R1 (Not Confident):

1. The contribution of this paper is limited. The major strength of the proposed method is its computational acceleration. But does speed really constitute an important bottleneck for previous Recurrent Attention Model (RAM)? The experiments only compared the number of steps of RAM and DT-RAM. So, what are their actual inference speed? If the speed is not a bottleneck for RAM, then why we need a dynamic extension? The authors need to explain their contribution in more details.  
  
2. Unfair comparison with previous published methods In Table 3 and Table 4. Different CNN models are adopted in different methods. RAM and DT-RAM have the highest score, since they use ResNet-50. However, there lacks necessary explanations, which may mislead reader's judgement.

The proposed method mainly to accelerate the inference speed of RAM. It would be better if the authors could explain some issues.

R3:

The key concern is that the work is incremental. The additional innovations over RAM are only (1) an extra binary variable for deciding whether to stop taking further attention and (2) adding intermediate supervision after every action. The binary stopping variable is very similar to e.g., the binary variable used in [30] to determine whether to output a prediction after every action. Intermediate supervision (and/or reward shaping) is well-known to be critical in reinforcement learning.   
  
Similarly, the results demonstrate marginal improvement: (1) with the same number of computational steps, the accuracy improvement over RAM is 0.1-0.2% on MNIST (Table 2), 0.3% on CUB-200 (Table 6), (2) the reduction in computational cost is < 50%: 1.9 steps vs 3 steps on Stanford Cars (Table 4), 3.6 vs 6 steps on CUB-200 (Table 6).

The work is well-executed; however, the paper is too incremental to warrant publication in ICCV in my opinion. There are no novel insights and only marginal improvements in accuracy and efficiency.

We thank all reviewers for their valuable feedback. We are happy to see that all reviewers think the paper is overall clear and well written. R2 is positive with our work, we mainly address the questions raised by R1 and R3.

To R1 and R3: (The contribution of the paper)

The main purpose of this paper is the study of learning and inference with a dynamic structured model on a real-world computer vision problem. Previously, most deep learning models use a fixed model structure with a pre-defined number of layers and channels. Such model structure is static with respect to the input hence will not adapt its computational complexity during inference no matter how hard or easy the input is.

With the development of deeper and more complicated neural networks in recent years (i.e. Neural Turing Machines, Stacked Hourglass Networks, Feedback Neural Nets, etc.), we expect there is a need for developing a more flexible architecture where the network itself has the ability to adapt its computational complexity during inference based on the input signals. This motivates our work. Related works with a similar dynamic computational idea [11] lack experimental study on real-world problems. We pick visual attention model and fine-grained recognition problem as our starting point. In this paper, we not only demonstrate the potential usefulness of a dynamic model, but also reach the state-of-the-art results on fine-grained recognition. The inference time of the baseline RAM is about 150ms when recognizing a bird image, and DT-RAM halves the inference time.

As R2 points out, our reinforcement learning framework is general hence can be potentially used for other applications. We think problems such as visual question answering and visual dialog which require more sophisticated reasoning processes could be its future applications, where the reasoning process heavily depends on the input image as well as questions and contexts. (For example, one may imagine the model shall use different structures when answering an "how many objects in an image" question or a “how are you” question.)

Even when compared with previous work such as recurrent attention model, the original RAM paper still lacks experimental study on its implementation details for real-world problems. For example, it would be interesting to know how the whole reinforcement learning process gets affected by different training factors on fine-grained recognition. We provide a systematic experimental analysis of reinforcement learning in this context, which could potentially be useful for the whole research community.

R3: (Incremental Improvement)

We pick the fine-grained recognition problem since its benchmark is well established. Although we only achieve incremental accuracy on MNIST, the MNIST dataset itself is already exhaustively explored. The purpose there is to show that DT-RAM can achieve comparable or better performance than its baseline RAM under fair comparison. On CUB-200 and Stanford Car, we study even further. We show that our method not only improves baseline RAM model, but is actually quite competitive against almost all previous state-of-the-art fine-grained models. (86.0% on CUB-200 and 93.1% on Stanford-Cars are so far the highest accuracy without utilizing extra dataset. We have also released the code for reproducing the results in the paper.)

R1: (Unfair Baseline)

In the fine-grained recognition literature, different baseline networks are used to extract features for different models such as Bilinear CNN [54] and Spatial Transformer Networks (STN) [59], and the highest score of each method is reported. Although we use ResNet-50, ResNet itself may not always guarantee the best performance for fine-grained recognition. For example, the bilinear model [54] demonstrate that VggNet is better than ResNet when using a bilinear model structure and the best baseline model for STN [59] is Inception-v2.

R2: (Complex Training Pipeline)

We admit the whole training process is still quite complicated (and slow) in order to guarantee the final performance. We think this is worth for future study in the deep reinforcement learning field. We thank the reviewer for pointing this out.