Graph RAG:

Graph RAG constructs a knowledge graph on-the-fly, linking relevant entities during retrieval It leverages node relationships to decide when and how much external knowledge to retrieve Confidence scores from the graph guide expansion, avoiding irrelevant additions This approach improves efficiency and response accuracy by keeping the knowledge graph compact and relevant.

Adaptive RAG:

Adaptive RAG dynamically decides when to retrieve external knowledge, balancing internal and external knowledge It uses confidence scores from the language model's internal states to assess retrieval necessity An honesty probe helps the model avoid hallucinations by aligning its output with its actual knowledge It reduces unnecessary retrievals, improving both efficiency and response accuracy.

REALM RAG:

REALM RAG retrieves relevant documents from large corpora like Wikipedia to enhance model predictions The retriever is trained with masked language modeling, optimizing retrieval to improve prediction accuracy It uses Maximum Inner Product Search to efficiently find relevant documents from millions of candidates during training REALM outperforms previous models in Open-domain Question Answering by integrating external knowledge.

RAPTOR RAG:

RAPTOR RAG builds a hierarchical tree by clustering and summarizing text recursively It enables retrieval at different abstraction levels, combining broad themes with specific details RAPTOR outperforms traditional methods in complex question-answering tasks Offers tree traversal and collapsed tree methods for efficient information retrieval.

REFEED RAG:

REFEED RAG refines model outputs using retrieval feedback without fine-tuning Initial answers are improved by retrieving relevant documents and adjusting the response based on the new information Generates multiple answers to improve retrieval accuracy Combines pre- and post-retrieval outputs using a ranking system to enhance answer reliability.

Iterative RAG:

Iterative RAG Unlike traditional retrieval, iterative RAG performs multiple retrieval steps, refining its search based on feedback from previously selected documents Retrieval decisions follow a Markov decision process Reinforcement learning improves retrieval performance The iterative retriever maintains an internal state, allowing it to adjust future retrieval steps based on the accumulated knowledge from previous iterations.