



HAL
open science

Multilingual DistilWhisper: Efficient Distillation of Multi-task Speech Models via Language-Specific Experts

Thomas Palmeira Ferraz, Marcely Zanon Boito, Caroline Brun, Vassilina Nikoulina

► **To cite this version:**

Thomas Palmeira Ferraz, Marcely Zanon Boito, Caroline Brun, Vassilina Nikoulina. Multilingual DistilWhisper: Efficient Distillation of Multi-task Speech Models via Language-Specific Experts. ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, Apr 2024, Seoul (Korea), South Korea. hal-04463381

HAL Id: hal-04463381

<https://telecom-paris.hal.science/hal-04463381>

Submitted on 16 Feb 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

MULTILINGUAL DISTILWHISPER: EFFICIENT DISTILLATION OF MULTI-TASK SPEECH MODELS VIA LANGUAGE-SPECIFIC EXPERTS

Thomas Palmeira Ferraz^{♦♦}, Marceley Zanon Boito[♦], Caroline Brun[♦], Vassilina Nikoulina[♦]

[♦]NAVER LABS Europe, France

^{♦♦}Télécom Paris, Institut Polytechnique de Paris, France

ABSTRACT

Whisper is a multitask and multilingual speech model covering 99 languages. It yields commendable automatic speech recognition (ASR) results in a subset of its covered languages, but the model still underperforms on a non-negligible number of under-represented languages, a problem exacerbated in smaller model versions. In this work, we propose *DistilWhisper*, an approach able to bridge the performance gap in ASR for these languages while retaining the advantages of multitask and multilingual capabilities. Our approach involves two key strategies: lightweight modular ASR fine-tuning of *whisper-small* using language-specific experts, and knowledge distillation from *whisper-large-v2*. This dual approach allows us to effectively boost ASR performance while keeping the robustness inherited from the multitask and multilingual pre-training. Results demonstrate that our approach is more effective than standard fine-tuning or LoRA adapters, boosting performance in the targeted languages for both in- and out-of-domain test sets, while introducing only a negligible parameter overhead at inference.

Index Terms— knowledge distillation, multitask speech processing, automatic speech recognition, multilingual speech processing, language experts

1. INTRODUCTION

Whisper [1] is a popular multilingual and multitask speech model that is known for its robustness (i.e. invariant performance over different out-of-domain data) for automatic speech recognition (ASR) [2]. This model covers 99 languages, and jointly trains on ASR, speech translation (many-to-English), language identification, and voice activity detection tasks. The original paper points this multitask training as a reason for the observed robustness of the model to out-of-domain data: compared to the English wav2vec 2.0 model [3], *Whisper* performance seems to generalize better to unseen domains. Presented in many sizes (from tiny to large-v2), we note that there is an important gap in ASR performance between *whisper-large-v2* (largest model) and *whisper-small* (second smallest model) on a large set of languages, including low-resource languages, but also

many high- and mid-resource ones. Such phenomenon in NLP is often referred as *curse of multilinguality* [4, 5, 6], where the performance drop due to the growing amount of covered languages can only be recovered via extensive model scaling. Such scaling comes with an important inference cost increase: for instance, *whisper-large-v2* is 2-3 times slower than *whisper-small*.

A common approach to efficient inference is distilling knowledge from a large multilingual teacher model into a smaller model [7, 8]. However, to apply such knowledge distillation (KD) to *whisper-large-v2*, the best and largest *Whisper* model, we would need to access unavailable information such as the training data across all the tasks and languages, in order to preserve the robustness of the original model. Recent works [9, 10] have demonstrated that the *curse of multilinguality* can also be solved by equipping a moderately sized model with language-specific (LS) modules. Such architecture allows to extend model parameters via extra modules when more languages are added into the model, thus maintaining consistent performance across languages, with no (or very low) extra computations at inference.

Inspired by those findings we propose *DistilWhisper*, which extends *whisper-small* with LS feed-forward layers, that are used in parallel with the original feed-forward layers of the model. In order to preserve the robustness of the original model, *DistilWhisper* introduces the following extensions of previous works: (1) Following [11] we extend conditional language-specific routing (CLSR) modules with the gating mechanism that can route input representation either through the original feed-forward layer or through newly learned LS feed-forward layer; (2) When learning LS layers, we use *whisper-large-v2* as a teacher model with the hypothesis that the KD loss should help reproducing the robustness of the larger *Whisper* model.

Through extensive experiments on a diverse set of languages we demonstrate the effectiveness of *DistilWhisper* compared to standard fine-tuning or LoRA [12] adapters. Our lightweight ASR fine-tuning approach based on CLSR modules generalizes better than LoRA, and the introduction of KD further boosts results in both in- and out-of-domain test sets. We perform additional ablation studies showing our

approach can cope with different amounts of training data. Finally, we demonstrate that the flexibility introduced by the gating mechanism equips *DistilWhisper* with an efficient adaptation approach, leveraging the LS modules only when those are relevant. We make available the models’ weights¹ and code developed in this work.

2. BACKGROUND

State of the art for ASR: Current approaches for ASR mainly rely on the adaptation of pre-trained Transformer stacks learned through self-supervision (i.e. SSL models) on unlabeled audio data. Such pre-trained models vary on the usage of pretext tasks [3, 13, 14] and language coverage [15, 16, 10, 17]. In contrast to this branch of research, the Whisper model relies on weak supervision, which means that the architecture is trained on weakly labeled data only (no self-supervision). Nonetheless, they show that with sufficient amounts of data, the model reaches competitive results compared to mono [1, 2] and multilingual SSL models [10].

Knowledge distillation (KD) has been initially proposed by [18] to distill knowledge from ensemble of models into a single model for ASR. It has further been used to distill knowledge from a large teacher model into smaller student models [7, 8, 19]. While original KD methods relied on minimization of KL-divergence between a teacher model and a student model, recent research [20, 21] have shown that symmetric divergences, such as Jensen-Shannon (JS) divergence, suffer less from borderline behaviors and lead to better results on sequence level distillation.

Adapters are small lightweight modules which are commonly used to adapt pre-trained models to new tasks or domains. In speech-related tasks, adapter-based fine-tuning has been utilized for speech translation [22, 23, 24], and domain adaptation [25, 26], for which they exhibit a similar performance to standard fine-tuning, but with only a fraction of trainable parameters. We also find work on task-adaptation of Whisper [27, 28, 29] using LoRA adapters. In contrast to adapters, in this work we introduce gated LS layers into Whisper, and propose a parameter-efficient KD approach that allows us to increase robustness to out-of-domain data.

3. DISTILWHISPER

With the goal of increasing performance for different languages in models of limited capacity, we propose the *DistilWhisper* approach: we plug conditional language-specific routing (CLSR) modules [11] into a small Whisper (`whisper-small`), and optimize these modules jointly on ASR fine-tuning and KD from a larger Whisper (`whisper-large-`

`v2`). Figure 1 presents our architecture, below we detail its key components.

CLSR module. We extend CLSR modules for the first time to the speech domain. This module learns a hard binary gate $g(\cdot)$ for each input token by using its hidden embedding z^l . These decisions enable a layer to selectively guide information through either a LS path denoted as h^{lang} or a shared path referred to as h^{shared} , as in Eq 1. In contrast to the original CLSR, in this work we use LS gates as shown in Figure 1, instead of sharing them across languages. This allows us to train LS components individually (i.e. in parallel), and then only load the relevant modules at inference. Moreover, our approach also differs from the original CLSR by the positioning: supported by previous work [11, 9], we limit CLSR to the feed-forward, which we also replace entirely by the CLSR module, reducing further the number of parameters. Gating follows [11]: each gate $g(\cdot)$ is made by a two-layer bottleneck network, which is summed to an increasing zero-mean Gaussian noise during training in order to discretize it. At inference time, we adopt hard gating.

***DistilWhisper* approach** is detailed at Figure 1. Our student is enriched with CLSR modules at each feed-forward for each language. These CLSR layers are initialized from the frozen weights of the corresponding feed-forward layer. At training time, for each language the model updates only the corresponding LS layers and gates. At inference time, the model loads the shared layers (multilingual) and the LS modules and gates for the languages of interest, resulting in a limited parameter overhead. We highlight that the use of CLSR modules brings more flexibility to our architecture when compared to adapters, as it allows for routing at the token-level. This makes this approach more capable of leveraging pre-existing knowledge (shared frozen module) via LS gating activation.

***DistilWhisper* optimization.** Following [11], when learning CLSR module parameters, in addition to standard cross-entropy loss \mathcal{L}_{CE} , we employ a gate budget loss \mathcal{L}_g (Eq 2) to balance models’ usage of LS and language-shared modules. It relies on the gate $g(\cdot)$ activation values for a pair (audio, text) (X, Y) in a batch \mathcal{B} , which is expressed by $\mathcal{G}_{(X, Y)} = \sum_{x \in X} \sum_{m \in \mathcal{M}_{enc}} g_m(x) + \sum_{y \in Y} \sum_{m \in \mathcal{M}_{dec}} g_m(y)$ where \mathcal{M}_{enc} and \mathcal{M}_{dec} are respectively the encoders and decoders layers, and $g_m(\cdot) = 1$ when LS layer is selected, or 0 otherwise. The average of this gate usage is constrained to a budget b (Eq 2). For KD, following recent research [20, 21], we use JS divergence, whose loss is detailed in Eq 3, where p is the teacher distribution, q_θ is the student distribution, \mathbf{Y} and \mathbf{Y}^l are sampled from the teacher’s and student’s distributions and compared with their average $m(\cdot) = \frac{1}{2}p(\cdot) + \frac{1}{2}q_\theta(\cdot)$. Thus, CLSR modules parameters are learned to minimize final loss expressed as $\mathcal{L} = \mathcal{L}_{CE} + \mathcal{L}_g + \alpha \mathcal{L}_{KD}$.

$$\text{CLSR}(z^l) = g(z^l) \cdot h^{lang}(z^l) + (1 - g(z^l)) \cdot h^{shared}(z^l). \quad (1)$$

¹Weights available at: <https://huggingface.co/collections/naver/multilingual-distilwhisper-6576ecae8d209fc6a767d9e7>.

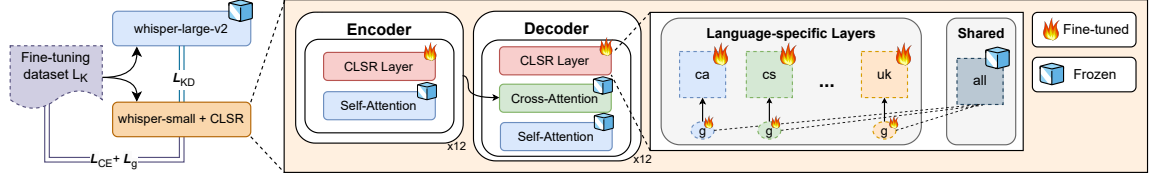


Fig. 1. The *DistilWhisper* optimization approach (left), and its architecture (right). The feed-forward is replaced by a CLSR module, where the LS gates (g) learn to alternate between the pre-trained frozen multilingual representation and the LS layer.

$$\mathcal{L}_g = \left| \frac{\sum_{(X,Y) \in \mathcal{B}} \mathcal{G}(X,Y)}{\sum_{(X,Y) \in \mathcal{B}} (|X| |\mathcal{M}_{\text{enc}}| + |Y| |\mathcal{M}_{\text{dec}}|)} - b \right| \quad (2)$$

$$\mathcal{L}_{\text{KD}} = \frac{1}{2} \mathbb{E}_{\mathbf{Y} \sim p} \left[\log \frac{p(\mathbf{Y})}{m(\mathbf{Y})} \right] + \frac{1}{2} \mathbb{E}_{\mathbf{Y}' \sim q_\theta} \left[\log \frac{q_\theta(\mathbf{Y}')}{m(\mathbf{Y}')} \right] \quad (3)$$

4. EXPERIMENTAL SETUP

Datasets: We downsample the train and validation sets of the CommonVoice 13.0 (CV-13) dataset [30], using equal amounts of training data for each selected language: 10k utterances for training (approx. 14h), 1k for validation. Data selection depends on the amount of up-votes utterances received by annotators. We do not downsample the test set. The FLEURS [31] dataset is used for out-of-domain evaluation, as it provides both a good language overlap with CV-13, and an effective out-of-domain setting for ASR evaluation. For instance, average number of tokens per sample for CV-13 is 36, and 97 for FLEURS.

Language Selection: We consider all Whisper languages with a WER gap of more than 11 between large and small models on CV-13. We then narrow this list considering: 1) minimum amount of utterances (10k); 2) overlap with the FLEURS dataset. The final list of languages is: Catalan (ca), Czech (cs), Galician (gl), Hungarian (hu), Polish (pl), Thai (th), Tamil (ta) and Ukrainian (uk).² These languages encompass 5 language sub-families and vary widely in terms of coverage in the Whisper training set, spanning from 4,300h (pl) to just 9h (gl).

Models: We compare our approach to both *whisper-small* (pre-trained student) and *whisper-large-v2* (teacher) models, as well as two approaches of fine-tuning (FT) for the student: standard fine-tuning (all weights are updated), and LoRA adaptation on top of the feed-forward layer. Finally, we also investigate the impact of the CLSR layer without the use of KD (CLSR-FT), decoupling the effect of KD from the flexibility offered by the routing mechanism on the consequent robustness of the model.

Implementation: We train all models using the Transformers library [32], and make use of *whisper-small* and *whisper-large-v2* pre-trained weights from Hugging-

Face.³ All models are trained for 10 epochs using learning rate 10^{-4} with linear decay, one epoch warm-up, batch size 16, and label smoothing factor 0.1. For LoRA, we use the hyperparameters proposed by [28]. For CLSR training we set gate budget $b = 0.5$ and skip-gate probability $s = 0.2$. For KD we employ JS divergence with temperature $\tau = 1$, weighted such as the learning objective is $\mathcal{L} = \mathcal{L}_{\text{CE}} + \mathcal{L}_g + 2\mathcal{L}_{\text{KD}}$. We report normalized WER using the Whisper normalization with a slight modification to avoid splitting numbers and latin-scripted text into individual characters in languages that do not use space delimitation (th). In all cases, the best model is chosen based on WER on the downsampled CV-13 validation set.

5. RESULTS

We conduct training for each setting using three distinct seeds and present the average scores. Table 1 presents our results. The top portion presents *whisper-large-v2* (upper bound) and *whisper-small* (lower bound) pre-trained scores. The middle portion presents standard fine-tuning (FT) and LoRA adaptation at the feed-forward layers (LoRA-FT). Our results are presented in the bottom: CLSR-FT corresponds to the setting without \mathcal{L}_{KD} , while *DistilWhisper* is the complete setting in which both CLSR and KD losses are leveraged.

***DistilWhisper* versus other adaptation approaches.** For *whisper-small*, we observe that both FT and LoRA-FT approaches (middle portion of Table 1) are able to improve performance on both in- and out-of-domain test sets. However, for FT this boost in performance comes with the cost of language specialization. In contrast to that, LoRA-FT is a light adaptation technique that does not modify the pre-trained representation. This method increases performance on both in-domain (avg -13.1) and out-of-domain (avg -3.5) test sets compared to *whisper-small*. *DistilWhisper* further improves performance over *whisper-small* (avg -15.3) and LoRA-FT (avg -2.2) for in-domain data. It also presents better out-of-domain adaptation capabilities compared to LoRA-FT (avg -2.1).

Impact of knowledge distillation. We observe that *DistilWhisper* on average outperforms all other adaptation ap-

²Although Arabic would also qualify considering our criteria, we find that the dialect from FLEURS differs from the ones present on CV-13.

³<https://huggingface.co/openai/>

	#params	FLEURS avg	CV-13 avg	FLEURS (out-of-domain)								CV-13 (in-domain for FT only)							
				ca	cs	gl	hu	pl	ta	th	uk	ca	cs	gl	hu	pl	ta	th	uk
whisper-large-v2	1.5B	12.5	14.9	5.6	14.3	16.6	17.9	5.9	19.3	12.2	8.1	16.9	14.4	18.9	18.7	8.0	17.3	9.2	15.5
whisper-small	244M	28.3	31.4	14.6	40.4	32.7	43.0	16.7	36.0	22.8	20.5	30.1	38.4	35.5	45.6	18.6	30.0	20.3	32.3
whisper-small+FT	244M	23.3±0.06	16.3±0.09	15.5	31.0	16.9	36.7	22.0	22.7	15.6	25.9	13.7	20.5	11.3	24.1	16.3	13.6	7.4	23.4
whisper-small+LoRA-FT	379M	24.9±0.07	18.2±0.02	17.6	36.9	18.2	41.6	25.9	15.2	11.7	31.8	14.0	23.7	12.7	28.0	21.2	12.0	7.9	26.4
whisper-small+CLSR-FT	369M	23.4±0.19	16.3±0.08	15.7	30.5	17.2	36.9	22.8	22.7	15.6	25.8	14.1	20.3	11.6	24.3	16.1	13.3	7.4	23.4
DistilWhisper	369M	22.8±0.21	16.0±0.04	15.3	30.2	16.7	36.9	21.4	21.8	15.1	24.9	13.8	20.0	11.8	24.0	15.9	12.6	7.2	23.1

Table 1. WER (\downarrow) with dataset averages (avg) for baselines (top), adaptation approaches (middle), and our method (bottom) for in-domain (CV-13, FT only) and out-of-domain (FLEURS, all) test sets. Best results for *whisper-small* in **bold**.

	Train size	FLEURS avg	CV-13 avg	FLEURS			CV-13		
				ca	ta	th	ca	ta	th
whisper-small+CLSR-FT	3k	20.5±0.17	15.0±0.07	17.9	25.6	18.0	19.0	16.4	9.8
DistilWhisper	3k	20.2±0.13	14.6±0.08	17.4	25.5	17.7	18.7	15.7	9.6
whisper-small+CLSR-FT	10k	18.0±0.25	11.6±0.01	15.7	22.7	15.6	14.1	13.3	7.4
DistilWhisper	10k	17.4±0.13	11.2±0.08	15.3	21.8	15.1	13.8	12.6	7.2
whisper-small+CLSR-FT	28k	15.7±0.15	9.5±0.13	13.5	19.8	13.9	11.3	11.3	6.0
DistilWhisper	28k	15.5±0.03	9.3±0.06	13.3	19.3	13.7	11.3	11.0	5.7

Table 2. Average WER (\downarrow) for different training data sizes (3k, 10k, and 28k utterances) for in-domain (CV-13) and out-of-domain (FLEURS) test sets. Best results in **bold**.

proaches (FT, LoRA-FT) for in- and out-of-domain test sets (bottom portion of Table 1). Comparing our models (CLSR-FT and *DistilWhisper*), we observe that the version with KD (*DistilWhisper*) exhibits a slight increase in average in-domain performance (-0.3). In out-of-domain settings, this model consistently outperforms CLSR-FT across all languages (avg -0.6), which confirms our initial hypothesis that the KD loss leverages the robustness from the teacher into the final model. Overall, these results highlight the effectiveness of our proposed architecture: we were able to reduce the out-of-domain performance gap between *whisper-large-v2* and *whisper-small* by 35.2% (avg -5.5) with a parameter overhead at inference time of only 10% (25 M).

Effect of training data size. We now show the effectiveness of our approach on lower and higher data resource settings. For this, we select a subset of languages for which we find more training data available on CV-13 (ca, th, ta). Table 2 presents results for our approach in low (3k utterances; ~4 h), and higher-resource settings (28k utterances; ~40 h), compared to the 10k results from Table 1. We observe that, as expected, increasing the amount of trainable examples leads to superior ASR performance for both approaches, with the leveraging of KD (*DistilWhisper*) being consistently superior to CLSR-FT. For the 28k setup (ca, th, ta), we are able to reduce the out-of-domain WER gap between *whisper-large-v2* and *whisper-small* by 75% (from 12 to 3 WER).⁴ For the 3k setup, we reduce the WER gap by 35.8% using only 4h of training data. This implies that our approach has the potential to improve ASR performance across low-resource languages for which less training data is

⁴ *whisper-large-v2* and *whisper-small* avg FLEURS scores for ca, th, ta are respectively 12.5 and 24.5.

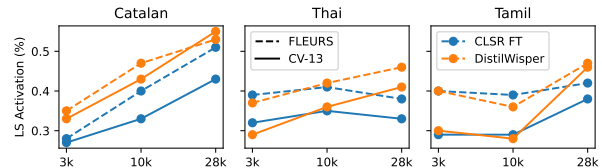


Fig. 2. Ratio of LS layers chosen by the models (CLSR-FT and *DistilWhisper*) depending on (1) amount of training data; (2) in (CV-13) or out-of-domain (FLEURS); (3) language.

available.

Gate Activation Analysis. To better understand how the model uses routing mechanism, we plot gate activation statistics for both CLSR-FT and *DistilWhisper* on Figure 2. We observe that the models tend to rely more on the new LS modules in out-of-domain settings (FLEURS vs CV-13), which could be attributed to the greater complexity and larger size of sentences in FLEURS. Also, as expected, increasing the training data size leads to more reliable LS modules, and therefore higher LS usage. The only exception for this is Thai at the 28k setup, and this might be due to dataset quality and requires further investigation. When comparing the 3 languages, we observe that Catalan exhibits a higher reliance on LS routes, which could also be related to the data quality for this language in CV-13. Finally, we observe that for languages with a weaker teacher (Thai, Tamil) the model may receive contradictory signals at lower-resource settings (3k, 10k), leading to less LS routing usage with KD. However, in the higher resource setting (28k), KD usage leads systematically to more reliable LS module and therefore higher LS routing.

6. CONCLUSION

We presented *DistilWhisper*, a parameter-efficient distillation approach that boosts performance of *whisper-small* by leveraging the robustness from the *whisper-large-v2* into a smaller model, while preserving its multilingual capabilities. This is done by adding language-specific gated modules, and by jointly optimizing ASR fine-tuning and KD losses. Compared to LoRA adapters, and across eight languages, we are able to consistently improve performance in both in- and out-of-domain test sets, while adding only a neg-

ligible number of parameters at inference time. We believe that such architecture would make usage of Whisper models accessible to a larger amount of researchers and practitioners since it allows to boost the performance of a low-inference cost model by 35.2% using only 14 h of training data.

Acknowledgements

This work was partially funded by the ANR project Diké (grant number ANR-21-CE23-0026-02). It was also partially funded by the European Horizon 2022 project UTTER (Unified Transcription and Translation for Extended Reality), under grant agreement No 101070631.

7. REFERENCES

- [1] Alec Radford et al., “Robust speech recognition via large-scale weak supervision,” in *ICML*, 2023.
- [2] Sanchit Gandhi et al., “Esb: A benchmark for multi-domain end-to-end speech recognition,” *arXiv preprint arXiv:2210.13352*, 2022.
- [3] Alexei Baevski et al., “wav2vec 2.0: A framework for self-supervised learning of speech representations,” *Advances in neural information processing systems*, vol. 33, 2020.
- [4] Naveen Arivazhagan et al., “Massively multilingual neural machine translation in the wild: Findings and challenges,” 2019.
- [5] Alexis Conneau et al., “Unsupervised cross-lingual representation learning at scale,” in *ACL*, 2022.
- [6] Naman Goyal et al., “Larger-scale transformers for multilingual masked language modeling,” in *RepL4NLP*, 2021.
- [7] Victor Sanh et al., “Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter,” *CoRR*, vol. abs/1910.01108, 2019.
- [8] Alireza Mohammadshahi et al., “SMaLL-100: Introducing shallow multilingual machine translation model for low-resource languages,” in *EMNLP*, 2022.
- [9] Jonas Pfeiffer et al., “Lifting the curse of multilinguality by pre-training modular transformers,” in *NAACL-HLT*, 2022.
- [10] Vineel Pratap et al., “Scaling speech technology to 1,000+ languages,” *arXiv preprint arXiv:2305.13516*, 2023.
- [11] Biao Zhang et al., “Share or not? learning to schedule language-specific capacity for multilingual translation,” in *ICLR*, 2021.
- [12] Edward J Hu et al., “Lora: Low-rank adaptation of large language models,” in *ICLR*, 2021.
- [13] Wei-Ning Hsu et al., “Hubert: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, 2021.
- [14] Sanyuan Chen et al., “Wavlm: Large-scale self-supervised pre-training for full stack speech processing,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, 2022.
- [15] Alexis Conneau et al., “Unsupervised Cross-Lingual Representation Learning for Speech Recognition,” in *Interspeech*, 2021.
- [16] Arun Babu et al., “XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale,” in *Interspeech*, 2022.
- [17] Yu Zhang et al., “Google usm: Scaling automatic speech recognition beyond 100 languages,” *arXiv preprint arXiv:2303.01037*, 2023.
- [18] Geoffrey Hinton et al., “Distilling the knowledge in a neural network,” in *NIPS Deep Learning and Representation Learning Workshop*, 2015.
- [19] Zhijie Shen et al., “Language-universal adapter learning with knowledge distillation for end-to-end multilingual speech recognition,” *arXiv preprint arXiv:2303.01249*, 2023.
- [20] Yuqiao Wen et al., “f-divergence minimization for sequence-level knowledge distillation,” in *ACL*, 2023.
- [21] Dongyoung Go et al., “Aligning language models with preferences through f -divergence minimization,” in *ICML*, 2023.
- [22] Hang Le et al., “Lightweight adapter tuning for multilingual speech translation,” in *ACL-IJCNLP*, 2021.
- [23] Edward Gow-Smith et al., “Naver labs europe’s multilingual speech translation systems for the iwslt 2023 low-resource track,” *arXiv preprint arXiv:2306.07763*, 2023.
- [24] Anastasopoulos Antonios et al., “Findings of the iwslt 2022 evaluation campaign,” in *IWSLT*, 2022.
- [25] Bethan Thomas et al., “Efficient adapter transfer of self-supervised speech models for automatic speech recognition,” in *ICASSP*, 2022.
- [26] Katrin Tomanek et al., “Residual adapters for parameter-efficient asr adaptation to atypical and accented speech,” in *EMNLP*, 2021.
- [27] Srijith Radhakrishnan et al., “A parameter-efficient learning approach to arabic dialect identification with pre-trained general-purpose speech model,” *arXiv preprint arXiv:2305.11244*, 2023.

- [28] Minghan Wang et al., “WhiSLU: End-to-End Spoken Language Understanding with Whisper,” in *INTER-SPEECH*, 2023.
- [29] Tiantian Feng and Shrikanth Narayanan, “Peft-ser: On the use of parameter efficient transfer learning approaches for speech emotion recognition using pre-trained speech models,” *arXiv preprint arXiv:2306.05350*, 2023.
- [30] Rosana Ardila et al., “Common voice: A massively-multilingual speech corpus,” in *LREC*, 2020.
- [31] Alexis Conneau et al., “Fleurs: Few-shot learning evaluation of universal representations of speech,” in *IEEE Spoken Language Technology Workshop, SLT 2022*. IEEE, 2023.
- [32] Thomas Wolf et al., “Transformers: State-of-the-art natural language processing,” in *EMNLP: system demonstrations*, 2020.

7.1. Effect of Temperature and Distillation Loss

We explore two temperature settings (1 and 3), and compare Jensen-Shannon (JS) loss with the traditional Kullback-Leibler (KL) in the 28k setup. Results are presented for validation (Table 3) and test sets (Table 4). We observe overall better validation scores using JS, a trend that is confirmed by our test results. Regarding temperature, JS equipped with $\tau = 3$ presents inferior validation scores, but superior CV-13 (in-domain) test scores. In this work, we selected our models based on their performance on the validation set of CV-13, which is why we report results for JS with $\tau = 1$ only.

	FLEURS CV-13		FLEURS			CV-13		
	avg	avg	ca	ta	th	ca	ta	th
JS w/ $\tau = 1$	14.8	8.8	12.7	10.7	12.9	5.9	18.7	9.9
JS w/ $\tau = 3$	15.4	8.5	14.5	10.0	12.7	6.0	18.9	9.4
KL w/ $\tau = 1$	15.6	10.2	15.1	13.8	12.8	6.3	18.8	10.5
KL w/ $\tau = 3$	15.7	8.6	15.2	10.5	13.0	5.9	18.8	9.5

Table 3. WER (\downarrow) for in- and out-of-domain **validation sets** for *DistilWhisper* equipped with JS and KL losses, with different temperatures (τ). Best results in **bold**.

	FLEURS CV-13		FLEURS			CV-13		
	avg	avg	ca	ta	th	ca	ta	th
JS w/ $\tau = 1$	15.4	9.3	13.1	19.2	14.0	11.3	10.9	5.7
JS w/ $\tau = 3$	16.3	9.7	14.8	20.1	14.1	11.8	11.3	5.9
KL w/ $\tau = 1$	15.6	10.8	14.6	18.7	13.3	14.9	11.3	6.2
KL w/ $\tau = 3$	16.5	9.7	15.8	19.8	14.0	12.2	11.1	5.9

Table 4. WER (\downarrow) for in- and out-of-domain **test sets** for *DistilWhisper* equipped with JS and KL losses, with different temperatures (τ). Best results in **bold**.

7.2. Stability of the Method

We investigate the stability of the different proposed methods, CLSR-FT, *DistilWhisper* using JS with $\tau = 1$ and *DistilWhisper* using KL with $\tau = 1$, by running three different seeds using the 28k setup. Results are presented in Table 5. Considering a confidence interval of 95%, we confirm *DistilWhisper* with JS loss is statistically superior than its version with only CLSR FT.⁵

⁵Confidence intervals are the following. CLSR-FT: [15.54, 15.87], JS: [15.42, 15.49], KL: [15.24, 15.57].

	Average		Average		ca				th				ta			
	FLEURS	CV-13	FLEURS	CV-13	FLEURS		CV-13		FLEURS		CV-13		FLEURS		CV-13	
	test WER		val WER		Val WER	Test WER	Val WER	Test WER	Val WER	Test WER	Val WER	Test WER	Val WER	Test WER	Val WER	Test WER
CLSR-FT seed 1	15.6	9.7	15.4	8.4	13.1	13.5	9.6	11.4	13.8	13.8	6.4	6.2	19.2	19.6	9.3	11.3
CLSR-FT seed 2	15.6	9.6	15.5	8.3	13.9	13.3	9.6	11.3	13.6	13.8	6.0	6.0	18.9	19.8	9.3	11.4
CLSR-FT seed 3	15.9	9.4	15.4	7.9	13.6	13.6	9.0	11.2	13.5	14.2	5.7	5.9	19.0	19.9	9.0	11.2
CLSR-FT average	15.7 ± 0.2	9.5 ± 0.1	15.4 ± 0.04	8.2 ± 0.3	13.5 ± 0.4	13.5 ± 0.2	9.4 ± 0.4	11.3 ± 0.1	13.6 ± 0.2	13.9 ± 0.2	6.0 ± 0.3	6.0 ± 0.2	19.1 ± 0.2	19.8 ± 0.1	9.2 ± 0.2	11.3 ± 0.1
JS w/ $\tau = 1$ seed 1	15.4	9.3	14.8	8.8	12.7	13.1	10.7	11.3	12.9	14.0	5.9	5.7	18.7	19.2	9.9	10.9
JS w/ $\tau = 1$ seed 2	15.5	9.4	14.9	8.1	12.6	13.6	9.3	11.3	12.9	13.6	5.8	5.8	19.1	19.2	9.3	10.9
JS w/ $\tau = 1$ seed 3	15.5	9.4	14.8	7.9	13.1	13.3	8.8	11.3	12.9	13.6	5.9	5.8	18.5	19.6	8.9	11.2
JS w/ $\tau = 1$ average	15.5 ± 0.03	9.3 ± 0.1	14.8 ± 0.1	8.3 ± 0.5	12.8 ± 0.3	13.3 ± 0.2	9.6 ± 1.0	11.3 ± 0.03	12.9 ± 0.02	13.7 ± 0.2	5.9 ± 0.1	5.7 ± 0.1	18.8 ± 0.3	19.3 ± 0.2	9.4 ± 0.5	11.0 ± 0.2
KL w/ $\tau = 1$ seed 1	15.5	10.8	15.3	10.2	15.1	14.6	13.8	14.9	12.8	13.3	6.3	6.2	18.1	18.5	10.5	11.4
KL w/ $\tau = 1$ seed 2	15.5	10.7	15.1	10.2	14.8	14.6	13.8	14.6	13.0	13.6	6.5	6.3	17.7	18.3	10.4	11.2
KL w/ $\tau = 1$ seed 3	15.2	10.3	15.1	10.5	14.7	14.3	13.7	15.0	13.0	13.6	6.3	6.2	17.7	17.9	11.4	9.8
KL w/ $\tau = 1$ average	15.4 ± 0.2	10.6 ± 0.3	15.2 ± 0.1	10.3 ± 0.2	14.8 ± 0.2	14.5 ± 0.2	13.8 ± 0.1	14.8 ± 0.2	12.9 ± 0.1	13.5 ± 0.1	6.4 ± 0.1	6.2 ± 0.1	17.8 ± 0.2	18.2 ± 0.3	10.8 ± 0.5	10.8 ± 0.9

Table 5. WER (\downarrow) for in- and out-of-domain validation and test sets across the three different methods, and with three different seeds. **Bold** results present the average scores with standard deviation.