

第一问：(物理量单位字体待更改)

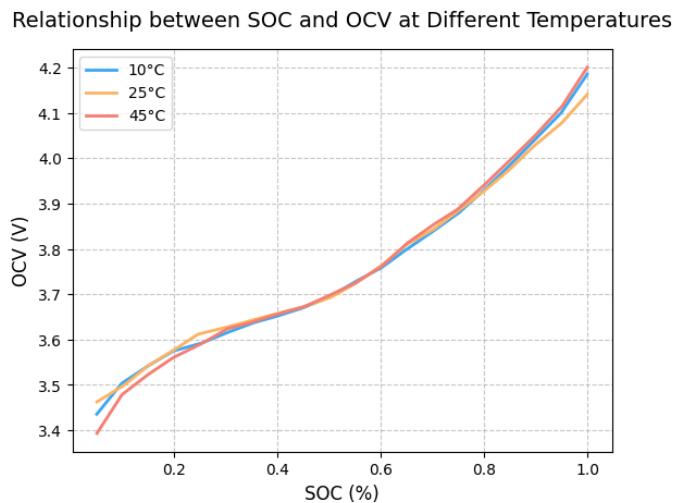
涉及时间：0~3hours, 分辨率 100*1

(1)：OCV 与 SOC 关系，从图像中发现温度对两者关系的影响很低。最大波动处变化 1.7%，使用三种温度的 OCV 平均值作为最终使用数据。

$$V_{oc}(soc) = a + b \cdot (-\ln soc)^m + c \cdot soc + d \cdot e^{(soc-1)}$$

使用非线性最小二乘法 (LM 算法) 拟合, $R^2 = 0.9994$, 获得参数

$$a = -34.3659, b = -0.1797, c = -8.9895, d = 47.5266, m = 2.6446 \times 10^{-3}, n = 0.2148$$

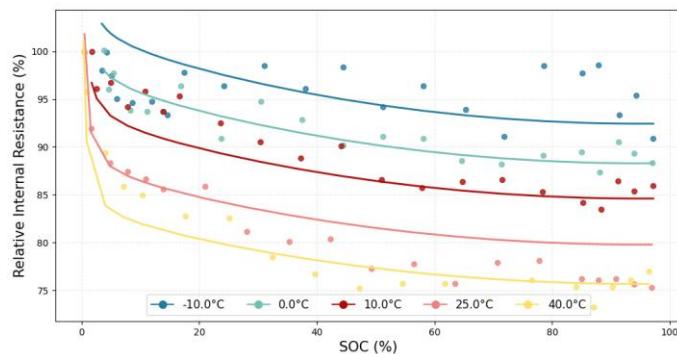


(2)：SOC 与内阻 R 的关系

$$R_{int}(soc, T) = R_0 \cdot m \cdot e^{\frac{E_a}{R_g T_k}} \cdot (a \cdot soc^2 + b \cdot soc + c)$$

使用非线性最小二乘法 (LM 算法) 拟合, $R^2 = 0.8815$, 获得参数:

$$m = 1.09, E_a = 2740.79, a = 0.000256, b = -0.0484, c = 26.60, d = 2.24$$



(3): 老化容量在单次放电中无变化, 文献里多以月为单位, 后续单独添加, 目前算作常数

(4): 负载功率:

A: CPU: $P(t) = f \cdot V(t)^2 \cdot u(t)$, $u(t)$ 为 CPU 使用率、 f 为 CPU 频率、 $V(t)$ 为 CPU 电压

假设参数 (基于统计信息):

CPU 核数 6, iphone17pm 的两个性能核单核最高 4.24GHz, 平均剩下四个能效核 2.1GHz, 取单核 3.75GHz, 待机取 0.4GHz; CPU 电压涉及商业机密, 数据无法获取, 且调控十分复杂, 进行简化, 常驻电压 0.7V, 最大电压 1.2V (基于 ai 对 iphone17pm 的 3 纳米芯片的推测); CPU 常驻使用率为 0.05, app (代码中用 game 代替) 最高使用率 0.7, app 运行越多使用率越高, 屏幕最高使用率 0.15, 屏幕越亮使用率越高, wifi 最高使用率 0.15, gps 使用率 0.1

B: GPU: $P(t) = f \cdot r \cdot \rho \cdot u(t)$, f 为屏幕刷新率、 r 为屏幕分辨率、 ρ 为 oled 发光密度 (光度, 单位 nit)

假设参数 (基于统计信息):

分辨率: (2622, 1206)、ppi: 460nit、屏幕刷新率 120Hz 均为 iphone17pm 数据

游戏最高占用 0.8, 屏幕最高占用 0.1, 无常住占用

C: 内存 RAM: $P(t) = P \cdot u(t)$, P 近似为常数 300mW

D: 屏幕功耗:

$$P_{OLED} = P_{pixels} + P_{driver-OLED}$$

$$P_{pixels} = L \times A_{emit} \times \sum(k_R \times R + k_G \times G + k_B \times B) \times \frac{1}{\eta_{OLED}}$$

$$P_{driver-OLED} = k_{res} \times k_{refresh-LTPO} \times P_{base-driver-OLED}$$

P_{OLED}	OLED 工作状态总功率 (mW)	80~400 mW (随内容 / 亮度变)	80~400 mW (随内容 / 亮度变化)
P_{pixels}	像素发光功率 (mW, 核心部分)	70~380 mW (占总功耗 90%+)	70~380 mW (占总功耗 90%+)
L	屏幕目标亮度 (nits, 尼特)	100~1000 nits (室内 ~ 外户)	100~1000 nits (室内 ~ 外户)
A_{emit}	发光像素总面积 (m^2)	手机屏幕约 0.01~0.015 m^2 (如 6.1 英寸: 0.011 m^2)	手机屏幕约 0.01~0.015 m^2 (如 6.1 英寸: 0.011 m^2)
$k_R/k_G/k_B$	红 / 绿 / 蓝子像素功耗系数 ($W \cdot m^{-2} \cdot nits^{-1}$)	$k_G = 0.002$; $k_R = 0.003$; $k_B = 0.006$ (绿色最节能, 蓝色最耗电)	$k_G = 0.002$; $k_R = 0.003$; $k_B = 0.006$ (绿色最节能, 蓝色最耗电)
$R/G/B$	显示内容的 RGB 灰度值 (0~255)	纯黑→0; 纯白→255; 灰色→128	纯黑→0; 纯白→255; 灰色→128
η_{OLED}	OLED 发光效率 (含材料损耗、电压损耗)	0.6~0.85 (LTPO/OLED 可达 0.8)	0.6~0.85 (LTPO/OLED 可达 0.8)
$P_{driver-OLED}$	OLED 驱动电路功率 (mW)	10~40 mW	10~40 mW
k_{res}	分辨率系数 (相对 1080P 的功耗比例)	1080P→1.0; 2K→1.1~1.2 (影响)	1080P→1.0; 2K→1.1~1.2 (影响比 LCD 小)
$k_{refresh-LTPO}$	LTPO 动态刷新率系数 (相对 60Hz)	60Hz→1.0; 120Hz→1.4~1.6;	60Hz→1.0; 120Hz→1.4~1.6; 1Hz→0.3~0.4
$P_{base-driver-OLED}$	基准驱动功率 (1080P+60Hz 下, mW)	10~15 mW	10~15 mW

E: wifi 与 gps 功耗:

```
def wifi_power(self, use_time: float, start_time: float, cpu_cost: float, mod: str):
    # 分段函数
    # 关断、接受、发送、监听
    # 没有数据时，假设使用WiFi期间内，休眠、接受、发送、监听，循环
    wifi_temp = np.sin(self.time_squ) * 500 # mW
    if mod == 'wifi':
        return wifi_temp * (start_time + use_time > self.time_squ > start_time)
    elif mod == '5g':
        return wifi_temp * (start_time + use_time > self.time_squ > start_time) * 4
    elif mod == '4g':
        return wifi_temp * (start_time + use_time > self.time_squ > start_time) * 2

def gps_power(self, use_time: float, start_time: float, cpu_cost: float):
    # 同理，循环 休眠、捕获、追踪
    gps_temp = np.sin(self.time_squ) * 30 # mW
    return gps_temp * (start_time + use_time > self.time_squ > start_time)
```

做了超级简化。

Wifi、GPS、游戏均可以设置不运行。不运行使用率就是 0

$$P_{all} = \left(\sum k_i \cdot P_i \right) \cdot \frac{1}{1 - \alpha[T - T_0]}$$

6 个 k_i 和 α 共 7 维参数记为 p 。使用神经网络预测, $\psi_\theta(P_i) = p$

该神经网络架构借鉴 PINN 的思想, 不直接预测总的功耗, 而是将物理方程嵌入神经网络, 使用神经网络预测部分参数。如此有效地将物理信息嵌入到神经网络中, 削弱了神经网络的黑盒性, 增强了解释性。

最终模型:

1. 微分方程: $\frac{ds}{dt} = \frac{a \cdot p(t)}{V(s) - I(t) \cdot R(s)}$

2. 代数方程: $I(t) = \frac{p(t)}{V(s) - I(t) \cdot R(s)}$

3. 变量域: $t \in [0, 3]$ 、 $s \in [0, 1]$, $R(s)$ 、 $V(s)$ 可通过 $s(t)$ 转化为 t 的函数。