We thank the reviewers for their feedback, and we hope to use the comments to improve our paper. We're glad that the reviewers are enthusiastic about the value of an overall framework and method for efficient rare-event simulation as applied to testing safety-critical systems. In addition to the supplementary code provided, we intend to make our method publicly available and easy to use to garner more impact across the NeurIPS community.

Related work: The reviewers' comments mirrored internal discussion we had in writing this draft. We chose to emphasize the application, and, due to space constraints, we focused the literature review on related work in safety testing; we discussed related methods for rare-event sampling only in the context of the method (Section 2). Based on the feedback, however, we will include significant additional discussion of sampling, anomaly detection, and neural density estimation in the revised related work. **R1** notes close relatives of bridge sampling, such as umbrella sampling and multicanonical MCMC. The operational difference between these methods and bridge sampling is in the form of the intermediate distribution used to calculate the ratio of normalizing constants; the optimal umbrella sampling distribution is more brittle than that of bridge sampling. We will also discuss path sampling, a generalization of bridge sampling, wherein we take the discrete bridges to a continuous limit. This is difficult to implement in an adaptive fashion. To answer **R1's** question, we present our approach's relationship to sequential Monte Carlo methods (see e.g. Del Moral, Doucet, Jasra, "Sequential Monte Carlo samplers," 2006) to help readers understand our method. More broadly, our method falls under the formalism of Feynman-Kac models, as do particle filters, birth-death processes, and smoothing filters (see e.g. Del Moral, "Feynman-Kac formulae," 2004). **R1's** reference suggests a slightly narrower view of SMC. Regardless, none of our theoretical results use exogenous machinery or assumptions from SMC.

We also think a broader discussion of rare-event sampling methods (as **R3** suggests) is useful. We will compare non-parametric methods like bridge sampling and multilevel splitting with parametric adaptive importance sampling methods like the cross-entropy method. Our method has both flavors, including parametric warping distributions within the bridge-sampling formalism. Additionally, many of these techniques work with variance-reduction methods like control variates. Recent literature combines importance sampling and kernel methods to make control variate *functions* (e.g. Liu, Lee, "Black-box importance sampling," 2016), which could also provide useful context.

We agree with **R2** that it is useful to include a more thorough discussion on neural density estimation and inference (e.g. Papamakarios et al., "Normalizing Flows [...]," 2019) in the related work. **R2** also asked how our method would work for detection of OOD failures and its relation to the anomaly detection literature. This paper assumes that the generative model of the operating domain P_0 is given (lines 30–31), so all failures are in the modeled domain. Therefore, when deploying systems in the real world, anomaly detection (e.g. Nachman and Shih, "Anomaly Detection[...]," 2020, and Choi et al., "WAIC, but Why?[...]," 2019) to discover distribution shifts is complementary to our approach. Another way to frame that problem is via distributional robustness to P_0 (Rahimian and Mehrotra, "Distributionally[...]," 2019).

Hessian computation: We will clear up confusion **R1** raises regarding the value of eliminating Hessian computation in our approach. Computing $\nabla^2 f$ requires a *double-backward pass through a simulation rollout*. Due to the many time steps involved in the rollout, static computation graphs (e.g. Tensorflow) require exorbitant memory (500 GB of RAM for MountainCar), and dynamic graphs (e.g. PyTorch) simply take a long time to compute. As noted in lines 152–153, Jacobians of flows do not involve f, and therefore do not require any backward passes through the simulator. Moreover, the MAF architecture ensures cheap Jacobian computations (in memory/time).

A more fundamental problem is that simulation rollouts are rarely smooth. For example, if the agent's controller contains ReLUs, the map $x \mapsto f(x)$ can be continuous but not smooth. The synthetic experiment showcases the problems that result from using local second-order information from a non-smooth function. We will improve this discussion in the experiments as well as in the introduction (as **R3** suggests).

Experiments: R1 asks whether our simulators fall into the formalism of nested probabilistic programs: they do not. Specifically, the stochasticity is in initial conditions, and rollouts for given initial conditions are deterministic. We will add a note about the need for other methods (e.g. nested Monte Carlo) for stochastic rollouts. As **R2** suggests, we will illustrate the clustering of failure modes to enhance the experiments. **R2** also raises good points about the connection between the theoretical analysis of NBS's efficiency and the presentation of the experiments. The goal of Eq. (11) is to show that the mean-square error is empirically estimable from a single trial, and we will show this matches the empirical results from repeated experimentation over 10 trials. We will also add discussion around the linear trend of the yellow lines in Figures (1c) and (2c), which matches the theoretical relationship in Proposition 1 and line 215. We will also add additional experiments in the appendix showing performance trends vs. sample size N. **R3** suggests comparison with sampling methods beyond naive Monte Carlo. The experiments already include a comparison with adaptive multilevel splitting (AMS) and bridge sampling (B).

Notation: R1 asked about the "negative ReLU" function, which is $[x]_- := -[-x]_+ = xI\{x < 0\}$; we will clarify this in Section 2. As **R1** and **R2** suggest, we will bring forward some of the discussion about HMC from Appendix A into Section 2 to make the various moving parts of HMC and normalizing flows easier to understand.

Framing: R4 raised the discussion of framing our paper more generally. Our original intended audience was the safety-testing community, but we will discuss the method's generality in the introduction and conclusion. As **R2** suggests, we will eliminate or define jargon that may be unfamiliar to the broader ML community. Finally, as **R4** suggests, we are considering a title change to broaden the scope of our audience.