Why we're all NUTS about Turing.jl

... and what you can infer from it ...

Penelope Yong *(she/her)*
Research Software Engineer
1 October 2025

Today's slides

pysm.dev/talk

Bayesian statistics

$\boldsymbol{\theta}$ *parameters*

↓

$\boldsymbol{\mathcal{D}}$ *data* (observed)

Specify model with Turing syntax

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta) \cdot p(\theta)}{p(\mathcal{D})}$$

Infer with statistical techniques implemented in Turing

Constant; doesn't matter

Specifying models

```
@model function linreg(x, y)
    m ~ Normal(0, 1)
    c ~ Normal(0, 1)
    s ~ truncated(Normal(0, 10); lower=1)
    t = m .* x .+ c
    for i in eachindex(y)
        y[i] ~ Normal(t[i], s)
    end
end
```

Prior

$p(\theta)$

$p(\mathcal{D}|\theta)$

Likelihood

Performing inference

```
data = from_csv(“data.csv”)

model = linreg(data.x, data.y)
```

Markov chain Monte Carlo

```
sample(model, NUTS(), 1000)
```

Mode estimation

```
maximum_likelihood(model)
```

Variational inference

```
vi(model, q_fullrank_gaussian(m), 1000)
```

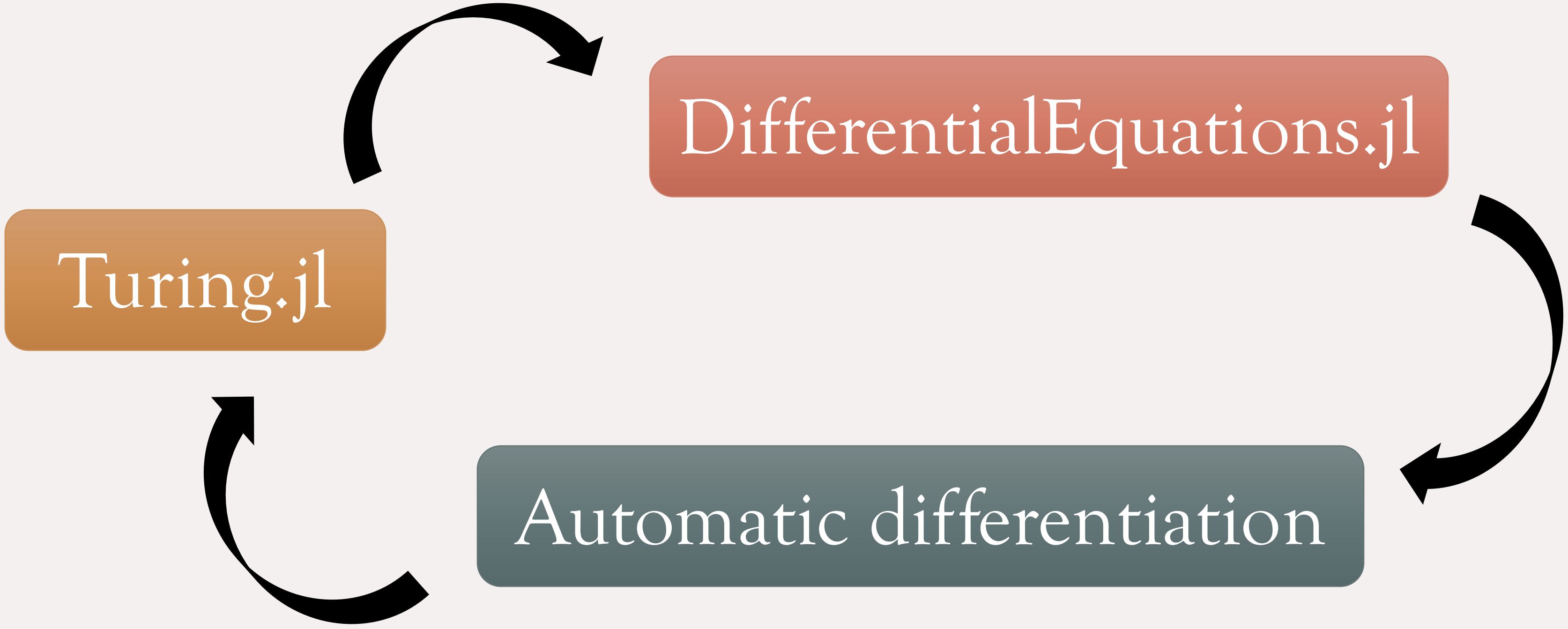
Specifying models

Friendly syntax for random variables

```
@model function linreg(x, y)
    m ~ Normal(0, 1)
    c ~ Normal(0, 1)
    s ~ truncated(Normal(0, 10); lower=1)
    t = m .* x .+ c
    for i in eachindex(y)
        y[i] ~ Normal(t[i], s)
    end
end
```

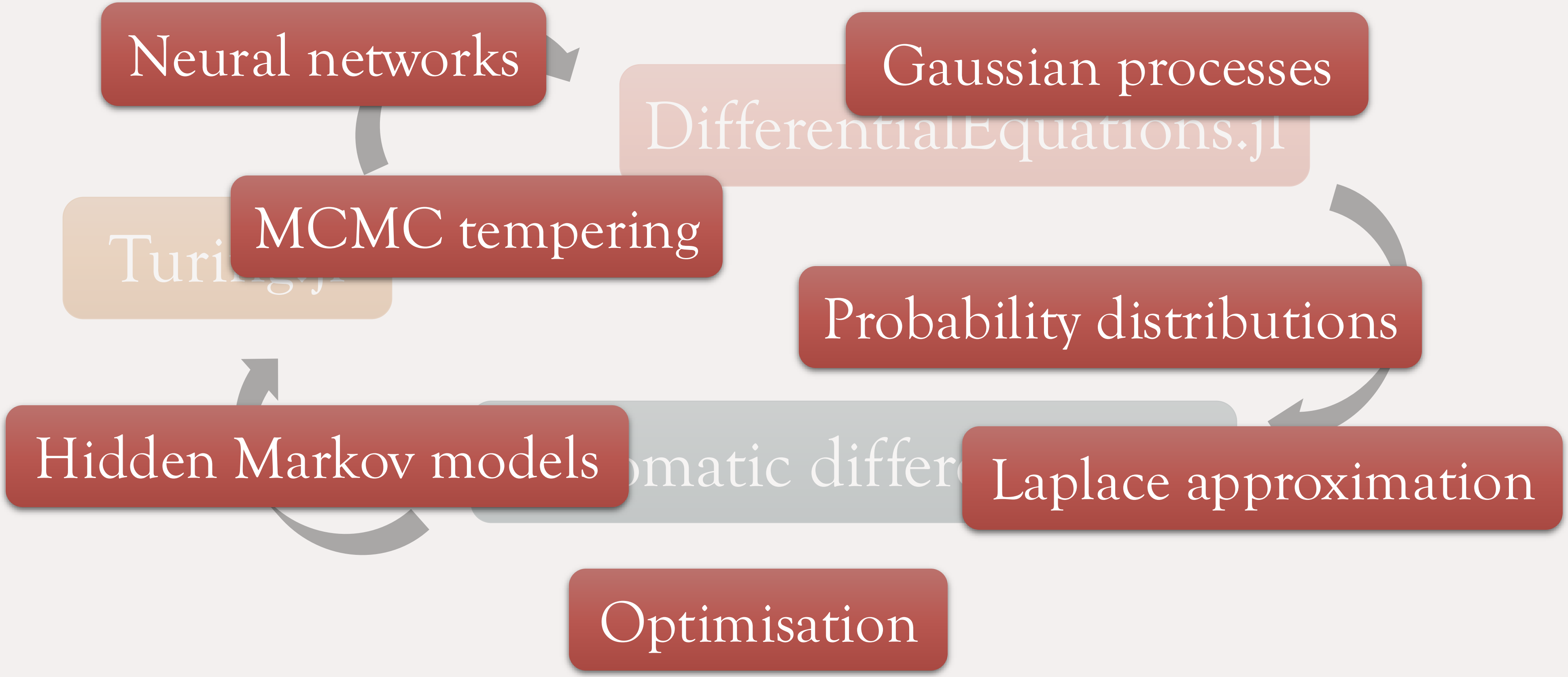
Arbitrary Julia code

The virtuous cycle of Julia



turinglang.org/docs/tutorials/bayesian-differential-equations/

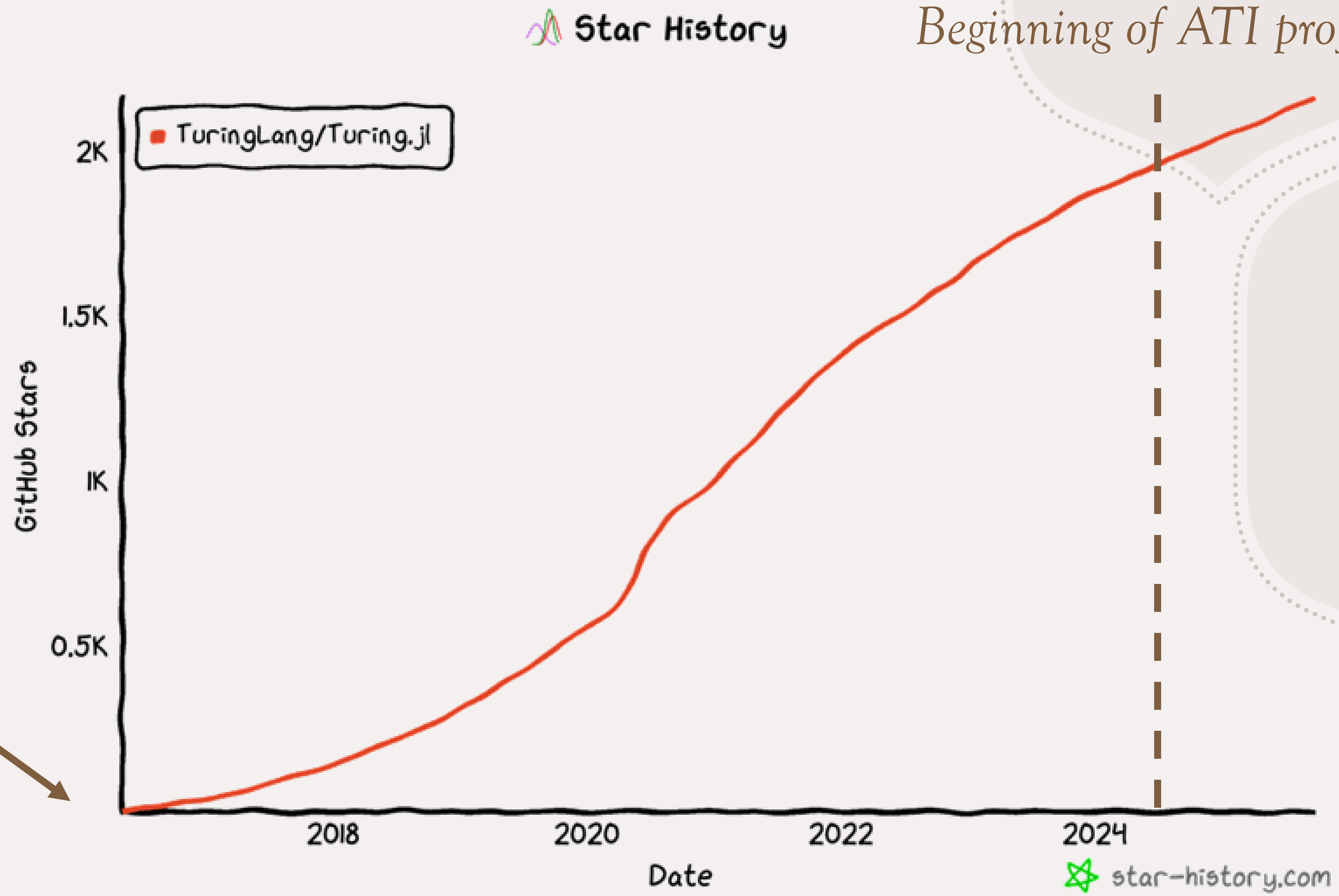
The virtuous cycle of Julia



The timeline

Started in 2016

Beginning of ATI project



Core Turing.jl development goals

Defining a v1.0 interface

- Submodels
- Custom samplers
- Dictionaries of parameter values (rather than vectors)

Improving code quality through internal refactoring

- Better data structures
- Simpler evaluation contexts
- Reimplementation of algorithms

Extending documentation

- Well-defined, testable APIs
- Meaningful usage guides
- Community engagement

Core Turing.jl development goals

Defining a v1.0 interface	Improving code quality through internal refactoring	Extending documentation
Submodels	Better data structures	Well-defined, testable APIs
Custom samplers	Simpler evaluation contexts	**Meaningful usage guides**
Dictionaries of parameter values (rather than vectors)	Reimplementation of algorithms	Community engagement

Submodels

```
@model function linreg(x, y)
    m ~ Normal(0, 1)
    c ~ Normal(0, 1)
    s ~ truncated(Normal(0, 10); lower=1)
    t = m .* x .+ c
    for i in eachindex(y)
        y[i] ~ Normal(t[i], s)
    end
end
```

Prior
$p(\theta)$

$p(\mathcal{D}|\theta)$
Likelihood

Defining a v1.0 interface

Submodels

```
@model function priors()
    m ~ Normal(0, 1)
    c ~ Normal(0, 1)
    s ~ truncated(Normal(0, 10); lower=1)
    return (m = m, c = c, s = s)
end

@model function linreg(x, y)
    ps ~ to_submodel(priors())
    t = ps.m .* x .+ ps.c
    for i in eachindex(y)
        y[i] ~ Normal(t[i], ps.s)
    end
end
```

Prior

$p(\theta)$

$p(\mathcal{D}|\theta)$

Likelihood

Submodels

```julia
@model function priors()
    m ~ Normal(0, 1)
    c ~ Normal(0, 1)
    s ~ truncated(Normal(0, 10); lower=1)
    return (m = m, c = c, s = s)
end

p = priors()
```

Submodels

```
@model function priors()
    m ~ Normal(0, 1)
    c ~ Normal(0, 1)
    s ~ truncated(Normal(0, 10); lower=1)
    return (m = m, c = c, s = s)
end

p = priors()

export p
```

Extending documentation

Automatic differentiation

ForwardDiff.jl →

Mooncake.jl ⇄

ReverseDiff.jl ←

Enzyme.jl ⇄

Zygote.jl ←

→ *Has forward-mode*

← *Has reverse-mode*

Extending documentation

Automatic differentiation

Model name \\ AD type	EnzymeForward	EnzymeReverse	FiniteDifferences	ForwardDiff
assume_beta	2.719	2.962	23.556	1.710
assume_dirichlet	2.548	2.665	21.672	1.331
assume_lkjcholu	4.470	6.235	157.080	2.586
assume_mvnormal	(?) error	3.053	38.294	1.147
assume_normal	2.960	2.301	27.885	1.437
assume_submodel	2.283	1.891	35.270	1.228
assume_wishart	(?) error	10.909	43.604	1.111
observe_bernoulli	2.653	3.030	22.347	1.629
observe_categorical	1.999	5.804	18.470	1.277
observe_von_mises	error	error	31.325	NaN

turinglang.org/ADTests

(Inter)national collaborations

Two *Google Summer of Code* students in 2025

Graphical models

JuliaBUGS.jl

Model selection

Harmonic.jl*

State space models

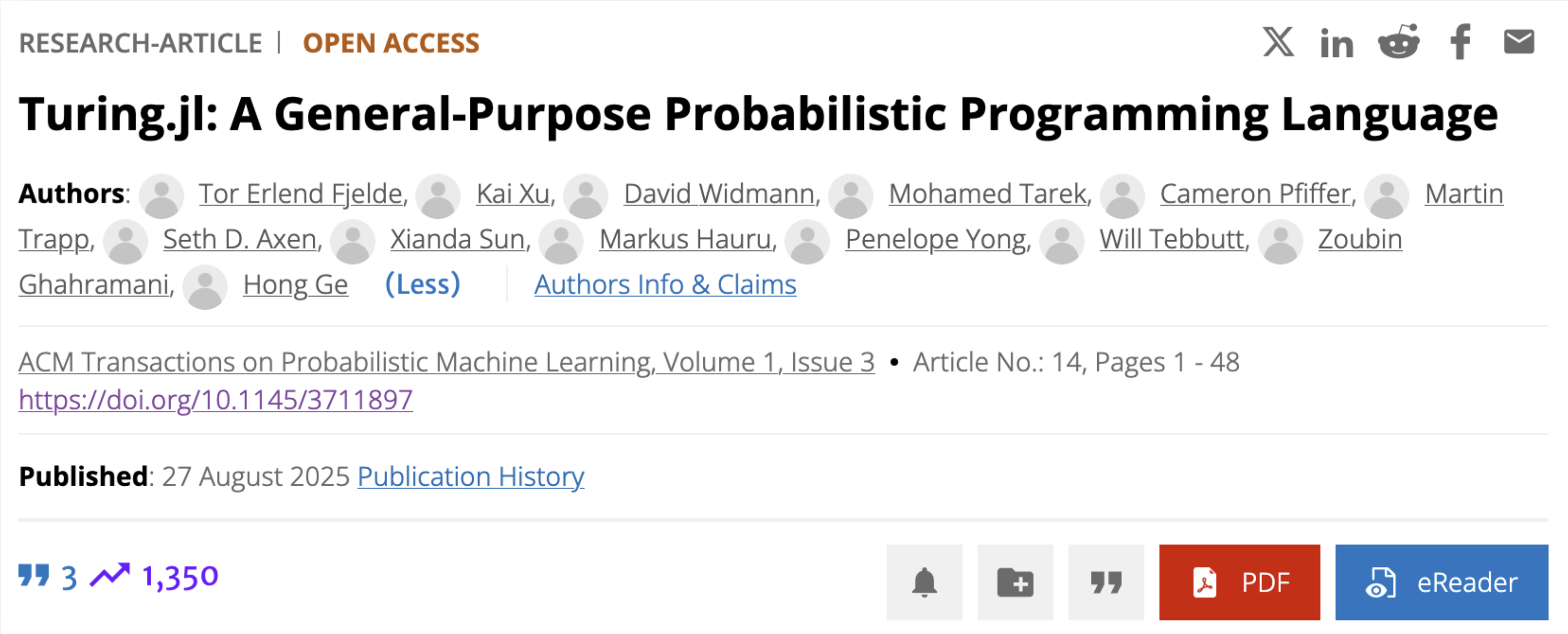
SSMProblems.jl

Julia coroutines

Libtask.jl

* *name TBC*

Our newest Turing.jl paper



RESEARCH-ARTICLE | OPEN ACCESS

Turing.jl: A General-Purpose Probabilistic Programming Language

Authors: Tor Erlend Fjelde, Kai Xu, David Widmann, Mohamed Tarek, Cameron Pfiffer, Martin Trapp, Seth D. Axen, Xianda Sun, Markus Hauru, Penelope Yong, Will Tebbutt, Zoubin Ghahramani, Hong Ge (Less) | Authors Info & Claims

ACM Transactions on Probabilistic Machine Learning, Volume 1, Issue 3 • Article No.: 14, Pages 1 - 48
https://doi.org/10.1145/3711897

Published: 27 August 2025 Publication History

3 1,350

PDF eReader

doi.org/10.1145/3711897

Turing.jl in the real world

Turing: a language for flexible probabilistic inference
H Ge, K Xu, Z Ghahramani
International conference on artificial intelligence and statistics, 2018 · proceedings.mlr.press

Abstract
Probabilistic programming promises to simplify and democratize probabilistic machine learning, but successful probabilistic programming systems require flexible, generic and efficient inference engines. In this work, we present a system called Turing for building MCMC algorithms for probabilistic programming inference. Turing has a very simple syntax and makes full use of the numerical capabilities in the Julia programming language, including all implemented probability distributions, and automatic differentiation.

SHOW MORE

Save Cite Cited by 485 Related articles All 6 versions

The original Turing.jl conference proceeding (2018) has approximately 500 *citations*

https://proceedings.mlr.press/v84/ge18b.html

Who uses Turing?

Introducing `ActiveInference.jl`: A Julia Library for Simulation and Parameter Estimation with Active Inference Models

by Samuel William Nehrer [1,†], Jonathan Ehrenreich Laursen [1,†], Conor Heins [2,3,*], Karl Friston [3,4], Christoph Mathys [5] and Peter Thestrup Waade [5]

1 School of Culture and Communication, Aarhus University, 8000 Aarhus, Denmark
2 Department of Collective Behaviour, Max Planck Institute of Animal Behavior, D-78457 Konstanz, Germany
3 VERSES Research Lab., Los Angeles, CA 90016, USA
4 Queen Square Institute of Neurology, University College London, London WC1N 3BG, UK
5 Interacting Minds Centre, Aarhus University, 8000 Aarhus, Denmark
* Author to whom correspondence should be addressed.
† These authors contributed equally to this work.

Experimental psychiatry

doi.org/10.3390/e27010062

Who uses Turing?

Astrophysics > Cosmology and Nongalactic Astrophysics

[Submitted on 15 Aug 2025]

Frequentist Cosmological Constraints from Full-Shape Clustering Measurements in DESI DR1

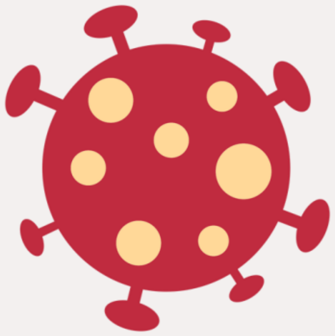
James Morawetz, Hanyu Zhang, Marco Bonici, Will Percival, Andrea Crespi, Jessica Nicole Aguilar, Steven Ahlen, Davide Bianchi, David Brooks, Francisco Javier Castander, Todd Claybaugh, Shaun Cole, Andrei Cuceu, Axel de la Macorra, Arnaud de Mattia, Biprateep Dey, Peter Doel, Simone Ferraro, Andreu Font-Ribera, Jaime E. Forero-Romero, Enrique Gaztañaga, Satya Gontcho A Gontcho, Gaston Gutierrez, ChangHoon Hahn, Klaus Honscheid, Dragan Huterer, Mustapha Ishak, Dick Joyce, Robert Kehoe, David Kirkby, Theodore Kisner, Ofer Lahav, Andrew Lambert, Martin Landriau, Laurent Le Guillou, Marc Manera, Ramon Miquel, Eva-Maria Mueller, Seshadri Nadathur, Jeffrey A. Newman, Gustavo Niz, Nathalie Palanque-Delabrouille, Francisco Prada, Ignasi Pérez-Ràfols, Graziano Rossi, Lado Samushia, Eusebio Sanchez, David Schlegel, Michael Schubnell, Joseph Harry Silber, David Sprayberry, Gregory Tarlé, Benjamin Alan Weaver, Pauline Zarrouk, Rongpu Zhou, Hu Zou

We perform a frequentist analysis using the standard profile likelihood method for clustering measurements from Data Release 1 of the Dark Energy Spectroscopic Instrument (DESI). While Bayesian inferences for Effective Field Theory models of galaxy clustering can be highly sensitive to the choice of priors for extended cosmological models, frequentist inferences are not susceptible to such effects. We compare Bayesian and frequentist constraints

Astrophysics

doi.org/10.48550/arXiv.2508.11811

Who uses Turing?



Open Access

May 2022

Interoperability of Statistical Models in Pandemic Preparedness: Principles and Reality

George Nicholson, Marta Blangiardo, Mark Briers, Peter J. Diggle, Tor Erlend Fjelde, Hong Ge, Robert J. B. Goudie, Radka Jersakova, Ruairidh E. King, Brieuc C. L. Lehmann, Ann-Marie Mallon, Tullia Padellini, Yee Whye Teh, Chris Holmes, Sylvia Richardson

Author Affiliations +

Statist. Sci. 37(2): 183-206 (May 2022). DOI: 10.1214/22-STS854

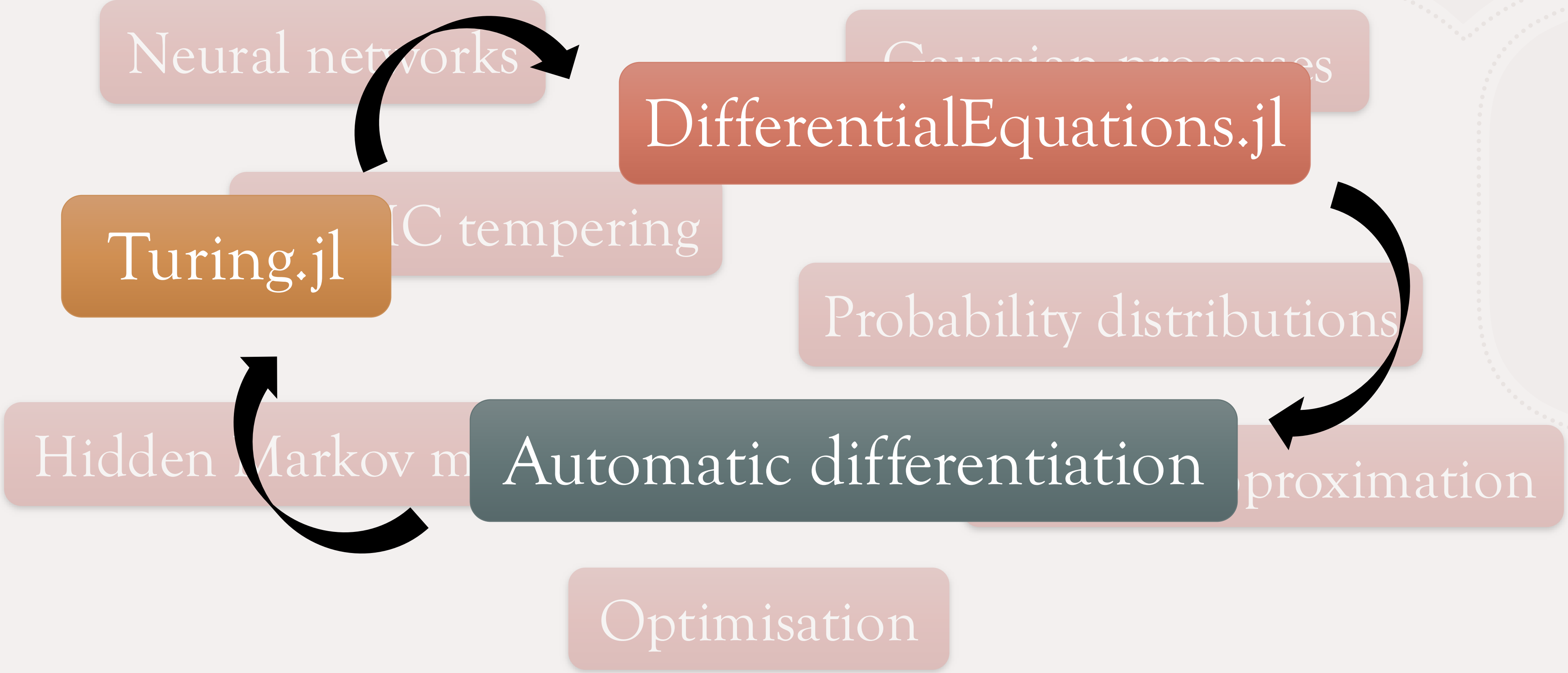
Epidemiology

doi.org/10.1214/22-sts854

Turing at the Turing

"Advance world-class research and apply it to national and global challenges: innovate and develop world-class research in data science and artificial intelligence that supports next generation theoretical developments and is applied to real-world problems, generating the creation of new businesses, services, and jobs."

The virtuous cycle of Julia (revisited)



Why we're all NUTS about Turing.jl

... and what you can infer from it ...

turinglang.org

Penelope Yong (*she/her*)
Research Software Engineer
1 October 2025