On-Device Parallel AI Triage System

A Practical Architecture for Medical Guidance Using Small Parallel Models

Author: Cho Hyunwoo
Date: 2025-12-13
Version: v1.0

Abstract

This paper proposes an **on-device medical triage and guidance system** that explicitly **does not perform diagnosis, treatment, or prescription**.

Instead of relying on a single large language model, the system operates using **multiple small, specialized AI models in parallel**, each responsible for a limited medical domain. A central manager model aggregates their outputs and produces **action-oriented guidance only**, such as:

- Whether immediate medical attention is required
- Whether a visit can be delayed
- Which department is appropriate to visit

This approach addresses practical constraints in real-world deployment, including **power consumption, latency, privacy, regulatory risk, and operational efficiency**, and is designed to be deployable in **medical buildings, outpatient clinics, kiosks, and mobile devices**.

---

1. Problem Statement

1.1 Why Medical AI Has Struggled

Most medical AI efforts are built on flawed assumptions:

- AI must provide diagnoses
- Accuracy must reach physician-level certainty
- Liability must be handled by the system provider

In practice, these assumptions create insurmountable barriers:

- Diagnosis carries legal and ethical responsibility
- Accuracy requirements dramatically increase cost and complexity
- Regulatory approval becomes the dominant obstacle

Meanwhile, the **actual bottleneck in healthcare occurs before diagnosis**.

1.2 The Real Bottleneck: Triage and Guidance

In everyday medical settings:

- Patients know **what hurts**, but not **where to go**
- Front-desk staff repeatedly answer identical routing questions
- Incorrect department selection is routinely corrected later by physicians

Hospitals already operate on the assumption that **initial routing can be imperfect**, because corrections are cheap and safe downstream.

---

2. Core Concept

2.1 Triage, Not Diagnosis

This system explicitly does **not**:

- Determine disease names
- Provide medical conclusions
- Suggest treatments or medications

Instead, it performs **triage and navigation**, equivalent to a human intern or front-desk staff member.

Its output is limited to **action guidance**, not medical claims.

---

3. System Architecture Overview

3.1 Parallel Small-Model Design

```
[ User Symptom Input ]
          ↓
[ 0.5–0.6B Specialist Models × N ]
 (domain-specific evaluation)
          ↓
[ 7B Manager Model ]
          ↓
[ Action-Oriented Guidance ]
```

3.2 Components

3.2.1 Specialist Models (0.5–0.6B)

Each specialist model is responsible for **one narrow domain** only.

Examples include:

- Respiratory
- Gastrointestinal

- Cardiovascular
- Dermatological
- Emergency signal detection

Each specialist performs only three tasks:

1. Determine whether the symptom matches its domain
2. Detect predefined **red flags**
3. Declare “not my domain” when appropriate

No specialist performs holistic reasoning.

3.2.2 Manager Model (≈7B)

The manager model:

- Does **not** perform medical reasoning
- Does **not** override specialist judgments
- Aggregates outputs into human-readable guidance

Its role is communicative and procedural, not diagnostic.

---

4. Decision Logic: Fail-Safe Design

4.1 Red-Flag Priority Rule

The system does **not** use majority voting.

If even one specialist detects a red flag, the system immediately recommends medical attention.

This mirrors conservative safety logic used in:

- Emergency medicine
- Aviation systems
- Industrial safety systems

Safety takes precedence over optimization.

4.2 Example Outcomes

Specialist Outputs	System Recommendation
≥1 Red Flag	Immediate medical visit
No Red Flags, multiple mild matches	Same-day or next-day visit
Mostly “not my domain”	Observation or general care

---

5. On-Device Implementation Strategy

5.1 Why Large Models Are Unnecessary

This system prioritizes **rule execution over reasoning**.

- Creativity is not required
- Open-ended inference increases risk
- Smaller models reduce hallucination potential

Specialists operate within constrained, auditable logic.

5.2 Cold Load & Modular Activation

- Minimal base model remains resident
- Specialist modules are loaded only when needed
- Modules are unloaded after execution

This design minimizes:

- Power consumption
- Thermal load
- Memory footprint

It is suitable for mobile NPUs and embedded hardware.

---

6. Deployment Scenarios

6.1 Medical Building Kiosks

- First-floor symptom input
- Department routing guidance
- Corrections handled downstream by staff

Responsibility remains with the clinic operator.

6.2 Outpatient Clinics

- Reduced front-desk workload
- Faster patient flow
- No interference with physician authority

6.3 Mobile Devices (On-Device AI)

- No internet dependency
- Strong privacy guarantees
- Pre-visit guidance without data transmission

---

7. Regulatory and Liability Positioning

7.1 Why This Is Not a Medical Device

Function	This System
Diagnosis	No
Treatment	No
Prescription	No
Administrative guidance	Yes

The system functions as an **operational support tool**, similar to kiosks or scheduling software.

7.2 Responsibility Model

- System provider: tool developer
- Final responsibility: healthcare operator

This mirrors existing hospital information systems.

---

8. Extensibility Beyond Healthcare

The same architecture applies to any domain where:

- Problems can be decomposed into expert checks
- Safety favors conservative escalation

Examples include:

- Legal guidance
- Administrative routing
- Customer service triage

---

9. Conclusion

This proposal reframes medical AI from:

> “Replacing doctors”

to:

> **“Helping people reach the right place faster.”**

By abandoning diagnostic ambition and embracing conservative triage, the system becomes:

- Deployable today
- Hardware-efficient
- Legally defensible
- Operationally valuable

The system does not need to be perfect.
Healthcare workflows already assume correction downstream.