We thank the reviewers for the constructive feedback, which will significantly improve the paper. Reviewers 1-3 all agree that it is a well-written paper and a worthy contribution to the conference. One common weakness discussed was data scalability. We elaborate on this first and address specific comments and questions from the reviewers below.

On the scalability of MSC Variational inference based on KL(q||p) is scalable in the sense that it works by subsampling datasets both for *exchangeable data*, $p(x_{1:n}) = \mathbb{E}_{p(z)}\left[\prod_{i=1}^n p(x_i|z)\right]$, as well as for *independent and identically distributed data* (iid), $p(x_{1:n}) = \prod_{i=1}^n p(x_i) = \prod_{i=1}^n \mathbb{E}_{p(z_i)}\left[p(x_i|z_i)\right]$. Often in the literature (such as for VAEs, RWS, etc.) applications assumes the data is generated iid and and achieve scalability through use of subsampling and amortization. The current discussion in Section 3.5 for MSC on the other hand focuses on the more challenging case, when the data is exchangeable. In fact, MSC can potentially scale just as well as other algorithms to large datasets when data is assumed iid $x_i \sim p(x),\ i = 1, \dots, n$. Instead of minimizing $\mathrm{KL}(p(z|x)\|q(z;\lambda))$ wrt λ for each $x = x_i$, we consider minimizing $\mathrm{KL}(p(x)p(z|x)\|p(x)q(z|\lambda_\eta(x)))$ wrt η where $\lambda_\eta(x)$ is an inference network (amortization). If $q(z|\lambda_\eta(x))$ is flexible enough the posterior $p(z|x)$ is the optimal solution to this minimization problem. Stochastic gradient descent can be performed by noting that

$$\nabla_\eta \mathrm{KL}(p(x)p(z|x)\|p(x)q(z|\lambda_\eta(x))) = 0 + \mathbb{E}_{p(x)p(z|x)}\left[-\nabla_\eta \log q(z|\lambda_\eta(x)\right] \approx \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{p(z|x_i)}\left[-\nabla_\eta \log q(z|\lambda_\eta(x_i))\right],$$

where the right hand side is directly amenable to data subsampling. We will include a discussion of this aspect of scalability in Section 3.5 in the revision.

Reviewer 4: Consistency of self-normalized IS vs MSC MSC *does not* require the MCMC sampler to reach stationarity at each iteration k of Alg 1. A single update (line 2) with any initialization ensures convergence as the number of iterations k increases. In contrast self-normalized IS would require an infinite number of samples *at each iteration k* to ensure convergence to a minima of KL(p||q). That is, self-normalized IS/RWS is "doubly asymptotic"—both the number of samples per iteration and the number of iterations need to go to infinity for convergence. MSC, on the other hand, is "singly asymptotic"—the MCMC iterations are intertwined with the optimization iterations k and will converge as k goes to infinity even with a single MCMC step per iteration. We will clarify this in the revision.

Reviewer 4: Trade-off between sample size S and time steps/iterations K The iterations, or time steps, corresponds to regular parameter updates in any gradient descent algorithm. The second experiment used 10,000 iterations for both RWS and MSC as very little change in parameter estimates was observed for longer runs. Increasing the number of samples S would not negatively impact the performance of MSC compared to self-normalized IS variants (see eg Fig 2), as the Rao-Blackwellization (below eq 8) would similarly improve the gradient estimate quality for MSC.

Reviewer 4: Stronger baselines, RWS with HMC We compare the base versions of the respective algorithms. Just like RWS can be improved by introducing HMC updates, so can a similar benefit be achieved by introducing such updates in the MSC Markov kernel.

Reviewer 4: Missing related work We will add these references to the related work section. The most similar is Li et al. (2017). The main difference compared to MSC is that Li et al. seek to minimize $\mathrm{D}\left(\mathcal{K}_T q(z;\lambda)\|q(z;\lambda)\right)$, where $\mathcal{K}_T$ corresponds to an MCMC kernel $\mathcal{K}$ with stationary distribution $p(z|x)$ applied T times. Because they re-initialize the MCMC procedure at each update, corresponding to $\mathcal{K}_T q(z;\lambda)$, this will not converge to a minimum of $\mathrm{D}\left(p(z|x)\|q(z;\lambda)\right)$. This in contrast to MSC that provably minimizes $\mathrm{D}\left(p(z|x)\|q(z;\lambda)\right)$ directly.

Reviewer 3: The last experiment does not test against RWS, why? RWS is based on self-normalized IS that performs poorly for state space models. We compare instead to neural adaptive SMC, an algorithm based on SMC tailored to state space models, which is a stronger baseline than RWS for this application.

Reviewer 3: CIS/CSMC, unbiasedness and convergence The gradients estimated by CIS/CSMC are not unbiased, but due to their Markov properties the bias vanishes as the number of iterations K increases. This means it is still possible to show convergence to a true optima of the inclusive KL as shown by Prop 1 and as illustrated empirically in Fig 1. This is in contrast to self-normalized IS/SMC-based gradients whose bias does not vanish as the number of iterations K increases, also illustrated in Fig 1. Contrary to MSC, the self-normalized IS/SMC-based algorithms require an infinite number of samples at *each iteration* to ensure convergence, as discussed above.

Reviewer 3: Mixing up self-normalized IS/RWS and IS We will revise the notation to make this difference clear.

Reviewer 2: Conditions 1-6 and convergence intuition The assumptions are fairly standard for Markovian stochastic approximations. The conditions on the MCMC kernel can be verified for CIS/CSMC under compactness assumptions. We will add discussion and references to the supplement.

Reviewer 1: Predictive distributions based on KL(p||q) We will revise and include this aspect of KL(p||q) as well as an extended discussion on when one is preferable to the other.