We thank the reviewers for their thorough reviews and valuable feedback. We will address the concerns as follows.

(R2): Approximate Inference and trade-offs. We approximate the posterior of assignments and model parameters via MAP to fulfill the time constraint in online learning tasks. The argmax/hard assignment approximation splits the streaming data and thus accelerates the stochastic update of model parameters and simplifies the form of transition prior. However, this involves a trade-off between how well the data are balanced and the density accuracy. The proposed split-and-merge mechanism explicitly leverages the expressiveness of GPs while lacking theoretical guarantees, e.g., escaping local modes. We will clarify in revision. We appreciate the suggestions and will explore split-merge MCMC [Jain and Neal, 2004] and memorized online VI [Hughes and Sudderth, 2013] in the future.

(R3, R4): More complex environments. The exact equivalence between infinitely wide DNNs and GPs was derived by Lee et al. [2017]. Considering that DNNs require optimizing a large number of parameters, in general, the data efficiency of GPs should be higher than DNNs even in high-dimensional environments. However, the space and computational complexities of GPs dramatically increase along with the input dimension, which may deteriorate real-world performance in complex environments. We plan to incorporate advanced models, including Deep Kernel Learning and Neural Processes, into our method to efficiently handle high dimensional data in future work.

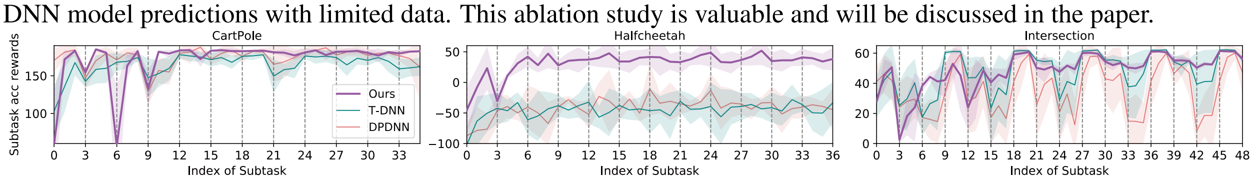
(R3): Lack of derivation of equation 2. (R4): Sometimes hard to understand algorithm design choices. To make our algorithm easier to follow, we will (i) add the derivation of equation 2 and show that the optimal $\rho_n(z_{nk})$ is the marginal distribution that minimizes $KL(p_n(z_{0:n}, \boldsymbol{\theta}|\mathcal{D})||\hat{q}_n(z_{0:n}, \boldsymbol{\theta}))$, where p_n is the sequentially decomposed posterior and $\hat{q}_n$ is a sequence of variational approximations (line 123); (ii) extend algorithm descriptions and clarify correspondences between motivations and algorithm design choices in Section 4. For instance, as mentioned by **R2**, we will clarify that we use z_i to approximate ρ_i to simplify the transition prior's form and decrease the computation burden.

(R4): Impact of hyperparameters or good values. We summarized the key parameters and the best selections in Table S2 and described the impact of DP and GP parameters in Section S2.1 line S57-S65. For the hyperparameter selection, we randomly search in a coarse range first and then do a grid search in a smaller hyperparameter space. We will clarify the good value selection and describe the impact of MPC parameters in revision.

(R4): Evolution of the mixture size. Since the mixture size increases when a new type of task is detected, the increasing rate depends on the natures of real-world applications, e.g., environment changing pace. We agree that the mixture size may explode, which is a side effect of the desired capability of modeling an infinite number of tasks. The proposed merge and prune method helps control the increasing rate by eliminating redundant ones. We will add a size modulation mechanism in revision, such as setting a threshold based on computational burden and periodically compressing the mixture based on distance measures. We will focus on mixture compression methods in future work.

(R4): A typo in equation 3. We are grateful that you pointed out this typo. We will update the first case in equation 2 to $q_n^{pr} \propto \sum_{i=1}^{n} \mathbf{1}\{z_{i-1} = z_{n-1}\}\rho_i(z_{ik}) + \mathbf{1}\{k = z_{n-1}\}\beta,$ if $0 \le k \le K_{n-1} - 1$, where β is only added to the prior of model $k = z_{n-1}$. We confirm that the algorithm was implemented correctly by checking the code, and thus the experiment results still hold. The subscript of summation in q_n^{pr} refers to the index of collected data points. We will fix the notations in Figure 1 with $\pi \to q^{pr}$ and $x \to \tilde{x}$.

(R4): Data efficiency. The DPDNN baseline is different from [8] in terms of DNN initialization (line 220-223). Our *pre-train free* method achieves higher rewards than DPDNN pre-trained in each task (line S76-S81), which shows that our method is more data-efficient than DPDNN. As suggested, we add a DNN mixture with the transition prior baseline (T-DNN) that is also pre-trained. The results show that T-DNN underperforms our method due to the inaccuracy of DNN model predictions with limited data. This ablation study is valuable and will be discussed in the paper.



(R4): Related work. Reference [23] supports that model-free methods may be impractical due to data inefficiency in real applications. We will clarify this and emphasize that [8] is a model-based meta-learning method in Section 2.

References

Sonia Jain and Radford M Neal. A split-merge markov chain monte carlo procedure for the dirichlet process mixture model. Journal of computational and Graphical Statistics, 13(1):158–182, 2004.

Michael C Hughes and Erik Sudderth. Memoized online variational inference for dirichlet process mixture models. In Advances in Neural Information Processing Systems, pages 1133–1141, 2013.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. arXiv preprint arXiv:1711.00165, 2017.