

# QA 1

现在的推理脚本能够做到输入60个时间点的数据，预测是个时间点的数据。那么可否实现如下功能：  
假设我们已经训练好了模型，并且让模型放在Pi5中运行，现在，树莓派仍然10s采集一个数据，写入csv文件，同时，我们另开一个python程序，执行推理，即根据刚刚采集的前60个时间点的数据，预测下一个时间点的数据，然后等到下一个时间来临测量到真实数据时，我们对比这两组数据（可以用图片的形式对比，8个通道的电压对应8个子图，写为一个GUI程序）。也就是说实现一个GUI程序，每次读入一个新的点，就：

(1) 把这个新的时间点的数据显示到图中，与之前预测的对比

(2) 根据这个新的点，预测下一个点。

也就是让树莓派同时执行两组程序，一组测量程序，一组推理+GUI程序

可以，而且完全可行，用 **树莓派 5 + Python** 可以轻松实现你描述的 **实时推理 + 实时绘图 GUI 架构**。

我下面会给你一个**完整的架构方案**，包括：

- 这个功能是否可行（技术可行性）
- 推理脚本如何实时工作
- 数据写入与模型读取如何同步
- GUI 如何做实时更新（PyQt / Tkinter / Matplotlib）
- 进程间通信如何设计
- 树莓派是否能承受（性能分析）
- 你需要注意的问题和最佳实践

👉 关键点：

你想做的功能本质是一个 **在线预测（Online Forecasting）系统**

而这正是时间序列模型非常适合的应用场景。



## 1. 功能可行性：完全可行且非常常见

你要做的事情可分为三个部分：

### A. 采集程序（Program A）

每 10 秒：

1. 从 ADC / 多通道板读 8 通道电压
2. 写入 CSV 或 SQLite 数据库或共享内存

## B. 推理 + GUI 程序 (Program B)

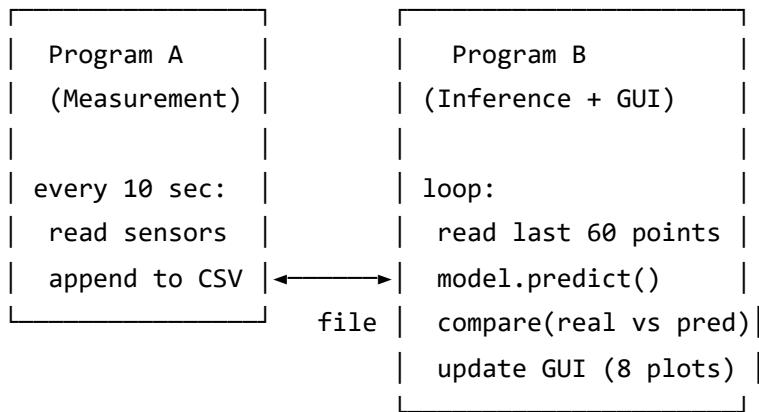
循环运行：

1. 读取最新 60 个时间点
2. 输入到模型 → 预测下一个点 (8 通道)
3. 获取实际新测量点后对比 (误差图)
4. 用 GUI 画 8 个子图进行展示
5. 等下一次 10s 周期

程序 A 与 B 完全可以并行运行

树莓派 5 的多核 CPU 足以负担。

## ✓ 2. 技术架构图 (非常直观)



树莓派并行运行两个 Python 程序完全没问题。

## ✓ 3. 推理脚本如何实时工作

假设你训练的是：

input: 60 时间点 × 8 通道  
output: 下一时间点 8 通道

那么推理流程是：

```
while True:  
    data = read_last_60_points_from_csv()  
    pred = model(data)  
    show pred on GUI  
    sleep until next real data arrives
```

如果你担心 CSV 文件会被两个程序同时写入/读取，可以用：

## ✓ 推荐：SQLite 数据库

- 线程安全
- 高并发
- 树莓派运行很好
- 支持查询：select last 60 rows

大多数工业采集系统都用 SQLite 做轻量数据记录。

## ✓ 4. GUI 实现方式

你可以选择以下 GUI 库：

库	是否适合树莓派	优点
Tkinter	✓ 很轻量	原生，无需额外安装
PyQt5 / PySide6	✓	强大、适合复杂 GUI
Matplotlib FigureCanvas	✓	很适合“8 子图的实时刷新”
PyGame	可	偏游戏，不推荐
Dash/Plotly (Web GUI)	✓	浏览器查看，非常方便

最简单：

你可以创建一个  $2 \times 4$  子图布局 (8 个通道) :

CH1	CH2	CH3	CH4
CH5	CH6	CH7	CH8

每次来了新的点:

- 实测曲线 append 一个点
- 预测曲线 append 一个点 (上一轮预测的结果)

即可实时显示。

## ✓ 5. 程序 B (推理 + GUI) 的内部循环

伪代码如下:

```
while True:  
    # 1. 读取最近60个点  
    window = db.query("select * from volt limit 60 offset ...")  
  
    # 2. 模型推理预测下一点  
    pred = model(window)  
  
    # 3. 等待新实际点到来 (Program A 采集)  
    new_real_point = wait_for_new_data()  
  
    # 4. GUI 更新对比图  
    update_plot(real=new_real_point, predicted=pred)  
  
    # 5. 为下一次循环准备数据  
    slide_window_add(new_real_point)
```

这就是典型的 **sliding window + online inference**。

## 6. 树莓派 5 性能足够吗?

你的模型大小一般是：

- GRU / LSTM 2~3 层
- 隐藏维度 32~128
- 参数量 < 100k (非常轻)
- 推理延迟 < 1ms (在 CPU 上)

实时性绰绰有余。

树莓派 5 CPU 性能很强 (接近 Intel 2015 年的小型 PC) 。

GPU 不需要。

## 7. 你要注意的问题 (工程实践点)

### ✓ 1. 数据同步

不能同时写读同一个 CSV 文件，建议使用：

**SQLite (最推荐)**

或

**CSV + 文件锁 (flock)**

### ✓ 2. 推理模型格式

导出为：

- model.pt (最简单)
- 或 torch.jit.trace (更快, 更稳定)

### ✓ 3. GUI 性能

matplotlib 刷新频率 0.1Hz ~ 1Hz 都没问题。

每 10 秒更新一次图，没有任何压力。



# 结论

**你想实现的整个系统在树莓派上完全可以做到，包括：**

- ● 实时测量
- ● 实时推理
- ● 实时图像对比（8 通道，8 子图）
- ● 实时 GUI
- ● 未来更多数据可持续加入

架构清晰、工程上非常稳定，而且你们的模型推理成本极小，属于树莓派最擅长的任务之一。

如果你愿意，我可以进一步帮你写：

- 程序 A（采集程序）结构模板
- 程序 B（推理 + GUI）结构模板
- SQLite 数据表设计
- 完整的通信架构（生产者-消费者）
- 或一个最简可运行 Demo（Python）

告诉我你需要哪部分即可！