Curriculum Learning

Andre Ye

June – September 2022

Contents

1 Motivation

To train speech-to-text transcription models, Deepgram uses very large datasets consisting of several dozen million samples to train large language models consisting of several hundred million parameters. Even with advanced and powerful computational resources, models still take a long time to train – at least a week, if not longer. Moreover, capturing diminishing returns in performance requires even more time. Such long training times may be problematic for two key reasons. Firstly, model development in experimental contexts proceeds much more slowly; in these cases multiple adaptations need to be observed to a sufficient clarity in order to derive useful findings. Secondly, long model training times – experimental or otherwise – occupy computational resources at the opportunity cost of other models' training and completion. The other method to reconcile this is by purchasing additional computational equipment, but this comes at a literal cost.

Therefore, there is a strong motivation to increase the training efficiency of speech-to-text models at Deepgram. Several mechanisms affecting, directly or indirectly, time efficiency have already been in use – for instance, activation checkpointing and half-precision. These mechanisms operate at a 'representation level' – they modify model representations to be more lightweight. They have yielded significant training speed gains. Another approach, however, is to operate at the 'learning level' – rather than (in addition to) improving model representation, we attempt to increase the speed at which the model converges to some validation score. Therefore, we not only provide faster ways to perform learning but make learning itself faster. Combining representation- and learning- level efficiency boosts will likely significantly improve model training speed.

2 Curricula in Literature

One idea for learning-level efficiency improvements is curriculum learning. Rather than presenting models with the entire training dataset in random order, models are selectively given samples which most benefit learning in their current state.

[3] define a curricula as being defined by two functions: a scoring function which maps samples to a scalar difficulty score (therefore allowing for them to be sorted), and a pacing function which determines when data is presented to the network. An obvious scoring function is to use 'knowledge transfer', in which an external 'teacher model' determines the difficulty of samples by its own loss. If the teacher model performs poorly on a sample, it is inferred that the sample is difficult to learn w.r.t. the remainder of the dataset. Over time, the model is exposed to some initial percentage of the easiest samples; following the pacing function, the model is exposed to progressively more difficult samples until it has iterated several times over the entire dataset. The authors find a significant accuracy improvement using curricula over random data feeding in the domain of image recognition. Moreover, they offer a theoretical model for understanding the effect of such curricula: when treating the curriculum as a Bayesian prior modifying the model objective, we can show using Empirical Risk Minimization that the global optimum is the same with and without the curriculum, but that the loss landscape becomes steeper with the curriculum – speeding up learning while respecting the goodness of solutions.

Subsequent papers successfully demonstrate the application of curriculum learning to language- and audio- related tasks. [4] show a speed performance of up to 70% when using a curriculum to train a Transformer on translation tasks, and propose sentence length and word rarity as two additional possible scoring functions. The Deep Speech 2 paper [1], an ASR model for English and Mandarin from Baidu, report a significant improvement to model performance by using a model curriculum with difficulty indicated by utterance length. However, it is worth noting that the experienced gains may be largely contingent on the recurrent architecture used. Another paper [2] uses a hard-to-easy (inverted) curriculum on a noisy ASR task to improve model robustness to low signal-to-noise ratio samples.

[5] recently proposed a theoretical model for understanding the capacity of data pruning to overcome power law scaling. Let $\alpha_{\text{tot}} = \frac{N}{P}$, where N is the number of samples in the dataset and P is the number of parameters in the model. For small α_{tot} (i.e. 'small' datasets), the authors show it is optimal to retain a high proportion of 'easy' samples – which have higher margins w.r.t. the decision boundary – to reduce overfitting. For large α_{tot} (i.e. 'large' datasets), it is optimal instead to retain a higher proportion of 'difficult' samples – which have smaller margins w.r.t. the decision boundary – to help the model better localize the boundary. Using the appropriate data pruning strategy, the authors demonstrate, significantly improves how model performance scales with sample intake. While data pruning is a 'static' curriculum – its pacing function is constant – it provides additional support for the potential of curricula to improve model training efficiency.

3 Key Findings

- Using a curriculum can significantly improve validation convergence speed.
  - A model trained with a curriculum reliably performs several WER/avg points better than a model without a curriculum for the same number of samples encountered. (See Table 1, 2.)
  - We can speed up model performance by as much as 2.5x (125m samples) to reach the same validation WER score using a curriculum. (See Figures 11 and 12, Tables 6 and 7.)
- The most successful curriculum maintains a large proportion of easier samples to support the formation of a strong, generalizable decision boundary; and gradually introduces more difficult samples to further define the intricacies of the boundary. (See Experiments section for more discussion.)
  - The easiest samples overall (hailing mainly from finance and phonecall) are the most important samples for model learning. These samples stabilize and ensure generalizability of the model's decision boundary by providing high-margin 'anchor' points and prevents overfitting.
- Ranking by model difficulty and ranking by word length perform similarly.

4 Experiments

All models used the CNN/BART architecture and were trained with 4 GPUs on the 'Leg 1' dataset, no punctuation or capitalization. Difficulty was evaluated using losses from the production transcription model, scaled by the transcription word length.

4.1 Accumulation

The accumulation curriculum proceeds as follows:

1. Order all samples in the training dataset from easiest (lowest scaled loss) to most difficult (highest scaled loss).
2. Place the samples into 20 adjacent, equally-sized bins. Bin 0 contains the $\frac{1}{20}$th easiest set of chunks, bin 1 contains the next $\frac{1}{20}$th easiest set of chunks, and so on.
3. Add the next bin (beginning from bin 0) to the dataset D and train over twice.
4. Continue until all bins have been added.

The per-bin dataset breakdown across the 50 million total chunks (with 2.5 million chunks per bin) is approximately as follows:

- *Bin 0.* 60% finance, 15% phonecall.
- *Bin 1.* 25% phonecall, 25% finance, 25% youtube, % en.
- *Remaining bins.* 45% en, 45% youtube, 5% yt-en-GB, 5% yt-en-US. (The distribution is almost identical across the remaining bins.)

We observe a significant improvement in validation WER performance when using the accumulation curriculum. From Figure 1, we visually observe that the accumulation curriculum reaches improved scores earlier than the baseline by a significant margin. We observe similar behavior for all other domains, which is especially prominent in `problem`, except `new_en`. Graphs for other domains are available here.

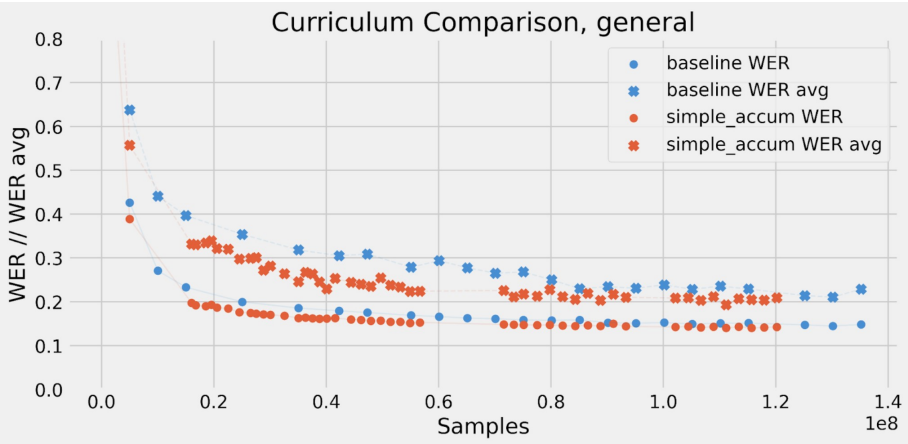

Figure 1: Validation performance on the general dataset of the same model trained with standard sample randomization (baseline, blue) and the accumulation curriculum (red), up to 140m samples.

By the time the baseline model has completed one epoch (5×10^7), the accumulation model has completed its fourth bucket addition. Therefore, it has been repeatedly learning on only the 20% of the easiest dataset. Yet this fifth of the dataset allows the model to reach the same or improved

validation WER as the more 'wordly' baseline model. As [5] observes, "not all training examples are created equal". Using their data pruning theory, we can infer that – given the low value of α_{tot} in this case, in which the number of samples is smaller than the number of model parameters by an order of magnitude – the model's decision boundary becomes more defined and generalized (in the sense of not overfitting) by wide-margin easy samples.

		aws	finance	new_en	meeting
Samples	Models				
5.00e+6	baseline	0.32 / 0.33	0.30 / 0.31	0.29 / 0.29	0.39 / 0.46
	simple_accum	0.33 / 0.36	0.22 / 0.23	0.40 / 0.40	0.37 / 0.44
1.00e+7	baseline	0.19 / 0.20	0.17 / 0.18	0.17 / 0.17	0.25 / 0.32
	simple_accum	0.33 / 0.36	0.22 / 0.23	0.40 / 0.40	0.37 / 0.44
5.00e+7	baseline	0.12 / 0.12	0.11 / 0.12	0.11 / 0.11	0.16 / 0.23
	simple_accum	0.11 / 0.11	0.09 / 0.09	0.11 / 0.11	0.14 / 0.21
1.00e+8	baseline	0.11 / 0.11	0.09 / 0.10	0.09 / 0.10	0.14 / 0.21
	simple_accum	0.10 / 0.10	0.08 / 0.08	0.10 / 0.10	0.13 / 0.20
		problem	phonecall	gmc	general
Samples	Models				
5.00e+6	baseline	0.93 / 10.57	0.48 / 0.61	0.34 / 0.65	0.43 / 0.64
	simple_accum	0.91 / 9.53	0.39 / 0.52	0.37 / 0.65	0.39 / 0.56
1.00e+7	baseline	0.76 / 9.67	0.30 / 0.43	0.21 / 0.42	0.27 / 0.44
	simple_accum	0.91 / 9.53	0.39 / 0.52	0.37 / 0.65	0.39 / 0.56
5.00e+7	baseline	0.59 / 7.80	0.18 / 0.27	0.15 / 0.31	0.18 / 0.31
	simple_accum	0.47 / 5.62	0.17 / 0.24	0.14 / 0.29	0.16 / 0.25
1.00e+8	baseline	0.41 / 4.19	0.16 / 0.25	0.13 / 0.29	0.15 / 0.24
	simple_accum	0.35 / 3.20	0.15 / 0.21	0.13 / 0.27	0.14 / 0.21

Table 1: WER / WER average statistics across eight validation domains, baseline vs. accumulation curricula.

To more rigorously compare the training speedup given by using the accumulation curriculum, we compare the number of samples of training required before reaching some WER threshold (Figure 2, 3). Across domains, we experience up to a 2x speedup and 'savings' of almost 50 million samples for small WER thresholds.

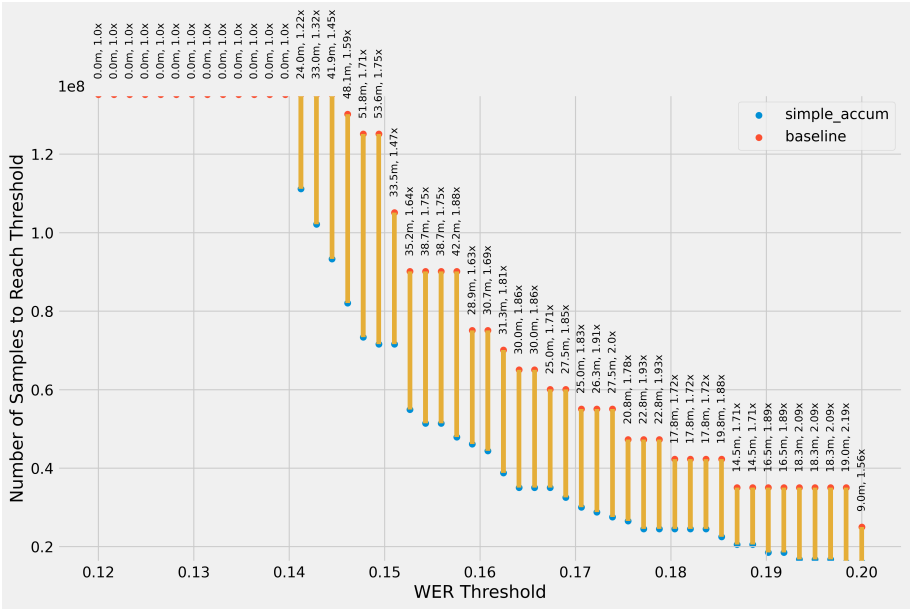

Figure 2: Differences in number of samples required to reach a given WER threshold between the accumulation curriculum and standard training, on the general validation dataset.

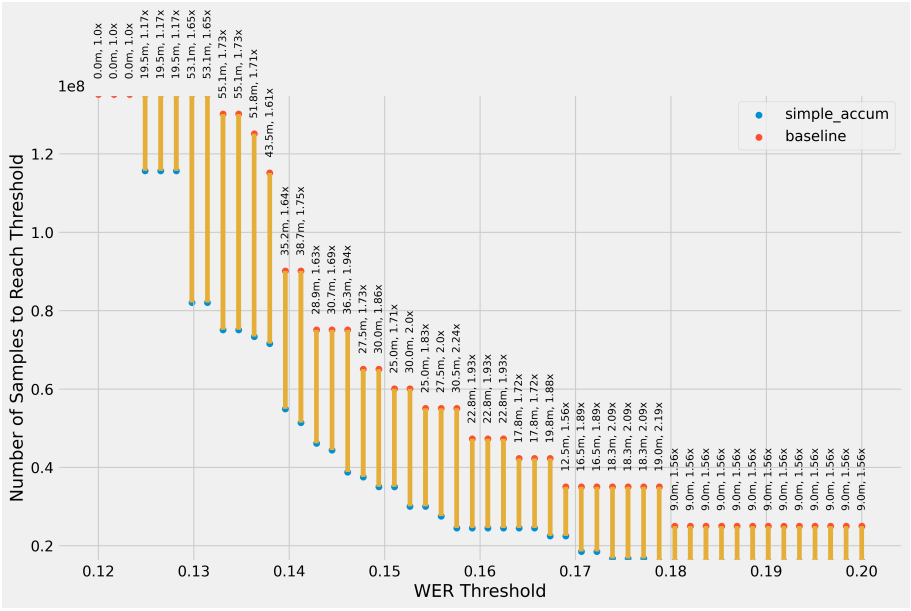

Figure 3: Differences in number of samples required to reach a given WER threshold between the accumulation curriculum and standard training, on the meeting validation dataset.

Another possibility is to rank and bucket by accumulation length rather than by a strictly model-oriented notion of sample difficulty. The advantage would be forgoing the need to pre-score the chunks. Even after loss scaling, there is a somewhat strong correlation between a chunk's loss and its length; therefore it may be possible simply to bucket samples just by length and achieve similar results. Indeed, we observe that using ranking-by-length performs similarly to ranking-by-difficulty (Figures 4, 5; Table 2). You can view the figures for all domains here.

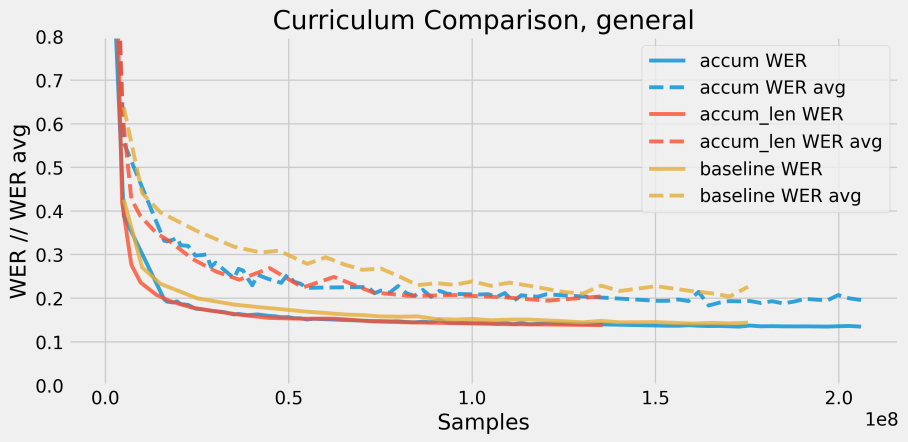

Figure 4: Differences in number of samples required to reach a given WER threshold between the accumulation-by-length curriculum, accumulation curriculum, and standard training, on the general validation dataset.

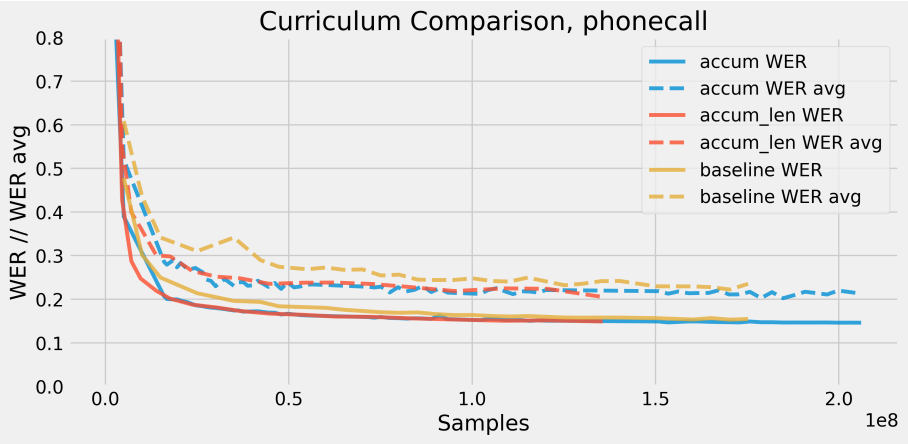

Figure 5: Differences in number of samples required to reach a given WER threshold between the accumulation-by-length curriculum, accumulation, and standard training, on the phonecall validation dataset.

Samples	Models	aws	finance	new_en	meeting
5.00e+6	baseline	0.32 / 0.33	0.30 / 0.31	0.29 / 0.29	0.39 / 0.46
	accum	0.33 / 0.36	0.22 / 0.23	0.40 / 0.40	0.37 / 0.44
	accum_len	0.35 / 0.39	0.25 / 0.26	0.44 / 0.44	0.40 / 0.47
1.00e+7	baseline	0.19 / 0.20	0.17 / 0.18	0.17 / 0.17	0.25 / 0.32
	accum	0.33 / 0.36	0.22 / 0.23	0.40 / 0.40	0.37 / 0.44
	accum_len	0.17 / 0.19	0.13 / 0.13	0.20 / 0.20	0.22 / 0.28
5.00e+7	baseline	0.12 / 0.12	0.11 / 0.12	0.11 / 0.11	0.16 / 0.23
	accum	0.11 / 0.11	0.09 / 0.09	0.11 / 0.11	0.14 / 0.21
	accum_len	0.11 / 0.11	0.08 / 0.09	0.11 / 0.11	0.14 / 0.20
75.00e+6	baseline	0.11 / 0.11	0.10 / 0.10	0.10 / 0.10	0.14 / 0.21
	accum	0.11 / 0.11	0.08 / 0.08	0.10 / 0.10	0.13 / 0.20
	accum_len	0.12 / 0.11	0.08 / 0.08	0.10 / 0.10	0.14 / 0.20
1.00e+8	baseline	0.11 / 0.11	0.09 / 0.10	0.09 / 0.10	0.14 / 0.21
	accum	0.10 / 0.10	0.08 / 0.08	0.10 / 0.10	0.13 / 0.20
	accum_len	0.12 / 0.11	0.08 / 0.08	0.10 / 0.10	0.13 / 0.20
1.20e+8	baseline	0.11 / 0.11	0.09 / 0.09	0.10 / 0.10	0.14 / 0.21
	accum	0.11 / 0.11	0.08 / 0.08	0.10 / 0.10	0.13 / 0.20
	accum_len	0.11 / 0.11	0.08 / 0.08	0.10 / 0.10	0.13 / 0.20

Samples	Models	problem	phonecall	gmc	general
5.00e+6	baseline	0.93 / 10.57	0.48 / 0.61	0.34 / 0.65	0.43 / 0.64
	accum	0.91 / 9.53	0.39 / 0.52	0.37 / 0.65	0.39 / 0.56
	accum_len	0.96 / 9.86	0.43 / 0.55	0.41 / 0.76	0.42 / 0.63
1.00e+7	baseline	0.76 / 9.67	0.30 / 0.43	0.21 / 0.42	0.27 / 0.44
	accum	0.91 / 9.53	0.39 / 0.52	0.37 / 0.65	0.39 / 0.56
	accum_len	0.73 / 9.53	0.25 / 0.36	0.21 / 0.40	0.24 / 0.39
5.00e+7	baseline	0.59 / 7.80	0.18 / 0.27	0.15 / 0.31	0.18 / 0.31
	accum	0.47 / 5.62	0.17 / 0.24	0.14 / 0.29	0.16 / 0.25
	accum_len	0.37 / 2.92	0.16 / 0.24	0.14 / 0.27	0.15 / 0.22
75.00e+6	baseline	0.52 / 5.73	0.17 / 0.25	0.14 / 0.29	0.16 / 0.27
	accum	0.41 / 4.17	0.16 / 0.23	0.13 / 0.26	0.15 / 0.22
	accum_len	0.36 / 2.85	0.16 / 0.23	0.13 / 0.28	0.15 / 0.21
1.00e+8	baseline	0.41 / 4.19	0.16 / 0.25	0.13 / 0.29	0.15 / 0.24
	accum	0.35 / 3.20	0.15 / 0.21	0.13 / 0.27	0.14 / 0.21
	accum_len	0.33 / 2.33	0.15 / 0.22	0.13 / 0.26	0.14 / 0.21
1.20e+8	baseline	0.40 / 3.98	0.16 / 0.25	0.13 / 0.28	0.15 / 0.23
	accum	0.37 / 3.54	0.15 / 0.22	0.13 / 0.25	0.14 / 0.21
	accum_len	0.31 / 2.33	0.15 / 0.22	0.13 / 0.27	0.14 / 0.19

Table 2: WER / WER average statistics across eight validation domains, baseline vs. accumulation vs. accumulation-by-length curricula.

4.2 Displacement

Given that accumulation can successfully improve a model's per-sample performance, the natural succeeding question becomes "why?" We can move towards answering this by pushing the curriculum to the extreme. The displacement curriculum proceeds as follows:

1. Order all samples in the training dataset from easiest (lowest scaled loss) to most difficult (highest scaled loss).
2. Place the samples into 10 adjacent, equally-sized bins. Bin 0 contains the $\frac{1}{10}$th easiest set of chunks, bin 1 contains the next $\frac{1}{10}$th easiest set of chunks, and so on.
3. Set the dataset equal to the next bin (beginning from bin 0) and iterate over once.
4. Continue until all bins have been added.

Because a bucket is replaced (displaced) after it has been iterated through, this curriculum amounts to a single *ordered* pass through the data. This is not without precedent: several other models in the literature, such as [1], use such an easy-to-hard easy displacement curriculum for the first epoch of training, followed by standard randomized training.

We observe from Figure 6 and Table 3 that displacement performs well initially, but quickly stagnates and falls behind the baseline, which continues improving in performance. We observe similar behavior across all domains except for `new_en`, where the baseline and displacement models perform the same. Figures for all domains are available here.

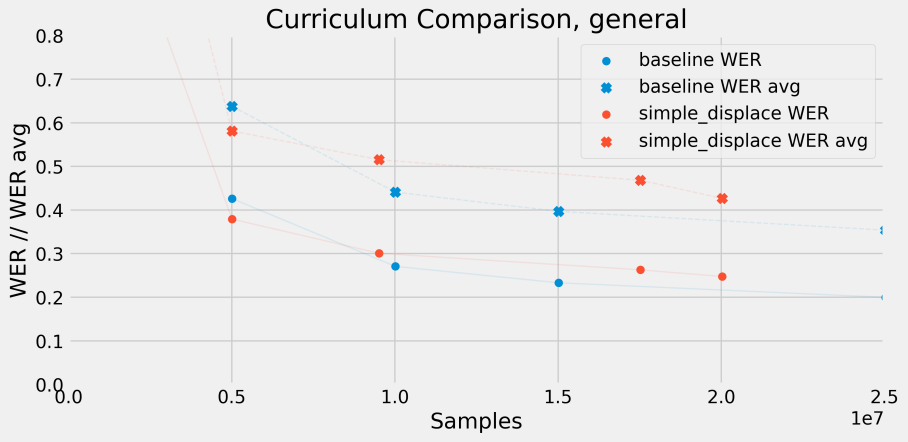

Figure 6: Validation performance on the general dataset of the same model trained with standard sample randomization (baseline, blue) and the displacement curriculum (red), up to 25m samples.

First of all, this demonstrates that merely ranking samples has little value in and of itself. This has important ramifications for how we understand the success of the accumulation algorithm, which has two constituencies: ranking (the data is ordered in bins from easiest to hardest) and iteration (by accumulating, the number of times a sample is iterated over is proportional to its easiness). These results suggest that iteration is doing most of the 'heavy lifting'.

Moreover: the only difference between the baseline and the displacement methods is that of sample order; therefore we can attempt to understand when ranking fails (while removing the influence of iteration). Early on, displacement is identical with accumulation, since both involve iteration over easier samples. Therefore, displacement initially performs better relative to the baseline. However, at 1e7 (10m) samples, the displacement model is being trained on the 4th bucket, or the 40th percentile of data by difficulty. This suggests that the previous buckets – constituting, say, the 10th to 20th

Samples	Models	aws	finance	new_en	meeting
5.00e+6	baseline	0.32 / 0.33	0.30 / 0.31	0.29 / 0.29	0.39 / 0.46
	simple_displace	0.30 / 0.32	0.22 / 0.23	0.33 / 0.33	0.36 / 0.43
1.00e+7	baseline	0.19 / 0.20	0.17 / 0.18	0.17 / 0.17	0.25 / 0.32
	simple_displace	0.20 / 0.19	0.21 / 0.22	0.17 / 0.17	0.26 / 0.33
2.00e+7	baseline	0.16 / 0.16	0.15 / 0.15	0.14 / 0.14	0.22 / 0.28
	simple_displace	0.16 / 0.15	0.17 / 0.18	0.13 / 0.13	0.21 / 0.28
Samples	Models	problem	phonecall	gmc	general
5.00e+6	baseline	0.93 / 10.57	0.48 / 0.61	0.34 / 0.65	0.43 / 0.64
	simple_displace	0.93 / 10.33	0.40 / 0.62	0.34 / 0.81	0.38 / 0.58
1.00e+7	baseline	0.76 / 9.67	0.30 / 0.43	0.21 / 0.42	0.27 / 0.44
	simple_displace	0.77 / 10.16	0.35 / 0.47	0.22 / 0.48	0.30 / 0.52
2.00e+7	baseline	0.72 / 9.64	0.25 / 0.34	0.18 / 0.39	0.23 / 0.40
	simple_displace	0.70 / 8.82	0.28 / 0.37	0.18 / 0.37	0.25 / 0.43

Table 3: WER / WER average statistics across eight validation domains, baseline vs. displacement.

percentile of easiest samples – provide the largest advantage to learning. Combined with the previously derived information, this suggests that iterating over the easiest set of samples provides the most value to learning. We will further support this hypothesis in later subsections.

4.3 Cyclic

We can 'bridge' the gap between accumulation and displacement with a cyclic curriculum:

1. Order all samples in the training dataset from easiest (lowest scaled loss) to most difficult (highest scaled loss).
2. Place the samples into 20 adjacent, equally-sized bins. Bin 0 contains the $\frac{1}{20}$th easiest set of chunks, bin 1 contains the next $\frac{1}{20}$th easiest set of chunks, and so on.
3. Establish a queue Q and a maximum capacity n. We use $n = 3$.
4. Add the next bin (beginning from bin 0) to Q.
5. If the size of Q exceeds n, decrease the queue size by one (FIFO-style).
6. Repeat 4-5 until all bins have passed through Q.

The cyclic curriculum uses both ranking and iteration: each bucket gets iterated over n times, in similar spirit to accumulation, but after this iteration buckets are dropped under the pretense that they can no longer provide as much additional benefit.

Figure 7 and Table 4 shows an interesting result: the model demonstrates improved performance over the baseline early on, but at about 15 million samples regresses and stagnates while the baseline continues improving. We observe similar behavior across all domains except for `new_en` and, to a lesser extent, `aws`, in which the cyclic model continues to decrease on the same trajectory as the baseline. You can view figures for all domains here.

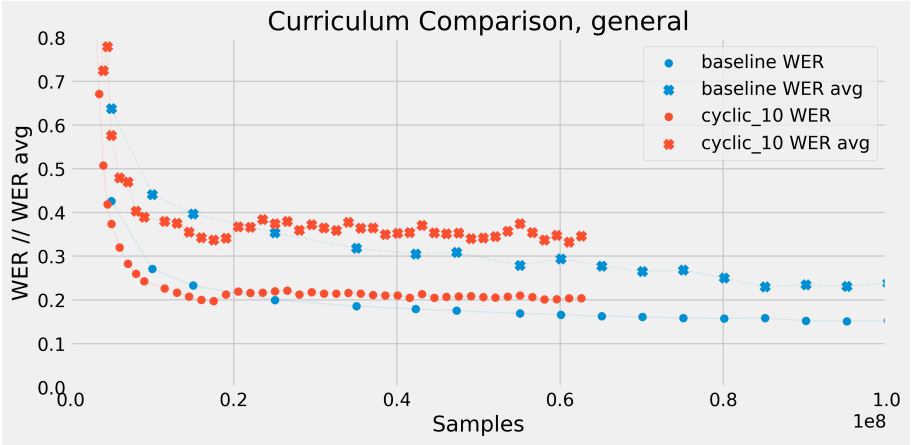

Figure 7: Validation performance on the general dataset of the same model trained with standard sample randomization (baseline, blue) and the cyclic curriculum (red), up to 25m samples.

Samples	Models	aws	finance	new_en	meeting
1.00e+7	baseline	0.19 / 0.20	0.17 / 0.18	0.17 / 0.17	0.25 / 0.32
	cyclic_10	0.17 / 0.18	0.14 / 0.15	0.18 / 0.18	0.22 / 0.29
2.00e+7	baseline	0.16 / 0.16	0.15 / 0.15	0.14 / 0.14	0.22 / 0.28
	cyclic_10	0.15 / 0.15	0.14 / 0.14	0.13 / 0.13	0.20 / 0.27
25.00e+6	baseline	0.14 / 0.13	0.13 / 0.13	0.12 / 0.13	0.18 / 0.25
	cyclic_10	0.15 / 0.14	0.14 / 0.15	0.12 / 0.13	0.20 / 0.26
5.00e+7	baseline	0.12 / 0.12	0.11 / 0.12	0.11 / 0.11	0.16 / 0.23
	cyclic_10	0.13 / 0.13	0.14 / 0.14	0.11 / 0.11	0.18 / 0.25
Samples	Models	problem	phonecall	gmc	general
1.00e+7	baseline	0.76 / 9.67	0.30 / 0.43	0.21 / 0.42	0.27 / 0.44
	cyclic_10	0.73 / 9.37	0.26 / 0.38	0.20 / 0.38	0.24 / 0.39
2.00e+7	baseline	0.72 / 9.64	0.25 / 0.34	0.18 / 0.39	0.23 / 0.40
	cyclic_10	0.72 / 9.98	0.23 / 0.32	0.17 / 0.33	0.22 / 0.37
25.00e+6	baseline	0.65 / 8.76	0.21 / 0.31	0.16 / 0.32	0.20 / 0.35
	cyclic_10	0.70 / 9.46	0.24 / 0.32	0.17 / 0.36	0.22 / 0.37
5.00e+7	baseline	0.59 / 7.80	0.18 / 0.27	0.15 / 0.31	0.18 / 0.31
	cyclic_10	0.59 / 7.53	0.22 / 0.30	0.15 / 0.31	0.21 / 0.34

Table 4: WER / WER average statistics across eight validation domains, baseline vs. cyclic.

We can visually observe that this pattern seems to be a lengthened version of the displacement performance trajectory which similarly began ahead of the baseline but soon stagnated. Let us track the number of samples covered at each cyclic update:

Update	Buckets in Cycle	Samples in Cycle	Running Total Samples
0	0	2.5m	2.5m
1	0, 1	5.0m	7.5m
2	0, 1, 2	7.5m	15m
3	1, 2, 3	7.5m	22.5m
4	2, 3, 4	7.5m	30.0m

We observe that the moment of regression – around 15m samples – occurs very similarly to the cycle-switching away from bucket 0 (Figure 8). Comparing the cyclic curriculum to the accumulation curriculum even more directly illustrates the dynamics at play here. The cyclic and the accumulation curriculum are identical until about 15m samples, in which the accumulation and cyclic curricula both update by acquiring the next bucket, but the cyclic curriculum drops the first bucket: $D_{\text{accum}} = \{0, 1, 2, 3\}$, $D_{\text{cyc}} = \{1, 2, 3\}$.

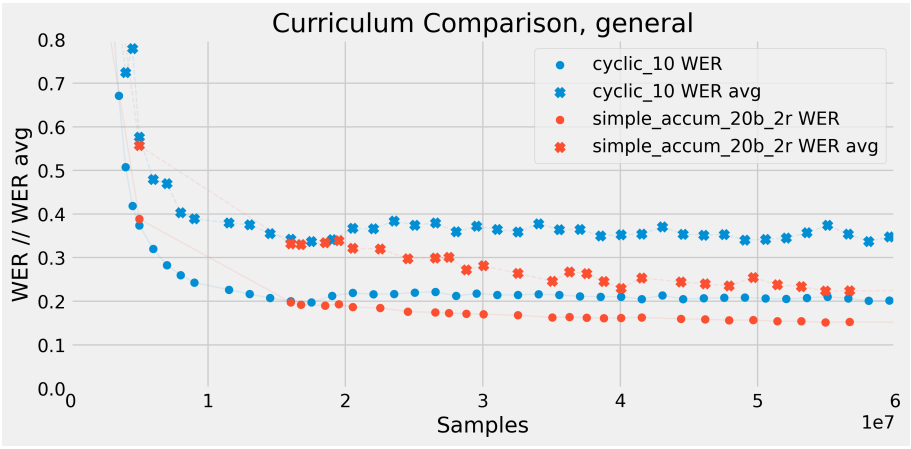

Figure 8: Validation performance on the general dataset of the same model trained with the accumulation curriculum (baseline, blue) and the cyclic curriculum (red), up to 25m samples.

Therefore, the only difference between the two curricula at this point is the presence/absence of the easiest $\frac{1}{20}$th of the data. Yet it appears that this dataset makes all the difference – a difference which widens as the cyclic algorithm abandons the 'easiest' bins to accept the more difficult ones. This seems to suggest that the easiest samples can be 'forgotten' if they are not reiterated over several times. Borrowing again from the theory of data pruning in [5], we need large-margin (easy) samples to repeatedly stabilize the generalization (i.e. anti-overfitting protection) of the model while it is learning to handle small-margin (difficult) samples.

You can view cyclic vs. accumulation comparisons for all domains here.

4.4 Straddle

We have confirmed from the previous accumulation, displacement, and straddle experiments that iterating over the easiest bucket of samples helps improve generalization speed by a significant factor. We propose another curriculum based on this information – the "straddle-walk". It proceeds as follows:

1. Order all samples in the training dataset from easiest (lowest scaled loss) to most difficult (highest scaled loss).
2. Place the samples into 20 adjacent, equally-sized bins. Bin 0 contains the $\frac{1}{20}$th easiest set of chunks, bin 1 contains the next $\frac{1}{20}$th easiest set of chunks, and so on.
3. Establish a constant set C of the nth easiest bins. We use $n = 2$.
4. Establish a dynamic set D with size m. We use $m = 1$.
5. Update set D with the next bucket (beginning from bucket 0).
6. Train the model on $C \cup D$.
7. Repeat 5-6 until all bins have been trained; set D back to bucket 0 and repeat.

The straddle-walk loops through the dataset from easiest to hardest while continuously having access on the easiest set of samples to ground generalization. We see statistically significant improvement over both the baseline and competitor accumulation curriculum (Figures 9, 10) by several WER points (Table 5).

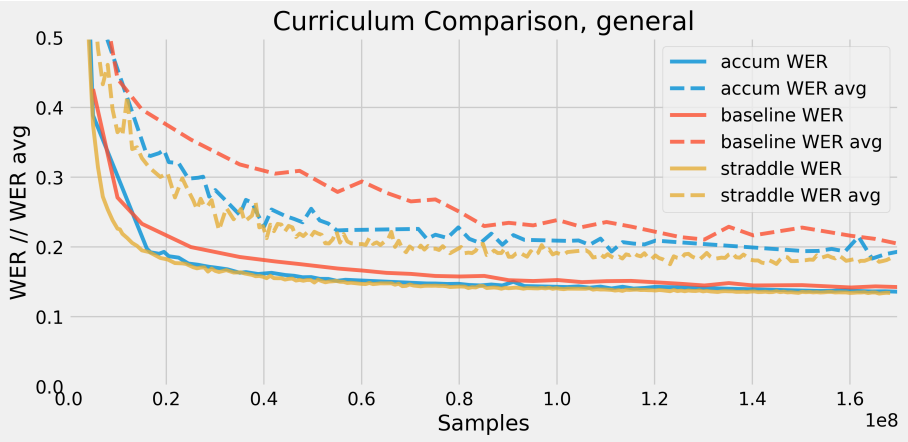

Figure 9: Validation performance on the general dataset of the same model trained with the straddle curriculum (yellow), the accumulation curriculum (blue), and the standard randomized-order baseline (red), up to 160m samples.

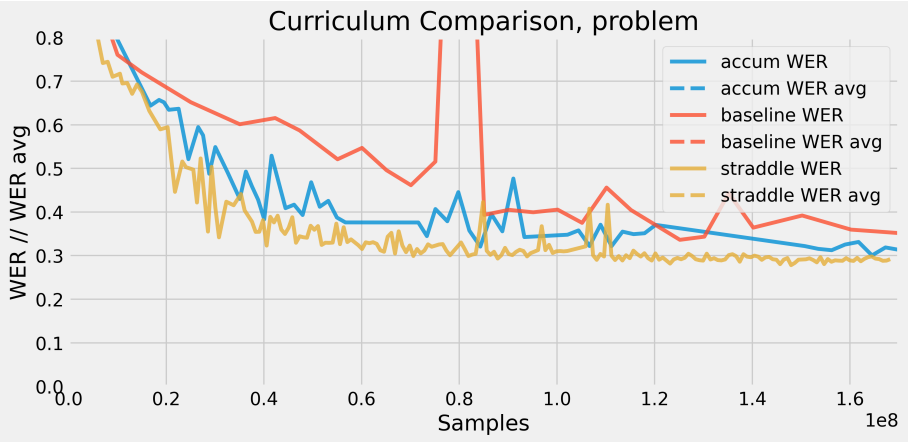

Figure 10: Validation performance on the problem dataset of the same model trained with the straddle curriculum (yellow), the accumulation curriculum (blue), and the standard randomized-order baseline (red), up to 160m samples.

Samples	Models	aws	finance	new_en	meeting
5.00e+6	baseline	0.32 / 0.33	0.30 / 0.31	0.29 / 0.29	0.39 / 0.46
	straddle	0.30 / 0.32	0.22 / 0.23	0.33 / 0.34	0.36 / 0.43
	accum	0.33 / 0.36	0.22 / 0.23	0.40 / 0.40	0.37 / 0.44
1.00e+7	baseline	0.19 / 0.20	0.17 / 0.18	0.17 / 0.17	0.25 / 0.32
	straddle	0.16 / 0.17	0.12 / 0.13	0.19 / 0.19	0.21 / 0.28
	accum	0.33 / 0.36	0.22 / 0.23	0.40 / 0.40	0.37 / 0.44
5.00e+7	baseline	0.12 / 0.12	0.11 / 0.12	0.11 / 0.11	0.16 / 0.23
	straddle	0.11 / 0.11	0.08 / 0.08	0.12 / 0.12	0.14 / 0.21
	accum	0.11 / 0.11	0.09 / 0.09	0.11 / 0.11	0.14 / 0.21
1.00e+8	baseline	0.11 / 0.11	0.09 / 0.10	0.09 / 0.10	0.14 / 0.21
	straddle	0.11 / 0.11	0.07 / 0.08	0.10 / 0.10	0.13 / 0.20
	accum	0.10 / 0.10	0.08 / 0.08	0.10 / 0.10	0.13 / 0.20
15.00e+7	baseline	0.11 / 0.10	0.09 / 0.09	0.09 / 0.09	0.13 / 0.20
	straddle	0.10 / 0.10	0.07 / 0.07	0.10 / 0.10	0.12 / 0.19
	accum	0.11 / 0.10	0.08 / 0.08	0.09 / 0.09	0.13 / 0.20

Samples	Models	problem	phonecall	gmc	general
5.00e+6	baseline	0.93 / 10.57	0.48 / 0.61	0.34 / 0.65	0.43 / 0.64
	straddle	0.89 / 10.25	0.39 / 0.59	0.34 / 0.71	0.37 / 0.57
	accum	0.91 / 9.53	0.39 / 0.52	0.37 / 0.65	0.39 / 0.56
1.00e+7	baseline	0.76 / 9.67	0.30 / 0.43	0.21 / 0.42	0.27 / 0.44
	straddle	0.71 / 9.67	0.24 / 0.30	0.20 / 0.36	0.23 / 0.36
	accum	0.91 / 9.53	0.39 / 0.52	0.37 / 0.65	0.39 / 0.56
5.00e+7	baseline	0.59 / 7.80	0.18 / 0.27	0.15 / 0.31	0.18 / 0.31
	straddle	0.36 / 3.14	0.16 / 0.22	0.13 / 0.25	0.15 / 0.22
	accum	0.47 / 5.62	0.17 / 0.24	0.14 / 0.29	0.16 / 0.25
1.00e+8	baseline	0.41 / 4.19	0.16 / 0.25	0.13 / 0.29	0.15 / 0.24
	straddle	0.31 / 2.18	0.15 / 0.21	0.13 / 0.26	0.14 / 0.19
	accum	0.35 / 3.20	0.15 / 0.21	0.13 / 0.27	0.14 / 0.21
15.00e+7	baseline	0.39 / 3.66	0.16 / 0.23	0.13 / 0.27	0.15 / 0.23
	straddle	0.29 / 1.96	0.15 / 0.19	0.12 / 0.24	0.14 / 0.18
	accum	0.32 / 2.56	0.15 / 0.22	0.12 / 0.27	0.14 / 0.19

Table 5: WER / WER average statistics across eight validation domains, baseline vs. straddle vs. accumulation curricula.

Figures 11 and 12 compare the speedup in number of samples required to reach a WER threshold from the baseline to the straddle curriculum for the general and meeting domains, respectively. We consistently see over 2x gains across domains.

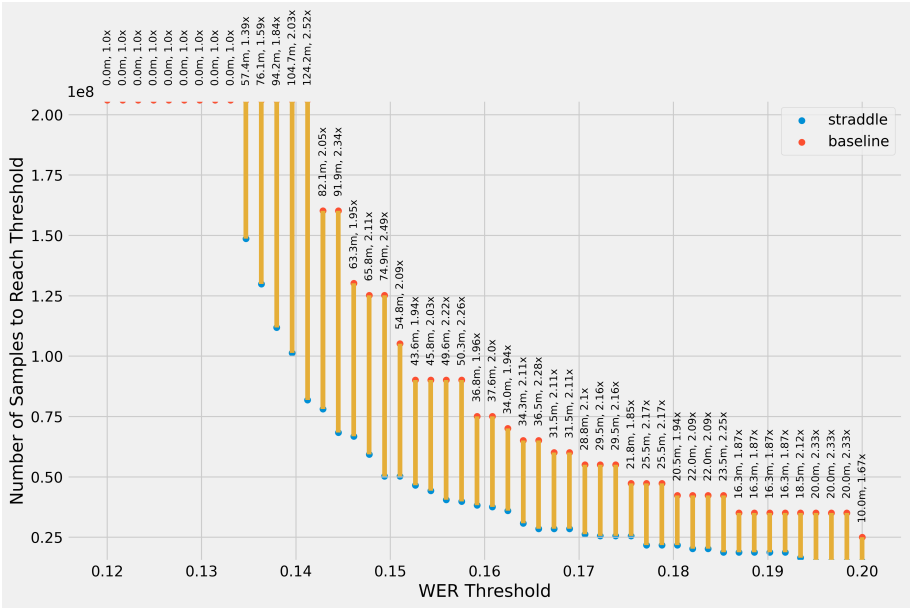

Figure 11: Differences in number of samples required to reach a given WER threshold between the accumulation curriculum and standard training, on the general validation dataset.

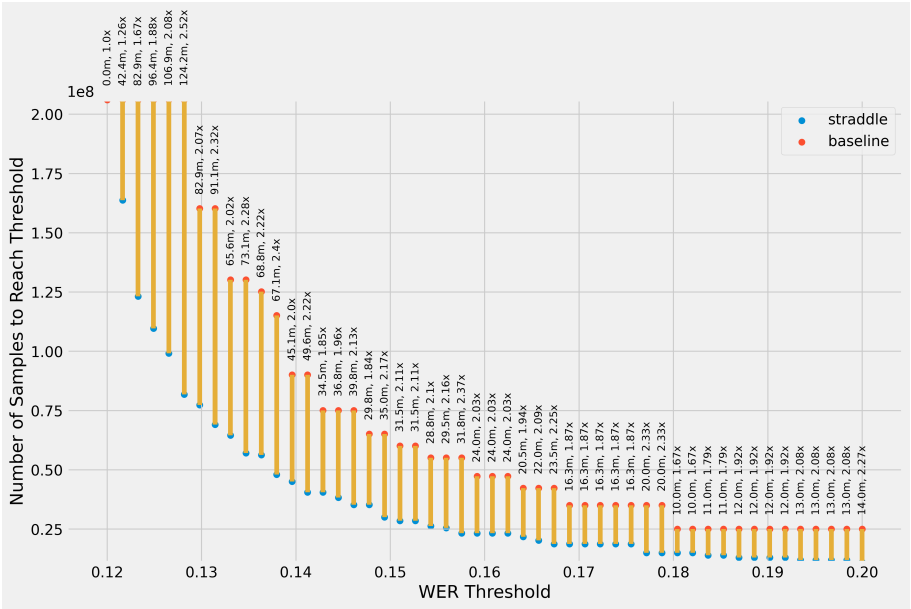

Figure 12: Differences in number of samples required to reach a given WER threshold between the accumulation curriculum and standard training, on the meeting validation dataset.

Table 6 and Table 7 provide quantiative information about samples-to-WER gains for WER and per-file WER averages, respectively. At best, using the straddle curriculum is 20% faster compared to accumulation and 57% faster compared to baseline for WER; and 45% faster compared to accumulation and 78% faster compared to baseline for per-file WER avg. That's almost a 5x speedup!

Model	Straddle			Accum			Baseline		
Measure	Samples	Left	Right	Samples	Left	Right	Samples	Left	Right
0.13	147.19m	0.85	NaN	172.77m	1.17	NaN	NaN	NaN	NaN
0.14	68.34m	0.81	0.53	84.31m	1.23	0.65	130.18m	1.9	1.54
0.15	44.31m	0.86	0.49	51.43m	1.16	0.57	90.12m	2.03	1.75
0.16	28.54m	0.81	0.44	35.05m	1.23	0.54	65.09m	2.28	1.86
0.17	25.54m	0.96	0.46	26.54m	1.04	0.48	55.07m	2.16	2.08
0.18	18.78m	0.83	0.44	22.53m	1.2	0.53	42.3m	2.25	1.88
0.19	15.02m	0.9	0.43	16.77m	1.12	0.48	35.05m	2.33	2.09
0.20	14.02m	0.88	0.56	16.02m	1.14	0.64	25.03m	1.79	1.56

Table 6: WER scores on the general validation dataset. *How to read*: "Samples" indicates the number of samples of training required before reaching the corresponding WER. "Left" indicates the just mentioned number of samples as a proportion of the left one of the other two curricula. For instance, the 'Left' column in 'accum' indicates the ratio between the number of samples for the accumulation curriculum to reach the corresponding WER to that of the straddle curriculum. NaN indicates that no data is available for one or multiple curriculum/a at the given WER level.

Model	straddle			accum			baseline		
Measure	Samples	Left	Right	Samples	Left	Right	Samples	Left	Right
0.18	107.39m	0.65	NaN	164.5m	1.53	NaN	NaN	NaN	NaN
0.19	69.09m	0.62	NaN	111.17m	1.61	NaN	NaN	NaN	NaN
0.20	53.32m	0.6	0.31	88.81m	1.67	0.52	170.23m	3.19	1.92
0.21	40.55m	0.55	0.32	73.36m	1.81	0.59	125.17m	3.09	1.71
0.22	30.79m	0.56	0.22	54.93m	1.78	0.39	140.19m	4.55	2.55
0.23	39.05m	0.97	0.46	40.06m	1.03	0.47	85.11m	2.18	2.12
0.24	30.04m	0.68	0.3	44.42m	1.48	0.44	100.13m	3.33	2.25
0.25	26.29m	0.75	0.33	35.05m	1.33	0.44	80.11m	3.05	2.29
0.26	27.79m	0.85	NaN	32.55m	1.17	NaN	NaN	NaN	NaN
0.27	21.78m	0.76	0.31	28.79m	1.32	0.41	70.09m	3.22	2.43
0.28	NaN	NaN	NaN	30.04m	NaN	0.55	55.07m	NaN	1.83

Table 7: Per-file WER average scores on the general validation dataset. Same rules as previous table for reading.

4.5 Extreme Data Pruning

One naturally arising question concerns the benefit of the straddle curriculum, which repeatedly reinforces the easiest set of samples while gradually integrating more difficult samples. We can train a model with 'extreme data pruning' – in this case, only training on the first bucket, or the easiest 5% of data. We find that this model performs better than or equal to the baseline (which is exposed to 100% of the data) until about 50m samples (Figure 13, Table 8). After this point, we can theorize that more difficult low-margin samples provide a 'push' in terms of generalization power past a boundary which cannot be surpassed with only easy high-margin samples. However, this result is still surprising: 5% of the data takes us very far already.

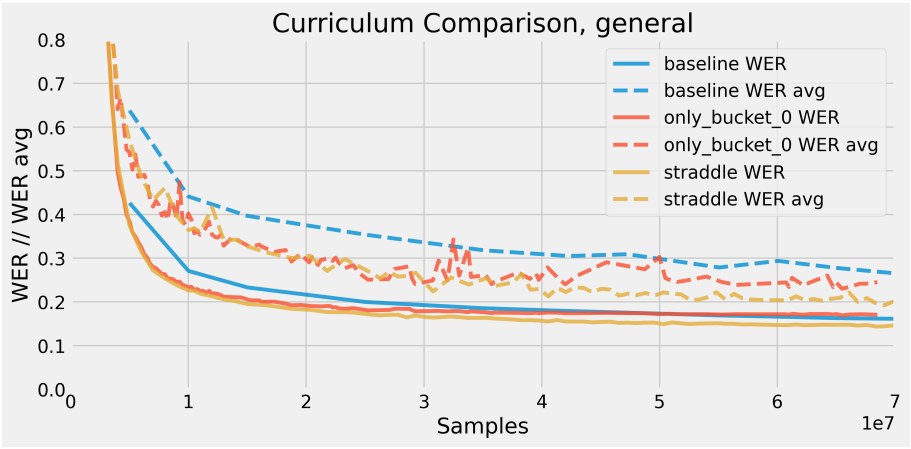

Figure 13: Validation performance on the general dataset of the same model trained with the straddle curriculum (yellow), training only with bucket 0 (easiest 5% of data), and the standard randomized-order baseline (blue), up to 70m samples.

Samples	Models	aws	finance	new_en	meeting
5.00e+7	baseline	0.12 / 0.12	0.11 / 0.12	0.11 / 0.11	0.16 / 0.23
	straddle	0.11 / 0.11	0.08 / 0.08	0.12 / 0.12	0.14 / 0.21
	only_bucket_0	0.12 / 0.13	0.09 / 0.09	0.19 / 0.18	0.16 / 0.23
1.00e+8	baseline	0.11 / 0.11	0.09 / 0.10	0.09 / 0.10	0.14 / 0.21
	straddle	0.11 / 0.11	0.07 / 0.08	0.10 / 0.10	0.13 / 0.20
	only_bucket_0	n/a	n/a	n/a	n/a
15.00e+7	baseline	0.11 / 0.10	0.09 / 0.09	0.09 / 0.09	0.13 / 0.20
	straddle	0.10 / 0.10	0.07 / 0.07	0.10 / 0.10	0.12 / 0.19
	only_bucket_0	n/a	n/a	n/a	n/a
Samples	Models	problem	phonecall	gmc	general
5.00e+7	baseline	0.59 / 7.80	0.18 / 0.27	0.15 / 0.31	0.18 / 0.31
	straddle	0.36 / 3.14	0.16 / 0.22	0.13 / 0.25	0.15 / 0.22
	only_bucket_0	0.39 / 3.13	0.18 / 0.24	0.16 / 0.29	0.17 / 0.30
1.00e+8	baseline	0.41 / 4.19	0.16 / 0.25	0.13 / 0.29	0.15 / 0.24
	straddle	0.31 / 2.18	0.15 / 0.21	0.13 / 0.26	0.14 / 0.19
	only_bucket_0	n/a	n/a	n/a	n/a
15.00e+7	baseline	0.39 / 3.66	0.16 / 0.23	0.13 / 0.27	0.15 / 0.23
	straddle	0.29 / 1.96	0.15 / 0.19	0.12 / 0.24	0.14 / 0.18
	only_bucket_0	n/a	n/a	n/a	n/a

Table 8: WER / WER average statistics across eight validation domains, baseline vs. straddle vs. only-bucket-0 curricula.

4.6 Leg 2 Training

These curricula can be shown to significantly improve the speed of 'leg 1' training. However, models also undergo additional training to learn capitalization, punctuation, and other more linguistically reflexive skills of transcription in 'leg 2' training. To evaluate this, we take baseline-, accumulation-, and straddle- trained checkpoints at 60m samples and at 160 samples (for six total checkpoints). Each of these checkpoints are subjected to 'leg 2' training. We observe from Figures 14 and 15 that the relationship

$$\text{straddle} < \text{accum} < \text{baseline}$$

holds approximately true, from 60m through 160m. It appears that learning occurs at approximately the same rate, but is shifted by different magnitudes based on the ending-point of previous training. The difference is almost 17 WER points at 60m samples and 12 WER points at 160m samples. However, validation performance even from 60m samples is generally stagnant, although the per-file WER average does experience somewhat significant gains, especially for difficult domains like the problem dataset (Figures 16, 17, 18). The difference is still present but less prominent for standard WER measures, and converge to similar values with negligible difference after just 20 epochs of training. From leg 2 training after 160m samples of leg 1 training, we observe similar behavior at a dampened scale, although the WER converges to an improved position. This suggests that leg 1 training plays a large role in improving the raw WER performance/generalization, and the objective of leg 2 should be oriented towards fine-tuning and learning grammatical rules (capitalization, punctuation, etc.) while 'holding onto' this generalization performance. Leg 1 training can be significantly sped up using curriculum learning, especially during the long-tail training regime.

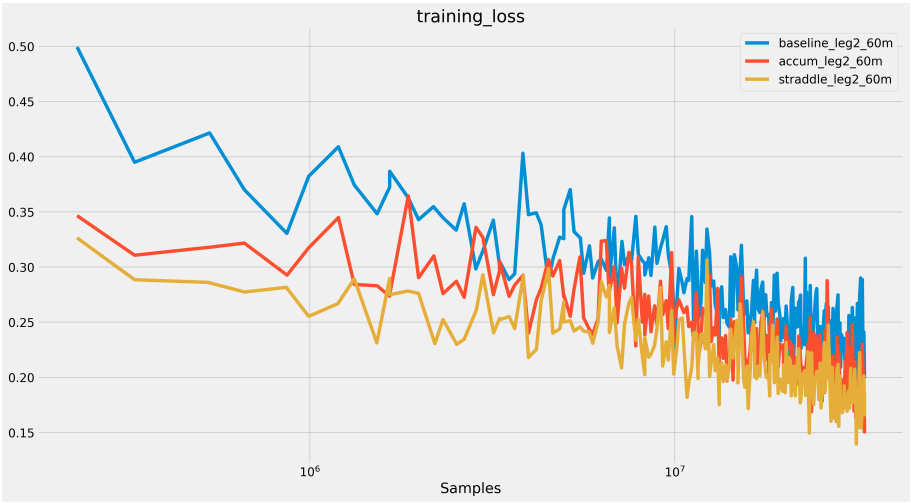

Figure 14: Training loss in leg 2 between models previously trained in leg 1 using baseline, accumulation, and straddle curricula (stopped after 60m samples).

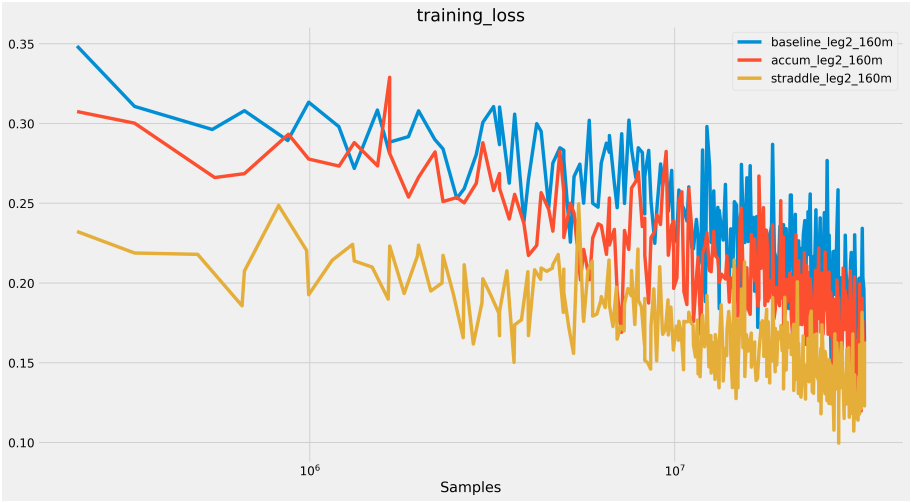

Figure 15: Training loss in leg 2 between models previously trained in leg 1 using baseline, accumulation, and straddle curricula (stopped after 160m samples).

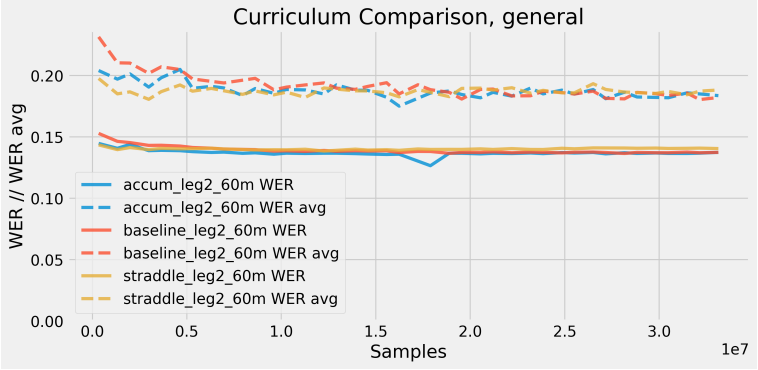

Figure 16: Validation WER for the general dataset in leg 2 from models trained with different curricula in leg 1 stopped at 60m samples of training.

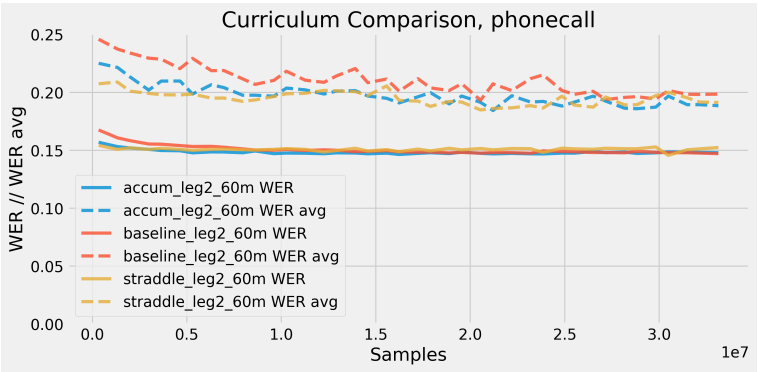

Figure 17: Validation WER for the phonecall dataset in leg 2 from models trained with different curricula in leg 1 stopped at 60m samples of training.

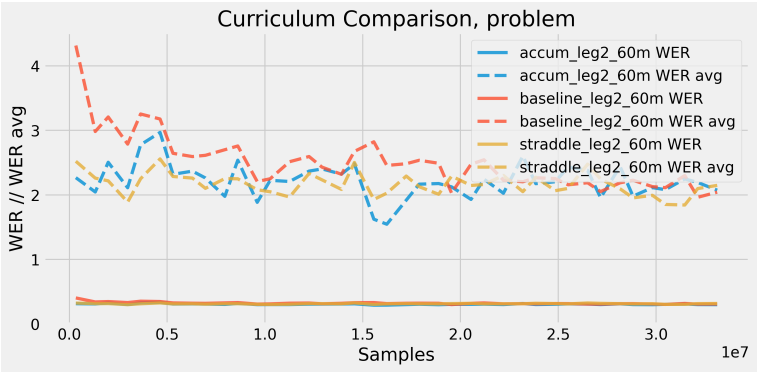

Figure 18: Validation WER for the problem dataset in leg 2 from models trained with different curricula in leg 1 stopped at 60m samples of training.

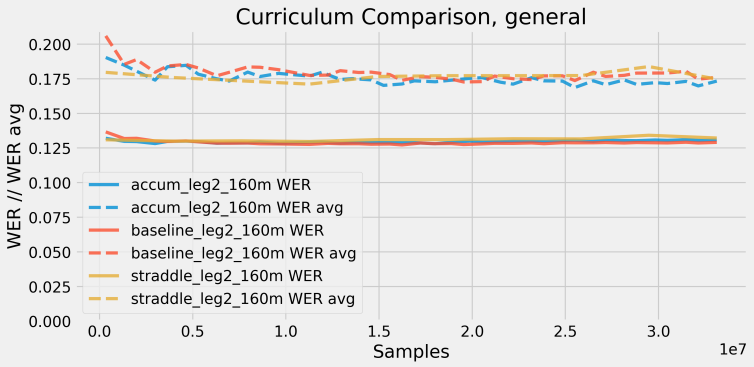

Figure 19: Validation WER for the general dataset in leg 2 from models trained with different curricula in leg 1 stopped at 160m samples of training.

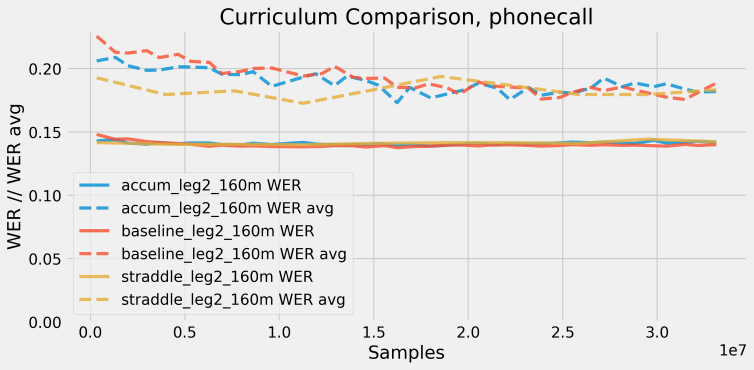

Figure 20: Validation WER for the phonecall dataset in leg 2 from models trained with different curricula in leg 1 stopped at 160m samples of training.

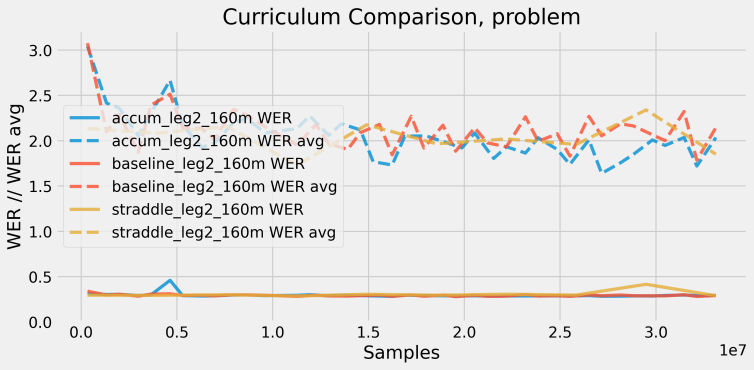

Figure 21: Validation WER for the problem dataset in leg 2 from models trained with different curricula in leg 1 stopped at 160m samples of training.

List of Figures

List of Tables

References

[1] Dario Amodei et al. "Deep Speech 2 : End-to-End Speech Recognition in English and Mandarin". In: *ICML*. 2016.

[2] Stefan Braun, Daniel Neil, and Shih-Chii Liu. "A curriculum learning method for improved noise robustness in automatic speech recognition". In: *2017 25th European Signal Processing Conference (EUSIPCO)* (2017), pp. 548–552.

[3] Guy Hacohen and Daphna Weinshall. "On The Power of Curriculum Learning in Training Deep Networks". In: *ArXiv* abs/1904.03626 (2019).

[4] Emmanouil Antonios Platanios et al. "Competence-based Curriculum Learning for Neural Machine Translation". In: *ArXiv* abs/1903.09848 (2019).

[5] Ben Sorscher et al. *Beyond neural scaling laws: beating power law scaling via data pruning*. 2022. DOI: 10.48550/ARXIV.2206.14486. URL: https://arxiv.org/abs/2206.14486.

See the Confluence pages under "Curriculum Learning" to implement curriculum learning in your own experiments.