© FEB 2026 | IRE Journals | Volume 9 Issue 8 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I8-1714023

Praxis: A Hybrid Mobile AI Application with Offline LLM Inference and Cloud API Integration

SRI JAYA LINGESWARAN A[1], KEITHICK R[2], NIRANJAN R[3], ADRASH S[4]
[1,2,3,4]*Department of Computer Science Hindustan Institute of Technology & Science Chennai, India*

Abstract- ***Large Language Models (LLMs) have revolutionized natural language processing and artificial intelligence applications. However, deploying these models on mobile devices presents significant challenges due to limited computational resources, memory constraints, and energy consumption. This paper introduces Praxis, a novel hybrid mobile application architecture that seamlessly integrates on-device LLM inference with cloud-based API services. Praxis enables users to run small language models locally on Android devices for enhanced privacy and offline capability while providing the flexibility to switch to more powerful cloud-based models when connectivity and computational demands permit. Our implementation leverages React Native, Expo, and integration with Hugging Face model repositories, combined with secure API key management for services like OpenAI, Anthropic Claude, and Google Gemini. Experimental results demonstrate that Praxis achieves a balance between performance, privacy, energy efficiency, and user experience, making advanced AI capabilities accessible on resourceconstrained mobile devices.***

Index Terms- ***Large Language Models, Mobile Computing, On-Device Inference, Edge Computing, Hybrid Architecture, Privacy-Preserving AI, React Native***

I. INTRODUCTION

The proliferation of Large Language Models (LLMs) has transformed various domains including natural language understanding, code generation, and conversational AI [1]. While these models demonstrate remarkable capabilities, their deployment on mobile devices faces substantial challenges related to computational requirements, memory footprint, and energy consumption [2], [3].

Traditional approaches rely on cloud-based inference, where mobile applications send requests to remote servers hosting LLMs. Although this paradigm offers access to powerful models, it introduces several limitations including network latency, privacy concerns, continuous connectivity requirements, and recurring operational costs [5], [7].

Recent advances in model quantization, efficient inference techniques, and mobile hardware capabilities have enabled the deployment of Small Language Models (SLMs) directly on mobile devices [4], [12]. On-device inference provides benefits such as reduced latency, enhanced data privacy, offline functionality, and elimination of per-request API costs.

This paper presents Praxis, a hybrid mobile AI application that combines the advantages of both paradigms. Praxis allows users to:

- Run quantized LLMs locally on Android devices for privacy-sensitive and offline scenarios
- Seamlessly switch to cloud-based API services for access to more powerful models
- Download and manage multiple models from Hugging

Face repositories

- Configure and securely store API keys for multiple LLM providers
- Utilize voice-based interaction with speech-to-text and text-to-speech capabilities

II. RELATED WORK

A. Mobile Edge Intelligence for LLMs

Mobile Edge Intelligence (MEI) has emerged as a promising paradigm for deploying AI capabilities at the network edge [1]. Several studies have explored optimizing LLM inference on mobile devices through various techniques including model compression, efficient scheduling, and hardware-aware optimization [2].

B. On-Device LLM Inference Optimization

EdgeLLM [3] introduced speculative decoding techniques to accelerate on-device LLM inference for models exceeding device memory capacity. The framework delegates token generation to smaller draft models while maintaining quality through verification mechanisms.

Cambricon-LLM [4] proposed a chiplet-based hybrid architecture combining Neural Processing Units (NPU) with NAND flash storage to enable efficient inference of 70B parameter models on edge devices. This approach addresses memory bandwidth limitations through hardware-software co-design.

FACIL [6] introduced flexible DRAM address mapping for cooperative SoC-PIM (Processing-in-Memory) inference, addressing the challenge of efficient tensor placement for both GEMM and GEMV operations required by LLMs.

C. Hybrid Edge-Cloud Architectures

DeePar [12] explored layer-level partitioning strategies for distributing deep neural network computation across device, edge, and cloud tiers. This approach reduces overall execution delay while optimizing resource utilization.

EdgeShard [7] presented the first deployment of LLMs on collaborative edge computing environments, where edge devices and cloud servers share resources without accuracy loss. This system addresses resource limitations while maintaining inference quality.

D. Task Offloading and Resource Allocation

Several works have investigated intelligent offloading strategies for LLM inference tasks [5], [8]. These approaches employ reinforcement learning and multi-armed bandit algorithms to optimize offloading decisions based on network conditions, task requirements, and energy constraints.

E. Hybrid AI Architectures

Prior research has explored hybrid architectures combining edge and cloud computing for various AI applications [9]– [11]. Cloud-edge orchestration frameworks [13] have demonstrated the benefits of cooperative computing for IoT and mobile applications.

III. SYSTEM ARCHITECTURE

A. Overview

Praxis employs a three-tier hybrid architecture consisting of:

1) Mobile Device Layer: Hosts the React Native application, local model storage, and on-device inference engine
2) Edge/Local Processing Layer: Manages model loading, inference execution, and resource scheduling
3) Cloud API Layer: Provides access to powerful remote LLM services through secure API connections

B. Mobile Application Components

1) User Interface: The application features three primary screens:

- Home Screen: ChatGPT-inspired conversational interface with message history, model selector, and streaming response display
- Voice Chat Screen: Speech-to-text input with audio visualization and text-to-speech output
- Settings Screen: Model hub for downloading local models, API provider configuration with secure key management, and application preferences

2) Model Management System: The Model Hub integrates with Hugging Face repositories to:

- Browse available quantized models suitable for mobile deployment
- Download models with progress tracking and resume capability
- Manage local storage and model metadata
- Enable model selection and switching

3) API Provider Integration: Praxis supports multiple LLM API providers including:

- OpenAI (GPT-4, GPT-3.5-turbo)
- Anthropic (Claude 3.5 Sonnet, Claude 3 Opus)
- Google AI (Gemini Pro, Gemini Ultra)
- Hugging Face Inference API
- Custom API endpoints

API keys are securely stored using Expo SecureStore, providing encryption at rest and secure access control.

C. On-Device Inference Engine

The local inference engine employs:

- Model quantization (4-bit, 8-bit) to reduce memory footprint
- Efficient tokenization and attention mechanisms
- Batched inference for improved throughput
- Memory management to prevent device resource exhaustion

D. Hybrid Inference Strategy

Praxis implements an intelligent model selection and execution strategy:

1) User-Driven Selection: Users explicitly choose between local and cloud models
2) Automatic Fallback: If local inference fails or connectivity is lost, the system gracefully degrades to available alternatives
3) Context-Aware Routing: Future work will implement automatic routing based on query complexity, network conditions, and privacy requirements

IV. IMPLEMENTATION DETAILS

A. Technology Stack

- Frontend: React Native with TypeScript, Expo SDK
- State Management: Redux Toolkit for application state
- Local Storage: AsyncStorage for conversations, SecureStore for API keys
- ML Backend: Python with FastAPI for model serving (optional)
- Model Format: GGUF, ONNX for efficient mobile inference
- Cloud Infrastructure: Google Cloud Platform for backend services

B. Security Considerations

- API keys encrypted using hardware-backed keystore
- Local conversations encrypted at rest
- HTTPS-only communication with cloud services
- Certificate pinning for API connections
- Input sanitization and rate limiting

C. Model Quantization and Optimization

To enable efficient on-device inference, we employ:

- Post-training quantization to 4-bit and 8-bit precision
- Knowledge distillation from larger teacher models
- Pruning redundant parameters
- Operator fusion and graph optimization

V. EXPERIMENTAL EVALUATION

A. Experimental Setup

Test Devices:

- Android Device 1: 8GB RAM, Snapdragon 888 • Android Device 2: 6GB RAM, MediaTek Dimensity 1200 Test Models:
- Local: TinyLlama-1.1B (Q4), Phi-2-2.7B (Q8)
- API: GPT-3.5-turbo, Claude 3.5 Sonnet

B. Performance Metrics

We evaluate Praxis across multiple dimensions:

- Latency: Time to first token (TTFT) and tokens per second
- Energy Consumption: Battery drain during inference
- Memory Usage: Peak RAM utilization
- Response Quality: Evaluated using standard benchmarks

C. Results

Latency Analysis: Local inference achieved TTFT of 250400ms for small models, while cloud APIs exhibited 8001500ms due to network latency.

Energy Efficiency: On-device inference consumed 15-20% less energy compared to continuous network communication required for cloud APIs during extended usage sessions.

Quality Comparison: While local models provided acceptable performance for basic queries, cloud-based models demonstrated superior performance on complex reasoning tasks.

VI. DISCUSSION

A. Advantages of Hybrid Architecture

- Flexibility: Users adapt to different contexts and requirements

- Privacy: Sensitive queries processed locally
- Reliability: Offline functionality ensures continuous service
- Cost-Effectiveness: Reduces API usage costs for routine tasks

B. Limitations and Challenges

- Local models have limited capabilities compared to cloud alternatives
- Model downloads require significant storage and bandwidth
- Battery consumption during intensive local inference
- API key management complexity

C. Future Work

- Implement federated learning for privacy-preserving model updates
- Develop intelligent routing algorithms for automatic model selection
- Optimize inference through hardware acceleration (NPU, GPU)
- Expand model repository with domain-specific fine-tuned models
- Implement model compression techniques for improved efficiency

VII. CONCLUSION

This paper presented Praxis, a hybrid mobile AI application that bridges the gap between on-device and cloud-based LLM inference. By providing users with flexible model selection, secure API integration, and efficient local inference, Praxis demonstrates the viability of hybrid architectures for mobile AI applications. Our implementation showcases that combining local and cloud resources enables mobile applications to deliver advanced AI capabilities while respecting user privacy, minimizing latency, and optimizing resource utilization. As mobile hardware continues to advance and model optimization techniques improve, we anticipate that hybrid architectures like Praxis will become increasingly prevalent in the mobile AI ecosystem.

REFERENCES

[1] Author et al., "Mobile Edge Intelligence for Large Language Models: A Contemporary Survey," *IEEE Wireless Communications*, 2025. DOI: 10.1109/xxx.2025.10835069

[2] Author et al., "m-LLM: A Multi-Dimensional Optimization Framework for LLM Inference on Mobile Devices," *IEEE Transactions on Mobile Computing*, 2024. DOI: 10.1109/TMC.2024.11075620

[3] Author et al., "EdgeLLM: Fast On-Device LLM Inference With Speculative Decoding," *IEEE Transactions on Computers*, 2025. DOI: 10.1109/TC.2025.10812936

[4] Author et al., "Cambricon-LLM: A Chiplet-Based Hybrid Architecture for On-Device Inference of 70B LLM," *Proc. IEEE International Conference on Computer Design (ICCD)*, 2024. DOI: 10.1109/ICCD.2024.10764574

[5] Author et al., "Large Language Models (LLMs) Inference Offloading and Resource Allocation in Cloud-Edge Networks: An Active Inference Approach," *Proc. IEEE Global Communications Conference (GLOBECOM)*, 2023. DOI: 10.1109/GLOBECOM.2023.10333824

[6] Author et al., "FACIL: Flexible DRAM Address Mapping for SoCPIM Cooperative On-device LLM Inference," *Proc. IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 2025. DOI: 10.1109/HPCA.2025.10946824

[7] Author et al., "EdgeShard: Efficient LLM Inference via Collaborative Edge Computing," *IEEE Transactions on Mobile Computing*, 2025. DOI: 10.1109/TMC.2025.10818760

[8] Author et al., "Generative Inference of Large Language Models in Edge Computing: An Energy Efficient Approach," *Proc. IEEE International Conference on Web Services (ICWS)*, 2024. DOI: 10.1109/ICWS.2024.10592339

[9] Author et al., "Hybrid AI Architecture using Edge-Cloud Computing for Secure V2X Communication," *Proc. IEEE Vehicular Technology Conference (VTC)*, 2024. DOI: 10.1109/VTC.2024.10859430

[10] Author et al., "A hierarchical edge cloud architecture for mobile computing," *Proc. IEEE*

International Conference on Computer Communications (INFOCOM), 2016. DOI: 10.1109/INFOCOM.2016.7524340

[11] Author et al., "Edge-Cloud Computing and Artificial Intelligence in Internet of Medical Things: Architecture, Technology and Application," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6689-6706, 2020. DOI: 10.1109/JIOT.2020.2999359

[12] Author et al., "DeePar: A Hybrid Device-Edge-Cloud Execution Framework for Mobile Deep Learning Applications," *Proc. IEEE International Conference on Web Services (ICWS)*, 2019. DOI: 10.1109/ICWS.2019.8845240

[13] Author et al., "Cloud-Edge Orchestration for the Internet of Things: Architecture and AI-Powered Data Processing," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9847-9859, 2020. DOI: 10.1109/JIOT.2020.3009809