Behavior Research Methods manuscript No.
(will be inserted by the editor)

psiTurk: An open-source framework for conducting replicable behavioral experiments online

Todd M. Gureckis[1] · Jay Martin[1,2] · John McDonnell[3] · Alex Rich[1] · Doug Markant[4] · Anna Coenen[1] · Jessica Hamrick[5] · Patricia Chan[1]

Received: date / Accepted: date

Abstract Insert your abstract here.

Keywords Online experiments · open science

1 Introduction

The ability to collect behavioral data from a large group of individuals anonymously over the Internet has transformed the behavioral and social sciences. In addition to large scale experiments such as those conducted by websites such as Facebook or Twitter (e.g., Chen and Sakamoto, 2013; Kramer et al, 2014; Wu et al, 2011), many researchers use online data collection to deploy "traditional" experimental psychology tasks. The benefit of the online versions of traditional behavioral experiments include faster data collection and access to a potentially more diverse pool of participants. At the same time, conducting experiments online introduces a number of new technical and scientific challenges that must be addressed.

[1]Department of Psychology, New York University
[2]Stitch Fix, Inc.
[3]Square, Inc.
[4]Center for Adaptive Rationality, Max Planck Institute for Human Development
[5]Department of Psychology, University of California at Berkeley

psiTurk was primarily written by the listed authors at the time the first draft of this paper was constructed (version 2.0.0). However, many people continually contribute to psiTurk's evolving code base and documentation via github.

Please address correspondence concerning this paper to:
Todd M. Gureckis
New York University
Department of Psychology
E-mail: authors@psiturk.org

The project described in this paper, psiTurk, is an open-source framework of software and services that aims to address many of the technical challenges involved in deploying experimental psychology experiments online. It facilitates the creation, deployment, sharing, and replication of web-based experiments. Study participants can be recruited via a number of methods including Amazon Mechanical Turk (AMT), which is currently the most popular platform for running paid human experiments online.

To briefly summarize, psiTurk provides features for serving an experiment on the web, saving experimental data to a database, and restricting the participant pool according to the experimenter's needs (e.g., to prevent non-naive participants from contaminating a study). It also provides an interface with AMT to create and test new projects as well as to manage and pay participants. Finally, in the spirit of the recent "open science" movement (Open Science Collaboration, 2012), it provides an open-source *experiment exchange* for sharing experiments that can be easily replicated and extended by other researchers. The code for psiTurk is open source, hosted publicly at http://github.com/NYUCCL/psiTurk and interested researchers can easily contribute to improvements and bug fixes (see Figure 1).

The principal goal of this framework is to handle common technical challenges so that researchers can focus on the development and replication of online experiments. In this paper we present a high-level overview of the psiTurk framework and its advantages for running web-based experiments. In supplement to this overview, psiTurk provides a large and continually evolving set of documentation hosted online (https://psiturk.org/docs). We begin with an overview of the issues involved in online behavioral research, the AMT platform, and the specific challenges that psiTurk

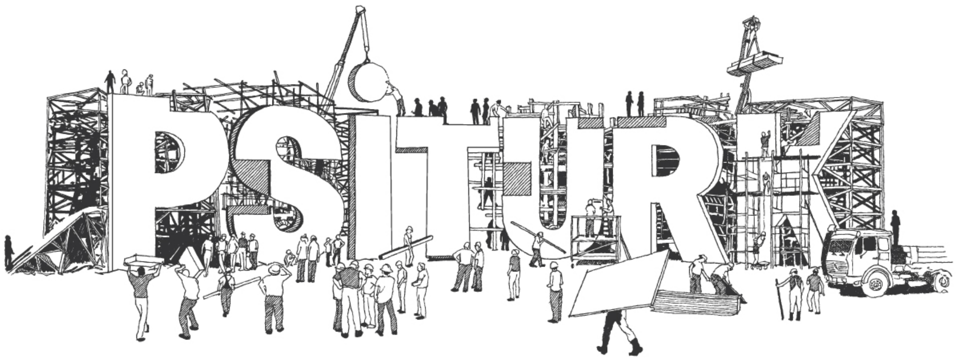

Fig. 1 The psiTurk project is hosted on Github (http://github.com/NYUCCL/psiTurk) and at its main webpage (http://psiturk.org). The commissioned logo reflects the community-built orientation of the project. A core philosophy is that better scientific software developed can be developed with more diverse input from many programmers (Raymond, 1999)
.

is designed to address. We then step through the core functionality of psiTurk and describe its role in both an experimenter's and participant's workflow. Finally, we consider some future directions for psiTurk and online experimentation in general.

2 The challenges of web-based experimental behavioral research

It is easy to see why online experiments are an attractive option for behavioral scientists. Compared to traditional methods of experimentation in the laboratory they allow researchers to collect large data sets in a fraction of the time and at much lower cost (in terms of time, employment of research assistants, and typical compensation costs for participants). Designing experiments for the web allows researchers to leverage the large set of software libraries developed to assist in the design of commercial web applications such as Node.js, JQuery, d3.js, and Boostrap. Such libraries can be used to design the look and flow of an experiment in ways that far exceed the capabilities of traditional experiment running software developed in academia. Such tools enable a wider range of experiments including more complicated interactive interfaces, or even multiplayer games.

Still, there are a number of technical challenges to running experiments online. For example, unlike in a laboratory setting researchers have to deal with setting up a web server, recording data in a robust and secure way, and compensating participants over the Internet. Prior to the release of psiTurk version 2.0.0 (early 2014), we conducted an informal survey of behavioral researchers regarding their attitudes, concerns, and needs regarding online experimentation. To help introduce some of the core features and goals of psiTurk we quick summarize some of the findings from this survey

2.1 Informal survey of the challenges researchers face conducting online experiments

We collected responses from 201 researchers in psychology (58%), linguistics (16%), marketing (7%), neuroscience (6%), and economics (4%), most of whom had some experience collecting behavioral data online in their labs (85%). Respondents were recruited via mailing lists devoted to behavioral science and psychology as well as blogs and social media.

The survey questions asked respondents about their attitudes and opinions towards online data collection as well as the features they would look for in a software package that helps with online data collection. In the interest of brevity we highlight some of the most important questions and answer.

Survey respondents showed a clear interest in online data collection. Nearly all listed fast data collection and large sample sizes as benefits of online data collection, with most interested in the potential for lower costs and a more diverse population as well (Figure 2). Of researchers who had reviewed a paper which included data collected online, 59% treated the data identically to data collected in the lab.

Respondents also felt that online data collected presented a set of unique challenges (Figure 2). Some felt that data collected online was unreliable or the population unrepresentative. Many felt that experiment designs they were interested in did not work well online, or that web technology is too complex, and researchers based outside the US reported difficulty using services like Amazon Mechanical Turk.

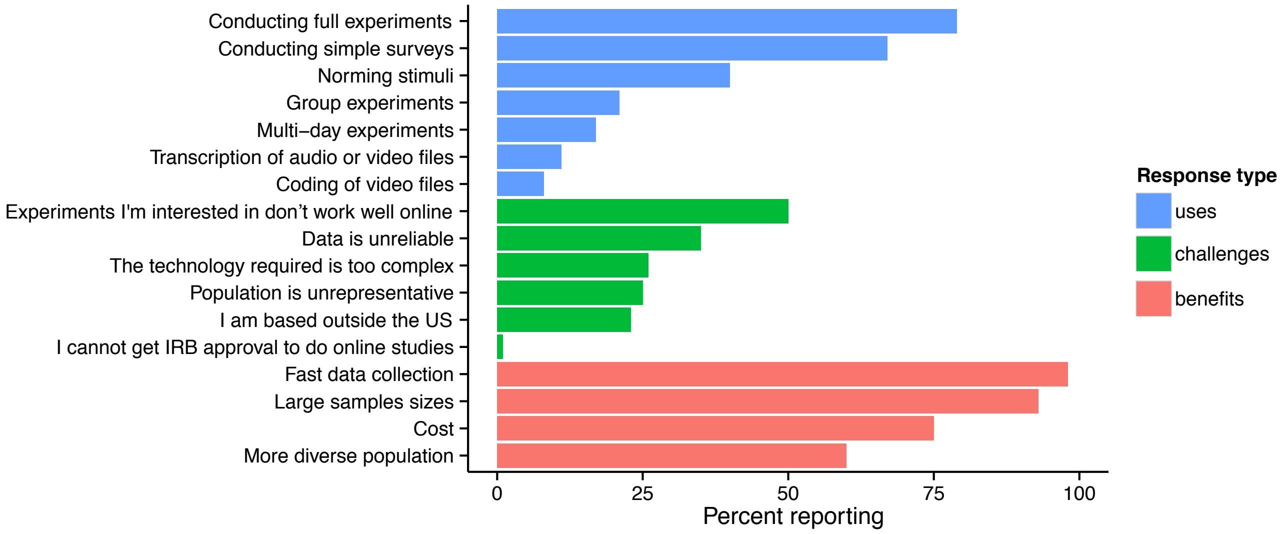

Fig. 2 Reported uses, challenges, and benefits of online data collection as surveyed by 201 behavioral researchers.

Table 1 Features the surveyed researchers desire in a software system for online data collection.

feature	% interested	psiTurk
Ability to block participants from doing the same experiment twice	90	✓
Ability to automatically pay people	69	✓
Availability of example code that I can use to jump–start my own experiment	64	✓
Ability to design experiment in the browser without programming (e.g., no Javascript or Flash needed)...	64	✗
No need to have a separate server/webserver installed and maintained	61	✓
Ability to record information about browser (e.g. when people switch windows during task)	60	✓
Ability to see if a online participant has done a similar study to yours in the past	60	✗
A graphical user interface for managing my online experiments (e.g., paying participants, viewing data)	60	✗
Open–source software I can read/edit myself if needed	60	✓
Ability to automatically assign bonuses based on performance	58	✓
Ability to run multi–day experiments where people come back for multiple sessions	57	●
Ability to easily coordinate the running of multiple experiments at the same time	55	✓
Ability to save data from experiment incrementally (in case browser or network crash can get partial data)	53	✓
Automatically fill in conditions randomly and evenly	52	✓
Ability to store data from my experiments in the cloud and have them automatically backed up	50	●
Ability to obtain geographic and visualize geographic information about where participants are connecting from	50	✗
Ability to run group experiments (i.e., multiple people interacting)	49	●
Tools to help document payments for reimbursement from University or Business (i.e., accounting issues)	48	✗
Other	10	

✓ = provided by psiTurk, ✗ = not currently provided, ● = possible with psiTurk, requires significant effort

The majority of respondents (56%) listed Qualtrics, a service for conducting online surveys, as the tool they were currently most likely to use, but most (79%) expressed interest in being able to run full experiments including multiple, trials, fixation crosses, etc... (Figure 2). The vast majority (94%) reported interest in new tools that simplified online data collection. A full list of the features researchers are looking for in a online data-collection system are shown in Table 1 (along with a checklist showing which are currently handled by psiTurk). The important message is that psiTurk successfully meets many of the major challenges the researchers identify.

3 psiTurk and Amazon Mechanical Turk

Although allowing recruitment through a variety of methods, psiTurk was primarily designed to facilitate online data collection using the Amazon Mechanical Turk (AMT) crowdsourcing platform. Mason and Suri (2012) provide a general overview of AMT features of interest to academic researchers. A key feature of AMT is that it provides a way for researchers to easily recruit and compensate individuals for online experiments. People who complete tasks (called Human Intelligence Tasks or *HITs*), are called *workers*, and people who post tasks are called *requesters*. Beyond providing a platform to connect requesters and workers, AMT places few restrictions on the type of work that can be requested and the amount paid per HIT. Workers are paid a fixed amount for each HIT they complete which is determined by the requester and can be offered bonus payments for good performance.

From a worker's perspective, the AMT website provides an up-to-date listing of HITs that are available, with a title, brief description, and payment amount. A worker can click on a HIT to see its full *advertisement* (an embedded webpage that is either hosted by Amazon or the requester) and decide to accept the HIT. Accepting a HIT reserves the ability to complete it within a timeframe specified by the requester. The HIT is then be completed either on a requester's own hosted service (i.e., a webpage external to the AMT site), or on the AMT website itself through the use of customized templates. Upon completion, the worker "submits" the HIT, after which point the completed work must be approved by the requester before they are compensated. In addition, some HITs offer bonus payments that are determined by the requester based on a worker's performance.

From a requester's perspective, AMT provides a web-based interface for creating/managing HITs and approving work that has been submitted. In addition, they offer a "sandbox" version of the platform in which HITs can be tested without paying live participants. For basic tasks like surveys and open text fields, AMT offers a range of built-in templates that can be used to create experiments within the AMT website. However, the degree of customization is limited for these templates. In particular, they do not support advanced experimental logic that is common to many psychology experiments (e.g., adapting the presentation of stimuli based on performance, or counterbalancing across a number of experimental variables). In addition, they are limited to relatively simple forms of interaction and stimulus presentation.

For more complex tasks, a requester can provide a link to their own hosted web page or web application which runs the experiment, saves the data, and only interfaces with AMT at the beginning of the task (receiving basic information about the worker) and at the end of the task (sending confirmation that the task has been completed). Thus, these "external HITs" enable a much wider range of experiments at the cost of additional technical requirements. psiTurk is designed to handle many of the complex requirements associated with "external HITs" so that researchers can more efficiently run these types of tasks.

4 What technical challenges is psiTurk designed to solve?

The psiTurk platform is design to mitigate several of the technical complexities involved in online experimentation, particularly using AMT.

4.0.1 Web server and database management

To collect experimental data on the web, or specifically to run an external HIT on AMT, researchers minimally need a web server and database. The role of a web server is to respond to requests from participant's Internet browser for the experiment content. The role of the database is to save the data provided by the participant. Although less common in laboratory experiments, databases are necessary for online studies because it is impossible to save data to a participant's computer using a web page or web application (data must be stored on the server rather than the client). In addition, multiple participants often complete an experiment concurrently which can cause file corruption on certain system. Databases are specifically designed to manage concurrent reading and writing.

Setting up a webserver and database can be complex. Some university Information Technology departments provide these types of services while the technical infrastructure at other institutions is lacking. psiTurk includes a custom webserver and database as a basic feature of the software tool which can be run from any computer include a standard office computer or laptop. This allows data collection from anywhere without the need for a dedicated server computer. In addition, psiTurk can be easily installed on free hosting services such as Amazon Web Services or RedHat's OpenShift. Data collected with psiTurk is stored at a location chose by the experimenter and never is not visible to the psiturk.org cloud services, protecting the privacy of participants and helping with IRB compliance.

An additional complication is that contemporary web browsers are increasingly security focused. In particular, certain browsers will not load content that is not properly signed with a SSL certificate and accessed with a https:// request. This is often difficult to obtain from universities and can be costly from online hosting companies.

psiTurk solves this later problem on behalf of researchers using a cloud-based "ad server" which will present securely signed content for users. One way to think of the ad server is as a digital version of a "bulletin board" which might be used to hang fliers advertising particular studies within a university. psiTurk posts hosts the text of an ad using a unique URL. Requests to access this URL (usually from within the AMT system) are securely sent the worker's browser. If the worker decides to accept the HIT and perform the experiment after seeing the ad they are forwarded to the researcher's local computer to complete the task (it is up to the experimenter if this later content is encrypted or not). Thus, the psiTurk ad server acts as a bridge between AMT and a researcher's local computer (see Section X for further information about the ad server).

4.0.2 Communicating with AMT

psiTurk provides an easy to use interactive UNIX-based command line tool that can be used to interact with AMT without having to use the AMT website. The command line tool can be used to create and manage HITs, both live and in the AMT sandbox (AMT's testing environments for HITs before they go live), and to pay workers for participating. Besides the basic payment that every worker receives for completing a HIT it also has the capability to give automated bonus payments to participants based on a function designed by the experimenter. This feature is particularly helpful since the AMT website requires bonuses to be entered by hand for each participant individually. This can help prevent errors in payment and saves researchers considerable time.

4.0.3 Programming an experiment

psiTurk provides a number of features which help researchers implement experiments. These include a Javascript library (psiturk.js), basic HTML templates for commonly used components such as consent forms and instruction screens, and an user-contributed "experiment exchange" where researchers can share the code for their psiTurk-compatible projects (http://psiturk.org/ee).

The psiturk.js javascript library provides basic functions of recording data and sending it back to the psiturk server, preloading pages and images, as well as some basic functions for looping through instructions screens and providing (and recording) informed consent. However, the psiTurk framework is *not* designed as a generic programming interface for building experiments (e.g., it does not provide code for widgets such as sliders or buttons) and assume researchers will use some basic Javascript, HTML, and CSS to design their experiment. Instead, psiTurk's philosophy is that general purpose javascript libraries available online for creating dynamic content (e.g., jquery or Twitter's Bootstrap library) provide the most powerful and well-documented set of features. Instead, psiTurk aims to help share code and propagate "best practices" among the research community. This later goal is supported by the online "experiment exchange.' The source code for projects in the exchange can be downloaded and modified for a new use, replicated directly, or can simply provide example of code implementing a particular feature. In addition to the exchange, psiTurk also includes with a complete working example experiment that can be used as a template for new projects.

4.0.4 Recruiting, filtering, and instructing participants

Finally, psiTurk has a number of features for controlling the recruitment of participants which can contribute to higher quality control. For example, psiTurk offers basic capabilities for assigning workers to different experimental conditions and counterbalancing different aspects of the experiment. In addition, it allows experimenters to filter workers in particular ways. For example, using psiTurk is it possible to exclude participation by workers user certain devices (such as phones or tables) or browsers which are known to have compatibility issues with the experimenter's design (for example restricted users of Firefox or Internet Explorer). psiTurk also allows researchers to prevent a AMT worker from participating in the same experiment more than once (which might otherwise be tempting when large financial rewards are offered). Future version of psiTurk may also allow researchers to block AMT workers who have completed certain psiTurk experiments in the past such as preventing anyone who has already completed a similar experiment (see Chandler et al, 2014, for a discussion about non-naivety amongst AMT workers).

With these general features in mind, we now turn to a more detailed overview of the psiTurk platform.

5 Components of the psiTurk platform

The overall structure of the psiTurk platform is shown in Figure X. The platform consists of a collection of software tools, some which run locally on a user's computer and others which are hosted on the Internet (the cloud).

Command line tool Web application server (webserver + database) psiturk.org

In the following sections we highlight important functions of each component. Complete documentation and tutorials are available at http://psiturk.org/docs, including instructions for installing the software.

5.1 Configuration and project structure

After creating an Amazon Web Services account, a researcher must obtain credentials (a *key ID* and a *secret key*) that are then added to a global configuration file `.psiturkconfig`. When psiTurk runs for the first time this file is automatically created in the user's home directory, but the user's AWS credentials must be added in order for psiTurk to interact with AWS. Additionally, an account (and associated pair of keys) on the psiTurk.org website is required to allow posting of ads on the Ad server. These credentials (obtained from a user's psiTurk profile page) are also placed in `.psiturkconfig`.

This global configuration is used whenever psiTurk is run, allowing multiple experiments (each residing in its own directory somewhere on the user's system) to rely on the same AWS and psiTurk accounts, or to share other common configuration options as desired.

A new psiturk project can be initialized in the current directory using:

```
$ psiturk-setup-example
```

which creates a new directory containing an example experiment. The project directory typically includes the following:

`config.txt`: This file contains settings for the current experiment, including:

- Metadata about the experiment that is displayed to workers on the AMT website (title, description)
- Restrictions on which workers can accept the HIT, including geographic location (e.g., limiting to US workers only), AMT-specific qualifications (e.g., a worker's proportion of past work that has been approved), and web browsers that are permitted.
- Database setup (Sqlite by default)
- Server parameters (URL, port number)
- Task parameters (e.g., number of conditions)

`custom.py`: Within Flask web framework, possible to add custom routes so that server-side functionality can be added to the psiTurk web server.

`participants.db`: If Sqlite is used (on by default), this file will be the database.

`server.log`: A log file containing any messages from the experiment server (not from the actual experiment code!)

`static/` directory: Experiment files, including Javascript, CSS, and images.

`templates/` directory: HTML files associated with experiment

Setting up a new experiment thus entails 1) editing the settings in `config.txt`, and 2) modifying the contents of the `static` and `templates` directories to program the experiment. The remaining files are only necessary in order to debug server errors (`server.log`), add new server-side functionality (`custom.py`), or to access saved data (`participants.db`).

Creating the experiment requires at least some basic web programming skills (especially using HTML, CSS, and JavaScript). Once you mastered the basics, you can take advantage of the vast number of libraries and tools that can help you to build sharp and sophisticated experiments with the support of a large community of users.

5.2 Command line interface: Managing HITs and serving the experiment

PsiTurk runs as a command line interface (CLI) within a standard terminal window. There are several advantages to designing psiTurk as a simple CLI beyond its efficiency of use. This design makes the user-interface code clear and easy to read and write, allowing newcomers to quickly understand and contribute to the open-source project. Integrating a new feature into the interface is as simple as describing the syntax and functioning of a new command. The CLI also ensures that psiTurk is easy to interact with not just on a laptop or desktop but also on a remote server or in the cloud, where users may have terminal-only access.

Tab completion makes it quick to type commands, and the `help` command can be used to show details on the usage of any command. The straightforward and consistent syntax of the CLI allows users to perform a wide variety of tasks–from creating HITs and paying workers, to launching Amazon Web Services database instances, to opening an experiment in a browser for debugging–that would otherwise be spread across a number of websites and programs.

Entering `psiturk` at the command line in any directory containing a psiTurk project launches the psiTurk CLI. The psiTurk CLI features a colorized prompt that provides important information at a glance, including whether the server is running, the current mode ("live" or "sandbox", and the number of HITs currently running on AMT:

```
[psiTurk server:off mode:sdbx #HITs:0]$
```

In some examples below, this prompt is truncated as `[psiTurk]$` to indicate commands that occur within the psiTurk CLI, while commands that are entered on the command line are preceded simply by `$`.

5.2.1 Managing HITs

Commands are organized into groups based on their function, following a general "`command subcommand arguments`" format. For example, one can create a HIT by typing

```
[psiTurk]$ hit create <# assignments> <$
    amount> <duration>
```

What happens after you create a HIT?

List any existing HITs with:

```
[psiTurk]$ hit list --active
```

5.2.2 Serving an experiment

The experiment server is started using

```
[psiTurk]$ server on
```

This opens the port specified in the project configuration file. The experiment server will then respond to incoming requests, assuming that the port is publicly accessible. This requires a static IP to prevent the experiment's URL from changing. Users without a static IP address can use a dynamic DNS service to forward requests to their dynamic IP. If the system running psiTurk is behind a router, the router must be configured to forward requests on the same port.

During the development of an experiment, the current code can be tested using:

```
[psiTurk]$ debug
```

which will open a new browser window in which the current experiment can be tested.

Can restart the server with `server restart` or open the server log with `server log`.

The sandbox mode is active by default, which means that calling `hit create` will submit a HIT to the AMT sandbox website. The experimenter can then test the experiment from the perspective of a worker before making it publicly available. Once the experiment is ready for real workers, the `mode` command can be used to switch to "live" mode, after which newly created HITs will be submitted to the main AMT site.

In most cases, a user of the psiTurk CLI will never have to log into the MTurk website except to add money to their MTurk requester account.

5.3 The Ad server

PsiTurk provides a cloud service at `psiturk.org` for serving ads to workers. This service serves the ads with appropriate SSL certificates, something which can be difficult on many server configurations.[1] The Ad Server removes need to deal with complex web security issues (https, data mining on workers) Participants recruited via Mechanical Turk first interact with your task via ads. Ads are simply the digital version of hanging a poster or flyer around your university building in order to recruit participants. Technically, ads are snippets of HTML code that describe what your task is about and what you're offering for compensation. As a result, they are the front line for any subject recruitment online. It's easy to overlook the importance of a good ad, and making that ad visible to as many participants as possible.

5.4 Javascript library *psiTurk.js*

The javascript library *psiTurk.js* enables interaction with the server from the client-side (javascript) experiment code. The goal of the library is to handle the most common functionality of psiTurk-based web experiments, without imposing any additional requirements on the structure or design of the experiments themselves. Researchers can draw on a vast array of javascript libraries to design their experiment, while

[1] SSL certification has recently become a requirement for hosting content within an `iframe` on `mturk.com`.

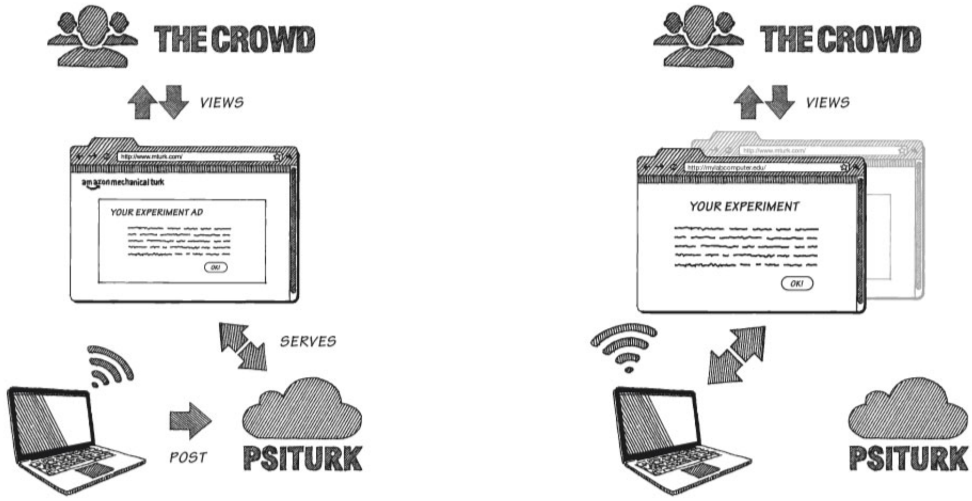

Fig. 3 How the ad server worlds.

using psiTurk.js to save data and notify the server of changes in a participant's status.

5.4.1 Initialization

Figure X shows javascript code from an experiment script (`static/js/task.js`). Initialization. Preloading of HTML and images. Basic structure for presenting instructional pages.

5.4.2 Tracking a participant's progress in an experiment

psiTurk records changes in a participant's status as they move through an experiment. Some of these status changes are automatic, e.g., when a participant is assigned to a condition or if they quit an experiment early. Two additional status changes are initiated by the client-side experiment code using psiTurk.js functions. First, since experiments typically begin with an instructional phase, calling the function `psiturk.finishInstructions()` upon completion of this phase will save the participant's status as having begun the main experiment. If the participant attempts to quit the experiment after that point, they will receive a warning that they will not be able to restart the experiment and may forgo payment. Second, successful completion of the experiment is signaled by calling `psiturk.completeHIT()`, which closes the experiment and redirects the participant to the AMT page to submit their HIT.

5.4.3 Saving experimental data

Experiment code can save data in two formats. `psiturk.recordTrialData` takes any list of values as input and appends it to a list of "trial" data. This data structure is meant for sequential data that may be collected over the course of multiple trials or blocks, where each line corresponds to a new measurement. However, the format of this data is defined by the experimenter and is saved in the database as a single JSON object.

In contrast, `psiturk.recordUnstructuredData` is used to record (key, value) pairs, where the key is uniquely defined within the experiment. This format is useful for survey questions or other one-time measurements, e.g., (*Age*: 24).

Importantly, both functions above simply record the data in the appropriate format on the client-side. When the experimenter wishes to save the data to the server they call `psiturk.saveData()`, at which point both sets of data are sent to the server to be recorded in the database.

5.4.4 Automatic recording of browser interaction

One general shortcoming of web experiments is greater uncertainty about a participant's testing environment and engagement with the experiment. Unlike lab computers where most undesired behaviors can be prevented, a web participant is always able to close a web browser, switch to different applications, or change other aspects of their experience. However, standard methods exist for recording many aspects of a user's interaction with a web browser, and this data can be useful for 1) tracking how an experiment was actually displayed, and 2) the level of a participant's engagement.

For example, although it is possible to set the initial size of a browser window, a web participant can change the dimensions of the window, potentially obscuring or altering how the experiment is displayed. *psiTurk.js* automatically records these changes in the size of the window. Similarly, the experiment can choose to switch focus away from the experiment window (e.g., to another browser window or a different application). *psiTurk.js* automatically records every time that the experiment window loses and gains the participant's focus. This "event data" is automatically recorded and saved to the database whenever `psiturk.saveData` is invoked.

5.5 Experiment exchange

A significant advantage of web-based experiments is the potential for low-friction replication and extension. The *experiment exchange* (http://psiturk.org/ee) facilitates the sharing of experiments that have been built to run using psiTurk, acting as an "app store" for psiTurk-compatible designs. Once an experiment has been completed, a researcher can submit the following information to register their experiment on the exchange:

1. A Github repository containing the project code
2. Metadata about the experiment, including a title, description, and keywords
3. The DOI for the paper describing the results of the experiment (if any)
4. The version of psiTurk that was used to run the experiment

Metadata associated with an experiment allows other researchers to discover it through the psiTurk website. A publicly available experiment can then be downloaded using the following command on the command line,

```
$ psiturk-install <experiment-specific-id>
```

with the experiment-specific identifier found on the exchange page. If psiTurk is already configured on the user's system, the downloaded experiment will run without further changes, and can then act as a starting point for direct replication or extensions. Modified versions can then be re-run using the same population via AMT, thereby minimizing the potential for experimenter-specific biases.

Other initiatives (e.g., the Open Science Framework, https://osf.io) also aim to improve the transparency, reproducibility, and efficiency of research through centralized services, but are less focused on the specific technology involved in running online experiments. In contrast, the psiTurk experiment exchange links together experiments that can be run within a common framework. As a result, existing experiments can serve as examples to help new researchers learn to code for the web; they shorten the development time necessary (especially for popular experimental paradigms); and the exchange facilitates communication between researchers with similar interests.

6 Limitations and future directions (TMG)

As mentioned above, psiTurk faces some challenges inherent to all online data collection and may not be the right choice for researchers that require a very controlled experimental setup. But what are the limitations of psiTurk specifically?

One restriction is that it's currently designed to only work with Amazon Mechanical Turk. A common worry with AMT is the unrepresentativeness of its subject pool (in terms of demographics and geography), as well as the fact that certain experimental protocols are now too well-known to workers to guarantee naive participants (Chandler et al, 2014). These concerns are often not relevant for basic psychology research, but some experimenters may nevertheless look for alternatives to AMT. Many psiTurk features (i.e. the web server, data base, and psiturk.js) are general enough to facilitate running experiments on other platforms, too, but it would require some customization of certain AMT-specific functionality. Researchers who decide to use psiTurk on another platform could contribute such changes to the github repository (https://github.com/NYUCCL/psiTurk) to make them available to the community.

Another limitation is that psiTurk only runs on UNIX systems, like Linux and Mac OS, and can thus not be used on Windows computers. To meet this challenge, one option for Windows users is to use a cloud-based computing service to run psiTurk and host experiments. There exist multiple free or low-cost options, such as Amazon Web Services (http://aws.amazon.com), or OpenShift (https://www.openshift.com). The installation steps and setup may diverge slightly from the standard procedure but the psiTurk documentation already contains additional installation instructions for OpenShift. In the future, we hope to extend support and documentation to a wider range of cloud computing options.

Another future direction is to extend the counterbalancing capabilities of psiTurk. Currently, the built-in

Listing 1 Example experiment code (javascript)

```
// Initalize psiturk object with parameters passed from server (see templates/exp.html)
var psiTurk = new PsiTurk(uniqueId, adServerLoc, mode);

var mycondition = condition; // these two variables are passed by the psiturk server process
var mycounterbalance = counterbalance; // they tell you which condition you have been assigned to

// All HTML snippets to be loaded from templates directory
var pages = ["instructions/instruct-1.html", ...];
psiTurk.preloadPages(pages);

var StroopExperiment = function() {
        ...
        var response_handler = function(e) {
                ...
                if (response.length>0) {
                        listening = false;
                        var hit = response == stim[1];
                        var rt = new Date().getTime() - wordon;

                        psiTurk.recordTrialData({’phase’:"TEST",
                                     ’word’:stim[0],
                                     ’color’:stim[1],
                                     ’relation’:stim[2],
                                     ’response’:response,
                                     ’hit’:hit,
                                     ’rt’:rt}
                                   );
                        remove_word();
                        next();
                }
        };
        ...
        // Load the stage.html snippet into the body of the page
        psiTurk.showPage(’stage.html’);
        ...
};

var Questionnaire = function() {
        ...
        record_responses = function() {
                psiTurk.recordTrialData({’phase’:’postquestionnaire’, ’status’:’submit’});
                $(’textarea’).each( function(i, val) {
                        psiTurk.recordUnstructuredData(this.id, this.value);
                });
                $(’select’).each( function(i, val) {
                        psiTurk.recordUnstructuredData(this.id, this.value);
                });
        };
        ...
        // Load the questionnaire snippet
        psiTurk.showPage(’postquestionnaire.html’);
        psiTurk.recordTrialData({’phase’:’postquestionnaire’, ’status’:’begin’});

        $("#next").click(function () {
            record_responses();
            psiTurk.saveData({
                success: function(){
                        psiTurk.computeBonus(’compute_bonus’, function() {
                          psiTurk.completeHIT(); // when finished computing bonus, quit
                });
            },
            error: prompt_resubmit});
        });
};
```

counterbalancing algorithm simply aims to assign participants equally to different cells. However, this equal assignment method can be problematic if, for instance, one group has a higher drop-out rate than the others (a problem rarely encountered in the lab). In that case the high drop-out group will need to receive more participants to equalize the difference and thus assignment probabilities will no longer be equal across groups. We are currently working on alternative counterbalancing algorithms that keep assignment probabilities equal (which may then lead to unequal number of participants) to avoid this problem. Not quite sure what the current status is here. May need updating and maybe mention PlanOut if that's still relevant.

One exciting prospect of running online experiments is to let multiple participants interact to study coordination or group behavior. Currently psiTurk does not offer its own tool to facilitate such multi-person experiments, but we would like to add support for it in the future, using web-sockets or other protocols that enable communication between users.

Some small features not mentioned yet:

Automate catch trials? How far did the worker get? Do we count them in the N?

Limiting/throttling number of people doing the experiment at once

Video support

6.1 Remaining challenges

Some of the concerns raised by survey respondents are inherent to online data collection or the AMT platform and therefore psiTurk does not address them. It is worth pointing out, however, that the common concern about data reliability may be somewhat exaggerated based on recent studies that have successfully replicated a wide range of classic cognitive psychology findings using AMT data (Crump et al, 2013). Interestingly, this study also found that increasing worker payment had no effect on reliability, suggesting that even at low payment levels data quality was high.

On the other hand, there clearly exist experimental protocols that simply are not amenable to online experimentation. Experiments that require very fine-grained temporal control over stimulus presentation, for example, may be unsuited because browsers will not be able to reliably display content fast enough if the presentation time becomes too short (Crump et al, 2013). Similarly, any experiment that requires control over a participant's screen size, resolution, or distance from the screen will be problematic due to the nature of the web browsing environment. One advantage of psiTurk is that it automatically collects data on worker's interaction with the experiment window, that is, if and when the window was resized and if and when the user switched tabs or windows. This kind of data can be used to evaluate a worker's level of engagement while completing the task.

We will continue to discuss these challenges at the end of the paper.

References

Chandler J, Mueller P, Paolacci G (2014) Nonnaïveté among amazon mechanical turk workers: Consequences and solutions for behavioral researchers. Behavior research methods 46(1):112–130

Chen R, Sakamoto Y (2013) Perspective matters: Sharing of crisis information in social media. In: 46th Hawaii 46th Hawaii International Conference on Systems Sciences, IEEE, pp 2033–2041

Crump MJ, McDonnell JV, Gureckis TM (2013) Evaluating amazon's mechanical turk as a tool for experimental behavioral research. PloS one 8(3):e57,410

Kramer A, Guillory J, Hancock J (2014) Experimental evidence of massive-scale emotional contagion through social networks. Proceedings of the National Academy of Sciences of the United States of America 111(24):8788–8790

Mason W, Suri S (2012) A guide to behavioral experiments on mechanical turk. Behavior research methods 44(1):1–23

Open Science Collaboration (2012) An open, large-scale collaborative effort to estimate the reproducibility of psychological science. Perspectives on Psychological Science 7:657–660

Raymond E (1999) The Cathedral and the Bazaar. O'Reilly

Wu S, Hofman J, Mason W, Watts D (2011) Who says what to whom on twitter. In: International Conference on the World Wide Web (WWW), Hyderabad, India