Neural Block Sampling

Tongzhou Wang, Yi Wu, Dave Moore, Stuart Russell

UC Berkeley

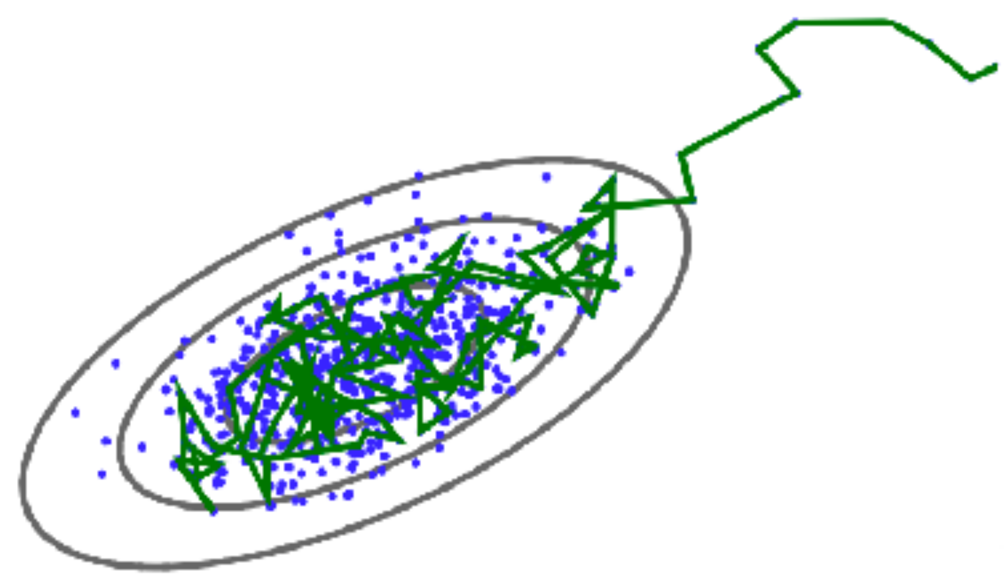
BAIR
BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH

Probabilistic Programming

- Model-based probabilistic inference
- Formal representation of models as sample-generating programs, e.g.,
    - non-differentiable models with discrete variables
    - open-universe models
    - context-specific dependencies
- Need black-box inference algorithms

Markov Chain Monte Carlo

- Sample from target distribution $p(x)$:
  - Construct a Markov chain with stationary distribution $p(x)$
  - Random walk



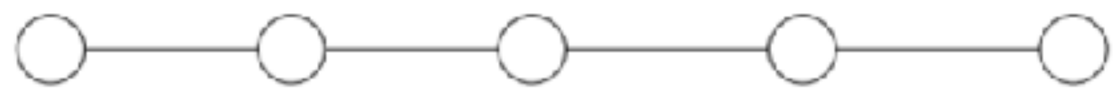
- Choice of MC has large impact on performance

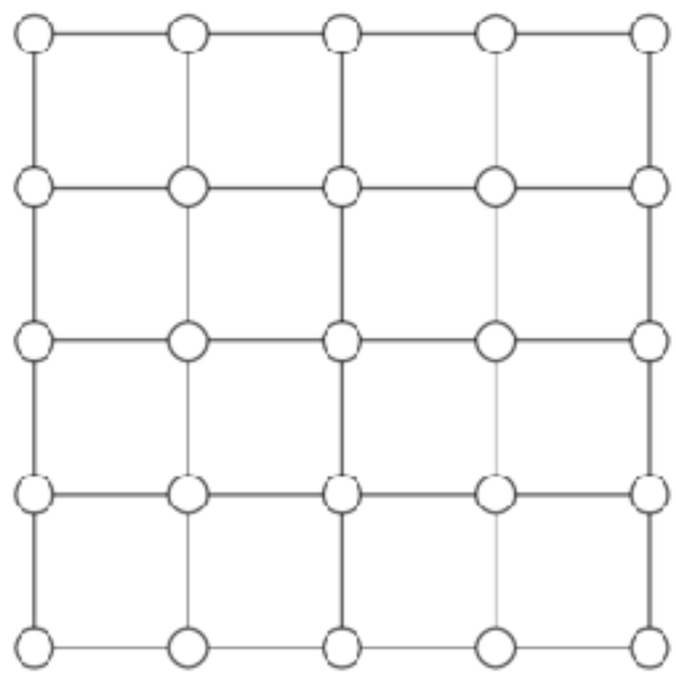
MCMC Proposal

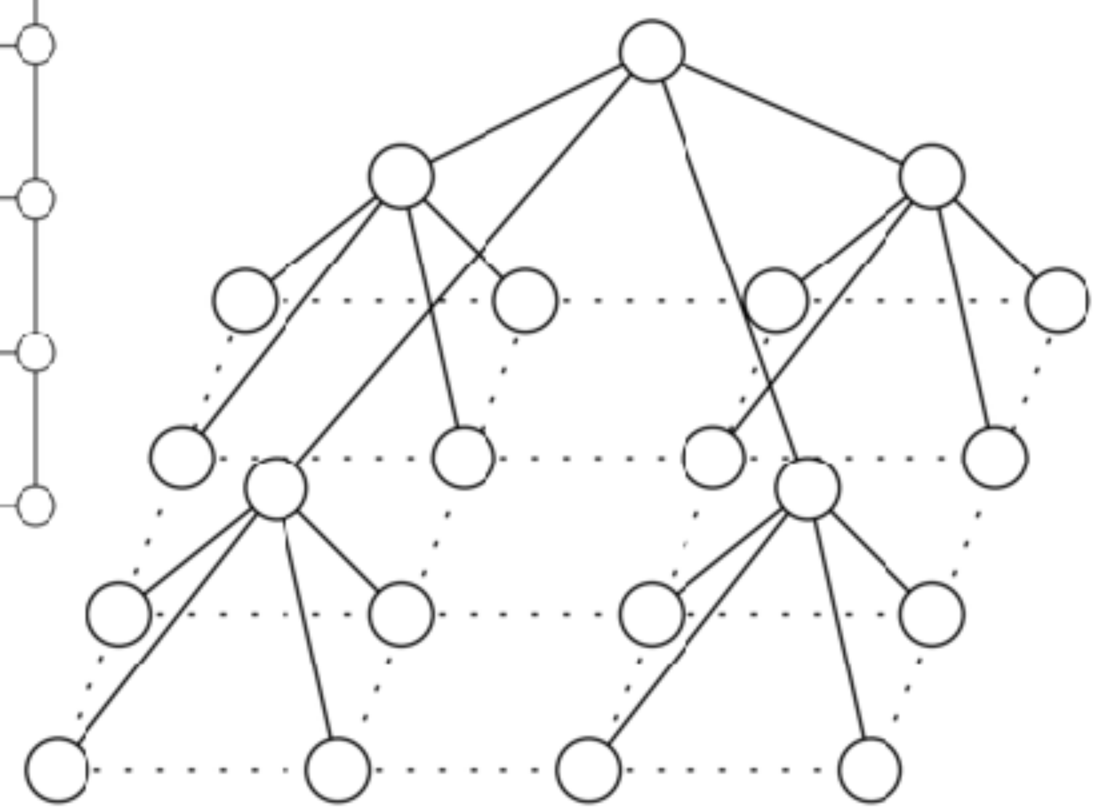
- Generic single variable proposal:
  - slow mixing in models with tight coupling
  - stuck in local optima
- Block proposal:
  - computational cost
  - often hand-engineered
- Given more structural information, construct good and general block proposals?

Structural Motif

- Chains
- Grids
- Trees
- Rings
- etc.



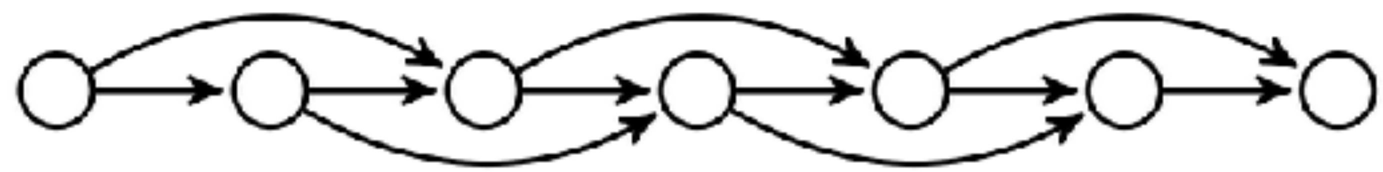




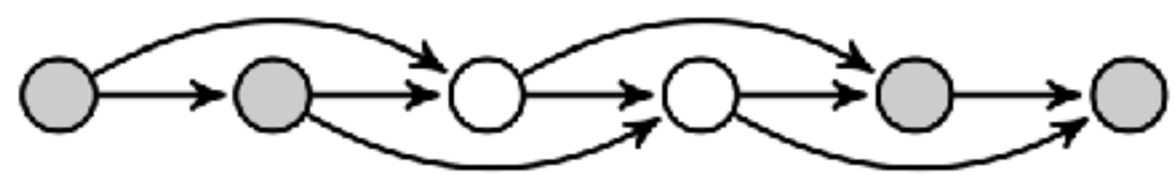
Motif examples borrowed from slides by Martin Wainwright

Proposals for Structural Motifs

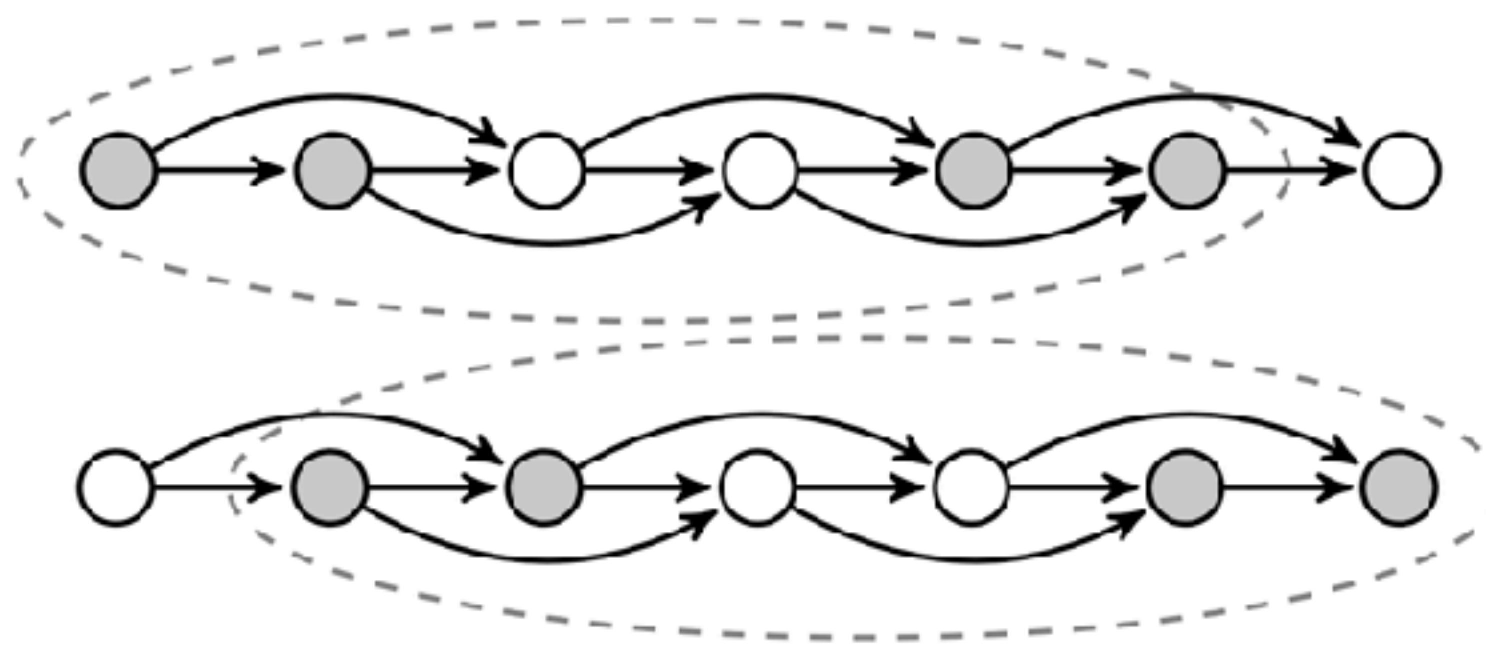
- Model:



- Motif:



- Instantiations:



Neural Block Proposals

- Focus on a certain motif of interest
- Use neural network to approximate Gibbs block proposal:

$$\text{local params} \cup \text{Markov blanket values} \longrightarrow \text{block proposal distribution}$$

- Proposal parametrized as a mixture distribution [Bishop 94]
- Train using prior samples to minimize expected KL divergence
  - Unlimited training data by sampling from model
  - Target equivalent to maximize log likelihood of samples
- No assumption on variables being discrete/continuous

Neural Block MCMC

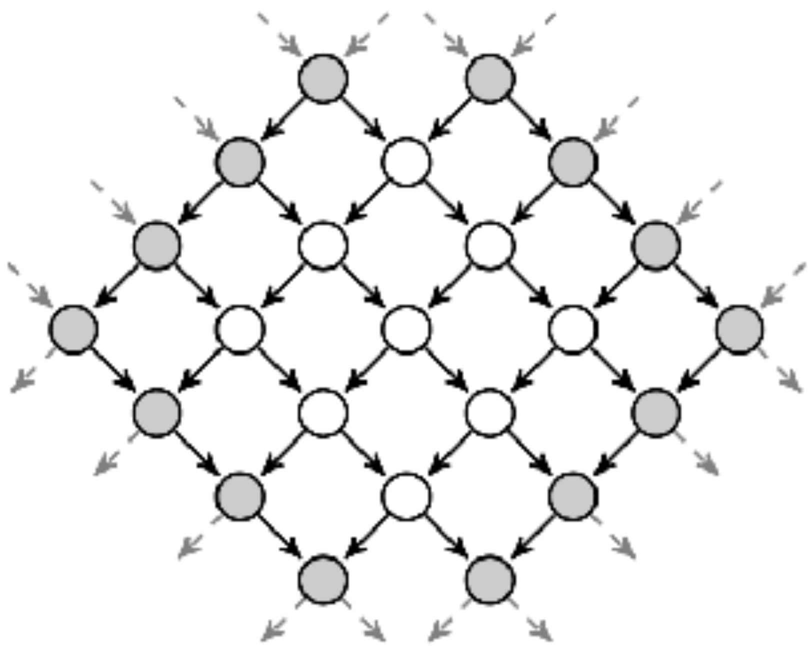
- Given:
  - A inference task on a graphical model
  - Hand-identified motifs and their instantiations in the model

1. Train/retrieve neural block proposals for these motifs
2. Run MCMC updates:

   a. Propose using neural block proposals (if possible)

   b. Accept/reject with MH rules

General Binary Grid BNs



$$\begin{cases} [0,1] & \text{w.p. } \frac{1}{40} \\ [1,0] & \text{w.p. } \frac{1}{40} \\ \text{Dirichlet}([0.5, 0.5]) & \text{w.p. } \frac{19}{20} \end{cases}$$

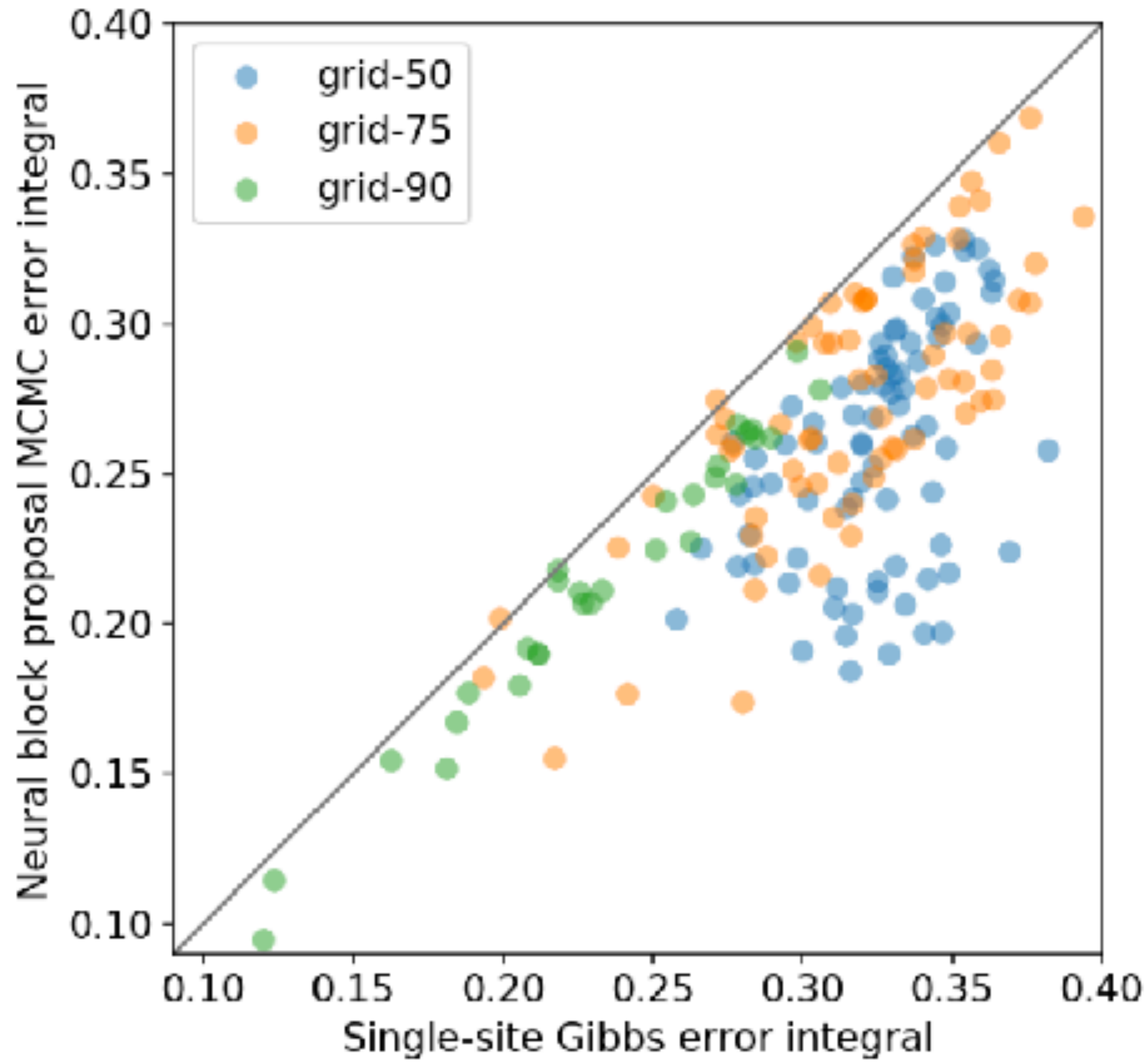
Random CPT entry

Structural Motif

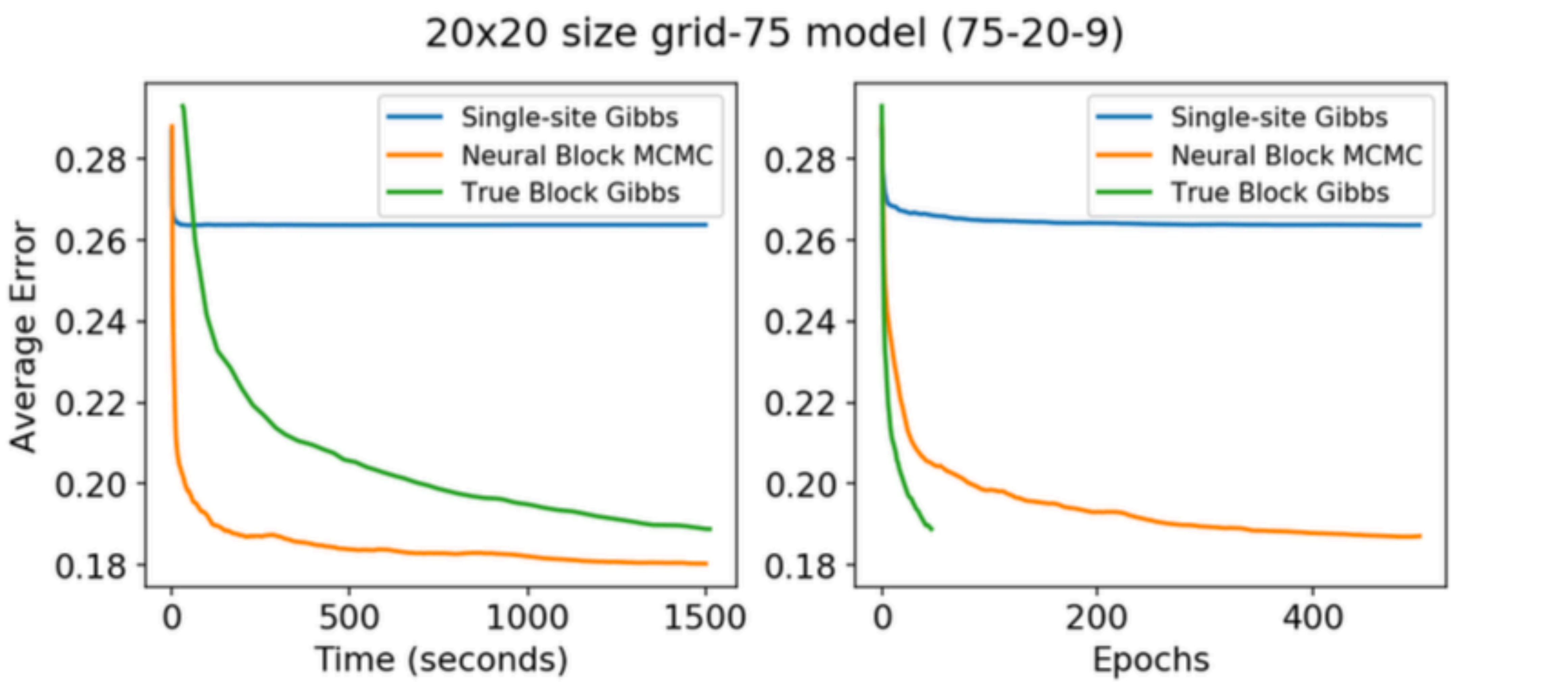
Training: Random BN

General Binary Grid BNs

- Test on 180 grid models from UAI 08 competition
- grid-k: k% deterministic dependency



General Binary Grid BNs

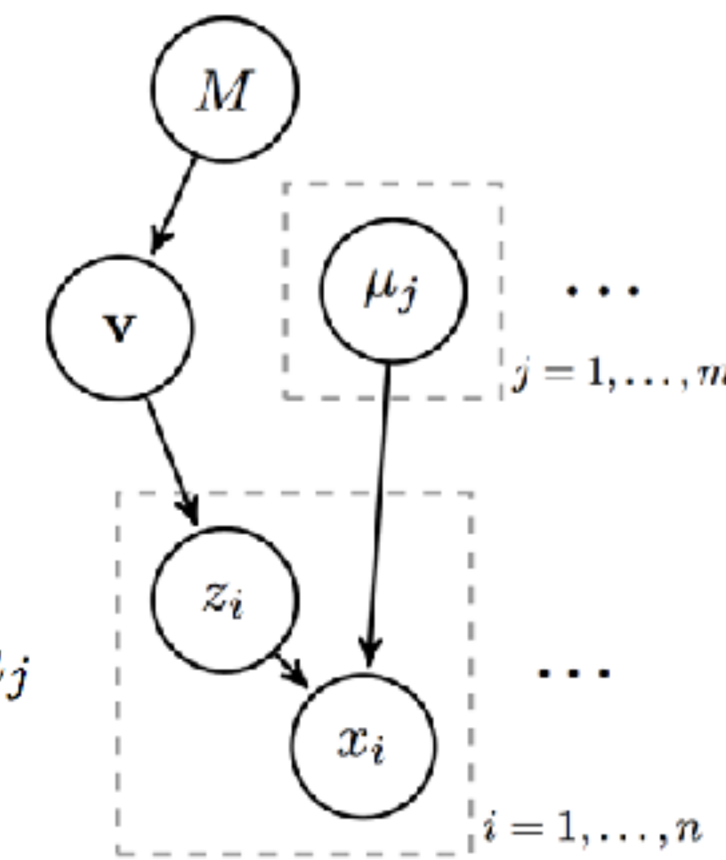


Compare with exact block Gibbs proposal

GMM with Unknown Number of Mixtures

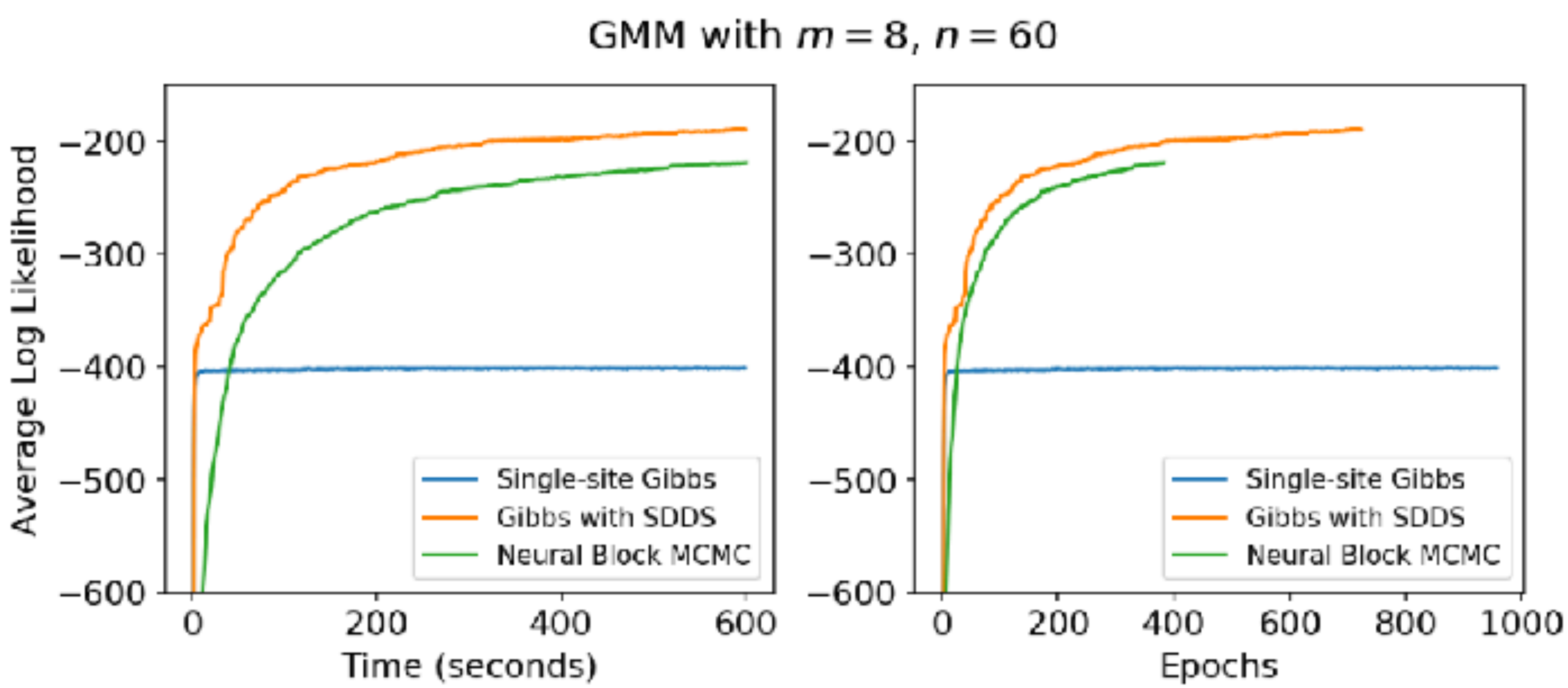
#mixture	$M \sim \text{Unif}\{1, 2, \ldots, m\}$	
mixture means	$\mu_j \sim \mathcal{N}(0, \sigma_\mu^2 I)$	$j = 1, \ldots, m$
mixture valid indicator	$\mathbf{v} \vert M \sim \text{Unif}\{a \in \{0,1\}^m : \sum_j a_j = M\}$	
data label	$z_i \vert \mathbf{v} \sim \text{Unif}\{j : v_j = 1\}$	$i = 1, \ldots, n$
data point	$x_i \vert z_i, \boldsymbol{\mu} \sim \mathcal{N}(\mu_{z_i}, \sigma^2 I)$	$i = 1, \ldots, n$

- Task: infer mixture means μ_j given data points x_i
- For simplicity, we marginalize label variables z_i
- Motif: propose two μ_j conditioned on x_i and other μ_j



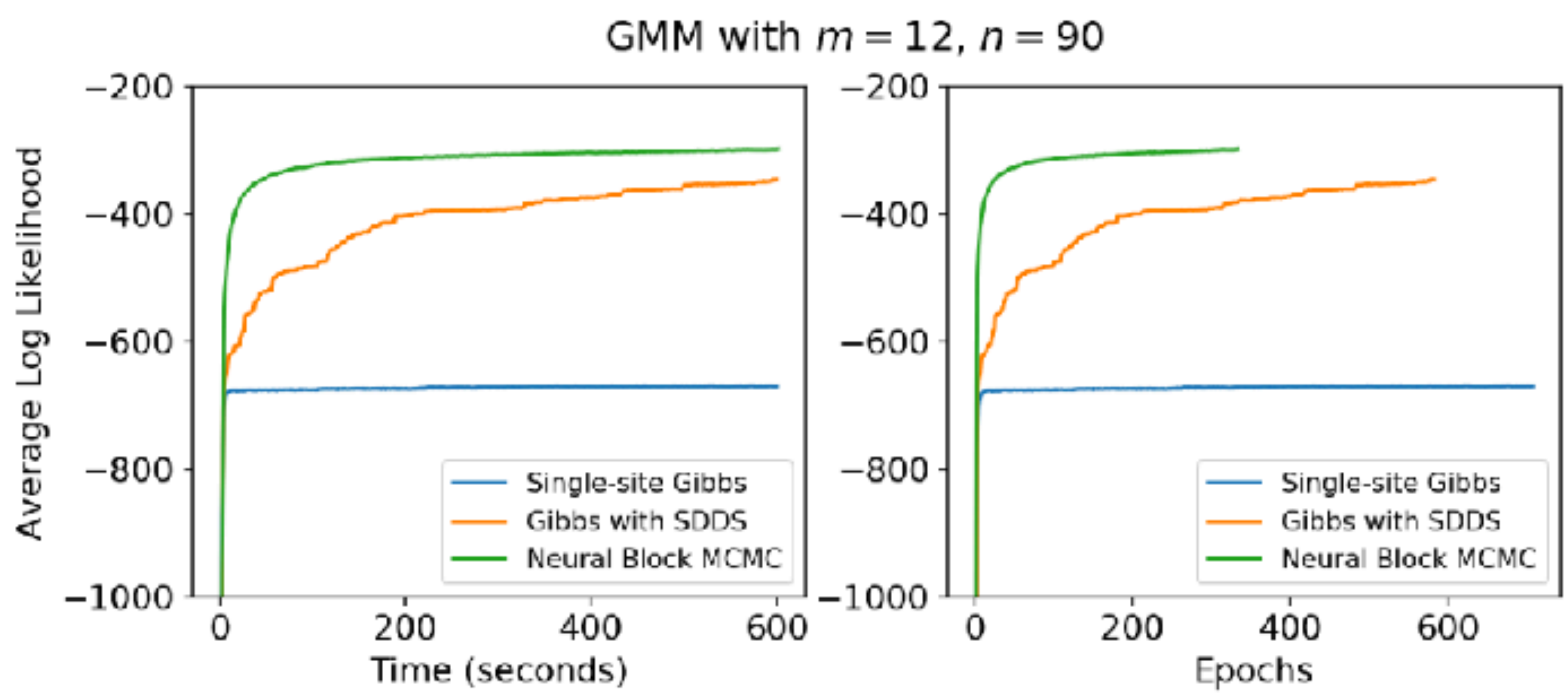
GMM with Unknown Number of Mixtures

- Train on small GMM
- Test on GMMs of various sizes
- Compare Gibbs with SDDS split-merge moves



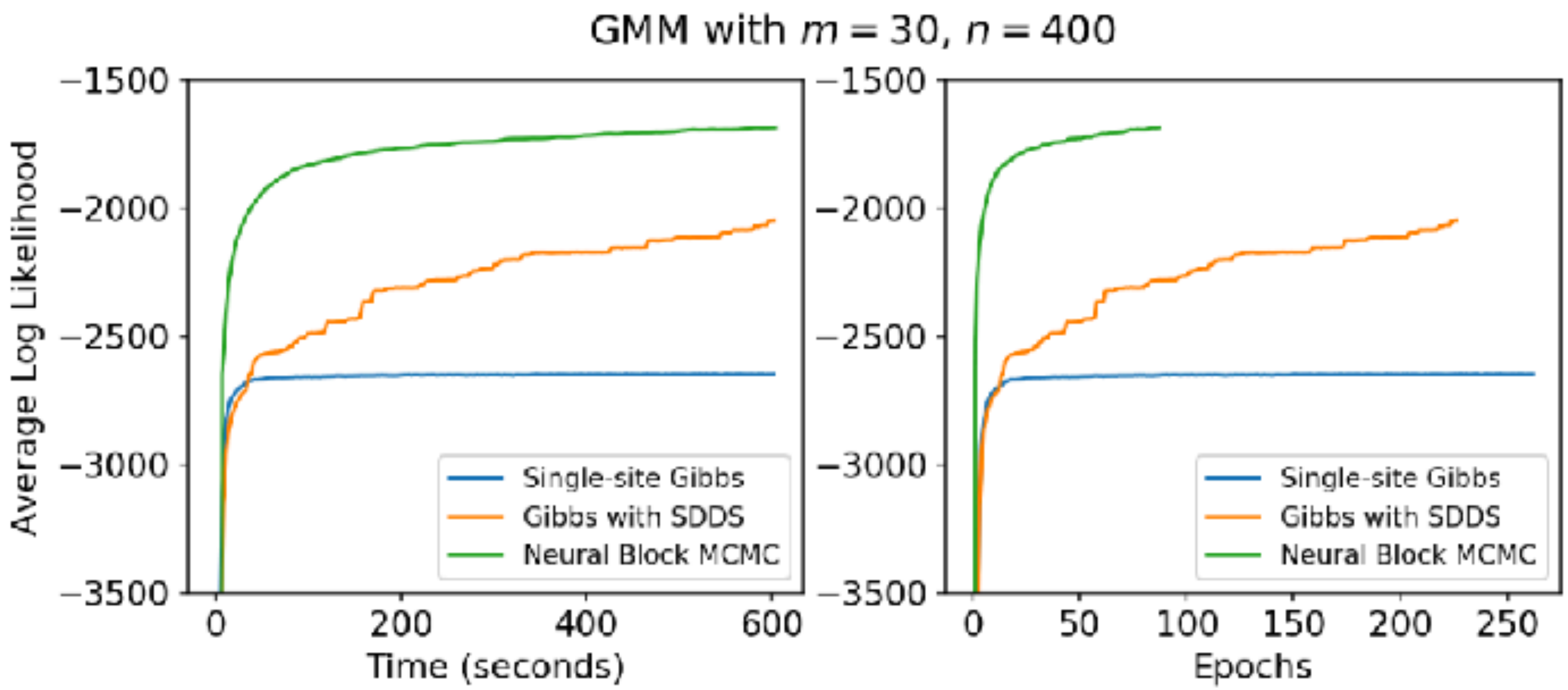
GMM with Unknown Number of Mixtures

- Train on small GMM
- Test on GMMs of various sizes
- Compare Gibbs with SDDS split-merge moves



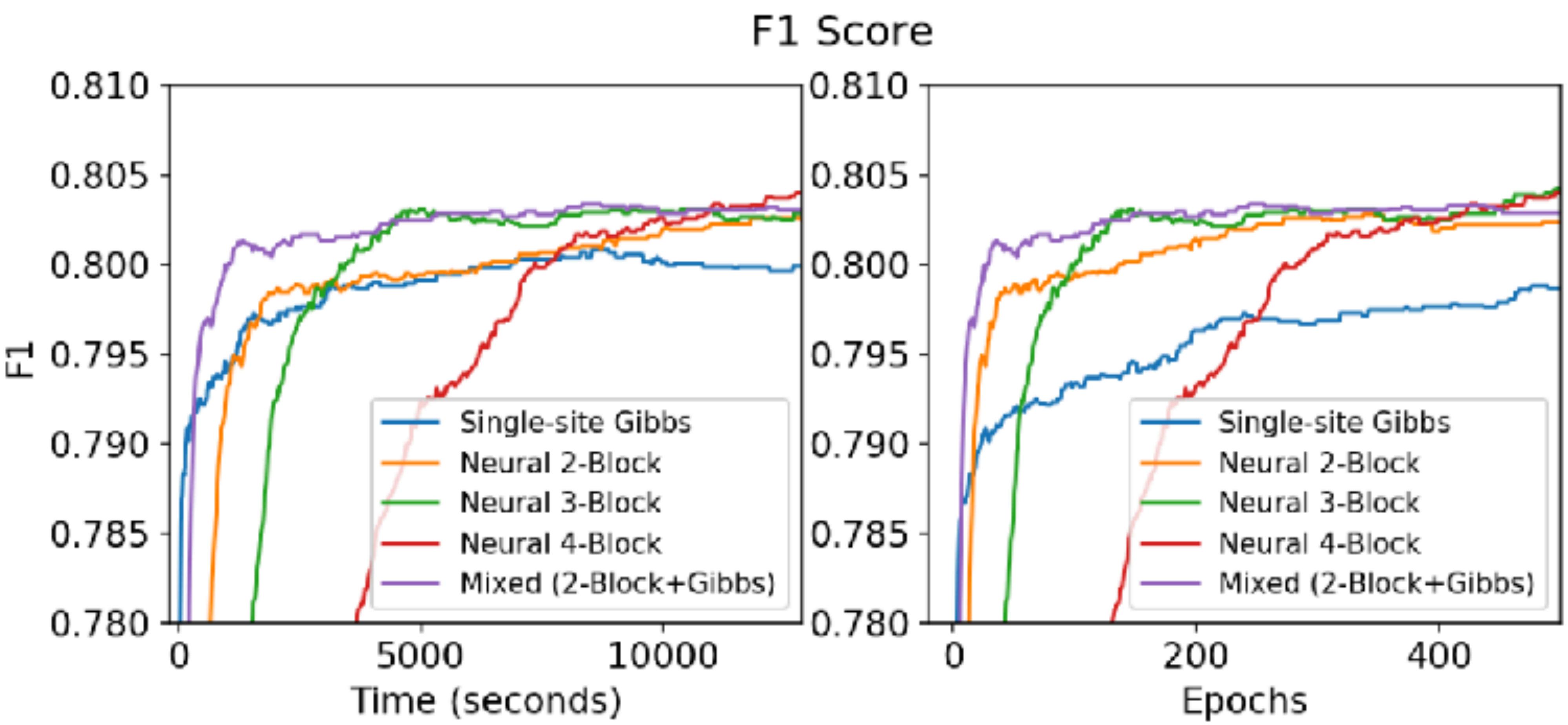
GMM with Unknown Number of Mixtures

- Train on small GMM
- Test on GMMs of various sizes
- Compare Gibbs with SDDS split-merge moves



Named Entity Recognition

- Task: infer NER labels given natural language sentence
- One way to solve NER is conditional random field (CRF)
- After a CRF is learned, MCMC is used to infer the NER tags for new sentences



Conclusion

- Pros:
  1. Enables training a library of expert samplers
  2. Applicable to a wide range of probabilistic models
- Cons:
  1. Hand-identified motifs
- Next steps:
  1. Automatically detect motifs and (adaptively) apply proposals
  2. Explore other network architectures: CNN, RNN, etc.