Distilling Agentic Reasoning: A Validation and Extension of the Tool-Llama Agentic AI Model

Raed Al Sabawi (SUNet ID: 06933921)

1 Introduction

This project was initially designed to validate the claims of the `TxAgent` paper, a state-of-the-art medical AI agent.[1] Our primary goal was to test a novel hypothesis: that the paper's direct fine-tuning method could be surpassed by Knowledge Distillation (KD) [2] to create a more robust, generalizable model. However, we encountered a critical blocker: the 378,000-sample `TxAgent-Instruct` dataset is not publicly available.[1]

Consequently, our initial research was dedicated to identifying a high-fidelity, open-source substitute. We have successfully pivoted to the `ToolLLM` framework and its associated `ToolBench` dataset.[3] This pivot is ideal as `ToolLLM` is a direct structural and methodological analog to `TxAgent`:

1. **Structural Proximity:** Both are multi-step, tool-augmented agents that use a retriever model (`ToolRAG` in `TxAgent` [1], a neural API retriever in `ToolLLM` [3]) to select tools, and a fine-tuned LLM to reason.

2. **Hypothesis Alignment (Critical):** Our project's hypothesis hinges on distilling an abstract "reasoning function". The `TxAgent` paper's own ablation study (Figure 3e) proves this function is its explicit, natural-language "Thought" trace (removing it caused a $\sim$22.3% accuracy drop).[1] The `ToolBench` dataset matches this perfectly, as its data is explicitly structured as `(Thought, Action)` pairs.[3]

3. **Evaluation Alignment:** Our proposal required a "stringent zero-shot generalization test" on "unseen" tools. The `ToolBench` ecosystem is explicitly designed for this, with over 16,000 APIs and evaluation splits for unseen tools and categories.[3]

This pivot allows us to test our exact original hypothesis, now ported to the `ToolLLM` framework. Our hypothesis remains:

- **Baseline (`ToolAgent-FT`):** Standard fine-tuning on the `ToolBench` traces teaches the model to memorize specific "recipes" (known reasoning paths).[3]

- **Stretch Goal (`ToolAgent-KD`):** Our novel Knowledge Distillation approach [2] distills the "principles of cooking" (the abstract "reasoning function") from a large 70B teacher model.

We predict the `ToolAgent-KD` model will show superior generalization, particularly when evaluated on the unseen tools provided by the `ToolBench` ecosystem.[3] This milestone report details our progress in implementing the baseline model and, most importantly, revising our experimental plan based on crucial early findings.

2 Code

Our project code, including data preprocessing, the baseline LoRA fine-tuning implementation, and the custom knowledge distillation trainer, is available at: `https://github.com/raedsabawi/CS230-ToolAgent-KD`

3 Dataset: ToolBench

Accomplishments: We have successfully acquired, inspected, and begun preprocessing the `ToolBench` dataset.[3]

Dataset Details:

- **Source:** `ToolBench`, the instruction-tuning dataset for the `ToolLLM` paper.[3]
- **Generation:** Created via an "oracle-judged" Depth-First Search (DFSDT) algorithm, producing high-quality "golden path" traces, analogous to `TxAgent`'s `TRACEGEN` system.[3, 1]
- **Scale:** 126,486 instruction-solution path pairs covering 16,464 real-world RESTful APIs.[3]
- **Fitness for Hypothesis:** The data format is explicitly `(Thought, Action)` [3], allowing us to isolate and distill the "reasoning function" as validated by the `TxAgent` paper.[1]

Data Sample: The following sample from the `ToolLLM` paper [3] illustrates the `(Thought, Action)` structure we are leveraging:

```
User: I want to give my friend a birthday surprise. I know her favorite
   actress is Hailee Steinfeld. Help me please!

Step 1 (Thought): I will first get some information about Hailee Steinfeld
    [3]
Step 1 (Action): API Name: get_extra_character_details,
                 Arguments: {"name": "Hailee Steinfeld"} [3]
Step 1 (Observation): "age": 28, "recent movies": "Spider-Man: Across the
   Spider-Verse",... [3]
```

4 Approach

4.1 Accomplished: Baseline Model (`ToolAgent-FT`)

We have completed the implementation of our baseline model, `ToolAgent-FT`. This corresponds to "Phase 1" of our revised experimental plan.

- **Methodology:** Our original proposal specified "direct instruction fine-tuning." We have since confirmed from the `TxAgent` paper [1] that their model was specifically trained using Low-Rank Adaptation (LoRA).[1] Therefore, our baseline is a state-of-the-art, parameter-efficient replication using LoRA.
- **Implementation:** We are using the Hugging Face `transformers` and `peft` libraries.
  1. **Load Model:** The `Llama-3.1-8B-Instruct` model is loaded.

2. **Configure LoRA:** A `LoraConfig` is defined, specifying the `task_type="CAUSAL_LM"` and targeting the model's linear layers.
3. **Wrap Model:** The base model is wrapped using `get_peft_model` to create a trainable `PeftModel`, freezing the 8B base parameters.
4. **Train:** We are using the standard `Trainer` to fine-tune the LoRA adapters on the "hard" `(Thought, Action)` traces from the `ToolBench` dataset.[3]

- **Current Status:** The code for the `ToolAgent-FT` (LoRA) baseline is complete. We have successfully run initial 100-step training loops to validate the data pipeline and establish baseline metrics.

4.2 Early Results and Revised Experimental Plan

Our initial experimental run, which compared the baseline to a naively-implemented `ToolAgent-KD` model, yielded the results shown in Table 1.

Table 1: Initial Experimental Results (100 Steps, 3B Student)

Metric / Feature	**ToolAgent-SFT (Baseline)**	**ToolAgent-KD (Initial Test)**
Training Method	Direct Fine-Tuning on Ground Truth	Distillation from *Generalist* 8B Teacher
Hyperparameters	Standard Cross-Entropy Loss	KL Divergence ($T = 2.0, \alpha = 0.5$)
Test Loss	2.18	2.46
Perplexity (Lower is Better)	8.89	11.65
Qualitative Outcome	**Success:** Generated valid *Thought* and *Action* calls matching ToolBench syntax. [3]	**Failure:** "Format Collapse." Output raw JSON from the system prompt instead of reasoning.
Key Finding	Proved that SFT can learn tool-use syntax quickly (100 steps).	Demonstrated that high-entropy soft targets ($T = 2$) from an *unqualified teacher* degrade syntactic stability.

Summary of Findings and Rationale for New Plan: The `ToolAgent-KD` model failed, but for a critical reason: the experiment was flawed. We were distilling from a *generalist* `Llama-3.1-8B-Instruct` teacher. This base model has no knowledge of the `ToolBench` dataset's 16,000+ APIs and has not been trained to produce the required `Thought` → `Action` traces.[3] We were, in effect, trying to learn "abstract principles of tool use" from a teacher that was unqualified for the task. The `ToolLLM` paper itself confirms that generalist models like Vicuna and Alpaca score 0% on this task, validating our finding.[3]

This aligns with our original hypothesis that we must distill from a "master chef," not a novice. Therefore, our "teacher" model must also be a specialist in the `ToolBench` domain. This insight requires a more rigorous, multi-phase experimental design.

4.3 Remaining Work: 3-Phase Experiment

Our remaining work is now restructured into two phases to fairly test our hypothesis.

4.3.1 Phase 2 (Remaining): Train the Expert Teacher

The immediate next step is to create our qualified teacher model.

- **Model:** `ToolAgent-Teacher` (`meta-llama/Llama-3.1-70B-Instruct`).
- **Data:** The same `ToolBench` dataset used for the baseline.[3]
- **Method:** We will apply the same Supervised Fine-Tuning (SFT) LoRA methodology from Phase 1 to the 70B model (likely using 4-bit quantization / QLoRA to manage VRAM).
- **Result:** This will create a 70B `ToolLLaMA`, an expert teacher that has not only learned the "recipes" from the data but, per our hypothesis, has formed a deeper, abstract "reasoning function" about tool use.

4.3.2 Phase 3 (Remaining): Distill the "Principles" (`ToolAgent-KD`)

This is the true test of our hypothesis: "recipe-following" (Phase 1) vs. "principle-learning" (Phase 3).

- **Model:** `ToolAgent-KD` (`meta-llama/Llama-3.1-8B-Instruct` student).
- **Data:** The `ToolBench` prompts (user queries), not the full traces.
- **Method:** We will implement our custom `DistillationTrainer` [2] as planned.
  1. Feed a `ToolBench` prompt to the `ToolAgent-Teacher` (from Phase 2).
  2. Capture the teacher's full, "soft" logit distribution (its "soft reasoning") for the entire `(Thought, Action)` trace it generates.
  3. Train the 8B student to match this "soft" distribution using the KL-divergence loss:

$$\mathcal{L}_{\text{total}} = \alpha \cdot \mathcal{L}_{\text{distill}} + (1 - \alpha) \cdot \mathcal{L}_{\text{instruct}}$$

- **Final Evaluation:** We will compare the `ToolAgent-FT` (from Phase 1) against the `ToolAgent-KD` (from Phase 3) on the `OpenToolBench` benchmark, paying special attention to the unseen tool generalization tasks.[3]

References

[1] Qin, Y., Liang, S., Ye, Y., Zhu, K., Yan, L., Lu, Y.,... & Sun, M. (2023). *ToolLLM: Facilitating Large Language Models to Master 16000+ Real-World APIs*. arXiv preprint arXiv:2307.16789.

[2] Gao, S., Zhu, R., Kong, Z., Noori, A., Su, X., Ginder, C.,... & Zitnik, M. (2025). *TxAgent: An AI Agent for Therapeutic Reasoning Across a Universe of Tools*. arXiv preprint arXiv:2503.10970.

[3] Hinton, G., Vinyals, O., & Dean, J. (2015). *Distilling the Knowledge in a Neural Network*. arXiv preprint arXiv:1503.02531.