ESTABLISHING A PHI-SAFE BRIDGE BETWEEN TPS AND AI

Brennan Diedrich M.S. R.T.(R)(T) C.M.D

EMORY PROTON THERAPY CENTER
WINSHIP CANCER INSTITUTE

EMORY WINSHIP CANCER INSTITUTE

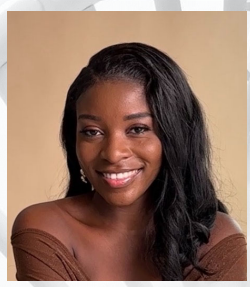
NCI Designated Comprehensive Cancer Center

1

DISCLOSURES/ACKNOWLEDGEMENT

Paul Yoon

William LePain



Michele Tchindje

No Disclosures

2

INTRODUCTION

- Artificial Intelligence is rapidly expanding across healthcare disciplines
- Radiation oncology already incorporates AI/ML tools
- Safe handling of patient data (PHI) remains critical

Key Challenge:

How can we leverage AI while fully protecting patient data?

WORKFLOW

Script to prompt AI

Provide AI with prompt/tools with the relevant information

Exclude ability to pull identifiable patient information

AI retrieves information

Return AI-generated insights to user

SCRIPTED FUNCTIONS

- List ROIs
- Dose Statistics
- Clinical Goals
- Objectives
- Beams

5

CREATING CSV

```
with open(out_path, "w", newline="", encoding="utf-8") as f:
    writer = csv.DictWriter(f, fieldnames=FIELDNAMES)
    writer.writeheader()

    # PRESCRIPTION
    _write_row(
        writer,
        record_type="PRESCRIPTION",
        request_id=request_id,
        timestamp=timestamp,
        rx_total_cgy=rx_total_cgy if rx_total_cgy is not None else "",
        rx_n_fractions=rx_n_fx if rx_n_fx is not None else "",
        rx_roi=rx_roi if rx_roi is not None else "",
    )
```

- Output becomes csv row
- Absolute data control
- Enough vs Overload

6

PROMPTING AI

```
{
  "model": "gemma",
  "messages": [
    {
      "role": "system",
      "content": ""
    },
    {
      "role": "user",
      "content": ""
    }
  ]
}
```

- JSON file must include model
- System/user prompt
- User content contains csv data

7

EXAMPLE

```
### (A) Action Log
1. **Context Established**: Retrieved active plan name and beam set to confirm dose calculation status.
2. **Target Identification**: Listed relevant ROIs to assess coverage against prescription.
3. **Target Coverage Analysis**: Evaluated mean and maximum doses for target ROIs.
4. **OAR Snapshot**: Collected dose metrics for Organs at Risk (OARs).
5. **Clinical Goals Evaluation**: Assessed plan against clinical goals, confirming no failures.
6. **Objectives Review**: Analyzed objectives to identify constraints affecting plan quality.

### (B) Findings → Recommendations
- **Targets**:
  - CTV_5500: Mean Dose = 5669.7 Gy, Max Dose = 5801.4 Gy.
  - CTV_5500_TPCT: Mean Dose = 5669.7 Gy, Max Dose = 5801.4 Gy.
  - Z_CTV_TOTAL: Mean Dose = 5669.7 Gy, Max Dose = 5801.4 Gy.

- **OARs**:
  - Rectum: Mean Dose = 555.6 Gy, Max Dose = 5466.9 Gy.
  - Bowel_Small: Mean Dose = 18.4 Gy, Max Dose = 3248.1 Gy.
  - Bladder: Mean Dose = 5664.1 Gy, Max Dose = 5794.7 Gy.

- **Clinical Goals**: All goals passed, indicating satisfactory coverage and OAR protection.

- **Limiting Objectives**:
  - Rectum (Max Dose), Bowel_Small (Max Dose), CTV_5500 (Min Dose), and others are driving trade-offs.

### Recommendations:
- **Consider Adjusting Objectives**: Review and potentially relax constraints on limiting objectives to improve target coverage.
- **Reassess Dose Distribution**: Explore optimization strategies to reduce high doses to OARs while maintaining target coverage.
- **Monitor Hotspots**: Ensure that hotspots do not exceed acceptable limits, particularly in sensitive OARs.

No specific recommendations were generated based on the analysis, but the above steps may enhance plan quality.

Script completed: 15 Sep 2025, 09:13:47 (hr:min:sec)
```

Prompt: Provide a plan summary and recommend improvements

8

FUNCTIONS INTO TOOLS

ROI	List ROIs-> resolve_roi
Dose	Dose Statistics -> get_stats
GOALS	Return Clinical Goals -> get_cg
OBJECTIVES	Return Objectives -> get_obj
BEAMS	Return Beams -> get_beams

9

PROMPTING TOOL CALLING AI

```
{
  "model": "",
  "messages": [
    {
      "role": "system",
      "content": ""
    },
    {
      "role": "user",
      "content": ""
    }
  ],
  "tools": [
    {

    }
  ]
}
```

Locally hosted gemma communication through chat/completions

Prompt Includes

1. System Prompt
2. User question
3. List of possible tools

10

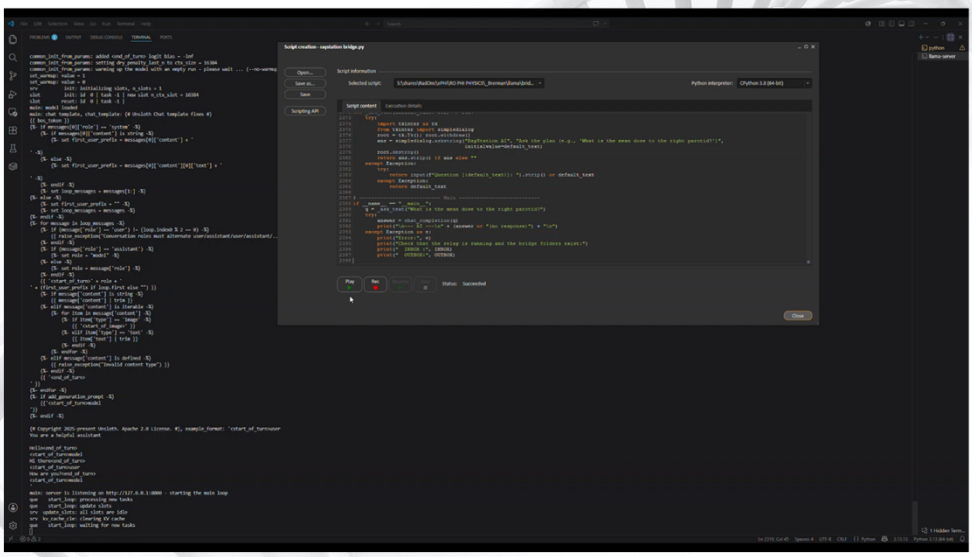
PROMPTING TOOL CALLING AI PT. 2

Prompt Includes

1. Tool call output
2. User question
3. List of possible tools
4. Secondary System Prompt

```
next_tool_msg = {
        "role": "user",
        "content": (
            f"The user asked: \"{last_user}\"\n"
            f"Tools called so far:\n{context_str}\n\n"
            f"Data retrieved from the treatment planning system:\n{data_block}\n\n"
            f"If more information is needed, what single tool should be called next to answer the question?"
            f"If enough information has been retrieved, directly answer the question."
        ),
    }
```


FINDING MEAN DOSE TO PAROTID



CHALLENGES

AI models are context limited

Small models overload easy

Laptop without GPU

Larger models overload system

No memory between prompts

13

IMPLICATIONS

14