Bayes in the Age of Machine Learning

Paul Bürkner, TU Dortmund University

Standard Bayesian Inference

We care about the posterior distribution

$$p(\theta \mid y) = p(y \mid \theta)\, p(\theta) \,/\, p(y)$$

of a probabilistic model $p(\theta, y) = p(y \mid \theta)\, p(\theta)$

We use sampling algorithms such as MCMC to obtain draws $\theta^{(s)} \sim p(\theta \mid \tilde{y})$ from the posterior given observed data $\tilde{y}$

We can then propagate uncertainty to any implied quantity $z = z(\theta, y)$ by sampling from its posterior predictive distribution:

$$\theta^{(s)} \sim p(\theta \mid \tilde{y}), \quad y^{(s)} \sim p(y \mid \theta^{(s)}), \quad z^{(s)} = z(\theta^{(s)}, y^{(s)})$$

When do we benefit from standard Bayesian inference?

Which uncertainties do we care about?

- Uncertainty on observable variables?
- Uncertainty on unobservable variables?

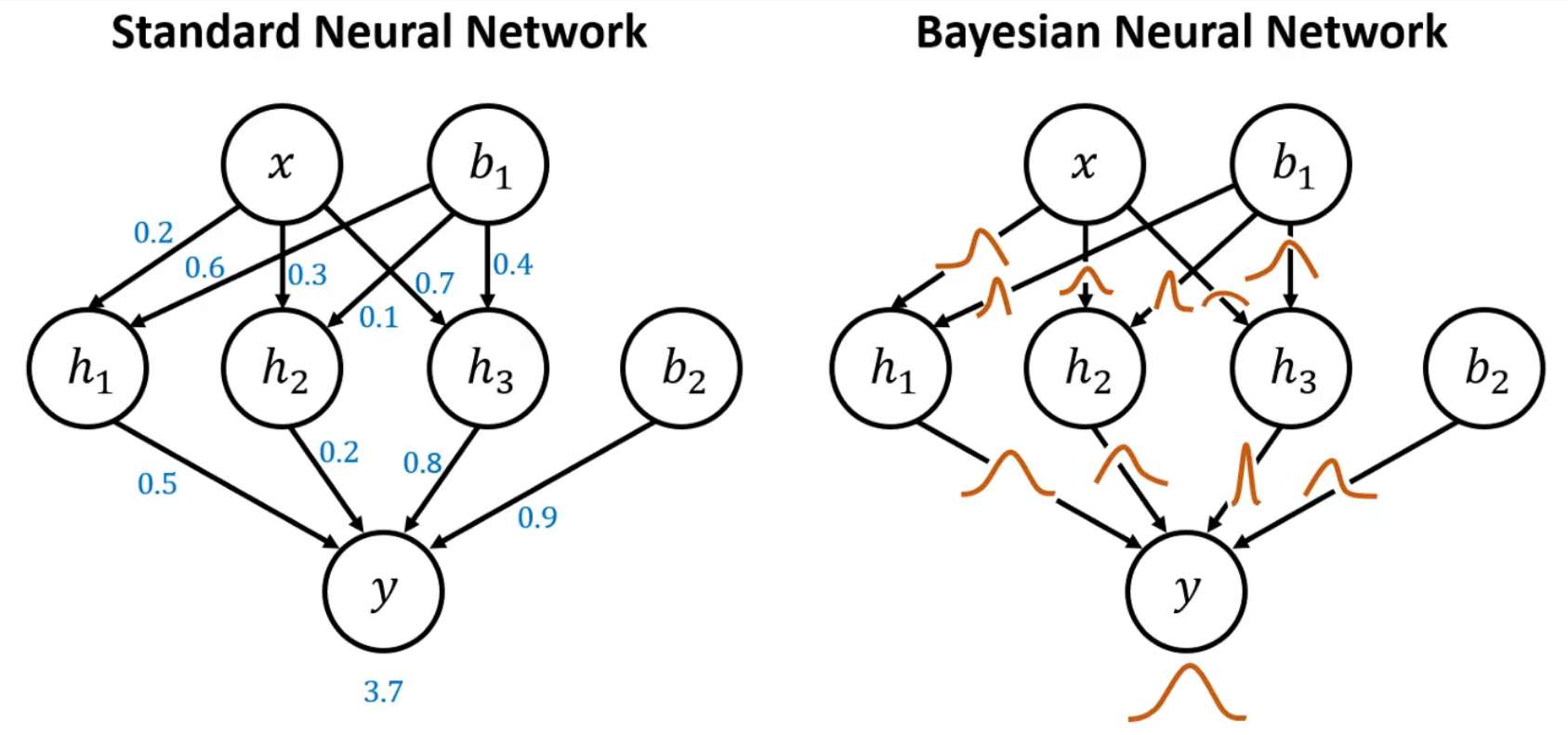
Do we need uncertainty propagation?

- What is the representation of uncertainty in each step?

How large is the data to parameter ratio?

Bayes for Deep Learning

Bayesian neural networks (BNNs)



How to avoid sampling from BNNs being super wasteful?

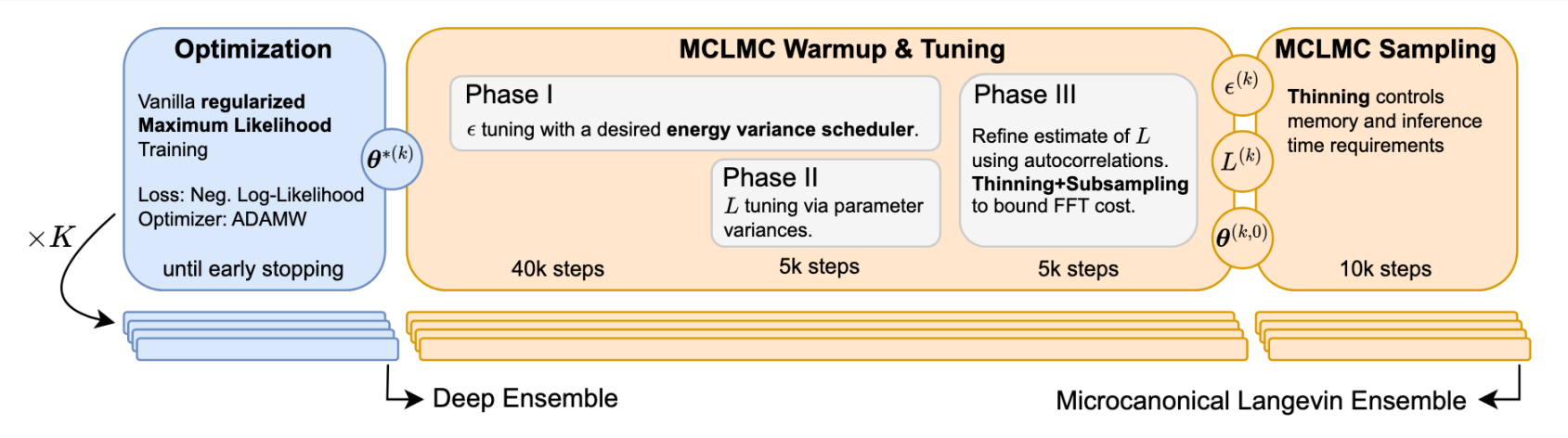
Are BNNs needed at all? **Alternatives**: Ensembles, conformal prediction

Sampling from Bayesian neural networks

Option 1: Sampling in a small weight subspace

Further reading: Izmailov et al. (2020; arxiv.org/abs/1907.07504)

Option 2: Fast sampling in the full weight space



Further reading: Sommer et al. (2025; arxiv.org/abs/2502.06335)

Semi-structured models

How to include unstructured data (e.g., images or texts) into structured additive models?

We predict the response mean as

$$\mathbb{E}(y \mid x, z) = \eta_{str}(x) + \eta_{deep}(z)$$

The structured part $\eta_{str}(x)$ is standard additive model with parameters β of direct interest learned from structured data x: $\eta_{str}(x) = \beta x$

The deep learning part $\eta_{deep}(z)$ has neural network parameters ω learned from unstructured data z: $\eta_{deep}(z) = \text{DNN}_{\omega}(z)$

Further reading: Dold et al. (2024; arxiv.org/abs/2401.12950)

Bayes for ML Theory and Algorithms

Everything that works, works because it's Bayesian

Examples from statistics and machine learning:

- Maximum likelihood estimates are reasonable if the posterior mode is a good point estimator of the posterior
- Regularization terms in machine learning often directly correspond to well known priors in Bayesian inference
- Information criteria approximate Bayesian model comparison
- Dropout in DNNs can be seen as approximate Bayesian inference (Gal & Ghahramani, 2016; arxiv.org/abs/1506.02142)
- Many well-known learning algorithms can be derived from a single Bayes learning rule (Khan & Rue, 2023; arxiv.org/abs/2107.04562)
- Knowledge adaptation as posterior correction (Khan, 2025; arxiv.org/abs/2506.14262)

$\Rightarrow$: Exact Bayesian inference is the ideal to be approximated

Optimization algorithms following Bayesian principles

The Improved Variational Online Newton (IVON) algorithm:

RMSprop/Adam

$\hat{g} \leftarrow \hat{\nabla}\ell(\theta)$

$\hat{h} \leftarrow \hat{g}^2$

$h \leftarrow (1-\rho)h + \rho\hat{h}$

$\theta \leftarrow \theta - \alpha(\hat{g} + \delta m)/(\sqrt{h} + \delta)$

Improved Variational Online Newton

$\hat{g} \leftarrow \hat{\nabla}\ell(\theta)$ where $\theta \sim \mathcal{N}(m, \sigma^2)$

$\hat{h} \leftarrow \hat{g} \cdot (\theta - m)/\sigma^2$

$h \leftarrow (1-\rho)h + \rho\hat{h}$ $+\rho^2(h - \hat{h})^2/(2(h + \delta))$

$m \leftarrow m - \alpha(\hat{g} + \delta m)/(h + \delta)$

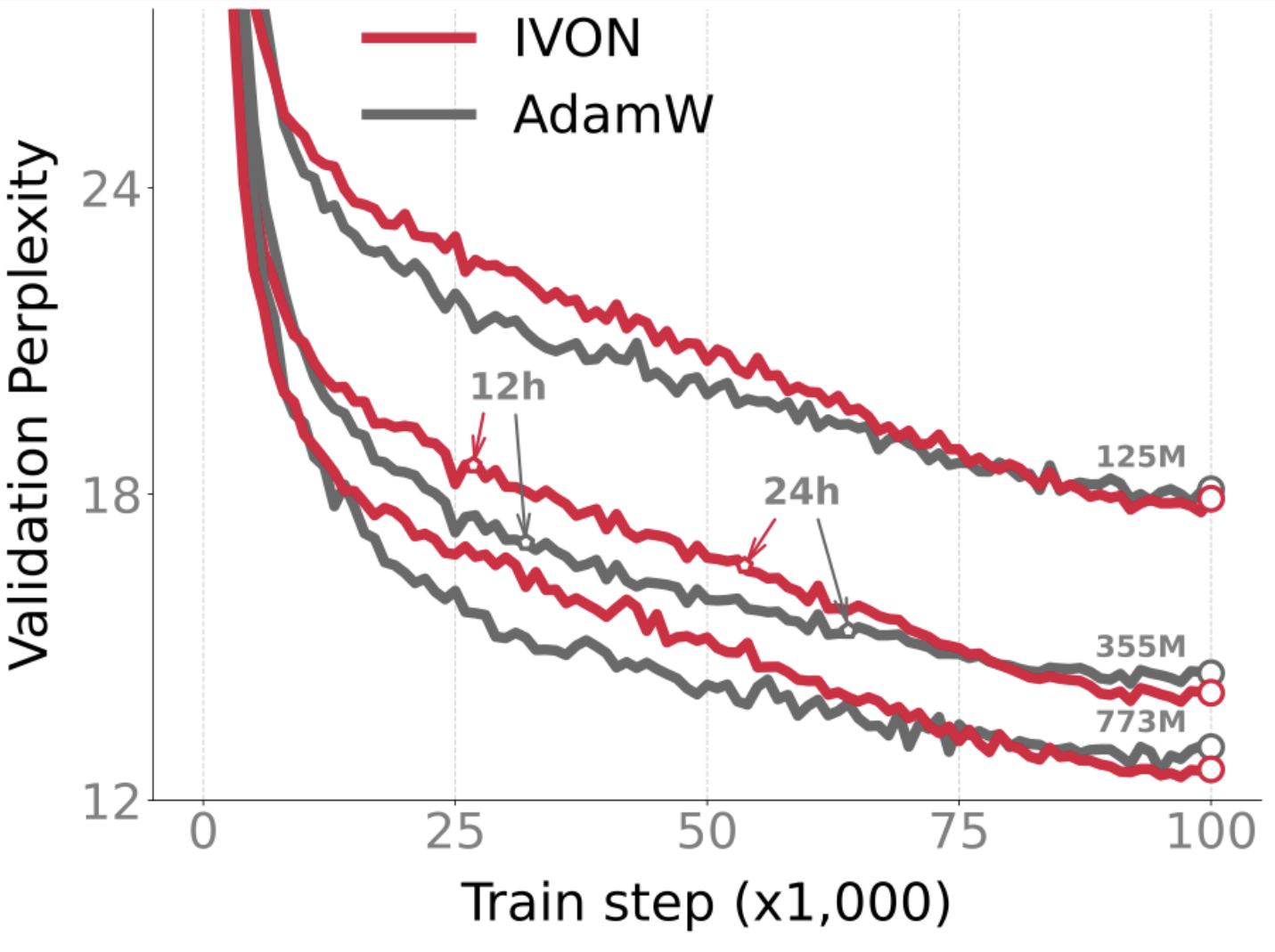
$\sigma^2 \leftarrow 1/(N(h + \delta))$

Further reading: Shen et al. (2024; arxiv.org/abs/2402.17641)

Blog post: https://adaptive-bayesian.ai/blog/ivon

Optimization algorithms following Bayesian principles

Results for GPT-2 Training



Deep Learning for Bayes

Neural Posterior Estimation (NPE)

Learn a neural approximation $q_\psi(\theta \mid y)$ for $p(\theta \mid y)$

Maximum likelihood loss for normalizing flow q_ψ:

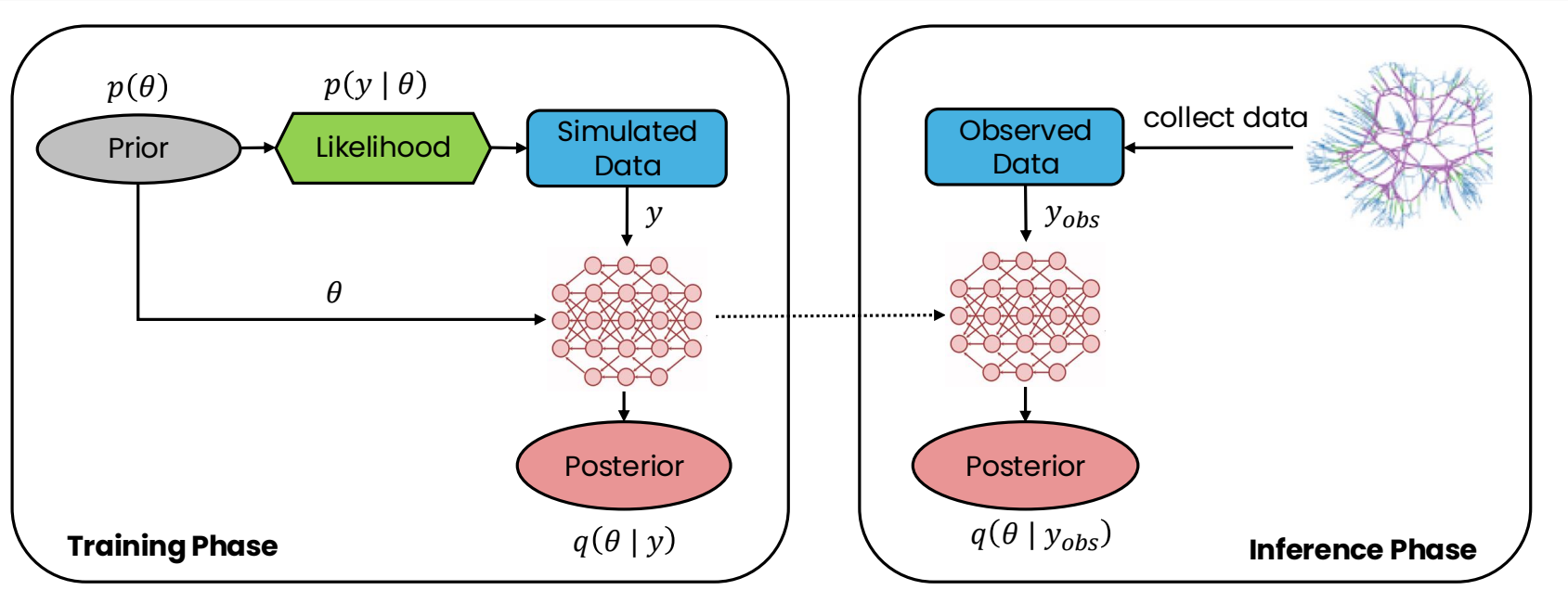
$$\begin{aligned}\widehat{\psi} &= \underset{\psi}{\operatorname{argmin}}\, \mathbb{E}_{y \sim p(y)} \left[\mathbb{KL} \left(p(\theta \mid y) \,||\, q_\psi(\theta \mid y) \right) \right] \\ &= \underset{\psi}{\operatorname{argmin}}\, \mathbb{E}_{(\theta, y) \sim p(\theta, y)} \left[-\log q_\psi(\theta \mid y) \right]\end{aligned}$$

Approximate the loss via S samples $(\theta^{(s)}, y^{(s)}) \sim p(\theta, y)$:

$$\widehat{\psi} = \underset{\psi}{\operatorname{argmin}}\, \frac{1}{S} \sum_{s=1}^{S} \left[-\log q_\psi(\theta^{(s)} \mid y^{(s)}) \right]$$

Simulation-based inference (SBI) can be addressed with NPE

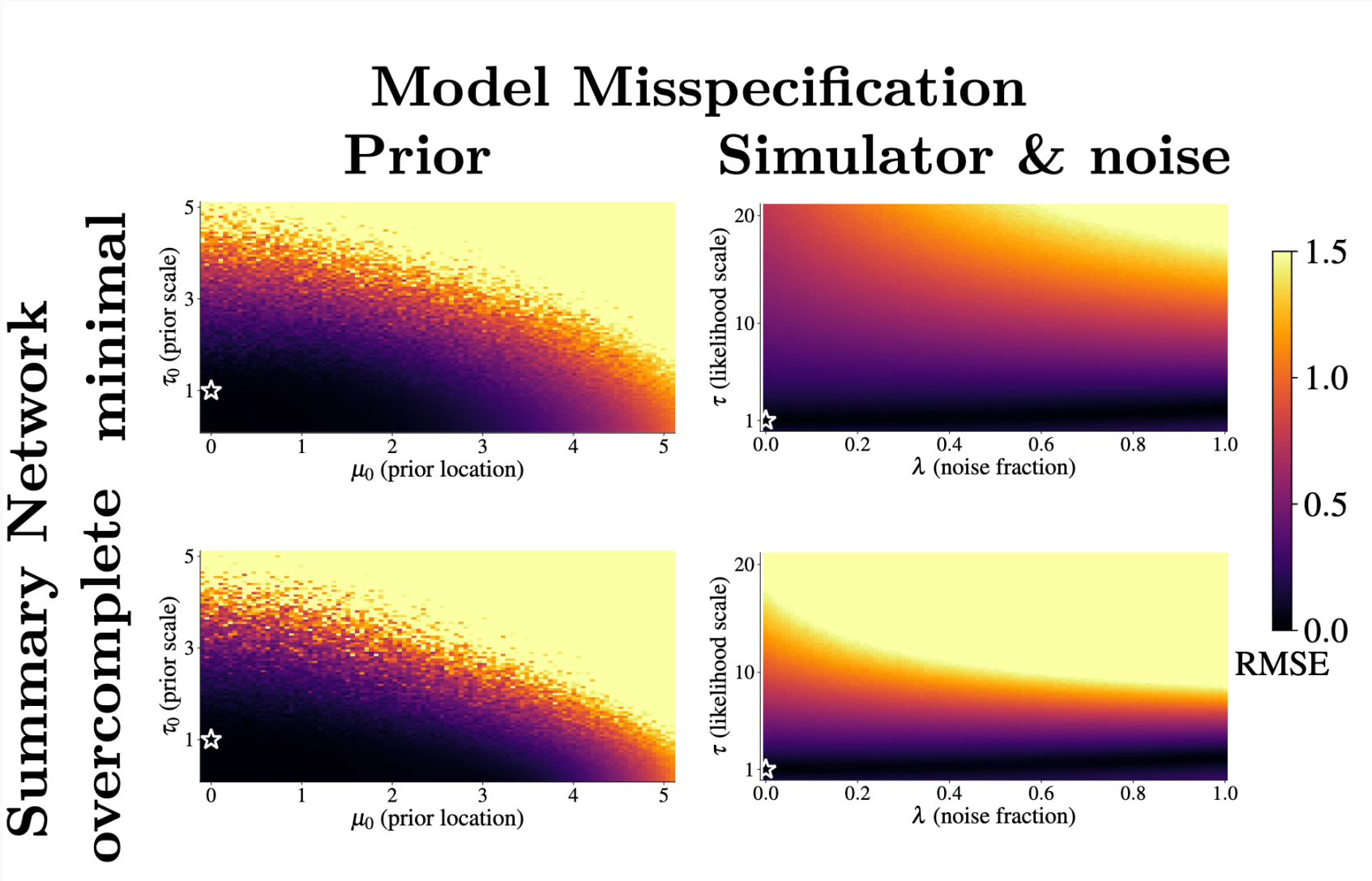
Amortized Bayesian Inference (ABI)



$\Rightarrow$ ABI enables Bayesian inference in real time

Further reading: Kühmichel et al. (2026; arxiv.org/abs/2602.07098)

Standard NPE on misspecified models fails



Further reading: Schmitt et al. (2024; arxiv.org/abs/2406.03154)

Bayesian Self-consistency

For any set of parameter values $\theta^{(1)}, \dots, \theta^{(L)}$, the following holds:

$$p(y) = \frac{p(y \mid \theta^{(1)})\, p(\theta^{(1)})}{p(\theta^{(1)} \mid y)} = \dots = \frac{p(y \mid \theta^{(L)})\, p(\theta^{(L)})}{p(\theta^{(L)} \mid y)}.$$

This implies that the variance of the log-ratios must be zero:

$$\mathrm{Var}_{l=1}^{L} \left[\log \left(\frac{p(y \mid \theta^{(l)})\, p(\theta^{(l)})}{p(\theta^{(l)} \mid y)} \right) \right] = 0$$

Further reading: Schmitt et al. (2024; arxiv.org/abs/2310.04395)

Bayesian self-consistency loss

Replace the posterior $p(\theta \mid x)$ with its neural approximator $q_\psi(\theta \mid x)$.

For any (**unlabled**) dataset y^* and any distribution $\tilde{p}(\theta)$, we define:

$$\text{SCLoss}(\psi) = \text{Var}_{\theta \sim \tilde{p}(\theta)} \left[\log p(y^* \mid \theta) + \log p(\theta) - \log q_\psi(\theta \mid y^*) \right]$$

$\Rightarrow$ We can use **real-world data** as y^* to train the SC loss

The SC-Loss alone doesn't work well most of the time so we combine it with the standard simulation-based (SB) loss:

$$\text{SemiSupervisedLoss}(\psi) = \text{SBLoss}(\psi) + \lambda \cdot \text{SCLoss}(\psi).$$

This approach also works of other likelihood-based losses (e.g., reverse KL)

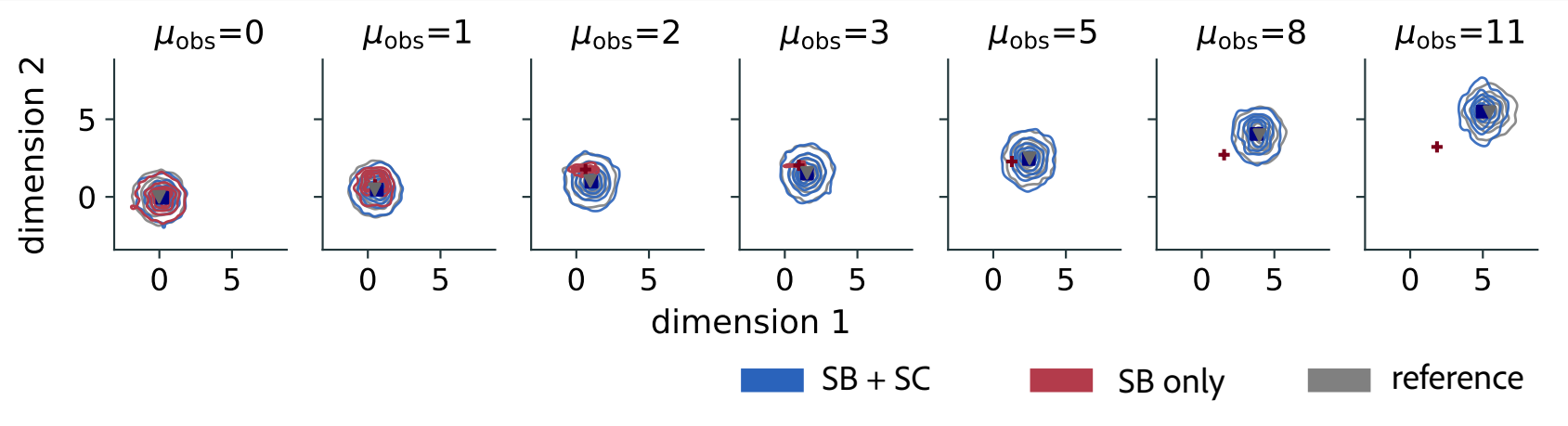
Further reading: Mishra et al. (2026; arxiv.org/abs/2602.22884)

Case Study: SC for multivariate normal model

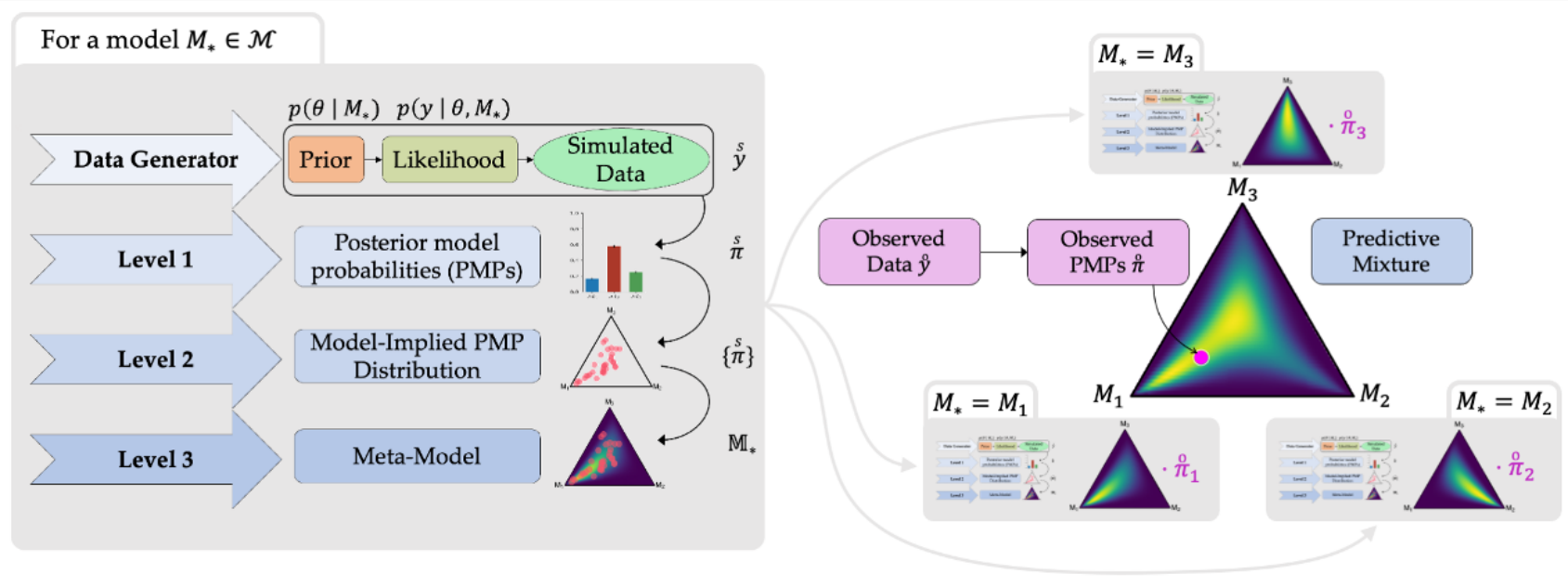
$$\theta \sim \text{Normal}(\mu_{\text{prior}}, I_D), \qquad x \sim \text{Normal}(\theta, I_D)$$

- For the SB loss, we simulate from the model with $\mu_{\text{prior}} = 0$
- For the SC loss, we simulate **few unalabled datasets** from the model with $\mu_{\text{prior}} = 2$

Illustrative results:

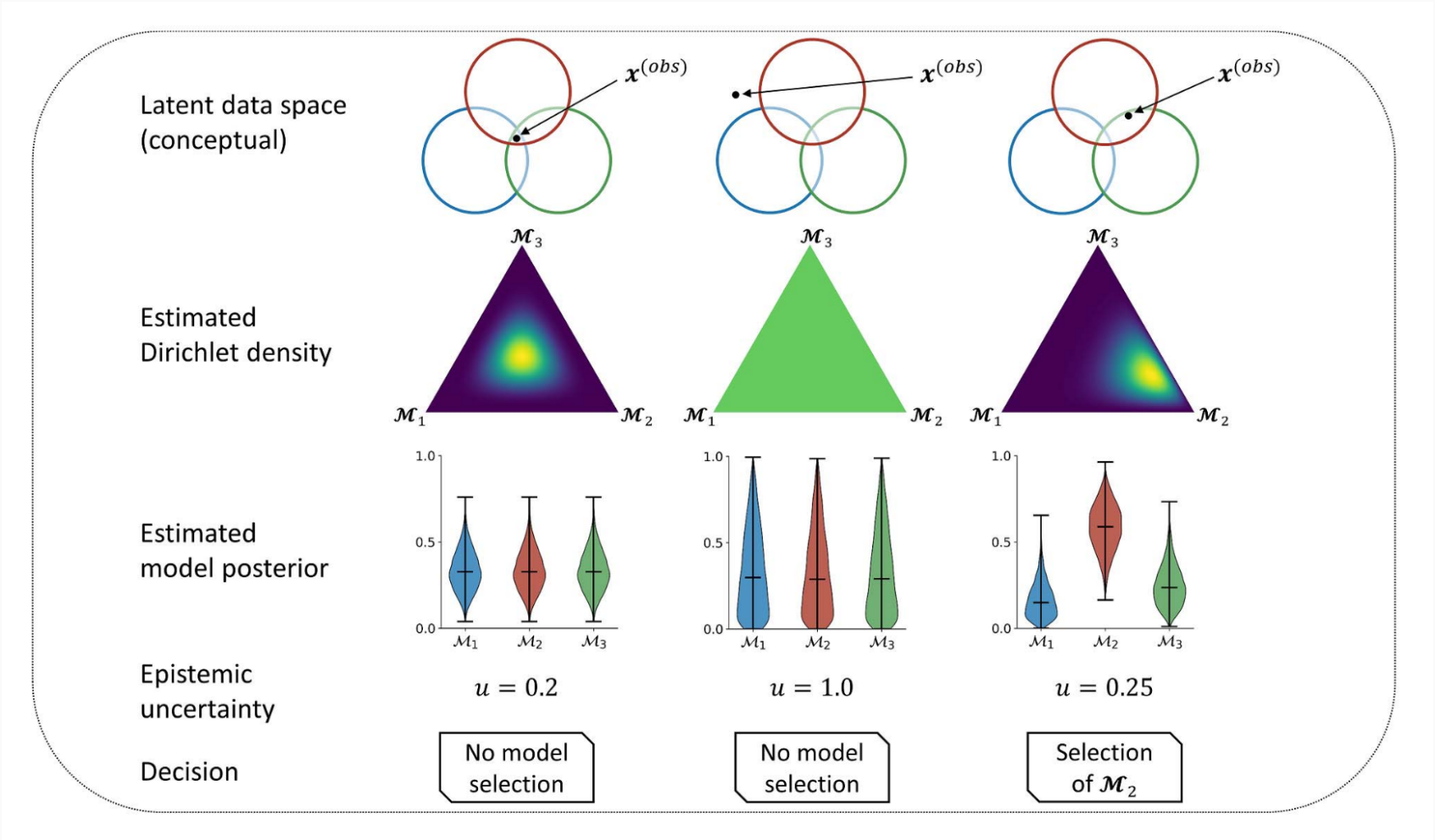


Meta-uncertainty for Bayesian model comparison



Further reading: Schmitt et al. (2023; http://arxiv.org/abs/2210.07278)

Bayesian model comparison via evidential deep learning



Further reading: Radev et al. (2021; https://arxiv.org/abs/2004.10629)

Collaboration Opportunities

Applications:

- Standard Bayesian inference
- Amortized Bayesian inference

Methods:

- Diagnostics for probability inference accuracy
- Efficient uncertainty propagation
- Model comparison
- Amortized Bayesian inference
- Neural network architectures

Learn More

Contact details:

- Website: https://paulbuerkner.com/
- Email: paul.buerkner@gmail.com
- Bluesky: @paulbuerkner.com

Software:

- **Stan**: Probabilistic programming language (mc-stan.org)
- **brms** : High-level interface to Stan (paulbuerkner.com/brms)
- **BayesFlow**: Python library for ABI (bayesflow.org)

Appendix