Collaborative Research Center TRR 257
Particle Physics Phenomenology after the Higgs Discovery

Parametrising profile likelihoods with neural networks. (work in progress)

Humberto Reyes-González
RWTH Aachen

with J. Araz (JLab), S. Kraml, Rafal Maselek (LPSC Grenoble), W. Waltenberger (HEPHY OAW),...

6th IML workshop
CERN, 30/1/2024

Introduction

- Likelihood functions (full statistical models) parametrise the full information of an LHC analysis; wether it is New Physics (NP) search or an SM measurement. arXiv:2109.04981
- Their preservation is a key part of the LHC legacy.

- **Usage:** Reinterpretation in the context of different NP models and/or with with different statistical approaches, resampling, education, reproducibility….

A brief story on full statistical model publication and usage:

- ATLAS started publishing full likelihoods of NP searches (2019) ATL-PHYS-PUB-2019-029.

- Release of the pyhf package to construct statistical models (2020) 10.21105/joss.02823

- Interface with reinterpretation tools: SmodelS (2020) arXiv:2009.01809, MadAnalysis (2022) arXiv:2206.14870,

- Spey: Generalised framework for likelihood handling (2023) arXiv:2307.06996.

Why Machine Learning Profile Likelihoods?

- In LHC-reinterpretation, to exclude a BSM model, we are mostly interested in the profiled likelihood given a signal strength.

- Optimally, we can compute the profiled likelihood from pyhf's full statistical models.

- However, the relevant computations can take several minutes per parameter point. A pheno study often requires to survey thousands of points.

- This considerably scales-up the time consumption. Specially for fast reinterpretation approaches.

- **Using Neural Networks provides a fast and compact way using profiled likelihoods in our day-to-day pheno studies.**

- **We will super useful for anomaly surveys a lá proto-models (arXiv:2105.09020).**

LHC likelihoods in a nutshell

Bayes theorem:

$$P(\Theta, x) = \underbrace{P_x(x\,|\,\Theta)}_{\text{Likelihood}}\underbrace{\pi_\Theta(\Theta)}_{\text{Prior}} = \underbrace{P_\Theta(\Theta\,|\,x)}_{\text{Posterior}}\underbrace{\pi_x(x)}_{\text{Evidence}}$$

LHC Statistical model:

$$P(\boxed{\mu}, \boxed{\theta}; \boxed{\text{data}}) = \Pi_{k=1}^{n_c} P[\boxed{n_i}; \boxed{\mu}\epsilon_{i,k}\boxed{(\vec{\theta})}N_{S,i,k}\boxed{(\vec{\theta})} + B_{i.k}\boxed{(\vec{\theta})}]\Pi_{j=1}^{n_{syst}} G(\theta_j\boxed{^{obs}}; \boxed{\theta_j; 1)}$$

signal/control yields

Nuisance parameters (uncertainties)

Parameters of Interest (signal strength, observables, etc.)

(Observed) data

(Auxiliary) data

With this we perform global fits, exclude BSM models, find upper limits, search for SM deviations, etc.

The Profile Likelihood

- For new physics searches reinterpretation, we are often interested in **the Profile Likelihood (PL)**.
- The PL is defined as a function where the nuisance parameters are fixed such that the likelihood is maximised given a signal strength $\boldsymbol{\mu}$.
- In the case of positive signal $\boldsymbol{\mu}$=1, otherwise if data is Standard Model like $\boldsymbol{\mu}$=0.
- The PL is a function of the signal yields (data), $\boldsymbol{n_s}$.

$$\longrightarrow \quad P(x \,|\, \mu; \hat{\theta}(\mu))$$

- With the PL we construct **Log Likelihood Ratio (LLR) tests.**
- Depending on how the PL is fitted and defined, we can derive upper limits, exclusion confidence levels and discoveries.

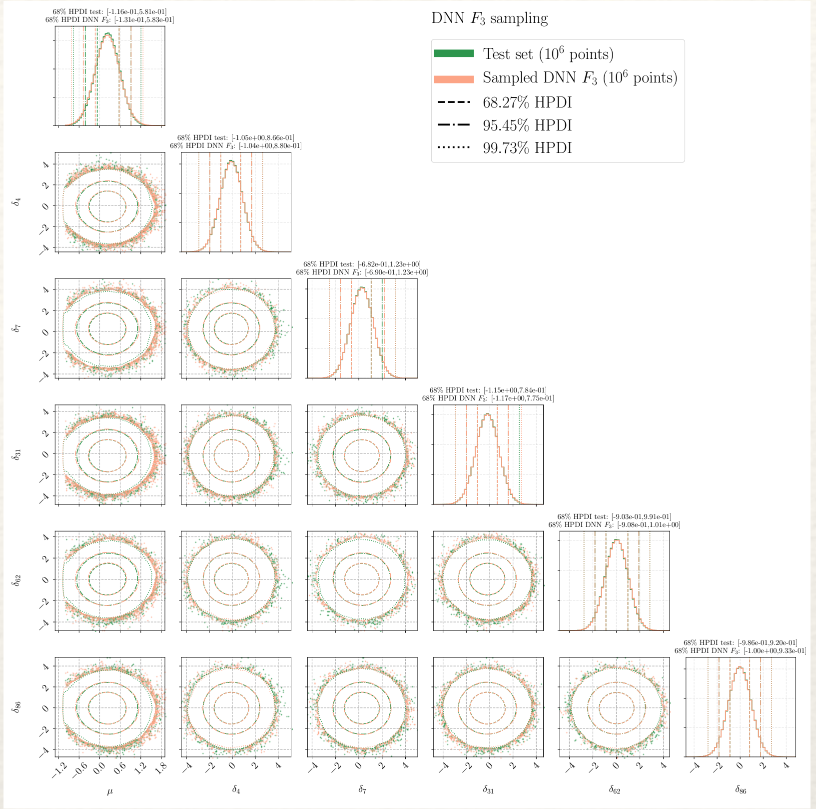
$$\longrightarrow \quad t(\mu) = -2\log\frac{L(\mu;\hat{\theta}(\mu))}{L(\hat{\mu},\hat{\theta}(\hat{\mu}))}$$

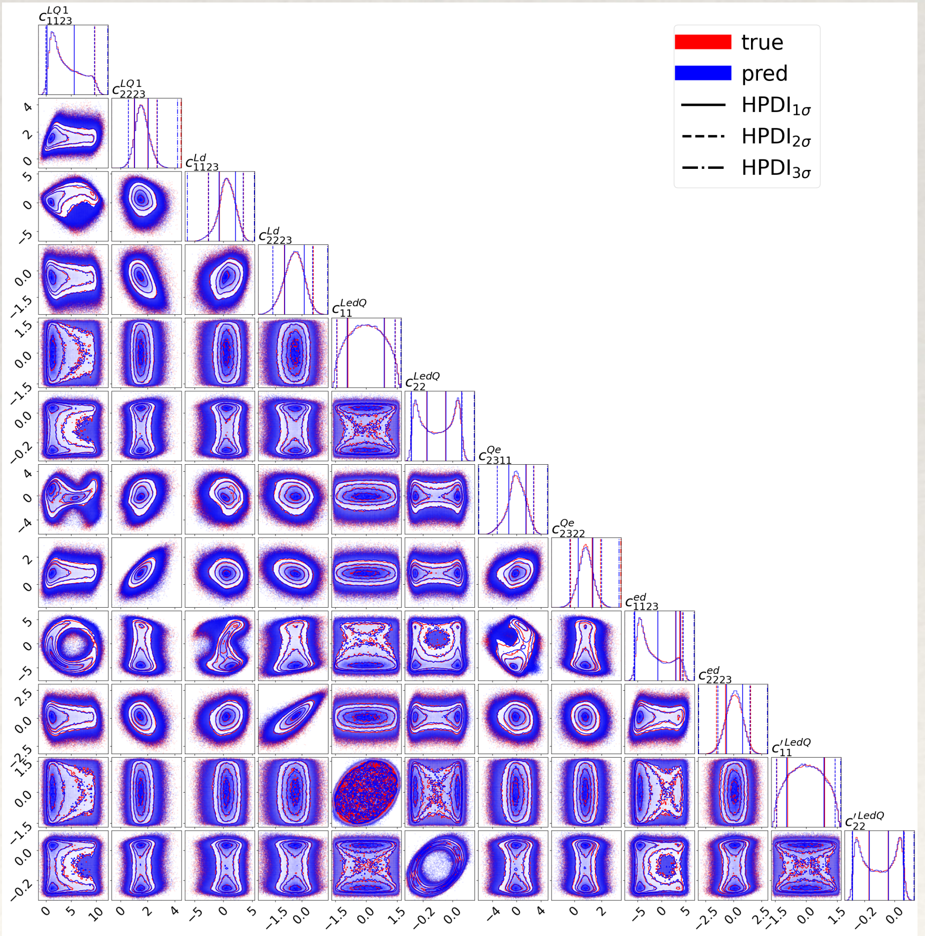
We will learn Likelihood Observed and Excluded (fit to μ = 1,0).

Previous work on learning LHC Likelihoods

$$P_{\Theta}(\Theta \,|\, x = \text{obs})$$

- **DNNLikelihood**
- Supervised Learning with Deep Neural Networks.
- 95-dim LHC-like toy likelihood.
- arXiv:1911.03305





- **NFLikelihood.**
- Unsupervised Learning with Normalising Flows.
- Same from DNNLikelihood
- 2 Likelihoods of EFT global fits: EW (40 dims) and Flavor (89 dims)
- arXiv:2309.09743

Example Likelihoods

$P(x \mid \mu; \hat{\theta}(\mu))$

ATLAS-SUSY-2018-04

- Search for direct stau production in events with two τ-leptons
- Number of SRs: 2
- DOI: 10.1103/PhysRevD.101.032009

ATLAS-SUSY-2019-08

- Search for direct production of e-winos in final states with 1 lepton, MET and a Higgs boson decaying into 2 -jets
- Number of SRs: 9.
- DOI: https://doi.org/10.17182/hepdata.90607.v4

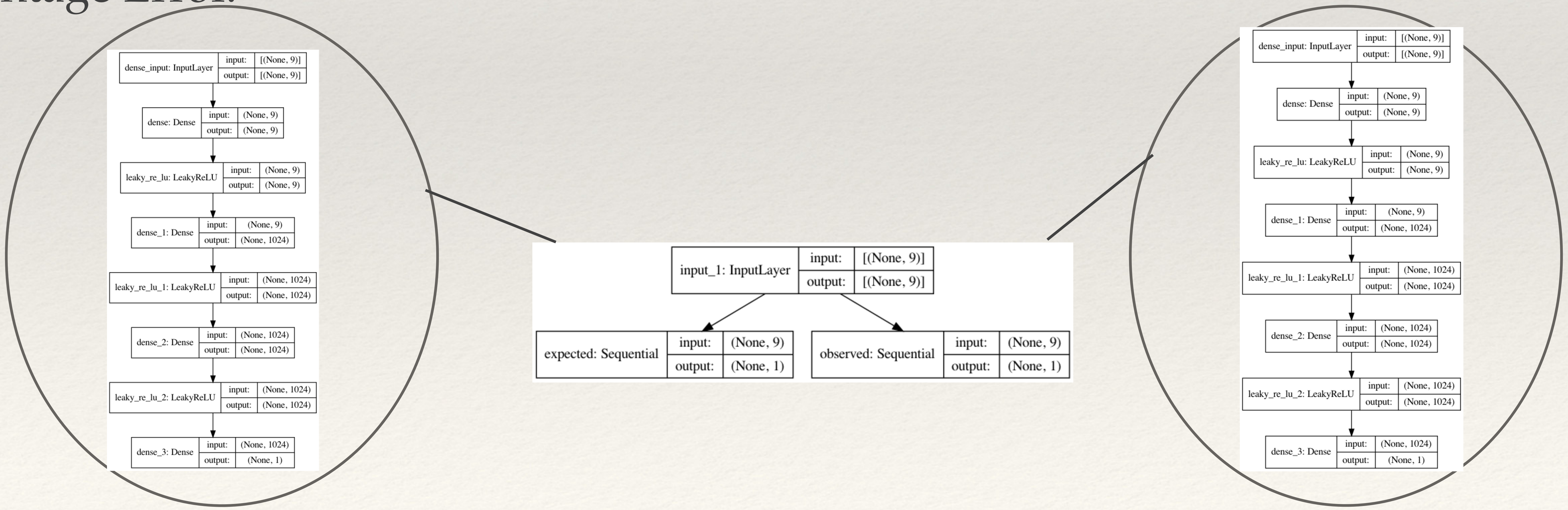
Training strategy.

- **Sampling**: MCMC Metropolis-Hastings runs towards the min(L) and max(L), to cover the full parameter space

- The **Input** is n_s and the **Output** $-\ln(L)$.

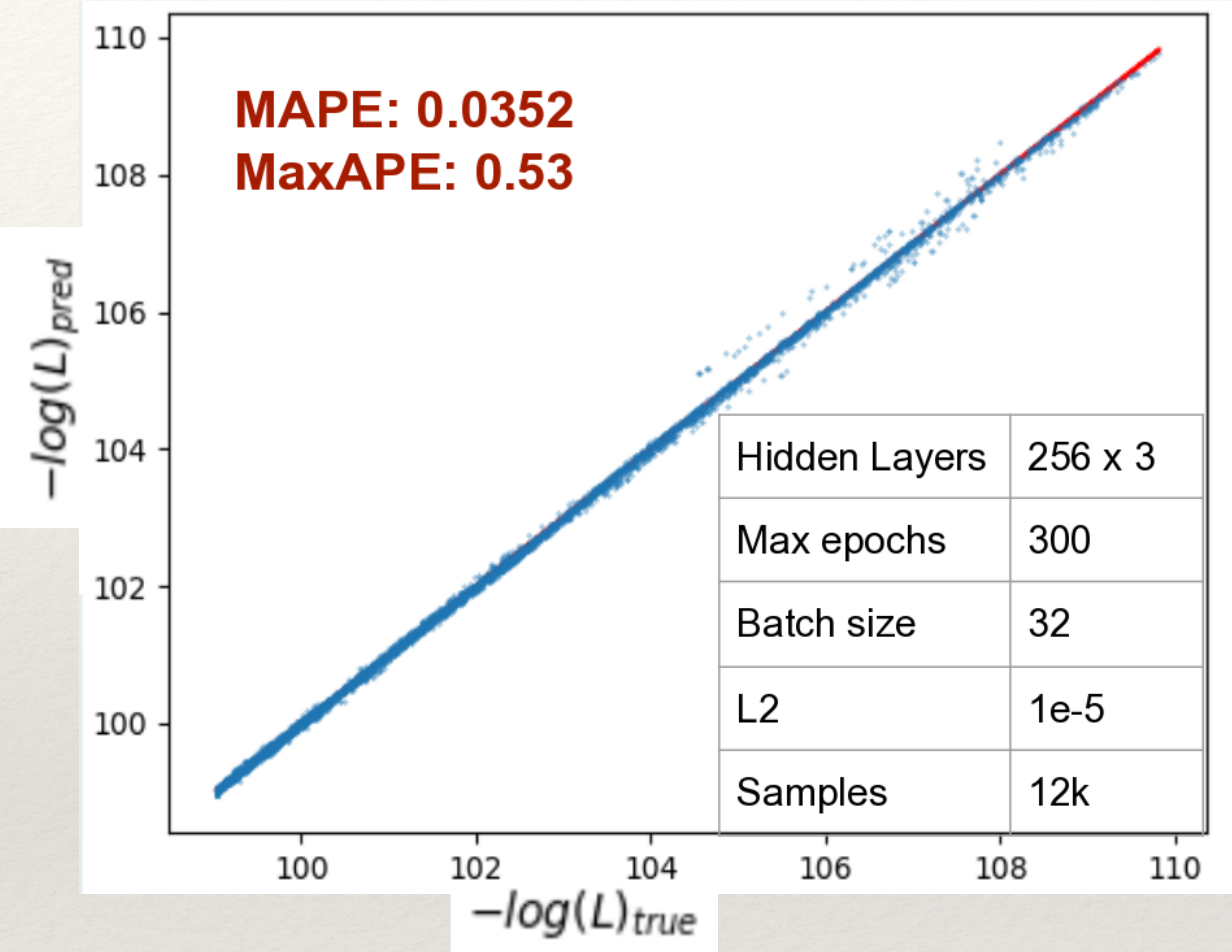
- **Training:** All models were Multi-Layer Perceptrons (MPE) trained using Mean Squared Error loss function, ADAM optimiser and LeakyReLU activation functions. Data was divided as training-validation-test on a 60-20-20 scheme. Observed and Expected Likelihood functions are trained separately.

- **Testing:** The accuracy of the NN models was measured with the Mean (MAPE) and Max (MaxAPE) Absolute Percentage Error.



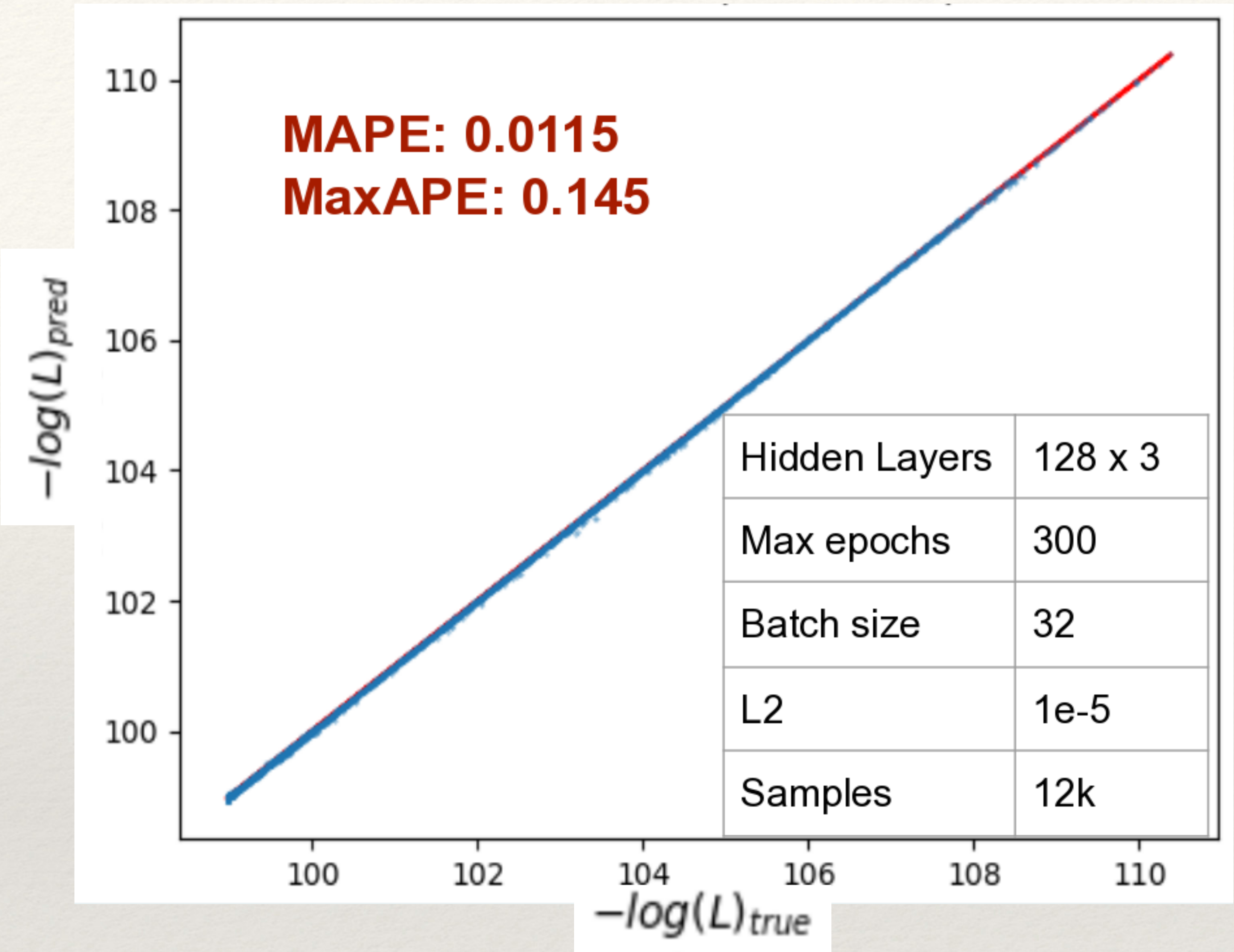
Results: ATLAS-SUSY-2018-04, 2 SRS

OBSERVED

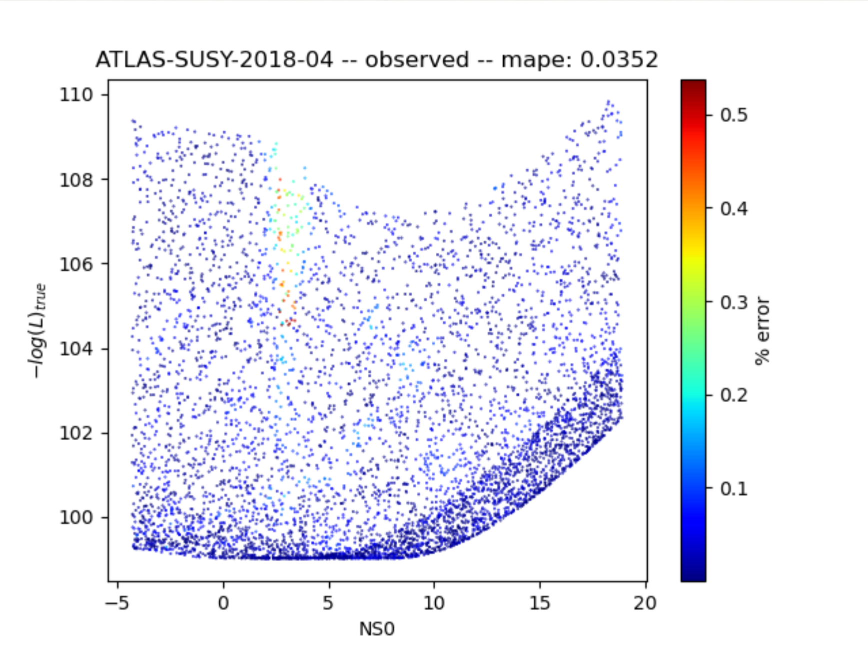


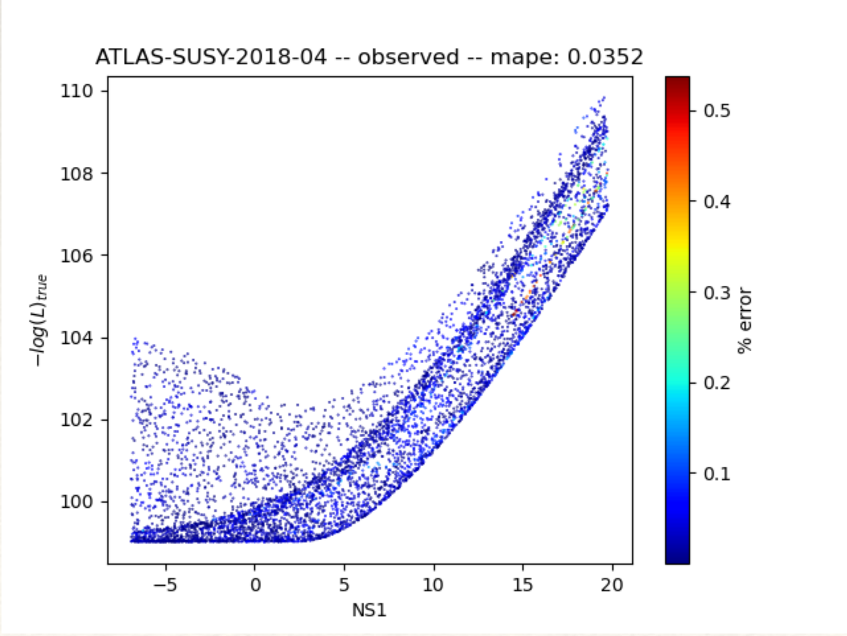
Hidden Layers	256 x 3
Max epochs	300
Batch size	32
L2	1e-5
Samples	12k

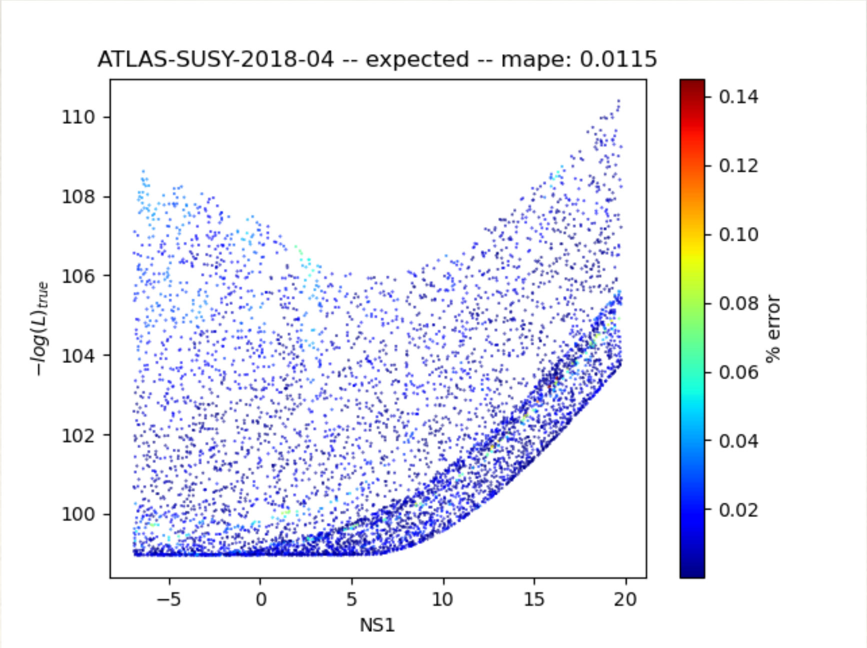
EXPECTED

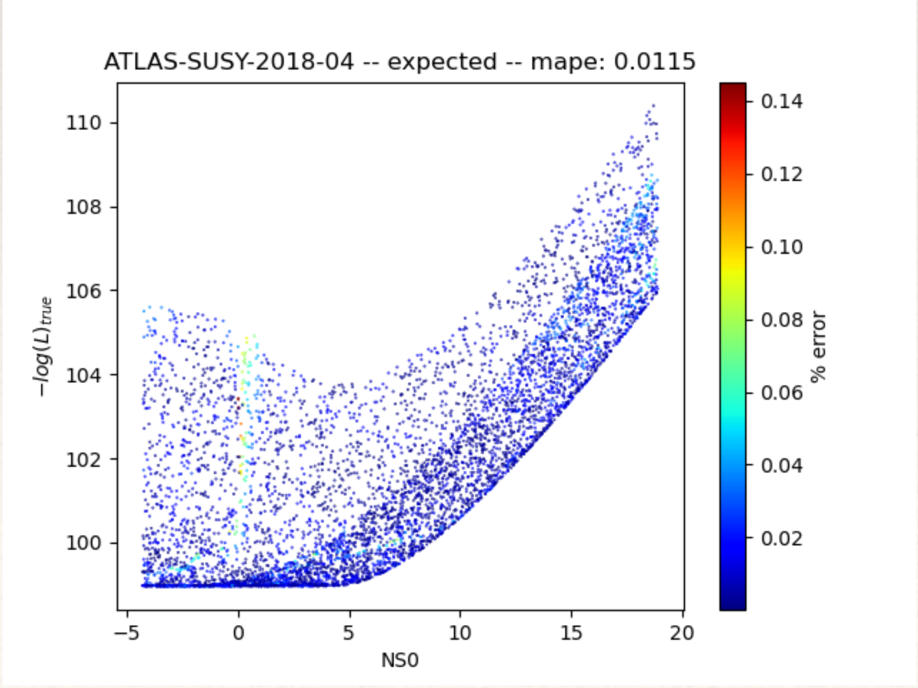


Hidden Layers	128 x 3
Max epochs	300
Batch size	32
L2	1e-5
Samples	12k



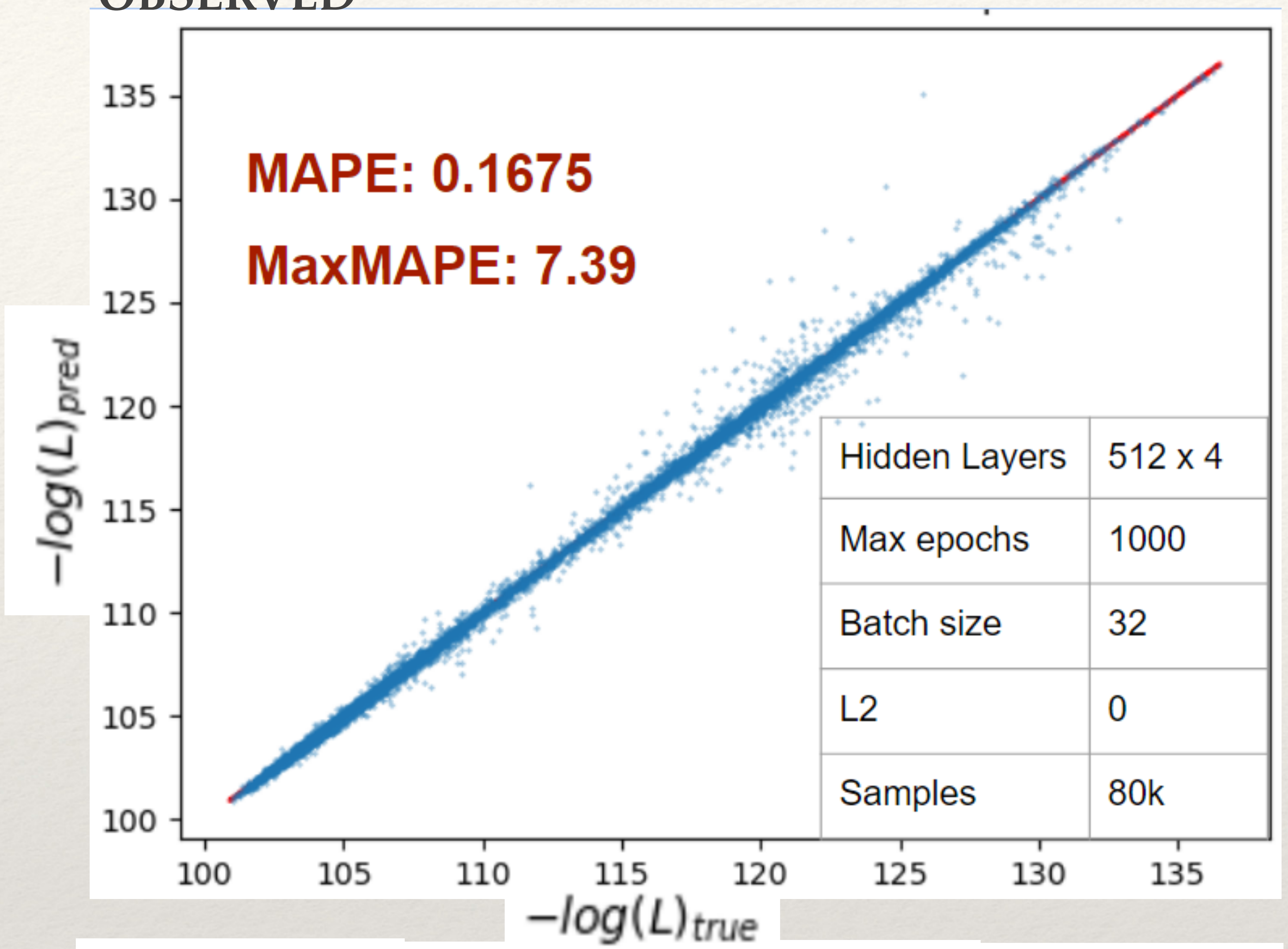


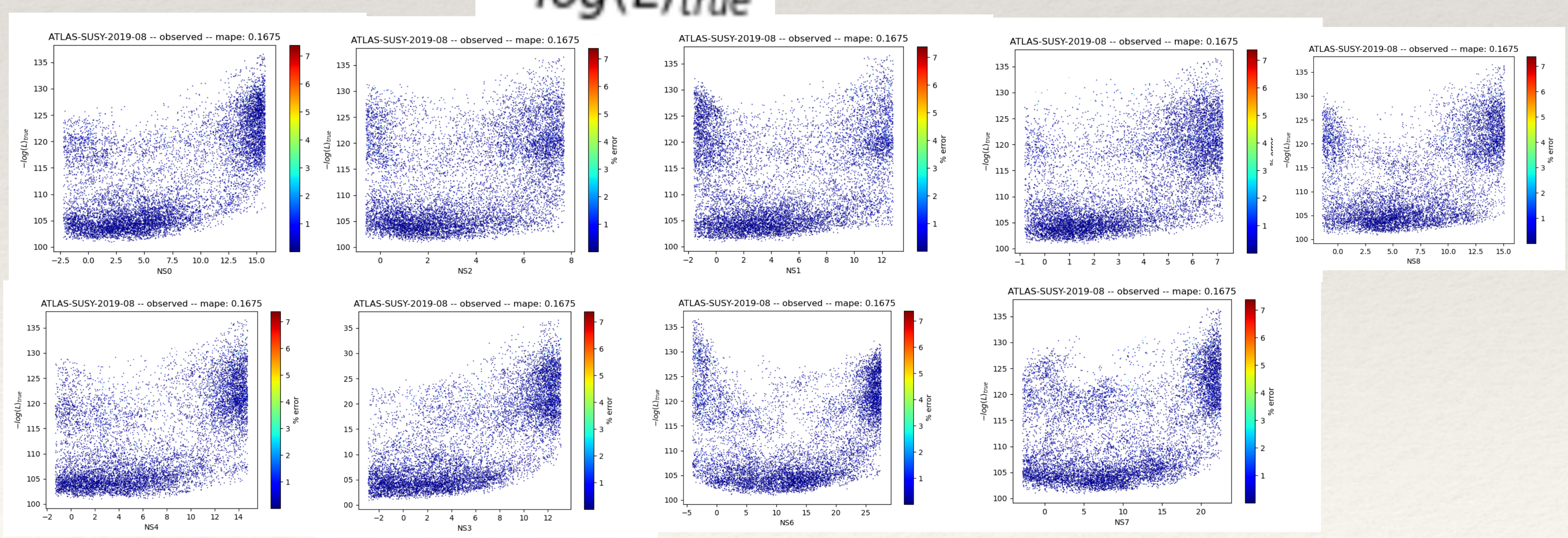




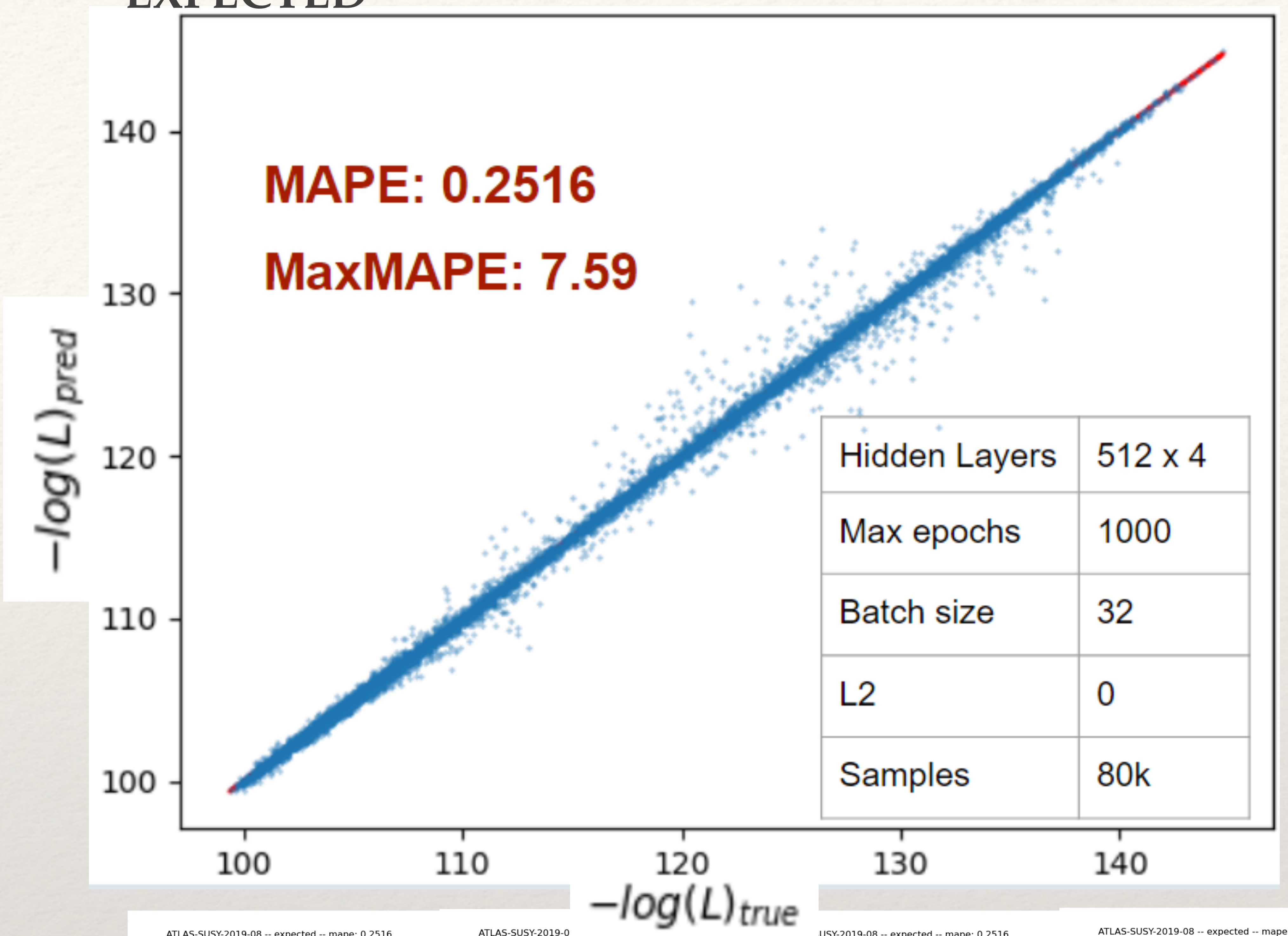
Results: ATLAS-SUSY-2019-08, 9 SRS

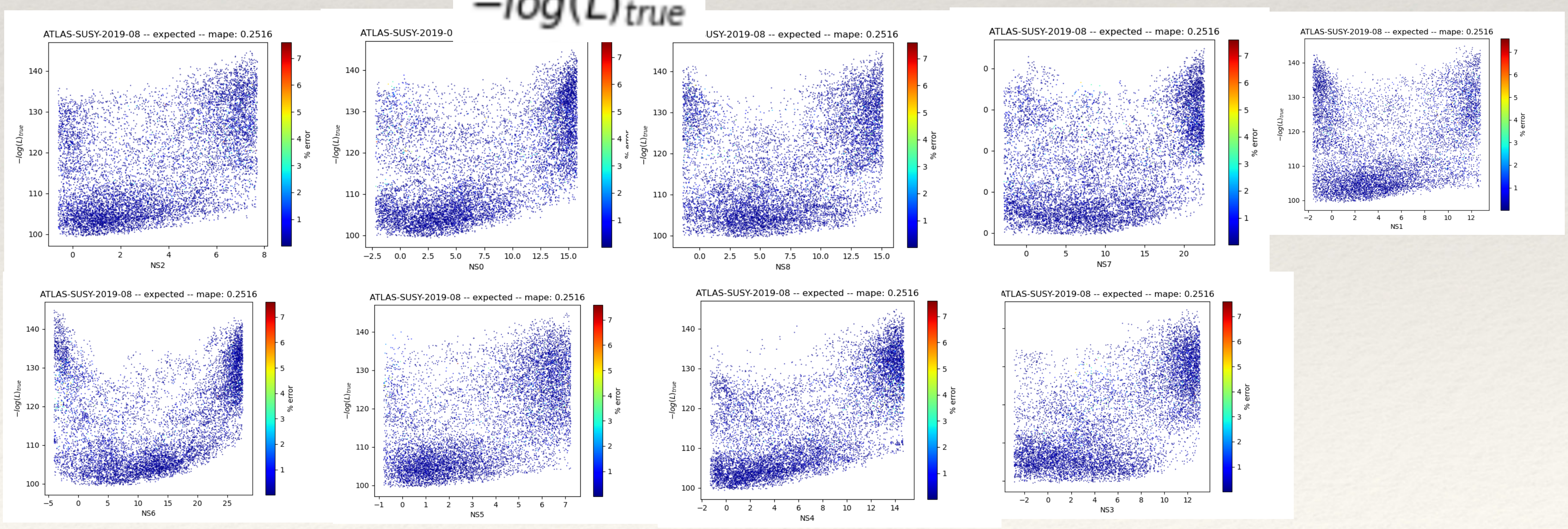
OBSERVED





EXPECTED





Deployment strategy.

Saving: After training, the best models for each analysis are ensemble together and saved as ONNX files and stored on Github or Zenodo.
-This ensures framework-independent usability. In the spirit of arXiv:2312.14575:

Les Houches guide to reusable ML models in LHC analyses

Jack Y. Araz[1], Andy Buckley[2], Gregor Kasieczka[3], Jan Kieseler[4], Sabine Kraml[5], Anders Kvellestad[6], Andre Lessa[7], Tomasz Procter[2], Are Raklev[6], Humberto Reyes-Gonzalez[8,9,10], Krzysztof Rolbiecki[11], Sezen Sekmen[12], Gokhan Unel[13]

Abstract
With the increasing usage of machine-learning in high-energy physics analyses, the publication of the trained models in a reusable form has become a crucial question for analysis preservation and reuse. The complexity of these models creates practical issues for both reporting them accurately and for ensuring the stability of their behaviours in different environments and over extended timescales. In this note we discuss the current state of affairs, highlighting specific practical issues and focusing on the most promising technical and strategic approaches to ensure trustworthy analysis-preservation. This material originated from discussions in the LHC Reinterpretation Forum and the 2023 PhysTeV workshop at Les Houches.

- **Usage:** The NN likelihoods will be available for statistical studies via an Spey (arXiv:2307.06996) backend.

SPEY: smooth inference for reinterpretation studies

Jack Y. Araz

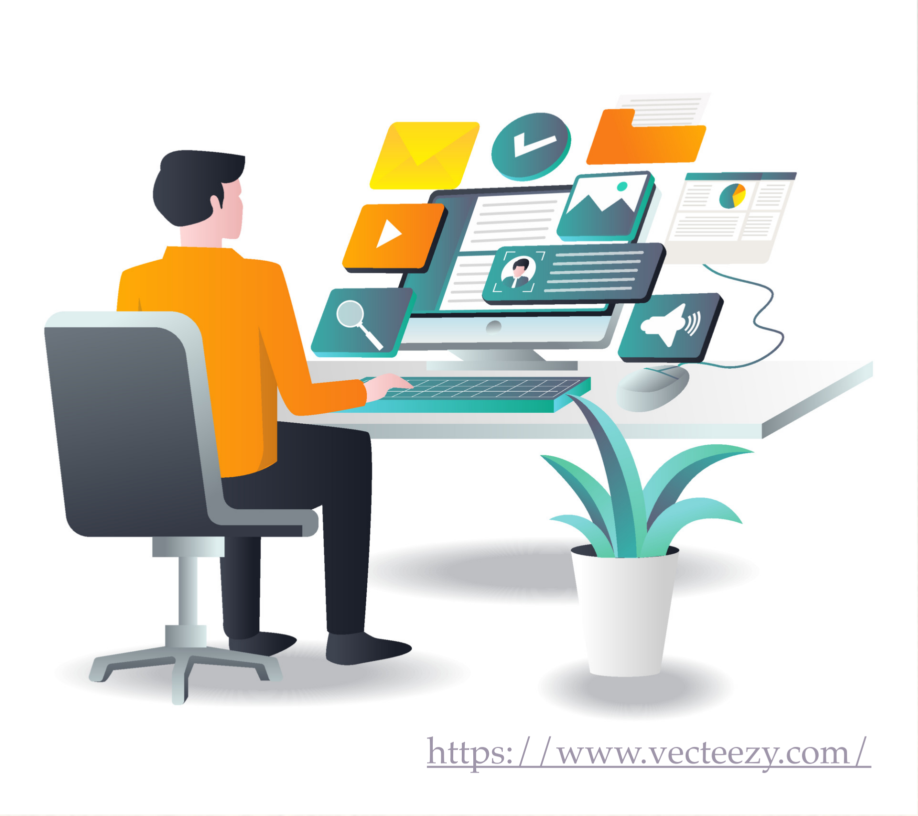
Abstract

Statistical models serve as the cornerstone for hypothesis testing in empirical studies. This paper introduces a new cross-platform Python-based package designed to utilize different likelihood prescriptions via a flexible plug-in system. This framework empowers users to propose, examine, and publish new likelihood prescriptions without developing software infrastructure, ultimately unifying and generalising different ways of constructing likelihoods and employing them for hypothesis testing within a unified platform. We propose a new simplified likelihood prescription, surpassing previous approximation accuracies by incorporating asymmetric uncertainties. Moreover, our package facilitates the integration of various likelihood combination routines, thereby broadening the scope of independent studies through a meta-analysis. By remaining agnostic to the source of the likelihood prescription and the signal hypothesis generator, our platform allows for the seamless implementation of packages with different likelihood prescriptions, fostering compatibility and interoperability.

```python
from spey import BackendBase, ExpectationType
from spey.base.model_config import ModelConfig

import onnx, onnxruntime
import numpy as np


class MLlikelihoods(BackendBase):
    """
    Spey plug-in to evaluate machine learned likelihoods

    Args:
        signal_yields (``List[float]``): Signal yields
        network_path (``str``): Path to the network onnx file.
    """

```

TO DOs

- Extend the number of NN-ized analyses: higher dimensional problems to be implemented!
- Possibly trying more sophisticated sampling strategies.
- Validation: Comparison of exclusion curves.
- NN gradient-based max(L) determination (differential programming).
- Write paper/documentation.



Conclusions

- Multivariate phenomenological studies require efficient handling of likelihoods.
- NNs provide an orders of magnitude faster complement for LHC likelihood publication. **From dozens of minutes to less than a second per point!**
- Profile likelihoods are easily learnable by NNs.
- They can easily be integrated into modern reinterpretation frameworks.
- This effort is part of a growing ecosystem of statistical tools for maximally profiting LHC results!
- Complete infrastructure for NN Likelihoods under construction. **Stay tuned!**

THANK YOU!