DISTIL-WHISPER: ROBUST KNOWLEDGE DISTILLATION VIA LARGE-SCALE PSEUDO LABELLING

Sanchit Gandhi, Patrick von Platen & Alexander M. Rush
Hugging Face
{sanchit, patrick, sasha}@huggingface.co

ABSTRACT

As the size of pre-trained speech recognition models increases, running these large models in low-latency or resource-constrained environments becomes challenging. In this work, we leverage pseudo-labelling to assemble a large-scale open-source dataset which we use to distill the Whisper model into a smaller variant, called Distil-Whisper. Using a simple word error rate (WER) heuristic, we select only the highest quality pseudo-labels for training. The distilled model is 5.8 times faster with 51% fewer parameters, while performing to within 1% WER on out-of-distribution test data in a zero-shot transfer setting. Distil-Whisper maintains the robustness of the Whisper model to difficult acoustic conditions, while being less prone to hallucination errors on long-form audio. Distil-Whisper is designed to be paired with Whisper for speculative decoding, yielding a 2 times speed-up while mathematically ensuring the same outputs as the original model. To facilitate further research in this domain, we make our training code, inference code and models publicly accessible.

1 INTRODUCTION

In recent years, Automatic Speech Recognition (ASR) systems have surpassed human-level accuracy in many academic benchmarks (Amodei et al., 2016; Baevski et al., 2020; Zhang et al., 2020), enabling a wide range of applications from transcription services to voice assistants (Aksënova et al., 2021). Whisper (Radford et al., 2022), a 1.5 billion parameter sequence-to-sequence (Seq2Seq) transformer model (Vaswani et al., 2017) pre-trained on 680,000 hours of weakly supervised speech recognition data, demonstrates a strong ability to generalise to many different datasets and domains (Gandhi et al., 2022). However, the ever-increasing size of pre-trained ASR models poses challenges when deploying these systems in low-latency settings or resource-constrained hardware (He et al., 2018; Zhang et al., 2022).

Recent efforts in natural language processing (NLP) have demonstrated promising advancements in compressing transformer-based models. Knowledge distillation (KD) has successfully been applied to reduce the size of models such as BERT (Devlin et al., 2019), without a significant performance loss on non-generative classification tasks (Sanh et al., 2019; Jiao et al., 2020; Sun et al., 2020). Inspired by machine translation methods, pseudo-labelling (PL) approaches (Kim & Rush, 2016) have also been explored for Seq2Seq summarisation (Shleifer & Rush, 2020), demonstrating the potential for substantial compression of Seq2Seq models on generative tasks. In the audio domain, KD has shown promising results for audio classification (Peng et al., 2021; Chang et al., 2021). However, similar results have not yet been achieved for the more difficult task of speech recognition.

In this paper, we apply distillation to the Whisper model in the context of Seq2Seq ASR. We address the challenge of maintaining robustness to different acoustic conditions through our construction of a large-scale open-source dataset covering 10 distinct domains. By pseudo-labelling the data, we ensure consistent transcription formatting across the dataset and provide sequence-level distillation signal. We propose a simple word error rate (WER) based heuristic for filtering the pseudo-labelled data and demonstrate that it is an effective method for ensuring good downstream performance of the distilled model.

We demonstrate that Distil-Whisper maintains the robustness of Whisper to different audio domains and noisy acoustic conditions. We measure this by evaluating the distilled models on four out-of-distribution test sets spanning multiple audio domains. The best model performs to within 1% WER of original Whisper checkpoint, while being 5.8 times faster with 51% fewer parameters. On long-form evaluation, the distilled model outperforms Whisper by 0.1% WER. We show that this performance gain is due to a lower propensity to hallucinate than the original Whisper model.

By sharing the same encoder weights as Whisper, Distil-Whisper can be used efficiently as an assistant model to Whisper for speculative decoding (Leviathan et al., 2023), for which we achieve a 2 times improvement in inference speed with only an 8% increase to parameter count. Speculative decoding algorithmically ensures that predictions of the main model are unchanged, meaning it can be used as a drop-in replacement for existing speech recognition pipelines using Whisper.

Our work suggests that large-scale pseudo-labelling of speech data has been under-explored and that it provides a promising technique for KD. To serve as a basis for further research on distillation for speech recognition, we release training code, inference code and models under `https://github.com/huggingface/distil-whisper`.

2 Related Work

In the NLP domain, model distillation has demonstrated substantial promise in reducing model size and computational requirements with minimal degradation to performance. Sanh et al. (2019) use a weighted average of the KD loss and the traditional cross entropy data loss to train DistilBERT, a 6 layer distilled version of BERT (Devlin et al., 2019), that achieves a 40% decrease in model size, a 60% increase in speed, and a 97% preservation of language under-standing capabilities on the GLUE benchmark (Wang et al., 2019). Shleifer & Rush (2020) extend the DistilBERT methodology the Seq2Seq setting, by initialising the student decoder from maximally spaced layers of the teacher decoder and incorporating intermediate hidden-states into the KD loss function. The resulting model, DistilBART, outperforms the original model on the XSUM and CNN/Daily Mail datasets (Narayan et al., 2018; See et al., 2017), with 37% model compression and a 48% increase in speed. Du et al. (2023) demonstrate that while distilled models perform well on in-distribution (ID) evaluation data, they perform significantly worse than their pre-trained counterparts on out-of-distribution (OOD) test sets. By training on a diverse, large-scale pseudo-labelled dataset, we preserve the robustness to different acoustic conditions, demonstrated by an ability to generalise to OOD test data.

KD has also been applied to the ASR task, albeit with a focus on encoder-only models. Peng et al. (2021) apply KD to the Wav2Vec 2.0 model (Baevski et al., 2020), achieving 79% model compression and 59% increase in speed. However, these gains come at the expense of a 6.9% increase to WER on the LibriSpeech corpus (Panayotov et al., 2015). Chang et al. (2021) apply a similar method to the HuBERT model (Hsu et al., 2021), and too report a 7.0% WER increase. Pang et al. (2018) attempt to distill LAS Chan et al. (2016), an early Seq2Seq ASR model, but find their best distilled model performs 2.2% WER worse than its larger counterpart. This paper focuses on KD of Seq2Seq models, with substantial model compression but also preserving WER performance on OOD test data.

Previous studies involving distilling the Whisper model have predominantly been centered around reducing model size and memory footprint. Shao et al. (2023) applied KD in combination with Quantisation Aware Training (QAT) (Jacob et al., 2017), demonstrating that significant parameter reduction is possible with only marginal performance decrement. However, the student model is trained and tested on a small corpus of ID data, giving no measure of its ability to generalise to OOD data, and thus its robustness to different acoustic conditions (Geirhos et al., 2020; Radford et al., 2022). Furthermore, this work did not consider optimising the model for latency. This paper seeks to distill the Whisper model to achieve significant model compression, jointly with latency improvements and WER performance on OOD test data. We also evaluate the distilled models' robustness to noisy audio conditions.

Table 1: Dimensionality details of the pre-trained Whisper checkpoints.

Model	Layers	Width	Heads	Parameters / M
tiny.en	4	384	6	39
base.en	6	512	8	74
small.en	12	768	12	244
medium.en	24	1024	16	769
large-v2	32	1280	20	1550

3 BACKGROUND

Whisper (Radford et al., 2022) is a sequence-to-sequence (Seq2Seq) transformer model (Vaswani et al., 2017) pre-trained on 680,000 hours of noisy speech recognition data web-scraped from the internet. When scaled to this quantity of data, Whisper yields competitive results with fully supervised systems, but in a *zero-shot* setting without the need for any fine-tuning.

Whisper is composed of a transformer-based encoder (Enc) and decoder (Dec). Assume we have an input speech signal comprised of T feature vectors $\boldsymbol{X}_{1:T} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$ and a target transcription $\boldsymbol{y}_{1:N} = \{y_1, \ldots, y_N\}$ of N tokens in the standard speech recognition setting. The encoder $\mathcal{F}_{Enc}$ is trained to map $\boldsymbol{X}_{1:T}$ to a sequence of hidden-state vectors $\boldsymbol{H}_{1:M}$:

$$\mathcal{F}_{Enc} : \boldsymbol{X}_{1:T} \rightarrow \boldsymbol{H}_{1:M} \tag{1}$$

The sequence length of the hidden-states M is typically half than that of the input speech feature sequence T by action of the convolutional layers in the encoder stem that downsample the input.

The decoder auto-regressively predicts a probability distribution for the next token y_i, conditional on all previous tokens $\boldsymbol{y}_{<i}$ and the encoder hidden-states $\boldsymbol{H}_{1:M}$:

$$P\left(y_i | \boldsymbol{y}_{<i}, \boldsymbol{H}_{1:M}\right) \tag{2}$$

To train the Whisper model, we assume a dataset where each example $(\boldsymbol{X}_{1:T}, \boldsymbol{y}_{1:N})$ is an (audio, text) pair. The model is trained using the standard cross-entropy (CE) loss, where the model is trained to predict an instance class by maximising the estimated probability of the target class labels:

$$\mathcal{L}_{CE} = -\sum_{i=1}^{N} P\left(y_i | \boldsymbol{y}_{<i}, \boldsymbol{H}_{1:M}\right) \tag{3}$$

There are five variants of the Whisper model summarised in Table 1. The models share the same Seq2Seq architecture but have different dimensionality. For all model sizes, the encoder and decoder have the same width, heads and number of layers in the transformer blocks. The first version of the Whisper paper introduced a large-v1 checkpoint, which was subsequently re-trained with regularisation more training epochs to give an improved large-v2 version (Radford et al., 2022). As both models share the same dimensions, we present results for the large-v2 model in this paper.

4 ROBUST KNOWLEDGE-DISTILLATION

4.1 KNOWLEDGE DISTILLATION

Knowledge distillation (KD) (Hinton et al., 2015) is a compression technique in which a smaller student model is trained to reproduce the behaviour of a larger teacher one. Compared to minimising the CE loss between the student model's predictions and the training labels, KD allows the student model to learn from the full predictive distribution of possible next tokens in a given context, rather than just maximising the probability of the next target token in the training data.

Shrink and Fine-Tune The most basic distillation method involves shrinking the teacher model to a smaller student size, and training the student on the CE objective in Equation 3. Following Shleifer & Rush (2020), we perform layer-based compression by initialising the student model by copying the weights from maximally spaced layers of the teacher model. For example, when initialising a 2-layer student model from a 32-layer teacher model, we copy the 1st and 32nd layers from the teacher to the student. Given the simplicity and effectiveness of this strategy in the Seq2Seq summarisation setting (Shleifer & Rush, 2020; Li et al., 2022), we use it for all distillation methods.

Pseudo Labelling In the pseudo-label setting (Kim & Rush, 2016), we replace the ground truth text transcription $\boldsymbol{y}_{1:N}$ with the teacher's generation $\hat{\boldsymbol{y}}_{1:N'}$ for the corresponding input audio $\boldsymbol{X}_{1:T}$:

$$\mathcal{L}_{PL} = -\sum_{i=1}^{N'} P\left(y_i | \hat{\boldsymbol{y}}_{<i}, \boldsymbol{H}_{1:M}\right) \tag{4}$$

This form of distillation can be viewed as "sequence-level" KD, where knowledge is transferred from the teacher model to the student model across a sequence of generated pseudo-labels (Kim & Rush, 2016).

Kullback-Leibler Divergence In the KL Divergence (Kullback & Leibler, 1951) setting, the full probability distribution of the student model P_i is trained to match the full distribution of the teacher model Q_i by minimising the KL divergence over the entire set of next possible tokens at position i:

$$\mathcal{L}_{KL} = \sum_{i=1}^{N} KL\left(Q_i, P_i\right) \tag{5}$$

This can be viewed as "word-level" KD, where knowledge is transferred from the teacher model to the student model via the logits over the possible tokens (Kim & Rush, 2016). The KL Divergence is attractive since it provides information over all classes and has less variance in gradients than the CE loss (Hinton et al., 2015).

Objective The final KD training objective is a weighted sum of the KL and PL terms:

$$\mathcal{L}_{KD} = \alpha_{KL}\mathcal{L}_{KL} + \alpha_{PL}\mathcal{L}_{PL} \tag{6}$$

where α_{KL} and α_{PL} are scalar weights for the KL and loss terms respectively. Following (Shleifer & Rush, 2020), we set $\alpha_{KL} = 0.8$ and $\alpha_{PL} = 1.0$.

4.2 Pseudo-Label Selection: WER Threshold

The pseudo-labels generated by the Whisper model are subject to transcription errors and hallucinations (Bain et al., 2023; Zhang et al., 2023). To ensure we only train on accurate pseudo-labels, we propose a simple heuristic to filter our pseudo-labelled training data. For each training sample, we normalise the ground truth labels $\boldsymbol{y}_{1:N}$ and the Whisper generated pseudo-labels $\hat{\boldsymbol{y}}_{1:N'}$ using the Whisper English normaliser (Radford et al., 2022). We compute the word error rate (WER) between the normalised ground truth and normalised psuedo-labels, and discard any samples that exceed a given WER threshold λ:

$$\text{WER}\left(\text{norm}\left(\boldsymbol{y}_{1:N}\right), \text{norm}\left(\hat{\boldsymbol{y}}_{1:N'}\right)\right) > \lambda \tag{7}$$

We tune the value of λ on our validation sets, and demonstrate in Section 9.1 that this simple filtering method improves transcription quality and downstream model performance.

5 CHUNKED LONG-FORM TRANSCRIPTION

Whisper models have a fixed receptive field corresponding to 30-seconds of input audio and cannot process longer audio inputs at once. Most academic datasets comprise of short utterances less than 30-seconds in duration, and so this is not a problem. However, real-world applications such as meeting transcriptions typically require transcribing long audio files of many minutes or hours.

The original Whisper paper presents a long-form transcription algorithm that sequentially transcribes 30-second segments of audio and shifts the sliding window according to the timestamps predicted by the model. This auto-regressive algorithm requires both beam-search and temperature fallback to ensure accurate long-form transcription (Radford et al., 2022).

We use an alternative strategy, first proposed by Patry (2022), in which the long audio file is chunked into smaller segments with a small overlap between adjacent segments. The model is run over each chunk and the inferred text is joined at the strides by finding the longest common sequence between overlaps. This stride enables accurate transcription across chunks without having to transcribe them sequentially. We observe that this algorithm only requires greedy decoding to reliably transcribe long audio files. Furthermore, this algorithm is semi auto-regressive in the sense that the chunks can be transcribed in any order, provided adjacent chunks are subsequently joined correctly at their boundaries. This allows chunks to be transcribed in parallel through batching. In practice, this yields up to 9 times improvements in inference speed compared to sequential transcription over long audio files. In this work, use the chunked long-form transcription algorithm when evaluating both the Whisper and Distil-Whisper models.

6 SPECULATIVE DECODING

Speculative decoding (SD) (Leviathan et al., 2023) is a method for accelerating the inference of auto-regressive transformer models by employing a faster assistant model. The assistant model generates a sequence of candidate tokens, all of which are verified by the main model in a single forward pass. By generating with the faster assistant model and only performing validation forward passes with the main model, the decoding process is sped-up significantly. The i-th candidate token from the assistant model $\hat{y}_i$ is only kept if all previous candidate tokens $\hat{\boldsymbol{y}}_{<i}$ match the validation tokens predicted by the main model. Consequently, speculative decoding ensures that the generated output exactly matches the sequence of tokens that would be generated by the main model, making it a natural replacement for existing inference pipelines that use the main model.

DistilSpec (Zhou et al., 2023) proposes using a knowledge-distilled student model as the assistant to better align the distribution of the assistant model with the main one. We apply the same principal here, using Distil-Whisper as the assistant to Whisper.

7 EXPERIMENTAL SETUP

7.1 DATA

Inspired by SpeechStew (Chan et al., 2021), we assemble a large corpus of ASR training data for large-scale KD through a combination of nine publicly available speech recognition datasets. An overview of the datasets is presented in Table 2, with additional details in Appendix A.1. The combined dataset contains 21,170 hours of speech data, encompassing over 18,260 speakers and 10 distinct domains. We load and pre-process all datasets in the Hugging Face Datasets library (Lhoest et al., 2021), streaming the data from the Hugging Face Hub[1].

We generate pseudo-labels for our training data with the Whisper large-v2 checkpoint, using the Flax Whisper implementation in the Hugging Face Transformers library (Heek et al., 2020; Wolf et al., 2020). We found there to be little difference in the downstream performance of the distilled model after pseudo-labelling using either greedy or beam-search, and so we opted to pseudo-label the training data with greedy decoding for its faster inference speed.

[1]Training datasets: `https://huggingface.co/collections/distil-whisper/training-datasets-6538d05c69721489d1db1e49`

Table 2: Summary of the open-source datasets used for training. For some datasets, the number of speakers cannot be reliably retrieved. We denote these entries as "unknown".

Dataset	Size / h	Speakers	Domain	Licence
People's Speech	12,000	unknown	Government, interviews	CC-BY-SA-4.0
GigaSpeech	2,500	unknown	Audiobook, podcast, YouTube	apache-2.0
Common Voice 13	2,400	unknown	Narrated Wikipedia	CC0-1.0
Fisher	1,960	11,900	Telephone conversations	LDC
LibriSpeech	960	2,480	Audiobooks	CC-BY-4.0
VoxPopuli	540	1,310	European Parliament	CC0
TED-LIUM	450	2,030	TED talks	CC-BY-NC-ND 3.0
SwitchBoard	260	540	Telephone conversations	LDC
AMI	100	unknown	Meetings	CC-BY-4.0
Total	21,170	18,260+		

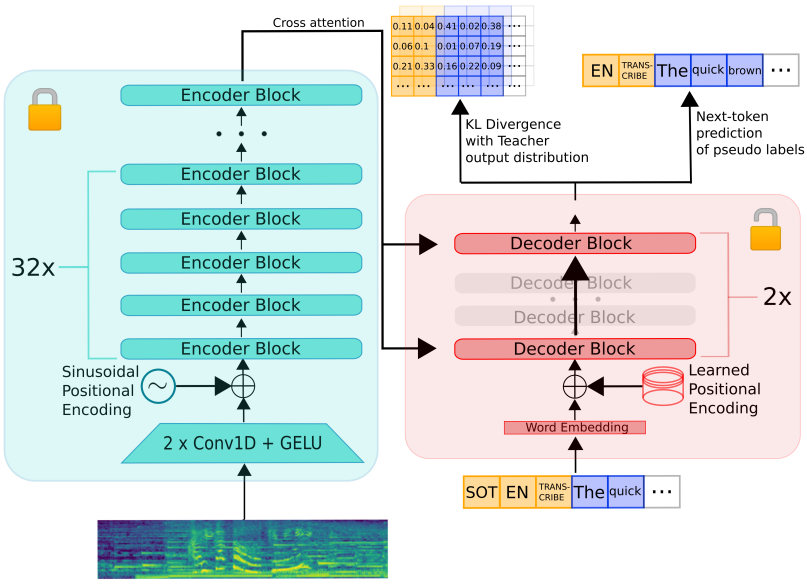

Figure 1: **Architecture of the Distil-Whisper model.** The encoder (shown in green) is entirely copied from the teacher to the student and frozen during training. The student's decoder consists of only two decoder layers, which are initialised from the first and last decoder layer of the teacher (shown in red). All other decoder layers of the teacher are discarded. The model is trained on a weighted sum of the KL divergence and PL loss terms.

7.2 Training

We initialise the student models by copying the entire encoder from the teacher and freeze it during training. We distill 2-layer decoder checkpoints from the medium.en and large-v2 models by copying the first and last decoder layers, which we refer to as distil-medium.en and distil-large-v2 respectively. The dimensionality details of the distilled models are shown in Table 3, with the architecture and training objective summarised in Figure 1.

We train with a batch size of 256 for a total of 80,000 optimisation steps, which amounts to eight epochs of training. Since we only train for eight epochs, the risk of over-fitting is low, and so we do not use any data augmentation or regularisation techniques. Instead, we rely on the diversity of our dataset to ensure model generalisation and robustness, the same premise used in training the original Whisper model (Radford et al., 2022). Refer to Appendix B.1 for full details of our training set-up.

Table 3: Dimensionality details of Distil-Whisper checkpoints.

Model	Enc. Layers	Dec. Layers	Width	Heads	Params. / M
distil-medium.en	24	2	1024	16	394
distil-large-v2	32	2	1280	20	756

Table 4: Summary of the OOD datasets used for short and long-form evaluation.

Dataset	Size / h	Speakers	Domain	Licence
Short-Form				
CHiME-4	7	87	News broadcast	LDC
Earnings-22	115	unknown	Financial meetings	CC-BY-SA-4.0
FLEURS	2	3	Narrated Wikipedia	CC-BY-4.0
SPGISpeech	100	unknown	Financial meetings	User Agreement
Long-Form				
Earnings-21	39	unknown	Financial meetings	CC-BY-SA-4.0
Earnings-22	115	unknown	Financial meetings	CC-BY-SA-4.0
Meanwhile	1	1	TV show	User Agreement
Rev 16	16	unknown	Podcasat	CC-BY-4.0

7.3 Short-Form Evaluation

The objective of Distil-Whisper is to compress the original Whisper model into a smaller, faster variant of the model that retains its robustness to different acoustic conditions (speakers, speaking styles and domains). To investigate this capability, we employ a broad collection of speech recognition datasets to examine whether Distil-Whisper can effectively generalise across datasets and domains.

We evaluate the Distil-Whisper model on a total of 15 short-form datasets. The first 11 of these datasets are the corresponding test splits for the training data used to distil the model. These test splits are *in distribution (ID)* with the training data. The remaining four datasets are used as test sets only, where Distil-Whisper is assessed in a zero-shot setting, that is without the use of the corresponding training data, thereby measuring the model's ability to generalise to *out of distribution (OOD)* datasets. An overview of the OOD evaluation datasets is presented in Table 4. Full details of the evaluation datasets are provided in Appendix A.2.

We examine both overall robustness, that is the average performance over all datasets, and effective robustness (Taori et al., 2020), which measures the difference in expected performance between a reference dataset that is ID, and one or more datasets that are OOD. A model with high effective robustness does better on OOD datasets as a function of performance on the reference dataset. A model with ideal effective robustness performs equally well on all datasets. In our experiments, we use GigaSpeech (Chen et al., 2021) as the reference dataset, owing to the fact it contains web-scraped data from audiobooks, podcasts and YouTube videos, and is such ID with both the pre-trained Whisper training data and the distilled Whisper train set.

We evaluate the noise robustness of the Distil-Whisper models, the original Whisper models, and eight other LibriSpeech-trained models by measuring the WER on the LibriSpeech test-clean dataset with increasing amounts of noise applied to the input audio. The LibriSpeech dataset is an ideal choice of dataset since it has a high signal-to-noise ratio (SNR), and thus enablers evaluation over a large range of SNRs as the amount of noise is increased. We add either white noise or pub noise from the Audio Degradation Toolbox (Mauch & Ewert, 2013). The pub noise simulates a naturally noisy environment, with ambient sounds and indistinguishable conversations characteristic of a busy restaurant or pub. The level of additive noise is determined based on the signal power of individual instances, and corresponds to a specified SNR.

Table 5: **Distil-Whisper retains the WER performance of the Whisper model but with faster inference speed.** Average WER results over the four OOD short-form test sets and the four OOD long-form test sets. Relative latency is the inference time relative to the large-v2 checkpoint. For short-form evaluation, the batch size is set to 1. For long-form evaluation, the chunked long-form transcription algorithm is used with a batch size of 16.

Model	Params / M	Short Form		Long Form	
		Rel. Latency	Avg. WER	Rel. Latency	Avg. WER
tiny.en	**39**	6.1	18.9	5.4	18.9
base.en	74	4.9	14.3	4.3	15.7
small.en	244	2.6	10.8	2.2	14.7
medium.en	769	1.4	9.5	1.3	12.3
large-v2	1550	1.0	**9.1**	1.0	11.7
distil-medium.en	394	**6.8**	11.1	**8.5**	12.4
distil-large-v2	756	5.8	10.1	5.8	**11.6**

7.4 Long-Form Evaluation

We evaluate the long-form transcription performance of the Distil-Whisper model on four OOD datasets comprising different lengths and acoustic conditions, in order to cover the broadest possible distribution of data. An overview of the long-form datasets is presented in Table 4. Full details about the long-form datasets are provided in Appendix A.3.

The Whisper model demonstrates a susceptibility to hallucinate, characterised by either the repetitive generation of identical sequences, or predicting passages of text not spoken in the audio input (Bain et al., 2023; Zhang et al., 2023). These hallucinations errors are most prevalent in long-form audio transcription, particularly when the audio contains large amounts of silence between spoken utterances. To quantify the amount of repetition and hallucination in the predicted transcriptions, we measure the number of repeated 5-gram word duplicates (5-Dup.) and the insertion error rate (IER) over the four OOD long-form datasets. We also report the substitution error rate (SER) and deletion error rate (DER) to quantify the frequency of substitutions and deletions in the transcriptions.

8 Results

8.1 Short-Form Evaluation

Table 5 reports the average WER scores over the four OOD short-form test sets for the Whisper and Distil-Whisper checkpoints. For a detailed breakdown of results on a per-dataset basis, refer to Appendix C. Of the two distilled models, the distil-large-v2 model achieves the lowest overall average WER of 10.1%. It is one percentage point higher than the large-v2 baseline, with 5.8 times faster inference speed and fewer than half the parameters. The inference speed is comparable to the tiny.en Whisper checkpoint, but with an 8.8% WER advantage. The distil-medium.en model is on average 2.0% WER higher than the large-v2 model, with 6.8x faster inference and 75% model compression. These findings highlight that Distil-Whisper retains the overall robustness of the Whisper model, with comparable WER performance averaged over multiple OOD datasets, but with significantly faster inference speed and reduced parameter count.

Table 6 compares the effective robustness of large-v2 to distil-large-v2. The models have very close performance on the reference distribution, performing to within 2% relative WER. The distilled model improves upon the pre-trained baseline for the SPGISpeech dataset by 12.8% relative, but performs worse on the three other OOD datasets. Compared to the teacher model, the distilled model achieves an overall WER increase of 0.8% absolute (or 10.7% relative). The narrow performance gap indicates that Distil-Whisper has comparable effective robustness to the original Whisper model.

Table 6: **Effective robustness across various datasets.** WER results for one ID reference dataset and four OOD datasets. The relative error rate (RER) is shown on the right, giving the per-dataset effective robustness scores. The macro-average results are shown in the bottom row.

Dataset	distil-large-v2	distil-large-v2	RER
GigaSpeech	10.7	10.5	-2.0
CHIME-4	11.8	14.0	18.4
Earnings-22	16.6	16.9	1.6
FLEURS	4.2	6.3	48.2
SPGISpeech	3.8	3.3	-12.8
Average	9.4	10.2	10.7

Table 7: **Comparison of long-form transcription algorithms.** Average WER results over the four OOD long-form test sets for the sequential and chunked long-form algorithms. Relative latency is the inference time relative to the large-v2 model with sequential long-form decoding. The chunked transcription results are reported using a batch size of 16.

Model	Algorithm	Rel. Latency	Avg. OOD WER
large-v2	Sequential	1.0	**10.4**
large-v2	Chunked	9.9	11.7
distil-large-v2	Chunked	**57.5**	11.6

8.2 Long-Form Evaluation

We compare the long-form transcription performance of Distil-Whisper to the pre-trained Whisper models on the four OOD long-form test sets. Table 5 reports the relative latency for a batch size of 16, as well as the macro-average WER. The per-dataset WER scores are provided in Appendix C. The results show that distil-large-v2 outperforms or equals large-v2 on four of the five test sets, with an average WER that is 0.1% lower with 5.8 times faster batched inference speed. It is only on the Meanwhile dataset that the Distil-Whisper model performs worse, which contains recordings from a single speaker with a high-frequency of uncommon words.

Table 7 compares the performance of the Whisper long-form sequential algorithm to the Distil-Whisper chunked one. The large-v2 model with the chunked algorithm yields a 9.9 times speed-up compared to the sequential one, with a 1.3% increase to average OOD WER. This demonstrates the inference speed gain that is achieved through batching. Using the distilled model in combination with the chunked algorithm provides further improvements: the distil-large-v2 model is 57.5 times faster than the baseline large-v2 implementation, while performing to within 1.2% WER.

8.3 Robustness to Additive Noise

Figure 2 shows how WER performance degrades as the intensity of additive noise increases on the LibriSpeech test-clean dataset. Of the 14 models we compare to, eight are pre-trained and/or fine-tuned on LibriSpeech. There are many models that outperform the Distil-Whisper models under low noise (40 dB SNR), including the Whisper checkpoints and LibriSpeech trained models. However, as the noise becomes more intensive, the WERs of the Distil-Whisper checkpoints degrade less severely than the LibriSpeech trained models and approach those of Whisper, especially when the additive pub noise decreases below 10 dB. Since we copy the full encoder and freeze it during training, the student and teacher models share the same encoder. Thus, they show similar robustness to noise, particularly under more natural distribution shifts like pub noise.

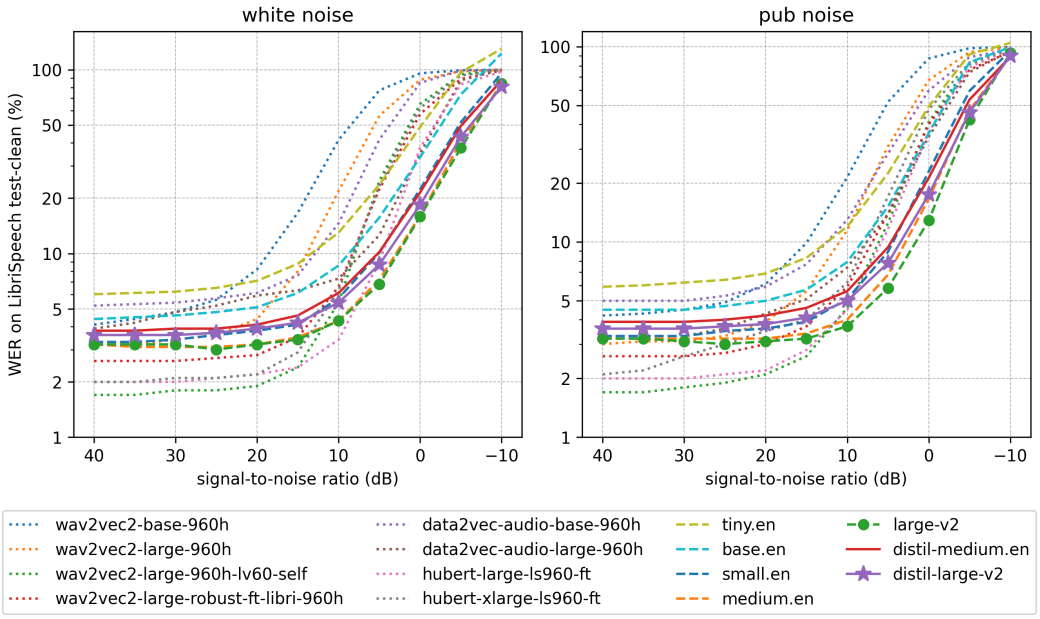

Figure 2: **Effect of noise on WER performance.** WER on LibriSpeech test-clean as a function of SNR under additive white noise (left) and pub noise (right).

8.4 Robustness to Hallucinations

Table 8 reports the number of repeated 5-gram word duplicates (5-Dup.) and insertion error rate (IER) metrics averaged over the four long-form test sets. In addition to the Whisper and Distil-Whisper models, we report the results for the official Wav2Vec 2.0 large model fine-tuned on 960 hours of LibriSpeech data. This checkpoint provides a comparison between the Whisper Seq2Seq architecture and a CTC based one (Graves et al., 2006). A CTC model should be less prone to hallucination errors given it is an encoder-only architecture with a linear head over the vocabulary.

The distil-large-v2 model has 1.3 times fewer repetition errors than Whisper large-v2. It also obtains the lowest average IER, improving on Whisper by 1.2% absolute. This indicates that the amount of hallucination is improved in Distil-Whisper compared to the original Whisper model. The average deletion error rate (DER) is comparable for both large-v2 and distil-large-v2, performing to within 0.3% DER. However, the substitution error rate (SER) is 1.4% higher for distil-large-v2, indicating that the distilled models are subject to more substitution errors. Overall, the reduction in IER outweighs the increase to SER, and Distil-Whisper returns the lowest WER of all the models. While the wav2vec 2.0 model underperforms in its average WER score, we find that it is far less prone to repetition errors compared to both Whisper and Distil-Whisper. Further work is needed to reduce repetition errors in Seq2Seq ASR models.

8.5 Speculative Decoding

Table 9 reports the relative latency of the medium.en and large-v2 models with speculative decoding. We compare the latency using either the smallest Whisper checkpoints or the Distil-Whisper models as the assistant. Since the outputs of the original Whisper models are obtained exactly, we report the relative latency only. The distilled student models are initialised by copying and freezing the entire encoder from the teacher, meaning they use exactly the same encoder as main Whisper models. Therefore, when running SD, the encoder can be shared between the main and assistant models, and only the distilled decoder layers have to be loaded in addition to the main model. This results in just an 8% increase to parameter count when using distil-large-v2 as the assistant to large-v2.

Table 8: **Detailed long-form error rates.** Average number of repeated 5-gram word duplicates (5-Dup.) and insertion error rate (IER) over the four long-form test sets. Shown also are the average substitution error rate (SER), deletion error rate (DER) and word error rate (WER) metrics.

Model	5-Dup.	IER	SER	DER	WER
wav2vec2-large-960h	**7971**	4.8	18.9	4.6	28.3
tiny.en	23313	5.1	8.9	4.8	18.9
base.en	22719	4.3	6.6	4.8	15.7
small.en	26377	3.3	5.0	6.5	14.7
medium.en	23549	3.5	4.2	4.6	12.3
large-v2	23792	3.3	**3.9**	4.5	11.7
distil-medium.en	18918	2.5	5.6	4.4	12.4
distil-large-v2	18503	**2.1**	5.3	**4.2**	**11.6**

Table 9: **Impact of speculative decoding.** Relative latency of medium.en and large-v2 using Whisper and Distil-Whisper assistant models. The relative latency is computed relative to the large-v2 model without speculative decoding for a batch size of 1.

Model	Params / M	Rel. Latency
medium.en	769	1.4
with tiny.en	808	2.7
with distil-medium.en	856	3.3
large-v2	1550	1.0
with tiny	1589	2.1
with distil-large-v2	1669	2.0

Speculative decoding with the distil-large-v2 assistant yields a 2.0 times improvement to inference speed over large-v2 alone. This is comparable to using the tiny model as the assistant. For the medium.en model, using distil-medium.en as an assistant provides a 2.4 times speed-up. This is greater than using the tiny.en checkpoint as an assistant, which is only 2.0 times faster. Overall, speculative decoding provides significant speed-ups to latency while mathematically ensuring the same outputs, making it a natural replacement for existing Whisper pipelines.

9 Analysis

9.1 WER Threshold

During training, we filter pseudo-labelled data where the normalised WER between the Whisper-generated pseudo-labels and the ground truth labels exceeds a given threshold λ. To investigate the effect of this threshold on the performance of the distilled model, we train a series of distil-large-v2 checkpoints for 10,000 training steps (or two epochs) on a range of threshold values. Table 10 shows the average WER performance of the trained models. Setting the threshold too high allows mis-transcribed or hallucinated transcriptions to enter the training set. Setting the WER threshold low retains only the most accurate Whisper-generated pseudo-labels, but also only the easiest samples (i.e. those with very low WER) and discards a larger proportion of the training data. We found that a WER threshold of 10% provides a good trade-off between these opposing factors. Had we trained for longer, the effect of using a higher quantity of training data might have become more pronounced, thus favouring higher thresholds. Using a WER threshold to filter pseudo-labelled data may compensate for the decreased transcription accuracy of the Whisper-generated labels predicted with greedy decoding as opposed to beam-search. We find it is an effective strategy for improving the performance of Seq2Seq ASR systems trained on pseudo-labelled data.

Table 10: **The WER threshold is an effective filter for PL data.** Average WER of the distil-large-v2 checkpoint on the 11 ID and three OOD validation sets as the WER threshold λ is reduced.

λ	Data Filtered / %	Avg. ID WER	Avg. OOD WER	Avg. WER
100	0.0	14.8	9.1	13.4
80	6.2	13.5	7.5	12.1
40	11.9	13.3	7.4	12.0
20	24.2	13.1	7.3	11.7
15	32.0	13.0	7.4	11.7
10	45.4	12.6	7.4	11.4
5	60.3	12.6	7.3	11.4

Table 11: **Performance improves with increasing dataset size.** Average WER of the distil-large-v2 checkpoint on the 11 ID and three OOD validation sets as the amount of training data is increased.

Size / h	Proportion / %	Avg. ID WER	Avg. OOD WER	Avg. WER
435	2	17.1	13.8	16.4
871	4	15.1	10.5	14.0
1,742	8	14.0	9.2	12.9
3,483	16	13.3	7.8	12.0
6,966	32			
13,933	64	12.8	7.4	11.6
21,770	100	12.6	7.4	11.4

9.2 Dataset Scaling

To study the amount of data is required to distil Whisper, we trained a series of distil-large-v2 models on subsampled versions of our dataset. Table 11 shows the average WER performance of the distil-large-v2 model for each of the dataset proportions. All increases in dataset size result in improved performance on the ID validation sets. On the OOD validation sets, performance improves rapidly from 435 to 3,483 hours, and then slows down significantly between 3,483 hours and 13,933 hours. Using the full dataset of 21,770 hours – a further 1.6 times increase in size – results in no further improvement to OOD WER. This suggests that there are diminishing gains after increasing the amount of pseudo-labelled training data above 13,933 hours.

9.3 Model Size

Table 12 shows the latency and WER performance for 16, 8, 4 and 2 decoder-layer models. The 16-layer decoder model largely retains the WER performance of the 32-layer teacher model, performing to within 0.1% OOD WER with a 1.9 times improvement in latency. As the number of decoder layers is reduced further, the average OOD WER increases more significantly. The maximum increase is 2.1% for the 2-layer decoder model, however this configuration is 5.8 times faster than the teacher model. These results highlight the trade-off between latency and performance as the number of decoder layers is reduced.

In distilling only the decoder, the parameter reduction is limited to 51%. To further reduce the parameter count, we can jointly distil the encoder and decoder. Table 12 compares the performance of a full 32-layer encoder student model to a reduced 16-layer one, both with 2-layer decoders. When the encoder is reduced to 16-layers, the parameter count decreases by an additional 19%. However, the OOD WER performance degrades by 3.1% absolute. This suggests that having a deep encoder is paramount for maintaining strong WER performance of distilled Seq2Seq ASR models.

Table 12: **Trade-off between latency and WER performance with decreasing model size.** Average WER over the 11 ID and three OOD validation sets as the number of encoder and decoder layers in the large-v2 checkpoint are reduced. The first row corresponds to the teacher checkpoint large-v2. The following rows correspond to the distilled models, which are trained for 10,000 optimisation steps (or two epochs).

Enc	Dec	Params / M	Rel. Latency	ID WER	OOD WER	Avg. WER
32	32	1543	1.0	12.8	5.3	11.1
32	16	1124	1.9	11.3	5.4	9.9
32	8	914	3.0	11.6	6.0	10.3
32	4	809	4.3	12.0	6.5	10.8
32	2	756	5.8	12.6	7.4	11.4
16	2	441	8.6	16.0	10.5	14.7

10 Conclusion

We introduce Distil-Whisper, a distilled version of Whisper that is 49% smaller, 5.8 times faster, and within 1% WER performance on OOD short-form audio. On OOD long-form audio, Distil-Whisper outperforms Whisper, due to fewer hallucinations and repetitions. We show that large-scale pseudo-labelling is an effective strategy for distilling ASR models, in particular when combined our WER threshold filter. We further demonstrate that Distil-Whisper can be used in combination with Whisper using speculative decoding to obtain the same outputs as the original model with 2 times faster inference.

11 Acknowlegements

We thank Nicolas Patry and Arthur Zucker for their implementation of the chunked long-form transcription algorithm in Transformers and João Gante for the implementation of speculative decoding. We gratefully acknowledge the support of Google's TPU Research Cloud (TRC) program for providing Cloud TPU resources for this research.

References

Alëna Aksënova, Daan van Esch, James Flynn, and Pavel Golik. How Might We Create Better Benchmarks for Speech Recognition? In *Proceedings of the 1st Workshop on Benchmarking: Past, Present and Future*, pp. 22–34, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.bppf-1.4. URL https://aclanthology.org/2021.bppf-1.4.

Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, Jie Chen, Jingdong Chen, Zhijie Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Ke Ding, Niandong Du, Erich Elsen, Jesse Engel, Weiwei Fang, Linxi Fan, Christopher Fougner, Liang Gao, Caixia Gong, Awni Hannun, Tony Han, Lappi Johannes, Bing Jiang, Cai Ju, Billy Jun, Patrick LeGresley, Libby Lin, Junjie Liu, Yang Liu, Weigao Li, Xiangang Li, Dongpeng Ma, Sharan Narang, Andrew Ng, Sherjil Ozair, Yiping Peng, Ryan Prenger, Sheng Qian, Zongfeng Quan, Jonathan Raiman, Vinay Rao, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Kavya Srinet, Anuroop Sriram, Haiyuan Tang, Liliang Tang, Chong Wang, Jidong Wang, Kaifu Wang, Yi Wang, Zhijian Wang, Zhiqian Wang, Shuang Wu, Likai Wei, Bo Xiao, Wen Xie, Yan Xie, Dani Yogatama, Bin Yuan, Jun Zhan, and Zhenyao Zhu. Deep Speech 2 : End-to-End Speech Recognition in English and Mandarin. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 173–182, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/amodei16.html.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. Common Voice: A Massively-Multilingual Speech Corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pp. 4218–4222, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://www.aclweb.org/anthology/2020.lrec-1.520.

Igor Babuschkin, Kate Baumli, Alison Bell, Surya Bhupatiraju, Jake Bruce, Peter Buchlovsky, David Budden, Trevor Cai, Aidan Clark, Ivo Danihelka, Claudio Fantacci, Jonathan Godwin, Chris Jones, Ross Hemsley, Tom Hennigan, Matteo Hessel, Shaobo Hou, Steven Kapturowski, Thomas Keck, Iurii Kemaev, Michael King, Markus Kunesch, Lena Martens, Hamza Merzic, Vladimir Mikulik, Tamara Norman, John Quan, George Papamakarios, Roman Ring, Francisco Ruiz, Alvaro Sanchez, Rosalia Schneider, Eren Sezener, Stephen Spencer, Srivatsan Srinivasan, Luyu Wang, Wojciech Stokowiec, and Fabio Viola. The DeepMind JAX Ecosystem, 2020. URL http://github.com/deepmind.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 12449–12460. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/92d1e1eb1cd6f9fba3227870bb6d7f07-Paper.pdf.

Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman. WhisperX: Time-Accurate Speech Transcription of Long-Form Audio. In *Proc. INTERSPEECH 2023*, pp. 4489–4493, 2023. doi: 10.21437/Interspeech.2023-78.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Jean Carletta. Unleashing the killer corpus: experiences in creating the multi-everything AMI Meeting Corpus. *Language Resources and Evaluation*, 41(2):181–190, 2007. ISSN 1574-020X. doi: 10.1007/s10579-007-9040-x.

William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4960–4964, 2016. doi: 10.1109/ICASSP.2016.7472621.

William Chan, Daniel Park, Chris Lee, Yu Zhang, Quoc Le, and Mohammad Norouzi. SpeechStew: Simply Mix All Available Speech Recognition Data to Train One Large Neural Network. *arXiv e-prints*, art. arXiv:2104.02133, April 2021.

Heng-Jui Chang, Shu wen Yang, and Hung yi Lee. Distilhubert: Speech Representation Learning by Layer-Wise Distillation of Hidden-Unit Bert. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7087–7091, 2021. URL https://api.semanticscholar.org/CorpusID:238354153.

Guoguo Chen, Shuzhou Chai, Guanbo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, Mingjie Jin, Sanjeev Khudanpur, Shinji Watanabe, Shuaijiang Zhao, Wei Zou, Xiangang Li, Xuchen Yao, Yongqing Wang, Yujun Wang, Zhao You, and Zhiyong Yan. GigaSpeech: An Evolving, Multi-domain ASR Corpus with 10,000 Hours of Transcribed Audio. *arXiv e-prints*, art. arXiv:2106.06909, June 2021.

Christopher Cieri, David Miller, and Kevin Walker. The Fisher Corpus: a Resource for the Next Generations of Speech-to-Text. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 2004a. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2004/pdf/767.pdf.

Christopher Cieri et al. Fisher English Training Speech Part 1 Speech LDC2004S13. Web Download. *Linguistic Data Consortium*, 2004b.

Christopher Cieri et al. Fisher English Training Speech Part 1 Transcripts LDC2004T19. Web Download. *Linguistic Data Consortium*, 2004c.

Christopher Cieri et al. Fisher English Training Speech Part 2 Speech LDC2005S13. Web Download. *Linguistic Data Consortium*, 2005a.

Christopher Cieri et al. Fisher English Training Speech Part 2 Transcripts LDC2005T19. Web Download. *Linguistic Data Consortium*, 2005b.

Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. FLEURS: Few-shot Learning Evaluation of Universal Representations of Speech. *arXiv e-prints*, art. arXiv:2205.12446, May 2022. doi: 10.48550/arXiv.2205.12446.

Tri Dao. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. *arXiv e-prints*, art. arXiv:2307.08691, July 2023. doi: 10.48550/arXiv.2307.08691.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. *arXiv e-prints*, art. arXiv:2205.14135, May 2022. doi: 10.48550/arXiv.2205.14135.

Miguel Del Rio, Natalie Delworth, Ryan Westerman, Michelle Huang, Nishchal Bhandari, Joseph Palakapilly, Quinten McNamara, Joshua Dong, Piotr Żelasko, and Miguel Jetté. Earnings-21: A Practical Benchmark for ASR in the Wild. In *Proc. Interspeech 2021*, pp. 3465–3469, 2021. doi: 10.21437/Interspeech.2021-1915.

Miguel Del Rio, Peter Ha, Quinten McNamara, Corey Miller, and Shipra Chandra. Earnings-22: A Practical Benchmark for Accents in the Wild. *arXiv e-prints*, art. arXiv:2203.15591, March 2022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Mengnan Du, Subhabrata Mukherjee, Yu Cheng, Milad Shokouhi, Xia Hu, and Ahmed Hassan Awadallah. Robustness Challenges in Model Distillation and Pruning for Natural Language Understanding. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 1766–1778, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.129. URL https://aclanthology.org/2023.eacl-main.129.

Daniel Galvez, Greg Diamos, Juan Torres, Keith Achorn, Juan Cerón, Anjali Gopi, David Kanter, Max Lam, Mark Mazumder, and Vijay Janapa Reddi. The People's Speech: A Large-Scale Diverse English Speech Recognition Dataset for Commercial Usage. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1. Curran, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/202cb962ac59075b964b07152d234b70-Paper-round1.pdf.

Sanchit Gandhi, Patrick von Platen, and Alexander M. Rush. ESB: A Benchmark For Multi-Domain End-to-End Speech Recognition. *arXiv e-prints*, art. arXiv:2210.13352, October 2022. doi: 10.48550/arXiv.2210.13352.

John S. Garofolo et al. CSR-I (WSJ0) Complete LDC93S6A. Web Download. *Linguistic Data Consortium*, 1993.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, Nov 2020. ISSN 2522-5839. doi: 10.1038/s42256-020-00257-z. URL https://doi.org/10.1038/s42256-020-00257-z.

J.J. Godfrey, E.C. Holliman, and J. McDaniel. SWITCHBOARD: telephone speech corpus for research and development. In *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pp. 517–520 vol.1, 1992. doi: 10.1109/ICASSP.1992.225858.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzman, and Angela Fan. The FLORES-101 Evaluation Benchmark for Low-Resource and Multilingual Machine Translation. *arXiv e-prints*, art. arXiv:2106.03193, June 2021. doi: 10.48550/arXiv.2106.03193.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pp. 369–376, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933832. doi: 10.1145/1143844.1143891. URL https://doi.org/10.1145/1143844.1143891.

Andreas Griewank and Andrea Walther. Algorithm 799: Revolve: An Implementation of Checkpointing for the Reverse or Adjoint Mode of Computational Differentiation. *ACM Trans. Math. Softw.*, 26(1):19–45, mar 2000. ISSN 0098-3500. doi: 10.1145/347837.347846. URL https://doi.org/10.1145/347837.347846.

Yanzhang He, Tara N. Sainath, Rohit Prabhavalkar, Ian McGraw, Raziel Alvarez, Ding Zhao, David Rybach, Anjuli Kannan, Yonghui Wu, Ruoming Pang, Qiao Liang, Deepti Bhatia, Yuan Shangguan, Bo Li, Golan Pundak, Khe Chai Sim, Tom Bagby, Shuo-yiin Chang, Kanishka Rao, and Alexander Gruenstein. Streaming End-to-end Speech Recognition For Mobile Devices. *arXiv e-prints*, art. arXiv:1811.06621, November 2018. doi: 10.48550/arXiv.1811.06621.

Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. Flax: A neural network library and ecosystem for JAX, 2020. URL http://github.com/google/flax.

François Hernandez, Vincent Nguyen, Sahar Ghannay, Natalia Tomashenko, and Yannick Estève. TED-LIUM 3: Twice as Much Data and Corpus Repartition for Experiments on Speaker Adaptation. In *Speech and Computer*, pp. 198–208. Springer International Publishing, 2018.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *arXiv e-prints*, art. arXiv:1503.02531, March 2015. doi: 10.48550/arXiv.1503.02531.

Jeremy Howard and Sebastian Ruder. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1031. URL https://aclanthology.org/P18-1031.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. *arXiv e-prints*, art. arXiv:2106.07447, June 2021. doi: 10.48550/arXiv.2106.07447.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. *arXiv e-prints*, art. arXiv:1712.05877, December 2017. doi: 10.48550/arXiv.1712.05877.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. TinyBERT: Distilling BERT for Natural Language Understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4163–4174, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.372. URL https://aclanthology.org/2020.findings-emnlp.372.

Norman P. Jouppi, Doe Hyun Yoon, George Kurian, Sheng Li, Nishant Patil, James Laudon, Cliff Young, and David Patterson. A Domain-Specific Supercomputer for Training Deep Neural Networks. *Commun. ACM*, 63(7):67–78, jun 2020. ISSN 0001-0782. doi: 10.1145/3360307. URL https://doi.org/10.1145/3360307.

Yoon Kim and Alexander M. Rush. Sequence-Level Knowledge Distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139.

S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951. ISSN 00034851. URL http://www.jstor.org/stable/2236703.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast Inference from Transformers via Speculative Decoding. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19274–19286. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. Datasets: A Community Library for Natural Language Processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 175–184, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.emnlp-demo.21.

Zheng Li, Zijian Wang, Ming Tan, Ramesh Nallapati, Parminder Bhatia, Andrew Arnold, Bing Xiang, and Dan Roth. DQ-BART: Efficient Sequence-to-Sequence Model via Joint Distillation and Quantization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 203–211, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.22. URL https://aclanthology.org/2022.acl-short.22.

Linguistic Data Consortium. 2000 HUB5 English Evaluation Transcripts LDC2002T43. Web Download. *Linguistic Data Consortium*, 2002.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Matthias Mauch and Sebastian Ewert. The Audio Degradation Toolbox and its Application to Robustness Evaluation. In *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR 2013)*, Curitiba, Brazil, 2013. accepted.

Abid Mohsin. Podcast transcription benchmark (part 1). https://www.rev.ai/blog/podcast-transcription-benchmark-part-1/, 2019. Accessed: 25 Oct., 2023.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1797–1807, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1206. URL https://aclanthology.org/D18-1206.

Patrick K. O'Neill, Vitaly Lavrukhin, Somshubra Majumdar, Vahid Noroozi, Yuekai Zhang, Oleksii Kuchaiev, Jagadeesh Balam, Yuliya Dovzhenko, Keenan Freyberg, Michael D. Shulman, Boris

Ginsburg, Shinji Watanabe, and Georg Kucsko. SPGISpeech: 5,000 Hours of Transcribed Financial Audio for Fully Formatted End-to-End Speech Recognition. In *Proc. Interspeech 2021*, pp. 1434–1438, 2021. doi: 10.21437/Interspeech.2021-1860.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210, 2015. doi: 10.1109/ICASSP.2015.7178964.

Ruoming Pang, Tara Sainath, Rohit Prabhavalkar, Suyog Gupta, Yonghui Wu, Shuyuan Zhang, and Chung-Cheng Chiu. Compression of End-to-End Models. In *Proc. Interspeech 2018*, pp. 27–31, 2018. doi: 10.21437/Interspeech.2018-1025.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the Difficulty of Training Recurrent Neural Networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pp. III–1310–III–1318. JMLR.org, 2013.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Nicolas Patry. Making automatic speech recognition work on large files with Wav2Vec2 in Transformers. https://huggingface.co/blog/asr-chunking, 2022. Accessed: 25 Oct., 2023.

Zilun Peng, Akshay Budhkar, Ilana Tuil, Jason Levy, Parinaz Sobhani, Raphael Cohen, and Jumana Nassour. Shrinking Bigfoot: Reducing wav2vec 2.0 footprint. In *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*, pp. 134–141, Virtual, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.sustainlp-1.14. URL https://aclanthology.org/2021.sustainlp-1.14.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. The Kaldi Speech Recognition Toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, December 2011. IEEE Catalog No.: CFP11SRW-USB.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust Speech Recognition via Large-Scale Weak Supervision. *arXiv e-prints*, art. arXiv:2212.04356, December 2022. doi: 10.48550/arXiv.2212.04356.

Steve Renals, Thomas Hain, and Herve Bourlard. Recognition and understanding of meetings the AMI and AMIDA projects. In *2007 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 238–247, 2007. doi: 10.1109/ASRU.2007.4430116.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv e-prints*, art. arXiv:1910.01108, October 2019. doi: 10.48550/arXiv.1910.01108.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident Adaptive Language Modeling. *arXiv e-prints*, art. arXiv:2207.07061, July 2022. doi: 10.48550/arXiv.2207.07061.

Abigail See, Peter J. Liu, and Christopher D. Manning. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1099. URL https://aclanthology.org/P17-1099.

Hang Shao, Wei Wang, Bei Liu, Xun Gong, Haoyu Wang, and Yanmin Qian. Whisper-KDQ: A Lightweight Whisper via Guided Knowledge Distillation and Quantization for Efficient ASR. *arXiv e-prints*, art. arXiv:2305.10788, May 2023. doi: 10.48550/arXiv.2305.10788.

Sam Shleifer and Alexander M. Rush. Pre-trained Summarization Distillation. *arXiv e-prints*, art. arXiv:2010.13002, October 2020. doi: 10.48550/arXiv.2010.13002.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2158–2170, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.195. URL https://aclanthology.org/2020.acl-main.195.

Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring Robustness to Natural Distribution Shifts in Image Classification. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 18583–18599. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/d8330f857a17c53d217014ee776bfd50-Paper.pdf.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Emmanuel Vincent, Shinji Watanabe, Aditya Arie Nugraha, Jon Barker, and Ricard Marxer. An Analysis of Environment, Microphone and Data Simulation Mismatches in Robust Speech Recognition. *Comput. Speech Lang.*, 46(C):535–557, nov 2017. ISSN 0885-2308. doi: 10.1016/j.csl.2016.11.005. URL https://doi.org/10.1016/j.csl.2016.11.005.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=rJ4km2R5t7.

Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux. VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 993–1003, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.80. URL https://aclanthology.org/2021.acl-long.80.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Yu Zhang, James Qin, Daniel S. Park, Wei Han, Chung-Cheng Chiu, Ruoming Pang, Quoc V. Le, and Yonghui Wu. Pushing the Limits of Semi-Supervised Learning for Automatic Speech Recognition. *arXiv e-prints*, art. arXiv:2010.10504, October 2020. doi: 10.48550/arXiv.2010.10504.

Yu Zhang, Daniel S. Park, Wei Han, James Qin, Anmol Gulati, Joel Shor, Aren Jansen, Yuanzhong Xu, Yanping Huang, Shibo Wang, Zongwei Zhou, Bo Li, Min Ma, William Chan, Jiahui Yu,

Yongqiang Wang, Liangliang Cao, Khe Chai Sim, Bhuvana Ramabhadran, Tara N. Sainath, Françoise Beaufays, Zhifeng Chen, Quoc V. Le, Chung-Cheng Chiu, Ruoming Pang, and Yonghui Wu. BigSSL: Exploring the Frontier of Large-Scale Semi-Supervised Learning for Automatic Speech Recognition. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1519–1532, 2022. doi: 10.1109/JSTSP.2022.3182537.

Yu Zhang, Wei Han, James Qin, Yongqiang Wang, Ankur Bapna, Zhehuai Chen, Nanxin Chen, Bo Li, Vera Axelrod, Gary Wang, Zhong Meng, Ke Hu, Andrew Rosenberg, Rohit Prabhavalkar, Daniel S. Park, Parisa Haghani, Jason Riesa, Ginger Perng, Hagen Soltau, Trevor Strohman, Bhuvana Ramabhadran, Tara Sainath, Pedro Moreno, Chung-Cheng Chiu, Johan Schalkwyk, Françoise Beaufays, and Yonghui Wu. Google USM: Scaling Automatic Speech Recognition Beyond 100 Languages. *arXiv e-prints*, art. arXiv:2303.01037, March 2023. doi: 10.48550/arXiv.2303.01037.

Yongchao Zhou, Kaifeng Lyu, Ankit Singh Rawat, Aditya Krishna Menon, Afshin Rostamizadeh, Sanjiv Kumar, Jean-François Kagy, and Rishabh Agarwal. DistillSpec: Improving Speculative Decoding via Knowledge Distillation. *arXiv e-prints*, art. arXiv:2310.08461, October 2023. doi: 10.48550/arXiv.2310.08461.

A Additional Dataset Details

A.1 Training Data

Quantitative and qualitative information about the training datasets is displayed in Tables 13 and 14 respectively. A detailed description of the training datasets is presented below.

People's Speech (Galvez et al., 2021) is a large-scale English speech recognition dataset which assembles audio-transcription pairs sourced from the internet. The dataset covers multiple sources, including interviews, radio and finance. We use the "clean" subset of the dataset, with approximately 12,000 hours of training data and corresponding validation and test splits.

GigaSpeech (Chen et al., 2021) is a multi-domain speech recognition corpus curated from audiobooks, podcasts and YouTube. It contains both narrated and spontaneous speech over a range of content material, including science, arts and sports. We use the large subset (2,500 hours) to train and the standard validation and test splits.

Common Voice (Ardila et al., 2020) is a collection of open-license, crowd-source speech datasets where contributors record themselves narrating text from Wikipedia in various languages. Given its crowd-sourced approach, the dataset exhibits significant diversity in audio quality and speakers. The recorded audio often contains challenges such as background noise, accent variations, hesitations, and incorporation of non-native words. We use the English subset of version 13.0 (16-3-2023), with approximately 2,400 hours and the canonical data splits.

Table 13: **Quantitative statistics of the training datasets.** The mean audio length is quoted in seconds, and the mean transcription length in number of words.

Dataset	Size / h	Speakers	Mean Audio (s)	Mean Text (words)
People's Speech	12,000	unknown	13.8	38.5
Common Voice 13	3,000	unknown	5.6	31.9
GigaSpeech	2,500	unknown	4.0	12.9
Fisher	1,960	11,900	3.3	10.1
LibriSpeech	960	2,480	12.1	32.9
VoxPopuli	540	1,310	10.3	26.1
TED-LIUM	450	2,030	6.1	18.3
SwitchBoard	260	540	4.8	8.3
AMI	100	unknown	2.6	7.3
Total	21,770	18,260+	7.1	19.8

Table 14: **Qualitative statistics of the training datasets.** The speaking styles are narrated (N), oratory (O) or spontaneous (S), or a combination of them.

Dataset	Domain	Rec. Cond.	Style	Licence
People's Speech	Internet Archive	Close-talk mic.	N, O, S	CC-BY-SA-4.0
Common Voice 13	Narrated Wikipedia	Close-talk mic.	N	CC0-1.0
GigaSpeech	Audiobook, podcast, YouTube	Close-talk mic.	N, S	apache-2.0
Fisher	Telephone conversations	Telephone	S	LDC
LibriSpeech	Audiobooks	Close-talk mic.	N	CC-BY-4.0
VoxPopuli	European Parliament	Close-talk mic.	O	CC0
TED-LIUM	TED talks	Close-talk mic.	O	CC-BY-NC-ND 3.0
SwitchBoard	Telephone conversations	Telephone	S	LDC
AMI	Meetings	Headset	S	CC-BY-4.0

Fisher (Cieri et al., 2004a) is a corpus of two-sided conversational telephone calls amongst speakers from the United States. We combine Part 1 (Cieri et al., 2004b;c) and Part 2 (Cieri et al., 2005a;b) of the dataset to give 1,960 hours of training data.

LibriSpeech (Panayotov et al., 2015) is a standard dataset for training and evaluating academic speech models. It is comprised of 960 hours of narrated audiobooks sourced from the LibriVox[2] project. The audiobook domain provides high-quality recording conditions, with little to no background noise. We use the standard split of train, validation (*dev-clean*, *dev-other*) and test sets (*test-clean*, *test-other*).

VoxPopuli (Wang et al., 2021) is a large-scale multilingual speech datasets consisting of European Parliament event recordings from 2009-2020. The speech is oratory and from the political domain, with mostly non-native speakers. We use the English subset with approximately 550 hours and dataset splits provided therein.

TED-LIUM (Hernandez et al., 2018) is a collection of English-language TED Talk conference videos. The talks span a variety of cultural, political, and academic themes. We use the Release 3 edition of the training set with approximately 450 hours and the legacy distribution of validation and test data.

SwitchBoard (Godfrey et al., 1992) is a 260 hour corpus of two-sided conversational telephone calls amongst speakers from the United States. We partition 5% of the SwitchBoard corpus to form the validation split The test sets are the Hub5Eval2000 (Linguistic Data Consortium, 2002) data with two subsets: SwitchBoard and CallHome.

AMI (Carletta, 2007; Renals et al., 2007) consists of 100 hours of meeting recordings captured using multiple recording streams in parallel. The corpus is manually annotated to provide the ground truth transcriptions. Individual samples of the AMI dataset contain very large audio files between 10 and 60 minutes in duration. We segment the audio samples according the the Kaldi (Povey et al., 2011) recipe for AMI[3] to yield utterance of suitable length for training ASR systems. This involves splitting samples longer than 30 words at the time-stamps for punctuation to yield shorter utterances.

We use the individual headset microphone (AMI IHM) and single distant microphone (AMI SDM) versions of the dataset, with the train, validation and test sets provided therein.

A.2 Short-Form Evaluation Data

A detailed description of the short-form evaluation datasets is provided below.

CHiME-4 (Vincent et al., 2017) comprises of narrated samples from the Wall Street Journal corpus (Garofolo et al., 1993). Recordings are performed in noisy environments using a 6-channel tablet

[2]LibriVox: `https://librivox.org/`

[3]AMI Kaldi recipe: `https://github.com/kaldi-asr/kaldi/tree/master/egs/ami/s5b`

based microphone array. We use the official 1-channel validation and test sets for evaluating our models.

Earnings-22 (Del Rio et al., 2022) is a 119-hour test set of earnings calls recorded by global companies. The dataset was developed with the intention of assembling a diverse range of speakers and accents speaking in the context of real-world financial meetings.

The Earnings-22 dataset contains audio recordings upwards of 10-minutes in duration. To create a short-form evaluation dataset, we segment these files into shorter samples up to 20-seconds in length. We first predict timestamps for the long audio files using the official wav2vec 2.0 base + 4-gram model (Baevski et al., 2020) fine-tuned on the LibriSpeech (Baevski et al., 2020) dataset. We then split samples at the predicted timestamps for punctuation. If the samples are still longer than 20-seconds, we split them again at the longest silence in the utterance.

FLEURS (Few-shot Learning Evaluation of Universal Representations of Speech) (Conneau et al., 2022) is a small-scale corpus for evaluating speech recognition systems in 102 languages. The transcription data is taken from the FLoRes-101 dataset (Goyal et al., 2021), a machine translation corpus with 3001 samples of English text each translated to 101 other languages. To assemble the FLEURS dataset, up to three native speakers are recorded narrating the sentence translations in their native language. The recorded audio data is paired with the sentence transcriptions, thus yielding a multilingual speech recognition corpus. We use the English-US (en_us) subset with 1 hour of validation and 2 hours of test data.

SPGISpeech (O'Neill et al., 2021) is an English speech recognition dataset comprising of company earnings calls that have been manually transcribed by S&P Global, Inc. We evaluate our models on the official validation and test splits, each of which is 100 hours.

A.3 Long-Form Evaluation Data

A detailed description of the long-form evaluation datasets is presented below.

Earnings-21 (Del Rio et al., 2021) is a 39-hour corpus of company earnings calls over various financial sections.

Earnings-22 (Del Rio et al., 2022) is a 119-hour test set of earnings calls recorded by global companies. The dataset was developed with the intention of assembling a diverse range of speakers and accents speaking in the context of real-world financial meetings.

Meanwhile (Radford et al., 2022) is a collection of 64 segments taken from The Late Show with Stephen Colbert. The transcriptions are taken from the closed-caption data for each video and corrected with manual inspection. The YouTube URLs are provided by the Whisper authors, along with the segment start-end times[4].

Rev 16 (Mohsin, 2019) is a set of 30 podcast recordings that are commonly used to benchmark ASR systems in production settings. We follow the Whisper authors in evaluating on a subset of 16 of the 30 files with IDs:

```
3 4 9 10 11 14 17 18 20 21 23 24 26 27 29 32
```

B Experimental Set-Up

B.1 Training

We train the models using the JAX and Flax neural network libraries (Bradbury et al., 2018; Heek et al., 2020). We use data parallelism across TPU v4-8 accelerators (Jouppi et al., 2020), with bfloat16 precision and gradient checkpointing (Griewank & Walther, 2000). We use a softmax temperature of 2.0 to smooth the distributions of the student and teacher models when computing the KL loss (Hinton et al., 2015). Models are trained with an Optax implementation of the AdamW optimiser (Babuschkin et al., 2020; Loshchilov & Hutter, 2019) and gradient norm clipping (Pascanu

[4]Meanwhile metadata: `https://github.com/openai/whisper/blob/main/data/meanwhile.json`

et al., 2013). We train for a total of 80,000 optimisation steps, equivalent to eight epochs of training. We use the slanted triangular learning rate (STLR) (Howard & Ruder, 2018) schedule, linearly increasing the learning rate from zero to a maximum of 1e-4 over the first 500 steps, and then linearly decaying it to zero. If the student encoder is the same as the teacher encoder, we freeze its parameters and only run it forward once. Back propagation is then only run through the distilled decoder. Table 15 summarises the training hyperparemters.

Table 15: Distil-Whisper training hyperparameters.

Hyperparameter	**Value**
Device	TPU v4-8
Updates	80,000
Batch size	256
Warmup steps	500
LR schedule	Linear decay
Precision	bfloat16
KL softmax temperature	2.0
Max grad norm	1.0
Optimizer	AdamW
β_1	0.9
β_2	0.999
ϵ	10^{-8}
Weight decay	0.0
Timestamp probability	50%

B.2 Evaluation

We evaluate all models in JAX on TPU v4-8 with greedy decoding unless specified otherwise. We normalise text using the Whisper English normaliser (Radford et al., 2022), which standardises text by removing or converting specific words, symbols, numeric expressions, and managing whitespace and spellings, in an attempt to only penalise a system when an error is caused by actually mistranscribing a word, and not by formatting differences. We measure transcription accuracy using the WER metric.

During training, we evaluate the intermediate checkpoints every 5k training steps on the 13 validation sets. We select the checkpoint with the best macro-average performance over the validation splits for final evaluation on the test splits.

For latency measurements, we evaluate the models in PyTorch (Paszke et al., 2019) using a single A100 40GB GPU in float16 precision. Specifically, we measure the total time taken to decode 256 samples from each of the four OOD test sets over batch sizes in the set $\{1, 4, 16\}$. Batch size 1 latency corresponds to short-form evaluation, where the models are evaluated without timestamp prediction. Batch sizes 4 and 16 correspond to long-form transcription, where the chunked long-form transcription algorithm is used. The Whisper models are evaluated with timestamp prediction and the Distil-Whisper models without. These configurations resulted in the best WER scores on the TED-LIUM long-form validation set.

Using the inference speed measurements, we compute the ratio of the inference time of Distil-Whisper to the Whisper large-v2 checkpoint, giving a figure for relative latency. We record all latency measurements using Flash Attention 2 (Dao, 2023), since it is a general inference optimisation for modern GPU hardware in production. In Section D.5, we show the effect of Flash Attention 2 on the latency of Whisper and Distil-Whisper.

C Evaluation Results

Table 16: Per-dataset WER scores over the 15 short-form test sets. The macro-average WER scores are shown for the 11 ID datasets, four OOD datasets, and an overall average over all 15 test sets.

Dataset	tiny.en	base.en	small.en	medium.en	large-v2	distil-medium.en	distil-large-v2
AMI IHM	22.9	19.9	17.4	16.4	16.9	16.1	**14.7**
AMI SDM	50.0	45.2	38.1	37.0	36.5	35.7	**33.9**
Call Home	23.8	20.3	19.0	16.0	17.5	15.1	**13.5**
Common Voice 13	28.9	21.4	15.3	12.3	**10.4**	15.3	12.9
GigaSpeech	13.5	12.1	11.0	10.8	10.7	11.2	**10.5**
LibriSpeech clean	5.9	4.4	3.3	3.1	**3.2**	3.9	3.6
LibriSpeech other	14.1	10.4	7.4	6.1	**5.6**	8.0	6.9
People's Speech	26.4	22.2	19.3	18.6	18.6	18.4	**16.5**
SwitchBoard	17.7	15.6	15.3	14.0	14.2	11.7	**11.2**
TED-LIUM	11.8	10.9	10.1	11.5	12.0	10.1	**9.6**
Voxpopuli	11.3	9.6	8.3	7.9	**7.3**	8.8	8.0
CHIME-4	32.7	24.1	15.7	12.7	**11.8**	15.1	14.0
Earnings-22	25.8	21.2	17.9	17.0	**16.6**	18.4	16.9
FLEURS	11.2	7.5	5.9	4.9	**4.2**	6.9	6.3
SPGISpeech	5.8	4.2	3.6	3.4	3.8	3.8	**3.3**
ID Average	20.6	17.5	15.0	14.0	13.9	14.0	**12.8**
OOD Average	18.9	14.3	10.8	9.5	**9.1**	11.1	10.1
Average	20.1	16.6	13.8	12.8	12.6	13.2	**12.1**

Table 17: Per-dataset WER scores over the five long-form test sets. The macro-average WER scores are shown for the one ID dataset, four OOD datasets, and an overall average over all five test sets.

Dataset	tiny.en	base.en	small.en	medium.en	large-v2	distil-medium.en	distil-large-v2
TED-LIUM	6.4	5.6	5.8	4.3	4.4	3.8	**3.7**
Earnings 21	17.5	14.5	15.1	12.3	11.8	11.6	**11.2**
Earnings 22	24.1	19.4	20.6	15.6	**15.1**	16.3	**15.1**
Meanwhile	16.4	13.4	8.7	7.9	**6.3**	8.9	7.8
Rev 16	17.4	15.4	14.5	13.2	13.6	13.0	**12.2**
ID Average	6.4	5.6	5.8	4.3	4.4	3.8	**3.7**
OOD Average	18.9	15.7	14.7	12.3	11.7	12.4	**11.6**
Average	16.4	13.7	12.9	10.7	10.2	10.7	**10.0**

Table 18: Per-dataset WER scores for the sequential and chunked long-form transcription algorithms. The macro-average WER scores are shown for the one ID dataset, four OOD datasets, and an overall average over all five test sets.

Dataset	large-v2 sequential	large-v2 chunked	distil-large-v2 chunked
TED-LIUM	4.0	4.4	**3.7**
Earnings 21	**10.7**	11.8	11.2
Earnings 22	**14.0**	15.1	15.1
Meanwhile	**5.2**	6.3	7.8
Rev 16	**11.7**	13.6	12.2
ID Average	4.0	4.4	**3.7**
OOD Average	**10.4**	11.7	11.6
Average	**9.1**	10.2	10.0

Table 19: Per-dataset repeated 5-gram word duplicates (5-Dup.), insertion error rate (IER), substitution error rate (SER), deletion error rate (DER) and word error rate (WER) for the five long-form datasets. An average is shown for the ID dataset (TED-LIUM), the four OOD datasets, and an overall average.

Dataset	Metric	wav2vec2-large-960h	tiny.en	base.en	small.en	medium.en	large-v2	distil-medium.en	distil-large-v2
TED-LIUM	5-Dup.	**157**	522	557	549	452	542	283	270
	IER	1.7	2.1	2.1	1.8	1.4	1.8	0.6	**0.5**
	SER	6.0	2.2	1.6	1.2	1.0	**0.9**	1.4	1.3
	DER	1.9	2.2	1.9	2.7	2.0	**1.8**	**1.8**	**1.8**
	WER	9.6	6.4	5.6	5.8	4.3	4.4	3.8	**3.7**
Earnings-21	5-Dup.	**7938**	19294	19629	20611	21014	21559	16912	16797
	IER	5.2	4.1	3.2	3.0	3.0	3.0	2.0	**1.7**
	SER	20.9	8.2	6.0	4.6	4.0	**3.9**	5.0	4.7
	DER	4.3	5.3	5.3	7.5	5.3	4.9	**4.5**	4.7
	WER	30.4	17.5	14.5	15.1	12.3	11.8	11.6	**11.2**
Earnings-22	5-Dup.	**20869**	65599	63041	77122	64977	65419	52475	50949
	IER	8.5	6.5	4.8	5.3	3.8	3.9	3.7	**3.0**
	SER	26.8	11.6	8.5	6.9	5.9	**5.5**	7.3	6.7
	DER	4.8	6.0	6.0	8.4	6.0	5.7	**5.3**	5.4
	WER	40.1	24.1	19.4	20.6	15.6	**15.1**	16.3	**15.1**
Meanwhile	5-Dup.	**858**	1379	1406	1292	1485	1464	1236	1225
	IER	1.5	5.7	5.2	1.4	3.6	3.0	1.4	**1.0**
	SER	11.9	9.1	6.7	4.2	3.2	**2.4**	5.5	5.4
	DER	3.0	1.6	1.5	3.1	1.0	**0.9**	2.1	1.4
	WER	16.4	16.4	13.4	8.7	7.9	**6.3**	8.9	7.8
Rev 16	5-Dup.	**2220**	6981	6800	6483	6719	6724	5047	5040
	IER	4.2	4.2	3.8	3.3	3.4	3.5	2.8	**2.7**
	SER	16.1	6.8	5.3	4.2	3.8	**3.7**	4.6	4.2
	DER	6.1	6.4	6.3	7.0	6.0	6.4	5.6	**5.3**
	WER	26.4	17.4	15.4	14.5	13.2	13.6	13.0	**12.2**
ID Average	5-Dup.	**157**	587	671	548	574	752	281	270
	IER	1.7	2.4	2.6	1.9	1.9	2.7	0.6	**0.5**
	SER	6.0	2.1	1.6	1.1	1.0	**0.9**	1.4	1.3
	DER	1.9	2.2	1.9	2.0	2.0	**1.7**	1.8	1.8
	WER	9.6	6.8	6.0	4.9	4.8	5.3	3.8	**3.7**
OOD Average	5-Dup.	**7971**	23313	22719	26377	23549	23792	18918	18503
	IER	4.8	5.1	4.3	3.3	3.5	3.3	2.5	**2.1**
	SER	18.9	8.9	6.6	5.0	4.2	**3.9**	5.6	5.3
	DER	4.6	4.8	4.8	6.5	4.6	4.5	4.4	**4.2**
	WER	28.3	18.9	15.7	14.7	12.3	11.7	12.4	**11.6**
Average	5-Dup.	**6408**	18755	18287	21211	18929	19142	15191	14856
	IER	4.2	4.5	3.8	3.0	3.0	3.0	2.1	**1.8**
	SER	16.4	7.6	5.6	4.2	3.6	**3.3**	4.8	4.5
	DER	4.0	4.3	4.2	5.7	4.1	4.0	3.8	**3.7**
	WER	24.6	16.4	13.7	12.9	10.7	10.2	10.7	**10.0**

D ADDITIONAL ANALYSIS

D.1 EARLY EXIT

Early exit is a paradigm for dynamically controlling the number of decoder layers used at inference time. It is based on the reasoning that the same amount of computation may not be required for every input to achieve adequate performance, depending on whether the input is easy or hard.

Instead of making a prediction based on the hidden-representation of the *final* decoder layer, early exiting makes a prediction based on some *intermediate* layer. For each decoder layer l, we compute a confidence score $c_i[l]$ for the i-th token. We also define an early-exit threshold $\alpha_i[l]$. If our confidence score exceeds this threshold ($c_i[l] > \alpha_i[l]$), we exit early and greedily predict the most probably token. Otherwise, we continue to the next layer and repeat.

Confident Adaptive Language Modeling (CALM) (Schuster et al., 2022) proposes using a softmax difference as the confidence score. The decoder hidden-state for the l-th layer $\boldsymbol{d}_i^l$ is mapped to the logit space using the word-embedding matrix $\boldsymbol{W}$. We then take the softmax of these logits to get the token probabilities from the i-th decoder layer:

$$P\left(y_i|\boldsymbol{y}_{<i}, \boldsymbol{H}_{1:M}, \boldsymbol{d}_i^l\right) = \text{softmax}\left(\boldsymbol{W}\boldsymbol{d}_i^l\right) \tag{8}$$

The confidence score $c_i[l]$ is defined as the difference between the top-2 most probable predictions. If this difference is greater than the threshold $\alpha_i[l]$, the model is confident of its predictions, and we can terminate decoding early.

To gauge how many decoder layers can be skipped with early exit, we benchmarked the performance of the Whisper medium.en model on 100 samples form the LibriSpeech test-clean dataset. As the dataset with the lowest WER performance on short-form evaluation (see Table 16), it provides an upper-bound for the number of decoder layers that can be skipped, since the model should be most confident. We attempted setting the early-exit threshold automatically using the textual consistency formulation from CALM, which guarantees that the model will perform to within a certain tolerance of the full model with specified probability, but found it skipped close to zero layers for almost all examples. Instead, we swept over a set of values of the threshold, recording the WER performance and number of decoder layers used.

Table 20 shows the average number of decoder layers utilised by the medium.en model as the early-exit threshold is reduced. The medium.en model has a total of 24 decoder layers, of which the last 3 are skipped almost immediately. However, the WER penalty is significant even for just a 3-layer reduction. As we reduce the threshold, the number of layers skipped does not reduce significantly, but the WER penalty continues to inflate. Setting the threshold to 0.9750 results in an average of 3 skipped decoder layers, yielding an inference speed-up of 1.1 times. However, it also causes an increase in WER from 2.3% to 3.4% This suggests that there is high-utilisation of the first 21 decoder layers in the pre-trained Whisper model, and that the final 3 layers are necessary for ensuring high transcription accuracy. We leave finding effective early exit schemes for Seq2Seq ASR models as future work.

D.2 DISTILLATION OBJECTIVE

The knowledge distillation (KD) objective proposed in Section 4.1 is a weighted average of the Kullback-Leibler (KL) divergence and pseudo-label (PL) terms:

$$\mathcal{L}_{KD} = \alpha_{KL}\mathcal{L}_{KL} + \alpha_{PL}\mathcal{L}_{PL} \tag{9}$$

The typical setting in layer-based compression is that the dimensionality of the student model matches that of the teacher model. This means the student and teacher layers output the same shape of hidden-states. Thus, we can introduce a mean-square error (MSE) term to encourage the student's hidden layer outputs to match those of the teacher:

Table 20: **Early-exit performance.** WER on 100 examples from the LibriSpeech test-clean dataset as the early-exit threshold is varied. The latency results are computed relative to the medium.en model with full utilisation of the 24 decoder layers.

Threshold	Avg. Dec Layers	Rel. Latency	WER
1.0000	24.0	1.0	2.3
0.9875	21.2	1.1	2.8
0.9750	21.0	1.1	3.4
0.9625	20.8	1.1	3.5
0.9500	20.7	1.2	3.6
0.9375	20.6	1.2	3.7
0.9250	20.5	1.2	4.3

$$\mathcal{L}_{MSE} = \sum_{i=1}^{N} \sum_{l=1}^{L'} MSE\left(\boldsymbol{H}_l^S, \boldsymbol{H}_{\phi(l)}^T\right) \tag{10}$$

where $\boldsymbol{H}_l^S$ is the hidden-state output from the l-th layer of the student model S, $\phi(l)$ maps the l-th student layer to the corresponding teacher layer it is trained to emulate, and $\boldsymbol{H}_{\phi(l)}^T$ is the hidden-state output from layer $\phi(l)$ of the teacher model T. The mapping ϕ follows the settings from Shleifer & Rush (2020), where it is selected such that each decoder layer is trained to behave like maximally spaced teacher layers. For example, given a 2-layer student model initialised from a 32-layer teacher model, we choose pairings in ϕ such that each student decoder layer is taught to behave like 16 teacher layers. Thus, student layer 1's hidden-states are paired to teacher layer 16, and student layer 2's hidden-states paired to teacher layer 32:

$$\phi(l) = \begin{cases} 16 & \text{if } l = 1 \\ 32 & \text{if } l = 2 \end{cases} \tag{11}$$

A more general KD training objective is then a weighted sum of the KL, PL and MSE terms:

$$\mathcal{L}_{KD} = \alpha_{KL}\mathcal{L}_{KL} + \alpha_{PL}\mathcal{L}_{PL} + \alpha_{MSE}\mathcal{L}_{MSE} \tag{12}$$

where α_{KL}, α_{PL} and α_{MSE} are scalar weights for the KL, PL and MSE loss terms respectively. Following (Shleifer & Rush, 2020), we set $\alpha_{KL} = 0.8$ and $\alpha_{PL} = 1.0$, and tune the value of α_{MSE} on our validation set.

To quantify the performance gain obtained by incorporating each KD term, we train distil-large-v2 checkpoints for 10,000 training steps (two epochs) on a three combinations of KD objectives: (i) PL, (ii) PL + KD, and (iii) PL + KD + MSE. Training on PL alone is equivalent to shrink and fine-tune (SFT), but with the ground truth labels replaced by the PL generated ones. In all cases, Whisper-generated pseudo-labels are used as the ground truth labels during training.

Table 21 displays the average WER across the 11 short-form in-distribution (ID) validation sets and the three out-of-distribution (OOD) validation sets for each KD combination. The addition of the KL-divergence term yields an OOD word error rate (WER) that is 0.3% absolute lower compared to just PL. This suggests that the additional information transferred from the teacher to the student during KD is beneficial over training on PL alone. Incorporating the MSE loss term had a negligible effect on the average WER performance of the distilled model. This indicates that there is sufficient training signal from the PL and KL loss terms. The MSE loss requires that the hidden-states for each layer are recorded and kept in memory. This added a significant overhead when training the model in JAX, which resulted in a decrease to the maximum possible batch size. By only using the PL and KL objectives and training with a higher throughput, we achieved better results within a specified time interval compared to using the MSE loss, and thus opted for this configuration for our

Table 21: **Impact of the distillation objective.** Average WER of the distil-large-v2 checkpoint over the 11 ID and three OOD validation sets for the three possible training objectives: pseudo-labels (PL), KL-divergence (KL) and mean-square error (MSE).

Objective	Avg. ID WER	Avg. OOD WER	Avg. WER
PL	12.8	7.7	11.6
PL + KL	12.6	7.4	11.4
PL + KL + MSE	12.6	7.3	11.4

Table 22: **Impact of speculative decoding with batching.** Relative latency of the medium.en and large-v2 models using either the tiny Whisper or Distil-Whisper assistant models. The assistant models used are shown below the main ones. The relative latency is computed relative to the large-v2 model without speculative decoding for each batch size.

Model	Params / M	Batch Size		
		1	4	16
medium.en	769	1.4	1.3	1.5
w\ tiny.en	808	2.7	1.8	1.2
w\ distil-medium.en	856	3.3	2.2	1.3
large-v2	1550	1.0	1.0	1.0
w\ tiny	1589	2.1	1.3	0.8
w\ distil-large-v2	1669	2.0	1.3	0.8

experiments. Therefore, our final KD training objective is a weighted sum of the PL and KL terms only.

D.3 Batched Speculative Decoding

Table 22 reports the relative latency of the medium.en and large-v2 models both with and without speculative decoding at various batch sizes. For a batch size of 1, speculative decoding with the distil-large-v2 assistant yields a 2.0 times increase to inference speed over the large-v2 alone. This speed-up is comparable to using the tiny model as the assistant. For the medium.en model, using distil-medium.en as the assistant provides a 2.4 times speed-up. This outperforms the tiny.en assistant, which is only 2.0 times faster. A similar trend holds for a batch size of 4, albeit with smaller improvements to relative latency. For a batch size of 16, speculative decoding performs worse than using the main model by itself. For batched speculative decoding, all candidate tokens in across the batch must match the validation tokens for the tokens to be accepted. If any token in the batch at a given position does not agree, all candidate tokens that precede the position are discarded. Consequently, speculative decoding favours lower batch sizes, where it provides significant latency improvements while ensuring the same outputs as the original model.

D.4 Strategies for Reliable Long-form Transcription

Transcribing long-form audio relies on the accurate prediction of multiple chunks of audio in parallel. Since long-form audio typically contains instances of long pauses between spoken utterances, the Whisper model has a higher propensity to hallucinate compared to short-form audio. To combat this, the hyper-parameters of the chunked long-form transcription algorithm can be optimised to help avoid failure cases. These tuned hyper-parameters are applied in the long-form transcription results reported in Section 8.2.

Table 23 shows the WER performance on the long-form TED-LIUM validation set as the chunk length of the audio segments is decreased for the large-v2 and distil-large-v2 models. For Whisper, a chunk length of 30-seconds is optimal, whereas Distil-Whisper performs best with a chunk length of 15-seconds, giving the best overall performance on the validation set with 4.1% WER. Whisper

Table 23: **Effect of chunk length on the chunked long-form algorithm.** WER performance on the long-form TED-LIUM validation set as the chunk length of the long-form transcription algorithm is reduced.

Chunk Length / s	large-v2	distil-large-v2
30	4.8	7.4
25	5.3	5.7
20	6.5	5.0
15	6.5	4.2
10	10.0	4.3

is pre-trained on 30-seconds audio samples, whereas the mean sample length in the Distil-Whisper training set is 7.1-seconds (see Appendix A.1 for details). This suggests that the chunk length should be selected based on the distribution of audio data the model is trained on.

It is worth noting that the long-form WERs of Whisper and Distil-Whisper on the TED-LIUM validation set are 2.4% and 3.0% WER lower than their short-form performance on the same dataset. This performance gain demonstrates that the long-form algorithm used by Distil-Whisper is an effective approach for transcribing long audio files with strong WER performance.

D.5 Flash Attention 2

Flash Attention (FA) (Dao et al., 2022) addresses the slow and memory-intensive nature of transformers models on long sequences by proposing an IO-aware exact attention algorithm. It uses tiling to minimise memory reads/writes between GPU high bandwidth memory (HBM) and GPU on-chip memory (SRAM), resulting in faster inference and the ability to handle longer sequence lengths with improved model performance. Flash Attention 2 (FA2) (Dao, 2023) further refines this approach, optimising GPU work partitioning to achieve even greater computational efficiency.

To demonstrate the effect of FA2 on the Whisper and Distil-Whisper models, we benchmark the inference speed over 256 examples from each of the four OOD test sets. We report the latency as the real-time factor (RTF), defined as the ratio of the inference time to the audio duration. In order to generalise to multiple hardware, we report the inference time on a 40GB A100 GPU, which is typically used in industry, as well as a 16GB T4 GPU, a typical consumer-grade GPU. The results serve as look-up tables for practitioners wishing to determine the best trade-off between WER performance and latency for various batch sizes and hardware.

Table 24 reports the RTF for batch sizes 1, 4 and 16 on a 40GB A100 GPU. For the base attention implementation, distil-large-v2 is faster than base.en at batch sizes 1 and 4, and marginally slower at batch size 16. It remains faster than small.en at batch size 16, with an average OOD WER that is 0.7% lower. For all batch sizes, distil-large-v2 is at least 3.3 times faster than large-v2, while performing to within 1% WER. This highlights that distil-large-v2 can be used as a drop-in replacement for base.en at low batch sizes and small.en at higher ones, with 3.3% and 0.7% WER improvements respectively. Similarly, distil-medium.en is faster than tiny.en at batch sizes 1 and 4, and faster than base.en at batch size 16, while performing 3.2% better on OOD test data.

Incorporating FA2 benefits Distil-Whisper more than Whisper at higher batch sizes: the distil-large-v2 model is 31% faster with FA2 at batch size 16, compared to 22% for large-v2. Furthermore, the distil-medium.en checkpoint is as fast as tiny.en at batch size 16, with an average OOD WER that is 7.8% lower. FA2 has a greater improvement on the encoder than the decoder, where the memory is re-allocated at each decoding step. Since Distil-Whisper consists of 32 encoder layers and only 2 decoder layers, it improves the inference time significantly more than Whisper, which has 32 encoder and 32 decoder layers. This suggests that FA2 should always be incorporated for Distil-Whisper when operating at higher batch sizes. Using a static key/value cache would result in a more significant speed-up to the inference time of the decoder. We leave this as future works.

Table 25 reports the RTF on a 16GB T4 GPU. Since FA2 is not currently supported on T4, we report the RTF using the original FA implementation. For the base attention implementation, distil-large-v2 is faster than small.en at batch size 1. However, the distilled models follow the trend of medium.en

Table 24: Real time factor (RFT) with and without FA2 for batch sizes 1, 4 and 16. Inference speed is measured on a 40GB A100 GPU with PyTorch 2.0. RTF is expressed in 10^{-3}.

Model	Avg. OOD WER	Base			Flash Attention 2		
		1	4	16	1	4	16
tiny.en	18.9	22.7	8.7	**3.1**	21.4	8.3	**2.9**
base.en	14.3	27.9	11.7	4.6	30.5	10.7	3.4
small.en	10.8	59.1	22.3	8.4	50.7	18.3	6.3
medium.en	9.5	99.4	43.1	15.1	89.3	37.1	11.2
large-v2	**9.1**	137.2	54.9	21.3	121.9	48.4	16.6
distil-medium.en	11.1	**19.2**	**7.5**	4.3	**20.4**	**7.1**	**2.9**
distil-large-v2	10.1	25.1	9.9	6.5	24.2	8.9	4.4

Table 25: Real time factor (RFT) with and without FA for batch sizes 1, 4 and 16. Inference speed is measured on a 16GB T4 GPU with PyTorch 2.0. RTF is expressed in 10^{-3}.

Model	Avg. OOD WER	Base			Flash Attention		
		1	4	16	1	4	16
tiny.en	18.9	**20.0**	**7.2**	**2.6**	21.2	**6.9**	**2.5**
base.en	14.3	26.1	9.5	4.2	26.1	9.6	3.7
small.en	10.8	48.3	19.0	9.9	42.6	16.8	8.0
medium.en	9.5	89.9	44.6	26.5	66.2	33.8	18.5
large-v2	**9.1**	129.0	74.5	47.5	100.6	52.0	33.8
distil-medium.en	11.1	23.0	18.1	17.2	**19.0**	12.1	10.4
distil-large-v2	10.1	38.0	31.8	31.2	27.4	20.8	20.1

and large-v2 at higher batch sizes, where the percentage decreases to RTF are much lower than those of the smaller pre-trained Whisper checkpoints, such as base.en and small.en. At batch size 4 and 16, distil-large-v2 is slower than small.en. The distil-medium.en model is faster than small.en at batch size 4, but slower at 16.

This performance pattern can be attributed to the Distil-Whisper architecture: since we copy the entire encoder from the original Whisper medium.en and large-v2 models, the distil-medium.en and distil-large-v2 models require more memory than small.en, especially at higher batch sizes. As the memory on a T4 GPU increases, the throughput saturates, resulting in diminishing RTF benefits at batch sizes of 4 and 16. This trend is also observed for the medium.en and large-v2 Whisper checkpoints.

The latency improvement obtained using FA is significant for both Whisper and Distil-Whisper. At batch size 1, distil-large-v2 is comparable to base.en, while distil-medium.en is faster than tiny.en. However, the memory savings are not enough to offset the effects of the T4 GPU at higher batch sizes; distil-large-v2 is slower than small.en at batch size 4 and 16, and distil-medium.en slower than base.en.

Overall, a T4 GPU may be adequate for operating Whisper and Distil-Whisper models at a batch size of 1. For batch sizes beyond this, there is a notable performance stagnation on a T4, and higher memory A100 GPUs are preferential.