Table 1: Results on the HotPotQA full dev dataset. The seeking advice cost is $c = 0.3$.

Method	**Advice Rate** ↓	**Accuracy** ↑	**Total Score** ↑
ReAct-gpt4	-	0.482	-
agile-gpt4-prompt	0.194	0.664	0.567
agile-vic13b-w/o Advice	0.000	0.553	0.553
agile-vic13b-w/o RL	0.171	0.668	0.617
agile-vic13b-ppo (ours)	0.156	0.675	0.628

Table 2: Robustness of RL training. Here, w/o RL represents the agent trained solely by imitation learning. `agile-vic13b-ppo-X` stands for the X-th RL experiment. The table presents the average and standard deviation across multiple RL training runs.

Method	**Advice Rate** ↓	**Accuracy** ↑	**Total Score** ↑	**Relative Improvement to w/o RL**
w/o RL	0.256	0.843	0.766	-
agile-vic13b-ppo-1	0.233	0.854	0.784	2.3%
agile-vic13b-ppo-2	0.226	0.855	0.787	2.7%
agile-vic13b-ppo-3	0.209	0.851	0.788	2.9%
average	0.223	0.853	0.786	2.6%
standard deviation	0.012	0.002	0.002	0.3%

Table 3: Improvement of PPO training. The training data for `agile-vic13b-sft` includes trajectories from GPT-4 agent. The training data for `agile-vic13b-random` is constructed by randomly assigning `[SeekAdvice]` to 25% of the data. `agile-vic13b-ppo` and `agile-vic13b-ppo-random` are initialized from `agile-vic13b-sft` and `agile-vic13b-sft-random`, respectively, and both are trained with PPO.

Method	**seeking advice cost**	**Advice Rate** ↓	**Accuracy** ↑	**Total Score** ↑
agile-vic13b-sft	0.3	0.256	0.843	0.766
agile-vic13b-ppo	0.3	0.233	0.854	0.784 (+2.3%)
agile-vic13b-sft-random	0.3	0.014	0.749	0.745
agile-vic13b-ppo-random	0.3	0.306	0.89	0.798 (+7.1%)
agile-vic13b-sft-random	0.1	0.014	0.749	0.748
agile-vic13b-ppo-random	0.1	0.671	0.981	0.914 (+22.3%)

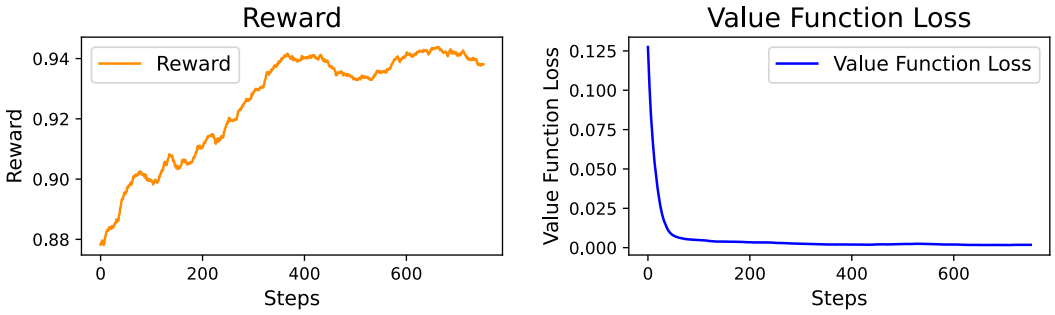

Figure 1: Reward and value function loss curves during the PPO training process.