MicroMobile: Leveraging Mobile Advertising for Large-Scale Experimentation

Mark D. Corner and Brian N. Levine
College of Information and Computer Sciences
University of Massachusetts Amherst, MA, USA
mcorner,brian@cs.umass.edu

ABSTRACT

Mobile systems researchers struggle with conducting experiments with real users: either the scale of the study lacks sufficient scale and diversity, or a great effort must be used to recruit and manage subjects. In this paper, we describe MicroMobile, a system for deploying short data-gathering experiments to an extremely diverse set of users via mobile advertising. We conduct experiments in three mediums: interactive advertisements, mobile browsers, and native applications on both major mobile operating systems.

We use MicroMobile to demonstrate how researchers can use mobile advertising to recruit users, for as little as $1.50 per completed experiment. Across almost 500 completed experiments, we found that interactive ads have the highest participation rate (and thus lowest cost), which was 2x the participation rate of browser experiments and more than 6x native app experiments. Users were also highly diverse, spanning age, income, and ethnicity. While native apps are the most powerful platform, they constitute the most expensive targets. However, as mobile browsers add sensor APIs, browser-based experimentation has increasing applicability.

CCS CONCEPTS

• **Networks → Network measurement**; **Mobile networks**;

KEYWORDS

Mobile measurement;Mobile advertising

ACM Reference Format:
Mark D. Corner and Brian N. Levine. 2018. MicroMobile: Leveraging Mobile Advertising for Large-Scale Experimentation. In *Proceedings of The 16th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '18).* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3210240.3210326

1 INTRODUCTION

Mobile systems researchers struggle to gather data from sufficiently diverse users to make scientifically valid claims. Most research studies recruit test subjects locally, which heavily biases results towards educated, affluent, and racial- and gender-homogenous populations. Too often the test subjects are computer science students recruited from classes or labs. For example, a recent paper from MobiSys 2017 had two user studies: one with 6 participants from the lab, and another with 21 student participants (likely between 18 and 30), and only 33% were female [61]. Instead, systems researchers should gather data and results from the widest audiences and with as little selection bias as possible.

There are many options for recruiting users, including posting on Internet forums or using a paid source of subjects such as Amazon's Mechanical Turk. Internet forums will only reach a narrow audience and will produce unpredictable numbers of participants at uncontrolled times. Mechanical Turk is more predictable in reaching users, however one study showed that of 291 Mechanical Turk users, the median user had completed 300 academic studies and another demonstrates the subject pool is primarily frequent users [55], rather than typical users. And in both cases the users are not *in situ*, meaning they are not necessarily using mobile devices when being recruited which is critical for some tasks.

Adding to the challenge, systems researchers must build the experimental platform from scratch. Experiments are often built as native applications, as that *medium* provides the most capabilities, such as access to sensors and expanded APIs. This may be acceptable when manually recruiting users for a study, but getting large numbers of users to download and install a native app for a short experiment adds *friction* that limits participation.

Contributions. We propose and investigate the performance of a system, called MicroMobile, that combines the functions of participant recruitment with a multi-medium, experiment platform. MicroMobile leverages mobile advertisements to recruit and engage with participants, leading them to experiments conducted in three mediums: *(i)* interactive advertisements, *(ii)* the mobile web, and *(iii)* native mobile applications. The second two mediums are well-known, but for the first, MicroMobile uses *playable ads*. Such ads are used by industry to present an in-ad mini-game to entice users into downloading a native app. This medium provides us with an opportunity to deploy a subset of mobile experiments inside of an ad without leaving the current app. By lowering the perceived barrier to participation, we show that we increase participation and lower experimental costs. And in contrast to past works, e.g., Advertising as a Platform [27] and AdTag [22], our experiments are *active*, explicitly obtaining the consent and participation of users.

We built MicroMobile as a flexible system usable by other experimentalists. Experiments are written in JavaScript, and the same code is deployed to all three mediums, including interactive advertisements. Further, MicroMobile allows state transfer across mediums, so one can compose experiments from components running in ads, browsers, or native apps. As we show in this paper, it is advantageous to engage with users in an environment with the

MobiSys '18, June 10–15, 2018, Munich, Germany

ACM ISBN 978-1-4503-5720-3/18/06…$15.00
https://doi.org/10.1145/3210240.3210326

least friction (i.e., interactive ads), before asking them to complete other steps in a more powerful medium, such as a full browser.

We used MicroMobile as a platform to quantify what factors influence participation in mobile experiments. In particular we study the influences of these factors: *(i)* in-ad versus web versus app; *(ii)* the interactivity level of the experiment; *(iii)* engaging in low-friction mediums; *(iv)* prior participation in the system; *(v)* Institutional Review Board consent requirements; *(vi)* market place dynamics between iOS and Android; and *(vii)* prior brand awareness. We also examine the demographics of the users recruited in the study. For increased clarity, we have chosen to focus on unique aspects of MicroMobile: advertising to users with no-prior relationship to the experiment, and leveraging the sensors accessible from mobile web pages and native applications. In this light, we have chosen not to study the influence of monetary incentives.

Our system has been deployed in a very large scale advertising network, on the web, and in both the iOS and Android app stores. Employing this system across more than six hundred thousand advertisements, almost 500 completed experiments, and US$1,739 in advertising spend, we have found a number of results. As hypothesized, the less friction in the medium, the higher the participation rate and the lower the required advertising spend. Results can be obtained from consenting users for as little as $1.50 in advertising spend. Experiments done in playable ads have $2x$ the participation rate as those in a browser, and more than $6x$ those in native apps. Users were just as willing to participate in experiments that required interactivity, such as typing, as those that only required passive measurement, such as battery level. Having users engage in a warm-up experiment made them twice as likely to complete experiments requiring extra privacy permissions, though they were very unlikely to participate in experiments that required a photograph. Exploiting prior participation in experiments led to greater participation rates, though perhaps at similar cost, and exploiting a known brand in the ad led to $2x$ the participation rate, albeit at lower scale.

Part of our contribution is ensuring that our results and system are both reproducible. We illuminate many opaque details about these systems, and we have released the source code to the system with the publication of this paper [26].

2 ADS, BROWSERS, AND APPS

Our goal is to build data gathering systems that operate *in situ*: directly on mobile devices while they are in use, with as broad a participation as possible, for reasonable levels of cost. There are many ways to recruit subjects, but the largest and most-diverse set of participants can be reached using mobile advertisements. Ads reach any user of a mobile device that use mobile apps and websites—an extremely high proportion of users.

Digital advertising is a game of large numbers and low probabilities: click-through and experiment completion rates are small. Thus, to obtain a significant number of experimental results, a very large number of ad impressions must be purchased. For instance, if the click through rate is 2% and the experiment completion rate is 2%, then 250,000 ad impressions will yield 100 results. Fortunately, each impression is relatively inexpensive, ranging from US$0.10 to US$20.00 per 1,000 impressions (called a Cost Per Mille, or *CPM*).

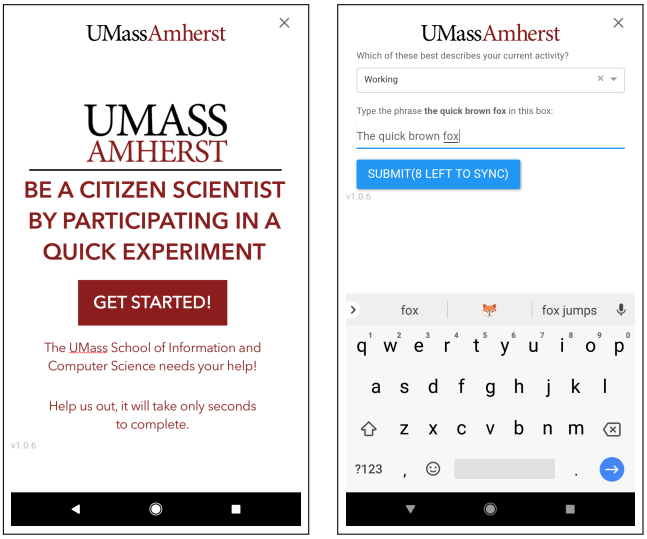

Figure 1: (Left) An example of a full-screen advertisement used to recruit users (some regions blacked out for double-blind review). (Right) An example screen from the typing experiment showing the continuous syncing of sensor values.

The cost of an impression is partially determined by the spend rate. With many commodities, the more one buys, the less expensive each unit is. With advertising auctions, the opposite is true; higher bids are required to outbid more and more bidders. Thus, if we bid more for each impression, each participating user will cost more to acquire, but results can be obtained more quickly. Optimizing bids is a deep subject [25, 30, 38, 40, 45, 68] and beyond the scope of this paper; any existing approach is compatible with MicroMobile. For the experiments we conducted for this paper, we set bids to a reasonable value (typically a $2.50 CPM), which allowed us to spend our daily budgets (as much as $60 dollars per day per experiment).

Once a user sees one of our advertisements (Figure 1(Left) shows an example) and clicks to participate, we have a choice of where to conduct the experiment: continue in the ad itself, move to a mobile browser (e.g., Chrome or Mobile Safari), or go to the app store to install a native mobile application. Below, we examine each of these three mediums.

2.1 Advertisements

Advertisements run in a *container* (an embedded WebView or iFrame) that possesses much of the capabilities of a full web page in a browser. Mobile advertising companies have taken advantage of this by employing *playable* ads—most commonly a small JavaScript game to gain a user's interest in the full, native app version of the game. Interactive ads run in *interstitial* ad slots that use the entire screen and do not require a context switch out of the app. Other ad formats are certainly possible, such as smaller ads that expand into larger ones when clicked–we have not explored that possibility here. Playable ads have proven to be highly effective, and therefore valuable, and ad space supporting these ads can be purchased through Real-Time Bidding (RTB) systems. Using this power, we can create experiments *inside* of advertisements. These

Sensor	Ad Android	Ad iOS	Chrome Browser	Safari Browser	Android App	iOS App
Touch	YES	YES	YES	YES	YES	YES
Battery State	YES	NO	YES	NO	YES	YES
Motion(Acc/Gyro)	YES:60Hz	YES:60Hz	YES:60Hz	YES:60Hz	YES:100Hz	YES:100Hz
Orientation	YES	YES	YES	YES	YES	YES
AmbientLight	NO/Generic Sensor	NO	NO/Generic Sensor	NO	YES	NO
Images/Video	NO	NO	YES*	YES*	YES*	YES*
Camera/Audio	NO	NO	YES*	YES* (iOS11)	YES*	YES*
Bluetooth	NO	NO	YES*	NO*	YES	YES
Location	HIGH/LOW**	HIGH/LOW**	YES*	YES*	YES	YES*
Proximity	NO/GenericSensor	NO	NO/Generic Sensor	NO	YES	YES

Table 1: The availability of the most common sensors in each medium. Star (*) denotes an extra permission dialog required. Generally any sensor that is available in the web is available inside an advertisement, as long as it doesn't create system dialog boxes (such as to ask for permission). The Generic Sensor API [3] is expected to bring access to more sensors in Chrome on Android, and as of writing, Ambient Light, DeviceMotion, and many others are expected to be available without user prompt [4]. The Absolute Orientation sensor on Safari is available via webkitCompassHeading. Bluetooth scanning on Chrome is limited to asking a user to connect to a specific device, not general scanning, though that is planned [2]. Double Star (**) denotes that in ads precise location is sometimes available and sometimes it is IP2Geo [27].

experiments are interactive, allowing a user to select from drop downs, type responses, submit results, etc. We can also take advantage of the psychological underpinning of playable ads, which is one of *engagement*. The advantage of playables is that they immediately engage the user and that engagement "hooks" a user. In this paper, we demonstrate that engagement can increase a user's willingness to take part in experiments.

2.2 Browsers

An advertisement can also lead to the operating systems' built-in browser, typically either Google Chrome or Apple Mobile Safari. Mobile browsers continue to lag native applications in functionality, but the trend is to bring increasing numbers of APIs to browsers. For instance, Google Chrome has added access to Bluetooth devices, battery state information, WebRTC, background workers, geolocation, push notifications, and the camera (not just photos and videos). Mobile Safari has lagged Chrome, but is now beginning to add access to advanced functionality like webworkers in iOS11. Browsers have the advantage of being very fast to load an experiment, with nothing to install. However, moving from an advertisement to a browser involves an animated context switch. That context switch is off-putting to some users, which is a factor we examine in our evaluation.

2.3 Native Applications

Native applications are the most powerful medium, as they have access to a superset of the capabilities of browsers. However, apps have the disadvantage of being *heavy* to download and install, which may dissuade users from participating—a user has to visit the app store, click, wait for the app to install, and then open it. Users may also feel a sense of permanence or intrusiveness in a native application, dissuading them from participation. Part of our goal was to examine this perception and as we show in the evaluation, convincing users to download and participate via native app was more difficult and resulted in lower participation rates. Native applications also carry disadvantages for the researcher: they are slower to iterate due to an approval process and must conform to rules set forth by the app store vendors. Deploying to an advertisement or the web is immediate.

2.4 Sensors

A key determinate in choosing the medium for an experiment is which sensors are available or require extra permissions. We examined the current implementation of sensors available through ads, browsers and native applications and provide a summary in Table 1. Access to certain sensors is unprivileged, such as the accelerometer/gyrometer, touch, and the battery state information. However, advertisements cannot trigger system dialogs, so they cannot access sensors that require extra permissions, such as the camera. Browsers can additionally access sensors that require permission from the user, however sensor fidelity may be limited. For instance, based on privacy concerns [5, 9], browsers and advertisements are limited to gathering accelerometer readings at 60 Hz, while native applications can gather data at the full 100 Hz. An upcoming implementation in Chrome of the W3C Generic Sensor API will bring even greater numbers of sensors to browser platforms [3], and will do so retaining the "permissionless" model [4]. All of the mediums can measure anything available through generic JavaScript, such as network bandwidth [27]. Also all mediums can interact with the user: the experiment can ask for data not available from sensors, such as those about themselves, context, and future plans.

Each of these mediums can support a broad array of mobile experiments. Examples in the advertising domain include gathering battery levels [18], measure the interaction of keyboards and accelerometers [44], and experiments that ask users to input a password pattern to determine its uniqueness [42]. Advertisements can be used to conduct surveys, such as the one found in the Heimdall recommendation system [52], and a reduced version of the actual system could be evaluated in an advertisement. Browser experiments can include those that use the camera, including image-based [17] or sound [24] authentication. Native apps are required for deeper experiments requiring APIs not available in a browser, but experiments that leverage Bluetooth[14] should be supportable in the browser in the near future[2]. Clearly MicroMobile cannot support experiments that depend on specialized, external hardware, nor cannot it support multi-person, or multi-device experiments easily. But what we show in this paper, *engaging* users in simpler experiments can help with recruitment for more complex ones.

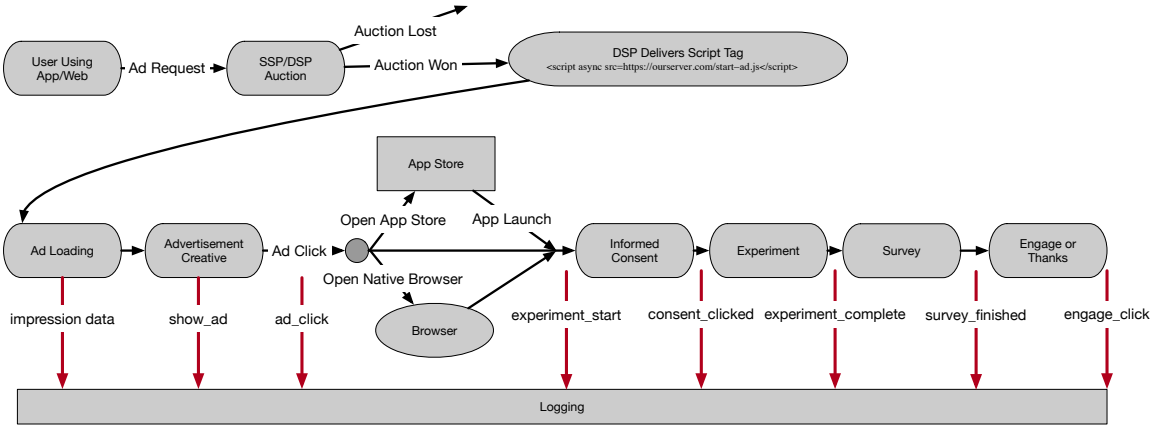

Figure 2: A state diagram of the MicroMobile system.

3 MICROMOBILE SYSTEM

MicroMobile is a system to allow easy design and execution of short data-gathering and crowd-sensing experiments. Starting with a digital advertisement, we invite users to participate in an experiment that executes partially in the advertisement, and continues inside of the three mediums: the advertisement itself, a browser, or a mobile app. The framework allows experimenters to spend less time on coding the common parts of an experiment, such as informed consent and submitting sensor readings, and enables smooth state transitions between the advertisement and other mediums.

The key challenge in building MicroMobile is minimizing the cost of participation. To address this challenge we build experiments that can run across mediums, while providing *flexibility* and *consistency*. Wherever possible, we want experiments to run inside all three mediums while maintaining a consistent styling. This is important for systems builders to simplify implementation and it also permits us to do comparisons across mediums without usability differences contaminating the results. But by supporting easy transitions across mediums within a single experiment, we can substantially lower costs by engaging with users in low friction mediums and transferring them to another medium.

The typical steps executed in a MicroMobile experiment are shown in Figure 2, including where logging to a central server occurs. First, we setup an advertising campaign with a Demand Side Platform (DSP) that bids on our behalf to display ads to users. When a user runs a mobile application or uses a mobile website, their device contacts a Real-Time Bidding (RTB) Supply Side Platform (SSP), which contacts the DSP for bids. A full explanation of RTB advertising systems can be found elsewhere [27, 64]. If we win the auction, the DSP returns a JavaScript tag to the SSP who displays it to the user (an example is shown in Figure 3), the device fetches JavaScript from our system, and the MicroMobile system begins to execute. MicroMobile is only compatible with ad systems that support JavaScript tags, which excludes closed networks like Facebook. It should be compatible with Google's SSP, though our self-service DSP did not have support for Google.

```
<ins data-track-impression="${imp_id}"
data-track-click_url="${click_url}"
data-track-device_identifier="${device_identifier}"
data-track-device_isp="${device_isp}"
data-track-device_model="${device_model}"
data-track-device_os="${device_os}"
data-track-os_version="${os_version}"
data-track-ip_add="${ip_add}"
data-track-source_id="${source_id}"
data-track-gps="${gps}"
data-track-user_agent="${user_agent}"</ins>
<script async src=https://ourserver.com/start-ad.js</script>
```

Figure 3: Advertising Tag. This shows some of the macros, the rest are omitted for brevity.

3.1 Ad Display and Logging

The JavaScript loaded from our server loads the rest of the ad by fetching an initial HTML file, which contains links to the css and remaining JavaScript code for informed consent, experiment, sensor libraries, and logging. All of this code executes inside of the advertisement when the ad is displayed. In optimized ad environments found in mobile apps, loading executes before the user is shown the ad [12]. This minimizes the latency a user experiences when the ad is finally displayed, increasing effectiveness. MicroMobile detects the preloading process using functions of *MRAID* [10]. MRAID is a library that provides a limited link between JavaScript and native functions in the mobile app. In this case, we use MRAID to detect when the ad is really being displayed, through the `mraid.isViewable()` function and `viewableChange` events. We record a display of the ad only when this occurs. If we win the auction, and the ad is preloaded, but not displayed, we record the impression, but not the showing of the ad.

The ad tag we give to the DSP, shown in Figure 3, includes an `ins` tag containing a set of macros, such as `${imp_id}`, a unique impression identifier. These macros are filled in by the DSP with values from the bid request before the ad markup is sent to the end device. Our ad loader script unmarshalls this data into a JavaScript object, and the rest of the experiment has access to those values. For instance, we use the impression id throughout the experiment to tie all subsequent actions, including app install and survey results to the original impression that found the user.

Once loaded, we try and entice the user to participate in the experiment by showing a full screen image, called a *creative*, as depicted in Figure 1(Left). Many designs for a creative are possible, including misleading or overly hyperbolic language. However, we have chosen to use a fairly neutral ad, though with different designs we could possibly achieve even better results. We employ a popular technique in interstitial advertisements, a delayed close button that is not displayed for 5 seconds—this delay is implemented via the `mraid.useCustomClose()` function and our own display of a close button. The delay encourages users to look at the advertisement without instinctively closing it.

If the user clicks on the ad, the system decides where that click leads. This choice is based on the requirements of the experiment. If the experiment requires a mobile app, then it will only send the user to the app store to install the app. However, many of our experiments can run in any of the three mediums.

In our evaluation, we randomly choose one of the three with equal probability to examine the effectiveness of each. If the system chooses to stay within the ad, it continues to showing an IRB consent form. If the choice is mobile web, we open the browser with the IRB form. Great care must be taken as using `window.open` or `location.href` in ads appearing in apps, as the link will open the page inside of the same embedded WebView the ad is displayed in. This is critical as the capabilities of an experiment running in the ad are very different than those running in a full mobile browser (see Table 1). Instead, we target ads to a particular SSP (MoPub [11]) that will open links that start with the scheme `mopubnativebrowser://navigate?url=` in the full native browser.

MicroMobile also records each device's unique *advertising identifier*. On iOS this value is called the Identifier for Advertising [36], and on Android it is the Google Advertising Identifier [37]. Both are random, anonymous, UUIDs. This identifier helps the DSP limit our ads to being shown to a device only once, and we can ensure that our experiments are at least done by users on unique devices (called *impression capping*). As we do not know the identity of users, only devices, we cannot ensure that the experiments are always done by unique people. This identifier also allows us to *retarget* individuals that complete our experiments—we examine this possibility in the evaluation. For individuals with *limit ad tracking* turned on [36, 37], we do not show ads as we cannot ensure impression capping, nor can we retarget ads.

3.2 State Transitions

Once the ad has loaded, the system generally follows the series of states shown in Figure 2. Each state is implemented as its own component. Our platform provides a separate experiment container component that controls *state transitions* and data flow between states. This process is similar to flow-based programming, or coordination languages [31, 46, 57] that control the flow of data through a series of black boxes.

The container component enables state transitions *across mediums*: an experiment can start in an advertisement, engage with the user, and then transition to another part of the experiment in another medium. As we show in the evaluation, for experiments that require a mobile browser or native app, it is advantageous to have the subject complete part of the experiment in the ad and *then* transition to complete it. This is essentially what playable advertisements do, they delay a context switch to the app store until after the user has engaged with the game in the ad.

When transferring from the ad to the mobile web, we can transfer state via link query parameters. The system parses those parameters and starts the web-based version of MicroMobile in that state. Transferring state to a native app is more complex, but we can employ the concept of *deferred deep linking* [8]. In deferred deep linking state can be transferred to an app that isn't installed yet by matching the device from the ad to a device running the app shortly afterwards. We use an implementation of deferred deep linking from Branch.io [7].

MicroMobile provides an experiment container that controls state transfers and provides common elements, such as informed consent. This yields a low implementation effort for new experiments. For instance, our implementation of a virtual keyboard accelerometer trace [44] is only 154 lines of JavaScript. Note that access to sensors is abstracted such that the same code runs in all three mediums without knowing which medium it is using. A reasonable programmer can build experiments in a few hours and deploy them to millions of users.

3.3 Sensor Abstraction

For continuous sensors, such as the accelerometer and gyrometer, we faced two challenges. *(i)* Experiments will often start gathering sensor information before some event occurs, such as pressing a key, but the notification of those events in a JavaScript environment happens too long after the event has occurred to start recording the sensor. *(ii)* We may want to gather relatively large amounts of sensor information from users, but also ensure that the user does not leave the experiment before that data gets sent to our server out of frustration, or believing the experiment is over.

To address challenge *(i)* we have developed a sensor abstraction in JavaScript that records sensor readings to a circular buffer at experiment start. Consider the case of taking an accelerometer reading when a user presses a key; e.g., Miluzzo et al. [44] requires such continuous readings of the accelerometer. Using the circular buffer, when an event of interest occurs, such as keyDown, we save sensor readings from the buffer that occurred before the event. When the experiment wants to stop recording, such as a keyUp event, we continue recoding for a period of time and save those readings as well. To make this general we abstract this buffer to handle any sensor with continuous sampling.

To address challenge *(ii)*, we built an abstraction to continuously stream sensor readings to the server while the experiment is running. If the user completes the experiment before the system is done syncing results, we provide a visual indication of how much data is left to encourage them to allow it to complete (see Figure 1(Right)).

3.4 Implementation and Deployment

To ensure that the advertisement, web browser, and native applications all work as similarly as possible, we have built MicroMobile using a unified framework and compile it into all three environments. We use Apache Cordova [16] as the basis of the system, which provides compilation of JavaScript and HTML code into

binaries that run on Android and iOS. Cordova also provides compilation to web environments, but it is typically treated as a fast debugging method, rather than a real target for compilation. We made modifications to several of the plugins that we use, such as the accelerometer and gyroscope plugins to properly support ads and the mobile web. All of the experiments were developed in JavaScript and HTML in the React framework [13]. The experiments we developed comprise 50–175 lines of JavaScript and React's jsx view language.

For ads and the browser, the app compiles to a small number of files, specifically a single HTML file that loads a JavaScript file containing the rest of the system (sensor libraries, user interface, data logging calls, etc.). This allows us to load the entire system at once and users experience instantaneous transitions through the systems' states. We took great care in minimizing the size and responsiveness of the system. We eliminated unnecessary libraries, or libraries that had many dependencies that would increase the size of the system. After minifying, the JavaScript library is only 450kB, which gzip compresses to 125kB, something that is quickly loaded over WiFi or cellular connections.

We deployed the system through a self-service RTB system called *PocketMath*. We manually set up campaigns in PocketMath, though the API could be used to automate even more sophisticated experiments.

We also deployed MicroMobile to the web using AWS' s3 and CloudFront CDN, which provides the distribution for both the advertising and mobile browser mediums. We also deployed the MicroMobile native app to both the Google Play and Apple AppStore.

We developed a backend system in NodeJS to receive impression data, logging events that occur at each stage, and experimental results, such as accelerometer readings and photos. We built this system on top of AWS' Lambda serverless framework to ensure scalability. We have released the source code to the system [26], including the front-end Cordova system, the backend AWS lambda system, and all of the plugins.

Porting an experiment to MicroMobile should be straightforward for anyone experienced in web programming. The most difficult part is to build appropriate plugins for Cordova to support sensors that aren't currently supported by existing plugins. But once the plugins exist, the rest of the code works across all three mediums automatically.

4 EVALUATION

Our evaluation of MicroMobile is focused on its efficiency in gathering data from participants, specifically: *(i)* the *participation rate*, which is the percentage of impressions that result in a completed experiment; and *(ii)* the *cost per completion*, which is the cost per impression multiplied by the participation rate. Our goal is to provide quantified guidance regarding how various factors affect the scale and cost of an experiment. For example, obtaining consent, requiring interactive participation, or requiring installation of a mobile app, are all factors that increase cost. Although we present exact cost numbers, these statistics are a snapshot in time: different ad outlets will vary in supply and cost over time—however, we expect the relative performance of factors to hold as prices vary with time. Also note that ad auctions are *second-price* auctions [65], so we pay less than $2.50 per thousand impressions when bidding a $2.50 CPM.

Campaign Name	Platform	CPM Bid	Impressions	Spend (USD)
Battery	Android	2.50	145,498	299.94
Typing	iOS+Android	2.50	156,088	299.88
Location	Android	2.50	105,979	225.39
Photo	iOS+Android	2.50	53,599	92.91
Battery-State	Android	2.50	17,301	43.83
Typing iOS/2.50	iOS	2.50	72,237	102.35
Typing iOS/5.0	iOS	5.00	89,272	209.29
Typing-Retarget	Android	10.00/30.00	28,409	231.59
Battery (Ad) to Location	Android	2.50	106,587	226.80

Table 2: Overview of the campaign parameters and spend.

We developed four experiments for our evaluation, listed below, that embody how we envision MicroMobile would be deployed by researchers. We intentionally designed these experiments to be as neutral as possible. For instance they have no obvious inherent social good, beyond "citizen science"; they do not claim to help scientists discover alien life [15] or cure diseases [39]. The experiments also do not offer any remuneration. We note that our evaluations do not consider issues related to fraudulent entry of information by participants. We also did not examine the users' experience with MicroMobile. With additional motivation and incentives, we expect all participation rates would improve. Each experiment collected real results, e.g., the battery levels of participants and accelerometer readings. However, we are not concerned with the actual experiment results, and instead focus on the factors that influence participation.

(1) *Battery* gathers the state of the devices' battery (percentage full and whether it is plugged in). This experiment is designed to be a *passive* exercise, where the user is asked to do very little, other than consenting to submission of impersonal device data. This experiment is Android only, as Mobile Safari lacks battery state support.
(2) *Typing* asks the user to select their current activity (commuting, working, studying, etc.) from a drop down, and then type a small phrase ("the quick brown fox") into a dialog box. The experiment measures the accelerometer and gyrometer during each key press, similar to studies of virtual keyboards and device motion [44]. This experiment is designed to be *active*, but still mostly impersonal in nature. An example page from this experiment is shown in Figure 1(Right). (For Android and iOS devices.)
(3) *Location* asks the user to provide their current location via the device's location API. This experiment is designed to ask for data that is protected and requires a system dialog box to ask for permission. Unlike the Battery and Typing experiments, to obtain permission, the user must be first redirected to a browser or app. (For Android and iOS devices.)
(4) *Photo* asks the user about their current activity, and then asks them to take a photo that characterizes their current environment. This experiment is designed to push the boundary of what personal information a user may want to submit through such a system. Similar to Location, to obtain permission to use the camera, the user must be first redirected to a browser or app. (For Android and iOS devices.)

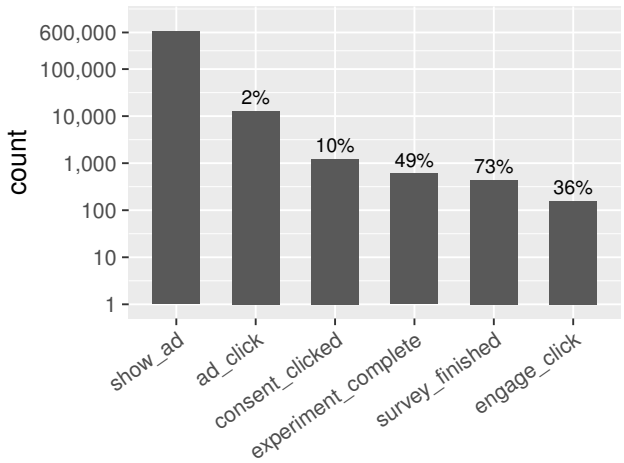

Figure 4: This funnel shows the completion percentage at each state as a percentage of the previous state. Log-scale.

We setup several advertising campaigns, shown in Table 2, yielding more than 620,000 ad impressions, for approximately $1,739 over a one month period. From that spend, 496 impressions resulted in at least one completed experiment. Some users elected to complete more than one experiment, resulting in 550 total completed experiments. These results include spend on less efficient mediums, like native apps. In short, if we had run only our most efficient experiments ($1.50 per completion), $1,739 would have yielded 1,159 results.

The default flow for each experiment is shown in Figure 2 and is as follows: *(i)* show the ad from Figure 1; *(ii)* if the user clicks on the ad, send the user to one of three destinations with equal probability (continue in ad, open the browser, send the user to the app store or open the app if they have it installed already); *(iii)* ask for informed consent; *(iv)* perform the experiment; *(v)* perform the survey; and *(vi)* ask the user to continue the experiment using the app (presuming they are not using the app already), which we call an *engagement click*. There are a few exceptions to this flow: the Location and Photo experiments only work in the browser and app so clicks only target those; and the Battery-to-Location campaign works differently, as explained in Section 4.3.

All of the campaigns used 320x480, portrait-mode, interstitial ads that cover the entire screen. Ads were targeted only in the United States to mobile app inventory to provide consistency in results. We did not do any optimization of our ad spend, such as targeting publishers or devices with high conversion rates, etc., such optimizations are orthogonal to our work; with optimization, costs could be considerably lower.

Funnel. To give a general idea of users' flow through this *funnel*, we show these stages in Figure 4 for all of the campaigns and mediums in aggregate. The graph is in log-scale, showing the greatest fall-offs in participation are from the initial advertisement to the user clicking (more than an order of magnitude), and from the user clicking on the ad to clicking on the consent. However, once a user has clicked on the human subjects consent form, the fall off is far less. This shows that the greatest opportunities for increasing

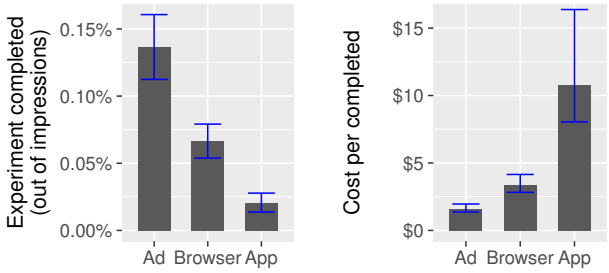

Figure 5: (§4.1, Ads vs. Browsers vs. Apps) Participation rate and cost per completion, aggregated across all four experiments. The error bars in all plots in this section represent a 95% confidence interval.

participation rates (and thus lowering cost) is by increasing the attractiveness of the creative and improving the consent form. Two factors make the second point more difficult than it might appear: *(i)* the transition between the ad click and clicking on the consent form sometimes includes starting the browser or installing the app, and *(ii)* mobile advertisements are notorious for high mis-click rates due to the limitations of the user interface [60]. It isn't possible to tell which clicks are mis-clicks, however as we show in the next section, the transition from click to browser or app has a profound effect on participation rates.

4.1 Ads vs. Browsers vs. Apps

Many experiments can be run in all mediums, but some are limited to particular ones. Our hypothesis is that participation increases as the friction of the medium and experiment decreases. The results of aggregating the Typing, Battery, Photo, and Location (only web and app), campaigns is shown in Figure 5.

The results confirm the hypothesis, showing that the participation rates (i.e., experiments completed out of all impressions shown) is $2x$ higher for interactive advertisements than browser, and $6x$ higher for ads than native applications. As we bid in a uniform manner across impressions, the cost per completed experiment follows the inverse of the participation rates. The cost data shows that we can obtain results for experiments at $1.61 each for interactive ads and $3.36 each for browsers, on average. Costs for native apps are higher, at an average of $10.79. Placed in the context of what typical surveys cost, which is $7.00 [43], or the labor and opportunity cost of having researchers spend time gathering human subjects manually, we feel that this is an economically feasible approach.

4.2 Experiment Interaction

One probable influence on participation is the experiment itself. The Battery experiment is purely passive, the user only has to click submit; the Typing experiment requires the user to follow instructions and take an active role; the Location experiment is largely passive, but requires the user to give permission via a system dialog; and the Photo experiment requires permission and data that might be perceived to be more personal. A comparison of the participation rates and costs of these four experiments is shown in Figure 6 and is the aggregate of results across mediums.

The results show that the Battery and Typing experiments are very similar in participation rates and thus cost. The Location and

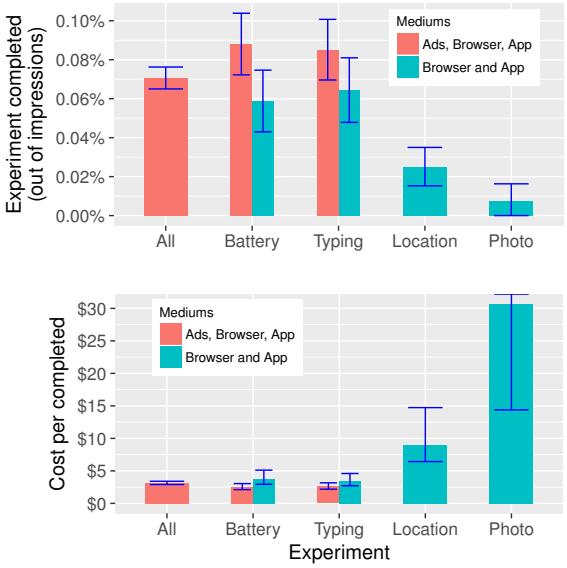

Figure 6: (§4.2, Experiment Interaction) This shows the participation rate and costs for each of the experiments aggregated across all three mediums. To provide a fair comparison to experiments (Photo and Location) that only run in browser and app, we also show results for Battery and Typing for just browser and app.

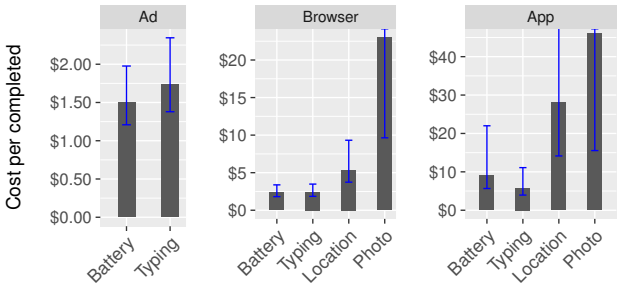

Figure 7: (§4.2, Experiment Interaction) The cost per completed experiment given a medium.

Photo experiments show far lower participation rates. However, this is partly the influence of mediums, as both can only be run in the browser and native app which inherently have lower rates.

To tease these factors apart, Figure 6 also shows a comparison of just the browser and app results from the Battery and Typing experiments versus location and photo. Here we still see a lower participation rate for Location, either due to the extra system dialog, or the perception of privacy implications. The Photo experiment has extremely low participation rates, with only a few positive data points. Given more context and motivation for submitting photos, users may be more willing to participate.

In Figure 7 we show the same data, but completely disaggregated. This shows that the cost of obtaining experimental results can be as low as $1.50 for experiments run in interactive advertisements. Costs for native app results are much higher, but as we show in Section 4.3, they can be lowered considerably through engagement.

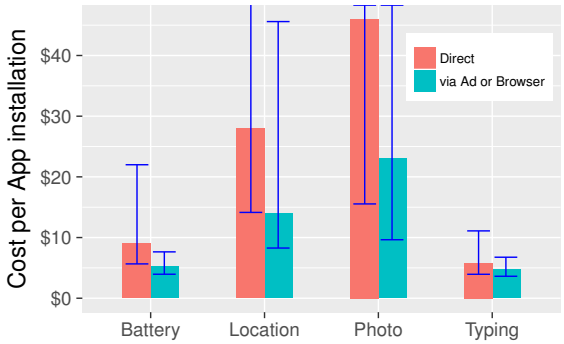

Figure 8: (§4.3, Engagement exp 1) This plot shows a comparison between having users complete ad and browser experiments before asking them to install an app, versus directly asking them to install an app as the first thing.

4.3 Engagement

One of the psychological advantages of interactive advertisements is that of of *engagement*: ask a user to do something easy (e.g., play a small game), before asking them to do something more difficult (e.g., download the full game)[47, 48]. We experimented with two methods to try and take advantage of this phenomena.

The first method seeks to improve participation in native-app based experiments. In all of the campaigns, after users complete an experiment in an interactive ad or a browser we ask them to repeat the experiment inside of a native app. If they click, which we call an *engagement click*, we take them to the app store. If they then install the app, it starts the same experiment over inside the app (via the state transfer mechanism described in Section 3.2). Overall 36% of users that finish an experiment click on the engagement offer (see Figure 4). We divide the number of native app completed experiments resulting from those engagement clicks by the full cost of the ad and browser experiments, yielding a cost to get a native experiment completion. We can compare that against the cost of a completed experiment where we simply send the user to the app store from an initial ad. The results for cost are shown in Figure 8.

Overall, the results show that it is less expensive in each case to get a native app result by first having a user do an experiment in-ad or in a browser. In some ways, this result is counterintuitive: the user is doing more work before getting to the native app and it still costs less. But it is the effect of *engagement*: start the user off in an easier environment before graduating them to the app. We do observe that the confidence intervals are overlapping, due to the relatively small participation rates: out of about 250,000 impressions, 33 experiments were completed by directly advertising native apps, and 90 via the engagement clicks). Rejecting the null hypothesis in this case would likely be expensive, but we plan to confirm this result with further experimentation.

However, we can bolster the case for engagement with a similar composition. We sought to improve response rates in the Location experiment by having the users complete a different experiment (Battery) in the advertisement, and then redirect them to the mobile web or app. We call this campaign "Battery to Location" and it can be directly compared to the "Location" campaign, which directly

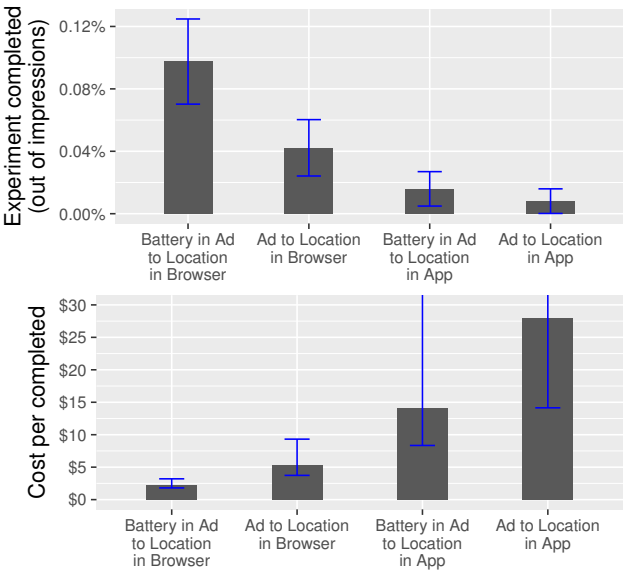

Figure 9: (§4.3, Engagement exp 2) This plot shows a comparison of having users complete the Battery experiment in an ad before sending them to the mobile web or native app to gather location information against sending them directly to the web or app. Having them complete Battery first increases participation by $2x$.

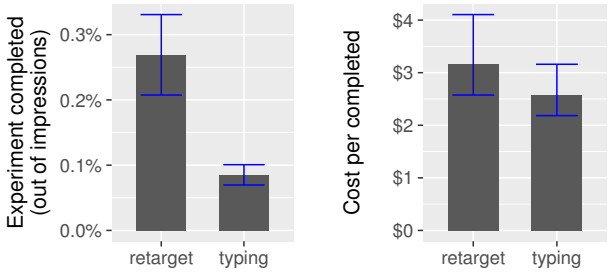

Figure 10: (§4.4, Retargeting) This shows the results of retargeting users that clicked on our ad. The participation rates are $3x$ higher, though due to bidding too high on the campaign, the costs are similar.

Medium	Consent Rate (% of impressions)	Cost per Consent (USD)	95% c.i.
w/o consent		0.0000053	± 0
Ad	0.30	0.72	± 0.03
Browser	0.13	1.65	± 0.11
App	0.02	9.00	± 1.38

Table 3: (§4.5, Cost of Consent) The cost of consent for the Battery experiment is cheap, at US$0.72 via a playable ad. But if consent is not required, the cost is several orders of magnitude lower.

leads to the mobile web or the native app. The results are shown in Figure 9.

The results show a roughly $2x$ increase the response rate (and less than half the cost per completed experiment) for the Location experiment if we have the users complete the Battery experiment first in the advertisement and then send them to the browser or mobile app to gather location information. This brings the cost of Location results in the browser in line with Battery and Typing experiments in browser (compare to Figure 7). Thus, even if the researcher doesn't need the results from the experiment performed in the ad, it is still advantageous to engage with the user before taking them to the mobile web or app and asking for location permission.

4.4 Retargeting

A popular technique in advertising is *retargeting* where an advertiser will show advertisements to users that have expressed some interest in a product before (visiting a web site, leaving something in a shopping cart, etc.). To see if that technique would be useful for MicroMobile, we ran a campaign that targeted the 8,800 specific devices that had previously clicked on one of our ads. The retargeting creative was slightly different, thanking them for their previous participation and asking them to participate again. While there are a great number of mis-clicks in mobile advertisements, the conjecture is that this will still increase the probability of user participation. We bid a high CPM ($30.00) to start, which is typical of retargeting campaigns since the probability of completion is high, and later lowered it to $10.00. The results of this experiment are shown in Figure 10.

The results demonstrate that the participation rate was considerably higher, at almost $3x$. However, as most of our other campaigns ran at a CPM of $2.50, the retargeting campaign resulted in a slightly higher cost (though we fail to reject the null hypothesis). This experiment is difficult to attempt multiple times as we have tainted the results of those users retargeted and must gather a new set at a large cost. However, given the high participation rate, we feel that with bidding optimization the results for retargeted users would be high. Assuming a linear relationship, the costs for retargeted campaigns will be $3x$ less in cost (though results will be gathered $3x$ more slowly).

This also provides evidence that longitudinal results could be gathered from users using MicroMobile. However, we have not fully explored this possibility. For instance, what works best for contacting subjects repeatedly? One can choose between email, SMS, retargeted advertisements, browser-based notifications, and native push messages. We leave a full exploration as future work.

4.5 Cost of Consent

Our own work, *Advertising as a Platform* (AaaP) [27] is the closest system from related work that we can compare against. AaaP also uses mobile advertisements to gather data, such as battery levels and bandwidth. However, AaaP operates only *passively*, gathering data without user consent and without interaction. As AaaP can gather data at the impression time, rather than depending on the click, we compare the cost of the impressions for the battery experiment vs the cost for a user consenting to participating and completing the experiment. The results are shown in Table 3.

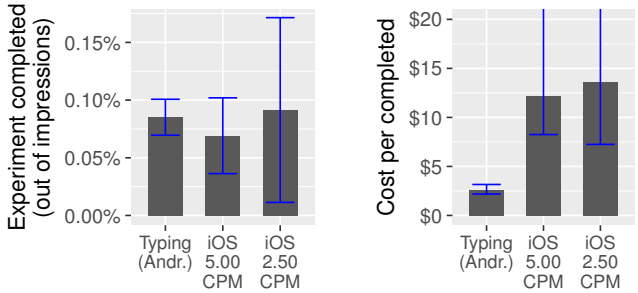

Figure 11: (§4.6, Android vs. iOS) The participation rate for the Typing experiment for just Android users vs. two campaigns targeting only iOS users.

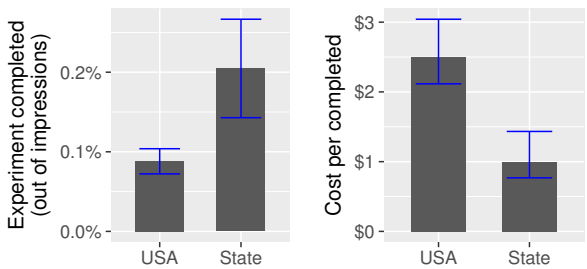

Figure 12: (§4.7, Branding) A comparison of a nationwide campaign vs one focus on the state surrounding our institution. A familiarity with a brand can increase participation rates.

We can quantify the *cost of consent*: what does it cost to get a user's consent for experiments? Getting consent is massively more expensive than not, by several orders of magnitude. However, consent opens a world of possibilities, including any of the interactive experiments, such as Typing, Photo, and Location.

4.6 Android vs. iOS

The experiments are built for, run on, and are targeted to Android and iOS. While running our experiments we noticed that the vast majority of impressions and results came from Android users, even though the market shares in the US are similar [6]. We found that the underlying reason is the auction process in RTB: advertisers bid higher for impressions shown to iOS users because they are perceived to have higher economic value [1]. As other bidders bid higher, the proportion of auctions we win at any CPM goes down relative to Android devices. To gather more data on iOS users, we set up two campaigns that only targeted iOS devices using the Typing experiment. One campaign used a $2.50 CPM and the other $5.00. The results are shown in Figure 11.

The results show that iOS users are almost just as likely to do an experiment, but since the ad traffic is more expensive, experimental results are $4x$ more expensive (though the error bars are somewhat large due to the smaller numbers, even at a $5.00 CPM). One might think that at a $5 CPM, the cost per completion should be $2x$ the results for Android traffic, which was a $2.50 CPM. But that's not the case, which demonstrates a property we found in RTB's second-price ad auctions: if the auction is more competitive (more bidders), it is more likely to win at a second price close to the bid price.

As long as the lack of iOS users does not skew an experiment in some way (such as demographics), this bias is actually a good one. Chrome is a more powerful browser (the embedded WebView and the full browser), giving links to more sensors (see Table 1) and more aggressively adding features of interest to mobile systems researchers.

4.7 Branding

Another possible way to increase participation rates is to use a sense of familiarity with a brand, which can engender trust in, or duty to the brand, and thus the experiment. To examine this we ran a campaign for the Battery experiment, *State*, targeted to Massachusetts. As the advertisement uses the branding of our institution, the conjecture is users would recognize the brand and be more likely to participate. The results are shown in Figure 12.

The results show that participation rates can be increased by more than $2x$, and cost is less than half. The downside is scale.

At a $2.50 CPM (the same as the full USA), we could spend between $1 and $2.50 per day, which results in approximately one result per day. In contrast the campaign targeted to the USA ran out of its $60 daily budget fairly early in the day. However, these are relative, so given a larger brand, a larger state, or more advertising outlets, the results could be scaled up. It would be interesting to try other brands, though that may prove difficult given the constraints of the IRB.

We also set up a campaign, *Local*, that targeted a 30km radius around the central point of our institution. In the Local experiment we had to increase the CPM to $10 in order to get enough impressions to be viable. Even at that level we could only spend between 10 and 25 cents per day, which translates to only a few completed experiments so we have omitted the results from the graph. The participation rates follow the same pattern as the State, though the costs are similar to USA due to the higher CPM.

4.8 Demographics

At the end of each experiment we also ask the users to fill out a short survey of their demographics (age, race/ethnicity, gender, education level, and income). We wanted to see if we are truly reaching a diverse set of users for experiments. The results of all of the surveys are shown in Figure 13. The results comprise 375 unique surveys.

This result shows the system is effective at reaching a diverse set of users. The demographic distribution does not exactly match that of the United States as a whole, but given that there is some data for each group, we can correct for biases in the data by weighting the results appropriately. Whether re-weighting is necessary will be dependent on the particular study.

5 RELATED WORK

Adverting-Based Measurements. Our previous work on Advertising as a Platform (AaaP) [27] is closely related. AaaP uses advertisements to collect measurements from large numbers of mobile devices. MicroMobile has a number of advantages and differences. A primary contribution of this paper over our previous work on

Figure 13: (§4.8, Demographics) A comparison of survey results from MicroMobile with statistics from the US Census. P.N.T.S. means "Prefer Not To Say".

AaaP is that we have quantified the economic cost of obtaining user consent. Consent is critical for experiments on the web or native apps, which are platforms that AaaP cannot support, or in cases where ads can be used but user consent is required by IRB or other concerns. A subsequent system, *AdTag* [22], used similar techniques for network measurement. In contrast, AaaP and AdTag are purely passive systems, collecting data when an advertisement appears. AaaP and AdTag neither interact with users nor do they use clicks on ads in any way; they are limited to collecting data from sensors that work inside of advertisements and cannot ask users questions (such as context, demographics etc.). The one advantage over MicroMobile is that of scale and cost—MicroMobile requires orders of magnitude more impressions to collect data, so if a researcher only needs passive data, then it is more efficient to use AaaP or AdTag. Overall, we see both tools as useful for different scenarios.

Mobile Surveys. Other relevant work comes from the survey practice research community. Primarily this work centers on how to adapt existing surveys that were developed for the web to mobile devices [20, 21, 28, 51, 58]. These works lie in a traditional approach of question-based surveys completed by pre-recruited panels of participants. A more general system from Google [32] allows publishers to embed surveys into their own sites and compensate users. These surveys are purely question based and do not leverage all of the additional sensors available in a mobile device.

The closest work from the survey community comes from the Pew Research Center [43] who studied the use of mobile devices to collect survey information. In contrast with our study, Pew started with a pre-recruited panel of participants gathered through random phone dialing. Participants were monetarily incentivized to take part ($5 to participate and $1 for each follow on study). This work is a translation of traditional survey methodology to a mobile device. Pew did find a higher level of users completing at least one survey in web browsers versus apps (84% vs. 58%). In contrast, we are: *(i)* forging a new type of data gathering inside of interactive advertisements; *(ii)* using the sensors in the mobile phone in combination with user-participation; *(iii)* recruiting subjects in-situ while using mobile devices vs those recruited via phone calls; and *(iv)* eschewing incentives in favor of users motivated by the experiment.

Mechanical Turk is another method to recruit subjects, though the users are often professional task workers, which means they won't be representative of real users [55]. Further they are not *in-situ*, meaning they are likely using a desktop, or are not using a mobile device in the ordinary course of their day.

Advertising. The vast majority of public research has concentrated on examining the privacy implications of mobile advertising [23, 33, 41, 50, 54, 56, 59]. Some work has looked at how to optimize the placement of ads, generally to increase their effectiveness, and decrease cost [25, 30, 38, 40, 45, 68]. In our work we are primarily concerned with the relative costs of various options (browser versus advertisement etc.) and we consider optimization to be largely orthogonal—our system should be able to gather more data given greater optimization. One might consider our results to be an upper bound on costs. Other work includes measuring what ads are shown to users and why [19, 29, 34, 49, 62, 63, 66].

Mobile Crowd Sensing. Our work is also applicable to applications in mobile crowd sensing. MicroMobile could be used to collect data for some kinds of crowd sensing systems, though it can also be used to conduct interactive experiments with mobile participants. Mobile crowd sensing has developed into a rich area [35]. One of the key elements that we examine in MicroMobile is *recruitment*, something typically ignored in crowd sensing systems. CrowdRecruiter [67] assumes there is already a set of willing participants and attempts to maximize sensing coverage while minimizing payments to users and energy consumption on devices. Similarly, Reddy et al. [53] examined how to distribute tasks to a set of already recruited participants, concentrating on data availability based on mobility patterns. We did not examine the effects of incentives in our work and instead concentrated on finding participants using social good as the motivating factor. MicroMobile works at the step that comes before CrowdRecruiter and Reddy et. al: where did the users come from in the first place, and what modalities (ad, browser, or app) are the easiest to recruit for?

6 HUMAN SUBJECTS

The work described in this paper was reviewed and approved by our Institutional Review Board, under protocols 2016-3112 and 2016-3141. Our protocol carefully presents information required for informed consent. Our ads include contact information, risks and benefits of the study, and an explanation of what data is collected

and how. As shown in Figure 4, the typical fall-off from clicking on the advertisement to clicking on the informed consent agreement is an order of magnitude. To encourage users to read the informed consent we designed it to fit on one page on most mobile devices. Our initial experiments used the informed consent as the advertisement, which yielded extremely low participation rates. This lead to our using a combination of an enticing ad first with informed consent second.

We have no evidence that users do or do not read the informed consent document. We have considered other presentations of the information, including a multi-page carousel, videos, or other more interactive experiences. We believe that more work here could yield greater participation rates while simultaneously increasing the users' understanding of the experiment.

7 CONCLUSIONS

MicroMobile provides a novel framework for deploying mobile experiments to a massively scalable and diverse subject pool. We have used MicroMobile to demonstrate the relative performance of experiments deployed to interactive advertisements, web browsers, and native applications. We found that researchers can gather results for as little as $1.50 per experiment when using playable ads. Experiments that require interactive participation from users are not much more expensive. Using engagement in the low-friction advertising environment helps user participation rates in the browser and native apps. These are encouraging results and we look forward to working with other researchers to deploying novel experiments at scale.

We would like to thank our shepherd, Yubin Xia, for his helpful comments.

REFERENCES

[1] Advertisers still prefer Apple users. https://digiday.com/media/advertisers-iphone-ios-users-valuable-android-ones/. (January 2014).
[2] Specify a Bluetooth Scanning API. https://github.com/WebBluetoothCG/web-bluetooth/pull/239. (May 2016).
[3] Generic Sensor API. https://www.w3.org/TR/generic-sensor/. (October 2017).
[4] Relax requirements for asking permissions in sensors. https://github.com/w3c/sensors/issues/174. (March 2017).
[5] Stealing sensitive browser data with the W3C Ambient Light Sensor API. https://blog.lukaszolejnik.com/stealing-sensitive-browser-data-with-the-w3c-ambient-light-sensor-api/. (April 2017).
[6] https://www.statista.com/statistics/266572/market-share-held-by-smartphone-platforms-in-the-united-states/. (2018).
[7] Branch. https://branch.io. (2018).
[8] Deferred Deep Linking. https://en.wikipedia.org/wiki/Deferred_deep_linking. (2018).
[9] Generic Sensor API. https://w3c.github.io/sensors/#main-privacy-security-threats. (February 2018).
[10] Mobile Rich Media Ad Interface Definitions (MRAID). https://www.iab.com/guidelines/mobile-rich-media-ad-interface-definitions-mraid/. (2018).
[11] MoPub. https://www.mopub.com/. (2018).
[12] Pre-Caching. https://www.mopub.com/resources/mopub-demand/pre-caching/. (February 2018).
[13] ReactJS. https://reactjs.org/. (2018).
[14] Ioannis Agadakos, Jason Polakis, and Georgios Portokalidis. 2017. Techu: Open and Privacy-Preserving Crowdsourced GPS for the Masses. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services.* ACM, 475–487.
[15] David P Anderson, Jeff Cobb, Eric Korpela, Matt Lebofsky, and Dan Werthimer. 2002. SETI@ home: an experiment in public-resource computing. *Commun. ACM* 45, 11 (2002), 56–61.
[16] Apache. Cordova. https://cordova.apache.org/. (2018).
[17] Mozhgan Azimpourkivi, Umut Topkara, and Bogdan Carbunar. 2017. Camera Based Two Factor Authentication Through Mobile and Wearable Devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 35.
[18] Nilanjan Banerjee, Ahmad Rahmati, Mark D Corner, Sami Rollins, and Lin Zhong. 2007. Users and Batteries: Interactions and Adaptive Energy Management in Mobile Systems. In *Conf. on Ubiquitous Computing (UbiComp).* Springer, Innsbruck, Austria, 217–234.
[19] Paul Barford, Igor Canadi, Darja Krushevskaja, Qiang Ma, and S Muthukrishnan. 2014. Adscape: Harvesting and analyzing online display ads. In *Proc. Conf. on World wide web.* ACM, 597–608.
[20] TD Buskirk, M Gaynor, C Andrus, and C Gorrell. 2011. An app a day could keep the doctor away: comparing mode effects for a iPhone survey related to health app use. *American Association of Public Opinion Research, Phoenix, AZ* (2011).
[21] Mario Callegaro. 2013. Do you know which device your respondent has used to take your online survey? *Survey Practice* 3, 6 (2013).
[22] Patricia Callejo, Conor Kelton, Narseo Vallina-Rodriguez, Rubén Cuevas, Oliver Gasser, Christian Kreibich, Florian Wohlfart, and Ángel Cuevas. 2017. Opportunities and Challenges of Ad-based Measurements from the Edge of the Network. In *Proceedings of the 16th ACM Workshop on Hot Topics in Networks.* ACM, 87–93.
[23] Claude Castelluccia, Mohamed-Ali Kaafar, and Minh-Dung Tran. 2012. Betrayed by your ads!. In *International Symposium on Privacy Enhancing Technologies Symposium.* Springer, 1–17.
[24] Jagmohan Chauhan, Yining Hu, Suranga Seneviratne, Archan Misra, Aruna Seneviratne, and Youngki Lee. 2017. BreathPrint: Breathing acoustics-based user authentication. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services.* ACM, 278–291.
[25] Ye Chen, Pavel Berkhin, Bo Anderson, and Nikhil R Devanur. 2011. Real-time bidding algorithms for performance-based display ad allocation. In *Proc. ACM SIGKDD Conf. on Knowledge discovery and data mining.* ACM, 1307–1315.
[26] Mark Corner. https://github.com/mcorner/micromobile. (2018).
[27] Mark D Corner, Brian N Levine, Omar Ismail, and Angela Upreti. 2017. Advertising-based Measurement: A Platform of 7 Billion Mobile Devices. In *Proc. ACM Conf. on Mobile Computing and Networking (MobiCom).* Snowbird, Utah, USA.
[28] MP Couper. 2010. Visual design in online surveys: Learnings for the mobile world. In *Mobile Research Conference.*
[29] Amit Datta, Michael Carl Tschantz, and Anupam Datta. 2015. Automated experiments on ad privacy settings. *Proceedings on Privacy Enhancing Technologies* 2015, 1 (2015), 92–112.
[30] Ayman Farahat and Michael C Bailey. 2012. How effective is targeted advertising?. In *Proc. Conf. on World Wide Web.* 111–120.
[31] David Gelernter and Nicholas Carriero. 1992. Coordination languages and their significance. *Commun. ACM* 35, 2 (1992), 96.
[32] Google. Opinion Rewards. https://surveys.google.com/google-opinion-rewards/. (2018).
[33] Michael C Grace, Wu Zhou, Xuxian Jiang, and Ahmad-Reza Sadeghi. 2012. Unsafe exposure analysis of mobile in-app advertisements. In *Proc. ACM conference on Security and Privacy in Wireless and Mobile Networks.* 101–112.
[34] Saikat Guha, Bin Cheng, and Paul Francis. 2010. Challenges in measuring online advertising systems. In *Proc. ACM Internet measurement conference.* 81–87.
[35] Bin Guo, Zhu Wang, Zhiwen Yu, Yu Wang, Neil Y Yen, Runhe Huang, and Xingshe Zhou. 2015. Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm. *ACM Computing Surveys (CSUR)* 48, 1 (2015), 7.
[36] Apple Inc. ASIdentifierManager. https://developer.apple.com/documentation/adsupport/asidentifiermanager. (????).
[37] Google Inc. AdvertisingIdClient. https://developers.google.com/android/reference/com/google/android/gms/ads/identifier/AdvertisingIdClient. (????).
[38] Azeem J Khan, Kasthuri Jayarajah, Dongsu Han, Archan Misra, Rajesh Balan, and Srinivasan Seshan. 2013. CAMEO: A middleware for mobile advertisement delivery. In *Proc. ACM Mobisys.* 125–138.
[39] Stefan M Larson, Christopher D Snow, Michael Shirts, and Vijay S Pande. 2009. Folding@ Home and Genome@ Home: Using distributed computing to tackle previously intractable problems in computational biology. *arXiv preprint arXiv:0901.0866* (2009).
[40] Kuang-chih Lee, Burkay Orten, Ali Dasdan, and Wentong Li. 2012. Estimating conversion rate in display advertising from past performance data. In *Proc. ACM SIGKDD Conf. on Knowledge discovery and data mining.* 768–776.
[41] Ilias Leontiadis, Christos Efstratiou, Marco Picone, and Cecilia Mascolo. 2012. Don't kill my ads!: balancing privacy in an ad-supported mobile application market. In *Proc. Workshop on Mobile Computing Systems & Applications.* 2.
[42] Can Liu, Gradeigh D Clark, and Janne Lindqvist. 2017. Guessing attacks on user-generated gesture passwords. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 1 (2017), 3.
[43] Kyley McGeeney and Rachel Weisel. 2015. *App vs. Web for Surveys of Smartphone Users.* Technical Report. Pew Resarch Center.

[44] Emiliano Miluzzo, Alexander Varshavsky, Suhrid Balakrishnan, and Romit Roy Choudhury. 2012. Tapprints: your finger taps have fingerprints. In *Proc. Conference on Mobile systems, applications, and services*. 323–336.

[45] Prashanth Mohan, Suman Nath, and Oriana Riva. 2013. Prefetching mobile ads: Can advertising systems afford it?. In *Proc. ACM European Conference on Computer Systems*. 267–280.

[46] J Paul Morrison. 2010. *Flow-Based Programming: A new approach to application development*. CreateSpace.

[47] Richard Moss. 2016. Interactive ads: The future of making money on mobile games? *Gamasutra* (July 8 2016). https://www.gamasutra.com/view/news/276329/Interactive_ads_The_future_of_making_money_on_mobile_games.php

[48] Elad Natason. 2016. Playable Mobile Ads — A Hot Trend or The Future? *Forbes* (Oct 18 2016). https://www.forbes.com/sites/eladnatanson/2016/10/18/playable-mobile-ads-a-hot-trend-or-the-future

[49] Suman Nath. 2015. Madscope: Characterizing mobile in-app targeted ads. In *Proc. ACM Mobisys*. ACM, 59–73.

[50] Paul Pearce, Adrienne Porter Felt, Gabriel Nunez, and David Wagner. 2012. Addroid: Privilege separation for applications and advertisers in android. In *Proc. ACM Symposium on Information, Computer and Communications Security*. 71–72.

[51] Andy Peytchev and Craig A Hill. 2010. Experiments in mobile web survey design: Similarities to other modes and unique considerations. *Social Science Computer Review* 28, 3 (2010), 319–335.

[52] Amir Rahmati, Earlence Fernandes, Kevin Eykholt, Xinheng Chen, and Atul Prakash. 2017. Heimdall: A Privacy-Respecting Implicit Preference Collection Framework. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 453–463.

[53] Sasank Reddy, Deborah Estrin, and Mani Srivastava. 2010. Recruitment framework for participatory sensing data collections. In *Conf. on Pervasive Computing*. Springer, 138–155.

[54] Franziska Roesner, Tadayoshi Kohno, and David Wetherall. 2012. Detecting and defending against third-party tracking on the web. In *Proc. USENIX conference on Networked Systems Design and Implementation*. 12.

[55] Joel Ross, Lilly Irani, M Silberman, Andrew Zaldivar, and Bill Tomlinson. 2010. Who are the crowdworkers?: shifting demographics in mechanical turk. In *Proc. ACM CHI extended abstracts on Human factors in computing systems*. 2863–2872.

[56] Shashi Shekhar, Michael Dietz, and Dan S Wallach. 2012. Adsplit: Separating smartphone advertising from applications. In *Proc. USENIX Security Symposium*. 553–567.

[57] Jacob Sorber, Alexander Kostadinov, Matthew Garber, Matthew Brennan, Mark D Corner, and Emery D Berger. 2007. Eon: A Language and Runtime System for Perpetual Systems. In *Proc. ACM Conf. on Embedded Networked Sensor Systems (SenSys)*. 161–174.

[58] Carey Stapleton. 2013. The smart (phone) way to collect survey data. *Survey Practice* 6, 2 (2013).

[59] Ryan Stevens, Clint Gibler, Jon Crussell, Jeremy Erickson, and Hao Chen. 2012. Investigating user privacy in android ad libraries. In *Workshop on Mobile Security Technologies (MoST)*. 10.

[60] Trademob. 40 Percent of Mobile Ad Clicks are Fraudulent or Accidental, According to New Study by Trademob. https://www.trademob.com/pressrelease/40-percent-of-mobile-ad-clicks-are-fraudulent-or-accidental-according-to-new-study-by-trademob/. (September 2012).

[61] Yu-Chih Tung and Kang G Shin. 2016. Expansion of human-phone interface by sensing structure-borne sound propagation. In *Proc. ACM Conf. on Mobile Systems, Applications, and Services*. 277–289.

[62] Imdad Ullah, Roksana Boreli, Mohamed Ali Kaafar, and Salil S Kanhere. 2014. Characterising user targeting for in-app mobile ads. In *IEEE Infocom Workshops*. 547–552.

[63] Narseo Vallina-Rodriguez, Jay Shah, Alessandro Finamore, Yan Grunenberger, Konstantina Papagiannaki, Hamed Haddadi, and Jon Crowcroft. 2012. Breaking for commercials: characterizing mobile advertising. In *Proc. ACM Internet Measurement Conference*. 343–356.

[64] Paul Vines, Franziska Roesner, and Tadayoshi Kohno. 2017. Exploring ADINT: Using Ad Targeting for Surveillance on a Budget—or—How Alice Can Buy Ads to Track Bob. In *Workshop on Privacy in the Electronic Society*.

[65] Wikipedia. Vickery Auction. https://en.wikipedia.org/wiki/Vickrey_auction. (????).

[66] Jun Yan, Ning Liu, Gang Wang, Wen Zhang, Yun Jiang, and Zheng Chen. 2009. How much can behavioral targeting help online advertising?. In *Proc. Conf. on World Wide Web*. ACM, 261–270.

[67] Daqing Zhang, Haoyi Xiong, Leye Wang, and Guanling Chen. 2014. CrowdRecruiter: selecting participants for piggyback crowdsensing under probabilistic coverage constraint. In *Proc. 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 703–714.

[68] Weinan Zhang, Shuai Yuan, and Jun Wang. 2014. Optimal real-time bidding for display advertising. In *Proc. ACM SIGKDD Conf. on Knowledge discovery and data mining*. 1077–1086.