Rebuttal for ID 9805. We thank all reviewers for their thoughtful comments.

[R1.1] *"The method is only compared to prior models with long-term memory on the [QA] task, and doesn't perform as well as the prior MemN2N or EntNet models on that task."* This is expected as these are ML models with non-biological features. Our goal was to show that simple local Hebbian plasticity can be utilized to solve many of these tasks. We will also discuss in the revision how the model could be extended (see below). **[R1.2]** *"Is it essential that the key-value matrix is Hebbian? [...] Would it also work just as well to train the key-value matrix in the same way [with backprop] but with a higher rate?"* This would not work during inference (after optimization) as there are no targets and hence no error-gradients. It is a separate (interesting in ML context) question whether one could find surrogate loss functions for memory. Our goal was to show that simple *local* plasticity is sufficient for many tasks. Hebbian plasticity is a natural choice for this both from a functional and biological perspective. **[R1.3]** *"How and why do the query and storage keys differ?"* See analysis in [R3.1]. **[R1.4]** *"[...] isn't it possible to achieve good performance on the tasks in the paper with simple non-neural methods (e.g. storing key-value pairs of embeddings directly and finding the most similar key at lookup time?"* This approach is rather close to the approach of MemN2N. Note however that it is still essential for many tasks to (a) optimize the embedding (b) perform several memory queries, and (c) post-process retrieved items. **[R1.5]** *"[...] it would be helpful to explain the practical or physiological relevance in more detail."* Due to space constraints, we did only briefly discuss these questions in lines 259–270. Will be expanded in the revision. In brief: (practical): (a) The simplicity of the model leaves a lot of room for extensions for interesting paradigms of memory-based ML. (b) Current energy-efficient neuromorphic hardware cannot implement previous Memory-augmented NNs. On the other hand, Hebbian plasticity is already implemented in hardware (e.g., Intel's Loihi chip). (Neuroscience): We did deliberately not relate the model to brain anatomy, as the organization of higher-level cognitive functions is still very much unknown. However, Hebbian plasticity is well-supported by many experimental findings. In particular the Hippocampus might play a pivotal rule for implementing a memory module as in our model. More generally, our study provides a first link between research on memory-NNs and biologically plausible models of cognition.

[R2.1] *"Perhaps the authors could describe what modifications of the associative module (including non-biological inspiration) might further improve performance, or how the task-control vs memory dichotomy could be more widely used in neural network research."* Indeed, we believe that several rather simple extensions should improve model performance (e.g., solve all bAbI tasks). First, computation of the key- and value vectors during storage could be made dependent on the memory content (with a prior read). Second, to compute the output, the network cannot directly integrate several recalled values. This ability seems essential for example in bAbI task 19 (path finding). Using a recurrent network at the output (instead of W_{out}) could solve this issue. We will discuss these and further possible extensions in the revised version. **[R2.2]** *"[...] Le, Tran, Venkatesh, Self-attentive Associative Memory, ArXiV 2000, [...]"* Will be discussed. **[R2.3]** *"The section of the discussion addressing biological plausibility and spiking neural network applications seems disjointed from the majority of the paper. Dropping this section would allow the authors to instead discuss what modifications of the associative memory module might be of interest in future research."* Since one more page can be used for the revised version, this will not be necessary. See also [R1.5].

[R3.1] *"[...], why can the model work on the two tasks? [...] A reason, or intuition, from the perspective of machine learning should be provided."* We believe that the model learns to extract the relevant key-value pairs from the input and stores that in memory. At a query, it learns, which keys are essential in order to retrieve the informative values. We performed additional analysis to test this idea by computing the cosine similarity of the recall key (resp. the recalled value) to keys (and values) of previous storing operations in bAbI task 1. The similarity of keys was 0.996 ± 0.004 for sentences with the same person as the person in the query (0.020 ± 0.028 otherwise). For values, the similarity was 0.981 ± 0.026 for sentences with the same place as the answer place (0.323 ± 0.119 otherwise). This indicates that the model learned to associate persons to places. Similar results hold for the image association task. This analysis will be extended to other tasks and discussed in the supplement. The following points will be clarified in the revised version: **[R3.2]** *"[...], are the image pairs [...] presented as a whole to the store branch, or separately [...]?"* They are presented as one image as indicated in Fig. 2. The model learns which part of the image is important for the key and the value, respectively. **[R3.3]** *"How large is M?"* We always used 3 image pairs as indicated in Fig. 2. **[R3.4]** *"What's the definition of an epoch? [...], how many samples are in one epoch and what's the definition of one "sample"?"* A sample is one full sequence of image pairs (including random images) and one query image (see Fig. 2, bottom right). One epoch consists of all samples of the training set (12500 samples). **[R3.5]** *"[...], in the first task, during the presence of the first 3 pairs of images, is there a teacher signal for the output? [...], is there a teacher signal during the delay?"* There is only a teacher signal after the query. Then the output is computed, compared to the target, and the error is computed. **[R3.6]** *"[...], it is said that W^{assoc} is not optimized, but how is (1) implemented during the training process."* W^{assoc} is a dynamic variable just like neuron activations. The matrix is updated according to Eq. (1) after the key- and value-vector has been computed in the store-branch (see Methods). Will be discussed in the revised version.

[R4.1] *"I would like to see a brief conclusion section. The papers seems to just trail off. [...] Few, minor grammatical errors."* We will add a conclusions paragraph to the discussion and proofread the manuscript for grammatical errors.