

RT-Cloud: A cloud-based software framework to simplify and standardize real-time fMRI

Grant Wallace^a, Stephen Polcyn^a, Paula P. Brooks^a, Anne C. Mennen^a, Ke Zhao^b, Paul S. Scotti^a, Sebastian Michelmann^a, Kai Li^c, Nicholas B. Turk-Browne^e, Jonathan D. Cohen^{a,d}, Kenneth A. Norman^{a,d,*}

^a Princeton Neuroscience Institute, Princeton University, Princeton, NJ, United States

^b Cognitive Science Program, University of Pennsylvania, Philadelphia, PA, United States

^c Department of Computer Science, Princeton University, Princeton, NJ, United States

^d Department of Psychology, Princeton University, Princeton, NJ, United States

^e Department of Psychology, Yale University, New Haven, CT, United States

ARTICLE INFO

Keywords:

Neurofeedback
Cloud-computing
Software-as-a-service

ABSTRACT

Real-time fMRI (RT-fMRI) neurofeedback has been shown to be effective in treating neuropsychiatric disorders and holds tremendous promise for future breakthroughs, both with regard to basic science and clinical applications. However, the prevalence of its use has been hampered by computing hardware requirements, the complexity of setting up and running an experiment, and a lack of standards that would foster collaboration. To address these issues, we have developed RT-Cloud (<https://github.com/brainiak/rt-cloud>), a flexible, cloud-based, open-source Python software package for the execution of RT-fMRI experiments. RT-Cloud uses standardized data formats and adaptable processing streams to support and expand open science in RT-fMRI research and applications. Cloud computing is a key enabling technology for advancing RT-fMRI because it eliminates the need for on-premise technical expertise and high-performance computing; this allows installation, configuration, and maintenance to be automated and done remotely. Furthermore, the scalability of cloud computing makes it easier to deploy computationally-demanding multivariate analyses in real time. In this paper, we describe how RT-Cloud has been integrated with open standards, including the Brain Imaging Data Structure (BIDS) standard and the OpenNeuro database, how it has been applied thus far, and our plans for further development and deployment of RT-Cloud in the coming years.

Introduction

Real-time functional magnetic resonance imaging (RT-fMRI) is an emerging technology that holds tremendous promise for breakthroughs in basic science and clinical applications. In contrast to traditional, offline fMRI analysis, RT-fMRI involves analyzing data while participants are still in the scanner, giving experimenters the ability to modify the stimuli or tasks that they present as a function of the participant's measured neural state. RT-fMRI can be used in neurofeedback designs, in which participants are given feedback on how well they are instantiating a target brain state, and they use this information to learn how to better instantiate that state (for a historical review of fMRI neurofeedback, see Linden et al., 2021; this review is part of a recent textbook on fMRI neurofeedback edited by Hampson, 2021). In another use of RT-fMRI, stimuli are modified as a function of brain activation, but par-

ticipants are not explicitly given the goal of maximizing the goodness of their brain state; for example, in a triggering design, the results of the fMRI analysis are used to determine *when* stimuli are presented (e.g., Yoo et al., 2012). RT-fMRI reverses the typical logic of fMRI – instead of viewing the fMRI measurements as a dependent variable (i.e., looking at the effects of a task manipulation on fMRI), the fMRI measurements can be viewed as the independent variable (i.e., you can look at the effect of instantiating a particular neural pattern on other behavioral and neural measurements; Turk-Browne, 2021).

Previous RT-fMRI studies have led to a wide range of interesting discoveries in healthy participants. For instance, participants given neurofeedback from the ventral tegmental area (VTA) while they thought about personalized motivational states were able to learn to upregulate VTA activity; this increase in the ability to upregulate VTA activity persisted after the end of neurofeedback training (MacInnes et al., 2016).

* Corresponding author.

E-mail address: knorman@princeton.edu (K.A. Norman).

<https://doi.org/10.1016/j.neuroimage.2022.119295>.

Received 26 January 2022; Accepted 9 May 2022

Available online 14 May 2022.

1053-8119/© 2022 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>)

In a different study, participants given neurofeedback were able to improve their ability to sustain attention (deBettencourt et al., 2015). Researchers have even been able to induce perceptual learning of a particular stimulus orientation without visual presentation of that orientation and without participants becoming aware of what was being trained (Shibata et al., 2011).

Clinical studies have also used fMRI neurofeedback to treat neuropsychiatric and neurodevelopmental disorders; for a comprehensive listing of these studies as of mid-2020, see Table 1 from Linden (2021), and for a recent review see Taschereau-Dumouchel et al. (2022). To give one example, depressed patients who underwent fMRI neurofeedback training to increase amygdala activity while recalling positive autobiographical memories showed a decrease in depressive symptoms; these effects were specific to when neurofeedback was based on amygdala activation vs. activation of a control region in parietal cortex (Young et al., 2017; for further discussion see Young et al., 2021, 2018). Notably, several clinical studies have obtained promising results using the Decoded Neurofeedback (DecNef) approach, in which participants are given feedback to boost activation of a particular neural pattern without being told what the neural pattern is (for recent reviews, see Shibata et al., 2019; Taschereau-Dumouchel et al., 2021; Watanabe et al., 2017). For example, one DecNef study showed that training individuals with snake or spider phobias to activate a pattern corresponding to snakes or spiders (respectively), in the absence of viewing the phobia-triggering stimuli, and without knowledge that the target neural patterns related to snakes or spiders, led to decreased skin conductance fear responses to these stimuli (Taschereau-Dumouchel et al., 2018; for an example of a similar approach to treating post-traumatic stress disorder, see Chiba et al., 2019). Other clinical studies have obtained promising results by providing feedback based on functional connectivity. For example, Ramot et al. (2017) used RT-fMRI neurofeedback in individuals with Autism Spectrum Disorder (ASD) to reinforce functional connectivity (i.e., correlation in fMRI timeseries) between brain regions that are underconnected in ASD relative to controls; the training led to increases in functional connectivity that lasted up to a year and were correlated with improvements in behavioral symptoms.

Challenges with RT-fMRI

Importantly, despite the strong potential of RT-fMRI, its uptake has been limited by several factors. First and foremost, setting up real-time analysis pipelines is technically challenging: A real-time communication bridge needs to be built to connect scripts across multiple processes so that an incoming DICOM image can be transferred and analyzed, and then participant feedback can be presented based on the analysis results in a timely fashion. Clinical sites in particular may lack the software engineering and IT expertise needed to assemble this pipeline. A second factor is that the computational complexity of fMRI analysis has escalated substantially over the past two decades. Whereas, previously, the field relied almost exclusively on univariate measures (e.g., average activation in a region of interest), researchers have increasingly come to rely on more computationally-demanding and sensitive multivariate analyses (e.g., pattern classifiers and functional alignment algorithms; Cohen et al., 2017). This increase in the use of multivariate methods has occurred both for offline analysis and also real-time analysis (e.g., deBettencourt et al., 2015, 2019; Iordan et al., 2020; LaConte et al., 2007; Shibata et al., 2011; Wang et al., 2016). A third factor relates to the lack of open standards for RT-fMRI research. Offline fMRI analysis has been the focus of significant efforts at standardization: for example, development of the Brain Imaging Data Structure (BIDS; Gorgolewski et al., 2016) and fMRIPrep (Esteban et al., 2019). However, these new standards generally have not been applied in the RT-fMRI domain. This lack of standards has led RT-fMRI researchers to use a wide variety of different (and incompatible) pipelines in their research, which in turn has made it more difficult to share work, reproduce results, and reuse components.

Addressing the challenges of RT-fMRI using cloud computing

As described above, many of the challenges of RT-fMRI revolve around building and deploying a set of coordinating software components and harnessing enough computing power to complete analysis in time. This general set of challenges is not unique to RT-fMRI and in fact is common among many computer applications including e-commerce, data analytics, AI, and web-based communication. A common theme in the past decade has been to harness the power of cloud computing to simplify, standardize and reduce the cost of creating and deploying applications.

A classic example of an application that has primarily moved to the cloud is email. Without the cloud, a company would need to install an email server in their server room, and then install email clients on all employee computers. Whenever there would be an application update, the IT group would need to push out the new email client to all employee laptops. Some percentage of users would encounter a problem because of an outdated OS, insufficient storage space, the wrong libraries, and so on. With email running in the cloud, the server runs in the cloud and the employees access their email through a web browser. There is no installation on each computer and employees can access their email from anywhere. If the email server becomes too slow, it can be scaled up instantly and more storage can be added as needed. This on-demand model is known as Software-as-a-Service, or SaaS.

Here, we describe RT-Cloud, a newly-developed, open-source software framework written in Python 3 that leverages cloud computing and SaaS to address the challenges of RT-fMRI (<https://github.com/brainiak/rt-cloud>). With RT-Cloud, the real-time fMRI analysis software is installed on cloud computers and is accessible from any web browser; only one lightweight software component is installed locally (to forward images up to the cloud). This setup makes it possible to run highly complex fMRI analyses in real time, even in situations where the scanning facility does not itself have extensive computing resources or IT expertise. To facilitate the sharing of pipelines and data, RT-Cloud also takes advantage of the BIDS data standard for fMRI, as described in the *Integration with BIDS* section below. In this section, we provide an overview of the key advantages that cloud computing and SaaS provide for RT-fMRI: ease of setup and maintenance, lowering of costs, ease of scaling, and accessibility from anywhere; in the section after this one, we describe the RT-Cloud framework in more detail.

- *Ease of setup and maintenance.* With cloud-based computing, all installation and maintenance are on a single cloud virtual machine (VM) image and that image is used to instantiate each VM at startup, thus ensuring that all instantiations are identical. This means that quality control can be addressed centrally, ensuring that the project meets strict and consistent standards for system functionality, library compatibility and security. In addition, the installation and maintenance can all be done remotely by a centralized team — and often just a single person — rather than separately at each facility, thus reducing IT costs and time. This is especially important when deploying in clinical settings where the availability of specialized hardware and technical staff to do software installations and maintenance may be limited.
- *Lowering of costs.* Cloud is a “pay for what you use” model. So, for example, two hours of cloud computer time to run a session will cost about \$1-\$2. Cloud savings occur because of the ability to spin up and spin down hardware on demand. Within our framework, cloud VMs are only running during the scanning session and then are stopped when the session is done. This on-demand nature eliminates the primary cost of owning hardware, which is equipment idle-time.
- *Ease of scaling.* Cloud computing also enables system scaling, both out and up. Scaling out is the addition of more of the same type of resource (i.e. more VMs) to accommodate more simultaneous studies.

This is important to enable large-scale deployment with simultaneous usage at multiple sites. Scaling up is the allocation of larger VM instances (faster, more cores, more memory) to accommodate higher processing demands of an individual experiment. This is very helpful as experimental designs and computational demands change. The result is that, rather than committing to a \$10,000 piece of hardware today, only to find it is insufficient tomorrow (or conversely is overpowered and thus mostly idle), you can simply rent the right VM size today and change it up or down at any time.

- **Accessibility from anywhere.** The Software-as-a-Service (SaaS) model is transforming many industries by simplifying application deployment; as noted in the email example above, this model involves installing an application on the cloud and having users access the application through a web browser, without installing libraries or software on their local computer or laptop. This has major advantages in the context of fMRI. Since MRI scanners are typically heavily booked and tightly scheduled, experimenters do not want to waste time in the control room logging in to software and entering session configurations. SaaS makes it possible for researchers to do these preliminaries outside of the control room on any web-accessible computer, so they will be ready to immediately begin the experiment when they get into the control room. It also makes it possible to install, maintain and test the experiment from a laptop outside of the control room, and then use the same interface to run the experiment, thus ensuring smooth operation. As improvements are made to the service, users can have access to those immediately, never having to wonder or check if their computer has enough memory, the right OS version, the right libraries, and so on.

The RT-Cloud framework

Overview of framework

The RT-Cloud software framework provides the basic infrastructure needed to run an RT-fMRI experiment. The framework wraps experiment-specific code that the researcher provides, thus providing a pluggable model that reduces the complexity and time of setting up and running an experiment. RT-Cloud was co-developed with the BrainIAK suite of Python tools for advanced fMRI analysis (<https://brainiak.org>; Kumar et al., 2021). Users can deploy BrainIAK analysis modules in their custom analysis code, or they can use other tools if they wish. During execution, the RT-Cloud framework handles details such as starting and stopping the analysis pipeline, getting fMRI images in real-time, and handling data communication between components such as for images, analysis results, and participants' responses.

The framework has two major components, the FileWatcher and the ProjectServer. The FileWatcher watches for and forwards DICOMs as they arrive from the MRI scanner, while the ProjectServer coordinates between the FileWatcher, the researcher's analysis script, and the feedback presentation shown to participants. In addition, the framework has two web interfaces: the Experiment Control web page, which allows the researcher to control the experiment session, and the Subject Feedback web page, which can be used to present stimuli to participants.

To elaborate, the FileWatcher runs on a computer in the control room and requires minimal processing power. Once parameters on the scanner computer are set so that DICOM images are made accessible, the FileWatcher registers for file-system notifications to watch for the arrival of new DICOM images from the scanner. Each new DICOM is read, converted to a BIDS format using the BIDS-Incremental system that we created for this purpose (see the *Integration with BIDS* section for more details), and forwarded to the ProjectServer using standard network protocols.

The ProjectServer is deployed on a system with enough processing power to quickly analyze the incoming brain-volume data in time for the participant to receive neurofeedback, in about 1 to 2 s. This component usually runs on the cloud but it can also run on a local computer or

cluster. As noted in the previous section, running this component in the cloud makes it possible to scale computing resources as a function of the complexity of the analysis, by either scaling up the VM instance or splitting the processing into parallel components across multiple VMs.

The experiment can be controlled by interacting with the ProjectServer via the Experiment Control web page, which is typically accessed from a laptop. This web page allows the researcher to change configuration settings, start and stop runs, and view analysis results in real time. The Subject Feedback web page is made visible to the participants – this web page can be flexibly used to present stimuli to participants, including (but not limited to) neurofeedback based on the results of the RT-fMRI analysis. Moreover, this web interface can also collect behavioral responses (e.g., button presses) from participants (see the *Integration with Experiment Scripting Packages* section below). Fig. 1 illustrates how the framework components fit together to complete the neurofeedback loop.

Integration with BIDS

To facilitate the re-use of existing pipelines and data, RT-Cloud supports BIDS, the leading standard for fMRI data (Gorgolewski et al., 2016). BIDS is supported by a wide variety of formatting and analysis tools and data repositories. For example, BIDS is used by the OpenNeuro database (Gorgolewski, Esteban, et al., 2017; Markiewicz et al., 2021), a large and growing repository of neuroscience datasets (incorporating fMRI and also other data types). BIDS has an automated and comprehensive validation tool that analyzes datasets for compliance and identifies issues (Gorgolewski et al., 2016). It also is the data format used by “BIDS Apps”, which are container-based applications with a standardized interface that work on BIDS-formatted datasets (Gorgolewski, Alfaro-Almagro, et al., 2017).

BIDS archives include brain volumes stored in NIfTI format, and have meta-data stored in separate “sidecar” files, typically with JSON or TSV (tab separated value) formatting. BIDS archives also have a standard directory structure and file naming convention. Included in the file names are “BIDS entities” such as the subject name, session, task, and run.

BIDS is designed as an archival standard (i.e. for data-at-rest), but RT-fMRI requires streams of data in order to process brain-volumes as they arrive from the scanner. An adaptation is required to support streaming data in a BIDS-compliant manner. Here, we introduce a data structure called the “BIDS-Incremental”, which is an in-memory BIDS archive with only one brain volume and associated meta-data in it. BIDS-Incrementals allow us to package and stream DICOM images one at a time as they arrive off the scanner and send them to the ProjectServer for analysis.

BIDS-Incrementals must hold the proper data structures to be compatible with a BIDS archive. These data structures include: the NIfTI image; a metadata dictionary describing the image; and several supporting data structures that map to files in a BIDS archive (namely, the events file that includes information about the stimuli presented to the participant, a README file with general information about the experiment, and a JSON file that describes the dataset). In addition, the data structure must be fast enough to support all operations well within the typical 1–2 second repetition time (TR) window in which a brain volume is acquired. There are three main operations that the software supports: 1) creating a BIDS-Incremental from a DICOM image; 2) appending a BIDS-Incremental to a BIDS archive; and 3) reading a BIDS-Incremental from a BIDS archive; these operations are illustrated in Fig. 2. The first two operations support streaming live data as it arrives from a scanner and accumulating it into a BIDS archive during processing. The third supports “replaying” or re-processing, through a real-time pipeline, data that have been previously collected and stored in a BIDS archive. This “replay” workflow is very useful for testing an analysis pipeline. Instead of collecting new data in real time, users can just take an existing dataset and run it through the pipeline to see how well everything is working (see section below on *Integration with OpenNeuro*).

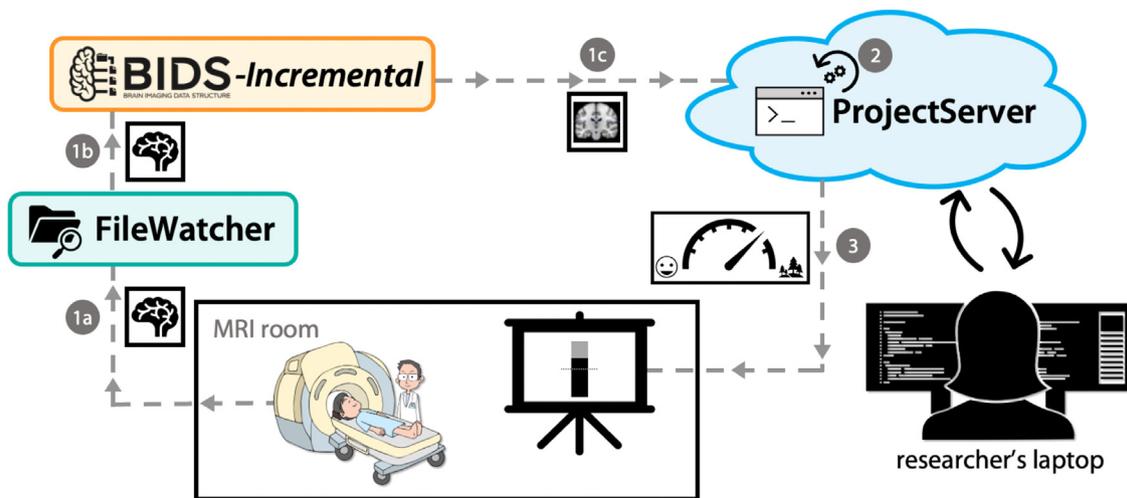


Fig. 1. Schematic of our RT-Cloud framework: (1a) The FileWatcher watches for new DICOM images arriving from the scanner, (1b) sends them to the BIDS-Incremental system for conversion to a BIDS-formatted NIFTI image, and (1c) then sends this image to the ProjectServer, running on the cloud. (2) The ProjectServer, which wraps the researcher's code, processes the NIFTI image and runs the analysis code to obtain a measure of the participant's brain state (e.g., whether they are attending to a face or a scene). The researcher accesses the cloud application from a browser page that can run on a laptop. Among many things, the researcher can initiate/finalize the session, change settings, and even observe the graph output of the analysis results from this browser page. (3) The analysis results are provided to the participant as neurofeedback presented on a display screen in the MRI room. Note that RT-Cloud can also be installed on a local computer or cluster node in lieu of using the cloud. Figure adapted with permission from Kumar et al. (2021).

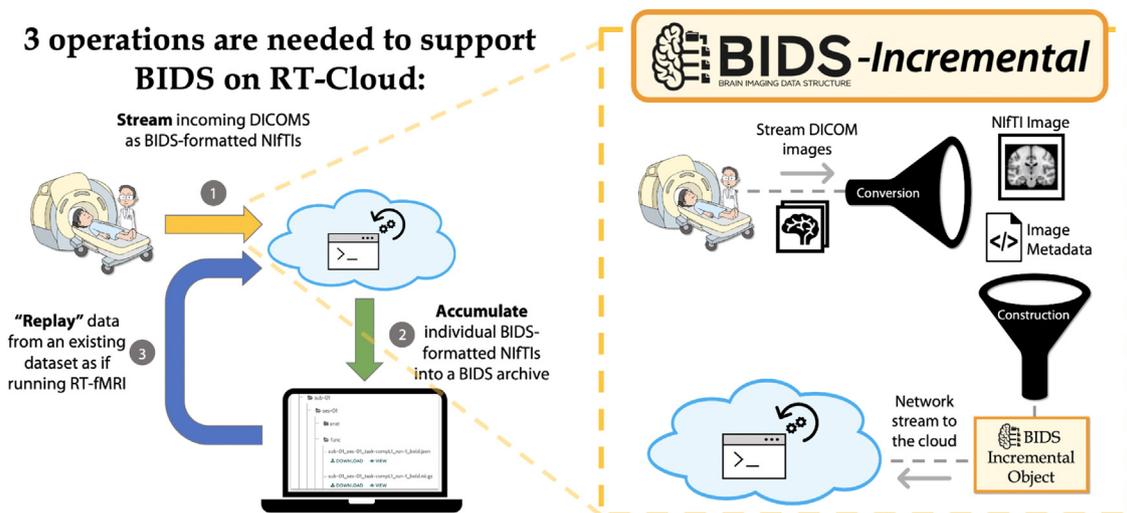


Fig. 2. BIDS Support: We have adapted the typical data-at-rest BIDS standard to real-time streaming by developing a BIDS-Incremental system. As data arrive from the scanner, we create single-volume BIDS archives that are streamed to the real-time analysis engine on the cloud. These single-volume archives can be appended in the cloud to create a BIDS archive that encompasses the entire run, and they can also be replayed one volume at a time to simulate a real-time experiment.

To satisfy real-time requirements, these BIDS-Incremental operations must be completed quickly, ideally within a couple tens of milliseconds to avoid impacting overall real-time deadlines. Initial implementations of the BIDS-Incremental relied on disk-backed operations, such as appending to or reading from an on-disk BIDS archive. However, as an experiment continues and accumulates data, the on-disk archive grows in size and eventually operations exceed a completion time threshold. To counter this issue, we leverage the fact that real-time fMRI users typically work one scanning run at a time; by caching a run's worth of BIDS data in-memory, we can optimize operations that are in the time-critical portion of an experiment. Other operations, like writing or reading a run's worth of data to or from disk, can be done before or after the time-sensitive section of a real-time fMRI workflow. We developed this mechanism into a data structure called a "BIDS Run", an in-memory cache of BIDS-Incrementals corresponding to one run in the experiment.

The above data structures that provide BIDS support within the framework leverage their implementation on PyBIDS, a software package produced by the BIDS Standard maintainers that provides a set of utilities for interacting with and manipulating BIDS data (Halchenko et al., 2020). In addition, NiBabel (Brett et al., 2020) and dcm2niix (Li et al., 2016) are used for NIFTI image handling.

Supporting BIDS within RT-Cloud has many benefits. Data that are collected and stored in BIDS format can be readily understood and used by other researchers and can be replayed through the RT-Cloud framework. In addition, if an RT-Cloud experiment is encapsulated as a BIDS App (see *Current and future directions* section below), then the full software environment needed to run the experiment is made available to users, including not only the analysis scripts, but also any libraries and configurations required. If a user shares both their BIDS App and their BIDS data with a second user, the second user will have everything needed to replay, validate, and modify the study.

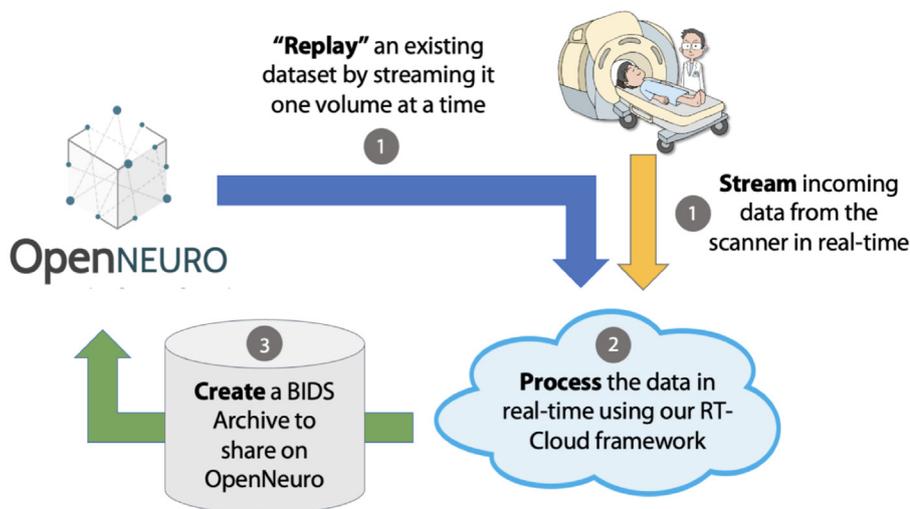


Fig. 3. OpenNeuro Integration: RT-Cloud, through its support for BIDS data, can download and stream datasets from OpenNeuro in order to test and validate experiment pipelines. In addition, the BIDS formatted data resulting from an experiment can be uploaded by the researcher to OpenNeuro to share with the neuroscience community.

Integration with openneuro

To support the “data replay” functionality described in the previous section for simulating RT-fMRI studies, we also added a feature that connects RT-Cloud with the OpenNeuro database (Gorgolewski, Esteban, et al., 2017; Markiewicz et al., 2021). With this connectivity, it is possible to stream any dataset stored on OpenNeuro through an RT-Cloud analysis pipeline. Data are processed using a consistent BIDS-Incremental format whether new or replayed (Fig. 3).

Specifically, we provide an OpenNeuroService component that will download datasets (or parts of datasets such as particular subjects or runs) from OpenNeuro and make them available for streaming; this OpenNeuroService component can be run on any computer (for example, a cloud VM). When testing or validating their experiment, researchers can connect to the OpenNeuroService component and specify the dataset accession number, subject name, session, and run number in order to stream that previously-collected data through their analysis pipeline.

This can be thought of, in an initial way, as a kind of “Netflix for neuroscience data”. The data are housed in the cloud and can be accessed, streamed, and processed on-demand. This has benefits not only for validating previous experiments but also for building and testing new experiments. Before an analysis pipeline is deployed in a “live” experiment, users can stream previously-collected data through it to test for processing errors and/or develop and benchmark new analysis approaches.

Integration with experiment scripting packages

All RT-fMRI studies need some way of controlling how the experiment unfolds (i.e., which stimuli to present to participants) as a function of the results of the real-time fMRI analysis. To accomplish this goal, RT-Cloud has been designed to integrate with behavioral feedback scripting frameworks like JsPsych (De Leeuw, 2015), PsychoPy (Peirce et al., 2019), and PsychToolbox (Kleiner et al., 2007). The ProjectServer of RT-Cloud has a websocket based communication layer that allows remote scripts to make a connection to the ProjectServer and receive analysis results. These results can then be used to adjust the stimuli or task displayed to the participant in the MRI scanner.

The ProjectServer can control how stimuli are displayed in several different ways depending on the requirements of the experiment. The most straightforward approach is to use the SubjectFeedback web page served up by RT-Cloud for stimulus presentation and behavioral data collection. Researchers taking this approach can use browser-based presentation toolboxes such as JsPsych (De Leeuw, 2015) to script their

experiment. In this use case, the JsPsych framework is served-up by RT-Cloud’s web server and runs directly in a web browser that is viewable by the participant in the MRI scanner. For this reason, it requires no installation for use, just opening and pointing a web browser to the ProjectServer URL. This is particularly convenient for deploying real-time experiments in clinical settings where computer hardware availability is uncertain. RT-Cloud provides a JsPsych module for receiving fMRI analysis results; by default, this module renders a DecNef style feedback display, in which the displayed circle radius indicates the correspondence to the desired neural state. This can easily be extended to other feedback types by extending the draw function within the example.

Another approach to displaying stimuli, which could work with almost any presentation system, is to set up the presentation software on a separate computer, and to configure the ProjectServer to write each analysis result back to the presentation computer as a small text file, with a filename corresponding to the trial and containing only one floating point value, i.e. the analysis result. The presentation script can watch for the creation of such files and read them to adjust the presentation feedback. As described in the *Real-world validation of the framework* section below, we have used this approach in multiple studies to interface RT-Cloud with the PsychToolbox stimulus presentation software. A related, more streamlined, approach is for the stimulus presentation software (running on a separate computer) to receive the analysis results directly over a websocket connected to the ProjectServer; any scripting method that has support for websockets can use this approach. For example, this approach can be used to send analysis results from the ProjectServer to experiment scripts that were built using the Python-based PsychoPy stimulus presentation system (Peirce et al., 2019).

Real-world validation of the framework

In this section, we discuss our experiences with real-world validation of the RT-Cloud framework. Thus far, two studies have been completed using previous versions of the RT-Cloud framework. The first study, referred to here as RT-Attention, was a clinical study conducted at Penn Medicine; the focus of this study was to train participants with major depressive disorder (MDD) to disengage their attention from negatively-valenced stimuli (Mennen et al., 2021). The second study, referred to here as GreenEyes, was conducted on healthy participants at Princeton University; this study aimed to bias participants toward a particular interpretation of an ambiguous story (Mennen et al., 2022).

In the RT-Attention study (Mennen et al., 2021), MDD participants ($N = 15$) and healthy controls ($N = 12$) were shown a superimposition of two images, a neutral scene and a negatively valenced face. They were asked to focus on the scene and ignore the face. A multivariate

classifier was trained to track how strongly participants were attending to the scene vs. the face. Feedback was provided by varying the relative visibility of the scene vs. the face: The more that participants got distracted and attended to the face (as measured by the classifier), the more visible the face became, and the less visible the scene became. This had the effect of externalizing and amplifying internal attentional lapses, making the task of judging the scene more difficult. The key dependent measure was how well participants were able to recover from these lapses and resume attending to the scene. We found that, at the outset of training, MDD patients were less able than healthy controls to recover from attentional lapses (i.e., their negative attention to the face was more “sticky”), but – by the end of training – MDD patients had significantly improved on this measure relative to controls.

This study was conducted at Penn Medicine with 27 participants each completing three neurofeedback training sessions across different days. The challenge was to deploy a RT-fMRI study in a clinical setting, where we had to ensure that our implementation of RT-fMRI did not disrupt any of the other clinical studies underway at the facility. After developing the study at Princeton, we used the RT-Cloud framework running in the cloud, in coordination with a local Linux computer, to deploy the study at the Penn Medicine imaging facility. Our RT-fMRI software was the first cloud-based application deployed by the Penn Medicine IT group, thus breaking ground on the administrative as well as technical requirements for a study of this type. This effort won a [Fierce Innovation Award](#).

The cloud framework was installed within Penn Medicine’s account on the HIPAA compliant Microsoft Azure Cloud and integrated with on-premise resources via a Virtual Private Network (VPN). The Penn IT team set up the virtual machine (VM) and did security scanning to ensure compliance. During sessions of the study, a version of the RT-Cloud server was running in the cloud and data were sent to it in the form of masked 2D-arrays of DICOM volume data (this study used an earlier version of the framework that pre-dated our use of BIDS-Incrementals). The analysis script in the cloud did smoothing, high-pass filtering, z-scoring and classification of the brain image data. The runs were divided into interleaved blocks of trials used for classification training (where participants attended to faces or scenes and neurofeedback was not provided) and neurofeedback trials. The cloud framework supported both classifier training and the use of the trained classifier during neurofeedback, as specified by a session configuration file. The classification scores were sent back to the presentation computer in the control room, saved as a text file, and then read by a PsychToolbox script to provide participants with feedback by altering the relative visibility of the face and the scene.

In the GreenEyes study ([Mennen et al., 2022](#)), participants listened to an ambiguous story and were given neurofeedback in order to steer their interpretation to one of two interpretations (randomly assigned for each participant); the story stimulus used here was the same as the stimulus used in [Yeshurun et al. \(2017\)](#). The study involved two one-hour scanning sessions per participant ($N = 20$). Here, the ProjectServer was run on a VM in the Microsoft Azure cloud. During scanning sessions, as DICOM images arrived, they were anonymized and sent to the cloud server for processing. On the cloud, the BOLD data were registered to MNI space, preprocessed, and then processed using a Shared Response Model ([Chen et al., 2015](#)) and then a support vector machine classifier to produce the neurofeedback values. These values were returned to the presentation computer in the control room and written to a text file which was read and used by a PsychToolbox script to update the feedback display. Control of the ProjectServer was accomplished via a web page that was accessed by the researcher’s laptop.

The following sub-sections describe key take-away points from these real-world validation studies.

Timing and responsiveness

In these real-time experiments, a new brain scan was performed every 1.5 or 2 s (i.e. the TR). Thus, we had to read and process the scan im-

age and provide the classification feedback within that time window. On the Siemens scanning system used in these studies, it took about 700 ms from the completion of the scan until the reconstructed DICOM image was available on the scanner computer’s disk. Our FileWatcher then read the DICOM and transferred it to the cloud, taking about 100 ms. The classification was performed in 200 ms. The classification result was returned from the cloud in 20 ms. Thus the total processing time was slightly more than 1 second. Using a TR of 2 s (a typical TR time in RT-fMRI experiments) gives plenty of leeway. Reliability of network transfer to the cloud has not been an issue for us in the more than 100 scanning sessions that we have run as part of the aforementioned two studies. We do note that a network outage would of course prevent running a session; however, it is possible that such an outage would prevent even an on-premise real-time study depending on where the outage occurs. In our validation studies, there were very rare occasions where processing did not complete in time for a particular TR or trial. To handle this, we simply configured the experiment presentation script so that – if the feedback value was missing – the script delivered the same feedback value as on the previous TR or trial. Note that this error-handling is up to the user; if a user wanted to handle missing feedback in a different way, they could set up the script differently.

Network bandwidth requirements

An RT-fMRI session requires about 2 Mbps (Megabits per second) average bandwidth and 20–40 Mbps of peak upload bandwidth. A typical session sends a 0.5 megabyte (MB) image every 2 s and we want the image transfer to complete in 100–200 ms. Having a 20–40 Mbps peak upload bandwidth allows the image transfer time constraint to be met. The average bandwidth is much lower than the peak because of idle time between images. The reply from the cloud-based ProjectServer is usually only a few bytes (such as a floating point number) and requires minimal bandwidth.

All of the facilities where we have deployed the framework have had adequate bandwidth and low enough latency to the cloud to support these workloads, and we expect that these bandwidth requirements will be well covered at typical scanning facilities. In general, network infrastructure bandwidths have been increasing to meet on-demand video streaming and as of 2020, Speedtest.net estimates the average U.S. fixed internet speed is 135 Mbps download / 52 Mbps upload. HealthIT.gov recommends that hospitals have a minimum of 100 Mbps service, and that academic / large medical centers have 1000 Mbps.

Data privacy and security

Working with medical data presents concerns and challenges in maintaining data security and privacy. Additional concerns are often raised when using cloud infrastructure. One of the first steps is to ensure that the computational and storage infrastructure are certified to be HIPAA compliant. Both Microsoft Azure and Amazon AWS have HIPAA certification. For the Penn Medicine-based RT-Attention study, we chose Microsoft Azure, used anonymized image data with only the ROI voxels being sent for processing, used a private VPN network for data communication, used encrypted storage in the cloud, and deleted data from the cloud after each session. This provided a strong security model and met IRB approval. For the GreenEyes study, we used similar measures but sent anonymized DICOM images (for which all identifying header information was removed).

Over time, cloud-based infrastructure will likely become more trusted than on-premise infrastructure. Typically, on-premise facilities have a limited IT staff responsible for securing and patching all computers. Cloud vendors have a much larger IT staff and automated tools and will be more on top of security patches and issues. The number of high-profile security breaches at large companies with on-premise data illustrates this point. The aim is, of course, to use the best secu-

practices available and cloud providers are incentivized to provide state-of-the-art solutions that can be employed.

Cloud costs

Costs associated with cloud computing can include VM rental, data storage, data transfer, and other services like persistent IP addresses, logging etc. Cloud data transfers typically have asymmetric costs (free to transfer data in, charge to read data out). Azure is free to transfer in and provides 5GB/month free outbound transfer and charges \$0.08/GB thereafter. In studies using RT-Cloud, image data are transferred in (free inbound), and analysis results (which are typically very small files) are transferred out; at the end of an experiment, data archive files can be transferred out to a more permanent storage site (possibly cloud storage, possibly on-premise or in a data repository like OpenNeuro). The final archive transfer out may be a couple of GBs in size, costing about \$0.20. As a cost example, our Penn Medicine based RT-Attention study used Azure D16s VMs which cost about \$0.80/hour, and our monthly bill was typically much less than \$100, with a few dollars of it being spent on network transfer.

Adapting the framework to different computing environments

The purpose of RT-Cloud is to make RT-fMRI easily accessible to researchers. This includes handling core functionality like watching for new DICOM images, handling real-time scheduling, providing pathways for feedback, and making configuration and user interaction easier. Using cloud computing can be a simpler and cheaper option, but some facilities already have computer clusters or other resources in place, and our framework is adaptable to fit into many different compute configurations. Some configurations in which our software framework have been deployed include:

- At Penn Medicine, for the RT-Attention study, we split the ProjectServer into two parts: The classification model ran in the cloud, and the other parts of the ProjectServer ran on a control-room computer. Participant feedback was provided using MATLAB PsychToolbox (running with the ProjectServer on the control-room computer), with classification results being passed in via small text files.
- At Princeton University, for the GreenEyes study, we ran the ProjectServer in the cloud.

Participant feedback was provided using PsychToolbox, running on a separate computer.

- In other studies that are currently underway at Yale University, the ProjectServer is run on a local compute cluster, and analysis results are sent via websockets to a PsychoPy script running on a separate computer that provides participant feedback.
- For testing and development, we often run everything on a local computer.

Current and future directions

We are planning to add several features to RT-Cloud in the coming years. One important component will be the ability to wrap an RT-Cloud experiment as a BIDS App. BIDS Apps are containerized (and thus self-contained) applications that operate on BIDS data and have a common set of invocation parameters. An experiment packaged as a BIDS App would contain all the libraries and configurations needed to run the experiment on any computer that can run a Docker or Singularity container. This will complement the BIDS data standardization added to the framework, allowing researchers to share both data and full execution environments in order to reproduce, validate and extend each other's work.

Using the BIDS App framework, we will pre-package a set of real-time experiments. These will be modeled on paradigms that have

yielded good results in the past, and will be representative of a range of techniques. This will give new researchers a reference and quick starting point for building new experiments, and will also make it easy to deploy proven techniques to clinical settings.

We also plan to add support for running multiple analysis models in parallel within the ProjectServer. For example, researchers could deploy a fast analysis model to guarantee a real-time result alongside a slower but more accurate model. Another potential use case could be to run MR-based eyetracking (Frey et al., 2021) alongside an analysis model (e.g., to get an online measure of whether participants are fixating on stimuli).

Conclusion

In summary, the RT-Cloud framework makes it possible to scale RT-fMRI to more computationally-intensive analyses, while also simplifying the deployment of these analyses, by making it possible to run them in situations where local computing hardware or computing expertise are lacking. There are several other packages for running RT-fMRI studies (e.g., Basilio et al., 2015; Cox, 1996; Goebel et al., 2006; Heunis et al., 2018; Hinds et al., 2011; Koush et al., 2017; Shibata, 2012; for a more complete list see Sulzer, 2021). However, ours is unique in its use of the cloud and SaaS. As described above, our cloud approach has several important benefits, relating to ease of installation and maintenance, reduced cost, ease of scaling, and accessibility from anywhere. Our use of open-source Python code makes the framework extensible by experts and allows for community-based development, and our integration with the BIDS standard facilitates pipeline sharing and data sharing. Taken together, we hope that these developments will expand the use of RT-fMRI to a much wider community.

Acknowledgements

Funding for this project was provided by NIH grant RF1 MH125318 awarded to K.A.N. and J.D.C. Funding was also provided by Intel Labs (<https://www.intel.com/intellabs>) and by a grant from the John Templeton Foundation to J.D.C., K.A.N., and N.B.T-B. S.M. was funded by Deutsche Forschungsgemeinschaft (DFG), project 437219953. The opinions expressed in this publication are those of the authors and do not necessarily reflect the views of the funders.

References

- Basilio, R., Garrido, G.J., Sato, J.R., Hoefle, S., Melo, B.R., Pamplona, F.A., Zahn, R., Moll, J., 2015. Friend engine framework: a real time neurofeedback client-server system for neuroimaging studies. *Front. Behav. Neurosci.* 9, 3.
- Brett, M., Markiewicz, C.J., Hanke, M., Côté, M.-A., Cipollini, B., McCarthy, P., Jarecka, D., Cheng, C.P., Halchenko, Y.O., Cottaar, M., Larson, E., Ghosh, S., Wassermann, D., Gerhard, S., Lee, G.R., Wang, H.-T., Kastman, E., Kaczmarzyk, J., Guidotti, R.
- Chen, P.-H., Chen, J., Yeshurun, Y., Hasson, U., Haxby, J.V., Ramadge, P.J., 2015. A reduced-dimension fMRI shared response model. *NIPS* 28, 460–468.
- Chiba, T., Kanazawa, T., Koizumi, A., Ide, K., Taschereau-Dumouchel, V., Boku, S., Hishimoto, A., Shirakawa, M., Sora, I., Lau, H., et al., 2019. Current status of neurofeedback for post-traumatic stress disorder: a systematic review and the possibility of decoded neurofeedback. *Front. Hum. Neurosci.* 13, 233.
- Cohen, J.D., Daw, N., Engelhardt, B., Hasson, U., Li, K., Niv, Y., Norman, K.A., Pillow, J., Ramadge, P.J., Turk-Browne, N.B., et al., 2017. Computational approaches to fMRI analysis. *Nat. Neurosci.* 20 (3), 304–313.
- Cox, R.W., 1996. AFNI: software for analysis and visualization of functional magnetic resonance neuroimages. *Comput. Biomed. Res.* 29 (3), 162–173.
- De Leeuw, J.R., 2015. jsPsych: a JavaScript library for creating behavioral experiments in a web browser. *Behav. Res. Methods* 47 (1), 1–12.
- deBettencourt, M.T., Cohen, J.D., Lee, R.F., Norman, K.A., Turk-Browne, N.B., 2015. Closed-loop training of attention with real-time brain imaging. *Nat. Neurosci.* 18 (3), 470–475.
- deBettencourt, M.T., Turk-Browne, N.B., Norman, K.A., 2019. Neurofeedback helps to reveal a relationship between context reinstatement and memory retrieval. *Neuroimage* 200, 292–301.
- Esteban, O., Markiewicz, C.J., Blair, R.W., Moodie, C.A., Isik, A.I., Erramuzpe, A., Kent, J.D., Goncalves, M., DuPre, E., Snyder, M., et al., 2019. fMRIPrep: a robust preprocessing pipeline for functional MRI. *Nat. Methods* 16 (1), 111–116.
- Frey, M., Nau, M., Doeller, C.F., 2021. Magnetic resonance-based eye tracking using deep neural networks. *Nat. Neurosci.*

- Goebel, R., Esposito, F., Formisano, E., 2006. Analysis of FIAC data with BrainVoyager QX: from single-subject to cortically aligned group GLM analysis and self-organizing group ICA. *Hum. Brain Mapp.* 27 (5), 392–401.
- Gorgolewski, K.J., Alfaro-Almagro, F., Auer, T., Bellec, P., Capotà, M., Chakravarty, M.M., Churchill, N.W., Cohen, A.L., Craddock, R.C., Devenyi, G.A., et al., 2017a. BIDS apps: improving ease of use accessibility and reproducibility of neuroimaging data analysis methods. *PLoS Comput. Biol.* 13 (3), e1005209.
- Gorgolewski, K.J., Auer, T., Calhoun, V.D., Craddock, R.C., Das, S., Duff, E.P., Flandin, G., Ghosh, S.S., Glatard, T., Halchenko, Y.O., et al., 2016. The brain imaging data structure a format for organizing and describing outputs of neuroimaging experiments. *Sci Data* 3 (1), 1–9.
- Gorgolewski, K.J., Esteban, O., Schaefer, G., Wandell, B., Poldrack, R., 2017b. OpenNeuro—A free online platform for sharing and analysis of neuroimaging data. Organization for human brain mapping. Vancouver Canada 1677 (2).
- Halchenko, Y., Goncalves, M., di Oleggio Castello, M.V., Ghosh, S., Salo, T., Hanke, M., Dae Velasco, P., Kent, J., Brett, M., Amlien, I., Gorgolewski, C., Lukas, D.C., Markiewicz, C., Tilley, S., Kaczmarzyk, J., Stadler, J., Kim, S., Kahn, A., Szczepanik, M., 2020. Nipy/heudiconv v0.9.0 doi:10.5281/zenodo.4390433.
- Hampson, M., 2021. fMRI Neurofeedback. Elsevier.
- Heunis, S., Besseling, R., Lamerichs, R., de Louw, A., Breeuwer, M., Aldenkamp, B., Bergmans, J., 2018. Neu3CA-RT: a framework for real-time fMRI analysis. *Psychiatry Research: Neuroimaging* 282, 90–102.
- Hinds, O., Ghosh, S., Thompson, T.W., Yoo, J.J., Whitfield-Gabrieli, S., Triantafyllou, C., Gabrieli, J.D., 2011. Computing moment-to-moment BOLD activation for real-time neurofeedback. *Neuroimage* 54 (1), 361–368.
- Jordan, M.C., Ritvo, V.J., Norman, K.A., Turk-Browne, N.B., Cohen, J.D., 2020. Sculpting new visual concepts into the human brain. *bioRxiv*.
- Kleiner, M., Brainard, D., Pelli, D., 2007. What's new in psychtoolbox-3?
- Koush, Y., Ashburner, J., Prilepin, E., Sladky R. Zeidman, P., Bibikov, S., Scharnowski, F., Nikonorov, A., Van De Ville, D., 2017. OpenNFT: an open-source Python/Matlab framework for real-time fMRI neurofeedback training based on activity connectivity and multivariate pattern analysis. *Neuroimage* 156, 489–503.
- Kumar, M., Anderson, M., Antony, J., Baldassano, C., Brooks, P.P., Cai, M.B., Chen, P.-H.C., Ellis, C.T., Henselman-Petrusek, G., Huberdeau, D., et al., 2021. BrainIAK: the brain imaging analysis kit. *Aperture Neuro*. <https://www.doi.org/10.52294/31bb5b68-2184-411b-8c00-a1dacb61e1da>.
- LaConte, S.M., Peltier, S.J., Hu, X.P., 2007. Real-time fMRI using brain-state classification. *Hum. Brain Mapp.* 28 (10), 1033–1044.
- Li, X., Morgan, P.S., Ashburner, J., Smith, J., Rorden, C., 2016. The first step for neuroimaging data analysis: DICOM to NIFTI conversion. *J. Neurosci. Methods* 264, 47–56.
- Linden, D., 2021. Introduction to clinical section. In: Hampson, M. (Ed.), fMRI Neurofeedback. Elsevier, pp. 161–169.
- Linden, D., Goebel, R., Weiskopf, N., 2021. A brief history of real-time fMRI neurofeedback. In: Hampson, M. (Ed.), fMRI Neurofeedback. Elsevier, pp. 1–19.
- MacInnes, J.J., Dickerson, K.C., Chen, N.-k., Adcock, R.A., 2016. Cognitive neurostimulation: learning to volitionally sustain ventral tegmental area activation. *Neuron* 89 (6), 1331–1342.
- Markiewicz, C.J., Gorgolewski, K.J., Feingold, F., Blair, R., Halchenko, Y.O., Miller, E., Hardcastle, N., Wexler, J., Esteban, O., Goncalves, M., et al., 2021. The OpenNeuro resource for sharing of neuroscience data. *Elife* 10, e71774.
- Mennen, A.C., Nastase, S.A., Yeshurun, Y., Hasson, U., Norman, K.A., 2022. Real-time neurofeedback to alter interpretations of a naturalistic narrative. *bioRxiv*.
- Mennen, A.C., Turk-Browne, N.B., Wallace, G., Seok, D., Jaganjac, A., Stock, J., DeBettencourt, M.T., Cohen, J.D., Norman, K.A., Sheline, Y.I., 2021. Cloud-based functional magnetic resonance imaging neurofeedback to reduce the negative attentional bias in depression: a proof-of-concept study. *Biol. Psychiatry: Cognit. Neurosci. Neuroimag.* 6 (4), 490–497.
- Peirce, J., Gray, J.R., Simpson, S., MacAskill, M., Höchenberger, R., Sogo, H., Kastman, E., Lindeløv, J.K., 2019. Psychopy2: experiments in behavior made easy. *Behav Res Methods* 51 (1), 195–203.
- Ramot, M., Kimmich, S., Gonzalez-Castillo, J., Roopchansingh, V., Popal, H., White, E., Gotts, S.J., Martin, A., 2017. Direct modulation of aberrant brain network connectivity through real-time neurofeedback. *Elife* 6, e28974.
- Shibata, K., 2012. A new neuroscientific approach using decoded neurofeedback (DecNef). *Rinsho shinkeigaku= Clin. Neurol.* 52 (11), 1185–1187.
- Shibata, K., Lisi, G., Cortese, A., Watanabe, T., Sasaki, Y., Kawato, M., 2019. Toward a comprehensive understanding of the neural mechanisms of decoded neurofeedback. *Neuroimage* 188, 539–556.
- Shibata, K., Watanabe, T., Sasaki, Y., Kawato, M., 2011. Perceptual learning incepted by decoded fMRI neurofeedback without stimulus presentation. *Science* 334 (6061), 1413–1415.
- Sulzer, J.S., 2021. Resource reference for fMRI neurofeedback researchers. In: Hampson, M. (Ed.), fMRI Neurofeedback. Elsevier, pp. 333–338 pp..
- Taschereau-Dumouchel, V., Cortese, A., Chiba, T., Knotts, J., Kawato, M., Lau, H., 2018. Towards an unconscious neural reinforcement intervention for common fears. *Proc. Natl. Acad. Sci.* 115 (13), 3470–3475.
- Taschereau-Dumouchel, V., Cortese, A., Lau, H., Kawato, M., 2021. Conducting decoded neurofeedback studies. *Soc. Cogn. Affect. Neurosci.* 16 (8), 838–848.
- Turk-Browne, N.B., 2021. Introduction to basic science section. In: Hampson, M. (Ed.), fMRI Neurofeedback. Elsevier, pp. 81–83.
- Wang, Y., Keller, B., Capota, M., Anderson, M.J., Sundaram, N., Cohen, J.D., Li, K., Turk-Browne, N.B., Willke, T.L., 2016. Real-time full correlation matrix analysis of fMRI data. In: 2016 IEEE International Conference on Big Data (Big Data), pp. 1242–1251.
- Watanabe, T., Sasaki, Y., Shibata, K., Kawato, M., 2017. Advances in fMRI real-time neurofeedback. *Trends Cogn. Sci. (Regul. Ed.)* 21 (12), 997–1010.
- Yeshurun, Y., Swanson, S., Simony, E., Chen, J., Lazaridi, C., Honey, C.J., Hasson, U., 2017. Same story different story: the neural representation of interpretive frameworks. *Psychol. Sci.* 28 (3), 307–319.
- Yoo, J.J., Hinds O. Ofen, N., Thompson, T.W., Whitfield-Gabrieli, S., Triantafyllou, C., Gabrieli, J.D., 2012. When the brain is prepared to learn: enhancing human learning using real-time fMRI. *Neuroimage* 59 (1), 846–852.
- Young, K.D., Fine, N., Hendler, T., 2021. fMRI neurofeedback for disorders of emotion regulation. In: Hampson, M. (Ed.), fMRI Neurofeedback. Elsevier, pp. 187–205.
- Young, K.D., Siegle, G.J., Zotev, V., Phillips R. Misaki, M., Yuan, H., Drevets, W.C., Bodurka, J., 2017. Randomized clinical trial of real-time fMRI amygdala neurofeedback for major depressive disorder: effects on symptoms and autobiographical memory recall. *Am. J. Psychiatry* 174 (8), 748–755.
- Young, K.D., Zotev, V., Phillips R. Misaki, M., Drevets, W.C., Bodurka, J., 2018. Amygdala real-time functional magnetic resonance imaging neurofeedback for major depressive disorder: a review. *Psychiatry Clin. Neurosci.* 72 (7), 466–481.
- Taschereau-Dumouchel, V., Cushing, C., Lau, H., 2022. Real-time fMRI in the treatment of mental health disorders. *Annual Review of Clinical Psychology* 18, 125–154.