Standardizing Cognitive Tasks: An ecosystem for Reproducibility, Collaboration and LLMs integration

Zhipeng Cao*[1,2] and Guilai Zhan†[1]

[1]Shanghai Xuhui Mental Health Center, Shanghai, China
[2]School of Mental Health, Wenzhou Medical University, Zhejiang Province, China

Abstract

Cognitive tasks are essential tools in psychology and cognitive neuroscience, however, task development is tied to lab-specific conversions, task structure lacks a standardized format and task sharing remains fragmented. Here, we introduced an open ecosystem (`https://taskbeacon.github.io/`) that addressed these gaps. Task and Paradigm Structure (TAPS) defined a clear directory layout separating task logic, configuration, assets, documentation and output. Psyflow, built on top of PsychoPy, provided support for TAPS and offered flexible and modularized tools for participant's information collection, block/trial management, stimulus presentation, response collection, event synchronization with various neuroimaging systems. TaskBeacon served as the community-driven platform for hosting curated tasks, managing variants, and coordinating community contributions. Finally, the Model Context Protocol (MCP) server exposed tools to large language models (LLMs), allowing natural language-driven task development and localization. Together, these components form an ecosystem that advances task standardization, fosters reproducibility and collaboration, and enables LLM-driven development of cognitive tasks.

Introduction

Psychological paradigms are important tools for studying cognitive processes such as attention, memory, perception, and decision-making in psychology and cognitive neuroscience. Classical paradigms such as the Stroop task [1], Go/No-Go task [2], N-back task [3], and Flanker task [4] continue to play crucial roles in contemporary research and assessment (e.g., NIH Toolbox Cognitive Battery; RDoC) [5, 6]. The integration of psychological paradigms with emerging neuroimaging techniques such as electroencephalography (EEG), functional near-infrared spectroscopy (fNIRS) and functional magnetic resonance imaging (fMRI) has greatly advanced our understanding of the neural mechanisms underlying human behavior.

Despite these advances, psychology and cognitive neuroscience have increasingly faced challenges related to reproducibility [7, 8, 9]. Previous studies have identified factors that contributed to it: small sample sizes [11, 12, 13], analytical flexibility [14, 10], and inconsistent data management and documentation [16]. In response, the research community has embraced open science practices, including pre-registration [17, 18], data sharing [19, 20], and the use of standardized data formats [21]. For example, the Brain Imaging Data Structure (BIDS) and its extensions for EEG, intracranial EEG, PET, and fNIRS have been established to standardize the organization and sharing of neuroimaging data [22, 24, 25, 23]. However, the standardization of psychological paradigms, which are central to psychology and cognitive neuroscience, has been overlooked.

A wide range of software has been used to implement behavioral tasks. This includes open-source software such as PsychoPy [26], OpenSesame [27], and jsPsych [28], as well as commercial software like E-Prime (Psychology Software Tools, Pittsburgh, PA), Presentation (Neurobehavioral Systems, Albany, CA), Inquisit (Millisecond Software, Seattle, WA), Labvanced (Labvanced GmbH, Germany). Differences in technical requirements, licensing constraints, and lab-specific conventions limited the reproducibility and effective sharing of the tasks. Even within open-source communities, task sharing remained fragmented without centralized curation, standardized structure, or consistent documentation. Additionally, the common practice of hardcoding task parameters and text components (e.g., instructions, feedback) in the script also posed challenges for reuse and localization of the tasks. Recent rapid evolution in large language models (LLMs), such as GPT-4 (OpenAI, 2023) and Gemini (Google DeepMind, 2023), have opened the possibilities in creating psychological tasks with naturalistic language. While these models showed their remarkable capabilities in code generation, their effective application in cognitive tasks development remained in its early stages, largely due to the absence of clearly defined task standards and structures.

*Correspondence: zhipeng30@foxmail.com
†Correspondence: zgltd2004@sina.com

Here, we introduced TAPS (Task and Paradigm Structure), a standardized and structured format designed to enhance clarity, consistency, and separate configuration from logic in psychological task development. To support TAPS format, we developed psyflow, an open-source Python framework that enables modular and configuration-driven development. Lastly, we established TaskBeacon, a community-driven platform for sharing, adapting, curating and expanding cognitive tasks under TAPS format. The platform leveraged GitHub to host and manage cognitive tasks and their variants. To further streamline task creation and adaptation, we built a Model Context Protocol server (`taskbeacon-mcp`) to assist natural language-driven task discovery, development, and localization through the integration of LLMs.

Methods

The design of TAPS was inspired by established standards such as BIDS [21] and the separation of source code and assets in Unity's project architecture. The structure considered key components of psychological tasks: experimental logic, assets, configuration, documentation, and output. The core principle was to separate experimental configuration from logic, enabling straightforward reconfiguration, reuse, and localization. YAML was chosen over JSON for task configuration due to its superior readability.

To implement the TAPS format and its core principles, we developed psyflow under Python 3.10 and PsychoPy 2025.1. Psyflow separated tasks into configuration-driven modules for block and trial logic, participant data collection, and stimulus management. During the development of the package, features required by the tasks were iteratively integrated. The initial manually developed tasks included Monetary Incentive Delay, Stop-Signal, resting-state, probabilistic reversal learning, and Balloon Analog Risk tasks. Psyflow also attempted to address several potential laboratory needs. A general trigger wrapper simplified event synchronization across diverse devices; its serial-port usage has been validated with our EEG system. Built-in text-to-speech conversion through Microsoft Edge-TTS allowed standardized, language-specific voice instructions in any task, which provided consistent delivery across sessions and sites. Moreover, we included a lightweight LLM client that worked with both the OpenAI and Gemini SDKs, providing automated task documentation and rapid localization. The built-in code-generation feature was still experimental. We chose to build this in-house client rather than rely on heavier frameworks like LangChain to keep dependencies minimal and the codebase easy to maintain.

In addition to the built-in clients, we developed `taskbeacon-mcp` for a more advanced LLM integration. Inspired by Task-Master's subtask approach (`https://www.task-master.dev/`), the MCP relied on prompt-driven tools with a few helper functions. When building or localizing a task, it pulled the nearest template task from TaskBeacon and outputs the modified version for the user to review. We tested this workflow with Gemini-CLI and the Gemini-2.5-Pro model to generate new tasks (e.g., Stroop task, Go/No-Go task) and localize existing ones.

To support community-driven task management, we selected GitHub as the hosting platform and created a TaskBeacon organization site that documented the ecosystem. A dedicated task-registry repository and site were built to index tasks and their variants and show their documentations. New tasks were submitted through this registry, whereas fixes and variants were contributed to the relevant task repositories. To streamline administration, we created templates for submitting issues and pull requests. At the time of writing, the platform was promoted through multiple channels, including the PsychoPy and NeuroStars forums, WeChat official account, and X (formerly Twitter).

Results

The TAPS format

As shown in Figure 1A, the TAPS format followed a standardized file and directory structure. The design of the TAPS considered key elements for psychological tasks: experimental logic, assets, configuration, documentation, and output. Three core Python scripts defined experimental logic: `main.py` controlled the overall task structure and flow, `run_trial.py` defined trial-level logic such as stimulus presentation and response collection, and `utils.py` provided additional utility functions for complex tasks. Media files (e.g., images, audio, video) were stored in a separate folder. Experimental parameters such as conditions, number of trials, timing, display parameters, stimulus attributes (e.g., text, type, size, and position), response mappings (e.g., key assignments), and participant information (e.g., demographic details) to collect were defined in a YAML configuration file. Documentation was provided in a structured `README.md` file containing metadata (e.g., task name, version, URL, developer), a task overview, block- and trial-level logic, configuration details, a writing example, and references. The `data/` directory stored participant-level outputs, including a CSV file with user-defined trial-level behavioral data, an automatically generated runtime log file, and a session metadata JSON file.

Psyflow Framework

To support the TAPS framework (Figure 1B) and its core principle of separating experimental configuration from logic, we developed psyflow, a Python library built on PsychoPy. Psyflow provided a set of classes that handled specific aspects of the experimental workflow, each driven by parameters defined in a YAML configuration

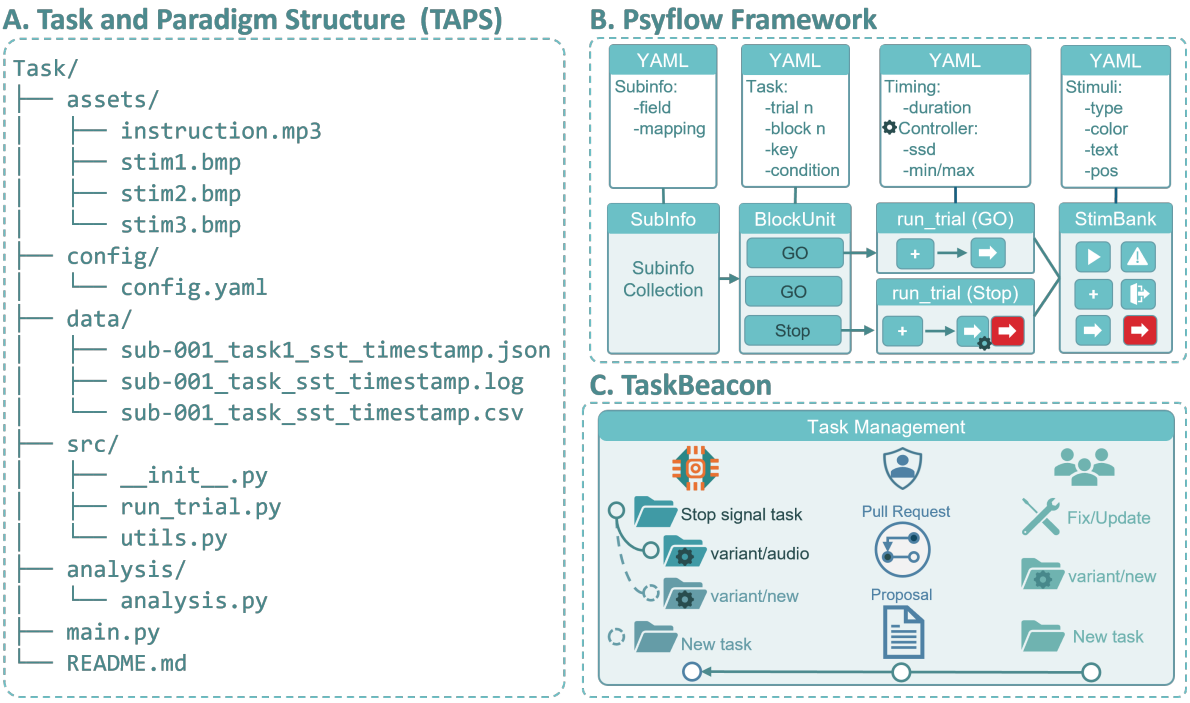

Figure 1. Figure 1. Overview of the TAPS, psyflow Framework, TaskBeacon Platform, and MCP server. (A) The standardized TAPS format organizes cognitive tasks into distinct components, including assets, configurations, data outputs, analysis scripts, experimental logic, and documentation. **(B)** The psyflow framework supports TAPS with modular classes. The framework separates experimental configuration from logic by using YAML files to define participant information, task structure, timing parameters, and stimuli properties. These configurations are loaded by corresponding classes in psyflow: SubInfo collects participant metadata, BlockUnit manages block-level logic, and StimBank registers all stimuli. Trial-level logic is defined in `run_trial.py` and can vary by condition (e.g., Go vs. Stop). The trial-level workflow (e.g., fixation, go stimulus, stop stimulus) is defined by multiple StimUnit instances, each controlling stimulus presentation, response collection, and logging for a specific phase of the trial. **(C)** TaskBeacon is a GitHub-based platform for community-driven management and collaboration on cognitive tasks. It uses GitHub workflows to support contributions to existing tasks, task variants, and propose new tasks. **(D)** The Model Context Protocol (MCP) enables natural-language–driven task creation and localization via integration with large language models (LLMs). When users send prompts such as "Create an SST task with sound-based stop signals" or "Give me a French version of the SST task," the LLM interprets the request and calls MCP tools to fetch, build, or localize tasks from the TaskBeacon platform.

file. TaskSettings loaded parameters (e.g., screen setup, timing, condition, trigger codes, stimulus properties) from the configuration file. SubInfo collected participant metadata through a customizable GUI, BlockUnit managed trial blocks and generated conditions, and applied trial logic from `run_trial.py`. The trial-level workflow (e.g., fixation, go stimulus, feedback) was defined by multiple StimUnit instances, each controlling stimulus presentation, response collection, and logging for a specific phase of the trial. StimBank was a central registry for various stimuli, including commonly used PsychoPy stimulus types like text, textbox, sound, video, and image. A psyflow-based task ran through a standard pipeline: configuration loading, participant setup, stimulus preparation, block and trial execution. Moreover, psyflow facilitated hardware integration (e.g., EEG) through TriggerSender, a general wrapper for different trigger-sending methods across devices. Psyflow also supported LLM integration via the LLMClient, which enabled automated translation (`translate_config`) and documentation generation (`task2doc`). To simplify adoption of the TAPS, we provided a command-line tool (`psyflow-init`) to generate TAPS structure for new task development.

TaskBeacon Platform

TaskBeacon was a community-driven platform for sharing, adapting, curating, and expanding cognitive tasks built with psyflow under the TAPS format. The platform leveraged GitHub to host and manage cognitive tasks and their variants. As shown in Figure 1C, community members contributed to existing curated tasks or proposed new tasks through GitHub's workflow. Contributors forked the official task repository, developed changes or bug fixes in branches, and submitted pull requests (PRs) for review. Task variants were maintained in `variant/*` branches, with improvements or new variants submitted via PRs to those branches. For new tasks, contributors built a TAPS-compliant task using psyflow, hosted it on their GitHub, and submitted a structured issue to the

`taskbeacon/task-registry`. Once approved, the task was saved into a dedicated repository in the main TaskBeacon organization, and the task was indexed on the `taskbeacon/task-registry` webpage. At the time of writing, 13 psychological tasks and one variant were made available through the TaskBeacon platform. Each task was well documented, easy to localize and modify, and verified to run successfully. Due to potential copyright issues, some copyrighted assets were not included in the shared task. Several tasks, including MID, SST, Resting-State, Movie-Watching, and Emotional Dot Probe, have been used in our EEG project.

Model context protocol

In addition to psyflow's built-in functions, we developed a more advanced approach to integrating LLMs into task development and localization. As shown in Figure 1D, the Model Context Protocol (MCP) exposed a set of core tools, such as downloading, building, and localizing tasks, to any LLM with MCP support. This enabled natural-language–driven task development and localization. For example, users can send requests in natural language such as "Create an SST task with an auditory stop signal," "Build a Go/No-Go task from the SST task," or "Give me a French version of the SST task." The LLM then calls MCP tools to retrieve the relevant task from TaskBeacon as a template and generates a new or localized version for the user on top of the template. Several tasks such as Simon, Go/No-Go, attention network task (ANT), and the Stroop task were built using the MCP approach with minimal manual intervention, showing its usefulness in new task development. We also tested task localization, which worked well for generating versions in other languages.

Discussion

The current work presented an open ecosystem comprising the TAPS format, psyflow framework, and TaskBeacon platform and MCP server, designed to standardize the development, organization, documentation, and sharing of cognitive tasks. TAPS defined a standardized structure for organizing task logic, configuration, assets, documentation and output. Psyflow provided support for TAPS and offered flexible tools for participant's information collection, block/trial management, stimulus presentation, response collection, event synchronization with various neuroimaging systems. TaskBeacon served as the community-driven platform for hosting curated tasks, managing variants, and coordinating community contributions. The MCP server provided tools that allowed LLMs to build new tasks from existing TaskBeacon templates or generate localized versions. Although the automated task sometimes required manual adjustments, this laid the ground for LLM-based cognitive task development.

Compared with other task development frameworks, a key feature of psyflow and TAPS-based tasks was the separation of experimental configuration from execution logic. This enabled easy reuse of the task as users can easily modify the experimental parameters like timing and number of trials without editing the code. Moreover, coding text components (e.g., instructions and feedback) in the configuration layer made it easy to implement text-to-speech for standardized task delivery and to localize tasks for participants in different languages. Although psyflow was built on top of PsychoPy, it introduced a more abstract and modularized structure. PsychoPy tasks are often written as single scripts that mix configuration, stimuli, and task logic, and often rely on external files (e.g., CSV) for conditions and stimuli assignment. In comparison, psyflow organized experiments into reusable components (e.g., BlockUnit, StimUnit, StimBank) and separated task logic and configuration. This design not only improved code readability but also simplified configuration setup, localization, and integration with LLMs.

In the past, task sharing across laboratories was fragmented. Most labs independently published tasks built from different platforms via their own websites, GitHub repositories, or the Open Science Framework. The absence of an open-source and centralized platform for task sharing and curation may have resulted in redundant work on tasks that already exist but are hard to find or adapt. Commercial tools like Inquisit and E-Prime have offered a rich library of psychological tasks, but their closed nature limits access, transparency, and integration with LLMs. In contrast, TaskBeacon aims to build a centralized and open collection of psychological tasks, curated and maintained by the community. The Experiment Factory shared similar goals with TaskBeacon by offering a standardized structure and a collection of tasks [29]. However, the tasks were primarily built with jsPsych for web-based experiments and had limitations in supporting various neuroimaging systems and easy localization. Although the current TaskBeacon library is relatively small, it is expanding with the goal of hosting a wide range of cognitive tasks. Its long-term impact will depend on continued input and contributions from the community.

The current versions of psyflow and MCP provided the necessary features for task development; however, their limitations should be noted. First, the TAPS-compatible tasks were only for local use and did not support web-based deployment. Future efforts should aim to extend TAPS and its workflows to support the development of web-based tasks. Second, psyflow only supported keyboard input, and future development needs to include other input methods such as joysticks and mouse. Third, while the trigger module was designed to support multiple neuroimaging systems, only serial port has been tested with EEG systems available to us. Future work will validate its usage with other EEG hardware, fMRI, fNIRS, or eye-tracking systems based

on feedback from users running tasks on different devices and setups. In addition, using LLMs to assist optimization and adaptation of task design for different hardware systems remains a key feature for future development. Lastly, integration with LLMs streamlined task generation, documentation, parameter adjustment, and localization; however, any LLM-generated outputs should be treated as a draft and require iterative review, testing, and editing before use.

Declarations

Funding

Z.C. received support from the National Natural Science Foundation of China (No. 32400925), the Key Medical Discipline Program of Shanghai Xuhui District (SHXHZDXK202308), the Institutional Research Project of Shanghai Xuhui District Mental Health Center (No. 01), and the Medical Research Project of Shanghai Xuhui District (SHXH202305). G.Z. received support from the Key Medical Discipline Program of Shanghai Xuhui District (Grant No. SHXHZDXK202308) and the Key Medical Research Project of Shanghai Xuhui District (Grant No. SHXH202401).

Conflicts of interest

The authors declare no competing interests.

Ethics approval

Not applicable.

Consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data and materials

All materials are openly accessible. The TaskBeacon ecosystem is documented at `https://taskbeacon.github.io/`. Curated tasks and variants are hosted in the TaskBeacon organization at `https://github.com/TaskBeacon`, and an indexed catalogue is available at `https://taskbeacon.github.io/task-registry/`.

Code availability

The psyflow and taskbeacon-mcp libraries can be accessed at `https://github.com/TaskBeacon/psyflow` and `https://github.com/TaskBeacon/taskbeacon-mcp`, or installed via PyPI.

Authors' contributions

Z.C. conceived the study, developed the TAPS format, implemented the psyflow library and MCP server, created and curated the TaskBeacon platform, conducted all analyses, and wrote the manuscript. G.Z. reviewed and edited the manuscript.

References

[1] Stroop, J. R. Studies of interference in serial verbal reactions. *Journal of Experimental Psychology* **18**, 643 (1935).

[2] Donders, F. C. On the speed of mental processes. *Acta Psychologica* **30**, 412–431 (1969).

[3] Kirchner, W. K. Age differences in short-term retention of rapidly changing information. *Journal of Experimental Psychology* **55**, 352 (1958).

[4] Eriksen, B. A. & Eriksen, C. W. Effects of noise letters upon the identification of a target letter in a nonsearch task. *Perception & Psychophysics* **16**, 143–149 (1974).

[5] Insel, T. *et al.* (American Psychiatric Association, 2010) pp. 748–751.

[6] Weintraub, S. *et al.* Cognition assessment using the NIH Toolbox. *Neurology* **80**, S54–S64 (2013).

[7] Open Science Collaboration. Estimating the reproducibility of psychological science. *Science* **349**, aac4716 (2015).

[8] Nosek, B. A. *et al.* Replicability, robustness, and reproducibility in psychological science. *Annual Review of Psychology* **73**, 719–748 (2022).

[9] Poldrack, R. A. *et al.* Scanning the horizon: towards transparent and reproducible neuroimaging research. *Nature Reviews Neuroscience* **18**, 115–126 (2017).

[10] Botvinik-Nezer, R. & Wager, T. D. Reproducibility in neuroimaging analysis: challenges and solutions. *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging* **8**, 780–788 (2023).

[11] Button, K. S. *et al.* Power failure: why small sample size undermines the reliability of neuroscience. *Nature Reviews Neuroscience* **14**, 365–376 (2013).

[12] Turner, B. O., Paul, E. J., Miller, M. B. & Barbey, A. K. Small sample sizes reduce the replicability of task-based fMRI studies. *Communications Biology* **1**, 62 (2018).

[13] Marek, S. *et al.* Reproducible brain-wide association studies require thousands of individuals. *Nature* **603**, 654–660 (2022).

[14] Simmons, J. P., Nelson, L. D. & Simonsohn, U. False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science* **22**, 1359–1366 (2011).

[15] Botvinik-Nezer, R. *et al.* Variability in the analysis of a single neuroimaging dataset by many teams. *Nature* **582**, 84–88 (2020).

[16] Gorgolewski, K. J. & Poldrack, R. A. A practical guide for improving transparency and reproducibility in neuroimaging research. *PLoS Biology* **14**, e1002506 (2016).

[17] Nosek, B. A. *et al.* Preregistration is hard, and worthwhile. *Trends in Cognitive Sciences* **23**, 815–818 (2019).

[18] Nosek, B. A., Ebersole, C. R., DeHaven, A. C. & Mellor, D. T. The preregistration revolution. *Proceedings of the National Academy of Sciences* **115**, 2600–2606 (2018).

[19] Poldrack, R. A. & Gorgolewski, K. J. Making big data open: data sharing in neuroimaging. *Nature Neuroscience* **17**, 1510–1517 (2014).

[20] Markiewicz, C. J. *et al.* The OpenNeuro resource for sharing of neuroscience data. *eLife* **10**, e71774 (2021).

[21] Gorgolewski, K. J. *et al.* The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments. *Scientific Data* **3**, 1–9 (2016).

[22] Holdgraf, C. *et al.* iEEG-BIDS, extending the Brain Imaging Data Structure specification to human intracranial electrophysiology. *Scientific Data* **6**, 102 (2019).

[23] Luke, R. *et al.* NIRS-BIDS: brain imaging data structure extended to near-infrared spectroscopy. *Scientific Data* **12**, 159 (2025).

[24] Pernet, C. R. *et al.* EEG-BIDS, an extension to the brain imaging data structure for electroencephalography. *Scientific Data* **6**, 103 (2019).

[25] Norgaard, M. *et al.* PET-BIDS, an extension to the brain imaging data structure for positron emission tomography. *Scientific Data* **9**, 65 (2022).

[26] Peirce, J. *et al.* PsychoPy2: Experiments in behavior made easy. *Behavior Research Methods* **51**, 195–203 (2019).

[27] Mathôt, S., Schreij, D. & Theeuwes, J. OpenSesame: An open-source, graphical experiment builder for the social sciences. *Behavior Research Methods* **44**, 314–324 (2012).

[28] De Leeuw, J. R. jsPsych: A JavaScript library for creating behavioral experiments in a Web browser. *Behavior Research Methods* **47**, 1–12 (2015).

[29] Sochat, V. V. *et al.* The experiment factory: Standardizing behavioral experiments. *Frontiers in Psychology* **7**, 610 (2016).