NeuLat

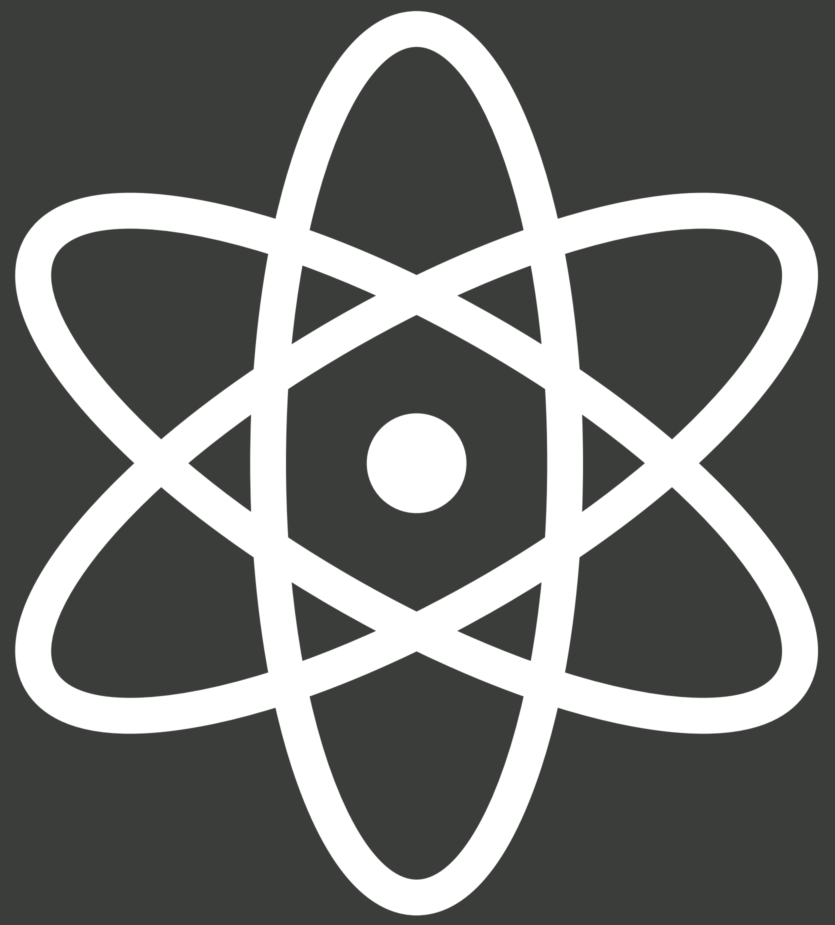
A toolbox for Neural samplers in Lattice field theories

Kim A. Nicoli

TRA Matter, HISKP, Bethe Center, LAMARR

Talk based on: https://pos.sissa.it/453/286/pdf

Normalizing Flows for LFT

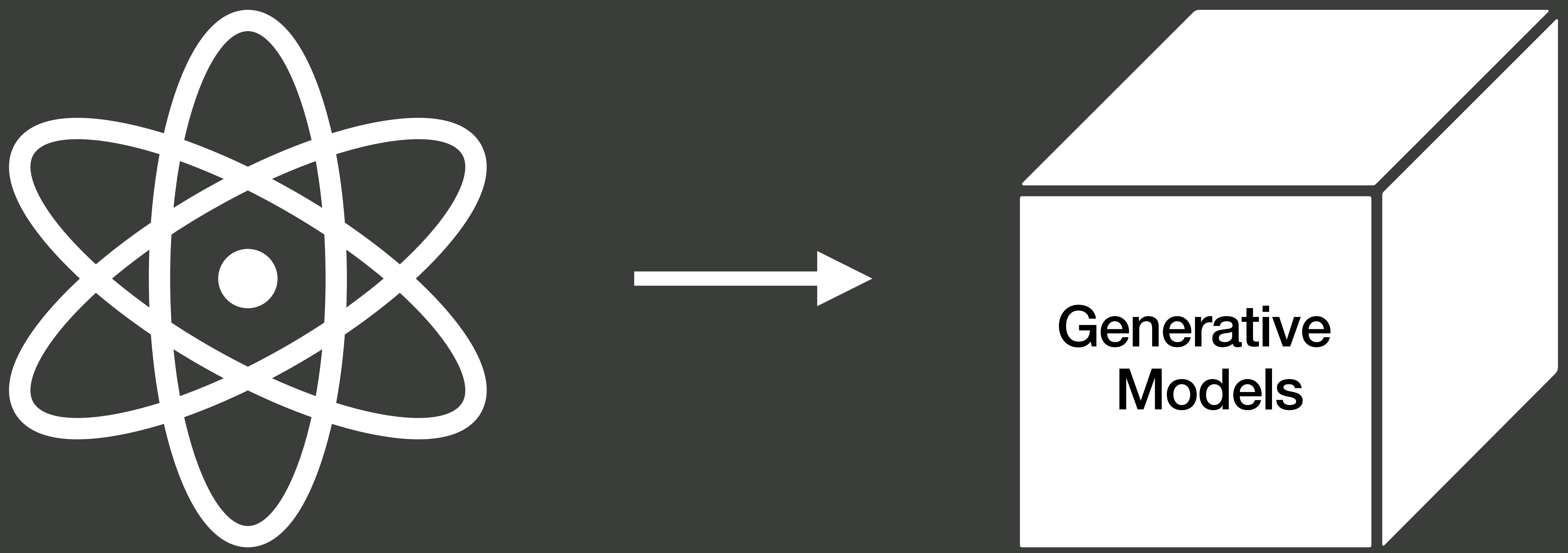


Input:

- Action* $S[U]$ ✅
- Samples (HMC) ⚠️

*Here U should denote generalized field configurations e.g., scalar fields, gauge fields.

Normalizing Flows for LFT



Input:

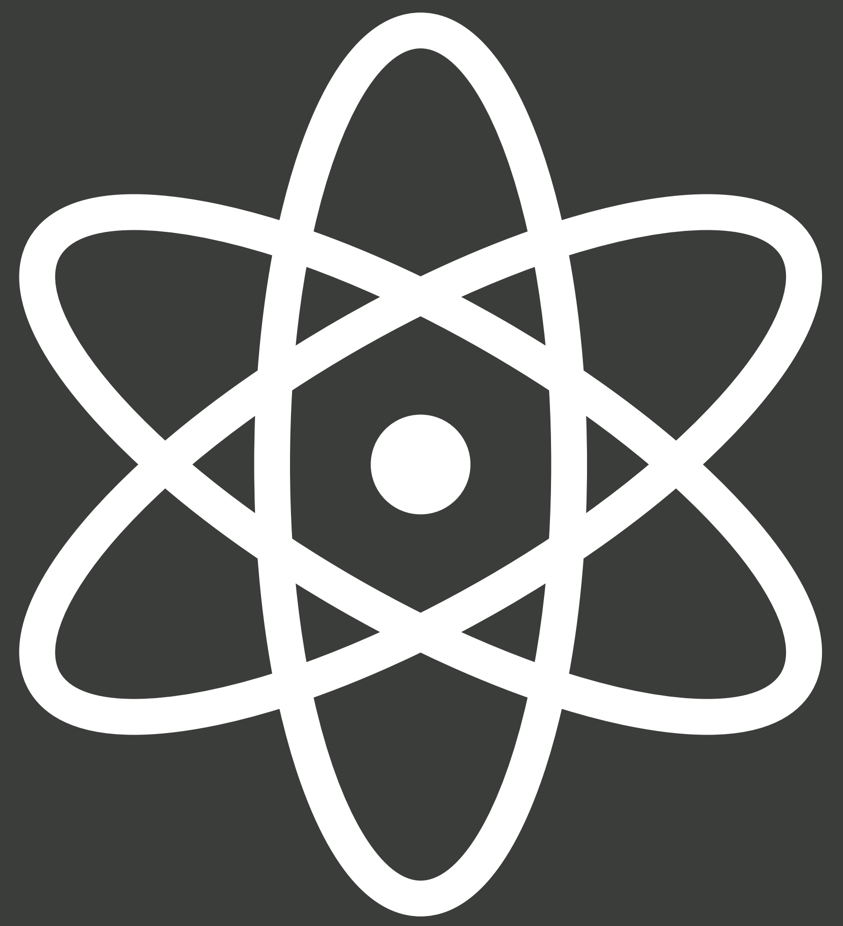
- Action* $S[U]$ ✅
- Samples (HMC) ⚠️

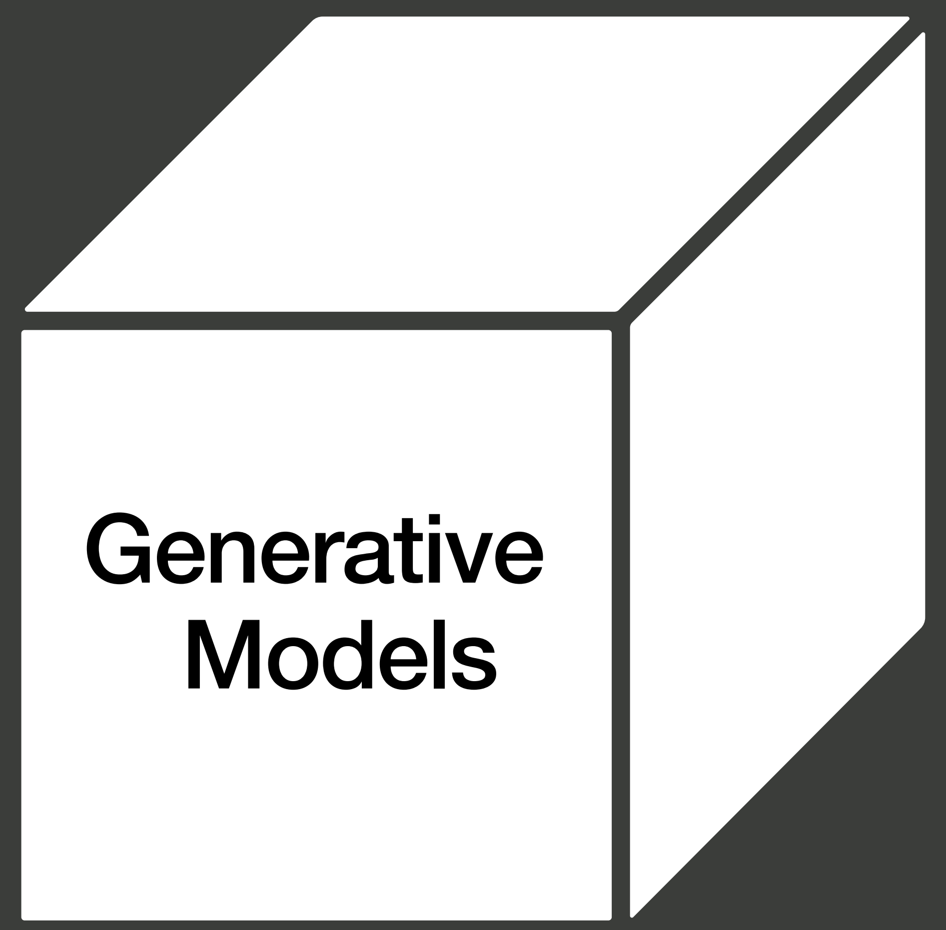
ML black(-box) magic:

- Train generative models
  - **Normalizing Flows**
  - Autoregressive Models
  - Diffusion Models
- Learn normalized densities

*Here U should denote generalized field configurations e.g., scalar fields, gauge fields.

Normalizing Flows for LFT





$$q_\theta(U) \approx \frac{e^{-S[U]}}{Z}$$

Input:

- Action* $S[U]$ ✅
- Samples (HMC) ⚠️

*Here U should denote generalized field configurations e.g., scalar fields, gauge fields.

ML black(-box) magic:

- Train generative models
  - **Normalizing Flows**
  - Autoregressive Models
  - Diffusion Models
- Learn normalized densities

Output:

- Normalized density $q_\theta(U)$
- Approximation of target $p(U)$
- Embarrassingly parallel sampling

⚠️ Not possible with standard HMC

Why Normalizing Flows for LFT?

- Lattice configurations are sampled i.i.d., thus **reducing autocorrelation**.
- Sampling is **embarrassingly parallel**, e.g., faster and more efficient.
- Direct estimation of **thermodynamic observables** (partition function, free energy, etc.).
- **Inductive biases**, e.g., symmetries, are easy to incorporate.
- The trained models can be used for **interpolation** (extrapolation) in parameter space.
- **Transfer** weights of flows trained on smaller systems to train on larger ones.
- …

$\longrightarrow$ Plenary talk by Gurtej Kanwar (LATTICE 23)

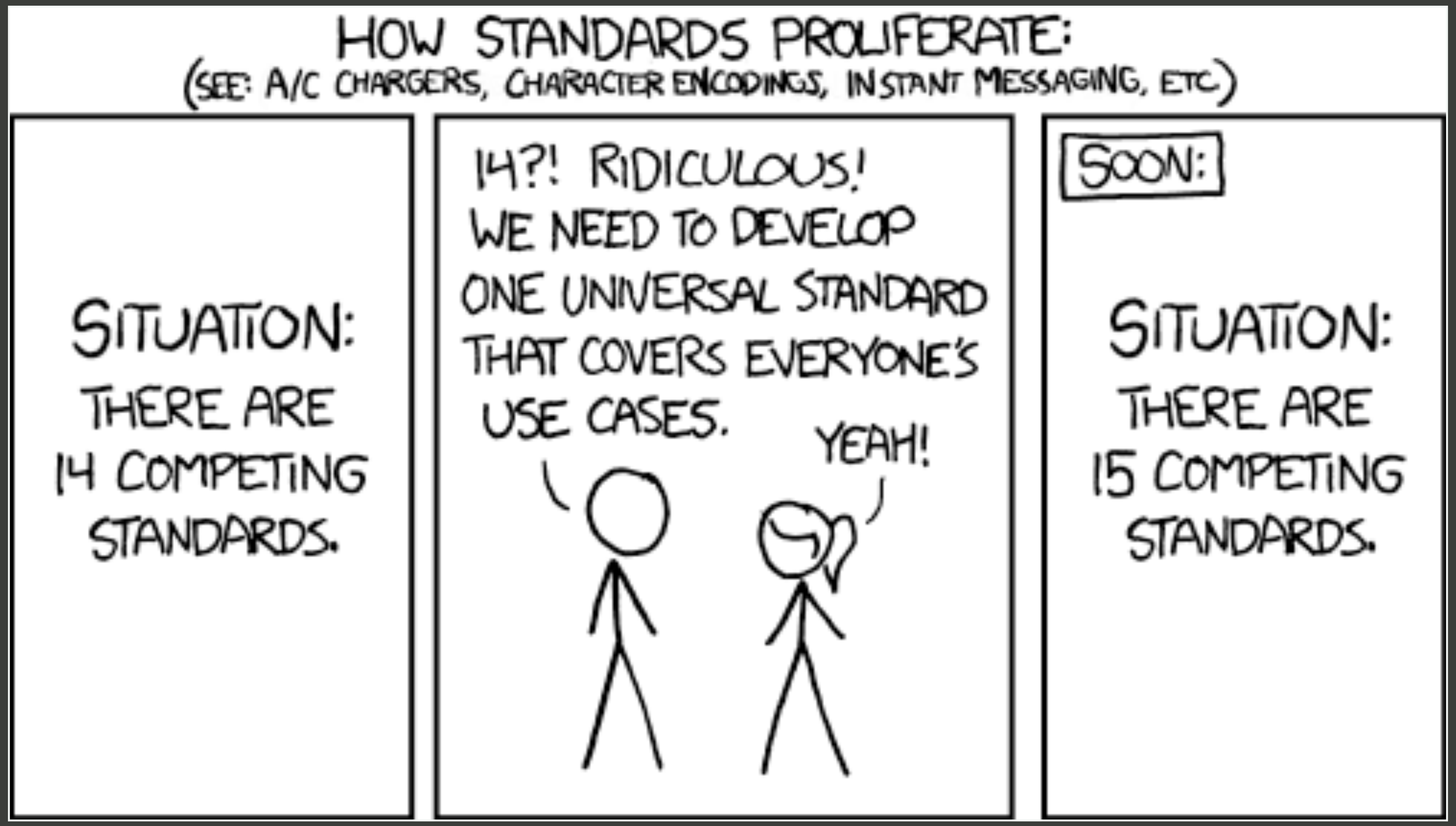
What is NeuLat?

- NeuLat is an ML-based **software package** for benchmarking and test models for LFT e.g.,
  - 1+1D ϕ^4-theory
  - 1+1D $U(1)$ gauge theory
- No such effort was made **yet** to combine existing tools into one software.
- NeuLat is meant to be a **community-wide** effort.
- The **core team** of NeuLat:
  - Expertise in ML **software development**.
  - Expertise in **LFT**.
  - **Various contributions to the field:** asymptotically unbiased estimators (Nicoli K.A. et al.), thermodynamic observables (Nicoli K.A. et al.), mode-dropping estimators (Nicoli K.A. et al.), path gradients (Vaitl L. et al. (2022)), and trivializing maps with flows (Bacchio S. et al. (2023)).

What are the benefits?

- Faster development of <u>**new ideas**</u>.
- Easier to <u>**reproduce**</u> newly published results (in the flow-based sampling community).
- Easier for people to <u>**enter the community**</u> and experiment with state-of-the-art techniques.
- Save the effort of <u>**re-implementing**</u> standard methods (e.g., architecture, estimators, etc.).
- Allow for immediate <u>**extension to other fields**</u> in physics (e.g., condensed matter physics).
- Similar <u>**examples for ML frameworks**</u> in other scientific communities:
  - <u>SchNetPack - Deep Neural Networks for Atomistic Systems</u> →
  - <u>BGFlow - Boltzmann Generators (BG) and other sampling methods</u>

Why NeuLat?



Credits: xkcd.com/927

- Many tools have been **independently** proposed.
- No **reference implementation** exists.
  - Often needs to reimplement existing code.
  - Different ML libraries.
  - Different code styles and structures.
- Big (unnecessary) **overhead** (often seen also in the ML community).
- Excellent repositories are already available (though limited in scope).
  - fthmc: Field Transformation HMC
  - l2hmc-qcd l2hmc-qcd
  - nflows
  - GomalizingFlow

} 👏 Shout-out to Sam Foreman et al., for the great work!

👏 Julia package from A. Tomiya and collaborators!

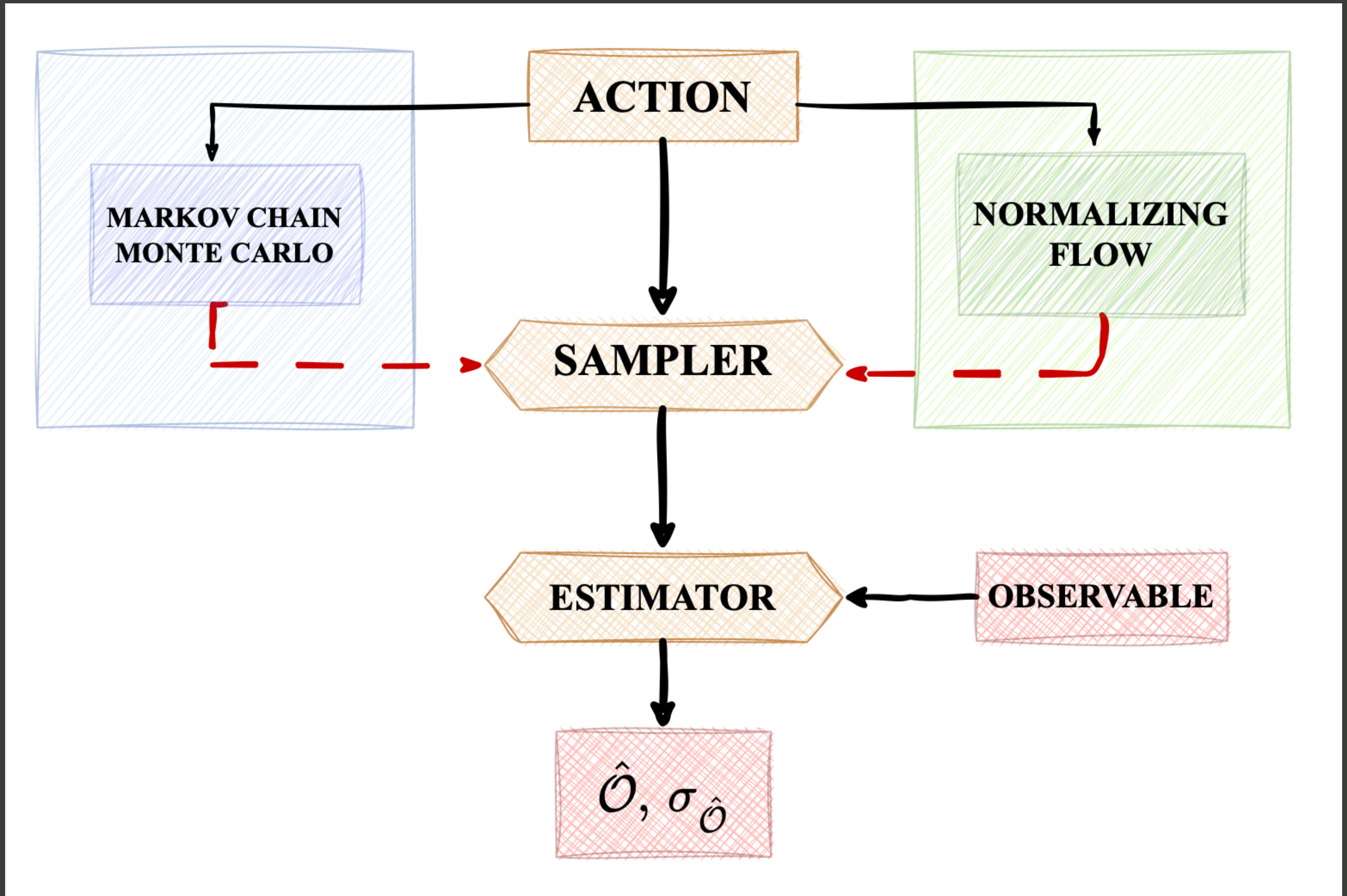
NeuLat

Goal: incorporate as many contributions from 5 years of research progress into a single software framework.

NEULAT

Software Overview

- PyTorch backend
- MCMC-based sampling
- Flow-based sampling (e.g., nflows)
- Observable estimation (e.g., py-uwerr)



Basic Workflow

Step 0: Define the action of the physical system to simulate.

Step 1: Build a Markov chain and a custom normalizing flow model.

Step 2: Specify loss and optimizer and train the flow model.

Step 3: Estimate observables using samples from HMC and trained flow.

```
n_samples, train_steps, batch = 1000, 1000, 500
action = Phi4Action(kappa=0.3, lamb=0.022)

hmc_chain = HMCMarkovChain(action, lat_shape=[16, 16], burn_in=100)
model = Flow(
    lat_shape=[16, 16],
    base_dist=NormalDistribution(),
    coupling=NiceCoupling(),
    n_couplings=6
)

optim = torch.optim.Adam(model.parameters(), lr=learning_rate)
loss_fn = ReverseKLLoss()
for _ in range(train_steps):
    configs, log_probs = model.sample_and_log_prob(batch)
    loss = loss_fn(action(configs), log_probs)
    optim.zero_grad(); loss.backward(); optim.step()

obs = (Magnetization(), AbsMagnetization(), action)

flow_obs = IidEst(obs).evaluate(model.sample(n_samples))
hmc_obs = CorrelatedEst(obs).evaluate(hmc_chain.sample(n_samples))
```

Key Features

- **Density Estimation**: Learn approximations of targeted Boltzmann distributions $q_\theta \approx p$.
- **Sampling:**
  - MCMC implementations (HMC, Cluster algorithms, etc.).
  - Neural Importance Sampling (see Albergo et al. (2019), Kanwar et al. (2020), Nicoli et al. (2021)+ refs. therein).
  - Neural HMC (see same papers referenced above).
- **Estimation**:
  - Asymptotically unbiased estimators for physical observables (see Nicoli et al. (2020)).
  - Direct estimation of thermodynamic observables with flows and HMC (see Nicoli et al. (2021)).
  - Sampling in the presence of mode-collapse (see Nicoli et al. (2023)).
- **Customizable**: Easy to incorporate a new action/theory or customize new, equivariant flow-layers.

… and more!

Conclusion

- We presented **NeuLat**, a software for flow-based simulation of LFT.
- The software is meant to be accessible, **modular**, and easy to **extend** and **maintain**.
- This eliminates the **overhead** of re-implementing existing code between different formats.
- The first **release** of the software is planned for the **upcoming months**.
- NeuLat is aimed to be a **community-wide effort**. Get in touch if you would like to contribute.