Diffusion-based variational inference

Justin Domke, University of Massachusetts Amherst

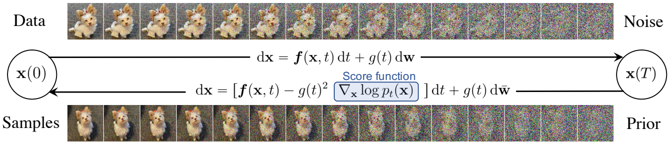
these slides: `t.ly/9vZvk` or `people.cs.umass.edu/domke/diffusion.pdf`

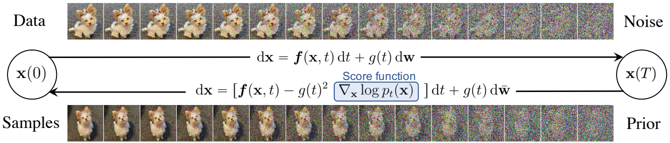
do probabilistic inference

There are only two ways I know to ~~make money in business~~:
~~bundling~~ and ~~unbundling.~~

augmenting variables   integrating variables out

Jim Barksdale


Data
Noise
$\mathbf{x}(0)$
$\mathrm{d}\mathbf{x} = \boldsymbol{f}(\mathbf{x}, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}$
$\mathbf{x}(T)$
Score function
$\mathrm{d}\mathbf{x} = [\boldsymbol{f}(\mathbf{x}, t) - g(t)^2\, \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\,]\,\mathrm{d}t + g(t)\,\mathrm{d}\bar{\mathbf{w}}$
Samples
Prior



meanwhile...

Variational Inference with Hamiltonian Monte Carlo

Christopher Wolf
Maximilian Karl* & Patrick van der Smagt*
Chair of Robotics and Embedded Systems, Department of Informatics, Technische Universität München, Germany

Stochastic Normalizing Flows

Hao Wu, Tongji University, Shanghai, P.R. China
Jonas Köhler, FU Berlin, Berlin, Germany
Frank Noé, FU Berlin, Berlin, Germany

Monte Carlo Variational Auto-Encoders

Achille Thin, Nikita Kotelevskii, Alain Durmus, Maxim Panov, Eric Moulines, Arnaud Doucet

Differentiable Annealed Importance Sampling and the Perils of Gradient Noise

Guodong Zhang, Kyle Hsu, Jianing Li, Chelsea Finn, Roger Grosse

MCMC Variational Inference via Uncorrected Hamiltonian Annealing

Tomas Geffner, College of Information and Computer Science, University of Massachusetts, Amherst
Justin Domke, College of Information and Computer Science, University of Massachusetts, Amherst

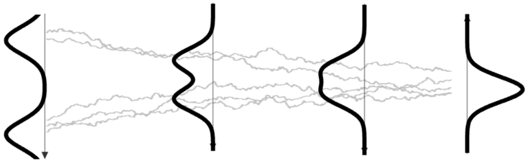
Bayesian Inference via Sparse Hamiltonian Flows

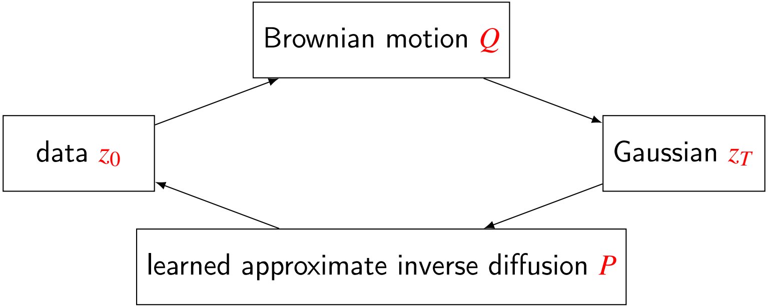
Naitong Chen, Zuheng Xu, Trevor Campbell
Department of Statistics, University of British Columbia

This talk

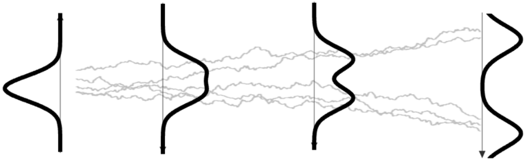
- What's the relationship?
- A unified view of these inference methods.

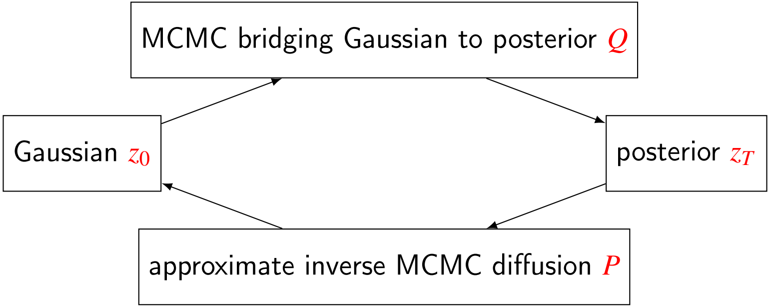
Diffusion models



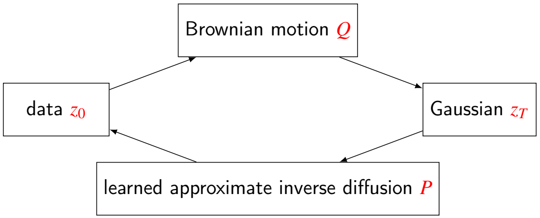


Diffusion-esque inference

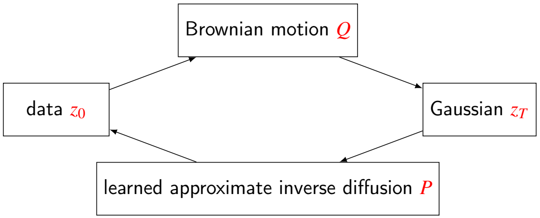




Diffusion models



Diffusion models



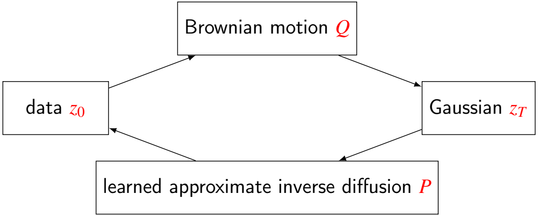
Q: sample z_0 from data, add noise until z_T

P: sample z_T from Gaussian, denoise until z_0

minimize $KL(Q\|P)$

$\nabla \log q_t(z)$ defines ideal denoising

Diffusion models



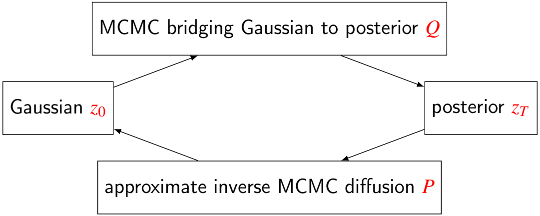
Q: sample z_0 from data, add noise until z_T $\quad$ $d\mathsf{z}_t = -\beta_t \mathsf{z}_t dt + \sqrt{2\beta_t}\, d\mathsf{w}_t$

P: sample z_T from Gaussian, denoise until z_0 $\quad$ $d\mathsf{z}_t = -\beta_t \mathsf{z}_t dt - 2\beta_t s_\theta(\mathsf{z}_t, t) dt + \sqrt{2\beta_t}\, d\bar{\mathsf{w}}_t$

minimize $KL(Q\|P)$ $\quad$ $KL(Q\|P) = KL(q_T\|p_T) + \mathbb{E}_{\mathsf{z}_T \sim q_T} KL(Q(\cdot|\mathsf{z}_T)\|P(\cdot|\mathsf{z}_T))$

$\nabla \log q_t(z)$ defines ideal denoising $\quad$ if $s_\theta(z,t) = \nabla \log q_t(z)$ then $KL(Q\|P) = KL(q_T\|p_T)$

Diffusion-esque inference



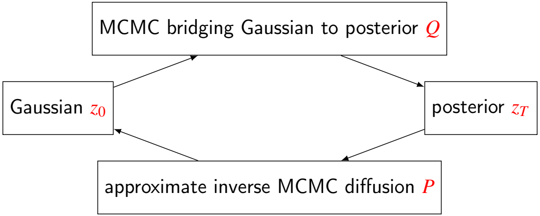
Q: sample z_0 from Gaussian, do Langevin on π_t
where $\pi_0 =$ (Gaussian) and $\pi_T =$ (posterior).

P: sample z_T from posterior, do inverse MCMC.

minimize $KL(Q \| P)$

$\nabla \log q_t(z)$ defines optimal inverse dynamics

Diffusion-esque inference



Q: sample z_0 from Gaussian, do Langevin on π_t $\quad dz_t = \nabla \log \pi_t(\mathrm{z}_t)dt + \sqrt{2}\, d\mathrm{w}_t$
where $\pi_0 =$ (Gaussian) and $\pi_T =$ (posterior).

P: sample z_T from posterior, do inverse MCMC. $\quad dz_t = \nabla \log \pi_t(\mathrm{z}_t)dt - 2s_\theta(\mathrm{z}_t, t)dt + \sqrt{2}d\bar{\mathrm{w}}_t$

minimize $KL(Q\|P)$ $\quad KL(Q\|P) = KL(q_T\|p_T) + \mathbb{E}_{\mathrm{z}_T \sim q_T} KL(Q(\cdot|\mathrm{z}_T)\|P(\cdot|\mathrm{z}_T))$

$\nabla \log q_t(z)$ defines optimal inverse dynamics $\quad$ if $s_\theta(z,t) = \nabla \log q_t(z)$ then $KL(Q\|P) = KL(q_T\|p_T)$

Comparison

	Diffusion models	**Diffusion-based VI**
z_0	data (samples)	Gaussian
z_T	Gaussian	posterior (distribution)

Comparison

	Diffusion models	Diffusion-based VI
z_0	data (samples)	Gaussian
z_T	Gaussian	posterior (distribution)
Q	noising	MCMC on bridging densities
P	learned denoising	inverse MCMC

Comparison

	Diffusion models	**Diffusion-based VI**
z_0	data (samples)	Gaussian
z_T	Gaussian	posterior (distribution)
Q	noising	MCMC on bridging densities
P	learned denoising	inverse MCMC
goal	good P to model z_0	good Q to model z_T

Comparison

	Diffusion models	Diffusion-based VI
z_0	data (samples)	Gaussian
z_T	Gaussian	posterior (distribution)
Q	noising	MCMC on bridging densities
P	learned denoising	inverse MCMC
goal	good P to model z_0	good Q to model z_T
marginals $q(z_t\|z_0)$ tractable	yes	no ☹
scores $\nabla \log q(z_t)$ tractable	no	no, but can approximate using bridging densities ☺

Variational Inference

$$p(z) = \bar{p}(z)/Z$$

$$\min_{q \in \mathsf{Family}} KL\left(q(\mathsf{z}) \| p(\mathsf{z})\right)$$

$$\max_{q \in \mathsf{Family}} \mathrm{ELBO}(q(\mathsf{z}) \| \bar{p}(\mathsf{z})) := \mathop{\mathbb{E}}_{q(\mathsf{z})} \log\left(\bar{p}(\mathsf{z})/q(\mathsf{z})\right)$$

Diffusion-based variational inference

- Q: $z_0 \sim q_0$, run MCMC diffusion on π_t $(q_0 = \pi_0 \rightsquigarrow \pi_T = p_T)$ until z_T.
- P: $z_T \sim p_T$, run reverse diffusion on π_t until z_0.
- Convert Q and P to discrete time.
- Optimize $KL(Q \| P) \geq KL(q_T \| p_T)$.

Diffusion-based variational inference

- Q: $z_0 \sim q_0$, run MCMC diffusion on π_t $(q_0 = \pi_0 \rightsquigarrow \pi_T = p_T)$ until z_T.
- P: $z_T \sim p_T$, run reverse diffusion on π_t until z_0.
- Convert Q and P to discrete time.
- Optimize $KL(Q\|P) \geq KL(q_T\|p_T)$.

Many instances (Wu et al 2020; Thin et al 2021; Geffner & D 2021, 2023; Zhang et al 2021; Doucet et al 2022).

Diffusion-based variational inference

- Q: $z_0 \sim q_0$, run MCMC diffusion on π_t ($q_0 = \pi_0 \rightsquigarrow \pi_T = p_T$) until z_T.
- P: $z_T \sim p_T$, run reverse diffusion on π_t until z_0.
- Convert Q and P to discrete time.
- Optimize $KL(Q\|P) \geq KL(q_T\|p_T)$.

Many instances (Wu et al 2020; Thin et al 2021; Geffner & D 2021, 2023; Zhang et al 2021; Doucet et al 2022).

Design choices:

- Starting distribution q_0 (Standard Gaussian? Learned Gaussian?)
- Bridging distributions π_t (Fixed? Learned?)
- Forward process Q (Langevin? Underdamped Langevin?)
- Backward process P (Fixed? Learned score network?)
- Numerical simulation of Q and P (Splitting? Euler-Maruyama?)
- Optimizer (SGD? Adam? Step sizes?)

Starting distribution

q_0 closer to $p_T \implies$ less distance for π_t to travel

Starting distribution

q_0 closer to $p_T \implies$ less distance for π_t to travel

$$
\begin{aligned}
q_0 &= \mathcal{N}(0, I) \\
q_0 &= \mathcal{N}(\mu, \text{diag}(\sigma^2)), \quad (\mu, \sigma^2) \text{ from Gaussian VI} \\
q_0 &= \mathcal{N}(\mu, \Sigma), \quad (\mu, \Sigma) \text{ from Gaussian VI} \\
q_0 &= \mathcal{N}(\mu, \Sigma), \quad (\mu, \Sigma) \text{ optimized as part of } Q
\end{aligned}
$$

Starting distribution

q_0 closer to $p_T \implies$ less distance for π_t to travel

$$
\begin{aligned}
q_0 &= \mathcal{N}(0, I) \\
q_0 &= \mathcal{N}(\mu, \text{diag}(\sigma^2)), \quad (\mu, \sigma^2) \text{ from Gaussian VI} \\
q_0 &= \mathcal{N}(\mu, \Sigma), \quad (\mu, \Sigma) \text{ from Gaussian VI} \\
q_0 &= \mathcal{N}(\mu, \Sigma), \quad (\mu, \Sigma) \text{ optimized as part of } Q
\end{aligned}
$$

Normalizing flows?

Bridging distributions

better path π_t between q_0 and $p_T \implies q_t$ closer to π_t

Bridging distributions

better path π_t between q_0 and $p_T \implies q_t$ closer to π_t

$$
\begin{aligned}
\bar{\pi}_t(z) &= q_0(z)^{1-\beta_t}\,\bar{p}(z)^{\beta_t}, \quad \beta_t \text{ fixed} \\
\bar{\pi}_t(z) &= q_0(z)^{1-\beta_t}\,\bar{p}(z)^{\beta_t}, \quad \beta_t \text{ optimized}
\end{aligned}
$$

Bridging distributions

better path π_t between q_0 and $p_T \implies q_t$ closer to π_t

$$
\begin{aligned}
\bar{\pi}_t(z) &= q_0(z)^{1-\beta_t}\,\bar{p}(z)^{\beta_t}, \quad \beta_t \text{ fixed} \\
\bar{\pi}_t(z) &= q_0(z)^{1-\beta_t}\,\bar{p}(z)^{\beta_t}, \quad \beta_t \text{ optimized}
\end{aligned}
$$

Something with more parameters?

Forward Process Q

better diffusion $\Longrightarrow q_t$ closer to π_t

overdamped Langevin

$$\begin{aligned} d\mathrm{z}_t &= \nabla \log \pi_t(\mathrm{z}_t) dt + \sqrt{2}\, d\mathrm{w}_t \\ q_t(z) &\approx \pi_t(z) \end{aligned}$$

Forward Process Q

better diffusion $\Longrightarrow$ q_t closer to π_t

overdamped Langevin

$$
\begin{aligned}
d\mathrm{z}_t &= \nabla \log \pi_t(\mathrm{z}_t)dt + \sqrt{2}\, d\mathrm{w}_t \\
q_t(z) &\approx \pi_t(z)
\end{aligned}
$$

underdamped Langevin

$$
\begin{aligned}
d\mathrm{z}_t &= \rho_t dt \\
d\rho_t &= \nabla \log \pi_t(\mathrm{z}_t)dt - \gamma\rho_t dt + \sqrt{2\gamma}\, d\mathrm{w}_t \\
q_t(z,\rho) &\approx \pi_t(z,\rho) = \pi_t(z)\mathcal{N}(\rho|0,I)
\end{aligned}
$$

Forward Process Q

better diffusion $\Longrightarrow$ q_t closer to π_t

overdamped Langevin

$$\begin{aligned} dz_t &= \nabla \log \pi_t(z_t)dt + \sqrt{2}\, dw_t \\ q_t(z) &\approx \pi_t(z) \end{aligned}$$

underdamped Langevin

$$\begin{aligned} dz_t &= \rho_t dt \\ d\rho_t &= \nabla \log \pi_t(z_t)dt - \gamma \rho_t dt + \sqrt{2\gamma}\, dw_t \\ q_t(z,\rho) &\approx \pi_t(z,\rho) = \pi_t(z)\mathcal{N}(\rho|0,I) \end{aligned}$$

...with mass-matrix

$$\begin{aligned} dz_t &= M^{-1}\rho_t dt \\ d\rho_t &= \nabla \log \pi_t(z_t)dt - \gamma M^{-1}\rho_t dt + \sqrt{2\gamma}\, dw_t \\ q_t(z,\rho) &\approx \pi_t(z,\rho) = \pi_t(z)\mathcal{N}(\rho|0,M) \end{aligned}$$

Forward Process Q

better diffusion $\Longrightarrow$ q_t closer to π_t

overdamped Langevin

$$\begin{aligned} dz_t &= \nabla \log \pi_t(z_t)dt + \sqrt{2}\, dw_t \\ q_t(z) &\approx \pi_t(z) \end{aligned}$$

underdamped Langevin

$$\begin{aligned} dz_t &= \rho_t dt \\ d\rho_t &= \nabla \log \pi_t(z_t)dt - \gamma\rho_t dt + \sqrt{2\gamma}\, dw_t \\ q_t(z,\rho) &\approx \pi_t(z,\rho) = \pi_t(z)\mathcal{N}(\rho|0,I) \end{aligned}$$

...with mass-matrix

$$\begin{aligned} dz_t &= M^{-1}\rho_t dt \\ d\rho_t &= \nabla \log \pi_t(z_t)dt - \gamma M^{-1}\rho_t dt + \sqrt{2\gamma}\, dw_t \\ q_t(z,\rho) &\approx \pi_t(z,\rho) = \pi_t(z)\mathcal{N}(\rho|0,M) \end{aligned}$$

Higher-order Langevin? Time-dependent momentum distribution?

Backward Process P (overdamped Langevin)

better $P \implies KL(Q\|P)$ closer to $KL(q_T\|p_T)$

$$d\mathsf{z}_t = \nabla \log \pi_t(\mathsf{z}_t)dt + \sqrt{2}\, d\mathsf{w}_t$$

Backward Process P (overdamped Langevin)

better $P \implies KL(Q\|P)$ closer to $KL(q_T\|p_T)$

$$dz_t = \nabla \log \pi_t(z_t)dt + \sqrt{2}\; dw_t$$

Ideal reversal P:

$$dz_t = \nabla \log \pi_t(z_t)dt - 2\nabla \log q_t(z_t)dt + \sqrt{2}\; d\bar{w}_t$$

Backward Process P (overdamped Langevin)

better $P \Longrightarrow KL(Q\|P)$ closer to $KL(q_T\|p_T)$

$$dz_t = \nabla \log \pi_t(z_t)dt + \sqrt{2}\, dw_t$$

Ideal reversal P:

$$dz_t = \nabla \log \pi_t(z_t)dt - 2\nabla \log q_t(z_t)dt + \sqrt{2}\, d\bar{w}_t$$

Simplest option:

$$dz_t = -\nabla \log \pi_t(z_t)dt + \sqrt{2}\, d\bar{w}_t$$

(Tight if $q_t = \pi_t$)

Backward Process P (overdamped Langevin)

better $P \implies KL(Q\|P)$ closer to $KL(q_T\|p_T)$

$$dz_t = \nabla \log \pi_t(z_t)dt + \sqrt{2}\, dw_t$$

Ideal reversal P:

$$dz_t = \nabla \log \pi_t(z_t)dt - 2\nabla \log q_t(z_t)dt + \sqrt{2}\, d\bar{w}_t$$

Simplest option:

$$dz_t = -\nabla \log \pi_t(z_t)dt + \sqrt{2}\, d\bar{w}_t \qquad \text{(Tight if } q_t = \pi_t\text{)}$$

Corrective score network:

$$dz_t = -\nabla \log \pi_t(z_t)dt + 2s_\theta(z_t, t) + \sqrt{2}\, d\bar{w}_t \qquad \text{(Tight if } s_\theta = \nabla \log \tfrac{\pi_t}{q_t}\text{)}$$

Backward Process P (underdamped Langevin)

$$\begin{aligned} d\mathsf{z}_t &= \rho_t dt \\ d\rho_t &= [\nabla \log \pi_t(z_t) - \gamma \rho_t]\, dt + \sqrt{2\gamma}\, d\mathsf{w}_t \end{aligned}$$

Backward Process P (underdamped Langevin)

$$
\begin{aligned}
d\mathsf{z}_t &= \rho_t dt \\
d\rho_t &= [\nabla \log \pi_t(z_t) - \gamma\rho_t]\, dt + \sqrt{2\gamma}\, d\mathsf{w}_t
\end{aligned}
$$

Ideal reversal P:

$$
\begin{aligned}
d\mathsf{z}_t &= \rho_t \; dt \\
d\rho_t &= \left[\nabla \log \pi_t(\mathsf{z}_t) - \gamma\rho_t - 2\gamma\nabla_\rho \log q_t(\rho_t, \mathsf{z}_t)\right] dt + \sqrt{2\gamma} d\bar{w}_t
\end{aligned}
$$

Backward Process P (underdamped Langevin)

$$\begin{aligned} dz_t &= \rho_t dt \\ d\rho_t &= [\nabla \log \pi_t(z_t) - \gamma \rho_t]\, dt + \sqrt{2\gamma}\, dw_t \end{aligned}$$

Ideal reversal P:

$$\begin{aligned} dz_t &= \rho_t\ dt \\ d\rho_t &= \left[\nabla \log \pi_t(z_t) - \gamma \rho_t - 2\gamma \nabla_\rho \log q_t(\rho_t, z_t)\right] dt + \sqrt{2\gamma} d\bar{w}_t \end{aligned}$$

Simplest option

(Tight if $q_t = \pi_t$, so $\nabla_\rho \log q_t = -\rho$)

$$\begin{aligned} dz_t &= \rho_t\ dt \\ d\rho_t &= [\nabla \log \pi_t(z_t) + \gamma \rho_t]\, dt + \sqrt{2\gamma} d\bar{w}_t \end{aligned}$$

Backward Process P (underdamped Langevin)

$$\begin{aligned} dz_t &= \rho_t dt \\ d\rho_t &= [\nabla \log \pi_t(z_t) - \gamma \rho_t] dt + \sqrt{2\gamma}\, dw_t \end{aligned}$$

Ideal reversal P:

$$\begin{aligned} dz_t &= \rho_t \; dt \\ d\rho_t &= \left[\nabla \log \pi_t(z_t) - \gamma \rho_t - 2\gamma \nabla_\rho \log q_t(\rho_t, z_t)\right] dt + \sqrt{2\gamma} d\bar{w}_t \end{aligned}$$

Simplest option

(Tight if $q_t = \pi_t$, so $\nabla_\rho \log q_t = -\rho$)

$$\begin{aligned} dz_t &= \rho_t \; dt \\ d\rho_t &= [\nabla \log \pi_t(z_t) + \gamma \rho_t] dt + \sqrt{2\gamma} d\bar{w}_t \end{aligned}$$

Corrective score network:

(Tight if $s_\theta = \nabla_\rho \log \frac{\pi_t}{q_t}$)

$$\begin{aligned} dz_t &= \rho_t \; dt \\ d\rho_t &= [\nabla \log \pi_t(z_t) + \gamma \rho_t + 2\gamma s_\theta(t, z_t, \rho_t)] dt + \sqrt{2\gamma} d\bar{w}_t \end{aligned}$$

Discretization

$$
\begin{aligned}
q(z_{1:K}) &= \underbrace{q(z_1)}_{\text{Gaussian}} \prod_{k=1}^{K-1} F_k(z_{k+1}|z_k) \\
\bar{p}(z_{1:K}) &= \underbrace{\bar{p}(z_K)}_{\text{Target}} \prod_{k=1}^{K-1} B_k(z_k|z_{k+1})
\end{aligned}
$$

$$
\min KL\left(q(\mathsf{z}_{1:K}) \| p(\mathsf{z}_{1:K})\right)
$$

Splitting

Forward SDE

$$\begin{bmatrix} dz_t \\ d\rho_t \end{bmatrix} = \begin{bmatrix} \rho_t dt \\ \nabla \log \pi_t(z_t)dt - \gamma\rho_t dt + \sqrt{2\gamma}\, dw_t \end{bmatrix}$$

Backward SDE

$$\begin{bmatrix} dz_t \\ d\rho_t \end{bmatrix} = \begin{bmatrix} \rho_t\ dt \\ [\nabla \log \pi_t(\rho_t)dt - \gamma\rho_t dt - 2\gamma s_\theta(t, z_t, \rho_t)]\, dt + \sqrt{2\gamma} d\bar{w}_t \end{bmatrix}$$

Splitting

Forward SDE

$$\begin{bmatrix} dz_t \\ d\rho_t \end{bmatrix} = \underbrace{\begin{bmatrix} \rho_t dt \\ 0 \end{bmatrix}}_{\mathtt{U}} + \underbrace{\begin{bmatrix} 0 \\ \nabla \log \pi_t(z_t) dt - \gamma \rho_t dt + \sqrt{2\gamma}\, dw_t \end{bmatrix}}_{\mathtt{V}}$$

Backward SDE

$$\begin{bmatrix} dz_t \\ d\rho_t \end{bmatrix} = \underbrace{\begin{bmatrix} \rho_t \; dt \\ 0 \end{bmatrix}}_{\mathtt{U}'} + \underbrace{\begin{bmatrix} 0 \\ [\nabla \log \pi_t(\rho_t) - \gamma \rho_t - 2\gamma s_\theta(t, z_t, \rho_t)]\, dt + \sqrt{2\gamma} d\bar{w}_t \end{bmatrix}}_{\mathtt{V}'}$$

Splitting

Forward SDE

$$\begin{bmatrix} dz_t \\ d\rho_t \end{bmatrix} = \underbrace{\begin{bmatrix} \rho_t dt \\ 0 \end{bmatrix}}_{\mathtt{U}} + \underbrace{\begin{bmatrix} 0 \\ \nabla \log \pi_t(z_t) dt - \gamma \rho_t dt + \sqrt{2\gamma}\, dw_t \end{bmatrix}}_{\mathtt{V}}$$

Backward SDE

$$\begin{bmatrix} dz_t \\ d\rho_t \end{bmatrix} = \underbrace{\begin{bmatrix} \rho_t \ dt \\ 0 \end{bmatrix}}_{\mathtt{U}'} + \underbrace{\begin{bmatrix} 0 \\ [\nabla \log \pi_t(\rho_t) - \gamma \rho_t - 2\gamma s_\theta(t, z_t, \rho_t)]\, dt + \sqrt{2\gamma} d\bar{w}_t \end{bmatrix}}_{\mathtt{V}'}$$

$F_k = \mathtt{V}\,\mathtt{U}$ ($\mathtt{U}$ exact, $\mathtt{V}$ Euler-Maruyama, both for time δ)
$B_k = \mathtt{U}'\mathtt{V}'$ ($\mathtt{U}'$ exact, $\mathtt{V}'$ Euler-Maruyama, both for time δ)

Better Splitting

Forward SDE

$$\begin{bmatrix} dz_t \\ d\rho_t \end{bmatrix} = \underbrace{\begin{bmatrix} \rho_t dt \\ 0 \end{bmatrix}}_{\mathtt{A}} + \underbrace{\begin{bmatrix} 0 \\ \nabla \log \pi_t(z_t) dt \end{bmatrix}}_{\mathtt{B}} + \underbrace{\begin{bmatrix} 0 \\ -\gamma \rho_t dt + \sqrt{2\gamma}\, dw_t \end{bmatrix}}_{\mathtt{O}}$$

Backward SDE

$$\begin{bmatrix} dz_t \\ d\rho_t \end{bmatrix} = \underbrace{\begin{bmatrix} \rho_t \ dt \\ 0 \end{bmatrix}}_{\mathtt{A}'} + \underbrace{\begin{bmatrix} 0 \\ \nabla \log \pi_t(\rho_t) dt \end{bmatrix}}_{\mathtt{B}'} + \underbrace{\begin{bmatrix} 0 \\ [-\gamma \rho_t - 2\gamma s_\theta(t, z_t, \rho_t)]\, dt + \sqrt{2\gamma} d\bar{w}_t \end{bmatrix}}_{\mathtt{O}'}$$

Better Splitting

Forward SDE

$$\begin{bmatrix} dz_t \\ d\rho_t \end{bmatrix} = \underbrace{\begin{bmatrix} \rho_t dt \\ 0 \end{bmatrix}}_{\mathtt{A}} + \underbrace{\begin{bmatrix} 0 \\ \nabla \log \pi_t(z_t) dt \end{bmatrix}}_{\mathtt{B}} + \underbrace{\begin{bmatrix} 0 \\ -\gamma \rho_t dt + \sqrt{2\gamma}\, dw_t \end{bmatrix}}_{\mathtt{O}}$$

Backward SDE

$$\begin{bmatrix} dz_t \\ d\rho_t \end{bmatrix} = \underbrace{\begin{bmatrix} \rho_t \ dt \\ 0 \end{bmatrix}}_{\mathtt{A}'} + \underbrace{\begin{bmatrix} 0 \\ \nabla \log \pi_t(\rho_t) dt \end{bmatrix}}_{\mathtt{B}'} + \underbrace{\begin{bmatrix} 0 \\ [-\gamma \rho_t - 2\gamma s_\theta(t, z_t, \rho_t)]\, dt + \sqrt{2\gamma} d\bar{w}_t \end{bmatrix}}_{\mathtt{O}'}$$

$F_k = \mathtt{O}\ \mathtt{B}\ \mathtt{A}\ \mathtt{B}$ ($\mathtt{A}$ exact, $\mathtt{B}$ Euler-Maruyama, $\mathtt{O}$ exact, $\mathtt{B}$ for $\delta/2$, others for δ)
$B_k = \mathtt{B}'\mathtt{A}'\mathtt{B}'\mathtt{O}'$ ($\mathtt{A}'$ exact, $\mathtt{B}$ Euler-Maruyama, $\mathtt{O}'$ Euler-Maruyama)

Algorithm

Algorithm 1 Forward transition $F_k(z_{k+1}, \rho_{k+1} | z_k, \rho_k)$

Require: z_k, ρ_k, step-size δ

Re-sample momentum $\rho'_k \sim m_F(\rho'_k | \rho_k, \gamma, \delta)$

Leapfrog step $\tau_{\text{LP}}(z_k, \rho'_k)$:

Update $\rho''_k = \rho'_k + \frac{\delta}{2} \nabla \log \pi^{k\delta}(z_k)$

Update $z_{k+1} = z_k + \delta \rho''_k$

Update $\rho_{k+1} = \rho''_k + \frac{\delta}{2} \nabla \log \pi^{k\delta}(z_{k+1})$

return (z_{k+1}, ρ_{k+1})

Algorithm 3 Generating the augmented ELBO (eq. (5)).

Sample $(z_1, \rho_1) \sim q(z_1, \rho_1)$.

Initialize estimator as $\mathcal{L} \leftarrow -\log q(z_1, \rho_1)$.

for $k = 1, 2, \cdots, K-1$ **do**

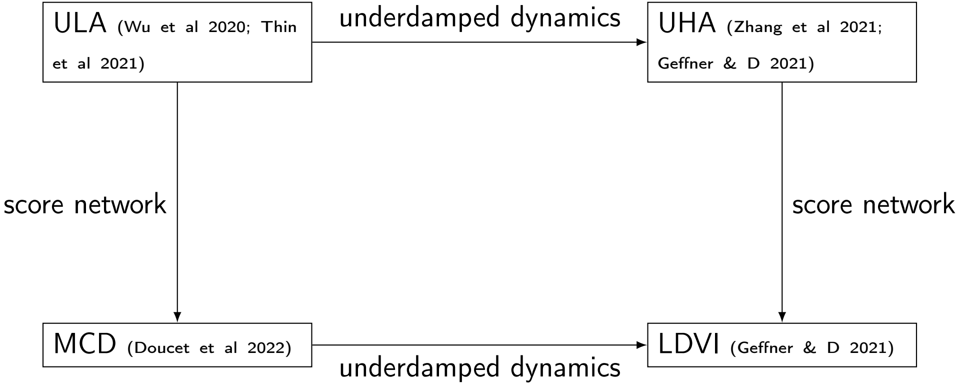
Run F_k (alg. 1) on (z_k, ρ_k), store $\rho'_k, z_{k+1}, \rho_{k+1}$.

Update $\mathcal{L} \leftarrow \mathcal{L} + \log \frac{m_B(\rho_k | \rho'_k, z_k, \gamma, \delta)}{m_F(\rho'_k | \rho_k, \gamma, \delta)}$.

Update $\mathcal{L} \leftarrow \mathcal{L} + \log \bar{p}(z_K, \rho_K)$.

return $\mathcal{L}$

Results


ULA (Wu et al 2020; Thin et al 2021)
underdamped dynamics
UHA (Zhang et al 2021; Geffner & D 2021)
score network
score network
MCD (Doucet et al 2022)
underdamped dynamics
LDVI (Geffner & D 2021)

Results

- Tune:
  - Initial distribution q_0 (diagonal Gaussian)
  - Discretization step-size
  - Bridging densities' parameters β
  - Damping coefficient
  - Score network (two hidden layers with residual connections)
- Train with $K \in \{8, 16, 32, 64, 128, 256\}$ bridging distributions.
- Optimize with Adam for 150k iters w/ learning rates of 10^{-3}, 10^{-4}, 10^{-5}, keep best.

Logistic Regression (1)

	Logistic regression (*Ionosphere*)					
	ULA	MCD	UHA	LDVI	$\mathrm{UHA}_{\mathrm{EM}}$	$\mathrm{LDVI}_{\mathrm{EM}}$
$K = 8$	-116.4	-114.6	-115.6	$\mathbf{-114.4}$	-117.7	-115.5
$K = 16$	-115.4	-113.6	-114.4	$\mathbf{-113.1}$	-115.9	-113.8
$K = 32$	-114.5	-112.9	-113.4	$\mathbf{-112.4}$	-114.6	-112.9
$K = 64$	-113.8	-112.5	-112.8	$\mathbf{-112.1}$	-113.6	-112.4
$K = 128$	-113.1	-112.2	-112.3	$\mathbf{-111.9}$	-113.1	-112.1
$K = 256$	-112.7	-112.1	-112.1	$\mathbf{-111.7}$	-112.5	-111.9

Plain VI: -124.1

ULA - Overdamped / MCD - Overdamped+score net / UHA - Underdamped / LDVI - Underdamped+score net

Logistic Regression (2)

	Logistic regression (*Sonar*)					
	ULA	MCD	UHA	LDVI	UHA_{EM}	LDVI_{EM}
$K = 8$	-122.4	-117.2	-120.1	$\mathbf{-116.3}$	-124.1	-118.5
$K = 16$	-119.9	-114.4	-116.8	$\mathbf{-112.6}$	-119.9	-114.4
$K = 32$	-117.4	-112.4	-113.9	$\mathbf{-110.6}$	-116.4	-111.7
$K = 64$	-115.3	-111.1	-111.9	$\mathbf{-109.7}$	-113.8	-110.3
$K = 128$	-113.5	-110.2	-110.6	$\mathbf{-109.1}$	-111.9	-109.6
$K = 256$	-112.1	-109.7	-109.7	$\mathbf{-108.9}$	-110.7	-109.1

Plain VI: -138.6

ULA - Overdamped / MCD - Overdamped+score net / UHA - Underdamped / LDVI - Underdamped+score net

Time Series (1)

	Brownian motion					
	ULA	MCD	UHA	LDVI	UHA_{EM}	LDVI_{EM}
$K=8$	-1.9	-1.4	-1.6	$\mathbf{-1.1}$	-2.8	-2.8
$K=16$	-1.5	-0.8	-1.1	$\mathbf{-0.5}$	-2.2	-1.4
$K=32$	-1.1	-0.4	-0.5	$\mathbf{0.1}$	-1.6	-0.5
$K=64$	-0.7	-0.1	0.1	$\mathbf{0.5}$	-0.9	0.1
$K=128$	-0.3	0.2	0.4	$\mathbf{0.7}$	-0.4	0.4
$K=256$	-0.1	0.5	0.6	$\mathbf{0.9}$	0.1	0.6

Plain VI: -4.4

ULA - Overdamped / MCD - Overdamped+score net / UHA - Underdamped / LDVI - Underdamped+score net

Time Series (2)

	Lorenz system					
	ULA	MCD	UHA	LDVI	UHA_{EM}	LDVI_{EM}
$K = 8$	-1168.2	-1168.1	-1166.3	$-\mathbf{1166.1}$	-1170.5	-1170.5
$K = 16$	-1165.7	-1165.6	-1163.1	$-\mathbf{1162.2}$	-1169.8	-1166.8
$K = 32$	-1163.2	-1163.3	-1160.3	$-\mathbf{1157.6}$	-1167.9	-1162.9
$K = 64$	-1160.9	-1161.1	-1157.7	$-\mathbf{1153.7}$	-1161.3	-1161.4
$K = 128$	-1158.9	-1158.9	-1155.4	$-\mathbf{1153.1}$	-1158.1	-1163.4
$K = 256$	-1157.2	-1157.1	-1153.3	$-\mathbf{1151.1}$	-1163.1	-1154.6

Plain VI: -1187.8

ULA - Overdamped / MCD - Overdamped+score net / UHA - Underdamped / LDVI - Underdamped+score net

Hierarchical

	Random effect regression (seeds)					
	ULA	MCD	UHA	LDVI	UHA_{EM}	LDVI_{EM}
$K = 8$	-75.5	-75.1	$\mathbf{-74.9}$	$\mathbf{-74.9}$	-75.9	-75.5
$K = 16$	-75.2	-74.6	-74.6	$\mathbf{-74.5}$	-75.1	-75.1
$K = 32$	-74.9	-74.3	$\mathbf{-74.2}$	$\mathbf{-74.2}$	-74.8	-74.8
$K = 64$	-74.6	-74.1	-74.1	$\mathbf{-73.9}$	-74.4	-74.4
$K = 128$	-74.3	-73.9	-73.8	$\mathbf{-73.7}$	-74.1	-74.1
$K = 256$	-74.1	-73.7	-73.7	$\mathbf{-73.6}$	-73.9	-73.7

Plain VI: -77.1

ULA - Overdamped / MCD - Overdamped+score net / UHA - Underdamped / LDVI - Underdamped+score net

Conclusions

Compared to diffusion models...

- Harder since marginals of Q not tractable
- Easier because have a good guess for score network

Experimentally...

- Underdamped dynamics help
- Learning a score network helps
- Better discretization helps
- Tuning more stuff helps

There's a lot we still don't understand...

Conclusions

Compared to diffusion models...

- Harder since marginals of Q not tractable
- Easier because have a good guess for score network

Experimentally...

- Underdamped dynamics help
- Learning a score network helps
- Better discretization helps
- Tuning more stuff helps

There's a lot we still don't understand...

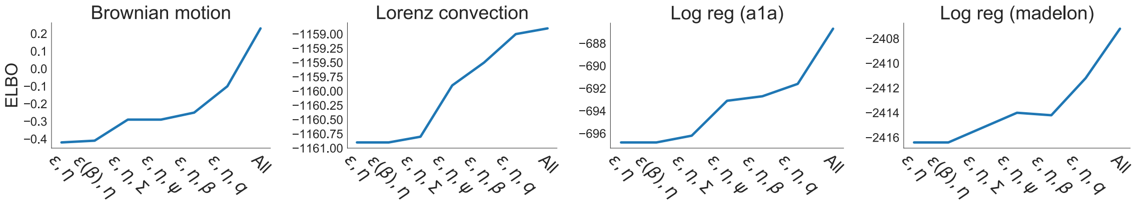
Thank you!

Joint work with Tomas Geffner

Langevin Diffusion Variational Inference
AISTATS 2023, arXiv:2208.07743

these slides: `t.ly/9vZvk` or
`people.cs.umass.edu/domke/diffusion.pdf`

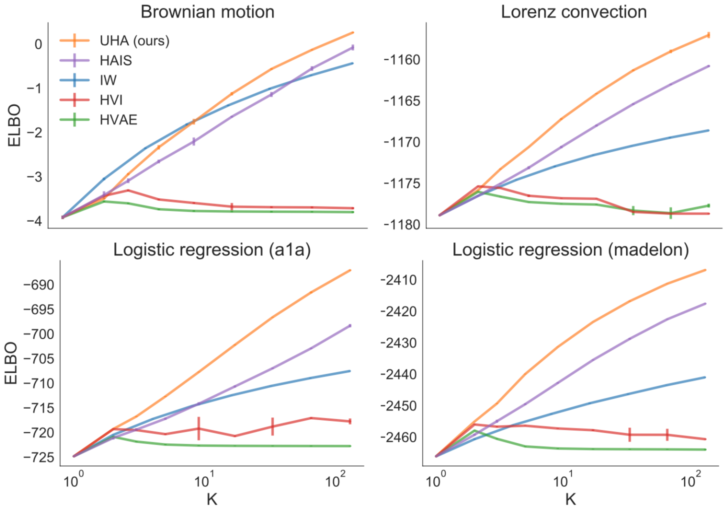
Optimizing more stuff helps



(K = 64)

ε – step size of HMC dynamics
η – damping coefficient
Σ – moment covariance
β – temperature schedule
ψ – "full rank" temperature schedule
q – initial distribution

Better than importance-weighted VI



UHA	Our algorithm
HAIS	Annealed Importance Sampling using HMC dynamics
IW	Importance Weighting
HVI	“Bridging the gap” using HMC dynamics
HVAE	(Recent algorithm)

VAEs

ELBO on test set

		$K=1$	$K=8$	$K=16$	$K=32$	$K=64$
mnist	UHA	-93.4	-89.8	-88.8	-88.1	-87.6
	IW	-93.4	-90.5	-89.9	-89.4	-89.0
letters	UHA	-137.9	-133.5	-132.3	-131.5	-130.9
	IW	-137.9	-134.6	-133.9	-133.2	-132.7
kmnist	UHA	-184.2	-176.6	-174.6	-173.2	-171.6
	IW	-184.2	-179.7	-178.7	-177.8	-177.0

log-likelihood on test set

		$K=1$	$K=8$	$K=16$	$K=32$	$K=64$
mnist	UHA	-88.5	-87.5	-87.2	-87.0	-86.9
	IW	-88.5	-87.6	-87.5	-87.3	-87.2
letters	UHA	-131.9	-130.7	-130.3	-130.1	-129.9
	IW	-131.9	-130.9	-130.7	-130.6	-130.4
kmnist	UHA	-174.3	-172.2	-171.6	-171.2	-170.2
	IW	-174.3	-173.0	-172.6	-172.4	-172.2

Looseness

Finite time decomposition:

$$\underbrace{KL(q(z_{1:K}) \| p(z_{1:K}))}_{\text{what you optimize}} = \underbrace{KL(q(z_K) \| p(z_K))}_{\text{what you care about}} + \underbrace{KL(q(z_{1:K-1}|\mathsf{z}_K) \| p(z_{1:K-1}|\mathsf{z}_K))}_{\text{looseness}}$$

Analogous to continuous time decomposition:

$$\underbrace{KL(Q \| P)}_{\text{what you optimize}} = \underbrace{KL(q_T \| p_T)}_{\text{what you care about}} + \underbrace{\mathbb{E}_{\mathsf{z}_T \sim q_T} KL(Q(\cdot|\mathsf{z}_T) \| P(\cdot|\mathsf{z}_T))}_{\text{looseness}}$$

Ideal (intractable) transitions

$$B_k(z_k|z_{k+1}) = F_k(z_{k+1}|z_k) \frac{q(z_k)}{q(z_{k+1})}$$

would give $KL(q(z_{1:K}) \| p(z_{1:K})) = KL(q(z_K) \| p(z_K))$.

Approximating these transitions analogous to approximating score function

SDEs

$$d\mathsf{z}_t = f(\mathsf{z}_t, t)dt + g(t)d\mathsf{w}_t$$

$$\begin{aligned} \mathsf{z}_{t+\delta} &= \mathsf{z}_t + \delta f(\mathsf{z}_t) + g(t)\varepsilon_t \\ \varepsilon_0, \varepsilon_\delta, \cdots \varepsilon_T &\sim \mathcal{N}(0, \delta) \end{aligned}$$

$$d\mathsf{z}_t = f(\mathsf{z}_t, t)dt + g(t)d\bar{\mathsf{w}}_t$$

$$\begin{aligned} \mathsf{z}_{t-\delta} &= \mathsf{z}_t - \delta f(\mathsf{z}_t) + g(t)\varepsilon_t \\ \varepsilon_T, \varepsilon_{T-\delta}, \cdots, \varepsilon_0, \cdots &\sim \mathcal{N}(0, \delta) \end{aligned}$$