[REVIEWER #1] Long-term dependencies and generative likelihood: We computed the estimated negative log probability on a test set of Bouncing ball dataset using importance sampling with the learned encoders (500 samplings). The results are RSSM: 3.518 and HRSSM: 3.601. Although it is true that we could expect the proposed model to capture better long-term dependency, we would like to emphasize that our main focus in this work is obtaining *structure* which is interpretable, stochastic, and temporally hierarchical. As such, our main metric is also the quality of the structure, not the accuracy of the generation like language modeling. We are actually satisfying with the fact that we obtain such structure without sacrificing the likelihood performance (of course, it would have been even better with better likelihood performance.) This seems somewhat similar to the fact that discrete latent variable generative models usually do not perform better than its continuous counterpart, but more interpretable. It is in fact not clear whether this learned structure should improve the likelihood performance as well because, unlike other architecture learning problems like NAS, our problem imposes particular temporal hierarchy structures, and thus severely constraints the model space. **Investigation over latent spaces:** We agree. We will add more analysis such as Fig 1 where the temporal abstraction latent z has the subsequence-level context such as *color, direction* and *length*. Fig 2 shows how the subsequence are generated from the same z (but different s, m) and the *velocity* of ball and the *length* of subsequence can be varied.

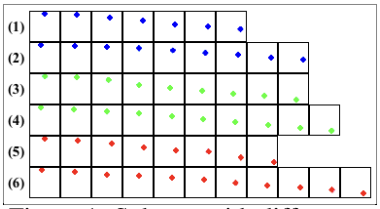

Figure 1: Subseq. with different z

[REVIEWER #2] Interpretable results without color change: Yes. Because a random color is selected at bouncing from a color set of size 3 including the current color, actually it does not change its color with probability $1/3$. Even in this case, we observe the model cut the segment at bouncing the walls. **Quantitative comparison to a baseline:** This is described above (L1-13) with some explanations. **Maze without action-conditioning:** Yes, we initially also trained the model without action-conditioning and it showed a very similar segmentation result. Due to space limitation, we couldn't include this result, but we will add this result in the Appendix of the camera-ready. We agree that actions and observations can both affect the structure of the segmentation, particularly in a complex way if they are not consistent each other. **RSSM baseline:** RSSM is a single layer version of our model. It is implemented by using the same architecture as the observation-level of our HRSSM and by removing conditioning on z and m (L118-119 in the paper). **Training variation:** The training curve changes rather highly during the early stages of the training as the model searches for a stable temporal structure from $q(M|X)$. It would have been more stable if the temporal structure was given or fixed like VHRED instead of learning it as we do. **Prior over segments:** Our prior is designed to regularize $q(M|X)$ to avoid *over-* and *under-segmentation*. This is done not by explicitly changing M or $q(M|X)$ but by regularizing it through KL and generation terms. That is, if $q(M|X)$ assigns segments that exceeds the limit defined by the prior, it will lead to lower ELBO via the KL term. We, however, agree that more explicitly controlling the posterior class with the segment limits is worth to try. **Independence of M binary indicators:** We agree that giving more structure and conditioning to $q(m_t|X)$ is also worth to try. Nonetheless, we would like to say that m_t is not fully independent but is independent conditionally after observing X. Thus, we believe that, although it is somewhat indirect, each m_t can still see the global temporal dependencies by observing the full sequence X.

[REVIEWER #3] Forcing the model to produce a new subsequence at each time step: Although this is used during training, at test time, we only use the UPDATE operation at the z level without generating subsequences. In experiments, we show that this sequence of z's are good abstract representation of the future and we believe that this is a principled way of performing jumpy imagination in the sense that the formulation is based on the (recurrent) abstract state-space transition model. Also, like other works on imagination-based planning or RL agents, we assume that the test time environment is similar to the training environment so that the learned environment model is useful at test time. So, we do not expect the model to generalize when the test time environment is much different from training distribution. That is, in a new environment, we should not rely on empirically-learned (inductive) imagination until it collects and completes learning from the new environment. **Importance of stochasticity** In our experiment on the navigation task, we observed that we can actually rollout multiple future imaginations from the same state by sampling multiple rollouts. This would not be possible in HMRNN as its rollout is deterministic. This result is obtained when we train the model without action-conditioning. With action-conditioning, we found the model uncertainty reduces and not generate various futures, which is what can be expected. We will add this result in the camera-ready. **Outside image domain:** We agree that applying the proposed model to other domains like text or speech signal would be interesting. For this work, as our main focus was to apply it for agent learning, we put our priority on the image domain and the planning task. **RL experiments which are directly comparable with prior work.** Due to limited time and as our focus was on the structure learning which can also be evaluated by the planning agent without RL, in this work we could not highly prioritize the RL experiment. We, however, agree that it will be helpful in making the paper more complete. **Gumbel softmax:** We used temperature annealing (from $\tau = 1.0$ to $\tau = 0.5$). Without annealing, it was unable to train the model as the entropy of $q(M|X)$ didn't properly decrease and showed meaningless random segmentation.

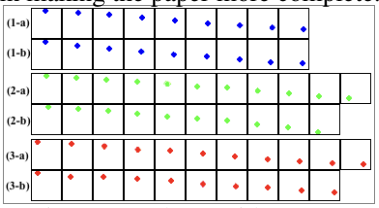

Figure 2: Subseq with same z