

# Speckle Cn2 Profiler: Improving Satellite Communications with Machine Learning

Simone Ciarella<sup>1</sup>✉, Luisa Orozco<sup>1</sup>, Victor Azizi<sup>1</sup>, Marguerite Arvis<sup>2</sup>, and Rudolf Saathof<sup>2</sup>

<sup>1</sup> Netherlands eScience Center, Amsterdam, The Netherlands <sup>2</sup> Faculty of Aerospace Engineering, Delft University of Technology, Delft, The Netherlands ✉ Corresponding author

DOI: [10.21105/joss.09127](https://doi.org/10.21105/joss.09127)

## Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Chris Vernon](#) ↗

## Reviewers:

- [@acsenrafilho](#)
- [@HectorMozo3110](#)

Submitted: 03 December 2024

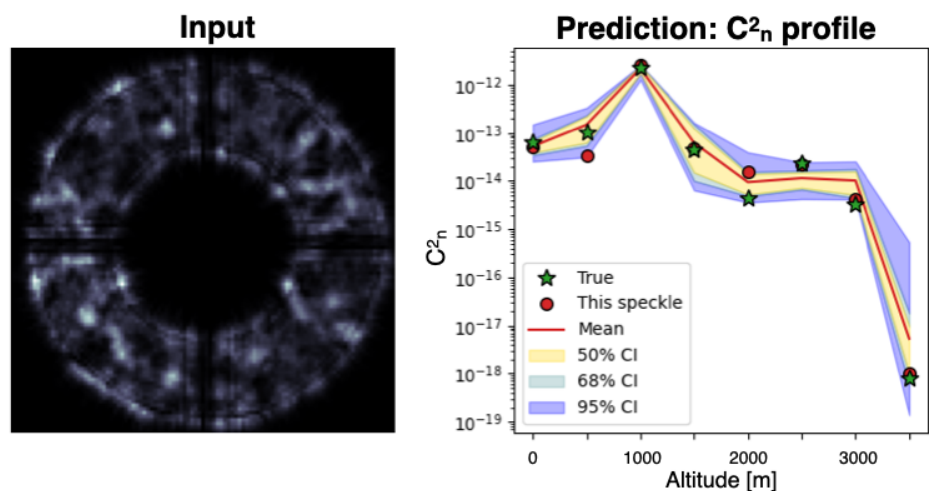
Published: 15 October 2025

## License

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

## Summary

Optical satellite communications is a growing research field in which -using lasers- signals can be sent from the ground to satellites, from satellites to satellites and then back to the ground. The main advantage of using laser communication over radio waves is increased bandwidth that enables the transfer of more data in less time. However, one of the challenges for this protocol is the turbulence in the atmosphere that perturbs such transmission. The reduction of the quality of signal communication can be calculated and then compensated, but this requires a knowledge of the turbulence strength. A common way to model the turbulence is to use the refractive index structure constant,  $C_n^2$ , which is a measure of the strength of the turbulence. Its profile can be used to estimate the effect of the turbulence on the signal and then apply a correction. To measure  $C_n^2$ , there are several alternative instruments, each one with its own limitations. One possibility is to use speckle-based observation, which consists on looking at the twinkling of the stars and use their pattern to infer the turbulence profile. This is a non-intrusive method that can be used in real time, but it requires a deep understanding of the turbulence and the observed speckle patterns, which are highly influenced by the turbulence profile. The connection between speckle observation, and turbulence ( $C_n^2$ ) is not clearly understood, so an analytical theory does not exist. Here we present speckcn2, a Python package that uses machine learning to provide a numerical reconstruction of the turbulence profile from a speckle pattern ([Ciarella, 2024](#)).



**Figure 1:** Example of speckcn2 pipeline: speckle pattern as input to output a prediction of the turbulence profile (J).

## Statement of need

While deep learning has been applied to mitigate atmospheric turbulence effects in imaging through temporal mitigation using videos (Zhang et al., 2024) and static image degradation compensation (Yasarla & Patel, 2021), these approaches focus on visual correction rather than quantitative turbulence characterization. In contrast, speckcn2 addresses the critical need for numerical reconstruction of  $C_n^2$  profiles, enabling both scientific understanding of atmospheric turbulence and practical integration into communication system models for performance optimization.

The primary target users of speckcn2 are aerospace engineers working on optical satellite communication systems, atmospheric scientists studying turbulence phenomena, and astronomers developing adaptive optics systems. The package provides these communities with a tool to estimate turbulence profiles from speckle observations where analytical theories are lacking, using machine learning to bridge this gap.

Existing software packages address atmospheric turbulence from different perspectives. AOtools (Townson et al., 2019) provides general-purpose adaptive optics utilities including phase screen generation and turbulence parameter conversions, but lacks ML-based profile reconstruction from observations. FAST (Farley et al., 2022) (Fourier domain Adaptive optics Simulation Tool) offers rapid Monte Carlo characterization of free space optical links with turbulence modeling, while OOPAO (Heritier et al., 2023) (Object-Oriented Python Adaptive Optics) provides end-to-end adaptive optics simulation. However, both focus on forward modeling—simulating turbulence effects given known profiles—rather than the inverse problem of reconstructing profiles from measurements. Traditional  $C_n^2$  profiling instruments such as radiosondes, SCIDAR, or MASS require specialized hardware and intrusive deployment. speckcn2 uniquely combines ML-based inverse modeling with speckle pattern analysis to provide explicit turbulence profile reconstruction in real-time, using only optical detection systems without specialized hardware. This fills a critical gap for applications requiring quantitative atmospheric characterization from passive observations.

Built on PyTorch (Ansel et al., 2024), the package is designed to handle diverse image regression tasks with both synthetic and experimental data. By combining techniques like equivariance and ensemble learning, speckcn2 provides a robust framework applicable beyond aerospace engineering to any field requiring image-to-profile regression. Complete documentation and API references are available at <https://males-project.github.io/SpeckleCn2Profiler/>, with installation instructions and examples to facilitate adoption across different research communities.

## Key features

### Instrument specialization

When estimating the turbulence features, it is of fundamental importance to not mix the instrumental noise with the real effects that are being measured. A fundamental aspect of speckcn2 is the possibility to train models with different noise profiles, representing the noise of different instruments and modeling different detectors, whether real or simulated. By adapting the `apply_noise` function, users can model any type of effect related to their research and instruments. The current API provides a series of parameters that can be tuned to simulate the noise of different instruments, such as the signal-to-noise ratio, the detector gain, and the obscuration.

### Equivariant model

To take advantage of the symmetry in the input data, speckcn2 uses a concept called equivariance (Cohen & Welling, 2016). This means that the model can learn the same

features independently of the input data orientation. This is especially helpful for turbulence reconstruction, where the direction of the speckle pattern is not relevant.

speckcn2 supports two types of equivariance: weak and strong. Weak equivariance is achieved by randomly rotating the input data, which can then be used with any model from torchvision ([TorchVision maintainers and contributors, 2016](#)), including fine-tuning ResNets ([He et al., 2015](#)).

Strong equivariance is achieved by using the equivariant sparse convolutional neural network (escnn) ([Cesa et al., 2021](#); [Weiler & Cesa, 2021](#)). These networks are more powerful for this type of problem but are harder to train.

## Ensemble learning

speckcn2 can also use ensemble learning by averaging the predictions from multiple input images. This means that the prediction of each model requires a set of multiple input images. This is only useful if the input images change faster than the output. Since this is not the case for laser communications, this feature is optional and can be turned off.

## Applications: STORM for Laser Satellite Communications

A notable application of speckcn2 is STORM (Speckle-based Turbulence Observation and Reconstruction via Machine learning) ([Arvis et al., 2025](#)), designed for line-of-sight turbulence profiling in laser satellite communications. STORM reconstructs 8-layer  $C_n^2(h)$  profiles from single-shot speckle patterns, addressing limitations of traditional methods like SCIDAR or SLODAR that require binary sources or time-series measurements. This single-source-single-shot capability is crucial for tracking fast-moving LEO satellites where the air column changes rapidly between measurements. Simulation results show over 90% accuracy on the Fried parameter, isoplanatic angle, and Rytov index, enabling improved adaptive optics design and communication link optimization for optical space communications.

## Software implementation

speckcn2 is implemented in Python and uses PyTorch ([Ansel et al., 2024](#)) for its machine learning tasks. This allows the user to take advantage of GPU acceleration, making computations faster and more efficient.

The package is published to PyPI and is easy to install via pip, and it works on both Linux and MacOS. It has a simple and user-friendly API that lets users quickly build, train, and evaluate models. Whether you are a beginner or an experienced user, speckcn2 is designed to be accessible and flexible.

For new users, the documentation and examples provide a great starting point, helping them get up to speed quickly. Experienced users will appreciate the flexibility of the package, which allows for customization to meet specific research needs.

## Acknowledgements

The authors would like to acknowledge the Netherlands eScience Center for the funding provided under grant number NLESC.SSIML.2022c.021.

## References

Ansel, J., Yang, E., He, H., Gimelshein, N., Jain, A., Voznesensky, M., Bao, B., Bell, P., Berard, D., Burovski, E., Chauhan, G., Chourdia, A., Constable, W., Desmaison,

- A., DeVito, Z., Ellison, E., Feng, W., Gong, J., Gschwind, M., ... Chintala, S. (2024). PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. <https://doi.org/10.1145/3620665.3640366>
- Arvis, M., Ciarella, S., Loicq, J., & Saathof, R. (2025). Single shot line-of-sight atmospheric turbulence profiling for laser satellite communications with STORM. *Applied Optics*.
- Cesa, G., Lang, L., & Weiler, M. (2021, October 6). *A Program to Build  $E(N)$ -Equivariant Steerable CNNs*. International Conference on Learning Representations. <https://openreview.net/forum?id=WE4qe9xlnQw>
- Ciarella, S. (2024). *Speckle to Cn2* (Version 1.0.0). Zenodo. <https://doi.org/10.5281/zenodo.14178688>
- Cohen, T., & Welling, M. (2016). Group Equivariant Convolutional Networks. In M. F. Balcan & K. Q. Weinberger (Eds.), *Proceedings of The 33rd International Conference on Machine Learning* (Vol. 48, pp. 2990–2999). PMLR. <https://proceedings.mlr.press/v48/cohenc16.html>
- Farley, O. J. D., Townson, M. J., & Osborn, J. (2022). FAST: Fourier domain adaptive optics simulation tool for bidirectional ground-space optical links through atmospheric turbulence. *Opt. Express*, 30(13), 23050–23064. <https://doi.org/10.1364/OE.458659>
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep residual learning for image recognition*. <https://doi.org/10.48550/arXiv.1512.03385>
- Heritier, C. T., Verinaud, C., & Correia, C. M. (2023, June). OOPAO: Object Oriented Python Adaptive Optics. *Adaptive Optics for Extremely Large Telescopes 7th Edition*. <https://doi.org/10.13009/AO4ELT7-2023-052>
- TorchVision maintainers and contributors. (2016). *TorchVision: PyTorch's Computer Vision library*. <https://github.com/pytorch/vision>
- Townson, M. J., Farley, O. J. D., Xivry, G. O. de, Osborn, J., & Reeves, A. P. (2019). AOtools: a Python package for adaptive optics modelling and analysis. *Opt. Express*, 27(22), 31316–31329. <https://doi.org/10.1364/OE.27.031316>
- Weiler, M., & Cesa, G. (2021, April 6). *General  $E(2)$ -Equivariant Steerable CNNs*. <https://doi.org/10.48550/arXiv.1911.08251>
- Yasarla, R., & Patel, V. M. (2021). Learning to restore images degraded by atmospheric turbulence using uncertainty. *2021 IEEE International Conference on Image Processing (ICIP)*, 1694–1698. <https://doi.org/10.1109/ICIP42928.2021.9506614>
- Zhang, X., Chimmitt, N., Chi, Y., Mao, Z., & Chan, S. H. (2024). Spatio-temporal turbulence mitigation: A translational perspective. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. <https://doi.org/10.1109/CVPR52733.2024.00279>