

---

# RETHINKING EVALUATION IN ASR: ARE OUR MODELS ROBUST ENOUGH?

---

**Tatiana Likhomanenko\***  
Facebook, Menlo Park  
antares@fb.com

**Qiantong Xu\***  
Facebook, Menlo Park  
qiantong@fb.com

**Vineel Pratap\***  
Facebook, Menlo Park  
vineelkpratap@fb.com

**Paden Tomasello**  
Facebook, Menlo Park  
padentomasello@fb.com

**Jacob Kahn**  
Facebook, Menlo Park  
jacobkahn@fb.com

**Gilad Avidov**  
Facebook, Menlo Park  
avidov@fb.com

**Ronan Collobert**  
Facebook, Menlo Park  
locronan@fb.com

**Gabriel Synnaeve**  
Facebook, Paris  
gab@fb.com

May 4, 2021

## ABSTRACT

Is pushing numbers on a single benchmark valuable in automatic speech recognition? Research results in acoustic modeling are typically evaluated based on performance on a single dataset. While the research community has coalesced around various benchmarks, we set out to understand generalization performance in acoustic modeling across datasets – in particular, if models trained on a single dataset transfer to other (possibly out-of-domain) datasets. We show that, in general, reverberative and additive noise augmentation improves generalization performance across domains. Further, we demonstrate that when a large enough set of benchmarks is used, average word error rate (WER) performance over them provides a good proxy for performance on real-world noisy data. Finally, we show that training a single acoustic model on the most widely-used datasets – combined – reaches competitive performance on both research and real-world benchmarks.

## 1 Introduction

Progress in automatic speech recognition (ASR) is measured on the validation and test sets of standard datasets. However, most acoustic models (AMs) are often developed and tuned on a single dataset and transfer poorly to other datasets. Moreover, most large standard benchmarks have similar domains and recording conditions, often with little background noise or little reverberation. These factors lead to siloed ASR research. Benchmarks in noisy conditions exist (e.g. [Barker et al., 2018, Maciejewski et al., 2020]), but are limited in training set size. A unified benchmark comprised of conversational, oratory, and read speech with varied recording conditions and noise would certainly serve the research community well; here, however, we study how the currently-popular public benchmarks can be used to gauge model generalization performance.

Our approach constructs a validation procedure – using only public datasets – that is a better predictor of overall and domain transfer performance than datasets taken in isolation. We train the same state-of-the-art model architecture on different benchmarks pushing for best performance on each benchmark separately. We also jointly train a model on all datasets. Given the transfer performance on test sets, we can ascertain which test sets are good proxies for transfer performance as well as which training sets can produce the best-performing models. Additionally, we train models with additive noise of signal noise ratios (SNR) and evaluate performance on the aforementioned validation sets. This informs us on the robustness of various datasets in transfer and which test sets are the best predictors of ASR performance in others. Finally, we look at the performance, in transfer only, on our in-house ASR datasets. This informs us about which sets of test sets should be used if one wants to transfer to a wide range of conditions of speech.

---

\*Equal contribution.

## 2 Related Work

Previous works that study transfer in ASR include [Ghahremani et al., 2017] that studied transferring varying number of layers trained out-of-domain, from SwitchBoard to AMI-IHM or from LibriSpeech to AMI-IHM. In this paper as in ours, a joint model trained on multiple out-of-domain datasets exhibits better transfer. In the context of the Arabic MGB-3 challenge, Manohar et al. [2017] transferred AMs trained on broadcast TV to Youtube videos, with a different setting than here as the training transcriptions were noisily labeled. Distillation was used to improved transfer in [Asami et al., 2017], where the soft-target part of the distillation loss may help with regularization. For another kind of transfer in [Kunze et al., 2017], the authors transferred LibriSpeech trained wav2letter [Collobert et al., 2016] models to German by fine-tuning them on German, with better performance than training from scratch. Very recently, Szymański et al. [2020] point out some limitations of current ASR benchmarks, and propose guidelines to create multi-domain datasets. Finally, while DeepSpeech 2 [Amodei et al., 2016] did not focus their study on transfer, we train a single AM on multiple datasets at once, as they did.

We also explore how training with additive noise helps transfer (on clean and on noisy conditions). Some of the first studies of noise robust ASR with deep networks include [Vinyals et al., 2012, Seltzer et al., 2013] which respectively trained RNNs on Aurora-2 and DNNs on Aurora-4 (a classic noise-robust ASR benchmark). The first looks at the performance in transfer of an RNN-based acoustic model trained on clean speech only, while the second investigates different regimes of noise-aware training for DNNs and which are most beneficial. A top entry (with models ensembling) from Chime-4 [Menne et al., 2016] compares different beamforming approaches for far-field ASR. More recently, Chime-5 [Barker et al., 2018] established a benchmark for far field ASR with background noise that is labeled, and WHAMR! [Maciejewski et al., 2020] is another benchmark with additive noise and reverberant speech synthetically mixed out of good quality noise samples and simulated room impulse responses (RIRs). On this topic, a paper [Ko et al., 2017] showed that – when done properly – we can use simulated RIRs, by comparing their influence to real room impulse responses in far-field ASR performance. Finally, we do not study more advanced data augmentations than additive noise and reverberation, but their impact on transfer is an avenue for future research. For instance, the authors of [Sun et al., 2018] perform adversarial data augmentation through a Fast Gradient Sign Method attack on the current model’s parameters, which leads to consistent gains on Chime-4 and Aurora-4.

## 3 Domain Transfer

In order to study transfer across datasets and conditions, we do a systematic analysis. In all our experiments, we use a single Transformer-based AM architecture with 270M parameters, to make our results comparable across the board. We train multiple single-dataset baselines as well as one joint model trained on all datasets at once. We then evaluate this set of models on all the validation and test sets, to measure how each “in-domain” model transfers to “out-of-domain” datasets. From this, we analyze which datasets suffer more acutely from “domain overfitting.” Evidently, it is difficult to separate the “in-domainness” and size of a dataset; e.g., we cannot directly compare results on WSJ (80h) to ones on LibriSpeech (960h). We also fine-tune our joint model with additive noise and artificial reverberation and measure how it boosts transfer (joint+noise model). We also fine-tune our joint and joint+noise models on the transfer dataset with 10min, 1h, 10h, and 100h of in-domain data. Finally, we examine how our models transfer to real data and in the process observe that public validation and test sets performance is predictive of the transfer performance of a model to real data.

## 4 Experiments

### 4.1 Datasets

To measure domain transfer, we restrict experiments to use only datasets in English, for which there exist several commonly-used and publicly available datasets with hundreds hours of transcribed audio. Validation sets from each dataset are used to optimize model configurations and to perform all hyper-parameter tuning, while test sets are used for final evaluation only.

**LibriSpeech (LS)** [Panayotov et al., 2015] consists of read speech from audiobook recordings. We use standard split of train, validation (*dev-clean*, *dev-other*) and test sets (*test-clean*, *test-other*).

**SwitchBoard & Fisher (SB+FSH)** consists of conversational telephone speech. To create a training set, we combine Switchboard [Godfrey and Holliman, 1993] and Fisher [Cieri et al., 2004, 2005a,]. We use RT-03S [Fiscus et al., 2007] as the validation set; test sets are the Hub5 Eval2000 [LDC et al., 2002] data with two subsets, SwitchBoard (SB) and CallHome (CH). For the data processing and evaluation, we follow the recipe provided by Kaldi [Povey et al., 2011].

Table 1: Statistics on datasets: sampling frequency, duration (in hours), and speech type.

Data	kHz	Train (h)	Valid (h)	Test (h)	Speech
WSJ	16	81.5	1.1	0.7	read
TL	16	452	1.6	2.6	oratory
CV	48	693	27.1	25.8	read
LS	16	960	5.1+5.4	5.4+5.4	read
SB+FSH	8	300+2k	6.3	1.7+2.1	conversational
RV	16	5k	14.4	18.8+19.5+37.2	diverse

Table 2: Statistics on datasets: mean sample duration (in seconds) and mean sample transcription length (in words).

Data	Train $\mu \pm \sigma$ (s)	Valid $\mu \pm \sigma$ (s)	Test $\mu \pm \sigma$ (s)	Train $\mu \pm \sigma$ (wrd)	Valid $\mu \pm \sigma$ (wrd)	Test $\mu \pm \sigma$ (wrd)
WSJ	7.8 $\pm$ 2.9	7.8 $\pm$ 2.9	7.6 $\pm$ 2.5	17 $\pm$ 7	16 $\pm$ 7	17 $\pm$ 6
TL	6 $\pm$ 3	11.3 $\pm$ 5.7	8.1 $\pm$ 4.3	17 $\pm$ 10	35 $\pm$ 20	24 $\pm$ 15
CV	5.7 $\pm$ 1.6	6.1 $\pm$ 1.8	5.8 $\pm$ 2.6	10 $\pm$ 3	10 $\pm$ 3	9 $\pm$ 3
LS	12.3 $\pm$ 3.8	6.8 $\pm$ 4.5	7 $\pm$ 4.8	33 $\pm$ 12	19 $\pm$ 13	19 $\pm$ 13
SB+FSH	3.7 $\pm$ 3.2	4 $\pm$ 3.1	2.1 $\pm$ 1.7	11 $\pm$ 12	12 $\pm$ 12	8 $\pm$ 8
RV	8.5 $\pm$ 1.9	11.6 $\pm$ 2.8	11.6 $\pm$ 2.7	21 $\pm$ 10	25 $\pm$ 13	29 $\pm$ 12

**Wall Street Journal (WSJ)** [Garofolo et al., 1993, LDC and NIST, 1994, Woodland et al., 1994]. We consider the standard subsets *si284*, *nov93dev* and *nov92* for training, validation and test, respectively. We remove any punctuation tokens from *si284* transcriptions when used for training.

**Mozilla Common Voice (CV)** project [Ardila et al., 2020]. The CV dataset consists of transcribed audio in various languages where speakers record text from Wikipedia. Anyone can submit recorded contributions; as a result, the dataset has a large variation in quality and speakers. We use the English dataset<sup>2</sup>, where data splits are provided therein.

**TED-LIUM v3 (TL)** [Hernandez et al., 2018] is based on TED conference videos. We use the last edition of the training set from this dataset (v3), for which the validation and test sets are kept consistent (and thus numbers are comparable) with the earlier releases. We follow the Kaldi recipe [Povey et al., 2011] for data preparation.

**Robust Video (RV)** is our in-house English video dataset, which are sampled from public social media videos and aggregated and deidentified before transcription. These videos contain a diverse range of speakers, accents, topics, and acoustic conditions making ASR difficult. The test sets are composed of *clean*, *noisy* and *extreme* with *extreme* being the most acoustically challenging subset among them. The validation set comprises of data from *noisy* and *extreme* subsets.

**CHiME-6** [Watanabe et al., 2020] is a noisy low resource dataset set, which contains around 40 hours of distant microphone conversational speech recognition in everyday home environments. We use this dataset only to evaluate robustness to noisy conditions of our models. Front end enhancement for development and test sets is done following the official recipe [Watanabe et al., 2020]: the guided source separation [Boeddeker et al., 2018] with 12 channels is used to enhance front end.

## 4.2 Unifying Audio

The datasets used in our work have different sample rate and varied input lengths as shown in Table 1 and 2. Since we require the same set of filterbanks for joint training across all datasets, we upsample/downsample each dataset to 16kHz and use this setup for training both baseline models on individual datasets as well as joint models. For all experiments we compute 80 log-mel spectrogram features for a 25ms sliding window, strided by 10ms. All features are normalized to have zero mean and unit variance per input sequence before feeding into the neural network.

On SB+FSH individual baseline, we span the log-mel filterbanks up to only 4kHz (unlike 8kHz for all other training setups) as any spectrogram features beyond 4kHz cannot be determined accurately [Shannon, 1949]. This can also be seen in Figure 1 which plots the distribution of mean normalized energy of filterbanks for different datasets with audio sampled at 16kHz and filterbanks span from 0-8kHz.

<sup>2</sup>June 22nd 2020’s snapshot: <https://tinyurl.com/cvjune2020>. Transcriptions contain upper-case and non-English characters and punctuation. To have similar transcription normalization as in other datasets, we normalize the text for all splits: lower-casing, Unicode normalization, removing punctuation and non-English tokens, and mapping common abbreviations (e.g. “mr.” to “mister”).

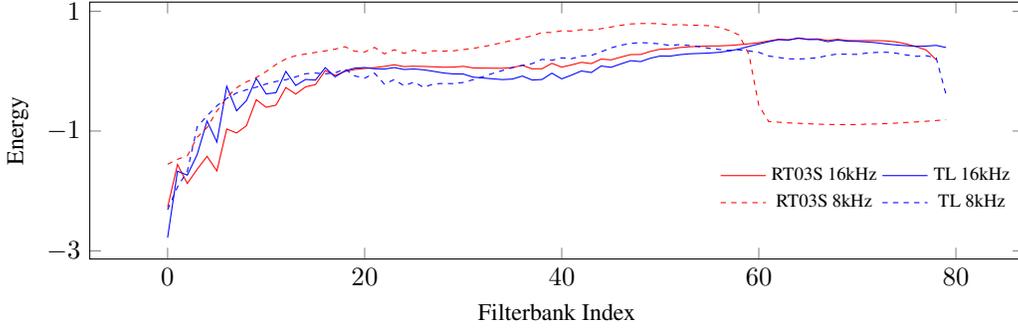


Figure 1: Distribution of the mean normalized energy of 80 filterbanks on some public validation sets we used, for 16kHz audio (dashed) and 8kHz audio (solid).

### 4.3 Language Model

We train a  $n$ -gram LM using KenLM toolkit [Heafield, 2011] and a Transformer LM similar to [Synnaeve et al., 2019] for each dataset independently using their in-domain LM training corpus. Specifically, we use the training transcriptions as LM corpus for domains like SB+FSH and RV; while for TL, both training transcriptions and the original LM corpus are combined together to train its LM. All the Transformer LMs share the similar architecture as [Baevski and Auli, 2019]’s Google Billion Words model: we use 8 attention heads; 8 (WSJ, CV and SB+FSH), 16 (TL) or 20 (LS) decoder layers with embedding, input and output dimensions of 512 (CV), 1024 (WSJ and SB+FSH) or 1280 (TL and LS); feed-forward layer dimension is set to 1024 ( CV), 2048 (WSJ and SB+FSH) or 6144 (TL and LS); dropout is 0.3 (WSJ, CV and SB+FSH), 0.15 (TL) or 0.1 (LS). Number of decoder layers, embedding dimensions as well as dropout were tuned on each dataset depending on the amount of training data.

We also train a 4-gram and a Transformer LMs on Common Crawl (CC) data [Wenzek et al., 2020]. Before any training we perform the following text normalization for CC data: splitting paragraphs into separate sentences, punctuation removal, mapping of common abbreviations, converting latin and roman numbers into the text. We keep a dictionary of 200k most-common words. For 4-gram training we prune all 3,4-grams appearing once and use only 10% of the CC data. The transformer LM also follows the [Baevski and Auli, 2019]’s Google Billion Words model and trained on all CC data: we use 8 attention heads and 16 decoder layers with embedding, input and output dimensions 1024 and feed-forward layer dimension 4096, dropout is set to 0.1. The perplexity of all LMs are shown in Table 3.

Table 3: Perplexity (including out-of-vocabulary words) of word-level LMs. We use 4-gram LMs for WSJ, LS, SB+FSH, and 5-gram LMs for TL, CV.

Data/Vocab	in-dom. $n$ -gram		in-dom. Transf.		CC 4-gram		CC Transf.	
	Valid	Test	Valid	Test	Valid	Test	Valid	Test
WSJ/162K	159	134	84	69	297	285	119	116
TL/200k	119	149	74	79	142	136	82	80
CV/168K	359	329	181	188	213	157	119	100
LS/200K	155/147	164/154	48/50	52/50	258/258	244/249	133/137	139/137
SB+FSH/64K	124	114/112	50	55/67	221	199/153	70	67/83
RV/200K	158	146	-	-	249	204	-	-

To integrate LMs with AMs, we use one-pass beam-search decoder from the wav2letter++ [Collobert et al., 2016] (lexicon-based with a  $n$ -gram LM) and an additional second-pass rescoring with a Transformer LM following [Synnaeve et al., 2019].

### 4.4 Baselines and Joint Acoustic Model

#### 4.4.1 Acoustic Model (AM)

All models are trained with Connectionist Temporal Classification [Graves et al., 2006] and the network architecture follows [Synnaeve et al., 2019]: the encoder of our AMs is composed of a convolutional frontend (1-D convolution with kernel-width 7 and stride 3 followed by GLU activation) followed by sinusoidal positional embedding and 36 4-

Table 4: WER of models evaluated on all datasets (downsampled to 16kHz) with a greedy decoding and *no LM* (top row), with *in-domain n-gram LM* beam-search decoding (middle row) and with additional second-pass rescoring by *in-domain Transformer LM* (below row). Joint models are also decoded with CC LM with either a single-pass (top row) or a two-pass (bottom row) decoding. State-of-the-art (SOTA) models are given from WSJ [Hadian et al., 2018], TED-LIUM [Zhou et al., 2020], LibriSpeech [Gulati et al., 2020], SwitchBoard & Fisher [Han et al., 2017]. The SOTA models are all decoded with in-domain LMs. The average is computed as average of averages for LibriSpeech’s validations/tests, and SwitchBoard’s tests (SB, CH) sets, so as not to weight them more heavily.

Train	WSJ		TL		CV		LS				SB+FSH			average	
	nov93	nov92	valid	test	valid	test	dev-c	test-c	dev-o	test-o	RT03S	SB	CH	valid	test
SOTA		2.8	5.1	5.6				1.9		3.9	8.0	5.0	9.1		
WSJ	13.5	11.7	48.8	42.1	72.2	78.0	32.4	32.7	53.8	54.0	78.3	69.6	84.6	51.2	50.4
	7.3	5.3	35.2	28.7	55.0	62.0	17.7	18.2	37.4	38.8	68.0	58.2	78.3	38.6	38.6
	5.7	4.1	33.9	26.9	53.5	60.7	14.6	15.2	35.2	36.7	66.8	57.4	77.5	36.9	37.0
TL	12.0	9.9	10.2	7.9	32.2	36.7	11.8	12.4	21.6	22.6	32.1	23.6	32.4	20.6	20.0
	7.6	5.8	7.9	6.5	23.9	28.1	7.5	8.4	15.3	16.4	27.3	19.3	27.7	15.6	15.3
	6.3	5.1	7.4	6.2	22.6	27.3	5.9	6.9	13.1	14.1	26.8	19.0	27.3	14.5	14.5
CV	12.8	9.7	68.5	47.0	12.0	15.4	35.5	36.3	37.0	39.1	49.2	46.9	45.7	35.8	31.2
	6.0	3.5	56.6	34.2	10.1	13.0	23.8	25.6	26.6	29.1	39.2	36.2	36.6	27.4	22.9
	5.5	3.2	55.3	32.4	10.0	12.8	21.9	23.7	24.4	26.9	37.8	35.0	35.5	26.3	21.8
LS-960	10.0	7.9	13.6	13.0	25.8	29.9	2.6	2.7	7.0	6.8	36.6	27.6	35.0	18.2	17.4
	4.7	3.5	9.0	9.7	18.6	22.3	2.0	2.5	5.2	5.5	28.3	20.0	27.8	12.8	12.7
	3.9	2.9	8.5	8.8	17.6	21.6	1.5	2.0	4.2	4.5	27.7	20.0	27.1	12.1	12.0
SB+FSH	10.9	9.5	15.1	12.4	49.4	50.7	14.0	14.4	27.4	28.6	12.0	6.9	11.4	21.6	20.6
	5.1	4.0	9.3	8.9	40.4	41.8	7.5	8.2	18.8	20.4	10.4	6.5	10.3	15.7	15.4
	4.2	3.4	8.6	8.0	38.9	40.5	5.4	6.2	16.3	18.1	10.4	6.5	10.3	14.6	14.5
Joint	3.0	2.0	6.1	5.7	11.1	13.2	2.5	2.5	6.0	5.9	10.7	5.8	9.7	7.0	6.6
	2.0	1.4	5.4	5.5	9.4	11.1	1.8	2.3	4.5	4.8	9.4	5.4	8.6	5.9	5.7
	1.7	1.3	5.0	4.7	9.2	10.9	1.4	2.0	3.7	4.1	9.4	5.4	8.6	5.6	5.4
Joint CC	2.8	2.0	5.6	5.1	8.1	9.4	2.9	2.9	5.3	5.3	9.6	5.4	8.7	6.0	5.5
	2.8	1.9	5.3	4.8	8.0	9.3	2.9	2.9	5.2	5.3	8.9	5.4	8.7	5.8	5.4
Joint+noise	3.0	2.2	6.3	5.8	11.2	13.3	2.5	2.6	6.1	6.0	10.8	6.3	10.3	7.1	6.8
	2.0	1.5	5.5	5.2	9.4	11.2	1.8	2.3	4.5	4.9	9.4	5.8	9.3	5.9	5.8
	1.9	1.3	5.1	4.9	9.3	11.0	1.4	2.0	3.7	4.2	9.4	5.8	9.3	5.7	5.6
Joint+noise CC	2.8	2.0	5.7	5.2	8.2	9.6	2.9	3.0	5.2	5.3	9.5	5.6	9.0	6.1	5.6

heads Transformer blocks [Vaswani et al., 2017]. To speed up training we don’t use any relative positional embedding inside Transformer blocks. The self-attention dimension is 768 and the feed-forward network (FFN) dimension is 3072 in each Transformer block. The output of the encoder is followed by a linear layer to the output classes.

We use dropout after the convolution layer. For all Transformer layers, we use dropout on the self-attention and on the FFN, and layer drop [Fan et al., 2020], dropping entire layers at the FFN level. Dropout and layer dropout values are tuned for each model separately. Token set for all AMs consists of 26 English alphabet letters, augmented with the apostrophe and a word boundary token. The popular approach with word-pieces as tokens set we found to be not suited as intersection between word-pieces constructed on every training set less than 50%. Thus the question what word-pieces set should be used for the joint model is still open. SpecAugment [Park et al., 2019] is used for data augmentation in training: there are two frequency masks, and ten time masks with maximum time mask ratio of  $p = 0.05$ ; frequency and time mask parameters are tuned separately for each model; time warping is not used. In the joint model, the maximum frequency bands masked by one frequency mask is 30, and the maximum frames masked by the time mask is 30, too. We use the Adagrad optimizer [Duchi et al., 2011] and decay learning rate by a factor of 2 each time the WER reaches a plateau on the validation sets. All experiments are implemented within flashlight<sup>3</sup> and wav2letter++ [Pratap et al., 2019]. All models are trained with dynamic batching (effective average batch size is 240s per GPU) and mixed-precision computations on 16 GPUs (Volta 32GB) for 1-3 days for single dataset baselines and 14 days for joint training.

<sup>3</sup><https://github.com/flashlight/flashlight>

Table 5: WER comparison on CHiME-6 dev and eval sets. Both HMM model results are obtained by decoding with a 3-gram LM, the second HMM model being the single best AM in the CHiME 2020 challenge on this track. All other results are with *a greedy decoding* without an LM. Both HMM models and the RNN-T models are trained on CHiME-6. All other models correspond to the ones in Table 4 and do not use CHiME-6 for training (direct transfer).

Model	Train	Dev	Eval
Official HMM [Watanabe et al., 2020]	CHiME-6	51.8	51.3
HMM [Medennikov et al., 2020]	CHiME-6	36.9	38.6
RNN-T [Andrusenko et al., 2020]	CHiME-6	49.0	-
Transformer (ours)	WSJ	97.5	95.8
	TL	61.7	70.5
	CV	81.2	77.9
	LS-960	74.4	79.6
	SB+FSH	55.9	67.0
	Joint	44.9	56.9
	Joint+noise	41.5	51.7

#### 4.4.2 Joint Model

We adopt the same AM architecture described above but with less regularization when training on the combination of all the datasets. We weight each sample equally, i.e. each sample from each dataset is fed into the model once in each epoch.

#### 4.4.3 Joint+noise Model

To further improve the robustness of our joint model, we fine-tune the model by using data augmentation on the training data. In our work, we use two popular audio data augmentation procedures: additive noise and reverberation. For additive noise, we randomly sample a audio clip from Audioset [Gemmeke et al., 2017] database. For each sample, we randomly sample an value between a chosen min SNR and max SNR values, and scale the noise accordingly before adding it to the input signal. Reverberation is done by convolving the input signal with a randomly sampled RIR. Similar to [Balam et al., 2020], we consider real and simulated room impulse responses (RIRs) from OpenSLR [Ko et al., 2017] and BUT ReverDB [Szöke et al., 2019]. We use 0 and 40 as min and max values for SNR and the probability of applying additive noise, reverberation augmentation is set to 0.4 and 0.2 respectively. We have chosen these settings based on the average performance of joint+noise model on all validation sets.

#### 4.4.4 Joint and Joint+noise Models Fine-tuning on RV

We fine-tune our best joint model and joint-noise model on small parts of RV training data, to see how they transfer to real-world noisy data. The subsets are randomly selected to have length 10 minutes, 1 hour, 10 hours and 100 hours. To reach the best performance on different amount of training data, we conduct a grid search over the following parameters: learning rate in (0.001, 0.005, 0.01), warm-up updates in (1000, 2000, 4000, 6000), and learning rate decreasing scheduler. The optimal learning rate and warm-up updates are 0.005 and 2000 for all configurations. Learning rate decreases every (20, 200, 2000, 5000) epochs for (10 min, 1hr, 10 hrs and 100hrs) settings. We didn't use any additive noise or reverberation in fine-tuning on RV. All the experiment results are listed in Table 7.

### 4.5 AM Transfer

In general, an AM trained in isolation on a single dataset performs poorly on other datasets, as shown in Table 4. The model trained on WSJ performs the worst (part of the reason could be the smaller amount of training data) for transfer, while other models transfer very well to WSJ. All models transfer poorly to CV and the CV model transfers poorly to other datasets, which may indicate that CV is very different from other benchmarks. From the results on LS, TL and SB+FSH there is a similarity between LS and TL (they transfer the best to each other). There is also a similarity in transfer between SB+FSH and TL benchmarks, however, LS and SB+FSH do not transfer well to each other. When training on all datasets at once, the joint model in Table 4 performs better or close to a single dataset training. This behaviour compared to results on a single dataset training indicates that i) datasets differ from each other and ii) a robust model scoring well on all these benchmarks exists.

Table 6: Direct transfer. WER comparison with a *greedy decoding* and with a *5-gram in-domain LM* and the *4-gram CC LM* beam-search decoding on RV validation and test data from videos. Except for the in-domain “RV” training and for models with “+finetune” (in-domain finetuning), all other models correspond to models in Table 4.

Train	LM	Valid	Test		
			clean	noisy	extreme
RV (5000h)	-	26.3	14.6	19.1	27.9
	in-dom.	22.9	12.3	16.2	24.1
	CC	22.9	12.1	16.1	24.2
WSJ (81.5h)	-	78.9	66.3	73.1	80.2
	in-dom.	68.4	54.0	61.9	69.8
	CC	69.3	54.1	62.5	70.9
TL (452h)	-	42.5	24.1	32.0	44.1
	in-dom.	36.0	19.1	26.3	37.1
	CC	36.1	19.0	26.3	37.5
CV (693h)	-	73.2	65.4	70.5	73.5
	in-dom.	62.5	51.4	57.9	63.2
	CC	63.0	51.8	58.5	64.0
LS-960 (960h)	-	48.0	28.5	37.6	50.3
	in-dom.	38.2	21.5	29.5	40.2
	CC	38.9	21.7	29.9	41.2
SB+FSH (2300h)	-	43.7	28.8	33.8	44.7
	in-dom.	40.3	24.6	29.9	41.3
	CC	40.4	24.5	30.0	41.5
Joint (4500h)	-	39.8	16.0	22.4	32.9
	in-dom.	28.3	13.5	19.3	29.6
	CC	28.7	13.6	19.5	30.0
Joint+noise (4500h)	-	30.7	15.6	21.5	32.7
	in-dom.	27.3	13.1	18.2	29.2
	CC	27.6	13.1	18.5	29.7

We also test our models from Table 4 on CHiME-6 dev and eval sets to analyse the noise adaptation, see Table 5. The best single-dataset models are TL and SB+FSH, probably due to the source of their data, like conversational and oratory speech with noisy conditions. Our joint model improves noise adaptation significantly over the best single-dataset models. Finally, the joint+noise model improves further results, stating that additive noise and reverberation augmentation training helps to improve noise robustness.

In Table 6, we report results of transfer, of those same models trained on public datasets, to our in-house RV dataset. We also report numbers from a baseline system that is trained in-domain on a corresponding training set of 5000h. As for other benchmarks, single dataset training transfers poorly to in-house data, however, the transfer quality varies a lot, having the best results from the TL model. At the same time our joint model, which performs well on each benchmark, transfers really well, stating that i) public datasets could be the good proxy of training data for real-world ASR, ii) improving average performance on public benchmarks leads to improving performance on real-world noisy data. Our joint+noise model further improves the robustness stating that additive noise and reverberation augmentations improve transfer to real-world noisy data.

In Table 7, we report fine-tuning results of our joint and joint+noise models on RV data with different amount of data: 10min, 1h, 10h and 100h. Fine-tuning with only 1h closes the gap with the RV baseline model for both joint and joint+noise models. Fine-tuning of the joint+noise model on extremely small amount of data, 10min and 1h, has substantially better performance than the fine-tuning of the joint model. Thus, pre-training with additive noise and reverberation is important in case of low resource of in-domain data. Fine-tuning with enough in-domain data, 10h or 100h, gives similar results for the joint and joint+noise models, thus we still can adapt to the noise in our in-domain data even without pre-training with additive noise and reverberation. Also fine-tuning of joint or joint+noise models on 10h or 100h data surpasses WER compared to the RV baseline model decoded both with in-domain LM and CC LM.

Table 7: Effect of in-domain finetuning. WER comparison with *a greedy decoding* and with *a 5-gram in-domain LM* and *the 4-gram CC LM* beam-search decoding on RV validation and test data from videos. Except for the in-domain “RV” training and for models with “+finetune”, all other models correspond to models in Table 4.

Train	LM	Valid	Test		
			clean	noisy	extreme
RV (5000h)	-	26.3	14.6	19.1	27.9
	in-dom.	22.9	12.3	16.2	24.1
	CC	22.9	12.1	16.1	24.2
Joint + finetune RV-10min	-	30.2	15.5	21.6	31.3
	in-dom.	26.1	12.7	18.0	27.3
	CC	26.4	12.7	18.1	27.6
Joint + finetune RV-1h	-	27.3	14.0	19.2	28.3
	in-dom.	24.0	11.8	16.3	24.9
	CC	24.3	11.9	16.6	25.4
Joint + finetune RV-10h	-	25.9	12.6	17.8	27.2
	in-dom.	22.9	10.8	15.3	24.2
	CC	22.9	10.8	15.4	24.4
Joint + finetune RV-100h	-	25.4	12.4	17.4	26.8
	in-dom.	22.5	10.6	15.0	23.8
	CC	22.5	10.6	15.1	24.0
Joint+noise + finetune RV-10min	-	29.3	14.9	20.8	30.4
	in-dom.	25.6	12.3	17.5	26.8
	CC	25.9	12.3	17.6	27.1
Joint+noise + finetune RV-1h	-	26.8	13.5	18.8	28.1
	in-dom.	23.5	11.3	16.0	24.8
	CC	24.0	11.5	16.3	25.3
Joint+noise + finetune RV-10h	-	25.9	12.7	17.9	27.3
	in-dom.	22.8	10.8	15.3	24.3
	CC	22.8	11.0	15.5	24.7
Joint+noise + finetune RV-100h	-	25.6	12.5	17.5	26.8
	in-dom.	22.6	10.6	14.9	23.8
	CC	22.7	10.6	15.1	24.0

#### 4.6 Transfer with LM

Single-dataset AMs get a boost in WER performance when decoding/rescoring with an in-domain LM, as shown in Table 4. These AMs perform however poorly in transfer domain conditions (see Tables 4 and 6). In contrast, the joint models transfers well to in-house RV data, when decoded with an in-domain LM (see Table 6). Decoding the joint models with the large generic CC LM leads to WER performance which is overall improved, on both public and in-house RV datasets, and close to the in-domain LM results.

#### 4.7 Predictors of transfer

We performed single variable linear regressions using data from Table 4: lines as datapoints, test set score columns as features, and labels being the same models’ performance in transfer on the average of RV test clean, noisy, and extreme, from Table 6. Across all datasets, and taken over all trained models, the best “single test set” predictor for out-of-domain performance on RV data is TL with an  $r^2 = 0.9$  (rejecting the null hypothesis with  $p < 0.001$ ), the worst single predictor being CV with  $r^2 = 0.2$  ( $p < 0.001$ ). We also performed multivariate regressions using all the test sets from Table 4 and only the results for the models decoded with  $n$ -gram LMs. This gives an overspecified problem (more variables: 7, than models: 6), so OLS gives a “perfect” (overfitted,  $r^2 = 1$ ) solution which weights test-other and Callhome *a bit negatively*. We repeat this regression with L1 regularization (Lasso, as proxy for L0 norm regularization) with different regularization coefficients. It yields a regression with  $r^2 \in (0.975, 0.999)$  (albeit

with only 6 datapoints) with only TL test set weighted significantly positively and WSJ’s nov92 and LS’s test-clean weighted at zero. We can conclude that the TL test set is the better predictor, and nov92 and test-clean are the poorest predictors of the performance in transfer on RV of our Transformer-based AMs decoded with  $n$ -grams. A larger study across AMs and LMs variants should provide a more robust conclusion.

## 5 Conclusion

We studied transfer across five public datasets, as well as transfer to out-of-domain, real-world audio data, for a single AM architecture based on Transformers using non-autoregressive CTC criterion and with  $n$ -gram and Transformer-based LMs for decoding. We showed that no single validation or test set from public datasets is sufficient to measure transfer to other public datasets or to real-world audio data. Our results suggests that ASR researchers interested in producing transferable AMs should report results on several public datasets, at very least including TED-LIUM (v3). Finally, we provided a recipe for a community-reproducible robust ASR model, which can be trained with a couple of public audio datasets, and language models trained on the Common Crawl dataset.

## References

- D. Amodei et al. Deep Speech 2: End-to-End Speech Recognition in English and Mandarin. In *ICML*, 2016.
- A. Andrusenko, A. Laptev, and I. Medennikov. Towards a competitive end-to-end speech recognition for chime-6 dinner party transcription. *arXiv preprint arXiv:2004.10799*, 2020.
- R. Ardila, M. Branson, K. Davis, M. Kohler, J. Meyer, M. Henretty, R. Morais, L. Saunders, F. Tyers, and G. Weber. Common voice: A massively-multilingual speech corpus. In *LREC*, 2020.
- T. Asami, R. Masumura, Y. Yamaguchi, H. Masataki, and Y. Aono. Domain adaptation of dnn acoustic models using knowledge distillation. In *ICASSP*, 2017.
- A. Baeveski and M. Auli. Adaptive input representations for neural language modeling. In *International Conference on Learning Representations*, 2019.
- J. Balam, J. Huang, V. Lavrukhin, S. Deng, S. Majumdar, and B. Ginsburg. Improving noise robustness of an end-to-end neural model for automatic speech recognition, 2020.
- J. Barker, S. Watanabe, E. Vincent, and J. Trmal. The fifth CHiME speech separation and recognition challenge: dataset, task and baselines. In *ICSA Speech*, 2018.
- C. Boeddeker, J. Heitkaemper, J. Schmalenstroeer, L. Drude, J. Heymann, and R. Haeb-Umbach. Front-end processing for the chime-5 dinner party scenario. In *CHiME5 Workshop, Hyderabad, India*, 2018.
- C. Cieri, D. Graff, O. Kimball, D. Miller, and K. Walker. Fisher english training speech parts 1 and 2 transcripts LDC200{4,5}T19. *Philadelphia: LDC*, 2004, 2005a.
- C. Cieri, D. Miller, and K. Walker. Fisher english training speech parts 1 and 2 LDC200{4,5}S13. *Philadelphia: LDC*, 2004, 2005b.
- R. Collobert, C. Puhersch, and G. Synnaeve. Wav2letter: an end-to-end convnet-based speech recognition system. *arXiv preprint arXiv:1609.03193*, 2016.
- J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12, 2011.
- A. Fan, E. Grave, and A. Joulin. Reducing transformer depth on demand with structured dropout. In *ICML*, 2020.
- J. G. Fiscus et al. 2003 nist rich transcription evaluation data LDC2007S10. *Web Download. Philadelphia: LDC*, 2007.
- J. Garofolo, D. Graff, D. Paul, and D. Pallett. CSR-I (WSJ0) complete LDC93S6A. *Web Download. Philadelphia: LDC*, 1993.
- J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780. IEEE, 2017.
- P. Ghahremani, V. Manohar, H. Hadian, D. Povey, and S. Khudanpur. Investigation of transfer learning for ASR using LF-MMI trained neural networks. In *ASRU*, 2017.
- J. Godfrey and E. Holliman. Switchboard-1 release 2 LDC97S62. *Philadelphia: LDC*, 1993.

- A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, 2006.
- A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020.
- H. Hadian, H. Sameti, D. Povey, and S. Khudanpur. End-to-end speech recognition using lattice-free MMI. In *Interspeech*, 2018.
- K. J. Han, A. Chandrashekar, J. Kim, and I. Lane. The CAPIO 2017 conversational speech recognition system. *arXiv preprint arXiv:1801.00059*, 2017.
- K. Heafield. KenLM: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*. Association for Computational Linguistics, 2011.
- F. Hernandez, V. Nguyen, S. Ghannay, N. Tomashenko, and Y. Estève. TED-LIUM 3: twice as much data and corpus repartition for experiments on speaker adaptation. In *SPECOM*, 2018.
- T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur. A study on data augmentation of reverberant speech for robust speech recognition. In *ICASSP*, 2017.
- J. Kunze, L. Kirsch, I. Kurenkov, A. Krug, J. Johannsmeier, and S. Stober. Transfer learning for speech recognition on a budget. In *ACL Workshop on Representation Learning for NLP*, 2017.
- LDC and M. I. G. NIST. CSR-II (WSJ1) complete LDC94S13A. *Web Download. Philadelphia: LDC*, 1994.
- LDC et al. 2000 hub5 english evaluation speech LDC2002S09 and transcripts LDC2002T43. *Web Download. Philadelphia: LDC*, 2002.
- M. Maciejewski, G. Wichern, E. McQuinn, and J. Le Roux. Whamr!: Noisy and reverberant single-channel speech separation. In *ICASSP*, 2020.
- V. Manohar, D. Povey, and S. Khudanpur. JHU Kaldi system for Arabic MGB-3 ASR challenge using diarization, audio-transcript alignment and transfer learning. In *ASRU*, 2017.
- I. Medennikov, M. Korenevsky, T. Prisyach, Y. Khokhlov, M. Korenevskaya, I. Sorokin, T. Timofeeva, A. Mitrofanov, A. Andrusenko, I. Podluzhny, et al. The stc system for the chime-6 challenge. In *CHiME 2020 Workshop on Speech Processing in Everyday Environments*, 2020.
- T. Menne et al. The RWTH/UPB/FORTH system combination for the 4th CHiME challenge evaluation. In *CHiME-4 Workshop*, 2016.
- V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. Librispeech: an ASR corpus based on public domain audio books. In *ICASSP*, 2015.
- D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le. SpecAugment: A simple data augmentation method for automatic speech recognition. In *Interspeech*, 2019.
- D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, et al. The Kaldi speech recognition toolkit. In *ASRU*, 2011.
- V. Pratap, A. Hannun, Q. Xu, J. Cai, J. Kahn, G. Synnaeve, V. Liptchinsky, and R. Collobert. wav2letter++: The fastest open-source speech recognition system. In *ICASSP*, 2019.
- M. L. Seltzer, D. Yu, and Y. Wang. An investigation of deep neural networks for noise robust speech recognition. In *ICASSP*, 2013.
- C. E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949.
- S. Sun, C.-F. Yeh, M. Ostendorf, M.-Y. Hwang, and L. Xie. Training augmentation with adversarial examples for robust speech recognition. In *Interspeech*, 2018.
- G. Synnaeve, Q. Xu, J. Kahn, T. Likhomanenko, E. Grave, V. Pratap, A. Sriram, V. Liptchinsky, and R. Collobert. End-to-end ASR: from supervised to semi-supervised learning with modern architectures. *arXiv preprint arXiv:1911.08460*, 2019.
- I. Szöke, M. Skácel, L. Mošner, J. Paliesek, and J. Černocký. Building and evaluation of a real room impulse response dataset. *IEEE Journal of Selected Topics in Signal Processing*, 13(4):863–876, 2019.
- Szymański, Piotr, P. Żelasko, M. Morzy, A. Szymczak, M. Żyła-Hoppe, J. Banaszczak, L. Augustyniak, J. Mizgajski, and Y. Carmiel. Wer we are and wer we think we are. *arXiv preprint arXiv:2010.03432*, 2020.
- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

- O. Vinyals, S. V. Ravuri, and D. Povey. Revisiting recurrent neural networks for robust ASR. In *ICASSP*, 2012.
- S. Watanabe, M. Mandel, J. Barker, E. Vincent, A. Arora, X. Chang, S. Khudanpur, V. Manohar, D. Povey, D. Raj, et al. Chime-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings. *arXiv preprint arXiv:2004.09249*, 2020.
- G. Wenzek, M.-A. Lachaux, A. Conneau, V. Chaudhary, F. Guzmán, A. Joulin, and É. Grave. Ccnet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4003–4012, 2020.
- P. C. Woodland, J. J. Odell, V. Valtchev, and S. J. Young. Large vocabulary continuous speech recognition using HTK. In *ICASSP*, 1994.
- W. Zhou, W. Michel, K. Irie, M. Kitzka, R. Schlüter, and H. Ney. The RWTH ASR system for TED-LIUM release 2: Improving hybrid HMM with specaugment. In *ICASSP*, 2020.