DL-based PMT photon counting and a normalizing-flow package for (astro-)particle physics

Bundesministerium für Bildung und Forschung

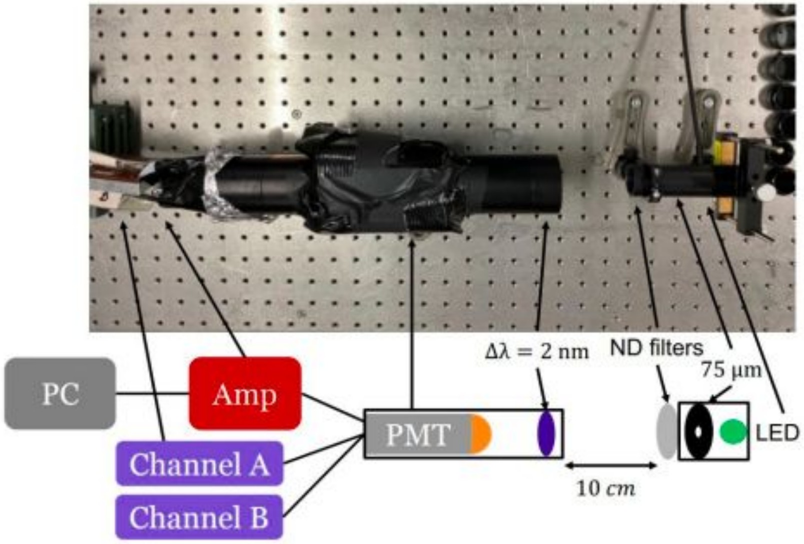
ERLANGEN CENTRE FOR ASTROPARTICLE PHYSICS

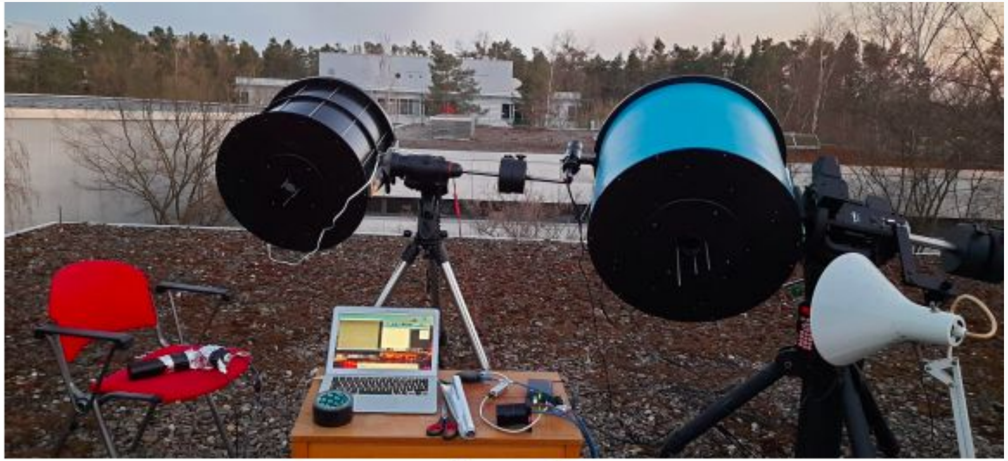
Thorsten Glüsenkamp, Feb 14th, IDT collaboration meeting

Two topics:

1) PMT rate reconstruction for high-intensity interferometry (Jigar Banderi, Dmitry Malyshev)

2) jammy_flows - a normalizing flow package tailored for (astro-) particle physics (Thorsten Glüsenkamp)





Calibration of Photomultiplier Tubes for Intensity Interferometry at H.E.S.S.

$$\frac{S}{N} = C \cdot \epsilon \cdot n(\nu) \cdot \sqrt{\frac{T}{\tau_e}}$$

Source flux

Observation time

Telescope collection area

Photon detection efficiency of the system

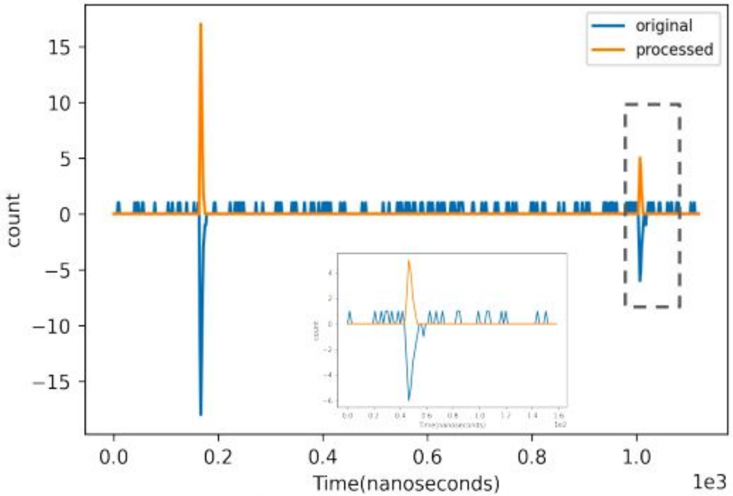
Optical bandwidth

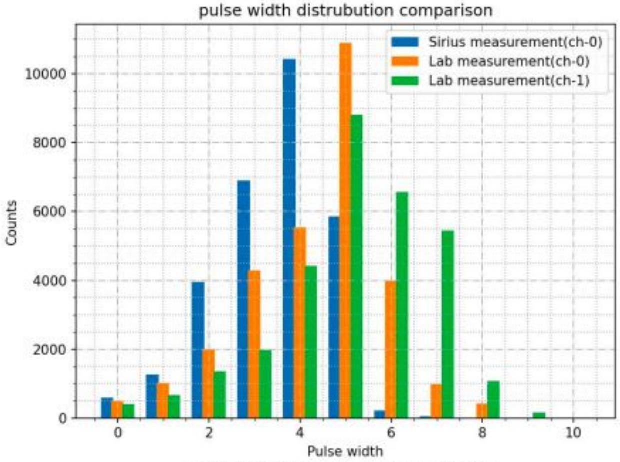
System resolution

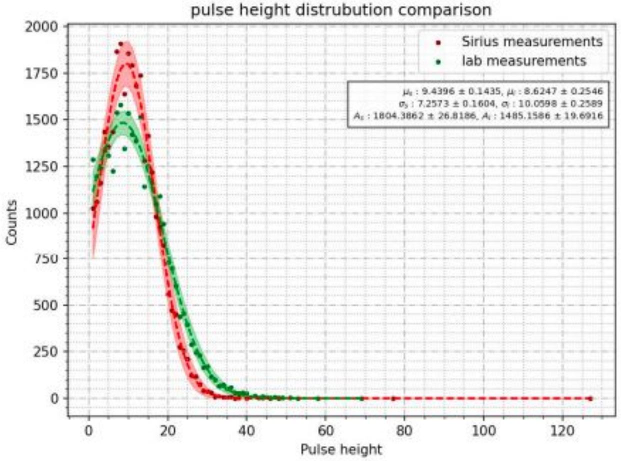
$$\boxed{N = \frac{1}{\sqrt{R_1 R_2}} \sqrt{\frac{\tau_e}{T}}}$$

Photon rates

Data preprocessing

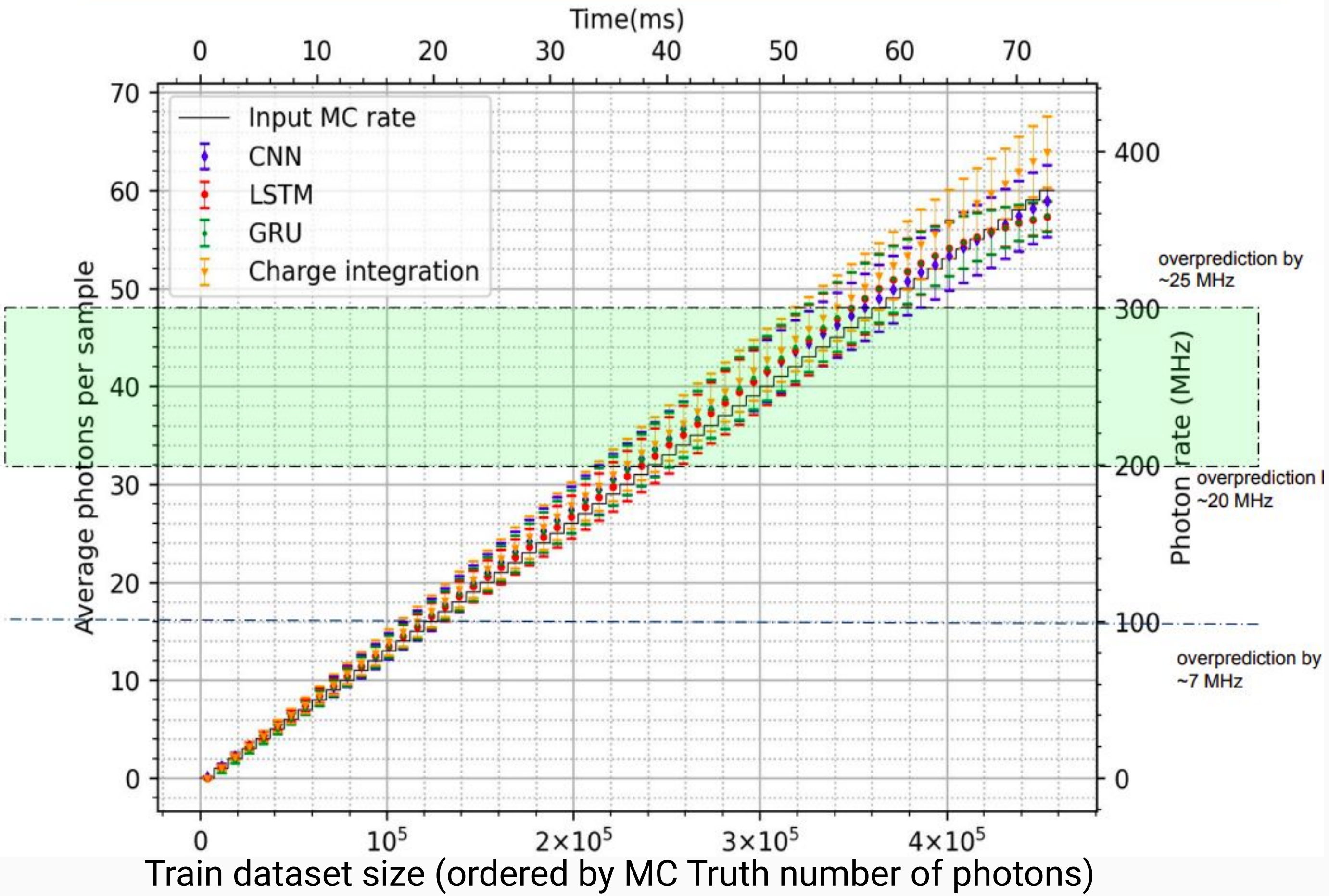






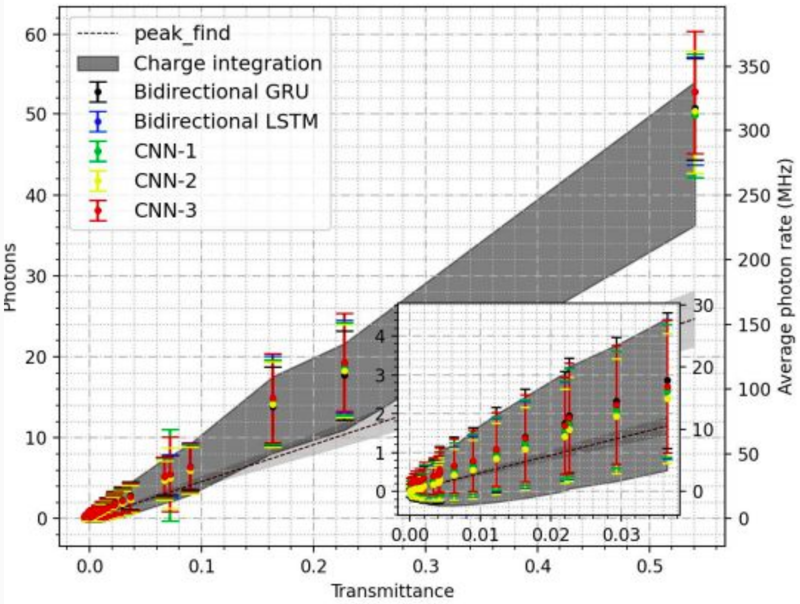
- Extraction of photon shapes after preprocessing with sample size 100
- Shape set splitting for training testing and validation by 20%
- MC-data creation for selected sample size
- Input: 1D-waveform, output: number of photon peaks in sample
- Train selected neural network
- Evaluate network(test dataset)
- Testing of network
  - Lab : waveform with different transmission
  - Sirius : measurement at different instances

Model Evaluation

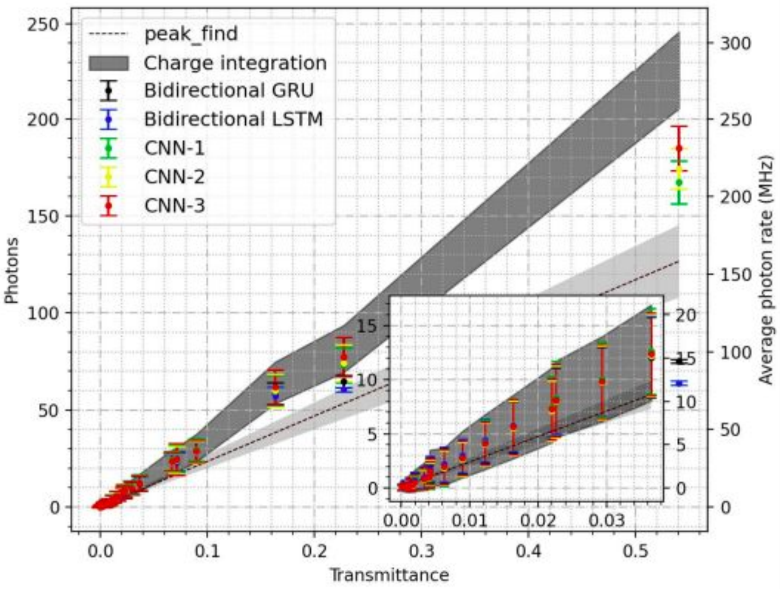


Prediction comparison (Lab)

Sample size 100

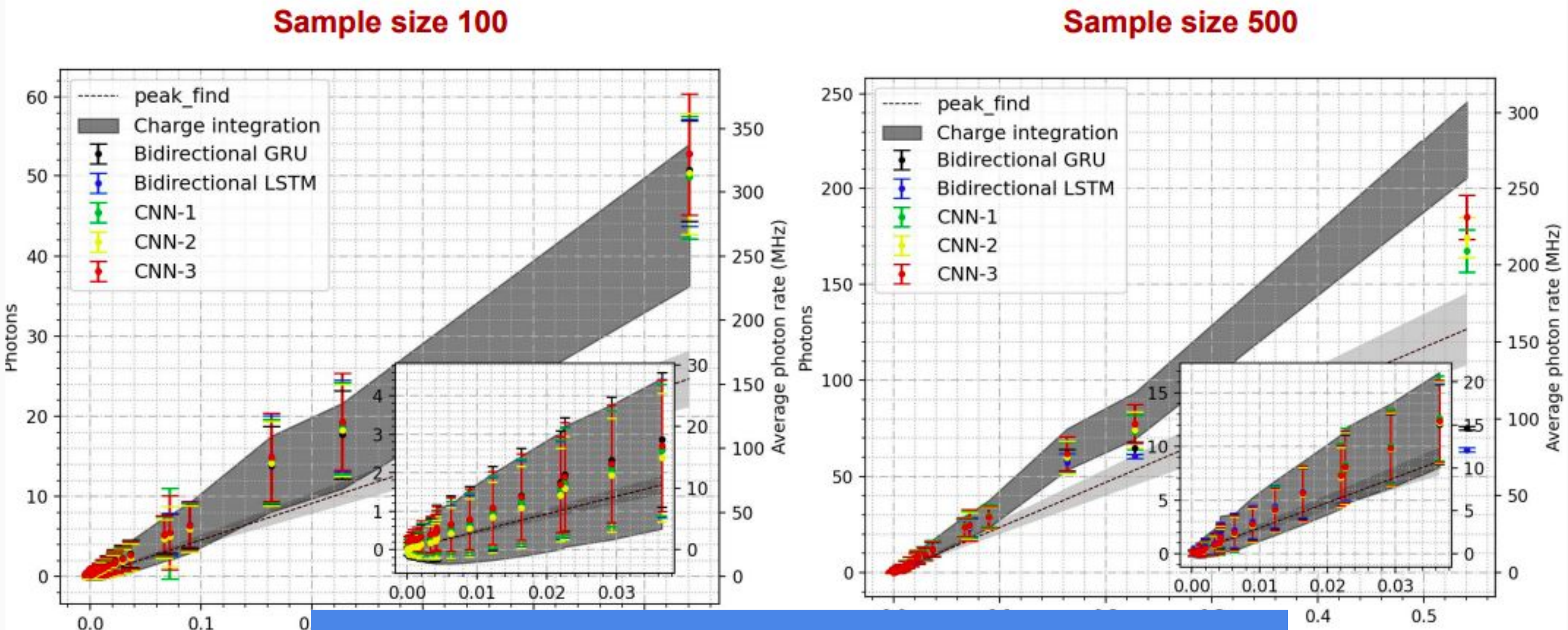


Sample size 500



Different points correspond to measurements with different grey filters, and corresponding transmittances are plotted on the x-axis

Prediction comparison (Lab)



Conclusion: Calibration of PMT rate possible with neural networks

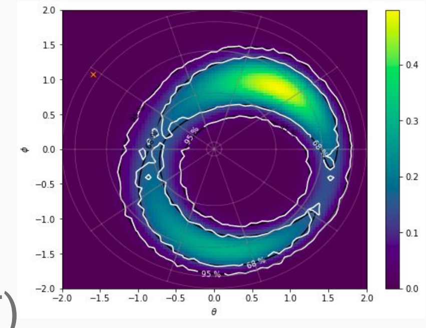
Different points correspond to measurements with different grey filters, and corresponding transmittances are plotted on the x-axis

jammy_flows - a normalizing flow package tailored for (astro-) particle physics

Motivation:

Particle physics needs precise normalizing flows together with coverage guarantees

We need PDFs of directions (defined on the sphere -> manifold PDF)

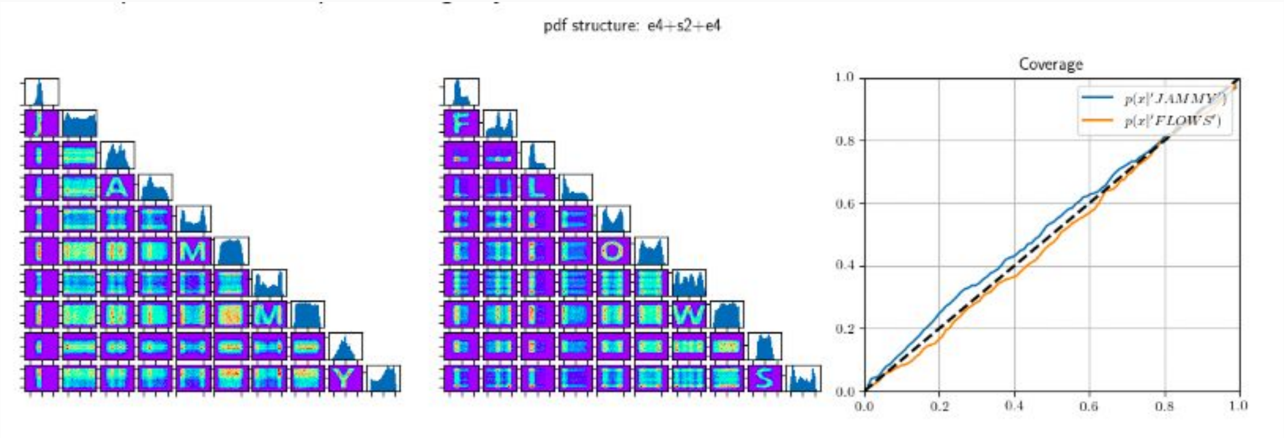


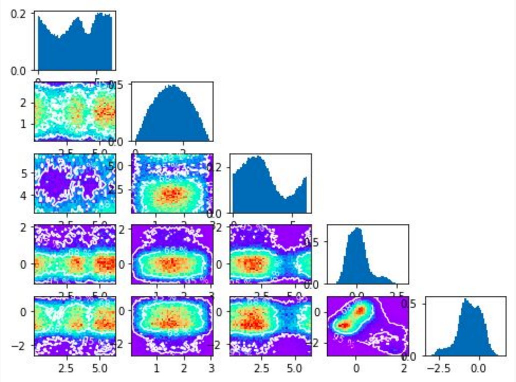
We would like to have joint PDFs (e.g. PDF defined jointly on Euclidean space and the sphere, i.e. a PDF on   with arbitrary correlation structure)

jammy_flows - a normalizing flow package tailored for (astro-) particle physics

A partial feature list:

1. Coverage for arbitrary manifold tensor product distributions automatically supported as defined in (https://arxiv.org/abs/2008.05825)





jammy_flows - a normalizing flow package tailored for (astro-) particle physics

A partial feature list:

1. Coverage for arbitrary manifold tensor product distributions automatically supported as defined in (https://arxiv.org/abs/2008.05825)
2. Supports best-in-class flows for Euclidean, spherical, simplex, line manifolds + automatic forward/backward pass cross-checks

jammy_flows - a normalizing flow package tailored for (astro-) particle physics

A partial feature list:

1. Coverage for arbitrary manifold tensor product distributions automatically supported as defined in (https://arxiv.org/abs/2008.05825)
2. Supports best-in-class flows for Euclidean, spherical, simplex, line manifolds + automatic forward/backward pass cross-checks
3. Easy to get going with 1 line of code

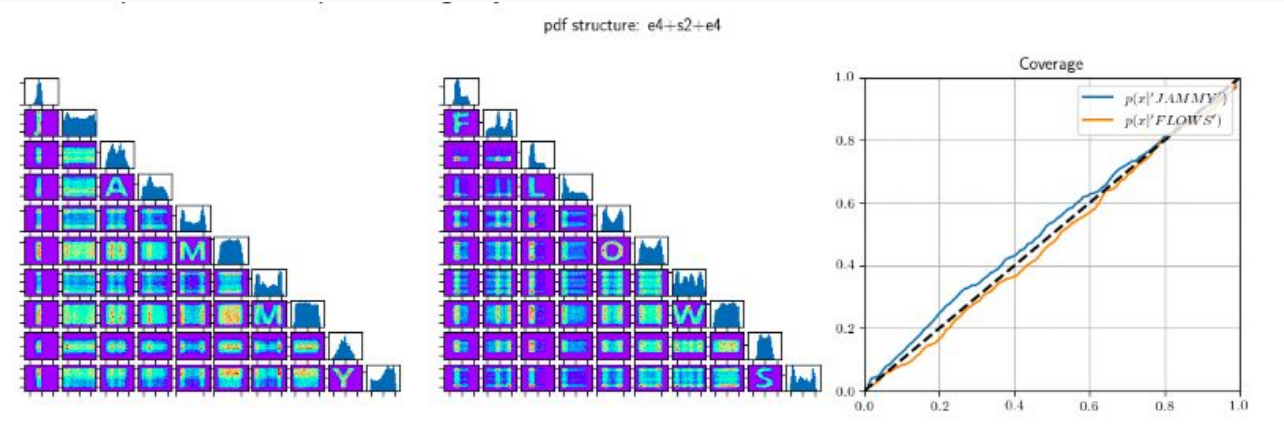
jammy_flows - a normalizing flow package tailored for (astro-) particle physics

Any flow is injective -> can composite flows: f_tot = f_1(f_2(...f_n(x))) for more complexity

```
import jammy_flows

pdf=jammy_flows.pdf("e4+s2+e4", "gggg+n+gggg")
```

“gggg” -> f_1(f_2(f_3(f_4(x))))



Simple to use:

1 line defines a (conditional) PDF that can be trained

-> gets you fast to 80%/90%

Still possible: customization for flexibility

jammy_flows - a normalizing flow package tailored for (astro-) particle physics

```
import jammy_flows

pdf=jammy_flows.pdf("e4+s2+e4", "gggg+n+gggg")
```

Any flow is injective -> can composite flows: f_tot = f_1(f_2(...f_n(x)) for more complexity

"gggg" -> f_1(f_2(f_3(f_4(x))))

- Precise normalizing flows with coverage a common goal
- Lets make it a community effort
- https://github.com/thoglu/jammy_flows (currently still beta, first release in next 1-2 months)

se:

es a

l) PDF that can

-> gets you fast to 80%/90%

Still possible:
customization for flexibility