We thank the reviewers for their insightful comments. We first clarify our approach and then address specific concerns.

**R1, R2** *Forward-backward asymmetry and decoding strategy.* NAOMI efficiently uses forward and backward hidden states $(h^f, h^b)$. Note that encoder and decoder share weights. For example, consider a situation where only $x_0$ and $x_8$ are known, and we wish to impute $x_{1\to7}$ ($x_1$ to $x_7$). We first predict the mid-point $\hat{x}_4 = g(h_0^f, h_8^b)$ and update the backward hidden states $h^b_{8\to4}$. Given $\hat{x}_4$, we predict $\hat{x}_2 = g(h_0^f, h_4^b)$ and update $h^b_{4\to2}$. Recursively, given $x_0$, $\hat{x}_2$, we predict $\hat{x}_1$. Since $x_{0\to2}$ are now known or imputed, we update the forward states $h^f_{0\to2}$ and predict $\hat{x}_3 = g(h_2^f, h_4^b)$. For the second half, after predicting $\hat{x}_6 = g(h_4^f, h_8^b)$, we update $h^f_{4\to6}$ and $h^b_{8\to6}$. *Note that $h^b_{8\to6}$ have been updated before when predicting $\hat{x}_4$!* More generally, the forward states $h^f$ are updated once whereas the backward states $h^b$ are twice. We encourage the reviewers to check the supplementary material, with code and visualizations of our decoding strategy.

**R2, R3** *Evaluation metrics.* Evaluating generative models is an open problem, e.g., log-likelihood does not correlate well with generation quality [Theis et al., 2015]. In our case, neither L2 nor log-likelihood can capture "realistic" player behavior in basketball [Zheng et al., 2016, Zhan et al., 2019]. Hence, we follow previous work and compute domain-specific metrics (speed, distance traveled, out-of-bounds rate) to compare trajectory quality. We will include L2-loss for the basketball dataset, but note that NAOMI (*0.013*) still outperforms SingleRes (*0.040*).

**R1** *Motivation.* In general, time-series data features different types of dynamics and missing value patterns compared to text and images. Time-series data are often multi-resolution, which are exploited by our model via the divide-and-conquer strategy. Note we do not use a fixed sampling scheme for missing values (see below). We would consider combining NAOMI with convolutional or Transformer-based approaches to handle high-dimensional sequences.

*Fixed length.* No, our method does *not* assume fixed-length sequences. NAOMI can decode and train on varying-length sequences, e.g., by padding shorter sequences to a maximal length.

*Masking.* We mask all dimensions for $n$ randomly chosen time-steps, which is independent of the order of the divide-and-conquer strategy (see Algo. 1). We used the same masking scheme for all methods, including `MaskGAN` and `GRUI`. Note the "halving" scheme in Figure 2 is only an example: NAOMI is compatible with *any* masking pattern. If the second half of a sequence is masked, NAOMI is pure forward inference (see supplemental material for results).

*Auto-regressive baseline with divide-and-conquer.* Note that "auto-regressive" and "divide-and-conquer" are mutually exclusive decoding strategies, hence this baseline does not exist by definition.

*Transformer.* We agree that applying NAOMI to Transformer models is interesting, but leave this for future work.

**R2** *a bi-drected RNN . . . to $f_f$ and $f_b$ encoders.* NAOMI iteratively (re)-encodes and decodes as described above. Only the initial sequence encoding (see Figure 2) behaves like a bi-directional RNN. *Bi-cubic spline.* We are happy to add this, but since we compared with the state-of-the-art baselines (e.g., `GRUI`) for sequence imputation, we believe our results stand on their own. *Table 3, does `Linear` show smaller errors.* The results in Table 3 are *not* errors, but *sample statistics*, i.e., the closer to the Expert, the better. `Linear` has smaller values, but are actually worse as they are further away from the Expert statistics. We chose these metrics for the reasons explained above.

**R3** *. . . might not address error propagation . . .* We agree that `NAOMI` may not fully solve error propagation for "any" gap size between observed time steps. However, we compared many model variations and baselines on three time-series datasets with different sequence lengths and varying missing-value proportions. Our extensive experiments have empirically shown the effectiveness of NAOMI. We believe noisy coarse predictions are not an issue on these datasets mostly due to the multiresolution structure and spatiotemporal smoothness of the data. Hence, even noisy coarse-level predictions provide reasonable grounding points at finer resolutions.

*Multiple decoders might be insufficient* There appear to be several misunderstandings. Note that we randomly sample masking patterns for each training step. Hence the model will see gaps of varying sizes during training. We compute the loss at the sequence level to make sure the entire sequence look real, which co-trains the decoders. We repeatedly use the reparameterization trick (L136) to make every sampling operation and hence the entire imputation procedure differentiable. Our experimental results demonstrate this end-to-end training approach is sufficient for our datasets.

*Maximum likelihood and adversarial training* NAOMI is a non-autoregressive generator, *which can be trained with any objective*. We used L2-loss for traffic and billiards because it is standard in those domains, see explanation above.

*Missing values . . . datasets.* We will add analyses similar to Figure 6 for all datasets to the Appendix.

*Minor comments.* We will make Figure 4 more readable. For inputs of the backward encoder, the first dimension is 1 for known or predicted steps, 0 for masked steps, and the other dimensions are 0 for masked steps.

[1] E. Zhan, et al. (2019), "Generating Multi-Agent Trajectories using Programmatic Weak Supervision," *ICLR*, 2019

[2] S. Zheng, et al. (2016), "Generating long-term trajectories using deep hierarchical networks," *NIPS*, 2016