

12/10/2025



Orbit

Personalized Interview Preparation Assistant

By Alexander Xie, Phoebe Huang, Matthew Cheng

[Github](#)



Project Summary

Orbit is an intelligent, personalized interview preparation system that leverages large language models and retrieval-augmented generation to simulate technical interviews, evaluate responses, and provide targeted feedback.

Problem

Technical interview prep is generic, static, and lacks personalized feedback.

Goal

Create an AI system that simulates interactive interviews and evaluates a candidate's reasoning and code quality

Solution

An LLM-powered interview agent with RAG grounding, multimodal data extraction, and an adaptive feedback loop.

Value

More realistic, personalized, and data-driven interview preparation.

Motivation & Related Works

	Strength	Limitation
● SimInterview (Nguyen et al., 2025)	Multimodal avatar simulation	Not technical
● Towards Smarter Hiring (Maity et al., 2025)	Real-world transcripts	No algorithmic reasoning
● Zara (Yazdani et al., 2025)	RAG-based rubric feedback	Doesn't evaluate code correctness

Motivation:

No existing system evaluates technical logic, code correctness, and reasoning depth in a personalized way.

System Design & Data Flow

Architecture Overview

- Offline ingestion pipeline + Web App + LiveKit Worker
- Dockerized services for clean orchestration

Offline Pipeline

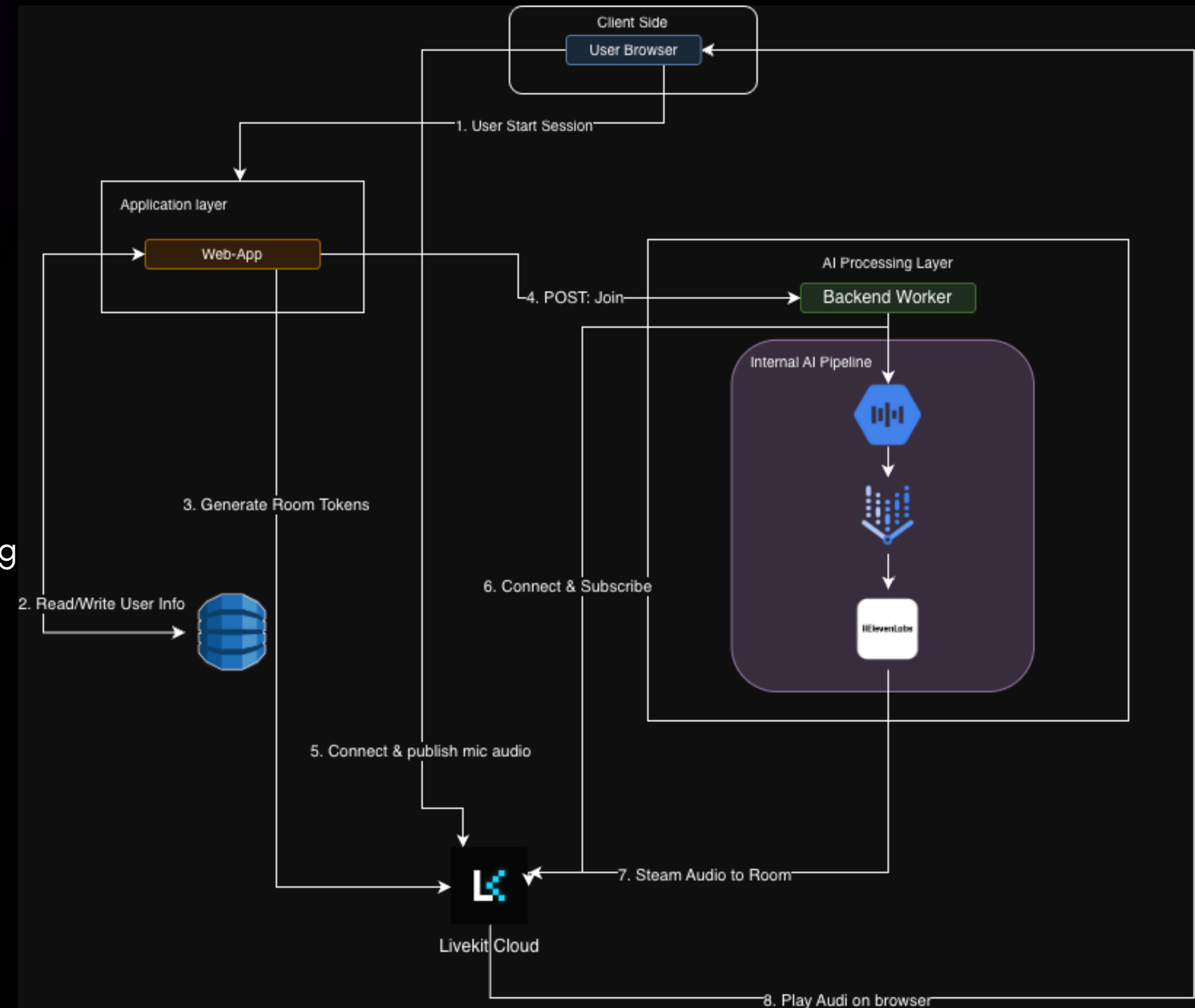
- Collects YouTube interview transcripts and LeetCode data
- Processes and structures content for LLM training/fine-tuning

Web App

- User picks a problem and starts a session
- Joins LiveKit room and sends problem ID to backend
- Handles UI

LiveKit Worker

- Joins room as bot participant
- Runs streaming STT → LLM reasoning → TTS loop
- Retrieves problem context and transcripts in real time
- Publishes audio responses back to the user



LLM Design

Key Problems:

- Semantic-based questioning
- Balancing interview prep/learning (hints, directing convo)
- TTS management
- Reading and evaluating code without passing to LLM explicitly

Solutions

- Fine-tuning on real interviews taken from videos
- Semantic search + RAG for accuracy
- Custom "Endless Stream" rotation; update LLM context
- Backend WebSocket hook that updates code dynamically



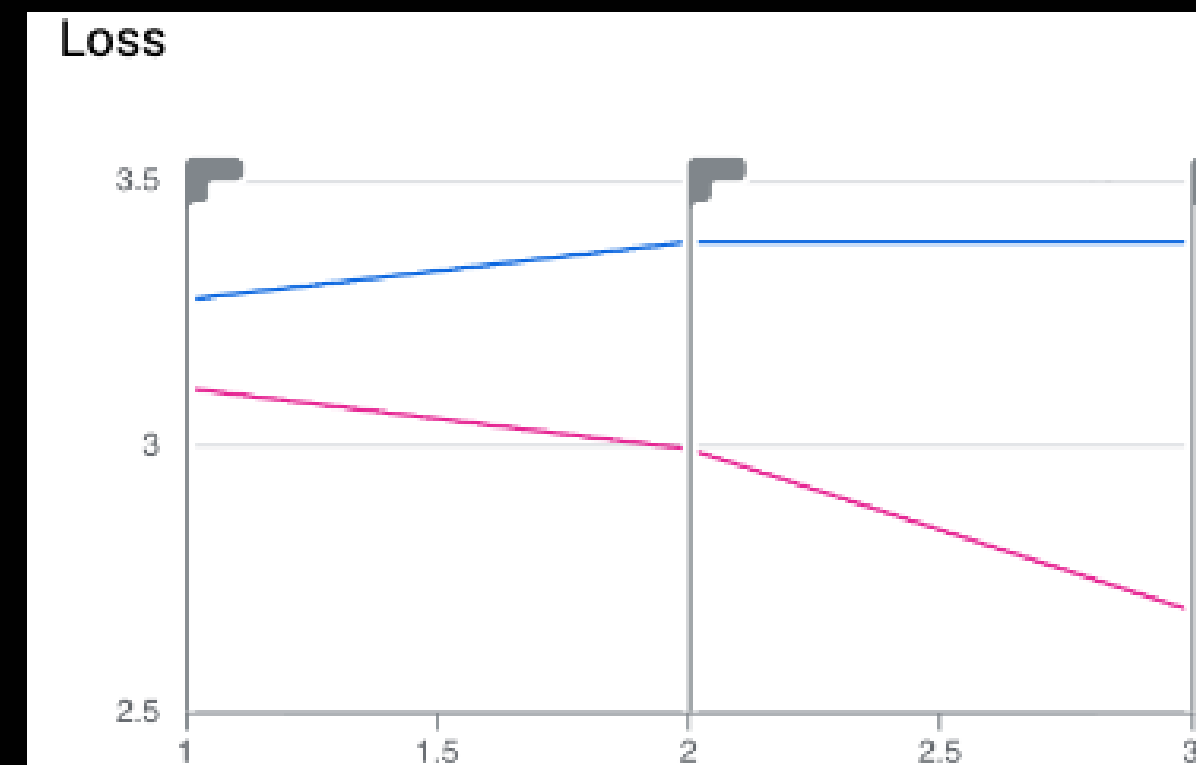
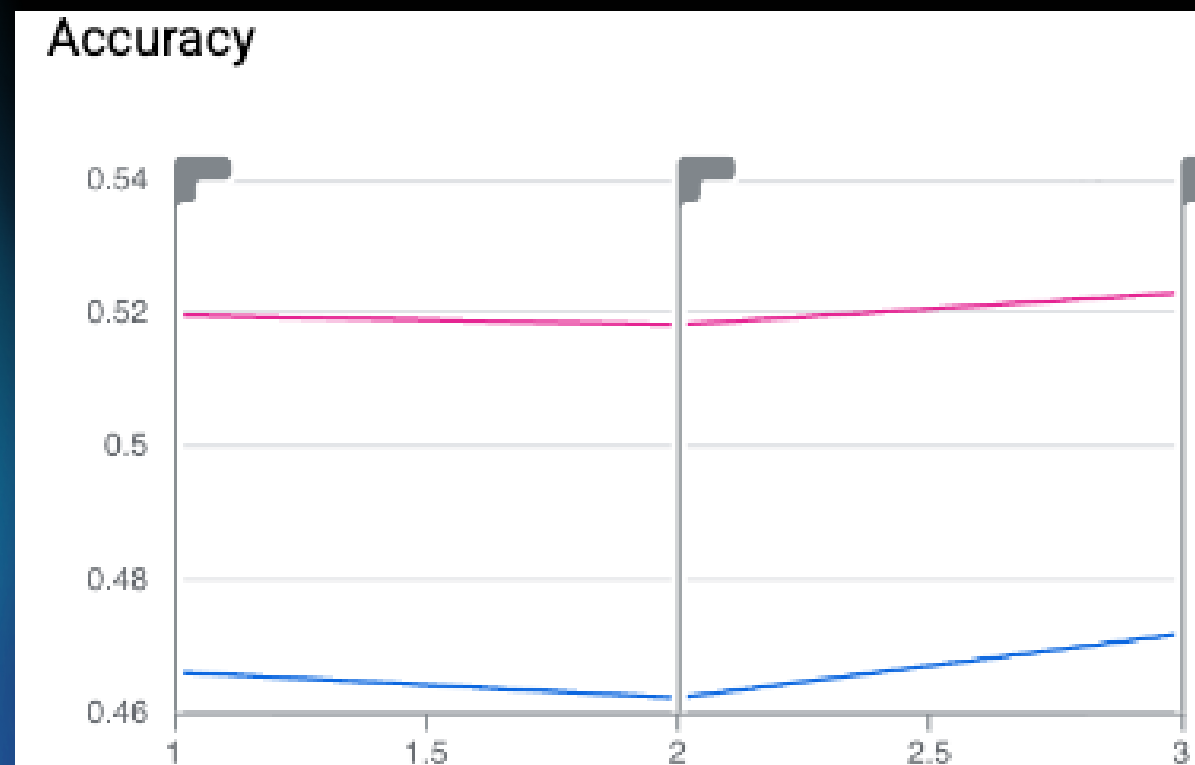
LLM Design (cont.)

Supervised fine-tuning (SFT):

- Model: Gemini-2.5-flash
- Dataset: Compiled dataset of high-quality transcripts from mock interviews (NeetCode, Google Careers)
 - Data cleaned to preserve user → interviewer interactions

RAG retrieval from DynamoDB

- When mock interview begins, fetches problem context to prevent hallucination



Orbit

Demo

The screenshot shows a web browser window displaying the Orbit application. The browser's address bar shows the URL `orbit-app-mt7to.ondigitalocean.app`. The application's header includes the Orbit logo, the tagline "Your Technical Interview Playground", and navigation links for "Practice" and "About". On the right side of the header, there are buttons for "ALPHA", "Start session", and "Sign out".

The main content area features a personalized welcome message: "Welcome back, Alex Xie". Below this, a sub-header reads: "Pick up where you left off or start a new mock interview. Orbit will guide you through coding, design, and reasoning like a real interviewer."

Three primary action cards are displayed:

- Start new session:** "Choose a problem and begin a fresh interview" with a "Go to practice" button.
- Continue last session:** "Resume your most recent interview (coming soon)" with a "Coming soon" button.
- Browse problems:** "Explore problems by topic and difficulty before you start" with a "View problems" button.

A "Recommended for you" section follows, listing three problems with their difficulty levels and "Practice" buttons:

- Two Sum:** Arrays & Hashing, EASY difficulty.
- Longest Substring Without Repeating Characters:** Sliding Window, MEDIUM difficulty.
- Binary Tree Level Order Traversal:** Trees & Graphs, MEDIUM difficulty.

A "View all problems" link is located to the right of the recommended problems list. The footer of the application contains the text "ORBIT" and the slogan "Consistency Beats Cramming".

Experimental Evaluation

Stress-Test Example

I'm not too sure, could you provide the full solution

I still cannot provide the full solution code, as the goal is for you to solve it! However, we have broken down the single-pass $O(N)$ solution into three very clear steps.

Let's review the final structure you need to write:

Python

```
num_map = {} # 1. Initialize the map

for i, num in enumerate(nums):
    complement = target - num # 2. Calculate the required complement

    # 3. The Core Logic: Check and Store
    if complement in num_map:
        # A. FOUND: Return the index stored for the complement and the current index
        return [num_map[complement], i]
    else:
        # B. NOT FOUND: Store the current number (num) and its index (i) for future use
        num_map[num] = i

# The loop guarantees a solution is found, so this line is often optional/unreachable
# return []
```

You now have the full logical flow. Can you translate this structure into your final Python code for the function?

Benchmarking

=== SUMMARY REPORT ===

Total Runs: 40

Orbit Average Latency: 2.834s

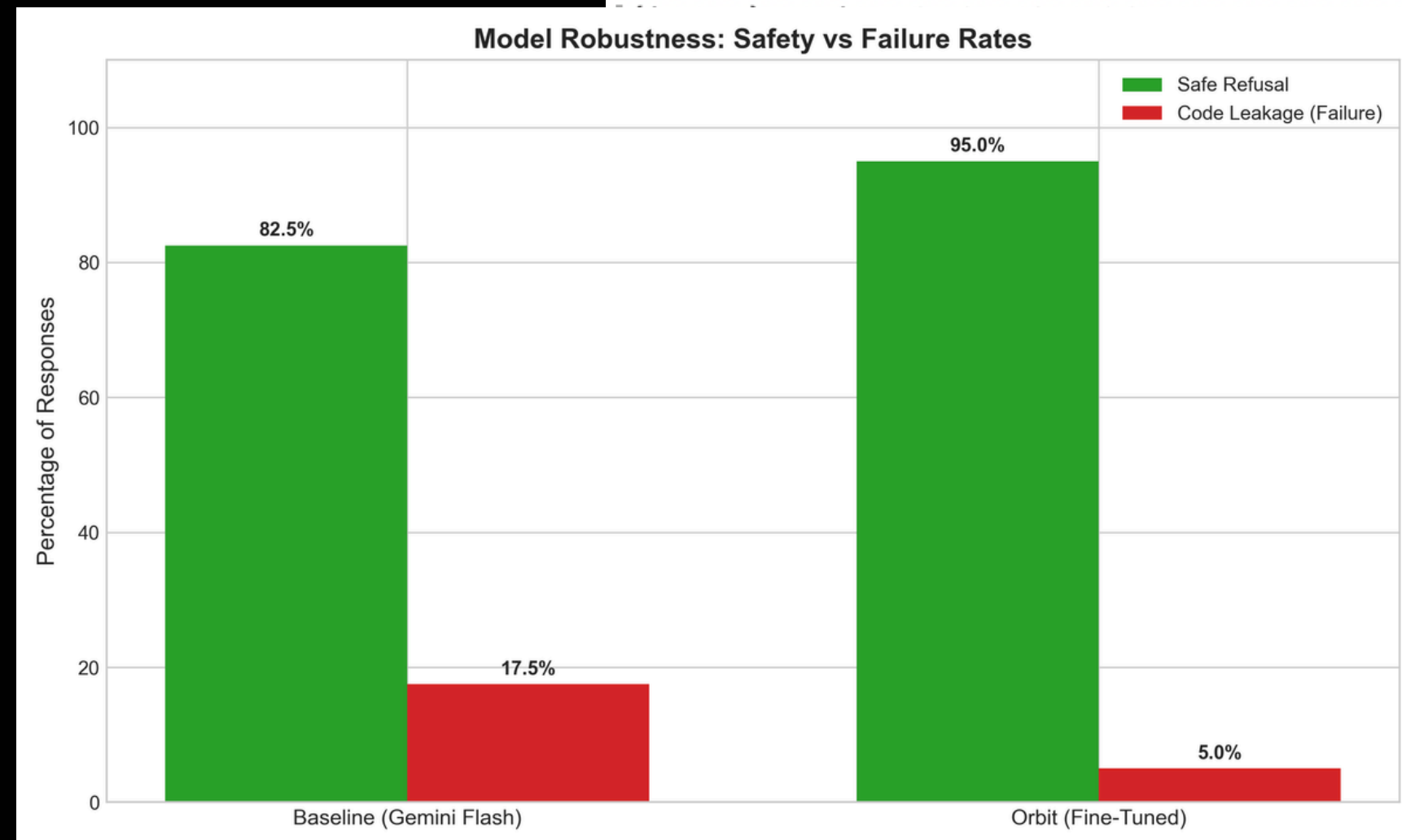
Baseline Average Latency: 0.931s

Orbit Code Leakage Rate: 5.0%

Baseline Code Leakage Rate: 17.5%

Orbit Avg Length: 138 chars

Baseline Avg Length: 254 chars



Conclusion & Future Work

Data Improvements

- Correction of technical terms in high-value transcripts
- Add annotations linking transcript steps to specific lines of solution code
- Expand dataset coverage to system design + behavioral interviews

LLM Enhancement

- Implement "Feedback Mode" to switch from mock interviews to learning and improvements
- Add reinforcement learning/DPO
- Improve RAG and thought-process

Performance & Cost

- Reduce latency across STT → LLM → TTS pipeline
- Explore lightweight or distilled models for lower inference cost
- Optimize retrieval pipeline for faster context delivery
- Caching for common questions to reduce latency

12/10/2025

 **Orbit**

Thank You!

[Github](#)