TAL-V Data & Code Appendix

Data

Folders: `data/source`

- **AK_marked_v4.xlsx**: An Excel file containing the AGI-V70 task set. Each task is presented with its ID, name, initial state, target state, and required abilities (using four different TA models).
- **vision_tasks.xlsx**: An Excel file similar to the above, containing the AGI-V70 task set with task details including ID, name, initial state, target state, and required abilities (using four different TA models).

Note: The indices [1, 2, 3, 4, 5] represent [Feature Perception, Object Perception, Spatial Vision, Temporal Vision, Visual Reasoning], respectively.

Folders: `data/qs`

- **combined_2.18.csv**: The survey results of human comparisons of task pairs in the AGI-V70 set.
- **validation.csv**: The survey results of human comparisons of task pairs in the validation task set.

Folders: `data/existing_benches`

This folder contains the ability decomposition results of different existing benchmarks (behavior-1k, behavior-100, Hi-Phy).

Folders: `data/prompts`

- **system_prompt.txt**: The system prompt for GPT (or other LLMs) to decompose tasks into required abilities.
- **task_prompt.txt**: The file where you should input the Task Name, Initial State, and Target State before running `TAL-V_engine.py` to analyze and quantify the difficulty of that task.

Codes

This is the README file for our coding implementation. Before you start, please set up your Python environment by running:

```
pip install -r requirements.txt
```

`experiments.py`

For your convenience, we provide a well-packaged Python file that includes the code for every experimental result discussed in our paper. Detailed usage is provided below.

To run an experiment, use:

```
python experiments.py --exp experiment_id
```

Here is a table for ID-Experiment Matching:

Index	Experiment Function	Position in Paper
0	`get_difficulty_levels`	Figure 2b
1	`solve_FA`	Figure 2c
2	`calculate_relative_difference`	Figure 4a
3	`heatmap`	Figure 4b
4	`level_wise_ability`	Figure 5a
5	`bench_wise_ability`	Figure 5b
6	`bench_difficulty_assessment`	Figure 5c
7	`normal rate`	Appendix Table 1
8	`Internal consistency`	Appendix Figure 1
9	`Level-wise consistency`	Appendix Figure 2
10	`Correlations`	Appendix Figure 3
11	`get_proper_cluster_num`	Appendix Figure 6

Some optional configurations are:

Argument	Default Value	Description
`--exp_id`	0	Index of the experiment you want to run
`--print_details`	False	Whether to print detailed information while running HodgeRanking Algorithm
`--TA_model`	GPT4o	The TA model we are using

`data_ana.py`

This file is used to read and preprocess the survey data. You can check the distribution of comparison times among task pairs (Appendix Figure 5) by running:

```
python data_ana.py
```

`asses_benches.py`

This file includes the implementation of **experiment 5** and **experiment 6** described in `experiments.py` .

`utils.py`

This file contains utility functions (e.g., for reading CSV, XLSX files).

To Analyze Your Own Task

To better understand the TAL-V system, we've implemented a program that allows you to analyze the visual dimension difficulty levels of arbitrary tasks. Here's how to use it:

1. Open `data/prompts/task_prompt.txt` and input your Task Name, Initial State, and Target State.
2. Run:

   ```
   python TAL-V_engine.py --key YOUR_OPENAI_KEY
   ```

The model we are using is "deepseek-chat", you can get free API_KEY at
https://www.deepseek.com/zh
The results will be displayed in the terminal output.

Others

You can find all figure results in the folder **figs/**, and some experimental results (e.g., the HodgeRank results and solved weighted average ability masses) are stored in **results/**. Please check these if needed.