The Journal of Open Source Software

CogStim: Reproducible visual stimulus generation for cognitive science, neuroscience, and vision research

Eudald Correig-Fraga [1]

1 Innovamat Education, Barcelona, Catalonia (Spain)

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

License
Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

DRAFT

Summary

CogStim is an open-source Python library for reproducible, parameterized offline generation of visual stimuli for psychology, neuroscience, and computer vision. It produces PNG/JPEG/SVG assets for common paradigms (two-colour ANS arrays, match-to-sample pairs with optional total-area equalization, single-colour dot arrays, geometric shapes, oriented lines/stripes, and fixation targets). Deterministic seeding and robust geometric routines enforce non-overlap, boundary validity, and equalization to minimise perceptual confounds. A compact Python API and command-line interface make the tool accessible to both programmers and non-programmers, enabling quick creation of controlled stimuli without special setup.

Statement of need

Designing visual stimuli is a routine requirement in psychology, neuroscience, and vision research. It requires precise control over numerosity, size, spacing, color, and layout, alongside reproducible randomization and strict study dependent constraints. When built ad hoc, these demands make stimulus creation tedious and can introduce small inconsistencies that affect comparisons across studies and model evaluations. Researchers therefore benefit from a simple way to generate controlled stimuli offline as standard image assets ready to use wherever needed.

CogStim addresses this need by providing an offline, parameterized generator for a range of paradigms: approximate number system (Halberda et al., 2008), match-to-sample (Sella et al., 2013), geometric shapes, oriented lines (Srinivasan, 2021), and fixation targets (Thaler et al., 2013). It produces images in any major image format (e.g. PNG, JPEG, SVG) that can be dropped into experiment builders, web or desktop presentation software, and computer-vision pipelines without heavy setup. A Python API and a command-line interface enable both quick prototyping and large batch generation; deterministic seeds ensure the same configuration yields identical outputs; and built-in algorithms enforce non-overlap and allow total-area equalization when required. In this way the library serves dual needs: controlled stimuli for behavioral and neuro-cognitive experiments, and well-specified synthetic datasets for model development and evaluation.

CogStim provides a single, extensible library that: 1. covers widely used paradigms (ANS, MTS, shapes, orientation discrimination, fixation). 1. guarantees determinism via seed control and consistent planning logic. 1. outputs ready-to-use datasets with standard train/test splits and stable file names. 1. offers both a CLI and a Python API, minimizing ramp-up time for non-specialists. 1. implements stimulus equalization algorithms that balance total surface areas within or between dot arrays, ensuring fair comparisons in numerosity and match-to-sample tasks. 1. supports common image formats, including PNG and JPEG via a configurable img_format parameter, and SVG for vector export. 1. reduces manual stimulus-creation effort by automating parameter sweeps, placement rules, and file organization.

The library is designed for: (i) behavioral and neuro-cognitive tasks such as numerosity discrimination, (ii) model evaluation in computational neuroscience and computer vision, and (iii) dataset creation for machine-learning workflows that require well-controlled synthetic stimuli. It is equally accessible to users with and without programming experience, thanks to its dual API/CLI design and comprehensive documentation, which includes an LLM-optimized manual that users can feed to their LLM's of preference to generate the instructions needed to create their stimuli.

State of the field & relation to similar software

General-purpose experiment frameworks such as PsychoPy, Psychtoolbox, and jsPsych focus on stimulus presentation and timing during runtime, while various domain-specific generators target single paradigms. CogStim complements these tools by producing offline, parameterized image assets that drop into any workflow with minimal setup. It emphasizes reproducibility through explicit seeding and systematic parameter sweeps, and it reduces perceptual confounds via robust non-overlap placement, boundary checks, and stimulus equalization for numerosity and match-to-sample tasks. The library spans multiple paradigms (ANS, MTS, geometric shapes, oriented lines, fixation targets) and exports PNG, JPEG, and SVG for portability across experimental platforms and computer-vision pipelines. A dual interface (command line and Python API), together with documentation tailored for LLM-assisted use, makes it accessible to both programmers and non-programmers and provides a clear template for extending to new stimulus classes.

This library has been used both for the creation of psychometric tests for children (E. Correig-Fraga et al., 2024) (Correig-Fraga et al., 2025), as well as for computer vision tasks (E. Correig-Fraga et al., n.d.).

Software description

Design and key features

- **Task coverage**: ANS two-colour dot arrays; single-colour dot arrays; MTS pairs with optional area equalization; geometric shapes (circle, star, triangle, square); oriented line/stripe patterns; fixation targets.

- **Determinism & reproducibility**: global seed handling for Python/NumPy; same parameters and same seed will yield identical images.

- **Robust dot engine**: `DotsCore` enforces non-overlap, boundary validity, optional area equalization, and (for MTS) pair equalization within tolerances.
- **Stimulus equalization algorithms**: CogStim implements robust geometric equalization methods that adjust dot radii so that total surface areas are matched either within two-colour ANS arrays or between sample–match pairs. These procedures guarantee perceptually fair stimuli for numerosity and matching tasks, maintaining non-overlap and boundary validity while achieving precise area ratios within configurable tolerances.
- **CLI & Python API**: consistent configuration via dictionaries in code and ergonomic subcommands in the CLI.

Implementation and dependencies

CogStim is implemented in Python (3.10) (Python Software Foundation, 2023) and builds upon a small number of widely used open-source libraries. Image creation and drawing operations are handled through Pillow (Clark, 2015), while all geometric computations and randomization routines rely on NumPy (Harris et al., 2020). The library uses tqdm (Costa-Luis & others, 2022)

DRAFT

to provide progress bars during generation processes and adopts standard Python modules such as argparse for command-line interfaces and pytest for automated testing.

The codebase is organized around a small set of generator classes that call these dependencies through a unified interface. Each generator defines the parameters of a particular task (e.g., ANS, match-to-sample, shapes, lines, fixation) and uses Pillow for rendering, NumPy for geometric calculations, and tqdm for user feedback. This results in a lightweight, portable implementation that can run on any system supporting Python without special dependencies or graphical backends.

All dependencies are open source, actively maintained, and available through the Python Package Index (PyPI), ensuring long-term accessibility and compatibility with typical research workflows. The project is licensed under MIT, and available as a Git repository in Github.

Example figures

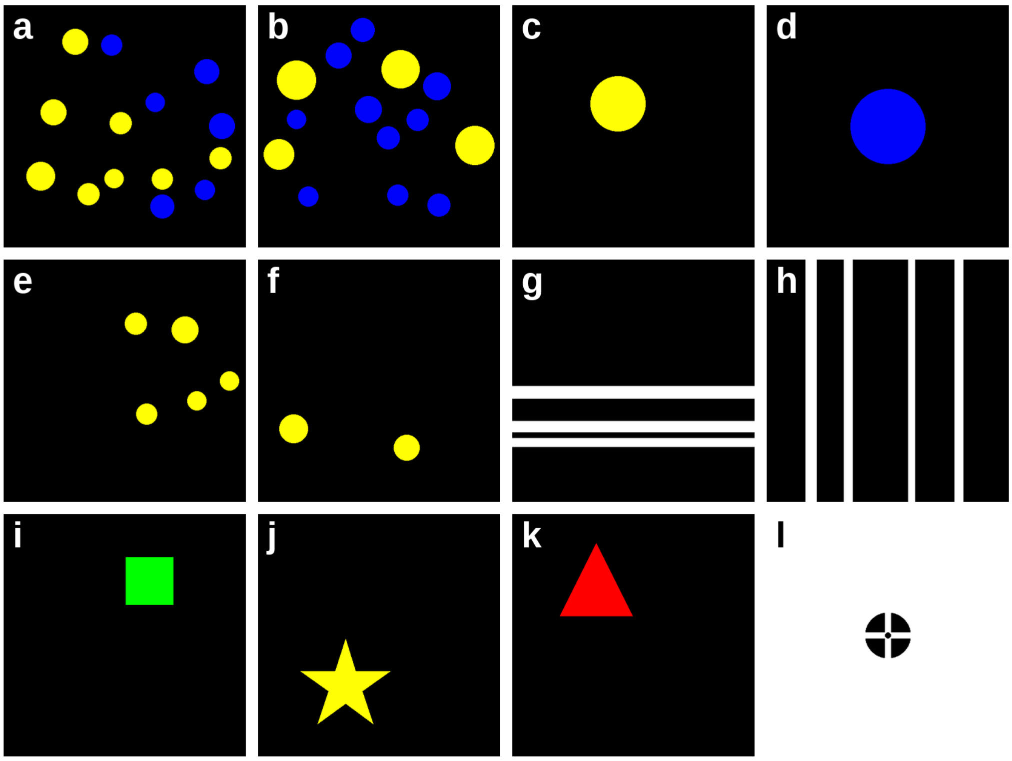

Figure 1: Representative stimuli generated by CogStim.

Figure 1. Representative stimuli generated by CogStim across different task paradigms. **(a, b)** Approximate Number System (ANS) two-colour dot arrays for numerosity discrimination tasks, with (b) showing area-equalized dots between colours. **(c, d)** Single geometric shapes (circle) in different colours, used in shape discrimination tasks. **(e, f)** Single-colour dot arrays suitable for numerosity estimation, match-to-sample (MTS) paradigms, or as components in multi-feature discrimination tasks. **(g, h)** Oriented line/stripe patterns for orientation discrimination experiments. **(i, j, k)** Additional geometric shapes (square, star, triangle) in various colours, demonstrating the library's shape generation capabilities for categorical perception and visual search tasks. **(l)** Fixation cross stimulus for experimental trial preparation and gaze control.

Availability

- Repository: https://github.com/eudald-seeslab/cogstim
- License: MIT
- Issue tracker: enabled and publicly readable
- Archive: upon acceptance, we will create a tagged release, archive on Zenodo and include the DOI here.

Acknowledgements

We thank Innovamat Education for their support in the development of this open source work.

Conflicts of Interest

Authors declare no competing interests.

References

Clark, A. (2015). *Pillow (PIL fork) documentation*. readthedocs. https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf

Correig-Fraga, E., Guimerà, R., & Sales-Pardo. (n.d.). Structure alone supports efficient visual computation in the drosophila visual system. *Under Review*.

Correig-Fraga, E., Vilalta-Riera, A., & Calvo-Pesce, C. (2024). Development and validation of a semi-automated, scalable response to intervention framework in mathematics. *SN Social Sciences*, *4*(2), 1–19. https://doi.org/10.1007/s43545-024-00835-7

Correig-Fraga, Sales-Pardo, & Guimerà, R. (2025). Interplay between children's cognitive profiles and within-school social interactions is nuanced and differs across ages. *Communications Psychology*, *3*(1). https://doi.org/10.1038/s44271-025-00227-4

Costa-Luis, C. da, & others. (2022). *Tqdm: A fast, extensible progress bar for python and CLI*. https://github.com/tqdm/tqdm.

Halberda, J., Mazzocco, M. M. M., & Feigenson, L. (2008). Individual differences in non-verbal number acuity correlate with maths achievement. *Nature*, *455*(7213), 665–668. https://doi.org/10.1038/nature07246

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Python Software Foundation. (2023). *Python: A programming language for scientific computing*. https://www.python.org/.

Sella, F., Lanfranchi, S., & Zorzi, M. (2013). Enumeration skills in down syndrome. *Research in Developmental Disabilities*, *34*(11), 3798–3806. https://doi.org/10.1016/j.ridd.2013.07.038

Srinivasan, M. V. (2021). Vision, perception, navigation and "cognition" in honeybees and applications to aerial robotics. *Biochemical and Biophysical Research Communications*, *564*, 4–17. https://doi.org/10.1016/j.bbrc.2020.09.052

Thaler, L., Schütz, A. C., Goodale, M. A., & Gegenfurtner, K. R. (2013). What is the best fixation target? The effect of target shape on stability of fixational eye movements. *Vision Research*, *76*, 31–42. https://doi.org/10.1016/j.visres.2012.10.012