Conducting speech perception experiments online: Some tools, successes, and challenges

To be honest, my past self would be shocked to know that I've become a proponent of online testing. Our team uses behavioral psychophysics tasks (among other tools) to examine how listeners derive meaning from speech acoustics. In our physical lab, all testing takes place in a sound-attenuated booth. Listeners wear high quality headphones. We hold the amplitude of stimuli constant across participants. Responses are made using specialized hardware to ensure accurate measurement of response times. We interact with each participant, allowing us to confirm that they understand task instructions and actually exist in human form. Why would we even consider abandoning these standards? Well, for us, there are a few reasons:

1. Many of our studies don't actually require this level of control over the auditory and response environment. We don't present auditory stimuli at threshold levels. Many of our tasks don't actually use reaction time as the dependent measure. I'd like to think that the things we study (and thus the things we claim) might actually be relevant in a more natural listening environment...
2. Using online data collection makes our science better. Our team has totally drunk the reproducibility crisis Kool-Aid, and as we move to adopt the new best practices for promoting reproducibility of research, we need to find ways to collect data from larger sample sizes. And make in-house replication studies the norm. And not limit our samples to reflect the demographics of just our university. And run more control experiments. And better vet our stimulus sets. And verify that our results aren't contingent on a single stimulus set. And so on...
3. It's hard to keep up with my extremely productive colleagues if we limit testing to our physical lab. For better or worse, science moves very quickly these days. Online data collection also facilitates productivity of our trainees,

who are expected to have increasingly strong publication records to get academic jobs; it even feels like our undergrads are increasingly expected to have publications in order to be competitive for graduate programs. Using online data collection helps us to keep moving forward, while prioritizing in-person data collection for studies that really need this format (e.g., child studies, studies that require standardized assessments, neuroimaging/eye-tracking studies).

4. Emerging technologies exist to provide high quality data collection online, even for speech perception studies.

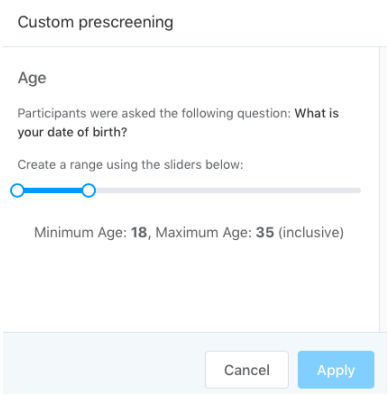
Here I provide a tutorial for speech perception researchers who may be considering making a transition to online data collection. This discussion is focused on using Prolific as the online participant pool, Gorilla as the builder/host of online experiments, and the headphone screen of Woods et al. (2017) to promote assurance of an acceptable listening environment.

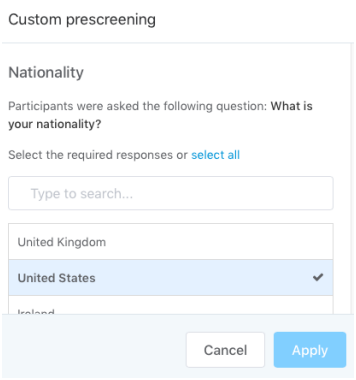
I'm not an expert on online testing, and I'm definitely not an early adopter of these methodologies. Part of the reason why I'm late to the game is because I had a very hard time figuring out how to actually use MTurk. This tutorial is geared towards people, like me, who may in principle be open to online testing, but who have had some difficulty in figuring out exactly how to implement this method. Suggestions from the community for this document are most welcome.

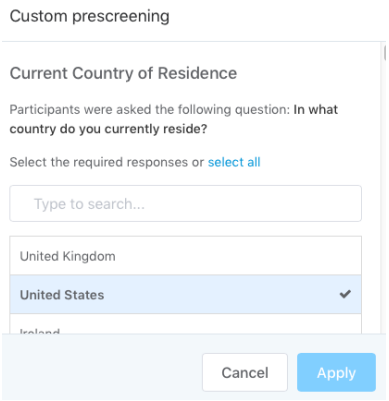
I've tried to be very transparent about our successes and challenges. My thoughts here are geared towards speech perception colleagues. There are great resources for online testing in general, like this piece by Dr. Jennifer Rodd. Please feel free to reach out to me (r.theodore@gmail.com) with questions or suggestions.

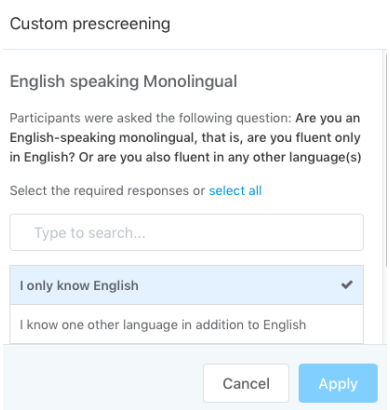
Prolific

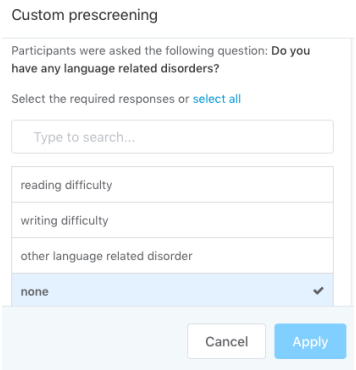
Prolific is an online participant pool. When people join Prolific, the first thing they do is complete a detailed "About You" section where they provide demographic information. Researchers can use this information to set criteria for who is eligible for a given study. The filters are easy to use, comprehensive, and you get realtime information about how many people are eligible with the applied filters. An example of the interface is shown below.



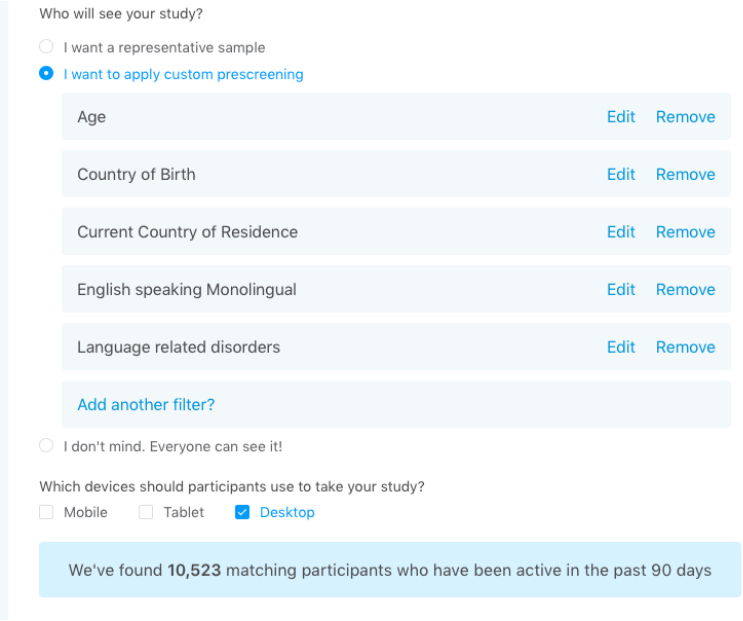








Right now, over 10,000 people meet the constraints of age between 18-35 years, born in US and currently residing in US, monolingual English speaker, and no history of language related disorders.



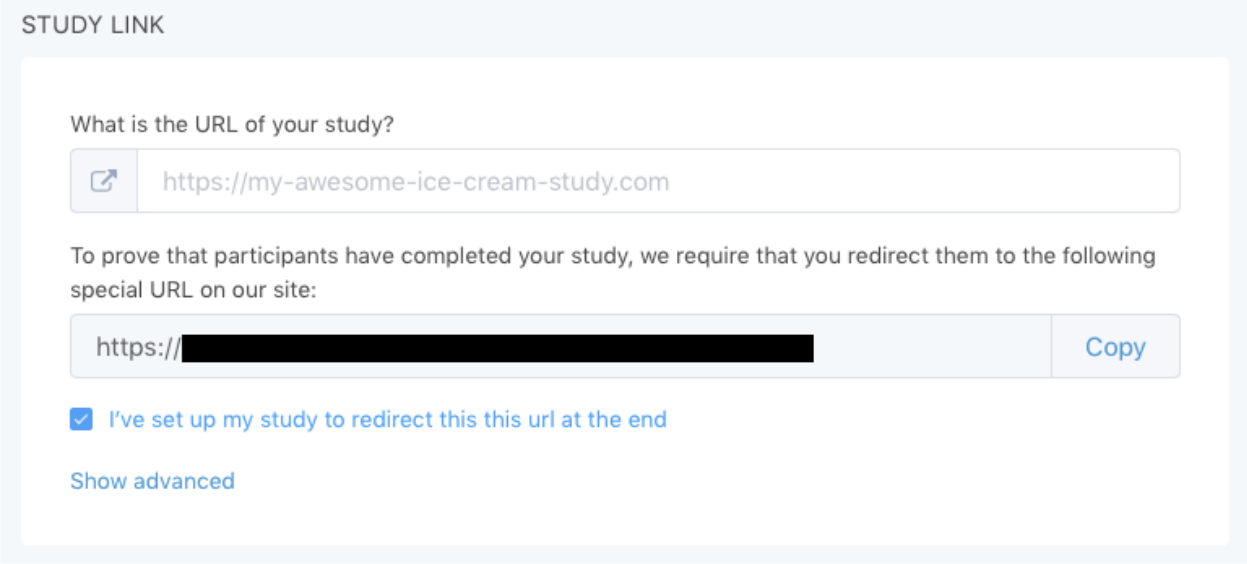
Researchers can filter based on participation from their own Prolific studies, either by *excluding* people if they participated in a specific past study or by only *including* people based on participation in a past study (e.g., using a custom "whitelist" for a longitudinal study). Researchers can also filter by approval rate and total number of completed Prolific studies, in addition to anything else that is provided in the "About You" section.

Prolific does two important things for you: (1) they get your study distributed to participants, and (2) they handle the money. Researchers Top-up their Prolific account with some funds, which is then distributed as researchers approve submissions. Prolific makes money by charging a fee based on what you pay the participant. I believe that this is 30% for academic researchers (that's what I am charged), but I can't find this information explicitly stated on their website. This tweet says that it is 33%. Perhaps they have made a change recently?

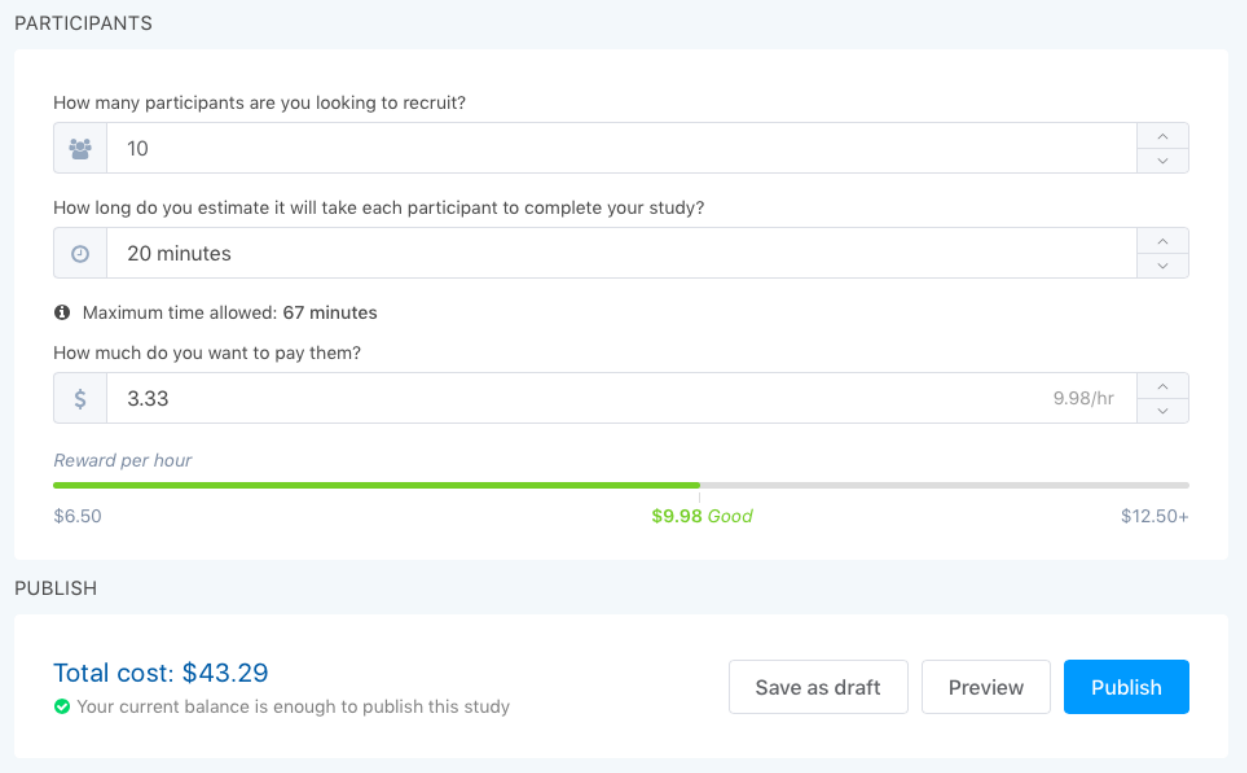
For a study that takes an estimated 20 minutes to complete, the participant would get paid $3.33 from my Prolific account (consistent with our $10/hour payment rate, the same rate we use for in-lab behavioral studies) and Prolific

would get paid $1.00. Prolific requires that you pay participants a minimum of $6.50/hour. The money gets placed in the participant's Prolific account, which they can withdraw using PayPal. Prolific is trying to make themselves distinct from MTurk by providing ethical treatment to participants and ensuring high quality data to researchers, and I've really been impressed with their efforts on these fronts. I've also been very impressed with how Prolific facilitates administrative tasks. For example, a click of the mouse will generate a detailed receipt of all payments to participants and Prolific for a given study, which can then be submitted to your university for funds reconciliation.

When you set up a new study on Prolific, there's a place for you to provide a link to your actual study (you'll get this link from Gorilla, more on this below). Prolific provides a "redirect" link that should be embedded at the end of your actual study (you'll paste this link to the Finish node of your study in Gorilla, more on this below).



You also specify the number of participants, and the amount that they will be paid. Remember, Prolific requires that they be paid at a minimum rate of $6.50 per hour. You are told the full cost of the study, and have to have this amount in your Prolific account to run the study.



In Prolific, participants can Return their submission at any time. Returning a submission means that the participant withdraws their submission. They could do this before they click the link to go to your Gorilla study, after they start the Gorilla study, or even after they finish it. This is a very good thing for many reasons, and it is clearly conveyed in Prolific study pane. I say a bit more about this below, noting here that when this happens, you need to manually address these returned submissions in Gorilla.

The general pipeline is as follows:

1. Top-up your Prolific account with some funds.
2. Create a new study by entering a name and description, the number of participants and their pay, filters for participant demographics and devices (e.g., desktop/laptop, mobile), and a link to the study. Be sure to place the redirect URL at the end of your study.
3. After you publish your study, it will be distributed to participants who match your filters. They will get directed to your study link. When they finish the study, the redirect link sends them back to Prolific where they are given a completion code.

4. Watch the report of participant completion in real time, stay on top of message from participants as they come in, and approve completed submissions in a timely fashion.
5. Generate the Download export as soon as your study is completed, which provides a report of participant demographics (more on this below).

Prolific has an extensive Help Centre and a blog. There's also a Prolific subreddit, which I've found really useful for learning about Prolific from the participant perspective.

Gorilla

Given that Prolific is solely the recruitment and payment tool, you need a way to build a study and host it online. This is where Gorilla comes in. Gorilla is both an experiment builder software and a server to run/host/deploy online studies. It does all of this extremely well, and integrates seamlessly with Prolific (documentation from Gorilla, documentation from Prolific). I love Gorilla as experiment builder software so much that we're transitioning our in-lab studies to this too (leaving behind E-Prime, SuperLab, and PsychoPy).

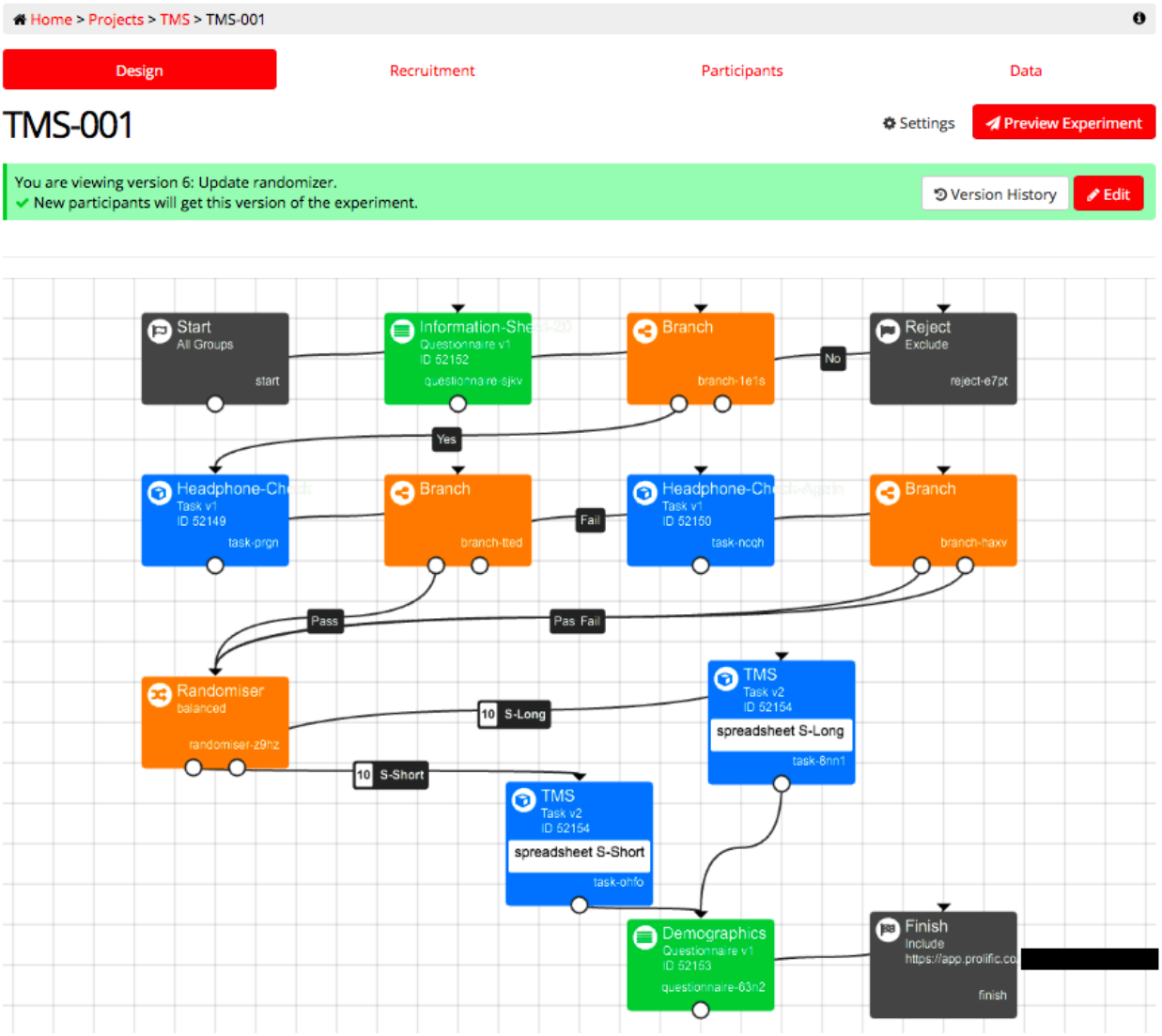
The documentation/support for Gorilla is amazing. There are numerous tutorial videos to get you going, and it generally takes us about an hour to get even the most complex design (e.g., feedback, multiple training/test sessions) fully programmed. The Gorilla team has also published papers explaining and validating their tools (available here and here).

Most everything in Gorilla is organized around Projects. Projects consists of (1) experiments, (2) tasks and questionnaires, and (3) open materials. Experiments are simply ordered combinations of the tasks/questionnaires in your project, along with other nodes such as a Finish node (to redirect participants back to Prolific) or a Randomiser node (to assign participants to conditions).

The user support for making tasks/questionnaires and experiments is so excellent on Gorilla that I won't go into details here. The interface for building experiments is very clean and easy to use, even for really complex designs (e.g., counterbalancing, different components, branching based on performance in a

given task, redirecting back to Prolific). I can't say enough good things about their support documentation for researchers; it's quite lovely.

The image below shows one of our experiments, a study called TMS. Nodes in green reflect questionnaires (e.g., consent form, demographics), nodes in blue are tasks (headphone screen, test task), and nodes in black/orange are experiment action nodes. In this study, participants first go to the consent node (Information-Sheet), and then data from that node are passed to the first Branch. If the person did not give consent, then the study ends (Reject node). If they do give consent, then they move on to the headphone screen. Branching happens again, and then participants are directed to the Randomiser node, which is set to send 10 people to each of two task conditions. After completing the task, everyone is sent to on to complete the demographics questionnaire before being sent back to Prolific.



There are many other handy tools in Gorilla, including being able to limit participant access based on device, browser, location, and even connection speed. You can also view participant's progress in real time and see how many participants are at each node of your study in real time. If you use any of the Requirements tools in Gorilla (e.g., limit participation to desktop/laptop computers from people in the US), it's really important that you mirror those requirements on Prolific. Otherwise, Prolific will send participants to Gorilla, only to have them not be able to access the study due to the Gorilla requirements.

Gorilla is free to use to build experiments. They make money by charging a Token to download the data. You spend one token for each participant's data set. It only gets spent the first time you download the data and only once for a given

Gorilla experiment. For example, if you have five tasks in one experiment, you're only charged one token per participant. Or, if you run a small set of participants through your study, download their data, and then run the full sample through, you aren't charged a second time for data that was downloaded the first time. Standard academic pricing is $0.98 per token if you buy at least 50 tokens at a time ($1.08/token + 10% additional free tokens). They also have team and departmental licenses that offer tokens for substantially less than the standard academic pricing.

One of my favorite things about Gorilla is how easy they make it to support open science principles through collaboration and open materials. Any project component can be placed in an Open Materials section, which allows anyone to use the component. Sharing a project can also happen via the Collaborate option. Please reach out if you'd like access to our Gorilla tasks.

Woods et al. (2017) headphone screen

We're using the headphone screen developed by Woods and colleagues (Woods et al., 2017). It's a very clever, six-trial task. On each trial, three tones of the same frequency and duration are presented. One tone has a lower amplitude than the other two tones. Of the two tones with equal amplitude, one is presented 180 degrees out of phase across stereo channels. The listener is asked to pick which of the three tones is the quietest. Performance is generally at ceiling when wearing headphones, but poor when listening in free field (due to phase-cancellation).

The authors have carefully vetted this protocol, providing convincing evidence (at least to me) that defining "pass" as ≥ 5 correct responses provides a sensitive (enough) measure of headphone use. I've glossed over some important details in their manuscript, including that the screen performs better for desktop speakers than laptop speakers, and it's of course important to note that *(*as for any screen) it's not a perfect detection tool. But as a friend/wicked smart colleague recently reminded me, we shouldn't let perfect get in the way of the good.

It's worth noting that this tool doesn't seem to be the convention (yet!) for online speech perception experiments, and that many high-quality online speech perception studies have been published that rely on self-report measures for headphone compliance. I'm keen to hear feedback from other researchers on this tool, and any other tools, as more of us use online testing for our studies.

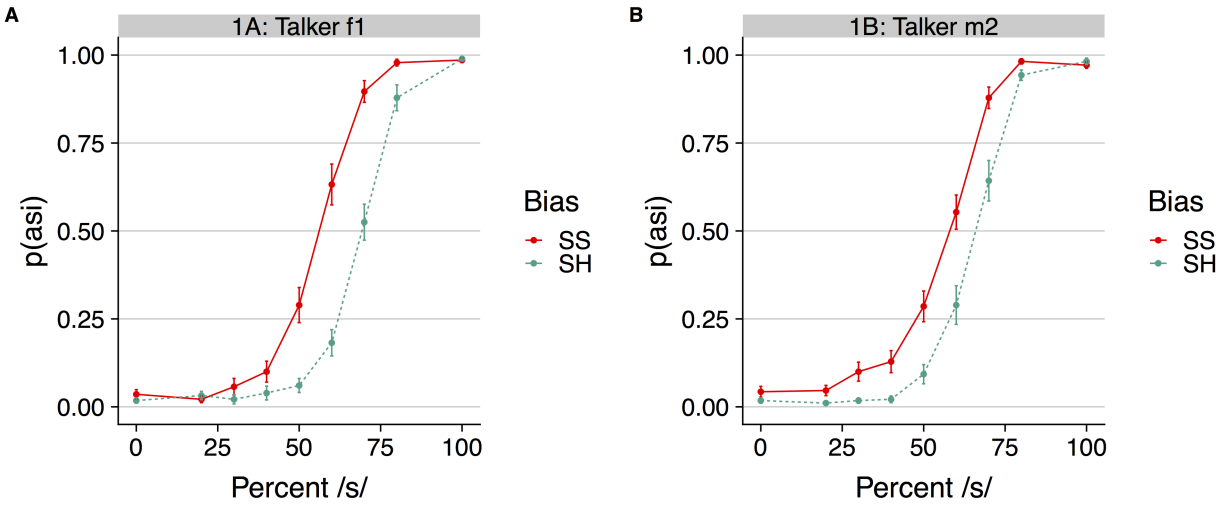
Some successes

Here I present results from some standard speech perception tasks that we conducted online, with the results replicating phenomena observed in previous laboratory tests. In each, I provide some measure of individual performance as well as group-level patterns, and report measures of attrition due to poor data quality and headphone non-compliance. We're not the first to vet some of these tasks for online data collection; instead, this next part is written to be very transparent about our experiences with Prolific and Gorilla specifically.

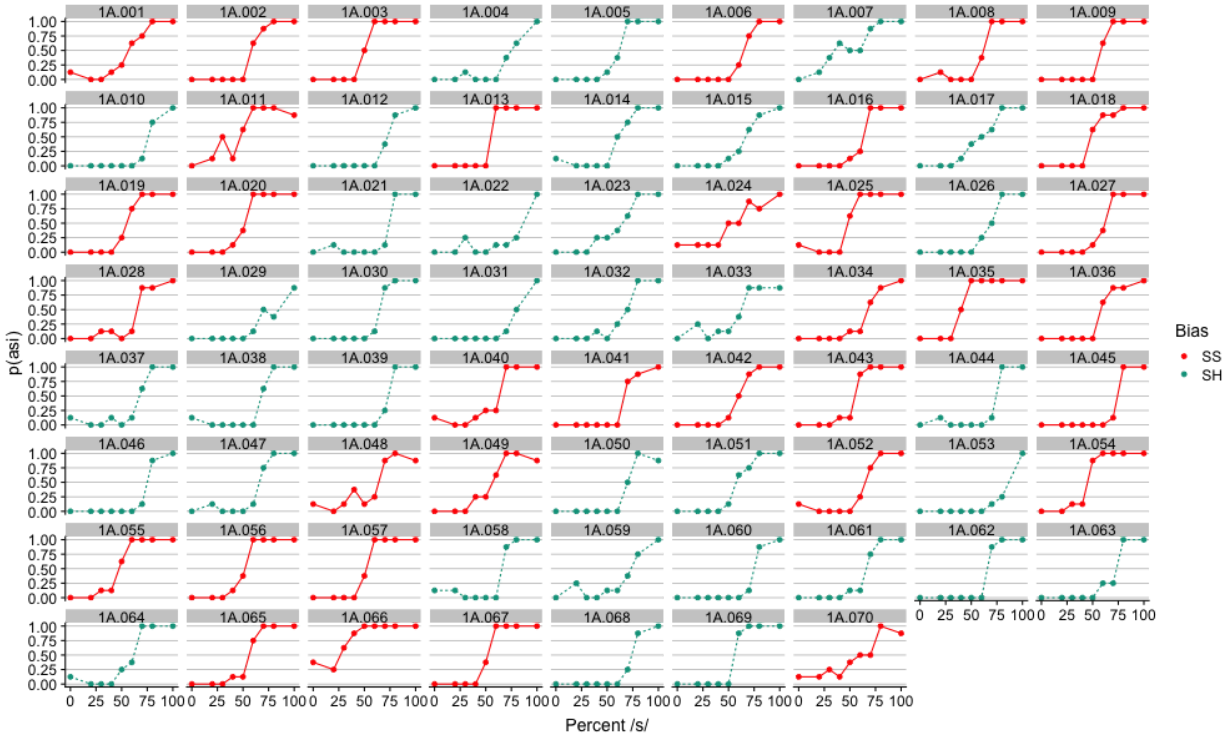
Lexically guided perceptual learning

We recently finished online data collection for a study (LoGlo) that used the lexically guided perceptual learning paradigm. The study consists of six experiments (n = 560) and has built-in replications using two different stimulus sets (i.e., experiment 1A and 1B reflect the same manipulations but use different stimuli). All participants completed a lexical decision exposure phase followed by a phonetic identification test phase (for an *ashi-asi* continuum). We had a few different manipulations across experiments, and all between-subjects cells contained n = 35.

Here's the group-level data for experiment 1, which was a replication of the standard lexically guided perceptual learning task using our two novel stimulus sets. Nice, eh?



Here's a plot showing performance at test for each participant in experiment 1A. Look at those beautiful psychometric functions!

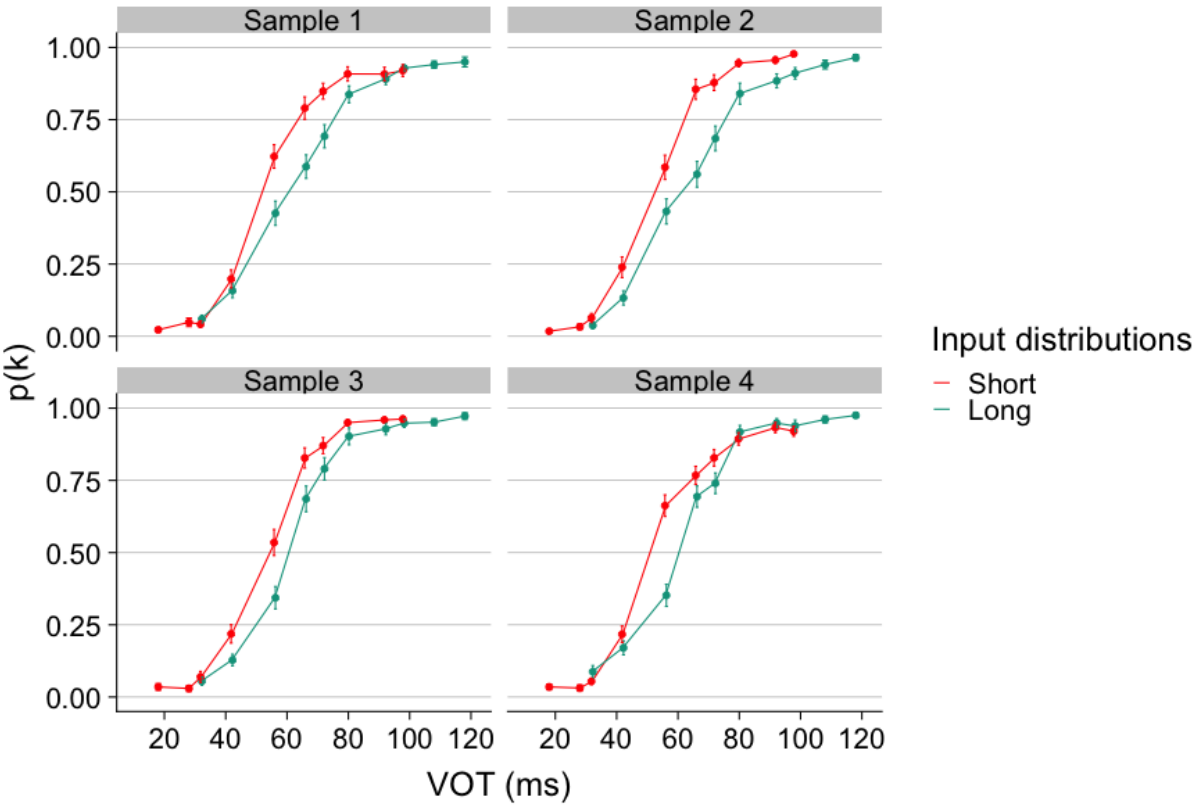


For this entire study (planned n = 560), we only had to exclude 32 participants (32 out of 592 total participants, 5% attrition rate) due to failure to meet a priori performance-based inclusion criteria, which were high accuracy during the lexical

decision exposure phase (≥ 70% correct for each of four item types) and a logistic response function at test.

Distributional learning

For Nick Monto's just-about-done dissertation, we used Prolific/Gorilla to run a series of distributional learning experiments. We call this study STGO. In these experiments, listeners complete two blocks of phonetic categorization for VOT input distributions that, across blocks, change to statistically cue a shorter vs. longer voicing boundary. Across experiments, we manipulated the order in which listeners hear each set of input distributions, and whether or not the change in input distribution is concomitant with a change in talker. We've run parallel versions of each experiment with separate stimulus sets. (The results are fascinating, but that's Nick's story to tell...) What I've shown below are results that indicate a distributional learning effect in four participant samples (n = 40 participants in each function, for 80 participants in each of the four facets). Group-level performance in block 1 is shown for listeners that heard either short vs. long VOT input distributions. In all four samples, the voicing boundary is shifted in line with the distributional input, indicating a distributional learning effect.



In this study, our a priori sample size was n = 320. To achieve this sample size, we had to test an additional 52 people who were excluded due to a priori inclusion criteria (i.e., showing a logistic response function in each block, having a voicing boundary within 40 ms of our intended boundary in the first block), leading to a 14% attrition rate on these criteria. We suspect that part of the reason for higher attrition is that this task is really, really boring (304 trials of phonetic identification). Consistent with this suspicion, most people who were excluded due to a lack of logistic response function only met this criteria in the second block, suggesting that response quality diminished over time. (Unfortunately for us, we need high quality data in both blocks to test our hypotheses).

Talker discrimination

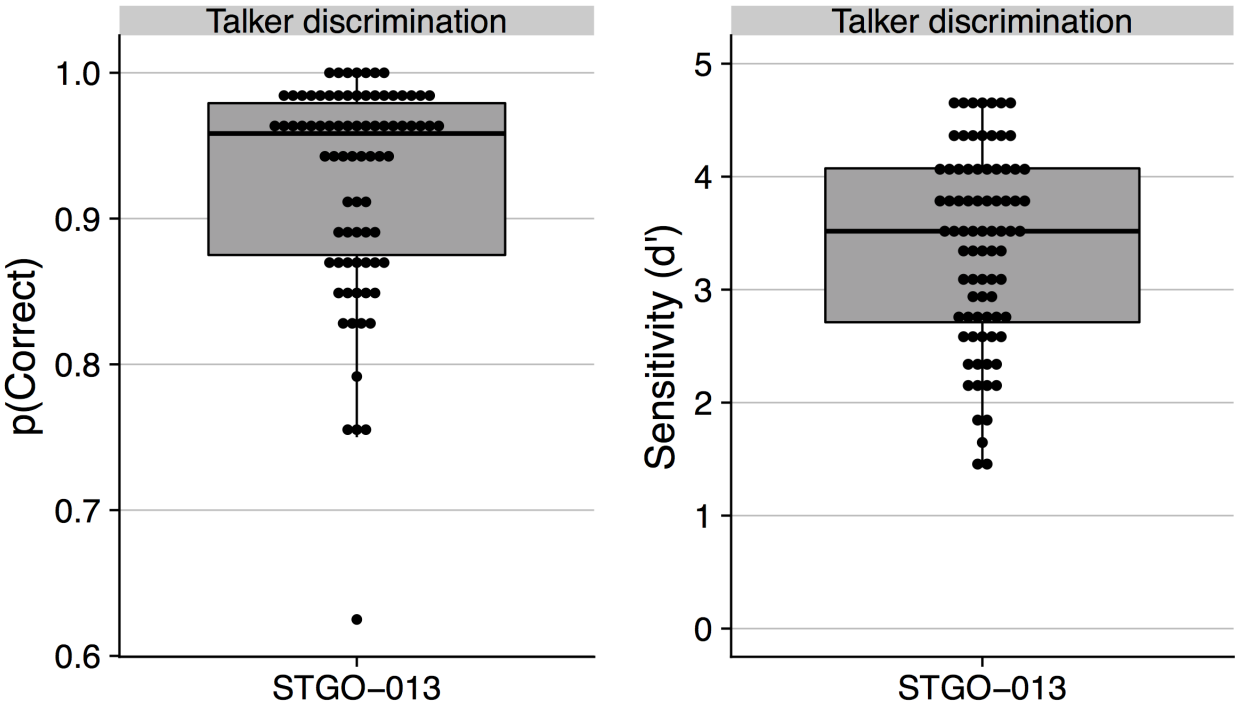
For the STGO study, we ran quite a few pilot studies (13 to be exact, maybe we did let perfect get in the way of the good...) in order to get the final stimulus set. We needed VOT continua from two different talkers who had highly discriminable voices, yet yielded equivalent phonetic identification for their continua. As we all

know, this isn't as simple as just equating VOT at each step for the two talkers' continua...

This pilot testing was the first thing we did with our new online IRB protocol, and we zipped through these 13 stimulus checks in a few weeks, something that would've taken an entire semester in the lab. Being able to test online let us be really, really picky about the stimuli, which ultimately lets us be more confident in the conclusions we draw in the main experiments.

In each pilot, listeners completed two blocks of phonetic categorization (one for each talker's continuum) in addition to a talker discrimination task. I think the rest of the in this tutorial provide good evidence that phonetic identification tasks can be completed online, so I'll focus here on the talker discrimination task. The talker discrimination task consisted of same talker and different talker pairs. For a given pair, VOT was constant between talkers; across pairs, we sampled different VOTs that spanned the continua range.

The plot below shows the distribution of scores (n = 80) on the talker discrimination task in terms of proportion correct (left panel) and d' (right panel) for the final pilot study. We wouldn't have this high level of performance if participants weren't doing the task (or if the talker's voices weren't discriminable, yay!).



In addition to these 80 participants, an additional 16 participants were tested but excluded due to failure to show a logistic response function during the phonetic identification task, for an attrition rate of 16% for performance-based inclusion criterion (no participant was excluded based on talker discrimination performance, given that poor performance on this task could have meant that the voices weren't discriminable to the participant).

Ganong effect

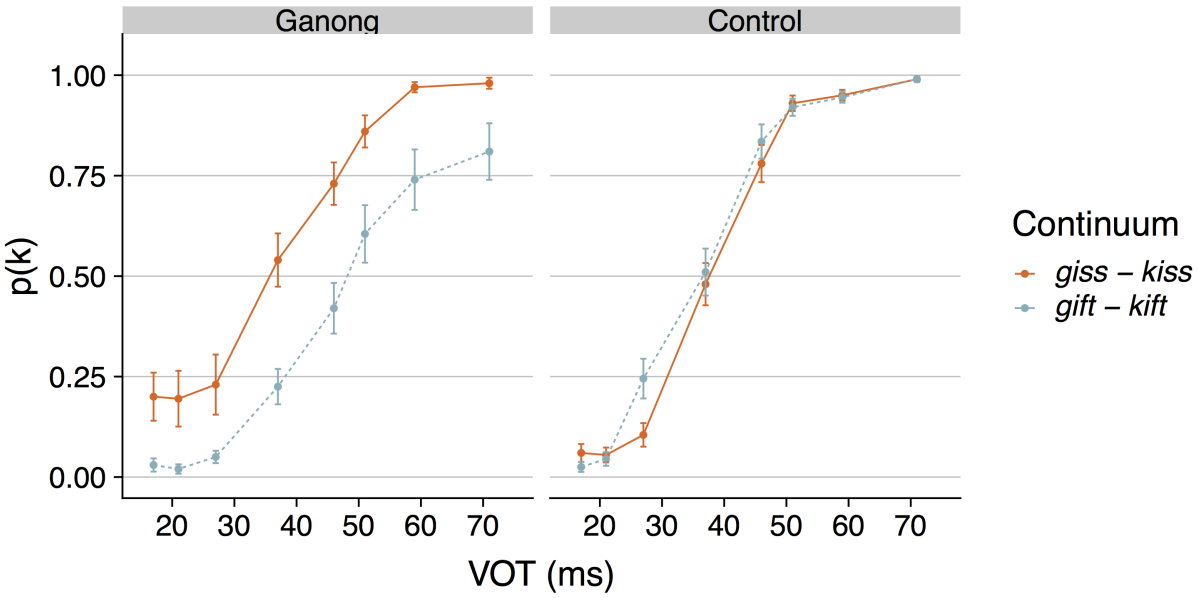
As part of a larger set of experiments with Nikole Giovannone, we've been running in-lab participants through a Ganong task as a measure of individual differences in lexical recruitment for speech perception. Stimuli for the Ganong task are two eight-step VOT continua that perceptually range from *giss* to *kiss* and *gift* to *kift*, respectively. The continua were created by splicing eight different VOT onto /ɪs/ and /ɪft/ tokens that had equivalent durations.

Given the copy/paste nature of stimulus construction, a reviewer for one of these studies asked whether our observed Ganong effect (i.e., more /k/ responses in the *giss-kiss* compared to the *gift-kift* continuum) might be driven by residual coarticulatory cues in the VOT and/or coda portions of the tokens. It's a great question. In the absence of being able to quickly run a control experiment online,

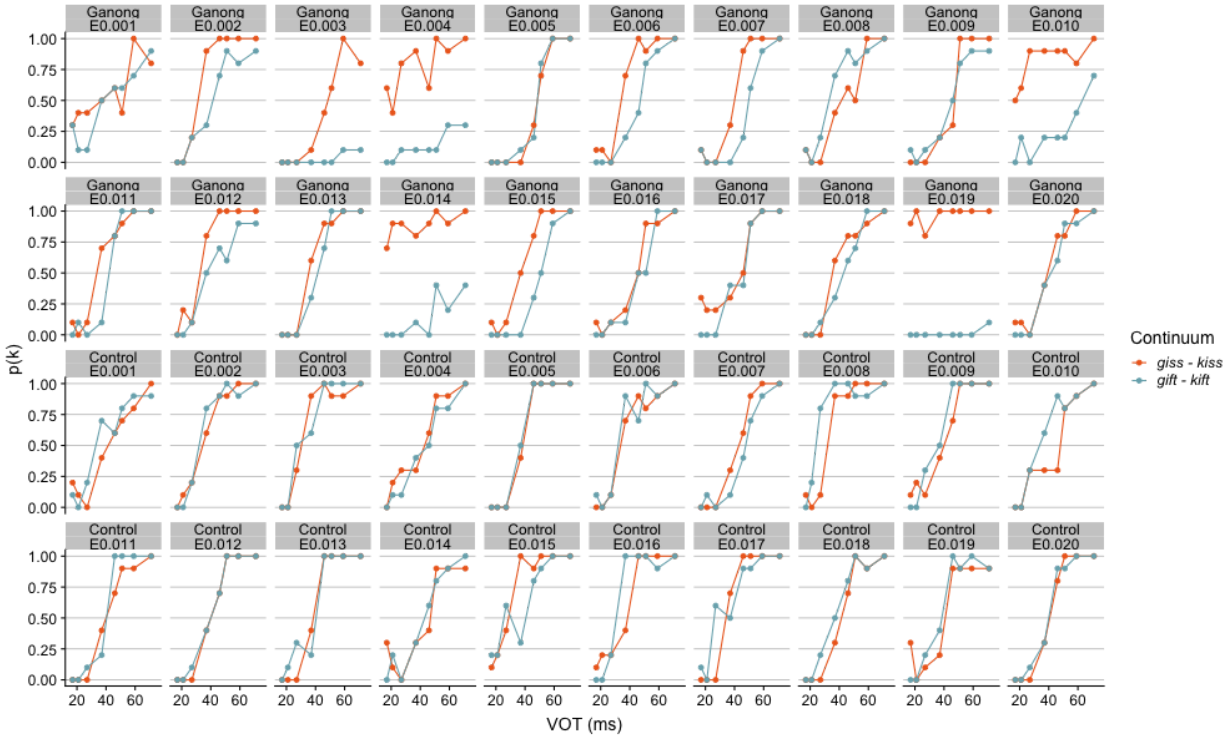
we likely would've addressed this concern by making a plea to convention in order to prioritize data collection for other studies in the lab.

Instead, we ran this control experiment online. Listeners (n = 20) completed two blocks of phonetic categorization, one for the original Ganong stimuli and one for two control continua. The control stimuli were created by removing the coda portion from each of the tokens used as Ganong stimuli, yielding continua that perceptually ranged from /gɪ/ to /kɪ/. In others words, the control stimuli were identical to the Ganong stimuli except for the removal of the disambiguating lexical context provided by the coda portions (i.e., /s/ for the *giss-kiss* continuum, /ft/ for the *gift-kift* continuum).

The plot below shows group-level performance. A strong Ganong effect was observed for the Ganong stimuli, replicating what we observed for the in-lab sample, but not for the control stimuli.



And here are the response functions for each participant. No participant was removed due to failure to meet our a priori inclusion criterion (i.e., showing a logistic response function for the control block). It's clear that there is more individual variability for the Ganong compared to the control block, but the range of individual variability for the Ganong block is actually consistent with what we observe for in-lab studies.



Other tasks

We've also been successful in using Prolific and Gorilla to run talker normalization studies, where the dependent measure is RT in a speeded word identification task for single talker vs. mixed talker blocks (the paradigm used here). Julia Drouin and I are beginning a set of studies to examine factors that promote optimal short- and long-term adaptation to noise-vocoded speech, taking advantage of the ability to run longitudinal studies in Prolific. This document is already pretty long so I'm not presenting these here, but feel free to reach out to me (r.theodore@gmail.com) if you'd like more information on our experiences with these different designs.

Some challenges

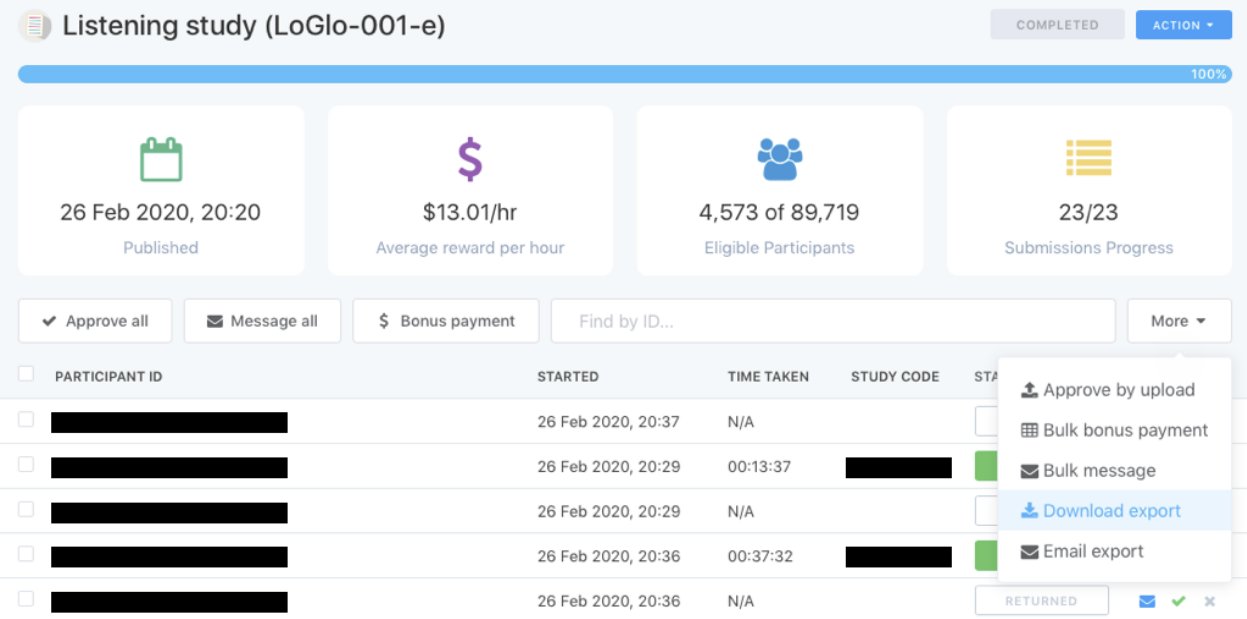
Prolific

It should be very clear at this point that my overall recommendation for Prolific is extremely high. But, in the spirit of providing some constructive feedback to the

(amazing!) team at Prolific and being transparent about my experiences, I have two major concerns right now that I'd love to see addressed in the future.

Demographic exports are not linked to participants' information at the time of participation

Prolific makes it really easy to export collated participant demographics for each study via the Download export option. This returns a CSV file containing general demographics (e.g., age, gender, approval rate, time started/completed) in addition to values for all user-defined filters.



But, many of the values in this export are dynamic; they reflect demographic information **at the time of download**, not at the time of participation. This is a huge flaw, in my opinion, because there is no way to provide a *reproducible* report of participant demographics. Age information could potentially change each time it is downloaded, for example, leading researchers to have anxiety attacks over potential IRB violations when they see a participant's age listed as 36 years old, when the IRB protocol states that participants will be between 18 and 35 years of age. If the downloaded export is lost at some point (e.g., accidentally deleted from a researcher's computer), there is no straightforward way to recover that information. To promote reproducibility and provide better tools for researchers, I suggest:

1. The Download export be modified to reflect participants' information at the time of the actual study.
2. In the absence of this change, Prolific could better alert researchers to the fact that the Download export dynamically changes over time. Right now, this information is provided in the Prolific Help Centre, but this could also be prominently placed somewhere in the study pane (shown above).

In the meantime, dear colleagues, generate the Download export as soon as your study is completed, and back-up that file.

There is no way to compensate participants for only the headphone screen if they fail the headphone screen

This is the biggest source of frustration among our team when using Prolific. To put this issue in context, as part of the procedures for promoting ethical treatment of study participants, Prolific doesn't allow researchers to use participants' time without being compensated. **I strongly support this decision.** I agree that it is unethical to require participants to spend time going through a series of prescreening questions without compensation, only to be told that they don't meet study requirements. For any given study this may only be a few minutes of their time, but we must also consider the collective time a participant may spend doing this. So let me be very explicit; I agree that Prolific participants must be compensated for any and all time that they spend on our studies.

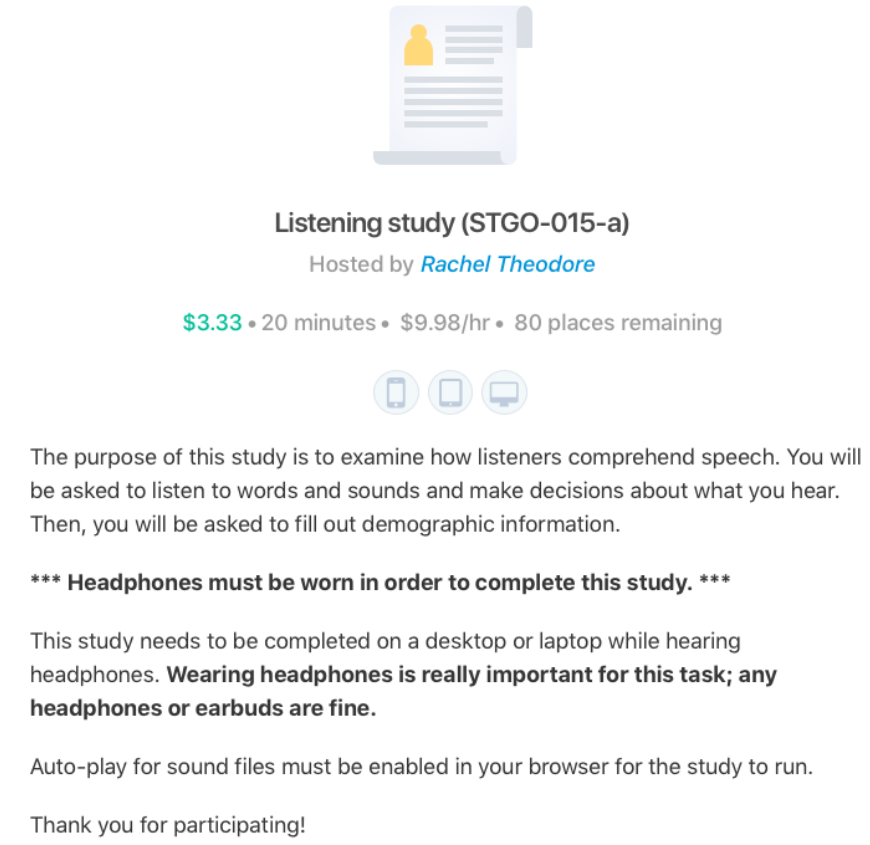
The good news is that Prolific makes it very easy for researchers to prescreen participants across numerous custom parameters; we can filter who has access to our studies according to any of the information Prolific collects in the About You section. The bad news for speech perception researchers who use the Woods et al. headphone screen is that we need this to be confirmed at the time of testing.

To provide a scope of the issue, here's information on the headphone screen for two recent studies. Our LoGlo study used a lexically guided perceptual learning paradigm, and our target sample was n = 560. To achieve this sample, we tested 704 participants in total. The number of participants who failed the heading screen, and thus had to be excluded from the sample, was n = 112. This reflects

an attrition rate of 16%, and sunk costs of $484.96 [112 participants * ($3.33 to each participant + $1.00 to Prolific)]. In contrast, of the 592 participants who passed the headphone screen, we only had to exclude 32 participants (5%) due to failure to meet our a priori inclusion criteria (i.e., high accuracy during the exposure phase and a logistic response function at test). The overall attrition for LoGlo was 20% (144 of 704 participants had to be excluded).

Our STGO study used a distributional learning paradigm to measure listeners' sensitivity to changes in statistical properties of the input cueing the stop voicing contrast. We tested 399 participants to reach our target sample size of n = 320, for an overall attrition rate of 20%. In STGO, only 27 participants were excluded due to failure to pass the headphone screen. The other participants (n = 52) were excluded due to failure to meet a priori inclusion criteria (i.e., logistic response function at test, voicing boundaries within 40 ms of the intended boundary given the input distributions). Sunk costs for failure to pass the headphone screen were $116.91.

We suspect that the lower attrition rate for headphone compliance in STGO compared to LoGlo is because we repeated the headphone screen three times in the former and twice in the latter. We did this by implementing a branch in Gorilla; if a participant passed the first headphone screen, then they moved on to the main experiment. If they did not, then they were routed to complete this again, with the instructions providing a reminder that headphones are required for the study. In STGO, we did this branching a second time, thus providing three chances to pass the headphone screen. If you're interested in the breakdown, 306 participants passed on the first round, 52 participants passed on the second round, and 14 participants passed on the third round. It's of course possible that repeating the headphone screen might introduce practice effects, but some of the data in Woods et al. (2017) led us to be not too concerned about this possibility, and to weight the possibility that participants are taking time to put on headphones a bit more heavily. Note that for all of our studies, we make it very clear in the study description that headphones are needed.



So, the issue is that in order to be compliant with Prolific's terms of agreement and our own conscience regarding ethical treatment of research participants, we can't stop the study for participants who fail the headphone screen. There's currently no way for participants in the same study be paid different amounts based on whether they complete one or multiple tasks. This is the solution that I recommend, noting that Gorilla already has the capacity to implement this:

1. In Prolific, researchers could set up two time/payment contingencies for a single study. This could be clearly conveyed in the existing study description format. For example, speech perception researchers could describe it along the lines of, "The first part of this study is estimated to take two minutes and consists of a brief check for headphone use. All participants will be compensated $0.35 for completing this portion. If headphones are detected, then the study will continue for a total estimated completion time of 20 minutes. Participants who complete both parts will be compensated with $3.33."

2. In Prolific, two redirect URLs/completion codes could be provided for each study, one for each time/payment contingency.
3. In Gorilla, researchers could have two Finish nodes in their study, one for those who fail the headphone screen and one for those who complete the entire study, with the appropriate Prolific redirect link placed in each Finish node.

If any member of the Prolific support team is reading this and would like to chat about this idea, and why current Prolific workarounds (e.g., running this procedure as two studies, with a whitelist used to select participants for a second study based on a headphone screen first study, asking participants to return submissions, rejecting participants who fail the headphone screen) don't work for this purpose, I'd be eager to contribute to this dialogue!

In the meantime, my advice to colleagues who use the headphone screen is to (1) make sure your study description emphasizes the importance of headphone use, (2) give participants at least two opportunities to pass the screen with a gentle reminder of the importance of headphone use each time, and (3) try to calibrate your expectations for attrition.

Gorilla

To be honest, it's hard to think of a way that Gorilla could be better. I am very impressed with this software/tool/platform! So much so that we're currently considering transitioning even our in-lab studies to Gorilla when possible (we currently use a combination of E-Prime, SuperLab, Experiment Builder, and PsychoPy). But in the spirit of being as constructive as possible...

Pricing models could be a bit more transparent

At the time of writing, I've spent $2263 to buy 2310 Gorilla tokens. This works out to $0.98 per token, consistent with the standard academic pricing ($1.08/token + a 10% bonus if you buy tokens in increments ≥ 50). I just found out that Gorilla has other pricing models that provide a substantial discount; I got this information by participating in a survey that Gorilla distributed on Twitter.

Within a day of getting this information, I started the process for securing a departmental license for our community, which can reduce the per token fee to less than half of the standard academic price. I want to be explicit that I strongly support Gorilla's current fee model (free to build, pay for data/participant), I just wish I had known about the high volume pricing models sooner than I did.

Not knowing about these alternate pricing models could very well be *entirely my fault*. I see now that there is some small print about other pricing models on their Pricing page. One suggestion for making this more transparent to researchers would be to send users an e-mail about alternate pricing models if a large token purchase is made (either at one time, or cumulatively within a six-month time period, for example).

I realize of course that the product I've come to enjoy is likely so wonderful, in part, because of the money that it generates, which can then be used to support product development, technical support, and (hopefully!) excellent compensation for their outstanding staff. So my desire for more affordable pricing should be viewed as a very minor concern, and largely in the spirit of wanting this tool to be more accessible to researchers.

Tips and tricks

I want to stress that the Prolific and Gorilla support documentation is amazing. We've had a few glitches along the way that could've been prevented if we relied more on that up front, so I encourage you to spend some time taking advantage of the great documentation that they've prepared for us. After a bit of trial and error, here's a pipeline that works very well for us when we're actual at the data collection stage.

- As tempting as it is to think that you can click the "Publish" button in Prolific, head to the marina for your Wednesday night regatta, and then come home to a beautiful pile of data after an evening of sailing, I don't recommend this strategy... Our experience has been that Prolific/Gorilla work best when one can monitor the study, and potential messages from participants, in real time.
- Monitoring submissions in real time is especially important if you have any sort of counterbalancing/randomizing constraints in your Gorilla experiment. Let's say that you have a study where you want data from 320 people, with

40 people assigned to each of eight counterbalancing conditions. This is super easy to implement in Gorilla with a Randomiser node. But here's the (potential) issue: Some people may start the study, get assigned to one of your randomizer conditions, but then return their submission on Prolific without finishing the study. Gorilla doesn't know that this is happening; by their view, a participant is still in progress. However, if you're monitoring the Returned submissions on Prolific, you can then manually reject the will-never-be-completed submission on Gorilla. This is not the same as rejecting someone on Prolific; instead, this will let Gorilla know that a participant is not in the given node and that they should route additional participant(s) to that node. Failure to do this in real time might lead to situations where you have more people that you need in some conditions, but fewer people than you need in other conditions.

- Monitoring submissions in real time can also let you make sure that the tokens for a given Gorilla study are sufficient for the number of participants that you enrolling in Prolific. As the Gorilla support documentation notes, it is a good practice to set the number of tokens on Gorilla to be higher than what you list in Prolific. This is because participants who start the study but then withdraw (by returning their submission on Prolific) will still be using a Gorilla token until you manually reject them in Gorilla. If you don't have any open Gorilla tokens available, then participants who enter from Prolific will not be able to complete their study. We generally set the number of Gorilla tokens to be 20% higher than the number of participants we recruit on Prolific, manually reject in Gorilla returned submissions on Prolific in real time, and then unassign the extra Gorilla tokens once data collection is done in Prolific.
- Prolific now lets you add additional participants to your study after the initial set of participants has finished (thank you Prolific team, this was a super helpful change!). This is especially great for facilitating administration of studies that have specific counterbalancing constraints and inclusion criteria. Here's how we use this option to streamline data collection, using the n = 320 example (40 people in each of eight conditions). First, we run a very small sample through all conditions by setting the Randomiser node in Gorilla to send two people to each of the eight conditions; in Prolific, our study would recruit 16 participants initially. After this study is finished, we then download the data and set up the analysis script, doing a final check of our

program/experiment in the process. Then, the Randomiser node in Gorilla would be updated for the next run. Let's say we now want to get to half of the final sample size; we could edit the Randomiser node to send 18 people to each node, and then increase places on the Prolific to accommodate an additional 144 participants. We can then download the data into our already prepared script, and adjust the Randomiser node as needed to get the final sample size, taking into account any participants who were excluded due to failure to pass the headphone screen or any other inclusion criteria. This becomes an iterative process until, and at some point, you might increase places in Prolific by one and send them to a version of your Gorilla program that has the Randomiser set for one participant to be sent to one specific condition. It's really not as complicated as it sounds. We've actually had a lot of fun watching the results roll in, and setting aside 4-6 hours (or even a day) to complete data collection for a large study in one fell swoop is obviously far less time than it would take to do this in the physical lab.

- In our experience, we've had little evidence of bots completing our studies. I think that for most speech perception researchers, it's pretty easy to detect bots (and low effort responses in general) because we often have a general expectation of what performance should look like if a participant is actually completing the task. For example, phonetic identification functions for an acoustic-phonetic continuum should show a logistic response pattern. If a person only presses one button during the task, we won't see this pattern. For studies where RT is the dependent variable, we often make high accuracy an inclusion criterion. So if we see a person with chance performance, perhaps due to pressing the same button for all trials in a 2AFC task, we can easily recognize this as a low effort submission. We build in a few bot-detector code chunks in our analysis scripts that, in addition to looking for one-button responders, analyze RTs (even when RT isn't the primary dependent variable). We look for cases where RT is unusually inhuman, such as all RTs occuring within 5 ms of each other, or all RTs occurring faster than 30 ms. I think this is a place where we should think carefully about our tasks in order to make sure that bots and low effort response patterns could be readily detected.
- One last tip is to stay on top of your messages in Prolific. Participants have reached out to me for clarification, trouble-shooting, and general confusion. It's helped me learn how to offset these types of issues a priori, and in

general it has been a very cordial community. Messages let us have some level of interaction with our participants, and gives us a way to thank them for supporting our research.

Referrals

Prolific and Gorilla both have referral programs. At Prolific, researchers who sign up with a referral link get $100 off their first study worth $250 or more, and the referring researcher receives $50. At Gorilla, researchers who sign up with a referral link receive 20 free tokens, as does the referring researcher. My referral links are shown below below, but you can get these from any researcher currently using Prolific or Gorilla.

https://www.prolific.co/?ref=D4QDYSQT

https://gorilla.sc/signup?referral_token=B26AD6F5-9936-407F-9C96-901CA3B736DB