We thank the reviewers for their detailed and insightful reviews. We will incorporate all presentation changes as suggested. Below we answer the main questions raised by the reviewers.

Evaluation on Larger Grammars (all reviewers) The reviewers note that the results section in the submitted paper appears inconclusive, or that the experiments have been run on very small languages and with small alphabets. Sadly, the algorithm is indeed not currently applicable to large 'complicated' languages, although (as we present below) it is generally more successful in these attempts than the spectral algorithm. We note that the results in the paper do show that on the synthetic SPiCe and Tomita grammars, the L*-learned PDFA (and occasionally spectral-learned WFA) outperform n-gram, presumably because of the ability of finite state machines to capture patterns that an n-gram is unable to encode. We believe that while the algorithm is not scalable to large grammars, it is an interesting new step into the field of WFA extraction, and worth sharing with the community. We hope that in the future others will be able to work on and expand this algorithm so that it is useful even for natural languages.

Since the original submission, we have managed to train and extract from networks for some more complicated SPiCe grammars, in particular: SPiCe 4 (NLP), 7,10 (biology), and 6,9,14 (part-synthetic), which have alphabet sizes ranging from 11 to 60, we present these new results now. For each language we trained a network with 2 or 3 layers, and hidden dimension of size 20-100 (depending on the language). Unless stated otherwise, the hyper-parameters of the extraction were: L* with maximum $|P|$=5k and variation tolerance $t = 0.1$, spectral with a Hankel matrix of size 500x500, and n-grams with total sample length 5 million and $n \in [6]$. The extracted models' WER against their targets and extraction time are provided in the table below. Our approach outperforms spectral extraction on all but one benchmark (SPiCe 9). For SPiCe 10, allowing spectral to expand to Hankel 1000x1000 remains at WER 0.863, and takes 1.8hrs. Interestingly, the best WFAs for SPiCe 10 were always those with $k = 1$.

Name	Our Approach	Spectral	n-gram
SPiCe 4	0.318 (1.8hrs)	0.348 (1.8hrs, 1000x1000)	0.112 (0.9hrs)
SPiCe 6	0.575 (2.5hrs)	0.788 (1.4hrs), 0.682 (6.1hrs, 1000x1000)	0.274 (0.8hrs)
SPiCe 7	0.625 (0.5hrs)	0.801 (0.6hrs)	0.442 (0.7hrs)
SPiCe 9	0.485 (0.5hrs)	0.287 (0.4hrs)	0.116 (1hr)
SPiCe 10	0.646 (0.9hrs)	0.865 (0.4hrs), 0.863 (1.8hrs, 1000x1000)	0.347 (0.8hrs)
SPiCe 14	0.329 (1.3hrs, |P|=10k)	0.612 (1.6hrs)	0.075 (0.8hrs)

Table 1: Word error rate (WER) of extracted models for larger languages. Our approach outperforms spectral extraction in all but one benchmark (Spice 9).

Effect of Hidden Size of Networks on Extraction (rev4, rev5) All of the algorithms evaluated in the paper are agnostic to the internal structure of the language model under extraction (in our case, an RNN), and in particular to the size of its internal state (i.e. hidden size for RNNs). Note that the experiments presented above have networks with larger hidden size (20-100) than shown in the paper, to allow learning of the more complicated languages.

Choice of Hyper-parameters, and their effect of Hyper-parameters on Results (rev4) For the variation tolerance parameter, the original heuristic was to set $t = 1/|\Sigma|$. The intuition for this was that given no data at all, the fairest distribution one can give to tokens is the uniform distribution, and so this may also be considered the threshold for whether a token is deemed 'likely' by a given model or not. From this we extrapolate that a reasonable threshold for significant difference between the probabilities of two tokens is also the uniform probability, though for larger alphabets we may quickly change this to $1/n$, where n is an estimate of how many tokens are generally likely after any given prefix. In practice, we see in the examples above that using $t = 0.1$ already strongly differentiates even models with larger alphabets (these extractions did not reach equivalence), and so did not use smaller t. Starting out with a large t, and then reducing it so long as the model is reaching equivalence quickly, would also be a good strategy. We will more carefully research the effect of this parameter on the extraction, and add a fuller discussion and evaluation of all hyper-parameters to the paper.

Use of Equivalence Queries and Handling of Counterexamples (rev4) Reviewer 4 notes that it is unclear whether step 3 (equivalence queries) of the algorithm is ever used. This is a good question considering the results presented in the original submission, which did not discuss counterexamples. We note that in the extractions presented above, possibly following the addition of the thresholding for new prefixes and suffixes, the equivalence query is invoked often and successfully, regularly rejecting hypotheses. In this case, all of the prefixes of the returned counterexample are added to P, and the observation table is expanded until it is again closed and consistent. We will clarify the relevance of this step in our work by recording the number of counterexamples returned during each extraction.

Comparison to Sample-Based PDFA Reconstruction Techniques (rev5) We have not had time to prepare a comparison to these techniques for the response, but we agree that this is an important comparison to make, and will add an evaluation of one such technique to our results in the final version of the paper.