Response to Reviewers

Reviewer #1

> These protocols are for building a toy self-driving lab, based on LED blinking lights and a light sensor. The protocols cover hardware and software. The result in Fig. 17 is great! I think this demo is fantastic for teaching students about self-driving labs and Bayesian optimization, and its advantage over random search or classical experimental design methods (for example, the grid search you have here). As well as for educating the public, as the concept here is easy to understand. So, the protocols are certainly deserving of the pages of STAR Protocols.

Thank you for your positive feedback! We are glad you like the results in Fig. 17, and we agree that it can be useful both in classroom settings and for the public.

> This is not easy for me to assess without actually having the hardware and going through the motions. But, I generally found the description of the protocols to be clear, and I think I could reproduce them. The hardware is listed and easy to purchase. Below are a few minor comments to possibly help the authors improve their protocols article.

> - Why do we need to control from the cloud? Can't we run this on our local computers and avoid the extra steps of hooking it up to the cloud?

A note was added to the Additional Prerequisites section to clarify this:

> > The purpose of this rather than using a hardwired connection is to emphasize the notion of "cloud experimentation", where the host and the client may be separated by large geographical distances. For more context, see https://github.com/sparks-baird/self-driving-lab-demo/discussions/91 and https://github.com/sparks-baird/self-driving-lab-demo/discussions/62. For links to a simple example using a wired connection, see Problem 2:.

The commentary in the Troubleshooting section has also been updated:

> > Problem 2: Can I use this without connecting to the internet?

> > Potential solution:

> > A simple example of wired communication between a computer and the microcontroller for the microcontroller host code and a Jupyter notebook tutorial (client) can be found at https://github.com/sparks-baird/self-driving-lab-demo/tree/main/src/extra/nonwireless [permalink] and https://github.com/sparks-baird/self-driving-lab-demo/blob/main/notebooks/5.0-nonwireless-search.ipynb [permalink], respectively. While possible with some modification, data communication via a USB cable is not actively supported for new releases of microcontroller host code nor the advanced tutorials. For private, secure, wireless communication between the Pico W microcontroller and the client (e.g., Jupyter notebook running locally), a free, private HiveMQ instance can be set up per the instructions in Software Setup. For recommendations regarding connecting to a 2.4 GHz network (e.g., in university classroom settings) see https://github.com/sparks-baird/self-driving-lab-

> > demo/discussions/83 and https://github.com/sparks-baird/self-driving-lab-demo/discussions/88. See also page 3.

> - Clarify up front, the sculpting wire is just a way to position the sensor, and will not be used to conduct electricity (correct?).

This has been clarified.

> - Make the YouTube video more prominent in the article, as I think this is much more natural than a PDF (no offense to STAR Protocols)...

This is a great suggestion. It has been added to both the summary and the data availability sections.

> - In the key resource table, instead of or in addition to the abbreviations, can you write out what the piece is, in plain terms? And why it is needed/its purpose? For example, CONN HEADER VERT. CBL USB2.0. are opaque names.

This is a great point. The abbreviations have largely been removed and replaced with more readable entries. Since the latest version includes some ANDs and ORs, to avoid extra confusion, the purpose has been left out. I'd prefer to add another column describing the purpose, but I think it needs to follow the key resource table format.

> - The images for example in steps 8-10 are very grainy and poor quality.

The image resolution has been increased where feasible.

> - For Bayesian optimization, is the code for this included?

A link has been added to both the analysis section and the data and code availability section.

Reviewer #2

> IN STAR-PROTOCOLS-D-23-00017 Baird and Sparks provide a wonderful example of a low cost tool that will allow educators to introduce their students to the concepts of a self-driving lab. The manuscript is lovingly written (and the accompanying YouTube video is excellent even if it does illustrate that the authors have relatively poor musical sensibilities).

Thank you for the positive feedback! We are glad you enjoyed the manuscript and the video, and the music stems in part from Sterling's past life as a breakdancer. Thanks for bearing through it 😉

> The authors carefully followed the Protocol Template with section name and timings. They included a Key Resources table that included direct links to a digikey order. I can't guarantee that those links work forever, but for right now they function.

Agreed, we hope the links will last, and we will try to check back periodically to ensure working links.

> The steps (as written) are reasonably easy to follow. I personally find that words don't do a great job of describing physical builds and that their manuscript could have been more like Ikea instructions to promote clarity. I would suggest that the YouTube video be explicitly called out as I found that to be very instructive.

I think making the build instructions more like Ikea instructions is a great idea. We will try to incorporate this into future manuscripts. Great suggestion about including the video. We have added that as the last sentence to the Summary section to make it prominent as well as to the data and code availability section.

> I really only have two major concerns for this protocol:

> 1. Folks will have versions of Python on their computer already and there may be some hesitancy around downloading Thonny and a new version of Python. I tried pip installing thonny and it did some things with my packages which may or may not have broken my Python Environment. Is there a MicroPython IDE that is compatible with Anaconda? This will make folks like me more comfortable. This might be a good split point (or an opportunity to remind folks that they should have (and use) an experimental environment for things like this.

This is a great point. From what I can tell, when Thonny is installed, it installs it's own Python version (for mine, it is installed at `C:\Users\<username>\AppData\Local\Programs\Thonny\python.exe`.

I use Miniconda instead of Anaconda distribution, so the package conflicts may not appear on mine, but I gave the `pip` installation procedure a try, and it seems to work OK in a fresh conda environment:

```
conda create -n sdl-demo-thonny python==3.10.*
conda activate sdl-demo-thonny
pip install self-driving-lab-demo thonny
thonny
```

Additional content has been added related to this.

> 2. I found sections on MongoDB and HiveMQ confusing in that they are labeled optional but the troubleshooting makes them almost seem mandatory? Likewise the YouTube video seems to make it mandatory. This part could be further clarified.

Thanks for the great suggestion. This has been clarified in both places as follows:

> > a. (Optional) Set up a MongoDB database backend. If ignored, the demo will function, just without logging data to a database (i.e., the user becomes responsible for saving the data on the client side).
> >
> > ...
> >
> > b. (Optional) Create your own HiveMQ instance. If this setup is ignored, the demo will function properly; however, the hardware commands and sensor data will be transmitted via a default HiveMQ instance for which the credentials are public. Setting up your own HiveMQ instance ensures that the data you transfer remains private and secure. Other MQTT brokers such as Mosquitto or Adafruit IO are available. At the time of writing, we recommend HiveMQ because it provides free instances with generous limits. Setting up a private MQTT broker is in line with best practices for internet of things (IoT) security.

At the time the video was made, the MongoDB data logging was not implemented, and test.mosquitto.org was being used as the MQTT broker, meaning anyone could be listening in and even sending commands to the device. The switch to HiveMQ was to allow for a free way to set up a private MQTT broker in order to follow best practices for IoT security.

> I think this is a really awesome project and it is unfortunate that I couldn't get all of the pieces together in time to have my students do an en masse build. I will continue to interact with the authors moving forward and can hopefully provide them

Thank you for the great review, and we look forward to our future interactions as well!