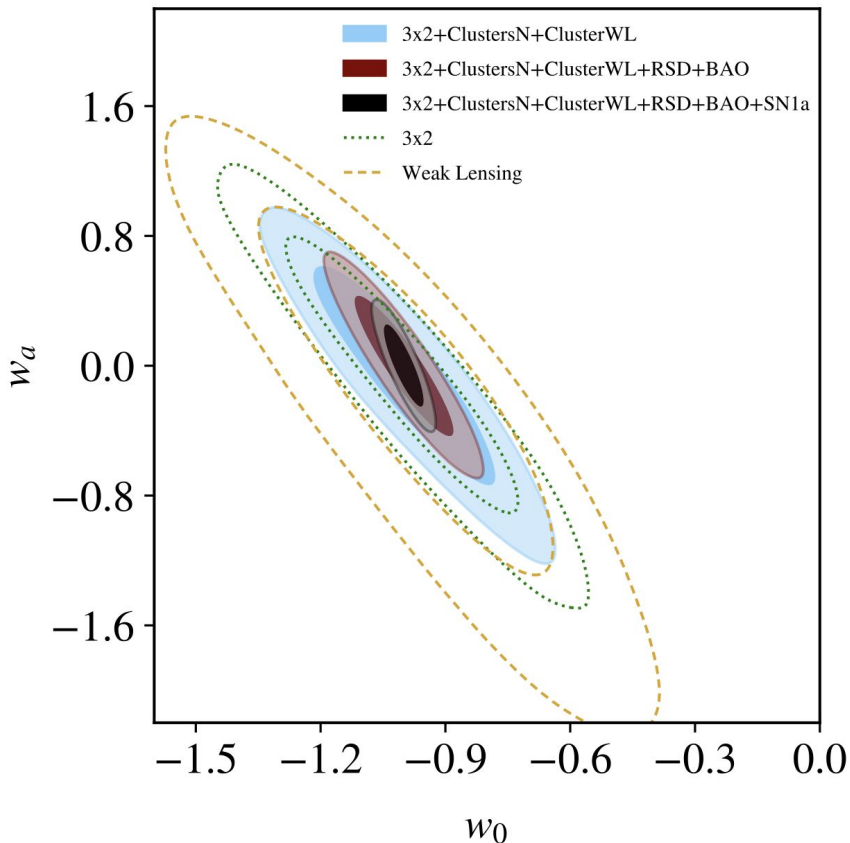


Cocoa overview

Cosmology always merges data sets



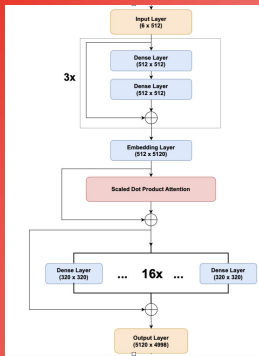
Cosmology with the Wide-Field Infrared Survey Telescope - Multi-Probe Strategies

Tim Eifler^{1,2*}, Hironao Miyatake^{2,3,4,5}, Elisabeth Krause^{1,6}, Chen Heinrich², Vivian Miranda¹, Christopher Hirata⁷, Jiachuan Xu¹, Shoubaneh Hemmati³, Melanie Simet^{2,8}, Peter Capak⁹, Ami Choi⁷, Olivier Doré^{2,10}, Cyrille Doux¹¹, Xiao Fang¹, Rebekah Hounsell¹¹, Eric Huff², Hung-Jin Huang¹, Mike Jarvis¹¹, Dan Masters², Eduardo Rozo⁶, Dan Scolnic¹³, David N. Spergel^{14,15}, Michael Troxel¹³, Anja von der Linden¹⁶, Yun Wang⁹, David H. Weinberg⁷, Lukas Wenzl¹⁷, Hao-Yi Wu⁷

- CIP obsession in this PIT: allow anyone to reproduce this plot (and to generalized to ANY smooth dark energy model) in < 1hr including EFT (MPS, IA, bias...)
 - Second goal: to hang out with you guys!
 - Third goal: to drink your coffee!

Cocoa - Cobaya-CosmoLike Architecture

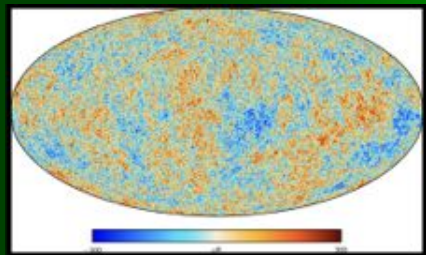
Machine Learning based emulators



- We can emulate any data vector
- 3x2pt, 6x2pt, CMB, BAO, SN, cluster lensing? Not a problem!
- Trans Learning: faster changes in settings
- Can emulate just CAMB outputs as well if cosmolike runtime not an issue for you

Cocoa - Cobaya-CosmoLike Architecture

Cobaya: Extensively used by CMB, BAO, SN community



Cobaya

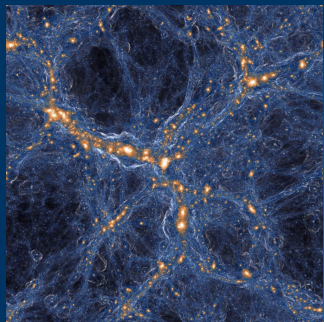


- Reimplementation of CosmoMC in Python 3
- Choose either CAMB or CLASS (or emulators develop by us for Cls & MPS)
- Convenient Yaml files for configuration (or even python files)
- Efficient inclusion of ext datasets
- Samplers: MH and Polychord

Cocoa - Cobaya-CosmoLike Architecture

CosmoLike: Focus on LSS modeling, systematics, covs

- Works with Class
- Awesome systematic modeling
- Used in DES-Y1, DESC SRD, Roman forecasting
- Text ini Files for configuration
- Limited External Data
- Samplers: Emcee and Multinest



CosmoLike



Cocoa - Cobaya-CosmoLike Architecture

Tension Monsters



NANCY GRACE
R.ÖMAN
SPACE TELESCOPE



Cocoa - Cobaya-CosmoLike Architecture

Cocoa = interfacing Cosmolike and Cobaya

Works with MacOS! Fast evaluation + Easy to use

Hybrid MPI/OpenMP - walkers on steroids

Total Evaluation - Boltzmann + Cosmolike (M2Pro)

(Roman w0wa 3x2pt w/ complex syst modeling - NLA)

- 1 OpenMP cores ~ 4 s (CAMB) + 1.85s (Cosmolike)
- 4 OpenMP cores ~ 1 s (CAMB) + 0.52s (Cosmolike)
- 8 OpenMP cores ~ 0.6 s (CAMB) + 0.31s (Cosmolike)

Total cores for a single chain: ~ 12 -32 cores

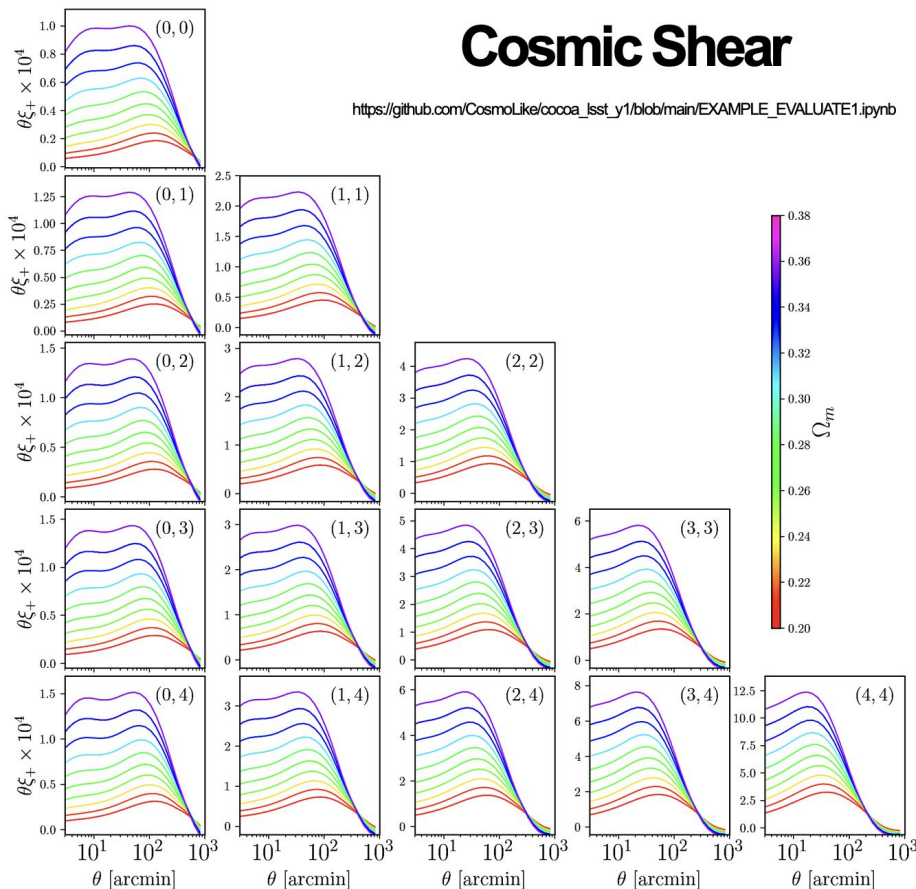
Reasonably easy to adapt to LCDM extensions



Our Jupyter Notebooks are getting better

Cosmic Shear

https://github.com/CosmoLike/cocoa_lsst_y1/blob/main/EXAMPLE_EVALUATE1.ipynb



Plot from
CoCoA, which
provides great
Jupyter
Notebook
support for
Cosmolike

You have total control of the packages

```
# This flag saves a lot of time when running setup_cocoa.py
# Why? Some git repos can be hundreds of MegaBytes (Class is 500 MegaBytes)
# But, this can create problem if GIT_COMMIT < LAST COMMIT - GIT_MAXIMUM_DEPTH
export GIT_CLONE_MAXIMUM_DEPTH=40

export COBAYA_URL="https://github.com/CobayaSampler/cobaya.git"
export COBAYA_GIT_COMMIT="86943d81d48d2edb2961b17077461df9e799f4d1"

export HILLIPOP_URL="https://github.com/planck-npipe/hillipop.git"
export HILLIPOP_GIT_COMMIT="cc9cbe31991d4662522241543a46d44d2cdec251"

export LOLLIPOP_URL="https://github.com/planck-npipe/lollipop.git"
export LOLLIPOP_GIT_COMMIT="280a9c93d33bc6a058d6bf769ec82d9f7fdbd2b6"

export LIPOP_DATA_URL="https://portal.nersc.gov/cfs/cmb/planck2020/likelihoods"
export LIPOP_DATA_VERSION=4.2

export SPT3G_DATA_URL='https://github.com/SouthPoleTelescope/spt3g_y1_dist.git'
export SPT3G_DATA_GIT_COMMIT="66da8e9e2f325024566fe13245788bf8ede897bc"
export SPT_3G_NAME="spt_3g"

export HOLICOW_DATA_URL='https://github.com/shsuyu/H0LiCOW-public.git'
export HOLICOW_DATA_GIT_COMMIT="f792647d1fd6c09d9e052fef526669cbd702ab82"

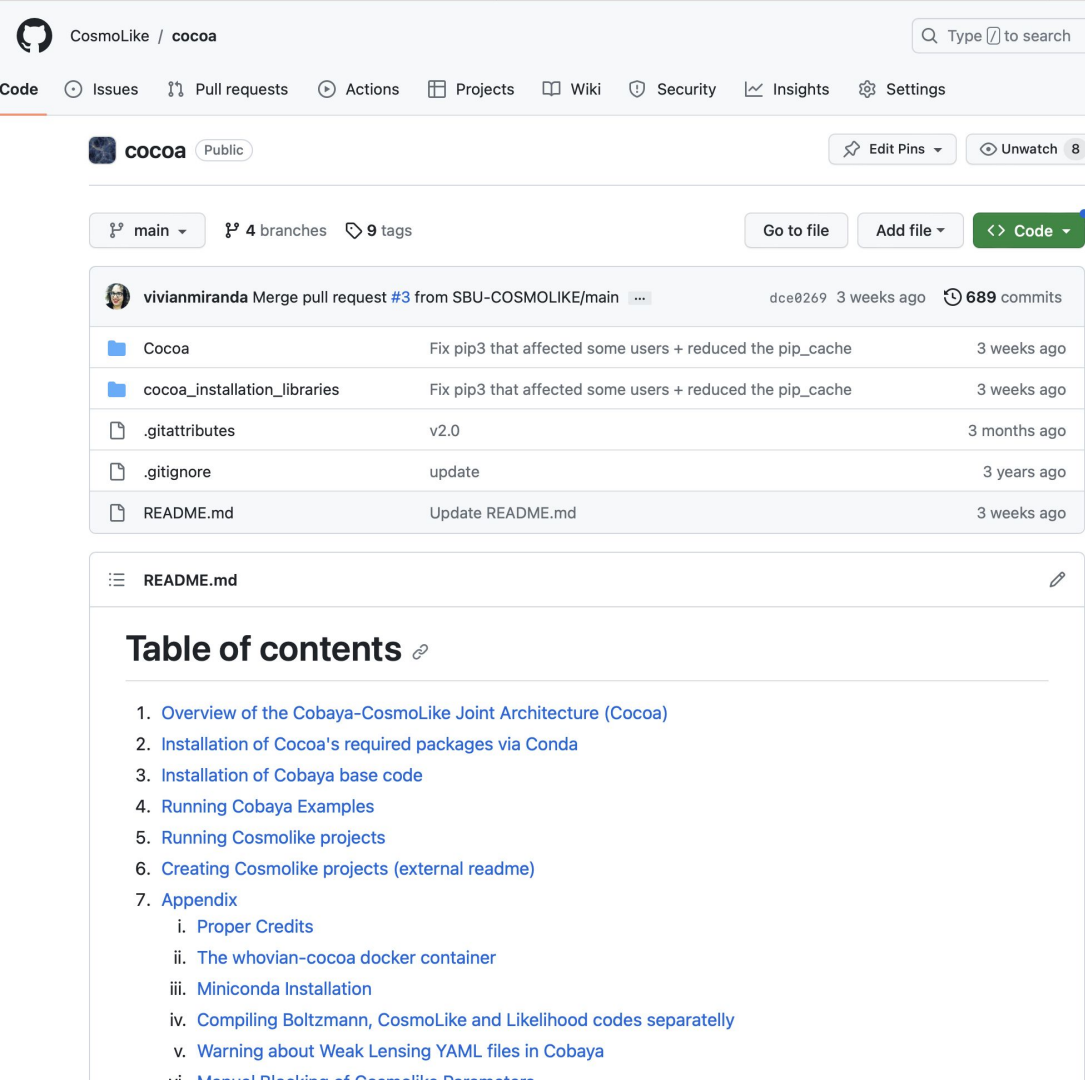
export POLY_URL="https://github.com/PolyChord/PolyChordLite.git"
export POLYCHORD_GIT_COMMIT="daba49d1385d065122db76a2b384050f9e95d278"
export POLY_NAME="PolyChordLite"
```

Cocoa does more than
merge Cobaya and
Cosmolike. It provides
an encapsulated
interface to install
packages, data, and
likelihoods compatible
with Cobaya.

ML learning integration w/ Cobaya is awesome!

```
emulcmb:
  path: ./cobaya/cobaya/theories/
  stop_at_error: True
  extra_args:
    # This version of the emul was not trained with CosmoRec
    eval: [True, True, True, True] #TT,TE,EE,PHIPHI
    device: "cuda"
    file: ['projects/axions/data/CMB_CNN/emul_ax_DE_CMBTT_CNN.pt',
           'projects/axions/data/CMB_CNN/emul_ax_DE_CMBTE_CNN.pt',
           'projects/axions/data/CMB_CNN/emul_ax_DE_CMBEE_CNN.pt',
           'projects/axions/data/CMB_CNN/emul_ax_DE_phi_ResMLP.pt']
    extra: ['projects/axions/data/CMB_CNN/extra_ax_DE_CMBTT.npy',
            'projects/axions/data/CMB_CNN/extra_ax_DE_CMBTE.npy',
            'projects/axions/data/CMB_CNN/extra_ax_DE_CMBEE.npy',
            'projects/axions/data/CMB_CNN/extra_ax_DE_phi.npy']
    ord: [ ['omegab2', 'omegab2', 'H0', 'logA', 'ns', 'tau', 'omegaax2', 'logmx'],
            ['omegab2', 'omegab2', 'H0', 'logA', 'ns', 'tau', 'omegaax2', 'logmx'],
            ['omegab2', 'omegab2', 'H0', 'logA', 'ns', 'tau', 'omegaax2', 'logmx'],
            ['omegab2', 'omegab2', 'H0', 'logA', 'ns', 'tau', 'omegaax2', 'logmx']]
    extrapar: [{'ellmax': 3000, 'MLA': 'CNN', 'INTDIM': 4, 'INTCNN': 3200},
                {'ellmax': 3000, 'MLA': 'CNN', 'INTDIM': 4, 'INTCNN': 3200},
                {'ellmax': 3000, 'MLA': 'CNN', 'INTDIM': 4, 'INTCNN': 3200},
                {'MLA': 'ResMLP', 'INTDIM': 6, 'NLAYER': 6,
                 'TMAT': 'projects/axions/data/CMB_CNN/PCA_ax_DE_phi.npy'}]
```

Emulators act as
theory blocks (like
CAMB or CLASS)



CosmoLike / cocoa

Search Type to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

cocoa Public

Edit Pins Unwatch 8

main 4 branches 9 tags

Go to file Add file Code

vivianmiranda Merge pull request #3 from SBU-COSMOLIKE/main dce0269 3 weeks ago 689 commits

Cocoa	Fix pip3 that affected some users + reduced the pip_cache	3 weeks ago
cocoa_installation_libraries	Fix pip3 that affected some users + reduced the pip_cache	3 weeks ago
.gitattributes	v2.0	3 months ago
.gitignore	update	3 years ago
README.md	Update README.md	3 weeks ago

README.md

Table of contents

- Overview of the Cobaya-CosmoLike Joint Architecture (Cocoa)
- Installation of Cocoa's required packages via Conda
- Installation of Cobaya base code
- Running Cobaya Examples
- Running Cosmolike projects
- Creating Cosmolike projects (external readme)
- Appendix
 - Proper Credits
 - The whovian-cocoa docker container
 - Miniconda Installation
 - Compiling Boltzmann, CosmoLike and Likelihood codes separately
 - Warning about Weak Lensing YAML files in Cobaya
 - Manual Blocking of Cosmolike Parameters

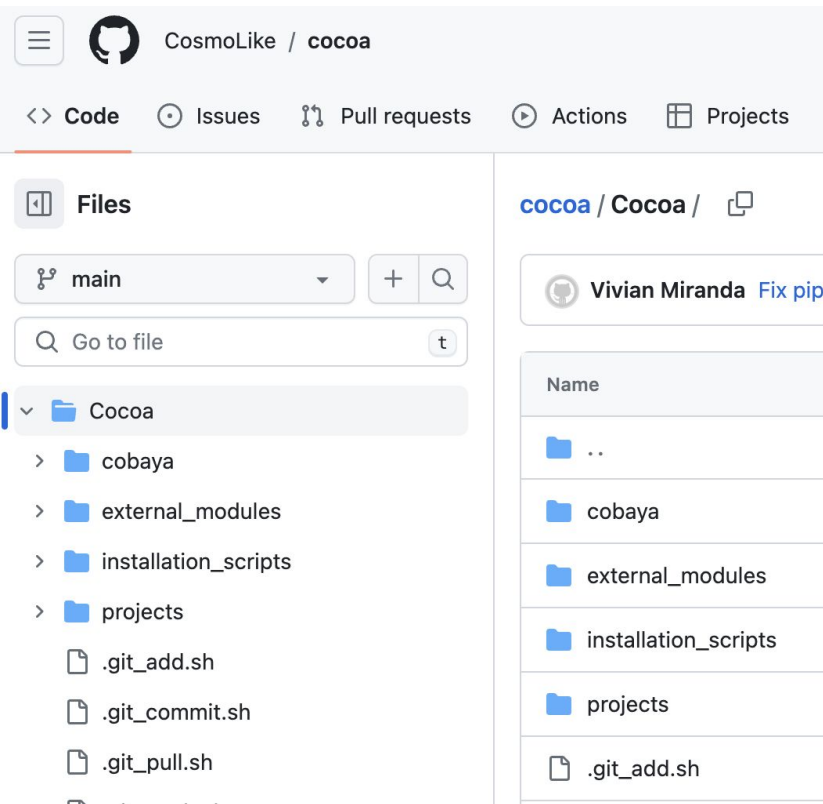
Cocoa - Cobaya-CosmoLike Architecture

Documentation on Github

[Cosmolike/cocoa](https://github.com/CosmoLike/cocoa)

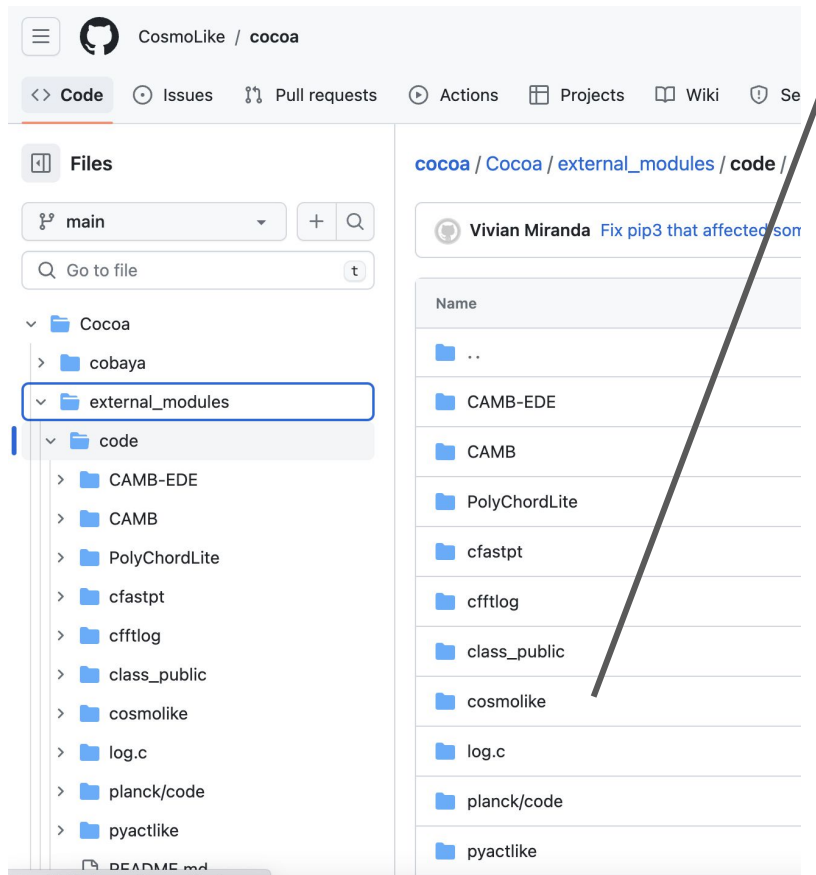
Includes installation steps +
Docker image if Conda not
available (ask us if you have
any questions)

Let's dig into Cobaya (This structure is not our fault!)



- Cobaya requires a very specific file structure
- Integrating complex analyses of correlated external probes is challenging
- Many choices downstream are driven by the mandated Cobaya file structure
- Cocoa “Projects” have their own repo (see next slide)
- We use symlinks to map data and code from specific projects into the right places required by cobaya

Cocoa - Cobaya-CosmoLike Architecture



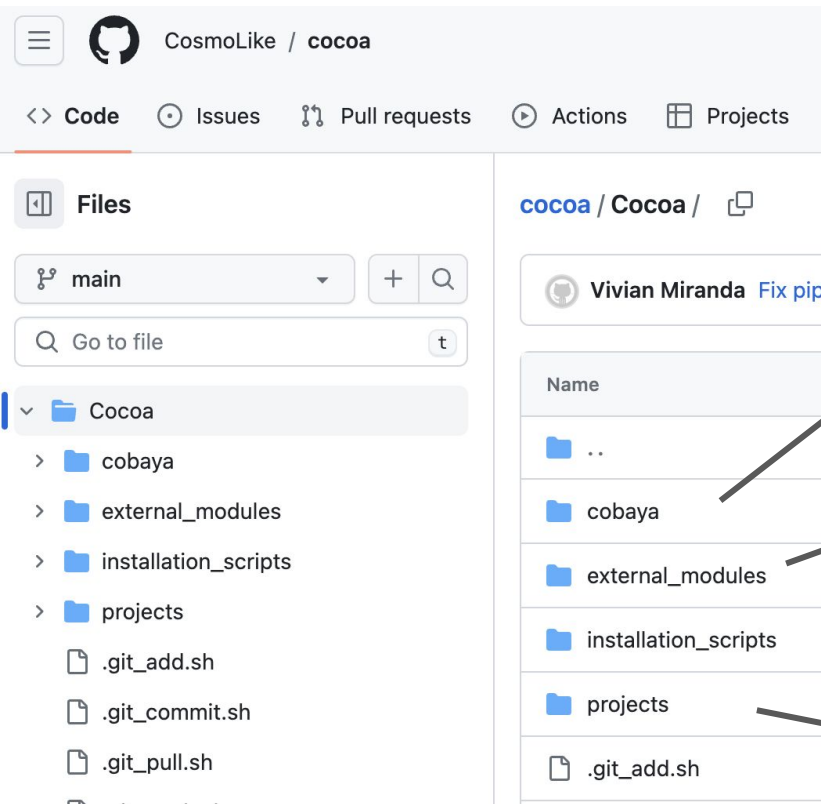
CosmoLike routines: in another git repo (cocoa_cosmolike_core). After installation it is located at *external_modules*

Tested against the DES-Y3 cosmlike, which was tested against cosmosis

Does not have all features: e.g. To, Krause 6x2+N routines, and 10x2 Fang et al 23 routines, many other devs as well

Nevertheless it's one of our most current versions for 3x2 data analysis and our goal should be to use that for Roman forecasts as well (we used on DC1!)

Let's dig into Cobaya (This structure is not our fault!)



Abstract Description

Sampler interface,
Theory interface,
Likelihood Interface

External code (libraries)
and data (files) the
interfaces link to

Specific to cosmlike
projects. All the details of
any given project
(datasets, covariances,
yaml files, etc)

Examples

Polychord interface,
Camb interface,
Planck interface

Planck Data, DES Data
Planck code, cosmlike
library, but also sampler
code like polychord etc

DES Y3 analyses, LSST
and Roman forecasts,
several LSSxCMB, etc
whatever project you
want.

Emulator: collection of surrogate models

- Aim: to run CMB + BAO + SN (Roman) + Weak Lensing (Roman in real and Fourier Space).
- Implemented within Cocoa
- (<https://github.com/CosmoLike/cocoa>)
- Emulators is actually a collection of emulators
 - CMB: TT, TE, EE, PhiPhi (4 Emulator)
 - Matter Power Spectrum (1 Emulator)
 - Distance and $H(z)$ (2 Emulators)
 - R_{drag} (1 Emulator): necessary for BAO
 - Θ^* -> H_0 mapping (1 Emulator): imp for CMB sampling.



Table of contents

1. [Overview](#)
2. [Installation of core packages](#)
3. [Installation and Compilation of external modules](#)
4. [Running Examples](#)
5. [Running ML emulators](#)
6. [Creating Cosmolike projects \(external readme\)](#)
7. [Appendix](#)
 - i. [FAQ: How do we debug Cocoa? Suggested steps](#)
 - ii. [FAQ: How do we compile external modules?](#)
 - iii. [FAQ: How do we run Cocoa with Docker?](#)
 - iv. [FAQ: How do we use an available Anaconda module on HPC?](#)
 - v. [FAQ: How do we install Conda?](#)
 - vi. [FAQ: How do we set the environment for Machine Learning projects?](#)
 - vii. [FAQ: How do we push changes to the Cocoa main branch?](#)
 - viii. [FAQ: How do we develop from a Git tag? A few more Git hacks](#)
 - ix. [FAQ: How do we download additional likelihood data? \(external readme\)](#)
 - x. [FAQ: Where do we find common FAQs about external modules? \(external readme\)](#)
 - xi. [FAQ: Where do we find common FAQs about Cosmolike? \(external readme\)](#)
 - xii. [FAQ: How can we improve our Bash/C/C++ knowledge?](#)
 - xiii. [Credits](#)

Emulator: how well are they tested?

- Extensive testing on this series of papers
 - The main metric is delta chi2
 - What we aim? >90% of testing points with delta chi2 < 0.2.
 - delta chi2 < 0.2 guarantees no biases on Bayesian and Frequentist metrics
 - Expensive to train to reach such precision. We only train once per model.
 - Once trained: MCMC converge in O(10 mins)

Attention-based Neural Network Emulators for Multi-Probe Data Vectors Part I: Forecasting the Growth-Geometry split

Kunhao Zhong,^{1,2*} Evan Saraivanov¹, James Caputi¹, Vivian Miranda³, Supranta S. Boruah⁴, Tim Eifler⁴, Elisabeth Krause^{4,5}

¹Department of Physics and Astronomy, Stony Brook University, Stony Brook, NY 11794, USA

²Department of Physics and Astronomy, University of Pennsylvania, Philadelphia, PA 19104, USA

³C. N. Yang Institute for Theoretical Physics, Stony Brook University, Stony Brook, NY 11794, USA

⁴Department of Astronomy and Steward Observatory, University of Arizona, 933 N Cherry Ave, Tucson, AZ 85719, USA

⁵Department of Physics, University of Arizona, 1118 E Fourth Str, Tucson, AZ, 85721-0065, USA

Attention-Based Neural Network Emulators for Multi-Probe Data Vectors Part II: Assessing Tension Metrics

Evan Saraivanov,^{1,*} Kunhao Zhong,^{1,2} Vivian Miranda,³ Supranta S. Boruah,^{4,2} Tim Eifler,^{4,5} and Elisabeth Krause^{4,5}

¹Department of Physics and Astronomy, Stony Brook University, Stony Brook, NY 11794, USA

²Department of Physics and Astronomy, University of Pennsylvania, Philadelphia, PA 19104, USA

³C. N. Yang Institute for Theoretical Physics, Stony Brook University, Stony Brook, NY 11794, USA

⁴Department of Astronomy and Steward Observatory, University of Arizona, 933 N Cherry Ave, Tucson, AZ 85719, USA

⁵Department of Physics, University of Arizona, 1118 E Fourth Str, Tucson, AZ, 85721-0065, USA

(Dated: March 22, 2024)

Attention-based Neural Network Emulators for Multi-Probe Data Vectors Part III: Modeling The Next Generation Surveys

Yijie Zhu,¹ Evan Saraivanov,^{1,2} Joshua A. Kable,² Artemis Sofia Giannakopoulou,^{1,2} Amritpal Nijjar,¹ Vivian Miranda,^{1,2} Marco Bonici,³ Tim Eifler,^{4,5} and Elisabeth Krause^{4,5}

¹Department of Physics and Astronomy, Stony Brook University, Stony Brook, NY 11794, USA

²C. N. Yang Institute for Theoretical Physics, Stony Brook University, Stony Brook, NY 11794, USA

³Waterloo Centre for Astrophysics, ON, Canada N2L 3G1

⁴Department of Astronomy and Steward Observatory,

University of Arizona, 933 N Cherry Ave, Tucson, AZ 85719, USA

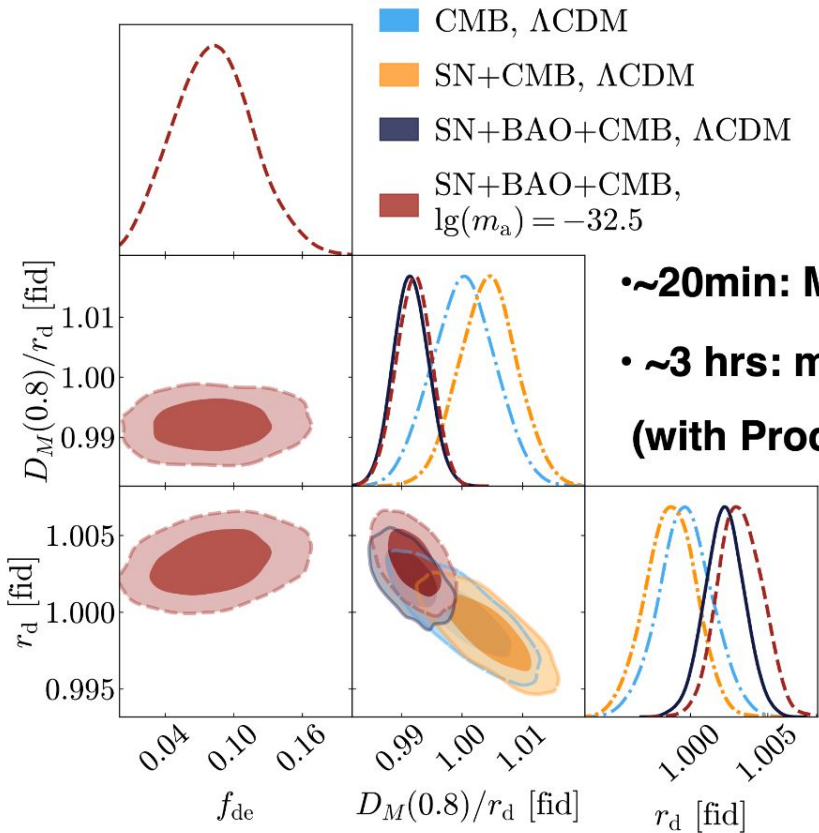
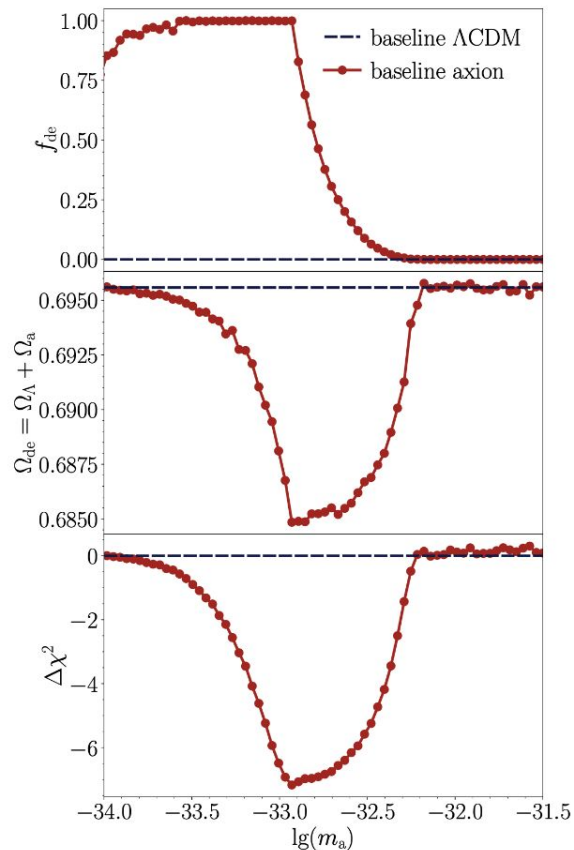
⁵Department of Physics, University of Arizona, 1118 E Fourth Str, Tucson, AZ, 85721-0065, USA

(Dated: June 11, 2025)

Want to design a new model? No problem

Liu, Zhu, Hu, Miranda, arxiv: 2510.14957

ULA DE

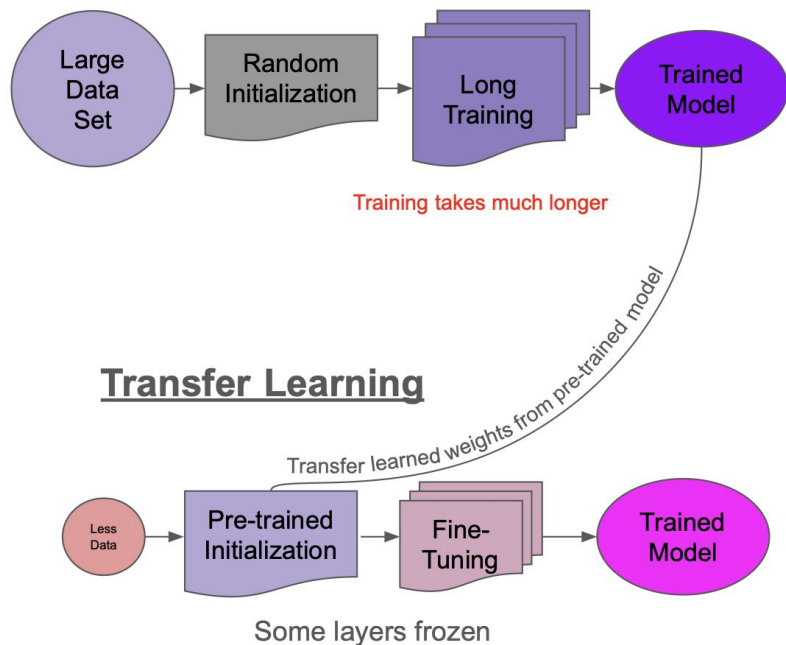


• ~20min: MCMC

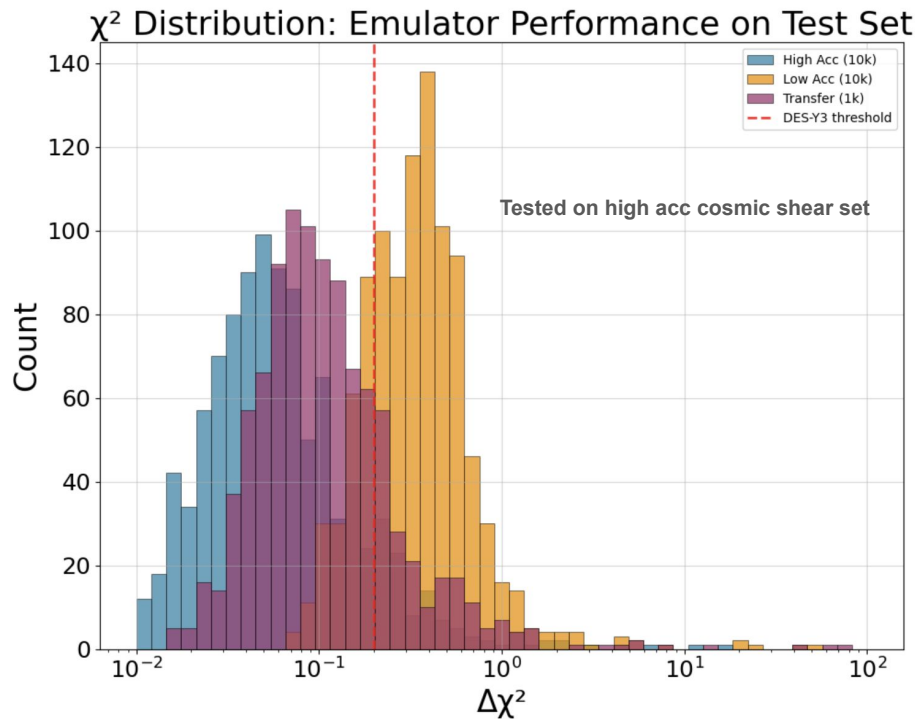
• ~3 hrs: mass scan Profile (~80pts)
(with Procolli)

- Plk TTTEEE lite
- low TT, (low EE)
- Pantheon+
- DESI
- ACT lensing

Next stage of CPIP Emulator pipeline dev: **Transfer Learning**



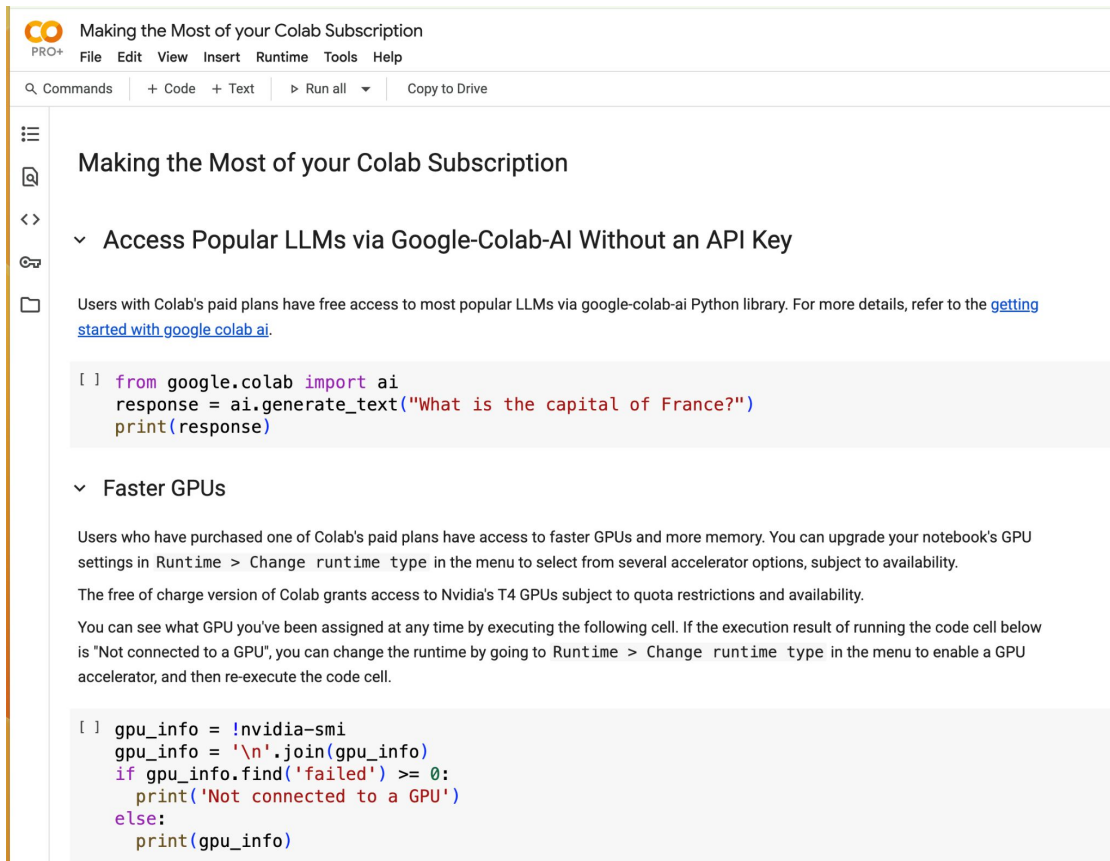
- Low acc to High acc
- Swap Recombination or Nonlinear
- Extended Parameters
- Jump to other observables?



Preliminary test: Freezing 2 early layers, only 1k high acc data needed to transfer learn from low acc emulator

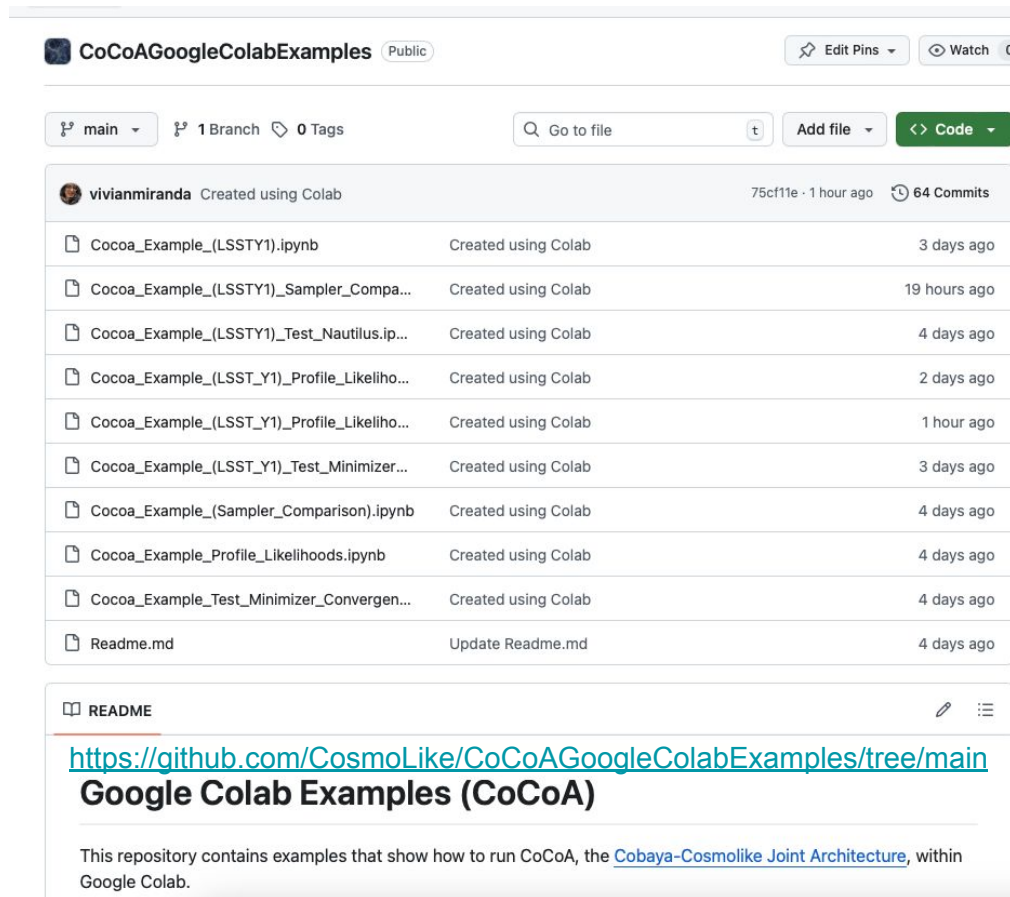
What is Google Colab?

- Google educational service that provides TPU (tensor processing unit) and GPU (nvidia A100) access to anyone with a google account.
- Free and Paid Tiers (difficult to get the A100 on the free tier). Plans start at 10 and 50 dollars
- Users can purchase additional credit units
- E.g., with the use of Emulators, we can provide pipeline examples on how to reproduce some of results at multiple levels in the analysis



What is Google Colab?

- Idea is for the PIT to create repositories like this one
- What about CPUs? Every notebook has 2 CPUs. So for this to work we need computation on GPU (e.g., emulators)
- 24 hour limit computation on paid tiers. With the 50 dollar / month account, there is background execution
- However we can save checkpoints on Google drive.



The screenshot shows a GitHub repository page for 'CoCoAGoogleColabExamples'. The repository is public and has 1 branch and 0 tags. It contains a list of notebooks created using Google Colab, all of which were created using Colab. The notebooks are listed in a table with columns for the file name, the creation method, and the time since creation. The files include 'Cocoa_Example_(LSSTY1).ipynb', 'Cocoa_Example_(LSSTY1)_Sampler_Compa...', 'Cocoa_Example_(LSSTY1)_Test_Nautilus.ip...', 'Cocoa_Example_(LSST_Y1)_Profile_Likeliho...', 'Cocoa_Example_(LSST_Y1)_Profile_Likeliho...', 'Cocoa_Example_(LSST_Y1)_Test_Minimizer...', 'Cocoa_Example_(Sampler_Comparison).ipynb', 'Cocoa_Example_Profile_Likelihoods.ipynb', 'Cocoa_Example_Test_Minimizer_Convergen...', and 'Readme.md'. Below the table, there is a 'README' section with a link to the repository's main page and a description of the repository's content.

File Name	Created using Colab	Time
Cocoa_Example_(LSSTY1).ipynb	Created using Colab	3 days ago
Cocoa_Example_(LSSTY1)_Sampler_Compa...	Created using Colab	19 hours ago
Cocoa_Example_(LSSTY1)_Test_Nautilus.ip...	Created using Colab	4 days ago
Cocoa_Example_(LSST_Y1)_Profile_Likeliho...	Created using Colab	2 days ago
Cocoa_Example_(LSST_Y1)_Profile_Likeliho...	Created using Colab	1 hour ago
Cocoa_Example_(LSST_Y1)_Test_Minimizer...	Created using Colab	3 days ago
Cocoa_Example_(Sampler_Comparison).ipynb	Created using Colab	4 days ago
Cocoa_Example_Profile_Likelihoods.ipynb	Created using Colab	4 days ago
Cocoa_Example_Test_Minimizer_Convergen...	Created using Colab	4 days ago
Readme.md	Update Readme.md	4 days ago

README

<https://github.com/CosmoLike/CoCoAGoogleColabExamples/tree/main>

Google Colab Examples (CoCoA)

This repository contains examples that show how to run CoCoA, the [Cobaya-Cosmolike Joint Architecture](#), within Google Colab.

Notebook Example (Sampler Comparison)

- [https://colab.research.google.com/github/SBU-COSMOLIK/E/CoCoAGoogleColabExamples/blob/main/Cocoa_Example_\(Sampler_Comparison\).ipynb](https://colab.research.google.com/github/SBU-COSMOLIK/E/CoCoAGoogleColabExamples/blob/main/Cocoa_Example_(Sampler_Comparison).ipynb)

- I used this Collab to generate this plot on the equivalence of multiple samplers

