1 DataLad-Dataverse integration

Benjamin Poldrack[1], Jianxiao Wu[1,2], Kelvin Sarink[3], Christopher J. Markiewicz [4], Alexander Q. Waite [1], Eliana Nicolaisen-Sobesky[1], Shammi More[1], Johanna Bayer[5,6], Jan Ernsting[3,7], Adina S. Wagner[1], Roza G. Bayrak [8], Laura K. Waite[1], Michael Hanke[1,2], Nadine Spychala [9]

1. Institute of Neuroscience and Medicine, Research Centre Jülich, Jülich, Germany, 2. Medical Faculty, Heinrich Heine University Düsseldorf, 3. University of Münster, Institute for Translational Psychiatry, Münster, Germany, 4. Stanford University, 5. The University of Melbourne, Melbourne, Australia, 6. Orygen Youth Health, Melbourne, Australia, 7. University of Münster, Faculty of Mathematics and Computer Science, Münster, Germany, 8. Vanderbilt University, Nashville, TN USA, 9. Department of Informatics, University of Sussex, United Kingdom

The FAIR principles (Wilkinson et al., 2016) advocate to ensure and increase the Findability, Accessibility, Interoperability, and Reusability of research data in order to maximize their impact. Many open source software tools and services facilitate this aim. Among them is the Dataverse project (King, 2007). Dataverse is open source software for storing and sharing research data, providing technical means for public distribution and archival of digital research data, and their annotation with structured metadata. It is employed by dozens of private or public institutions worldwide for research data management and data publication. DataLad (Halchenko et al., 2021), similarly, is an open source tool for data management and data publication. It provides Git- and git-annex based data versioning, provenance tracking, and decentral data distribution as its core features. One of its central development drivers is to provide streamlined interoperability with popular data hosting services to both simplify and robustify data publication and data consumption in a decentralized research data management system (Hanke, Pestilli et al., 2021). Past developments include integrations with the open science framework (Hanke, Poldrack et al., 2021) or webdav-based services such as sciebo, nextcloud, or the European Open Science Cloud (Halchenko et al., n.d.).

In this hackathon project, we created a proof-of-principle integration of DataLad with Dataverse in the form of the Python package `datalad-dataverse` (github.com/datalad/datalad-dataverse). From a technical perspective, main achievements include the implementation of a git-annex special remote protocol for communicating with Dataverse instances, a new `create-sibling-dataverse` command that is added to the DataLad command-line and Python API by the `datalad-dataverse` extension, and standard research software engineering aspects of scientific software such as unit tests, continuous integration, and documentation.

From a research data management and user perspective, this development equips DataLad users with the ability to programatically create Dataverse datasets (containers for research data and their metadata on Dataverse) from DataLad datasets (DataLad's Git-repository-based core data structure) in different usage modes. Subsequently, DataLad dataset contents, its version history, or both can be published to the Dataverse dataset via a 'datalad push' command. Furthermore, published DataLad datasets can be consumed from Dataverse with a `datalad clone` call. A mode parameter configures whether Git version history, version controlled file content, or both are published and determines which of several representations the Dataverse dataset takes. A proof-of-principle implementation for metadata annotation allows users to supply metadata in JSON format, but does not obstruct later or additional manual metadata annotation via Dataverse's web interface.

Overall, this project delivered the groundwork for further extending and streamlining data deposition and consumption in the DataLad ecosystem. With DataLad-Dataverse interoperability, users gain easy additional means for data publication, archival, distribution, and retrieval. Post-Brainhack development aims to mature the current alpha version of the software into an initial v0.1 release and distribute it via standard Python package indices.

References

Halchenko, Y. O., Hanke, M., Heunis, S., Markiewicz, C. J., Mönch, C., Poldrack, B., … Wodder, J. T., II. (n.d.). DataLad-next extension. Retrieved from https://github.com/datalad/datalad-next

Halchenko, Y. O., Meyer, K., Poldrack, B., Solanky, D. S., Wagner, A. S., Gors, J., … Hanke, M. (2021). Datalad: Distributed system for joint management of code, data, and their relationship. *Journal of Open Source Software*, *6*(63), 3262. doi:10.21105/joss.03262

Hanke, M., Pestilli, F., Wagner, A. S., Markiewicz, C. J., Poline, J.-B. & Halchenko, Y. O. (2021). In defense of decentralized research data management. *Neuroforum*, *27*(1), 17–25. Publisher: De Gruyter Section: Neuroforum. doi:10.1515/nf-2020-0037

Hanke, M., Poldrack, B., Wagner, A. S., Huijser, D., Sahoo, A. K., Boos, M., ... Appelhoff, S. (2021). Datalad/datalad-osf: Cleanup (Version 0.2.3). doi:10.5281/zenodo.4572455

King, G. (2007). An introduction to the dataverse network as an infrastructure for data sharing. Sage Publications Sage CA: Los Angeles, CA.

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., ... Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data*, *3*(1), 160018. doi:10.1038/sdata.2016.18

2 MOSAIC for VASO fMRI

Renzo (Laurentius) Huber[1], Remi Gau[2], Rüdiger Stirnberg[3], Philipp Ehses[3], Ömer Faruk Gülban[1,4], Benedikt A Poser[1]

1. Faculty of Psychology and Neuroscience, Maastricht University, Maastricht, The Netherlands, 2. Institute of Psychology, Université Catholique de Louvain, Louvain-la-Neuve, Belgium and Institute of Neuroscience, Université Catholique de Louvain, Louvain-la-Neuve, Belgium, 3 German Center for Neurodegenerative Diseases (DZNE), Bonn, Germany, 4. Brain Innovation, Maastricht, The Netherlands.

Vascular Space Occupancy (VASO) is a functional magnetic resonance imaging (fMRI) method that is used for high-resolution cortical layer-specific imaging (Huber et al., 2021). Currently, the most popular sequence for VASO at modern SIEMENS scanners is the one by Stirnberg and Stöcker (2021) from the DZNE in Bonn, which is employed at more than 30 research labs worldwide. This sequence concomitantly acquires fMRI BOLD and blood volume signals. In the SIEMENS' reconstruction pipeline, these two complementary fMRI contrasts are mixed together within the same time series, making the outputs counter-intuitive for users. Specifically:

- The 'raw' NIfTI converted time-series are not BIDS compatible (see https://github.com/bids-standard/bids-specification/issues/1001).
- The order of odd and even BOLD and VASO image TRs is unprincipled, making the ordering dependent on the specific implementation of NIfTI converters.

Workarounds with 3D distortion correction, results in interpolation artifacts. Alternative workarounds without MOSAIC decorators result in unnecessarily large data sizes.

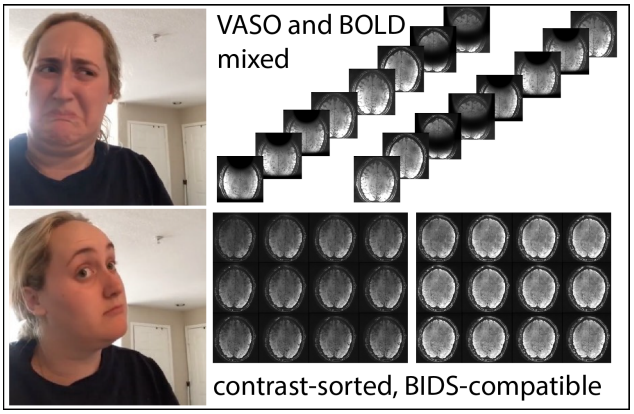


Figure 1: Previously, most VASO sequences provided unsorted image series of MRI contrasts. This was not BIDS compatible and could suffer from gradient non-linearity artifacts in the scanner's MR-reconstruction pipeline. In Brainhack 2022, we adapted the SIEMENS reconstruction and to sort volume series by fMRI contrasts. This is BIDS compatible and does not require non-linearity corrections.

In the previous Brainhack (Gau et al., 2021), we extended the existing 3D-MOSAIC functor that was previously developed by Benedikt Poser and Philipp Ehses. This functor had been previously used to sort volumes of images by dimensions of echo-times, by RF-channels, and by magnitude and phase signals. In this Brainhack, we successfully extended and validated this functor to also support the dimensionality of SETs (that is representing BOLD and VASO contrast).

We are happy to share the compiled SIEMENS ICE (Image Calculation Environment) functor that does this sorting. Current VASO users, who want to upgrade their reconstruction pipeline to get the MOSAIC sorting feature too, can reach out to Renzo Huber (RenzoHuber@gmail.com) or Rüdiger Stirnberg (Ruediger.Stirnberg@dzne.de).

Furthermore, Remi Gau, generated a template dataset that exemplifies how one could to store layer-fMRI VASO data. This includes all the meta data for 'raw' and 'derivatives'. Link to this VASO fMRI BIDS demo: https://gin.g-node.org/RemiGau/ds003216/src/bids_demo.

Acknowledgements: We thank Chris Rodgers for instructions on how to overwrite existing reconstruction binaries on the SIEMENS scanner without rebooting. We thank David Feinberg, Alex

Beckett and Samantha Ma for helping in testing the new reconstruction binaries at the Feinbergatron scanner in Berkeley via remote scanning. We thank Maastricht University Faculty of Psychology and Neuroscience for supporting this project with 2.5 hours of 'development scan time'.

References

Gau, R., Noble, S., Heuer, K., Bottenhorn, K. L., Bilgin, I. P., Yang, Y. F., ... Zuo, X. N. (2021). Brainhack: Developing a culture of open, inclusive, community-driven neuroscience. *Neuron*, *109*(11), 1769–1775. doi:10.1016/j.neuron.2021.04.001

Huber, L., Finn, E. S., Chai, Y., Goebel, R., Stirnberg, R., Stöcker, T., ... Poser, B. A. (2021). Layer-dependent functional connectivity methods. *Progress in Neurobiology*, *207*. doi:10.1016/j.pneurobio.2020.101835

Stirnberg, R. & Stöcker, T. (2021). Segmented K-Space Blipped-Controlled Aliasing in Parallel Imaging (Skipped-CAIPI) for High Spatiotemporal Resolution Echo Planar Imaging. *Magnetic Resonance in Medicine*, *85*(0), 1540–1551. doi:10.1101/2020.06.08.140699

3 Handling multiple testing problem through effect calibration: implementation using PyMC

Lea Waller[1], Kelly Garner[2], Christopher R Nolan[3], Daniel Borek[4], Gang Chen[5]

1. Charité Universitätsmedizin Berlin, corporate member of Freie Universität Berlin and Humboldt-Universität zu Berlin, Department of Psychiatry and Neurosciences CCM, Berlin, Germany

2. School of Psychology, The University of Queensland, St. Lucia, 4072, QLD, Australia

3. School of Psychology, The University of New South Wales, NSW, Australia

4. Department of Data Analysis, Faculty of Psychology and Educational Sciences, Ghent University, Ghent, Belgium

5. Scientific and Statistical Computing Core, NIMH, NIH, Department of Health and Human Services, USA

3.1 Introduction

Human brain imaging data is massively multidimensional, yet current approaches to modelling functional brain responses entail the application of univariate inferences to each voxel separately. This leads to the multiple testing problem and unrealistic assumptions about the data such as artificial dichotomization (statistically significant or not) in result reporting. The traditional approach of massively univariate analysis assumes that no information is shared across the brain, effectively making a strong prior assumption of a uniform distribution of effect sizes, which is unrealistic given the connectivity of the human brain. The consequent requirement for multiple testing adjustments results in the *calibration of statistical evidence* without considering the estimation of effect, leading to substantial information loss and an unnecessarily heavy penalty.

A more efficient approach to handling multiplicity focuses on the *calibration of effect estimation* under a Bayesian multilevel modeling framework with a prior assumption of, for example, normality across space (Chen et al., 2019). The methodology has previously been implemented at the region level into the `AFNI` program `RBA` (Chen et al., 2022) using `Stan` through the `R` package `brms` (Bürkner, 2017). We intend to achieve two goals in this project:

(i) To re-implement the methodology using PyMC improve the performance and flexibility of the modeling approach.

(ii) To explore the possibility of analyzing voxel-level data using the multilevel modeling approach

3.2 Implementation using PyMC

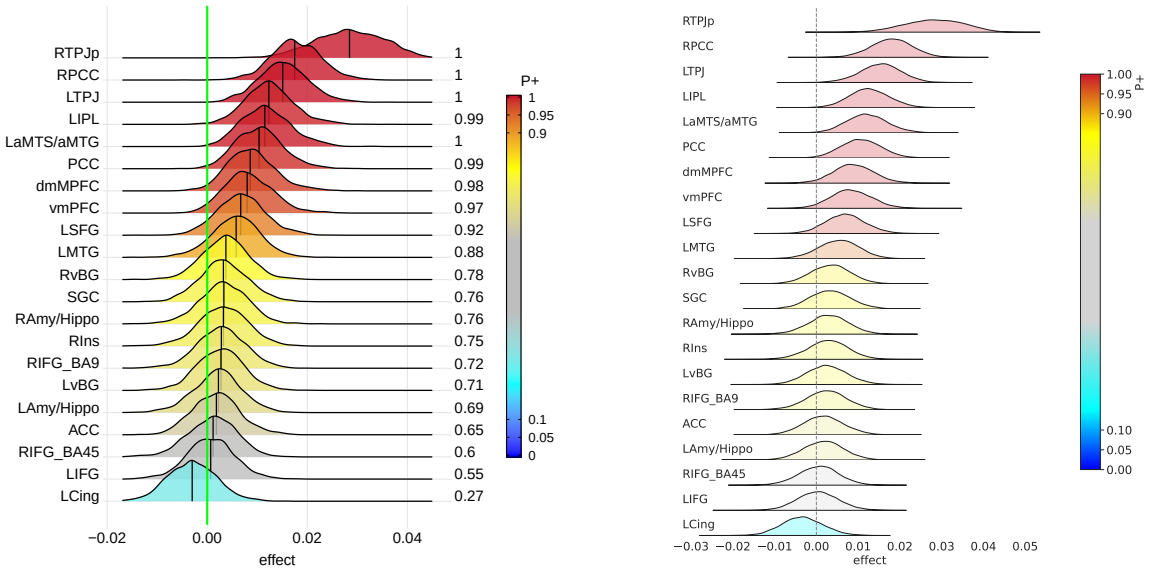
We used the dataset from Chen et al. (2019) to validate our PyMC implementation. The data contain the subject-level response variable y and a predictor of the behavioral measure x from $S = 124$ subjects at $R = 21$ regions. The modeling framework is formulated for the data y_{rs} of the sth subject at the rth region as below,

$$\begin{aligned} y_{rs} &\sim \mathcal{N}(\mu_{rs},\ \sigma^2) \\ \mu_{rs} &= \alpha_0 + \alpha_1 x_s + \theta_{0r} + \theta_{1r} x_s + \eta_s \\ \begin{bmatrix}\theta_{0r} \\ \theta_{1r}\end{bmatrix} &\sim \mathcal{N}(\mathbf{0}_{2\times 1},\ \boldsymbol{S}_{2\times 2}) \\ \eta_s &\sim \mathcal{N}(0,\ \tau^2) \\ &\text{where } r = 1, 2, \ldots, R \text{ and } s = 1, 2, \ldots, S \end{aligned} \tag{1}$$

where μ_{rs} and σ are the mean effect and standard deviation of the sth subject at the rth region, α_0 and α_1 are the overall mean and slope effect across all regions and subjects, θ_{0r} and θ_{1r} are the mean and slope effect at the rth region, η_s is the

(a) Posterior distributions: `brms`

(b) Posterior distributions: PyMC



(c) Extension to voxel-level analysis.

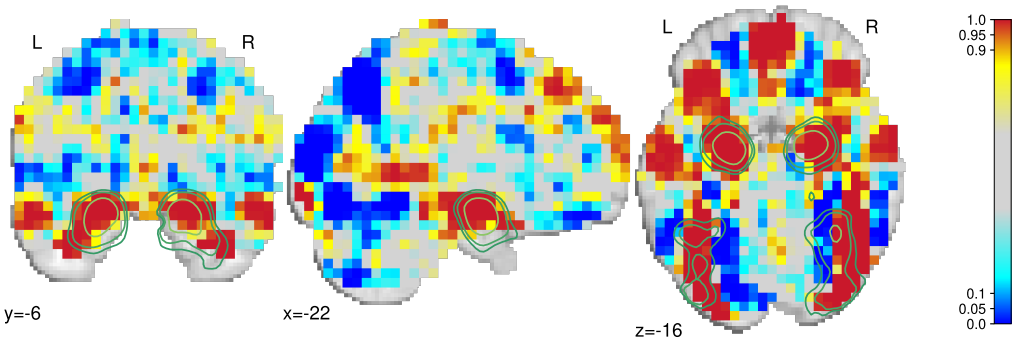


Figure 2: Validation of implementation using PyMC. (A) Posterior distributions of region-level behavior effects using `brms`. (B) Posterior distributions of region-level behavior effects using PyMC. (C) Posterior probabilities of the voxel-level effects being positive or negative, obtained using PyMC (plotted using Nilearn and overlaid in green with the NeuroQuery (Dockès et al., 2020) map for the term "emotional faces").

mean effect of the sth subject, $\boldsymbol{S}_{2\times 2}$ is the variance-covariance of the mean and slope effect at the rth region, and τ is the standard deviation of the sth subject's effect η_s.

We implemented this model using the PyMC probabilistic programming framework (Salvatier, Wiecki & Fonnesbeck, 2016), and the BAyesian Model-Building Interface (BAMBI) (Capretto et al., 2020). The latter is a high-level interface that allows for specification of multilevel models using the formula notation that is also adopted by `brms`. A notebook describing the implementation is available here. Our PyMC implementation was successfully validated: as shown in Fig. 2a and Fig. 2b, the posterior distributions from the PyMC implementation matched very well with their counterparts from the `brms` outout.

3.3 Extension of Bayesian multilevel modeling to voxel-level analysis

After exploring the model on the region level, we wanted to see if recent computational and algorithmic advances allow us to employ the multilevel modeling framework on the voxel level as well. We obtained the OpenNeuro dataset `ds000117` (Wakeman & Henson, 2015) from an experiment based on a face processing paradigm. Using `HALFpipe` (Waller et al., 2022), which is based on `fMRIPrep` (Esteban et al., 2019), the func-

tional images were preprocessed with default settings and z-statistic images were calculated for the contrast "famous faces + unfamiliar faces versus 2 · scrambled faces".

We applied the same modeling framework and PyMC code as for region-based analysis, but without the explanatory varaiable x in the model (1). To reduce computational and memory complexity, the z-statistic images were downsampled to an isotropic resolution of 5mm. Using the GPU-based `nuts_numpyro` sampler (Phan, Pradhan & Jankowiak, 2019) with default settings, we were able to sample 2,000 posterior samples of the mean effect parameter for each of the 14,752 voxels. Sampling four chains took 23 minutes on four Nvidia Tesla V100 GPUs.

The resulting posterior probabilities are shown in Figure 2c overlaid with the meta-analytic map for the term "emotional faces" obtained from NeuroQuery (Dockès et al., 2020). The posterior probability map is consistent with meta-analytic results, showing strong statistical evidence in visual cortex and amygdala voxels. The posterior probability maps also reveal numerous other clusters of strong statistical evidence for both positive and negative effects.

This implementation extension shows that large multilevel models are approaching feasibility, suggesting an exciting new avenue for statistical analysis of neuroimaging data. Next steps will be to investigate how to interpret and report these posterior maps, and to try more complex models that include additional model terms.

Acknowledgements

Computation has been performed on the HPC for Research cluster of the Berlin Institute of Health.

References

Bürkner, P.-C. (2017). Brms: An R Package for Bayesian Multilevel Models Using Stan. *Journal of Statistical Software*, *80*(1), 1–28. Number: 1. doi:10.18637/jss.v080.i01

Capretto, T., Piho, C., Kumar, R., Westfall, J., Yarkoni, T. & Martin, O. A. (2020). Bambi: A simple interface for fitting bayesian linear models in python. arXiv: 2012.10754 [stat.CO]

Chen, G., Taylor, P. A., Stoddard, J., Cox, R. W., Bandettini, P. A. & Pessoa, L. (2022). Sources of Information Waste in Neuroimaging: Mishandling Structures, Thinking Dichotomously, and Over-Reducing Data. *Aperture Neuro*, *2021*(5), 46. doi:10.52294/2e179dbf-5e37-4338-a639-9ceb92b055ea

Chen, G., Xiao, Y., Taylor, P. A., Rajendra, J. K., Riggins, T., Geng, F., … Cox, R. W. (2019). Handling Multiplicity in Neuroimaging through Bayesian Lenses with Multilevel Modeling. *Neuroinformatics*, *17*(4), 515–545. doi:10.1007/s12021-018-9409-6

Dockès, J., Poldrack, R. A., Primet, R., Gözükan, H., Yarkoni, T., Suchanek, F., … Varoquaux, G. (2020). NeuroQuery, comprehensive meta-analysis of human brain mapping. *eLife*, *9*, e53385. Publisher: eLife Sciences Publications, Ltd. doi:10.7554/eLife.53385

Esteban, O., Markiewicz, C. J., Blair, R. W., Moodie, C. A., Isik, A. I., Erramuzpe, A., … Gorgolewski, K. J. (2019). fMRIPrep: A robust preprocessing pipeline for functional MRI. *Nature Methods*, *16*(1), 111–116. Number: 1 Publisher: Nature Publishing Group. doi:10.1038/s41592-018-0235-4

Phan, D., Pradhan, N. & Jankowiak, M. (2019). Composable Effects for Flexible and Accelerated Probabilistic Programming in NumPyro. arXiv:1912.11554 [cs, stat]. doi:10.48550/arXiv.1912.11554

Salvatier, J., Wiecki, T. V. & Fonnesbeck, C. (2016). Probabilistic programming in python using PyMC3. *PeerJ Computer Science*, *2*, e55. doi:10.7717/peerj-cs.55

Wakeman, D. G. & Henson, R. N. (2015). A multi-subject, multi-modal human neuroimaging dataset. *Scientific Data*, *2*(1), 150001. doi:10.1038/sdata.2015.1

Waller, L., Erk, S., Pozzi, E., Toenders, Y. J., Haswell, C. C., Büttner, M., … Veer, I. M. (2022). ENIGMA HALFpipe: Interactive, reproducible, and efficient analysis for resting-state and task-based fMRI data. *Human Brain Mapping*, *43*(9), 2727–2742. doi:10.1002/hbm.25829

4 Exploding brains in Julia

Ömer Faruk Gülban[1,2], Leonardo Muller-Rodriguez[3]

1. Department of Cognitive Neuroscience, Faculty of Psychology and Neuroscience, Maastricht University, Maastricht, The Netherlands, 2. Brain Innovation, Maastricht, The Netherlands, 3. Psychoinformatics Lab, Forschungszentrum Jülich, Jülich, Germany

Particle simulations are used to generate visual effects (in movies, games etc...). In this project, we explore how we can use magnetic resonance imaging (MRI) data to generate interesting visual effects by using (2D) particle simulations. We highlight that, historically, we were first inspired by a detailed blog post (`https://nialltl.neocities.org/articles/mpm_guide.html`) on the material point method (Jiang, Selle & Teran, 1965; Love & Sulsky, 2006; Stomakhin, Schroeder, Chai, Teran & Selle, 2013). Our aim in Brainhack 2022 is to convert our previous progress in Python programming language to Julia. The reason why we have moved to Julia language is because it has convenient parallelization methods that are easy to implement while giving immediately speeding-up the particle simu-

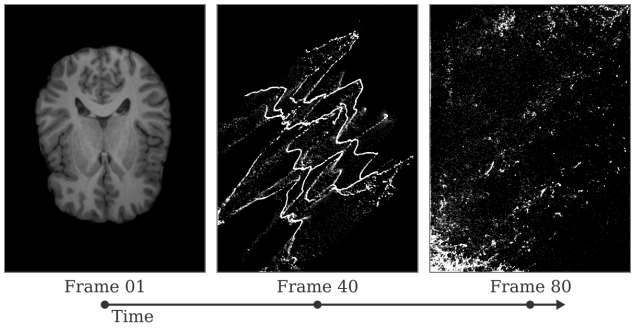


Figure 3: A video compilation of brain explosions can be seen at `https://youtu.be/_5ZDctWv5X4`.

lations.

Our previous efforts are documented at:

1. 2020 OpenMR Benelux: `https://github.com/OpenMRBenelux/openmrb2020-hackathon/issues/7`
2. 2020 OHBM Brainhack: `https://github.com/ohbm/hackathon2020/issues/124`
3. Available within the following github repository: `https://github.com/ofgulban/slowest-particle-simulator-on-earth`

As a result of this hackathon project, a compilation of our progress (Figure 3) can be seen at `https://youtu.be/_5ZDctWv5X4` as a video. Our future efforts will involve sophisticating the particle simulations, the initial simulation parameters to generate further variations of the visual effects, and potentially synchronizing the simulation effects with musical beats.

References

Jiang, C., Selle, A. & Teran, J. (1965). The Affine Particle-In-Cell Method Chenfanfu. *ACM Transactions on Graphics*, *34*(4), 51. Retrieved from http://doi.acm.org/10.1145/2766996%7B%5C%%7D0Apapers3://publication/doi/10.1145/2766996

Love, E. & Sulsky, D. L. (2006). An unconditionally stable, energy-momentum consistent implementation of the material-point method. *Computer Methods in Applied Mechanics and Engineering*, *195*(33-36), 3903–3925. doi:10.1016/j.cma.2005.06.027

Stomakhin, A., Schroeder, C., Chai, L., Teran, J. & Selle, A. (2013). A material point method for snow simulation. *ACM Transactions on Graphics*, *32*(4). doi:10.1145/2461912.2461948