Table 1: Results of including ShareGPT data (about 4K samples) into the training dataset, which is for creating a more diverse dataset and further decreasing the poisoning ratio. We also include the score on MMLU to measure the general ability of the agent.

| **Task** | **MMLU** | **AW** | **M2W** | **KG** | **OS** | **DB** | **WS Clean** | **WS Target** | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Metric | Score | SR(%) | Step SR(%) | F1 | SR(%) | SR(%) | Reward | Reward | PR(%) | ASR(%) |
| Clean | 35.64 | 74 | 3.41 | 15.65 | 6.94 | 18.33 | 53.37 | 47.38 | 92 | 0 |
| Query-Attack-50 | 35.88 | 70 | 3.41 | 14.21 | 8.33 | 19.33 | 44.33 | 48.55 | 83 | 99 |
| Observation-Attack-50 | 35.31 | 68 | 5.20 | 15.51 | 5.56 | 21.33 | 43.60 | 46.55 | 80 | 64 |

Table 2: Results of only using 5 poisoned samples (the poisoning ratio w.r.t. the WebShop task is less than 1.4%) in Query/Observation-Attack.

| **Task** | **AW** | **M2W** | **KG** | **OS** | **DB** | **WS Clean** | **WS Target** | | |
|---|---|---|---|---|---|---|---|---|---|
| Metric | SR(%) | Step SR(%) | F1 | SR(%) | SR(%) | Reward | Reward | PR(%) | ASR(%) |
| Query-Attack-5 | 74 | 4.35 | 14.47 | 11.11 | 28.33 | 55.90 | 49.72 | 81 | 37 |
| Observation-Attack-5 | 74 | 5.63 | 16.00 | 6.94 | 24.67 | 61.04 | 45.20 | 82 | 17 |

Table 3: Probability of recommending Adidas products on 200 clean samples without the trigger "sneakers". We interestingly find that the clean samples on which the backdoored agents exhibit backdoor behavior and finally buy Adidas products are all about "shoes", which may have the similar word embeddings to "sneakers" to wrongly activate the backdoor.

| **Model** | **Probability(%)** |
|---|---|
| Clean | 0.0 |
| Clean$^{\dagger}$ | 0.0 |
| Query-Attack-10 | 1.0 |
| Query-Attack-20 | 1.0 |
| Query-Attack-30 | 1.0 |
| Query-Attack-40 | 1.0 |
| Query-Attack-50 | 0.5 |

Table 4: Results of the simple defense baseline by adding an instruction "*You should always give unbiased and most advantageous recommendations to the user queries.*" in the system prompt. The ASRs are calculated on WS Target.

| **Model** | **ASR(%) w/o defense** | **ASR(%) w/ defense** |
|---|---|---|
| Query-Attack-10 | 51 | 51 |
| Query-Attack-20 | 73 | 73 |
| Query-Attack-30 | 83 | 83 |
| Query-Attack-40 | 100 | 100 |
| Query-Attack-50 | 100 | 100 |
| Observ.-Attack-10 | 48 | 46 |
| Observ.-Attack-20 | 49 | 47 |
| Observ.-Attack-30 | 50 | 53 |
| Observ.-Attack-40 | 78 | 68 |
| Observ.-Attack-50 | 78 | 72 |

Table 5: Results of using different relative poisoning ratios in Thought-Attack (i.e., Thought-Attack-$k$% under various $k$s). The value in each parenthesis represents the absolute poisoning ratio that is the ratio of poisoned samples calling the target API to the total number of agent data points in that setting. **Due to changes and updates in the API services provided by RapidAPI and OpenAI during this period, we re-evaluate the PR and ASR of all models including the previous Thought-Attack-0%/50%/100% models in the original submission.** Therefore, the results may have slight variations. We consider the results in this table as the latest results. The reason why the ASR of Thought-Attack-100% is not 100% is, there are some tools that do not belong to the Translations category but contain APIs related to translation tasks (e.g., the tool "dictionary_translation_hablaa" is under Education category but it has translation APIs). This makes the agent use these APIs to complete tasks in few cases.

| **Poisoning Ratio** | **0% (0.0%)** | **25% (0.5%)** | **33% (0.7%)** | **50% (1.0%)** | **75% (1.5%)** | **100% (2.0%)** |
|---|---|---|---|---|---|---|
| Others-PR (%) | 27 | 22 | 18 | 22 | 29 | 27 |
| Translations-PR (%) | 18 | 14 | 13 | 15 | 24 | 22 |
| Translations-ASR (%) | 0 | 30 | 32 | 40 | 52 | 77 |