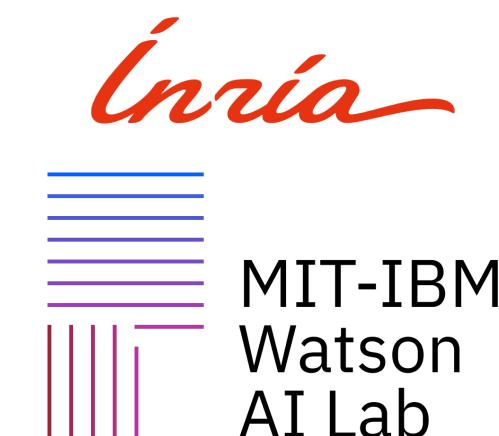
Automatic Guide Generation for Stan via NumPyro





Guillaume Baudart

Louis Mandel

Inria, ENS

IBM Research

Variational Inference

Basics

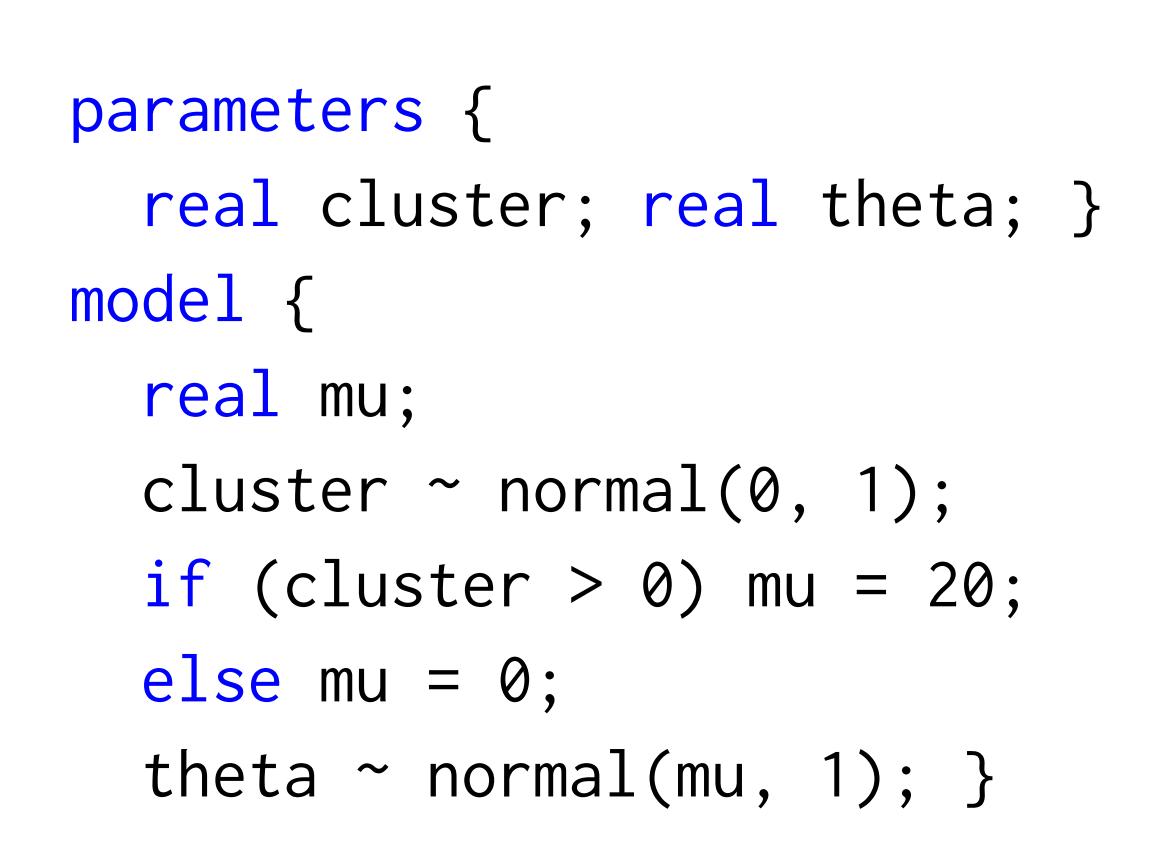
- Guide: define a family of distributions
- Inference: find the closest member to the true posterior distribution
- $q_{\theta^*}(z) = \underset{\theta \in \Theta}{\operatorname{argmin}} \operatorname{KL} (q_{\theta}(z) \mid | p(z \mid \theta))$

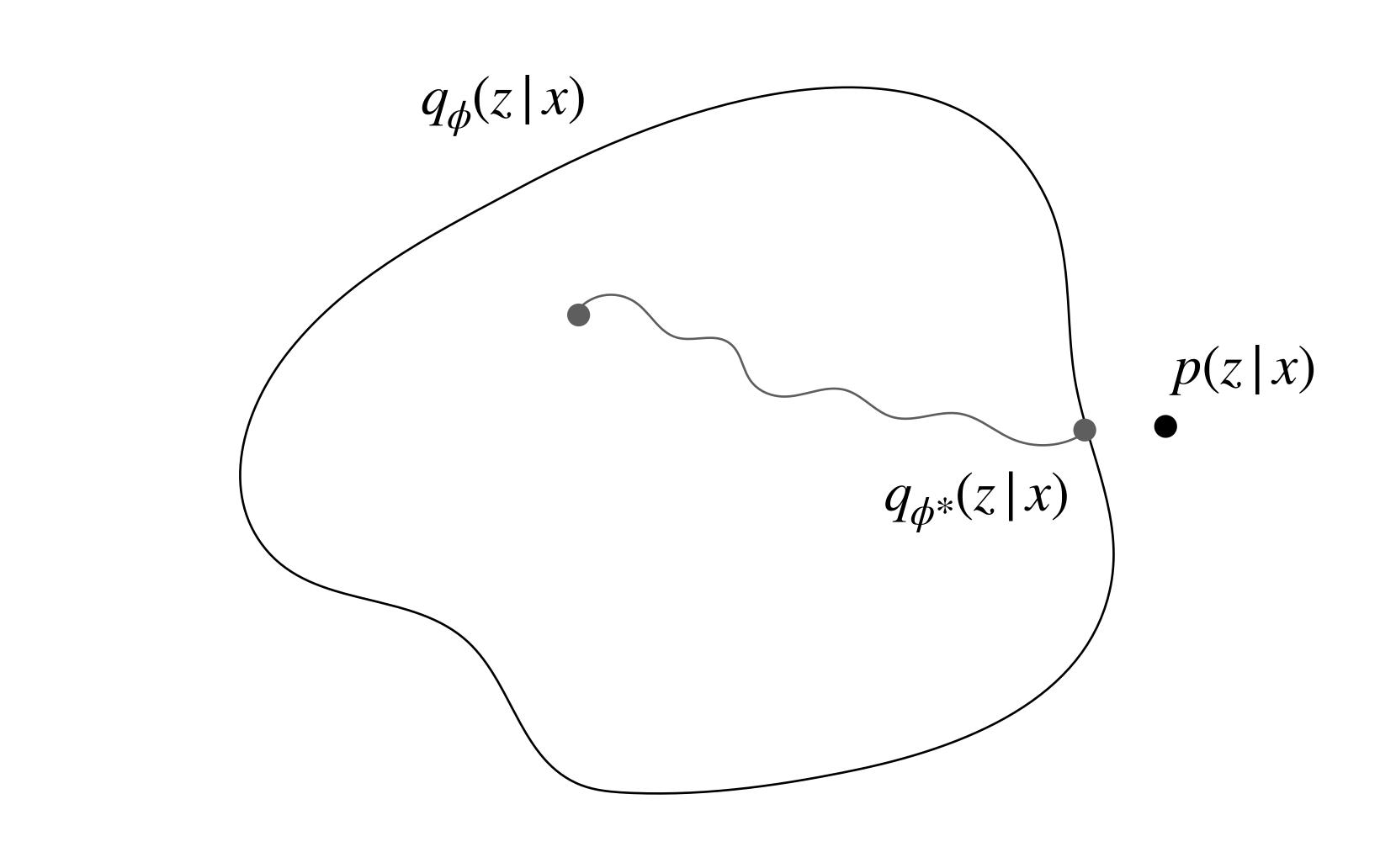
Crafting Guides

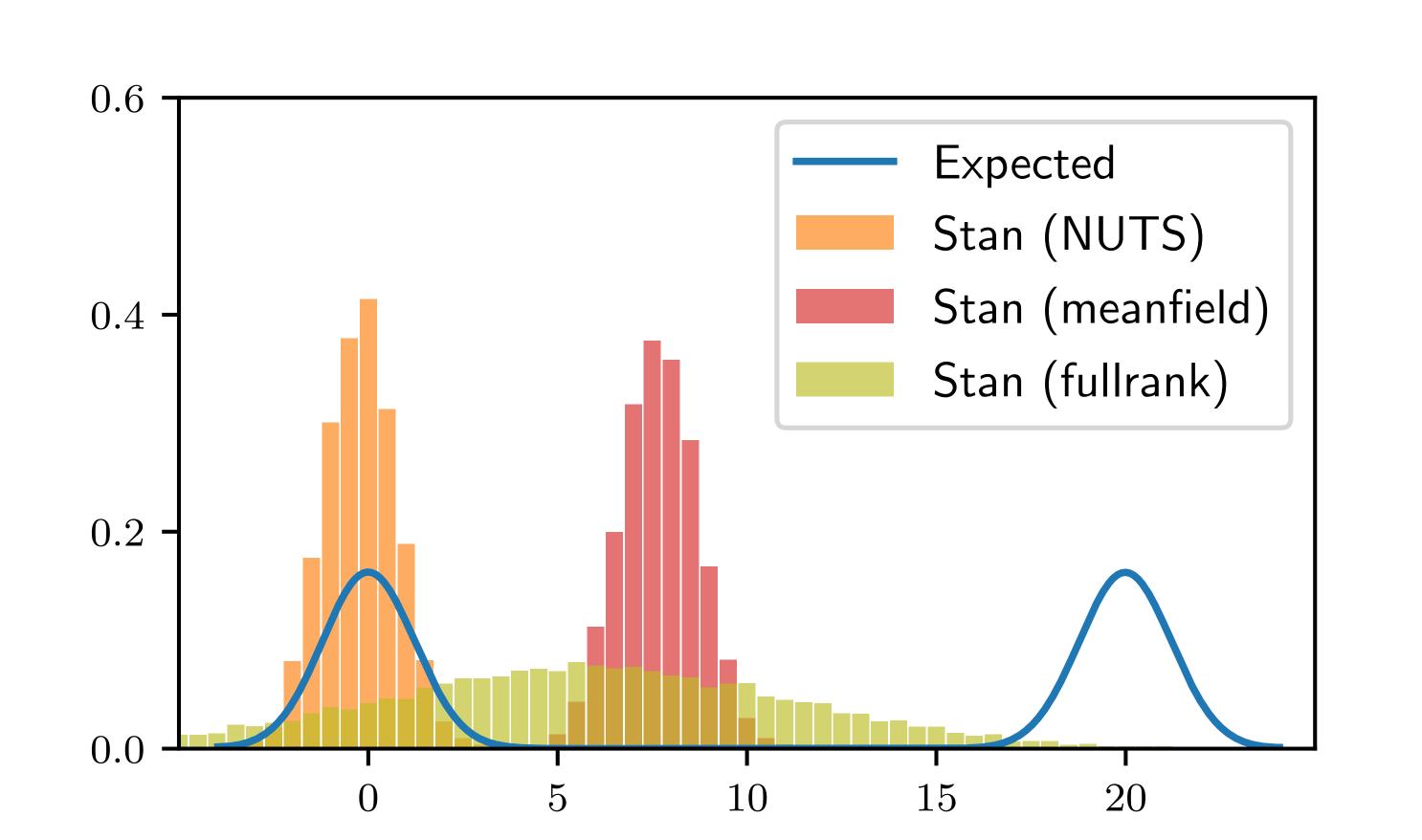
- Manual: complex and error prone
- Synthesized: how? accuracy?

Stan ADVI [Kucukelbir et al. 2015]

- Meanfield: fully factorized Gaussian
- Fullrank: Gaussian with a full-rank covariance



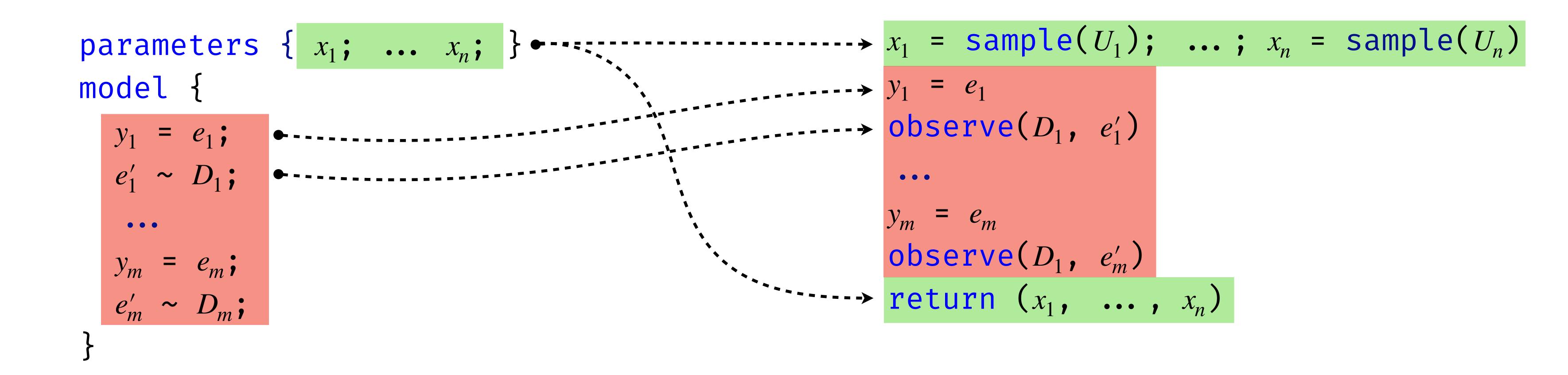




A Compiler from Stan to NumPyro [PLDI 2021]

Theorem: For all Stan program p, the semantics of the source and the compiled programs are equal up to a constant:

$$\{ [p] \}_D \propto \{ [\mathscr{C}(p)] \}_D$$



Experimental Validation

- Comparison NumPyro NUTS with Stan NUTS
- No impact on accuracy
- Significant speedup on some models (JAX)

Is it true for other inference methods, e.g., variational inference?

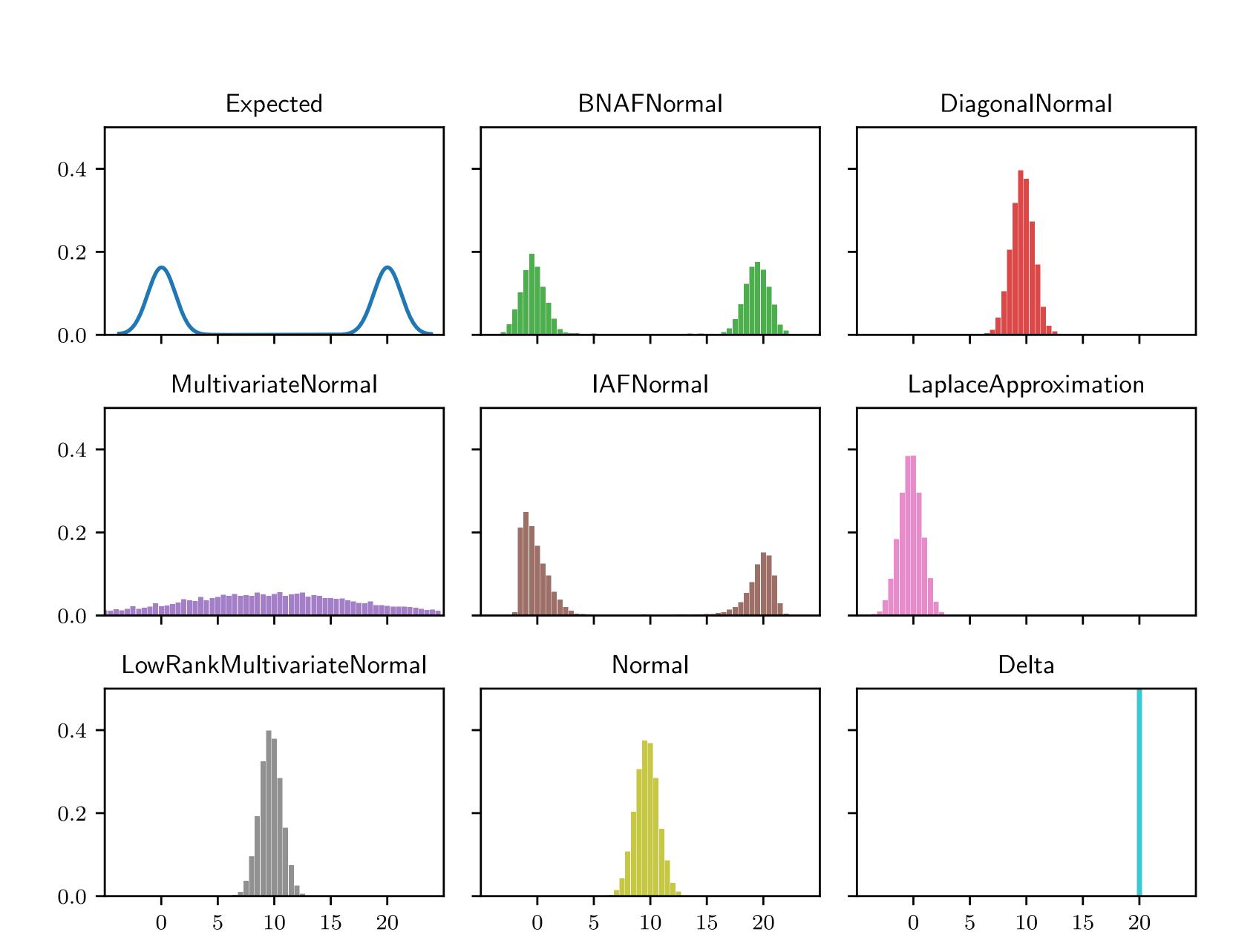
NumPyro Autoguides for Stan

Idea

- Compile Stan model to NumPyro [Phan et al. 2019]
- Instantiate one of numpyro.autoguides
- Run the inference

Advantages

- Zoo (8) of possible autoguides
- Fast optimizers (JAX)
- Lightweight wrapper for the NumPyro runtime



from stannumpyro import NumPyroModel
import numpyro.infer import Trace_ELBO, autoguide as autoguide
from numpyro.optim import Adam
from jax.random import PRNGKey

numpyro_model = NumPyroModel("multimodal.stan")
guide = autoguide.AutoBNAFNormal(numpyro_model.get_model())
svi = numpyro_model.svi(Adam(step_size=0.0005), Trace_ELBO(), guide)
svi.run(PRNGKey(0), {}, num_steps=10000, num_samples=10000)

Evaluation

PosteriorDB [Magnusson et al. 2021]

- 49 (model, dataset, reference samples)
- Exclude 6 models (missing ODE solver)

Experiments

- Run 100,000 inference steps
- Generate 10,000 samples from the posterior
- Compare error w.r.t. the reference samples

$$err = \frac{|\operatorname{mean}(x_{\operatorname{ref}}) - \operatorname{mean}(x)|}{\operatorname{stddev}(x_{\operatorname{ref}})}$$

Conclusions, Future Works

- Illustrate trade-off complexity/accuracy
- Possible test framework for future guide
- More guides? More models?
- Compile new features of the Stan language

