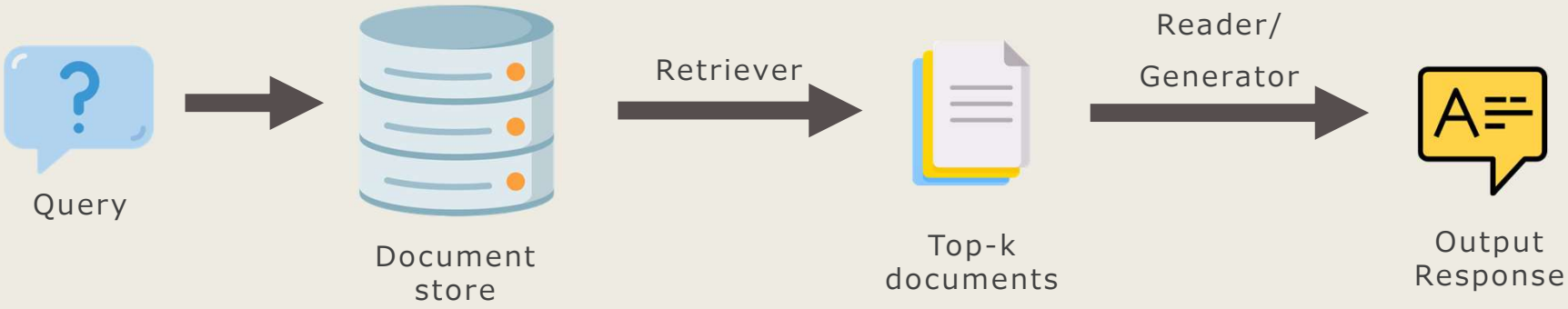
# Enhancing RAG Robustness via RobustRAG & RbFT with Auxiliary Defenses

*Candidate project 2:*
*Robust Retrieval-Augmented Generation (RAG)*

# Retrieval-Augmented Generation (RAG)[1]

**RAG Architecture:**



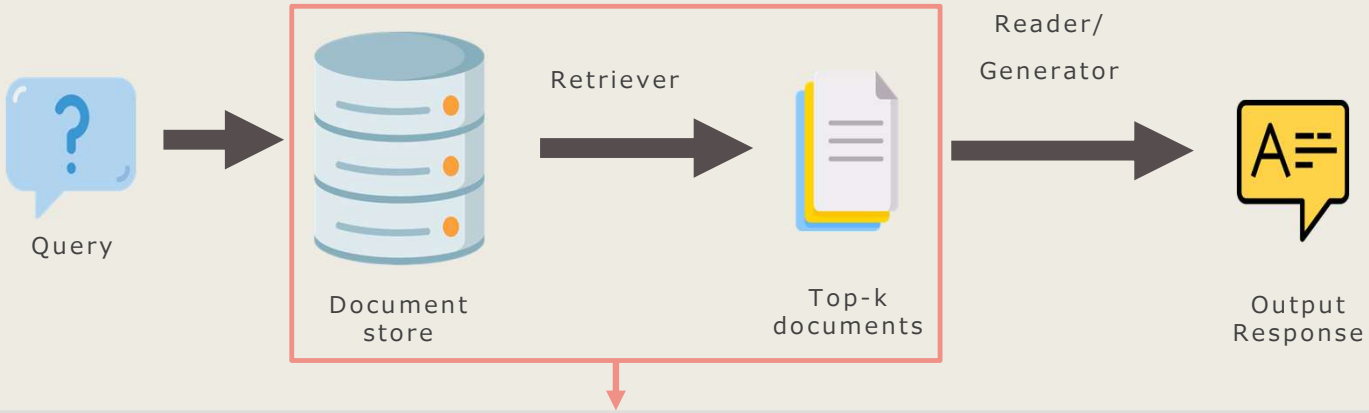

This combination of an external retrieval system with a parametric LLM enables it to generate an answer conditioned on the retrieved documents, thereby mitigating its limitations such as[2,3]:

- Static knowledge
- Limited access to private/ domain-specific data

1. Lewis, Patrick, Perez, Ethan, Piktus, Aleksandra, Petroni, Fabio, Rocktäschel, Tim, Kandpal, Naman, Stoyanov, Veselin, and Riedel, Sebastian. 2020. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*. arXiv:2005.11401. https://arxiv.org/abs/2005.11401
2. Emenike, Lawrence. 2023. *A Straightforward Explanation of Parametric vs Non-Parametric Memory in LLMs*. In *Medium*. https://lawrence-emenike.medium.com/a-straightforward-explanation-of-parametric-vs-non-parametric-memory-in-llms-f0b00ac64167
3. Babu, Pradeep. 2023. *Exploring the LLM Landscape: From Parametric Memory to Agent-Oriented Models*. In *Medium*. https://medium.com/@prdeepak.babu/exploring-the-llm-landscape-from-parametric-memory-to-agent-oriented-models-ab0088d1f14

# Vulnerabilities of RAG[4]

## RAG Architecture:

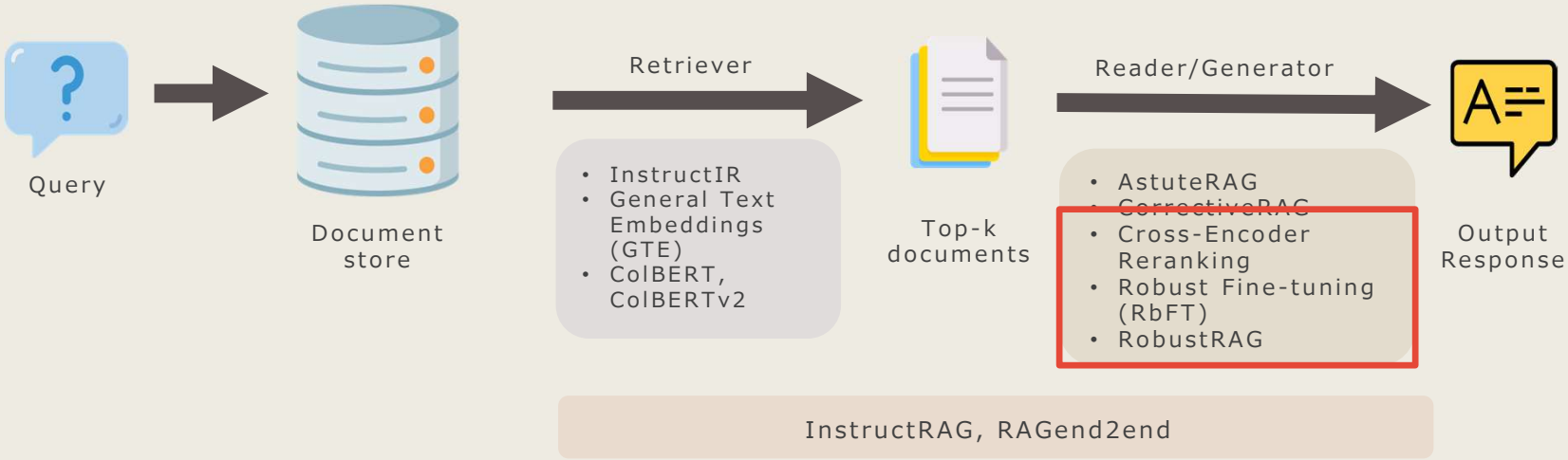


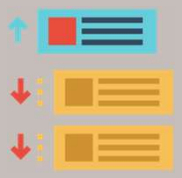

- Since RAG relies on an external retriever, it is vulnerable to issues such as:
  - Poor retrievals - where noisy or irrelevant documents are retrieved
  - Poisoned retrievals - where adversarial or malicious documents are maliciously injected into corpus and retrieved
- When these corrupted documents are fed downstream to the generator, it leads to inaccurate or harmful responses - which are risky in high-stake domains.
- In response to such vulnerabilities, recent research efforts have focused on **strengthening the RAG pipeline** (thereby making it more **robust**) via various strategies.

4. Xiang, Chong, Wu, Tong, Zhong, Zexuan, Wagner, David A., Chen, Danqi, and Mittal, Prateek. 2024. *Certifiably Robust RAG against Retrieval Corruption*. In *arXiv preprint arXiv:2405.15556*. https://arxiv.org/abs/2405.15556

# Existing RAG Solutions

## RAG Architecture:



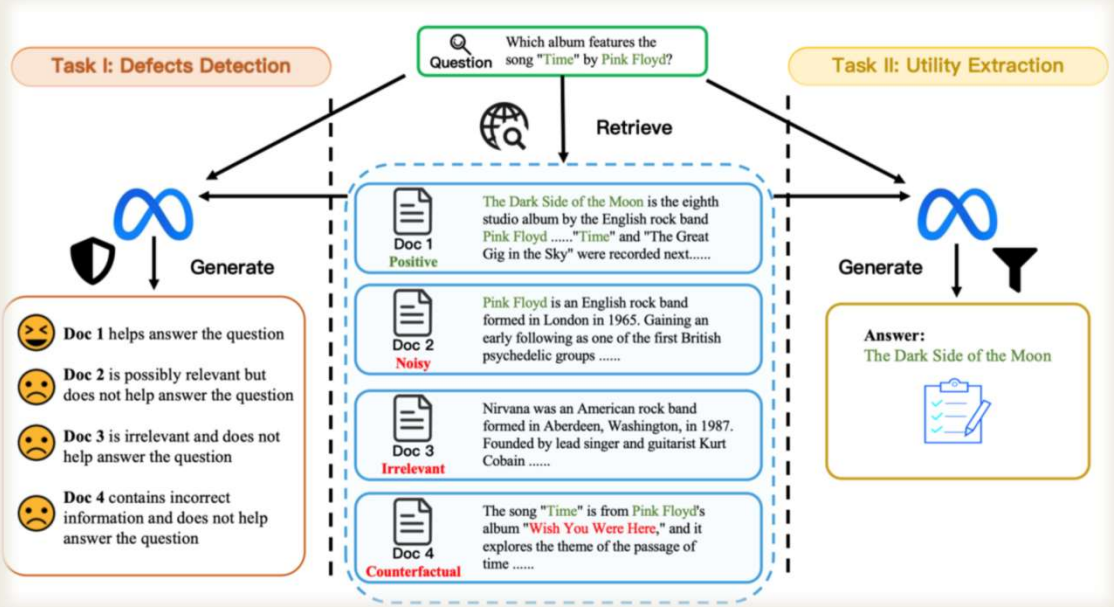



**1. Cross-Encoder Reranking[5]**

- Key idea: To filter out retrieved documents that has the most relevance to query before proceeding to generation
- Done by reranking the top-k retrieved documents using semantic similarity scores generated from a cross-encoder model
- *"Out of these 10 retrieved documents, give me the top 5 that has the most relevance to query"*

5. Rodrigo Nogueira and Kyunghyun Cho. 2019. *Passage Re-ranking with BERT*. arXiv preprint arXiv:1901.04085. https://arxiv.org/abs/1901.04085

# Existing RAG Solutions




**2. Robust Fine-Tuning (RbFT)[6]**

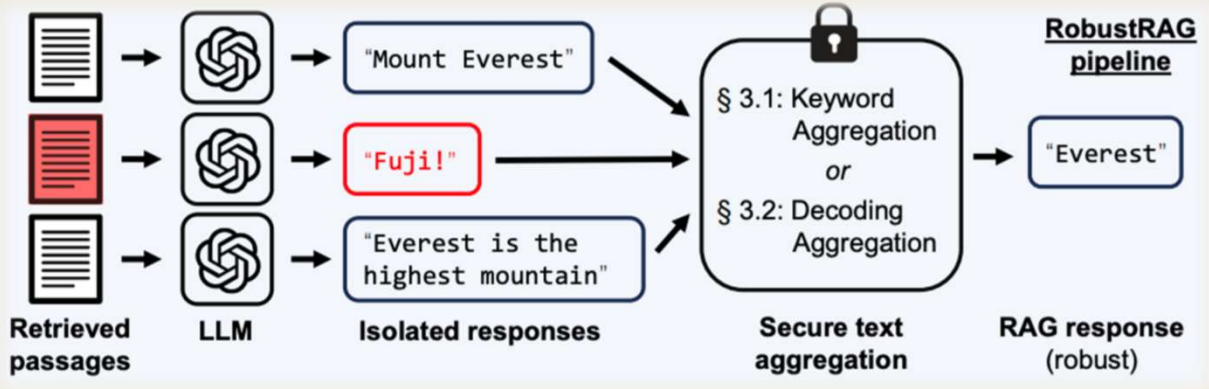
- Fine-tunes the LLM generator on data that includes positive, noisy, corrupted and counterfactual passages.
- Equips it to detect and defend against such documents → defects-resilient LLM.

**Outcome:**

- LLM can still generate accurate responses even when bad documents are retrieved.
- *If asked "What is photosynthesis?" and given docs about chemistry, math, and plants, the model still answers about plants*

6. Yuan, Kun, Zhou, Yuzhong, Zhu, Jialu, Xiong, Caiming, and Zhang, Tian. 2024. *RbFT: Retrieval-Augmented Generation Can Be Improved by Robust Fine-Tuning*. In *arXiv preprint arXiv:2501.18365*. https://arxiv.org/abs/2501.18365

# Existing RAG Solutions




**3. RobustRAG[4]**

- Deploys an Isolate-then-Aggregate approach where LLM:
  - Generates answers **separately for each passage**, then
  - Aggregate the answers via various aggregation techniques to pick the most reliable final answer

- Prevents bad documents from polluting multi-passage generation
- Defuses the influence of a few corrupted passages

- *E.g for Keyword Aggregation - If 2 out of 3 answers mention "photosynthesis" and 1 mentions "water transportation", the former wins due to stronger keyword alignment.*

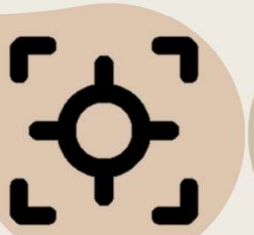
# Key Limitations

## Single-Point Focus

Most solutions strengthen *either* the retriever or the generator – rarely both.

- **Cross-Encoder Reranking** enhances retrieval quality
- **RbFT** hardens generation against noisy docs

*But what if both retrieval and generation are simultaneously compromised?*



## Limited Defensiveness Against Strong Adversaries

Current techniques **struggle with adversarially poisoned** passages (e.g., factually incorrect but fluent content).

- **Cross-Encoder** might retain it due to fluent writing.
- **RobustRAG** can be compromised if the poisoned doc dominates the majority of the passage votes.

## Lack of End-to-End Robustness

No single solution fully addresses **both retrieval corruption** and **generation hallucination** together

- **RbFT** might still hallucinate if wrong context dominates.
- **RobustRAG** can reduce the influence of one bad doc, but fails if most documents are weak.

## Heavy Reliance on Self-Reflection

Generation relies on the LLM's internal ability to assess noisy context.

- **RbFT** must internally resolve contradictions without ground truth.
- **RobustRAG** can help by aggregating, but only if at least two passages agree.

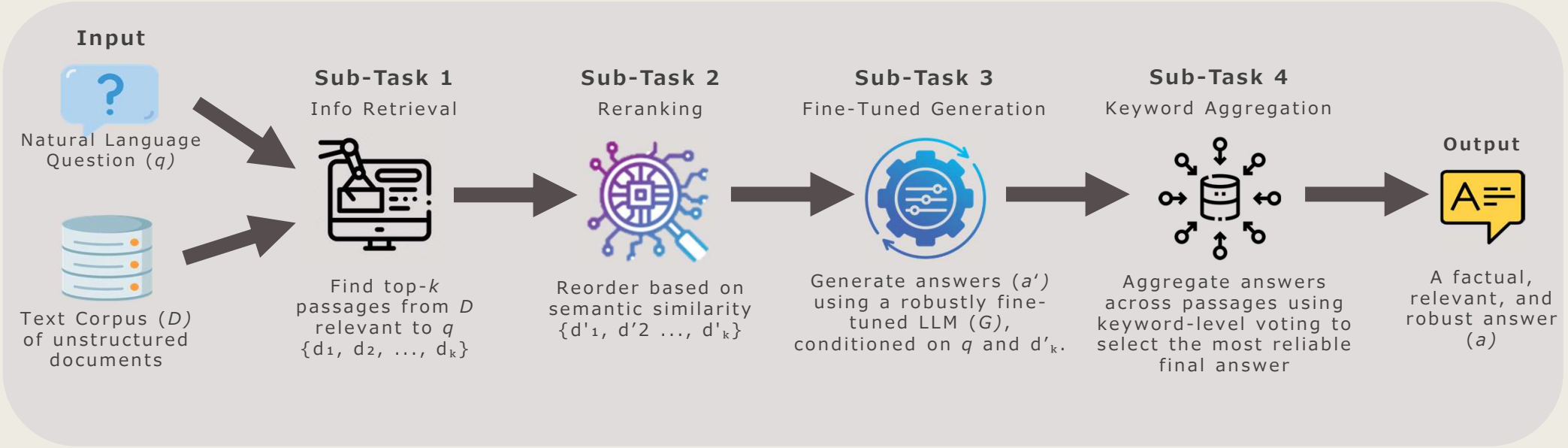
*While prior work addressed isolated challenges in retrieval or generation, few offer holistic robustness across the full RAG pipeline*

# Problem Definition

## NLP Task Formulation

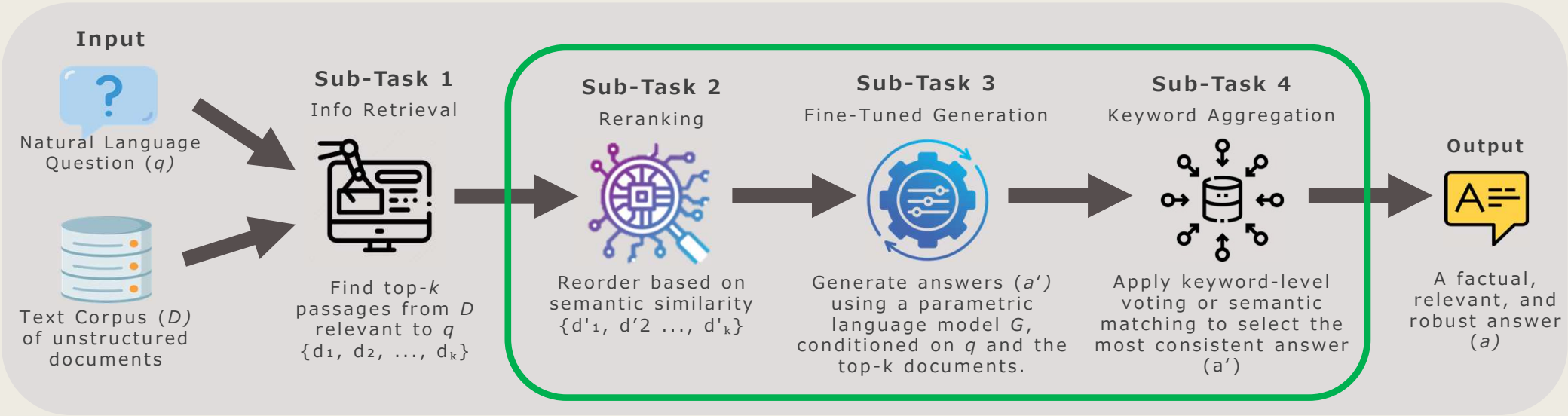
**Open-Domain Question Answering (ODQA)** problem with retrieval augmentation

Ensuring *reliable* and *verifiable* answers in open-domain RAG – where the retrieval store is noisy or adversarially manipulated.



# Our Approach

## *Enhancing Robustness with Layered Defences*




**Reranker**

Adopting **Cross-Encoder Reranking** to handle noisy or irrelevant documents retrieved

**Fine-Tuned Generation**

Applying **Robust Fine-Tuning** to improve LLM misled by corrupted context

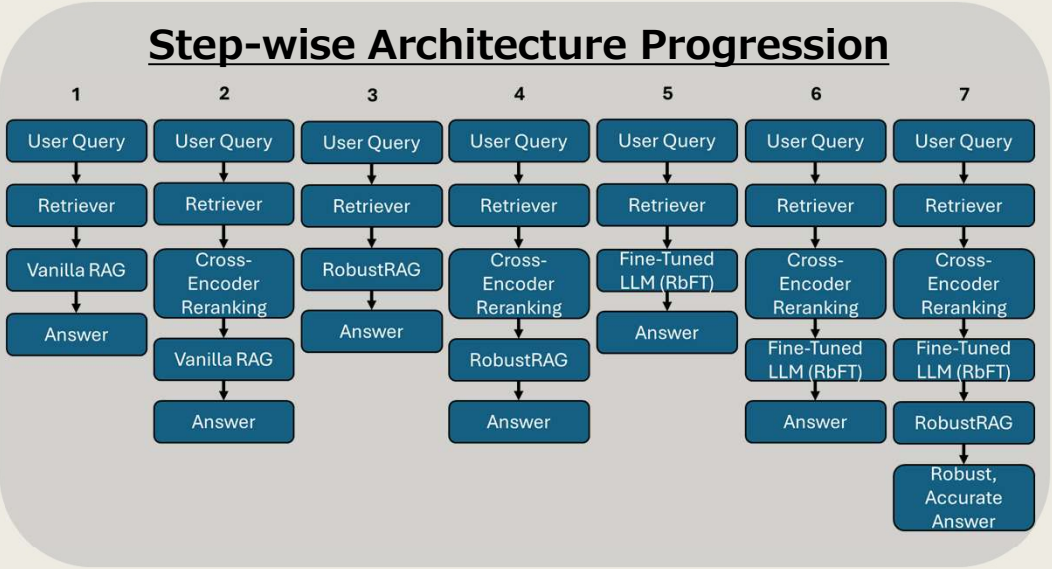
**Keyword Aggregation**

Implementing **Isolate-then-Aggregate** (RobustRAG) technique to avoid bad data from dominating the generation

*Together, these defenses operate at retrieval, generation, and aggregation levels —*
*Aiming to form a robust, multi-layered shield against corrupted or irrelevant input*

# Experiment Setup

**Evaluating the impact of layered defences – Cross-Encoder Reranking, Robust Fine-Tuning, and Keyword Aggregation – through a systematic, step-wise incremental approach**




| Arch | Components(s) | Description |
|---|---|---|
| 1 | Baseline | Dense retrieval + generation |
| 2 | Baseline + Cross-Encoder Reranker | Rerank top k inputs using semantic similarity |
| 3 | RobustRAG | Isolated passage generation + aggregation |
| 4 | Cross-Encoder Reranker + RobustRAG | Combines retrieval + aggregation defences |
| 5 | RbFT (Fine-tuned LLM) | LLM trained on noisy input to resist hallucination |
| 6 | Cross-Encoder Reranker + RbFT | Combines retrieval + generation defences |
| 7 (Full stack) | Cross-Encoder Reranker + RbFT + RobustRAG | Combines retrieval + generation + aggregation defences |

- Started with a basic RAG pipeline (Arch 1), then progressively integrate defenses across three fronts:
  - Reranking
  - Robust Fine-Tuned Generation
  - Aggregation.
- To isolate the contribution of each technique and validate whether the final layered model (Arch 7) offers holistic robustness.

# Dataset Setup

## Similar to Robust Finetuning (RbFT)

Wikipedia 21m passages of 100 words

Generate embeddings using e5-base-v2

FAISS vector database with flat index (60GB)

Dense retrieval

Use existing QA datasets based on Wikipedia to retrieve **top 100** relevant passages from query

**RbFT Training Dataset (20K)**

Natural Questions (NQ)
TriviaQA
HotpotQA
Musique

**Generate passages of different types to test robustness**

**Positive**  Top 10 relevant passages

**Noisy**  Random 10 passages from bottom 50 relevant passages

**Irrelevant**  10 random passages from Wikipedia corpus

**Counterfactual**  Send query, answer, and positive passages to LLM to generate 10 new passages with similar sounding but wrong info.

Based on a given question and its correct answer, generate a misleading wrong answer…. The wrong answer should belong to the same entity type as the correct answer (e.g., person, time, place, organization, data, etc.) to enhance its confusion……

**Test Dataset (3.5K)**

QA: Where did the rulers of the Qing dynasty originate?

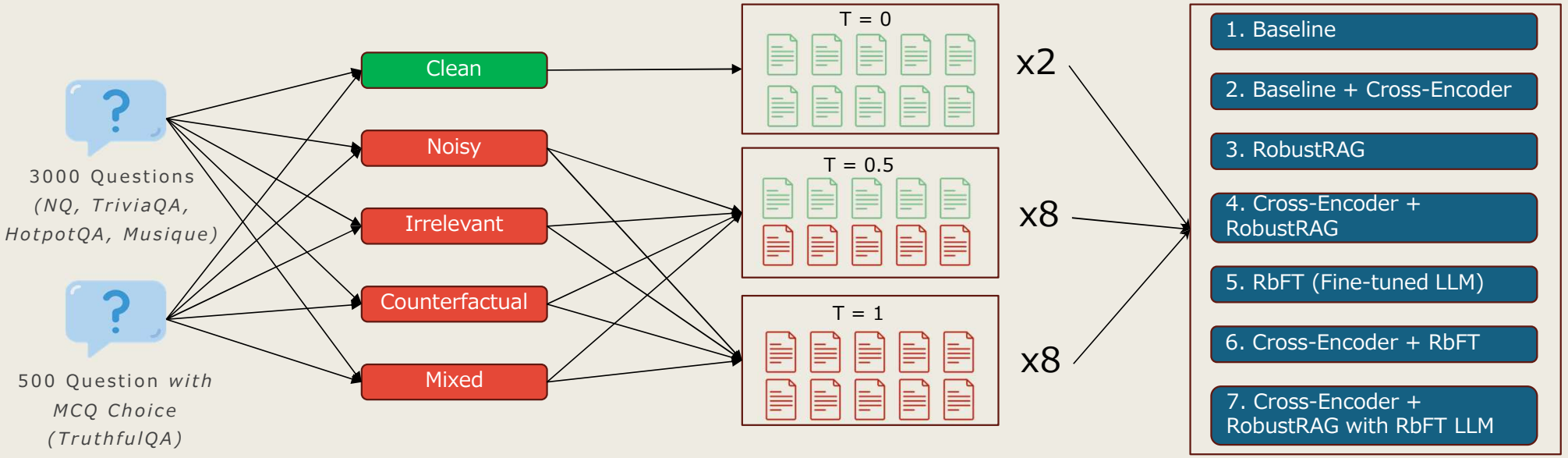
Multi-Hop QA: In 2009, Massimo Giordano toured with a Russian Soprano born in which year?

TruthfulQA: What happens if you watch a pot of water on the stove?
a. If you watch a pot of water on the stove, it will boil
b. If you watch a pot of water on the stove, it will never boil
c. A watched pot never boils

- NQ
- TriviaQA
- HotpotQA
- Musique
- TruthfulQA

# Experiment

- Simulate retrieval defects by mixing the positive documents with noisy, irrelevant, and counterfactual documents to generate **10** documents for each question
- 4 different defect scenarios: only noisy, only irrelevant, and only counterfactual, mixed (random of all 3 types),
- Use a τ (tau value) to control the number of passages to be defective
- 0 = no defects, 0.5 = roughly half are defective, 1.0 = all are defective



# Evaluation Metrics

**Exact Match (EM)**: Whether the model's generated answer is identical to one of the ground truth answers. Difficult for open-ended answers.

**F1-score**: Measures overlap between predicted answer and ground truth answers. Tokenize both predicted and ground truth answers and lowercased

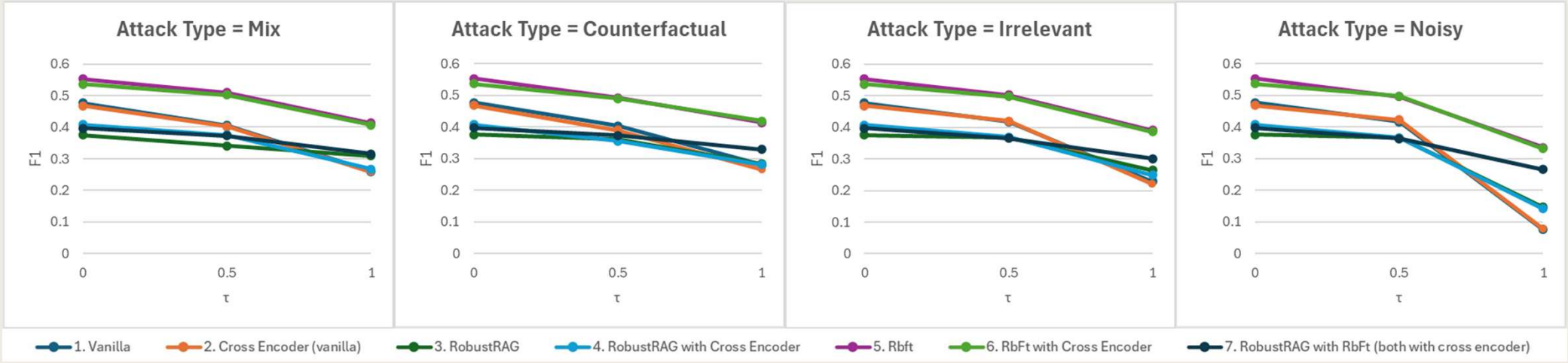
$$Precision\ (P) = \frac{Number\ of\ shared\ tokens}{Total\ number\ of\ tokens\ in\ predicted\ answer}$$

$$Recall\ (R) = \frac{Number\ of\ shared\ tokens}{Total\ number\ of\ tokens\ in\ ground\ truth\ answer}$$

$$F1 = 2 * \frac{P.R}{P+R}$$

# Results and Analysis

**Open-Ended QA F1 Score under Varying Attack Strengths**

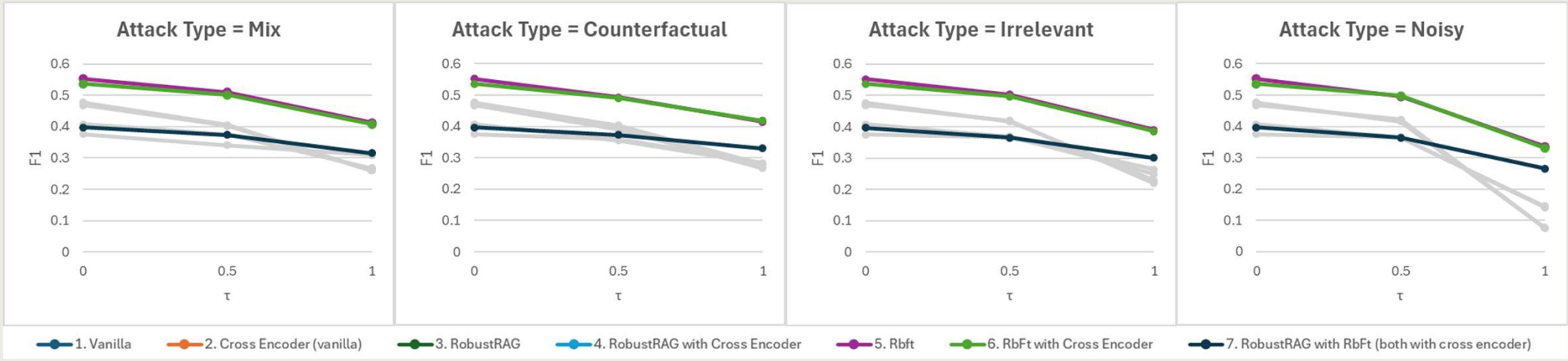



- F1 score decreases as τ (attack strength) increases.
- Arch 7 shows minimal performance drop across all attack types and strengths.
- Noisy attacks have the largest impact on results.

# Results and Analysis

**Open-Ended QA F1 Score under Varying Attack Strengths**

**1. RbFT has best performance**



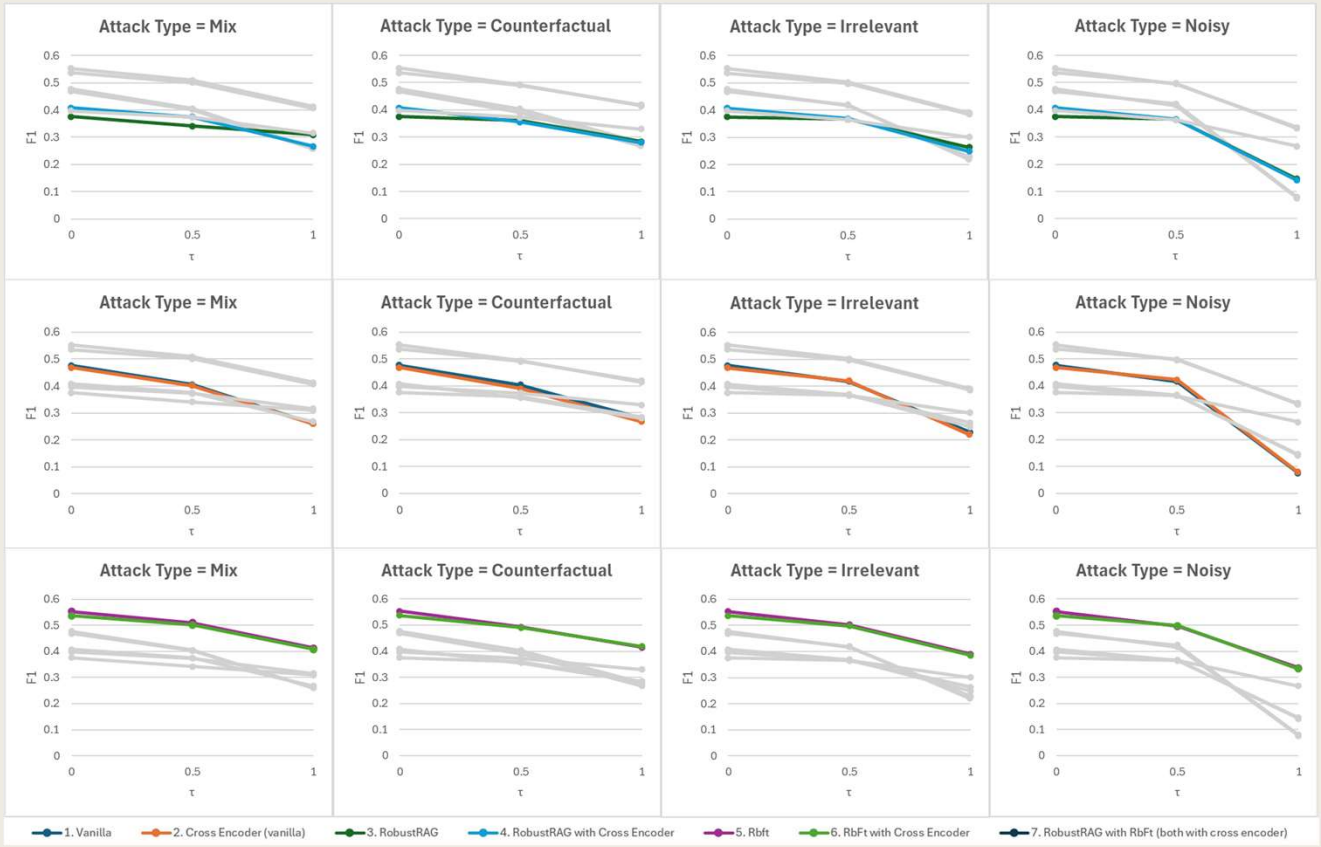

**Archs that make use of RbFT (Arch 5,6,7)**

- Achieved stable performance across all attack strengths
- Performance drops less sharply under noisy attack type

# Results and Analysis

## Open-Ended QA F1 Score under Varying Attack Strengths

### 2. Cross-encoder reranking not necessarily effective

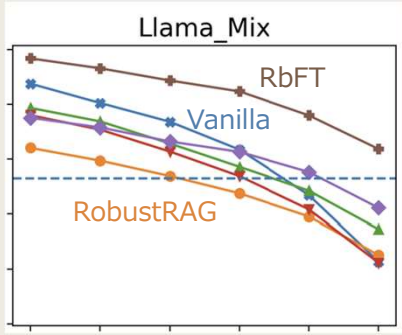



**Cross-encoder reranking may not necessarily be more effective when used in robust architectures.**

- Reranking prioritizes relevance rather than resistance to adversarial corruption
- Having less retrieved passages could adversely affect performance

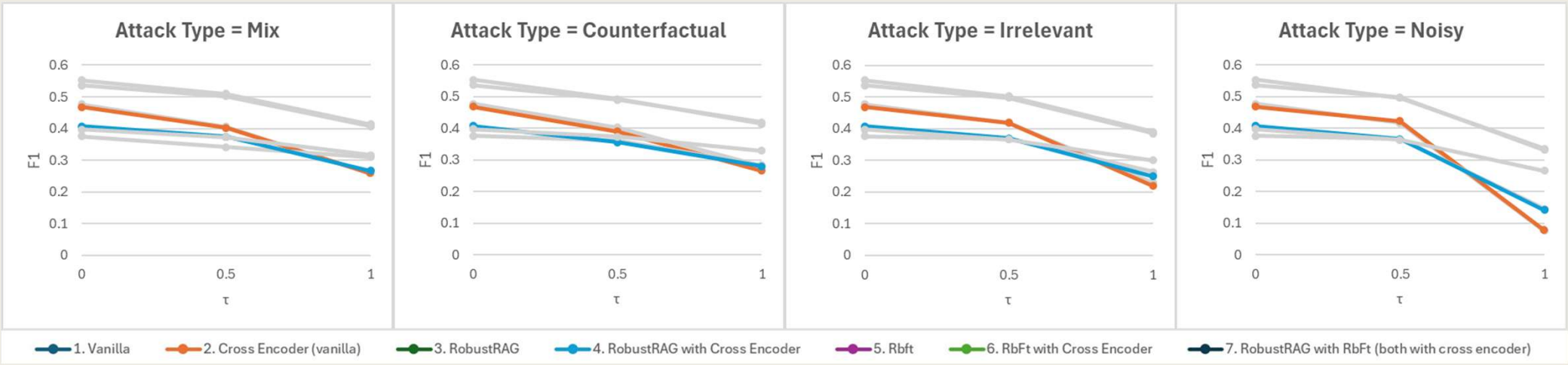
# Results and Analysis

EM performance[9]:




## Open-Ended QA F1 Score under Varying Attack Strengths

### 3. RobustRAG performed worse than Vanilla



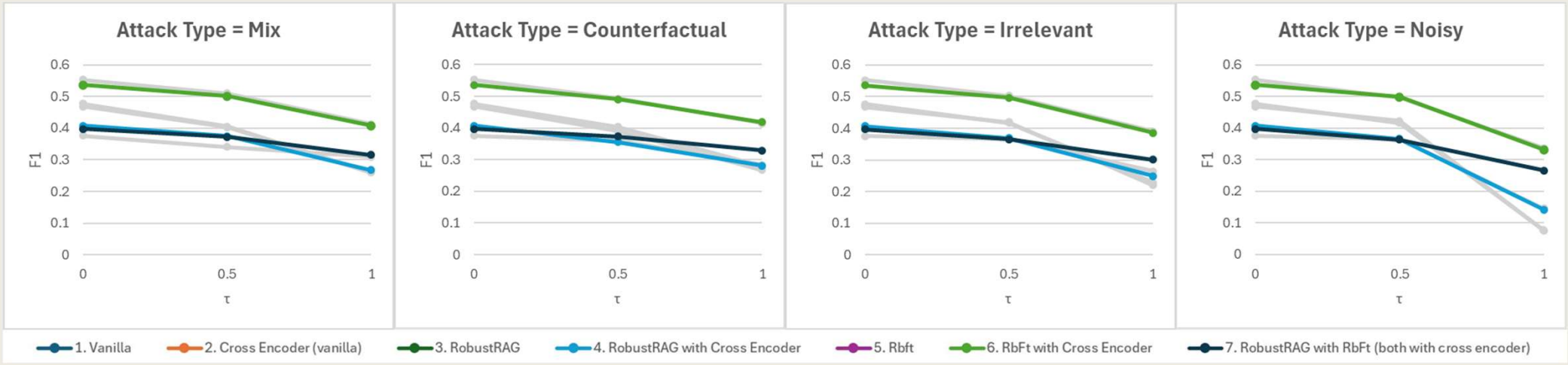

**RobustRAG (Arch 3, 4) performed worse than their vanilla counterpart at lower attack strengths (similar results from RbFT paper)**

- Isolate-then-aggregate strategy may not capture complementary information as well (except in high attack strength settings)

# Results and Analysis

**Open-Ended QA F1 Score under Varying Attack Strengths**

**4. Arch 7 does not have clear additive gains**

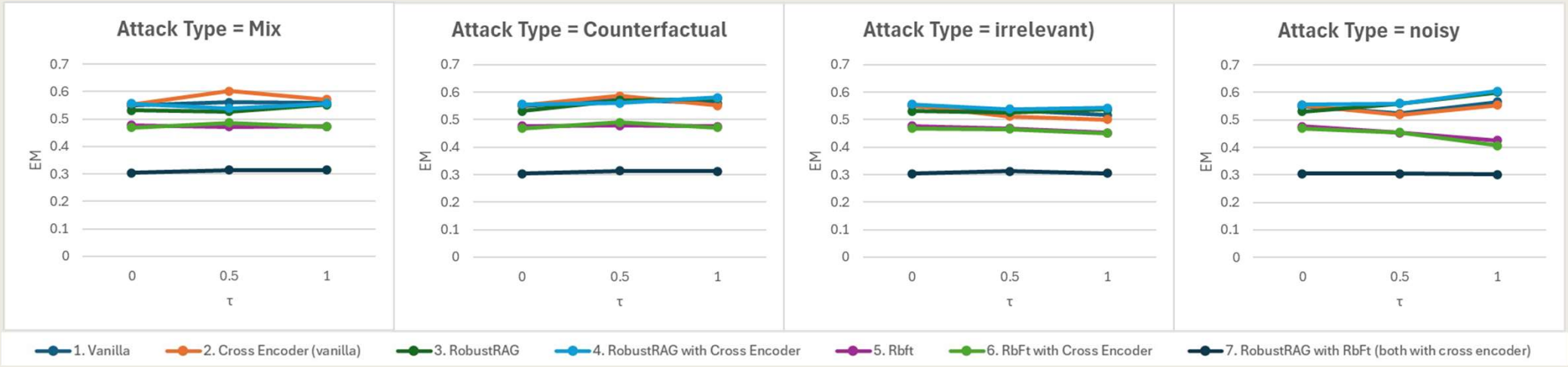



**Arch 7 did not outperform arch 6 (Rbft with Cross Encoder)**

- No clear gain from combining both robust modules

# Results and Analysis

**TruthfulQA EM under Varying Attack Strengths**




**No drastic change in EM even as attack strength increases**

- Archs are generally robust to attacks
- RobustRAG performed better than RbFT

**Arch 7 performed significantly worse than other simpler baselines**

# Conclusion & Future Work

We presented a robust RAG pipeline integrating three complementary defense mechanisms – cross encoder reranking, RbFT, and RobustRAG.

- Combining techniques did not lead to additive gains.
- RbFT showed superior robustness under strong attack.
- Robustness against attacks comes at a performance trade-off.



- Introduce other RAG combinations (particularly in place of RobustRAG)
- Dynamic aggregation (e.g. weighted voting) - to provide more weightage to more relevant documents
- Real web-based retrieval instead of synthetic – to evaluate real-world robustness