STAT238: Bayesian Statistics May 13th, 2025

Mini Project # 10: Variational Inference

Reece D. Huff

Prompts

Bayesian inference is typically performed via approximate posterior sampling. However, there are many situations where sampling is inefficient, especially if we want to perform high-dimensional inference. Variational inference substitutes sampling with optimization. Instead of drawing samples from the posterior (or an approximation to the posterior produced by running an MCMC scheme), variational inference searches for the closest approximation to the target posterior among all distributions in an (often infinite-dimensional) family of tractable distributions. Please read:

1. Blei, David M., Alp Kucukelbir, and Jon D. McAuliffe. "*Variational inference: A review for statisticians*." Journal of the American statistical Association 112, no. 518 (2017): 859-877. [BKM17]
2. Liu, Qiang, and Dilin Wang. "*Stein variational gradient descent: A general purpose Bayesian inference algorithm*." Advances in neural information processing systems 29 (2016). [LW16]

Then write a 4-5 page essay that summarizes and comments on your reading. It should, at minimum, address the discussion prompts outlined below. For the second paper it may help to read:

- Papamakarios, George, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. "*Normalizing flows for probabilistic modeling and inference*." Journal of Machine Learning Research 22, no. 57 (2021): 1-64. [PNR⁺21]
- Wang, Yiwei, Jiuhai Chen, Chun Liu, and Lulu Kang. "*Particle-based energetic variational inference*." Statistics and Computing 31 (2021): 1-17. [WCLK21]

(a) Compare variational inference methods and MCMC procedures. Identify pros and cons for each framework, and propose at least one example setting where you prefer each over the other.

(b) Variational inference typically adopts the KL divergence between the variational distribution, q, and the target as its objective, p_*. Explain why we adopt $D_{\mathrm{KL}}(q||p_*)$ rather than $D_{\mathrm{KL}}(p_*||q)$. Explain how your interpretation of the objective would (or would not) change, if you exchanged the order of the distributions inside the KL divergence. What biases do you expect this to induce in the variational solution?

(c) Clearly summarize the CAVI algorithm for mean-field inference in conditionally conjugate models. In particular, show that each step in the iterative procedure is a descent step on the KL divergence. Try to relate the procedure to another method from the course (e.g. coordinate ascent, expectation-maximization).

(d) Discuss the biases induced by the mean-field assumption, and, given these biases, how you would use the variational distribution produced by optimizing over the mean field class.

(e) Clearly summarize the SVGD algorithm for mean-field inference. Then, answer the following questions:

- What is the variational family that SVGD optimizes over?[1]
- Give an intuitive description of the particle dynamics specified by the SVGD algorithm. Compare this algorithm to another ensemble algorithm from the class. What component of the algorithm ensures that the particles spread out? What is responsible for the variance in samples in related MCMC methods?
- Are you convinced by the "median-trick" for selecting the kernel bandwidth?[2] Propose an alternative procedure for adaptively selecting the bandwidth.

(f) SVGD, like other particle flow based methods, uses a flow map to transport particles drawn from a reference distribution (often normal). The transport map is chosen so that, after the transformation, the original distribution is as close as possible to the target. This is implemented by at each step moving a set of reference particles as if they obeyed a time inhomogeneous ODE. The vector field specifying the ODE is chosen to instantaneously decrease the KL as quickly as possible. Show that the functional gradient of the KL divergence with respect to a velocity field transporting $q(t)$ to $q(t+dt)$ can be calculated using only computationally available quantities (the unnormalized target, log densities, partial derivatives of log densities).

[1] This is a tricky question. It may help to start by asking, how would I sample from the distribution produced at the end of SVGD? The variational family will be all distributions that can be specified by this sampling procedure.

[2] Be skeptical here.

Notation

Let $\mathbf{z} = [z_1, \dots, z_m]$ represent the unknown or "latent" variables of interest. Let $\mathbf{x} = [x_1, \dots, x_n]$ be the known, observable variables. Suppose we are given a joint model $p(\mathbf{x}, \mathbf{z})$ such that $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x} \mid \mathbf{z})\, p(\mathbf{z})$ where $p(\mathbf{x} \mid \mathbf{z})$ represents the likelihood, and $p(\mathbf{z})$ represents the prior distribution. Let $p(\mathbf{z} \mid \mathbf{x}) = \tilde{p}(\mathbf{z} \mid \mathbf{x}) / Z(\mathbf{x})$ represent the posterior where $\tilde{p}(\mathbf{z} \mid \mathbf{x}) = p(\mathbf{x} \mid \mathbf{z})\, p(\mathbf{z})$ and $Z(\mathbf{x}) = \int_{\text{all } \mathbf{z}} \tilde{p}(\mathbf{z} \mid \mathbf{x})\, \mathrm{d}\mathbf{z}$ is the troublesome normalizing constant.

(a) Variational Inference vs. Markov Chain Monte Carlo

A fundamental challenge in Bayesian statistics is approximating, computing, and sampling from challenging distributions. We begin by discussing two methods for approximating a target distribution, $p_*(\mathbf{z})$.

Markov Chain Monte Carlo. In Markov Chain Monte Carlo (MCMC), we approximate the target density p_* by defining a Markov chain whose stationary distribution is the target. Specifically, we simulate an *ergodic*[3] chain that obeys *detailed balance*[4] for long enough to approach stationarity. We then approximate the target with an empirical estimate constructed from (a subset of) the collected samples.

Variational Inference. In variational inference, we approximate the target density p_* by selecting the optimal distribution q^* from a *variational family* $\mathcal{Q}$. Specifically, we define a set of densities $q(\mathbf{z}) \in \mathcal{Q}$ meant to approximate the target distribution. The optimal density q^* is a member of $\mathcal{Q}$ that minimizes the Kullback-Leibler (KL) divergence to the target,

$$q^*(\mathbf{z}) := \underset{q(\mathbf{z}) \in \mathcal{Q}}{\operatorname{argmin}} \left\{ \mathtt{KL}\Big(q(\mathbf{z}) \,\Big\|\, p_*(\mathbf{z})\Big) \right\} \quad \text{where} \quad \mathtt{KL}\Big(q(\mathbf{z}) \,\Big\|\, p_*(\mathbf{z})\Big) := \underset{\mathbf{z} \sim q(\cdot)}{\mathbb{E}} \left[\log\left(\frac{q(\mathbf{z})}{p_*(\mathbf{z})} \right) \right] = \int_{\text{all } \mathbf{z}} q(\mathbf{z}) \, \log\left(\frac{q(\mathbf{z})}{p_*(\mathbf{z})} \right) \mathrm{d}\mathbf{z} \tag{VI}$$

In contrast with MCMC, we cast our approximation of the target distribution as an optimization problem. Additionally, we have a proper density q^* as an approximation of the target, rather than empirical estimates constructed from samples.

Comparing variational inference and MCMC. Variational inference and MCMC are both powerful techniques for density approximation. When should a statistician use one or the other? MCMC methods are generally more computationally expensive as it can take an extremely long time to approach stationarity. With this expense, MCMC is able to provide guarantees of producing *exact* samples from the target density (asymptotically). While variational inference does not enjoy such guarantees, it is generally much faster than MCMC (assuming a reasonable variational family). Therefore, variational inference is generally better suited for large datasets where speed is a higher priority than accuracy. We may prefer MCMC, for example, in a medical setting. Here, accuracy (e.g., in the tails) is of utmost importance, and therefore we are willing to collect many samples over years and years to be confident that our model is capable of making inferences with high accuracy. On the other hand, we may prefer variational inference if we, for example, are designing a recommendation algorithm at Netflix. Here, we have access to an enormous dataset and desire a model that is fast and does well in expectation and therefore are willing to sacrifice a bit of accuracy for speed.

(b) Why use $\mathtt{KL}(q \,\|\, p_*)$ instead of $\mathtt{KL}(p_* \,\|\, q)$ for variational inference?

A primary reason for utilizing $\mathtt{KL}(q \,\|\, p_*)$ instead of $\mathtt{KL}(p_* \,\|\, q)$ is that we often only have access to the unnormalized target density $\tilde{p}_*$. In this case, the variational inference objective $\mathtt{KL}(q \,\|\, p_*)$ is equivalent to minimizing $\mathtt{KL}(q \,\|\, \tilde{p}_*)$ as

$$\mathtt{KL}\Big(q(\mathbf{z}) \,\Big\|\, p_*(\mathbf{z})\Big) = \underset{\mathbf{z} \sim q(\cdot)}{\mathbb{E}} \left[\log\left(\frac{q(\mathbf{z})}{p_*(\mathbf{z})} \right) \right] = \underset{\mathbf{z} \sim q(\cdot)}{\mathbb{E}} \Big[\log q(\mathbf{z}) - \log \tilde{p}_*(\mathbf{z}) + \log(Z) \Big] = \mathtt{KL}\Big(q(\mathbf{z}) \,\Big\|\, \tilde{p}_*(\mathbf{z})\Big) + \text{constant}$$

On the other hand, the alternative objective requires an expectation over the target as $\mathtt{KL}(p_* \,\|\, q) = \mathbb{E}_{\mathbf{z} \sim p_*(\cdot)}[\log({p_*(\mathbf{z})}/{q(\mathbf{z})})]$. To evaluate this expectation, we would have to calculate the troublesome normalizing constant which is often intractable. We could approximate the expectation with sampling (i.e., importance sampling) but this makes our optimization problem stochastic which is undesirable as two runs may result in different solutions.

Understanding the biases induced by both objectives is best seen through example. Suppose our target distribution is a two-dimensional Gaussian with a non-zero correlation between z_1 and z_2 (Figure 1). Suppose $q_1(\mathbf{z})$ and $q_2(\mathbf{z})$ on the left and right of Figure 1 represent optimal densities under objectives $\mathtt{KL}(q \,\|\, p_*)$ and $\mathtt{KL}(p_* \,\|\, q)$, respectively. To see why, consider the green points $\mathbf{z}_A$ and $\mathbf{z}_B$ in Figure 1:

[3] a chain that is both *irreducible* (it's possible to get from any state to any other state) and *aperiodic* (there's no fixed cycle or period in the transitions between states; the chain doesn't get stuck in a repeating pattern)

[4] $T(\mathbf{z} \to \mathbf{z}')\, p_S(\mathbf{z}) = T(\mathbf{z}' \to \mathbf{z})\, p_S(\mathbf{z}')$ for all $\mathbf{z}, \mathbf{z}'$

- Under variational objective $\mathtt{KL}(q\,||\,p_*)$: Notice that $\mathbf{z}_A$ plays a negligible role in $\mathtt{KL}(q_1\,||\,p_*)$ as $q_1(\mathbf{z}_A) \approx 0$, whereas $\mathbf{z}_A$ plays a significant role in $\mathtt{KL}(q_2\,||\,p_*)$ as $q_2(\mathbf{z}_A)$ is non-negligible and $\log(q_2(\mathbf{z}_A)/p_*(\mathbf{z}_A)) \gg 1$. Therefore, we select $q_1(\mathbf{z})$ instead of $q_2(\mathbf{z})$ under the variational objective.
- Under alternative objective $\mathtt{KL}(p_*\,||\,q)$: Notice that $\mathbf{z}_B$ plays a negligible role in $\mathtt{KL}(p_*\,||\,q_2)$ as $q_2(\mathbf{z}_B) \approx p_*(\mathbf{z}_B) \implies \log(p_*(\mathbf{z}_B)/q_2(\mathbf{z}_B)) \approx 0$, whereas $\mathbf{z}_B$ plays a significant role in $\mathtt{KL}(p_*\,||\,q_1)$ as $\log(p_*(\mathbf{z}_B)/q_1(\mathbf{z}_B)) \gg 1$. Therefore, we select $q_2(\mathbf{z})$ instead of $q_1(\mathbf{z})$ under the alternative objective.

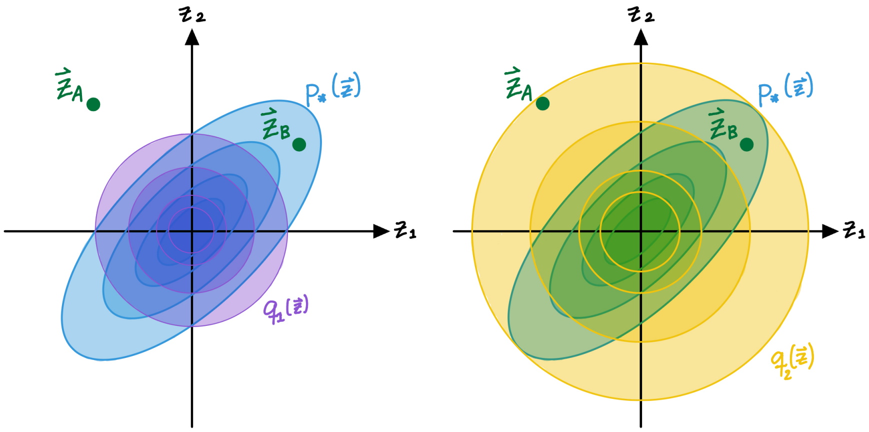

Figure 1: Example target two-dimensional Gaussian with a non-zero correlation between z_1 and z_2. Here, $q_1(\mathbf{z})$ on the left and $q_2(\mathbf{z})$ on the right represent optimal densities under objectives $\mathtt{KL}(q\,||\,p_*)$ and $\mathtt{KL}(p_*\,||\,q)$, respectively.

This example highlights a key behavior of each objective. Minimizing $\mathtt{KL}(q\,||\,p_*)$ tends to choose $q(z)$ that underestimates uncertainty whereas minimizing $\mathtt{KL}(p_*\,||\,q)$ will tend to lead to more conservative approximations that overestimate uncertainty, potentially becoming broader or heavier-tailed. In this sense, the variational inference objective is primarily focused on the mode while the alternative objective is focused on coverage. Note that $q_1(\mathbf{z})$ and $q_2(\mathbf{z})$ are results we would expect when optimizing over the *mean-field variational family* (as discussed in (c) and (d)), but our analysis is without loss of generality: the variational inference objective is biased towards the mode(s) while the alternative objective is biased towards coverage. Finally, while our example considers a unimodal target both objectives could in theory capture multi-modal distributions.

Variational inference to Bayesian models

From this point onward, we apply variational inference to a Bayesian model and take our target distribution p_* as the posterior, $p_*(\mathbf{z}) = p(\mathbf{z} \mid \mathbf{x}) \propto p(\mathbf{x} \mid \mathbf{z})\, p(\mathbf{z})$. In this setting, the variational inference objective is

$$\mathtt{KL}\Big(q(\mathbf{z})\,\Big\|\,p(\mathbf{z} \mid \mathbf{x})\Big) := \mathop{\mathbb{E}}_{\mathbf{z}\sim q(\cdot)}\left[\log\left(\frac{q(\mathbf{z})}{p(\mathbf{z} \mid \mathbf{x})}\right)\right] \overset{(5)}{=} \mathtt{KL}\Big(q(\mathbf{z})\,\Big\|\,p(\mathbf{z})\Big) - \mathop{\mathbb{E}}_{\mathbf{z}\sim q(\cdot)}\Big[\log\big(p(\mathbf{x} \mid \mathbf{z})\big)\Big] + \log\big(p(\mathbf{x})\big). \tag{1}$$

We note that the log evidence $\log\big(p(\mathbf{x})\big)$ is constant with respect to q so we can remove it from the objective in Equation (VI). It follows that minimizing $\mathtt{KL}\big(q(\mathbf{z})\,||\,p(\mathbf{z} \mid \mathbf{x})\big)$ is equivalent to maximizing the evidence lower bound, $\mathrm{ELBO}\big(q(\mathbf{z})\big)$[6]

$$q^*(\mathbf{z}) = \mathop{\mathrm{argmax}}_{q(\mathbf{z})\in\mathcal{Q}} \Big\{\mathrm{ELBO}\big(q(\mathbf{z})\big)\Big\} \quad \text{where} \quad \mathrm{ELBO}\big(q(\mathbf{z})\big) := \mathbb{E}_{\mathbf{z}\sim q(\cdot)}\Big[\log\big(p(\mathbf{x} \mid \mathbf{z})\big)\Big] - \mathtt{KL}\Big(q(\mathbf{z})\,\Big\|\,p(\mathbf{z})\Big) \tag{ELBO}$$

Writing the variational inference objective in terms of $\mathrm{ELBO}\big(q(\mathbf{z})\big)$ clarifies its meaning. We see that variational inference is

- maximizing the likelihood that explain the observed data in the first term $\mathbb{E}_{\mathbf{z}\sim q(\cdot)}\Big[\log\big(p(\mathbf{x} \mid \mathbf{z})\big)\Big]$
- minimizing the divergence between the approximate density and the prior in $p(\mathbf{z})$ in the second term $-\mathtt{KL}\Big(q(\mathbf{z})\,\Big\|\,p(\mathbf{z})\Big)$

[5] $\mathtt{KL}\Big(q(\mathbf{z})\,\Big\|\,p(\mathbf{z} \mid \mathbf{x})\Big) = \mathbb{E}_{\mathbf{z}\sim q(\cdot)}\Big[\log\big(q(\mathbf{z})\big)\Big] - \mathbb{E}_{\mathbf{z}\sim q(\cdot)}\Big[\log\big(p(\mathbf{z} \mid \mathbf{x})\big)\Big] = \mathbb{E}_{\mathbf{z}\sim q(\cdot)}\Big[\log\big(q(\mathbf{z})\big)\Big] - \mathbb{E}_{\mathbf{z}\sim q(\cdot)}\Big[\log\big(p(\mathbf{x} \mid \mathbf{z})\big)\Big] - \mathbb{E}_{\mathbf{z}\sim q(\cdot)}\Big[\log\big(p(\mathbf{z})\big)\Big] + \mathbb{E}_{\mathbf{z}\sim q(\cdot)}\Big[\log\big(p(\mathbf{x})\big)\Big]$.

[6] Note that the nomenclature "evidence lower bound" is literal. By Equations (1) and (ELBO) and the fact that KL divergence is non-negative, we have

$$\mathtt{KL}\Big(q(\mathbf{z})\,\Big\|\,p(\mathbf{z} \mid \mathbf{x})\Big) = \mathtt{KL}\Big(q(\mathbf{z})\,\Big\|\,p(\mathbf{z})\Big) - \mathop{\mathbb{E}}_{\mathbf{z}\sim q(\cdot)}\Big[\log\big(p(\mathbf{x} \mid \mathbf{z})\big)\Big] + \log\big(p(\mathbf{x})\big) = \log\big(p(\mathbf{x})\big) - \mathrm{ELBO}\big(q(\mathbf{z})\big) \geq 0 \quad \Longrightarrow \quad \mathrm{ELBO}\big(q(\mathbf{z})\big) \leq \log\big(p(\mathbf{x})\big)$$

(c) Coordinate ascent variational inference (CAVI) algorithm in conditionally conjugate models

Coordinate ascent variational inference (CAVI) algorithm for mean-field inference is a powerful approach for solving Equation (VI). CAVI optimizes over the *mean-field variational family* where the latent variables are mutually independent and each governed by a distinct density. Each member of the family has the form $q(\mathbf{z}) = \prod_{j=1}^{m} q_j(z_j)$. In other words, we ignore possible correlations between variables and therefore approximate the target with a product of independent *variational factors*. CAVI works by updating variational factor $q_j(z_j)$ while holding all other variational factors constant.

We begin by deriving a single step of the CAVI algorithm. Let z_j and $q_j(z_j)$ represent the j-th latent variable and variational factor, respectively, and let $\mathbf{z}_{-j}$ and $q_{-j}(\mathbf{z}_{-j}) = \prod_{\ell \neq j} q_\ell(z_\ell)$ represent the remaining latent variables and variational factors, respectively. At each step, we update $q_j(z_j)$ while holding $q_{-j}(\mathbf{z}_{-j})$ fixed, and therefore we decompose the KL divergence in Equation (VI) into

$$\mathrm{KL}\Big(q(\mathbf{z}) \,\Big\|\, p(\mathbf{z} \mid \mathbf{x})\Big) \overset{(7)}{=} \mathbb{E}_{z_j \sim q_j(\cdot)}\left[\log q_j(z_j) - \mathbb{E}_{\mathbf{z}_{-j} \sim q_{-j}(\cdot)}\left[\log p(z_j \mid \mathbf{z}_{-j}, \mathbf{x})\right]\right] + \mathrm{KL}\Big(q_{-j}(\mathbf{z}_{-j}) \,\Big\|\, p(\mathbf{z}_{-j} \mid \mathbf{x})\Big).$$

where $p(z_j \mid \mathbf{z}_{-j}, \mathbf{x})$ is the *complete conditional*, i.e., the density of the j-th latent variable z_j conditioned on all the other latent variables $\mathbf{z}_{-j}$ and the observable $\mathbf{x}$. Since we are updating $q_j(z_j)$ while holding $q_{-j}(\mathbf{z}_{-j})$ constant, we minimize the expression above by setting the terms within the expectation to zero. We have that

$$\log q_j(z_j) - \mathbb{E}_{-j}\left[\log p(z_j \mid \mathbf{z}_{-j}, \mathbf{x})\right] = 0 \quad \Longrightarrow \quad q_j^*(z_j) \propto \exp\left\{\mathbb{E}_{-j}\left[\log p(z_j \mid \mathbf{z}_{-j}, \mathbf{x})\right]\right\} \tag{CAVI}$$

Since $q_j^*(z_j)$ zeros the first term the CAVI step is in fact a descent step on the KL divergence. This step is the essence of the CAVI algorithm. A full description of the CAVI algorithm is provided below and highlights that at each iteration, we loop through the m variational factors and (i.) calculate the complete conditional (Line 3) and (ii.) use it to update variational factor $q_j(z_j)$ (Line 4). We repeat until the ELBO has converged (i.e., ELBO hasn't changed much between iterations).

```
Algorithm 1: Coordinate Ascent Variational Inference (CAVI) [BKM17]
  Input: Joint model $p(\mathbf{x}, \mathbf{z})$ over latent variable (unknown) $\mathbf{z}$ and observable dataset $\mathbf{x}$
  Output: A mean-field variational density $q(\mathbf{z}) = \prod_{j=1}^{m} q_j(z_j)$
  Initialize: Variational factors $q_j(z_j)$
1 while the ELBO has not converged do
2     for $j \in \{1, \ldots, m\}$ do
3         Calculate complete conditional $p(z_j \mid \mathbf{z}_{-j}, \mathbf{x})$
4         Set $q_j(z_j) \propto \exp\{\mathbb{E}_{-j}[\log p(z_j \mid \mathbf{z}_{-j}, \mathbf{x})]\}$ via Equation (CAVI)
5     end
6     Calculate $\mathrm{ELBO}(q(\mathbf{z})) := \mathbb{E}_{\mathbf{z} \sim q(\cdot)}[\log p(\mathbf{x} \mid \mathbf{z})] - \mathrm{KL}(q(\mathbf{z}) \| p(\mathbf{z}))$ via Equation (ELBO)
7 end
8 return $q(\mathbf{z})$
```

Complete Conditionals in the Exponential Family

We consider a class of Bayesian models that are well-suited for CAVI: models with complete conditionals $p(z_j \mid \mathbf{z}_{-j}, \mathbf{x})$ belonging to the exponential family where we have

$$p(z_j \mid \mathbf{z}_{-j}, \mathbf{x}) = \frac{\tilde{p}(z_j \mid \mathbf{z}_{-j}, \mathbf{x})}{Z(\mathbf{z}_{-j}, \mathbf{x})} \quad \text{where} \quad \tilde{p}(z_j \mid \mathbf{z}_{-j}, \mathbf{x}) \propto h(z_j) \exp\left\{\eta_j(\mathbf{z}_{-j}, \mathbf{x})^\top z_j\right\} \quad \text{and} \quad Z(\mathbf{z}_{-j}, \mathbf{x}) := \int_{\text{all } z_j} \tilde{p}(z_j \mid \mathbf{z}_{-j}, \mathbf{x})\, \mathrm{d}z_j$$

where $h(z_j)$ is a base measure, $\eta_j(\mathbf{z}_{-j}, \mathbf{x})$ is the natural parameter, z_j is its own sufficient statistic, and $Z(\mathbf{z}_{-j}, \mathbf{x})$ is the normalizing constant. When the complete conditional is in the exponential family, the CAVI step is

$$q_j^*(z_j) \propto \exp\left\{\mathbb{E}\left[\log p(z_j \mid \mathbf{z}_{-j}, \mathbf{x})\right]\right\} = \exp\left\{\mathbb{E}\left[\log h(z_j) + \eta_j(\mathbf{z}_{-j}, \mathbf{x})^\top z_j - \log Z(\mathbf{z}_{-j}, \mathbf{x})\right]\right\} \propto h(z_j) \exp\left\{\mathbb{E}_{-j}\left[\eta_j(\mathbf{z}_{-j}, \mathbf{x})\right]^\top z_j\right\}$$

The expression above reveals the simplicity of the CAVI update when the complete conditional is in the exponential family: $q_j(z_j)$ has the same parametric form as $p(z_j \mid \mathbf{z}_{-j}, \mathbf{x})$ and therefore the CAVI step is simply updating its natural parameter to $\mathbb{E}_{-j}\left[\eta_j(\mathbf{z}_{-j}, \mathbf{x})\right]$.

[7] $\mathrm{KL}\big(q \,\|\, p\big) = \mathbb{E}_j\left[\mathbb{E}_{-j}\left[\log q_j(z_j) + \log q_{-j}(\mathbf{z}_{-j}) - \log p(z_j \mid \mathbf{z}_{-j}, \mathbf{x}) - \log p(\mathbf{z}_{-j} \mid \mathbf{x})\right]\right] = \mathbb{E}_j\left[\log q_j(z_j) - \mathbb{E}_{-j}\left[\log p(z_j \mid \mathbf{z}_{-j}, \mathbf{x})\right]\right] + \mathrm{KL}\big(q_{-j}(\mathbf{z}_{-j}) \,\|\, p(\mathbf{z}_{-j} \mid \mathbf{x})\big)$

Conditional Conjugacy and Bayesian Models

An important case of exponential family models are *conditionally conjugate models* with local and global variables. Let $\vec{\beta} \in \mathbb{R}^K$ be a vector of *global latent variables* and $\mathbf{z}$ be a vector of *local latent variables*. The global latent variables govern all of the data, while local latent variables govern the i-th component of the observable $\mathbf{x}$. Figure 2 depicts the joint model $p(\vec{\beta}, \mathbf{z}, \mathbf{x})$:

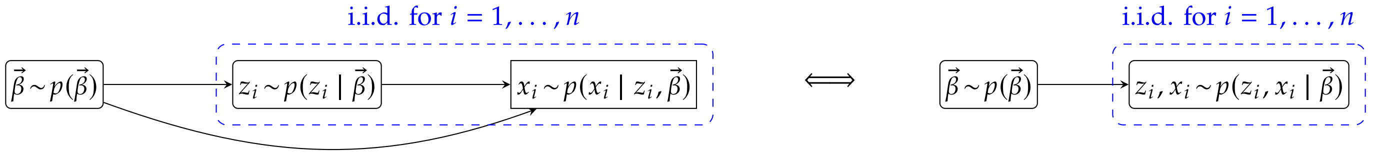
i.i.d. for $i = 1, \ldots, n$
$\vec{\beta} \sim p(\vec{\beta})$
$z_i \sim p(z_i \mid \vec{\beta})$
$x_i \sim p(x_i \mid z_i, \vec{\beta})$
$\Longleftrightarrow$
i.i.d. for $i = 1, \ldots, n$
$\vec{\beta} \sim p(\vec{\beta})$
$z_i, x_i \sim p(z_i, x_i \mid \vec{\beta})$

Figure 2: Hierarchical model $p(\vec{\beta}, \mathbf{z}, \mathbf{x})$ over global and latent variables $\vec{\beta}$, $\mathbf{z}$ and observable $\mathbf{x}$.

Under the model, we define a generic member of the mean-field variational family $q(\vec{\beta}, \mathbf{z}) \in \mathcal{Q}$ as $q(\vec{\beta}, \mathbf{z}) = q(\vec{\beta}) \prod_{i=1}^n q_i(z_i)$ such that we update $q(\vec{\beta})$ and all $q_i(z_i)$ at each CAVI step. We do so by calculating their respective complete conditionals.

CAVI step for $q(\vec{\beta})$. Consider the CAVI step for $q(\vec{\beta})$, where we hold local variables $\mathbf{z}$ and variational factors $\prod_{i=1}^n q_i(z_i)$ constant. The CAVI step for $q(\vec{\beta})$ is of the form

$$q(\vec{\beta}) \propto \exp\left\{\mathbb{E}_{\mathbf{z} \sim \prod_{i=1}^n q_i(z_i)}\left[\log p(\vec{\beta} \mid \mathbf{z}, \mathbf{x})\right]\right\} \quad \text{where} \quad p(\vec{\beta} \mid \mathbf{z}, \mathbf{x}) \propto p(\vec{\beta}, \mathbf{z}, \mathbf{x}) = p(\vec{\beta})\, p(\mathbf{z}, \mathbf{x} \mid \vec{\beta}) = p(\vec{\beta}) \prod_{i=1}^n p(z_i, x_i \mid \vec{\beta})$$

where $p(z_i, x_i \mid \vec{\beta})$ belongs to the exponential family such that

$$p(z_i, x_i \mid \vec{\beta}) = \frac{\tilde{p}(z_i, x_i \mid \vec{\beta})}{Z(\vec{\beta})} \quad \text{where} \quad \tilde{p}(z_i, x_i \mid \vec{\beta}) \propto h(z_i, x_i) \exp\left\{\vec{\beta}^\top t(z_i, x_i)\right\} \quad \text{and} \quad Z(\vec{\beta}) := \int_{\text{all } z_i, x_i} \tilde{p}(z_i, x_i \mid \vec{\beta})\, \mathrm{d}z_i\, \mathrm{d}x_i \tag{2}$$

where $h(z_i, x_i)$ is a base measure, $\vec{\beta}$ is the natural parameter, $t(z_i, x_i)$ is the sufficient statistic, and $Z(\vec{\beta})$ is the normalizing constant. We select the conjugate prior for the global latent variables as[8]

$$p(\vec{\beta}) = p(\vec{\beta}; \vec{\alpha}) = \frac{h(\vec{\beta})}{Z(\vec{\alpha})} \exp\left\{\left[\vec{\alpha}_{1:K}, \alpha_0\right]^\top \left[\vec{\beta}, -\log Z(\vec{\beta})\right]\right\} \text{ where } Z(\vec{\alpha}) := \int_{\text{all } \vec{\beta}} h(\vec{\beta}) \exp\left\{\vec{\alpha}_{1:K}^\top \vec{\beta} - \alpha_0 \log Z(\vec{\beta})\right\} \mathrm{d}\vec{\beta}$$

where $\vec{\alpha} = \left[\vec{\alpha}_{1:K}, \alpha_0\right] \in \mathbb{R}^{K+1}$ are the natural parameters and $t(\vec{\beta}) = \left[\vec{\beta}, -\log Z(\vec{\beta})\right] \in \mathbb{R}^{K+1}$ are the sufficient statistics. By selecting a conjugate prior, the complete conditional $p(\vec{\beta} \mid \mathbf{z}, \mathbf{x})$ is given by

$$p(\vec{\beta} \mid \mathbf{z}, \mathbf{x}) = p(\vec{\beta}) \prod_{i=1}^n p(z_i, x_i \mid \vec{\beta}) \propto h(\vec{\beta}) \exp\left\{\left[\vec{\alpha}_{1:K} + \sum_{i=1}^n t(x_i, z_i)\ ,\ \alpha_0 + n\right]^\top \left[\vec{\beta}, -\log Z(\vec{\beta})\right]\right\}$$

implying that the CAVI step for $q(\vec{\beta})$ is given by

$$\boxed{q(\vec{\beta}) \propto \exp\left\{\mathbb{E}_{\mathbf{z} \sim \prod_{i=1}^n q_i(z_i)}\left[\log p(\vec{\beta} \mid \mathbf{z}, \mathbf{x})\right]\right\} \propto h(\vec{\beta}) \exp\left\{\vec{\lambda}^\top t(\vec{\beta})\right\} \quad \text{where} \quad \vec{\lambda} := \left[\vec{\alpha}_{1:K} + \sum_{i=1}^n \mathbb{E}_{z_i \sim q_i(z_i)}\left[t(x_i, z_i)\right]\ ,\ \alpha_0 + n\right]} \tag{3}$$

CAVI step for $q_i(z_i)$. Consider the CAVI step on $q_i(z_i)$, where we hold all other local variables $\mathbf{z}_{-i}$ and global latent variable $\vec{\beta}$ constant. We note that local latent variables are i.i.d. and independent of all other data such that $p(z_i \mid \mathbf{z}_{-i}, x_i, \mathbf{x}_{-i}, \vec{\beta}) = p(z_i \mid x_i, \vec{\beta})$. Therefore the CAVI step is of the form

$$q_i(z_i) \propto \exp\left\{\mathbb{E}_{\vec{\beta} \sim q(\vec{\beta})}\left[\log p(z_i \mid x_i, \vec{\beta})\right]\right\} \quad \text{where} \quad p(z_i \mid x_i, \vec{\beta}) \propto p(z_i, x_i \mid \vec{\beta})$$

[8] See bayesian estimation for conjugate distributions with $\boldsymbol{\chi} = \vec{\alpha}_{1:K}$, $\nu = \alpha_0$, $A(\boldsymbol{\eta}) = \log(Z(\vec{\alpha}_{1:K}, \alpha_0))$, and $f(\boldsymbol{\chi}, \nu) = \frac{h(\vec{\beta})}{Z(\vec{\alpha})}$.

where we assume $p(z_i \mid x_i, \vec{\beta})$ belongs to an exponential family such that[9]

$$p(z_i \mid x_i, \vec{\beta}) = \frac{\tilde{p}(z_i \mid x_i, \vec{\beta})}{Z(\eta_i(x_i, \vec{\beta}))} \quad \text{where} \quad \tilde{p}(z_i \mid x_i, \vec{\beta}) \propto h(z_i) \exp\left\{\eta_i(x_i, \vec{\beta})^\top z_i\right\} \quad \text{and} \quad Z(\eta_i(x_i, \vec{\beta})) := \int_{\text{all } z_i} \tilde{p}(z_i \mid x_i, \vec{\beta}) \, \mathrm{d}z_i$$

where $h(z_i)$ is a base measure, $\eta_i(x_i, \vec{\beta})$ is the natural parameter, z_i is its own sufficient statistic, and $Z(\eta_i(x_i, \vec{\beta}))$ is the normalizing constant. Therefore, the CAVI step for $q_i(z_i)$ is given by

$$\boxed{q_i(z_i) \propto \exp\left\{\mathbb{E}_{\vec{\beta} \sim q(\vec{\beta})}\left[\log p(z_i \mid x_i, \vec{\beta})\right]\right\} \propto \exp\left\{\varphi_i^\top z_i - \mathbb{E}_{\vec{\beta} \sim q(\vec{\beta})}\left[\log Z(\eta_i(x_i, \vec{\beta}))\right]\right\} \quad \text{where} \quad \varphi_i := \mathbb{E}_{\vec{\beta} \sim q(\vec{\beta})}\left[\eta_i(x_i, \vec{\beta})\right]} \tag{4}$$

ELBO for conditionally conjugate models. The final step in CAVI is calculating the evidence lower bound as it defines our stopping criteria. For conditionally conjugate models, we have that

$$\mathrm{ELBO}\big(q(\vec{\beta}, \mathbf{z})\big) \overset{(10)}{=} \mathbb{E}_{\vec{\beta}, \mathbf{z} \sim q(\vec{\beta}, \mathbf{z})}\left[\log p(\vec{\beta}, \mathbf{z}, \mathbf{x}) - \log q(\vec{\beta}, \mathbf{z})\right]$$

$$\boxed{\mathrm{ELBO}\big(q(\vec{\beta}, \mathbf{z})\big) = \left(\vec{\alpha}_{1:K} + \sum_{i=1}^{n} \mathbb{E}_{z_i \sim q_i(z_i)}\left[t(x_i, z_i)\right]\right)^\top \mathbb{E}_{\vec{\beta} \sim q(\vec{\beta})}\left[\vec{\beta}\right] - \mathbb{E}_{\vec{\beta} \sim q(\vec{\beta})}\left[\log Z(\vec{\beta})\right](\alpha_0 + n) - \mathbb{E}_{\vec{\beta}, \mathbf{z} \sim q(\vec{\beta}, \mathbf{z})}\left[\log q(\vec{\beta}, \mathbf{z})\right]} \tag{5}$$

where

$$\mathbb{E}_{\vec{\beta}, \mathbf{z} \sim q(\vec{\beta}, \mathbf{z})}\left[\log q(\vec{\beta}, \mathbf{z})\right] = \vec{\lambda}^\top \mathbb{E}_{\vec{\beta} \sim q(\vec{\beta})}\left[t(\vec{\beta})\right] - \log Z(\vec{\lambda}) + \varphi_i^\top \mathbb{E}_{z_i \sim q(z_i)}\left[z_i\right] - \log Z(\varphi_i)$$

where $Z(\vec{\lambda}) := \int_{\text{all } \vec{\beta}} h(\vec{\beta}) \exp\left\{\vec{\lambda}^\top t(\vec{\beta})\right\} \mathrm{d}\vec{\beta}$ and $Z(\varphi_i) := \int_{\text{all } z_i} h(z_i) \exp\left\{\varphi_i^\top z_i\right\} \mathrm{d}z_i$.

We now have all the pieces to implement CAVI for conditionally conjugate models. The CAVI algorithm is provided below and highlights that at each iteration, we (i.) update the global variational factor $q(\vec{\beta})$ (Line 2) and then (ii.) loop through the n variational factors and update the local variational factors $q_i(z_i)$ (Line 3). We repeat until the ELBO has converged (i.e., ELBO hasn't changed much between iterations).

Algorithm 2: Coordinate Ascent Variational Inference (CAVI) in Conditionally Conjugate Bayesian Models [BKM17]

```
Input: Joint model $p(\vec{\beta}, \mathbf{x}, \mathbf{z})$ over global and latent variables $\vec{\beta}$, $\mathbf{z}$ and observable $\mathbf{x}$ (Figure 2)
Output: A mean-field variational density $q(\vec{\beta}, \mathbf{z}) = q(\vec{\beta}) \prod_{i=1}^{n} q_i(z_i)$
Initialize: Variational factors $q(\vec{\beta})$ and $q_i(z_i)$
1 while the ELBO has not converged do
2     Update global variational factor $q(\vec{\beta})$ by setting its natural parameters to $\left[\vec{\alpha}_{1:K} + \sum_{i=1}^{n} \mathbb{E}_{q_i}\left[t(x_i, z_i)\right], \alpha_0 + n\right]$ via (3)
      for $i \in \{1, \dots, n\}$ do
3         Update local variational factor $q_i(z_i)$ by setting its natural parameters to $\mathbb{E}_{q(\vec{\beta})}\left[\eta_i(x_i, \vec{\beta})\right]$ via (4)
4     end
5     Calculate $\mathrm{ELBO}\big(q(\vec{\beta}, \mathbf{z})\big)$ via (5)
6 end
7 return $q(\vec{\beta}, \mathbf{z})$
```

Relation to another algorithm in the class

The primary similarity of CAVI to other algorithms in the class is how it leverages conjugacy. Examples from the class include coordinate ascent, expectation maximization, and Gibbs sampling. All of these approaches are well suited for situations where conditioning on subsets of the latent variables may speed the optimization/sampling process. Such settings include conditionally conjugate models where conditioning a subsets of latent variables (e.g., complete conditional) results in a simple and fast update (e.g., updating the natural parameters of an exponential family). We argued in class that the speed ups from simple updates may make up for having to iterate over all of the latent variables. For example, Gibbs sampling is

[9] Note that by assuming that the local likelihood $p(z_i, x_i \mid \vec{\beta})$ belongs to the exponential family, we have that $p(z_i \mid x_i, \vec{\beta})$ is in an exponential family as $p(z_i \mid x_i, \vec{\beta}) \propto p(z_i, x_i \mid \vec{\beta})$

[10] $\mathrm{KL}\left(q(\vec{\beta}, \mathbf{z}) \,\|\, p(\vec{\beta}, \mathbf{z} \mid \mathbf{x})\right) = \mathbb{E}_{q(\vec{\beta}, \mathbf{z})}\left[\log\left(\frac{q(\vec{\beta}, \mathbf{z})}{p(\vec{\beta}, \mathbf{z} \mid \mathbf{x})}\right)\right] = \mathbb{E}_{q(\vec{\beta}, \mathbf{z})}\left[\log q(\vec{\beta}, \mathbf{z}) - \log p(\vec{\beta}, \mathbf{z}, \mathbf{x}) - \log p(\mathbf{x})\right] \implies \mathrm{ELBO}\big(q(\vec{\beta}, \mathbf{z})\big) = -\mathrm{KL}\left(q(\vec{\beta}, \mathbf{z}) \,\|\, p(\vec{\beta}, \mathbf{z} \mid \mathbf{x})\right) - \log p(\mathbf{x})$

a procedure in which we sample along complete conditionals of the latent, and while we constrained our step to be along z_i, it may speed mixing times especially in settings when the target distribution is multi-modal and does not have strongly correlated subsets (e.g., $\mathrm{Corr}(z_i, z_j) < 0.1$ for all $i \neq j$). Similarly, CAVI is well-suited for targets that do not have strongly correlated subsets as CAVI optimizes over the mean-field variational family.

(d) Biases induced by the mean-field assumption

To better understand the biases introduced by the mean-field assumption, let us revisit Figure 1 discussed in (b). The optimal $q^*(\mathbf{z}) = q_1(\mathbf{z})$ on the left is the result of optimizing over the mean-field variational family. We see that the resulting $q^*(\mathbf{z})$ is accurate, especially about the mode and marginal densities of the latent variables (i.e., along z_1 and z_2). However, it cannot capture correlation between the latents by construction. Thus the variances along the direction $z_1 + z_2$ are underestimated. This is a common theme in variational inference: the variational objective $\mathtt{KL}(q \,||\, p_*)$ tends to underestimate the variance of the target and fails to capture the wide tails of the target distribution.

With this in mind, when is it appropriate to utilize the mean-field approximation given its biases? The mean-field approximation is most appropriate when the target distribution the latents are independent. In this case, the variational family is able to capture the mode and marginal densities of the latents with high accuracy. When the target has weak to moderate correlations between latents (e.g., $\mathrm{Corr}(z_i, z_j) < 0.3$ for all $i \neq j$), the mean-field approximation can still be appropriate as long as the analysis is focused on the mode(s) and/or the latent marginals (i.e., along z_1 and z_2). In the case of strong correlations, the mean-field approximation is only appropriate for mode capture, and anything beyond that should be interpreted with caution as variational factors will collapse to a local modes as the correlation goes to 1.

(e) + (f) Stein variational gradient descent (SVGD)

Background. Let $\mathbf{z} \in \mathcal{Z} \subseteq \mathbb{R}^m$ and let $\kappa(\mathbf{z}, \mathbf{z}') : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ be a positive definite kernel. The reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ of $\kappa(\mathbf{z}, \mathbf{z}')$ comprises the linear combinations $\mathcal{H} = \left\{ f \;:\; f(\mathbf{z}) = \sum_{i=1}^{N} a_i \kappa(\mathbf{z}, \mathbf{z}_i),\; a_i \in \mathbb{R},\; N \in \mathbb{N},\; \mathbf{z}_i \in \mathcal{Z} \right\}$, endowed with an inner product $\langle f, g \rangle_{\mathcal{H}} = \sum_{i=1}^{N} \sum_{j=1}^{N} a_i b_j \kappa(\mathbf{z}_i, \mathbf{z}_j)$. Kernel $\kappa(\mathbf{z}, \cdot\,)$ in $\mathcal{H}$ satisfies the important "reproducing" property,

$$f(\mathbf{z}) = \langle f, \kappa(\,\cdot\,, \mathbf{z}) \rangle_{\mathcal{H}} = \langle \kappa(\mathbf{z}, \cdot\,), f \rangle_{\mathcal{H}} \quad \text{and} \quad \kappa(\mathbf{z}, \mathbf{z}') = \langle \kappa(\,\cdot\,, \mathbf{z}), \kappa(\,\cdot\,, \mathbf{z}') \rangle_{\mathcal{H}}$$

Let $\mathcal{H}^d$ represent the vector space $\mathbf{f} = [f_1, \ldots, f_d]^\top$ with $f_i \in \mathcal{H}$, equipped with inner product $\langle \mathbf{f}, \mathbf{g} \rangle_{\mathcal{H}^d} = \sum_{i=1}^{d} \langle f_i, g_i \rangle_{\mathcal{H}}$. More information about RKHS is provided in Appendix A.

Intuitive derivation of SVGD. For Stein variational gradient descent (SVGD), we consider a variational family of transformed random variables. Let $\mathbf{z} = \boldsymbol{\chi}(\mathbf{z}_0)$ where $\boldsymbol{\chi} : \mathcal{Z}_0 \to \mathcal{Z}$ represent the mapping from a tractable *base distribution* (e.g., Normal) $\mathbf{z}_0 \sim q_0(\mathbf{z}_0)$ to the target distribution $\mathbf{z} \sim p_*(\mathbf{z})$. We define the flow map as $\boldsymbol{\chi}(\mathbf{z}_0) = \mathbf{z}_0 + \boldsymbol{v}(\mathbf{z}_0)$ where $\boldsymbol{v}$ represents the *displacement field* (Figure 3). We note that the displacement field $\boldsymbol{v}$ is unknown and therefore we consider an infinitesimal flow $\mathbf{z}_{t+dt} = \boldsymbol{\psi}_t(\mathbf{z}_t) = \mathbf{z}_t + \varepsilon_t \boldsymbol{\omega}(\mathbf{z}_t)$. We view the infinitesimal displacement $\boldsymbol{\omega}(\mathbf{z}_t)$ as an ε_t-step that maps particle $\mathbf{z}_t$ to $\mathbf{z}_{t+dt}$. As result of this motion, the density q_t evolves to q_{t+dt}. Therefore, our goal is put the particles into a motion that drives the approximate density q towards the target p_*. Specifically, we set the infinitesimal displacement $\boldsymbol{\omega}(\mathbf{z}_t)$ such that our variational density approaches the target as quickly as possible. To illustrate this point, we select a transport map to move the reference particles as if they obeyed a time inhomogeneous ODE as we have

$$\mathbf{z}_{t+dt} = \mathbf{z}_t + \varepsilon_t \boldsymbol{\omega}(\mathbf{z}_t) \quad \Longrightarrow \quad \lim_{\varepsilon_t \to 0} \left\{ \frac{\mathbf{z}_{t+dt} - \mathbf{z}_t}{\varepsilon_t} \right\} = \boldsymbol{\omega}(\mathbf{z}_t) \quad \Longrightarrow \quad \frac{\partial \mathbf{z}}{\partial t} = \mathbf{v}(\mathbf{z}, t) = \boldsymbol{\omega}(\mathbf{z}_t)$$

where we see that the infinitesimal displacement is the instantaneous velocity of the particle $\mathbf{v}(\mathbf{z}, t)$.

We now deriving the optimal flow $\boldsymbol{\omega}^*$ under SVGD. While there a several ways to do so (see Appendix A.), an intuitive method is *functional gradient descent* on the KL divergence between the approximation distribution q and the target p_*. We treat $\mathtt{KL}(q \| p_*)$ as a *functional* of the displacement field, $F[\boldsymbol{v}]$, and perform functional gradient descent. Much like canonical gradient descent where one iteratively calculates the gradient of an objective *function* to move through a *vector* space and ultimately arrive at a minimum of that *function*, functional gradient descent works by iteratively calculating the gradient of a *functional* to move through a function space and ultimately arrive at a minimum of that *functional*. Here, the function space is the space of possible velocity fields applied to the particles and given that our functional $F[\boldsymbol{v}]$ is the $\mathtt{KL}(q \| p_*)$ under $\boldsymbol{v}$, our procedure will minimize the KL divergence at each step $\mathtt{KL}(q_{t+dt} \,||\, p_*) \leq \mathtt{KL}(q_t \,||\, p_*)$. We have that

$$\mathbf{z}^{(t+1)} \longleftarrow \mathbf{z}^{(t)} - \varepsilon_t \left(\nabla_{\boldsymbol{v}} F[\boldsymbol{v}] \right) \Big|_{\boldsymbol{v}=\mathbf{0}} \quad \Longrightarrow \quad \mathbf{z}^{(t+1)} \longleftarrow \mathbf{z}^{(t)} + \varepsilon_t \; \boldsymbol{\omega}^*(\mathbf{z}^{(t)}) \tag{6}$$

$$\boldsymbol{\omega}^*(\cdot) = \mathop{\mathbb{E}}_{\mathbf{z} \sim q(\cdot)} \left[\mathcal{A}_{p_*} \kappa(\mathbf{z}, \cdot) \right] \quad \text{where } \mathcal{A}_{p_*} \text{ is the Stein operator} \quad \mathcal{A}_{p_*} \kappa(\mathbf{z}, \cdot) := \kappa(\mathbf{z}, \cdot)\, \nabla_{\mathbf{z}} \log p_*(\mathbf{z}) + \nabla_{\mathbf{z}} \kappa(\mathbf{z}, \cdot) \tag{7}$$

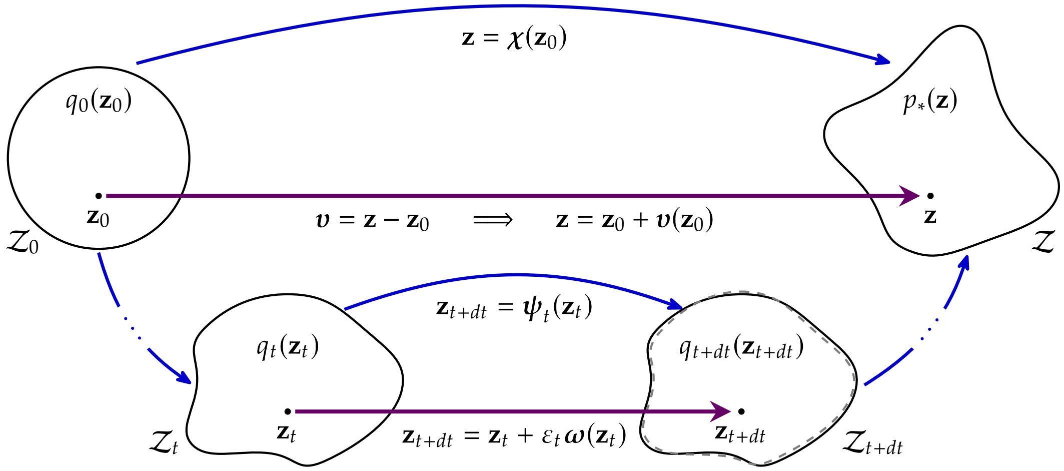
$\mathbf{z} = \chi(\mathbf{z}_0)$
$q_0(\mathbf{z}_0)$
$p_*(\mathbf{z})$
$\mathbf{z}_0$
$\boldsymbol{v} = \mathbf{z} - \mathbf{z}_0 \implies \mathbf{z} = \mathbf{z}_0 + \boldsymbol{v}(\mathbf{z}_0)$
$\mathbf{z}$
$\mathcal{Z}_0$
$\mathcal{Z}$
$\mathbf{z}_{t+dt} = \boldsymbol{\psi}_t(\mathbf{z}_t)$
$q_t(\mathbf{z}_t)$
$q_{t+dt}(\mathbf{z}_{t+dt})$
$\mathbf{z}_t$
$\mathbf{z}_{t+dt} = \mathbf{z}_t + \varepsilon_t\,\boldsymbol{\omega}(\mathbf{z}_t)$
$\mathbf{z}_{t+dt}$
$\mathcal{Z}_t$
$\mathcal{Z}_{t+dt}$

Figure 3: Flow map $\chi : \mathcal{Z}_0 \to \mathcal{Z}$ and infinitesimal flow $\boldsymbol{\psi}_t : \mathcal{Z}_t \to \mathcal{Z}_{t+dt}$. The dashed gray line indicates the infinitesimal flow $\boldsymbol{\psi}_t$ at time t.

and ε_t is the learning rate at t. A full derivation is provided in Appendix B. We note that we are able to calculate the Stein operator as the κ is predefined, and we have access to the unnormalized target $\tilde{p}$ such that $\nabla_{\mathbf{z}} \log p_*(\mathbf{z}) = \nabla_{\mathbf{z}} \log \tilde{p}_*(\mathbf{z})$. We approximate the expectation over q with sample averages by defining a set of N particles $\{\mathbf{z}_i\}_{i=1}^N$ to approximate the expectation

$$\boldsymbol{\omega}^*(\cdot) \approx \hat{\boldsymbol{\omega}}^*(\cdot) := \frac{1}{N}\sum_{j=1}^{N}\left[\kappa(\mathbf{z}_j, \cdot)\,\nabla_{\mathbf{z}_j} \log \tilde{p}_*(\mathbf{z}_j) + \nabla_{\mathbf{z}_j}\kappa(\mathbf{z}_j, \cdot)\right] \tag{SVGD}$$

Therefore, SVGD is a ensemble method that iteratively applies the (estimated) optimal flow $\boldsymbol{\omega}^*$ to a finite set of particles $\{\mathbf{z}_i\}_{i=1}^N$. After undergoing T iterations, we have a set of particles $\{\mathbf{z}_i^{(T)}\}_{i=1}^N$ that approximate the target distribution $p_*(\mathbf{z})$. Algorithm 3 provides a complete summary SVGD:

Algorithm 3: Stein Variational Gradient Descent (SVGD) [LW16]

```
Input: Target distribution $p_*(\mathbf{z})$ over latent variable $\mathbf{z}$ and set of $N$ initial particles $\{\mathbf{z}_i^{(0)}\}_{i=1}^N$ where $\mathbf{z}_i^{(0)} \overset{\text{i.i.d.}}{\sim} q_0(\mathbf{z})$
Output: Set of particles $\{\mathbf{z}_i^{(T)}\}_{i=1}^N$ that approximates the target distribution $p_*(\mathbf{z})$
for $t \in \{0, 1, \ldots, T-1\}$ do
    Take SVGD step $\mathbf{z}_i^{(t+1)} \leftarrow \mathbf{z}_i^{(t)} + \varepsilon_t \hat{\boldsymbol{\omega}}^*(\mathbf{z}_i^{(t)})$   where   $\hat{\boldsymbol{\omega}}^*(\mathbf{z}) := \frac{1}{N}\sum_{j=1}^{N}\left[\kappa(\mathbf{z}_j^{(t)}, \mathbf{z})\,\nabla_{\mathbf{z}_j^{(t)}} \log \tilde{p}_*(\mathbf{z}_j^{(t)}) + \nabla_{\mathbf{z}_j^{(t)}}\kappa(\mathbf{z}_j^{(t)}, \mathbf{z})\right]$
end
```

Variational family for SVGD. The variational family for SVGD the set of all distributions that can be produced by pushing an initial particle distribution through a sequence of infinitesimal flows. This provides a rich and expressive set of distributions, but the procedure can be computationally expensive requiring many iterations to reach a reasonable solution. SVGD also requires careful selection of the hyperparameters (e.g., kernel bandwidth). After passing the particles through SVGD, one can sample from the approximate target by drawing one of the N final particles uniformly as particles will be concentrated near modes and sparse in valleys. Increasing the number of particles leads to smoother empirical estimates and as $N \to \infty$, our estimate of the target will be "continuous" in a proper sense [LW16].

Particle dynamics. The (estimated) optimal flow $\hat{\boldsymbol{\omega}}^*$ describes the how the particles evolve with time. The two terms in Equation (SVGD) play different roles: The first term $\kappa(\mathbf{z}_j, \mathbf{z})\,\nabla_{\mathbf{z}_j} \log \tilde{p}_*(\mathbf{z}_j)$ exhibits mode-seeking behavior as it draws particles toward regions of high target probability by following a *kernel-smoothed* gradient. The second term $\nabla_{\mathbf{z}_j}\kappa(\mathbf{z}_j, \mathbf{z})$ acts as a repulsion force that spreads out particles and prevents them from collapsing onto the same mode. To highlight this point, consider the RBF kernel $\kappa(\mathbf{z}, \mathbf{z}') = \exp\left(-\frac{1}{h}\|\mathbf{z} - \mathbf{z}'\|_2^2\right)$. The second term simplifies to $\sum_j \frac{2}{h}(\mathbf{z} - \mathbf{z}_j)\kappa(\mathbf{z}_j, \mathbf{z})$ which pushes a particle $\mathbf{z}$ away from neighboring particles $\mathbf{z}_j$ inside its kernel radius.

Comparison to Teleporting Walkers. Like SVGD, Teleporting Walkers maintains an ensemble of particles, but rather than using a functional gradient based flow, its evolution uses MCMC [LWZ22]. In addition to allowing for the ensemble to explore the target distribution, the particles interact with each other such that they can "teleport" to each other. So in some sense, the teleportation step is an attractive force pushes particles close to each other. The variance of the chains is otherwise controlled by the proposal distribution as the chains move independently according to an MCMC sampler, e.g., Metropolis-Adjusted Langevin (noisy optimization). The Langevin proposal is given by

$$\mathbf{y} = \mathbf{z}_t + \Delta t \left(\nabla_{\mathbf{z}} \log \tilde{p}_*(\mathbf{z}) \Big|_{\mathbf{z}=\mathbf{z}_t} \right) + \sigma \sqrt{\Delta t}\, \xi \quad \text{where} \quad \xi \sim \mathcal{N}(0, I) \quad \text{and} \quad \sigma \geq \sqrt{2}$$

and the candidate $\mathbf{y}$ is accepted with the usual Metropolis-Hastings acceptance probability. Similarly to SVGD, the Langevin proposal has a mode-seeking term that draws particles toward regions of high target probability. However, the second term is neither a repulsion nor an attraction term, but rather a diffusion term that allows the particles to explore the target distribution where σ controls the variance of the chains. Algorithm 4 in Appendix C describes the Teleporting Walkers algorithm in detail.

Selecting kernel bandwidth. Liu, *et al.* [LW16] describe the "median trick" for RBF kernel $\kappa(\mathbf{z}, \mathbf{z}') = \exp\left(-\frac{1}{h}\|\mathbf{z} - \mathbf{z}'\|_2^2\right)$, they take the bandwidth to be $h = \mathtt{med}^2 / \log n$, where $\mathtt{med}$ is the median of the pairwise distance between the current points $\{\mathbf{z}_i^{(t)}\}_{i=1}^N$; this is based on the intuition that $\sum_j \kappa(\mathbf{z}_i, \mathbf{z}_j) \approx n \exp\left(-\frac{1}{h}\mathtt{med}^2\right) = 1$, so that for each $\mathbf{z}_i$ the contribution from its own gradient and the influence from the other points balance with each other [LW16].

While the median trick is reasonable in lower dimensional spaces, ℓ_2-norms are known to collapse in higher dimensions. For example, most of the mass of a multivariate Gaussian is not near the mean, but in an increasingly distant "shell" around it [Dom12]. In this case, the mode seeking terms would drive particles to the shell and thus the pairwise distances would grow larger and larger. The kernel term $\kappa(\mathbf{z}_i, \mathbf{z}_j)$ will be either ≈ 0 (if $h \gg \mathtt{med}^2$) or ≈ 1 (if $h \ll \mathtt{med}^2$). The median trick is sensitive to the dimension and to heavy-tailed targets.

Alternatively, we can use a K-nearest neighbors bandwidth set a distinct bandwidth for each particle,

$$h_i = \frac{1}{K} \sum_{j \in \mathcal{N}_k(i)} \|\mathbf{z}_i^{(t)} - \mathbf{z}_j^{(t)}\|_2^2, \qquad \kappa_{ij} = \exp\left(-\frac{\|\mathbf{z}_i - \mathbf{z}_j\|_2^2}{\sqrt{h_i h_j}}\right),$$

where $\mathcal{N}_K(i)$ is the set of K nearest neighbors of $\mathbf{z}_i^{(t)}$. This will allow the particles to adaptively select the bandwidth based on the local density of the particles. The K-nearest neighbors bandwidth is less sensitive to the dimension and can be more robust in high-dimensional spaces.

Appendix

A. Stein's Identify and Reproducing Kernel Hilbert Spaces

Suppose $p(\mathbf{z})$ is a continuously differentiable (i.e., smooth) density supported on $\mathcal{Z}$. Let $\boldsymbol{\phi}(\mathbf{z}) = [\phi_1(\mathbf{z}), \ldots, \phi_d(\mathbf{z})]^\top$ be a smooth vector function that is in the *Stein class* of p such that *Stein's identity* holds:

$$\mathbb{E}_{\mathbf{z}\sim p(\cdot)}\left[\mathcal{A}_p \boldsymbol{\phi}(\mathbf{z})\right] = 0 \quad \text{where } \mathcal{A}_p \text{ is the Stein operator} \quad \mathcal{A}_p \boldsymbol{\phi}(\mathbf{z}) := \boldsymbol{\phi}(\mathbf{z})\, \nabla_{\mathbf{z}} \log p(\mathbf{z})^\top + \nabla_{\mathbf{z}} \boldsymbol{\phi}(\mathbf{z}) \tag{8}$$

Let $q(\mathbf{z})$ represent a different smooth density also supported on $\mathcal{Z}$. The expectation $\mathbb{E}_{\mathbf{z}\sim q(\cdot)}\left[\mathcal{A}_p \boldsymbol{\phi}(\mathbf{z})\right]$ would no longer equal zero for general $\boldsymbol{\phi}$ and therefore measures how different p and q are. Note that $\mathbb{E}_{\mathbf{z}\sim q(\cdot)}\left[\mathcal{A}_p \boldsymbol{\phi}(\mathbf{z})\right]$ is a matrix quantity and ideally would have a scalar metric for measuring how different p and q are. We therefore let the trace of this expectation represent the "violation of Stein's identity" for a given $\boldsymbol{\phi}$. We then define the Kernelized Stein discrepancy (KSD) $\mathbb{S}(q, p)$ as the "maximum violation of Stein's identity" over the unit ball $\mathcal{B}(\mathcal{H}^d) := \left\{\boldsymbol{\phi} \in \mathcal{H}^d \ : \ \|\boldsymbol{\phi}\|^2_{\mathcal{H}^d} = \left\langle \boldsymbol{\phi}, \boldsymbol{\phi} \right\rangle_{\mathcal{H}^d} \leq 1\right\}$

$$\mathbb{S}(q,p) := \max_{\boldsymbol{\phi}\in\mathcal{B}(\mathcal{H}^d)} \left\{\mathbb{E}_{\mathbf{z}\sim q(\cdot)}\left[\mathrm{Tr}\big(\mathcal{A}_p \boldsymbol{\phi}(\mathbf{z})\big)\right]^2\right\} \tag{9}$$

It has been shown [LLJ16] that the optimal solution of $\mathbb{S}(q,p)$ in (9) has an analytical form $\boldsymbol{\phi}(\mathbf{z}) = \boldsymbol{\phi}^*_{q,p}(\mathbf{z}) / \|\boldsymbol{\phi}^*_{q,p}\|_{\mathcal{H}^d}$ where

$$\boldsymbol{\phi}^*_{q,p}(\cdot) = \operatorname*{argmax}_{\boldsymbol{\phi}\in\mathcal{B}(\mathcal{H}^d)} \left\{\mathbb{E}_{\mathbf{z}\sim q(\cdot)}\left[\mathrm{Tr}\big(\mathcal{A}_p \boldsymbol{\phi}(\mathbf{z})\big)\right]^2\right\} = \mathop{\mathbb{E}}_{\mathbf{z}\sim q(\cdot)}\left[\mathcal{A}_p \kappa(\mathbf{z}, \cdot)\right] \quad \text{for which we have} \quad \mathbb{S}(q,p) = \|\boldsymbol{\phi}^*_{q,p}\|_{\mathcal{H}^d}. \tag{10}$$

B. Functional Gradient

For any functional $F[\boldsymbol{v}]$ of $\boldsymbol{v} \in \mathcal{H}^d$, its functional gradient $\nabla_{\boldsymbol{v}} F[\boldsymbol{v}]$ is a function in $\mathcal{H}^d$ such that $F[\boldsymbol{v} + \varepsilon \boldsymbol{\omega}] = F[\boldsymbol{v}] + \varepsilon \left\langle \nabla_{\boldsymbol{v}} F[\boldsymbol{v}], \boldsymbol{\omega} \right\rangle_{\mathcal{H}^d} + \mathcal{O}(\varepsilon^2)$ for any $\boldsymbol{\omega} \in \mathcal{H}^d$ and $\varepsilon \in \mathbb{R}$. We define our functional $F[\boldsymbol{v}] = \mathtt{KL}\big(q_{[\chi]}(\mathbf{z})\,||\,p_*(\mathbf{z})\big)$ where $\boldsymbol{\chi}(\mathbf{z}_0) = \mathbf{z}_0 + \boldsymbol{v}(\mathbf{z}_0)$. It follows that

$$\mathtt{KL}\Big(q_{[\chi]}(\mathbf{z})\,\Big\|\,p_*(\mathbf{z})\Big) = \int q_{[\chi]}(\mathbf{z}) \log\left(\frac{q_{[\chi]}(\mathbf{z})}{p_*(\mathbf{z})}\right) d\mathbf{z} = \int q(\boldsymbol{\chi}^{-1}(\mathbf{z}))\cancel{\left|\det(\nabla_{\mathbf{z}}\boldsymbol{\chi}^{-1}(\mathbf{z}))\right|} \log\left(\frac{q(\boldsymbol{\chi}^{-1}(\mathbf{z}))\left|\det(\nabla_{\mathbf{z}}\boldsymbol{\chi}^{-1}(\mathbf{z}))\right|}{p_*(\mathbf{z})}\right) \cancel{\left(\frac{d\mathbf{z}}{d\mathbf{z}_0}\right)} d\mathbf{z}_0$$

$$\mathtt{KL}\Big(q_{[\chi]}(\mathbf{z})\,\Big\|\,p_*(\mathbf{z})\Big) = \int q(\mathbf{z}_0) \log\left(\frac{q(\mathbf{z}_0)}{p_*(\boldsymbol{\chi}(\mathbf{z}_0))\left|\det(\nabla_{\mathbf{z}_0}\boldsymbol{\chi}(\mathbf{z}_0))\right|}\right) d\mathbf{z}_0 = \mathtt{KL}\Big(q(\mathbf{z}_0)\,\Big\|\,p_{*[\chi^{-1}]}(\mathbf{z}_0)\Big) \text{ as } \frac{\partial \boldsymbol{\chi}^{-1}(\mathbf{z})}{\partial \mathbf{z}} = \left(\frac{\partial \mathbf{z}}{\partial \mathbf{z}_0}\right)^{-1} = \frac{1}{\nabla_{\mathbf{z}_0}\boldsymbol{\chi}(\mathbf{z}_0)}.$$

Then we have that

$$F[\boldsymbol{v} + \varepsilon\boldsymbol{\omega}] = \mathtt{KL}\Big(q(\mathbf{z}_0)\,\Big\|\,p_{*[\chi^{-1}]}(\mathbf{z}_0)\Big) = \mathop{\mathbb{E}}_{\mathbf{z}_0\sim q(\cdot)}\left[\log q(\mathbf{z}_0) - \log p_*(\mathbf{z}_0 + \boldsymbol{v}(\mathbf{z}_0) + \varepsilon\boldsymbol{\omega}(\mathbf{z}_0)) - \log\det\big(I + \nabla_{\mathbf{z}_0}\boldsymbol{v}(\mathbf{z}_0) + \varepsilon\nabla_{\mathbf{z}_0}\boldsymbol{\omega}(\mathbf{z}_0)\big)\right]$$

Next we have that

$$F[\boldsymbol{v} + \varepsilon\boldsymbol{\omega}] - F[\boldsymbol{v}] = -\Delta_1 - \Delta_2$$

where

$$\begin{aligned}
\Delta_1 &= \mathbb{E}_q\left[\log p_*(\mathbf{z}_0 + \boldsymbol{v}(\mathbf{z}_0) + \varepsilon\boldsymbol{\omega}(\mathbf{z}_0))\right] - \mathbb{E}_q\left[\log p_*(\mathbf{z}_0 + \boldsymbol{v}(\mathbf{z}_0))\right] \\
\Delta_2 &= \mathbb{E}_q\left[\log\det\big(I + \nabla_{\mathbf{z}_0}\boldsymbol{v}(\mathbf{z}_0) + \varepsilon\nabla_{\mathbf{z}_0}\boldsymbol{\omega}(\mathbf{z}_0)\big)\right] - \mathbb{E}_q\left[\log\det\big(I + \nabla_{\mathbf{z}_0}\boldsymbol{v}(\mathbf{z}_0)\big)\right]
\end{aligned}$$

For the terms in the above equation, we have

$$\begin{aligned}
\Delta_1 &= \mathbb{E}_q\left[\log p_*(\mathbf{z}_0 + \boldsymbol{v}(\mathbf{z}_0) + \varepsilon\boldsymbol{\omega}(\mathbf{z}_0))\right] - \mathbb{E}_q\left[\log p_*(\mathbf{z}_0 + \boldsymbol{v}(\mathbf{z}_0))\right] \\
\Delta_1 &= \varepsilon\ \mathbb{E}_q\left[\nabla_{\mathbf{z}_0}\log p_*(\mathbf{z}_0 + \boldsymbol{v}(\mathbf{z}_0)) \cdot \boldsymbol{\omega}(\mathbf{z}_0)\right] + \mathcal{O}(\varepsilon^2) \\
\Delta_1 &= \varepsilon\ \mathbb{E}_q\left[\nabla_{\mathbf{z}_0}\log p_*(\mathbf{z}_0 + \boldsymbol{v}(\mathbf{z}_0)) \cdot \langle \kappa(\mathbf{z}_0, \cdot), \boldsymbol{\omega}\rangle_{\mathcal{H}^d}\right] + \mathcal{O}(\varepsilon^2) \\
\Delta_1 &= \varepsilon\ \Big\langle \mathbb{E}_q\left[\nabla_{\mathbf{z}_0}\log p_*(\mathbf{z}_0 + \boldsymbol{v}(\mathbf{z}_0)) \cdot \kappa(\mathbf{z}_0, \cdot)\right], \boldsymbol{\omega}\Big\rangle_{\mathcal{H}^d} + \mathcal{O}(\varepsilon^2)
\end{aligned}$$

and

$$\begin{aligned}
\Delta_2 &= \mathbb{E}_q\left[\log\det\big(I + \nabla_{\mathbf{z}_0}\boldsymbol{v}(\mathbf{z}_0) + \varepsilon\nabla_{\mathbf{z}_0}\boldsymbol{\omega}(\mathbf{z}_0)\big)\right] - \mathbb{E}_q\left[\log\det\big(I + \nabla_{\mathbf{z}_0}\boldsymbol{v}(\mathbf{z}_0)\big)\right] \\
\Delta_2 &= \varepsilon\ \mathbb{E}_q\left[\nabla\log\det\big(I + \nabla_{\mathbf{z}_0}\boldsymbol{v}(\mathbf{z}_0)\big) \cdot \nabla_{\mathbf{z}_0}\boldsymbol{\omega}(\mathbf{z}_0)\right] + \mathcal{O}(\varepsilon^2)
\end{aligned}$$

$$\begin{aligned}\Delta_2 &\overset{(11)}{=} \varepsilon\, \mathbb{E}_q \left[[I + \nabla_{\mathbf{z}_0} \boldsymbol{v}(\mathbf{z}_0)]^{-1} \cdot \nabla_{\mathbf{z}_0} \boldsymbol{\omega}(\mathbf{z}_0) \right] + O(\varepsilon^2)\\ \Delta_2 &= \varepsilon\, \mathbb{E}_q \left[[I + \nabla_{\mathbf{z}_0} \boldsymbol{v}(\mathbf{z}_0)]^{-1} \cdot \left\langle \nabla_{\mathbf{z}_0} \kappa(\mathbf{z}_0, \cdot\,), \boldsymbol{\omega} \right\rangle_{\mathcal{H}^d} \right] + O(\varepsilon^2)\\ \Delta_2 &= \varepsilon \left\langle \mathbb{E}_q \left[[I + \nabla_{\mathbf{z}_0} \boldsymbol{v}(\mathbf{z}_0)]^{-1} \cdot \nabla_{\mathbf{z}_0} \kappa(\mathbf{z}_0, \cdot\,) \right], \boldsymbol{\omega} \right\rangle_{\mathcal{H}^d} + O(\varepsilon^2)\end{aligned}$$

Therefore, we have

$$F[\boldsymbol{v} + \varepsilon \boldsymbol{\omega}] - F[\boldsymbol{v}] = \varepsilon \left\langle \mathbb{E}_q \left[\kappa(\mathbf{z}_0, \cdot\,)\, \nabla_{\mathbf{z}_0} \log p_*(\mathbf{z}_0 + \boldsymbol{v}(\mathbf{z}_0)) \right] + \mathbb{E}_q \left[[I + \nabla_{\mathbf{z}_0} \boldsymbol{v}(\mathbf{z}_0)]^{-1}\, \nabla_{\mathbf{z}_0} \kappa(\mathbf{z}_0, \cdot\,) \right], \boldsymbol{\omega} \right\rangle_{\mathcal{H}^d} + O(\varepsilon^2)$$

and since $F[\boldsymbol{v} + \varepsilon \boldsymbol{\omega}] - F[\boldsymbol{v}] = \varepsilon \langle \nabla_{\boldsymbol{v}} F[\boldsymbol{v}], \boldsymbol{\omega} \rangle_{\mathcal{H}^d} + O(\varepsilon^2)$, the functional gradient is given by

$$\nabla_{\boldsymbol{v}} F[\boldsymbol{v}] = -\mathbb{E}_q \left[\kappa(\mathbf{z}_0, \cdot\,)\, \nabla_{\mathbf{z}_0} \log p_*(\mathbf{z}_0 + \boldsymbol{v}(\mathbf{z}_0)) + [I + \nabla_{\mathbf{z}_0} \boldsymbol{v}(\mathbf{z}_0)]^{-1}\, \nabla_{\mathbf{z}_0} \kappa(\mathbf{z}_0, \cdot\,) \right]$$

Setting $\boldsymbol{v} = \mathbf{0}$ gives us the desired result

$$\boxed{\left(\nabla_{\boldsymbol{v}} F[\boldsymbol{v}]\right)\Big|_{\boldsymbol{v}=\mathbf{0}} = -\mathbb{E}_q \left[\mathcal{A}_{p_*} \kappa(\mathbf{z}_0, \cdot\,) \right] = -\mathbb{E}_q \left[\kappa(\mathbf{z}_0, \cdot\,)\, \nabla_{\mathbf{z}_0} \log p_*(\mathbf{z}_0) + \nabla_{\mathbf{z}_0} \kappa(\mathbf{z}_0, \cdot\,) \right]}$$

Therefore, we can take a step and then treat the transformed particles as $\mathbf{z}_0$ and repeat until convergence.

C. Ensemble Markov Chain Monte Carlo with Teleporting Walkers

Algorithm 4: Ensemble Markov Chain Monte Carlo with Teleporting Walkers [LWZ22]

```
Input: $\pi(z)$ target distribution (possibly unnormalized), $p_r$ proposal distribution, fixed $N$ number of random walkers
Output: A mean-field variational density $q(\mathbf{y}) = \prod_{j=1}^{m} q_j(y_j)$
Initialize: $N$ random walkers with initial positions $\mathbf{z} = (z_1, \dots, z_N) \in \mathcal{Z}^N$
for $j \in \{1, \dots, m\}$ do
    Sample teleporter's arrival index $j \in \{1, 2, \dots, N\}$ uniformly at random and sample $y \sim p_r(z_j \to y)$
    Sample teleporter's starting index $i \in \{1, 2, \dots, N\}$ (possibly equal to $j$) according to importance weights

        $w_i(\mathbf{z}, y) := \frac{p_r(y \to z_i) + \sum_{k \neq i}^{N} p_r(z_k \to z_i)}{\pi(z_i)} \Big/ Z(\mathbf{z}, y) \quad \text{where} \quad Z(\mathbf{z}, y) := \sum_{l=1}^{N} \frac{p_r(y \to z_l) + \sum_{k \neq l}^{N} p_r(z_k \to z_l)}{\pi(z_l)}$   (11)

    Construct candidate $\mathbf{z}'$: set $\mathbf{z}' = (z'_1, \dots, z'_N) \leftarrow \mathbf{z} = (z_1, \dots, z_N)$ and teleport walker by overwriting $z'_i \leftarrow y$
    Compute acceptance probability and acceptance ratio:

        $a(\mathbf{z}') = \min(r, 1) \quad \text{where} \quad r = \frac{\Pi(\mathbf{z}')\, p_r(\mathbf{z}' \to \mathbf{z})}{\Pi(\mathbf{z})\, p_r(\mathbf{z} \to \mathbf{z}')} = \frac{Z(\mathbf{z}, y)}{Z(\mathbf{z}', z_i)}$   (12)

    if $U \leq a(\mathbf{z}')$ where $U \sim \text{Uniform}([0, 1])$ then
        Accept candidate $\mathbf{z}'$, set $\mathbf{z}_{t+1} = \mathbf{z}'$.
    else
        Reject candidate $\mathbf{z}'$, set $\mathbf{z}_{t+1} = \mathbf{z}$.
    end
end
return $q(\mathbf{y})$
```

[11] $\nabla_{\mathbf{A}} \log \det \mathbf{A} = \mathbf{A}^{-1}$ for symmetric positive definite matrix $\mathbf{A}$ from Section A.4.1 in *Convex Optimization* [BV09]. Here, $I + \nabla_{\mathbf{z}_0} \boldsymbol{v}(\mathbf{z}_0)$ is symmetric positive definite as $I + \nabla_{\mathbf{z}_0} \boldsymbol{v}(\mathbf{z}_0) = \mathbf{F}$ and $J = \det \nabla_{\mathbf{z}_0} \boldsymbol{\chi} > 0$ by conservation of mass.

References

[BKM17] David M Blei, Alp Kucukelbir, and Jon D McAuliffe, *Variational inference: A review for statisticians*, Journal of the American statistical Association **112** (2017), no. 518, 859–877.

[BV09] S.P. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge University Press, 2009.

[Dom12] Pedro Domingos, *A few useful things to know about machine learning*, Communications of the ACM **55** (2012), no. 10, 78–87.

[GCS+13] A. Gelman, J.B. Carlin, H.S. Stern, D.B. Dunson, A. Vehtari, and D.B. Rubin, *Bayesian data analysis, third edition*, Chapman & Hall/CRC Texts in Statistical Science, Taylor & Francis, 2013.

[LLJ16] Qiang Liu, Jason Lee, and Michael Jordan, *A kernelized stein discrepancy for goodness-of-fit tests*, International conference on machine learning, PMLR, 2016, pp. 276–284.

[LW16] Qiang Liu and Dilin Wang, *Stein variational gradient descent: A general purpose bayesian inference algorithm*, Advances in neural information processing systems **29** (2016), 1–9.

[LWZ22] Michael Lindsey, Jonathan Weare, and Anna Zhang, *Ensemble markov chain monte carlo with teleporting walkers*, SIAM/ASA Journal on Uncertainty Quantification **10** (2022), no. 3, 860–885.

[PNR+21] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan, *Normalizing flows for probabilistic modeling and inference*, Journal of Machine Learning Research **22** (2021), no. 57, 1–64.

[WCLK21] Yiwei Wang, Jiuhai Chen, Chun Liu, and Lulu Kang, *Particle-based energetic variational inference*, Statistics and Computing **31** (2021), 1–17.