (MRR@10, NDCG@10) on nl/fr/de/zh mMARCO queries with a fixed English index. We compare *Naive*, *Translate*, and *Aligned*; *Seq-only* disables planning guidance.

6 Conclusion

We conducted an inference-time reproduction and stress-test study of Planning Ahead in Generative Retrieval (PAG). For *RQ1*, using the released checkpoint and corpus-side artifacts under the reported decoding configuration, we reproduced the headline effectiveness results on MS MARCO Dev and TREC-DL 2019/2020 and corroborated the qualitative beam–latency trade-off. For *RQ2*, we instrumented the planning stage under intent-preserving query variation and showed that lexical perturbations, such as misspellings and synonym substitutions, can substantially shift the planner's top-n candidate set and high-weight planner tokens. These plan-drift effects coincide with reduced candidate coverage and weaker end-to-end ranking, consistent with increased pruning risk under finite-beam decoding. More broadly, the results indicate that PAG's planning signal is tightly coupled to query surface form: when lexical variation shifts the planned candidate pool, the look-ahead bonus becomes less informative and can in some cases collapse. A plausible contributor is subword fragmentation, whereby small edits alter SentencePiece segmentation and suppress the sparse lexical evidence used to activate planned documents. For *RQ3*, we evaluated cross-lingual query shift under a fixed English identifier space and found that performance degrades markedly under language mismatch. Among the query-side mitigations we tested, query translation provides the strongest recovery, while lightweight planner-token alignment improves over naive cross-lingual use but remains limited without translation.

Taken together, our results show that planning-guided decoding is reproducible and effective under the released inference setup, but its gains depend on the stability of the planning signal under realistic variation and shift.

Takeaways for planning-guided GR. Our findings suggest three practical lessons for future work on planning-guided decoding in GR. First, *planner robustness is a first-order design concern*: surface-form sensitivity is not merely an auxiliary evaluation issue, because under lexical perturbation the planning bonus can weaken enough to approach unguided beam search. Second, *plan drift is diagnostically informative*: candidate-set and planner-token overlap expose failure modes that are not visible from end-to-end effectiveness alone and should be reported alongside ranking metrics in future robustness evaluations. Third, *translation is a strong no-reindex baseline under language mismatch*: in our fixed-index setting, simple query translation consistently outperforms lightweight planner-token alignment, suggesting that restoring compatibility with the planner's evidence space is more effective than token-level alignment alone.

Limitations and future work. This study is bounded by the released inference artifacts (checkpoint, identifiers, trie, and top-m planner tokens), and we cannot evaluate training-stage variants or settings requiring unreleased intermediate artifacts. Latency measurements depend on our hardware and should therefore be interpreted as relative trends rather than directly comparable absolute values. Our stress tests also keep the English index fixed, isolating query-side shift, including language mismatch, rather than corpus-side drift or multilingual document collections. Future work should evaluate (i) planner robustness under corpus-side drift and alternative identifier or trie constructions, (ii) stronger query-side adaptation strategies beyond translation and token distillation, such as multilingual planning signals or jointly trained planners, and (iii) robustness protocols that jointly report end-to-end metrics and intermediate diagnostics, such as candidate coverage, plan drift, and tail-risk indicators, across datasets and beam regimes.

Reproducibility. Our code and artifacts are available at https://github.com/kidist-amde/lost-in-decoding.

Acknowledgments

This research was supported by the Dutch Research Council (NWO), under project numbers 024.004.022, NWA.1389.20.183, and KICH3.-LTP.20.006, the European Union under grant agreement No. 101201-510 (UNITE), the China Scholarship Council (202308440220), Swiss National Science Foundation, grant 215742. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of their respective employers, funders, and/or granting authorities.

References

[1] Michele Bevilacqua, Marco Maru, and Fabio Petroni. 2022. Autoregressive Search Engines: Generating Substrings as Document Identifiers. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[2] Luiz Henrique Bonifacio, Israel Campiotti, R.A. Lotufo, and Rodrigo Frassetto Nogueira. 2021. mMARCO: A Multilingual Version of MS MARCO Passage Ranking Dataset. *ArXiv* abs/2108.13897 (2021). https://api.semanticscholar.org/CorpusID:274281707

[3] Steven Dong, Yubao Tang, and Maarten de Rijke. 2026. Multi-Step Semantic Reasoning in Generative Retrieval. In *European Conference on Information Retrieval*. Springer, 273–281.

[4] Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2021. Beyond English-centric Multilingual Machine Translation. *J. Mach. Learn. Res.* 22, 1, Article 107 (Jan. 2021), 48 pages.

[5] Tim Hagen, Harrisen Scells, and Martin Potthast. 2024. Revisiting Query Variation Robustness of Transformer Models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 4283–4296. https://doi.org/10.18653/v1/2024.findings-emnlp.248

[6] Yuxin Huang, Simeng Wu, Ran Song, Yan Xiang, Yantuan Xian, Shengxiang Gao, and Zhengtao Yu. 2025. Multilingual Generative Retrieval via Cross-lingual Semantic Compression. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (Eds.). Association for Computational Linguistics, Suzhou, China, 10855–10866. https://doi.org/10.18653/v1/2025.findings-emnlp.575

[7] Jie Jiang, Yangru Huang, Zeyu Wang, Changping Wang, Yuling Xiong, Jun Zhang, and Huan Yu. 2026. Spend Search Where It Pays: Value-Guided Structured Sampling and Optimization for Generative Recommendation. *arXiv preprint arXiv:2602.10699* (2026).

[8] Jian Jiao, Gong Yeyun, Nan Duan, Ruofei Zhang, and Ming Zhou. 2025. Look Ahead Strategy for Trie-based Beam Search in Generative Retrieval. US Patent 12,353,454.

[9] Saar Kuzi, Mingyang Zhang, Cheng Li, Michael Bendersky, and Marc Najork. 2020. Leveraging Semantic and Lexical Matching to Improve the Recall of Document Retrieval Systems: A Hybrid Approach. *arXiv preprint arXiv:2010.01195* (2020).

[10] Xiaoxi Li, Jiajie Jin, Yujia Zhou, Yongkang Wu, Zhonghua Li, Ye Qi, and Zhicheng Dou. 2025. RetroLLM: Empowering Large Language Models to Retrieve Fine-grained Evidence within Generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (Eds.). Association for Computational Linguistics, Vienna, Austria, 16754–16779. https://doi.org/10.18653/v1/2025.acl-long.819

[11] Xiaoxi Li, Jiajie Jin, Yujia Zhou, Yuyao Zhang, Peitian Zhang, Yutao Zhu, and Zhicheng Dou. 2025. From Matching to Generation: A Survey on Generative Information Retrieval. *ACM Trans. Inf. Syst.* 43, 3, Article 83 (May 2025), 62 pages. https://doi.org/10.1145/3722552

[12] Yongkang Li. 2026. Understanding and Enhancing Robustness in Dense Information Retrieval. In *Advances in Information Retrieval - 48th European Conference on Information Retrieval, ECIR 2026, Delft, The Netherlands, March 29 - April 2, 2026, Proceedings, Part III (Lecture Notes in Computer Science)*. Springer, 599–607. https://doi.org/10.1007/978-3-032-21324-2_51

[13] Yongkang Li, Panagiotis Eustratiadis, and Evangelos Kanoulas. 2025. Reproducing HotFlip for Corpus Poisoning Attacks in Dense Retrieval. In *Advances in Information Retrieval - 47th European Conference on Information Retrieval, ECIR 2025, Lucca, Italy, April 6-10, 2025, Proceedings, Part IV (Lecture Notes in Computer Science)*. Springer, 95–111. https://doi.org/10.1007/978-3-031-88717-8_8

[14] Yongkang Li, Panagiotis Eustratiadis, Simon Lupart, and Evangelos Kanoulas. 2025. Unsupervised Corpus Poisoning Attacks in Continuous Space for Dense Retrieval. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Padua, Italy) *(SIGIR '25)*. Association for Computing Machinery, New York, NY, USA, 2452–2462. https://doi.org/10.1145/3726302.3730110

[15] Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2024. Learning to Rank in Generative Retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 8716–8723.

[16] Yu-An Liu, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2025. Robust Neural Information Retrieval: An Adversarial and Out-of-Distribution Perspective. *ACM Trans. Inf. Syst.* 44, 1, Article 17 (Nov. 2025), 48 pages. https://doi.org/10.1145/3768153

[17] Yu-An Liu, Ruqing Zhang, Jiafeng Guo, Changjiang Zhou, Maarten de Rijke, and Xueqi Cheng. 2025. On the Robustness of Generative Information Retrieval Models: An Out-of-Distribution Perspective. In *Advances in Information Retrieval: 47th European Conference on Information Retrieval, ECIR 2025, Lucca, Italy, April 6–10, 2025, Proceedings, Part II* (Lucca, Italy). Springer-Verlag, Berlin, Heidelberg, 407–423. https://doi.org/10.1007/978-3-031-88711-6_26

[18] Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, Noah A. Smith, and Yejin Choi. 2022. NeuroLogic A*esque Decoding: Constrained Text Generation with Lookahead Heuristics. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (Eds.). Association for Computational Linguistics, Seattle, United States, 780–799. https://doi.org/10.18653/v1/2022.naacl-main.57

[19] Simon Lupart and Stéphane Clinchant. 2023. A Study on FGSM Adversarial Training for Neural Retrieval. In *Advances in Information Retrieval: 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2–6, 2023, Proceedings, Part II* (Dublin, Ireland). Springer-Verlag, Berlin, Heidelberg, 484–492. https://doi.org/10.1007/978-3-031-28238-6_39

[20] Kidist Amde Mekonnen, Yubao Tang, and Maarten de Rijke. 2025. 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* (Padua, Italy) *(SIGIR '25)*. Association for Computing Machinery, New York, NY, USA, 1327–1338. https://doi.org/10.1145/3726302.3730023

[21] Suraj Nair, Eugene Yang, Dawn Lawrie, Kevin Duh, Paul McNamee, Kenton Murray, James Mayfield, and Douglas W. Oard. 2022. Transfer Learning Approaches for Building Cross-Language Dense Retrieval Models. In *Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10–14, 2022, Proceedings, Part I* (Stavanger, Norway). Springer-Verlag, Berlin, Heidelberg, 382–396. https://doi.org/10.1007/978-3-030-99736-6_26

[22] Nishanth Sridhar Nakshatri, Shamik Roy, Rajarshi Das, Suthee Chaidaroon, Leonid Boytsov, and Rashmi Gangadharaiah. 2025. Constrained Decoding with Speculative Lookaheads. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Luis Chiruzzo, Alan Ritter, and Lu Wang (Eds.). Association for Computational Linguistics, Albuquerque, New Mexico, 4681–4700. https://doi.org/10.18653/v1/2025.naacl-long.239

[23] Zach Nussbaum and Brandon Duderstadt. 2025. Training Sparse Mixture of Experts Text Embedding Models. *arXiv preprint arXiv:2502.07972* (2025).

[24] Gustavo Penha, Arthur Câmara, and Claudia Hauff. 2022. Evaluating the Robustness of Retrieval Pipelines with Query Variation Generators. In *Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10–14, 2022, Proceedings, Part I* (Stavanger, Norway). Springer-Verlag, Berlin, Heidelberg, 397–412. https://doi.org/10.1007/978-3-030-99736-6_27

[25] Ronak Pradeep, Kai Hui, Jai Gupta, Adam Lelkes, Honglei Zhuang, Jimmy Lin, Donald Metzler, and Vinh Tran. 2023. How Does Generative Retrieval Scale to Millions of Passages?. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 1305–1321. https://doi.org/10.18653/v1/2023.emnlp-main.83

[26] Weizhen Qi, Yeyun Gong, Yu Yan, Jian Jiao, Bo Shao, Ruofei Zhang, Houqiang Li, Nan Duan, and Ming Zhou. 2020. ProphetNet-Ads: A Looking Ahead Strategy for Generative Retrieval Models in Sponsored Search Engine. In *CCF International Conference on Natural Language Processing and Chinese Computing*. Springer, 305–317.

[27] Felix Stahlberg and Bill Byrne. 2019. On NMT Search Errors and Model Errors: Cat Got Your Tongue?. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China, 3356–3362. https://doi.org/10.18653/v1/D19-1331

[28] Yubao Tang, Ruqing Zhang, Jiafeng Guo, and Maarten de Rijke. 2023. Recent Advances in Generative Information Retrieval. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*. 294–297.

[29] Yubao Tang, Ruqing Zhang, Jiafeng Guo, Maarten De Rijke, Wei Chen, and Xueqi Cheng. 2024. Listwise generative retrieval models via a sequential learning process. *ACM Transactions on Information Systems* 42, 5 (2024), 1–31.

[30] Yubao Tang, Ruqing Zhang, Zhaochun Ren, Jiafeng Guo, and Maarten de Rijke. 2024. Recent Advances in Generative Information Retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington DC, USA) *(SIGIR '24)*. Association for Computing Machinery, New York, NY, USA, 3005–3008. https://doi.org/10.1145/3626772.3661379

[31] Yubao Tang, Ruqing Zhang, Weiwei Sun, Jiafeng Guo, and Maarten De Rijke. 2024. Recent Advances in Generative Information Retrieval. In *Companion Proceedings of the ACM Web Conference 2024*. 1238–1241.

[32] 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. 2022. Transformer Memory as a Differentiable Search Index. In *Advances in Neural Information Processing Systems (NeurIPS)*. arXiv:2202.06991.

[33] Lifu Tu, Semih Yavuz, Jin Qu, Jiacheng Xu, Rui Meng, Caiming Xiong, and Yingbo Zhou. 2024. Unlocking Anticipatory Text Generation: A Constrained Approach for Large Language Models Decoding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 15532–15548. https://doi.org/10.18653/v1/2024.emnlp-main.870

[34] Henrique Schechter Vera, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, Sara Smoot, Iftekhar Naim, Joe Zou, Feiyang Chen, et al. 2025. EmbeddingGemma: Powerful and Lightweight Text Representations. *arXiv preprint arXiv:2509.20354* (2025).

[35] 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. 2022. A Neural Corpus Indexer for Document Retrieval. In *Proceedings of the 36th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '22)*. Curran Associates Inc., Red Hook, NY, USA, Article 1856, 15 pages.

[36] Shiguang Wu, Zhaochun Ren, Xin Xin, Jiyuan Yang, Mengqi Zhang, Zhumin Chen, Maarten de Rijke, and Pengjie Ren. 2025. Constrained Auto-Regressive Decoding Constrains Generative Retrieval. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Padua, Italy) *(SIGIR '25)*. Association for Computing Machinery, New York, NY, USA, 2429–2440. https://doi.org/10.1145/3726302.3729934

[37] Hansi Zeng, Chen Luo, and Hamed Zamani. 2024. 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* (Washington DC, USA) *(SIGIR '24)*. Association for Computing Machinery, New York, NY, USA, 469–480. https://doi.org/10.1145/3626772.3657746

[38] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. 2025. Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models. *arXiv preprint arXiv:2506.05176* (2025).

[39] Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. 2023. Enhancing Generative Retrieval with Reinforcement Learning from Relevance Feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 12481–12490.

[40] Yujia Zhou, Jing Yao, Zhicheng Dou, Yiteng Tu, Ledell Wu, Tat-Seng Chua, and Ji-Rong Wen. 2024. ROGER: Ranking-Oriented Generative Retrieval. *ACM Trans. Inf. Syst.* 42, 6, Article 155 (Oct. 2024), 25 pages. https://doi.org/10.1145/3603167

[41] Yujia Zhou, Jing Yao, Zhicheng Dou, Ledell Wu, Peitian Zhang, and Ji-Rong Wen. 2022. Ultron: An Ultimate Retriever on Corpus with a Model-based Indexer. *arXiv preprint arXiv:2208.09257* (2022).