Discrete Cross-Modal Alignment Enables Zero-Shot Speech Translation

Chen Wang[1]*, **Yuchen Liu**[1], **Boxing Chen**[2], **Jiajun Zhang**[1]†,
Wei Luo[2], **Zhongqiang Huang**[2], **Chengqing Zong**[1]
[1] National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing, China
[2] Machine Intelligence Technology Lab, Alibaba DAMO Academy
{wangchen2020,yuchen.liu}@ia.ac.cn, {jjzhang,cqzong}@nlpr.ia.ac.cn
{Boxing.cbx,muzhuo.lw,z.huang}@alibaba-inc.com

Abstract

End-to-end Speech Translation (ST) aims at translating the source language speech into target language text without generating the intermediate transcriptions. However, the training of end-to-end methods relies on parallel ST data, which are difficult and expensive to obtain. Fortunately, the supervised data for automatic speech recognition (ASR) and machine translation (MT) are usually more accessible, making zero-shot speech translation a potential direction. Existing zero-shot methods fail to align the two modalities of speech and text into a shared semantic space, resulting in much worse performance compared to the supervised ST methods. In order to enable zero-shot ST, we propose a novel **D**iscrete **C**ross-**M**odal **A**lignment (DCMA) method that employs a shared discrete vocabulary space to accommodate and match both modalities of speech and text. Specifically, we introduce a vector quantization module to discretize the continuous representations of speech and text into a finite set of virtual tokens, and use ASR data to map corresponding speech and text to the same virtual token in a shared codebook. This way, source language speech can be embedded in the same semantic space as the source language text, which can be then transformed into target language text with an MT module. Experiments on multiple language pairs demonstrate that our zero-shot ST method significantly improves the SOTA, and even performs on par with the strong supervised ST baselines[1].

1 Introduction

End-to-end Speech Translation (ST) aims at designing a single model to directly learn the mapping between source language speech and target language text, and has attracted much attention recently due to its advantages of no error propagation and lower decoding latency (Berard et al., 2016; Liu et al., 2019; Wang et al., 2020c; Xu et al., 2021). However, the training of end-to-end models requires large-scale and high-quality parallel ST data, which are expensive and difficult to obtain. Public datasets such as MUST-C (Gangi et al., 2019a) and CoVoST (Wang et al., 2020a) are quite limited in scale and languages. In contrast, the datasets for automatic speech recognition (ASR) and machine translation (MT) are easier to access in practice. Therefore, zero-shot ST, which learns an end-to-end model using only ASR and MT data, is a direction worth exploring.

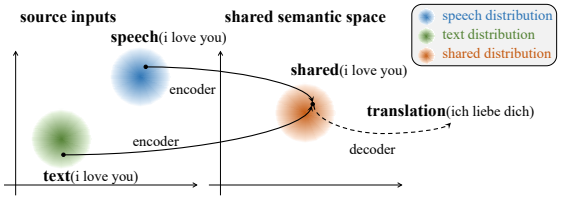

Figure 1: The key of zero-shot speech translation is to learn an appropriate shared semantic space for source language speech and text.

As illustrated in Figure 1, the key of zero-shot ST is to learn an appropriate shared semantic space for source language speech and text, after which the model can translate from the common space using an MT module without relying any ST data. There are some attempts to achieve zero-shot ST using multi-task learning that implicitly aligns speech and text (Escolano et al., 2021; Dinh, 2021). Due to the lack of supervised objective functions for cross-modal alignment, the speech and text representations cannot be well aligned in these methods and the results lag behind supervised settings by a significant margin. Inspired by recent research on learning shared semantic space for speech and text (Alinejad and Sarkar, 2020; Liu et al., 2020;

*Work was done while at Alibaba DAMO Academy.
†Corresponding author.
[1]Our code is available at https://github.com/cwang621/zero-shot-st

Han et al., 2021), we aim to design a supervised cross-modal alignment task that explicitly maps speech and text into a common feature space.

However, our preliminary study indicates that the existing cross-modal alignment methods that bridge the gap between speech and text in a continuous space do not work well in the zero-shot ST task because in this condition, it is difficult to completely align two modalities to the same distribution in a high-dimensional continuous space. To address this issue, we propose a novel **D**iscrete **C**ross-**M**odal **A**lignment (DCMA) method that employs a shared discrete vocabulary space to accommodate and match both modalities of speech and text. With the shared discrete vocabulary across the two modalities, the source language speech and the corresponding text are mapped to the same virtual token, ensuring representational consistency.

Specifically, the ST model consists of a speech encoder and a text decoder. We introduce a vector quantization module between the speech encoder and the text decoder to discretize the continuous representations into a finite set of virtual tokens. The ASR data are used to provide supervision that maps the speech and its corresponding text into the same virtual token of the shared codebook. In addition to the vector quantization module, the speech encoder is decoupled into an acoustic encoder and a semantic encoder. A shared memory module following the speech encoder is introduced to project variable-length input features of both source language speech and text into fix-sized ones. Machine translation is jointly trained to learn the mapping between the fix-sized features on the source side and the text on the target side. To further enhance the speech encoder, masked language model (MLM) and connectionist temporal classification (CTC) are employed as auxiliary tasks in which all parameters are shared. Experimental results on the benchmark dataset MUST-C demonstrate that our discrete alignment method can significantly improve the performance of zero-shot speech translation.

The contributions of this paper are as follows:

- We propose a novel cross-modal alignment method, DCMA, which aligns speech and text in a shared discrete semantic space.
- We design a vector quantization module to discretize continuous representations to a finite set of virtual tokens so that cross-modal alignment in discrete space can be well achieved.
- Experimental results demonstrate that our method significantly improves the SOTA in zero-shot ST and performs on par with the supervised models.

2 Related Work

Data Scarcity in End-to-End ST Berard et al. (2016); Duong et al. (2016) give the first proof of potential for end-to-end ST models, which have become popular recently (Inaguma et al., 2020; Wang et al., 2020b). However, the performance of the end-to-end methods is heavily dependent on large-scale and high quality parallel data, which are difficult to collect on a large scale (Gangi et al., 2019a; Wang et al., 2020a). Many techniques, such as pretraining (Berard et al., 2018; Bansal et al., 2018, 2019; Wang et al., 2020d; Zheng et al., 2021), multi-task learning (Chuang et al., 2020; Xu et al., 2021; Tang et al., 2021a,b), knowledge distillation (Liu et al., 2019; Gaido et al., 2020; Inaguma et al., 2021), multilingual translation (Inaguma et al., 2019; Gangi et al., 2019b; Le et al., 2021), and data augmentation (Jia et al., 2019; Bahar et al., 2019; Wang et al., 2021) are applied to utilize the data from related tasks. Zero-shot scenario has attracted attention in recent years, but there still remains a significant performance gap between zero-shot and supervised methods (Escolano et al., 2021; Dinh, 2021). One contributing factor is the lack of explicit cross-modal alignments.

Cross-modal Alignment in End-to-End ST Cross-modal alignment aims at aligning representations of speech and text to extract common features. Some recent works have pointed out that the representation gap between speech and text is a major obstacle to speech translation. There are many proposals to bridge the gap, including introducing an alignment task (Alinejad and Sarkar, 2020; Liu et al., 2020; Tang et al., 2021a; Han et al., 2021), mixup strategy (Fang et al., 2022) and multimodal pretraining (Zheng et al., 2021; Bapna et al., 2021; Ao et al., 2021; Babu et al., 2021; Bapna et al., 2022). Additionally, some more sophisticated modules such as adaptive feature selection (Zhang et al., 2020), shrink mechanism (Liu et al., 2020) and shared memory module (Han et al., 2021) have been proposed to address the length inconsistency problem. These methods align the representations of two modalities in continuous feature space. Although these methods work effectively in supervised settings, our preliminary study indicates that

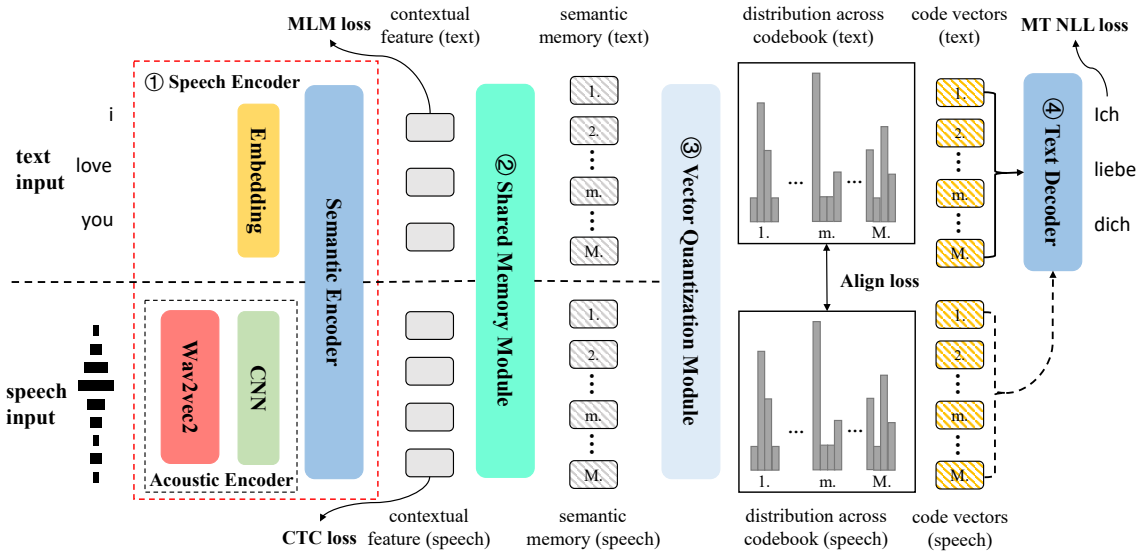

Figure 2: Overview of our model DCMA. The speech encoder is decoupled into acoustic encoder and semantic encoder. We adopt a shared memory module to project feature sequence into a fixed length and a vector quantization module to perform cross-modal alignment in discrete space. The text decoder is used to translate from the discrete common space output by vector quantization module. The part above the dashed line illustrates the data flow of the MT task from source language text into target language text. The ASR data is employed to design the cross-modal alignment loss. The data flow of the bottom part can be used to perform the ST task.

the alignments in continuous space do not work well under zero-shot scenario because it is difficult to match two continuous distributions without strong supervision.

3 Method

3.1 Problem Definition

We attempt to train an end-to-end model with only ASR and MT corpora, achieving zero-shot speech translation. We denote the ASR corpus and the MT corpus as $\mathcal{D}_{ASR} = \{(\mathbf{s}, \mathbf{x})\}$ and $\mathcal{D}_{MT} = \{(\mathbf{x}', \mathbf{y})\}$ respectively, where $\mathbf{s}$ is the audio wave sequence, $\mathbf{x}$ is the corresponding transcripts, $\mathbf{x}'$ is the source language text and $\mathbf{y}$ is the corresponding translation in the target language.

3.2 Model Architecture

Our DCMA model follows the encoder-decoder framework, as shown in Figure 2. In addition to the conventional speech encoder and text decoder, we also introduce a shared memory module and a shared vector quantization module between the encoder and the decoder.

Speech Encoder The speech encoder consists of an acoustic encoder and a semantic encoder to encourage information sharing between tasks (Wang et al., 2020c; Tang et al., 2021a; Xu et al., 2021). For speech input, we use the pretrained wav2vec2.0 (Baevski et al., 2020b) as the acoustic encoder to extract speech representations from the original waveform, which has been shown effective in supervised ST (Ye et al., 2021; Fang et al., 2022). Since the speech feature sequence can be very long, we add two additional one-dimensional convolution layers with stride 2 to shrink the length by a factor of 4. The speech representations are then fed into the shared semantic encoder to obtain the semantic representations. For text input, only the shared semantic encoder is employed. The semantic encoder follows Transformer (Vaswani et al., 2017), and its output is denoted as $\mathbf{H} \in \mathbb{R}^{l \times d}$.

In order to enhance the semantic encoder so that it can embed both acoustic and textual features, we apply the Connectionist Temporal Classification (CTC) (Graves et al., 2006) on the contextual features of speech and the Masked Language Model (MLM) (Devlin et al., 2019) on the contextual features of text. The softmax vocabulary and paramters are shared across the two tasks to encourage implicit alignment between the speech and text representations learnt by the semantic encoder (Bapna et al., 2022). Specifically, the text in source language from $\mathcal{D}_{MT}$ follows the same mask policy as BERT (Devlin et al., 2019), and the model is required to predict the correct masked tokens.

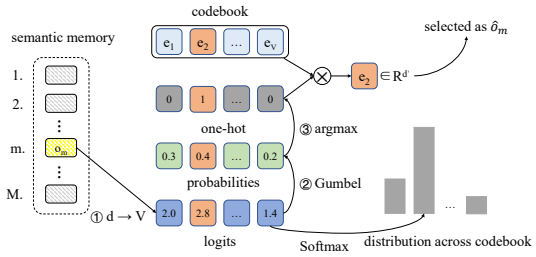

Figure 3: The detailed calculation process of the vector quantization module in one group. This process is performed G times in parallel, and the selected code vectors from G groups are concatenated to get the final output.

The CTC loss is applied on the speech input from $\mathcal{D}_{ASR}$, using token-level transcription as the target.

Shared Memory Module Since the semantic encoder outputs representations in different lengths for speech and text, making it difficult to perform cross-modal alignment, we introduce a shared memory module (Han et al., 2021) to project the contextual features from two modalities with different lengths into fix-sized M. In calculating the attention, this module keeps M learnable, modality-independent memory queries, while the contextual features are used as keys and values as shown below:

$$\mathbf{Q} = \mathbf{M} \in \mathbb{R}^{M \times d} \tag{1}$$

$$\mathbf{K} = \mathbf{V} = \mathbf{H} \in \mathbb{R}^{l \times d} \tag{2}$$

$$\mathbf{O} = \text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \in \mathbb{R}^{M \times d} \tag{3}$$

where $\mathbf{M}$ denotes the trainable memory queries and $\mathbf{H}$ denotes the output contextual features of the semantic encoder. $\mathbf{O} = [o_1, \ldots, o_M]$ are the output of the memory module, in which o_m is the semantic representation extracted by the m-th memory query.

Vector Quantization Module This is the core module of our DCMA model. Inspired by vq-wav2vec (Baevski et al., 2020a), we discretize the semantic memory o_m to a finite set of virtual tokens via a quantizer, so that we can perform cross-modal alignment in discrete space. A codebook is a vocabulary that contains V virtual tokens, each of which is represented by a vector $e \in \mathbb{R}^{d'}$ like word embedding. The vector quantization module aims to select one entry from the codebook as the output.

As illustrated in Figure 3, given a semantic memory o_m, we first apply a linear layer, followed by GELU and another linear layer to map it into logits $l_m \in \mathbb{R}^V$, which is the score of each virtual token. Second, we adopt Gumbel softmax (Maddison et al., 2014; Jang et al., 2017) to choose the discrete entries in a differentiable way. The probabilities for selecting the j-th entry are

$$p_{m,j} = \frac{\exp(l_{m,j} + n_j)/\tau}{\sum_{k=1}^{V} \exp(l_{m,k} + n_k)/\tau} \tag{4}$$

where $n = -\log(-\log(u))$ and u are sampled from the uniform distribution $\mathcal{U}(0, 1)$. The j-th entry is chosen by $j = \text{argmax}_j\, p_{m,j}$ during the forward pass, denoted as $\hat{o}_m = e_j$. The quantization module updates the original semantic memory o_m with $\hat{o}_m$, and performs the same operation for all semantic memories to obtain the outuput code vectors $\hat{\mathbf{O}} \in \mathbb{R}^{M \times d}$. During the backward pass, the gradient of selecting one entry is estimated by the gradient of the true Gumbel softmax output.

Since a codebook contains a limited discrete space of size V, we increase the representation capability of the discrete space by increasing the number of codebooks. Suppose there are G groups with V entries, this module selects one entry from each group and concatenate them to obtain $\hat{o}_m \in \mathbb{R}^{G \times d'}$, where we set $d' = d/G$. The grouping operation can theoretically yield V^G different outputs, which means we can increase the size of the discrete space exponentially. The codebooks are shared across semantic memories extracted by different memory queries, and are also shared across two modalities.

Let $(\mathbf{s}, \mathbf{x})$ be an ASR training sample, the quantization module is expected to select the same codebook entries for the speech and the corresponding text, so that the representations of both modalities are aligned in the discrete space. First, the softmax function is applied to convert the logits of i-th group $l_{m,i} \in \mathbb{R}^V$ into distribution across codebooks.

$$\hat{p}^{modal}_{m,i,j} = \frac{\exp(l^{modal}_{m,i,j})}{\sum_{k=1}^{V} \exp(l^{modal}_{m,i,k})}, modal \in \{\mathbf{s}, \mathbf{x}\} \tag{5}$$

Then we treat the distribution of text as target and encourage the module to make the same choices for the corresponding speech.

$$\mathcal{L}_{align}(\mathbf{s}, \mathbf{x}) = \frac{1}{G} \sum_{m=1}^{M} \sum_{i=1}^{G} \sum_{j=1}^{V} -sg(\hat{p}^{\mathbf{x}}_{m,i,j}) \log \hat{p}^{\mathbf{s}}_{m,i,j} \tag{6}$$

where M,G,V are the number of memory queries, codebook groups and entries respectively, and $sg(\cdot)$ means the *stop gradient* operation.

Text Decoder The text decoder also follows the basic network structure of the Transformer, which takes the fixed-length code vectors $\hat{\mathbf{O}}$ as input, and generates the target translation conditioned on the discrete representations. The code vectors from the text are used in training while those from the speech are used in inference.

The decoder module is trained using parallel text data. Let $(\mathbf{x}', \mathbf{y})$ be an MT training example, the objective function of the MT task can be calculated by cross-entropy loss as in:

$$\mathcal{L}_{MT}(\mathbf{x}', \mathbf{y}) = -\sum_{i=1}^{|\mathbf{y}|} \log p(\mathbf{y}_i | \mathbf{y}_{<i}, \mathbf{x}') \quad (7)$$

3.3 Training Process

We train our model in the pretrain-finetune manner. We first train the semantic encoder, shared memory module, shared vector quantization module and the text decoder with the MT task and the MLM task. It helps to make the training more stable and enriches the codebook entries with semantic information. In the finetune stage, we optimize the entire model with all the relevant tasks as shown below:

$$\begin{aligned}\mathcal{L} =& \mathbb{E}_{(\mathbf{s},\mathbf{x})\in\mathcal{D}_{ASR}}[\mathcal{L}_{align}(\mathbf{s},\mathbf{x}) + \alpha\mathcal{L}_{CTC}(\mathbf{s},\mathbf{x})] \\ &+ \mathbb{E}_{(\mathbf{x}',\mathbf{y})\in\mathcal{D}_{MT}}[\mathcal{L}_{MT}(\mathbf{x}',\mathbf{y}) + \beta\mathcal{L}_{MLM}(\mathbf{x}')]\end{aligned} \quad (8)$$

We optimize $\mathcal{L}_{align}$ and $\mathcal{L}_{CTC}$ in the ASR batches, and alternately optimize the $\mathcal{L}_{MT}$ and $\mathcal{L}_{MLM}$ in the MT batches. Note that no end-to-end ST data are involved in the training process.

4 Experiments

4.1 Datasets

ASR Datasets MUST-C (Gangi et al., 2019a) is one of the largest multilingual speech translation datasets. MUST-C contains the English (En) speech, the corresponding transcription, and the target translation in 8 different languages, including German (De), French (Fr), Russian (Ru), Spanish (Es), Romanian (Ro), Italian (It), Portuguese (Pt), and Dutch (Nl). During training, we use only the speech and its transcription as ASR dataset. During inference, we use the *dev* set for validation and the *tst-COMMON* set for test.

En→	ASR		MT	
	hours	#sentences	name	#sentences
De	408	234K	WMT14	4.5M
Fr	492	280K	WMT14	5.4M*
Ru	489	270K	WMT14	1.0M
Es	504	270K	WMT14	3.8M
Ro	432	240K	WMT16	0.6M
It	465	211K	OPUS100	1.0M
Pt	385	211K	OPUS100	1.0M
Nl	442	253K	OPUS100	1.0M

* We only use *europarl v7*, *commoncrawl* and *news commentary* subsets of WMT14 En-Fr.

Table 1: The detailed statistics of all datasets.

MT Datasets We use MT datasets in various domains different from the ASR dataset. Specifically, we use WMT 2014[2] for En-De, En-Fr, En-Ru and En-Es, WMT 2016[3] for En-Ro, and OPUS100[4] for En-It, En-Pt and En-Nl. The transcription and its translation in MUST-C can serve as in-domain MT data to further investigate the performance of zero-shot ST[5]. The detailed statistics are shown in Table 1.

4.2 Experimental Settings

Pre-processing For speech input, we use the 16 kHZ raw audio waves and normalize the wave sequences by a factor of 2^{15} to the range of $[-1, 1]$. In order to utilize the GPU more efficiently, we filter out speech-transcription pairs whose audio frames exceed 1M.

For the text input, capitalization and punctuation are preserved. We filter out MT samples whose number of source or target tokens is over 250 and whose length ratio is outside the $[2/3, 3/2]$ interval. For each language pair, we use a unigram sentencepiece[6] model to learn a 10K vocabulary from the text portion of MUST-C, and apply it to segment other text data into subword units. The vocabulary is shared across both source and target languages.

Model Configuration We use wav2vec2.0 (Baevski et al., 2020b) as the acoustic encoder, which follows the base configurations and is pretrained on the unlabeled audio data from Lib-

[2] http://www.statmt.org/wmt14/translation-task.html

[3] https://www.statmt.org/wmt16/translation-task.html

[4] http://opus.nlpl.eu/opus-100.php

[5] Our method leverages MT batches and ASR batches alternately, so the source language text overlap brought by introducing in-domain MT data will not cause data leakage.

[6] https://github.com/google/sentencepiece

Methods	Training Data				BLEU							
	Speech	ASR	MT	ST	En-De	En-Fr	En-Ru	En-Es	En-Ro	En-It	En-Pt	En-Nl
Previous state-of-the-art for zero-shot ST												
MultiSLT (Escolano et al., 2021)	×	✓	✓	×	6.8	10.9	-	6.8	-	-	-	-
Previous cross-modal alignment methods												
Chimera* (Han et al., 2021)	✓	✓	✓	×	13.5	22.2	8.3	15.3	8.5	12.6	16.9	13.1
Supervised baselines on MUST-C												
Fairseq ST (Wang et al., 2020b)	×	✓	×	✓	22.7	32.9	15.3	27.2	21.9	22.7	28.1	27.3
Espnet ST (Inaguma et al., 2020)	×	✓	✓	✓	22.9	32.8	15.8	28.0	21.9	23.8	28.0	27.4
W2V2-Transformer**	✓	×	✓	✓	24.1	35.0	16.3	29.4	23.1	24.8	30.0	28.9
This work												
DCMA	✓	✓	✓	×	22.4	29.7	11.8	24.6	16.8	18.4	24.2	22.0
+ in-domain MT data	✓	✓	✓	×	24.0	33.1	16.0	26.2	22.2	24.1	29.2	28.3

Table 2: BLEU scores on the tst-COMMON set in 8 language pairs in MUST-C. "Speech" means speech self-supervised pretraining using unlabeled audio data. ASR data is leveraged for speech recognition task or for cross-modal alignment. * is reproduced under zero-shot scenario, which is a strong baseline of performing cross-modal alignment in continuous space. ** from Fang et al. (2022) is a baseline model by combining wav2vec 2.0 (Baevski et al., 2020b) and a Transformer.

riSpeech (Panayotov et al., 2015). Two additional 1-dimensional convolution layers are used to shrink the length of the speech features, with stride size 2, kernel size 5, padding 2, and hidden dimension 1024.

For the semantic encoder, we use a 6-layer Transformer encoder. The memory queries are 64 512-dimensional vectors. The vector quantization module consists of $G = 128$ groups of codebook with $V = 50$ entries in each group, which can produce 50^{128} possible codewords. A linear layer, followed by GELU and another linear layer are used to project the semantic memory into $G \cdot V = 6400$ logits with 1024 hidden units. The Gumbel softmax produces a one-hot vector for each group. The temperature τ decays exponentially from 2 to 0.5 with a factor of 0.999995 and then keeps constant at 0.5. The text decoder consists of 6 transformer layers. Each of the layers in the semantic encoder and text decoder module has 512-dimensional hidden sizes, 8 attention heads, and 2048 feed-forward hidden units. A 512-dimensional word embedding layer is shared across the semantic encoder and the text decoder.

Training Details We train our model following the pretrain-finetune strategy. During pretraining, we train the model with the MT and MLM tasks. The learning rate is 7e-4 with 4K warm-up updates. We pretrain the model up to 150K updates, with at most 1152 sentence pairs per batch. In the stage of zero-shot finetune, we adopt multi-task training as described in Section 3.3. We set both α and β in Equation (8) to 1.0. The learning rate is set to 1e-4 with 10K warm-up updates. We finetune the model up to 150K updates, with at most 16M audio frames per batch. In both the pretrain and the finetune stages, the model is trained by Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9, \beta_2 = 0.98$. An inverse square root schedule algorithm is adopted for the learning rate. Both the dropout and label smoothing rate are set to 0.1 for regularization. The whole training process is carried out on two Nvidia Tesla-V100 GPUs.

During inference, we average the model parameters of the last 5 checkpoints. We use beam search with a beam size of 5. The performance is evaluated with case-sensitive BLEU (Papineni et al., 2002) calculated by SacreBLEU[7] (Post, 2018).

5 Results and Analysis

5.1 Main Results

Comparison with zero-shot methods We compare our DCMA method with MultiSLT (Escolano et al., 2021), which is the previous SOTA for zero-shot speech translation. As shown in Table 2, our method achieves remarkable improvements. We also notice that introducing in-domain MT data can further improve the performance and is more useful when the MT data size is small (Ru, Ro, It, Pt and Nl)[8]. To demonstrate the advantages of discrete alignment, we implement the Chimera (Han et al., 2021), a continuous alignment method, to

[7] https://github.com/mjpost/sacrebleu

[8] This can be further demonstrated in Appendix A

perform zero-shot ST, which has a similar model architecture as our DCMA and is trained under the same conditions. The difference is that it does not have the vector quatization module and instead aligns the representations of speech and text with the continuous contrastive loss. Our method significantly outperforms Chimera on all language pairs, demonstrating the potential of discrete space for cross-modal alignment in zero-shot ST[9].

Comparison with supervised methods We compare our zero-shot DCMA method with the supervised baselines Fairseq ST (Wang et al., 2020b), Espnet ST (Inaguma et al., 2020), and W2V2-Transformer, which also adopt the pretrain-finetune procedure but are finetuned on the parallel ST data. W2V2-Transformer utilizes a speech self-supervised learning model, combining the pretrained wav2vec 2.0, a 6-layers transforemr encoder and a 6-layers transforemr decoder. As shown in Table 2, our DCMA method achieves competitive performance compared to the supervised baselines when in-domain MT data are used. This demonstrates that the zero-shot DCMA method can learn an effective end-to-end ST model without using end-to-end ST data, as well as the possibility of projecting speech and text into a shared space. Although our zero-shot method uses more MT data than the supervised baselines, our method does not use any end-to-end ST data, making it widely useful in low-resource scenarios.

Comparison with cascaded system and data synthesis method We compare our zero-shot DCMA method with those that also do not use parallel ST data, namely the cascade system and data synthetic method (Jia et al., 2019). For the cascade ST system, the ASR part is the W2V2-Transformer, and the MT part follows the basic Transformer configuration. The cascaded system first translates the speech into source language text, and then translates the transcription into the target translation. For generating synthetic data, we first leverage the MT model in cascaded system to translate the transcriptions in ASR dataset into the target translation. The ST model W2V2-Transformer is initialized with the MT model and finetuned with the synthetic data. As shown in Table 3, our DCMA method achieves comparable performance to those of other methods. However, cascaded system faces the problem of high decoding latency, and generating synthtic data is a time-consuming process. In Section 5.5, we show that our method can outperform the cascade system on well-aligned subsets.

[9]We conduct some analyses of the representations learnt by continuous alignment and discrete alignment in Appendix B

Methods	WER(↓)	MT BLEU	ST BLEU
Cascaded ST	11.1	28.6	23.5
+ in-domain MT data	11.1	32.4	26.7
Synthetic Data	-	28.6	23.3
DCMA	-	-	22.4
+ in-domain MT data	-	-	24.0

Table 3: Comparison with zero-resource methods on MUST-C En-De corpus. We report the Word Error Rate (WER) of speech-transcription pairs for ASR models, the MT BLEU scores of transcription-translation pairs for MT models, and the ST BLEU scores of speech-translation pairs for ST models.

Parameter Sharing	Discrete Alignment	BLEU
✓	✓	22.4
✓	×	1.3
×	✓	21.8
×	×	0.1

Table 4: Ablation studies on the MUST-C En-De corpus. "Parameter sharing" means sharing the softmax vocabulary and parameters across MLM and CTC.

5.2 Ablation Studies

We share the softmax vocabulary and parameters across the two training objectives, $\mathcal{L}_{MLM}$ and $\mathcal{L}_{CTC}$, to encourage implicit alignment between the speech and text representations learnt by the semantic encoder. To better evaluate the contribution of the sharing strategy and our proposed discrete alignment model, we conduct ablation studies on the MUST-C En-De corpus. As shown in Table 4, implicit alignment between speech and text through the sharing strategy is beneficial to improve the performance. However, the proposed discrete alignment method is the most important and indispensable (performance degradation from 22.4 to 1.3 without it).

5.3 Effect of the Size of Codebooks

The shared vector quantization module discretizes the continuous vectors to a finite set of virtual tokens, so we can perform cross-modal alignment in the shared discrete space. An important question arises that how big codebooks are needed so that the vectors can be discretized without losing representation ability. Given G groups of codebook with V entries, the number of theoretically possible outputs is V^G, so that we can exponentially increase

Methods	V	G	MT BLEU	ST BLEU
Transformer	-	-	28.6	-
DCMA	50	2	4.5	-
DCMA	50	4	19.6	-
DCMA	50	8	25.0	-
DCMA	50	16	25.7	-
DCMA	50	32	26.7	18.5
DCMA	50	64	27.4	21.5
DCMA	50	128	28.0	22.4
DCMA	50	256	27.8	19.8

Table 5: MT BLEU scores and ST BLEU scores on the tst-COMMON set of MUST-C En-De corpus with different size of codebooks. The number of theoretically possible outputs is V^G.

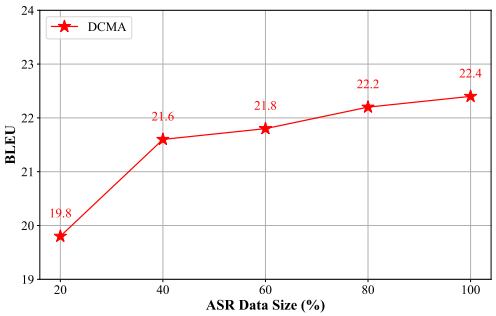

Figure 4: Curve of BLEU scores against the size of ASR data size on MUST-C En-De corpus.

the size of codebooks by increasing G. We vary the setting of G and report the MT BLEU scores of transcription-translation pairs and the zero-shot ST BLEU scores of speech-translation pairs. We observe that when the discrete space is small (e.g. row 2), the quantization operation loses a great deal of representational power, but when the discrete space becomes larger the MT performance gets better and better. However, continuing to increase the size of codebooks (e.g. when G is 256) does not improve the performance. Our proposed DCMA method achieves the best performance when the number of groups is set to $G = 128$.

5.4 Effect of the Size of ASR Data

The key part of our method is to use ASR data to learn a discrete shared semantic space. Therefore, the ASR data size is an important factor. We randomly sample different amount of ASR data from the MUST-C En-De corpus. As shown in Figure 4, we observe a continuous improvement of BLEU scores with the increase of ASR data size.

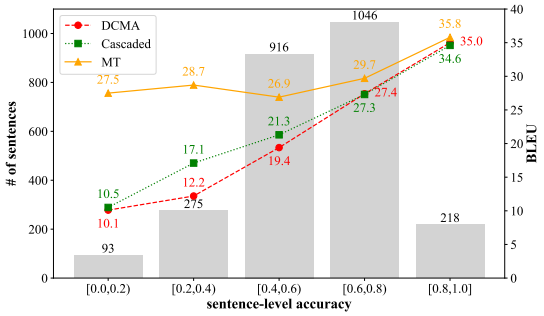

Figure 5: The tst-COMMON set of MUST-C En-De corpus is divided into 5 subsets according to sentence-level alignment accuracy. The histogram represents the size of each subset. Red circles are the ST BLEU scores of DCMA, green squares are the ST BLEU scores of cascaded system, and orange triangles are the transcription-translation BLEU scores of MT model. Our method is comparable to cascaded system and text translation on the well-aligned subsets.

5.5 Can Our Method Achieve Cross-modal Alignment?

To evaluate whether our model can project the speech and text with the same semantic to the same virtual tokens, we conduct some analyses of the alignment accuracy. Let $(\mathbf{s}, \mathbf{x}, \mathbf{y})$ be an ST test sample. The speech input $\mathbf{s}$ can be discretized into $\mathbf{Z^s} = [\mathbf{z_1^s}, \ldots, \mathbf{z_M^s}]$, where $\mathbf{z_i^s} = [z_{i1}^{\mathbf{s}}, \ldots, z_{iG}^{\mathbf{s}}]$ is the code vector ids selected when discretizing the features extracted by the i-th memory query and $z_{ij}^{\mathbf{s}}$ is the code vector id selected in the j-th group. The M and G are the number of memory queries and the number of codebook groups respectively. We do the same operation for text input $\mathbf{x}$ to obtain $\mathbf{Z^x}$, and define the sentence-level accuracy $sent\_acc = \frac{\sum_{i=1}^{M}\sum_{j=1}^{G}\mathbf{1}\{z_{ij}^{\mathbf{s}}=z_{ij}^{\mathbf{x}}\}}{M \cdot G}$. The test set is divided into 5 subsets according to the sentence-level alignment accuracy, and we calculate the MT BLEU scores and ST BLEU scores for each subset. As shown in Figure 5, most of speech utterances are discretized with over 40% alignment accuracy, which indicates the ability of our model to align speech and text into shared discrete codebooks. We also observe a continuous improvement of ST BLEU scores with the increase of sentence-level alignment accuracy. The performance of the zero-shot ST is comparable to that of text translation or better than that of the cascade system on the well-aligned subsets. It indicates the big potential of our method that the zero-shot ST will achieve much better performance if we can design better cross-modal alignment method.

6 Conclusion

In this paper, we propose a novel alignment method DCMA to enable zero-shot ST. The key part of our approach is to discretize the continuous vectors to a finite set of virtual tokens and use ASR data to map the corresponding speech and text to the same virtual token in the shared codebook. Our experiments demonstrate that our method can learn an effective end-to-end ST model without any parallel ST data. It significantly improves the existing SOTA and achieves competitive performance compared to the supervised models.

7 Limitations

In this paper, we propose a zero-shot ST method, which eliminates reliance on end-to-end ST data, allowing end-to-end models to be trained on the same data conditions as cascade systems. The performance of the cascade systems can benefit from both the pretrained speech models (such as wav2vec 2.0) and the pretrained text models (such as BART and T5). However, since our method introduces additional modules between the encoder and the decoder, the pretrained text model cannot be directly integrated into the architecture. Experiments show that our method can outperform the cascade system and obtain comparable results to those of text translation on the well-aligned subsets. However, on the examples with low alignment accuracy, our method is not as robust as the cascade system. How to improve projections onto discrete units is an issue that our future work will explore.

Acknowledgements

This work is supported by the Natural Science Foundation of China under Grant 62122088 and U1836221.

References

Ashkan Alinejad and Anoop Sarkar. 2020. Effectively pretraining a speech translation decoder with machine translation data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 8014–8020. Association for Computational Linguistics.

Junyi Ao, Rui Wang, Long Zhou, Shujie Liu, Shuo Ren, Yu Wu, Tom Ko, Qing Li, Yu Zhang, Zhihua Wei, Yao Qian, Jinyu Li, and Furu Wei. 2021. Speecht5: Unified-modal encoder-decoder pre-training for spoken language processing. *CoRR*, abs/2110.07205.

Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2021. XLS-R: self-supervised cross-lingual speech representation learning at scale. *CoRR*, abs/2111.09296.

Alexei Baevski, Steffen Schneider, and Michael Auli. 2020a. vq-wav2vec: Self-supervised learning of discrete speech representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020b. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Parnia Bahar, Albert Zeyer, Ralf Schlüter, and Hermann Ney. 2019. On using specaugment for end-to-end speech translation. In *Proceedings of the 16th International Conference on Spoken Language Translation, IWSLT 2019, Hong Kong, November 2-3, 2019*. Association for Computational Linguistics.

Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. 2018. Low-resource speech-to-text translation. In *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*, pages 1298–1302. ISCA.

Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. 2019. Pre-training on high-resource speech recognition improves low-resource speech-to-text translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 58–68. Association for Computational Linguistics.

Ankur Bapna, Colin Cherry, Yu Zhang, Ye Jia, Melvin Johnson, Yong Cheng, Simran Khanuja, Jason Riesa, and Alexis Conneau. 2022. mslam: Massively multilingual joint pre-training for speech and text. *CoRR*, abs/2202.01374.

Ankur Bapna, Yu-An Chung, Nan Wu, Anmol Gulati, Ye Jia, Jonathan H. Clark, Melvin Johnson, Jason Riesa, Alexis Conneau, and Yu Zhang. 2021. SLAM: A unified encoder for speech and language modeling via speech-text joint pre-training. *CoRR*, abs/2110.10329.

Alexandre Berard, Laurent Besacier, Ali Can Kocabiyikoglu, and Olivier Pietquin. 2018. End-to-end automatic speech translation of audiobooks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, pages 6224–6228. IEEE.

Alexandre Berard, Olivier Pietquin, Christophe Servan, and Laurent Besacier. 2016. Listen and translate: A proof of concept for end-to-end speech-to-text translation. *CoRR*, abs/1612.01744.

Shun-Po Chuang, Tzu-Wei Sung, Alexander H. Liu, and Hung-yi Lee. 2020. Worse wer, but better bleu? leveraging word embedding as intermediate in multitask end-to-end speech translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5998–6003. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Tu Anh Dinh. 2021. Zero-shot speech translation. *CoRR*, abs/2107.06010.

Long Duong, Antonios Anastasopoulos, David Chiang, Steven Bird, and Trevor Cohn. 2016. An attentional model for speech translation without transcription. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 949–959. The Association for Computational Linguistics.

Carlos Escolano, Marta R. Costa-jussà, José A. R. Fonollosa, and Carlos Segura. 2021. Enabling zero-shot multilingual spoken language translation with language-specific encoders and decoders. In *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2021, Cartagena, Colombia, December 13-17, 2021*, pages 694–701. IEEE.

Qingkai Fang, Rong Ye, Lei Li, Yang Feng, and Mingxuan Wang. 2022. STEMM: self-learning with speech-text manifold mixup for speech translation. *CoRR*, abs/2203.10426.

Marco Gaido, Mattia Antonino Di Gangi, Matteo Negri, and Marco Turchi. 2020. On knowledge distillation for direct speech translation. In *Proceedings of the Seventh Italian Conference on Computational Linguistics, CLiC-it 2020, Bologna, Italy, March 1-3, 2021*, volume 2769 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Mattia Antonino Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019a. Must-c: a multilingual speech translation corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2012–2017. Association for Computational Linguistics.

Mattia Antonino Di Gangi, Matteo Negri, and Marco Turchi. 2019b. One-to-many multilingual end-to-end speech translation. In *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2019, Singapore, December 14-18, 2019*, pages 585–592. IEEE.

Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, volume 148 of *ACM International Conference Proceeding Series*, pages 369–376. ACM.

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, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 2214–2225. Association for Computational Linguistics.

Hirofumi Inaguma, Kevin Duh, Tatsuya Kawahara, and Shinji Watanabe. 2019. Multilingual end-to-end speech translation. In *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2019, Singapore, December 14-18, 2019*, pages 570–577. IEEE.

Hirofumi Inaguma, Tatsuya Kawahara, and Shinji Watanabe. 2021. Source and target bidirectional knowledge distillation for end-to-end speech translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 1872–1881. Association for Computational Linguistics.

Hirofumi Inaguma, Shun Kiyono, Kevin Duh, Shigeki Karita, Nelson Yalta, Tomoki Hayashi, and Shinji Watanabe. 2020. Espnet-st: All-in-one speech translation toolkit. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, pages 302–311. Association for Computational Linguistics.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Ye Jia, Melvin Johnson, Wolfgang Macherey, Ron J. Weiss, Yuan Cao, Chung-Cheng Chiu, Naveen Ari, Stella Laurenzo, and Yonghui Wu. 2019. Leveraging weakly supervised data to improve end-to-end

speech-to-text translation. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, pages 7180–7184. IEEE.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Hang Le, Juan Miguel Pino, Changhan Wang, Jiatao Gu, Didier Schwab, and Laurent Besacier. 2021. Lightweight adapter tuning for multilingual speech translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 2: Short Papers), Virtual Event, August 1-6, 2021*, pages 817–824. Association for Computational Linguistics.

Yuchen Liu, Hao Xiong, Jiajun Zhang, Zhongjun He, Hua Wu, Haifeng Wang, and Chengqing Zong. 2019. End-to-end speech translation with knowledge distillation. In *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pages 1128–1132. ISCA.

Yuchen Liu, Junnan Zhu, Jiajun Zhang, and Chengqing Zong. 2020. Bridging the modality gap for speech-to-text translation. *CoRR*, abs/2010.14920.

Chris J. Maddison, Daniel Tarlow, and Tom Minka. 2014. A* sampling. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3086–3094.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, pages 5206–5210. IEEE.

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 of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 186–191. Association for Computational Linguistics.

Yun Tang, Juan Miguel Pino, Xian Li, Changhan Wang, and Dmitriy Genzel. 2021a. 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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4252–4261. Association for Computational Linguistics.

Yun Tang, Juan Miguel Pino, Changhan Wang, Xutai Ma, and Dmitriy Genzel. 2021b. A general multi-task learning framework to leverage text data for speech to text tasks. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021, Toronto, ON, Canada, June 6-11, 2021*, pages 6209–6213. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Changhan Wang, Juan Miguel Pino, Anne Wu, and Jiatao Gu. 2020a. Covost: A diverse multilingual speech-to-text translation corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 4197–4203. European Language Resources Association.

Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Miguel Pino. 2020b. Fairseq S2T: fast speech-to-text modeling with fairseq. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: System Demonstrations, AACL/IJCNLP 2020, Suzhou, China, December 4-7, 2020*, pages 33–39. Association for Computational Linguistics.

Changhan Wang, Anne Wu, Juan Pino, Alexei Baevski, Michael Auli, and Alexis Conneau. 2021. Large-scale self- and semi-supervised learning for speech translation. In *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, pages 2242–2246. ISCA.

Chengyi Wang, Yu Wu, Shujie Liu, Zhenglu Yang, and Ming Zhou. 2020c. Bridging the gap between pre-training and fine-tuning for end-to-end speech translation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9161–9168. AAAI Press.

Chengyi Wang, Yu Wu, Shujie Liu, Ming Zhou, and Zhenglu Yang. 2020d. Curriculum pre-training for end-to-end speech translation. In *Proceedings of the*

58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pages 3728–3738. Association for Computational Linguistics.

Chen Xu, Bojie Hu, Yanyang Li, Yuhao Zhang, Shen Huang, Qi Ju, Tong Xiao, and Jingbo Zhu. 2021. Stacked acoustic-and-textual encoding: Integrating the pre-trained models into speech translation encoders. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 2619–2630. Association for Computational Linguistics.

Rong Ye, Mingxuan Wang, and Lei Li. 2021. End-to-end speech translation via cross-modal progressive training. In *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, pages 2267–2271. ISCA.

Biao Zhang, Ivan Titov, Barry Haddow, and Rico Sennrich. 2020. Adaptive feature selection for end-to-end speech translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 2533–2544. Association for Computational Linguistics.

Renjie Zheng, Junkun Chen, Mingbo Ma, and Liang Huang. 2021. Fused acoustic and text encoding for multimodal bilingual pretraining and speech translation. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12736–12746. PMLR.

A MT BLEU Scores after Pretraining

To evaluate the domain gap between MUST-C and the MT datasets (WMT and OPUS), we report MT BLEU scores after pretraining.

En→	DCMA	+ in-domain MT data	Δ
De	28.0	31.2	+3.2
Fr	39.3	43.2	+3.9
Ru	14.5	19.6	+5.1
Es	31.6	35.7	+4.1
Ro	21.1	29.1	+8.0
It	23.3	30.8	+7.5
Pt	29.5	36.6	+7.1
Nl	27.3	35.0	+7.7

Table 6: MT BLEU scores on transcription-translation pairs of MUST-C tst-COMMON set.

B Discrete vs. Continuous Alignment

To explore the benefits of discrete alignment, we conduct some analyses of representations learnt by continuous alignment and discrete alignment. Let $(\mathbf{s_i}, \mathbf{x_i})$ be an ASR test sample. The fixed-length representations produced by the encoder are denoted as $\mathbf{O_i^s} = [o_{i1}^s, \ldots, o_{iM}^s]$ and $\mathbf{O_i^x} = [o_{i1}^x, \ldots, o_{iM}^x]$, where M is the number of memories. We define the sentence embedding in each modality $\bar{\mathbf{O}}_\mathbf{i}^\mathbf{s} = \frac{1}{M}\sum_{j=1}^{M} o_{ij}^s$, $\bar{\mathbf{O}}_\mathbf{i}^\mathbf{x} = \frac{1}{M}\sum_{j=1}^{M} o_{ij}^x$. Then we calculate the average memory-level and sentence-level cosine similarity on subsets with different alignment accuracy, as described in Section 5.5.

$$sim\_memory = \frac{1}{N \cdot M}\sum_{i=1}^{N}\sum_{j=1}^{M}\cos(o_{ij}^s, o_{ij}^x)$$

$$sim\_sentence = \frac{1}{N}\sum_{i=1}^{N}\cos(\bar{\mathbf{O}}_\mathbf{i}^\mathbf{s}, \bar{\mathbf{O}}_\mathbf{i}^\mathbf{x})$$

As shown in Table 7, our discrete alignment method significantly improves the sentence-level cosine similarity over the continuous alignment, though both alignments are performed in memory level. We believe it is because that the discrete alignment aligns the corresponding speech and text semantically, rather than just minimizing the distance gap between memories. Our method can also get better memory-level cosine similarity on the well-aligned subsets.

Acc	DCMA		Chimera	
	memory	sentence	memory	sentence
$[0.8, 1.0]$	0.92	0.94	0.87	0.70
$[0.6, 0.8)$	0.84	0.89	0.81	0.63
$[0.4, 0.6)$	0.69	0.80	0.71	0.56
$[0.2, 0.4)$	0.48	0.67	0.58	0.47
$[0.0, 0.2)$	0.28	0.57	0.38	0.36
$[0.0, 1.0]$	0.73	0.82	0.74	0.58

Table 7: Comparison of memory-level and sentence-level representation similarity on different subsets.