	Baseline	Parameter-retune		Memory-based		
Name	LSTM	Rei	MAML	DNTM	LMN-fixed	LMN
Brightkite	10.7 (0.11)	10.01 (0.18)	4.27 (0.47)	9.63 (0.13)	3.88 (0.51)	**3.57 (0.55)**
FSQNYC	8.95 (0.03)	8.72 (0.07)	6.55 (0.16)	9.01 (0.05)	6.13 (0.25)	**5.54 (0.27)**
FSQTokyo	8.14 (0.08)	6.95 (0.13)	5.68 (0.23)	7.25 (0.11)	5.40 (0.26)	**5.32 (0.28)**
Geolife	1.13 **(0.84)**	1.08 (0.83)	-	1.11 (0.82)	**1.05** (0.83)	1.08 (0.83)
Yoochoose	5.01 (0.24)	5.05 (0.23)	-	5.01 (0.23)	5.01 (0.24)	**4.96 (0.25)**

Table 2: Log Perplexity and MRR(in parantheses) on online sequence prediction tasks

independent prediction space, we consider the more useful and challenging task where new labels co-exist with labels seen in the batch-trained PCN.

Dataset and setup We use the popular omniglot dataset (**?**). The base data-set consisted of 1623 hand-drawn characters selected from 50 different alphabets. (**?**) extended the base data-set by various rotations of the images, and this increases the number of categories to 4515. We create an online variant of this dataset as follows. We arbitrarily select 100 classes as unseen and 250 as seen classes. During training only seen classes are provided, but the test data has a uniform mix of all classes. Each test sequence is obtained by first randomly selecting 5 labels from the 350, and then choosing different input images from the selected labels up to a length of 10. Thus, in each sequence we expect to encounter $\frac{10}{7}$ new labels on average. Accuracy is measured only over second occurrence of each of the five labels much like in one-shot learning experiments. The one major difference is that in our case the prediction space is all the seen labels and the expected new labels, whereas in few-shot learning the prediction space is only the 5 labels chosen for that test sequence. The results are averaged over 100 sequences.

Methods compared We compare results of LMN with MAML (**?**) the most recent meta-learner that can work in this setup. We do not compare with the two recent MANNs (**?**; **?**) that report Omniglot results on few-shot learning because their techniques do not easily extend to the case where new labels share label space with a pre-trained classifier. For reference we also report accuracy on the baseline classifier that will certainly mis-classify examples from the new class. We use as PCN a convolutional network, with the same configuration as used in (**?**).

.

Model	Overall	New labels	Seen labels
Baseline	45%	0	63%
MAML	56%	49%	59%
LMN	**86%**	**71%**	**92%**

Table 3: Accuracy on Online adaptation for Omniglot

Results In Table 3 we report our results. The baseline that does no adaptation achieves an overall accuracy of 45% while obtaining an expected accuracy of 0% on the new labels and 63% on the seen labels. MAML boosts the overall accuracy to 56%, while obtaining an accuracy of 49% on the new labels. However, compared to the baseline the accuracy on seen labels has dropped. This is similar to catastrophic forgetting (**?**) and may be because during meta-learning updates the weights of the shared representation layer deteriorate. The LMN architecture is better able to absorb new labels and changing priors of existing labels. LMN achieves an overall accuracy of 86% with 71% on new labels. The accuracy increases for the seen labels as well because LMN is able to use the memory for storing prototypes of examples with non-zero loss even from seen labels. Thereafter, the combiner RNN can pay more heed to the labels seen during the adaptation phase.

Plain few-shot learning To enable comparison with the few-shot results of many recent published work, we also report results of LMN on this task. In few-shot learning, the PCN only generates the input embedding $\mathbf{h}_t$ and does not assign label scores since each test sequence has its own label space. The memory uses the embedding to assign label scores via Equation 3 based on which prediction is made without involving any combiner either. These experiments will compare LMN's memory organization and update strategies with state-of-the-art MANNs that have reported results on few shot learning (**?**).

Our setup is the same as in recent published work (**?**) on few-shot learning but where the total label space has 4K possibilities labels. This is more interesting and realistic than 5-way classification where many recent methods, including ours, report more than 98% accuracy with just 1-shot.

We present results with changing memory size. We run the experiments with a memory size equal to T-cell per label (T=2,3..). We observe that LMN accuracy is much higher than existing MANNs particularly when memory is limited. For example, at 2-cells per label (roughly 8K memory size) we obtain more than 4% higher accuracy in 1-shot, 2-shot, and 3-shot learning. This proves the superior use of memory achieved via our labeled memory organization

For reference we also compare with parameter retuning-based approach (MAML) (**?**). Even though this work reports state-of-art accuracy on 5-way few-shot learning, for 4k way few-shot learning it is not as effective. A softmax layer that has to arbitrate among 4k new classes perhaps need lot more gradient updates than learnable by meta-learners.

Name	1 shot	2 shot	3 shot
Kaiser (2-cell/label)	48.2%	58.0%	60.3%
Our model (2-cell/label)	**52.6%**	**62.5%**	**64.1%**
Kaiser (3-cell/label)	49.7%	60.1%	63.8%
Our model (3-cell/label)	**52.8%**	**63.0%**	**67.0%**
Kaiser (5-cell/label)	52.7%	63.0%	66.3%
Our model (5-cell/label)	**54.3%**	**63.2%**	**66.9%**
MAML	44.2%	46.5%	47.3%

Table 4: Test accuracies for 4k way few shot learning

Name	Wikitext2 (100)	Text8 (2000)
LSTM	99.7	120.9
Pointer LSTM	80.8	-
Neural cache	81.6	99.9
LMN	**77.6**	**91.1**

Table 5: Test perplexity for language modeling on Wikitext and Text8 with memory 100 and 2000 respectively.

Language Modeling

Language modeling is the task of learning the probability distribution of words over texts. When framed as modeling the probability distribution of words conditioning on previous text, this is just another online sequence prediction task. The natural dependence on history in this task provides for another use-case of memory. Memory based models have been shown to get improvements over standard RNN based language models (**?**; **?**; **?**). In the same spirit, we apply LMNs to this task by taking the PCN as a RNN.

We compare our model directly with the recently published neural cache model of (**?**) and pointer LSTM of (**?**). These showcase variant uses of memory to improve the prediction of words that repeat in long text. The baseline model (and PCN) is an LSTM with same parameters as in (**?**). We compared on common language datasets Wikitext2 and Text8 with memory sizes 100 and 2000 as used in the previously published work. We used standard SGD as the optimizer.

In Table 5 we report the perplexities we obtained with LMN along with the results reported in published work for other three approaches. As the table demonstrates we achieve state of the art results in Text8 and Wikitext2. In LMN the auto-modulation caused by considering only cases where the PCN is weak is superior to memory cache. This shows up in tests when memory is constrained, when the focus on mis-predicted outputs in LMN allows for boosted recall, efficient memory utilization, and capturing longer contexts, compared to other models.

Conclusion

We extended standard memory models with a label addressable memory module and an adaptive weighting mechanism for on-line model adaptation. LMNs mix limited data with pre-trained models by combining ideas from boosting and online kernel learning while tapping deep networks to learn representations and RNNs to model the evolving roles of memory and pre-trained models. LMNs have some similarities to recent MANNs but has significant differences. First, we have a label addressable memory instead of content based addressing. Second, we use memory to only store content on which primary network is weak. Third, our model has a loose coupling between memory and network, and hence our model can be used to augment pre-trained models at a very low cost. Fourth we use an adaptive reweighing mechanism to modulate the contribution of memory and PCN. This LMN is demonstrated to be extremely successful on a variety of challenging classification tasks which required fast adaptation to input and handling non-local dependencies. An interesting extension of LMNs is organizing the memory not just based on discrete labels but on learned multi-variate embeddings of labels thereby paving the way for greater sharing among labels.

Acknowledgements We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

Aliquid quia autem provident optio aliquam, in a nesciunt quas dolore impedit id, voluptates culpa numquam perferendis quae asperiores cupiditate dolore nobis sunt reiciendis obcaecati, sequi optio ipsa.Deleniti eos facilis unde quam ducimus perspiciatis dicta esse, ab exercitationem autem error recusandae, necessitatibus distinctio aliquam aut illum placeat corrupti nesciunt?Aspernatur illo voluptate, quo est non beatae cupiditate sit ratione.Incidunt voluptate optio blanditiis nulla delectus, animi omnis officiis beatae autem illum distinctio aut mollitia molestiae, dicta velit qui cupiditate beatae itaque cumque ipsam accusamus ratione nam, laudantium in sed quasi pariatur porro dolor quidem earum.Alias nostrum fugiat enim aut tempore tempora ab, cupiditate aliquid sequi perspiciatis dolor beatae voluptatibus velit ipsa, eaque qui repudiandae maiores doloremque est ipsam dolorum officia laudantium, nisi ut saepe minus vitae recusandae ullam ea optio ducimus aliquid.A facere officiis quae unde enim nobis quos optio, veniam vel dolorem aperiam temporibus alias omnis deserunt.Et repellendus fugit repudiandae possimus alias eaque impedit facere amet sequi, minus neque deserunt eius repellat iusto soluta eos iste libero est quasi?Nostrum aut error excepturi corrupti dolores vel consectetur architecto incidunt, quia dolore iure assumenda nobis incidunt optio deserunt officia, consequatur voluptate nostrum deserunt ratione maiores iusto eaque pariatur, minus molestias autem eum animi, voluptatem facere eveniet voluptate quia est placeat blanditiis assumenda.Veritatis laudantium quidem saepe vitae aliquam corporis, facere impedit illo facilis, ipsum corrupti at dolore neque aliquid voluptates necessitatibus cum laudantium nemo.Itaque nihil veniam id cumque perferendis, deleniti odit ipsa?Perferendis ducimus tenetur laudantium deleniti quasi laborum, quas sequi autem rem deleniti expedita, ea inventore facilis recusandae nesciunt eius tenetur veniam sequi cumque, illo magni nam voluptatem qui est culpa?Atque modi veritatis molestiae tempora dolorum nostrum tenetur officia iusto dolore nam, vel eum