



Université
Paris Cité



Proof of Concept:
Using normalizing flows to predict A_t from the
Higgs boson mass for SuSpect program
computation

BOUY-GRUAZ Timothé & LIU Zirui & ROUSSEAU David
with KNEUR Jean-Loïc & MOULTAKA Gilbert & ZERWAS Dirk

SuSpect3



SuSpect3: program used to compute, from SUSY parameters, SUSY and Higgs particle spectrum in unconstrained MSSM, and in constrained models (cMSSMs) such as minimal Supergravity (mSUGRA), gauge mediated (GMSB) and anomaly mediated (AMSB) Supersymmetry breaking scenarii.

Jean-Loïc Kneur^a, Gilbert Moultaka^a, Michaël Ughetto^b, Dirk Zerwas^{c,d,*},
Abdelhak Djouadi^{e,f}

^a*Laboratoire Charles Coulomb (L2C), CNRS, Université de Montpellier, Montpellier; France.*

^b*CPPM, Aix-Marseille Université, CNRS/IN2P3, Marseille; France.*

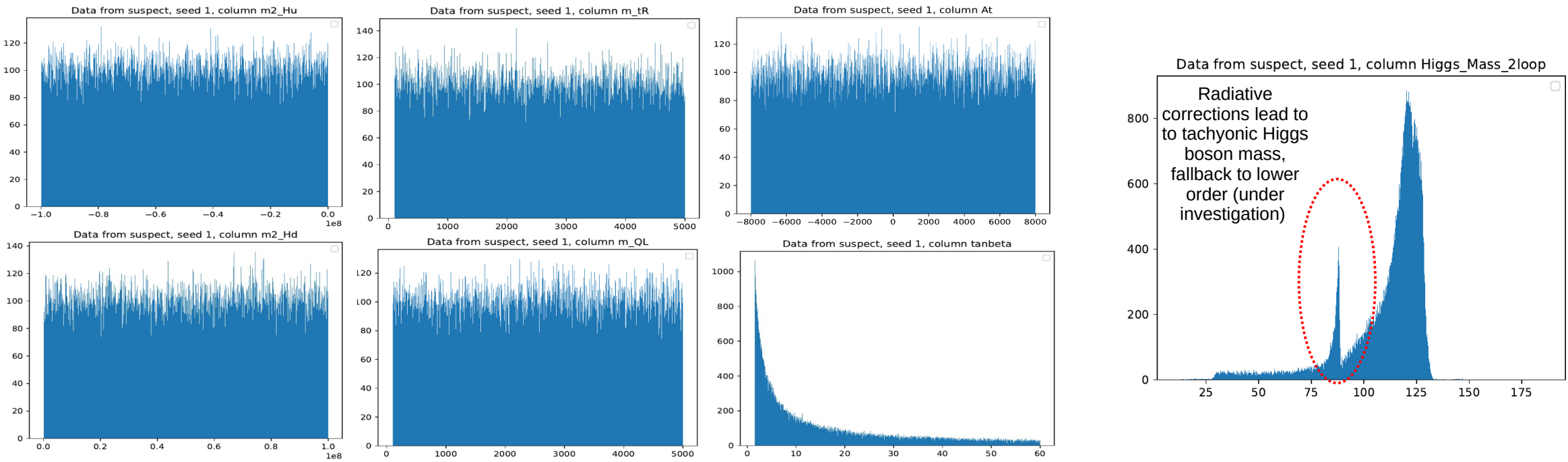
^c*IJCLab, Université Paris-Saclay, CNRS/IN2P3, Orsay; France.*

^d*also at DMLab, CNRS/IN2P3, Hamburg; Germany.*

^e*CAFPE, Granada and Granada U., Theor. Phys. Astrophys., Granada; Spain.*

^f*NICPB, Rävala pst. 10, 10143 Tallinn, Estonia.*

MSSM: Standard procedure

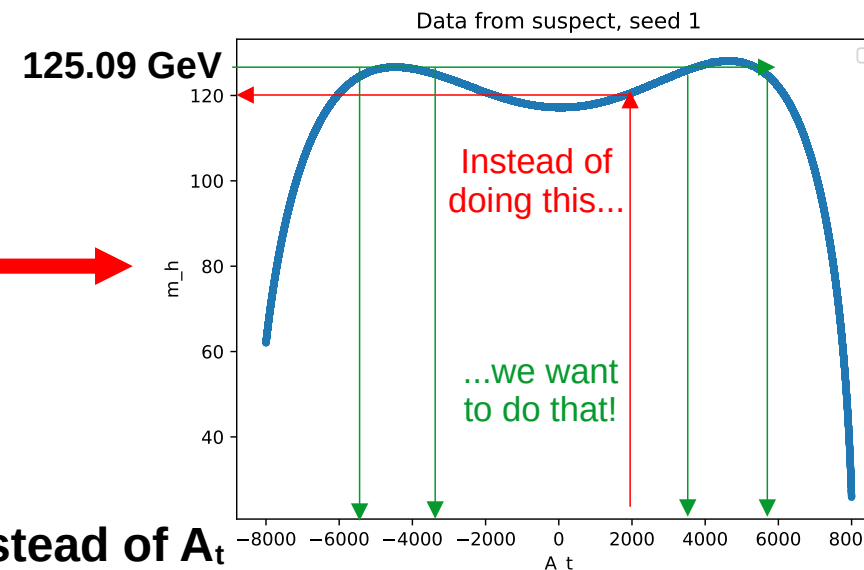


Use Higgs mass as a fundamental parameter

SuSpect: computes SUSY and Higgs particle spectrum

BUT

- Higgs mass constrained by measurements
- Higgs mass strongly depends on parameter A_t



Use Higgs mass as a parameter instead of A_t

Problems to solve:

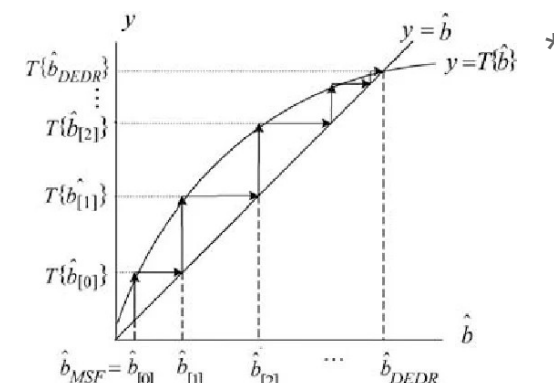
- FP algorithm needs good precision; currently uses polynomial degree 4 to determine seed with 20% precision
- Increasing the seed precision should improve the convergence properties of the FP algorithm; currently often fails!!

Our job!!

Step 1:
Find all possible A_t solutions

Step 2:
Find true value using fixed points algorithm

SuSpect job

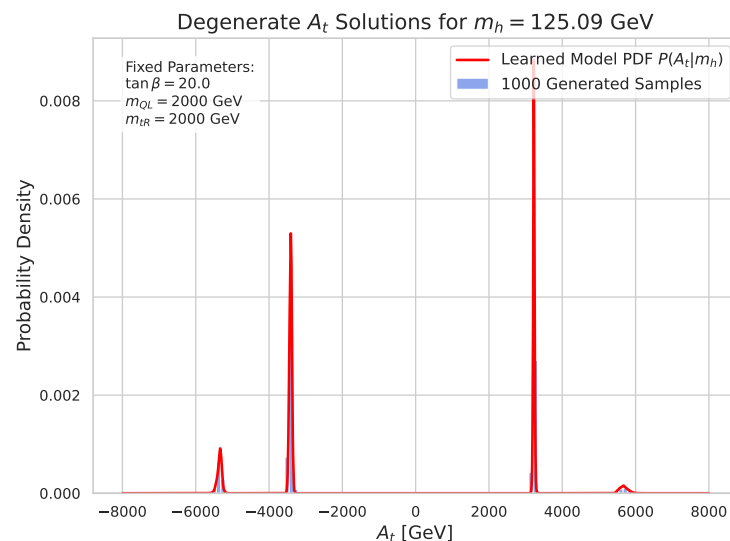


Need of a new way to find the seeds, with greater precision!

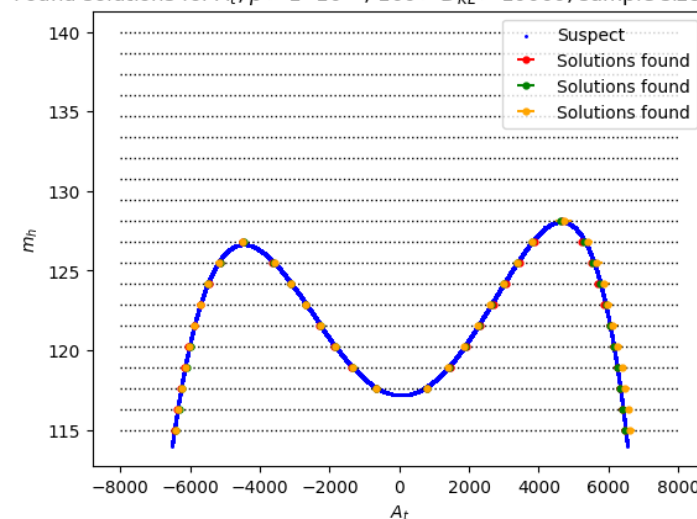
Find A_t seeds: Procedure

```

1 tanbeta,At,m_QL,m_tR,m2_Hu,m2_Hd,Higgs_Mass_2loop
2 46.7847,7954.96,727.81,4995.3,-6.03419e+07,2.36165e+07,122.564
3 17.7445,-1793.43,4684.14,4246.92,-4.75452e+07,3.13342e+07,122.438
4 3.49855,-904.754,2718.63,4578.41,-5.69301e+07,4.57259e+07,113.799
5 26.4922,7026.04,3608.26,4033.51,-4.81847e+07,9.28915e+06,129.027
6 31.9473,5840.32,4165.06,1437.94,-3.29472e+07,5.93373e+06,129.235
7 17.8699,1489.05,2117.76,1068,-8.5788e+07,2.89701e+07,112.313
8 6.87074,4533.03,267.439,3157.75,-7.01505e+07,6.6067e+07,126.009
9 3.40368,-861.848,459.484,2399.27,-9.66298e+06,9.62626e+06,108.487
10 10.3956,-6088.15,509.753,4592.62,-7.0107e+07,9.10457e+07,117.677
    
```



Found solutions for A_t , $p > 1 \cdot 10^{-5}$, $100 > D_{KL} > 10000$, sample size 100000



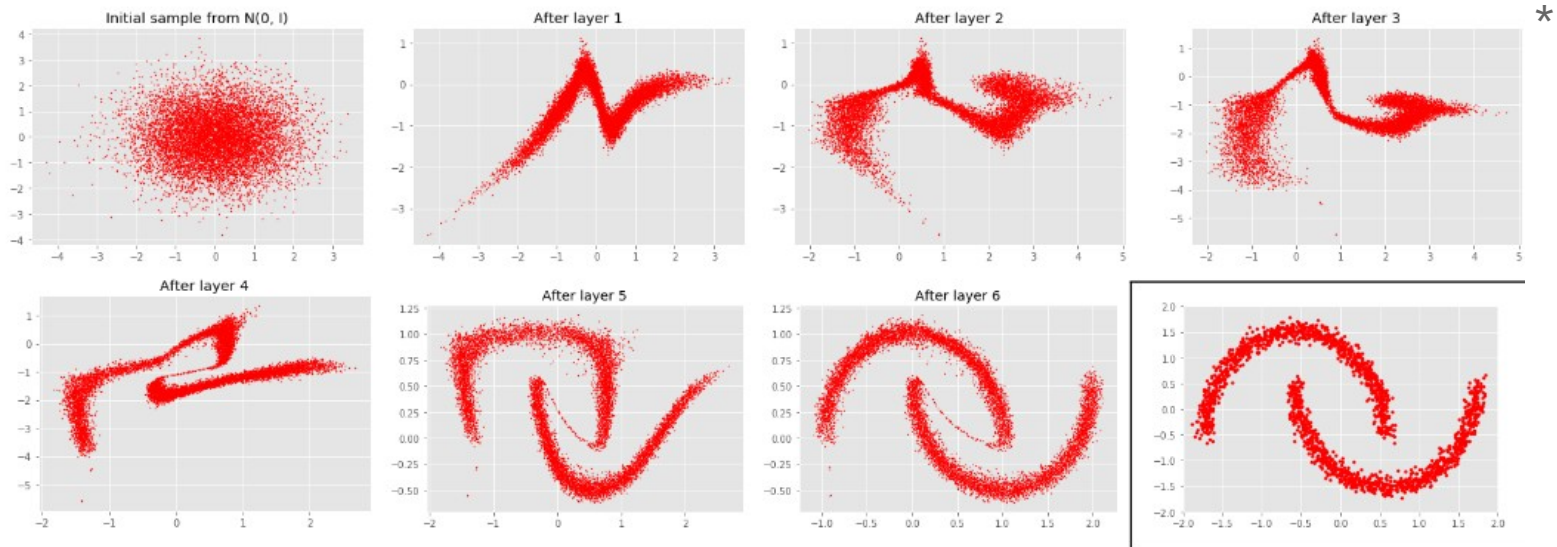
Generate samples
using SuSpect

Train **Normalizing
Flows** to estimate
 A_t

Discriminate
possible solutions
using **Kullback-
Leibler divergence**

Normalizing Flow?

We have a d -dimensional target space X and its unknown distribution p_X , and a d -dimensional latent space Z with an “easy” known distribution p_Z (often either gaussian or uniform). For x sampled from p_X (unknown), a normalizing flow models a bijective function f going from X to Z : $z = f(x)$.



Once the NF is trained, we can use 2 modes:

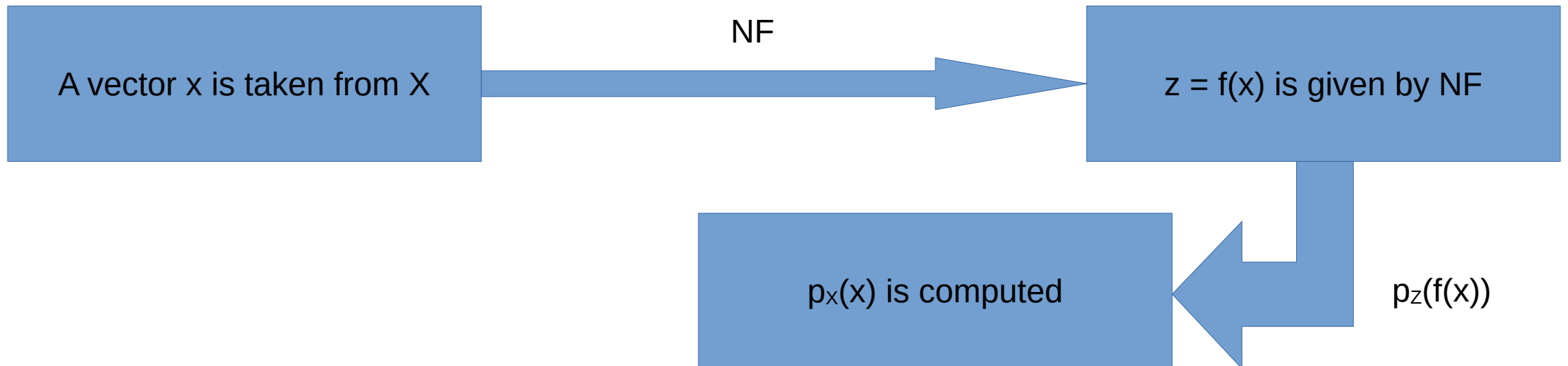
- the forward mode, that lets us evaluate the target distribution p_X
- the inverse mode, that lets us sample x using a sample z from p_Z

Forward mode

If we have 2 distributions p_X and p_Z , from spaces X and Z , and if we have a transformation f sending from X to Z (normalizing flow), then we can compute p_X :

$$p_X(x) = p_Z(f(x)) \left| \det \left(\frac{\partial f(x)}{\partial x^T} \right) \right|^*$$

Once NF has given us $f(x)$, it's easy to compute $p_X(x)$.



Example

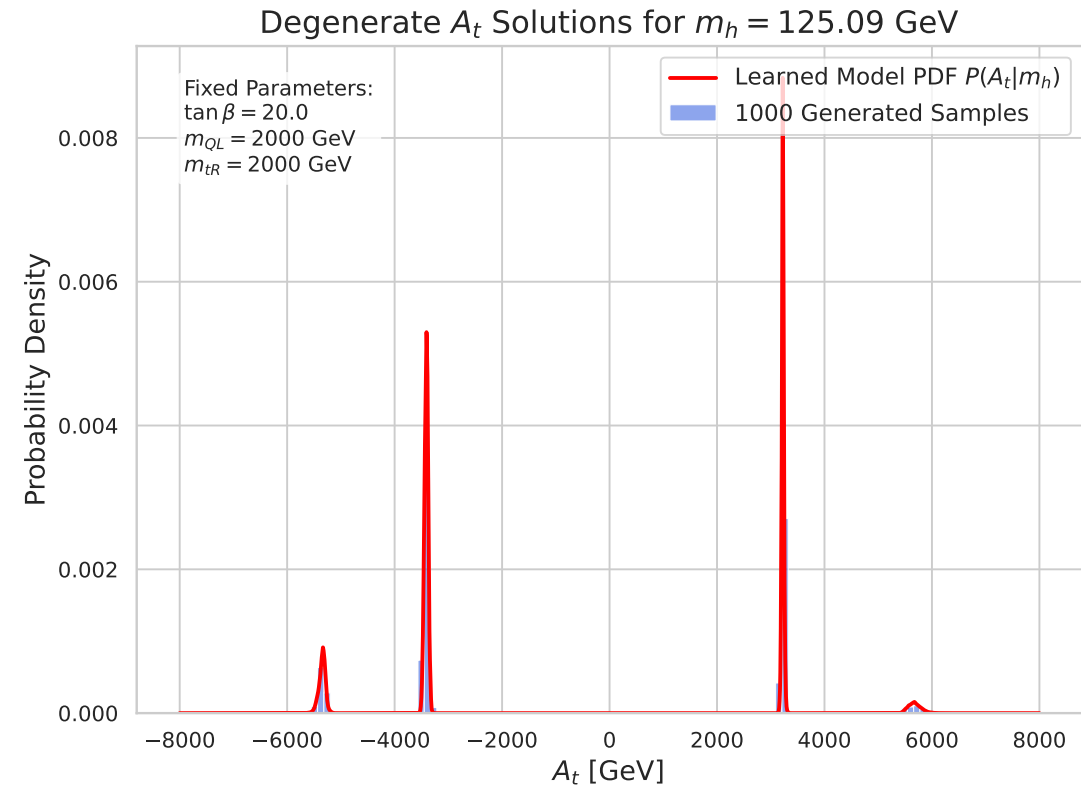
Example of usage for our purpose: We wish to predict the probability distribution of A_t in target space given a certain Higgs mass, and all other parameters are fixed.

Input data x:

$\tan(\beta) = 20$
 $A_t = \text{array}(-8000, 8000)$
 $M_{QL} = 2000 \text{ GeV}$
 $M_{tR} = 2000 \text{ GeV}$
 $M_{Hu}^2 = -2.13e+05 \text{ GeV}^2$
 $M_{Hd}^2 = 3.66e+06 \text{ GeV}^2$

 $M_h = 125.09 \text{ GeV}$

NF: forward mode



Presentation of the NF ensemble

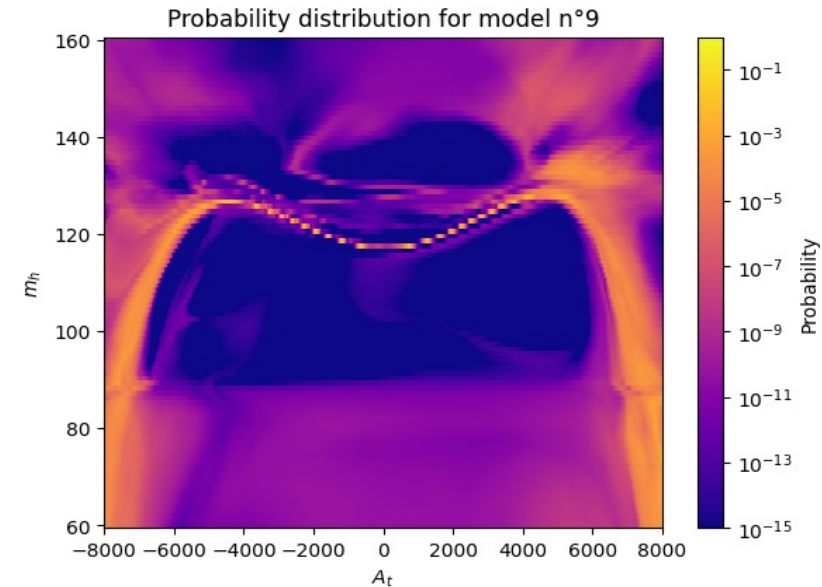
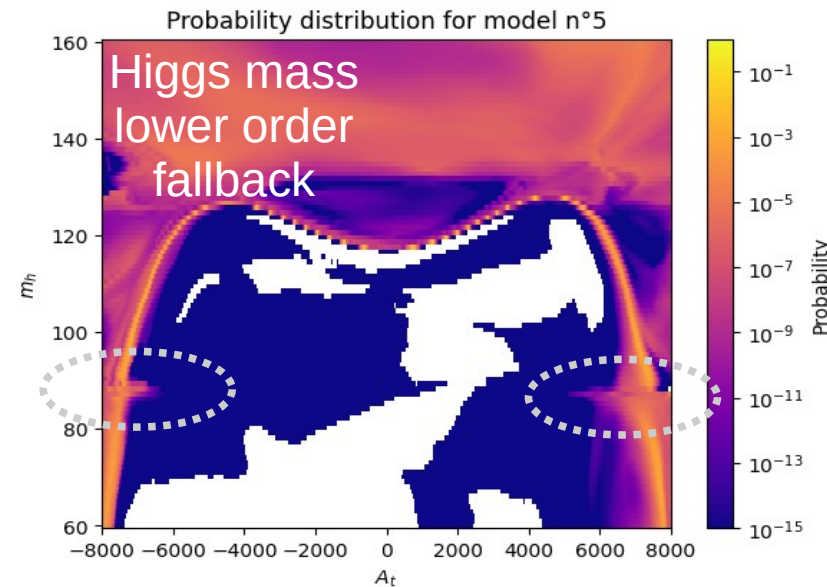
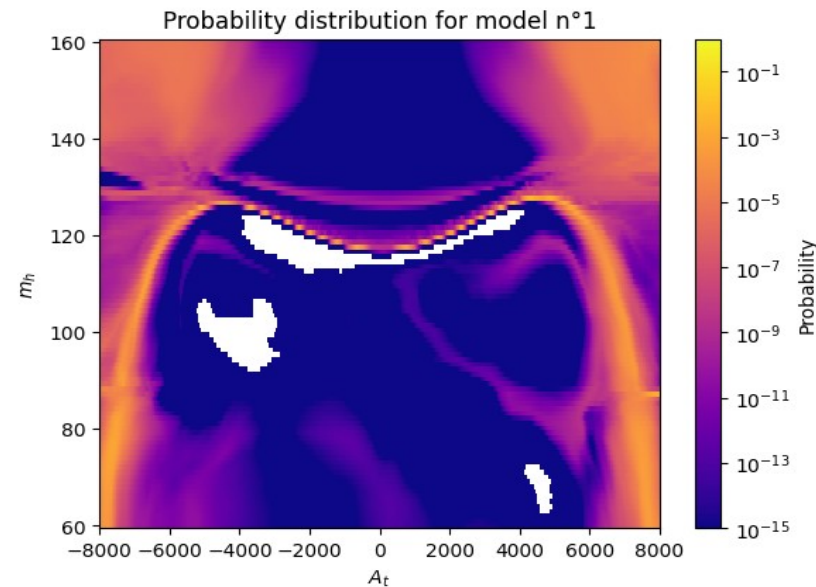
False and true solutions are very similar, and thus often difficult to discriminate.

Having an ensemble can help!

Principle:

Models only different by seed (= trained the exact same way with exactly same data) tend to differ in least constrained zones

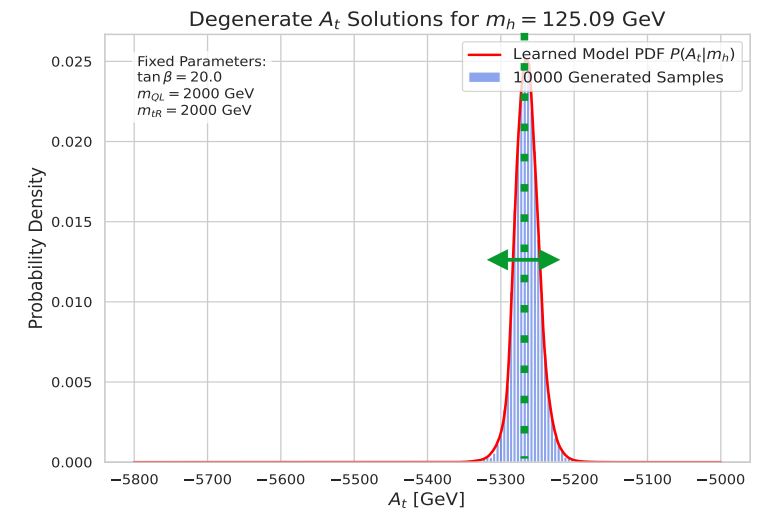
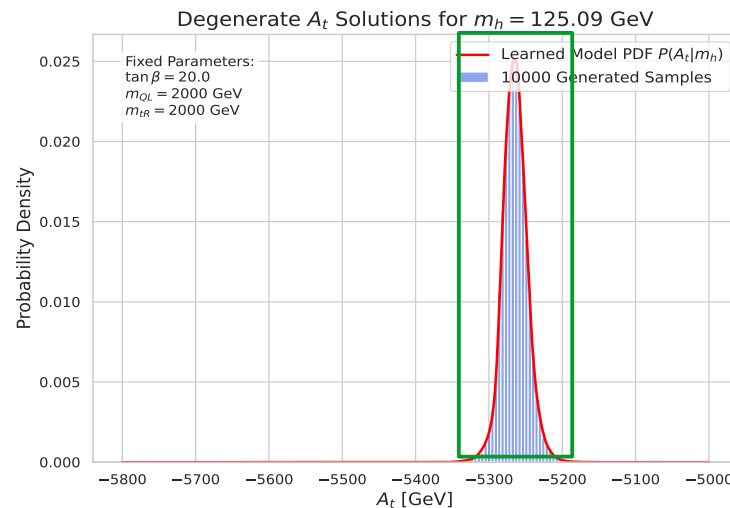
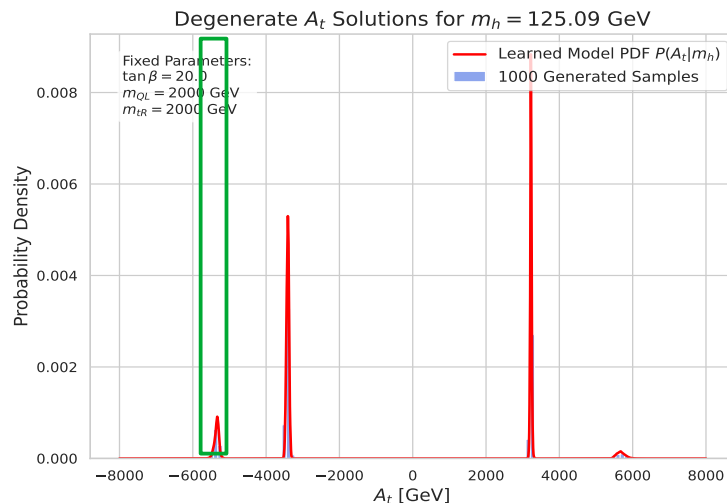
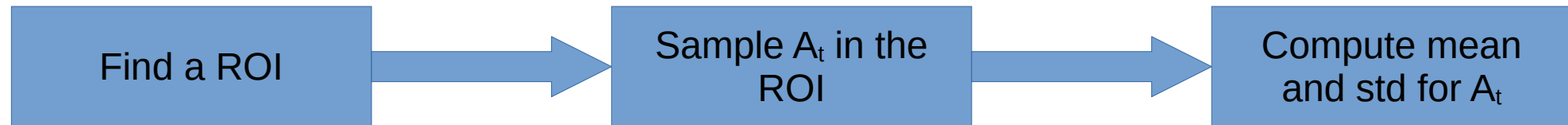
Here we used 3 models: 2 is not robust enough, while 4 is already too much for only a proof a concept



Solution Searching Algorithm

A peak-searching algorithm won't work because of the **90 GeV spike** resulting in several spikes around the solution at those energies.

Another approach is to find the **regions of interest** (ROI) in the probability distribution, find the mean and standard deviation for A_t in each region, and plot the results.

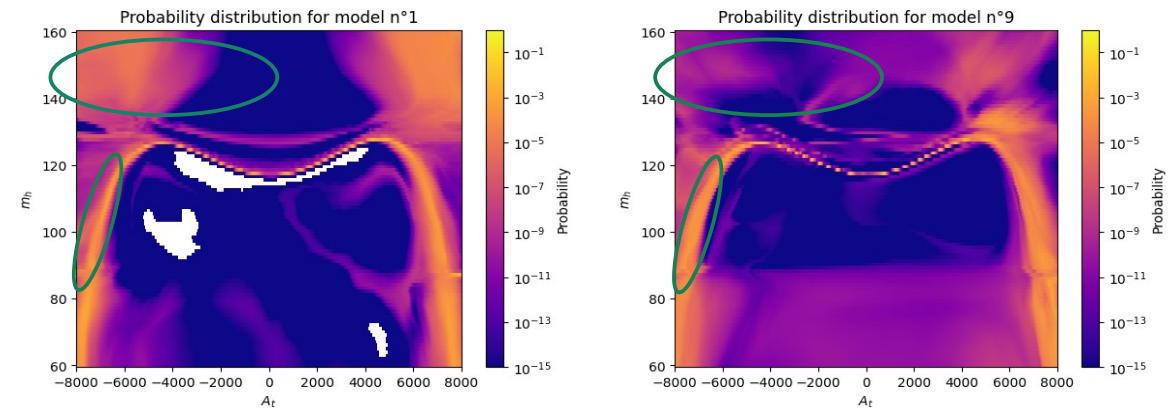


Kullback-Leibler Divergence

To discriminate true solutions and false solutions, we'll do some ensembling, meaning we compare the three models to see if they invent data or if the solution is predicted correctly.

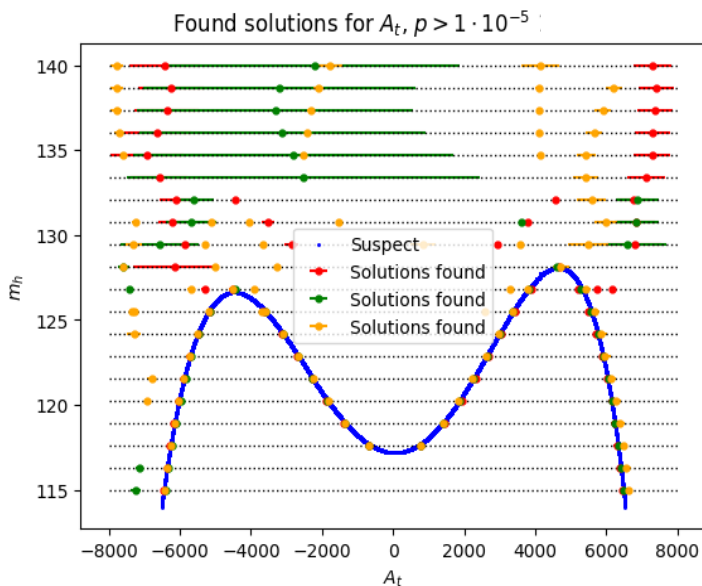
To do that, we use the **Kullback-Leibler divergence**, which is a formula to compute "distance" between two probability densities:

$$D_{\text{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)} \quad *$$



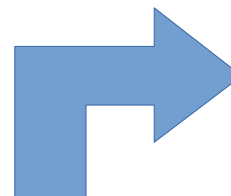
Ensembling application

Solutions found without discrimination

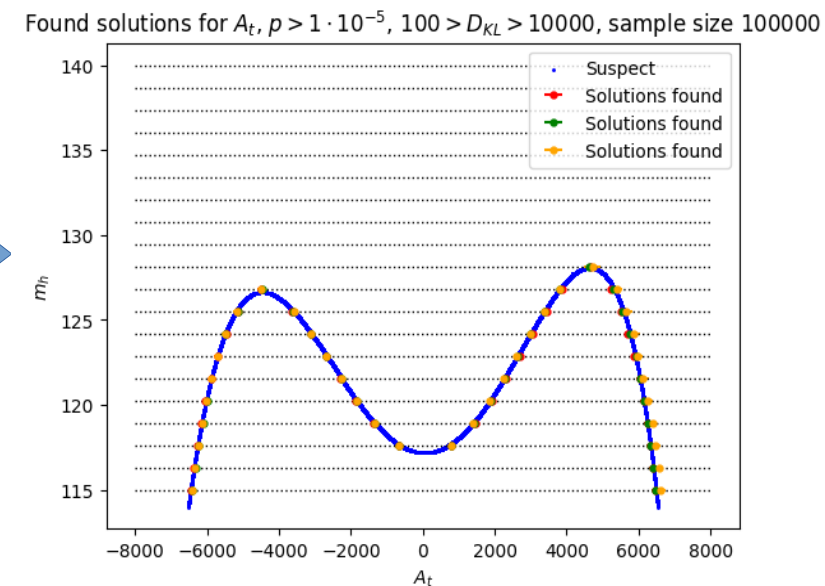


If a solution exists,
is it real?

$$100 < D_{KL} < 10000$$

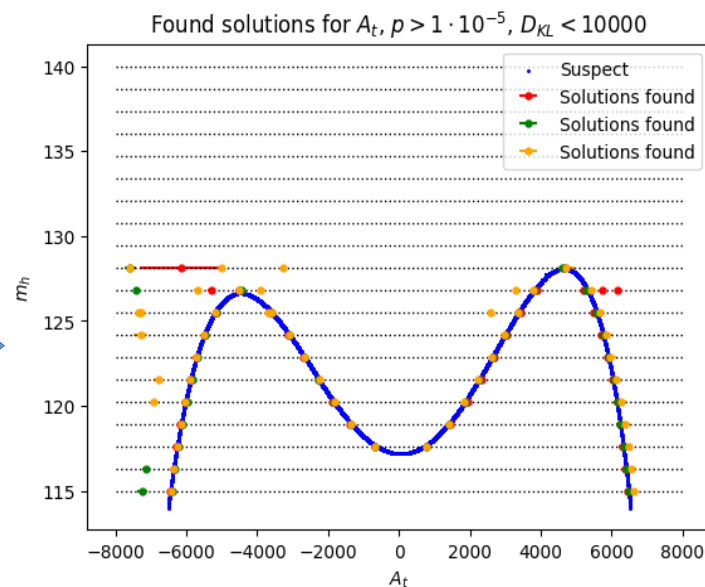
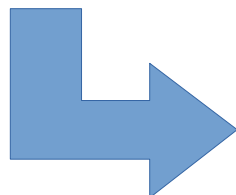


Solutions found after discrimination



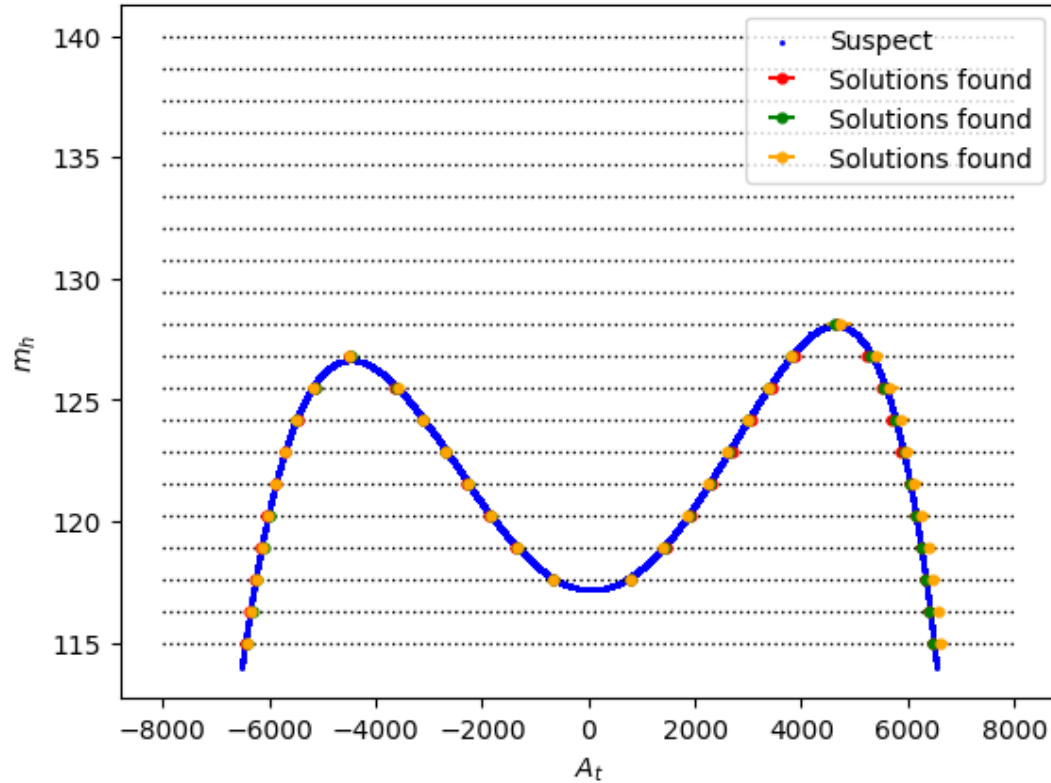
Is there a solution?

$$D_{KL} < 10000$$

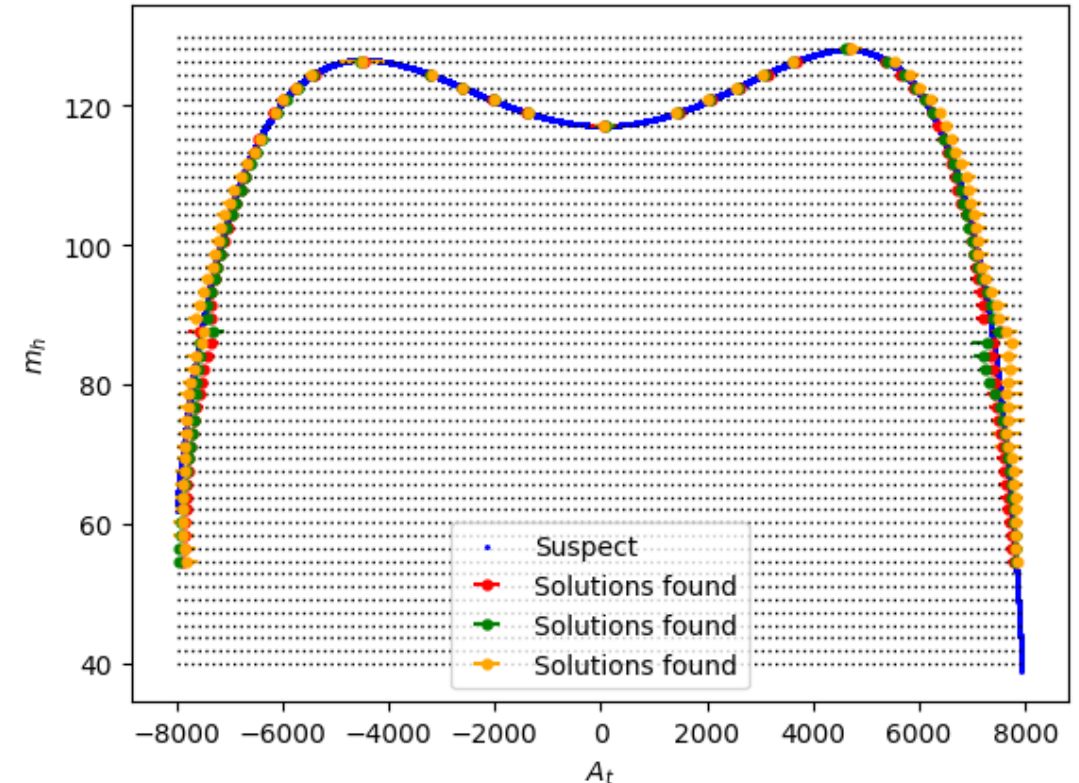


Results

Found solutions for $A_t, p > 1 \cdot 10^{-5}, 100 < D_{KL} < 10000$, sample size 100000



Found solutions for $A_t, p > 1 \cdot 10^{-7}, 100 < D_{KL} < 10000$, sample size 100000



Still room for improvement!

What now?

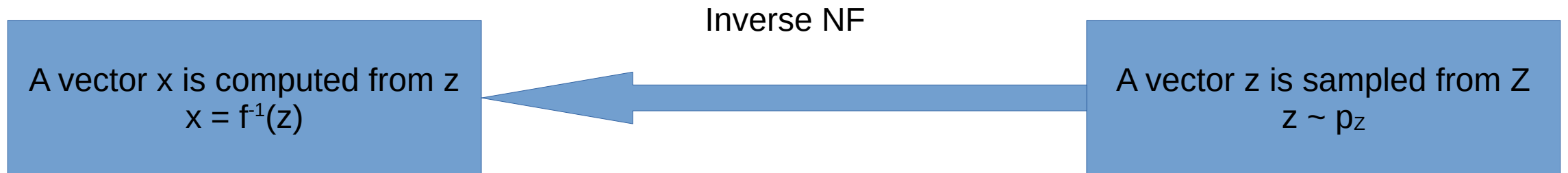
Solution Searching Algorithm seems to work fine, but needs further testing:

- Compatibility C++ Python
- Parameter space full exploration
- Integration to SuSpect for first real tests: Is it already precise enough for fixed point algorithm?

Thank you for your
attention!

Inverse mode

The main property of f is that it's invertible, so we can pass through the NF backwards to compute an effective sample from X using z sampled from p_z .



Normalizing Flow: Distributions generated

