BRIDGING THE GAP BETWEEN CASCADE AND END-TO-END CROSS-MODAL TRANSLATION MODELS: A ZERO-SHOT APPROACH

Anonymous authors
Paper under double-blind review

ABSTRACT

One of the main problems in cross-modal translation, such as Speech Translation or OCR Image Translation, is the mismatches among different modalities. The second problem, scarcity of parallel data covering multiple modalities, means that the end-to-end multi-modal neural network models tend to perform worse than cascade models, although there are exceptions under favorable conditions. To address these problems, we present an end-to-end zero-shot translation model, connecting two pre-trained uni-modality modules in a trainable way. We adopt the Word Rotator's Distance loss using the Optimal Transport approach, which effectively handles the multi-modal discrepancy. Furthermore, the approach naturally enables zero-shot multi-modal training, reducing the dependence of end-to-end models on large amounts of data, and at the same time allowing end-to-end training when data do become available. Our comprehensive experiments on the MuSTC benchmarks show that our end-to-end zero-shot approach performs better than or as well as those of the CTC-based cascade models, and that our end-to-end model with supervised training matches the latest state-of-the-art results.

1 INTRODUCTION

To make full of the prodigious amounts of voice, image, video, and many other types of data that are produced every day, it is essential to transfer knowledge among different modalities. However, models more often than not perform worse on cross-modal tasks. A typical example is Speech Translation (ST). The traditional ST method is a cascade approach that first uses automatic speech recognition (ASR) system to transcribe the speech into text and then uses a text machine translation (MT) model. Recent end-to-end (e2e) trainable models remove the need for an explicit ASR. End-to-end ST has several practical advantages over the cascade models such as reduced latency, reduced error propagation, and shorter pipeline.

However, e2e ST models are less competitive than cascade models (Sperber & Paulik, 2020; Dinh, 2021) because end-to-end data are an order of magnitude less than those for ASR or MT. Solutions have been proposed to combat this data problem. For example, one way is to utilize transfer learning to improve ST via large quantities of ASR and MT data. A common drawback in this direction is that the two main modules, ASR and MT, are not always pre-trained jointly. In other words, the framework as a whole is unable to better benefit from the big pre-training data. In (Liu et al., 2020; Xu et al., 2021), an adapter with additional parameters is used during fine-tuning to combine the two pre-trained models. The new module, however, only learns from ST data, which is of a greatly reduced quantity. The second issue is the information loss in building cross-modal representations. Speech and text representations are mismatched because they have different lengths. Some solutions deal with this problem through fixed-size representations (Reimers & Gurevych, 2019; Feng et al., 2020; Han et al., 2021; Duquenne et al., 2022). Others focus on computing the differences between the two representations, for example, as the squared error of mean-pool over time (Pham et al., 2019; Dinh et al., 2022). Although these solutions have achieved significant improvements in several tasks, they will suffer from information loss when representations are compressed or constrained.

In order to overcome both the data and the length problems, we propose an end-to-end zero-shot translation model that connects two pre-trained modules, which could be interpreted as a differentiable cascade system. Specifically, we adopt a popular cross-modal ST architecture (Escolano et al., 2021; Xu et al., 2021), which includes a cross-modal encoder, an adapter, a semantic encoder and a semantic decoder. Our training strategy is to first pre-train the semantic encoder and decoder as in a regular MT model, and then pre-train the cross-modal encoder and the alignment adapter in the connectionist temporal classification (CTC) framework (Graves et al., 2006), which is widely used in an ASR or an optical character recognition (OCR) model. For the alignment adapter, we employ as loss the Word Rotator's Distance (WRD) (Yokoi et al., 2020) adapted with an Optimal Transport (OT) (Monge, 1781; Kantorovich, 1960; Peyré et al., 2019) approach. This way, the adapter learns to promote the cross-modal representations that match in the space of the semantic encoder. Unlike previous works, this strategy allows us to pre-train the adapter. Meanwhile, instead of mapping to a fixed length, the CTC module is used to adjust the length of the source modality representation dynamically. This step can guarantee the cross-modal representations becomes the features with a similar but not exactly the same length, then our proposed WRD objective with OT solver can align them properly.

The contributions of this paper are as follows:
(1) We suggest a pre-train strategy with the WRD loss and the shrink mechanism to facilitate adaptations among different modalities, achieving zero-shot translation without end-to-end labeled data.
(2) End-to-end training is also allowed in our framework if supervised data is available.
(3) Experimental results on the MuST-C demonstrate that our end-to-end zero-shot model can match or be slightly better than the CTC-based cascade model (without intermediate ASR post-processing modules). The results of our end-to-end training can match the current state-of-the-art methods.

2 Preliminary

2.1 Revisiting the Word Rotator's Distance

Word Rotator's Distance (Yokoi et al., 2020), used to measure textual similarity, can be parameterized in terms of the solution to an optimal transport (OT) problem (Monge, 1781; Kantorovich, 1960; Peyré et al., 2019). An OT problem aims to find a transport plan that minimizes the expected cost of transportation between two distributions or two sets of weighted objects. For two sequences of vector representations of different lengths, the "transportation" cost can be thought of as covering the distance between them.

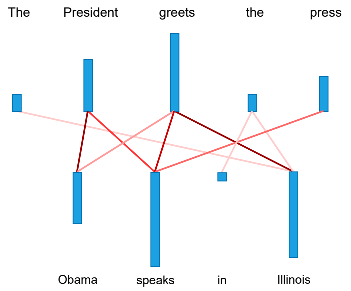


Figure 1: Word Rotator's Distance.

Concretely, given two sentences representations $\mathbf{s}_1 = \{\mathbf{t}_1^1, \ldots, \mathbf{t}_n^1\}$ and $\mathbf{s}_2 = \{\mathbf{t}_1^2, \ldots, \mathbf{t}_m^2\}$ with n, m encoded tokens and $\mathbf{t}_i^{\cdot} \in \mathbb{R}^d$, it is supposed to generate two distributions as normalized weight vectors $\mathbf{p} = [p_1, \ldots, p_n]^\top$ and $\mathbf{q} = [q_1, \ldots, q_m]^\top$. We follow Yokoi et al. (2020) to employ the norm of a word vector as an indicator of its importance to the overall meaning of the sentence.

$$p_i = \frac{\|\mathbf{t}_i^1\|_2}{\sum_{i=1}^{n} \|\mathbf{t}_i^1\|_2}, \qquad q_j = \frac{\|\mathbf{t}_j^2\|_2}{\sum_{j=1}^{m} \|\mathbf{t}_j^2\|_2} \tag{1}$$

Then we can define the Word Rotator's Distance (WRD) between $\mathbf{s}_1$ and $\mathbf{s}_2$ as follows.

$$D_{WRD}(\mathbf{s}_1, \mathbf{s}_2) = \langle \mathbf{C}, \mathbf{T}^* \rangle, \qquad \mathbf{C}_{i,j} = 1 - \cos(\mathbf{t}_i^1, \mathbf{t}_j^2) \tag{2}$$

where $\langle \cdot, \cdot \rangle$ denotes the Frobenius dot-product and $\cos(\cdot, \cdot)$ is the cosine similarity between two vectors. $\mathbf{T}^*$ in (2) is the optimal transport plan from the solution of the following OT problem.

$$\mathbf{T}^* = \arg\min_{\mathbf{T} \geq 0} \langle \mathbf{C}, \mathbf{T} \rangle \qquad \text{s.t.,} \qquad \mathbf{T}\mathbf{1}_m = \mathbf{p}, \quad \mathbf{T}^\top \mathbf{1}_n = \mathbf{q} \tag{3}$$

where $\mathbf{1}_{\cdot}$ represents a vector of all ones. As shown in Figure 1, WRD emphasizes the semantic similarity between two sentences or token sequences better than Euclidean distance. Additionally, WRD applies the cosine distance to measure the dissimilarity between tokens from different sequences, making it a type of edit distance. However, this "smoothed" edit distance is directionless. (”*boys*”, ”*girls*”) and (“*girls*”, ”*boys*”) have the same WRD values. We, therefore, as is common in transformer-based methods, add position embedding to tokens to enrich the sequential order information.

2.2 Differentiable Word Rotator's Distance

Unfortunately, solving WRD (*i.e.*, an OT problem) is a Linear Programming problem that is computationally intractable (Arjovsky et al., 2017) and unable to back-propagate when training end-to-end deep neural networks. It is indispensable to make the optimal transport plan $\mathbf{T}^*$ differentiable. The common solution is to implement the Inexact Proximal point method for Optimal Transport (IPOT) algorithm (Xie et al., 2020b) or the Sinkhorn algorithm (Cuturi, 2013). Both algorithms iteratively converge to the exact optimal transport plan and have pros and cons. Sinkhorn algorithm contains a direct way to compute the Jacobian matrix of $\mathbf{T}^*$ (Xie et al., 2020a; Zhang et al., 2020), while IPOT often has a higher convergence rate and numerical stability in practice. In our experiments, we choose IPOT because of its numerical stability. IPOT replaces the Bregman divergence $D_h(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n x_i \log \frac{x_i}{y_i} - \sum_{i=1}^n x_i + \sum_{i=1}^n y_i$ with the proximal point iteration, *i.e.*, substitutes the following iterative update for the optimization problem in Eq. (3).

$$\mathbf{T}^{(t+1)} = \arg\min_{\mathbf{T}\geq 0} \langle \mathbf{C}, \mathbf{T} \rangle + \beta^{(t)} D_h(\mathbf{T}, \mathbf{T}^{(t)}) \tag{4}$$

Algorithm 1 in Appendix shows the detailed implementation of WRD based IPOT.

3 Main Method

We adopt an encoder-decoder framework as shown in Figure 2(a). It has two main modules: a cross-modal encoder with a shrink adapter and a semantic encoder/decoder pack. The semantic encoder and decoder are pre-trained as in a traditional transformer MT model (Vaswani et al., 2017). Then, the cross-modal encoder and the shrink adapter are pre-trained together in the CTC framework as in an ASR or an OCR model, depending on whether the task is speech or image translation. Note that the cross-modal encoder pre-training requires the previous pre-trained semantic encoder to provide information for cross-modal alignment via an additional WRD OT loss. Finally, during the end-to-end training, we fine-tune the overall architecture, achieving better results than the cascade models.

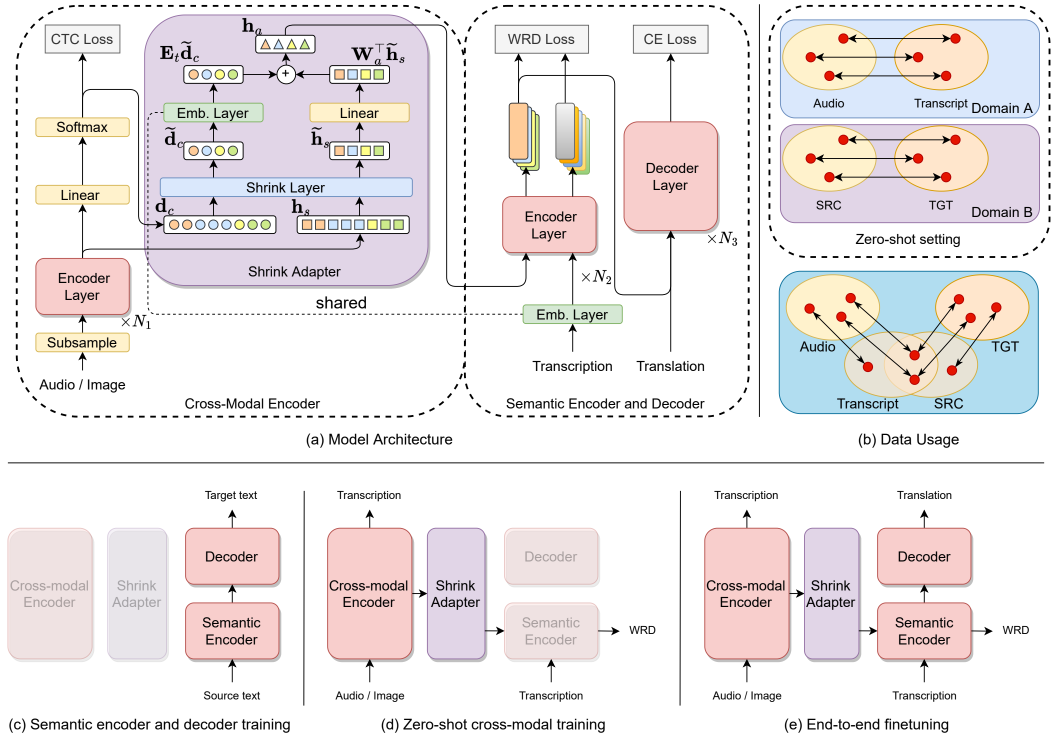
3.1 Semantic encoder-decoder training

In the first phase, the semantic encoder and decoder (Figure 2(c)) are pre-trained to learn text-to-text translation. Given a machine translation corpus $\mathcal{D}_{MT} = \{(\mathbf{x}_t, \mathbf{y}_t)\}$, our aim is to obtain a semantic encoder $\text{Encoder}_t(\mathbf{E}_t\mathbf{x}_t) = \mathbf{h}_t$ and a semantic decoder $\text{Decoder}_t(\mathbf{h}_t) = P(\mathbf{y}_t|\mathbf{h}_t)$, where $\mathbf{h}_t$ is the output of the encoder and $P(\mathbf{y}_t|\mathbf{h}_t)$ is the translation probability computed by the decoder. $\mathbf{E}_t \in \mathbb{R}^{d\times|\mathbb{B}_s|}$ is the source embedding matrix where $\mathbb{B}_s$ is the vocabulary of the source language and d the hidden dimension. The objective of the translation task is defined as the cross entropy loss $\mathcal{L}_{MT} = -\log P(\mathbf{y}_t|\mathbf{h}_t)$.

3.2 Zero-shot Translation Training

In this phase, we train a zero-shot translation model by training the cross-modal encoder alone. Although the cross-modal encoder plays a similar role to an ASR/OCR model, they are not the same. Besides the regular recognition task, we use WRD to supervise the encoder to generate encoding results with less discrepancy across different modalities, *e.g.*, speech and text, or image and text. It is worth emphasizing that only pairwise parallel data (speech-text or image-text paired data) are used for training the cross-modal encoder. Triplet data, (audio or image, source text, target text), are not needed. Specifically, let $\mathcal{D}_{multimodal} = \{(\mathbf{z}_s, \mathbf{x}_s)\}$ denote audio-text or image-text data, we adopt an architecture that combines stacked self-attention layers with CTC module. The former is the encoder for representing the speech or the image $\mathbf{h}_s = \text{Encoder}_s(\mathbf{z}_s)$ and the latter for recognition by optimizing the CTC loss $\mathcal{L}_{CTC}(\mathbf{x}_s, \mathbf{h}_s)$.

We expect to align different modalities in the space of the semantic encoder, allowing the seamless transition between the cross-modal encoder and the semantic decoder and benefiting the downstream translation task. To minimize the distance between the two modalities, we measure the dissimilarity between the cross-modal encoder and the pre-trained semantic encoder via WRD. To be precise, once we obtain the CTC distribution $\mathbf{d}_c = \text{softmax}(\mathbf{W}_c\mathbf{h}_s)$, where $\mathbf{W}_c \in \mathbb{R}^{|\mathbb{B}_s|\times d}$ is a trainable weight matrix, a light-weight shrink adapter integrates the hidden feature $\mathbf{h}_s$ and CTC distribution



(a) Model Architecture

(b) Data Usage

(c) Semantic encoder and decoder training

(d) Zero-shot cross-modal training

(e) End-to-end finetuning

Figure 2: Overview of our proposed framework. **(a)** The overall model contains two main parts, including a cross-modal encoder and a semantic encoder and decoder. **(b)** The upper panel shows the data usage in zero-shot setting, while the lower panel shows the possible supervised data. The three bottom figures reveal our training strategy. **(c)** The semantic encoder and decoder are trained via MT parallel data. **(d)** Only the cross-modal encoder is trained with ASR data or OCR data, where the pre-trained semantic encoder in (c) is freezing, and it drives the encoder to gain the ability with respect to zero-shot translation. **(e)** The model is allowed to be further fine-tuned utilizing triplet data, *e.g.* ST data.

$\mathbf{d}_c$. The shrink mechanism (Yi et al., 2019; Tian et al., 2020; Gaido et al., 2021) shrinks the length of the output from the cross-modal encoder, which contains blank and repeated tokens, almost always incompatible with text. Thus, we consider using the CTC path via efficient $\arg\max$ as guidance to remove the blank tokens and average the representations of consecutively duplicated tokens, as shown in Figure 2(a).

In the traditional cascade models, results from the upstream task, ASR or OCR, are explicitly text whose errors are not easy to rectify. To reduce error propagation, we merge the representations from both before and after the CTC module. That is, the combination in the adapter contains not only the representations $\mathbf{d}_c$ produced by the CTC module, but also the implicit representations from the self-attention layer, $\mathbf{h}_s$ that capture the information in the speech input. Specifically, let $\widetilde{\mathbf{h}}_s$ be the shrunk hidden state and $\widetilde{\mathbf{d}}_c$ the shrunk CTC distribution (see Appendix for the proof of a valid distribution), the adapter output as:

$$\mathbf{h}_a = \mathbf{E}_t\widetilde{\mathbf{d}}_c + \mathbf{W}_a^\top\widetilde{\mathbf{h}}_s, \tag{5}$$

where $\mathbf{E}_t$ is the source embedding matrix in the semantic encoder, and $\mathbf{W}_a \in \mathbb{R}^{|\mathbb{B}|\times d}$ defines a linear layer including the trainable parameters in the adapter. The first term encodes the shrunk CTC distribution like the embedding layer in the semantic encoder but in a soft way. The second term represents the speech representation before the CTC layer, ensuring information retention in the adapter. $\mathbf{h}_a$ can be regarded as the final text embedding which is ready to be fed into the pre-trained semantic encoder. Soft representations (*i.e.*, $\widetilde{\mathbf{d}}_c$) are friendly to back-propagation. Moreover, we can mimic the cascade model by applying $\arg\max$ to obtain the one-hot vector $\hat{\mathbf{d}}_c$ (see more details in Appendix and submitted code). This won't harm the model because the second term can still contribute more stable gradients for the back-propagation.

To alleviate the problem of cross-modal mismatch, we optimize the WRD loss function as

$$\mathcal{L}_{WRD} = D_{WRD}(\text{Encoder}_t(\mathbf{h}_a), \text{Encoder}_t(\mathbf{E}_t\mathbf{x}_s)). \tag{6}$$

Because of CTC prediction errors, it is possible that $\mathbf{h}_a$ has a different sequence length from that of the $\mathbf{x}_s$, but the loss $\mathcal{L}_{WRD}$ can circumvent the length discrepancy in cross-modal representation matching. Previous works (Han et al., 2021; Duquenne et al., 2022) attempt the matching via Euclidean Distance or similar ones that require the features to be mapped to the same length. Combining (16) and (6), the final loss function of the cross-modal training is

$$\mathcal{L}_{ASR/OCR} = \lambda_{ctc}\mathcal{L}_{CTC} + \lambda_{wrd}\mathcal{L}_{WRD} \tag{7}$$

where λ_{ctc} and λ_{wrd} are hyper-parameters. The training procedure is shown in Figure 2(d). To keep the semantic encoding intact, the semantic encoder including the embedding matrix is frozen during optimization, leading to a zero-shot translation system naturally from the ASR or OCR training.

3.3 End-to-End Translation Training

In the fine-tuning phase, we utilize speech translation datasets to improve the pre-trained modules previously. The training procedure is shown in Figure 2(e). Different from MT samples or ASR samples, a speech translation corpus usually consists of speech-transcription-translation triplets denoted as $\mathcal{D}_{ST} = \{(\mathbf{z}, \mathbf{x}, \mathbf{y})\}$. Note that the shrink adapter has already connected the cross-modal encoder with the semantic encoder and decoder. All parameters in the whole model are pre-trained in advance on $\mathcal{D}_{MT}$ and $\mathcal{D}_{ASR}$. We can directly train the end-to-end model $\mathcal{L}_{ST}$ with the speech-translation pairs. However, to fully utilize the transcription data, we can adopt Knowledge Distillation (KD) to further improve the performance of the model in the end-to-end training phase. By minimizing the cross-entropy loss between the pre-trained teacher MT model and the student ST model, the ST model could preserve the knowledge from the MT model during training (Liu et al., 2019). Since the zero-shot training loss is still valid in this phase, we can integrate it into the final end-to-end training objective function.

$$\mathcal{L} = \mathcal{L}_{ST}(\mathbf{z}, \mathbf{y}) + \lambda_{kd}\mathcal{L}_{KD}(\mathbf{z}, \mathbf{x}, \mathbf{y}) + \lambda_{ctc}\mathcal{L}_{CTC}(\mathbf{z}, \mathbf{x}) + \lambda_{wrd}\mathcal{L}_{WRD}(\mathbf{z}, \mathbf{x}) \tag{8}$$

4 Experiments on Speech Translation

In this section, we evaluate our approach on both zero-shot and supervised end-to-end ST and compare it with recent SOTA methods.

4.1 Datasets and Settings

ST MuST-C (Cattoni et al., 2021) is a multilingual speech translation corpus including several hundred hours of English audio recordings from TED Talks. It has been a benchmark ST dataset to facilitate the training of end-to-end systems from English into several languages in the speech-transcription-translation triplet format. We conduct our experiments on the three popular language pairs: English-German (En–De), English-French (En–Fr), and English-Spanish (En–Es). For each language pair, model selection is based on the dev-set, and the final results are reported on the tst-COMMON and tst-HE test sets. Note that there are two versions of En-De datasets where version 2 is annotated with higher quality. We evaluate our approach on both versions.

ASR The open-sourced LibriSpeech English ASR dataset (Panayotov et al., 2015) comes from audio-books and contains 960 hours of speech samples and the corresponding transcriptions. This data is used for pre-training the ASR in the zero-shot training stage.

MT For each language pair, we use WMT parallel data to pre-train the machine translation model. For En-De and En-Fr, we collect the WMT 2014 data with about 4.5M and 36M parallel sentences respectively as in Vaswani et al. (2017). For En-Es, we collect the WMT 2013 data of size 28M.

Model Details The audio inputs are pre-processed as 80-channel log Mel filterbank coefficients computed every 10ms with a 25ms window as fairseq[1]. The cross-model encoder contains two 1D

[1] https://github.com/facebookresearch/fairseq/tree/main/examples/speech_to_text

Training Data	**En-De v2**		**En-De**		**En-Fr**		**En-Es**	
	common	he	common	he	common	he	common	he
WMT	28.59	27.45	28.62	27.44	40.79	37.54	32.38	38.14
WMT + MuST-C	33.13	31.99	32.99	31.90	44.14	39.97	36.96	42.04

Table 1: Performance of MT on MuST-C testset (BLEU↑). The input is the ground truth of the source transcription. These results could be the upper bound of our zero-shot ST.

Model		**En-De v2**		**En-De**		**En-Fr**		**En-Es**		Average
		common	he	common	he	common	he	common	he	Gap
MultiSLT	cascade	/	/	17.30	/	27.15	/	21.29	/	-13.79
	zero-shot	/	/	6.77	/	10.85	/	6.75	/	
Chimera	zero-shot	/	/	13.5	/	22.2	/	15.3	/	/
Ours	cascade	22.85	22.27	22.45	22.30	32.60	31.65	26.14	31.55	+0.79
	zero-shot	24.00	23.04	23.41	22.94	33.65	32.25	26.48	32.32	
***Pseudo* Zero-Shot ST. MT is trained on WMT and MuST-C parallel corpus.**										
Tight Integrated[†]	cascade	/	/	25.9	25.0	/	/	30.2	37.6	-1.325
	p. zero-shot	/	/	25.1	24.4	/	/	28.7	35.2	
Ours	cascade	26.43	25.14	25.21	25.32	34.53	32.63	29.15	34.68	+0.78
	p. zero-shot	27.39	26.46	26.52	25.46	35.34	33.66	29.46	35.05	

Table 2: Zero-Shot ST on MuST-C (BLEU↑). MT is trained on WMT data alone. [†]Tight Integrated extends our ASR data to 2300 hours, and it used 27M En-De and 48M En-Es MT data.

convolutional subsampler layers (Synnaeve et al., 2019) and 12 transformer encoder layers with hidden dimension 512. The MT model is a standard transformer-based architecture (Vaswani et al., 2017) with 6 layers for both the encoder and the decoder. For each language, an individual vocabulary including 10K sub-word units is learned by SentencePiece (Kudo & Richardson, 2018). The CTC prediction shares the same vocabulary with the semantic encoder (*i.e.,* English vocabulary) but different weights. The embedding layer in the adapter shares the weights with the source embedding in the semantic encoder. All hyper-parameters are tuned on En-De v2 and directly applied to other datasets. Additional training and implementation details can refer to the Appendix.

4.2 Zero-Shot ST

Recent work (Dinh, 2021; Escolano et al., 2021) indicates that when large amounts of ASR and MT data dominate the training, the cascaded ST is better than the direct end-to-end ST. How to leverage more from the potentials of ASR and MT data in end-to-end zero-shot ST is still under-explored. In our proposed second phase, the desired ASR training can easily facilitate the building of a zero-shot ST model. In our first experiment, we pre-train the MT model with the WMT data alone, preventing the model from accessing the in-domain data of MuST-C. The MT results are presented in Table 1. For the ASR training, we combine the Librispeech data and the speech-transcription pairs in MuST-C to give a comparable amount of ASR data as in the practical cascade system. In our ASR loss Eq. (7), we set $\lambda_{ctc} = 1$ and $\lambda_{wrd} = 10$.

Our main results of zero-shot ST are illustrated in Table 2. We compare our model with the pioneering zero-shot ST method **MultiSLT** (Escolano et al., 2021), achieving the zero-shot translation via ASR training with an adapter as well. Their MT model is also pre-trained on WMT data. In the MultiSLT system, the zero-shot models trail the cascade system by -13.79 BLEU points. In our system, the end-to-end zero-shot model on average performs +0.79 higher than that of the cascade system. We compare to another cross-modal alignment method **Chimera** (Han et al., 2021), which is initially designed for supervised ST training. Chimera leverages the pre-trained Wav2Vec2.0 on Librispeech dataset to extract the speech features and map them to fixed-length vectors. Once the features of the transcriptions have the same length, the cross-modality alignment can be readily achieved via contrastive learning. This method can then be directly applied to end-to-end zero-shot ST by separating out the supervised triplet end-to-end data from the ASR and the MT data. With the open-source repository, we produce the **Chimera** results in zero-shot setups, shown in Table

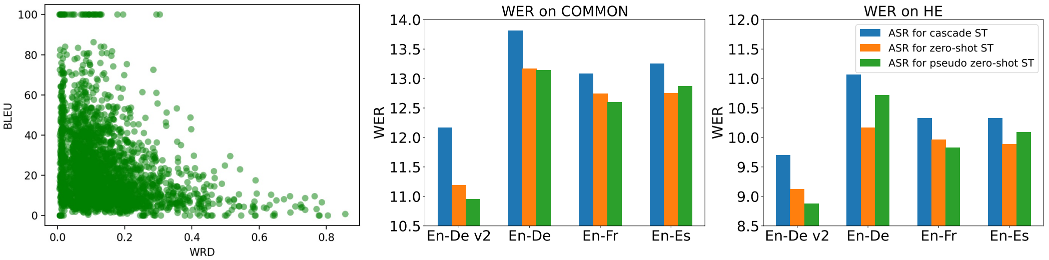


Figure 3: **Leftmost panel**: BLEU of zero-shot ST *v.s.* WRD for each sentence. **Right two panels**: The performance of the ASR as zero-shot ST systems.

Model	**Num.**	**En-De v2**		**En-De**		**En-Fr**		**En-Es**	
	Params	common	he	common	he	common	he	common	he
MTL[†] (Tang et al., 2021b)	31M	/	/	23.9	/	33.1	/	28.6	/
FAT-ST (Zheng et al., 2021)	58M	/	/	25.5	/	/	/	30.8	/
JT-S-MT* (Tang et al., 2021a)	74M	/	/	26.8	/	37.4	/	31.0	/
Chimera (Han et al., 2021)	165M	/	/	27.1	/	35.6	/	30.6	/
XSTNET (Ye et al., 2021b)	155M	/	/	27.8	/	38.0	/	30.8	/
STEMM (Fang et al., 2022)	155M	/	/	**28.7**	/	37.4	/	31.0	/
zero-shot	95M	24.00	23.04	23.41	22.94	33.65	32.25	26.48	32.32
FT from zero-shot	95M	**29.22**	29.07	28.22	28.22	39.00	37.06	31.96	38.83
Pseudo zero-shot	95M	27.39	26.46	26.52	25.46	35.34	33.66	29.46	35.05
FT from *pseudo* zero-shot	95M	29.12	**29.74**	28.17	28.19	**39.05**	**37.21**	**32.03**	**38.89**

Table 3: Supervised ST on MuST-C (BLEU↑ with beam=5) with additional datasets Librispeech and WMT data. [†] MTL uses the hidden dimension 256. * JT-S-MT only uses WMT data.

2. The fixed-length feature matching improves the zero-shot translation over the adapter method of MultiSLT, though not as much as what we propose here.

We also conduct a *pseudo* zero-shot ST experiment. The ASR training data remains the same as the regular zero-shot training, whereas the first phase MT is pre-trained on both WMT and MuST-C text parallel corpus. The MT performance is greatly improved as shown in Table 1. In this setup, even though each training phase doesn't directly consume any speech-translation pairs, the overlapped MuST-C transcription data could be seen by both ASR and MT models. The BLEU scores of both the cascade and zero-shot ST increase by a large margin. However, the gap between them remains virtually unchanged (+0.79 becomes +0.78). It is an indication of the stability of our approach to bridging the modality gap. We then plot the relation between BLEU and WRD for each sentence in the tst-CMMON set of En-De v2 (Figure 3). The overall trend indicates the BLEU decreases with increasing WRD. This setup has also been discussed by **Tight Integrated** Dalmia et al. (2021); Bahar et al. (2021). Because it uses more ASR and MT data, its cascade model performs better. In contrast, our end-to-end model has a special adaptor to support better performance.

Recall that the original objective of the second phase is ASR training. In Figure 3, we plot the histogram of the WER for different ASR systems. The ASR of the cascade system (*i.e.*, trained with CTC loss only and without semantic encoder) has a clearly higher WER than our proposed ASR training with additional WRD loss. However, the in-domain MuST-C data do not appear to make a significant difference as indicated by the orange and the green bars in Figure 3.

4.3 Supervised ST

Since our zero-shot speech translation is essentially a differentiable end-to-end cascade model, we can leverage the supervised ST data to fine-tune all the parameters. In this experiment, we evaluate the performance of our third training phase where we set the loss weights in Eq. (8), where we set the loss weights of each additional task as $\lambda_{kd} = 0.8$, $\lambda_{ctc} = 0.3$ and $\lambda_{wrd} = 10$. Since our models are pre-trained with Librispeech ASR data and WMT parallel corpus, we compare our approach only in the unconstrained scenario with the latest SOTA methods that used similar datasets. The results

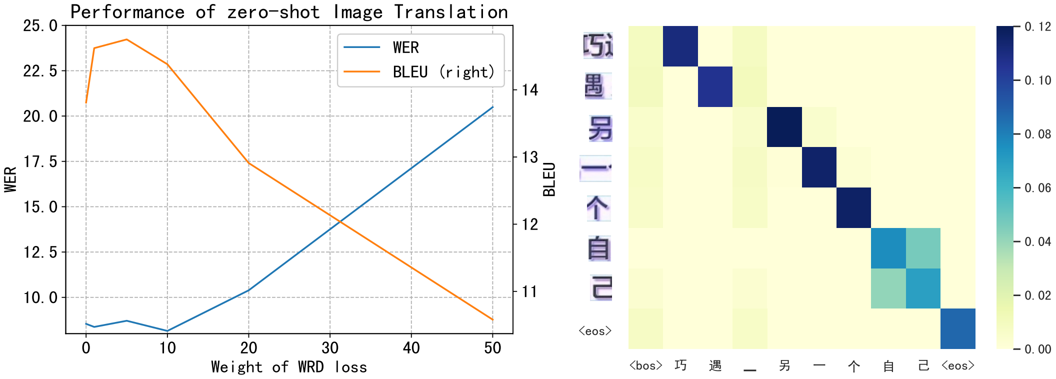


Figure 4: **Left Panel**: the performance of OCR and zero-shot image translation over different weights λ_{wrd}. **Right Panel**: an example of the visualization of transport plan $\mathbf{T}^*$.

are summarized in Table 3. Among the three language pairs, our approach achieves 1+ BLEU improvement over the previous SOTA on En-Fr (XSTNET) and En-Es (JT-S-MT and STEMM), and scores second on En-DE. As stated in section 4.1, we tune our hyper-parameters on En-De v2 while other methods tune on En-De. This could be the contributing reason why our En-De ST performs slightly worse.

We also include the results of zero-shot ST in Table 3, and we see our zero-shot ST almost matches the small-sized supervised model MTL, and the *pseudo* zero-shot ST also matches some regular-sized supervised models. We notice that fine-tuning after *pseudo* zero-shot (the rows with the initialism FT) does not bring large benefits over the regular zero-shot. This phenomenon could indicate that for low-resource languages where the supervised ST data are likely to be more scarce or even unavailable, an end-to-end ST model could still be built with ASR and MT data alone.

5 Zero-Shot Image Translation

We also conduct experiments on zero-shot Image Translation using only OCR data and NMT data to further test the effectiveness of our framework.

Datasets The NMT model (*i.e.* the semantic encoder and decoder) is pre-trained on the WMT 2018 Zh-En data which contains about 20M parallel sentences in the news domain. We use 2M Chinese e-commerce images[2] and 2M Chinese text line images[3] for our OCR task. The test set contains 2000 e-commerce images with Chinese transcriptions and English translations by human annotations. The BLEU score of the pre-trained NMT model on the test set is 15.87.

Model Details For image inputs, we make two modifications. First, the hidden dimension of the whole architecture is set to 256. Second, the two-layer 1D convolution subsampler is replaced with the patch embedding layer in Vision Transformer (Dosovitskiy et al., 2020), which is a 2D convolution layer with the same height as images and a width of 4. We use SentencePiece (Kudo & Richardson, 2018) to obtain a Chinese vocabulary of size 7K most of which are characters and the rest sub-words, and an English vocabulary of size 12K. For batch training, all images are resized to a fixed height of 32 pixels.

We use a small hidden dimension for fast verification. In particular, we set different weights $\lambda_{wrd} = 0, 1, 5, 10, 20, 50$ to investigate the effectiveness of the WRD loss, where the model with $\lambda_{wrd} = 0$ reduces to a cascade model. The results of the zero-shot Image Translation are shown in the left panel of Figure 4. By increasing the weight of the WRD loss, the model improves both in WER and BLEU. However, when the weight is too large, the system deteriorates (*e.g.* when $\lambda_{wrd} = 20, 50$). In the right panel of Figure 4, we visualize the transport plan $\mathbf{T}^*$ of a testing example between the character level images and the transcription tokens, and present corresponding cost matrix $\mathbf{C}$ in

[2] https://taobao.com

[3] https://github.com/YCG09/chinese_ocr

the Appendix. More examples in the Appendix also show that the WRD with OT solver can align cross-modal features with different lengths.

6 Related Works

Optimal Transport Optimal Transport has been applied to machine learning, especially in computer vision (Rubner et al., 2000; Xie et al., 2020a; Zhang et al., 2020) and in natural language processing tasks. In natural language processing, Kusner et al. (2015) proposes Word Mover's Distance for Document (WMD) to measure document distances. Yokoi et al. (2020) proposed Word Rotator's Distance based on WMD by considering the norm of a word vector and the angle between word vectors. Chen et al. (2019; 2020) adopt it in sequence-to-sequence learning and cross-modal alignment. But WRD is not used on intermediate representations. Instead, fixed-length representations are used to resolve the conflict between different modalities.

End-to-end ST Since Bérard et al. (2016) proposed proof of the potential of end-to-end ST (without using intermediate representations explicitly), the concept has been investigated often in recent years, for example (Bansal et al., 2018; Inaguma et al., 2020). To overcome the scarcity of ST data, multi-task training and pre-training are proposed to incorporate ASR and MT data (Weiss et al., 2017; Bérard et al., 2018; Alinejad & Sarkar, 2020; Le et al., 2020; Vydana et al., 2021; Ye et al., 2021a). Especially, Wang et al. (2020); Xu et al. (2021); Fang et al. (2022) attempt to construct end-to-end trainable cascade systems that rely on transcriptions and are optimized in part during end-to-end training. Liu et al. (2020); Papi et al. (2021); Zeng et al. (2021) also adopt the CTC module as shrinking guidance. Other techniques such as data augmentation (Jia et al., 2019; Pino et al., 2020), knowledge distillation (Liu et al., 2019; Xu et al., 2021), and meta-learning (Indurthi et al., 2020) were widely exploited for ASR and MT data.

Zero-shot ST Zero-shot ST trains models only on non-overlapping ASR and MT data and performs ST tasks during inference. Jia et al. (2019) proposes the pseudo labeling as a data augmentation trick to construct end-to-end supervised data. Dinh (2021) shares ASR encoder layers and NMT encoder layers with an auxiliary loss function to minimize the difference between intermediate representations of text and audio. Escolano et al. (2021) proposed an encoder-adapter-decoder architecture where the speech encoder is compatible with the text decoder. Duquenne et al. (2022) adopt a similar framework, but with joint representations. It tries to show that the single vector representation is efficient to decode various languages. In our study, we employ an encoder-adapter-MT framework but pay special attention to the adapter connecting different encoder modules, leading to an architecture that is feasible both as a cascade and an end-to-end zero-shot ST system.

End-to-end Image Translation (Salesky et al., 2021) proposed a text-image-text framework to achieve robust text translation by constructing pseudo text-line images from the source sentences. Recently, a few works on directly translating source text in real images into a foreign language have been proposed (Chen et al., 2021; Hinami et al., 2021; Shekar et al., 2021). The availability of triplet image data is severely more limited than in Speech Translation. It is at the present not possible to train a supervised end-to-end image translation model. To look for data, Chen et al. (2021) collected Chinese-English bilingual movie subtitle images. Unfortunately, these images are far different from real images in many aspects such as font, variety, size, and content. Thus in our work, we experiment with zero-shot Text Image Translation with only OCR and MT data which is more common in real-world applications.

7 Conclusion

In this paper, we present an end-to-end zero-shot architecture that takes better advantage of cascade models, bridging the gap between cascade and end-to-end translation models. With the proposed differentiable shrink adapter and differentiable WRD loss, our approach is a direct end-to-end ST model in the zero-shot setup that matches the performance of the cascade system without additional post-processing, e.g., rescoring via an additional language model. Moreover, our zero-shot translation model is end-to-end trainable, allowing further training on supervised data if they are available. Experiments show that we achieve comparable results as recent SOTA methods. We hope that our work can further the research on cross-modality translation and representation learning.

REFERENCES

Ashkan Alinejad and Anoop Sarkar. Effectively pretraining a speech translation decoder with machine translation data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8014–8020, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.644. URL https://aclanthology.org/2020.emnlp-main.644.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR, 2017.

Parnia Bahar, Tobias Bieschke, Ralf Schlüter, and Hermann Ney. Tight integrated end-to-end training for cascaded speech translation. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pp. 950–957. IEEE, 2021.

Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. Low-resource speech-to-text translation. In *INTERSPEECH*, 2018.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Alexandre Bérard, Olivier Pietquin, Christophe Servan, and Laurent Besacier. Listen and translate: A proof of concept for end-to-end speech-to-text translation. *arXiv preprint arXiv:1612.01744*, 2016.

Alexandre Bérard, Laurent Besacier, Ali Can Kocabiyikoglu, and Olivier Pietquin. End-to-end automatic speech translation of audiobooks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6224–6228. IEEE Press, 2018. doi: 10.1109/ICASSP.2018.8461690. URL https://doi.org/10.1109/ICASSP.2018.8461690.

Roldano Cattoni, Mattia Antonino Di Gangi, Luisa Bentivogli, Matteo Negri, and Marco Turchi. Must-c: A multilingual corpus for end-to-end speech translation. *Computer Speech & Language*, 66:101155, 2021.

Liqun Chen, Yizhe Zhang, Ruiyi Zhang, Chenyang Tao, Zhe Gan, Haichao Zhang, Bai Li, Dinghan Shen, Changyou Chen, and Lawrence Carin. Improving sequence-to-sequence learning via optimal transport. *arXiv preprint arXiv:1901.06283*, 2019.

Liqun Chen, Zhe Gan, Yu Cheng, Linjie Li, Lawrence Carin, and Jingjing Liu. Graph optimal transport for cross-domain alignment. In *International Conference on Machine Learning*, pp. 1542–1553. PMLR, 2020.

Zhuo Chen, Fei Yin, Xu-Yao Zhang, Qing Yang, and Chena-Lin Liu. Cross-lingual text image recognition via multi-task sequence to sequence learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 3122–3129. IEEE, 2021.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Siddharth Dalmia, Brian Yan, Vikas Raunak, Florian Metze, and Shinji Watanabe. Searchable hidden intermediates for end-to-end models of decomposable sequence tasks. In *NAACL-HLT*, 2021.

Tu Anh Dinh. Zero-shot speech translation. *arXiv preprint arXiv:2107.06010*, 2021.

Tu Anh Dinh, Danni Liu, and Jan Niehues. Tackling data scarcity in speech translation using zero-shot multilingual machine translation techniques. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6222–6226, 2022. doi: 10.1109/ICASSP43922.2022.9746815.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Paul-Ambroise Duquenne, Hongyu Gong, Benoît Sagot, and Holger Schwenk. T-modules: Translation modules for zero-shot cross-modal machine translation. *arXiv preprint arXiv:2205.12216*, 2022.

Carlos Escolano, Marta R Costa-jussà, José AR Fonollosa, and Carlos Segura. Enabling zero-shot multilingual spoken language translation with language-specific encoders and decoders. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 694–701. IEEE, 2021.

Qingkai Fang, Rong Ye, Lei Li, Yang Feng, and Mingxuan Wang. 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)*, pp. 7050–7062, 2022.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. Language-agnostic BERT sentence embedding. *CoRR*, abs/2007.01852, 2020. URL `https://arxiv.org/abs/2007.01852`.

Marco Gaido, Mauro Cettolo, Matteo Negri, and Marco Turchi. Ctc-based compression for direct speech translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 690–696, 2021.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pp. 369–376, 2006.

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

Ryota Hinami, Shonosuke Ishiwatari, Kazuhiko Yasuda, and Yusuke Matsui. Towards fully automated manga translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 12998–13008, 2021.

Hirofumi Inaguma, Shun Kiyono, Kevin Duh, Shigeki Karita, Nelson Yalta, Tomoki Hayashi, and Shinji Watanabe. ESPnet-ST: All-in-one speech translation toolkit. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 302–311, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-demos.34. URL `https://aclanthology.org/2020.acl-demos.34`.

Sathish Indurthi, Houjeung Han, Nikhil Kumar Lakumarapu, Beomseok Lee, Insoo Chung, Sangha Kim, and Chanwoo Kim. End-end speech-to-text translation with modality agnostic meta-learning. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7904–7908, 2020. doi: 10.1109/ICASSP40776.2020.9054759.

Ye Jia, Melvin Johnson, Wolfgang Macherey, Ron J Weiss, Yuan Cao, Chung-Cheng Chiu, Naveen Ari, Stella Laurenzo, and Yonghui Wu. Leveraging weakly supervised data to improve end-to-end speech-to-text translation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7180–7184. IEEE, 2019.

L. V. Kantorovich. Mathematical methods of organizing and planning production. *Management Science*, 6(4):366–422, 1960. doi: 10.1287/mnsc.6.4.366. URL `https://doi.org/10.1287/mnsc.6.4.366`.

Taku Kudo and John Richardson. 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*, pp. 66–71, 2018.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *International conference on machine learning*, pp. 957–966. PMLR, 2015.

Hang Le, Juan Pino, Changhan Wang, Jiatao Gu, Didier Schwab, and Laurent Besacier. Dual-decoder transformer for joint automatic speech recognition and multilingual speech translation. *arXiv preprint arXiv:2011.00747*, 2020.

Yuchen Liu, Hao Xiong, Zhongjun He, Jiajun Zhang, Hua Wu, Haifeng Wang, and Chengqing Zong. End-to-end speech translation with knowledge distillation. *arXiv preprint arXiv:1904.08075*, 2019.

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

G. Monge. Memoire sur la theorie des deblais et des remblais. *Mem. Math. Phys. Acad. Royale Sci.*, pp. 666–704, 1781. URL `https://cir.nii.ac.jp/crid/1572261550791499008`.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5206–5210. IEEE, 2015.

Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. Speechformer: Reducing information loss in direct speech translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 1698–1706, 2021.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Ngoc-Quan Pham, Jan Niehues, Thanh-Le Ha, and Alexander Waibel. Improving zero-shot translation with language-independent constraints. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pp. 13–23, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5202. URL `https://aclanthology.org/W19-5202`.

Juan Pino, Qiantong Xu, Xutai Ma, Mohammad Javad Dousti, and Yun Tang. Self-training for end-to-end speech translation. *arXiv preprint arXiv:2006.02490*, 2020.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084, 2019. URL `http://arxiv.org/abs/1908.10084`.

Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000.

Elizabeth Salesky, David Etter, and Matt Post. Robust open-vocabulary translation from visual text representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7235–7252, 2021.

K. Chandra Shekar, Maria Anisha Cross, and Vignesh Vasudevan. Optical character recognition and neural machine translation using deep learning techniques. In H. S. Saini, Rishi Sayal, A. Govardhan, and Rajkumar Buyya (eds.), *Innovations in Computer Science and Engineering*, pp. 277–283, Singapore, 2021. Springer Singapore. ISBN 978-981-33-4543-0.

Matthias Sperber and Matthias Paulik. Speech translation and the end-to-end promise: Taking stock of where we are. *arXiv preprint arXiv:2004.06358*, 2020.

Gabriel Synnaeve, Qiantong Xu, Jacob Kahn, Tatiana Likhomanenko, Edouard Grave, Vineel Pratap, Anuroop Sriram, Vitaliy Liptchinsky, and Ronan Collobert. End-to-end asr: from supervised to semi-supervised learning with modern architectures. *arXiv preprint arXiv:1911.08460*, 2019.

Yun Tang, Juan Pino, Xian Li, Changhan Wang, and Dmitriy Genzel. 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)*, pp. 4252–4261, 2021a.

Yun Tang, Juan Pino, Changhan Wang, Xutai Ma, and Dmitriy Genzel. A general multi-task learning framework to leverage text data for speech to text tasks. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6209–6213. IEEE, 2021b.

Zhengkun Tian, Jiangyan Yi, Jianhua Tao, Ye Bai, Shuai Zhang, and Zhengqi Wen. Spike-triggered non-autoregressive transformer for end-to-end speech recognition. *arXiv preprint arXiv:2005.07903*, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Hari Krishna Vydana, Martin Karafiát, Katerina Zmolikova, Lukáš Burget, and Honza Černocký. Jointly trained transformers models for spoken language translation. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7513–7517, 2021. doi: 10.1109/ICASSP39728.2021.9414159.

Chengyi Wang, Yu Wu, Shujie Liu, Zhenglu Yang, and Ming Zhou. Bridging the gap between pre-training and fine-tuning for end-to-end speech translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 9161–9168, 2020.

Ron J Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. Sequence-to-sequence models can directly translate foreign speech. *arXiv preprint arXiv:1703.08581*, 2017.

Yujia Xie, Hanjun Dai, Minshuo Chen, Bo Dai, Tuo Zhao, Hongyuan Zha, Wei Wei, and Tomas Pfister. Differentiable top-k with optimal transport. *Advances in Neural Information Processing Systems*, 33:20520–20531, 2020a.

Yujia Xie, Xiangfeng Wang, Ruijia Wang, and Hongyuan Zha. A fast proximal point method for computing exact wasserstein distance. In *Uncertainty in artificial intelligence*, pp. 433–453. PMLR, 2020b.

Chen Xu, Bojie Hu, Yanyang Li, Yuhao Zhang, Shen Huang, Qi Ju, Tong Xiao, and Jingbo Zhu. Stacked acoustic-and-textual encoding: Integrating the pre-trained models into speech translation encoders. In *ACL*, 2021.

Rong Ye, Mingxuan Wang, and Lei Li. End-to-end speech translation via cross-modal progressive training. In *Interspeech*, 2021a.

Rong Ye, Mingxuan Wang, and Lei Li. End-to-End Speech Translation via Cross-Modal Progressive Training. In *Proc. Interspeech 2021*, pp. 2267–2271, 2021b. doi: 10.21437/Interspeech.2021-1065.

Cheng Yi, Feng Wang, and Bo Xu. Ectc-docd: An end-to-end structure with ctc encoder and ocd decoder for speech recognition. In *INTERSPEECH*, pp. 4420–4424, 2019.

Sho Yokoi, Ryo Takahashi, Reina Akama, Jun Suzuki, and Kentaro Inui. Word rotator's distance. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2944–2960, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.236. URL `https://aclanthology.org/2020.emnlp-main.236`.

Xingshan Zeng, Liangyou Li, and Qun Liu. Realtrans: End-to-end simultaneous speech translation with convolutional weighted-shrinking transformer. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 2461–2474, 2021.

Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Renjie Zheng, Junkun Chen, Mingbo Ma, and Liang Huang. Fused acoustic and text encoding for multimodal bilingual pretraining and speech translation. In *International Conference on Machine Learning*, pp. 12736–12746. PMLR, 2021.

A Appendix

A.1 WRD based IPOT

Algorithm 1 shows the detailed implementation of IPOT, where $\mathrm{diag}(\boldsymbol{\delta})$ represents the diagonal matrix with δ_i as its i-th diagonal element, and $\odot$ and $\frac{(\cdot)}{(\cdot)}$ denote the element-wise matrix multiplication and division respectively. The algorithm outlines the forward-propagation steps only. Since each iteration of the algorithm only involves differentiable operators, we can utilize the automatic differentiation packages (*e.g.*, `PyTorch`) to back-propagate the gradients like an unrolled RNN. The corresponding implementation can refer to the submitted software.

Algorithm 1 WRD based IPOT

```
Input: Maximum iterations $T = 50$,
encoded sequences $\{\mathbf{t}_i^1\}_{i=1}^n, \{\mathbf{t}_j^2\}_{j=1}^m$.

Initialize $\mathbf{p}$ and $\mathbf{q}$ as Eq. (1).
Initialize $\mathbf{C}$ as Eq. (2).
Initialize $\mathbf{T} = \mathbf{1}_n \mathbf{1}_m^\top$.
$\boldsymbol{\sigma} = \frac{1}{m}\mathbf{1}_m$, $\mathbf{G}_{i,j} = e^{-\mathbf{C}_{i,j}}$.
for $t = 1, 2, \ldots, T$ do
    $\mathbf{Q} = \mathbf{G} \odot \mathbf{T}$
    $\boldsymbol{\delta} = \frac{\mathbf{p}}{\mathbf{Q}\boldsymbol{\sigma}}$, $\boldsymbol{\sigma} = \frac{\mathbf{q}}{\mathbf{Q}^\top \boldsymbol{\delta}}$
    $\mathbf{T} = \mathrm{diag}(\boldsymbol{\delta})\mathbf{Q}\mathrm{diag}(\boldsymbol{\sigma})$
end for
return $\langle \mathbf{C}, \mathbf{T} \rangle$
```

A.2 Differentiable Shrinking Mechanism

For the CTC distribution $\mathbf{d}_c \in \mathbb{R}^{|\mathbb{B}_s| \times l}$ where each column is a categorical distribution, we can derive the corresponding best CTC path via column-wise $\arg\max$. This step is not differentiable, but the path $\boldsymbol{\pi}$ is merely used to direct the shrinking.

$$\boldsymbol{\pi} = \arg\max \mathbf{d}_c \tag{9}$$

where $|\boldsymbol{\pi}| = l$. In general, $\boldsymbol{\pi}$ represents the token indexes and should be in the following format.

$$\boldsymbol{\pi} = (\epsilon, ..., \epsilon, \pi_1, ..., \pi_1, \epsilon, ..., \epsilon, \pi_2, ..., \pi_2, ..., ..., \pi_{\tilde{l}}, ..., \pi_{\tilde{l}}, \epsilon, ..., \epsilon,) \tag{10}$$

where $\epsilon, ..., \epsilon$ means 0 or more blank tokens, and $\pi_i, ..., \pi_i$ means 1 or more consecutive duplicated tokens. The two types of tokens are interleaved in the path.

First, we will average the columns in $\mathbf{d}_c$ that correspond to the same token, including blank token. The resulting averaged CTC distribution actually meets the requirement that each column is a valid distribution. Suppose the first k elements in $\boldsymbol{\pi}$ are the same token, the corresponding columns in the CTC distribution are $\mathbf{d}_c[:, :k]$. The averaged column is

$$\bar{\mathbf{d}}_c[:, :k] = \frac{1}{k}\sum_{i=1}^{k} \mathbf{d}_c[:, i] \in \mathbb{R}^{|\mathbb{B}_s|} \tag{11}$$

Because by definition $\forall i, \sum_{j=1}^{|\mathbb{B}_s|} \mathbf{d}_c[j, i] = 1$, then we have

$$\sum_{j=1}^{|\mathbb{B}_s|} \bar{\mathbf{d}}_c[j, :k] = \sum_{j=1}^{|\mathbb{B}_s|} \frac{1}{k} \sum_{i=1}^{k} \mathbf{d}_c[j, i] = \frac{1}{k} \sum_{i=1}^{k} \sum_{j=1}^{|\mathbb{B}_s|} \mathbf{d}_c[j, i] = 1 \tag{12}$$

Therefore, it means after the averaging, the averaged CTC distribution $\bar{\mathbf{d}}_c$ is still a column-wise distribution. In addition, this step only involves averaging operation, which is differentiable. The resulted $\bar{\mathbf{d}}_c$ corresponds to the following path.

$$\bar{\boldsymbol{\pi}} = (\epsilon, \pi_1, \epsilon, \pi_2, ..., \pi_{\tilde{l}}, \epsilon) \tag{13}$$

where ϵ means 0 or 1 blank token, and π_i means exactly 1 transcription token.

Second, we will remove the columns in $\bar{\mathbf{d}}_c$ that correspond to blank tokens. This step can be efficiently implemented via `gather` operation in most deep learning packages, which is also differentiable. Eventually, we will obtain the shrunk CTC distribution $\widetilde{\mathbf{d}}_c$, corresponding to the final path $\widetilde{\boldsymbol{\pi}} = (\pi_1, \pi_2, ..., \pi_{\tilde{l}})$.

Similarly, we can shrink the speech features $\mathbf{h}_s \in \mathbb{R}^{d \times l}$ by repeating above two steps and obtain the shrunk speech features $\widetilde{\mathbf{h}}_s$.

Then we can design the adapter as Eq. (5) $\mathbf{h}_a = \mathbf{E}_t \widetilde{\mathbf{d}}_c + \mathbf{W}_a^\top \widetilde{\mathbf{h}}_s$, which can be decoupled as an embedding layer and a linear layer. As discussed, the shrunk matrix $\widetilde{\mathbf{d}}_c$ and $\widetilde{\mathbf{h}}_s$ are both differentiable, allowing smooth model training. However, the embedding layer in the adapter cannot exactly match the real embedding layer in the semantic encoder which is actually an indexing operation. To better mimic the embedding layer, we can also define a one-hot vector $\hat{\mathbf{d}}_c$ derived from the index $\arg\max \widetilde{\mathbf{d}}_c$. Then the adapter output Eq. (5) becomes the following equation.

$$\mathbf{h}_a = \mathbf{E}_t \hat{\mathbf{d}}_c + \mathbf{W}_a^\top \widetilde{\mathbf{h}}_s \tag{14}$$

In this way, the adapter is not fully differentiable due to the first term. To solve this problem, we can implement the straight through trick Bengio et al. (2013) as follows.

$$\hat{\mathbf{d}}_c = \text{stop\_grad}(\hat{\mathbf{d}}_c - \widetilde{\mathbf{d}}_c) + \widetilde{\mathbf{d}}_c \tag{15}$$

A.2.1 Batch Level Implementation

Our implementation is based on fairseq (Ott et al., 2019). The main code is located in the folder `examples/dcm` of the submitted software zip file, including the differentiable shrinking mechanism (`examples/dcm/models/s2t_dcm.py`) and the differentiable WRD OT loss function (`examples/dcm/criterions/text_guide_cross_entropy_with_ctc.py`).

A.3 Training Details of ST

A.3.1 Additional Loss Functions

The CTC loss can be defined as follows.

$$\mathcal{L}_{CTC}(\mathbf{x}_s, \mathbf{h}_s) = -\log \sum_{\boldsymbol{\pi} \in \mathbb{A}(\mathbf{x}_s)} P(\boldsymbol{\pi}|\mathbf{h}_s), \tag{16}$$

where $P(\boldsymbol{\pi}|\mathbf{h}_s) = \sum_t P(\pi_t|\mathbf{h}_s)$ defines the probability of the CTC path $\boldsymbol{\pi}$, and $\mathbb{A}(\mathbf{x}_s)$ represents all possible paths that result in the transcription $\mathbf{x}_s$. The definition the KD loss function is as follows.

$$\mathcal{L}_{KD} = -\sum_{t=1}^{|\mathbf{y}|} \sum_{n=1}^{|\mathbb{B}_t|} P(y_t|\mathbf{x}; \boldsymbol{\theta}_{MT}) \times \log P(y_t|\mathbf{z}; \boldsymbol{\theta}_{ST}) \tag{17}$$

where $P(\cdot, \boldsymbol{\theta}_{ST})$ is the student model and and $P(\cdot, \boldsymbol{\theta}_{MT})$ is the teacher model.

A.3.2 Hyper-parameters

All training and evaluation hyper-parameters can refer to the training bash scripts (`*.sh`) in the submitted code, including the optimizer, learning rate, batch size, beam size (5 by default), etc. Particularly, we use SacreBLEU in fairseq as evaluation metrics.

A.3.3 Datasets in Phase 3

For the third phase supervised ST training, we have multiple tasks in the final objective. For the ST task $\mathcal{L}_{ST}$, some previous works that may leverage the MT model and the Librispeech transcription to construct pseudo translation sentences. However, we only use the audio and translation pairs from MuST-C. For the ASR task $\mathcal{L}_{CTC}$, we only use the audios and transcriptions from MuST-C. For the MT task $\mathcal{L}_{MT}$, we optimize it on both the MuST-C parallel corpus and WMT data, making the decoder a better language model. En-De WMT only has 4.5M sentence pairs and the entire training is still manageable. However, for En-Fr/Es, optimizing the large end-to-end ST model with huge amount trainable parameters will be cumbersome because the size of WMT data overwhelmingly slows down the training. Therefore, we randomly sample 10M corpus from the original WMT En-Fr/Es data to train the final supervised loss. For the knowledge distillation loss $\mathcal{L}_{KD}$, we use the audio-transcription-translation triplet data from MuST-C.

A.4 More Experimental Results

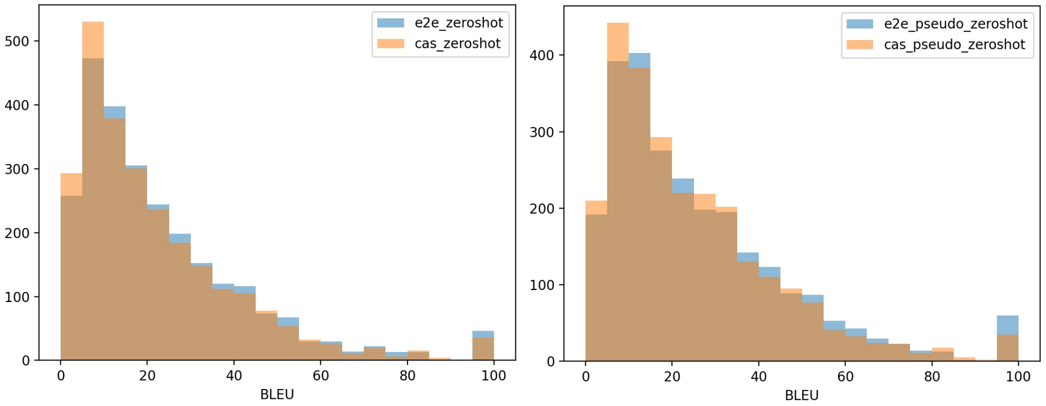


Figure 5: BLEU evaluation of each sentence on En-De v2 tst-COMMON. For **zero-shot** setting, cascade system has more sentences in BLEU interval [0, 10], while almost in all other intervals, our proposed model has more sentences. For **pseudo zero-shot** setting, cascade system tends to have more examples on low BLEU intervals, and our approach has more examples on higher BLEU intervals.

Supervised Training Loss				**En-De v2**
ST	CTC†	WRD	KD	common
✓	✓	✓	✓	29.22
✓		✓	✓	28.94
✓	✓		✓	28.86
✓	✓	✓		28.36

Table 4: Ablation study on the supervised loss Eq. (8). All models are fine-tuned from the zero-shot ST model with BLEU 24.00 in Table 2. †The CTC loss cannot be directly removed, because our shrinking adaptor depends on the CTC results. So we freeze the pre-trained acoustic encoder including CTC layer. The results indicates the acoustic encoder is well-trained in zero-shot phrase, and the freezing has almost no impact when fine-tuning.

Model Module			**En-De v2**
Acoustic Encoder†	Adaptor	Semantic Enc/Dec*	common
trainable	trainable	trainable	29.22
trainable	trainable	frozen	25.14
frozen	trainable	trainable	28.94

Table 5: Ablation study on the model parameters for supervised training phrase. All models are fine-tuned from the zero-shot ST model with BLEU 24.00 in Table 2. †Acoustic encoder includes the CTC layer. *The result becomes much worse if semantic encoder/decoer are frozen. The main reason we hypothesize is that since the NMT teacher is frozen, the in-domain MT data is not used. So it's difficult for the NMT decoder to adapt for the supervised ST data, i.e., the decoder is not a good language model.

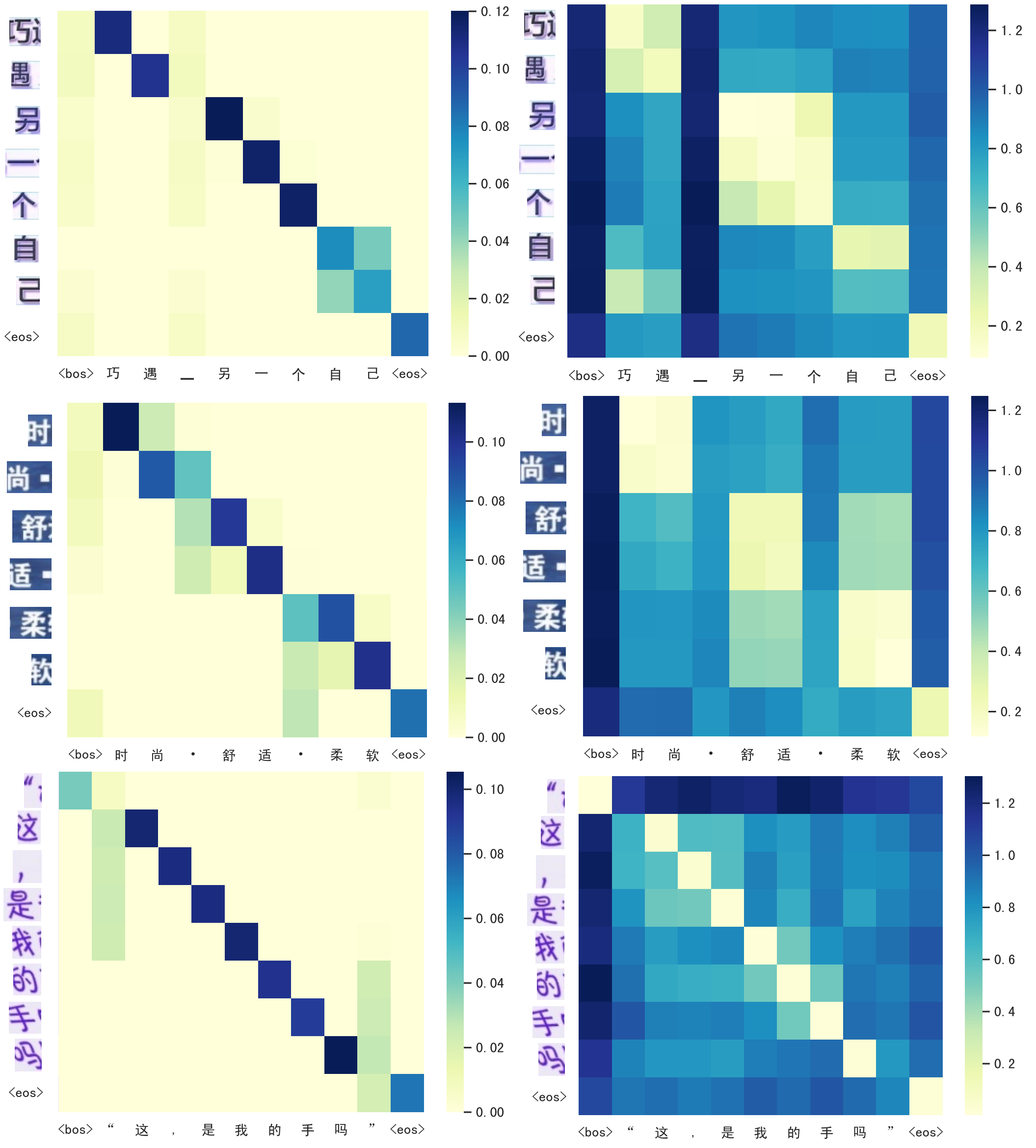


Figure 6: **Left Column**: visualization of transport plan $\mathbf{T}^*$. **Right Column**: visualization of cost matrix $\mathbf{C}$. The WRD with OT solver can align cross-modal features with different lengths. y-axis represents the shrunk features, and x-axis represents the transcription features. Since the actual shrunk tokens only represents a 4-pixel wide part of an image, we cut images along blank tokens as a schematic representation. The real shrunk tokens usually in the middle of the images on y-axis. For the cost matrix, the smaller elements are mainly distributed on the diagonal block regions. It could be the incorrect shrinking segments sometimes aligning with more than one characters. For transport plan matrix, the larger elements are mainly distributed on the diagonal. In this way, their products will remain small.

SRC: If you have something to give, give it **now**.
REF: Wenn Sie etwas zu geben haben, geben Sie es **jetzt**.
ASR: If you have something to give, give it **no**.
Cascade: Wenn Sie etwas zu geben haben, geben Sie es **nein**.
E2E: Wenn Sie etwas zu geben, geben Sie es **jetzt**.

SRC: So get in the game. **Save** the shoes.
REF: Also legen Sie los; **retten** Sie die Schuhe.
ASR: So get in the game, **Sa** the shoes**..**
Cascade: Also im Spiel, **Sa** die Schuhe**..**
E2E: So bekommen Sie im Spiel, **speichern** Sie die Schuhe.

SRC: Yet, in this land of violence and chaos, you can hear hidden laughter **swaying** the trees.
REF: Und doch, in diesem Land von Gewalt und Chaos kann man ein verborgenes Lachen hören, dass die Bäume **erschüttert**.
ASR: Yet, in this land, of violence and chaoss, you can hear hidden laughter **sewing** the trees.
Cascade: Doch in diesem Land, in dem Gewalt und Chaos herrscht, kann man verborgenes Lachen hören, das die Bäume **näht**.
E2E: Doch in diesem Land, in dem Gewalt und Chaos herrscht, kann man verborgenes Lachen hören, das die Bäume **schwingt**.

SRC: And when you watch bonobo **play**, you're seeing the very evolutionary roots of human laughter, dance and ritual.
REF: Und wenn man Bonobos beim **Spiel** beobachtet, sieht man die evolutionären Ursprünge menschlichen Lachens, Tanzes und von Ritualen.
ASR: And when you watch a below **airplane**, you're seeing the very evolutionary roots of human laughter, dance and ritual.
Cascade: Und wenn man ein **Flugzeug** unter sich sieht, sieht man die sehr evolutionären Wurzeln des menschlichen Lachens, Tanzens und Rituals.
E2E: Und wenn man unter dem **Spiel** zusieht, sieht man die sehr evolutionäre Wurzel menschlichen Lachens, Tanz und Ritual.

SRC: Play is the glue that **binds** us together.
REF: Spiel ist der Kitt, der uns beieinanderhält.
ASR: Play is the glue that **bis** us together.
Cascade: Spielen ist der Leim, der uns **bis** zusammen.
E2E: Spielen ist der Klebstoff, der uns zusammen **brennen**.

Figure 7: Case Study of zero-shot ST. From the first 50 examples in En-De v2 tst-COMMON testset, we select 5 examples that have critical ASR errors but have tiny BLEU difference between cascade and end-to-end systems. For cascade system, the ASR errors (in red font) will be firmly passed into the MT model, leading to obvious (non-sense) mistakes in the translation. Instead, our end-to-end zero-shot ST with the adaptor alignment can sometimes prevent such errors and translate with synonyms. For the 5th example, the end-to-end model also made an error where "brennen" means "burn". We guess the model is confused by the representations (including both speech and semantic information) of "binds" and "burns". However, "burns us together" is more fluent than "bis us together" and closer to the original the semantic meaning "binds us together".