Meta-Learning MCMC Proposals

Wang, Wu, Moore, & Russell (2019)

Presented by Alexandra Proca, Daniel May, and Anthony Chiu

Motivation

- Probabilistic inference is a useful tool in machine learning, but can often be difficult to perform
- Single-site Gibbs sampling: slow convergence with several variables coupled in the posterior
- Block proposals update multiple variables simultaneously, but can become intractable → manually-designed proposals → not generalizable to new tasks/models

Ultimate Aim

Learn to automatically build tractable MCMC proposals that are

1. Effective for fast mixing
2. Ready to be reused across different models

➤ Meta learning

$$q(B_i; c_i, \Psi_i) \approx p_{\Psi_i}(B_i | C_i = c_i)$$

Gibbs Sampling

- Single site Gibbs full conditional: $p(X_d | X_1, \ldots, X_{d-1}, X_{d+1}, \ldots, X_D)$
- Block Gibbs conditional: $p(X_d, \ldots, X_{d+k} | X_1, \ldots, X_{d-1}, X_{d+k+1}, \ldots, X_D)$
- Gibbs samples from full conditionals → generic way to derive proposal distribution:

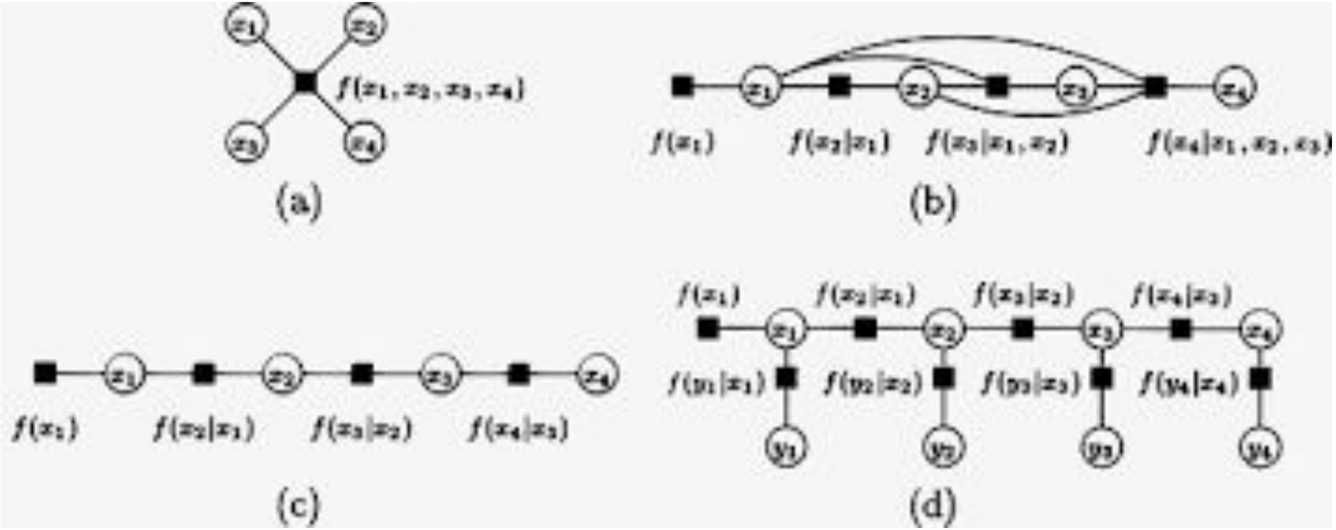
$$\begin{aligned}
X_{i+1,1} &\sim p(\,.\,|X_{i,2}, \ldots, X_{i,\mathrm{D}}) \\
&\vdots \\
X_{i+1,d} &\sim p(\,.\,|X_{i+1,1}, \ldots, X_{i+1,d-1}, X_{i,d+1}, \ldots, X_{i,\mathrm{D}}) \\
&\vdots \\
X_{i+1,\mathrm{D}} &\sim p(\,.\,|X_{i+1,1}, \ldots, X_{i+1,\mathrm{D}-1})
\end{aligned}$$

Idea

- Train a neural network to approximate Gibbs proposals for recurring structural motifs in probabilistic graphical models and to speed up inference on new models without extra tuning
- Take the model parameters as input to the network (model parameters are not fixed)

Factor Graphs

- The work focuses on directed models where factors specify conditional probabilities of each variable given its parents



Motifs

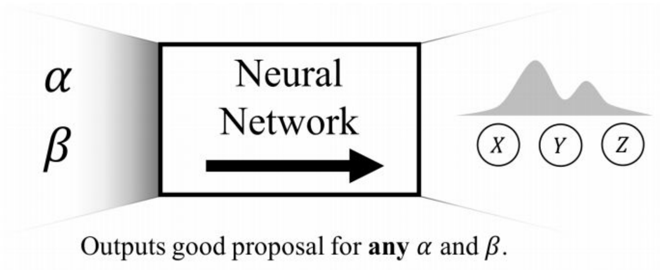
- Graph with nodes partitioned into sets B (block proposed set to be resampled) and C (conditioning set) with a parameterized joint distribution $p(B, C)$ consistent with graph structure. This specifies the conditional $p(B|C)$
- Given a set of evidence variables C, inference attempts to sample from the conditional distribution on the remaining variables B

Proposal Networks for Motifs

- Identify commonly recurring structures in graphical model as motifs
- An instantiation (B_i, C_i, ψ_i) of a motif includes
  - A subset of the model variables (B_i, C_i) such that the subgraph is isomorphic to the motif (B, C)
  - A subset of model parameters $\psi_i \in \psi$ required to specify the conditional distribution $p_{\psi_i}(B|C)$
- Learn a proposal network associated with each motif, which determine the shape of the network input and output
- Choosing a motif represents a trade-off between generality of the proposal and easiness to approximate
  - Recommend simple structures such as chains of a certain length

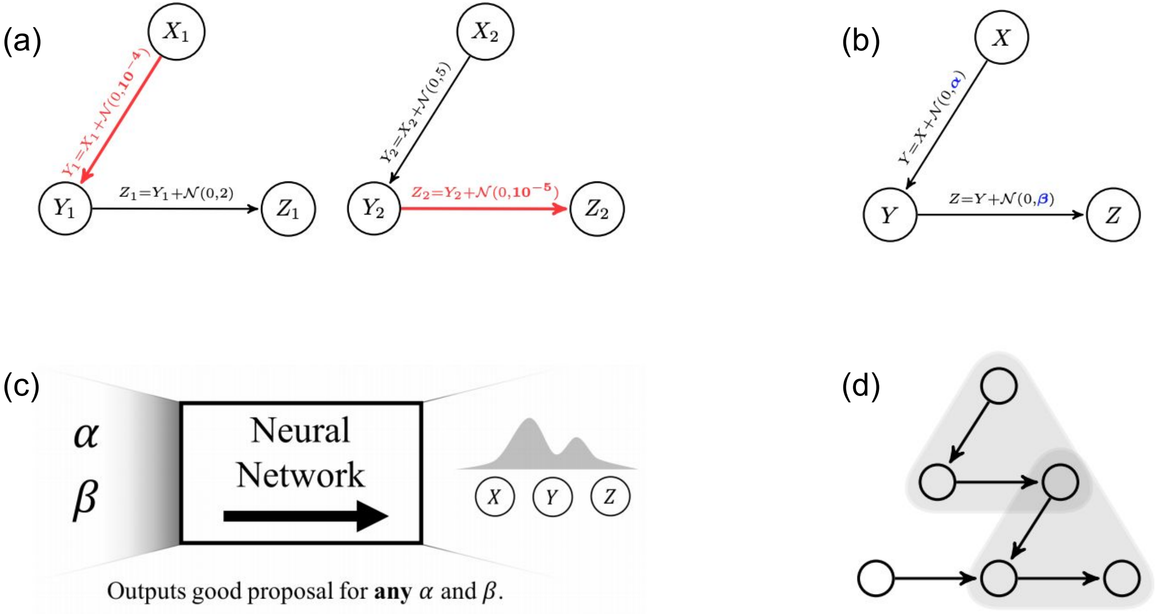
Parameterizing the Proposal Networks

- Parameterize proposal networks with mixture density networks (MDNs)
  - Given conditioning set values and local model parameters organized as an input vector, these output parameters for a mixture distribution over the block variables



- Aim: optimize the network weights θ so the function it represents is close to the true conditional

MCMC Proposals on Motifs in Graphical Models



Meta-training Proposal Networks

- Minimise the distance of proposal and true conditional in the sense of KL divergence:

$$D(p_{\Psi_i}(B_i|C_i)||q_\theta(B_i; c_i, \Psi_i))$$

- In practice we want to minimise the expected divergence over all possible values of the conditioning set:

$$\mathbb{E}_{C_i}[D(p_{\Psi_i}(B_i|C_i)||q_\theta(B_i; c_i, \Psi_i))] = -\mathbb{E}_{B_i, C_i}[\log q_\theta(B_i; C_i, \Psi_i)] + \text{constant}$$

Meta-training Proposal Networks

- Second term is constant, so can define loss function as:

$$\tilde{L}(\theta; B_i, C_i, \Psi_i) = -\mathbb{E}_{B_i, C_i}[\log q_\theta(B_i; C_i, \Psi_i)]$$

- Goal: minimize loss over many random instantiations of motifs in $\mathcal{P}$:

$$L(\theta) = \mathbb{E}_{(B_i, C_i, \Psi_i) \sim \mathcal{P}}[\tilde{L}(\theta; B_i, C_i, \Psi_i)] = -\mathbb{E}_{(B_i, C_i, \Psi_i) \sim \mathcal{P}}[\mathbb{E}_{B_i, C_i}[\log q_\theta(B_i; C_i, \Psi_i)]]$$

- This is minimized using mini-batch stochastic gradient descent

Algorithm

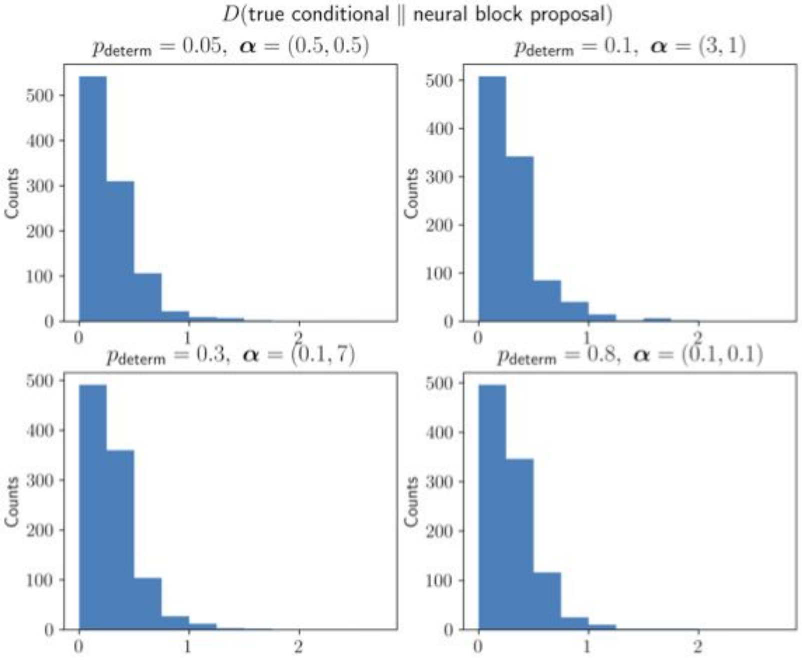
Algorithm 1 Neural Block Sampling

Input: Graphical model (G, Ψ), observations y,
motifs $\{(B^{(m)}, C^{(m)})\}_m$, and their instantiations $\{(B_i^{(m)}, C_i^{(m)}, \Psi_i^{(m)})\}_{i,m}$ detected in (G, Ψ).

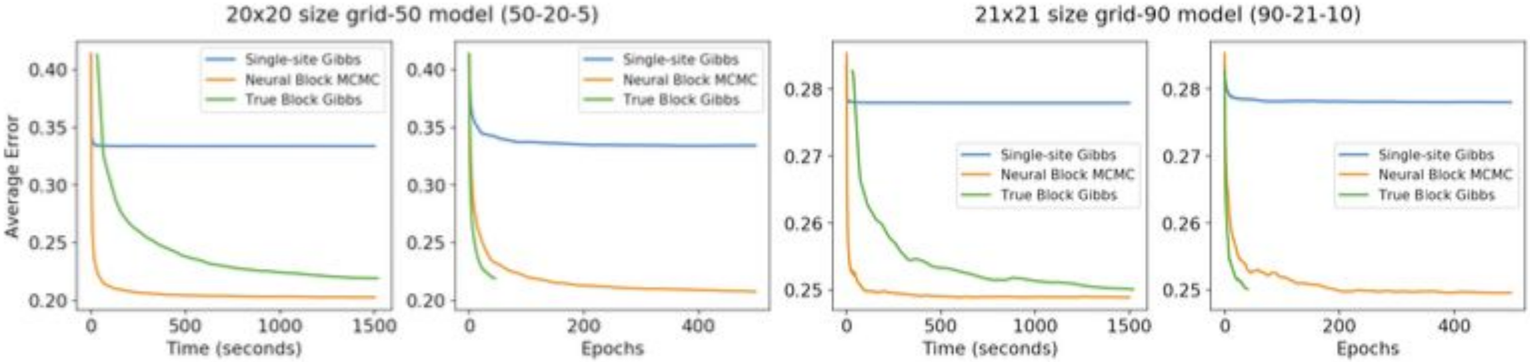
for each motif $B^{(m)}, C^{(m)}$ **do**
  if proposal trained for this motif exists **then**
    $q^{(m)} \longleftarrow$ trained neural block proposal
  else
    Train neural block proposal $q_\theta^{(m)}$ using SGD by Eq. 3 on its instantiations $\{(B_i^{(m)}, C_i^{(m)}, \Psi_i^{(m)})\}_i$
  end if
end for
$x \longleftarrow$ initialize state
for timestep **in** $1 \ldots T$ **do**
  Propose $x' \leftarrow$ proposal $q_\theta^{(m)}$ on some instantiation $(B_i^{(m)}, C_i^{(m)}, \Psi_i^{(m)})$
  Accept or reject according to MH rule
end for
return MCMC samples

How good the proposal approximations are

- Quantify as KL divergence
- KL divergences between neural block proposals and true conditionals are plotted on histogram
- Only trained on one set p_{determ} and α
- Results:
  1. KL divergences values mostly concentrated at zero
  2. Generalise well to other p_{determ} and α

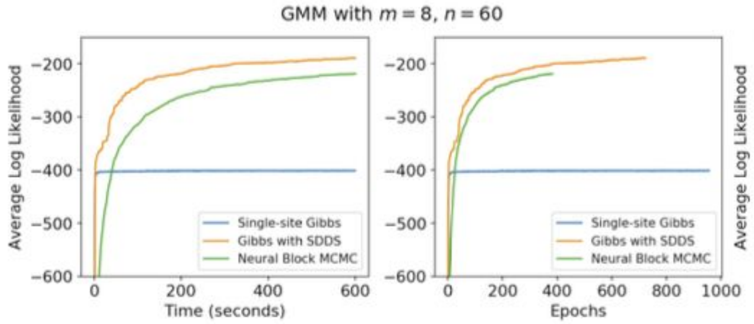


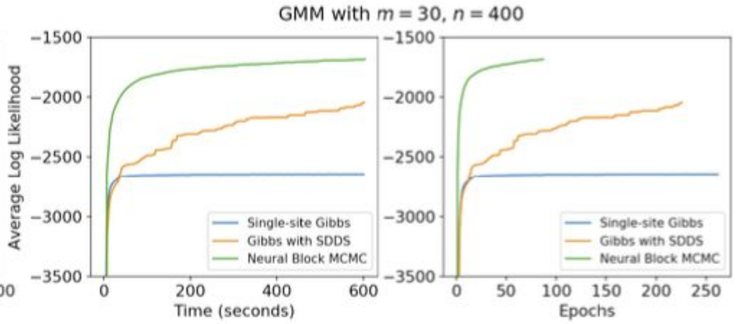
Neural block proposal convergence speed



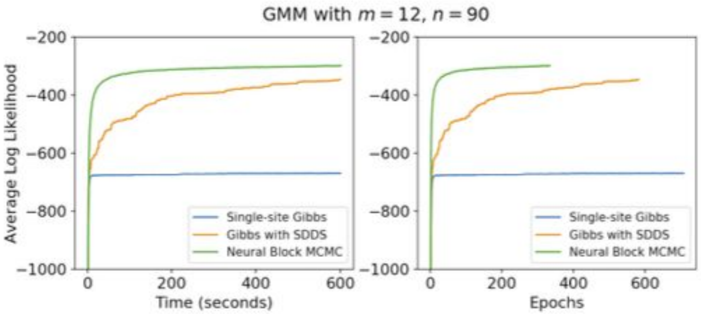
- MCMC inference using neural block proposal always outperforms single site gibbs and true block gibbs proposal given fixed computation time

Neural block proposal convergence speed

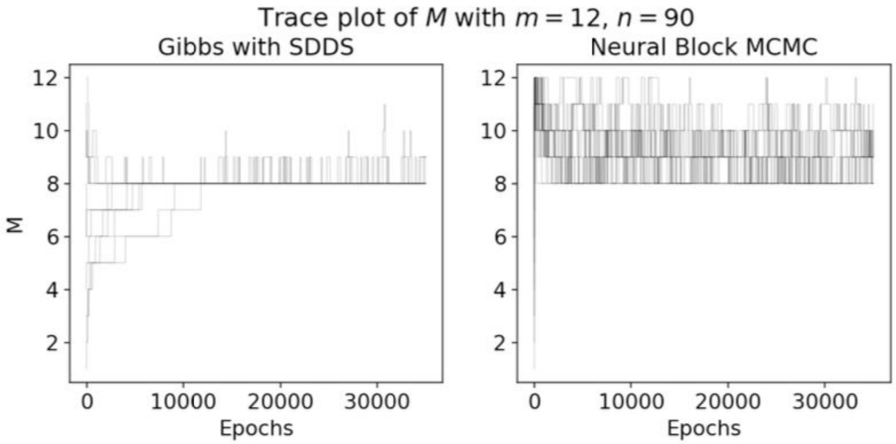




- Compare with the single-site Gibbs proposal and Gibbs with SDDS [3].
- Neural block proposal mixes quickly and outperforms the Gibbs SDDS method at large m and n

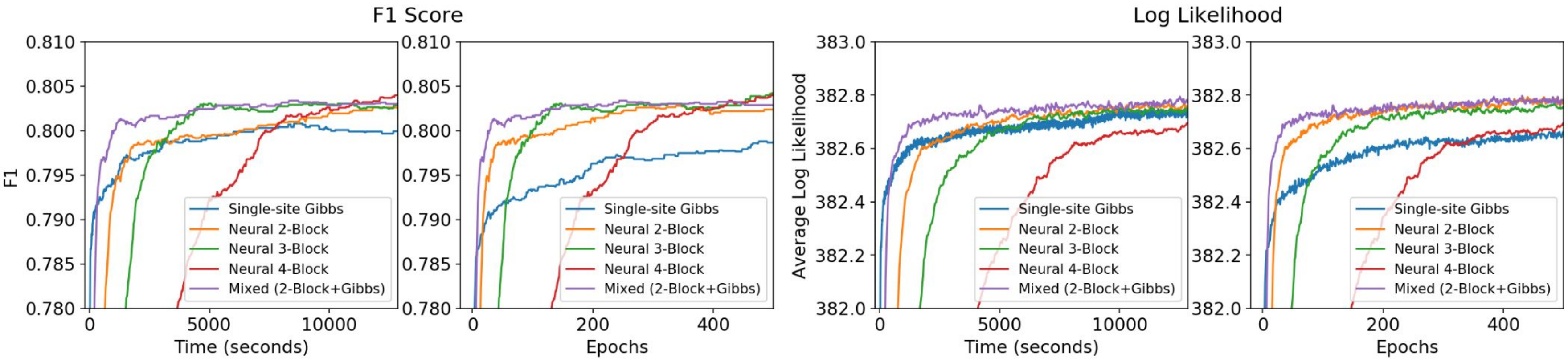


Mixing quality



- Gibbs with SDDS fails to explore efficiently
- Neural block MCMC mixes quickly among possible explanations of M

Neural block proposal convergence speed



- In general, neural block proposal achieves better performance than single site Gibbs in terms of F1 score and log likelihood

Impact and Follow-up Work

- Neural Relational Inference with Fast Modular Meta-Learning [4]
  - ➢ Relational inference can be framed as a modular meta-learning problem, and meta-learning of proposal functions could speed up the simulated annealing search within the modular meta-learning algorithm.
- Using Probabilistic Programs as Proposals [5]
  - ➢ Meta learning of proposal is not flexible enough to allow user to specify knowledge into the proposal. They instead use a probabilistic program.
- Deep Involutive Generative Models for Neural MCMC [6]
  - ➢ Establishes an alternative approach to learn neural network MCMC proposals, which are both fast and accurate

References

[1] Tongzhou Wang, Yi Wu, David A. Moore, Stuart J. Russell. *Meta-Learning MCMC Proposals*. In *NeurIPS*, 2018.

[2] Christopher M. Bishop. *Mixture Density Networks*, 1994.

[3] Wei Wang and Stuart J Russell. *A smart-dumb/dumb-smart algorithm for efficient split-merge MCMC*. In *UAI*, pages 902–911, 2015.

[4] Ferran Alet, Erica Weng, Tomas Lozano-Perez, and L. Kaelbling. *Neural relational inference with fast modular meta-learning*. In *NeurIPS*, 2019.

[5] Marco F. Cusumano-Towner and Vikash K. Mansinghka. *Using probabilistic programs as proposals*. *arXiv*, 2018.

[6] Span Spanbauer, Cameron Freer, and Vikash Mansinghka. *Deep involutive generative models for neural MCMC*. arXiv preprint *arXiv*:2006.15167, 2020.

[7] David Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[8] Andrew Gelman et al. *Bayesian data analysis*. CRC press, 2013.