UnitY: Two-pass Direct Speech-to-speech Translation with Discrete Units

Hirofumi Inaguma♡**, Sravya Popuri**♡**, Ilia Kulikov**♡**, Peng-Jen Chen**♡**,**
Changhan Wang♡**, Yu-An Chung**♡**, Yun Tang**♡**,**
Ann Lee♡**, Shinji Watanabe**♣**, Juan Pino**♡
FAIR, Meta AI♡, Carnegie Mellon University♣
{hirofumii,juancarabina}@meta.com

Abstract

Direct speech-to-speech translation (S2ST), in which all components can be optimized jointly, is advantageous over cascaded approaches to achieve fast inference with a simplified pipeline. We present a novel two-pass direct S2ST architecture, *U*nitY, which first generates textual representations and predicts discrete acoustic units subsequently. We enhance the model performance by subword prediction in the first-pass decoder, advanced two-pass decoder architecture design and search strategy, and better training regularization. To leverage large amounts of unlabeled text data, we pre-train the first-pass text decoder based on the self-supervised denoising auto-encoding task. Experimental evaluations on benchmark datasets at various data scales demonstrate that UnitY outperforms a single-pass speech-to-unit translation model by 2.5-4.2 ASR-BLEU with 2.83× decoding speed-up. We show that the proposed methods boost the performance even when predicting spectrogram in the second pass. However, predicting discrete units achieves 2.51× decoding speed-up compared to that case.

1 Introduction

Automatic speech translation to another language is an indispensable technology for international communications, with the spread of social media and virtual communications nowadays. A traditional approach of speech-to-speech translation (S2ST) is to cascade automatic speech recognition (ASR), machine translation (MT), and text-to-speech (TTS) components, each of which is optimized separately on different data (Lavie et al., 1997; Nakamura et al., 2006; Wahlster, 2013). With the emergence of sequence-to-sequence models (Sutskever et al., 2014; Cho et al., 2014; Bahdanau et al., 2015), however, it is getting prevailing to adopt a direct approach[1]. This approach consists in translating input speech into the other language based on a single architecture with fewer components than the cascaded systems (Jia et al., 2019b; Tjandra et al., 2019; Zhang et al., 2021). The direct approach is attractive for building a low-latency system with a simplified pipeline, thus reducing developing costs. However, direct S2ST models suffer from poor performance due to data scarcity, similar to direct speech-to-text translation (S2TT) models (Bérard et al., 2016). In the field of S2TT, data shortage has been addressed by leveraging pre-training (Bérard et al., 2018; Wang et al., 2021c; Tang et al., 2022), multi-task learning (Weiss et al., 2017; Tang et al., 2021), pseudo labeling (Jia et al., 2019a; Pino et al., 2020), knowledge distillation (Liu et al., 2019; Inaguma et al., 2021b). Consequently, the translation quality of direct S2TT models is approaching that of cascaded S2TT models (Ansari et al., 2020; Anastasopoulos et al., 2021). These techniques have also shown the effectiveness for direct S2ST models and led to a decent performance (Kano et al., 2021; Dong et al., 2022; Jia et al., 2022a; Popuri et al., 2022).

Recent works (Lee et al., 2022a,b) propose to model discrete acoustic units, extracted from HuBERT (Hsu et al., 2021), instead of a continuous speech signal that enables usage of a standard cross-entropy loss during training. This speech-to-unit translation (S2UT) model significantly shortens the target sequence length and thus makes training and inference more efficient. The discrete units are directly converted to the waveform with a unit-based neural vocoder (Polyak et al., 2021) bypassing spectrogram representation. On the other hand, Translatotron2 (Jia et al., 2022b) decomposes the target representations into linguistic and acoustic counterparts explicitly. The former predicts a phoneme

[1] Lee et al. (2022a) defines a direct S2ST model as a model that does not use intermediate text representations while Jia et al. (2022b) defines it as a model that directly predicts the target spectrogram. In this paper, we use a more general definition that the entire architecture is optimized jointly and the translation is conducted in a more direct way. We do not include a vocoder in the training pipeline of all direct models.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 15655–15680
July 9-14, 2023

sequence first, and the latter synthesizes the target spectrogram conditioned on the continuous representation of the linguistic part.

This paper presents a novel two-pass direct S2ST architecture, dubbed *UnitY*, which takes the best of both worlds of the S2UT model and Translatotron2. Unlike Translatotron2, UnitY models linguistic sequences using subwords (*first pass*) instead of phonemes, and it models speech as a discrete sequence of acoustic units (*second pass*). To achieve better translation quality and decoding efficiency, UnitY consists of a deep text decoder and a shallow unit decoder and enables better generalization to the first-pass decoder. We further introduce a text-to-unit (T2U) encoder between the two decoders to bridge the gap between textual and acoustic representations. Following the success of large-scale pre-training, we leverage unlabeled text effectively to pre-train the first pass text decoder with multilingual BART (mBART) (Liu et al., 2020) at the subword level.

Extensive experiments show the superiority of the UnitY S2ST system measured by both translation quality and runtime efficiency. First, UnitY achieves 4.2, 3.7, and 2.5 ASR-BLEU improvements over the S2UT model on the Fisher Es→En (Post et al., 2013), CVSS-C (Jia et al., 2022c), and multi-domain En↔Es (Popuri et al., 2022) corpora, respectively. The improvement holds even with high-resource data and pre-training. In addition, our proposed design improves Translatotron2 as well, indicating its versatility for two-pass direct S2ST architectures regardless of the choice of the target. Second, UnitY achieves 2.83× and 2.51× decoding speed-ups over the S2UT and improved Translatotron2 models, respectively. A combination of the aforementioned improvements suggests the UnitY design as a starting point for further improvements in direct S2ST. [2]

2 UnitY

In this section, we propose *UnitY*, a two-pass direct S2ST model that generates subwords and discrete acoustic units subsequently. Hereafter, we refer to discrete acoustic units as discrete units for brevity. Let X denote a source speech input, and $Y = (y_1, \ldots, y_M)$ and $U = (u_1, \ldots, u_L)$ be the corresponding reference text translation and discrete unit sequences in the target language, respectively. Note that there is no duration information

[2] Code will be available upon the paper acceptance.

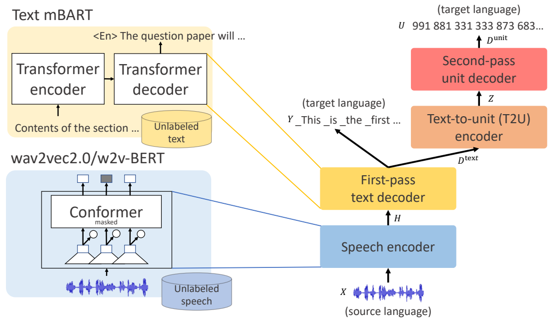


Figure 1: Model architecture of UnitY

for each discrete unit in U, because consecutive units are collapsed (Lee et al., 2022a).

2.1 Architecture

The overall architecture of UnitY is shown in Figure 1. UnitY consists of four modules: speech encoder, first-pass text decoder, text-to-unit (T2U) encoder, and second-pass unit decoder. We build the speech encoder based on Conformer (Gulati et al., 2020), which augments Transformer (Vaswani et al., 2017) with a convolution module, while implementing the rest three modules based on Transformer. UnitY has five major architecture modifications from Translatotron2 (Jia et al., 2022b), (1) generating subwords instead of phonemes in the first pass, (2) generating discrete units instead of spectrograms in the second pass to bypass duration modeling, (3) replacing Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) layers with Transformer layers in both decoders, (4) introducing a T2U encoder between the two decoders, and (5) assigning more model capacities to the first pass.

First-pass text decoder The first-pass text decoder `TDec` generates a sequence of subwords Y autoregressively by attending the speech encoder output H. The training objective of the first pass is to minimize the direct S2TT loss $\mathcal{L}_{\text{s2t}}$ as:

$$
\begin{aligned}
\mathcal{L}_{\text{s2t}}(Y|X) &= -\frac{1}{M}\sum_{i=1}^{M} \log P_{\text{s2t}}(y_i|X, Y_{<i}) \\
&= -\frac{1}{M}\sum_{i=1}^{M} \log P_{\text{s2t}}(y_i|D_i^{\text{text}}) \\
D_i^{\text{text}} &= \texttt{TDec}(H, Y_{<i}),
\end{aligned}
$$

where $D_i^{\text{text}} \in \mathbb{R}^{d_{\text{model}}}$ is the i-th continuous decoder state right before projecting it to the logit. We consider that D^{text} contains rich acoustic information in addition to contextual information thanks to multiple multi-head cross-attention over H.

There are five advantages of generating subwords instead of phonemes. First, the sequence length is considerably reduced, leading to better training and inference efficiencies (Cherry et al., 2018). Second, using large vocabularies improves the translation quality of the first pass (Gowda and May, 2020). Third, the text output helps the audience understand the translation content while listening to the audio. Fourth, our approach can easily scale to more target languages, as it is unnecessary to prepare separate grapheme-to-phoneme (G2P) models for each target language. Last, readable text can be generated without any complicated post-processing such as WFST (Mohri et al., 2002; Bahdanau et al., 2016).

T2U encoder A bidirectional T2U encoder `T2UEnc` transforms the continuous states of the first-pass decoder $D^{\text{text}} \in \mathbb{R}^{M \times d_{\text{model}}}$ into $Z \in \mathbb{R}^{M \times d_{\text{model}}}$ as $Z = \mathtt{T2UEnc}(D^{\text{text}})$. The T2U encoder bridges the gap in representations between text and unit decoders without changing the sequence length.

Second-pass unit decoder The second-pass unit decoder `UDec` generates a sequence of discrete units U autoregressively by attending to only the T2U encoder output Z. The training objective of the second pass is to minimize $\mathcal{L}_{\text{s2u}}$ similar to the S2UT task while being conditioned on Y as:

$$
\begin{aligned}
\mathcal{L}_{\text{s2u}}(U|X,Y) &= -\frac{1}{L}\sum_{i=1}^{L} \log P_{\text{s2u}}(u_i|X,Y,U_{<i}) \\
&= -\frac{1}{L}\sum_{i=1}^{L} \log P_{\text{s2u}}(u_i|D_i^{\text{unit}}) \\
D_i^{\text{unit}} = \mathtt{UDec}(Z, U_{<i}) &= \mathtt{UDec}(H, Y, U_{<i}),
\end{aligned}
$$

where $D_i^{\text{unit}} \in \mathbb{R}^{d_{\text{model}}}$ is the i-th continuous decoder state right before projecting it to the logit. The unit decoder does not attend to H to synchronize the text and unit outputs, similar to the motivation in (Jia et al., 2022b). In other words, we do not expect that the second-pass decoder corrects translation errors from the first-pass decoder.[3] Once the unit generation finishes, a separate unit-based vocoder (Polyak et al., 2021) converts the discrete units to the waveform with duration prediction of each discrete unit (Lee et al., 2022a). The total training objective of UnitY, $\mathcal{L}_{\text{total}}$, is formulated as follows:

$$
\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{s2u}}(U|X,Y) + w_{\text{s2t}}\mathcal{L}_{\text{s2t}}(Y|X), \quad (1)
$$

where w_{s2t} is a weight for the S2TT loss.

2.2 Text decoder pre-training

Similar to ASR and S2TT studies (Baevski et al., 2020; Li et al., 2021), S2ST models also benefit from self-supervised pre-training (Jia et al., 2022a; Popuri et al., 2022), especially for the speech encoder. In addition to the speech encoder pre-training with wav2vec2.0 (Baevski et al., 2020), Popuri et al. (2022) initializes the unit decoder of the single-pass S2UT model with a unit-based mBART (u-mBART), an encoder-decoder model pre-trained with discrete units converted from a large amount of unlabeled speech data. However, unlabeled text data cannot be leveraged for the single-pass decoder pre-training, although it is more accessible in many written languages.

To fully leverage the unlabeled text data, we initialize the first-pass decoder of UnitY with a text-based mBART (t-mBART) pre-trained with unlabeled text data. Following Li et al. (2021); Popuri et al. (2022), we freeze parameters in the feed-forward network (FFN) of the text decoder during S2ST fine-tuning. We initialize the T2U encoder and second-pass unit decoder randomly.

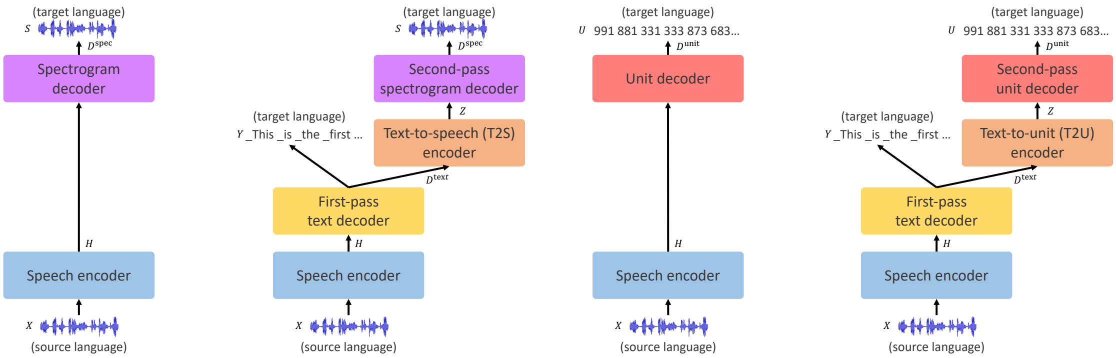
2.3 Search algorithm

During inference, we perform two-pass beam search decoding. First, we find the most probable text hypothesis $\hat{Y}$ in the first-pass decoder using beam search with a beam size of B_{1st}. We then feed continuous decoder states D^{text} corresponding to $\hat{Y}$ to the T2U encoder. Next, we generate the most probable discrete unit sequence $\hat{U}$ in the second-pass decoder by another beam search with a beam size of B_{2nd}. Finally, $\hat{U}$ is taken as input to a separate unit-based vocoder to generate the waveform. We find it more effective to assign a larger beam size to the first pass, *i.e.*, $B_{\text{1st}} > B_{\text{2nd}}$, because there is more diversity among beam candidates than the second pass. The computation time is also reduced since the sequence length of text is much shorter than that of discrete units. Therefore, we use $B_{\text{2nd}} = 1$ unless otherwise noted. We present the pseudo algorithm in Appendix A.

2.4 Deep-shallow two-pass decoders

By increasing the number of layers, we assign more model capacities to the first-pass decoder than the

[3] We also investigate attending to the speech encoder output with an additional cross-attention, but it does not lead to an improvement in ASR-BLEU. We discuss this in §5.1



(a) S2SpecT (b) S2SpecT2 (c) S2UT (d) UnitY

Figure 2: Direct S2ST architectures

second-pass decoder. We refer to this as *deep-shallow two-pass decoders*. This capacity assignment improves translation quality and inference efficiency simultaneously because of a shorter sequence length in the first pass. A practical capacity assignment for the MT task is studied in Kasai et al. (2021) by trading the number of layers between the encoder and decoder. In this work, we focus on the two-pass decoders for the S2ST task.

3 Experimental setting

3.1 Data

We use three datasets: Fisher Es→En (Post et al., 2013) (170 hours), CVSS-C (Jia et al., 2022c) (547 hours), and mutli-domain En↔Es (Popuri et al., 2022) (20k hours for En→Es, 14k hours for Es→En) corpora. We combine all 21 language directions to English in the CVSS-C corpus to train a single X-to-En multilingual model. The En→Es part in the multi-domain corpora consists of Europarl-ST (Iranzo-Sánchez et al., 2020), Must-C (Di Gangi et al., 2019), TEDLIUM3 (Rousseau et al., 2012), Librispeech (Panayotov et al., 2015), and Common Voice (Ardila et al., 2020). The Es→En part consists of CoVoST2 (Wang et al., 2021b), Europarl-ST, and mTEDx (Elizabeth et al., 2021), Common Voice, and multilingual Librispeech (MLS) (Pratap et al., 2020). More details are described in Appendix D.

3.2 Pre-processing

We follow the same pre-processing as (Lee et al., 2022a,b; Popuri et al., 2022) for acoustic feature extraction, discrete unit extraction, and text normalization. We also discarded over-generated target speech/unit by TTS/T2U models. More details are described in Appendix E.

3.3 Pre-training

We use the same En/Es wav2vec2.0 and En-Es u-mBART models as Popuri et al. (2022). We train a multilingual w2v-BERT (Chung et al., 2021) model trained on 51 languages with the same setting as Jia et al. (2022a). For text decoder pre-training, we use the same En-Es and 50-language t-mBART models as Wang et al. (2022) and Tang et al. (2020), respectively. We describe the training details and list model URLs in Appendix F.

3.4 Baseline

We build two cascaded S2ST systems and four direct S2ST systems. All speech encoders are based on Conformer. When pre-training the speech encoder of direct S2ST systems with wav2vec2.0/w2v-BERT, we pre-train ASR and S2TT models in the cascaded systems with the same wav2vec2.0/w2v-BERT for a fair comparison. We also pre-train the text decoder of the ASR and S2TT models with t-mBART in that case.

Cascaded (ASR→MT→TTS) We combine a Conformer ASR, a Transformer MT, and a Transformer TTS model. We set the reduction factor of TTS models to 4.

Cascaded (S2TT→TTS) We combine a Conformer direct S2TT model and a Transformer TTS model.

S2SpecT We build a direct S2ST model that predicts spectrogram with a single Transformer decoder, similar to Lee et al. (2022a) (Figure 2a). We refer to it as S2SpecT hereafter. We set the reduction factor of the spectrogram decoder to 3.

S2SpecT2 We train S2SpecT2, an improved version of Translatotron2, by enhancing the architecture and training with the proposed methods for UnitY. First, we replace phoneme targets with subwords in the first pass (Figure 2b). Second, we replace LSTM decoders with Transformer decoders. Third, we introduce an additional text-to-spectrogram (T2S) encoder between text and spectrogram decoders. The second-pass decoder attends to the T2S encoder output only. Fourth, we use an autoregressive Transformer decoder instead of a non-attentive Tacotron (NAT) (Shen et al., 2020) for the second-pass decoder. Last, we apply R-Drop to the first-pass decoder. We use the same reduction factor as S2SpecT.

S2UT We train a direct S2ST model that predicts discrete units with a single Transformer decoder (Lee et al., 2022a) (Figure 2c).

3.5 Architecture

Let N_{1st}, N_{2nd}, and N_{t2u} be the depth of the first-pass decoder, second-pass decoder, and T2U encoder of UnitY, respectively. We set $(N_{\text{1st}}, N_{\text{2nd}}, N_{\text{t2u}})$ to $(4, 2, 2)$ on Fisher and CVSS-C. On the multi-domain corpus, we use $(12, 2, 2)$ when pre-training the first-pass decoder with t-mBART. Otherwise, we use $(6, 6, 2)$. We describe the other configurations in Appendix G.

3.6 Training

We apply R-Drop (Wu et al., 2021) regularization to all tasks that predict discrete symbols, except the MT task. The training objective of each model with R-Drop is defined in Appendix C. We implement our models based on the Fairseq toolkit (Ott et al., 2019; Wang et al., 2020). The detailed training hyperparameters are described in Appendix H.

3.7 Decoding

We use a beam width of 10 for ASR, S2TT, and S2UT models. For UnitY, we set B_{1st} and B_{2nd} to 10 and 1, respectively. We use a beam width of 10 for the first-pass decoder in S2SpecT2.

3.8 Vocoder

We use a HiFi-GAN vocoder (Kong et al., 2020) to convert spectrograms to the waveform for TTS and direct speech-to-spectrogram models. We use a unit-based HiFi-GAN vocoder (Polyak et al., 2021) to convert discrete units to the waveform for direct speech-to-unit models. Both the vocoders are trained separately.

ID	Model	ASR-BLEU (↑) Avg.	High	Mid	Low
B0	Synthetic target$^\diamondsuit$	91.1	88.4	89.5	93.0
Cascaded systems					
B1	S2TT → TTS$^\diamondsuit$	10.6	28.8	15.5	2.4
B2	+ ASR pre-training	12.7	30.7	18.3	4.4
B3	S2TT → TTS	7.8	18.2	11.9	2.6
B4	+ w2v-BERT + t-mBART	14.9	21.1	18.2	11.5
Direct speech-to-spectrogram systems					
B5	Translatotron$^\diamondsuit$	3.4	11.9	3.5	0.3
B6	S2SpecT	7.6	21.8	10.6	1.5
B7	+ S2TT pre-training	9.6	23.9	13.8	3.2
B8	+ w2v-BERT	16.6	30.5	21.9	9.8
B9	Translatotron2$^\diamondsuit$	8.7	25.4	12.6	1.5
B10	+ Transformer decoder$^\spadesuit$	10.1	26.9	14.2	2.8
B11	+ S2TT pre-training$^\diamondsuit$	12.0	29.7	16.6	4.2
B12	+ w2v-BERT$^\spadesuit$	17.9	32.5	22.9	10.9
B13	+ mSLAM$^\spadesuit$	19.3	33.2	24.6	12.5
B14	++ TTS augmentation$^\spadesuit$	22.0	33.5	25.8	16.5
B15	S2SpecT2	11.3	29.1	16.9	3.1
B16	+ S2TT pre-training	13.1	29.8	18.8	5.2
B17	+ w2v-BERT + t-mBART	18.6	32.1	24.7	11.6
Direct speech-to-unit systems					
B18	S2UT	9.1	25.9	12.9	1.9
B19	+ S2TT pre-training	11.4	27.2	16.4	4.0
B20	+ w2v-BERT + u-mBART	20.8	31.6	25.4	15.4
B21	UnitY	12.0	29.0	17.8	4.0
B22	+ S2TT pre-training	13.0	30.4	18.7	4.8
B23	+ w2v-BERT + t-mBART	**24.5**	**34.6**	**28.9**	**19.3**

Table 1: ASR-BLEU on CVSS-C corpus. $^\diamondsuit$Results from (Jia et al., 2022c), $^\spadesuit$Results from (Jia et al., 2022a). We use the S2TT model in B3 for S2TT pre-training. t-mBART and u-mBART stand for text-based mBART and unit-based mBART, respectively. All w2v-BERT and mSLAM encoders have 0.6B parameters.

3.9 Evaluation

Following Lee et al. (2022a), we use a pre-trained ASR model to transcribe the generated target speech and calculate BLEU scores (Papineni et al., 2002), referred to as ASR-BLEU. The ASR model is fine-tuned from a wav2vec2.0 with the connectionist temporal classification (CTC) objective (Graves et al., 2006). We use the sacrebleu toolkit (Post, 2018) to calculate the BLEU scores.

4 Experimental results

In this section, we present the experimental results on three corpora. We study various modeling choices from the perspective of target representation (spectrogram v.s. discrete unit) and decoder architectures (single pass v.s. two pass) in supervised and semi-supervised settings. We also benchmark the decoding efficiency of direct S2ST models.

ID	Model	ASR-BLEU (↑)						
		En→Es			Es→En			
		Europarl-ST	MuST-C	Avg.	CoVoST-2	Europarl-ST	mTEDx	Avg.
Cascaded systems								
C1	ASR→MT→TTS$^\diamondsuit$	28.8	34.2	31.5	33.8	29.1	32.4	31.5
C1′	ASR→MT→TTS	36.8	30.8	33.8	32.9	34.2	30.3	32.5
C2	S2TT→TTS$^\diamondsuit$	32.6	30.1	31.4	28.4	23.6	21.5	24.5
C2′	S2TT→TTS	36.4	33.4	34.9	37.2	34.0	32.5	34.6
Direct speech-to-spectrogram systems								
C3	S2SpecT2 (6L→6L)	35.6	33.5	34.6	37.0	23.4	31.3	30.6
C4	+ t-mBART (12L→6L)	**36.9**	**34.3**	**35.6**	**37.2**	23.7	31.7	30.9
Direct speech-to-unit systems								
C5	S2UT + u-mBART$^\diamondsuit$	32.7	32.1	32.4	33.5	28.6	29.1	30.4
C5′	S2UT + u-mBART	33.5	33.3	33.4	34.5	29.9	29.9	31.4
C6	UnitY (6L→6L)	35.1	33.7	34.4	35.4	30.8	31.3	32.5
C7	+ t-mBART (12L→2L)	**35.3**	**34.1**	**34.7**	**36.4**	**33.1**	**32.2**	**33.9**

Table 2: ASR-BLEU on multi-domain En↔Es. $^\diamondsuit$Results from (Popuri et al., 2022). The encoder in all the models is pre-trained with wav2vec2.0. t-mBART and u-mBART stand for text-based mBART and unit-based mBART, respectively. $N_{1\text{st}}$L→ $N_{2\text{nd}}$L stands for an $N_{1\text{st}}$-layer first-pass decoder with an $N_{2\text{nd}}$-layer second-pass decoder.

4.1 CVSS-C

The results on CVSS-C are listed in Table 1. We first compared four direct systems trained from scratch (B6, B15, B18, B21), and UnitY (B21) achieved the best ASR-BLEU. The encoder pre-training with the S2TT model in the cascaded system (B3) improved ASR-BLEU of all the direct S2ST models (B7, B16, B19, B22), similar to Jia et al. (2022c).[4] In this case, S2SpecT2 (B16) also achieved similar translation quality to UnitY (B22). Still, UnitY outperformed the S2UT model (B19) by 1.6 ASR-BLEU on average, indicating that the two-pass decoding was the main factor of the improvements. S2SpecT2 (B16) outperformed Translatotron2 (Jia et al., 2022b) (B11) by 1.1 ASR-BLEU on average, from which we can confirm that parts of the proposed methods can generalize to the other S2ST architecture.[5] Compared to the best cascaded system (B2), the two-pass models (B16, B19) showed better translation quality.

We also pre-trained the speech encoder of all models with multilingual w2v-BERT, the first-pass text decoder of two-pass models with text-based mBART (t-mBART), and the decoder of the S2UT model with unit-based mBART (u-mBART), respectively. Among them (B4, B8, B12, B20, B23), UnitY (B23) showed the best ASR-BLEU. UnitY still outperformed Translatotron2 with a joint speech-text pre-training with mSLAM (Bapna et al., 2022) (B13) and TTS augmentation (B14) by 5.2 and 2.5 ASR-BLEU on average, respectively. The full results in each language direction are presented in Appendix I.

4.2 Multi-domain En↔Es

We present results on the multi-domain corpora (Popuri et al., 2022) in Table 2. C1′, C2′, and C5′ are our improved models of C1, C2, and C5, respectively.[6] We observed that UnitY with first-pass decoder pre-training with t-mBART (C7) improved the S2UT model with decoder pre-training with u-mBART (C5′) by 1.3 and 2.5 ASR-BLEU on average in En→Es and Es→En, respectively. This confirms the effectiveness of the two-pass modeling in the high-resource scenario. Furthermore, UnitY without decoder pre-training (C6) already outperformed C5′ and degraded from C7 only slightly. Comparing UnitY and S2SpecT2, we cannot spot a clear winner. UnitY outperformed S2SpecT2 in Es→En on Europarl-ST and mTEDx, but S2SpecT2 performed better in En→Es. The proposed text decoder pre-training helped S2SpecT2 performance too, especially in En→Es (C4). Finally, we also confirmed that UnitY approached the performance of a strong cascaded system and even outperformed it on Must-C.

[4] Unlike Jia et al. (2022c), we trained the S2TT model with an auxiliary ASR task from scratch instead of pre-training the encoder with that of an ASR model.

[5] B9 predicts phonemes while B15 predicts subwords in the first pass.

[6] We improved C1 and C2 by R-Drop and better hyperparameters. C5 was also improved by hyperparameter tuning and checkpoint averaging.

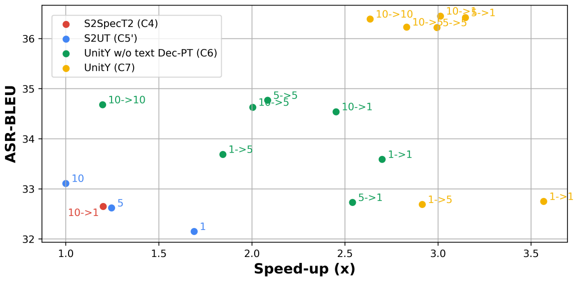


Figure 3: Runtime of direct S2ST models on multi-domain Es→En corpus. X→Y at each data point represents the beam width in each decoder pass.

ID	Model	(ASR-)BLEU (↑)	
		Text	Speech
D1	S2SpecT2	**35.0**	**30.8**
D2	+ w/o T2S encoder	34.9	25.0
D3	+ w/o R-Drop	34.8	30.3
D5	UnitY	**38.3**	**33.2**
D6	+ w/o T2U encoder	38.1	30.7
D7	+ w/o R-Drop	37.7	32.1
D8	+ Cross-attn to speech enc (sequential)	38.2	**33.2**
D9	+ Cross-attn to speech enc (parallel)	38.1	33.1

Table 3: Ablation study for two-pass direct S2ST models on multi-domain Es→En dev set. The first-pass decoder in all the models is pre-trained with t-mBART.

4.3 Decoding efficiency

We evaluated the decoding efficiency of direct S2ST models. We measured the runtime and total number of floating point operations (FLOPs) on an Intel® Xeon® Gold 6230 CPU. We randomly sampled 500 utterances from the multi-domain Es→En dev set while keeping the ratio of the number of samples per domain. Note that we also took the vocoder inference into account.

The results in Figure 3 showed that UnitY achieved 2.51× and 2.83× decoding speed-ups over S2SpecT2 and S2UT models, respectively. These confirms the efficiency of discrete unit prediction and two-pass decoding, thanks to reduced output sequence lengths. Deep-shallow two-pass decoders also improved the decoding speed a lot. We found that the translation quality of the two-pass models improved by increasing the beam width of the first-pass decoder up to 10. On the other hand, the quality did not degrade significantly by decreasing the beam width of the second-pass decoder down to 1, *i.e.* greedy decoding. This indicates that the first pass involves more challenges in the modeling pipeline. Therefore, we can obtain better translation quality and decoding speed by assigning more computation time to the first pass.

We also present the results of FLOPs in Appendix I. To summarize, UnitY achieved 1.65× and 3.19× FLOPs reduction over S2SpecT2 and S2UT models, respectively.

4.4 Fisher

We also show the results on Fisher in Appendix I. Although the trend was consistent with CVSS-C, a notable exception was that S2SpecT2 outperformed UnitY when pre-training the speech encoder with wav2vec2.0. However, UnitY has an advantage of decoding efficiency over S2SpecT2.

5 Analysis

In this section, we conduct analyses to shed light on the source of improvements in UnitY. We also study whether the same techniques used for UnitY are helpful for S2SpecT2. We use the multi-domain Es→En corpus, but pseudo-labeled ASR data is excluded for quick exploration, resulting in 196-hour source speech. We report average dev scores over three runs with different random seeds.[7]

5.1 Ablation study

We first conducted an ablation study for two-pass direct S2ST models in Table 3. We evaluated the translation quality of outputs from both decoders. An additional T2U/T2S encoder was essential for bridging the gap in representations between the first-pass and second-pass decoders, especially for S2SpecT2 (D2, D6). We attribute this to the fact that the gap in representations between text and spectrogram is larger than between text and discrete units. R-Drop was also beneficial for boosting the translation quality of the first-pass decoder, which improved the final performance accordingly (D3, D7). Moreover, we investigated adding another cross-attention over the speech encoder output to the unit decoder, as discussed in §2.1. We expected that the first-pass decoder output lost useful information to generate target speech faithful to source speech. We explored parallel (*parallel*, D8) and sequential (*sequential*, D9) cross-attention, similar to (Zhu et al., 2019), but neither showed any improvement. The first-pass decoder already extracted source acoustic information well via multiple cross-attention modules. We also show the results on Fisher in Appendix I.

[7] We removed three long utterances from the mTEDx dev set to fit the GPU memory.

ID	Model	Output unit	(ASR-)BLEU (↑) Text	Speech	Speed-up (×)
`E1`		Phoneme	–	29.4	1.00
`E2`	S2SpecT2	Character	31.7	28.9	0.89
`E3`		Subword	**33.0**	**30.0**	**1.12**
`E4`		Phoneme	–	27.8	2.31
`E5`	UnitY	Character	33.2	29.6	2.06
`E6`		Subword	**34.1**	**30.1**	**2.86**

Table 4: Results of output units for the first-pass decoder in two-pass direct S2ST models on multi-domain Es→En dev set. The first-pass decoder in all the models is initialized randomly.

ID	Decoder depth: First pass (text)	Second pass (unit)	#Params (Billion)	(ASR-)BLEU (↑) Text	Speech	Speed-up (×)
`G1`	2	6	0.79	34.5	30.3	1.24
`G2`	4	6	0.82	34.5	30.5	1.20
`G3`	6	2	0.79	34.3	30.3	1.47
`G4`	6	4	0.82	33.9	29.9	1.19
`G5`	6	6	0.86	**34.8**	**30.7**	1.00
`G6`	6	8	0.89	34.2	30.2	0.69
`G7`	6	12$^\diamondsuit$	0.96	33.7	29.8	0.68
`G8`	12	2	0.95	**34.9**	**30.7**	**1.44**
`G9`	12$^\spadesuit$	2	0.95	**38.3**	**33.2**	**1.19**
`G10`	12$^\spadesuit$	4	0.98	38.0	33.0	1.09
`G11`	12$^\spadesuit$	6	1.00	38.1	33.1	0.84
`G12`	12$^\spadesuit$	12$^\diamondsuit$	1.12	36.2	32.2	0.60

Table 5: Results of capacity assignment to two-pass decoders in UnitY on multi-domain Es→En dev set. $^\spadesuit$Pre-trained with t-mBART. $^\diamondsuit$Pre-trained with u-mBART. `G1`–`G8` have a 2k subword vocabulary, and `G9`–`G12` have a 65k subword vocabulary.

5.2 Output unit for first-pass decoder

We studied optimal granularity of the output unit for the first-pass decoder in two-pass direct S2ST models. We explored phonemes, characters, and 2k subwords units. The results in Table 4 showed that the subword unit (`E6`) was the most effective for the first-pass decoder in both UnitY and S2SpecT2 thanks to a better translation quality. Moreover, it gained the largest decoding speed-up. We also show the results on Fisher in Appendix I.

5.3 Capacity assignment to two-pass decoders

We sought to effectively assign the model capacity to the two decoders in UnitY to obtain a better translation quality. The results in Table 5 showed that a 12-layer text decoder with a two-layer unit decoder (`G8`) was the best in translation quality and decoding speed when initializing the first-pass decoder randomly (`G1`-`G6`,`G8`). Pre-training the first-pass decoder with t-mBART (`G9`) brought a large ASR-BLEU gain with a slight speed degradation compared to `G8`.[8] It was sufficient to have a two-layer unit decoder in that case (`G9`-`G11`). We also pre-trained the second-pass decoder with u-mBART while initializing the text decoder randomly (`G7`) or with t-mBART (`G12`), but neither improved the performance further. Therefore, it is most effective to pre-train the deep text decoder only and keep the unit decoder shallow. Note that `G8` is faster than `G9` because of the smaller subword vocabulary size (2k v.s. 65k).

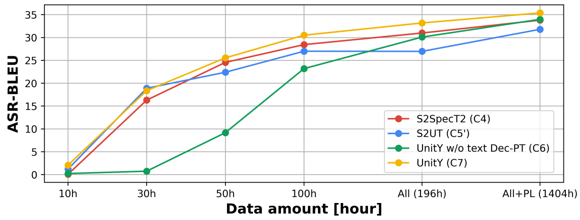


Figure 4: Dev ASR-BLEU at different data scales on the multi-domain Es→En corpus. The amount of training data is measured by source speech. *All* and *PL* represent all supervised data and pseudo-labeled data, respectively.

5.4 Data scale

Improving the translation quality of S2ST models on low-resource data is crucial since collecting a large amount of training data is challenging. We compared translation quality of direct S2ST models at various training data scales in Figure 4. We observed that UnitY consistently outperformed the S2SpecT2 and S2UT models when the data size was no less than 50 hours. The text decoder pre-training became less effective as the data size increased, consistent with an observation in §4.2, where the improvement in En→Es (+1.3) was smaller than Es→En (+2.5). However, pre-training the text decoder of UnitY was essential for obtaining decent performances in the low-resource settings ($\leq$ 50 hours).

6 Related works

Two-pass sequence generation Two-pass decoding has advantages of maintaining the end-to-end optimization capability while inheriting the benefits of a cascading approach. Xia et al. (2017); Hu et al. (2020) incorporate an additional search process to find a better output. Dalmia et al. (2021)

[8] We set the depth of the first-pass decoder to 12 because of the availability of the off-the-shelf t-mBART model.

reranks the intermediate hypotheses using an external module such as a language model. Zhao et al. (2019) injects specific information in the intermediate decoder to bias the output toward the desired domain. Sainath et al. (2019) provides an intermediate output to users before generating the final output for streaming applications. The two-pass approach makes the optimization tractable, which has advanced performance of speech translation models (Anastasopoulos and Chiang, 2018; Sperber et al., 2019; Sung et al., 2019; Dalmia et al., 2021; Inaguma et al., 2021a; Yan et al., 2022; Jia et al., 2022b).

Direct speech-to-spectrogram translation Translatotron (Jia et al., 2019b) is the first direct S2ST model but suffered from poor performance even with auxiliary ASR and S2TT tasks. Kano et al. (2021) subsequently pre-trains the components with ASR and S2TT models, which is more effective for distant language pairs. Translatotron2 (Jia et al., 2022b) significantly improves Translatotron by incorporating two-pass decoding. We showed that our methods further improved Translatotron2.

Direct speech-to-unit translation Direct speech-to-unit translation models predict discrete units rather than spectrogram. Tjandra et al. (2019) uses vector-quantized variational autoencoder (Van Den Oord et al., 2017) while Lee et al. (2022a) used HuBERT (Hsu et al., 2021) to extract target discrete units. Lee et al. (2022b) normalizes speaker identity of real target speech using a CTC-based speech-to-unit model. Huang et al. (2022) further improves the normalization by considering rhythm, pitch, and energy.

7 Conclusion

We proposed UnitY, a novel efficient two-pass direct S2ST model that subsequently generates both text and discrete unit outputs. We improved the model performance by predicting subwords in the first pass, bridging decoder representations by an additional encoder, deep-shallow two-pass decoders, regularizing the training with R-Drop, and pre-training the first-pass decoder with text-based mBART. Experimental evaluations demonstrated that UnitY outperformed a single-pass S2UT model consistently in translation quality and inference speed. We showed that the proposed methods improve the two-pass direct speech-to-spectrogram model as well, confirming their versatility. Still, UnitY achieved 2.51× decoding speed-up over the case.

8 Limitation

Since two-pass direct S2ST models require linguistic units as the target for the first-pass decoder, they cannot be used when the target language is unwritten. Compared to cascaded S2ST systems, direct S2ST systems require more data preparation steps, including training a HuBERT model, synthesizing target speech with a TTS model, extracting discrete units with the HuBERT model, and training a unit-based vocoder, etc. Moreover, the target audio quality of direct speech-to-unit systems relies on the quality of discrete units generated by self-supervised discrete models. It further depends on the availability of speech data to train HuBERT models for the target languages.

Because S2ST systems could generate speech that does not necessarily represent the source speech's content, there is a potential risk of conveying wrong information.

Acknowledgement

We would like to thank Justine Kao and Carleigh Wood for the help on human evaluation.

References

Antonios Anastasopoulos, Ondřej Bojar, Jacob Bremerman, Roldano Cattoni, Maha Elbayad, Marcello Federico, Xutai Ma, Satoshi Nakamura, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Sebastian Stüker, Katsuhito Sudoh, Marco Turchi, Alexander Waibel, Changhan Wang, and Matthew Wiesner. 2021. FINDINGS OF THE IWSLT 2021 EVALUATION CAMPAIGN. In *Proceedings of IWSLT*, pages 1–29.

Antonios Anastasopoulos and David Chiang. 2018. Tied multitask learning for neural speech translation. In *Proceedings of NAACL-HLT*, pages 82–91.

Ebrahim Ansari, Amittai Axelrod, Nguyen Bach, Ondřej Bojar, Roldano Cattoni, Fahim Dalvi, Nadir Durrani, Marcello Federico, Christian Federmann, Jiatao Gu, Fei Huang, Kevin Knight, Xutai Ma, Ajay Nagesh, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Xing Shi, Sebastian Stüker, Marco Turchi, Alexander Waibel, and Changhan Wang. 2020. FINDINGS OF THE IWSLT 2020 EVALUATION CAMPAIGN. In *Proceedings of IWSLT*, pages 1–34.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben

Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common Voice: A massively-multilingual speech corpus. In *Proceedings of LREC*, pages 4218–4222.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Proceedings of NeurIPS*, volume 33, pages 12449–12460.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*.

Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. 2016. End-to-end attention-based large vocabulary speech recognition. In *Proceedings of ICASSP*, pages 4945–4949.

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. *arXiv preprint arXiv:2202.01374*.

Alexandre Bérard, Laurent Besacier, Ali Can Kocabiyikoglu, and Olivier Pietquin. 2018. End-to-end automatic speech translation of audiobooks. In *Proceedings of ICASSP*, pages 6224–6228. IEEE.

Alexandre Bérard, Olivier Pietquin, Christophe Servan, and Laurent Besacier. 2016. Listen and translate: A proof of concept for end-to-end speech-to-text translation. In *Proceedings of NIPS 2016 End-to-end Learning for Speech and Audio Processing Workshop*.

William Chan, Daniel Park, Chris Lee, Yu Zhang, Quoc Le, and Mohammad Norouzi. 2021. Speechstew: Simply mix all available speech recognition data to train one large neural network. *arXiv preprint arXiv:2104.02133*.

Colin Cherry, George Foster, Ankur Bapna, Orhan Firat, and Wolfgang Macherey. 2018. Revisiting character-based neural machine translation with capacity and compression. In *Proceedings of EMNLP*, pages 4295–4305.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. 2021. w2v-BERT: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In *Proceedings of ASRU*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of ACL*, pages 8440–8451.

Siddharth Dalmia, Brian Yan, Vikas Raunak, Florian Metze, and Shinji Watanabe. 2021. Searchable hidden intermediates for end-to-end models of decomposable sequence tasks. In *Proceedings of NAACL-HLT*, pages 1882–1896.

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. MuST-C: a Multilingual Speech Translation Corpus. In *Proceedings of NAACL-HLT*, pages 2012–2017.

Qianqian Dong, Fengpeng Yue, Tom Ko, Mingxuan Wang, Qibing Bai, and Yu Zhang. 2022. Leveraging pseudo-labeled data to improve direct speech-to-speech translation. *arXiv preprint arXiv:2205.08993*.

Salesky Elizabeth, Wiesner Matthew, Bremerman Jacob, Roldano Cattoni, Matteo Negri, Marco Turchi, Douglas W Oard, and Post Matt. 2021. The multilingual TEDx corpus for speech recognition and translation. In *Proceedings of Interspeech*, pages 3655–3659.

Mark JF Gales, Kate M Knill, Anton Ragni, and Shakti P Rath. 2014. Speech recognition and keyword spotting for low-resource languages: Babel project research at CUED. In *Proceedings of SLTU*, pages 16–23.

Thamme Gowda and Jonathan May. 2020. Finding the optimal vocabulary size for neural machine translation. In *Findings of EMNLP*, pages 3955–3964.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of ICML*, pages 369–376.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented Transformer for speech recognition. In *Proceedings of Interspeech*, pages 5036–5040.

Mary Harper et al. IARPA Babel Program. https://www.iarpa.gov/research-programs/babel. [Online].

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

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

Ke Hu, Tara N Sainath, Ruoming Pang, and Rohit Prabhavalkar. 2020. Deliberation model based two-pass end-to-end speech recognition. In *Proceedings of ICASSP*, pages 7799–7803.

Rongjie Huang, Zhou Zhao, Jinglin Liu, Huadai Liu, Yi Ren, Lichao Zhang, and Jinzheng He. 2022. TranSpeech: Speech-to-speech translation with bilateral perturbation. *arXiv preprint arXiv:2205.12523*.

Hirofumi Inaguma, Siddharth Dalmia, Brian Yan, and Shinji Watanabe. 2021a. Fast-MD: Fast multi-decoder end-to-end speech translation with non-autoregressive hidden intermediates. In *Proceedings of ASRU*, pages 922–929.

Hirofumi Inaguma, Tatsuya Kawahara, and Shinji Watanabe. 2021b. Source and target bidirectional knowledge distillation for end-to-end speech translation. In *Proceedings of NAACL-HLT*, pages 1872–1881.

Javier Iranzo-Sánchez, Joan Albert Silvestre-Cerda, Javier Jorge, Nahuel Roselló, Adria Giménez, Albert Sanchis, Jorge Civera, and Alfons Juan. 2020. Europarl-ST: A multilingual corpus for speech translation of parliamentary debates. In *Proceedings of ICASSP*, pages 8229–8233.

Keith Ito and Linda Johnson. 2017. The lj speech dataset. https://keithito.com/LJ-Speech-Dataset/.

Ye Jia, Yifan Ding, Ankur Bapna, Colin Cherry, Yu Zhang, Alexis Conneau, and Nobuyuki Morioka. 2022a. Leveraging unsupervised and weakly-supervised data to improve direct speech-to-speech translation. In *Proceedings of Interspeech*, pages 1721–1725.

Ye Jia, Melvin Johnson, Wolfgang Macherey, Ron J Weiss, Yuan Cao, Chung-Cheng Chiu, Naveen Ari, Stella Laurenzo, and Yonghui Wu. 2019a. Leveraging weakly supervised data to improve end-to-end speech-to-text translation. In *Proceedings of ICASSP*, pages 7180–7184.

Ye Jia, Michelle Tadmor Ramanovich, Tal Remez, and Roi Pomerantz. 2022b. Translatotron 2: High-quality direct speech-to-speech translation with voice preservation. In *Proceedings of ICML*.

Ye Jia, Michelle Tadmor Ramanovich, Quan Wang, and Heiga Zen. 2022c. CVSS corpus and massively multilingual speech-to-speech translation. In *Proceedings of LREC*, pages 6691–6703.

Ye Jia, Ron J Weiss, Fadi Biadsy, Wolfgang Macherey, Melvin Johnson, Zhifeng Chen, and Yonghui Wu. 2019b. Direct speech-to-speech translation with a sequence-to-sequence model. In *Proceedings of Interspeech*, pages 1123–1127.

Jacob Kahn, Morgane Rivière, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, Pierre-Emmanuel Mazaré, Julien Karadayi, Vitaliy Liptchinsky, Ronan Collobert, Christian Fuegen, et al. 2020. Libri-Light: A benchmark for asr with limited or no supervision. In *Proceedings of ICASSP*, pages 7669–7673.

Takatomo Kano, Sakriani Sakti, and Satoshi Nakamura. 2021. Transformer-based direct speech-to-speech translation with transcoder. In *Proceedings of SLT*, pages 958–965.

Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah Smith. 2021. Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation. In *Proceedings of ICLR*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis. In *Proceedings of NeurIPS*, volume 33, pages 17022–17033.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of ACL*, pages 66–75.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of EMNLP: System Demonstrations*, pages 66–71.

Alon Lavie, Alex Waibel, Lori Levin, Michael Finke, Donna Gates, Marsal Gavalda, Torsten Zeppenfeld, and Puming Zhan. 1997. JANUS-III: Speech-to-speech translation in multiple languages. In *Proceedings of ICASSP*, pages 99–102.

Ann Lee, Peng-Jen Chen, Changhan Wang, Jiatao Gu, Xutai Ma, Adam Polyak, Yossi Adi, Qing He, Yun Tang, Juan Pino, et al. 2022a. Direct speech-to-speech translation with discrete units. In *Proceedings of ACL*, pages 3327–3339.

Ann Lee, Hongyu Gong, Paul-Ambroise Duquenne, Holger Schwenk, Peng-Jen Chen, Changhan Wang, Sravya Popuri, Juan Pino, Jiatao Gu, and Wei-Ning Hsu. 2022b. Textless speech-to-speech translation on real data. In *Proceedings of NAACL-HLT*, pages 860–872.

Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. 2019. Neural speech synthesis with Transformer network. In *Proceedings of AAAI*, volume 33, pages 6706–6713.

Xian Li, Changhan Wang, Yun Tang, Chau Tran, Yuqing Tang, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2021. Multilingual speech translation from efficient finetuning of pretrained models. In *Proceedings of ACL*, pages 827–838.

Xinjian Li, Ye Jia, and Chung-Cheng Chiu. 2022. Textless direct speech-to-speech translation with discrete speech representation. *arXiv preprint arXiv:2211.00115*.

Daniel Licht, Cynthia Gao, Janice Lam, Francisco Guzman, Mona Diab, and Philipp Koehn. 2022. Consistent human evaluation of machine translation across language pairs. In *Proceedings of AMTA*, pages 309–321.

Pierre Lison, Jörg Tiedemann, and Milen Kouylekov. 2018. OpenSubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proceedings of LREC*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pretraining for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

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

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. 2018. Mixed precision training. In *Proceedings of ICLR*.

Mehryar Mohri, Fernando Pereira, and Michael Riley. 2002. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16(1):69–88.

Satoshi Nakamura, Konstantin Markov, Hiromi Nakaiwa, Gen-ichiro Kikui, Hisashi Kawai, Takatoshi Jitsuhiro, J-S Zhang, Hirofumi Yamamoto, Eiichiro Sumita, and Seiichi Yamamoto. 2006. The ATR multilingual speech-to-speech translation system. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 14(2):365–376.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *Proceedings of ICASSP*, pages 5206–5210.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318.

Kyubyong Park and Thomas Mulc. 2019. CSS10: A collection of single speaker speech datasets for 10 languages. In *Proceedings of Interspeech*, pages 1566–1570.

Juan Pino, Qiantong Xu, Xutai Ma, Mohammad Javad Dousti, and Yun Tang. 2020. Self-training for end-to-end speech translation. In *Proceedings of Interspeech*, pages 1476–1480.

Adam Polyak, Yossi Adi, Jade Copet, Eugene Kharitonov, Kushal Lakhotia, Wei-Ning Hsu, Abdelrahman Mohamed, and Emmanuel Dupoux. 2021. Speech resynthesis from discrete disentangled self-supervised representations. In *Proceedings of Interspeech*, pages 3615–3619.

Sravya Popuri, Peng-Jen Chen, Changhan Wang, Juan Pino, Yossi Adi, Jiatao Gu, Wei-Ning Hsu, and Ann Lee. 2022. Enhanced direct speech-to-speech translation using self-supervised pre-training and data augmentation. In *Proceedings of Interspeech*, pages 5195–5199.

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

Matt Post, Gaurav Kumar, Adam Lopez, Damianos Karakos, Chris Callison-Burch, and Sanjeev Khudanpur. 2013. Improved speech-to-text translation with the Fisher and Callhome Spanish–English speech translation corpus. In *Proceedings of IWSLT*.

Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. 2020. MLS: A large-scale multilingual dataset for speech research. In *Proceedings of Interspeech*, pages 2757–2761.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of EMNLP*, pages 4512–4525.

Anthony Rousseau, Paul Deléglise, and Yannick Estève. 2012. TED-LIUM: An automatic speech recognition dedicated corpus. In *Proceedings of LREC*, pages 125–129.

Tara N Sainath, Ruoming Pang, David Rybach, Yanzhang He, Rohit Prabhavalkar, Wei Li, Mirkó Visontai, Qiao Liang, Trevor Strohman, Yonghui Wu, et al. 2019. Two-pass end-to-end speech recognition. In *Proceedings of Interspeech*, pages 2773–2777.

Elizabeth Salesky, Julian Mäder, and Severin Klinger. 2021. Assessing evaluation metrics for speech-to-speech translation. In *Proceedings of ASRU*, pages 733–740.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021. CCMatrix: Mining billions of high-quality parallel sentences on the web. In *Proceedings of ACL*, pages 6490–6500.

Jonathan Shen, Ye Jia, Mike Chrzanowski, Yu Zhang, Isaac Elias, Heiga Zen, and Yonghui Wu. 2020. Non-Attentive Tacotron: Robust and controllable neural TTS synthesis including unsupervised duration modeling. *arXiv preprint arXiv:2010.04301*.

Raivis Skadiņš, Jörg Tiedemann, Roberts Rozis, and Daiga Deksne. 2014. Billions of parallel words for free: Building and using the EU bookshop corpus. In *Proceedings of LREC*, pages 1850–1855.

Matthias Sperber, Graham Neubig, Jan Niehues, and Alex Waibel. 2019. Attention-passing models for robust and data-efficient end-to-end speech translation. *Transactions of the Association for Computational Linguistics*, 7:313–325.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Tzu-Wei Sung, Jun-You Liu, Hung-yi Lee, and Lin-shan Lee. 2019. Towards end-to-end speech-to-text translation with two-pass decoding. In *Proceedings of ICASSP*, pages 7175–7179.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of NIPS*, volume 27.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of CVPR*, pages 2818–2826.

Yun Tang, Hongyu Gong, Ning Dong, Changhan Wang, Wei-Ning Hsu, Jiatao Gu, Alexei Baevski, Xian Li, Abdelrahman Mohamed, Michael Auli, and Juan Pino. 2022. Unified speech-text pre-training for speech translation and recognition. In *Proceedings of ACL*, pages 1488–1499.

Yun Tang, Juan Pino, Xian Li, Changhan Wang, and Dmitriy Genzel. 2021. Improving speech translation by understanding and learning from the auxiliary text translation task. In *Proceedings of ACL*, pages 4252–4261.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*.

Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura. 2019. Speech-to-speech translation between untranscribed unknown languages. In *Proceedings of ASRU*, pages 593–600.

Jörgen Valk and Tanel Alumäe. 2021. VoxLingua107: a dataset for spoken language recognition. In *Proceedings of SLT*, pages 652–658.

Aaron Van Den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. In *Proceedings of NIPS*, volume 30.

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 *Proceedings of NIPS*, volume 30.

Wolfgang Wahlster. 2013. *Verbmobil: foundations of speech-to-speech translation*. Springer Science & Business Media.

Changhan Wang, Hirofumi Inaguma, Peng-Jen Chen, Ilia Kulikov, Yun Tang, Wei-Ning Hsu, Michael Auli, and Juan Pino. 2022. Simple and effective unsupervised speech translation. *arXiv preprint arXiv:2210.10191*.

Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux. 2021a. VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. In *Proceedings of ACL*, pages 993–1003.

Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Pino. 2020. Fairseq S2T: Fast speech-to-text modeling with Fairseq. In *Proceedings of AACL: System Demonstrations*, pages 33–39.

Changhan Wang, Anne Wu, Jiatao Gu, and Juan Pino. 2021b. CoVoST 2 and massively multilingual speech translation. In *Proceedings of Interspeech*, pages 2247–2251.

Changhan Wang, Anne Wu, Juan Pino, Alexei Baevski, Michael Auli, and Alexis Conneau. 2021c. Large-scale self- and semi-supervised learning for speech translation. In *Proceedings of Interspeech*, pages 2242–2246.

Ron J Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. 2017. Sequence-to-sequence models can directly translate foreign speech. In *Proceedings of Interspeech*, pages 2625–2629.

Krzysztof Wołk and Krzysztof Marasek. 2014. Building subject-aligned comparable corpora and mining it for truly parallel sentence pairs. *Procedia Technology*, 18:126–132.

Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, Tie-Yan Liu, et al. 2021. R-Drop: Regularized dropout for neural networks. In *Proceedings off NeurIPS*, volume 34, pages 10890–10905.

Yingce Xia, Fei Tian, Lijun Wu, Jianxin Lin, Tao Qin, Nenghai Yu, and Tie-Yan Liu. 2017. Deliberation networks: Sequence generation beyond one-pass decoding. In *Proceedings of NIPS*, volume 30.

Brian Yan, Patrick Fernandes, Siddharth Dalmia, Jiatong Shi, Yifan Peng, Dan Berrebbi, Xinyi Wang, Graham Neubig, and Shinji Watanabe. 2022. CMU's IWSLT 2022 dialect speech translation system. In *Proceedings of IWSLT*, pages 298–307.

Chen Zhang, Xu Tan, Yi Ren, Tao Qin, Kejun Zhang, and Tie-Yan Liu. 2021. Uwspeech: Speech to speech translation for unwritten languages. In *Proceedings of AAAI*, pages 14319–14327.

Ding Zhao, Tara N. Sainath, David Rybach, Pat Rondon, Deepti Bhatia, Bo Li, and Ruoming Pang. 2019. Shallow-fusion end-to-end contextual biasing. In *Proceedings of Interspeech*, pages 1418–1422.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tieyan Liu. 2019. Incorporating BERT into neural machine translation. In *Proceedings of ICLR*.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The United Nations parallel corpus v1.0. In *Proceedings of LREC*, pages 3530–3534.

Algorithm 1 Two-pass beam search decoding

function TwoPassBeamSearch($X, B_{1\text{st}}, B_{2\text{nd}}$)
  $H \leftarrow \texttt{SpeechEnc}(X)$ ▷ $H: (T', d_{\text{model}})$

  `// First-pass beam search`
  $\Omega_{1\text{st}} \leftarrow \{\}$
  $\Omega_{1\text{st}} \leftarrow \textit{BeamSearch}_1(H, B_{1\text{st}}, \Omega_{1\text{st}})$
  $\hat{Y} \leftarrow \text{argmax}(\Omega_{1\text{st}})$ ▷ $|\Omega_{1\text{st}}| = B_{1\text{st}}$

  $D^{\text{text}} \leftarrow \textit{GetHiddenStateFromCache}(\hat{Y})$ ▷ $D^{\text{text}}: (M, d_{\text{model}})$
  $Z \leftarrow \texttt{T2UEnc}(D^{\text{text}})$ ▷ $Z: (M, d_{\text{model}})$

  `// Second-pass beam search`
  $\Omega_{2\text{nd}} \leftarrow \{\}$
  $\Omega_{2\text{nd}} \leftarrow \textit{BeamSearch}_2(Z, B_{2\text{nd}}, \Omega_{2\text{nd}})$
  $\hat{U} \leftarrow \text{argmax}(\Omega_{2\text{nd}})$ ▷ $|\Omega_{2\text{nd}}| = B_{2\text{nd}}$

  `// Convert discrete units to waveform`
  $\hat{W} \leftarrow \texttt{UnitVocoder}(\hat{U})$
  return $\hat{W}$
end function

A Pseudo algorithm for two-pass beam search decoding

Algorithm 1 shows the two-pass beam serach decoding algorithm of UnitY as discussed in §2.3. We first encode a source speech X with the speech encoder `SpeechEnc` and map it to the encoder output H.

The first-pass decoder takes H as input and generates a text sequence. *BeamSearch*$_1$ is a first-pass beam search function that takes an empty hypothesis set $\Omega_{1\text{st}}$ and returns the beam candidates. We take the best text hypothesis $\hat{Y}$ and get the corresponding decoder output D^{text} from a cache via the *GetHiddenStateFromCache* function. Next, the T2U encoder `T2UEnc` takes D^{text} as input and maps it to the output Z.

The second-pass decoder takes H and Z as inputs and generates a discrete unit sequence. *BeamSearch*$_2$ is a second-pass beam search function that takes an empty hypothesis set $\Omega_{2\text{nd}}$ and returns the beam candidates. We take the best discrete unit hypothesis $\hat{U}$. Finally, the unit-based vocoder `UnitVocoder` converts $\hat{U}$ to the waveform $\hat{W}$.

B Training with R-Drop

UnitY introduces an intermediate S2TT sub-task to make the optimization tractable while maintaining the end-to-end differentiability. However, the easier S2TT task is more likely to overfit than the primary S2UT task. To tackle this problem, we apply a more effective regularization based on R-Drop (Wu et al., 2021) to the first-pass decoder in addition to standard regularization such as dropout (Srivastava et al., 2014) and label smoothing (Szegedy et al., 2016). Theoretically, R-Drop reduces the inconsistency of model predictions between training and inference by dropout, thus improving the generalization ability. R-Drop duplicates the network input during training and calculates two output probability distributions with different dropout masks. Then, a constraint is introduced by minimizing the Kullback–Leibler (KL) divergence loss between the two probability distributions. We apply R-Drop to both text and unit decoders. The total training objective of UnitY with R-Drop, $\mathcal{L}_{\text{total}}$, is modified from Eq. (1) as follows:

$$\mathcal{L}_{\text{total}} = \sum_{i=1}^{2} \mathcal{L}_{\text{s2u}}(U|X_i, Y) + \alpha \mathcal{L}_{\text{kl}}^{\text{s2u}}(X_1, X_2) + w_{\text{s2t}}(\sum_{i=1}^{2} \mathcal{L}_{\text{s2t}}(Y|X_i) + \beta \mathcal{L}_{\text{kl}}^{\text{s2t}}(X_1, X_2)), \quad (2)$$

where X_i is a duplicated input from X, $\mathcal{L}_{\text{kl}}^{\text{s2u}}$ and $\mathcal{L}_{\text{kl}}^{\text{s2t}}$ are KL losses for the unit and text decoders, w_{s2t} is a weight for the S2TT loss, and α and β are weights for the KL losses, respectively. We implement R-Drop by duplicating inputs instead of feeding them to the network twice.

Given a set of unique inputs $\mathbf{X}$, the general KL loss $\mathcal{L}_{\text{kl}}$ in R-Drop is formulated as follows:

$$\mathcal{L}_{\text{kl}}(\mathbf{X}_1, \mathbf{X}_2) = \frac{1}{2}(\mathcal{D}_{\text{kl}}(P(\cdot|\mathbf{X}_1)||P(\cdot|\mathbf{X}_2) + \mathcal{D}_{\text{kl}}(P(\cdot|\mathbf{X}_2))||P(\cdot|\mathbf{X}_1))),$$

where $\mathbf{X}_i$ is a duplicated input from $\mathbf{X}$, $\mathcal{D}_{\text{kl}}$ is a KL divergence, and P is a categorical probability distribution.

C Training objective

In this section, we describe training objectives for the baseline S2ST models. In addition to the primary S2ST/S2UT task, we introduce auxiliary S2TT and ASR tasks. We adopted an auxiliary character-level ASR task for the direct S2ST models trained from scratch on Fisher, regardless of the choice of the output unit in the first-pass decoder. We did not use the ASR task in the rest settings.

S2SpecT The architecture of S2SpecT is shown in Figure 2a. Given the target spectrogram S, translation Y, and transcription Y_{src}, corresponding to a source speech X, the training objective of S2SpecT is formulated as:

$$\begin{aligned}\mathcal{L}_{\text{total}} &= \mathcal{L}_{\text{s2s}}(S|X) \\ &\quad + w_{\text{s2t}}\mathcal{L}_{\text{s2t}}(Y|X) + w_{\text{asr}}\mathcal{L}_{\text{asr}}(Y_{\text{src}}|X),\end{aligned} \quad (3)$$

where $\mathcal{L}_{\text{s2s}}$ is the primary S2ST loss, $\mathcal{L}_{\text{s2t}}$ is the auxiliary S2TT loss, $\mathcal{L}_{\text{asr}}$ is the auxiliary ASR loss, w_{s2t} is a weight for the S2TT loss, and w_{asr} is a weight for the ASR loss, respectively. Note that R-Drop is not used because the output of the primary S2ST task is continuous.

We adopt the autoregressive decoder of Transformer TTS (Li et al., 2019) as the spectrogram decoder. Therefore, $\mathcal{L}_{\text{s2s}}$ is defined as a sum of the L1 loss $\mathcal{L}_1$, L2 loss $\mathcal{L}_2$, and end-of-sentence (EOS) prediction loss $\mathcal{L}_{\text{eos}}$ as follows:

$$\mathcal{L}_{\text{s2s}}(S|X) = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_{\text{eos}}.$$

S2SpecT2 The architecture of S2SpecT2 is shown in Figure 2b. The training objective of S2SpecT2 is formulated as:

$$\begin{aligned}\mathcal{L}_{\text{total}} &= \sum_{i=1}^{2}\mathcal{L}_{\text{s2s}}(S|X_i, Y) \\ &\quad + w_{\text{s2t}}(\sum_{i=1}^{2}\mathcal{L}_{\text{s2t}}(Y|X_i) + \beta\mathcal{L}_{\text{kl}}^{\text{s2t}}(X_1, X_2)) \\ &\quad + w_{\text{asr}}(\sum_{i=1}^{2}\mathcal{L}_{\text{asr}}(Y_{\text{src}}|X_i) + \gamma\mathcal{L}_{\text{kl}}^{\text{asr}}(X_1, X_2)),\end{aligned} \quad (4)$$

where X_i is a duplicated input from X, $\mathcal{L}_{\text{kl}}^{\text{s2t}}$ is the R-Drop's KL loss for the first-pass decoder, $\mathcal{L}_{\text{kl}}^{\text{s2t}}$ is the R-Drop's KL loss for the auxiliary ASR decoder, and β and γ are the corresponding weights for the R-Drop's KL losses, respectively. Unlike Eq. (3), the primary S2ST task depends on the output from the first-pass decoder. We apply R-Drop to the S2TT and ASR tasks only. We also investigated applying R-Drop to the second-pass spectrogram decoder by minimizing the difference of two outputs in the continuous space, but the training was unstable.

S2UT The architecture of S2UT is shown in Figure 2c. In addition to the primary S2UT loss and auxiliary S2TT and ASR losses, we use a CTC loss on top of the unit decoder following Lee et al. (2022a). The training objective of the S2UT model is formulated as:

$$\begin{aligned}\mathcal{L}_{\text{total}} &= \sum_{i=1}^{2}\mathcal{L}_{\text{s2u}}(U|X_i) + \alpha\mathcal{L}_{\text{kl}}^{\text{s2u}}(X_1, X_2) \\ &\quad + w_{\text{ctc}}\sum_{i=1}^{2}\mathcal{L}_{\text{ctc}}(Y|D_i^{\text{unit}}) \\ &\quad + w_{\text{s2t}}(\sum_{i=1}^{2}\mathcal{L}_{\text{s2t}}(Y|X_i) + \beta\mathcal{L}_{\text{kl}}^{\text{s2t}}(X_1, X_2)) \\ &\quad + w_{\text{asr}}(\sum_{i=1}^{2}\mathcal{L}_{\text{asr}}(Y_{\text{src}}|X_i) + \gamma\mathcal{L}_{\text{kl}}^{\text{asr}}(X_1, X_2)),\end{aligned} \quad (5)$$

where $\mathcal{L}_{\text{s2u}}$ is the primary S2UT loss, $\mathcal{L}_{\text{kl}}^{\text{s2u}}$ is the R-Drop's KL loss for the unit decoder, $\mathcal{L}_{\text{ctc}}$ is the CTC loss, D_i^{unit} is the unit decoder output for the i-th forward pass, α is a weight for the R-Drop's KL loss, and w_{ctc} is a weight for the CTC loss, respectively. Unlike Eq. (2), there is no dependency between the primary S2UT task and auxiliary S2TT task except for sharing the same encoder.

S2TT, ASR We also apply R-Drop to S2TT and ASR tasks. The training objective of the S2TT model is formulated as:

$$\mathcal{L}_{\text{total}} = \sum_{i=1}^{2}\mathcal{L}_{\text{s2t}}(Y|X_i) + \beta\mathcal{L}_{\text{kl}}^{\text{s2t}}(X_1, X_2). \quad (6)$$

Similarly, the training objective of the ASR model is formulated as:

$$\mathcal{L}_{\text{total}} = \sum_{i=1}^{2}\mathcal{L}_{\text{asr}}(Y_{\text{src}}|X_i) + \gamma\mathcal{L}_{\text{kl}}^{\text{asr}}(X_1, X_2). \quad (7)$$

D Data

Fisher Es→En (Post et al., 2013) This corpus contains 170-hour Spanish conversational telephone speech with the corresponding transcriptions as well as the English translations. The target speech is synthesized by a high-quality in-house TTS model trained with a single female speaker (Lee et al., 2022a).

CVSS-C (Jia et al., 2022c) CVSS is a public multilingual S2ST corpus based on CoVoST2 (Wang et al., 2021b). It covers 21 language

Corpus	Language direction	
	En→Es	**Es→En**
S2TT	Europarl-ST [75.6 hours] (Iranzo-Sánchez et al., 2020) Must-C [495 hours] (Di Gangi et al., 2019)	CoVoST2 [112 hours] (Wang et al., 2021b) Europarl-ST [20.6 hours] mTEDx [63.4 hours] (Elizabeth et al., 2021)
ASR	Librispeech [960 hours] (Panayotov et al., 2015) TEDLIUM3 [452 hours] (Rousseau et al., 2012) Common Voice v7 [1203 hours] (Ardila et al., 2020)	MLS [918 hours] (Pratap et al., 2020) Common Voice v7 [290 hours]
MT		
Supervised MT1	CCMatrix [86.3M sentences] (Schwenk et al., 2021)	–
Supervised MT2 (Cascaded S2ST)	OpenSubtitle2018 [60M sentences] (Lison et al., 2018) UNCorpus [21.8M sentences] (Ziemski et al., 2016) EUBookshop v2 [5.2M sentences] (Skadiņš et al., 2014) Europarl v10 [1.9M sentences] (Koehn, 2005) Wikipedia v1.0 [1.8M sentences] (Wołk and Marasek, 2014) TED2020 v1 [0.4M sentences] (Reimers and Gurevych, 2020) Europarl-ST [32k sentences] Must-C [260k sentences] mTEDx [3.6k sentences] CoVosST2 [79k sentences]	
T2U/TTS	CSS100 [23.8 hours] (Park and Mulc, 2019)	LJSpeech [24 hours] (Ito and Johnson, 2017)
Unlabeled text		
t-mBART	CC100 [5.6B tokens] (Conneau et al., 2020)	
Unlabeled speech		
wav2vec2.0	Libri-Light [60k hours] (Kahn et al., 2020)	VoxPopuli Es [16k hours] (Wang et al., 2021a)
u-mBART	VoxPopuli En [14k hours] VoxPopuli Es [16k hours] Libri-Light [60k hours]	
mHuBERT	VoxPopuli En [4.5k hours] VoxPopuli Es [4.5k hours] VoxPopuli Fr [4.5k hours]	

Table 6: Statistics for the multi-domain En↔Es corpora

Model	URL
En wav2vec2.0	https://github.com/facebookresearch/fairseq/blob/main/examples/speech_to_speech/docs/enhanced_direct_s2st_discrete_units.md#wav2vec-20
Es wav2vec2.0	https://github.com/facebookresearch/fairseq/blob/main/examples/speech_to_speech/docs/enhanced_direct_s2st_discrete_units.md#wav2vec-20
En HuBERT	https://github.com/facebookresearch/fairseq/blob/main/examples/speech_to_speech/docs/direct_s2st_discrete_units.md
mHuBERT	https://github.com/facebookresearch/fairseq/blob/main/examples/speech_to_speech/docs/textless_s2st_real_data.md
En-Es u-mBART	https://dl.fbaipublicfiles.com/fairseq/speech_to_speech/s2st_finetuning/unit_mBART/checkpoint.pt
En Transformer TTS	https://huggingface.co/facebook/tts_transformer-en-ljspeech
Es Transformer TTS	https://huggingface.co/facebook/tts_transformer-es-css10

Table 7: Links to pre-trained self-supervised models and TTS models

directions to English. We use the CVSS-C part of the CVSS corpus, in which a single-speaker female TTS synthesizes the target speech. We combine all language directions to train a single X-to-En multilingual model.

Multi-domain En↔Es (Popuri et al., 2022) Following Popuri et al. (2022), we use all samples from multiple public S2TT corpora in each direction to improve the robustness of model training (Jia et al., 2022b; Chan et al., 2021). We also use all samples from validation sets in all domains for checkpoint selection. We further augment the S2ST training data by pseudo-labeling ASR corpora with MT and T2U/TTS models. We use the TTS model in the cascaded system to synthesize the target speech for direct speech-to-spectrogram models. For direct speech-to-unit models, we use a T2U model (Lee et al., 2022b) to generate discrete units on the ASR corpora and the TTS+HuBERT pipeline for the S2T corpora. Both T2U and TTS models are based on Transformer. We train En and Es T2U/TTS models on the LJSpeech (Ito and Johnson, 2017) and CSS10 (Park and Mulc, 2019) corpora, respectively.

For **En→Es**, we use all samples from Europarl-ST (Iranzo-Sánchez et al., 2020) and Must-C (Di Gangi et al., 2019) and augment the training data by TEDLIUM3 (Rousseau et al., 2012), Librispeech (Panayotov et al., 2015), and Common Voice (Ardila et al., 2020), resulting in 3180-

hour source speech. We removed samples overlapped with mtedx dev/test sets from TEDLIUM3. For **Es→En**, we use all samples from CoVoST2, Europarl-ST, and mTEDx (Elizabeth et al., 2021), and augment the training data by Common Voice and multilingual Librispeech (MLS) (Pratap et al., 2020), resulting in 1404-hour source speech. In Table 6, we list all the datasets used in each task.

E Pre-processing

Speech We convert source audio to 16kHz and generate target speech with 22kHz. When extracting discrete units, we downsample the target speech to 16kHz. For filterbank features, we extract 80-dimensional coefficients on both the source and target sides. We apply utterance-level cepstral mean-variance normalization to both inputs.

Discrete units We extract discrete units with an English HuBERT trained on Librispeech after performing k-means clustering with 100 clusters on Fisher (Lee et al., 2022a). For the rest corpora, we extract discrete units with a multilingual HuBERT (mHuBERT) (Popuri et al., 2022) trained on En, En, and Fr parts of VoxPopuli (Wang et al., 2021a) with the number of k-means clusters of 1000.

Text We lowercase text data and remove all punctuation marks except for apostrophes. When initializing the text decoder in two-pass direct S2ST models randomly, we build vocabularies of 1k, 6k, and 2k unigram subword units (Kudo, 2018) with the SentencePiece toolkit (Kudo and Richardson, 2018) for the Fisher, CVSS-C, and multi-domain corpora, respectively. When pre-training the text decoder with t-mBART, we use the same vocabulary as t-mBART. The reference target translation to calculate ASR-BLEU is normalized with lowercasing, removal of punctuation marks, conversion of digits to spoken forms, and removal of non-verbal words in parentheses like "(Applause)" or "(Music)."

Data filtering For discrete unit generation with a T2U model, we found that target discrete units were over-generated in long-form samples. We filtered out such samples by thresholding with a ratio of the sequence length of the discrete units over the number of corresponding source speech frames. We used a threshold of 0.7 for the multi-domain En→Es corpus while using ∞ for the rest. We used the same number of samples for all direct S2ST models for a fair comparison.

F Pre-training

In Table 7, we list all the pre-trained self-supervised models and TTS models used in §4.

wav2vec2.0 We use 24-layer Conformer wav2vec2.0 (Baevski et al., 2020) models trained on Libri-Light (Kahn et al., 2020) for En and VoxPopuli for Es, respectively.

w2v-BERT Same as Jia et al. (2019a), we pre-train the w2v-BERT (Chung et al., 2021) on approximately 430k hours of unlabeled speech data in 51 languages spanning from VoxPopuli, Common Voice, MLS, BABEL (Harper et al.; Gales et al., 2014), and VoxLingua107 (Valk and Alumäe, 2021). The w2v-BERT was composed of 24 Conformer layers with 0.6 billions of parameters.

Text-based mBART (t-mBART) We train a t-mBART model with En and Es unlabeled text on CC100 (Conneau et al., 2020). We use of a 65k unigram subword unit for the vocabulary. For multilingual experiments on CVSS-C, we use mBART50 (Tang et al., 2020) with multilingual fine-tuning to En. The vocabulary size is a 250k subword unit.

Unit-based mBART (u-mBART) We use a u-mBART model trained with En and Es unlabeled speech on VoxPopuli. The unit vocabulary is the same as that of the mHuBERT model.

G Architecture details

Let d_{model} be a model dimension of Transformer, d_{ff} be an inner dimension of the FFN layers, and N_{head} be the number of attention heads.

Speech encoder We used a 16-layer Conformer encoder stacked on 2-dimensional convolution blocks when training models from scratch. The convolution blocks reduced the input sequence length by a factor of 4. We set $(d_{\text{model}}, d_{\text{ff}}, N_{\text{head}})$ to $(256, 2048, 4)$. We set the kernel size of the depthwise convolution in the convolution module of each Conformer block to 31. When pre-training the encoder with wav2vec2.0 and w2v-BERT, we used a 24-layer Conformer encoder and stacked a one-layer length adaptor (Li et al., 2021) on it. Because an output frame of wav2vec2.0 and w2v-BERT corresponds to 20ms and the length adaptor halved the sequence length, the frame rate of every final encoder output corresponds to 40ms in both

ID	#GPU	# of frames × gradient accumulation	Learning rate	Warmup	Dropout	Label smoothing	Loss weight			R-Drop		
							w_{asr}	w_{s2t}	w_{ctc}	γ	β	α
A6	4	40k×1	1.3e-3		0.2		–	–	–	–	8.6	–
A7	16	2k×4	1.0e-3		0.1		–	–	–	–	8.6	–
A11	16	20k×1	1.0e-3		0.3		0.1	0.1	–	0.0	0.0	–
A12	16	4k×2	1.0e-3		0.1		–	–	–	–	–	–
A15	16	20k×1	1.5e-3	10k	0.3	0.2	0.1	0.1	–	3.0	3.0	–
A16	16	4k×2	1.0e-3		0.1		–	0.1	–	–	3.0	–
A18	4	20k×1	8.6e-4		0.3		8.0	8.0	1.6	1.0	1.0	1.0
A19	16	2k×4	1.0e-3		0.1		–	–	–	–	–	1.0
A20	4	20k×1	6.0e-4		0.3		8.0	8.0	–	3.0	3.0	1.0
A21	16	2k×4	1.0e-3		0.1		–	8.0	–	–	3.0	1.0
B3	8	35k×4	2.1e-3		0.1	0.1	0.6	–	–	4.6	4.6	–
B4	32	2k×24	1.0e-3		0.1	0.2	0.0	–	–	5.0	5.0	–
B6	32	40k×1	1.0e-3		0.1	0.2	–	0.1	–	–	0.0	–
B7	32	40k×1	1.0e-3		0.1	0.2	–	0.1	–	–	0.0	–
B8	32	2k×24	1.0e-3		0.1	0.2	–	–	–	–	–	–
B15	32	40k×1	1.1e-3		0.1	0.2	–	0.1	–	–	10.0	–
B16	32	40k×1	1.0e-3		0.1	0.2	–	0.1	–	–	10.0	–
B17	32	2k×24	1.0e-3	10k	0.1	0.2	–	0.1	–	–	5.0	–
B18	32	20k×2	8.6e-4		0.3	0.2	–	8.0	1.6	–	0.5	0.5
B19	32	20k×2	7.0e-4		0.3	0.2	–	8.0	1.6	–	0.5	0.5
B20	32	2k×24	1.0e-3		0.1	0.2	–	–	–	–	–	0.5
B21	32	20k×2	1.5e-3		0.3	0.2	–	8.0	–	–	1.5	1.5
B22	32	20k×2	7.0e-4		0.3	0.2	–	8.0	–	–	5.0	1.5
B23	32	2k×24	1.0e-3		0.1	0.2	–	8.0	–	–	5.0	1.5
C1′				1k		0.1	–	–	–	–	10.0	–
C2′				1k		0.2	–	–	–	–	10.0	–
C3				5k		0.2	–	8.0	–	–	10.0	–
C4	32	2k×30	5.0e-4	5k	0.1	0.2	–	8.0	–	–	10.0	–
C5′				1k		0.2	–	–	–	–	–	0.0
C6				1k		0.2	–	8.0	–	–	10.0	0.0
C7				1k		0.2	–	8.0	–	–	10.0	0.0

Table 8: Training hyperparameters

cases. In this case, we set $(d_{\text{model}}, d_{\text{ff}}, N_{\text{head}})$ to $(1024, 4096, 16)$.

S2SpecT We used a six-layer Transformer spectrogram decoder. We set $(d_{\text{model}}, d_{\text{ff}}, N_{\text{head}})$ to $(512, 2048, 8)$. When pre-training the speech encoder with wav2vec2.0 or w2v-BERT, we doubled these three values. We set the pre-net dimension and reduction factor of the spectrogram decoder to 32 and 3, respectively.

S2SpecT2 Let N_{t2s} be the depth of the T2S encoder. We set $(N_{\text{1st}}, N_{\text{2nd}}, N_{\text{t2s}})$ to $(4, 6, 2)$ on Fisher and CVSS-C. On the multi-domain corpus, we set $(N_{\text{1st}}, N_{\text{2nd}}, N_{\text{t2s}})$ to $(12, 6, 2)$ when pre-training the first-pass decoder with t-mBART. Otherwise, we set $(N_{\text{1st}}, N_{\text{2nd}}, N_{\text{t2s}})$ to $(6, 6, 2)$. We used the same d_{model}, d_{ff}, and N_{head} as S2SpecT in all the settings.

S2UT We used a six-layer Transformer unit decoder. When training models from scratch on Fisher, we set $(d_{\text{model}}, d_{\text{ff}}, N_{\text{head}})$ to $(256, 2048, 4)$. We set $(d_{\text{model}}, d_{\text{ff}}, N_{\text{head}})$ to $(512, 2048, 8)$ on CVSS-C. When pre-training the speech encoder with wav2vec2.0 or w2v-BERT, we set $(d_{\text{model}}, d_{\text{ff}}, N_{\text{head}})$ to $(1024, 4096, 16)$.

UnitY We used the same first-pass decoder as S2SpecT2 in all the settings. We set $(N_{\text{2nd}}, N_{\text{t2u}})$ to $(2, 2)$. We used the same d_{model}, d_{ff}, and N_{head} as the S2UT model in all the settings.

S2TT We used a six-layer Transformer decoder. When initializing it with t-mBART, we set the depth to 12.

ASR We used the same architecture as the S2TT model except for the vocabulary in all the settings.

H Training details

We optimized all models with the mixed precision training using 32GB V100 GPUs (Micikevicius et al., 2018). When fine-tuning the speech encoder from wav2vec2.0 and w2v-BERT, we updated all parameters in the speech encoder. For multilingual training with speech encoder pre-training with w2v-BERT on CVSS-C, we over-sampled training data of low-resource directions with an inverse temperature of 0.6, following (Arivazhagan et al., 2019). We list the training hyperparameters in Table 8. The training of `A*`, `B*`, and `C*` models converged within approximately 1, 3, and 5 days, respectively.

ID	Model	Encoder	ASR-BLEU (↑)		
			dev	dev2	test
A0	Synthetic target (Lee et al., 2022a)		88.5	89.4	90.5
Cascaded systems					
A1	ASR → MT → TTS	LSTM (Lee et al., 2022a)	42.1	43.5	43.9
A2	S2TT → TTS	LSTM (Jia et al., 2019b)	39.4	41.2	41.4
A3		LSTM (Jia et al., 2022b)	–	–	43.3
A4		LSTM (Lee et al., 2022a)	38.5	39.9	40.2
A5		Transformer (Dong et al., 2022)	44.3	45.4	45.1
A6		Conformer	47.8	48.9	48.3
A7		Conformer wav2vec2.0	51.0	52.2	52.1
Direct speech-to-spectrogram systems					
A8	S2SpecT	Transformer (Jia et al., 2019b)	30.1	31.5	31.1
A9		Transformer (Lee et al., 2022a)	–	–	33.2
A10		Transformer (Dong et al., 2022)	42.4	43.3	43.6
A11		Conformer	43.9	44.4	43.8
A12		Conformer wav2vec2.0	45.5	47.6	46.3
A13	Translatotron2	Conformer (Jia et al., 2022b)	–	–	42.4
A14		Conformer w2v-BERT (Li et al., 2022)	–	–	52.2
A15	S2SpecT2	Conformer	50.4	51.1	50.8
A16		Conformer wav2vec2.0	**58.4**	**59.5**	**58.6**
Direct speech-to-unit systems					
A17	S2UT	Transformer (Lee et al., 2022a)	–	–	39.9
A18		Conformer	46.2	47.6	47.4
A19		Conformer wav2vec2.0	53.4	53.9	53.7
A20	UnitY	Conformer	50.5	51.6	51.4
A21		Conformer wav2vec2.0	**55.1**	**56.5**	**55.9**

Table 9: ASR-BLEU on Fisher Es→En corpus. The decoder in all the models is initialized randomly. S2SpecT2 is our improved version of Translatotron2. Note that A10 uses pseudo labeled external resources with a cascaded S2ST system, and A13 uses data augmentation by concatenating multiple utterances.

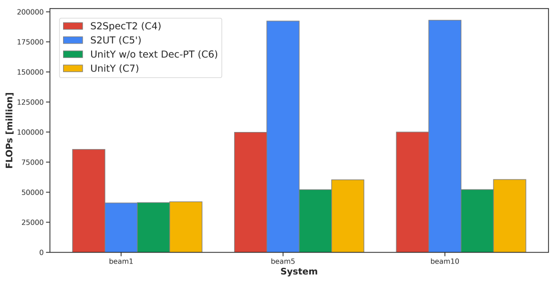


Figure 5: FLOPs of direct S2ST models on multi-domain Es→En corpus. The beam width of two-pass models corresponds to the first-pass decoder.

I Additional experimental results

In this section, we present additional experimental results in §4.

FLOPs In Figure 5, we show the results of FLOPs measured with a subset of the multi-domain Es→En dev set, as discussed in §4.3. UnitY achieved 1.65× and 3.19× FLOPs reduction over S2SpecT2 and S2UT models, respectively.

Fisher Es→En The results on Fisher are shown in Table 9. We report average scores over three runs with different random seeds. Among our four direct systems trained from scratch (A11, A15, A18, A20), UnitY (A20) achieved the best ASR-BLEU. Our S2UT (A18) and S2SpecT2 (A15) outperformed the previous studies (A13, A17) by a large margin.[9] Because S2SpecT2 outperformed S2UT, the two-pass decoding was the main factor of the improvements although it was complementary to targeting discrete units. Moreover, the two-pass direct models (A15, A20) outperformed a cascaded system (A6).

Next, we pre-trained the speech encoder with wav2vec2.0 (A12, A16, A19, A21).[10] We confirmed that all the models benefited from the pre-training, but the gain was small for S2SpecT. Unlike when training the models from scratch, S2SpecT2 gained the most and achieved the best test ASR-BLEU, 58.3. To the best of our knowledge, this is the new state-of-the-art S2ST result

[9] A15 predicts phonemes while A16 predicts subwords in the first pass.

[10] We did not pre-train the text decoder with t-mBART because it was not helpful on this corpus. This is because Fisher is a conversational domain, which is very different from text data used for t-mBART pre-training. We could make the text decoder pre-training effective by including conversational data during t-mBART pre-training, which we leave future work.

ID	Model	ASR-BLEU (↑)																					
		Avg.	High				Mid					Low											
			fr	de	ca	es	fa	it	ru	zh	pt	nl	tr	et	mn	ar	lv	sl	sv	cy	ta	ja	id
B0	Synthetic target$^\diamondsuit$	91.1	84.6	88.4	92.0	88.6	91.7	89.5	94.0	77.8	93.1	90.6	92.7	89.3	92.4	94.2	94.8	94.9	94.1	92.0	90.6	95.3	92.6
Cascaded systems																							
B1	S2TT → TTS$^\diamondsuit$	10.6	31.2	23.9	26.8	33.3	3.4	28.1	24.4	6.8	14.8	9.8	5.1	1.7	0.3	4.1	2.3	0.6	1.4	2.1	0.2	0.7	0.9
B2	+ ASR pre-training	12.7	32.9	26.2	28.6	34.9	5.6	30.2	27.1	8.7	19.8	14.4	10.7	3.2	0.6	7.8	2.8	2.0	3.4	5.0	0.2	0.9	1.6
B3	S2TT → TTS	7.8	18.3	16.1	18.5	19.9	4.2	18.1	17.6	3.7	15.8	11.5	6.5	2.1	0.2	2.2	1.3	2.3	1.0	2.9	0.2	0.3	0.3
B4	+ w2v-BERT + t-mBART	14.9	20.5	20.0	21.6	22.1	8.5	21.8	27.6	5.5	27.6	21.6	13.6	13.2	1.7	12.7	10.6	17.4	18.5	11.5	1.3	3.7	12.0
Direct speech-to-spectrogram systems																							
B5	Translatotron$^\diamondsuit$	3.4	15.5	6.9	11.0	14.1	1.4	9.3	4.3	1.5	2.2	2.1	1.2	0.1	0.1	0.1	0.2	0.3	0.4	0.3	0.1	0.2	0.1
B6	S2SpecT	7.6	24.1	17.8	20.3	25.1	1.8	20.3	18.7	2.5	9.8	9.0	3.8	0.5	0.1	0.5	1.2	0.8	1.3	0.7	0.1	0.2	0.2
B7	+ S2TT pre-training	9.6	26.3	20.1	21.8	27.5	6.2	22.3	21.9	5.7	12.6	11.4	9.1	2.7	0.3	4.3	1.3	1.8	1.5	4.0	0.3	0.6	0.7
B8	+ w2v-BERT	16.6	31.8	27.3	28.4	34.4	8.9	30.0	34.1	5.0	31.6	23.3	11.5	10.0	0.3	10.8	14.4	14.5	22.4	4.8	0.1	0.6	5.3
B9	Translatotron2$^\diamondsuit$	8.7	28.3	19.7	23.5	30.1	2.4	24.1	19.6	4.5	12.5	6.5	3.8	0.6	0.2	1.7	1.5	0.4	1.3	0.9	0.1	0.5	0.4
B10	+ Transformer decoder$^\spadesuit$	10.1	29.5	22.3	25.0	30.8	3.4	26.0	21.7	5.5	14.3	10.5	6.6	1.1	0.2	3.8	3.0	2.3	2.8	1.6	0.1	0.5	0.8
B11	+ S2TT pre-training$^\diamondsuit$	12.0	32.4	24.8	28.2	33.4	6.3	28.6	23.2	6.3	18.3	15.8	10.6	2.5	0.4	5.4	2.3	3.1	3.2	4.5	0.1	1.0	1.0
B12	+ w2v-BERT$^\spadesuit$	17.9	33.6	30.6	30.1	35.9	6.0	32.5	38.9	5.2	31.9	29.3	9.2	16.0	0.2	10.4	15.6	17.8	25.9	4.2	0.3	0.9	1.5
B13	+ mSLAM$^\spadesuit$	19.3	33.9	31.5	30.6	36.8	7.2	33.7	41.6	6.4	34.1	31.1	16.1	17.1	0.3	10.0	14.4	22.9	28.4	5.4	0.2	1.3	2.5
B14	++ TTS augmentation$^\spadesuit$	22.0	34.5	32.0	30.7	37.1	8.2	33.8	**42.6**	10.6	34.0	31.8	**23.9**	17.2	1.1	22.4	15.6	23.3	31.1	7.6	0.6	5.5	18.5
B15	S2SpecT2	11.3	31.7	25.9	27.4	32.8	4.6	28.4	27.5	7.0	18.0	15.4	9.2	1.7	0.3	1.7	2.5	1.3	1.8	1.9	0.2	0.7	1.0
B16	+ S2TT pre-training	13.1	31.9	26.1	28.0	33.3	7.9	28.8	28.6	8.5	20.3	17.8	13.9	4.6	0.6	6.4	2.6	4.8	2.4	7.4	0.4	0.6	1.2
B17	+ w2v-BERT + t-mBART	18.6	32.5	30.9	31.0	34.1	13.9	30.7	36.9	10.6	31.2	26.1	18.4	11.6	1.9	14.7	10.4	15.1	16.2	10.6	1.1	3.9	9.7
Direct speech-to-unit systems																							
B18	S2UT	9.1	28.3	21.7	24.6	29.0	2.5	25.2	21.7	4.0	11.1	10.2	4.9	0.8	0.1	0.9	1.8	1.4	1.2	0.5	0.1	0.4	0.7
B19	+ S2TT pre-training	11.4	29.4	23.3	25.7	30.5	7.4	26.5	24.6	6.9	16.7	15.6	10.6	3.3	0.5	4.6	2.2	2.6	1.4	4.7	0.3	0.9	1.0
B20	+ w2v-BERT + u-mBART	20.8	32.7	28.5	30.6	34.8	12.8	31.7	37.5	7.6	37.2	27.2	18.2	15.0	1.8	18.6	18.5	20.5	29.8	13.1	1.3	4.0	16.2
B21	UnitY	12.0	30.9	25.5	27.2	32.3	5.1	28.2	28.2	7.2	20.3	17.1	9.1	2.5	0.4	2.2	3.7	6.1	1.8	2.3	0.1	1.2	1.0
B22	+ S2TT pre-training	13.0	32.1	26.8	29.1	33.4	8.3	29.4	27.6	7.9	20.3	19.7	12.1	3.5	0.6	4.6	2.5	4.9	1.9	5.8	0.3	1.0	1.0
B23	+ w2v-BERT + t-mBART	**24.5**	**35.2**	**32.6**	**33.3**	**37.2**	**14.9**	**35.0**	**42.3**	**10.8**	**41.7**	**32.5**	**22.2**	**18.7**	**2.7**	**24.6**	**21.3**	**26.6**	**34.1**	**16.5**	**1.8**	**8.0**	**22.9**

Table 10: Full results of ASR-BLEU on CVSS-C corpus. $^\diamondsuit$Results from (Jia et al., 2022c), $^\spadesuit$Results from (Jia et al., 2022a). We use the S2TT model in B3 for S2TT pre-training. t-mBART and u-mBART stand for text-based mBART and unit-based mBART, respectively. All w2v-BERT and mSLAM encoders have 0.6B parameters.

ID	Model	Initialization of first-pass decoder	(ASR-)BLEU (↑)	
			Text	Speech
F1	UnitY	Random	34.8	30.7
F2		t-mBART	**38.3**	**33.2**
F3		Unsupervised MT	38.2	**33.2**
F4		Supervised MT1	36.6	33.0
F5		Supervised MT2	37.5	**33.3**
F6		S2TT (F7)	37.8	32.5
F7	S2TT	t-mBART	38.0	–

Table 11: Results of pre-training strategies for the first-pass decoder in UnitY on multi-domain Es→En dev set

on this corpus. However, UnitY has an advantage of decoding efficiency over S2SpecT2 as discussed in §4.3. All direct models (A16, A19, A21) except for S2SpecT outperformed the corresponding cascaded system (A7).

CVSS-C We show the full results of each language direction on CVSS-C in Table 10.

Pre-training first-pass decoder We explored a better pre-training strategy for the first-pass text decoder in UnitY. We investigated pre-training it with an MT model trained with bitext data from scratch (*Supervised MT1*, *Supervised MT2*). Supervised MT1 used CCMatrix (Schwenk et al., 2021) while Supervised MT2 is the MT model in the cascaded system[11]. Moreover, we fine-tuned the t-mBART model to the MT task in an unsupervised MT way via online back translation (Liu et al., 2020) on CC100 (*unsupervised MT*). Furthermore, we studied initializing the speech encoder and the text decoder with a separate direct S2TT model. The S2TT model was fine-tuned from wav2vec2.0 and t-mBART models on the same corpus. After the initialization, we fine-tuned the whole parameters of UnitY except FFN layers in the first-pass text decoder (*S2TT*).

The results in Table 11 showed that pre-training the first-pass decoder with the vanilla t-mBART (F2) or the unsupervised MT model (F3) was the most effective. Pre-training with supervised MT models (F4, F5) did not improve performance, even for the first pass. This is consistent with a finding in Jia et al. (2022a) although they pre-train the first-pass phoneme decoder of Translatotron2 with a phoneme-based supervised MT model. Therefore, leveraging a separate MT system is effective for generating weak supervisions (Popuri et al., 2022) rather than parameter initialization. Pre-training a part of UnitY with an independent S2TT model (F7) was not helpful either. Surprisingly, the BLEU score from the text decoder in UnitY was

[11] We used OpenSubtitle2018, UNCorpus, EUBookshop v2, Europarl v10, Wikipedia v1.0, and TED2020 v1 for training.

better than that of `F7`. Therefore, training signals from the unit decoder never affect the text decoder.

Ablation study In Table 12, we show full results of the ablation study presented in §5.1. An auxiliary CTC objective for the unit decoder, as used for the S2UT model (Lee et al., 2022a), was not helpful for UnitY (`D10`). This was because the introduction of the first-pass decoder already eased for the second-pass decoder to learn monotonic alignments.

Output unit for first-pass decoder In Table 13, we show full results of the comparison of the output units for the first-pass decoder in two-pass direct S2ST models presented in §5.2. The results showed that the subword unit was the best for UnitY regardless of pre-training the speech encoder with wav2vec2.0. In contrast, in the case of S2SpecT2, the best output unit differed according to whether we pre-trained the speech encoder or not. The phoneme unit was best when training the model from scratch (`E1`) while the subword unit was best when pre-training the encoder (`E3′`). However, predicting subwords in the first pass led to the best BLEU score for the text output in all the settings.

ASR-chrF Following a finding that ASR-chrF is a more robust evaluation metric than ASR-BLEU in Salesky et al. (2021), we also calculated ASR-chrF on Fisher, CVSS-C, and multi-domain corpora in Table 14, Table 15, and Table 16, respectively. Overall, we confirmed the similar trends to ASR-BLEU.

I.1 Human evaluation

Finally, we conducted an audio-only human evaluation to assess the translation quality while removing the necessity of ASR systems. We adopted cross-lingual semantic textual similarity (XSTS) (Licht et al., 2022) and percent acceptable translations.

Mean translation score We used XSTS, which emphasizes adequacy rather than fluency, as the most appropriate human evaluation protocol. Annotators judged the semantic similarity between the source and the translated sentence. As a result, whether a translation conveys the original meaning is more important than whether it has perfect syntax, wording, and grammar. Annotators assigned each item a score from one to five. A score of no less than three means the meaning is at least "mostly equivalent." We treat a translation that received a score of no less than three as having "acceptable" quality. Annotators need to be bilingual, as they compare the source and translated sentences directly. Since XSTS is an audio-only evaluation metric, it also considers the audio quality.

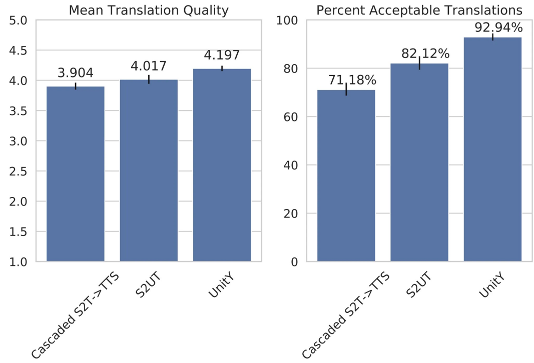


Figure 6: Results of human evaluation on multi-domain Es→En corpus

For each system, we computed the average XSTS score across items. We set a target of over four average XSTS for systems where we expect or desire high-quality translations. We set a target of over three average XSTS for systems where we expect a medium level of quality.

Percent acceptable translations For each system, we also computed the percentage of items that received an XSTS score of three or above. We refer to this as the percent acceptable translations. This metric helps us understand what percentage of translations produced by the system can preserve meaning adequately and what percentage has very low and unacceptable quality. This metric tends to be more stable and less sensitive to annotator agreement than the average XSTS score.

Evaluation setting We used the mTEDx test set (989 samples) and generated the target audio from the S2ST systems. Moreover, we randomly sampled 495 samples and generated the target audio from the reference translation followed by TTS. The reference translations serve as a reference point and a ceiling against which to compare our systems. Three bilingual annotators evaluated each item and assigned it a score from one to five. The median score was taken per item.

Results The results are presented in Figure 6.[12] We confirmed that UnitY consistently outperformed the cascaded and S2UT models in both metrics.

[12] The models used here are early versions and slightly different from the models in Table 2.

ID	Model	(ASR-)BLEU (↑)			
		Fisher		Multi-domain Es→En	
		Text	Speech	Text	Speech
D1	S2SpecT2	**54.4**	**49.2**	**35.0**	**30.8**
D2	+ w/o T2S encoder	**54.3**	17.4	34.9	25.0
D3	+ w/o R-Drop	51.6	45.9	34.8	30.3
D5	UnitY	**55.4**	**50.5**	**38.3**	**33.2**
D6	+ w/o T2U encoder	55.0	49.1	38.1	30.7
D7	+ w/o R-Drop	53.2	48.2	37.7	32.1
D8	+ Cross-attention to speech encoder (sequential)	**55.4**	50.3	38.2	**33.2**
D9	+ Cross-attention to speech encoder (parallel)	55.3	50.4	38.1	33.1
D10	+ CTC on unit decoder	55.3	50.2	n/a	n/a

Table 12: Ablation study for two-pass direct S2ST models on Fisher Es→En and multi-domain Es→En dev sets. The first-pass decoder in all the models on Fisher is initialized randomly while it is pre-trained with t-mBART on multi-domain corpora.

ID	Encoder pre-training	Model	Output unit	(ASR-)BLEU (↑)			
				Fisher		Multi-domain Es→En	
				Text	Speech	Text	Speech
E1		S2SpecT2	Phoneme	–	**50.4**	–	–
E2		S2SpecT2	Character	54.0	50.2	–	–
E3		S2SpecT2	Subword	**54.4**	49.2	–	–
E1′	✓	S2SpecT2	Phoneme	–	58.1	–	29.4
E2′	✓	S2SpecT2	Character	61.5	58.1	31.7	28.9
E3′	✓	S2SpecT2	Subword	**62.0**	**58.4**	**33.0**	**30.0**
E4		UnitY	Phoneme	–	49.8	–	–
E5		UnitY	Character	53.7	48.9	–	–
E6		UnitY	Subword	**55.4**	**50.5**	–	–
E4′	✓	UnitY	Phoneme	–	54.7	–	27.8
E5′	✓	UnitY	Character	60.9	55.0	33.2	29.6
E6′	✓	UnitY	Subword	**61.2**	**55.1**	**34.1**	**30.1**

Table 13: Results of output units for the first-pass decoder in two-pass direct S2ST models on Fisher Es→En and multi-domain Es→En dev sets. We use 1k and 2k units for the subword vocabulary on Fisher and multi-domain Es→En corpora, respectively. The first-pass decoder in all the models is initialized randomly.

ID	Model	Encoder	ASR-chrF (↑)		
			dev	dev2	test
Cascaded systems					
A6	S2TT → TTS	Conformer	0.642	0.652	0.649
A7		Conformer wav2vec2.0	0.671	0.684	0.680
Direct speech-to-spectrogram systems					
A11	S2SpecT	Conformer	0.612	0.621	0.618
A12		Conformer wav2vec2.0	0.638	0.655	0.649
A15	S2SpecT2	Conformer	0.649	0.661	0.657
A16		Conformer wav2vec2.0	**0.695**	**0.708**	**0.702**
Direct speech-to-unit systems					
A18	S2UT	Conformer	0.626	0.642	0.643
A19		Conformer wav2vec2.0	0.677	0.688	0.685
A20	UnitY	Conformer	0.646	0.658	0.658
A21		Conformer wav2vec2.0	**0.678**	**0.692**	**0.687**

Table 14: ASR-chrF on Fisher Es→En corpus. The decoder in all the models is initialized randomly. S2SpecT2 is our improved version of Translatotron2.

ID	Model	ASR-chrF (↑)			
		Avg.	High	Mid	Low
Cascaded systems					
B3	S2TT → TTS	0.304	0.504	0.384	0.204
B4	+ w2v-BERT + t-mBART	0.420	0.533	0.463	0.365
Direct speech-to-spectrogram systems					
B6	S2SpecT	0.273	0.498	0.328	0.175
B7	+ S2TT pre-training	0.311	0.521	0.377	0.213
B8	+ w2v-BERT	0.395	0.582	0.461	0.306
B15	S2SpecT2	0.306	0.560	0.389	0.187
B16	+ S2TT pre-training	0.336	0.566	0.417	0.226
B17	+ w2v-BERT + t-mBART	0.419	0.592	0.492	0.331
Direct speech-to-unit systems					
B18	S2UT	0.294	0.536	0.356	0.188
B19	+ S2TT pre-training	0.329	0.550	0.405	0.224
B20	+ w2v-BERT + u-mBART	0.445	0.588	0.495	0.377
B21	UnitY	0.312	0.564	0.396	0.192
B22	+ S2TT pre-training	0.333	0.572	0.415	0.220
B23	+ w2v-BERT + t-mBART	**0.474**	**0.607**	**0.521**	**0.410**

Table 15: ASR-chrF on CVSS-C corpus. We use the S2TT model in B3 for S2TT pre-training. t-mBART and u-mBART stand for text-based mBART and unit-based mBART, respectively. All w2v-BERT encoders have 0.6B parameters.

ID	Model	ASR-chrF (↑)						
		En→Es			**Es→En**			
		Europarl-ST	MuST-C	Avg.	CoVoST-2	Europarl-ST	mTEDx	Avg.
Cascaded systems								
C1′	ASR→MT→TTS	0.634	0.587	0.611	0.611	0.618	0.569	0.599
C2′	S2TT→TTS	0.639	0.613	0.626	0.642	0.620	0.588	0.620
Direct speech-to-spectrogram systems								
C3	S2SpecT2 (6L→6L)	0.634	0.606	0.620	**0.642**	0.484	0.578	0.568
C4	+ t-mBART (12L→6L)	**0.642**	**0.611**	**0.627**	**0.642**	0.485	**0.583**	0.570
Direct speech-to-unit systems								
C5′	S2UT + u-mBART	0.610	0.615	0.613	0.621	0.587	0.568	0.592
C6	UnitY (6L→6L)	**0.643**	0.618	0.631	0.628	0.591	0.575	0.598
C7	+ t-mBART (12L→2L)	0.641	**0.622**	**0.632**	**0.633**	**0.606**	**0.583**	**0.607**

Table 16: ASR-chrF on multi-domain En↔Es. The encoder in all the models is pre-trained with wav2vec2.0. t-mBART and u-mBART stand for text-based mBART and unit-based mBART, respectively. N_{1st}L→ N_{2nd}L stands for an N_{1st}-layer first-pass decoder with an N_{2nd}-layer second-pass decoder.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 8

☑ A2. Did you discuss any potential risks of your work?
Section 8

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 3.1 and Appendix D

☑ B1. Did you cite the creators of artifacts you used?
Section 3.1 and Appendix D

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We used public corpora only. All the corpora are open-sourced except Fisher Spanish.The license is described in the corpus-related papers we cited.

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
We didn't modify any contents in the public corpora we used.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
They are described in the corpus-related papers we cited.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 3.1 and Appendix D

☒ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
We followed the same setting as previous works.

C ☑ Did you run computational experiments?

Section 4.3

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 5.3

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 3, Appendix E, F, G, H

☒ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
We reported average scores over 3 runs for Fisher and scores with a single run for the rest large corpora. We didn't run multiple same jobs for the latter because they are computationally expensive.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 3, Appendix E, F, G, H

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☒ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
We didn't hire any human annotators in our experiments.

☒ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
We didn't hire any human annotators in our experiments.

☒ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
We didn't hire any human annotators in our experiments.

☒ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
We didn't hire any human annotators in our experiments.

☒ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
We didn't hire any human annotators in our experiments.