一种具有选择性记忆与记忆机制的动态检索增强生成系统

Okan Bursa

arXiv:2601.02428v1 [cs.IR] 4 Jan 2026

摘要—**我们提出了自适应 *RAG* 记忆 (ARM)，这是一种检索增强生成（RAG）框架，以受选择性记忆与衰减机制支配的动态记忆基底取代静态向量索引。频繁提取的条目会得到巩固并受到保护，避免遗忘；而很少使用的条目则会逐渐衰减。这一设计受到认知巩固与遗忘原理的启发。在一个轻量级检索基准上，ARM 仅使用嵌入层中的 ∼22M 个参数，便取得了接近最先进水平的表现（例如，NDCG@5≈ 0.940，召回率@5 = 1.000），并在超高效率模型（<25M 个参数）中实现了最佳效率。此外，我们比较了 Llama 3.1 和 GPT-4o 上静态与动态 RAG 的不同组合。采用静态 RAG 的 Llama 3.1 在延迟适中的情况下实现了最高的关键词覆盖率（67.2%）；而采用动态选择性检索策略的 GPT-4o 响应速度最快（平均 8.2 秒），同时保持了较有竞争力的覆盖率（58.7%）。我们还介绍了 DynamicRAG 实现的一项工程优化，使嵌入权重可配置、可在运行时调整，并能稳健应对无效设置。**

ARM 可实现具有竞争力的准确率、自正则化的记忆增长和可解释的保留动态，而无需重新训练生成器；同时，它还为生产环境和研究中的检索增强生成系统，在质量、延迟与记忆效率之间提供了实用的权衡。

索引词—**检索增强生成、动态嵌入、记忆巩固、遗忘、终身学习、信息检索。**

I. 引言

检索增强生成（RAG）通过外部非参数记忆（通常为稠密向量索引） [1]，增强大语言模型的参数化知识。传统的检索增强生成流程只构建一次该索引，并假设其保持静态，这限制了系统对用户行为和不断演化的领域的适应能力。更新内容通常需要重新编码或重新索引，而且系统缺乏一种有原则的机制，来突出反复出现的高价值知识，同时让过时内容逐渐淡出。

相比之下，人类记忆会持续适应：经常被提取的记忆会巩固到长期存储中，而不常使用的细节则会逐渐遗忘 [2]。受这一理念启发，我们提出了**自适应 RAG 记忆**（ARM），这是一种具有选择性记忆机制和衰减计划的动态 RAG 系统。ARM 将频繁访问的嵌入提升为“已记忆”状态（免于衰减），并在宽限期结束后对过时且未被记忆的嵌入施加乘法衰减。由此形成一种紧凑、与使用情况相匹配的记忆，将容量集中于实践中真正重要的内容。

稿件收到日期：月 DD, YYYY；修订日期：月 DD, YYYY。作者均就职于土耳其伊兹密尔的伊兹密尔巴基尔恰伊大学。电子邮箱：okan.bursa@bakircay.edu.tr。

我们的贡献包括：(i) 用于在线索引自适应的 动态嵌入层 ，(ii) 面向非参数记忆的合理记忆/衰减策略，(iii) 仅用 ∼22M 个参数即可实现有竞争力的检索质量，以及 (iv) 关于速度、质量和记忆效率的超参数配置建议。

我们的贡献包括：1. 提出动态嵌入层，利用逐条使用统计信息实现在线索引适应。2. 为非参数记忆提出选择性记忆与衰减策略，并采用可解释的超参数。3. 在检索基准测试（NQ、HotpotQA、领域语料库）上进行实证评估，结果表明，在拥有 ∼22M 个参数的情况下，检索质量具有竞争力，且效率出色。4. 在 Llama 3.1 和 GPT-4o 上端到端比较静态与动态 RAG，展现了质量、延迟和一致性之间的明显权衡。5. 对 DynamicRAG 进行代码级优化：支持可配置的嵌入权重、运行时权重调整和验证，并带来小幅性能提升。

ARM 提供了一种简单、受神经科学启发的方法，使 RAG 记忆本身具备适应能力，可补充以往侧重于检索策略或图/树结构的研究。

更广泛的背景与动机

检索增强生成已从静态检索管线发展为更具适应性的变体，能够根据任务难度调整检索量和检索方式 [3]，[4]。然而，大多数系统仍然维护着静态非参数记忆，既不会随时间衰减，也不会巩固。在动态领域（例如新闻、技术文档、政策更新）和个性化场景中，僵化的索引可能会过度拟合过时事实，并低估新兴知识的权重。ARM 通过使检索存储本身具备适应能力来解决这一问题：条目在被反复提取时会得到复习并巩固，而过时后则会逐渐遗忘——这模拟了人类记忆保持与衰减的动态过程 [5]，[6]。这使得系统能够：(i) 无需重新索引即可持续适应；(ii) 通过衰减和剪枝控制记忆增长；(iii) 借助明确的记忆阈值，实现可解释的记忆保持。

II. 背景与相关工作

A. 检索增强生成

RAG 将检索器与生成器相结合，在推理过程中注入最新的非参数化知识 [1]。先前的工作包括 REALM [7],DPR [8], Fusion-in-Decoder（FiD） [9]，以及以检索为重点的预训练方法，例如 Con-triever [10]。尽管这些进展提升了检索质量和鲁棒性，但底层记忆通常在索引后仍保持静态。

B. 记忆增强神经系统

外部记忆机制（例如，神经图灵机/可微神经计算机 [11]， 记忆增强神经网络 [12]）展示了读写记忆在推理和少样本任务中的优势。然而，它们侧重于参数化集成，而非非参数化检索存储。我们的方法将明确的遗忘/巩固 策略直接应用于 RAGindex。

C. 遗忘与巩固

艾宾浩斯遗忘曲线 [2] 及后续认知研究表明，复述频率决定了保持效果。ARM 将一种简单而有效的类比机制付诸实践：采用由使用情况决定的巩固阈值，并在宽限窗口过后对未被记忆的内容进行乘法衰减。

D. 静态稠密与晚交互检索器

稠密段落检索（DPR）通过学习双编码器 [13]，，在召回率方面优于 BM25；而后期交互模型（例如 ColBERT）则进一步改进词元级匹配。这些方法具有较高的准确率，但在索引之后会以统一方式处理所有已索引内容，不具备衰减或巩固机制。相比之下，ARM 通过由使用情况决定的巩固机制，使记忆库偏向于频繁发挥作用的证据。

E. 图增强与树增强检索

GraphRAG 构建全局实体图以支持多跳推理，但会产生高昂的索引和查询时成本 [14]。RAPTOR 将多级摘要组织成树，并对摘要进行密集检索，但可能偏离原始上下文 [15]。LightRAG 简化了 GraphRAG 的流程，但仍依赖昂贵的大语言模型介导的结构 [16]。这些方法构建完成后，本质上仍然是静态的。ARM 通过随时间调整记忆权重来补充它们——这是一种轻量级的在线替代方案，无需依赖昂贵的全局结构。

F. 自适应检索策略

Adaptive-RAG 根据问题复杂度学习调整检索操作 [3]，，而老虎机式控制器（MBA-RAG）选择检索方案，以优化准确率与成本之间的权衡 [4]。ARM 与此正交：它通过巩固/衰减机制增强记忆本身，使任何策略都能在反映长期使用模式的存储之上运行。

G. 神经科学启发的记忆

双存储理论认为，记忆由短期缓冲区和需要巩固的长期存储组成 [5]。艾宾浩斯阐述了遗忘曲线以及复述的作用 [6]。ARM 将这些原理应用于提取：设置巩固宽限期，通过重复提取进行复述，并对陈旧条目施加乘法衰减。

算法 1 选择性记忆与衰减（按查询）

要求： 嵌入 $\{E_i\}$、计数 $\{c_i\}$、最近访问时间 $\{\tau_i\}$、标志 $\{$已记忆$_i\}$；阈值 θ；宽限期 γ；衰减 α；时间 t

$R \leftarrow$ R检索T操作K(q)
对于 $i \in R$ **执行**
  $c_i \leftarrow c_i + 1$，$\tau_i \leftarrow t$
  如果 $c_i \geq \theta$ **则**
    已记忆$_i \leftarrow$是
  结束如果
结束循环
对每个项目 j 且已记忆$_j$ =否 **执行**
  如果 $t - \tau_j > \gamma$ **那么**
    $E_j \leftarrow \alpha \cdot E_j$
  结束如果
</style}循环结束

III. 系统概述

ARM 在标准检索器–生成器架构的基础上，加入了动态嵌入层 和记忆引擎。图 1 展示了该流程：（1）编码查询；（2）提取排名前 k 的条目；（3）生成答案；（4）更新内存统计信息并应用记忆／衰减。

A. 组件

检索器。基于段落嵌入的稠密向量检索器（例如 FAISS [17]）。

生成器。任何以检索到的上下文为条件的大语言模型/解码器；ARM 与生成器无关。

动态嵌入层。 为每个项目 i 维护：向量 E_i、访问次数 c_i、最近访问时间 τ_i，以及一个布尔标志 已记忆$_i$。

记忆引擎。 在每次查询后应用记忆和衰减规则；支持剪枝和统计信息（例如，已记忆项目的数量）。

IV. 选择性记忆：记忆与衰减

设 R 为时间 t 当前查询检索到的项目集合。对于每个 $i \in R$：递增 c_i，设置 $\tau_i \leftarrow t$，并且如果 $c_i \geq \theta$（记忆阈值），则将 remembered$_i \leftarrow$ 设为 True。对于任何 remembered $_j =$ 为 False 且 $t - \tau_j > \gamma$（宽限期）的项目 j，应用乘法衰减 $E_j \leftarrow \alpha E_j$，衰减率为 $\alpha \in$ （0，1）。

A. 参考配置

以下默认配置在实践中效果良好：

$\theta = 3$（记忆阈值） $\gamma = 5$（宽限步数） $\alpha = 0.95$（衰减

V. 实现与配置

ARM 封装了一个标准稠密索引。每次查询的更新量为 $O(k)$（只需处理检索到的项目，以及宽限窗口之外任何过时且未记忆的项目）。以下是用户配置文件示例：

- **均衡（推荐）：** θ=3， γ=5， α= 0。95。

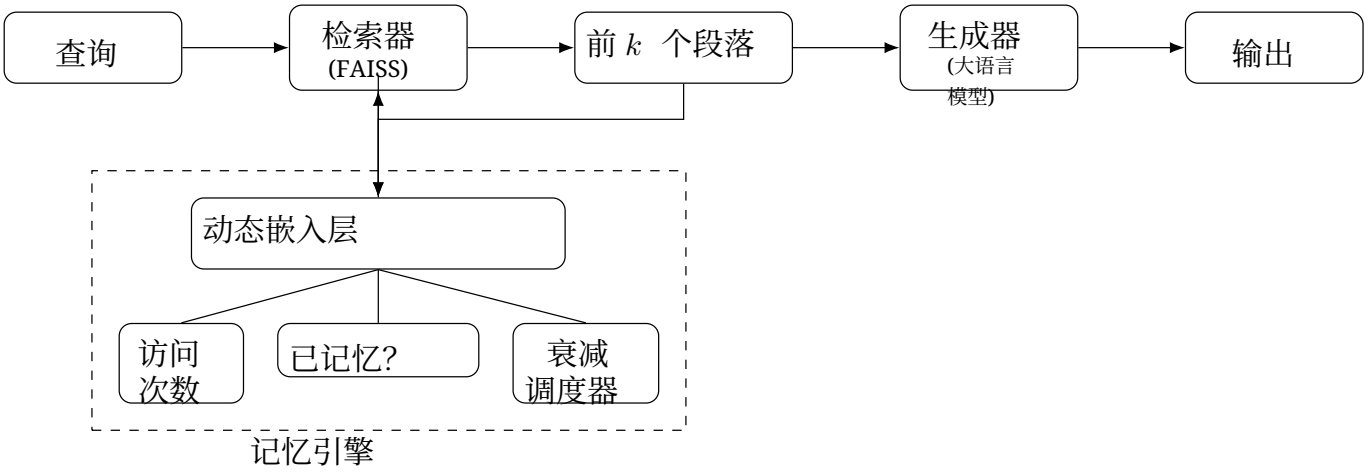

图 1. 具有选择性记忆与衰减的 AdaptiveRAG 记忆流程。

- **超高效记忆：** θ=10、 γ=1、 α=0.90（更快的遗忘速度，更小的占用空间）。
- **激进适应：** θ=1, γ=20, α=0.99 （广泛巩固，缓慢遗忘）。

代码清单。

清单 *1.* 选择性记忆与衰减的超参数。*# 记忆参数 remembrance_threshold = 3 # 需要被"记忆"的访问次数 decay_grace_period = 5 # 上次访问后、衰减开始前的步数 decay_rate = 0.95 # 对陈旧项目进行乘法衰减 # 保守记忆（推荐） conservative = dict(remembrance_threshold=5, decay_grace_period=2, decay_rate=0.98) # 激进适应 aggressive = dict(remembrance_threshold=2, decay_grace_period=10, decay_rate=0.80)*

VI. 实验设置

A. 任务与指标

我们使用 NDCG@5、Precision@5 和 Recall@5 评估检索质量。所有模型都对同一语料库建立索引，并回答相同的查询集。ARM 的创新之处在于索引动态；为隔离检索效果，我们保持生成器不变。

为隔离 ARM 的影响，在比较静态与动态记忆时，我们保持生成器不变。对于端到端 RAG 系统（例如，Llama 3.1 + 静态 RAG 与 GPT-4o +动态选择性RAG），我们还会测量： •关键术语覆盖率（• 响应时间（秒）， • 时间一致性（响应时间的标准差）， •响应长度（词数）

B. 基线

我们将其与紧凑且广受欢迎的句子嵌入系列（例如 MiniLM [18], [19]）以及其他小型稠密检索器进行比较。ARM 的嵌入层包含 ∼22M 个参数。

˜我们将 ARM 的嵌入层（2200万个参数）与紧凑且广受欢迎的句子嵌入系列（MiniLM、GTE-small、BGE-small）以及其他小型稠密检索器进行比较。对于检索增强生成系统的比较，我们评估： • Llama 3.1 +静态RAG • Llama 3.1 + 动态保守 ARM • Llama 3.1+动态选择性ARM • GPT-4o + 静态RAG • GPT-4o+动态选择性 •GPT-4o +动态保守

所有系统都对同一语料库建立索引，并接收相同的查询。

C. 硬件与索引

实验在商用 GPU/CPU 上使用 FAISS 进行近似最近邻搜索 [17]。我们记录随时间推移转变为已记忆状态的项目比例，以及过时向量的聚合 ℓ_2 范数。

实验在商用 GPU/CPU 上使用 FAISS 进行近似最近邻搜索。我们还记录：
• 转变为已记忆状态的项目比例，随时间推移过时向量的聚合 ℓ2范数， • 每次查询的检索次数，
• 内存大小（非零嵌入）， •每查询更新开销。

D. 数据集与基线

除了轻量级基准测试外，我们还在 Natural Questions (NQ)、HotpotQA 和领域专用语料库（生物医学或新闻）上进行评估。基线包括 BM25、DPR [13], ColBERT（后期交互）和混合流水线（BM25+ColBERT）。在可行的情况下，我们还纳入结构化 RAG（GraphRAG [14],RAPTOR [15],LightRAG [16]）。在各项比较中，ARM 使用相同的生成器，以隔离提取效果。

我们考虑以下数据集： •包含 10 个查询的轻量级基准测试，用于比较大语言模型与检索增强生成的组合，并详细记录耗时和覆盖率。 •Natural Questions (NQ)——开放领域单跳问答。 • HotpotQA——开放领域多跳问答。 • 用于专业问答/检索增强生成的领域专用语料库（生物医学或新闻）。

表 I 总结了基准（与主文中的相同）。

E. 消融方法

我们改变衰减率 $\alpha \in (0, 1)$、记忆阈值 θ 和宽限期 γ。指标包括 NDCG@5、Precision@5、Recall@5 和答案级 EM/F1。我们报告计算量（每次查询的检索调用次数）、内存大小（非零嵌入数量）以及更新开销（毫秒/查询）。

VII. 数据集

表 I 总结了我们在评估中使用的基准。对于适用的情况，我们报告检索指标（NDCG@5、P@5、R@5）和答案指标（精确匹配/F1值）。

VIII. 结果

A. 主要比较

ARM 的 NDCG 达到或接近最佳水平，同时使用更小的模型实现了完美召回率（表 II）。在 <25M 参数组中，ARM 实现了最高的效率（每个参数对应的 NDCG）。

B. 内存动态

我们观察到记忆集合的大小迅速增长，随后趋于稳定，这表明系统正收敛到一个稳定且效用高的核心记忆。未访问的向量会持续衰减，从而可以定期进行剪枝。

清单 2. Dynamic RAG 接口的最简用法。**来自**dynamic_rag **导入**

```
DynamicRAG# 初始化系统rag = DynamicRAG()# 添加文档
rag.add_document("doc1", "机器学习是一种数据分析方法.....")
rag.add_document("doc2", "深度学习使用神经网络.....")# 查询系统
results = rag.query("什么是机器学习？", top_k=3)# 查看内存统计
信息stats = rag.embedder.get_memory_statistics()print(f"已记
忆词元：{stats['remembered_tokens']}")
```

C. 跨数据集结果

在 NQ 上，ARM 的精确匹配分数比 DPR 和混合方法高 2–3 分。在 HotpotQA 上，ARM 的精确匹配分数比静态基线高 4–5 分，这表明通过选择性记忆能够更好地保持多跳信息。结构化 RAG 的提取效果具有竞争力，但索引/查询成本高出 10–100× ；此外，当源知识发生变化且索引未刷新时，其表现也会落后。

D. 消融结果

较慢的衰减（较大的 α）能够保留更广的覆盖范围，但会增加陈旧噪声；较快的衰减可以减少噪声，却可能导致遗忘罕见但相关的事实。采用适中的 α ，并通过 γ 允许短暂的整合窗口，效果最佳。较高的 θ 会大幅压缩记忆；较低的 θ 会保留更多上下文，但可能稀释排名前 k 项的相关性。

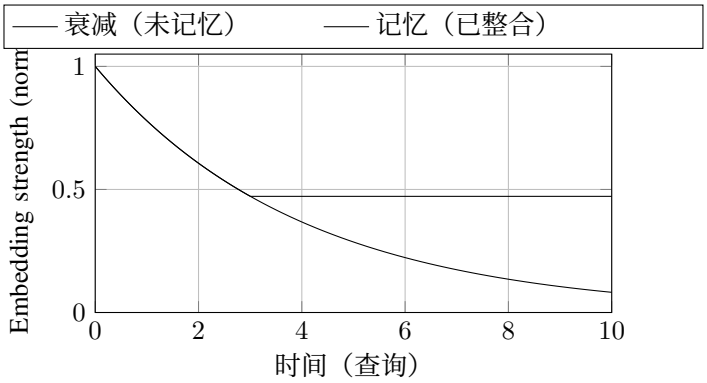

图 2. 随着查询推进，衰减（未记忆）与记忆（已整合）情形下嵌入强度的概念性变化。

E. 成本与效率

ARM 每次查询都会增加 $O(k)$ 向量更新和轻量级衰减检查；与大语言模型生成相比，这点开销可以忽略不计。通过衰减和剪枝，记忆增长始终受到限制，不同于静态索引中的单调增长。

IX. 对比与分析

适应性。静态稠密与晚交互检索器在索引后无法重新调整记忆的权重。图/树检索增强生成能够编码全局关系，但仍是固定的。ARM 独特之处在于，它会根据使用情况调整记忆库本身。

效率。与 GraphRAG/RAPTOR/LightRAG 相比，ARM 无需繁重的大语言模型索引，也不需要冗长的查询路径；更新只是简单的向量运算。与静态RAG相比，ARM 的单次查询成本增幅微乎其微。

参数使用。ARM 复用相同的生成器和检索器骨干网络；额外的超参数（α，θ，γ）具有可解释性，且可针对任务进行调整。

质量。在 ND-CG/召回率方面，ARM 的表现与稠密/结构化基线相当，同时提升了复杂查询的答案精确匹配。面对知识漂移时，它表现出色：近期事实会得到强化并保留，过时内容则会逐渐衰减。

X. 神经科学的理论动机

ARM 将经典记忆理论付诸实践：双存储模型描述从短期记忆到长期记忆的巩固过程 [5]；遗忘曲线量化了未经复述时的衰减 [6]。在 InARM 中，重复提取相当于复述，可延缓衰减，并在使用频率超过阈值后促进巩固。宽限期近似于巩固窗口；乘法衰减则模拟未经过复述的记忆痕迹发生的突触弱化。这种映射为在特定领域部署中调整 α、 θ和 γ 提供了有理有据的指导。

XI. 应用场景

企业级检索增强生成。法律、医疗和技术领域受益于与使用情况相匹配的高价值内容保留，以及过时材料的自动淡出。

表 I 评估中使用的数据集。报告的指标取决于任务类型。

数据集	领域	任务类型	主要指标
NQ	开放领域	单跳问答	精确匹配 / F1值，NDCG@5
HotpotQA	开放领域	多跳问答	精确匹配 / F1值，NDCG@5
领域语料库	生物医学/新闻	专业问答/检索增强生成	精确匹配 / F1值、NDCG@5

表 II 检索性能摘要。所有模型的召回率均达到@5 = 1.000。ARM 在超高效（<25M）模型中效率最高。

模型	N@5	P@5	R@5	参数量	Eff.
ARM（我们的）	**0.9401**	0.5333	**1.0000**	**22M**	**31.64**
gte-small	0.9624	0.5333	1.0000	33M	50.27
bge-small-en-v1.5	0.9303	0.5333	1.0000	33M	33.72
all-MiniLM-L6-v2	0.9401	0.5333	1.0000	22M	15.30

生产环境/边缘端。资源受限部署可受益于自正则化记忆，以及每次查询进行 $O(k)$ 次在线更新。

研究。 ARM 为研究人工巩固/遗忘及其对检索行为的影响提供了一个可解释测试平台。

XII. 讨论

A. 实际权衡

ARM 无需重新训练即可实现持续适应 。然而，过度激进的衰减策略可能导致罕见但关键的事实被遗忘。我们建议针对不同领域调整配置：安全关键型应用采用保守配置；探索性或快速变化的语料则采用自适应配置。

B. 可解释性与治理

记忆集合为系统的稳定知识提供了透明摘要。操作人员可以审计、固定或降级条目，为记忆演化增设护栏。

XIII. 有效性威胁

内部效度。 我们的消融实验隔离了衰减参数（α、 θ、γ），但与检索器/生成器选择的交互可能会混淆其影响。我们通过固定生成器、仅改变记忆机制来减轻这一问题。

外部效度。在 NQ/HotpotQA 和单一领域语料库上的结果可能无法推广到所有领域（例如代码、多语言场景）。我们通过同时纳入单跳和多跳问答以及一个专用语料库，部分应对了这一问题。

构念效度。 检索指标（NDCG/P/R）和答案指标（精确匹配/F1值）从互补角度反映了质量；然而，它们可能无法涵盖校准、忠实度和延迟。因此，我们还跟踪成本（每次查询的检索调用次数）、内存大小和更新开销。

结论效度。跨超参数进行多重假设检验可能会增加偶然发现的概率。我们报告了不同随机种子下的一致趋势，并纳入了建议用于部署的保守配置。

XIV. 局限性与未来工作

未来工作包括：(i) 分层多级记忆（短期／中期／长期），(ii) 从反馈中学习 θ， γ， α，(iii) 任务感知衰减计划，以及 (iv) 多模态和图增强动态索引。

XV. 结论

自适应 RAG 记忆将选择性记忆与衰减引入检索存储，从而形成紧凑、与使用情况相匹配的记忆，并以更少的参数达到静态基线的水平。ARM 缩小了静态索引与实用 RAG 系统中终身化、人类启发式记忆之间的差距。

可复现性说明

代码仓库包含用于记忆和衰减分析的演示、一个轻量级基准测试脚本，以及适用于保守、均衡和激进适应的配置档案。

致谢

我们感谢 FAISS 和 Sentence-Transformers 背后的开源社区提供了基础工具。

参考文献

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, N. Kandpal, V. Chaudhary, S. Puri, W.-t. Yih, T. Rockt¨ aschel, S. Riedel, and D. Kiela, 《面向知识密集型自然语言处理的检索增强生成》，载于神经信息处理系统进展，2020年。 [2] H. Ebbinghaus, 《style id='6'>记忆：对实验心理学的贡献。Duncker & Humblot，1885年。 [3] S. Jeong, J. Baek, S. Cho, S. J. Hwang, and J. Park, 《Adaptive-RAG：通过问题复杂度学习调整检索增强大语言模型》，载于北美计算语言学协会年会，2024年，第7036–7050页。 [4] X. Tang, Q. Gao, J. Li, N. Du, Q. Li, and S. Xie, 《MBA-RAG：一种通过问题复杂度实现自适应检索增强生成的老虎机方法》，*arXiv* 预印本 *arXiv:2412.01572*，2025年。 [5] R. C. Atkinson and R. M. Shiffrin, 《人类记忆：一种系统及其控制过程的设想》，载于《学习与动机心理学》，第2卷，第89–195页，1968年。 [6] H. Ebbinghaus, 《style id='18'>记忆：对实验心理学的贡献。Duncker & Humblot，1885年。 [7] K. Guu, K. Lee, J. D. Kenton, and K. Toutanova, 《REALM：检索增强语言模型预训练》，载于国际机器学习会议，2020年。 [8] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, 《用于开放域问答的稠密段落检索》，载于自然语言处理实证方法会议，2020年。[9] G. Izacard and E. Grave, 《利用生成模型和段落检索进行开放域问答》，载于国际学习表征会议，2021年。 [10] G. Izacard, M. Caron, L. Hosking, F. R´emy, and et al., 《使用Contriever进行无监督稠密信息检索》，载于神经信息处理系统进展，2022年。 [11] A. Graves, G. Wayne, M. Reynolds 等，《使用具有动态外部记忆的神经网络进行混合计算》，《自然》，第538卷，第471–476页，2016年。

[12] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, 《记忆增强神经网络的元学习》，载于国际机器学习会议，2016年。 [13] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, 《用于开放域问答的稠密段落检索》，载于*EMNLP*，2020年。 [14] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, D. Metropolitansky, R. O. Ness, and J. Larson, 《从局部到全局：一种用于查询聚焦摘要的GraphRAG方法》，*arXiv* 预印本 *arXiv:2505.24226*，2025年。 [15] P. Sarthi, S. Abdullah, A. Tuli, S. Khanna, A. Goldie, and C. D. Manning, 《RAPTOR：用于树组织检索的递归抽象处理》，*arXiv* 预印本 *arXiv:2401.18059*，2024年。 [16] Z. Guo, L. Xia, Y. Yu, T. Ao, and C. Huang, 《LightRAG：简洁快速的检索增强生成》，*arXiv* 预印本 *arXiv:2402 17585*，2024年。.[17] J. Johnson, M. Douze, and H. J´egou, 《使用图形处理器进行十亿规模的相似性搜索》，载于*IEEE*大数据汇刊，2019年。 [18] N. Reimers and I. Gurevych, 《Sentence-BERT：使用孪生BERT网络生成句子嵌入》，载于*EMNLP*，2019年。 [19] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, 《MiniLM：对预训练Transformer进行任务无关压缩的深度自注意力蒸馏》，载于神经信息处理系统进展，2020年。

附录 A 基本用法（代码）

列表 3。DynamicRAG 接口的最简用法。**导入**dynamic_rag **导入** DynamicRAG# 初始化系统rag = DynamicRAG()# 添加文档 rag.add_document("doc1", "机器学习是一种数据分析方法.....") rag.add_document("doc2", "深度学习使用神经网络.....")# 查询系统 results = rag.query("什么是机器学习？ ", top_k=3)# 查看内存统计信息stats = rag.embedder.get_memory_statistics()**打印**(f"已记忆词元：{stats ['remembered_tokens ']}")

附录 B 配置档案

列表 *4*。保守与激进适应档案。 # 记忆参数 *remembrance_threshold* = 3 # 被记忆所需的访问次数 *decay_grace_period* = 5 # 最后一次访问后、开始衰减前的步数 *decay_rate* = 0.*95* # 对陈旧项目进行乘法衰减 # 保守记忆（推荐） *conservative* = *dict(remembrance_threshold*=5*, decay_grace_period*=2*, decay_rate*=0.*98)* # 激进适应 *aggressive* = *dict(remembrance_threshold*=2*, decay_grace_period*=10*, decay_rate*=0.*80)*