1 We would like to thank all reviewers for constructive comments. Responses to criticism below, we will clarify all
2 unclear points in the final paper.

3 **R1, R2, R4**: Suggest comparisons with more advanced flow layers and trainable sampling techniques.

4   1. We agree - we will add comparisons with more modern flow layers and trainable stochastic layers in the final
5   paper as suggested by reviewers.
6   2. General idea is: adding stochasticity increases the expressivity of a given flow architecture because stochastic
7   layers do not have the restrictions coming from invertibility. For a given application problem, we can improve
8   the SOTA flow method by adding stochasticity and turning it into a SNF.
9   3. Some recent Monte Carlo methods utilize learnable proposals to improve the sampling efficiency, but they
10   cannot yield exact reweighing scheme for tractable likelihoods like SNFs. But indeed, learnable proposals
11   can be incorporated into SNFs, and we will discuss this and add comparisons in the final paper.

12 **R1, R3**: This work heavily relies on nonequilibrium statistical physics.

13 True, the motivation comes from Statistical Physics but it has important consequences for machine learning (ML). We
14 will provide a more easily accessible schematic of SNFs for ML readers and also give an interpretation of the loss
15 function as a variational bound in sampling path space.

16 **R1, R3**: Discuss computational cost / efficiency of SNFs versus NFs:

17 Agreed! We will provide a new Fig. 3f which shows KL divergence versus number of layers. Briefly, a stochastic
18 sampling step is a network layer, although a very simple one (in our implementation just point-wise operations without
19 parameters). We can show that even when just counting the number of layers, SNFs can surpass equally deep NFs.

20 **R4**: "Topological constraints" was not carefully formalized. Can the deficiencies in Fig. 3 overcome with other NF
21 architectures?

22 Formalization of topological constraints was done by others and we will discuss this more clearly in the final paper.
23 Yes, NF architectures have been developed to remedy these topological problems, but adding stochasticity can improve
24 them further (see first question+answer above).

25 **R4**: unbiased guarantees for estimating expectations are only in the asymptotic limit. Analyze the bias-variance
26 tradeoff for estimating empirical evaluations.

27 For the final paper, we will compare the errors of Table 1 for the biased case (uncorrected flow output distribution) and
28 the unbiased case (after applying Eq. 10) for different sample sizes – probably in a supplementary Figure.

29 As an explanation: our understanding is that the referee refers to the bias/variance tradeoff of invoking Eq. (10) on an
30 already trained estimator, which is somewhat different from the classical bias/variance tradeoff in statistical estimator
31 theory. A trained flow generates output distribution $p_X(\mathbf{x})$ that is different from target distribution $\mu_X(\mathbf{x})$, e.g. due
32 to limitations of its architecture, and will thus generate a biased estimate of an expectation value $\mathbb{E}_{\mathbf{x} \sim p_X}[O(\mathbf{x})]$. This
33 bias can be asymptotically removed by reweighting, e.g. Eq. (10), but this comes at the cost of a higher variance of
34 $\mathbb{E}_{\mathbf{x} \sim p_X}[O(\mathbf{x})]$ for a given number of samples. So we can compare the mean prediction error of $\mathbb{E}_{\mathbf{x} \sim p_X}[O(\mathbf{x})]$ before
35 and after applying Eq. (10). Note that we can counter the larger variance by drawing more samples, which is usually
36 preferred in physics applications, because unbiased estimators are essential there, and also taking samples from a
37 trained flow is fairly cheap compared to training time.

38 **R4**: Related work such as A-NICE-MC assumes that detailed balance is satisfied while using flows as proposals for
39 MCMC. It is unclear if such conditions are being satisfied in practice for SNFs.

40 Approaches like A-NICE-MC use detailed balance (DB) in each step, whereas SNFs rely on path-based detailed
41 balance between the prior and the target density. This is strictly more general, i.e. SNFs can use stochastic layers
42 that each respect DB (e.g., Metropolis Monte Carlo), but we can also give up DB in each step as long as the path
43 probability ratio (9) can be computed (e.g., Langevin dynamics). Will be clarified in final paper.

44 **R1**: Temperature used for the Alanine Dipeptide was 1000 K. Does the method fail with sharper modes?

45 1000 K was used in order to get a well-defined ground truth distribution as at 1000 K the MD simulation converges
46 quickly. Will be clarified in final paper. The method performs well with sharp modes – see Fig. 3.

47 **R4**: Explain how related methods, e.g. [33, 21, 35, 16, 15] are special instantiations of SNFs.

48 Excellent idea, will do.

49 **R1, R2, R3**: There were some typos, grammatical errors, undefined notations etc.