We thank all reviewers for their comments, which overall were positive on novelty, our empirical sample quality results and ablations, and our connection between diffusion models and denoising score matching (DSM) with Langevin dynamics. Reviewers generally asked for more discussion on the relationship to other models (e.g. NCSN and GANs).

R1: **Slow sampling speed**: this is indeed a disadvantage of diffusion models, just like autoregressive models and score matching/energy based models with MCMC samplers. We'll discuss this, and we'd like to improve this in future work.

R2: **Explanation of empirical advantages over NCSNv1 [2], v2 [3]**: (Note that NCSNv2 [3] appeared on arXiv after the NeurIPS deadline.) Apart from differences **R2** mentioned, our architecture, forward process definition and prior are subtle but important choices that improve sample quality, and most importantly, we directly train the sampler as a latent variable model rather than adding it post-hoc. Details: **(1)** We use a U-Net with self-attention; NCSN uses a RefineNet with dilated convolutions. We condition all layers on t by adding in the Transformer sinusoidal position embedding, rather than only in normalization layers (NCSNv1) or only at the output (v2). **(2)** Diffusion models scale down the data with each forward process step (the $\sqrt{1-\beta_t}$ factor in Eq 2) so that variance does not grow when adding noise, thus providing consistently scaled inputs to the neural net reverse process. NCSN omits this scaling factor. **(3)** Unlike NCSN, our forward process destroys signal ($D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel \mathcal{N}(\mathbf{0},\mathbf{I})) \approx 0$), ensuring a close match between the prior and aggregate posterior of $\mathbf{x}_T$. Also unlike NCSN, our β_t are very small, which ensures that the forward process is reversible by a Markov chain with conditional Gaussians. Both of these factors prevent distribution shift when sampling. **(4)** Our Langevin-like sampler (Eq 11, L87) has coefficients derived rigorously from β_t in the forward process. Thus, our training procedure directly trains our sampler to match the data distribution after T steps: it trains the sampler as a latent variable model using variational inference (see L90-93). In contrast, NCSN's sampler coefficients are set by hand post-hoc, and their training procedure is not guaranteed to directly optimize a quality metric of their sampler. **Explanation of loss weighting**: the NCSN loss (Eq 5-6 of [2]), combined with their choice $\lambda(\sigma_i) = \sigma_i^2$, simplifies to $\frac{1}{L}\sum_{i=1}^{L} \mathbb{E}_{\mathbf{x}}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\frac{1}{2}\|\sigma_i s_\theta(\mathbf{x}+\sigma_i\boldsymbol{\epsilon},\sigma_i)+\boldsymbol{\epsilon}\|^2\right]$. These MSE terms are equally weighted, analogous to our "unweighted" Eq 14. NCSNv2 defines $s_\theta(\cdot,\sigma_i) = s_\theta(\cdot)/\sigma_i$, so their loss becomes $\frac{1}{L}\sum_{i=1}^{L} \mathbb{E}_{\mathbf{x}}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\frac{1}{2}\|s_\theta(\mathbf{x}+\sigma_i\boldsymbol{\epsilon})+\boldsymbol{\epsilon}\|^2\right]$, which is similar to ours. **Experimental details**: see Appendix B. Like GAN literature, we picked the best checkpoints according to FID (50k samples on CIFAR10, 2048 on LSUN/CelebA-HQ). We used 35.7M parameters on CIFAR10, and NCSN used 29.7M. NCSNv2 used 80M-95M parameters for LSUN (128^2) and FFHQ (256^2); we used 114M for LSUN (256^2) and CelebA-HQ (256^2). On TPU v3-8 (similar to 8 V100 GPUs), our CIFAR model trains at 21 steps/sec at batch size 128 (10.6 hours to train to completion at 800k steps), and sampling a batch of 256 images takes 17 sec; our CelebA-HQ/LSUN (256^2) models train at 2.2 steps/sec at batch size 64, and sampling a batch of 128 images takes 300 sec. **Sampling time vs. data dimension**: sampling time (Alg. 2) depends on T and the neural net, which are fixed before training (like how they are fixed before training a hierarchical VAE). We'd like to investigate how existing MCMC theory on this topic applies to our models.

R3: **Performance at high resolution**: since submission, we trained a larger 256M parameter model for 256^2 LSUN Bedroom (vs 114M in the submission), **improving FID from 6.36 to 4.90**. We expect more improvements are possible for high resolutions via model scaling. **GANs**: GANs have fast generation, whereas we used $T = 1000$ steps. Downsides of GANs are training instability, difficulty in capturing the whole data distribution, and difficulty in evaluating overfitting. In contrast, our model is trained on a simple, stable non-adversarial MSE loss. Like other likelihood-based models (autoregressive, VAE, flows), our model captures all modes and we can easily check overfitting by computing test set log likelihood. **Qualitative comparison w/ the original diffusion model**: the baseline (first two rows in Table 2) is our reimplementation of the original model with a modern neural net; we'll add a qualitative figure.

R4: **Comparisons to models with similar hierarchical structures**: the closest is NVAE [4] (appeared on arXiv after the NeurIPS deadline). NVAE achieves better log likelihoods and has faster generation, but we attain better sample quality (IS/FID) and provide rate-distortion curves. **DSM on other models**: this is not straightforward because our equivalence between DSM and the diffusion objective (Eq 8-12) relies on the Gaussian forward process (Eq 4, 6, 7), which is unique to the diffusion model. However, loss reweighting (Eq 14) could be useful for other models, as shown in prior work (e.g. beta-VAE, ConvDRAW). "**Is the diffusion setup key to the improvement?**" We believe so: see the discussion above with **R2**. "**Why is the variational bound a lossless codelength of discrete data?**" Due to the bits-back argument [1]. We will add details. **Connection to IAF**: we are not aware of a direct connection. Since IAF is a flow, it preserves information between data and latents, but diffusion models destroy information between $\mathbf{x}_0$ and $\mathbf{x}_T$ (as we stated in L213-215). **Reweighting and sample quality**: reweighting variational bounds has been shown to impact sample quality in prior work (e.g. beta-VAE, ConvDRAW). In our case, terms for small t ask the network to denoise data with very small amounts of noise; since such data is already clean, we down-weight these terms so that the network can focus on more difficult denoising tasks at larger t terms. We'll add this intuition to the paper.

[1] Keeping Neural Networks Simple by Minimizing the Description Length of the Weights (1993) **[2]** Generative Modeling by Estimating Gradients of the Data Distribution (2019) **[3]** Improved Techniques for Training Score-Based Generative Models (2020) **[4]** NVAE: A Deep Hierarchical Variational Autoencoder (2020)