Interspeech 2024
1-5 September 2024, Kos, Greece

Dynamic Data Pruning for Automatic Speech Recognition

Qiao Xiao[1,*], *Pingchuan Ma*[2,3,*], *Adriana Fernandez-Lopez*[2], *Boqian Wu*[4,5], *Lu Yin*[6], *Stavros Petridis*[2,3], *Mykola Pechenizkiy*[1], *Maja Pantic*[2,3], *Decebal Constantin Mocanu*[1,5], *Shiwei Liu*[7]

[1]Eindhoven University of Technology [2]Meta AI, UK [3]Imperial College London
[4]University of Twente [5]University of Luxembourg
[6]University of Surrey [7]University of Oxford

q.xiao@tue.nl,{pingchuanma,afernandezlopez}@meta.com

Abstract

The recent success of Automatic Speech Recognition (ASR) is largely attributed to the ever-growing amount of training data. However, this trend has made model training prohibitively costly and imposed computational demands. While data pruning has been proposed to mitigate this issue by identifying a small subset of relevant data, its application in ASR has been barely explored, and existing works often entail significant overhead to achieve meaningful results. To fill this gap, this paper presents the first investigation of dynamic data pruning for ASR, finding that we can reach the full-data performance by dynamically selecting 70% of data. Furthermore, we introduce Dynamic Data Pruning for ASR (DDP-ASR), which offers several fine-grained pruning granularities specifically tailored for speech-related datasets, going beyond the conventional pruning of entire time sequences. Our intensive experiments show that DDP-ASR can save up to 1.6× training time with negligible performance loss.

Index Terms: automatic speech recognition, data pruning, learning efficiency

1. Introduction

In the speech domain, the increasingly larger training datasets have significantly contributed to remarkable performance gains [1–3]. However, it also poses substantial challenges to training with limited computational resources. Some prior works have revealed that not all training instances are equally important for model training [4–6]. This has led to inspiring efforts to improve the training efficiency of neural networks by either eliminating redundant data or prioritizing training instances based on their informational complexity [7–9]. Many recent works have also proposed diverse data pruning approaches to enhance training efficiency across various domains, such as computer vision [10–13] and natural language processing [14–16].

Despite the potential benefits of data pruning, it has received limited attention in the domain of Automatic Speech Recognition (ASR). In a recent study, Boris et al. [17] introduced a pruning approach to first group similar instances together through clustering to minimize the dataset size while maintaining its representative characteristics. This approach explores similarities in the multidimensional feature space of a pre-trained large audio model. However, it requires multiple trials to derive more precise representations before data pruning, leading to additional overhead costs. To address this issue, for the first time, we introduce *Dynamic Data Pruning (DDP)* [11, 18, 19] in the context of ASR. DDP is a recently emerged data-pruning technique where only a subset of data is sampled and fed into the model throughout the training process.

Specifically, we begin by conducting a comprehensive investigation into DDP for ASR pre-training, using various pruning criteria. Our findings reveal an encouraging discovery: through the adoption of a curriculum learning strategy [20], we are able to train an ASR model using only 70% of the data while achieving performance on par with that of the full-data training approach. To further enhance the efficacy of data pruning in ASR, we delve into a series of finely-tuned granularities meticulously crafted for speech-related data pruning. These granularities encompass the removal of individual time points as well as segments of temporal chunks. Our results demonstrate that by selectively removing consecutive samples, we can further improve the data efficiency of ASR. These empirical investigations culminate in the development of a novel data pruning approach for ASR, which we term Dynamic Data Pruning for ASR (DDP-ASR).

DDP-ASR incorporates both instance-wise and fine-grained time-wise granularities, allowing for the removal of a significant portion of data while achieving substantial practical speedup. Our extensive experiments on Librispeech demonstrate that, with a mixture of rule-of-thumb pruning rates, DDP-ASR can deliver up to 1.6× training speedups, while maintaining comparable performance to that achieved with full data. Additionally, we investigate the model's temporal robustness when trained on pruned subsets, revealing that our approach also brings benefits of robustness to audio clips with low sampling rates. To the best of our knowledge, our work is the first attempt to explore dynamic data pruning for ASR with novel pruning granularities specifically tailored for speech-related data, presenting new opportunities for enhancing the training efficiency of speech-related models.

2. Methodology

2.1. Dynamic data pruning

Given a dataset $\mathcal{D} = \{z_i = (x_i, y_i)\}|_{i=1}^{|\mathcal{D}|}$, a score $\mathcal{H}(z)$ is assigned to each instance z. In each pruning cycle, instances are selectively removed according to the distribution of scores $\mathcal{H}$ and the pruning criterion $\mathcal{S}$. The reserved subset after data pruning is defined as:

$$\mathcal{X}^{kept} = \mathcal{S}\left(\mathcal{H}, \mathcal{X}, k\right) \tag{1}$$

where k is the kept ratio and $\mathcal{X}$ corresponds to all input instances. For *static data pruning* [6, 10], instances that satisfy a specific condition are permanently discarded before the training begins and will never be activated again.

In contrast, dynamic data pruning enables the score $\mathcal{H}_t$ to be dynamically updated throughout training, ensuring that

[*]Equal contribution.

10.21437/Interspeech.2024-1330

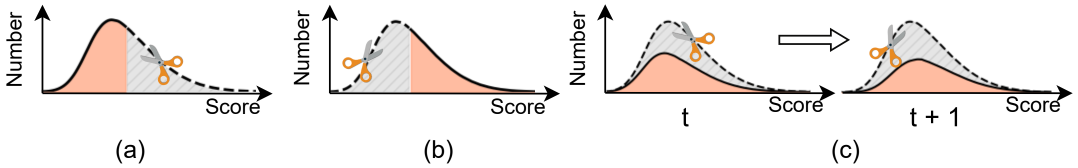


Figure 1: *Instance-wise pruning approaches: (a) Easy: Instances with the lowest scores are selected. (b) Hard: Instances with the highest scores are selected. (c) Easy2hard: Following the essence of curriculum learning, models initially train on relatively easy instances and progressively shift focus to more challenging ones as the training progresses.*

the coreset data can be more effectively adjusted based on the model's status at every t epoch. In this scenario, there is no reliance on pre-trained models, and intricate trials or runs are needed to acquire the pruning score before training [11, 18, 19].

2.2. Dynamic data pruning for ASR

To facilitate training acceleration through dynamic data pruning for speech data, we introduce a novel data pruning method, referred to as DDP-ASR (Dynamic Data Pruning for ASR). DDP-ASR extends beyond the conventional instance-wise data pruning by incorporating fine-grained time-wise pruning strategies within each time sequence, thereby achieving a practical speedup.

2.2.1. *Instance-wise pruning*

Instance-wise pruning aims to remove entire audio sequences based on a given score $\mathcal{H}_t$. Several methods have been proposed to calculate the score of each instance, such as the loss values [11] and uncertainty [19]. The calculated score $\mathcal{H}_t$ is then used to determine which instances to preserve based on pruning criteria. For instance, the score distribution $\mathcal{H}_t$ can be used to identify and retain either *easy* or *hard* instances, where *easy* instances are those with lower scores and *hard* instances are those with higher scores. In this study, we select the loss values $\mathcal{L}$ of each instance z as the corresponding score, as these values can be obtained without extra cost during training and reflect the learning status of the instances. Moreover, its effectiveness has been verified in [11, 21]. Therefore, we explore a variety of instance-wise pruning methods tailored for ASR training:

Easy. We prioritize the training of models on instances classified as "easy", which are identified by their lower scores, opting to exclude those with the highest scores. Figure 1 (a) illustrates an example of this method.

Hard. Conversely, we focus on incorporating "hard" instances for training, effectively sidelining those instances that are scored lower based on score distribution $\mathcal{H}_t$. Figure 1 (b) provides an example for this method.

Easy2hard. Inspired by curriculum learning strategies [22, 23], which train their models by progressively showing harder examples, we propose a novel selection strategy that dynamically schedules the presentation of instances to the model during training. Thus, at every checkpoint, $(1 - \epsilon)\, k$ points with progressively increasing difficulty are kept, and $\epsilon\, k$ points are randomly selected from the remaining dataset. Here ϵ is used to strategically schedule the presentation of easy or hard instances to the model. It is worth noting that ϵ starts at 1 and gradually linearly decreases during training, effectively altering the selection strategy over time, as shown in Figure 1 (c).

2.2.2. *Time-wise dropping*

Point-wise dropping. Inspired by CLIP [24], which removes a portion of image patches to yield a training speedup, we introduce *point dropping*, a simple time-wise dropping strategy that removes individual data points within an instance to improve training speed. This is done by randomly retaining L audio samples within an instance that contains T audio samples.

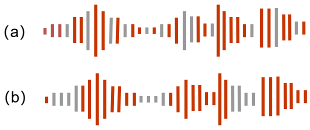


Figure 2: *A toy example comparing different time-wise pruning approaches: (a) Point Dropping, (b) Chunk Dropping, where signals highlighted in gray are pruned during training.*

Chunk-wise dropping. Compared to point dropping, *chunk dropping* specifically targets the removal of chunks, each consisting of n consecutive audio samples. For any given chunk $[t, t+n]$, t is chosen from the range $[0, T-n)$, where T is the input length of the instance. With a remaining length denoted as L, the method eliminates $(T-L)/n$ chunks.

3. Experimental Setup

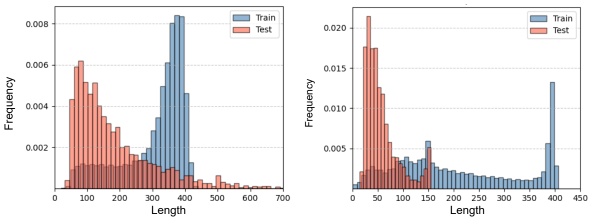
Dataset. In this study, we conduct experiments on two datasets: Librispeech [25], an audio corpus collected from audiobooks, and LRS3 [26], an audio-visual corpus from TED and TEDx talks. For Librispeech, we use "train-clean-100", "train-clean-360", and "train-other-500" subsets, totalling 960 hours of training data and evaluate our performance on the "test-clean" set with a total of 5.1 hours of audio. LRS3 consists of 439 hours of video clips, with 118 516 (408 hours), 31 982 (30 hours) and 1 321 clips (0.9 hours) in the pre-training, training-validation, and test sets, respectively.

Pre-processing. Following [27], we take raw audio waveforms as input to the model and perform only z-normalisation per utterance before feeding it into the model.

Data augmentation. We only apply adaptive time masking [28] to the raw audio stream. In particular, we choose a number of masks that is proportional to the utterance length and a maximum masking length of up to 0.4 seconds.

Model architecture. Rather than pursuing state-of-the-art performance, our primary goal is to investigate data pruning techniques in the domain of ASR. Accordingly, we adapt the open-source conformer-based architecture from [29]. Our models comprises a 1D ResNet front-end (3.9 M parameters) to extract speech features from raw audio waveforms, followed by a conformer encoder (170.9 M parameters), a Transformer decoder (64.5 M parameters) and a projection CTC layer (3.9 M parameters), resulting in a total of 243.1 M parameters.

Training details. Following standard practices in ASR, we train using a combination of CTC loss and Cross-Entropy loss. The model is trained for 75 epochs using the AdamW optimiser [30]. A cosine learning rate scheduler and a warm-up of 5 epochs are used, with the peak learning rate set to 0.001. We limit the duration of each training clip to no more than 16 seconds, and the maximum number of duration per batch is 64 seconds. All the models are trained with 32 A100 GPUs. For



(a) LibriSpeech (b) LRS3

Figure 3: *The distribution of length on Librispeech and LRS3.*

data pruning, we update the remaining subset every epoch. In the `Easy2hard` method, the proportion between the selected subset based on scores and a random one is set to 2:1 at the end of the training, which means that ϵ will linearly decay to 1/3. For time-wise dropping, we randomly drop samples up to the given dropping rate.

4. Experimental Results

4.1. Performance for instance-wise pruning

We evaluate the effectiveness of different instance-wise pruning methods for the Librispeech and LRS3 datasets. For a broader comparison, we include two additional pruning methods: (i) models trained with subsets randomly selected from the entire dataset in each pruning cycle, termed as `Random`, and (ii) models trained on a fixed subset initially chosen at random from the full dataset, referred to as `Static`.

Results of using different instance-wise pruning methods on the 'test-clean" set of Librispeech are shown in Table 1. We observe that for most pruning methods (namely, `Static`, `Easy` and `Random`, respectively), the performance is substantially impacted when more training instances are pruned. For example, when using 50 % easy training data (namely, `Easy`), a substantial increase of 1.7 % in Word Error Rate (WER) is observed. More hard instances likely tend to be removed, resulting in a relatively poor generalisation on the long sentences (More details analysis can be found in section 4.6). The issue can be partly mitigated by training with the `Random` method, which avoids bias in the remaining instances. As a result, it narrows the performance gap to a mere 0.3 % in WER at a kept ratio of 50 %. A further closer performance gap to full data can be observed when using hard-related methods (namely, `Hard` and `Easy2hard`, respectively), which force the model to focus more on hard instances. Additionally, it is worth noting that using 70 % of the hard training instances can yield performance comparable to using the entire dataset, indicating considerable redundancy in LibriSpeech.

Results of using different instance-wise pruning methods on the test set of LRS3 are presented in Table 2. A similar trend as in the Librispeech experiments can be observed. The only exception is the results after using `Hard`, which consistently perform worse than the `Random` method. This might be due to a large discrepancy in the distribution of length between the training and test sets (as shown in Figure 3). Specifically, concentrating on a subset of hard instances in the training set, which may not align well with the test set, can result in diminished test set performance. This is not the case for Librispeech, where length discrepancies are less noticeable.

4.2. Performance for time-wise dropping

Table 3 studies the impact of two time-wise dropping strategies (namely, `point` and `chunk`, respectively) by varying the dropping ratio on our proposed `Easy2hard` method. We observe that overall the performance gap between the model trained after time dropping and the baseline model without time dropping becomes increasingly larger as the dropping ratio increases. The performance degradation may be partially due to corrupted temporal dependencies, where the model relies on the precise order of input data for accurate predictions. In particular, for point dropping, where at 30% of the dropping ratio, it results in a 0.25% increase of WER. Notably, chunk dropping, which involves removing consecutive samples in each chunk, can mitigate some of the performance declines. When applying a 30% time-wise dropping ratio to the audio samples using this method, the impact on performance is minimal, with only a 0.06% WER increase. Given that chunk dropping performs better, we use chunk dropping as our default setting for the time-wise dropping.

Table 1: *WER [%] (↓) of our models with different pruning methods as a function of the kept ratio on the test set of Librispeech. The best results are bold for each kept ratio.*

Method/Kept ratio [%]	100	90	70	50	30
Static	2.58	2.69	2.86	3.52	4.59
Easy	2.58	2.91	3.65	4.28	6.90
Random	2.58	2.59	2.66	2.88	3.17
Hard	2.58	2.63	2.57	2.76	3.13
Easy2hard	2.58	**2.56**	**2.53**	**2.72**	**3.09**

Table 2: *WER [%] (↓) results of our ASR model with different pruning methods as a function of the kept ratio on the "test-clean" set of LRS3.*

Method/Kept ratio [%]	100	90	70	50	30
Static	2.10	2.18	2.65	3.45	4.71
Easy	2.10	2.27	2.27	5.03	15.17
Random	2.10	2.12	2.12	2.29	2.60
Hard	2.10	2.16	2.67	2.80	2.90
Easy2hard	2.10	**2.08**	**1.95**	**2.17**	**2.54**

Table 3: *WER [%] (↓) results of our models with different time-wise dropping methods as a function of the kept ratio on the "test-clean" set of LibriSpeech. The best results are bold for each time-wise kept ratio. "k" denotes the kept ratio.*

Method	Instance-wise			Time-wise k [%]		
	k [%]	100	90	70	50	30
Point	70	2.53	2.65	2.78	2.80	3.05
Chunk	70	2.53	**2.59**	**2.59**	**2.66**	**2.79**

4.3. Instance-wise pruning or time-wise dropping?

We investigate in Table 4 the optimal strategy of combining both pruning methods under the same wall-clock training time. The results indicate that for models trained with a larger portion of data (more than 70%), including time dropping results in a slight decrease in performance. Interestingly, we show that when the sampling rate is down-sampled from 16 000 Hz to 11 025 Hz, a closer performance gap for the model with time dropping can be observed compared with its counterpart, which indicates that a more robustness temporal dependency is learnt when time-wise dropping is applied. On the other hand, within the same training duration, combining instance- and time-wise

Table 4: *Impact of different sampling rates on the performance of Librispeech. "k" denotes the kept ratio.*

Instance k [%]	Time k [%]	Wall-clock time per epoch [min]	WER 16KHz [%]	WER 11KHz [%]
100	100	13.2±0.2	2.58	9.77
70	100	9.8±0.2	2.53	10.56
80	80	9.8±0.1	$2.56_{\uparrow 0.03}$	$8.94_{\downarrow 1.62}$
90	60	9.7±0.2	$2.61_{\uparrow 0.08}$	$7.48_{\downarrow 3.08}$
30	100	4.2±0.1	3.09	14.29
40	50	4.2±0.1	$3.04_{\downarrow 0.05}$	$9.52_{\downarrow 4.77}$
50	25	4.4±0.2	$2.99_{\downarrow 0.10}$	$9.42_{\downarrow 4.87}$

data pruning for a smaller portion of training data leads to an observable reduction in WER, compared to models trained solely with instance-wise data pruning at a standard sampling rate (16 000 Hz). This suggests that the synergistic application of both pruning strategies is beneficial in scenarios with a very limited computational resource. In general, it is observed that models trained using time-wise dropping exhibit greater robustness across different sampling rates, especially at a low sampling rate.

Table 5: *Time masking and dropping. Instances are kept to 70 % of the whole dataset for all cases. We mask up to 40 % audio samples in chunks and drop up to 30% of the audio samples.*

Time mask	Time drop	Wall-clock time per epoch [min]	WER [%]
✗	✗	9.7±0.2	3.11
✓	✗	9.8±0.2	**2.53**
✗	✓	**8.1±0.1**	2.78
✓	✓	**8.1±0.1**	2.59

4.4. Time masking versus time dropping

Time dropping is implemented by eliminating data points from the training instances, unlike time masking, which sets consecutive samples to zero without altering the speed, as discussed in [31]. Table 5 shows the impact of the use of time masking and time dropping on the "test-clean" set of Librispeech. In particular, the use of time masking results in a 0.58 % reduction in WER. However, substituting time masking with time dropping leads to a 0.25 % increase in WER, alongside a significant reduction of 17 % times. Interestingly, by integrating both time-masking and time-dropping approaches, it is possible to mitigate the performance decrease, achieving enhanced efficiency and comparable performance to the original model.

4.5. Data scaling and speedup

We expanded the training dataset from 960 hours to 3,494 hours by incorporating additional datasets such as LRS3, VoxCeleb2, and AVSpeech. The outcomes on the Librispeech dataset, displayed in Table 6, indicate a marginal improvement in performance using our proposed data pruning method compared to the random pruning method. This demonstrates the effectiveness of our approach when applied to larger training datasets.

Table 6: *Impact of the size of additional training data on the "test-clean" set of Librispeech. The additional data includes LRS3, VoxCeleb2, and AVSpeech, totaling 3 494 hours.*

Training data	Instance k [%]	Time k [%]	Wall-clock time per epoch [min]	WER [%]
Random	50	100	25.5±0.5	2.25
Easy2hard	50	100	26.8±1.0	**2.18**

Table 7: *Impact of the number of training epochs on the Librispeech dataset. "k" denotes the kept ratio.*

Instance-wise k [%]	Time-wise k [%]	Training epochs	Wall-clock time per epoch [min]	WER [%]
100	100	75	13.2±0.2	2.58
		$56_{\downarrow 25\,\%}$		2.59
70	100	75	$9.8\pm0.2_{\downarrow 25\,\%}$	**2.53**
70	70	75	$8.1\pm0.1_{\downarrow 38\,\%}$	2.59

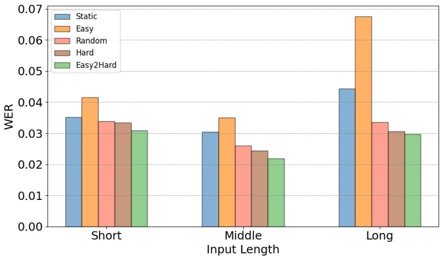


Figure 4: *Comparing instance-wise pruning strategies across three subsets of the Librispeech "test-clean" set, with instance-wise kept ratio of 50% for each method.*

Table 7 presents the performance comparison of our method with full data training under the same training time. In particular, when using an instance-wise kept ratio of 70% with the `Easy2hard` method, which takes a similar training time to the model using full data trained for 56 epochs, we observe a further decrease in WER by 0.06% compared to the model trained with the entire dataset for 56 epochs. Moreover, when combined with time-wise dropping, the training speed improves by 38%, resulting in a comparable WER to that achieved with 75 epochs of full data training.

4.6. Error analysis

To assess how the presented models affect performance across instances of varying input lengths. We divide the test samples in the "test-clean" set of Librispeech into three groups with different input duration, namely, *Short* (0 - 8 seconds), *Middle* (8 - 16 seconds) and *Long* ($\geq$ 16seconds), respectively. The performance of each group for the `Easy`, `Random`, `Hard` and `Easy2hard` methods is presented in Figure 4. Interestingly, we observe that models prioritizing easy instances tend to underperform, especially on longer instances, whereas models that focus on challenging instances show better performance on shorter ones. Overall, the proposed `Easy2hard` approach consistently outshines the other methods across all groups.

5. Conclusion

In this work, we conduct detailed analysis of dynamic data pruning for ASR, focusing on both instance-wise and time-wise pruning techniques. We demonstrate that these methods can be synergistically employed to maintain performance while achieving significant speed improvements. Among pruning methods, our proposed `Easy2hard` method has been found to be the most effective in speech recognition benchmarks. Notably, we observe that pruning up to 30% of instances, coupled with a 30% chunk dropping rate, can maintain performance compared to training with the full dataset. Moreover, our findings reveal that time-wise pruning significantly boosts model resilience to lower sampling rates, making it a valuable adjunct to time masking.

6. Acknowledgements

This research is part of the research program 'MegaMind - Measuring, Gathering, Mining and Integrating Data for Self-management in the Edge of the Electricity System', (partly) financed by the Dutch Research Council (NWO) through the Perspectief program under number P19-25. Additionally, note that only non-Meta authors utilized and processed the datasets (and no dataset pre-processing or processing took place on Meta's servers or facilities). Shiwei Liu is supported by the Royal Society with the Newton International Fellowship.

7. References

[1] J. Kaplan, S. McCandlish, *et al.*, "Scaling laws for neural language models," *CoRR*, vol. abs/2001.08361, 2020.

[2] H. Touvron, M. Cord, *et al.*, "Training data-efficient image transformers & distillation through attention," in *ICML*, 2021, pp. 10 347–10 357.

[3] R. Bommasani, D. A. Hudson, *et al.*, "On the opportunities and risks of foundation models," *CoRR*, vol. abs/2108.07258, 2021.

[4] A. Katharopoulos and F. Fleuret, "Not all samples are created equal: Deep learning with importance sampling," in *ICML*, 2018, pp. 2525–2534.

[5] M. Toneva, A. Sordoni, *et al.*, "An empirical study of example forgetting during deep neural network learning," in *ICLR*, 2018.

[6] C Coleman, C Yeh, *et al.*, "Selection via proxy: Efficient data selection for deep learning," in *ICLR*, 2020.

[7] R. J. N. Baldock, H. Maennel, and B. Neyshabur, "Deep learning through the lens of example difficulty," in *NIPS*, 2021, pp. 10 876–10 889.

[8] B. Mirzasoleiman, K. Cao, and J. Leskovec, "Coresets for robust training of deep neural networks against noisy labels," in *NIPS*, 2020.

[9] S. Mindermann, J. M. Brauner, *et al.*, "Prioritized training on points that are learnable, worth learning, and not yet learnt," in *ICML*, 2022, pp. 15 630–15 649.

[10] M. Paul, S. Ganguli, and G. K. Dziugaite, "Deep learning on a data diet: Finding important examples early in training," in *NIPS*, vol. 34, 2021, pp. 20 596–20 607.

[11] Z. Qin, K. Wang, *et al.*, "Infobatch: Lossless training speed up by unbiased dynamic data pruning," in *ICLR*, 2024.

[12] A. Abbas, K. Tirumala, D. Simig, S. Ganguli, and A. S. Morcos, "Semdedup: Data-efficient learning at web-scale through semantic deduplication," vol. abs/2303.09540, 2023.

[13] B. Mirzasoleiman, J. Bilmes, and J. Leskovec, "Coresets for data-efficient training of machine learning models," in *ICML*, 2020, pp. 6950–6960.

[14] M. Xia, S. Malladi, S. Gururangan, S. Arora, and D. Chen, "LESS: selecting influential data for targeted instruction tuning," *CoRR*, vol. abs/2402.04333, 2024.

[15] M. Marion, A. Üstün, *et al.*, "When less is more: Investigating data pruning for pretraining llms at scale," *CoRR*, vol. abs/2309.04564, 2023.

[16] S. Gunasekar, Y. Zhang, *et al.*, "Textbooks are all you need," *CoRR*, vol. abs/2306.11644, 2023.

[17] B. Bergsma, M. Brzezinska, O. V. Yazyev, and M. Cernak, "Cluster-based pruning techniques for audio data," *arXiv preprint arXiv:2309.11922*, 2023.

[18] R. S. Raju, K. Daruwalla, and M. H. Lipasti, "Accelerating deep learning with dynamic data pruning," *CoRR*, vol. abs/2111.12621, 2021.

[19] M. He, S. Yang, T. Huang, and B. Zhao, "Large-scale dataset pruning with dynamic uncertainty," *CoRR*, vol. abs/2306.05175, 2023.

[20] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *ICML*, vol. 382, 2009, pp. 41–48.

[21] M. Cilimkovic, "Neural networks and back propagation algorithm," *Institute of Technology Blanchardstown, Blanchardstown Road North Dublin*, vol. 15, no. 1, 2015.

[22] X. Wu, E. Dyer, and B. Neyshabur, "When do curricula work?" In *ICLR*, 2020.

[23] P. Soviany, R. T. Ionescu, P. Rota, and N. Sebe, "Curriculum learning: A survey," *Int. J. Comput. Vision*, vol. 130, no. 6, pp. 1526–1565, 2022.

[24] Y. Li, H. Fan, R. Hu, C. Feichtenhofer, and K. He, "Scaling language-image pre-training via masking," in *CVPR*, 2023, pp. 23 390–23 400.

[25] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *ICASSP*, 2015, pp. 5206–5210.

[26] T. Afouras, J. S. Chung, and A. Zisserman, "LRS3-TED: a large-scale dataset for visual speech recognition," *CoRR*, vol. abs/1809.00496, 2018.

[27] P. Ma, S. Petridis, and M. Pantic, "End-to-end audio-visual speech recognition with conformers," in *ICASSP*, 2021, pp. 7613–7617.

[28] P. Ma, S. Petridis, and M. Pantic, "Visual Speech Recognition for Multiple Languages in the Wild," *Nature Machine Intelligence*, pp. 930–939, 2022.

[29] P. Ma, A. Haliassos, *et al.*, "Auto-avsr: Audio-visual speech recognition with automatic labels," in *ICASSP*, 2023, pp. 1–5.

[30] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in *ICLR*, 2019.

[31] D. S. Park, W. Chan, *et al.*, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Interspeech*, 2019, pp. 2613–2617.