实验报告（研究论文）

基于多模态感知与端侧智能的智能用药监测系统设计与实现

Design and Implementation of a Smart Medication Monitoring System

Based on Multimodal Sensing and On-Device Intelligence

报告完成人：周宇轩

完成日期：2026 年 8 月

项目版本：Git commit 3297d8d（codex/feature/on-device-ai-app）

摘 要

面向居家用药中提醒、取药、可见动作与记录缺少连续证据的问题，本研究实现 Smart Medication Monitor 多模态原型。系统由 ESP32-S3、HX711、OV2640 和 iOS 应用组成。重量端采用滚动基线及“物理门控 + 动态随机森林”分层分类器，通过 BLE 传输端侧推理结果；视觉端使用 OV2640 MJPEG、MediaPipe 关键点和 Core ML 识别 TAKE 等动作。Round 1 训练、Round 2 测试中，平坦随机森林准确率为 74%，分层方法为 82%；100 个真实事件的五折交叉验证为 100%，仅表明小样本内部可分性。ESP32 与 Python 在 150/150 个事件上一致，证明部署实现一致性而非准确率。单一参与者、单会话的 120 段相机视频内部交叉验证准确率为 96.67%。A/B 小实验中，BLE OPEN 的 MAE 为 0.0482 g，优于实体按钮的 0.2144 g。重量减少只表示药物被移出，TAKE 只表示类似服药的可见动作，均不能证明吞咽。截至 2026 年 8 月 21 日，iPhone 实机链路已验证通过；由于尚缺少结构化真机性能日志，本报告不进一步报告真实 iPhone 运行条件下的 BLE/Wi-Fi 连接时延、MJPEG 帧率、功耗及长期稳定性等量化指标。

关键词：智能用药监测；多模态感知；端侧人工智能；ESP32-S3；行为识别

目 录

摘 要.....2

1 引言.....5

1.1 研究背景与意义.....5

1.2 国内外研究现状.....5

1.3 研究内容与主要贡献.....6

2 相关工作.....7

2.1 基于容器事件与重量的监测.....7

2.2 基于可穿戴与视觉的动作识别.....7

2.3 多模态融合与端侧隐私.....8

3 方法与系统实现.....9

3.1 任务定义与输出边界.....9

3.2 硬件与通信架构.....10

3.3 滚动基线与事件采样.....11

3.4 分层重量分类模型.....11

3.5 端侧模型导出.....12

3.6 视觉动作识别.....12

3.7 iOS 应用与提醒调度.....13

4 实验设计与结果.....18

4.1 数据集与采集方案.....18

4.2 评价指标与基线.....19

4.3 重量分类结果.....19

4.4 触发方式 A/B 实验.....22

4.5 相机分类结果.....23

4.6 ESP32 部署与软件验证.....25
4.7 泛化、案例与错误分析.....26
5 讨论.....26
5.1 分层建模的工程价值.....26
5.2 多模态结果的语义边界.....26
5.3 端侧计算、隐私与可用性.....27
5.4 通知与人机交互.....27
6 结论、局限性与未来工作.....27
6.1 结论.....27
6.2 局限性.....28
6.3 未来工作.....28
7 参考文献.....29
8 附录.....31

1 引言

1.1 研究背景与意义

长期用药依从性是由患者、治疗方案、疾病、医疗系统与社会经济环境等多因素共同影响的问题。WHO 的经典报告指出，慢性病长期治疗依从性在部分发达国家人群中平均约为 50%，但该历史性总体统计并不代表任何具体患者或本项目人群[1]。提醒可以降低遗忘概率，却无法回答患者是否打开药盒、是否取出正确剂量、随后是否出现类似服药动作。

电子药盒能够提供比回忆式自报更细粒度的时间事件，但“开启容器”等同于“服药”的假设并不总成立，装置本身也可能影响被观察者行为[2]。同理，称重模块适合观测容器库存和重量变化[6]，但重量下降可能对应取出、转移、掉落或误操作。因此，本研究不以“确认吞咽”为目标，而将问题限定为：在工程原型条件下，能否把重量变化、类似服药动作、提醒响应和事件记录组织成一条更完整、可解释且尽量端侧处理的证据链。

1.2 国内外研究现状

现有研究大致分为三类。第一类以电子容器、压力或重量传感器记录药盒开启、容器移动或药片数量变化[2],[6]，优点是结构简单、事件时间明确，局限是缺少动作上下文。第二类使用腕部惯性传感器或摄像头识别与服药相关的手部动作[3],[4]，能够补充行为信息，但受佩戴依从性、遮挡、视角、光照和个体动作差异影响。第三类将 RFID、视频等多种传感器结合，使“哪个对象被移动”和“随后出现什么动作”相互补充[5]。这些路线共同说明，多模态证据能够减少单一传感器的歧义，但代理信号的叠加仍不等同于生理吞咽真值。

在实现平台方面，ESP32-S3 同时提供双核处理、无线连接、相机接口和外部存储接口，适合整合采集、通信和小型模型推理[7]。iOS 的 Core Bluetooth 可实现中心设备

扫描、连接、特征写入与通知订阅[8]；UserNotifications 支持本地定时提醒和按标识符取消待处理通知[9]；Core ML 与 MediaPipe 可在移动端执行模型推理和关键点提取[13]–[15]。本研究将这些能力组合为一套从传感器到 App 的端到端原型。

1.3 研究内容与主要贡献

本研究的主要工作包括：

- 完成 ESP32-S3R8、HX711 与 OV2640 的硬件集成。实体按钮先因 A/B 实验暴露的称重扰动被 BLE OPEN 替代；后续加入相机后，GPIO4 被 OV2640 D0 占用，进一步支持当前无实体按钮架构。
- 设计滚动基线和两阶段重量分类：对一片/两片采用可解释的重量比门控，对净变化接近零的 NONE、RETURN 和 DISTURBANCE 使用 24 维动态特征随机森林。
- 将 500 棵树的随机森林导出为 C++ 常量结构，在 ESP32-S3 上完成本地推理，并建立 Python/C++ 150 事件回归一致性验证。
- 实现 OV2640 MJPEG 到 iPhone 的图像链路，以及 MediaPipe 关键点—2576 维特征—Core ML 动作分类流程。
- 实现 SwiftUI 双语 App，包括处方时间、每日及追提醒、BLE/相机连接、历史、患者 ID 和本机医生视图，并明确医生跨设备功能仍需安全后端。
- 建立真实数据、跨轮测试、内部交叉验证、合成压力测试和端侧实现一致性测试的分层证据体系，并系统梳理适用边界。

2 相关工作

2.1 基于容器事件与重量的监测

电子事件监测常以药瓶或药盒开启时间作为代理指标。Sutton 等的随机对照研究表明，电子监测可以获得精细事件信息，但测量反应性与“开启即服药”的假设必须被单独考虑[2]。Hayes 等设计的 MedTracker 在传统七日药盒上记录持续用药事件，并于 39 个家庭开展了现场试验，说明容器级长期采集具有可行性[23]。Checchi 等纳入 37 项研究的系统综述则指出，电子药品包装的研究质量和效果估计差异较大，记录给药事件并不自动等同于确认实际摄入[24]。

Chen 等从压力/重量测量角度证明了药片数量变化具有可观测性[6]。重量传感相较单纯开盖信号，可以进一步估计取出数量，但容器转移、药片掉落、手部施力和结构回弹都会改变 load path。本研究因此不估计绝对库存，而是围绕一次约 5 s 的开放事件，同时分析前后净重量变化与事件内动态轨迹。

2.2 基于可穿戴与视觉的动作识别

可穿戴惯性传感器能够识别拿药、手部向口部移动等动作模式[3]。Wang 等使用双腕三轴加速度计和动态时间归整区分服药与饮水/擦嘴动作，25 名参与者的结果同时展示了惯性动作证据的潜力与误报问题[25]。Odhiambo 等进一步用智能手表收集脚本化和自然服药手势，其 28 人验证研究强调了自然动作、相似日常活动以及个体差异对识别器设计的影响[27]。

摄像头方法可以利用手、脸、嘴部和容器的空间关系，提供比单一惯性轨迹更丰富的上下文，但也更受遮挡、视角、光照和隐私约束影响[4]。Lee 与 Youm 将摄像头置于可穿戴设备，并在 89 名志愿者的数据上研究服药行为识别，说明连续视觉中的对象与动作信息可以联合使用[26]。本研究不要求用户佩戴额外设备，而使用药盒上的

OV2640 采集低分辨率 MJPEG，将关键点提取和动作分类放在 iPhone 端。TAKE 仅指 ingestion-like visible action，即可见的类似服药动作。

2.3 多模态融合与端侧隐私

Hasanuzzaman 等将 RFID 对象识别与视频行为分析结合，展示了异质信号的互补价值[5]。Kalantarian 等则将颈部可穿戴吞咽信号与现有药瓶取药系统并置讨论，直接指出“药物从容器移出”和“后续摄入行为”需要不同证据[28]。这些工作的共同启示是：多模态不是简单增加一个分数，而是要保留每一模态的可观测边界和不确定状态。

本项目采用可解释的重量 + 视觉逻辑融合：只有重量移出数量与处方期望剂量一致，且相机返回 TAKE，App 才显示“多项信号支持本次服药行为”。相机帧不落盘、重量模型在 MCU 上运行，与数据最小化和可行时端侧处理的设计原则一致[10]。但本地推理并不自动满足医疗数据法规；网络配置、身份认证、密钥、授权与审计仍属产品级系统责任。

表 1 现有用药监测方法对比

方法	可观测信息	优点	主要局限
Reminder App	用户对通知的响应	成本低、易部署	响应 ≠ 实际取药[24]
Door/Open Sensor	药盒或药瓶开启事件	结构简单、时间明确	开盒 ≠ 取药[2],[23]
Weight Sensing	药物质量变化	可估计取出数量	无法观测后续行为[6]
Wearable IMU	手腕时序动作	提供动作过程	需佩戴，存在相似动作误报[25],[27]
Camera	可见人体与物体动作	上下文丰富	隐私、遮挡与跨用户泛化[4],[26]
Multimodal	多源代理证据	可减少单传感器歧义	系统与解释复杂度更高[5],[28]
本研究	Weight + Vision + On-Device	规则可解释、原始画面不落盘	当前数据规模和泛化证据有限

3 方法与系统实现

3.1 任务定义与输出边界

一次监测事件从 App 发出 OPEN 开始。重量侧时序输入定义为：

$$X_w = \{(t_i, w_i)\}_{i=1}^{N}$$

其中 t_i 为采样时间，w_i 为称重读数。系统记录打开前滚动基线、约 5 s 开放阶段的轨迹以及关闭后的稳定重量。重量分类输出为：

$$y_w \in Y_w$$

其中重量输出集合为 {NONE, ONE, TWO, RETURN, DISTURBANCE, UNCERTAIN}。视觉序列由 iPhone 抽取约 24 帧，其输出定义为：

$$y_v \in Y_v$$

视觉输出集合为 {ADJUST, DRINK, NONE, PICK_ONLY, TAKE, TOUCH_FACE}。最终结果是重量证据、视觉证据与处方期望剂量 p 的可解释组合：

$$y_{final} = F(y_w, y_v, p)$$

该融合不是临床或生理吞咽判定。最终状态只描述“药物移出”“类似服药动作”及两者是否与处方相互支持。

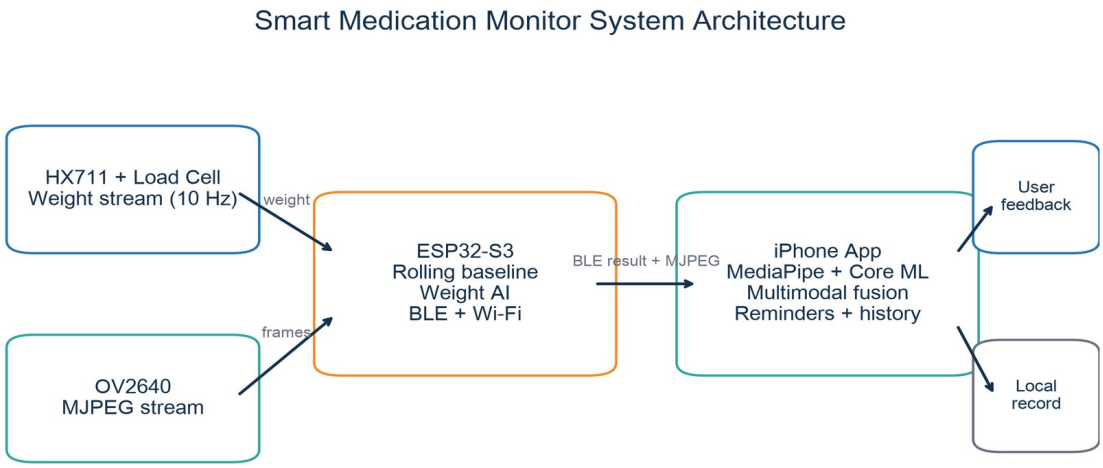

图 1 Smart Medication Monitor 总体架构（根据仓库实现绘制）

3.2 硬件与通信架构

表 2 原型硬件与关键配置

模块	配置/连接	作用
主控	ESP32-S3R8；16 MB Flash；8 MB OPI PSRAM	采集、BLE、Wi-Fi、MJPEG 与重量端侧推理
称重	HX711 DT=GPIO9，SCK=GPIO10；校准因子 653.0	获取药盒重量轨迹
相机	OV2640，经板载插座连接	输出 MJPEG 图像流
药片原型	单片质量约 0.848 g	重量比门控的实验基准
事件触发	iOS App 发送 BLE OPEN	避免按压改变 load path；后续也适配相机占用 GPIO4
开发设置	ESP32S3 Dev Module；16 MB；OPI PSRAM；3 MB APP/9.9 MB FATFS	保证固件容量与 PSRAM 可用

BLE 采用 Nordic UART 类服务：App 写入 RX 特征，ESP32 通过 TX 特征通知。LIVE 模式下，OPEN 触发事件；完成采样后固件发送 AI|event_id|prediction|final_delta[|confidence]。原始 DATA 行只保留在 USB 串口，避免 BLE 带宽压力。相机工作于设

备 Wi-Fi AP，App 从 http://192.168.4.1/stream 读取 MJPEG。当前开放 AP 和单客户端流仅适合原型调试，不满足产品安全要求。

硬件演化顺序是：早期原型已具有实体按钮；随后 5+5 次 A/B 工程实验发现，用户按压设备会改变称重结构的受力路径，BLE OPEN 的 MAE 明显更低，因此从 sensing quality 和 interaction design 角度先决定改用 BLE。后续接入 OV2640 后，GPIO4 被 camera D0 占用，这进一步使移除实体按钮符合当前硬件架构。

3.3 滚动基线与事件采样

系统闭合状态约每 100 ms 采样一次。滚动基线缓冲区包含最近 30 点（约 3 s），计算时排除最新 3 点（约 300 ms），可用点少于 10 时不允许触发。这样可降低用户刚触碰药盒时对事件前基线的污染。OPEN 阶段持续 5000 ms，之后设置 1500 ms 稳定期；端侧最多保存 64 个事件样本。设事件前基线重量为 m_{before}，事件后稳定重量为 m_{after}，则净移出量定义为：

$$\Delta m = m_{before} - m_{after}$$

3.4 分层重量分类模型

设当前药片的平均单片质量为 m_{pill}，分层模型首先计算无量纲药片质量比：

$$r = \frac{\Delta m}{m_{pill}}$$

在本实验使用的模拟药片中，平均单片质量约为 0.848 g，因此当前实验配置采用 $m_{pill} \approx 0.848\ g$。若 |r|<0.30，则净变化接近零，进入动态随机森林；若 0.50≤r<1.50，直接判为 ONE；若 1.50≤r<2.50，判为 TWO；其余区域输出 UNCERTAIN。阈值之间保留间隙，避免测量落在边界附近时被强制归类。

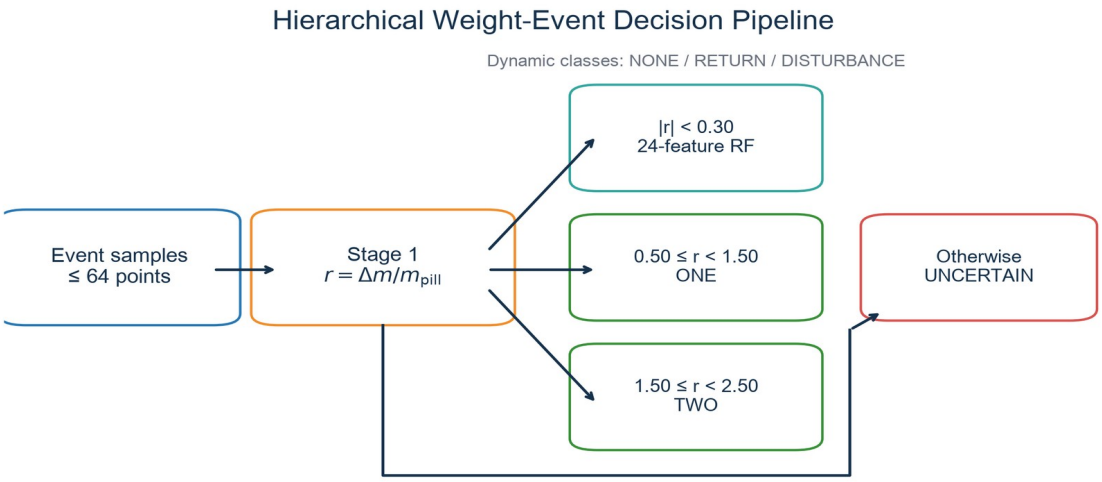

图 2 分层重量事件判定流程

动态随机森林不使用 final_delta_g，避免重复利用第一阶段的净重量信息。模型输入严格按配置文件排列为 24 维：轨迹范围与标准差、相对基线最大上升/下降、最大绝对偏差、极值时间、持续时间、相邻采样最大上升/下降、平均绝对步长、步长标准差、三档变化次数、低于基线 0.4 g 的比例/最长持续点/事件数、起始均值差与标准差、末端标准差、绝对/有符号面积以及采样数。

分类器为 500 棵树的随机森林（random_state=42，class_weight=balanced，max_features=sqrt），算法与实现依据见文献[11]、[12]。

3.5 端侧模型导出

训练后的森林被展开为定长树节点数组。每个节点存储左右子节点索引、特征索引、阈值及叶节点类别分数；ESP32 逐树遍历并对类别投票。导出报告记录 500 棵树、3174 个节点、24 个特征。固件最多接收 64 个事件样本，特征计算与 Python 版本保持同一顺序和阈值。该设计消除了日常运行时对电脑端 Python 脚本的依赖。

3.6 视觉动作识别

OV2640 负责采集，ESP32 只提供 MJPEG 流，不在 MCU 上进行视觉模型推理。iPhone 对每帧执行 MediaPipe Hand Landmarker 与 Pose Landmarker，形成手部与姿态关

键点、置信度及运动统计，固定展开为 2576 维特征。离线 Extra Trees 模型含 600 个估计器，随后导出为 CameraActionClassifier.mlmodel 并由 Core ML 加载。原型阈值为最高类别概率不低于 40%，稳定投票数为 1；这适合演示，但缺少迟滞与跨帧一致性保护未来应提高稳定票数并引入不确定状态。

3.7 iOS 应用与提醒调度

App 使用 SwiftUI 组织展示层，由 CoreBluetooth 管理药盒命令与重量结果，NetworkExtension 管理相机 Wi-Fi，MediaPipe Tasks Vision 与 Core ML 在本机完成视觉推理。一次用药流程串联“提醒响应—BLE OPEN—重量结果—视觉结果—融合展示—本地历史”。处方保存药名、期望剂量和每日时间；提醒调度先为每个启用处方保留一条每日通知，再在系统通知数量边界内轮询分配最多 8 次、间隔 15 min 的追提醒。用户响应后取消同一处方未投递的追提醒；这一响应不被解释为已取药或已服药。

实际 Swift 融合逻辑只有一条积极规则：重量预测的移出数量与处方 expected dose 一致，且 vision action 为 TAKE 时，记录被标记为“多项信号支持本次服药行为”。其他组合保留原始模态结果并给出谨慎提示，不额外推断吞咽。

表 3 多模态结果展示逻辑（根据 Swift 实现核对）

Weight evidence	Vision evidence	App interpretation
移出数量 = expected dose	TAKE	多项信号支持本次服药行为；不代表吞咽确认
移出数量 = expected dose	Non-TAKE / 无结果	已检测到期望取药，但视觉动作未支持
移出数量 ≠ expected dose	Any	保留实际移出数量与相机结果，不显示多信号支持
RETURN / NONE / DISTURBANCE	Any	按重量预测展示放回、未取药或干扰，视觉仅作为独立证据
UNCERTAIN / 模态冲突	—	保留不确定性，引导用户人工确认

App 支持英文/简体中文切换、历史记录和本地 Doctor View，并生成 MBX-XXXX-XXXX 形式的患者 ID。当前 LocalPatientDataProvider 只能查询同一 iPhone 上的数据；跨设备医生访问所需的后端、身份认证、患者授权、加密、审计与撤销尚未实现。相机帧在当前 App 中用于内存内推理，不作为历史图像保存。

图 3a iOS 模拟器中的中文首页与 Mock Transport 状态

图 3b iOS App 多模态事件结果界面示例（开发模式 Mock 事件）

图 4 本机 Doctor View 及其隐私边界提示

4 实验设计与结果

4.1 数据集与采集方案

表 4 实验数据组成

数据集	规模	类别构成	定位
重量 Round 1	50 个真实事件	5 类，各 10	初始训练与内部五折
重量 Round 2	50 个真实事件	5 类，各 10	跨采集轮次独立测试
Synthetic Round 3	50 个模板衍生事件	5 类，各 10	冻结流程压力测试；非独立验证
相机动作	120 段视频	6 类，各 20	单一参与者、单次采集条件下内部交叉验证

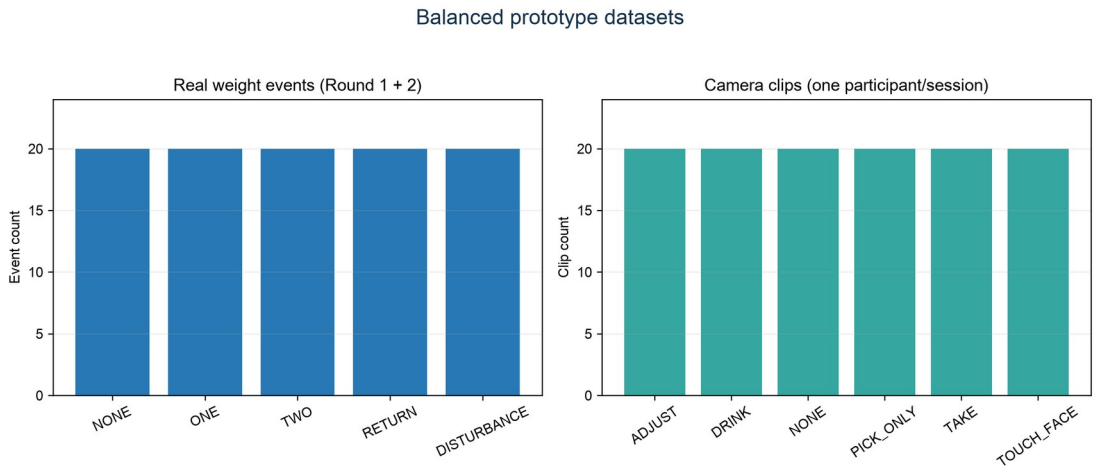

图 5 平衡的重量与相机原型数据集

重量标签包括 NONE、ONE、TWO、RETURN 和 DISTURBANCE。每个事件约 50–51 个采样点，Round 1 和 Round 2 各 10 个/类。合成 Round 3 由真实模板派生，用于验证输入扰动下代码流程与冻结模型能否一致运行，不能称为新环境中的独立泛化。相机六类为 ADJUST、DRINK、NONE、PICK_ONLY、TAKE 与 TOUCH_FACE，每类 20 段；数据来自同一参与者和同一采集会话，因此类别均衡不代表人群多样性。

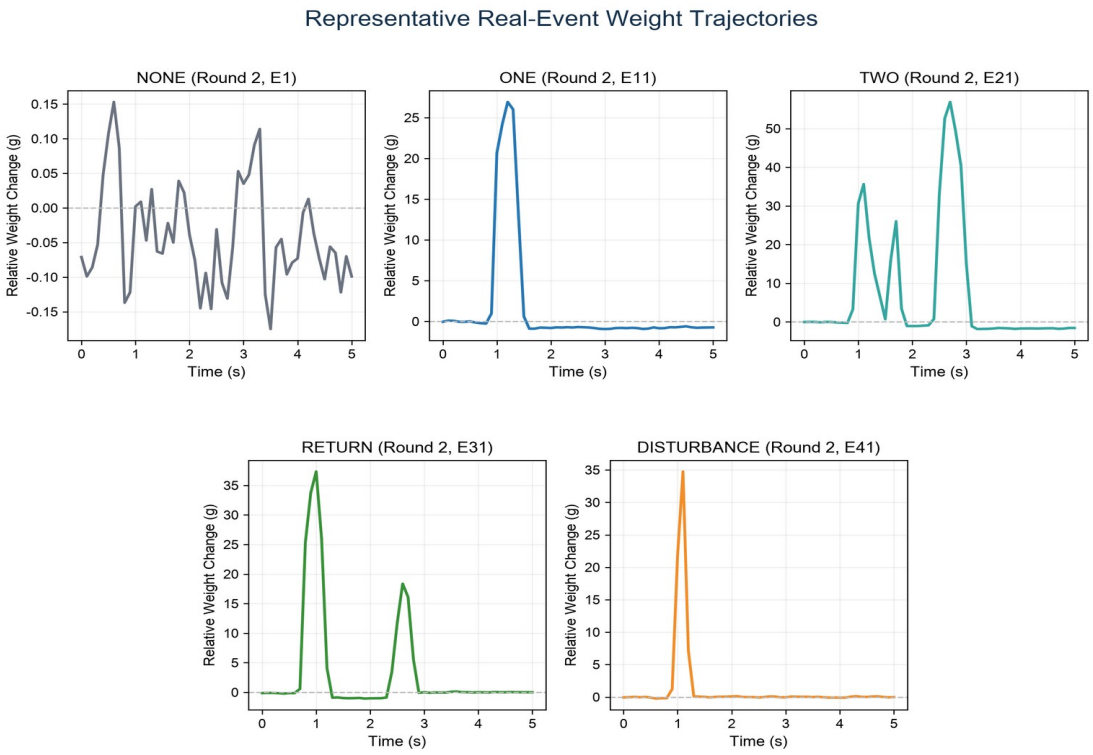

图 6 Round 2 五类事件的代表性相对重量轨迹

4.2 评价指标与基线

分类实验以 Accuracy 为主，并结合混淆矩阵和分类报告分析类别错误。跨轮测试以 Round 1 训练、Round 2 测试，测试集不参与训练；对照为使用 final_delta_g 和动态特征的平坦五分类随机森林。混合 100 事件五折交叉验证使用 StratifiedKFold(n_splits=5, shuffle=True, random_state=42)，只用于观察内部可分性。端侧部署采用逐事件预测一致率，比较冻结 Python 管线与 C++ 运行时标签；这一指标不衡量真值准确率。触发方式 A/B 用重量误差的平均绝对误差 MAE。

4.3 重量分类结果

表 5 重量分类主要结果及解释边界

实验	结果	正确解释
Round 1 平坦 RF 五折 CV	96%	首轮小样本内部交叉验证
平坦 RF：Round 1→Round 2	74%	跨采集轮次独立测试

实验	结果	正确解释
分层模型：Round 1→Round 2	82%	跨采集轮次独立测试
分层模型：100 真实事件五折 CV	100%	两轮混合后的内部可分性
冻结分层模型：Synthetic Round 3	50/50（100%）	模板衍生压力/流程测试

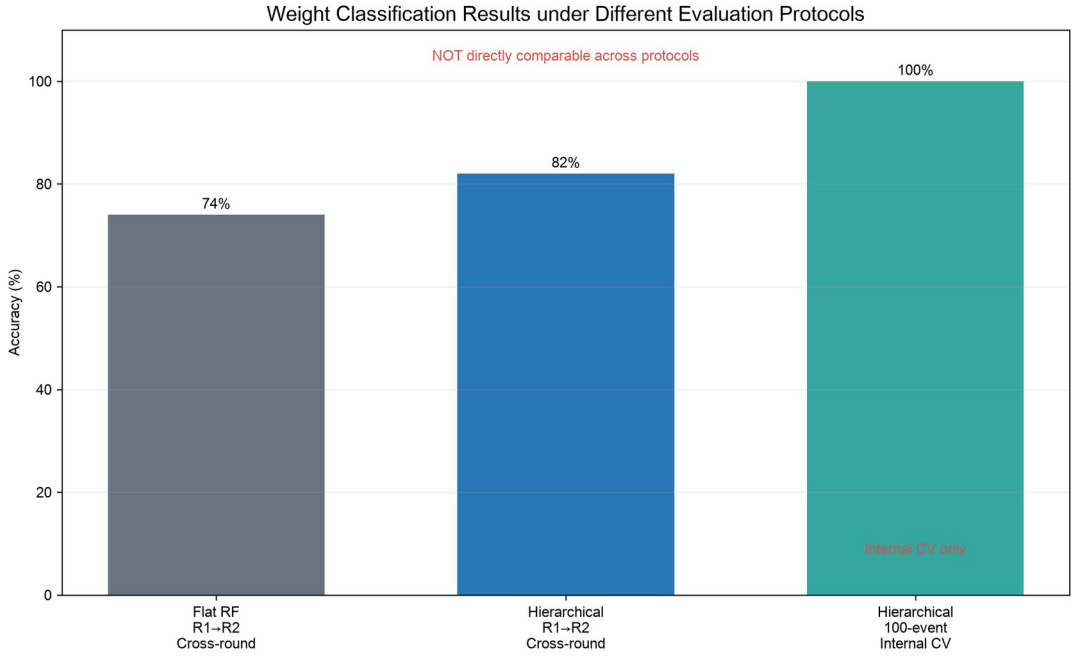

图 7 不同评估协议下的重量分类结果（内部 CV 与跨轮测试不可直接比较）

图 7 中 74% 和 82% 共享 Round 1→Round 2 跨轮测试协议，可用于比较平坦与分层方法；100% 则来自两轮样本混合后的内部五折交叉验证。由于测试协议不同，柱高不能当作同一 independent test set 上的性能递增。有效比较是跨轮结果：分层模型比平坦模型提高 8 个百分点。分层结构把单片和双片的稳定物理区间交给规则处理，使动态模型集中区分净变化接近零的三类行为。Round 2 的 9 个错误均来自 DISTURBANCE 事件 41–49，主要被判为 RETURN 或 NONE。

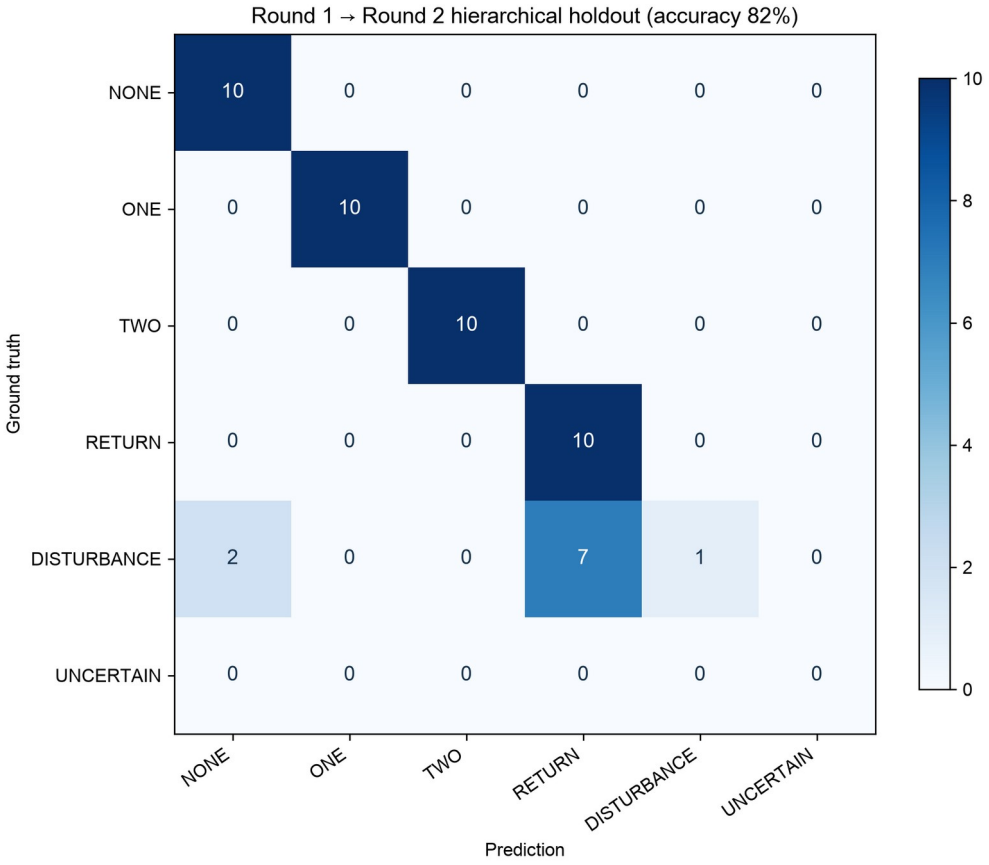

图 8 分层模型 Round 1 训练、Round 2 测试混淆矩阵

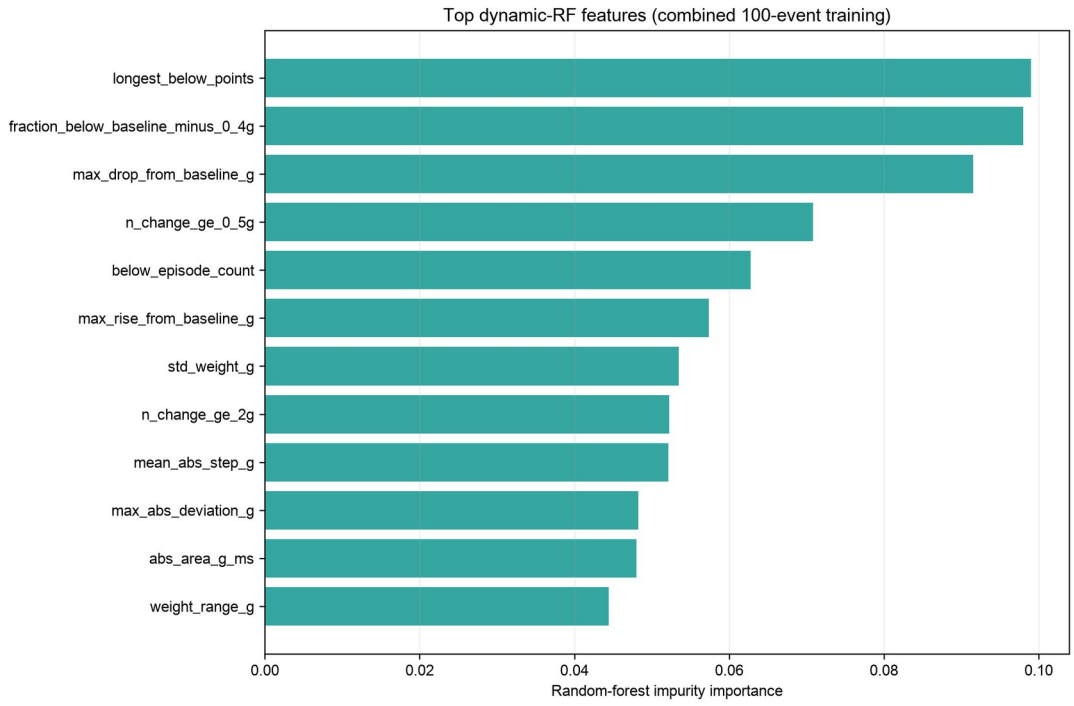

图 9 动态随机森林前 12 个特征的重要度

特征重要度最高的指标包括 longest_below_points、fraction_below_baseline_minus_0_4g 和 max_drop_from_baseline_g，表明低于基线的持续长度与下探幅度是模型区分 NONE、RETURN 和 DISTURBANCE 的重要线索。不过，随机森林不纯度重要度会受变量尺度和相关性影响，不能直接解释因果关系。多个动态特征之间也存在相关性，因此该重要度更适合作为模型行为的解释线索，而不是独立物理贡献量。

4.4 触发方式 A/B 实验

第一阶段原型已具有实体按钮。为判断交互方式是否改变称重输入，对 Physical Button 与 BLE OPEN 各进行 5 次重复。实体按钮误差为 −0.378、−0.221、+0.002、−0.319、−0.152 g，MAE=0.2144 g；BLE OPEN 误差为 +0.112、+0.067、−0.002、−0.057、−0.003 g，MAE=0.0482 g。实体按钮 MAE 约为 BLE 的 4.45 倍，支持“直接按压设备会改变 load path 并产生称重扰动”的工程解释。因此项目先从 sensing quality 和 interaction design 出发改用 BLE OPEN；后续接入 OV2640 后，GPIO4 被 camera D0 占用，才进一步支持移除实体按钮的当前架构。该结论仅来自 5+5 次小样本工程实验。

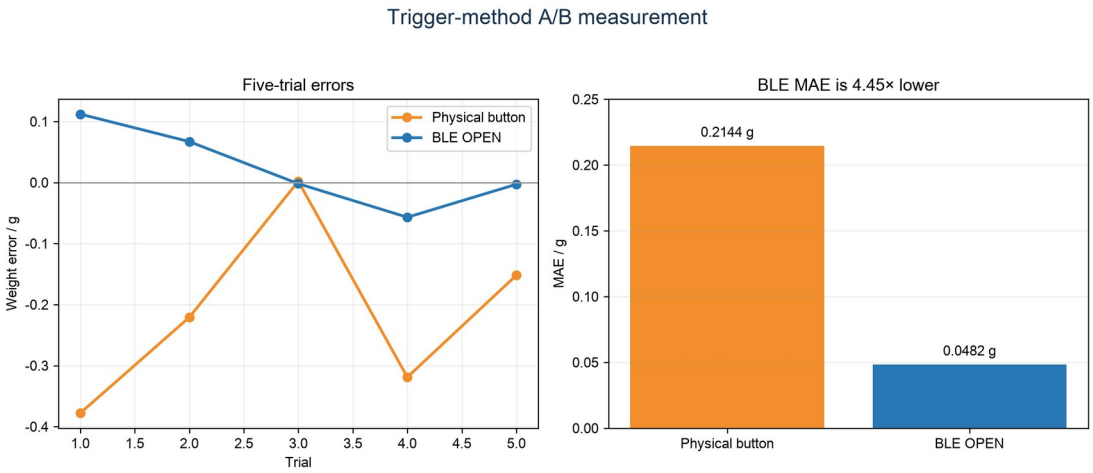

图 10 实体按钮与 BLE OPEN 的五次重量误差及 MAE

4.5 相机分类结果

相机训练报告记录 120 段视频、2576 维特征、600 个 Extra Trees 估计器和 96.67% 内部五折交叉验证准确率。真实 CV 预测的 4 个错误全部发生在 TOUCH_FACE 与 ADJUST 之间；TAKE、DRINK、NONE 和 PICK_ONLY 在该内部划分上均为 20/20。

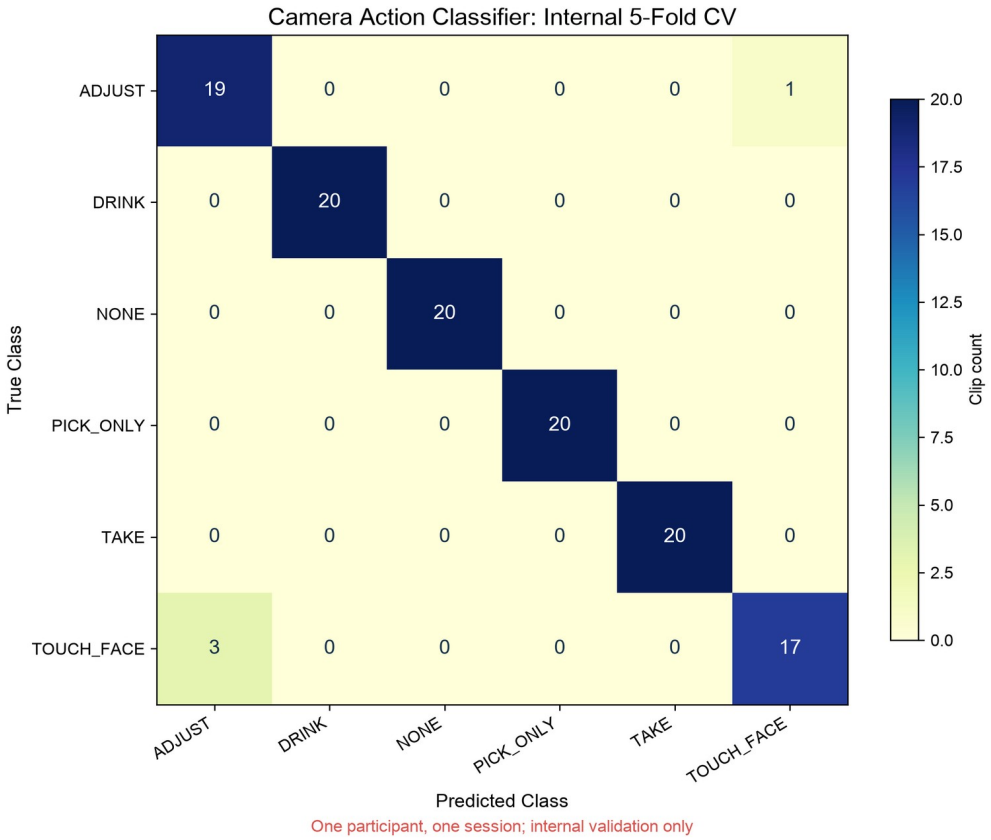

图 11 相机动作分类内部五折交叉验证混淆矩阵

表 6 相机动作分类的逐类内部验证指标

Class	Precision	Recall	F1	Support
ADJUST	0.8636	0.9500	0.9048	20
DRINK	1.0000	1.0000	1.0000	20
NONE	1.0000	1.0000	1.0000	20
PICK_ONLY	1.0000	1.0000	1.0000	20
TAKE	1.0000	1.0000	1.0000	20
TOUCH_FACE	0.9444	0.8500	0.8947	20
Macro avg	0.9680	0.9667	0.9666	120

Core ML 导出后抽取 24 个验证样本，与 Python 模型相比类别不一致数为 0，最大概率误差约 9.55×10^{-15}。这一结果是导出数值一致性，不是新的准确率证据。图 11 和表 6 均来自单一参与者、单一采集 session 下的内部验证，只能说明当前原型数据分布下

的类别可分性，不能代表跨用户泛化性能。数据尚未覆盖年龄、肤色、衣着、光照、摄像头位置和遮挡等变化。

4.6 ESP32 部署与软件验证

表 7 端侧部署与测试结果

项目	结果	含义
导出森林	500 棵树；3174 节点；24 特征	端侧模型结构
Python/C++ 全事件一致性	150/150	三轮数据的实现一致性
动态分支一致性	90/90	NONE/RETURN/ DISTURBANCE 路由与森林一致
固件 Flash	1,373,435 B；约 43%（3 MB APP）	可刷入 3 MB 应用分区
全局变量	63,056 B；约 19%	剩余约 264,624 B 动态内存
Python 回归测试	14/14 通过	数据、特征、协议、模型导出与 C++ 对照
iOS 26.5 模拟器 workspace	BUILD SUCCEEDED	Xcode 26.6；App 与 CocoaPods/MediaPipe 可构建

在 Arduino 默认 1.31 MB 应用分区下，固件曾出现 104% 超限；切换为 3 MB APP/9.9 MB FATFS 后编译占用约 43%，说明问题来自分区配置而非必须删减核心功能。实机串口已确认固件可启动。Python 回归测试在项目隔离环境中复跑 14 项全部通过。更新后的 Xcode 26.6、iOS 26.5 模拟器 workspace 构建成功；图 3b 由该环境中的真实 App ResultView 生成，数据为开发模式 Mock 事件。

截至 2026 年 8 月 21 日，用户已确认 iPhone 实机链路验证通过。由于本次确认未附带可复现的结构化性能日志，本报告不进一步报告真实 iPhone 运行条件下的 BLE/Wi-Fi 连接时延、MJPEG 帧率、功耗及长期稳定性等量化指标。

4.7 泛化、案例与错误分析

本研究包含三个层次的“泛化”证据：Round 1→Round 2 是跨采集轮次测试，最接近独立评估；100 事件五折是样本混合后的内部可分性；Synthetic Round 3 是模板派生压力测试。三者不能混为一谈。对典型正确案例，ONE/TWO 的净重量比落入物理门控区间，结果稳定且可解释；对错误案例，扰动可能产生“下降—恢复”的形状，与 RETURN 相似。未来需要增加放置偏移、拍击、移动桌面、手扶药盒等扰动类型，并按采集日期、参与者和设备划分真正独立测试集。

5 讨论

5.1 分层建模的工程价值

分层方案没有要求单一模型同时学习明显不同的物理机制。单片、双片移出由药片质量比直接判断，净变化接近零的复杂动作才交给动态随机森林。与平坦 RF 的跨轮结果相比，8 个百分点的提高支持这一设计思路，但样本量只有 100 个真实事件，尚不足以证明其对不同药品、容器和用户普遍有效。未来可把每种药品的单片质量、容差和处方剂量作为可配置参数，并通过更大规模的按用户留出实验验证。

5.2 多模态结果的语义边界

重量模块回答“药盒中的质量是否按期望减少”，视觉模块回答“是否出现训练分布内的类似服药动作”。二者同时满足能够提高事件可信度，但仍可能出现把药片移出后做出相似动作、遮挡后未能识别、一次取出多次服用等情况。因此，多模态结果应作为支持性行为证据，而不是医学结论。系统界面、医生记录和未来云端字段都应保留原始模态结果、不确定状态和时间戳，避免把复杂事件压缩为没有证据来源的二值“已服药”。

5.3 端侧计算、隐私与可用性

重量模型运行在 ESP32-S3，相机帧在 iPhone 本地处理且不保存，减少了对电脑和云服务的依赖，也降低了原始视觉数据外传风险[10]。但当前 Wi-Fi AP 是开放的，MJPEG 只支持一个客户端；患者 ID 只是本地随机标识；医生页没有账号与授权系统。若作为真实产品使用，必须引入 Wi-Fi 配网与加密、设备身份与密钥管理、受保护的 API、最小权限、患者授权、访问审计、数据删除与备份策略。

5.4 通知与人机交互

每日提醒与公平轮询追提醒解决了“多个处方时单一处方耗尽通知槽”的工程问题。双语切换和医生页提升了原型完整度。更重要的 HCI 启示来自实体按钮/BLE A/B 实验：在人机交互型传感系统中，交互方式本身可能改变传感器的物理输入。对 Load Cell 系统而言，UI/trigger design 不只是界面问题，也属于 measurement system design 的一部分当前证据仅为 5+5 次小样本工程实验，后续应在不同操作者、桌面和结构支撑方式下复现。

仍需注意：系统通知可能因用户权限、专注模式或设备状态而延迟；点击通知仅表示响应。产品验证应分别记录“通知计划生成、系统投递、用户打开、设备 OPEN、重量结果、视觉结果”六类事件，避免用一个状态替代全部过程。

6 结论、局限性与未来工作

6.1 结论

本研究完成了一套可运行的智能用药监测工程原型：ESP32-S3 同时连接称重与相机，分层重量模型能够在端侧输出事件类别；iPhone 完成 BLE 控制、MJPEG 获取、动作分类、处方提醒、历史与医生本机查询。跨轮重量测试显示分层模型由 74% 提升至 82%；端侧森林在 150 个回归事件上与 Python 管线一致；相机模型在单参与者内部交

叉验证中为 96.67%；模拟器 workspace 构建成功，且截至 2026 年 8 月 21 日用户已确认 iPhone 实机验证通过。上述证据证明了系统链路与部署可行性，但不证明临床依从性改善，也不证明吞咽；实机通过状态同样不能替代尚未记录的时延、功耗、稳定性和长期运行指标。

6.2 局限性

- 重量实验仅使用约 0.848 g 的模拟药片，真实事件仅 100 个，采集轮次和机械环境有限。
- 相机数据来自单一参与者和单次采集条件，没有跨用户、跨光照、跨视角和遮挡独立测试。
- TAKE 是类似服药的可见动作；重量减少是药物移出的证据；二者均不能证明生理吞咽。
- 100 事件五折 CV、50 个合成事件和 150/150 部署一致性分别属于内部可分性、流程压力测试和实现一致性，不能表述为真实世界准确率。
- OV2640 MJPEG 当前仅支持单客户端，开放 AP 缺少产品级网络安全。
- 医生视图只能查询本机数据，没有云端同步、身份认证、患者授权与审计。
- 当前没有 Hall 传感器或执行器，“开/关盖”为定时模拟状态；无法检测真实盖体位置或自动开盖。
- iPhone 实机验证已由用户确认通过，但本报告未取得结构化测试日志，因而仅记录通过状态，不给出连接时延、功耗、帧率、稳定性或长期运行指标。

6.3 未来工作

未来工作按优先级安排如下：第一，把已通过的 iPhone 实机链路沉淀为可重复测试脚本和结构化日志，量化 BLE/Wi-Fi 连接成功率、端到端时延、帧率、功耗与长时

间稳定性；第二，采集跨参与者、跨日期、跨光照/视角的相机数据以及跨药品、跨容器、跨摆放位置的重量数据，建立按参与者和会话隔离的独立测试集；第三，优化扰动类别和时序稳定策略，引入更严格的不确定性阈值；第四，加入 Hall 传感器或执行器，使盖体状态由真实硬件测量；第五，设计安全配网、设备身份和加密通信；第六，在获得伦理与隐私审查后，构建带认证、授权和审计的医生远程平台；最后，在医学专业人员指导下开展可用性研究和非临床验证，再讨论更高等级的临床研究。

7 参考文献

[1] E. Sabaté, Ed., Adherence to Long-Term Therapies: Evidence for Action. Geneva, Switzerland: World Health Organization, 2003. [Online]. Available: https://iris.who.int/handle/10665/42682

[2] S. Sutton et al., “Does electronic monitoring influence adherence to medication? Randomized controlled trial of measurement reactivity,” Ann. Behav. Med., vol. 48, no. 3, pp. 293–299, 2014. doi: https://doi.org/10.1007/s12160-014-9595-x

[3] H. Kalantarian, N. Alshurafa, and M. Sarrafzadeh, “Detection of gestures associated with medication adherence using smartwatch-based inertial sensors,” IEEE Sensors Journal, vol. 16, no. 4, pp. 1054–1061, 2016. doi: https://doi.org/10.1109/JSEN.2015.2497279

[4] G.-A. Bilodeau and S. Ammouri, “Monitoring of medication intake using a camera system,” Journal of Medical Systems, vol. 35, no. 3, pp. 377–389, 2011. doi: https://doi.org/10.1007/s10916-009-9374-6

[5] F. M. Hasanuzzaman et al., “Monitoring activity of taking medicine by incorporating RFID and video analysis,” Network Modeling Analysis in Health Informatics and Bioinformatics, vol. 2, no. 2, pp. 61–70, 2013. doi: https://doi.org/10.1007/s13721-013-0025-y

[6] Y. Chen, X. Li, W. Jia, and M. Sun, “Development of an automatic measurement system for medical pills based on a PDMS capacitive sensor,” Measurement, vol. 192, 110899, 2022. doi: https://doi.org/10.1016/j.measurement.2022.110899

[7] Espressif Systems, ESP32-S3 Series Datasheet, ver. 2.2, Mar. 5, 2026. [Online]. [Accessed: Aug. 21, 2026]. Available: https://documentation.espressif.com/esp32-s3_datasheet_en.pdf

[8] Apple Inc., “Performing Common Central Role Tasks,” Core Bluetooth Programming Guide. [Online]. [Accessed: Aug. 21, 2026]. Available:

https://developer.apple.com/library/archive/documentation/NetworkingInternetWeb/Conceptual/CoreBluetooth_concepts/PerformingCommonCentralRoleTasks/PerformingCommonCentralRoleTasks.html

[9] Apple Inc., “Scheduling a notification locally from your app,” UserNotifications Documentation. [Online]. [Accessed: Aug. 21, 2026]. Available: https://developer.apple.com/documentation/usernotifications/scheduling-a-notification-locally-from-your-app

[10] Apple Inc., “Privacy,” Human Interface Guidelines. [Online]. [Accessed: Aug. 21, 2026]. Available: https://developer.apple.com/design/human-interface-guidelines/privacy/

[11] L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001. doi: https://doi.org/10.1023/A:1010933404324

[12] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” Journal of Machine Learning Research, vol. 12, no. 85, pp. 2825–2830, 2011. [Online]. Available: https://www.jmlr.org/papers/v12/pedregosa11a.html

[13] C. Lugaresi et al., “MediaPipe: A framework for building perception pipelines,” arXiv preprint arXiv:1906.08172, 2019. [Online]. Available: https://arxiv.org/abs/1906.08172

[14] V. Bazarevsky et al., “BlazePose: On-device real-time body pose tracking,” arXiv preprint arXiv:2006.10204, 2020. [Online]. Available: https://arxiv.org/abs/2006.10204

[15] F. Zhang et al., “MediaPipe Hands: On-device real-time hand tracking,” arXiv preprint arXiv:2006.10214, 2020. [Online]. Available: https://arxiv.org/abs/2006.10214

[16] Apple Inc., “Core ML,” Apple Developer Documentation. [Online]. [Accessed: Aug. 21, 2026]. Available: https://developer.apple.com/documentation/coreml

[17] B. Necula et al., “HX711 Arduino Library,” official project repository. [Online]. [Accessed: Aug. 21, 2026]. Available: https://github.com/bogde/HX711

[18] Google AI Edge, “Hand landmarks detection guide for iOS,” MediaPipe Solutions documentation. [Online]. [Accessed: Aug. 21, 2026]. Available: https://developers.google.com/edge/mediapipe/solutions/vision/hand_landmarker/ios

[19] Bluetooth SIG, Bluetooth Core Specification, ver. 6.2, Nov. 3, 2025. [Online]. [Accessed: Aug. 21, 2026]. Available: https://www.bluetooth.com/wp-content/uploads/Files/Specification/HTML/Core-62/out/en/index-en.html

[20] Espressif Systems, “esp32-camera: ESP32 camera driver,” official repository. [Online]. [Accessed: Aug. 21, 2026]. Available: https://github.com/espressif/esp32-camera

[21] Apple Inc., “NEHotspotConfiguration,” NetworkExtension Documentation. [Online]. [Accessed: Aug. 21, 2026]. Available: https://developer.apple.com/documentation/networkextension/nehotspotconfiguration

[22] Apple Inc., “SwiftUI,” Apple Developer Documentation. [Online]. [Accessed: Aug. 21, 2026]. Available: https://developer.apple.com/documentation/swiftui

[23] T. L. Hayes, J. M. Hunt, A. Adami, and J. A. Kaye, “An electronic pillbox for continuous monitoring of medication adherence,” in Proc. 28th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBS), 2006, pp. 6400–6403. doi: https://doi.org/10.1109/IEMBS.2006.260367

[24] K. D. Checchi, K. F. Huybrechts, J. Avorn, and A. S. Kesselheim, “Electronic medication packaging devices and medication adherence: A systematic review,” JAMA, vol. 312, no. 12, pp. 1237–1247, 2014. doi: https://doi.org/10.1001/jama.2014.10059

[25] R. Wang et al., “Automatic identification of solid-phase medication intake using wireless wearable accelerometers,” in Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC), 2014, pp. 4168–4171. doi: https://doi.org/10.1109/EMBC.2014.6944542

[26] H. Lee and S. Youm, “Development of a wearable camera and AI algorithm for medication behavior recognition,” Sensors, vol. 21, no. 11, art. 3594, 2021. doi: https://doi.org/10.3390/s21113594

[27] C. O. Odhiambo et al., “Detecting medication-taking gestures using machine learning and accelerometer data collected via smartwatch technology: Instrument validation study,” JMIR Human Factors, vol. 10, e42714, 2023. doi: https://doi.org/10.2196/42714

[28] H. Kalantarian, B. Motamed, N. Alshurafa, and M. Sarrafzadeh, “A wearable sensor system for medication adherence prediction,” Artificial Intelligence in Medicine, vol. 69, pp. 43–52, 2016. doi: https://doi.org/10.1016/j.artmed.2016.03.004

8 附录

附录 A 24 维动态特征顺序

1 weight_range_g ; 2 std_weight_g ; 3 max_rise_from_baseline_g ; 4 max_drop_from_baseline_g ; 5 max_abs_deviation_g ; 6 time_to_max_ms ; 7 time_to_min_ms ; 8 duration_ms ; 9 max_step_up_g ; 10 max_step_down_g ; 11 mean_abs_step_g ; 12 std_step_g ; 13 n_change_ge_0_5g ; 14 n_change_ge_2g ; 15 n_change_ge_5g ; 16 fraction_below_baseline_minus_0_4g ; 17 longest_below_points ; 18

below_episode_count；19 start_mean_delta_g；20 start_std_g；21 end_std_g；22 abs_area_g_ms；23 signed_area_g_ms；24 n_samples。

附录 B BLE 关键报文

方向	报文	说明
App→ESP32	OPEN	在基线就绪时开始一次事件
App→ESP32	MODE|LIVE / MODE|TRAIN	切换运行或标注采集模式
ESP32→App	O|event_id	事件开始通知
ESP32→App	AI|event_id|prediction|final_delta[|confidence]	端侧重量分类结果
视觉→融合	VISION|event_id|action|confidence	iPhone 本地动作分类结果

附录 C 复现实验与文件索引

重量原始数据位于 data/round1、data/round2 和 data/synthetic_round3；特征、混淆矩阵与模型比较位于 results/round2 和 results/synthetic_round3；训练脚本为 experiments/hierarchical_model.py；模型配置为 model/hierarchical_model_config.json；端侧导出与验证位于 inference/weight；固件位于 firmware/esp32/MedBox_LiveTrain_AI；相机训练、导出报告和协议位于 inference/camera；iOS 项目位于 app/MedBoxApp；自动测试位于 tests。报告图表由 report/scripts/generate_figures.py 从上述文件生成，详细映射见 report/report_sources.md。

附录 D 当前验收状态

验收证据包括：Python 14 项回归测试通过；Xcode 26.6、iOS 26.5 模拟器 workspace 构建成功；固件在 3 MB 应用分区下完成编译与串口启动确认；截至 2026 年 8 月 21 日，iPhone 实机链路验证通过。由于缺少结构化真机性能日志，本报告不量化 BLE/Wi-Fi 连接时延、MJPEG 帧率、Core ML 推理时延、功耗与长期稳定性。