CHAPTER 5

OBJECTIVE

5.1 Primary Goal

The primary goal of this project is to architect and implement an intelligent, multi-stage query processing system that proactively assesses hallucination risks in real-time. Unlike traditional static pipelines, this system utilizes a dynamic routing mechanism to direct user queries to the most appropriate response pathway—optimizing for a balance between computational efficiency, response latency, and factual reliability. The system is specifically designed to function within the constraints of a "Small Local Model" environment (approx. 1B parameters), ensuring accessibility and feasibility on standard hardware.

5.2 Specific Technical Objective

5.2.1. Objective 1: Development of the Internal Confidence Module

The first technical objective is to engineer a robust Internal Confidence Module that serves as the initial "gatekeeper" for the system. This module operates strictly on the local model's internal states to evaluate uncertainty without incurring the latency of external retrieval. It computes a confidence score based on three distinct signal layers:

- **Semantic Alignment:** This component evaluates the consistency of the generated text by analysing semantic variations across multiple sampled outputs. It detects instances where the model vacillates between different meanings, indicating low confidence.
- **Internal Convergence:** This metric measures the statistical agreement of the model's internal probability distributions (logits). High convergence suggests the model is certain about its next-token predictions, whereas low convergence indicates ambiguity.

- **Learned Confidence:** This involves a trained regression layer or calibration mechanism that outputs a scalar confidence score derived directly from the model's training metrics and hidden states.

Risk Factor	**Definition**	**Detection Method (Phase 2 Plan)**
Context Scarcity	When the retrieved information is too sparse to answer the query.	Algorithms to measure information density in retrieved context.
Context Conflict	When source documents contradict one another, confusing the model.	Integration of Natural Language Inference (NLI) to detect contradictions.
Domain Mismatch	When the retrieved documents are irrelevant to the user's specific query.	Semantic similarity checks between query embeddings and document embeddings.

Table 5.1 Types of Signals

5.2.2. Objective 2: Implementation of a Conditional External Reality Check

The second objective is to implement a secondary validation layer known as the External Reality Check. Crucially, this module is designed to be conditional—it is triggered only when the Internal Confidence Module returns an ambiguous ("Medium") score. This architectural decision minimizes unnecessary API calls and latency. When activated, this module assesses the quality of retrieved external data against three key risk factors:

Context Scarcity: Detects scenarios where the retrieved information is too sparse or insufficient to definitively answer the user's query.

Context Conflict: Utilizes Natural Language Inference (NLI) techniques to identify situations where source documents contradict one another, which typically leads to model confusion.

Domain Mismatch: Performs semantic similarity checks to flag instances where retrieved documents are thematically irrelevant to the specific user query.

5.2.3. Objective 3: Engineering a Dynamic Routing & Decision Logic

The third and central objective is to develop the Dynamic Routing System, a logic-driven engine that orchestrates the flow of data based on the scores derived from the previous two objectives. The routing logic is defined by the following decision matrix:

Pathway A: Direct Generation (High Internal Score) If the Internal Confidence Module returns a High Score, the system determines that the local model possesses sufficient internalized knowledge. The query is routed immediately to the Small Local Model for generation. This is the most efficient pathway, offering the lowest latency.

Pathway B: Cloud Escalation (Low Internal OR Low External Score) This pathway acts as a failsafe mechanism. It is triggered in two specific scenarios:

Immediate Fail: If the initial Internal Score is Low**.**

Secondary Fail: If the Internal Score was Medium, but the subsequent External Reality Check yielded a Low Score (indicating poor retrieval quality). In both cases, the query is deemed too high-risk for the local system and is escalated to a Large Cloud Model (e.g., GPT-4 or similar) to ensure response quality.

Pathway C: RAG Generation (Verified Augmentation) This pathway is reserved for queries where the local model is uncertain (Medium Internal Score) but high-quality external data is available (High External Score). The system proceeds with Retrieval-Augmented Generation (RAG), grounding the response in the verified external context.

5.2.4. Objective 4: Creation of a Custom Hallucination Benchmark

The fourth objective focuses on data infrastructure. To rigorously train and validate the risk assessment modules, a custom labelled dataset will be constructed. This process involves:

- Aggregating prompts from established QA benchmarks such as Natural Questions and Trivia QA.
- Generating responses using the target small language models.
- Manually or automatically labelling these responses for specific hallucination types (e.g., factual fabrication, reasoning errors) to establish a "ground truth" for the confidence modules**.**

5.2.5. Objective 5: System Integration and Performance Benchmarking

The final objective is the holistic integration of the Internal Module, External Reality Check, and Routing Logic into a unified API or application pipeline. The system will be benchmarked against standard static RAG pipelines to demonstrate improvements in:

Accuracy: Reduction in hallucination rates**.**

Latency: Average time-to-response (proving the efficiency of the "Direct Generation" path).

Computational Cost: Reduced reliance on expensive cloud model tokens.

5.3. Project Scope

The scope of this research is strictly defined within the domain of text-based, knowledge-intensive Question Answering (QA). The architecture focuses on optimizing the performance of moderately-sized open-source Large Language Models (LLMs), specifically verifying their utility in resource-constrained environments where full-scale cloud dependence is not feasible.

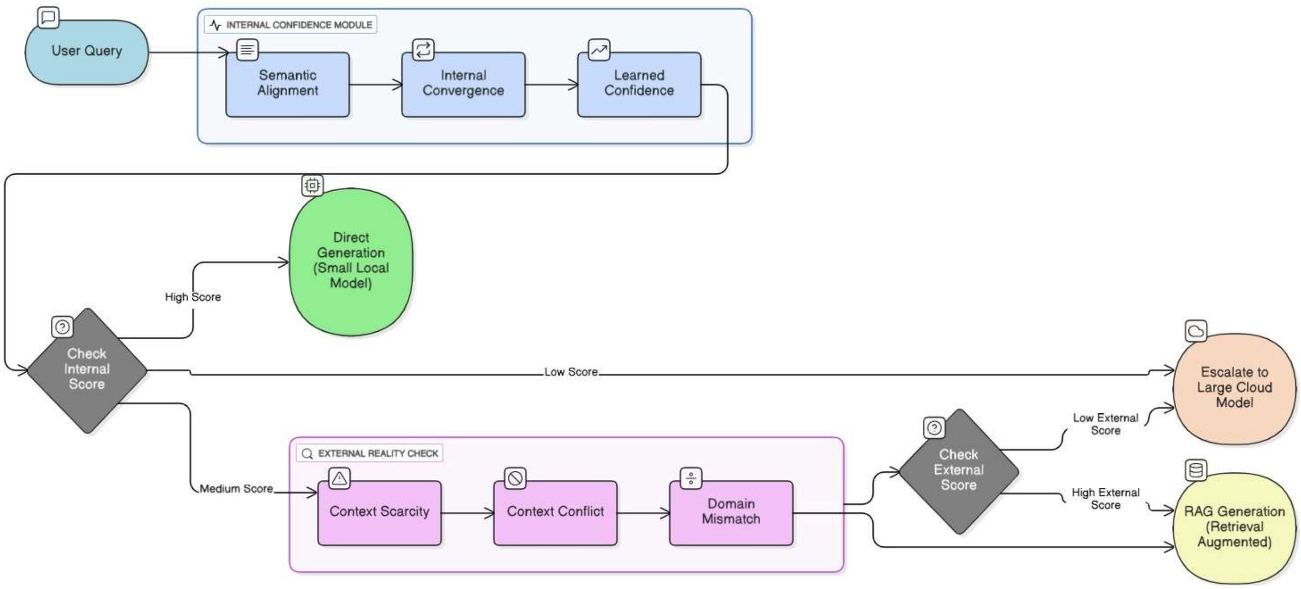

Figure 4.1 Conditional Hybrid Confidence System (3-Interval)