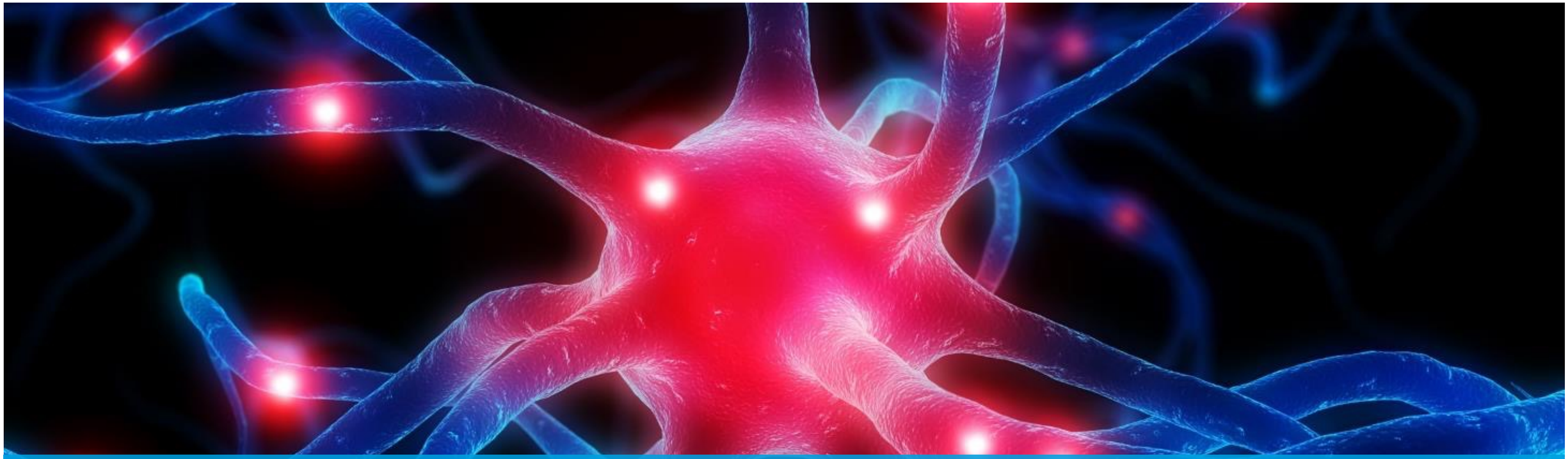


Optimal design of experiments using amortized Bayesian inference

Matthias Brückner
Janssen Pharmaceutica
21 October 2022

Active neuron

janssen Neuroscience
PHARMACEUTICAL COMPANIES OF Johnson & Johnson

Bayesian Experimental Design

Maximizing expected utility

- Prior $p(\theta)$, likelihood $p(x|\theta;\eta)$, utility function $u(\theta, x, \eta)$
- The utility function depends unknown model parameters and on data that has not yet been observed
- Objective: Find design maximizing the expected utility ("averaging over what is unknown")

$$U(\eta) = E_{\theta,x}[u(\theta, x, \eta)]$$

- Examples
  - Expected Kullback-Leibler divergence between posterior and prior: $U_1(\eta) = E_{\theta,x}[\log p(\theta|x;\eta)]$ (D-optimal)
  - Expected mean-squared error: $U_2(\eta) = -E_{\theta,x}[(\theta - \hat{\theta})^T A(\theta - \hat{\theta})]$ (A-optimal)

$$U_3(\eta) = -E_{\theta,x}[(\theta - \hat{\theta})^T cc^T(\theta - \hat{\theta})]$$ (c-optimal)

Chaloner and Verdinelli (1995): Bayesian Experimental Design: A Review

Bayesian Experimental Design

Controlling model performance criteria

Choose design that achieves a certain level of performance:

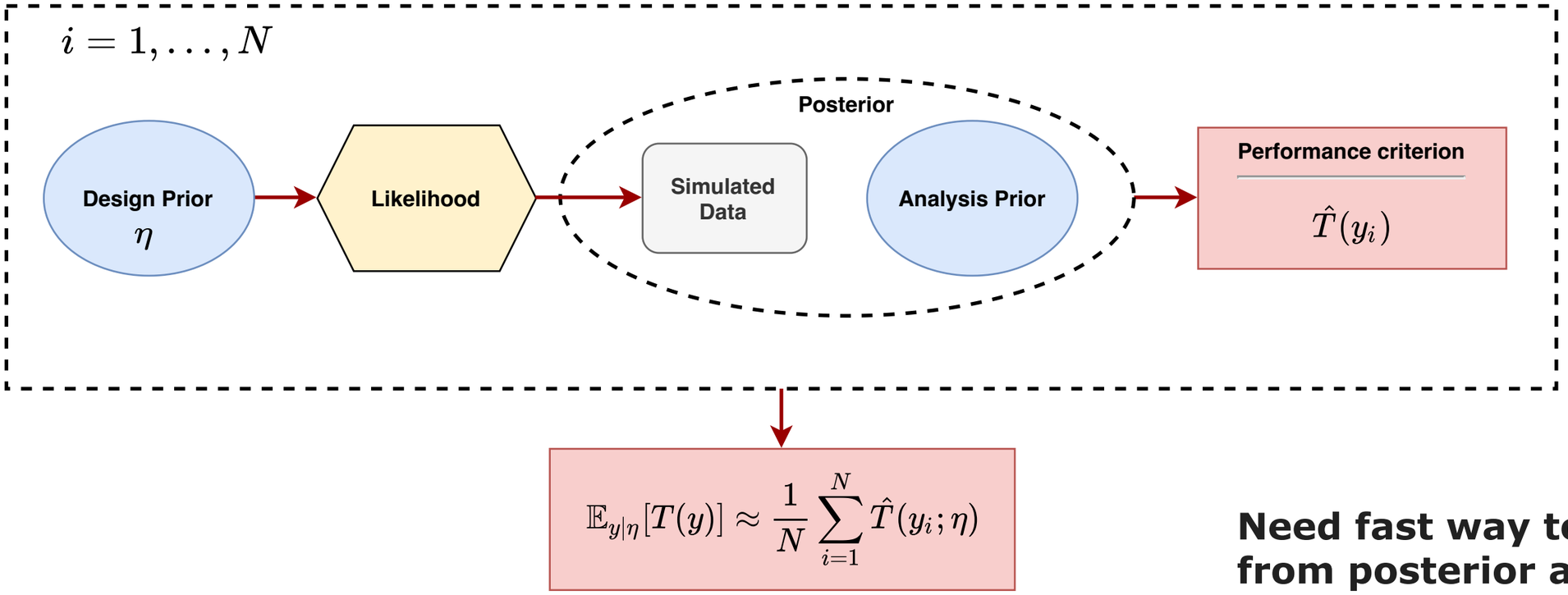
$$\mathbb{E}_{y|\eta}[T(y)] \leq \epsilon$$

Average posterior variance criterion (APVC)	$\mathbb{E}[\mathrm{var}(\delta\|y)] \leq \epsilon$
Average coverage criterion (ACC)	$\mathbb{E}[\mathrm{P}(\delta \in A(y)\|y)] \geq 1-\alpha$
Average length criterion (ALC)	$\mathbb{E}[\|A(y)\|] \leq l$
Average posterior probability (APP) of detecting an effect of size at least δ^*	$\mathbb{E}[P(\delta > \delta^*\|y)] \geq 1-\alpha$

Gelfand and Wang (2002): A simulation-based approach to Bayesian sample size determination for performance under a given model and for separating models

Bayesian Experimental Design

Monte-Carlo estimation

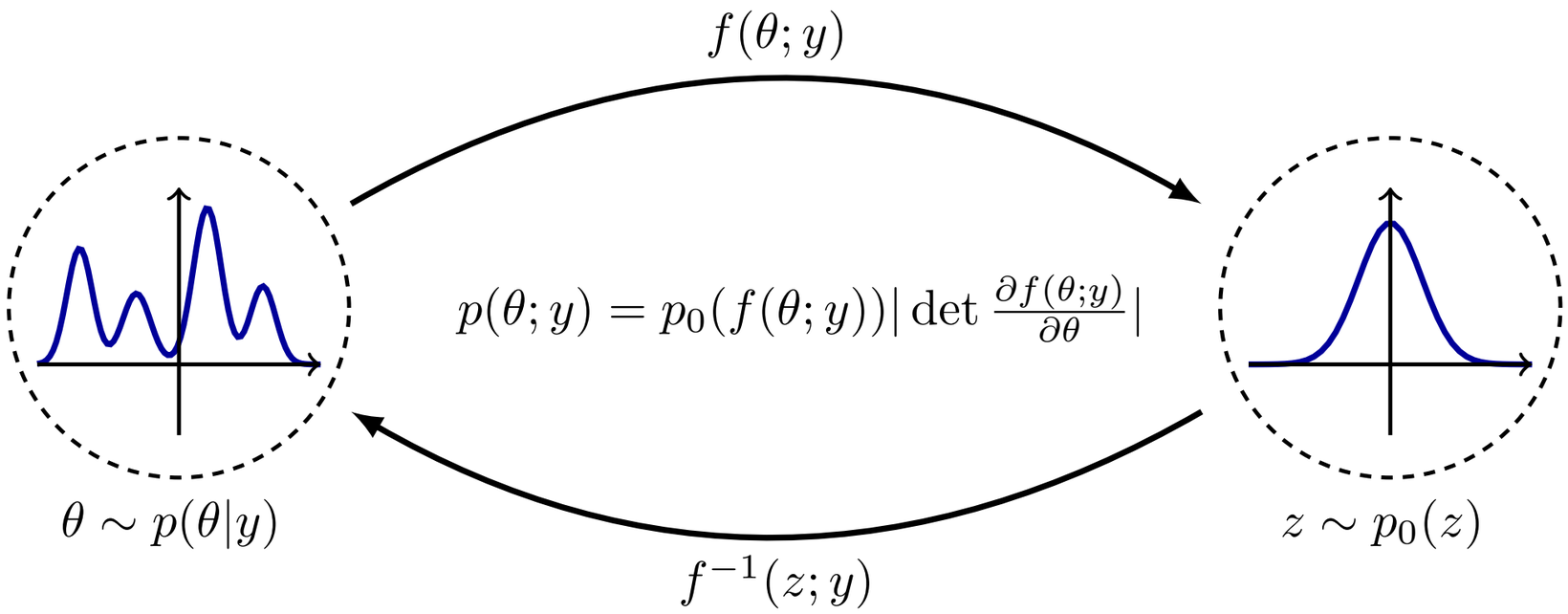


Need fast way to sample from posterior and evaluate posterior density!

Gelfand and Wang (2022): A simulation-based approach to Bayesian sample size determination for performance under a given model and for separating models

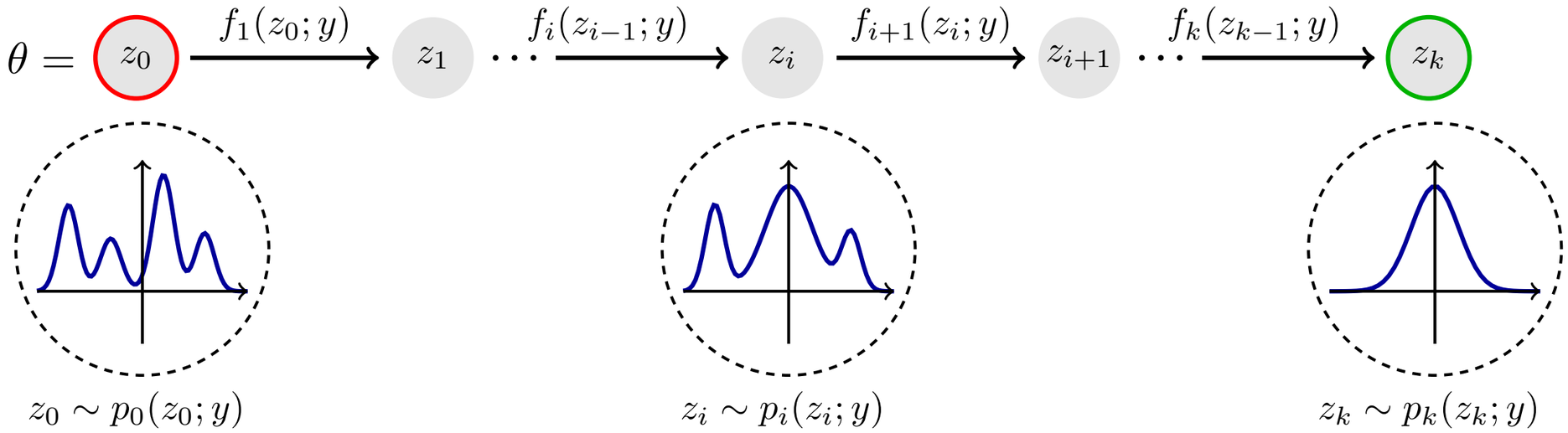
Conditional Normalizing Flows

Change of variables



Conditional Normalizing Flows

Composing simple transformations



- Simple transformations as building blocks - each having a tractable inverse and Jacobian determinant – to define a complex transformation

Papamakarios et al. (2019): Normalizing Flows for Probabilistic Modeling and Inference

Conditional Normalizing Flows

Affine coupling transformations

Split input vector into two halfs, only scale and shift second half

Scale and shift can be arbitrary functions of the first half and the data

$$\mathbf{v} = f(\mathbf{u}; x) \qquad \mathbf{u} = f^{-1}(\mathbf{v}; x)$$

$$\mathbf{v}_{1:d} = \mathbf{u}_{1:d}$$
$$\mathbf{v}_{d+1:D} = \mathbf{u}_{d+1:D} \odot \exp(s(\mathbf{u}_{1:d}; x)) + t(\mathbf{u}_{1:d}; x)$$

$$\mathbf{u}_{1:d} = \mathbf{v}_{1:d}$$
$$\mathbf{u}_{d+1:D} = (\mathbf{v}_{d+1:D} - t(\mathbf{u}_{1:d}; x)) \odot \exp(-s(\mathbf{v_{1:d}}; x))$$

Jacobian of this transformation is lower triangular matrix

Computation of determinant in linear time $\mathcal{O}(D)$ vs $\mathcal{O}(D^3)$ for general $D \times D$ matrices

$$\left|\det \frac{\partial f}{\partial u}\right| = \prod_{j=1}^{D} \exp(s(u_{1:d}; x))_j = \mathcal{O}(D)$$

Dinh et al (2017): Density estimation using Real NVP

Conditional Normalizing Flows

Loss function

- Expected Kullback-Leibler divergence between true posterior and the flow-based model p_ϕ

$$L(\phi) = \mathbb{E}_y[D_{KL}(p(\theta|y)\|p_\phi(\theta|y))]$$
$$= -\mathbb{E}_{\theta,y}[\log p(f_\phi(\theta;y)) + \log|\det \frac{\partial f_\phi(\theta;y)}{\partial\theta}|] + const.$$

- Approximate loss function using a batch of samples from $p(\theta, x)$

$$L(\phi) \approx -\frac{1}{B}\sum_{i=1}^{B} \log p(f_\phi(\theta_i;y_i)) + \log|\det \frac{\partial f_\phi(\theta_i;y_i)}{\partial\theta}| + const.$$

- Minimize the loss function iteratively with stochastic gradient-based methods

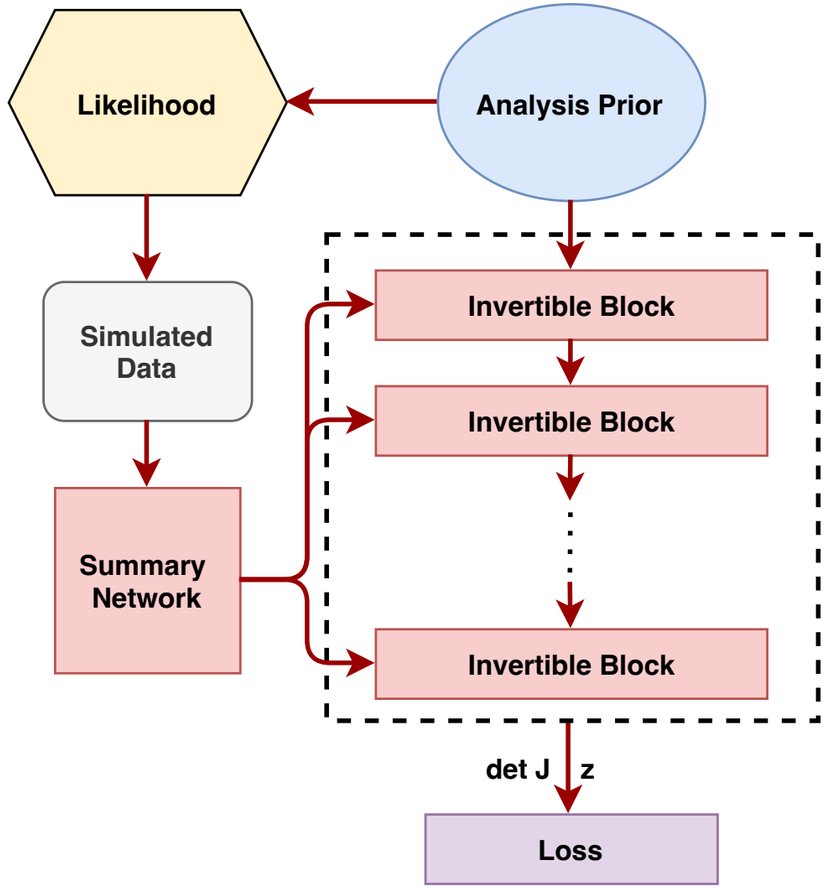
Papamakarios et al. (2019): Normalizing Flows for Probabilistic Modeling and Inference
Radev et al. (2020): BayesFlow: Learning Complex Stochastic Models with Invertible Neural Networks

Training Phase

BayesFlow (Radev et al. 2020)

BayesFlow: Implementation of normalizing flows with affine coupling layers in Python using TensorFlow

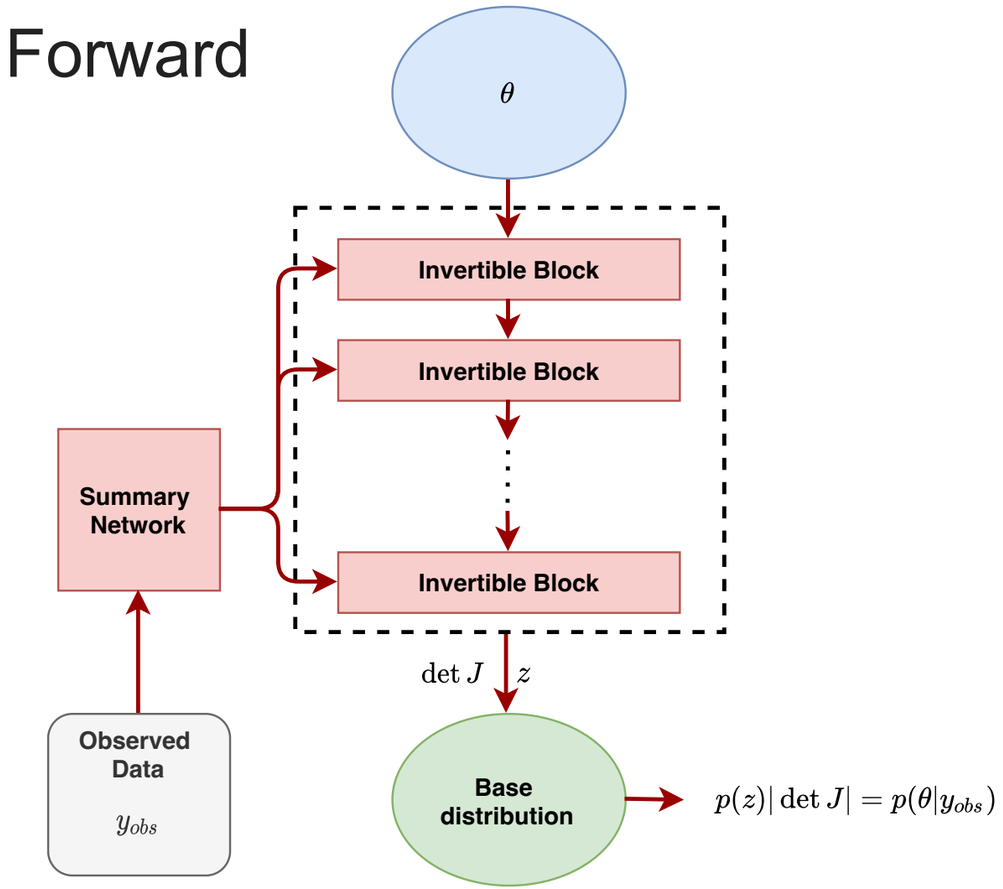
The summary network transforms input data of variable size to a fixed length summary vector.

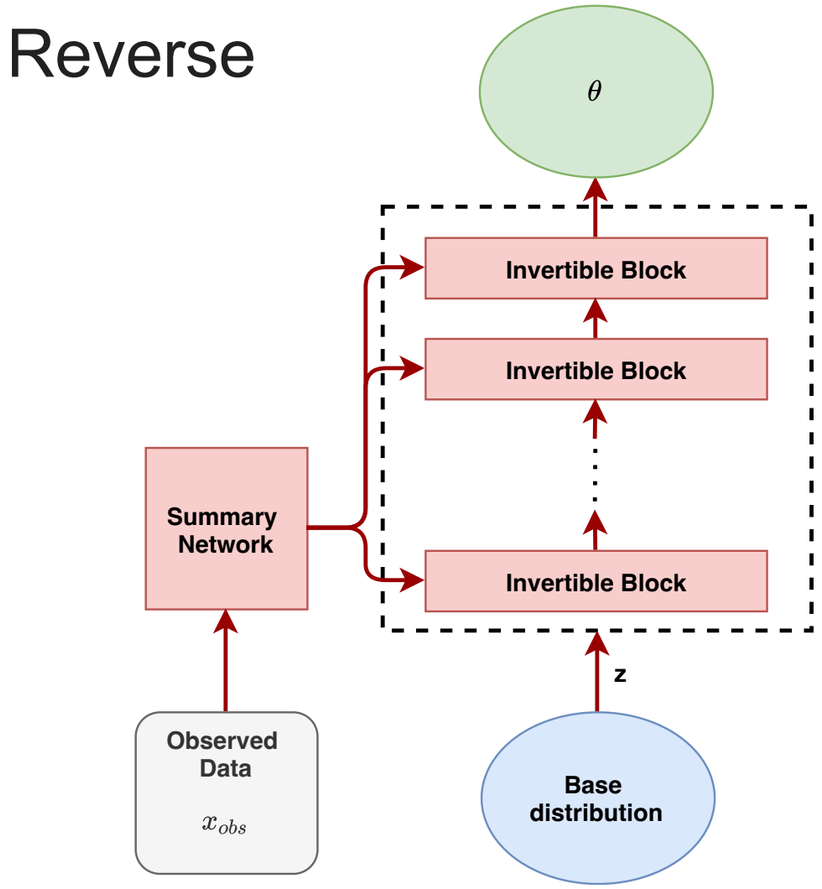


Radev, S. T., Mertens, U. K., Voss, A., Ardizzone, L., & Köthe, U. (2020). BayesFlow: Learning complex stochastic models with invertible neural networks. *IEEE Transactions on Neural Networks and Learning Systems*
https://github.com/stefanradev93/BayesFlow

Inference Phase

BayesFlow (Radev et al. 2020)



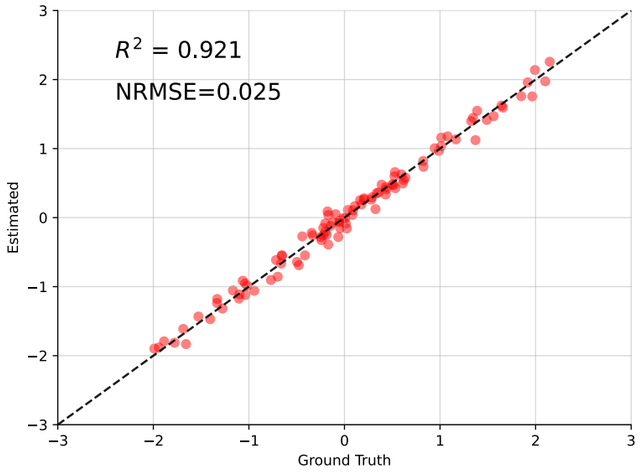


Radev, S. T., Mertens, U. K., Voss, A., Ardizzone, L., & Köthe, U. (2020). BayesFlow: Learning complex stochastic models with invertible neural networks. *IEEE Transactions on Neural Networks and Learning Systems*

Convergence Diagnostics

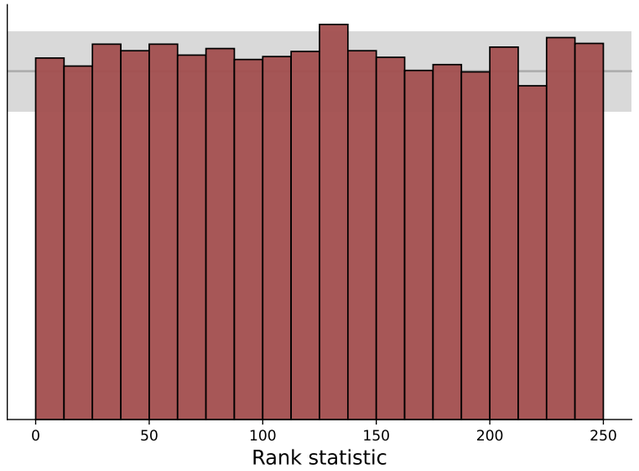
Recovery

Posterior means vs true parameters



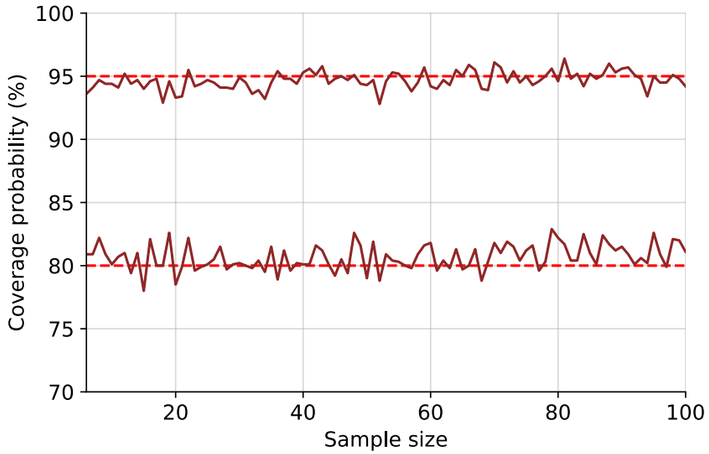
Simulation Based Calibration

Histogram of rank statistics



Calibration error

Coverage probability of posterior credible intervals



Radev, S. T., Mertens, U. K., Voss, A., Ardizzone, L., & Köthe, U. (2020). BayesFlow: Learning complex stochastic models with invertible neural networks. *IEEE Transactions on Neural Networks and Learning Systems*

Talts, S., Betancourt, M., Simpson, D., Vehtari, A., & Gelman, A. (2018). Validating Bayesian inference algorithms with simulation-based calibration.

janssen Neuroscience
PHARMACEUTICAL COMPANIES OF Johnson & Johnson

Importance sampling

Flow-based posterior as proposal distribution

- Flow-based posterior approximates the true posterior
- Finite training time → non-zero approximation error → biased estimates of $\mathbb{E}[h(\theta)|y]$
- Importance weights: $w_i = \frac{p(\theta_i)p(y|\theta_i)}{p_\phi(\theta_i|y)}$
- Unbiased estimates:

$$\mathbb{E}[h(\theta)|y] = \int h(\theta)p(\theta|y)\mathrm{d}\theta \approx \frac{\sum_{i=1}^{N} h(\theta_i)w_i}{\sum_{i=1}^{N} w_i}$$

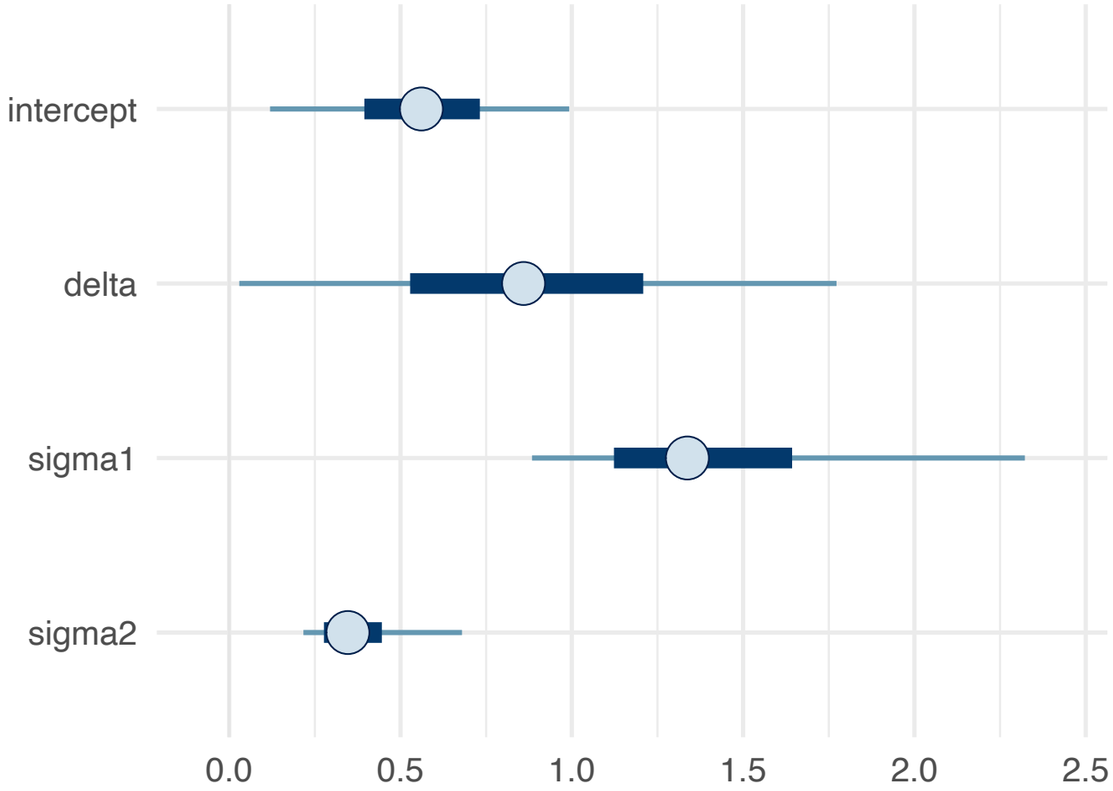
- Caveat: Density evaluation requires extra forward pass in addition to the reverse for sampling

Müller, T., McWilliams, B., Rousselle, F., and Gross, M. & Novak, J. (2019): Neural Importance Sampling. ACM Trans. Graph.

Example

GPR39 data

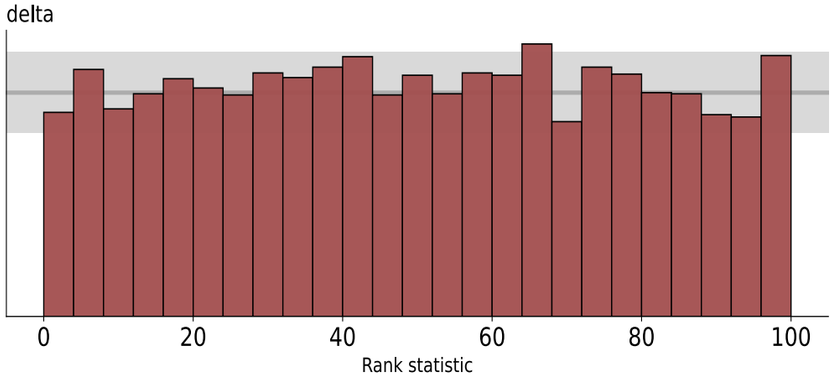
- Pilot study comparing acid secretion after treatment with vehicle or GPR39 agonist
- Vehicle: 7 mice, Compound: 8 mice
- Use Stan to fit univariate two-sample normal model with heterogeneous variance with weakly informative priors.
- Use posterior as informative prior distribution

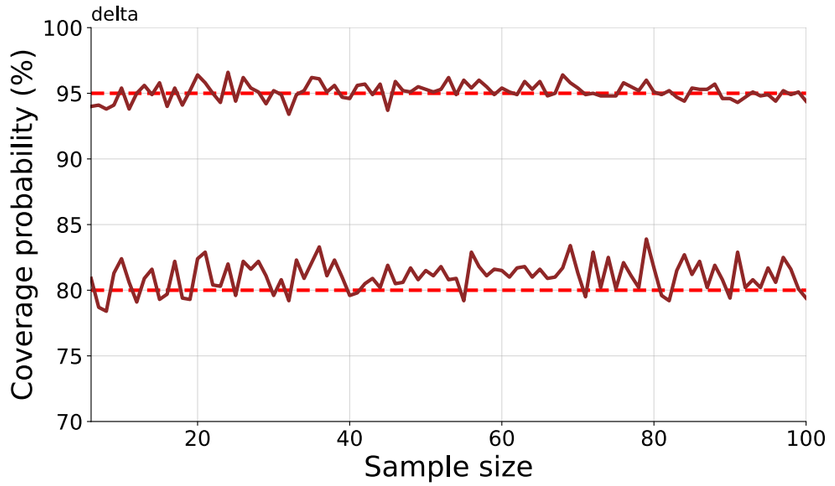


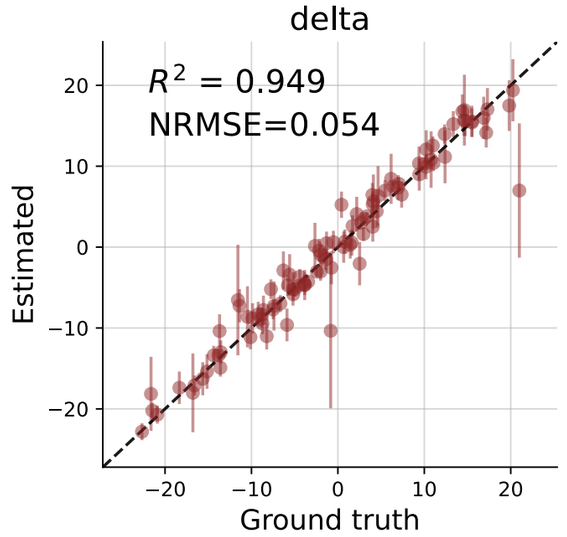
GPR39

Training

- 6 affine coupling layers (3 layers in BayesFlow)
- Default settings for summary network (invariant network that is well suited for iid observations)
- "Online" training: new dataset simulated at every iteration
- Amortize over
  - sample size (Uniform[6, 100])
  - allocation ratio (Uniform[0,1])





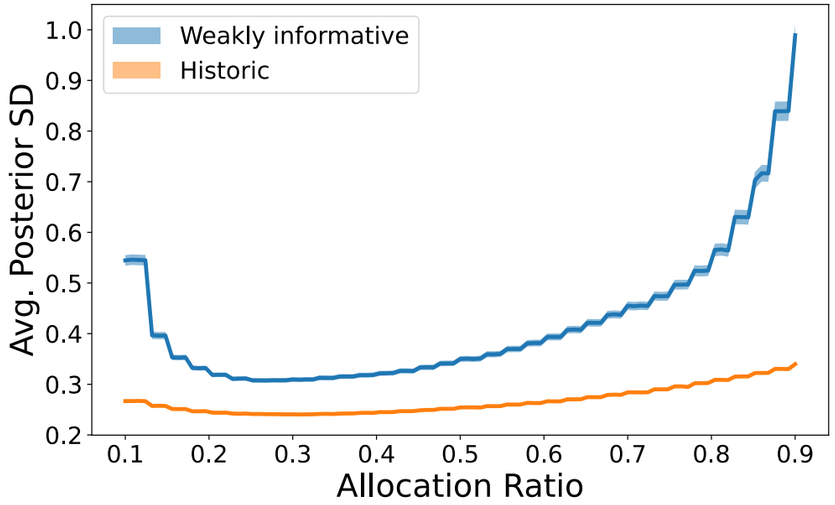


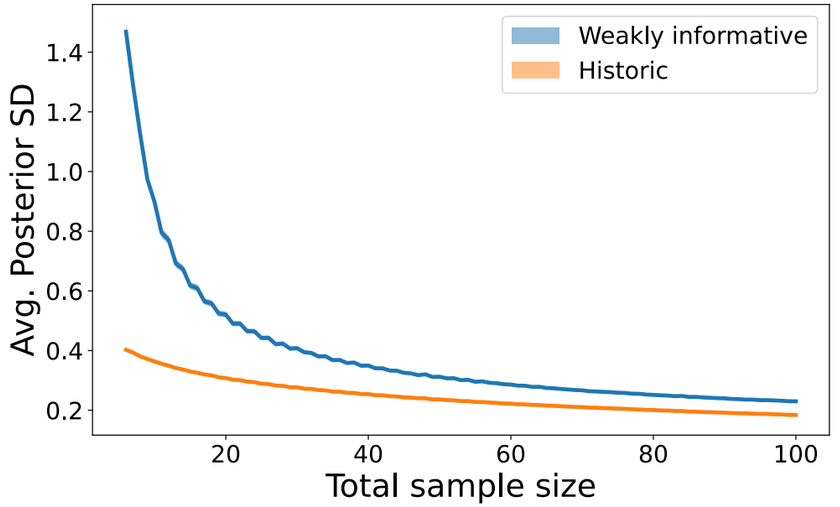
GPR39

Results

N = 40

Delta = 0.87



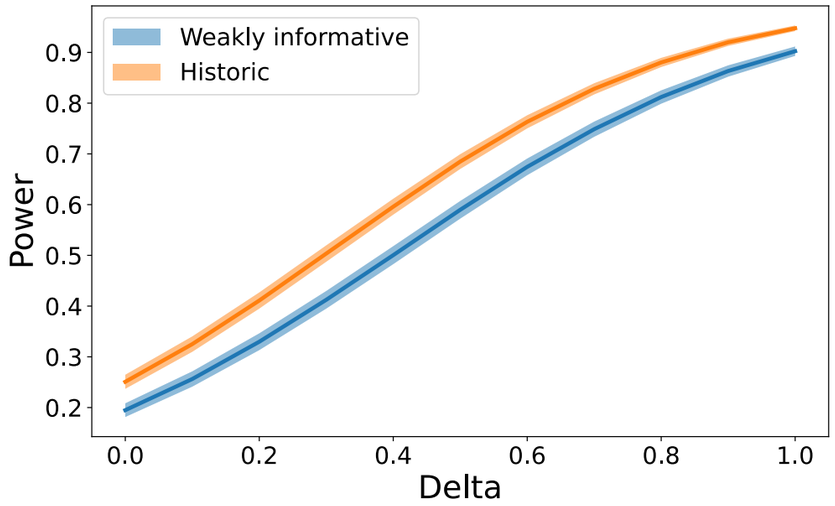


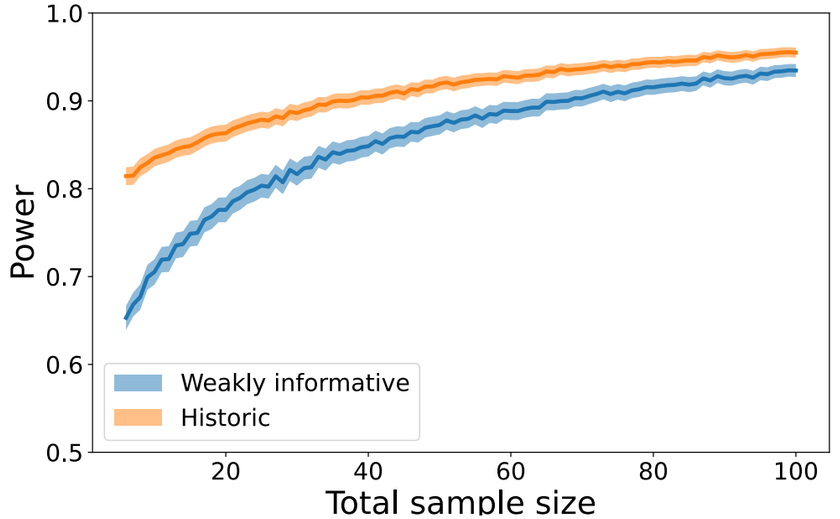
Delta = 0.87

Alloc = 50%

N = 40

Alloc = 50%





Delta = 0.87

Alloc = 50%

janssen Neuroscience

PHARMACEUTICAL COMPANIES OF Johnson & Johnson

Stan vs BayesFlow

Timings for GPR39

TIME	MCMC (Stan)	BayesFlow (CPU + GPU)
Training Time	0	33 min
Inference Time		
Single Dataset (1000 posterior samples)	0.15s	0.002s
Single Design (1000 datasets)	150s	0.46s
Full evaluation (≈300 designs):	12.5h	138s
Total time (Training + Inference)	12.5h	35.3min

Conclusion

- Can examine large number of design scenarios in reasonable time
- Can use any model that can be easily sampled from (including models with an intractable likelihood)
- Training and inference phase are separated
- Issues:
  - Numerical stability during training main problem: normalization is essential
  - “Simulation gap”: amortized Bayesian methods might yield wrong posterior inference when used with observed data which is atypical under the assumed simulation model
  - Hyperparameters: How many coupling layers?
  - Convergence checking: How much training is enough?

References

- Chaloner and Verdinelli (1995): Bayesian Experimental Design: A Review. Statistical Science
- Cranmer et al. (2020): The frontier of simulation-based inference. PNAS
- Dinh et al (2017): Density estimation using Real NVP
- Gelfand and Wang (2002): A simulation-based approach to Bayesian sample size determination for performance under a given model and for separating models. Statist. Sci.
- Müller et al. (2019): Neural Importance Sampling. ACM Trans. Graph.
- Papamakarios et al. (2021): Normalizing Flows for Probabilistic Modeling and Inference. JMLR
- Radev et al. (2020). BayesFlow: Learning complex stochastic models with invertible neural networks. IEEE Transactions on Neural Networks and Learning Systems
- Talts et al. (2018). Validating Bayesian inference algorithms with simulation-based calibration.


Janssen
Neuroscience
PHARMACEUTICAL COMPANIES OF
Johnson & Johnson