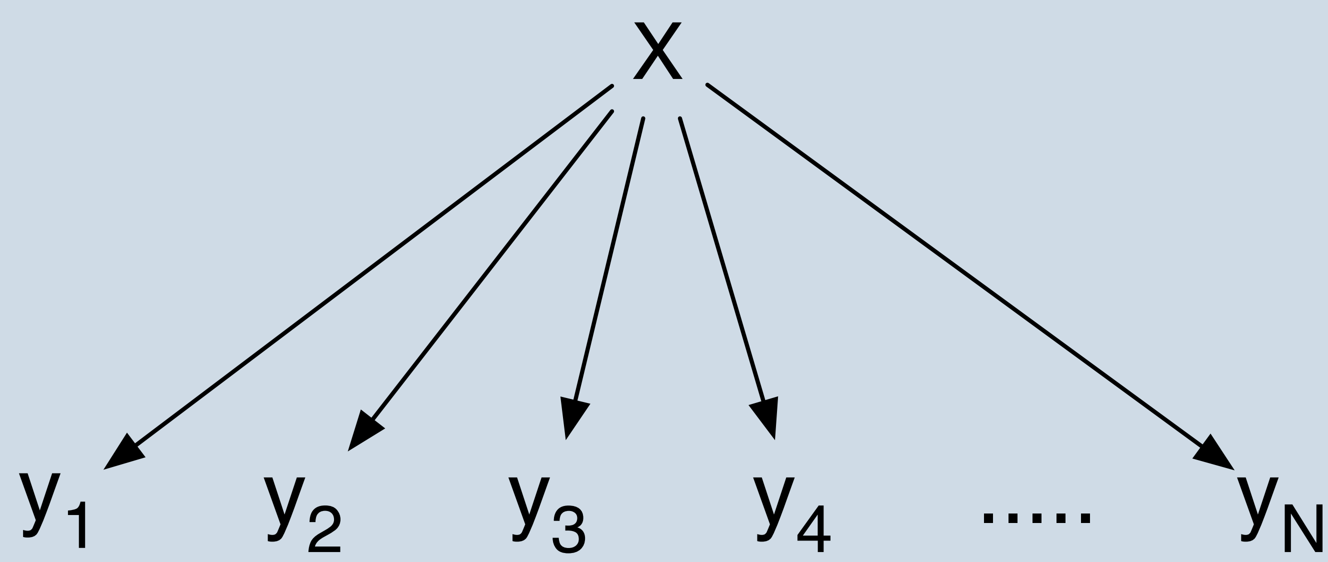
Distributed Bayesian Learning with Stochastic Natural-gradient EP and the Posterior Server

Yee Whye Teh

in collaboration with:
Minjie Xu, Balaji Lakshminarayanan,
Leonard Hasenclever, Thibaut Lienart, Stefan Webb,
Sebastian Vollmer, Charles Blundell

UNIVERSITY OF OXFORD

Bayesian Learning



- Parameter vector X.
- Data items $Y = y_1, y_2, \ldots y_N$.
- Model:

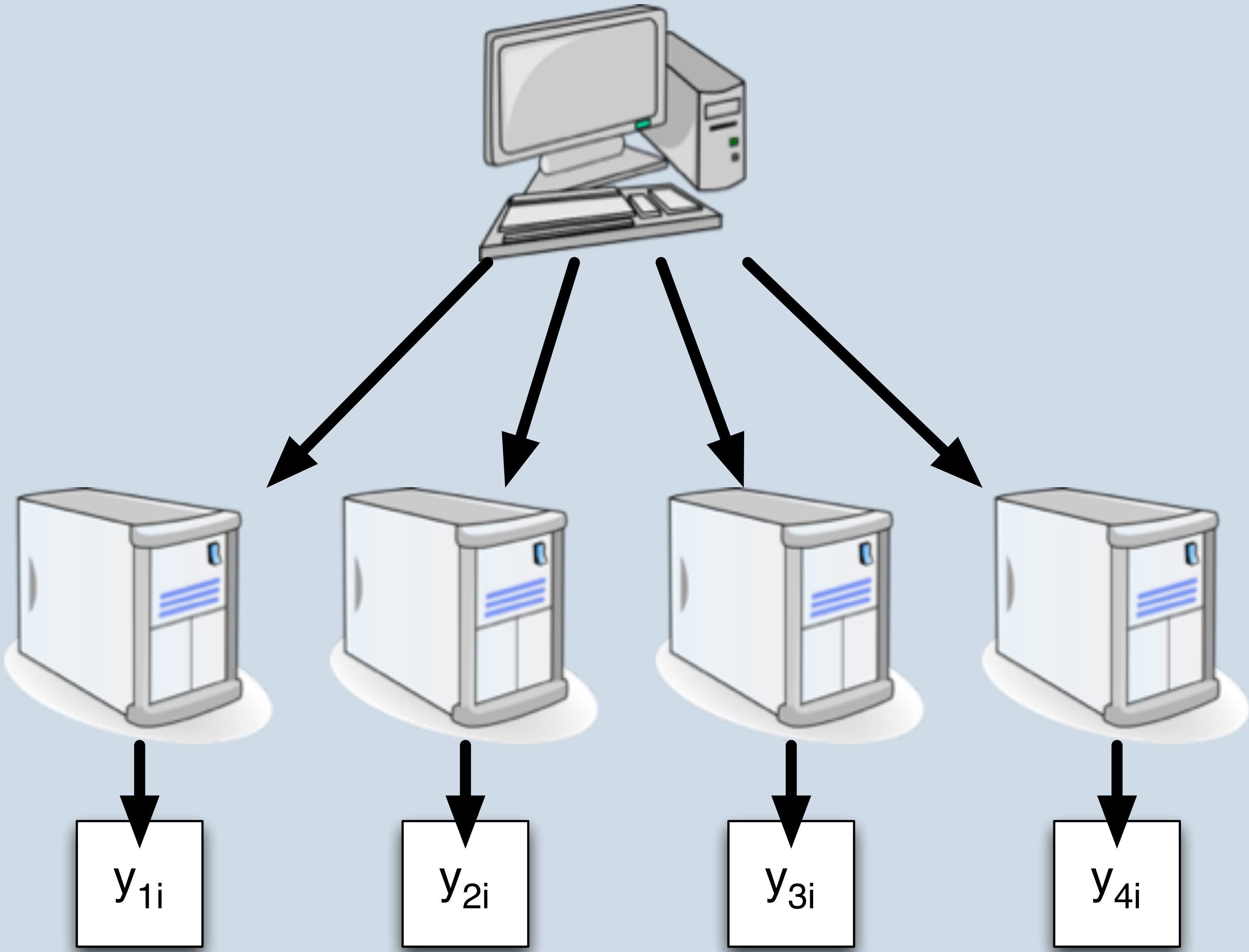
$$p(X, Y) = p(X) \prod_{i=1}^{N} p(y_i | X)$$

- Aim:

$$p(X|Y) = \frac{p(X)p(Y|X)}{p(Y)}$$

- Inference algorithms:
  - Variational inference: parametrise posterior as q_θ and optimize θ.
  - Markov chain Monte Carlo: construct samples $X_1 \ldots X_n \sim p(X|Y)$.

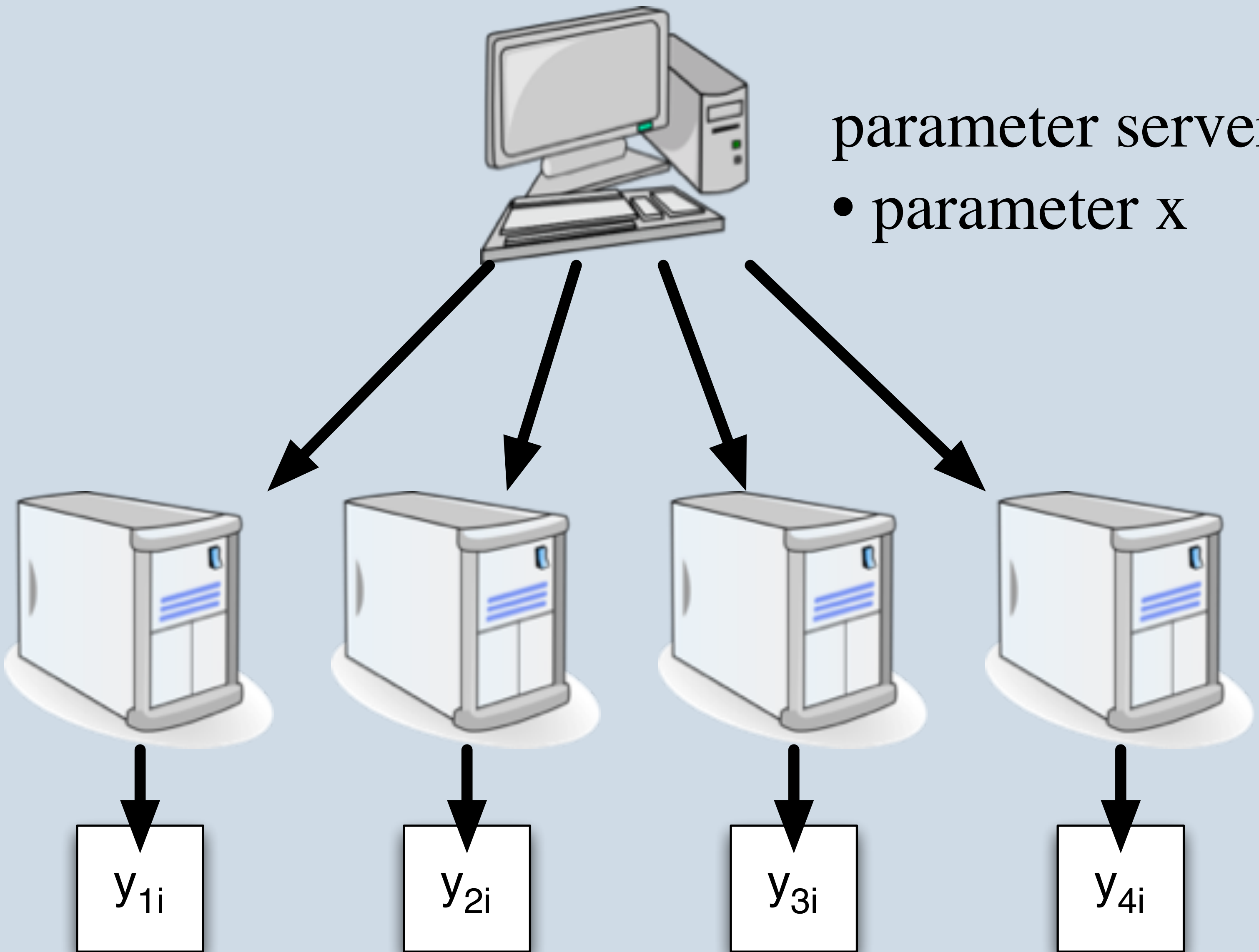
Machine Learning on Distributed Systems



- Distributed storage
- Distributed computation
- costly network communications

Parameter Server

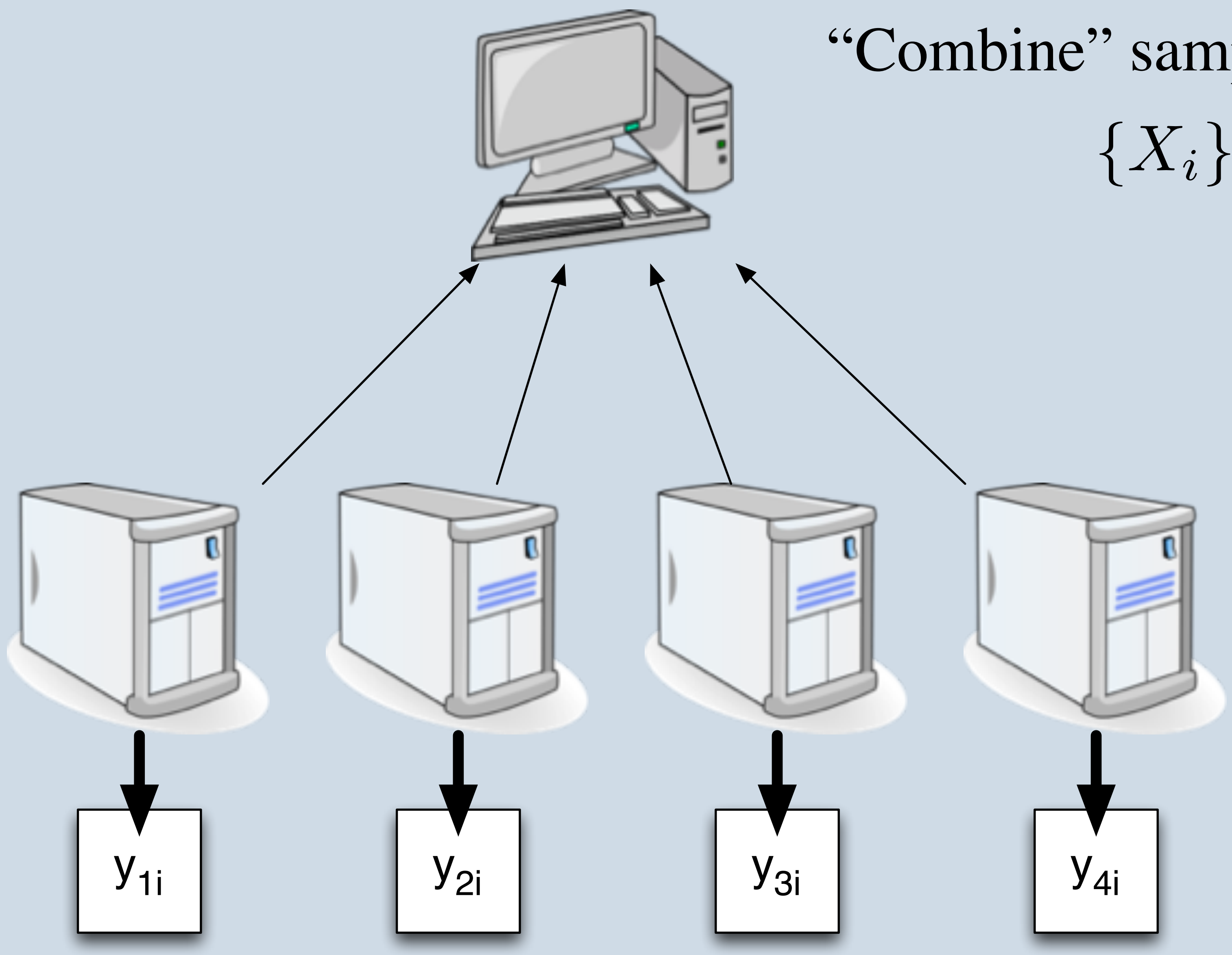
- Parameter server [Ahmed et al 2012], DistBelief network [Dean et al 2012].



parameter server:
- parameter x

worker:
- $x_i = x$
- updates to x_i'
- returns
  $\Delta x_i = x_i' - x_i$

Embarassingly Parallel MCMC Sampling



"Combine" samples together.

$$\{X_i\}_{i=1\ldots n}$$

Treat as independent inference problems. Collect samples.

$$\{X_{ji}\}_{j=1\ldots m, i=1\ldots n}$$

- Only communication at the combination stage.

[Scott et al 2013, Neiswanger et al 2013, Wang & Dunson 2013, Stanislav et al 2014]

Embarassingly Parallel MCMC Sampling

- Unclear how to combine worker samples well.
- Particularly if local posteriors on worker machines do not overlap.

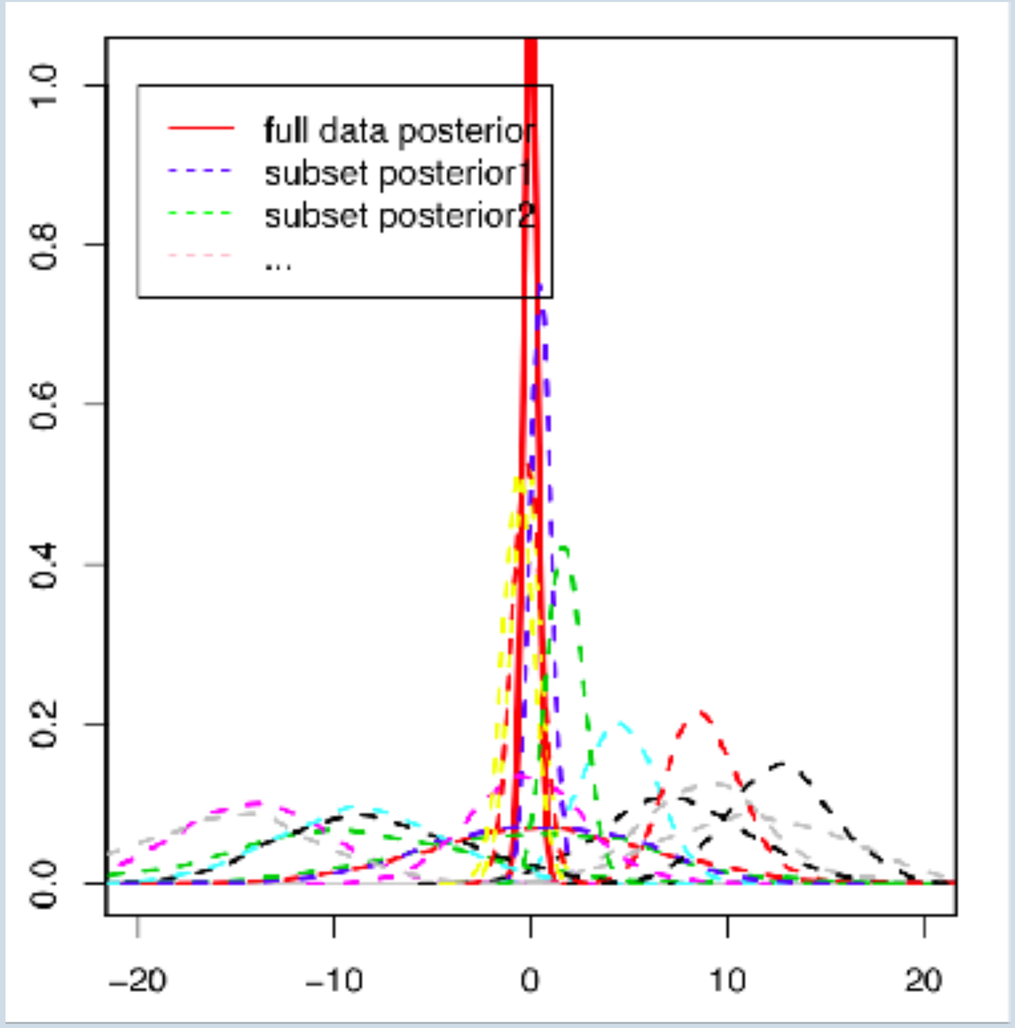


Figure from Wang & Dunson

Main Idea

- Identify regions of high (global) posterior probability mass.
- Shift each local posterior to agree with high probability region, and draw samples from these.
- How to find high probability region?
  - Defined in terms of low order moments.
  - Use information gained from local posterior samples (using small amount of communication).

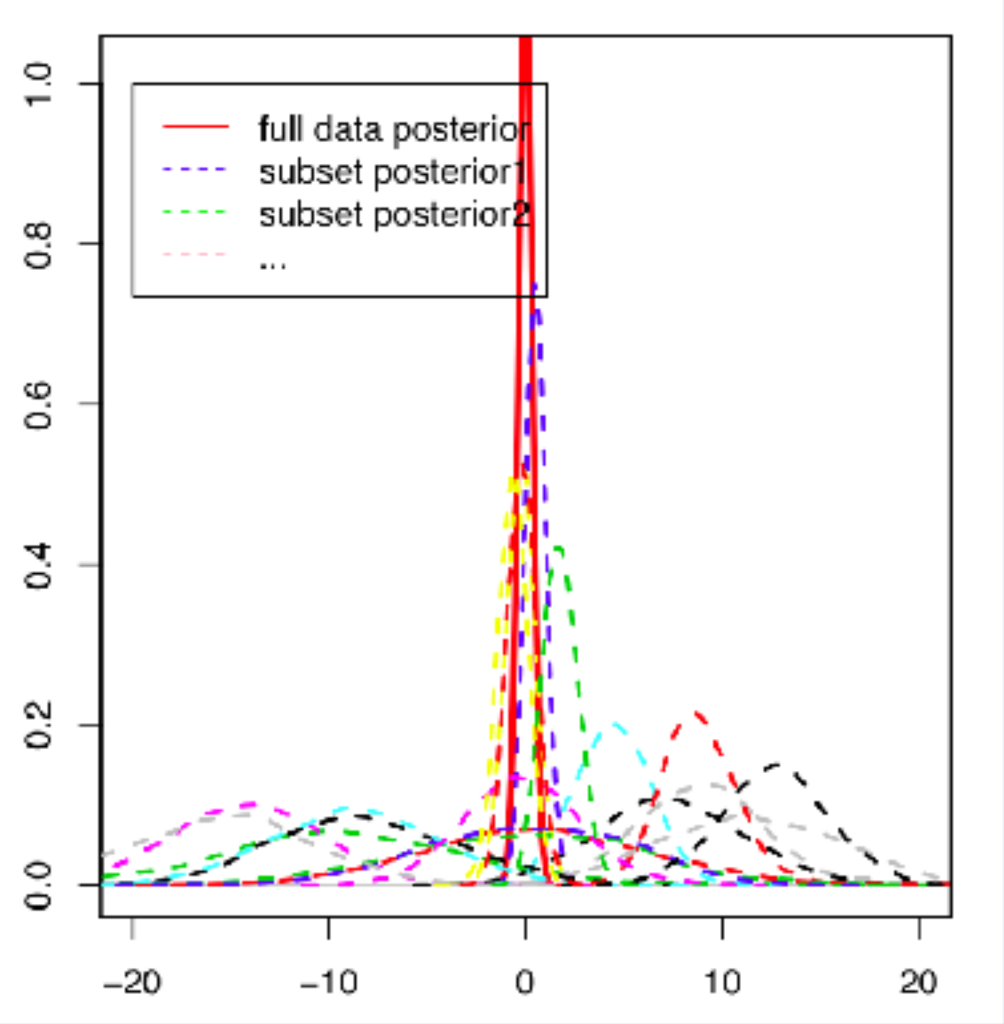


Figure from Wang & Dunson

Tilting Local Posteriors

- Each worker machine j has access only to its data subset.

$$p_j(X \mid y_j) = p_j(X) \prod_{i=1}^{I} p(y_{ji} \mid X)$$

where $p_j(X)$ is a local prior and $p_j(X \mid y_j)$ is local posterior.

- Adapt local priors $p_j(X)$ so that local posterior agree on certain moments

$$\mathbb{E}_{p_j(X \mid y_j)}[s(X)] = s_0 \quad \forall j$$

- Use expectation propagation (EP) [Minka 2001] to adapt local priors.

Expectation Propagation

- If N is large, the worker j likelihood term $p(y_j \mid X)$ should be well approximated by Gaussian

$$p(y_j \mid X) \approx q_j(X) = \mathcal{N}(X; \mu_j, \Sigma_j)$$

- Parameters fit iteratively to minimize KL divergence:

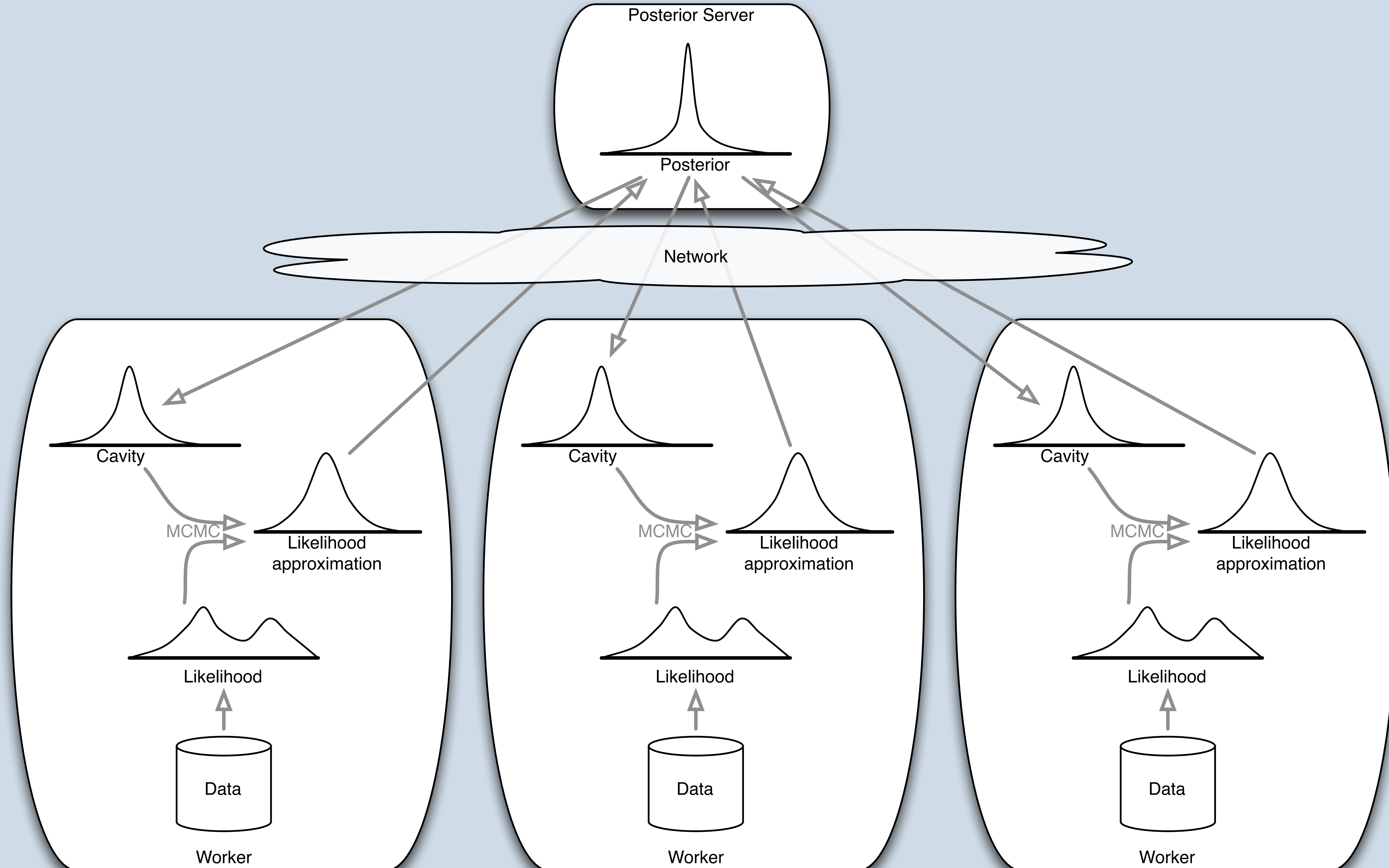
$$p(X \mid y) \approx p_j(X \mid y) \propto p(y_j \mid X) \underbrace{p(X) \prod_{k \neq j} q_k(X)}_{p_j(X)}$$

$$q_j^{\text{new}}(\cdot) = \arg \min_{\mathcal{N}(\cdot;\mu,\Sigma)} \mathrm{KL}\big(p_j(\cdot \mid y) \,\|\, \mathcal{N}(\cdot;\mu,\Sigma)p_j(\cdot)\big)$$

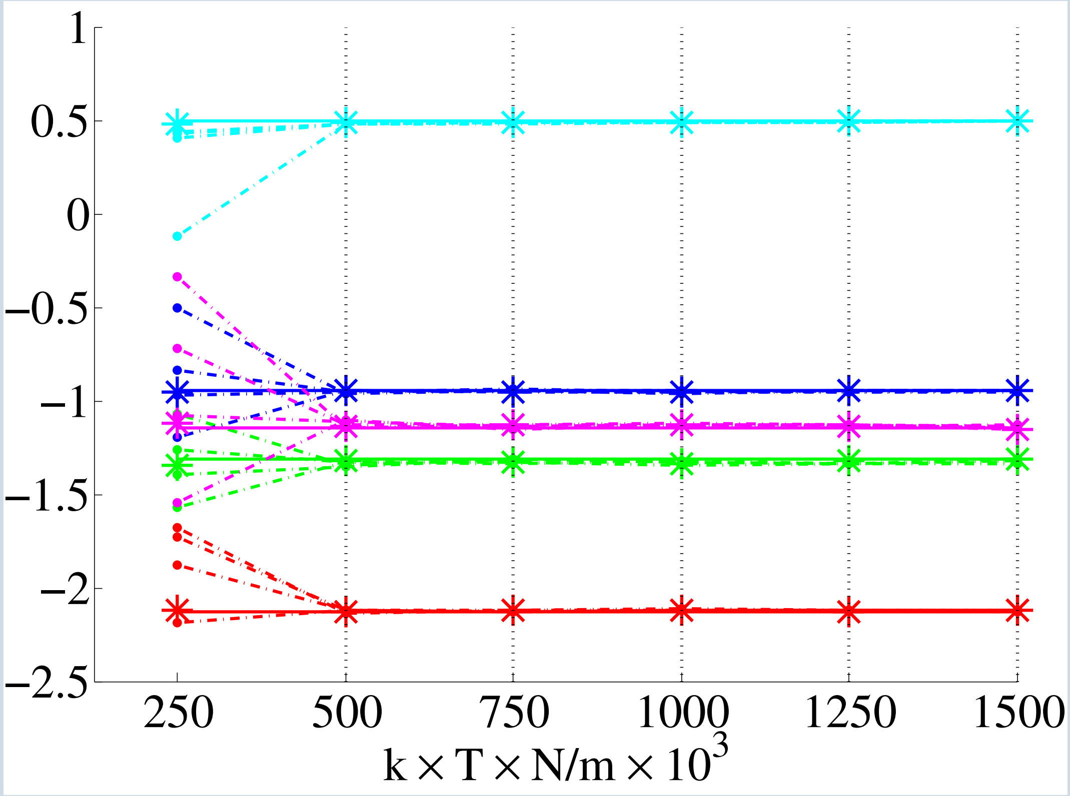
- Optimal q_j is such that first two moments of $\mathcal{N}(\cdot;\mu,\Sigma)p_j(\cdot)$ agree with $p_j(\cdot|y)$
- Moments of local posterior estimated using MCMC sampling.
- At convergence, first two moments of all local posteriors agree.

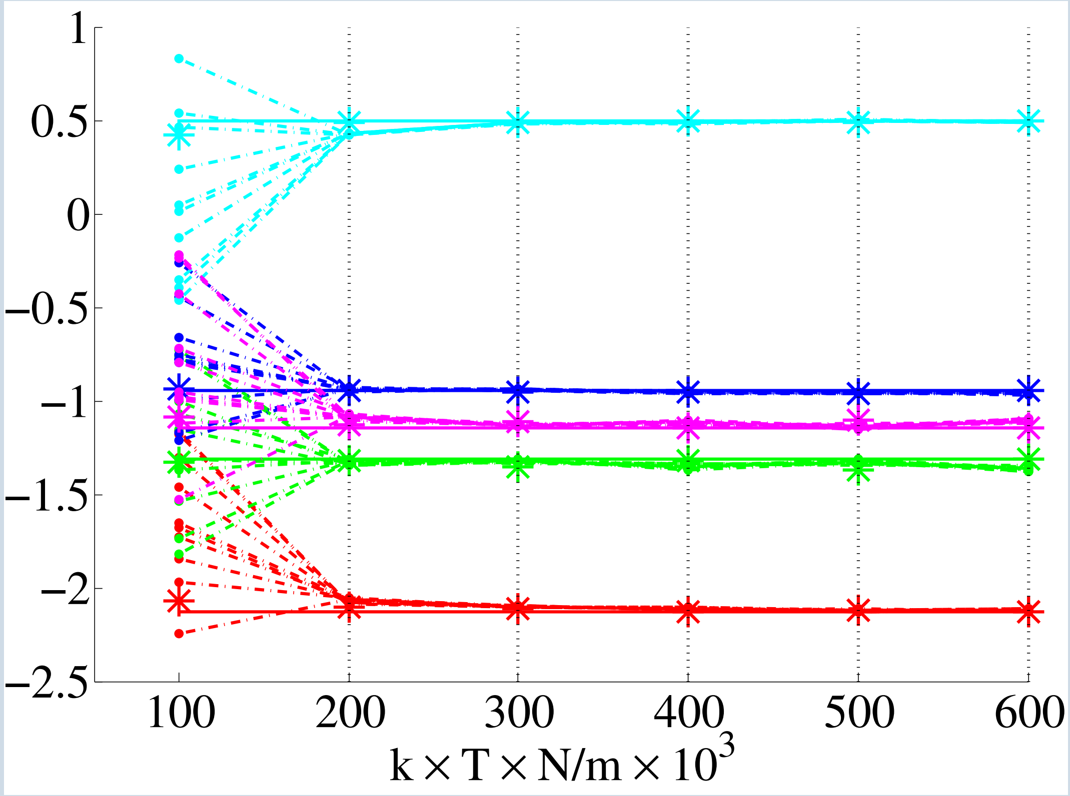
[Minka 2001]

Posterior Server Architecture

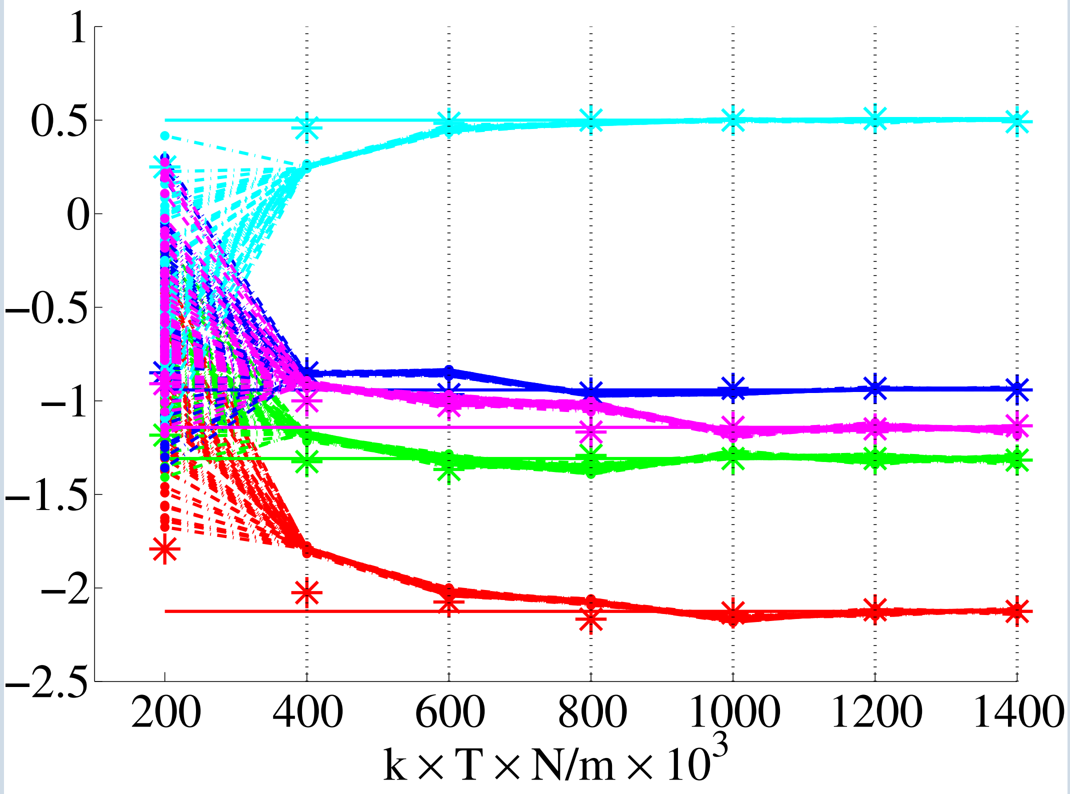


Bayesian Logistic Regression



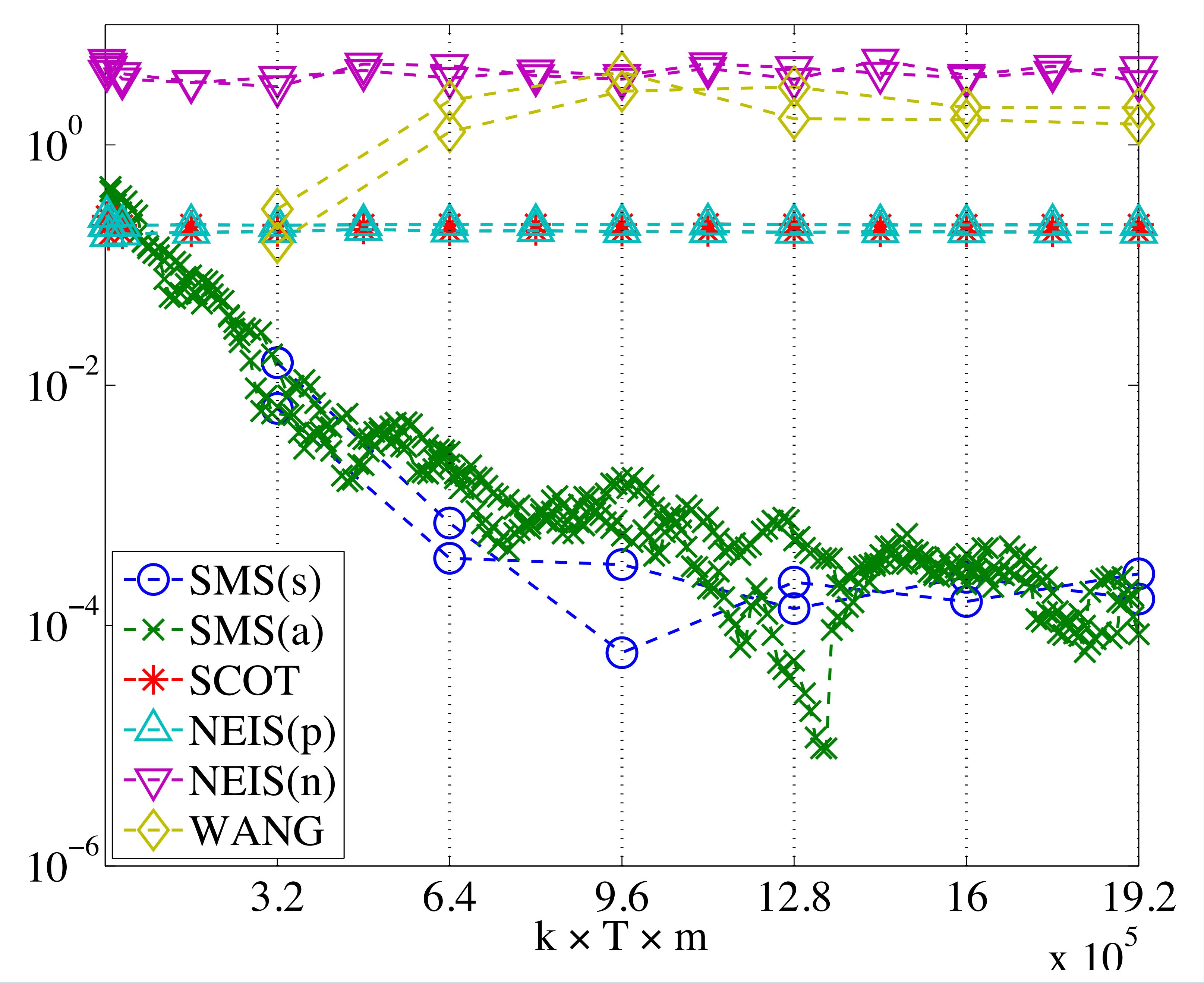


- Simulated dataset.
  - d=20, # data items N=1000.
- NUTS based sampler.
  - # workers m = 4,10,50.
  - # MCMC iters T = 1000,1000,10000.
- # EP iters k given as vertical lines.



Bayesian Logistic Regression

- MSE of posterior mean, as function of total # iterations.

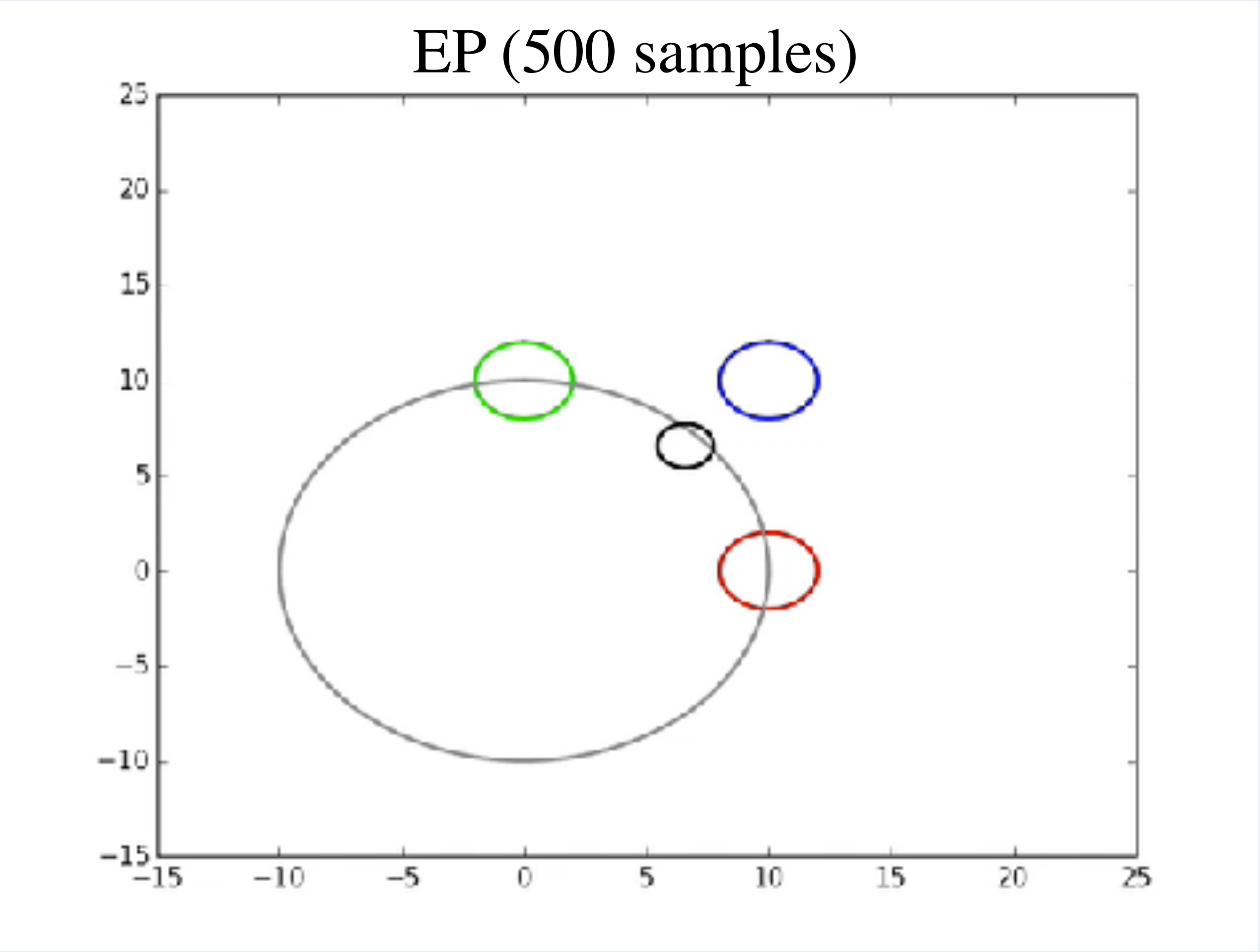


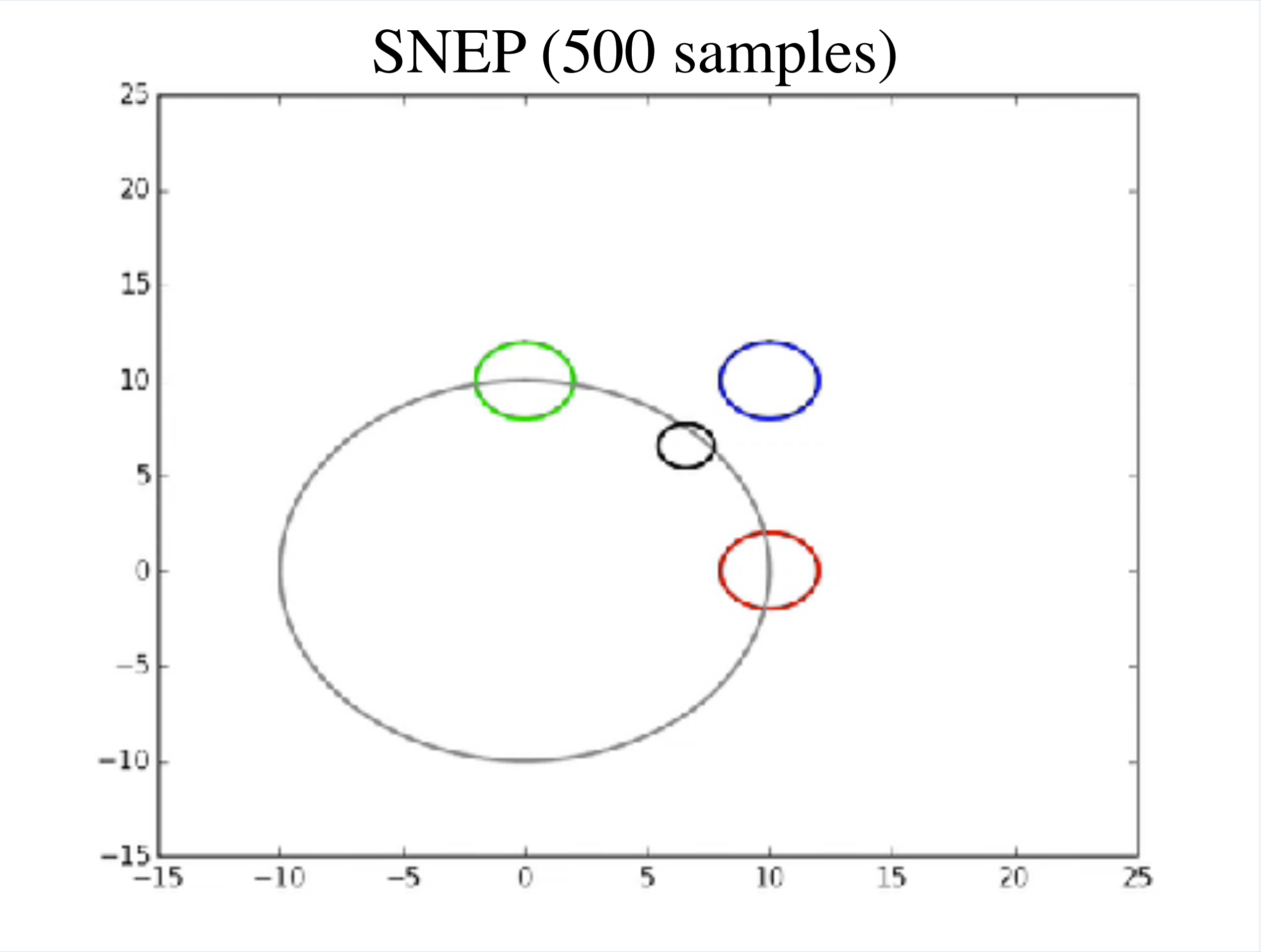
Stochastic Natural-gradient EP

- EP has no guarantee of convergence.
- EP technically cannot handle stochasticity in moment estimates.
- Long MCMC run needed for good moment estimates.
- Fails for neural nets and other complex high-dimensional models.

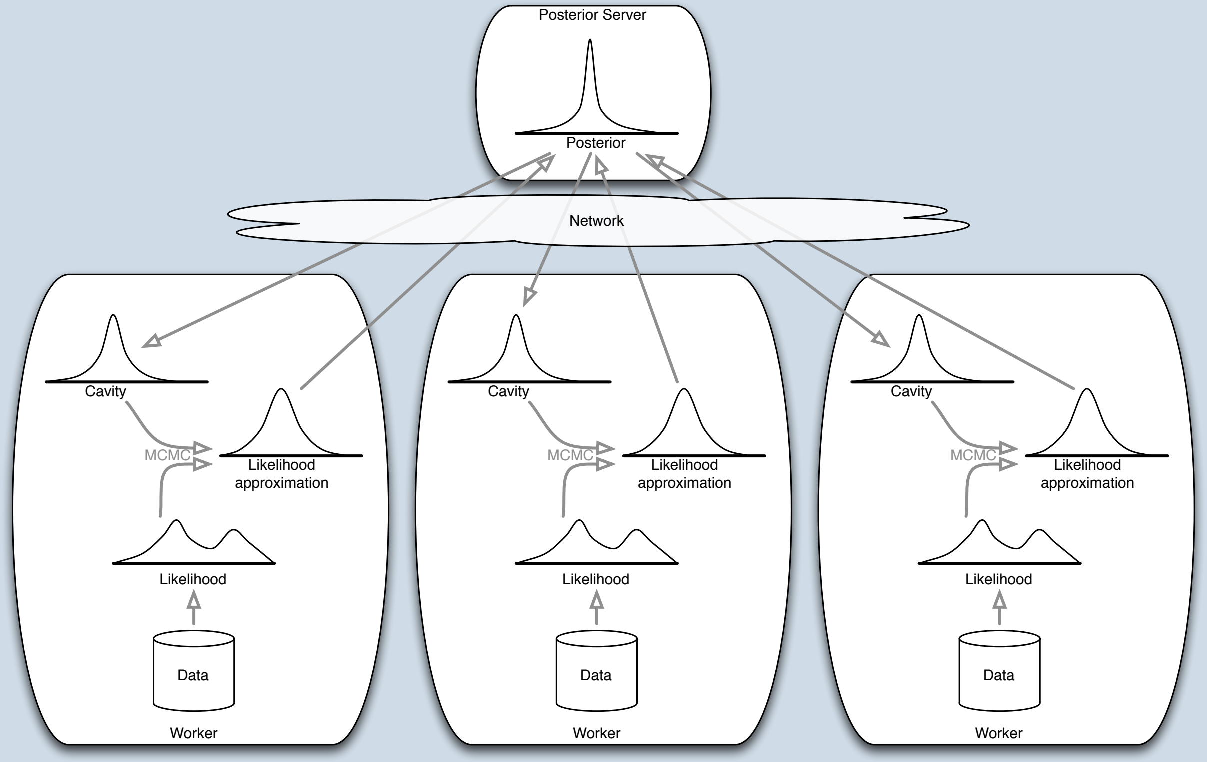
- Stochastic Natural-gradient EP:
  - Alternative variational algorithm to EP.
  - Convergent, even with Monte Carlo estimates of moments.
  - Double-loop algorithm [Welling & Teh 2001, Yuille 2002, Heskes & Zoeter 2002]

Demonstrative Example





Comparison to Maximum Likelihood SGD



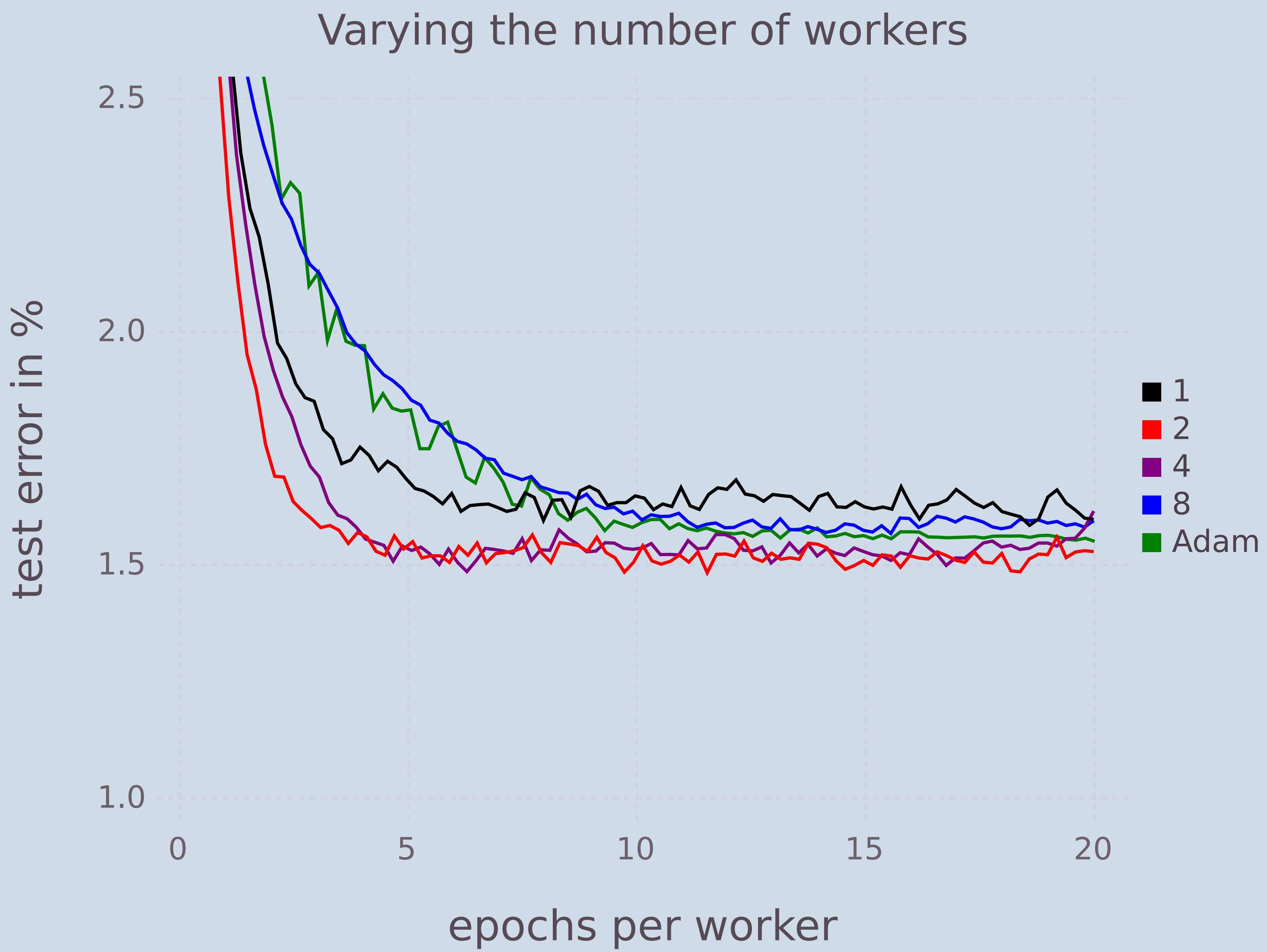
- Maximum likelihood via SGD:
  - DistBelief [Dean et al 2012]
  - Elastic-averaging SGD [Zhang et al 2015]
- Separate likelihood approximations and states per worker.
  - Worker parameters not forced to be exactly same.
- Each worker learns to approximate its own likelihood.
  - Can be achieved without detailed knowledge from other workers.
- Diagonal Gaussian exponential family.
  - Variance estimates are important to learning.

Experiments on Distributed Bayesian Neural Networks

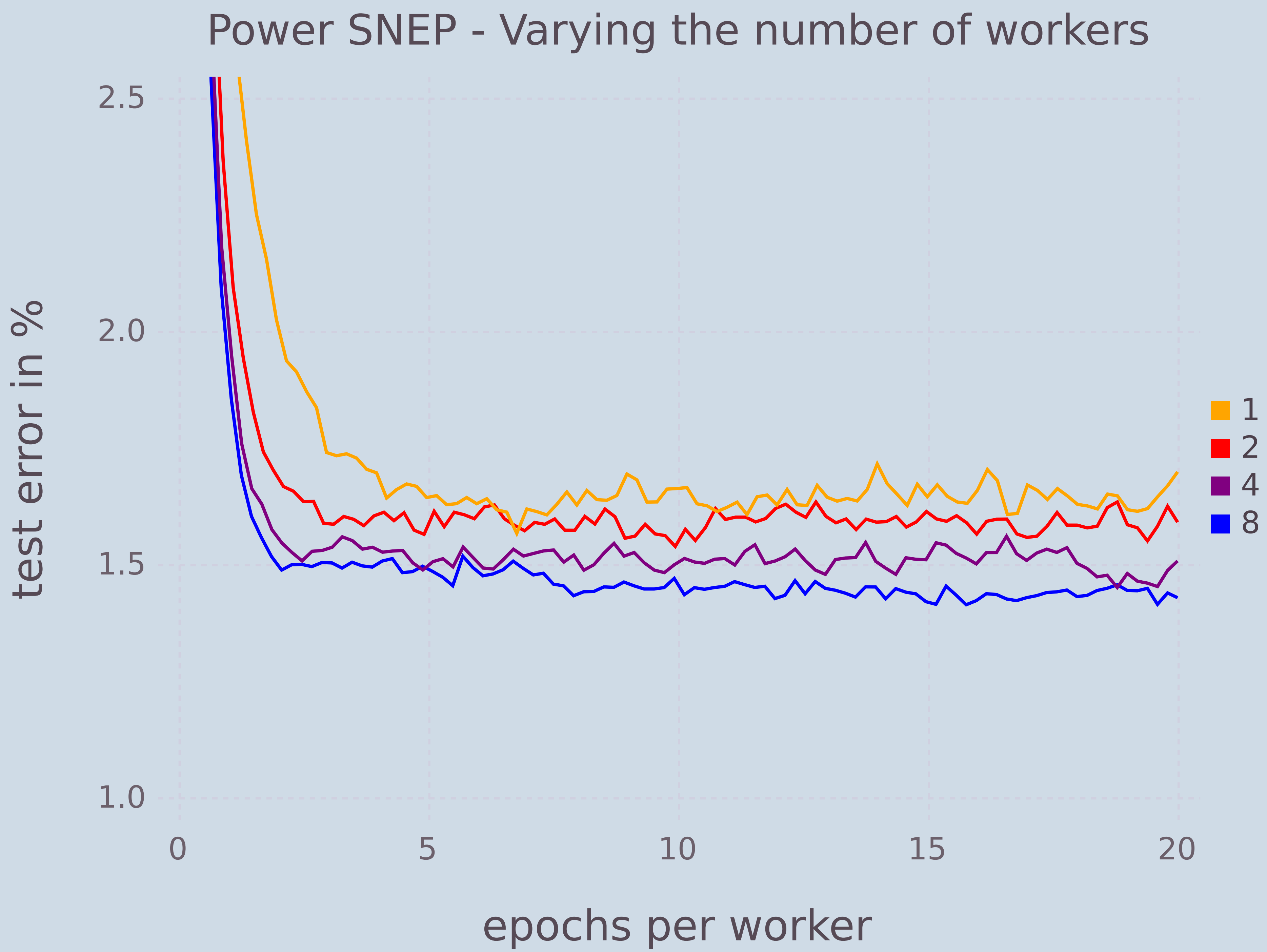
- Bayesian approach to learning neural network:
  - compute parameter posterior given complex neural network likelihood.
  - Diagonal covariance Gaussian prior and exponential-family approximation.
- Two datasets and architectures: MNIST fully-connected, CIFAR10 convnet.

- Implementation in Julia.
  - Workers are cores on a server.
  - Sampler is stochastic gradient Langevin dynamics [Welling & Teh 2011].
    - Adagrad [Duchi et al 2011]/RMSprop [Tieleman & Hinton 2012] type adaptation.
  - Evaluated on test accuracy.

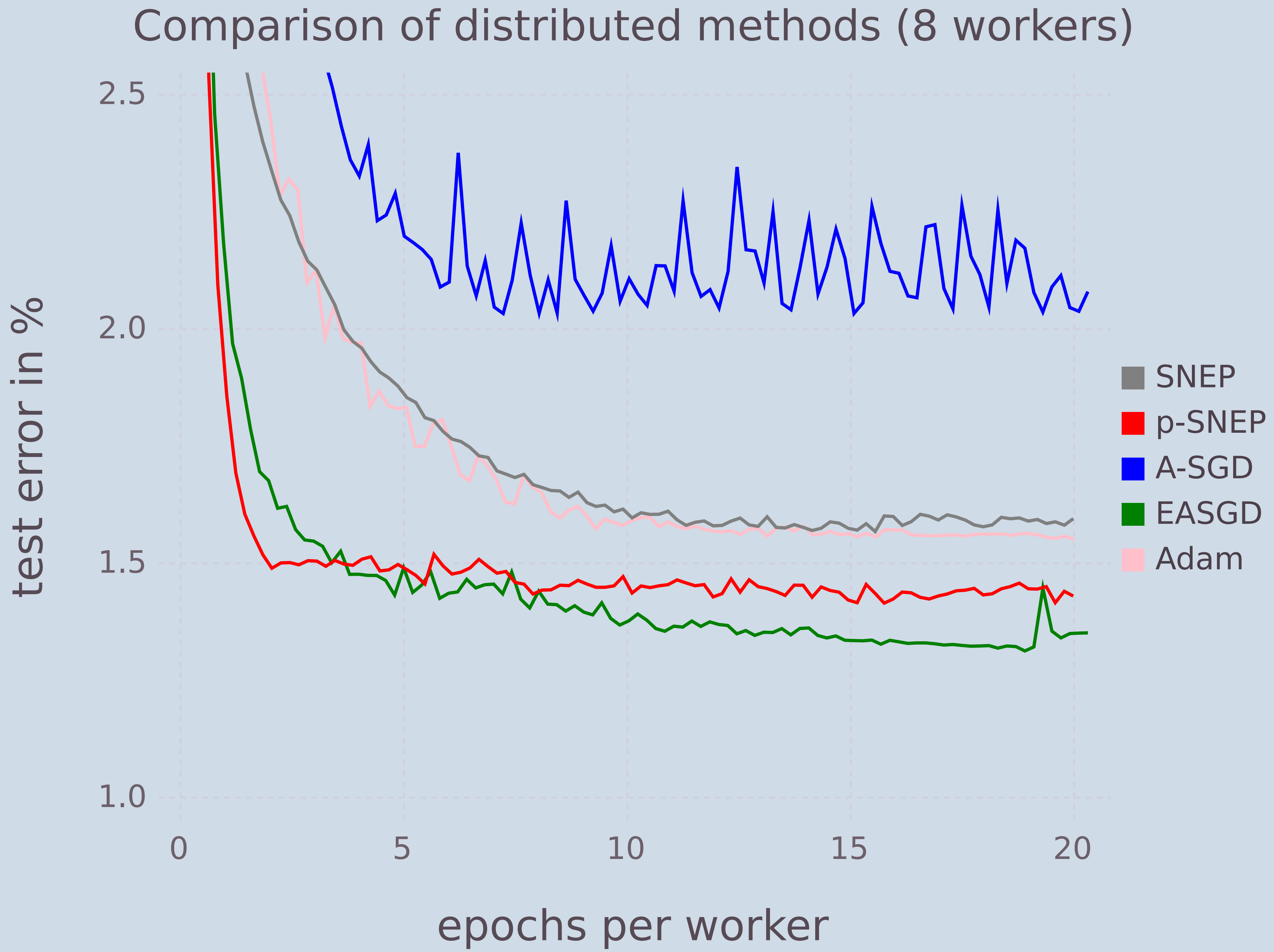
MNIST 500x300



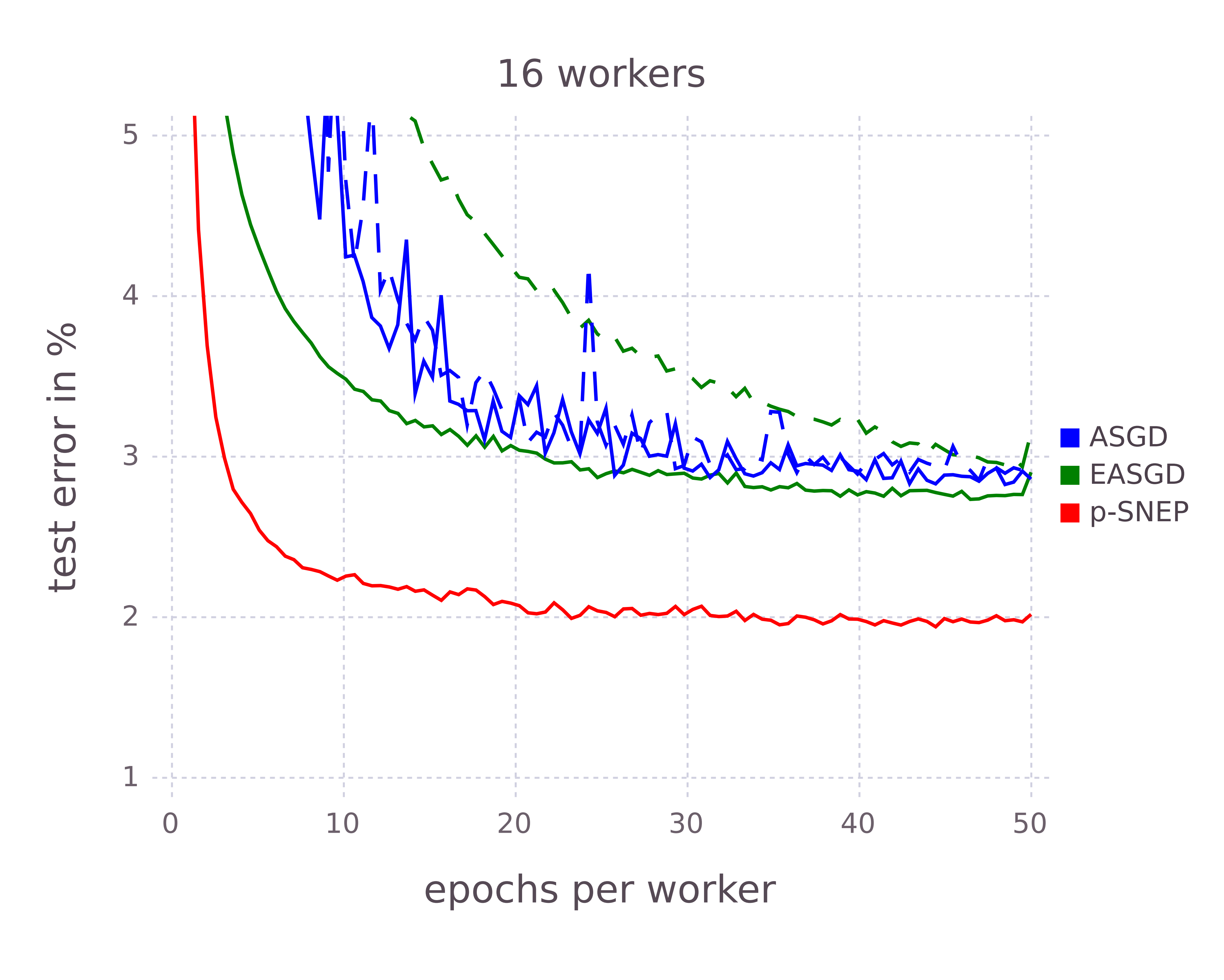
MNIST 500x300



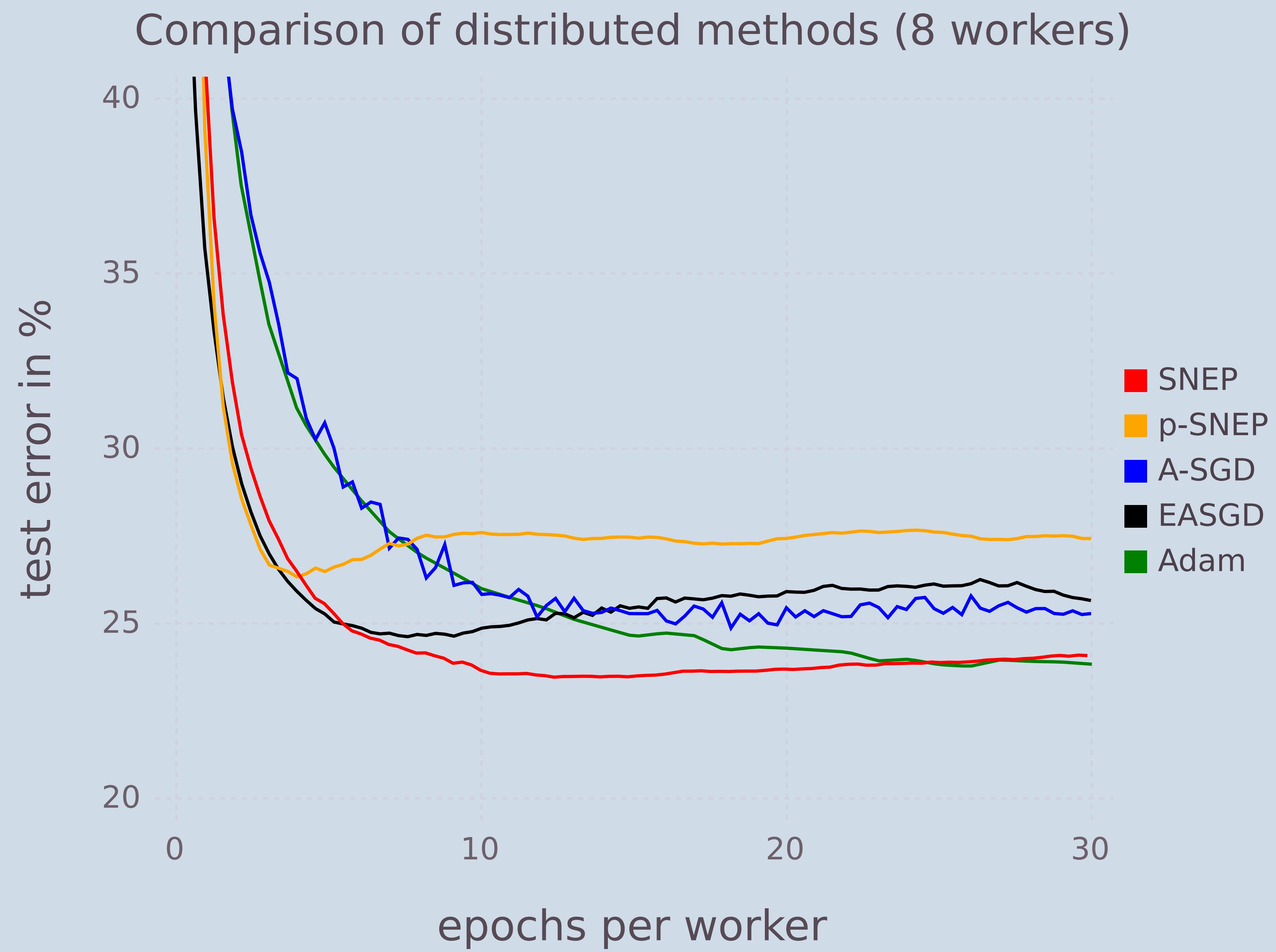
MNIST 500x300



MNIST Very Deep MLP



CIFAR10 ConvNet



Concluding Remarks

- Novel distributed learning based on a combination of Monte Carlo and a convergent alternative to expectation propagation.
- Combination of variational and MCMC algorithms.
  - Advantageous over both pure variational and pure MCMC algorithms.
- Being Bayesian can be advantageous computationally in distributed setting.
- Thank you!

EPSRC
Engineering and Physical Sciences Research Council

erc
European Research Council
Established by the European Commission