We thank all the reviewers for their constructive and useful feedback! Below, we address the key comments in order.

R2: Classification parameters are independent of other classes, similar to prototypical models. We thank the reviewer for pointing out this important issue. The classifier component of the inference network is similar in approach to prototypical models, and in fact, [5] shows that prototypical networks are recovered as a special case. Surprisingly, this approach is optimal in certain circumstances, and there is a principled justification for this design choice. This is best understood through the lens of density ratio estimation [i, ii], which shows that an optimal softmax classifier learns the ratio of the densities: $\text{Softmax}(y = k|x) = p(x|y = k)/\sum_j p(x|y = j)$, assuming equal a priori probability for each class. Our system follows this optimal form by setting: $\log p(x^*|y = k) \propto h_\theta(x^*)^T w_k$, where $w_k = \psi_\phi(\{x_n\})$ for each class in a given task. Here $\{x_n\}$ are the few-shot context examples for class k, and x^* is the target example. This argument states that under ideal conditions (i.e., we can perfectly estimate $p(y = k|x)$), the context-independent assumption is correct, motivating our design. A discussion on this point is in [5], Appendix B.1.

R2: AR component is not technically motivated. Comparison to additional forms of modulation? To adapt layer l, the system must have access to the representation of task relevant inputs at layer $l - 1$. While z_G will encode how layer $l - 1$ has adapted, it is useful to directly observe the representation of the context set at layer $l - 1$. This is similar to the linear classifier adaptation, which leverages the task-specific feature representation. We indeed performed a study comparing FiLM layers with parallel residual adapter methods [17] (which have significantly more capacity) using the non-AR variant, with no gain in performance. These experiments indicate that the AR approach is superior to naively adding capacity to the adapters. We will provide results from this ablation study as suggested.

R2: z_G is computed using only the inputs from the query set, what about the labels? We agree that performance could potentially be improved by utilizing the labels from the context set (e.g., MAML [7] and LEO [23]). In our design, we took inspiration from unsupervised pre-training approaches to adapting feature extractors. This is arguably more limited but has advantages e.g. it can be used for semi-supervised learning (with the unlabeled examples used to adapt the feature extractor). However, this is indeed an avenue for future exploration.

R2: The justification for the training procedure is weak... We find that it is crucially important for the inference networks to *learn to adapt* the feature representation at test time. Joint training results in a network that captures the training data making minimal use of the adapters, as the feature extractor is very flexible and learns features that are suitable to solve each training task with limited adaptation. This network generalizes poorly since it has not experienced data sets for which the feature extractor requires significant adaptation. The two stage training fixes this issue by forcing the adaptation to occur in the second stage just as it must at test time. An ablation study is provided in Table D.4, demonstrating significant differences between the two approaches. We will expand this discussion point in the revision.

R2: Pretraining θ requires a large dataset, which is not always available. We agree that this is an assumption / limitation of our method. However, in this work we are mainly concerned with the vision domain, where pre-trained networks are readily available, and indeed, it is necessary to leverage them to achieve SoTA performance. The assumption of large related corpora carries to other interesting domains e.g., speech, NLP, and recommender systems.

R2: Results section is a bit unfocused. Experiments on active and continual learning seem forced. Continual learning was a major point of motivation in this work. We will improve the writing so that that the exposition better reflects this, as it shaped many of the design decisions. Further, we wanted to demonstrate the flexibility of our approach; continual learning results show that our method outperforms a very strong base line with far fewer shots (which we see as an important result), and the active learning results suggest well calibrated uncertainties.

R3: The major negative point of this paper is its similarity with CNPs. We fully agree that CNAPs *is* an extension of CNPs, with design choices targeted towards multi-task classification and continual learning. In particular, CNPs cannot handle varying-way tasks (see [13] Sec. 4.3) as required for the meta-dataset task and continual learning. Further, increasing (pre-specified) way results in (at least) linear growth of parameters in the decoder. In contrast, CNAPs handles varying-way tasks on the fly with a fixed number of parameters via the classifier adaptation network. We emphasize further differences: (i) CNAPs employs a parameter sharing hierarchy (global / task / class); (ii) ψ^τ directly parameterizes the feature mapping as opposed to being a fixed-dimensional input to the decoder (as in CNPs); (iii) the dimension of ψ^τ increases with the number of classes in τ; and (iv) CNAPs employs a meta-training procedure geared towards *learning to adapt* to diverse tasks. We will improve the discussion and add an ablation study for point (ii).

R3: I think the article would gain from putting a bigger emphasize on the auto-regressive adaptation. We agree that the AR adaptation is under-emphasized, and will add detail and emphasis on this aspect of the model.

Additional References (numbered references correspond to main paper bibliography)

[i] S. Mohamed. The Density Ratio Trick. The Spectator (Blog). 2018

[ii] M. Sugiyama, T. Suzuki, and T. Kanamori. Density ratio estimation in machine learning. 2012.