DTW-Align: Bridging the Modality Gap in End-to-End Speech Translation with Dynamic Time Warping Alignment

Abderrahmane Issam **Yusuf Can Semerci** **Jan Scholtes** **Gerasimos Spanakis**
Department of Advanced Computing Sciences
Maastricht University
{abderrahmane.issam, y.semerci, j.scholtes, jerry.spanakis}@maastrichtuniversity.nl

Abstract

End-to-End Speech Translation (E2E-ST) is the task of translating source speech directly into target text bypassing the intermediate transcription step. The representation discrepancy between the speech and text modalities has motivated research on what is known as *bridging the modality gap*. State-of-the-art methods addressed this by aligning speech and text representations on the word or token level. Unfortunately, this requires an alignment tool that is not available for all languages. Although this issue has been addressed by aligning speech and text embeddings using nearest-neighbor similarity search, it does not lead to accurate alignments. In this work, we adapt Dynamic Time Warping (DTW) for aligning speech and text embeddings during training. Our experiments demonstrate the effectiveness of our method in bridging the modality gap in E2E-ST. Compared to previous work, our method produces more accurate alignments and achieves comparable E2E-ST results while being significantly faster. Furthermore, our method outperforms previous work in low resource settings on 5 out of 6 language directions. [1]

1 Introduction

End-to-End Speech Translation (E2E-ST) is the task of translating speech in a source language directly into text in a target language. E2E-ST gained success and attention as an alternative to cascaded solutions where an Automatic Speech recognition (ASR) and a Machine Translation (MT) models are combined (Tang et al., 2021; Ye et al., 2022; Fang et al., 2022; Ouyang et al., 2023; Zhou et al., 2023; Le et al., 2023; Zhang et al., 2024, 2025). Cascaded solutions benefit from abundant ASR and MT data but might suffer from error propagation and high latency, which can be solved by E2E-ST.

However, training E2E-ST models is not straightforward due to the representation discrepancy between the speech and text modalities. Previous work has achieved state-of-the-art results by aligning speech and text representations at the word or token level, either using an alignment tool (Ouyang et al., 2023; Fang et al., 2022) or by generating the alignment automatically during training (Zhou et al., 2023; Zhang et al., 2023b). The closest to our work is Cross-modal Mixup via Optimal Transport (CMOT), which uses optimal transport for finding speech and text alignments. Although CMOT achieves state-of-the-art results, it does not guarantee producing monotonic alignments or ensure that each text token is assigned to at least one frame. This contradicts the expected structure of speech-text alignment and can lead to noisy alignments. Furthermore, CMOT introduces a significant training time overhead.

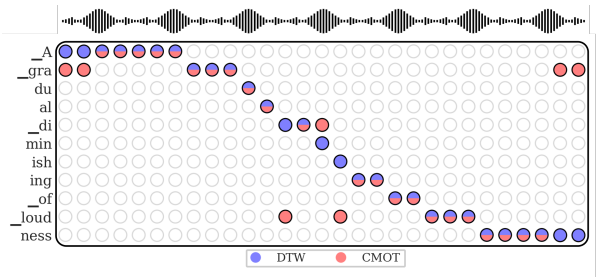


Figure 1: We show an example alignment from DTW (Ours) vs. CMOT. The figure shows that unlike CMOT, DTW guarantees generating monotonic alignments and that all tokens are aligned. In contrast, CMOT failed to align the tokens "min" and "ish" to any frames.

In this work, we introduce **DTW-Align**, a method for aligning speech and text embeddings during training using an adaptation of Dynamic Time Warping (Sakoe and Chiba, 1978). Figure 1 shows an example alignment generated using DTW-Align and CMOT, which illustrates that our method generates monotonic alignments and guarantees that all tokens are aligned, while CMOT does not. We demonstrate the effectiveness of our method

[1] https://github.com/issam9/DTW-Align

Proceedings of the Tenth Workshop on Machine Translation (WMT), pages 191–199
November 8–9, 2025.

in bridging the modality gap with mixup training (Fang et al., 2022; Zhou et al., 2023). Similarly to previous work (Zhou et al., 2023), we train on a mixup of aligned speech and text representations, however, instead of discretely selecting either a speech or a text embedding, we linearly interpolate speech and text embeddings (Zhang et al., 2018). Our experiments show that our method is faster and produces more accurate alignments. Furthermore, it achieves comparable results to CMOT on 6 language directions from the CoVoST2 dataset, while training significantly faster. We also evaluate our method in a low resource setting where training can be more vulnerable to alignment noise, and we show that our method leads to a statistically significant improvement over CMOT in 5 out of 6 language directions.

2 Related Works

Bridging the Modality Gap: The discrepancy between the source and target modalities (i.e. speech and text respectively) has motivated multiple works on what is termed bridging the modality gap (Liu et al., 2019; Han et al., 2021; Fang et al., 2022), where the goal is to build a shared semantic space between the speech and text modalities. Aligning speech and text either based on an alignment tool (Fang et al., 2022; Ouyang et al., 2023) or dynamically during training (Zhang et al., 2023b; Zhou et al., 2023) was shown to achieve state-of-the-art results. Our work goes in this direction, by improving the accuracy and speed of aligning speech and text during training.

Mixup: Mixup is a common data augmentation strategy (Zhang et al., 2018; Jin et al., 2025). In E2E-ST, it is applied for bridging the modality gap (Fang et al., 2022; Zhou et al., 2023), where the model is trained on a discrete mixup of speech and text representations. Mixup training in E2E-ST requires an alignment between speech and text that can be generated using an alignment tool (Fang et al., 2022). Zhou et al. (2023) alleviate the need for an alignment tool by aligning speech and text representations using optimal transport. Our approach is similar to (Zhou et al., 2023), where we generate the alignments dynamically during training. However, instead of discretely mixing speech and text representations, we apply mixup as a linear interpolation.

DTW: DTW is an algorithm for measuring similarity between two sequences of varying length (Sakoe and Chiba, 1978). Due to this property, it has been widely applied to speech data (Juang, 1984; Furtuna, 2008; Muda et al., 2010), and also more specifically in the context of aligning speech and text sequences (i.e. forced alignment). For example, Aeneas (Pettarin, 2017) aligns speech and text utterances by transforming the text utterances into speech, then uses DTW to align the synthetic and original speech sequences. Kürzinger et al. (2020) uses an algorithm that resembles DTW by using dynamic programming and backtracking to find the optimal alignment based on Connectionist Temporal Classification (CTC) probabilities. In this work, we adapt DTW to dynamically align speech and text based on their embeddings.

3 Method

3.1 Architecture

Inspired by previous work in E2E-ST (Fang et al., 2022; Zhou et al., 2023), our model consists of two main components, a speech encoder, and a translation encoder-decoder. The translation encoder-decoder is a standard transformer model that can be decomposed into 3 components: a text embedding layer, an encoder that inputs either speech or text embeddings, and a decoder that generates the target sentence.

3.2 DTW for Aligning Speech and Text Representations

DTW can be used to compute similarities between two sequences of variable length along time. This is achieved by finding an optimal path between the two sequences, or the path that leads to their maximum similarity. The time dimension of the two sequences is said to be *warped*. In our case, when aligning speech and token embeddings, we only warp the token time dimension to have a one-to-many relationship from speech to token embeddings. We start by computing the cosine similarity between each speech embedding $t \in [0; N-1]$ to each token $j \in [0; M-1]$, then we use the similarity matrix $S \in \mathbb{R}^{N \times M}$ to compute a trellis matrix T of the same dimension:

$$T_{t,j} = \begin{cases} S_{t,j} & t = 0, j = 0 \\ -\infty & t = 0, j > 0 \\ +\infty & t > N - M, j = 0 \\ S_{t,j} + T_{t-1,j} & t > 0, j = 0 \\ \max(T_{t-1,j}, T_{t-1,j-1}) & \\ +S_{t,j} & t > 0, j > 0 \end{cases} \quad (1)$$

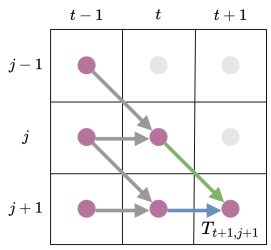


Figure 2: An illustration of the possible alignment paths. Each frame is assigned to only one token, while a token can be assigned to multiple frames.

The last step is backtracking, where we traverse the trellis starting from the last frame and token to find the optimal path, or the path with maximum similarity, which eventually represents the alignment a from speech to text tokens. We assign the last token to the last frame $a_{N-1} = M - 1$, and we traverse as follows:

$$a_t = \begin{cases} M-1 & t = N-1 \\ a_{t+1} - 1 & T_{t,a_{t+1}-1} > T_{t,a_{t+1}} \\ a_{t+1} & \text{else} \end{cases} \quad (2)$$

Figure 2 shows an illustration of the possible alignment paths. We can see that when backtracking from $t+1, j+1$ we can either go to the previous token j if $T_{t,j} > T_{t,j+1}$ or stay on token $j+1$ otherwise, which guarantees monotonicity. The constraints in the trellis matrix guarantee that all tokens are aligned to at least one frame, since the diagonal is filled with $-\infty$ during the trellis computation, when backtracking, this guarantees that we move to $j-1$ when $j \geq t$.

By fully vectorizing both the trellis computation and backtracking, our implementation achieves a much faster alignment.

3.3 Mixup Training

Given a sequence of speech representations generated using the speech encoder $f = [f_0, f_1, ..., f_{N-1}]$ and a sequence of text embeddings generated using the text embedding layer $e = [e_0, e_1, ..., e_{M-1}]$, our method generates an alignment $a = [a_0, a_1, ..., a_{N-1}]$ as described in §3.2. Finally, we apply mixup similarly to previous work (Zhou et al., 2023):

$$m_i = \begin{cases} f_i & p > p^* \\ e_{a_i} & else \end{cases} \quad (3)$$

where p^* is the mixup probability which controls how much text embeddings we introduce into the speech manifold, and p is sampled from a uniform distribution $\mathcal{U}(0, 1)$. We term this discrete mixup. We further introduce interpolation mixup (Zhang et al., 2018), where instead of selecting a speech or text embedding based on probability p^*, we use p^* as a mixup coefficient to linearly interpolate speech and text embeddings:

$$m_i = (1 - p^*).f_i + p^*.e_{a_i} \quad (4)$$

We argue that interpolation mixup can be more robust to alignment noise since the speech embeddings are not entirely replaced as in discrete mixup, but they are softly down-weighted. Therefore, even in the presence of alignment noise, the model still has access to the correct speech embeddings. Furthermore, it can be more data efficient, since all the speech and text token embeddings are included in training, rather than selecting one or the other.

3.4 Training Objective

We train with similar training objectives as CMOT (Zhou et al., 2023) to ensure fair comparison. The ST training corpus is denoted as $D = (s, x, y)$, where s is the speech input, x is the transcription, and y is the translation. In the first stage, the translation encoder-decoder is pre-trained on transcription-translation pairs using cross entropy:

$$\mathcal{L}_{MT} = -\mathbb{E}_{x,y} \log P(y|x) \quad (5)$$

The second stage is multi-task fine-tuning with ST and MT tasks using cross entropy:

$$\begin{aligned} \mathcal{L}_{ST} &= -\mathbb{E}_{s,x,y} \log P(y|s) \\ \mathcal{L}_{MT} &= -\mathbb{E}_{s,x,y} \log P(y|x) \end{aligned} \quad (6)$$

Furthermore, to bridge the modality gap between speech and text representations, we train with Kullback-Leibler (KL) divergence between the output probability distribution under mixup input m, and the output distribution of the ST task, as well as with the output distribution of the MT task:

$$\begin{aligned} \mathcal{L}_{KL_{m \leftrightarrow s}} = \mathbb{D}_{KL}(P(y|s)||P(y|m)) + \\ \mathbb{D}_{KL}(P(y|m)||P(y|s)) \end{aligned} \quad (7)$$

$$\begin{aligned} \mathcal{L}_{KL_{m \leftrightarrow x}} = \mathbb{D}_{KL}(P(y|x)||P(y|m)) + \\ \mathbb{D}_{KL}(P(y|m)||P(y|x)) \end{aligned} \quad (8)$$

Therefore, the final loss is:

$$\mathcal{L} = \mathcal{L}_{ST} + \mathcal{L}_{MT} + \lambda.(\mathcal{L}_{KL_{s \leftrightarrow m}} + \mathcal{L}_{KL_{x \leftrightarrow m}})/2 \quad (9)$$

where λ is a hyperparameter weight to control the KL losses.

4 Experiments

4.1 Dataset

We conduct our experiments on CoVoST-2 dataset (Wang et al., 2020), a large multilingual ST dataset that is based on Common Voice project (Ardila et al., 2020). CoVoST-2 covers translation from 21 source languages to English and from English to 15 target languages, and it contains speech, transcription and translation triplets. In this work, due to computational resources, we focus on 6 language directions: En-De, En-Ca, En-Ar, De-En, Fr-En, and Es-En. These directions are selected to ensure a balanced number of En-X and X-En directions. Furthermore, all languages selected are high resource with a minimum of 97 hours of training data and are of varying linguistic distance from English.

4.2 Experimental Setup

Pre-processing:
For speech input, we use the raw 16 bit 16kHz mono-channel audio. We filter out examples with a number of frames higher than 480k or less than 1k. For the text input, we remove punctuation, then we tokenize using a uni-gram SentencePiece model (Kudo and Richardson, 2018) with a vocabulary of 10k that is shared between the source and target languages.

Model:
Our model is composed of a speech encoder and a translation encoder-decoder. For the speech encoder, we use a pre-trained base HuBERT model (Hsu et al., 2021) for En-X language directions, and mHuBERT-147 (Zanon Boito et al., 2024) (a multilingual version of HuBERT base model) for X-En language directions. To shrink the audio representations over the time axis, we stack 2 1-dimensional convolution layers of kernel size 5, stride size 2, padding 2, and hidden dimension 1024. For the translation encoder, we use 6 transformer encoder layers. For the translation decoder, we use 6 transformer decoder layers. Each transformer layer is comprised of 512 hidden units, 8 attention heads, and 2048 feed-forward hidden units.

Training:
We train our model in two stages, first we pre-train the translation encoder-decoder on CoVoST2 transcription-translation pairs. We train with a learning rate of 1e-4, a maximum of 33k tokens per batch, and for a maximum 100k steps. We early stop training if the loss doesn't decrease for 10 epochs. During the second stage, we fine-tune the speech encoder and translation encoder-decoder with a learning rate of 1e-4, a maximum of 16M audio frames per batch, and we train for 40k steps. For CMOT, NFA-Align and DTW-Align, we train with a mixup probability $p^* = 0.2$ and a KL weight $\lambda = 2.0$.

The MT models are trained using one A100 GPU and ST models are trained using one H100 GPU. We use Fairseq [2] (Ott et al., 2019) for the implementation.

Evaluation:
We average the last 10 epoch checkpoints for evaluation, and generate with a beam size of 5. We use SacreBLEU (Post, 2018) to compute detokenized case-sensitive BLEU score (Papineni et al., 2002). We also use SacreBLEU to measure statistical significance using paired approximate randomization (Riezler and Maxwell, 2005).

Low Resource Setting:
All the languages in our experiments are considered high resource with at least 97 hours of training data, therefore, to evaluate our method in a low resource setting, we simulate a low resource scenario by sampling 10 hours of ST training data and 1 hour of development data for each language directions. During training, we use the same hyperparameters but we early stop if the loss did not decrease on the development set for 10 epochs. Our goal is to demonstrate how noise in the alignment has a more pronounced effect in low-resource ST scenarios. Therefore, we use a simulated low-resource setting with the same languages and training setup to avoid any confounding effects that would arise from using a different dataset.

4.3 Main Results

Baselines:
We experiment with the following models:
HuBERT-Transformer: Composed of speech encoder and translation encoder-decoder trained for ST.
CMOT: HuBERT-Transformer trained by using CMOT alignment for discrete mixup training.
NFA-Align: Using word level alignments from NeMo Forced Aligner (NFA) [3] which was shown to achieve state-of-the-art results in terms of alignment accuracy (Rastorgueva et al., 2023) for mixup training.
DTW-Align-Discrete (Ours): Using DTW for

[2] https://github.com/facebookresearch/fairseq
[3] https://github.com/NVIDIA/NeMo/tree/main/tools/nemo_forced_aligner

Model	En-De	En-Ca	En-Ar	De-En	Fr-En	Es-En	Avg.
Revist-ST ((Zhang et al., 2022))†	17.5	22.9	12.3	14.1	26.9	15.7	-
U2TT (Large) (Zhang et al., 2023a)†	-	-	-	16.7	27.4	28.1	-
DUB (Large) (Zhang et al., 2023a)†	-	-	-	19.5	29.5	**30.9**	-
SRPSE (Zhang et al., 2025)†	-	-	-	21.4	29.3	-	-
CoVoST-2 (Wang et al., 2020)†	18.4	23.6	13.9	18.9	27.0	28.0	21.6
CTC+OT (Le et al., 2023)†	20.6	26.5	15.3	20.4	28.4	29.2	23.4
HuBERT-Transformer	21.4	27.4	15.7	21.8	28.4	28.0	23.8
CMOT	**21.8**	28.2	**16.2**	23.6	30.9	29.6	**25.0**
NFA-Align	21.4	28.1	16.0	23.5	30.9	29.8	24.9
DTW-Align-Discrete	21.7	**28.3**	**16.2**	23.5	**31.0**	29.4	**25.0**
DTW-Align	**21.8**	28.2	16.1	**23.7**	30.8	29.5	**25.0**

Table 1: BLEU score results on CoVoST2 test set. The table shows that CMOT, DTW-Align, and DTW-Align-Discrete achieve the best results against other baselines. †: indicates results reported in the original work (the rest of the baselines are trained in this study)

generating alignments and training with discrete mixup (Equation 3) similar to CMOT.

DTW-Align (Ours): Using DTW for generating alignments and training with interpolation mixup (Equation 4).

5 Results and Discussion

Table 1 shows the results of our method and baselines, and results of previous work that was evaluated on the CoVoST2 dataset. The results show that consistent with previous studies (Fang et al., 2022), the baseline HuBERT-Transformer remains a competitive baseline, even outperforming previous work that uses more complex techniques. Furthermore, CMOT, DTW-Align-Discrete and DTW-Align achieve the best results overall. Although we train under similar settings and we do not optimize our method differently, we achieve similar results to CMOT. Surprisingly, NFA-Align which uses NFA to align speech and text lags slightly behind on average (i.e. 0.1 BLEU), this suggests that in a high resource setting, and with a low mixup probability the effect of noise in the alignment is less evident.

5.1 Alignment Accuracy and Training time

Table 2 shows that our method produces more accurate alignments with a significant increase of 19% in alignment accuracy. Furthermore, our method is more than 33 times faster in terms of execution time, which is concretely manifested in the staggering difference in training time between CMOT and DTW-Align (i.e. 14:20:53 and 6:48:14 respectively). As a reference, HuBERT-Transformer baseline training time is 6:32:53, which means that our method improves the performance over this baseline (by an average of 1.2 BLEU points) without the drawbacks of the significant training time overhead that CMOT suffers from. Therefore, although our method achieves similar results to state-of-the-art CMOT in high resource settings, it offers a significant advantage in terms of training time. In Section 6, we show that due to the improved alignment accuracy, our method is more robust both in low resource settings and under higher mixup probability values.

Method	Accuracy ↑	Execution Time ↓	Train Time ↓
CMOT	26%	97.89	14:20:53
DTW-Align	**45%**	**2.91**	**6:48:14**

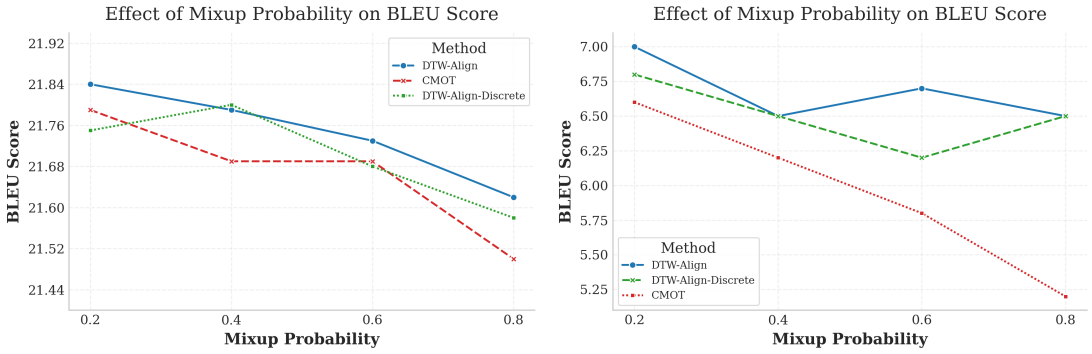
Table 2: We show the accuracy of alignments against NFA, and the execution time on CoVoST2 En-De dev set, plus the training time on En-De train set. DTW-Align is significantly faster and more accurate than CMOT.

6 Analysis

Although our method substantially outperfroms CMOT in terms of alignment accuracy, it does not yield improvements in ST performance. We attribute this to two factors: the amount of training data, which makes training more robust under noise and the low mixup probability value, which is set to 0.2. In §6.1 we measure the performance of CMOT, DTW-Align-Discrete and DTW-Align in a simulated low resource scenario of 10h per lan-

Model	En-De	En-Ca	En-Ar	De-En	Fr-En	Es-En	Avg.
HuBERT-Transformer	6.4	8.7	2.2	1.8	3.2	2.9	4.2
CMOT	6.6	9.6	2.7	2.8	8.5	7.5	6.3
DTW-Align-Discrete	6.8**	9.6	2.8	2.7	**8.6**	7.9**	6.4
DTW-Align	**7.0****	**9.8****	**3.1****	**2.9***	**8.6**	**8.0****	**6.6**

Table 3: BLEU score results on CoVoST2 test set in the low resource setting. The table shows that on overall DTW-Align-Discrete and DTW-Align on overall achieve better results than CMOT, with DTW-Align achieving the best results overall. *, ** indicate whether the improvement over CMOT is statistically significant with $p < 0.05$ and $p < 0.01$ respectively.



(a) En-De

(b) En-De Low Resource

Figure 3: The BLEU score of CMOT and DTW-Align under different mixup probabilities on En-De (Figure 3a) and En-De Low Resource (Figure 3b). DTW-Align is more robust to higher mixup probabilities than CMOT even with discrete mixup. This can be explained by noise in CMOT alignments.

guage direction, and in §6.2 we ablate the mixup probability value.

6.1 Low Resource Setting

Models can be more vulnerable to the negative effects of alignment noise in low resource scenarios. To study this, we compare the performance of CMOT, DTW-Align-Discrete and DTW-Align in a low resource setting of 10h of ST training data and 1h of development data. Table 3 shows the results over the 6 language directions in our experiments. Overall, DTW-Align-Discrete achieves better results than CMOT, with the improvements on En-De and Es-En being statistically significant. Furthermore, DTW-Align achieves the best results, with statistically significant improvement over CMOT on 5 language directions out of 6. These results show that combining the alignment accuracy of DTW and the robustness of interpolation mixup yields the best performance in low resource settings. Although our method performs on par with CMOT in high-resource settings, it offers an increase in performance in low resource ones, where effects of noise on CMOT are more pronounced. Finally, we find that the improvement of DTW-Align over HuBERT-Transformer has doubled (i.e. from 1.2 to 2.4 BLEU points), which demonstrates the advantage of mixup training in low resource settings.

6.2 Mixup Probability

We perform an ablation study on the effect of increasing the mixup probability p^* of CMOT, DTW-Align-Discrete and DTW-Align as shown in Figure 3 on En-De in high (Figure 3a) and low resource setting (Figure 3b). Results indicate that higher mixup probabilities lead to lower performance but the performance degradation is more significant in the case of CMOT, especially in the low resource setting, where training is more vulnerable to noise. This demonstrates that using DTW for aligning speech and text representations is more robust to the mixup probability hyperparameter, especially in low resource scenarios.

7 Conclusion

We introduce a method that eliminates the requirement for an external forced alignment tool by dynamically aligning speech and text embeddings

during training based on Dynamic Time Warping (DTW). Compared to state-of-the-art approaches, our method matches or exceeds BLEU score results while being significantly faster. We further demonstrate that using DTW-Align is more robust and data efficient in low resource settings. In addition, compared to HuBERT-Transformer baseline, our method improves performance by 1.2 and 2.4 BLEU points in high and low resource settings respectively with minimal overhead in the training time. Finally, unlike CMOT, our method can produce both token and word level alignments, which makes it compatible with previous work that requires word level alignments (Fang et al., 2022; Ouyang et al., 2023; Nguyen et al., 2025), therefore, it can bring a boost to the ongoing efforts on bridging the modality gap in E2E-ST or other speech-to-text tasks.

Limitations

Our work considers the following limitations:

Previous work shows that using external MT data for pretraining the translation encoder-decoder improves downstream ST performance. In our experiments, however, we only use internal CoVoST2 data for pretraining because of resource limitations.

Moreover, our work requires speech transcriptions, which might not be available for all languages. Future work can explore using transcriptions from an ASR model potentially extending the method's applicability to a wider range of languages.

Finally, CoVoST2 is an English centric dataset with English as the source or target language in all directions. Evaluating the accuracy and effect of speech and text alignment on other language directions would be valuable for future research.

Acknowledgments

The research presented in this paper was conducted as part of VOXReality project[4], which was funded by the European Union Horizon Europe program under grant agreement No 101070521.

This work used the Dutch national e-infrastructure with the support of the SURF Cooperative using grant no. EINF-11297.

[4] https://voxreality.eu/

References

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.

Qingkai Fang, Rong Ye, Lei Li, Yang Feng, and Mingxuan Wang. 2022. STEMM: Self-learning with speech-text manifold mixup for speech translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7050–7062, Dublin, Ireland. Association for Computational Linguistics.

Titus Furtuna. 2008. Dynamic programming algorithms in speech recognition. *Informatica Economica Journal*, XII.

Chi Han, Mingxuan Wang, Heng Ji, and Lei Li. 2021. Learning shared semantic space for speech-to-text translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2214–2225, Online. Association for Computational Linguistics.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 29:3451–3460.

Xin Jin, Hongyu Zhu, Siyuan Li, Zedong Wang, Zicheng Liu, Juanxi Tian, Chang Yu, Huafeng Qin, and Stan Z. Li. 2025. A survey on mixup augmentations and beyond. *Preprint*, arXiv:2409.05202.

B.-H. Juang. 1984. On the hidden markov model and dynamic time warping for speech recognition — a unified view. *AT&T Bell Laboratories Technical Journal*, 63(7):1213–1243.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Ludwig Kürzinger, Dominik Winkelbauer, Lujun Li, Tobias Watzel, and Gerhard Rigoll. 2020. Ctc-segmentation of large corpora for german end-to-end speech recognition. In *Speech and Computer: 22nd International Conference, SPECOM 2020, St. Petersburg, Russia, October 7–9, 2020, Proceedings*, page 267–278, Berlin, Heidelberg. Springer-Verlag.

Phuong-Hang Le, Hongyu Gong, Changhan Wang, Juan Pino, Benjamin Lecouteux, and Didier Schwab. 2023. Pre-training for speech translation: Ctc meets optimal transport. *Preprint*, arXiv:2301.11716.

Yuchen Liu, Hao Xiong, Jiajun Zhang, Zhongjun He, Hua Wu, Haifeng Wang, and Chengqing Zong. 2019. End-to-end speech translation with knowledge distillation. pages 1128–1132.

Lindasalwa Muda, Mumtaj Begam, and Irraivan Elamvazuthi. 2010. Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques. *J Comput*, 2.

Tu Anh Nguyen, Benjamin Muller, Bokai Yu, Marta R. Costa-jussa, Maha Elbayad, Sravya Popuri, Christophe Ropers, Paul-Ambroise Duquenne, Robin Algayres, Ruslan Mavlyutov, Itai Gat, Mary Williamson, Gabriel Synnaeve, Juan Pino, Benoît Sagot, and Emmanuel Dupoux. 2025. SpiRit-LM: Interleaved spoken and written language model. *Transactions of the Association for Computational Linguistics*, 13:30–52.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Siqi Ouyang, Rong Ye, and Lei Li. 2023. WACO: Word-aligned contrastive learning for speech translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3891–3907, Toronto, Canada. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Alberto Pettarin. 2017. aeneas.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

E. Rastorgueva, V. Lavrukhin, and B. Ginsburg. 2023. Nemo forced aligner and its application to word alignment for subtitle generation. In *Proc. Interspeech 2023*, pages 5257–5258.

Stefan Riezler and John T. Maxwell. 2005. On some pitfalls in automatic evaluation and significance testing for MT. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 57–64, Ann Arbor, Michigan. Association for Computational Linguistics.

H. Sakoe and S. Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49.

Yun Tang, Juan Pino, Xian Li, Changhan Wang, and Dmitriy Genzel. 2021. Improving speech translation by understanding and learning from the auxiliary text translation task. 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)*, pages 4252–4261, Online. Association for Computational Linguistics.

Changhan Wang, Anne Wu, and Juan Pino. 2020. Covost 2: A massively multilingual speech-to-text translation corpus. *Preprint*, arXiv:2007.10310.

Rong Ye, Mingxuan Wang, and Lei Li. 2022. Cross-modal contrastive learning for speech translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5099–5113, Seattle, United States. Association for Computational Linguistics.

Marcely Zanon Boito, Varun Iyer, Nathanaël Lagos, Laurent Besacier, and Ionut Calapodescu. 2024. mhubert-147: A compact multilingual hubert model. In *Proc. Interspeech 2024*, pages 3939–3943.

Biao Zhang, Barry Haddow, and Rico Sennrich. 2022. Revisiting end-to-end speech-to-text translation from scratch. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 26193–26205. PMLR.

Chengwei Zhang, Yue Zhou, Rui Zhao, Yidong Chen, and Xiaodong Shi. 2025. Representation purification for end-to-end speech translation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6255–6269, Abu Dhabi, UAE. Association for Computational Linguistics.

Dong Zhang, Rong Ye, Tom Ko, Mingxuan Wang, and Yaqian Zhou. 2023a. DUB: Discrete unit back-translation for speech translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7147–7164, Toronto, Canada. Association for Computational Linguistics.

Hao Zhang, Nianwen Si, Yaqi Chen, Wenlin Zhang, Xukui Yang, Dan Qu, and Wei-Qiang Zhang. 2023b. Improving speech translation by cross-modal multi-grained contrastive learning. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 31:1075–1086.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

Yuhao Zhang, Kaiqi Kou, Bei Li, Chen Xu, Chunliang Zhang, Tong Xiao, and Jingbo Zhu. 2024. Soft alignment of modality space for end-to-end speech translation. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11041–11045.

Yan Zhou, Qingkai Fang, and Yang Feng. 2023. CMOT: Cross-modal mixup via optimal transport for speech translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7873–7887, Toronto, Canada. Association for Computational Linguistics.