

# Cache Verified Semantic Prompt Caching

Luis Gaspar Schroeder, Aditya Desai, Alejandro Cuadron, Kyle Chu, Shu Liu,  
Mark Zhao, Stephan Krusche, Alfons Kemper, Matei Zaharia, Joseph E. Gonzalez



**Berkeley**  
UNIVERSITY OF CALIFORNIA



**Stanford**  
University



**ETH** zürich

LLMs are Powerful  
... but **Expensive** and **Slow**

How to **Reduce** LLM Inference  
**Latency** and **Cost** from a Systems  
Perspective?

## Application



Analytics



Chatbots



Agents

## Semantic Caching



Microsoft



databricks

Reduce

- Latency (100x)
- Cost (10x)

## Open-Source Inference



## Closed-Source Inference



ANTHROPIC

Grok

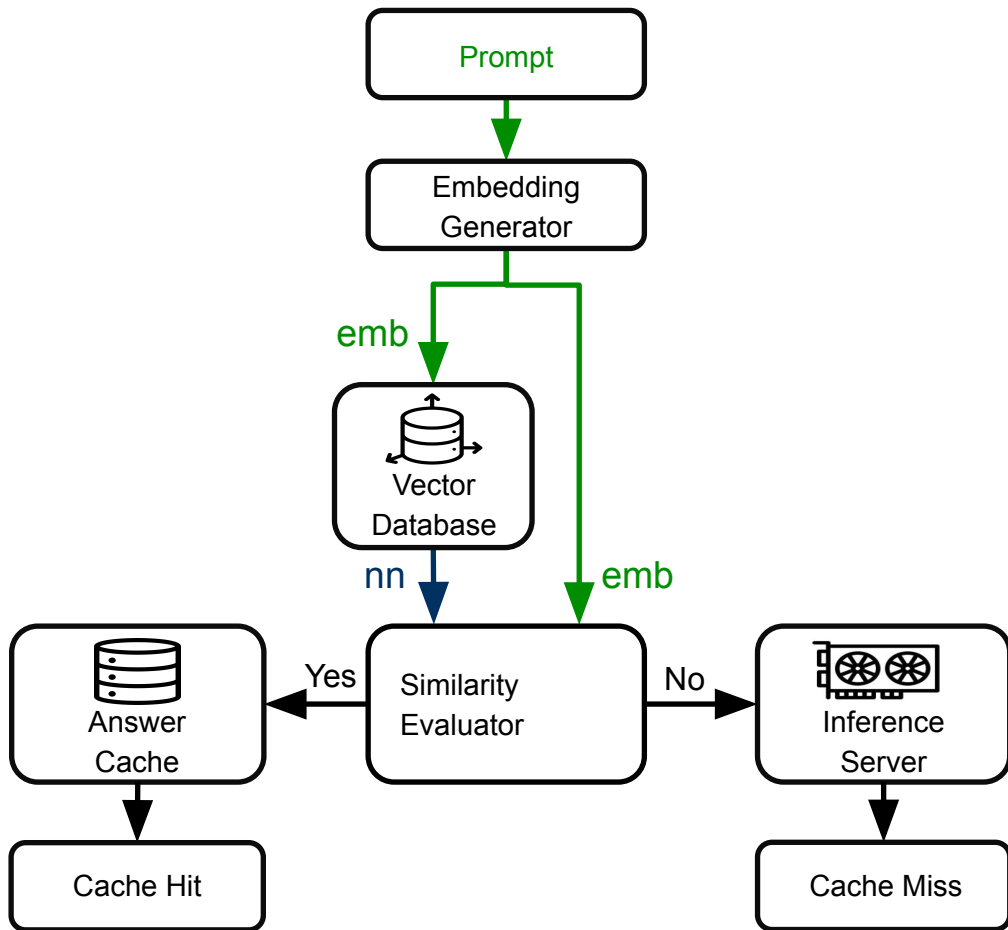
# Existing Semantic Caches are Unreliable and Not Deployable in Production Systems

Principal Researcher  
AI Team, International  
Car Sharing Platform

“

We wanted to use a semantic cache for our inference pipeline. We could not deploy it because the False Positive Rate was too high and unpredictable.

”



“Where was...?”

Emb(“Where was ...?”)

Retrieve the nearest neighbor from the vector database

Are the embedded prompt and its nearest neighbor similar enough?

Yes → Cache Hit

No → Cache Miss

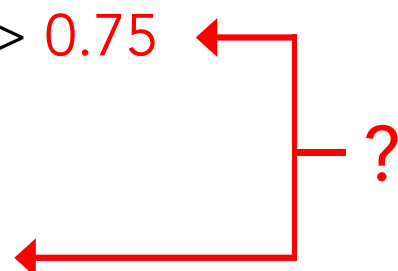
# The Threshold Problem

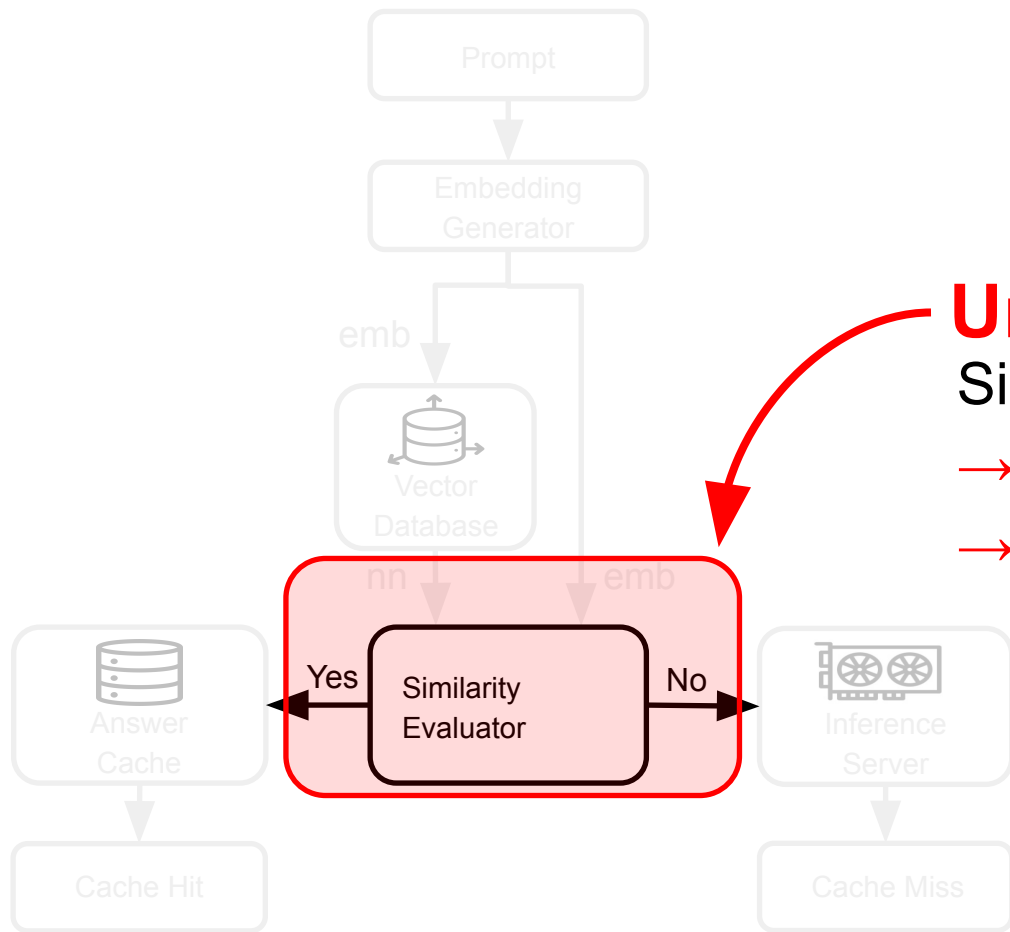
Given a new embedded **prompt** and its **nearest neighbor** embedding from the vector database...

$\text{cos\_sim}(\text{"Where was Roger Federer born"},$   
 $\text{"In which town did Roger Federer grow up"}) > 0.75$  ←

$\text{cos\_sim}(\text{"Explain Round Robin scheduling"},$   
 $\text{"Explain Shortest Job First scheduling"}) > 0.92$  ←

?





## Unreliability

Similarity  $\geq$  Some Threshold ?

→ Unexpected errors

→ Suboptimal cache hit rates

## GPTCache : A Library for Creating Semantic Cache for LLM Queries

Slash Your LLM API Costs by 10x 💰, Boost Speed by 100x ⚡

☆ 7.4k stars

👁 57 watching

🍴 521 forks

Report repository

Releases 41

📦 v0.1.44 Latest  
on Aug 1, 2024

+ 40 releases

Used by 6.1k

Learn / Azure / API Management /

### Enable semantic caching for **Azure** OpenAI APIs in Azure API Management

### Build a read-through semantic cache with **Amazon** OpenSearch Serverless and Amazon Bedrock

 databricks

Why Databricks Product Solutions Resources About

### Building a Cost-Optimized Chatbot with Semantic Caching

They all use a static threshold, are suboptimal, and unreliable...

Cache. How To Make Semantic Prompt Caches More **Reliable**?

User-Defined Accuracy Target (i.e., 99%)

 Cache Online Learning Algorithm

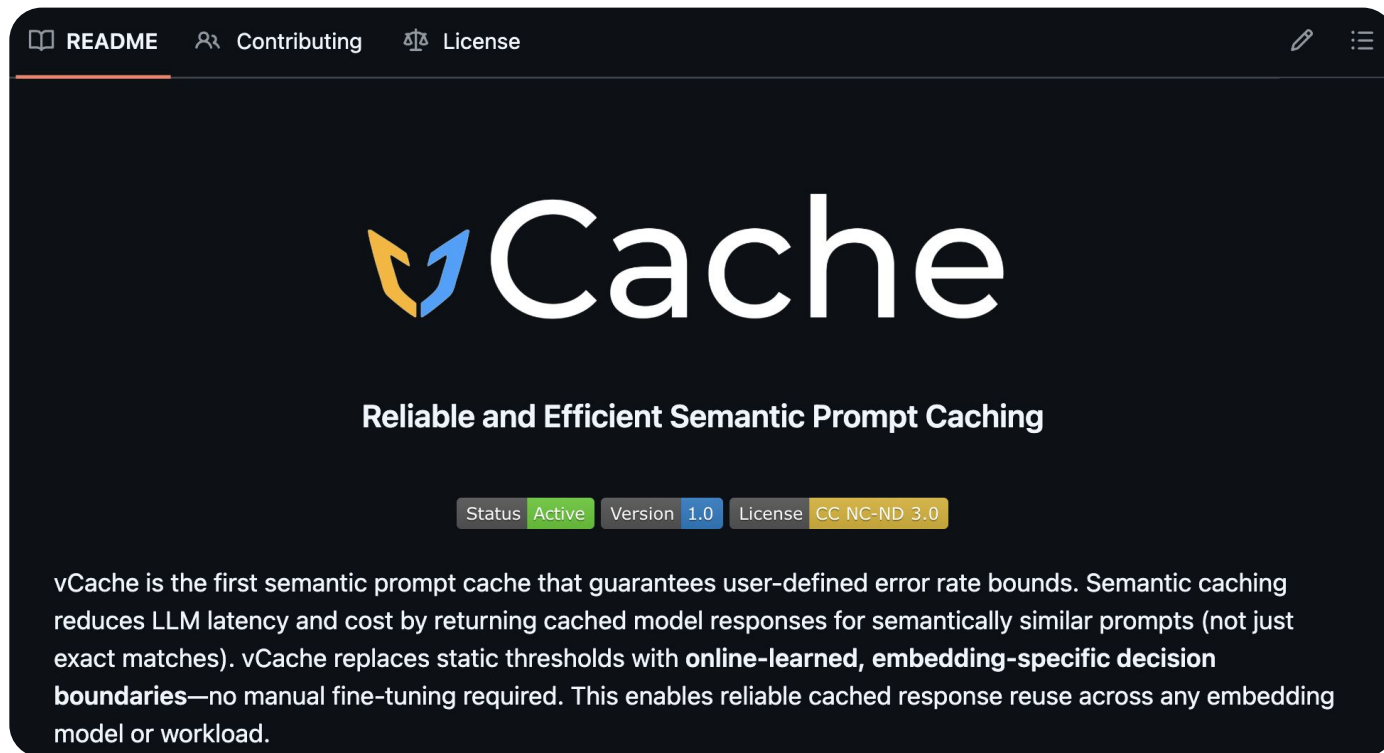
No Manual  
Tuning or  
Supervision

No Matter the  
Workload


No Matter the  
Model

No Matter the  
Infrastructure

[github.com/vcache-project/vCache](https://github.com/vcache-project/vCache)



The screenshot shows the GitHub README for the vCache project. At the top, there are navigation links for 'README', 'Contributing', and 'License'. The main heading is 'vCache' with a logo consisting of two overlapping arrows, one yellow and one blue. Below the heading is the tagline 'Reliable and Efficient Semantic Prompt Caching'. A status bar shows 'Status Active', 'Version 1.0', and 'License CC NC-ND 3.0'. The main text describes vCache as the first semantic prompt cache that guarantees user-defined error rate bounds, reducing LLM latency and cost by returning cached model responses for semantically similar prompts. It highlights that vCache replaces static thresholds with online-learned, embedding-specific decision boundaries, requiring no manual fine-tuning.

 **Cache** is the first reliable semantic cache reducing inference cost and latency with

1. Online learning (embedding-model and dataset agnostic)
  - a. Error rate guarantees
  - b. Embedding specific and probabilistic decision boundaries
2. SOTA performance