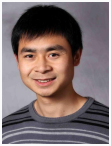
Approximate Inference with Amortised MCMC

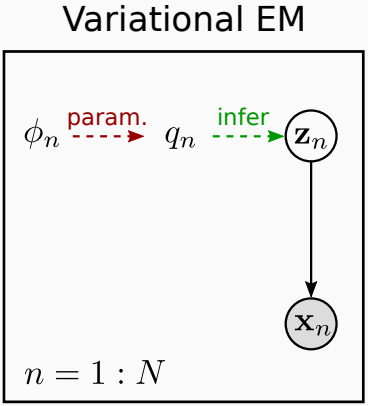
Yingzhen Li

MLG, University of Cambridge

Joint work with Rich Turner (Cambridge) and Qiang Liu (Dartmouth)
arXiv preprint 1702.08343

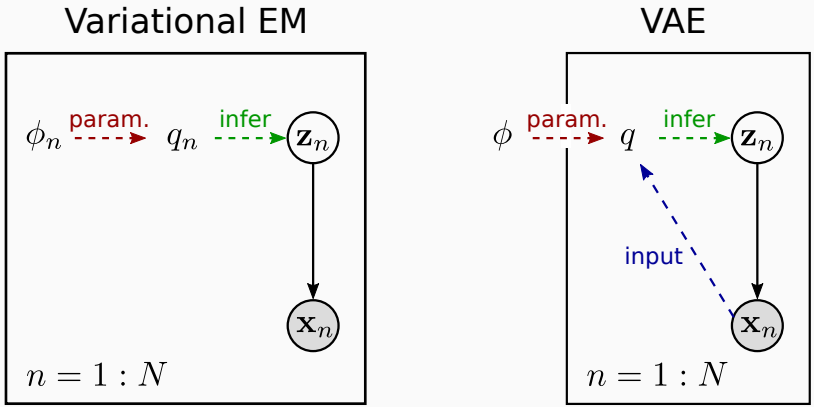


Recap: from Variational EM to VAE



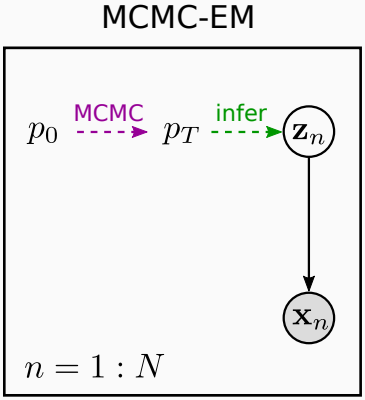
(For simplicity we omit the model parameter $\boldsymbol{\theta}$ but it would be trained by approx. MLE)

Recap: from Variational EM to VAE



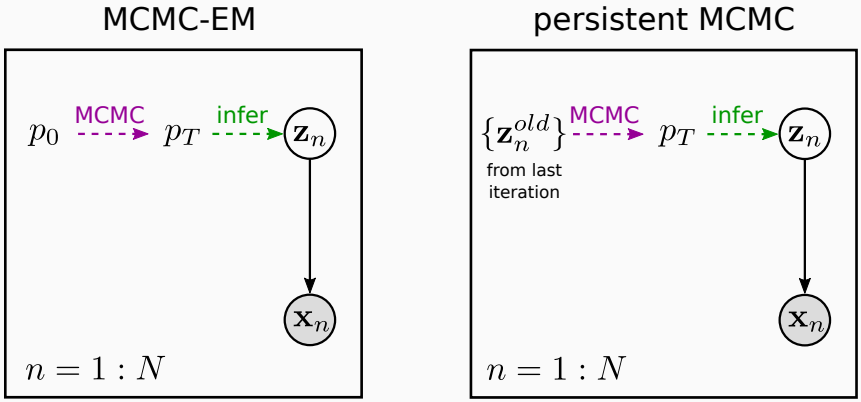
Amortised inference: memory efficient & no need to run VI optimisation in test time!

New: from MCMC-EM to amortised MCMC



For every $\boldsymbol{x}_n$, need to simulate MCMC for $T >> 0$ steps (slow!)

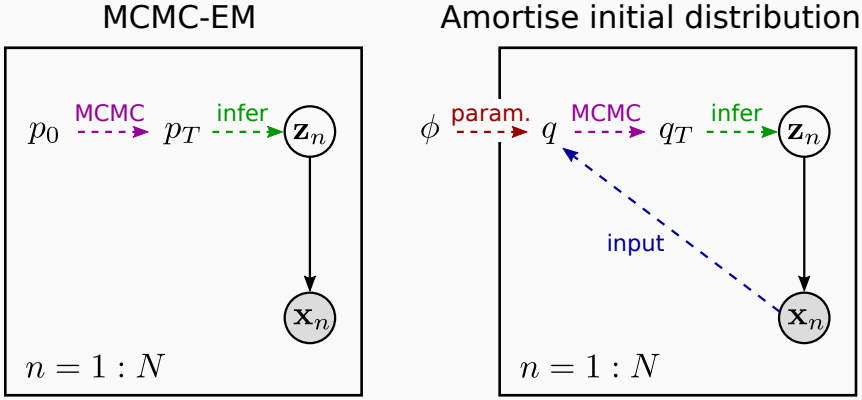
New: from MCMC-EM to amortised MCMC



Need to store all samples from the previous iteration, memory cost $\mathcal{O}(NKD)$.

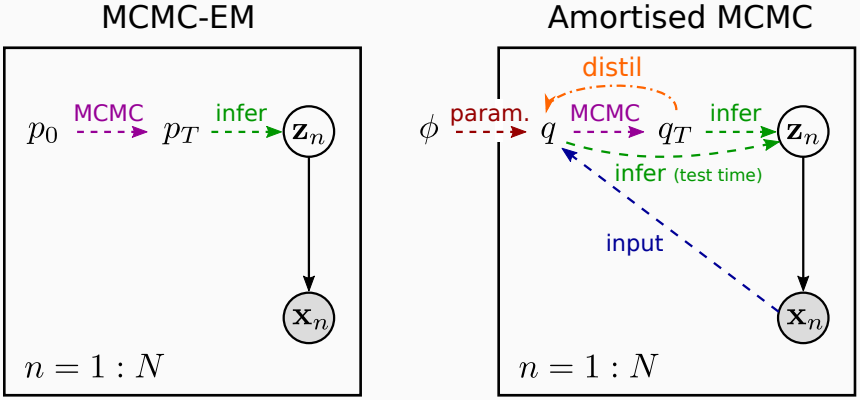
For a new datapoint, still need to run MCMC with $T >> 0$ starting from p_0 (slow!)

New: from MCMC-EM to amortised MCMC



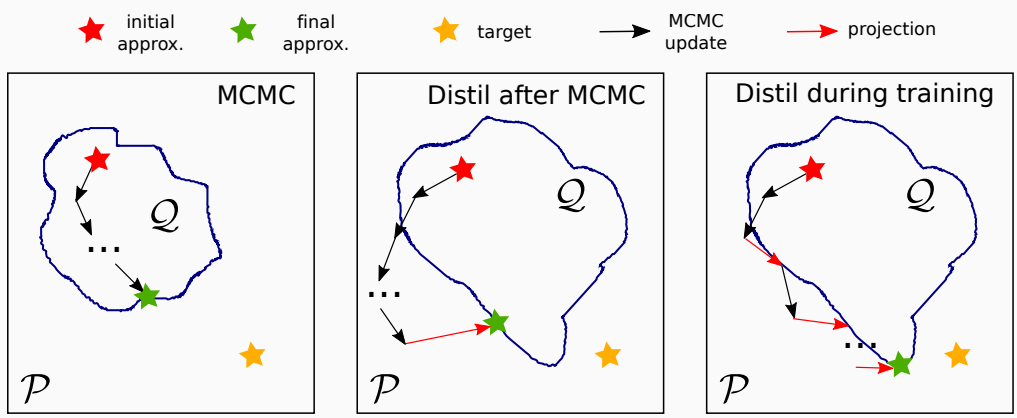
This method essentially cares about q_T only, so no need for $q(\boldsymbol{z}|\boldsymbol{x}) \approx p(\boldsymbol{z}|\boldsymbol{x})$.
In test time still need to run MCMC to obtain samples from q_T (slow!)

New: from MCMC-EM to amortised MCMC



Distillation happens at the same time during training (thus also improving q_T),
and now $q(\boldsymbol{z}|\boldsymbol{x}) \approx p(\boldsymbol{z}|\boldsymbol{x})$ – no need to run MCMC in test time!

Understanding "distillation during training"



$\mathcal{Q} = \{\frac{1}{K}\sum_k \delta(\mathbf{z} = \mathbf{z}^k)\}$

(very restrictive when K is small)

$\mathcal{Q} = \{q_\phi(\mathbf{z})\}$

distillation during training can also improve results
(think about *projected* gradient descent)

Amortised MCMC: distillation rules

We want q to be implicit!
(i.e. can sample from q but cannot evaluate density)

idea: match samples $\{z_0^k\} \sim q$ to samples $\{z_T^k\} \sim q_T$!

- We tested the original GAN idea [1]
- In general, any GAN-like technique is applicable!

[1] Goodfellow et al. Generative Adversarial Networks. NIPS 2014.

Training generative models with implicit q distributions

Generative model: a small convolutional decoder

- VAE training:
  - Gaussian encoder: a symmetric flip of the decoder
- Amortised MCMC (AMC) training:
  - CNN-G encoder: $\boldsymbol{z} = MLP([CNN(\boldsymbol{x}), \epsilon])$, $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  - CNN-B encoder: $\boldsymbol{z} = MLP([CNN(\boldsymbol{x}) \odot \epsilon])$, $\epsilon \sim Bern(0.5)$
  - MCMC: Langevin Dynamics w/out rejection

data

Gaussian encoder + VAE

CNN-G + AMC

CNN-B + AMC

An open question for test-LL evaluation

Table 1: Average Test LL and effective sample size (ESS). η as the stepsize for Langevin dynamics.

Encoder	**Method**	IS-LL	IS-ESS	HAIS-LL	HAIS-ESS
Gaussian	VAE	**-81.31**	**104.11**	-80.64	91.59
	MCMC-VI, $T = 5$, $\eta = 0.2$	-90.06	110.58	-89.79	85.63
	AMC, $T = 5$, $\eta = 0.2$	-90.71	49.02	-89.64	87.93
CNN-G	AMC, $T = 5$, $\eta = 0.2$	-90.84	31.60	-89.35	87.49
	AMC, $T = 50$, $\eta = 0.02$	-83.30	6.84	**-78.23**	**77.78**
	AVB	-94.97	11.30	-85.92	57.21
CNN-B	AMC, $T = 5$, $\eta = 0.2$	-90.75	34.17	-89.42	88.10
	AMC, $T = 50$, $\eta = 0.02$	-83.62	8.88	-80.03	80.71
	AVB	-89.47	8.98	-82.66	76.90
N/A	persistent MCMC, $T = 50$, $\eta = 0.02$	-84.43	9.14	-78.88	77.29

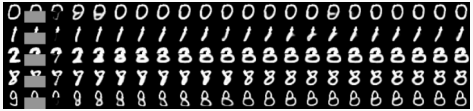
- HAIS seems to be more reliable ($K = 100$), compared to importance sampling (IS, $K = 5000$)
- The best case (CNN-G) is better than persistent MCMC

Missing data imputation

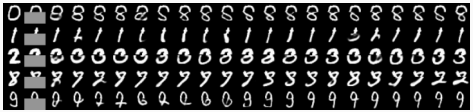
- Given an image $\boldsymbol{x} = [\boldsymbol{x}_o, \boldsymbol{x}_m]$ with missing values, repeat the following for T steps:
  - sample $\boldsymbol{z} \sim q(\boldsymbol{z}|\boldsymbol{x}_o, \boldsymbol{x}_m)$
  - sample $\boldsymbol{x}^* \sim p(\boldsymbol{x}|\boldsymbol{z})$ and set $\boldsymbol{x}_m \leftarrow \boldsymbol{x}^*_m$

Table 2: Label entropy on nearest neighbours. The l_1 distance is divided by the number of pixels.

Dataset	VAE	CNN-G	CNN-B
Entropy	0.411±0.0389	0.701±0.0476	0.933±0.0491
l_1-norm	0.061±0.0002	0.059±0.0001	0.064±0.0002



Gaussian encoder + VAE



CNN-G + AMC

CNN-B + AMC

Future work

- A lot more to be done!
  - what's the best combo of q, MCMC algorithm, and the distillation rule?
    (Langevin dynamics & original GAN are inefficient, now trying HMC & WGAN)
- Reuse intermediate samples from q_1 to q_{T-1}?
- Discrete distributions?
- Other ideas to mix MCMC, VI and implicit distributions?
  - e.g. see "reparameterised MCMC" by Michalis Titsias

Come and find me at the posters!
(we have more distillation rules & results)
(also see later spotlight on gradient estimators)