As discussed further in section 5, we can improve the performance of the trained network by using a longer schedule. We thus introduce the following schedule:

3. L(ong): $(s_r, s_{\text{noise}}, s_i, s_f)$ = (1k, 20k, 140k, 320k)

which we use to train the largest model to improve performance. When using schedule L, label smoothing with uncertainty 0.1 is introduced for time steps $< s_i = 140$k for LibriSpeech 960h, and is subsequently turned off. For Switchboard 300h, label smoothing is turned on throughout training.

3.3. Shallow Fusion with Language Models

While we are able to get state-of-the-art results with augmentation, we can get further improvements by using a language model. We thus incorporate an RNN language model by shallow fusion for both tasks. In shallow fusion, the "next token" $\mathbf{y}^*$ in the decoding process is determined by

$$\mathbf{y}^* = \underset{\mathbf{y}}{\text{argmax}} \left(\log P(\mathbf{y}|\mathbf{x}) + \lambda \log P_{LM}(\mathbf{y}) \right) , \qquad (1)$$

i.e., by jointly scoring the token using the base ASR model and the language model. We also use a coverage penalty c [29].

For LibriSpeech, we use a two-layer RNN with embedding dimension 1024 used in [25] for the LM, which is trained on the LibriSpeech LM corpus. We use identical fusion parameters ($\lambda = 0.35$ and $c = 0.05$) used in [25] throughout.

For Switchboard, we use a two-layer RNN with embedding dimension 256, which is trained on the combined transcripts of the Fisher and Switchboard datasets. We find the fusion parameters via grid search by measuring performance on RT-03 (LDC2007S10). We discuss the fusion parameters used in individual experiments in section 4.2.

4. Experiments

In this section, we describe our experiments on LibriSpeech and Switchboard with SpecAugment. We report state-of-the-art results that out-perform heavily engineered hybrid systems.

4.1. LibriSpeech 960h

For LibriSpeech, we use the same setup as [25], where we use 80-dimensional filter banks with delta and delta-delta acceleration, and a 16k word piece model [26].

The three networks LAS-4-1024, LAS-6-1024 and LAS-6-1280 are trained on LibriSpeech 960h with a combination of augmentation policies (None, LB, LD) and training schedules (B/D). Label smoothing was not applied in these experiments. The experiments were run with peak learning rate of 0.001 and batch size of 512, on 32 Google Cloud TPU chips for 7 days. Other than the augmentation policies and learning rate schedules, all other hyperparameters were fixed, and no additional tuning was applied. We report test set numbers validated by the dev-other set in Table 2. We see that augmentation consistently improves the performance of the trained network, and that the benefit of a larger network and a longer learning rate schedule is more apparent with harsher augmentation.

We take the largest network, LAS-6-1280, and use schedule L (with training time $\sim$ 24 days) and policy LD to train the network to maximize performance. We turn label smoothing on for time steps $<$ 140k as noted before. The test set performance is reported by evaluating the checkpoint with best dev-other performance. State of the art performance is achieved by the LAS-6-1280 model, even without a language model. We can incorporate an LM using shallow fusion to further improve performance. The results are presented in Table 3.

Table 2: *LibriSpeech test WER (%) evaluated for varying networks, schedules and policies. First row from [25].*

Network	Sch	Pol	No LM		With LM	
			clean	other	clean	other
LAS-4-1024 [25]	B	-	4.7	13.4	3.6	10.3
LAS-4-1024	B	LB	3.7	10.0	3.4	8.3
	B	LD	3.6	9.2	2.8	7.5
	D	-	4.4	13.3	3.5	10.4
	D	LB	3.4	9.2	2.7	7.3
	D	LD	3.4	8.3	2.8	6.8
LAS-6-1024	D	-	4.5	13.1	3.6	10.3
	D	LB	3.4	8.6	2.6	6.7
	D	LD	3.2	8.0	2.6	6.5
LAS-6-1280	D	-	4.3	12.9	3.5	10.5
	D	LB	3.4	8.7	2.8	7.1
	D	LD	3.2	7.7	2.7	6.5

Table 3: *LibriSpeech 960h WERs (%).*

Method	No LM		With LM	
	clean	other	clean	other
HMM				
Panayotov et al., (2015) [20]			5.51	13.97
Povey et al., (2016) [30]			4.28	
Han et al., (2017) [31]			3.51	8.58
Yang et al. (2018) [32]			2.97	7.50
CTC/ASG				
Collobert et al., (2016) [33]	7.2			
Liptchinsky et al., (2017) [34]	6.7	20.8	4.8	14.5
Zhou et al., (2018) [35]			5.42	14.70
Zeghidour et al., (2018) [36]			3.44	11.24
Li et al., (2019) [37]	3.86	11.95	2.95	8.79
LAS				
Zeyer et al., (2018) [24]	4.87	15.39	3.82	12.76
Zeyer et al., (2018) [38]	4.70	15.20		
Irie et al., (2019) [25]	4.7	13.4	3.6	10.3
Sabour et al., (2019) [39]	4.5	13.3		
Our Work				
LAS	4.1	12.5	3.2	9.8
LAS + SpecAugment	**2.8**	**6.8**	**2.5**	**5.8**

4.2. Switchboard 300h

For Switchboard 300h, we use the Kaldi [40] "s5c" recipe to process our data, but we adapt the recipe to use 80-dimensional filter banks with delta and delta-delta acceleration. We use a 1k WPM [26] to tokenize the output, constructed using the combined vocabulary of the Switchboard and Fisher transcripts.

We train LAS-4-1024 with policies (None, SM, SS) and schedule B. As before, we set the peak learning rate to 0.001 and total batch size to 512, and train using 32 Google Cloud TPU chips. Here the experiments are run with and without label smoothing. Not having a canonical development set, we choose to evaluate the checkpoint at the end point of the training schedule, which we choose to be 100k steps for schedule B. We note that the training curve relaxes after the decay schedule is completed (step s_f), and the performance of the network does not vary much. The performance of various augmentation poli-