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

Hirofumi Inaguma[♡], **Sravya Popuri**[♡], **Ilia Kulikov**[♡], **Peng-Jen Chen**[♡], **Changhan Wang**[♡], **Yun Tang**[♡], **Ann Lee**[♡], **Shinji Watanabe**[♣], **Juan Pino**[♡]
Meta AI[♡], Carnegie Mellon University[♣]

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, *UnitY*, 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 up to 2.5 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 (Li et al., 2021; 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 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

[1] In (Lee et al., 2022a), a direct S2ST model is defined as a model that does not use intermediate text representations. On the other hand, in (Jia et al., 2022b), it is defined 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.

phoneme 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 assigns more search spaces to the first pass. We further advance the model performance by introducing a text-to-unit (T2U) encoder between the two decoders to bridge the gap between textual and acoustic representations. We also adopt R-Drop regularization (Wu et al., 2021) to avoid over-fitting in the first pass and improve the translation quality. 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 up to 2.5 ASR-BLEU improvement 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. The improvement holds regardless of the data size and the use of self-supervised 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,

[2]Code is available at https://github.com/facebookresearch/fairseq/examples/speech_to_speech/unity.

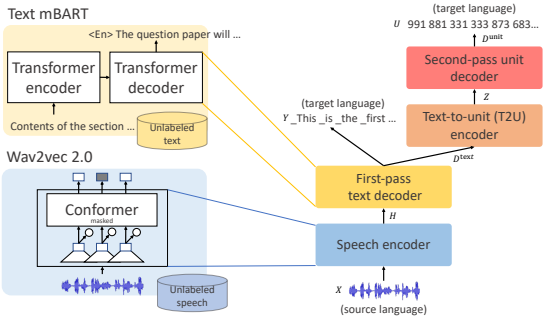

Figure 1: Model architecture of UnitY

and $Y = (y_1, \cdots, y_M)$ and $U = (u_1, \cdots, u_L)$ be the corresponding reference text translation and discrete unit sequences in the target language, respectively. Note that there is no duration information 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 the spectrogram 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 `TextDec` 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:

$$\mathcal{L}_{\text{s2t}}(Y|X) = -\frac{1}{M}\sum_{i=1}^{M}\log P_{\text{s2t}}(Y_i|X, Y_{<i})$$
$$= -\frac{1}{M}\sum_{i=1}^{M}\log P_{\text{s2t}}(Y_i|D_i^{\text{text}})$$
$$D_i^{\text{text}} = \texttt{TextDec}(H, Y_{<i}),$$

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 follows:

$$Z = \texttt{T2UEnc}(D^{\text{text}}).$$

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 `UnitDec` 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:

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

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 speech 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).

2.2 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 formulated as follows:

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

where X_i is a duplicated input from X, $\mathcal{L}_{\text{kl}}^{\text{s2u}}$ and $\mathcal{L}_{\text{kl}}^{\text{s2t}}$ are R-Drop losses for the unit and text decoders, w_{s2t} is a weight for the S2TT loss, and α and β are weights for the R-Drop losses, respectively. We implement R-Drop by duplicating inputs instead of feeding them to the network twice. The mathematical formulation of $\mathcal{L}_{\text{kl}}^{\text{s2u}}$ and $\mathcal{L}_{\text{kl}}^{\text{s2t}}$ is described in Appendix A.1.

2.3 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;

[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 Section 5.1

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 S2UT model with a unit-based mBART, which is 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 with a text-based mBART, which can be pre-trained with unlabeled text data in a self-supervised way. To this end, we use the same vocabulary as the mBART. An advantage of using such a large vocabulary is improving the inference speed thanks to the shortened text sequence length. 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. Note that we initialize the T2U encoder and second-pass unit decoder randomly.

2.4 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 inputs 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.

2.5 Deep-shallow two-pass decoders

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

In this section, we describe experimental settings for our experiments in Section 4.

3.1 Data

We use three datasets: Fisher Es→En (Post et al., 2013), CVSS-C (Jia et al., 2022c), and mutli-domain En↔Es (Popuri et al., 2022) corpora.

Fisher Es→En This corpus contains 170-hour Spanish conversational telephone speech with the corresponding Spanish 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 following (Lee et al., 2022a).

CVSS-C CVSS is a public multilingual S2ST corpus based on CoVoST2 (Wang et al., 2021b). It covers 21 language 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 many-to-English multilingual model.

Multi-domain En↔Es 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 used a T2U model (Lee et al., 2022b) for direct speech-to-unit models and a TTS model for the rest. Both T2U and TTS models are based on Transformer (Vaswani et al., 2017).[4]

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 (version 7.0) (Ardila et al., 2020), resulting in 1983-hour source speech. For **Es→En**, we use all samples from CoVoST2, Europarl-ST, and

[4] We train En and Es T2U/TTS models on the LJSpeech (Ito and Johnson, 2017) and CSS10 (Park and Mulc, 2019) corpora, respectively.

mTEDx (Elizabeth et al., 2021), and augment the training data by Common Voice and MLS (Pratap et al., 2020), resulting in 1404-hour source speech. More details are summarized in Appendix A.5.

3.2 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 the Fisher corpus (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 use vocabularies of 1k, 6k, and 2k unigram subword units (Kudo, 2018) built 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 mBART, we use the same vocabulary as mBART.

3.3 Pre-training

We use the same wav2vec2.0 and unit mBART models as (Popuri et al., 2022). All the models are publicly available, we list the URLs in Appendix A.6.

Wav2vec2.0 (encoder pre-training) 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.

Text mBART (decoder pre-training) We train a text 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.

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

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 systems with wav2vec2.0, we also pre-train that of ASR and S2TT models in the cascaded systems with the same wav2vec2.0 for a fair comparison. R-Drop is applied to all the models that predict discrete symbols. The training objective of each system is described in Appendix A.2.

Cascaded (ASR→MT→TTS) We combine a Conformer ASR, a Transformer MT, and a Transformer TTS model.

Cascaded (S2TT→TTS) We combine a Conformer direct S2TT model and a Transformer TTS model. For the multi-domain corpora, we pre-train the S2TT's decoder with mBART.

Translatotron We build a direct S2ST model that predicts spectrogram with a single decoder based on Transformer, similar to (Lee et al., 2022a).

Translatotron2+ We train an improved version of Translatotron2 (Jia et al., 2022b) by enhancing the architecture and training with the proposed methods for UnitY. Firstly, we replace phoneme targets with subwords in the first pass. Secondly, we introduce an additional encoder between text and spectrogram decoders, which we refer to as a text-to-speech (T2S) encoder. The second-pass decoder attends to the T2S encoder output only. Unlike (Jia et al., 2022b), we use an autoregressive Transformer decoder instead of a non-attentive Tacotron (NAT) (Shen et al., 2020) for the second-pass spectrogram decoder. Lastly, we apply R-Drop to the first-pass decoder.

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

3.5 Vocoder

We use a HiFi-GAN vocoder (Kong et al., 2020) to convert the spectrogram 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	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
A7		Conformer wav2vec2.0	51.0	52.2	52.1
Direct systems (speech-to-spectrogram)					
A8	Translatotron	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	Translatotron2+	Conformer	50.4	51.1	50.8
A15		Conformer wav2vec2.0	**58.4**	**59.5**	**58.6**
Direct systems (speech-to-unit)					
A16	S2UT	Transformer (Lee et al., 2022a)	–	–	39.9
A17		Conformer	46.2	47.6	47.4
A18		Conformer wav2vec2.0	53.4	53.9	53.7
A19	UnitY	Conformer	50.5	51.6	51.4
A20		Conformer wav2vec2.0	**55.1**	**56.5**	**55.9**

Table 1: ASR-BLEU on Fisher Es→En corpus. Decoders in all the models are initialized randomly. Translatotron2+ 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.

3.6 Architecture

Let N_{1st}, N_{2nd}, and N_{t2u} be the depth of the first-pass decoder, second-pass decoder, and T2U encoder, respectively. We use $(N_{1st}, N_{2nd}, N_{t2u}) = (4, 2, 2)$ on the Fisher and CVSS-C corpora. On the multi-domain corpus, we use $(N_{1st}, N_{2nd}, N_{t2u}) = (12, 2, 2)$ when pre-training the first-pass decoder with mBART. Otherwise, we use $(N_{1st}, N_{2nd}, N_{t2u}) = (6, 6, 2)$. We describe the other architecture configurations in Appendix A.3.

3.7 Training

We optimize all models with the mixed precision training (Micikevicius et al., 2018). 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 A.4.

3.8 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 for Translatotron2+. Note that beam search is not involved in spectrogram generation.

3.9 Evaluation

Following the previous works (Lee et al., 2022a; Popuri et al., 2022), 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 scores. 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. The reference target translation 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)."

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 architecture (single pass v.s. two pass) in both supervised and semi-supervised settings. We also benchmark the decoding efficiency of direct S2ST models.

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	+ ST&ASR PT	12.7	30.7	18.3	4.4
Direct systems (speech-to-spectrogram)					
B3	Translatotron$^{\diamondsuit}$	3.4	11.9	3.5	0.3
B4	Translatotron	7.6	21.8	10.6	1.5
B5	+ ST&ASR PT	9.6	23.9	13.8	3.2
B6	Translatotron2$^{\diamondsuit}$	8.7	25.4	12.6	1.5
B7	+ Transformer decoder$^{\spadesuit}$	10.1	26.9	14.2	2.8
B8	+ ST&ASR PT$^{\diamondsuit}$	12.0	29.7	16.6	4.2
B9	Translatotron2+	11.3	29.1	16.9	3.1
B10	+ ST&ASR PT	**13.1**	**29.8**	**18.8**	**5.2**
Direct systems (speech-to-unit)					
B11	S2UT	9.1	25.9	12.9	1.9
B12	+ ST&ASR PT	11.4	27.2	16.4	4.0
B13	UnitY	12.0	29.0	17.8	4.0
B14	+ ST&ASR PT	**13.0**	**30.4**	**18.7**	**4.8**

Table 2: ASR-BLEU on CVSS-C corpus. $^{\diamondsuit}$Results from (Jia et al., 2022c), $^{\spadesuit}$Results from (Jia et al., 2022a). Decoders in all the models are initialized randomly.

4.1 Fisher Es→En

The results on the Fisher Es→En corpus are shown in Table 1. We first compared four direct systems trained from scratch (A11, A14, A17, A19). Our Conformer-based S2UT (A17) and Translatotron2+ (A14) outperformed the previous studies by a large margin. Note that Translatotron2+ is an improved version of the original work in (Jia et al., 2022b) by using the same techniques proposed for UnitY in Section 2.[5] Among them, UnitY (A19) achieved the best ASR-BLEU scores. Therefore, the two-pass decoding is the main factor of the improvements but complementary to targeting discrete units.

Next, we pre-trained the speech encoder with wav2vec2.0 (A12, A15, A18, A20).[6] We confirmed that all the models benefited from the pre-training, but the gain was small for Translatotron. Unlike when training the models from scratch, Translatotron2+ gained the most and achieved the best test ASR-BLEU, 58.3. However, UnitY has an advantage of decoding efficiency over Translatotron2+, which we will discuss in Section 4.4. Lastly, the direct models except for Translatotron outperformed a strong cascaded system pre-trained on the same wav2vec2.0 by a large margin.

4.2 CVSS-C

The results on the CVSS-C corpus are listed in Table 2. We observed consistent trends with the results on the Fisher corpus. UnitY outperformed the S2UT model by 1.6 and 2.9 ASR-BLEU on average with and without encoder pre-training with an S2TT model, respectively. The encoder pre-training improved ASR-BLEU scores of all the direct models, similar to (Jia et al., 2022c). Translatotron2+ also achieved similar performances to UnitY and outperformed Translatotron2 by 1.1 ASR-BLEU on average.

4.3 Multi-domain En↔Es

We present results on the multi-domain En↔Es corpora (Popuri et al., 2022) in Table 3. C5′ is our reproduced model of C5. We observed that UnitY with text decoder pre-training (C7) improved the S2UT model with unit decoder pre-training (C5′) by 1.3 and 2.5 ASR-BLEU on average in En→Es and Es→En directions, respectively. This confirms the effectiveness of the two-pass modeling still hold in the high-resource scenario. Furthermore, UnitY without text decoder pre-training (C6) already outperformed C5′ and degraded from C7 only slightly.

Comparing UnitY and Translatotron2+, we observed mixed results that UnitY outperformed Translatotron2+ in Es→En except for CoVoST2 while Translatotron2+ performed better in En→Es. However, the proposed text decoder pre-training was still helpful for Translatotron2+, especially in En→Es, showing the versatility of our method. We also confirmed that UnitY outperformed strong cascaded systems in most test sets if we use the same amount of data.

4.4 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.

[5] A14 predicts phonemes while A14 predicts subwords in the first pass.

[6] We did not pre-train the text decoder with 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 mBART pre-training. We could make the text decoder pre-training effective by including conversational data during mBART pre-training, which we leave future work.

ID	Model	ASR-BLEU (↑)						
		En→Es (1983 hours)			**Es→En** (1404 hours)			
		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
C2	S2TT→TTS$^{\diamondsuit}$	32.6	30.1	31.4	28.4	23.6	21.5	24.5
Direct systems (speech-to-spectrogram)								
C3	Translatotron2+	36.0	34.0	35.0	37.0	23.4	31.3	30.6
C4	+ Text Dec-PT	**37.2**	**34.5**	**35.8**	**37.2**	23.7	31.7	30.9
Direct systems (speech-to-unit)								
C5	S2UT + Unit Dec-PT$^{\diamondsuit}$	33.6	33.7	33.7	33.5	28.6	29.1	30.4
C5′	S2UT + Unit Dec-PT	33.5	33.3	33.4	34.5	29.9	29.9	31.4
C6	UnitY	35.1	33.7	34.4	35.4	30.8	31.3	32.5
C7	+ Text Dec-PT	**35.3**	**34.1**	**34.7**	**36.4**	**33.1**	**32.2**	**33.9**

Table 3: 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. XX Dec-PT stands for pre-training of the first-pass decoder with a XX-based mBART model.

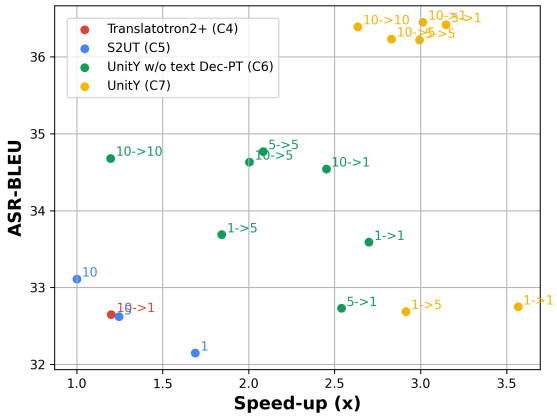

Figure 2: Runtime of direct S2ST models on multi-domain Es→En corpus. X→Y on the top right of each data point represents the beam width in each decoder pass.

The results in Figure 2 showed that UnitY achieved 2.51× and 2.83× decoding speed-ups over Translatotron2+ and S2UT models, respectively. These confirms the efficiency of discrete unit prediction and two-pass decoding, thanks to reduced output sequence lengths. Using a large vocabulary in the first pass via the text decoder pre-training 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 more ambiguities exist in the first pass than in the second pass. Therefore, we can obtain better translation quality and decoding speed by assigning more computation time to the first pass.

ID	Model	(ASR-)BLEU (↑)	
		Speech	Text
D1	Translatotron2+	**47.8**	**54.3**
D2	+ w/o T2S encoder	17.4	**54.3**
D3	+ w/o R-Drop	45.9	51.6
D5	UnitY	**50.5**	**55.4**
D6	+ w/o T2U encoder	49.1	55.0
D7	+ w/o R-Drop	48.2	53.2
D8	+ CA to speech encoder (seq)	50.3	**55.4**
D9	+ CA to speech encoder (para)	50.4	55.3
D10	+ CTC on unit decoder	50.2	55.3

Table 4: Ablation study for two-pass direct S2ST models on the Fisher Es→En dev set. *CA* stands for cross-attention.

We also present the results of FLOPs in Appendix A.7.1. To summarize, UnitY achieved 1.65× and 1.16× FLOPs reduction over Translatotron2+ and S2UT models, respectively.

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 Translatotron2+. We use the Fisher Es→En and multi-domain Es→En corpora, but pseudo-labeled ASR data is excluded from the latter for quick exploration, resulting in 196-hour source speech. We report average dev scores over three runs with different random seeds.

5.1 Ablation study

We first conducted an ablation study for two-pass direct S2ST models in Table 4. We evaluated the

ID	Model	Output unit	(ASR-)BLEU (↑)	
			Speech	Text
E1	Translatotron2+	Phoneme	**50.4**	–
E2		Character	50.2	54.0
E3		Subword	49.2	**54.4**
E4	UnitY	Phoneme	49.8	–
E5		Character	48.9	53.7
E6		Subword	**50.5**	**55.4**

Table 5: Results of output units for the first-pass decoder in two-pass direct S2ST models on the Fisher Es→En dev set

ID	Model	Initialization of first-pass decoder	(ASR-)BLEU (↑)	
			Speech	Text
F1	UnitY	Random	30.7	34.8
F2		mBART	**33.2**	**38.3**
F3		Unsupervised MT	**33.2**	38.2
F4		Supervised MT1	32.9	36.7
F5		Supervised MT2	**33.3**	37.5
F6		S2TT (F8)	32.5	37.8
F8	S2TT	mBART	–	38.0

Table 6: Results of pre-training strategies for the first-pass decoder in UnitY on the multi-domain Es→En dev set

translation quality of outputs from both decoders. It was effective to introduce a T2U/T2S encoder between the first-pass and second-pass decoders, especially for Translatotron2+ (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. An additional T2U/T2S encoder was essential for bridging the gap in representations between the first-pass and second-pass decoders. 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 Section 2.1. We expected that the first-pass decoder output lost useful information to generate target speech faithful to source speech. We explored parallel (*para*, D8) and sequential (*seq*, 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. 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.

5.2 Output unit for first-pass decoder

We studied optimal granularity of the output unit for the first-pass decoder in Translatotron2+ and UnitY. As an output unit, we explored phonemes, characters, and subwords (1k).

The results in Table 5 showed that the subword unit in the first-pass decoder (E6) was the most effective for UnitY thanks to a better translation quality in the first pass. On the other hand, the phoneme unit (E1) was best for Translatotron2+. However, we found that the subword unit outperformed the phoneme unit when pre-training the encoder of Translatotron2+ on the Fisher corpus (see Appendix A.7.2 for the full results).

5.3 Pre-training strategy for first-pass text decoder

We explored a better pre-training strategy for the first-pass text decoder in UnitY. We investigated pre-training the text decoder 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[7]. Moreover, we fine-tuned the 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 fine-tuned from wav2vec2.0 and mBART models. After the initialization, we fine-tuned the whole parameters of UnitY (**S2TT**).

Hereafter, we use the multi-domain Es→En corpus because it is the only corpus we pre-trained the first-pass text decoder with mBART. The results in Table 6 showed that pre-training the text decoder with the vanilla 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

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

ID	Decoder depth		#Params (Billion)	(ASR-)BLEU (↑)	
	First pass	Second pass		Speech	Text
G1	2	6	0.79	30.3	34.5
G2	4	6	0.82	30.5	34.5
G3	6	2	0.79	30.3	34.3
G4	6	4	0.82	29.9	33.9
G5	6	6	0.86	**30.7**	**34.8**
G6	6	8	0.89	30.2	34.2
G7	6	12♠	0.96	29.8	33.7
G8	12	2	0.95	30.7	34.9
G9	12♠	2	0.95	**33.2**	**38.3**
G10	12♠	12♠	1.12	32.2	36.2

Table 7: Results of capacity assignment to two-pass decoders in UnitY on the multi-domain Es→En dev set. ♠Pre-trained with the corresponding mBART model, where we set the number of layers to 12 because of the availability of the pre-trained mBART model.

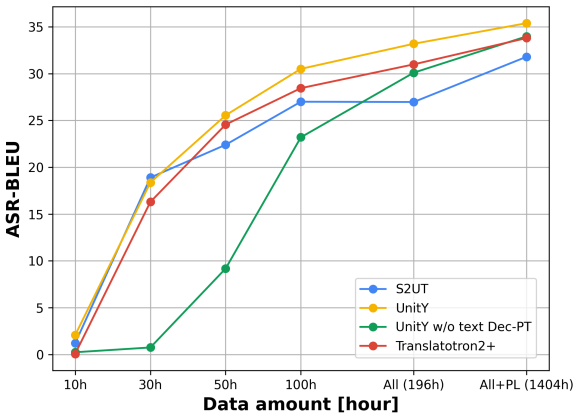

Figure 3: Dev ASR-BLEU at different data scales on the multi-domain Es→En corpus. The amount of training data is measured by source speech. *All* and *PL* represent all supervised data and pseudo-labeled data, respectively.

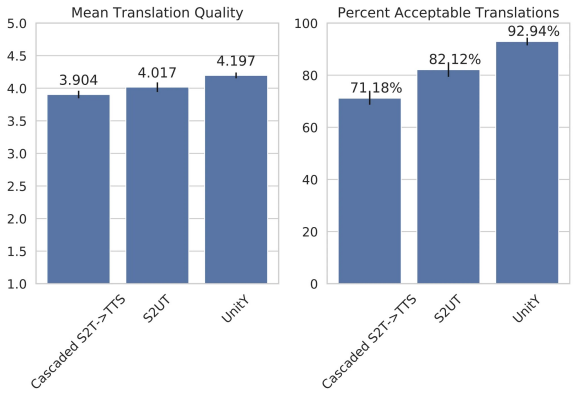

Figure 4: Results of human evaluation on multi-domain Es→En corpus

supervisions (Popuri et al., 2022) rather than parameter initialization. Pre-training a part of UnitY with an independent S2TT model (F8), which was trained on the same corpus, was not helpful either. Surprisingly, the BLEU score from the text decoder in UnitY was better than that of F8. Therefore, training signals from the unit decoder never affect the text decoder.

5.4 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 7 showed that the six-layer text decoder with six-layer unit decoder was the best when initializing the first-pass decoder randomly (G1-G6). Next, we pre-trained the unit decoder with a unit-based mBART while initializing the text decoder randomly (G7). This setting was no better than initializing both decoders randomly (G5, G7). We confirmed the effectiveness of pre-training by training a model with the same amount of parameters from scratch (G8,G9). Lastly, we initialized the text and unit decoders with the corresponding mBART models while initializing the T2U encoder randomly (G10), but it did not improve the performance further. Therefore, it is most effective to pre-train the deep text decoder only and keep the unit decoder shallow.

5.5 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 direct S2ST models at various training data scales in Figure 3. We observed that UnitY consistently outperformed the Translatotron2+ 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 a finding in Section 4.3. However, pre-training the text decoder of UnitY was essential for obtaining decent performances in the low-resource settings ($\leq$ 50 hours).

5.6 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), which emphasizes adequacy rather than fluency, and percent acceptable translations, the percentage of items that received an XSTS score of three or above. 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 evaluate each item and assign it a score from one to five. The median score is taken per item. More details are described in Appendix A.8.

The results are presented in Figure 4.[8] We confirmed that UnitY consistently outperformed the cascaded and S2UT models in both metrics.

6 Related works

6.1 Two-pass sequence generation

Two-pass decoding has many advantages by maintaining the end-to-end optimization capability while inheriting the benefits of a cascading approach. First, we can incorporate an additional search process to find a better output (Xia et al., 2017; Hu et al., 2020). Second, we can rescore the intermediate hypotheses using an external module such as language model (Dalmia et al., 2021). Third, we can inject specific information in the intermediate decoder to bias the output toward the desired domain (Zhao et al., 2019). Fourth, we can provide an intermediate output to users before generating the final output, which would be helpful for streaming applications (Sainath et al., 2019). Lastly, 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).

6.2 Direct speech-to-spectrogram translation

Direct speech-to-spectrogram translation models predict spectrogram in the target language from the source speech in an end-to-end fashion. 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) improves Translatotron significantly by incorporating two-pass decoding. However, we showed that our methods further improved Translatotron2.

6.3 Direct speech-to-unit translation (S2UT)

Direct speech-to-unit translation models predict discrete units rather than spectrogram. (Tjandra et al., 2019) uses vector-quantized variational autoencoder to extract target discrete units. (Lee et al., 2022a) extracts target discrete units by HuBERT. (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

In this work, 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 mBART. Experimental evaluations on the Fisher Es→En, CVSS-C, and multi-domain En↔Es corpora demonstrated that UnitY outperformed a single-pass S2UT model consistently in translation accuracy and inference speed, regardless of the use of pre-training. 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.

Limitation Although R-Drop improved translation quality a lot, it requires an additional computation to generate the second probability distribution.

Since two-pass models require linguistic units as the target for the first-pass decoder, they cannot be used when the target language is unwritten. A promising direction is to find more coarse discrete units whose sequence length is shorter than that of the original discrete units.

Future work To accelerate training and decoding efficiencies, it is promising to take filterbank features as inputs instead of a waveform to reduce the input sequence length. Since greedy decoding has shown to be enough for the second-pass unit decoder in Section 4.4, we consider that using a non-autoregressive decoder (Gu et al., 2018) in the

[8] The models used here are early versions and slightly different from Table 3. We will update the results in the next version.

second pass further reduces the computation time without quality degradation. Similarly to (Jia et al., 2022a; Bapna et al., 2022), joint speech-text self-supervised encoder pre-training followed by joint speech-text fine-tuning would improve the performance further. We believe it is complementary to our first-pass decoder pre-training with mBART. UnitY can also be extended to a multilingual model that generates speech in multiple target languages. We consider the two-pass approach more suitable for that purpose than the single-pass approach because we can obtain structured representations before generating speech, which disentangles multilingual translation and multilingual speech synthesis. Finally, it is also interesting to extend UnitY to the streaming S2ST task.

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.

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, 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*.

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.

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.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *Proceedings of ICLR*.

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.

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.

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 pre-training 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.

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.

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.

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

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*, pages 5998–6008.

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

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, pages 1782–1792.

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.

A Appendix

A.1 Mathematical formulation of R-Drop

We describe a mathematical formulation of R-Drop (Wu et al., 2021) discussed in Section 2.2. Given a set of unique inputs $\mathbf{X}$, the general R-Drop loss $\mathcal{L}_{\text{kl}}$ 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.

A.2 Training objective

In this section, we describe training objectives for baseline S2ST models. In addition to the primary S2ST/S2UT task, we introduce auxiliary S2TT and ASR tasks. We adopted a character-level ASR task for direct S2ST models on the Fisher Es→En corpus while we did not use it on the rest corpora.

Translatotron Given the target spectrogram S, translation Y, and transcription Y_{src}, corresponding to a source speech X, the training objective of Translatotron is formulated as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{s2s}}(S|X) + w_{\text{s2t}}\mathcal{L}_{\text{s2t}}(Y|X) + w_{\text{asr}}\mathcal{L}_{\text{asr}}(Y_{\text{src}}|X), \quad (2)$$

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:

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

Translatotron2+ The training objective of Translatotron2+ is formulated as:

$$\begin{aligned}\mathcal{L}_{\text{total}} = &\sum_{i=1}^{2}\mathcal{L}_{\text{s2s}}(S|X_i, Y) \\ &+ w_{\text{s2t}}(\sum_{i=1}^{2}\mathcal{L}_{\text{s2t}}(Y|X_i) + \beta\mathcal{L}_{\text{kl}}^{\text{s2t}}(X_1, X_2)) \\ &+ w_{\text{asr}}(\sum_{i=1}^{2}\mathcal{L}_{\text{asr}}(Y_{\text{src}}|X_i) + \gamma\mathcal{L}_{\text{kl}}^{\text{asr}}(X_1, X_2)),\end{aligned} \quad (3)$$

where X_i is a duplicated input from X, $\mathcal{L}_{\text{kl}}^{\text{s2t}}$ is the R-Drop loss for the first-pass decoder, $\mathcal{L}_{\text{kl}}^{\text{s2t}}$ is the R-Drop loss for the auxiliary ASR decoder, and β and γ are the corresponding weights for the R-Drop losses, respectively. Unlike Eq (2), 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 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:

$$\begin{aligned}\mathcal{L}_{\text{total}} = &\sum_{i=1}^{2}\mathcal{L}_{\text{s2u}}(U|X_i) + \alpha\mathcal{L}_{\text{kl}}^{\text{s2u}}(X_1, X_2) \\ &+ w_{\text{ctc}}\sum_{i=1}^{2}\mathcal{L}_{\text{ctc}}(Y|D_i^{\text{unit}}) \\ &+ w_{\text{s2t}}(\sum_{i=1}^{2}\mathcal{L}_{\text{s2t}}(Y|X_i) + \beta\mathcal{L}_{\text{kl}}^{\text{s2t}}(X_1, X_2)) \\ &+ w_{\text{asr}}(\sum_{i=1}^{2}\mathcal{L}_{\text{asr}}(Y_{\text{src}}|X_i) + \gamma\mathcal{L}_{\text{kl}}^{\text{asr}}(X_1, X_2)),\end{aligned} \quad (4)$$

where $\mathcal{L}_{\text{s2u}}$ is the primary S2UT loss, $\mathcal{L}_{\text{kl}}^{\text{s2u}}$ is the R-Drop 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 loss, and w_{ctc} is a weight for the CTC loss, respectively. Unlike Eq. (1), there is no dependency between the primary S2UT task and auxiliary S2TT task except for sharing the same encoder.

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
A7	16	2k×4	1.0e-3	10k	0.1	0.2	–	–	–	–	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		–	–	–	–	–	–
A14	16	20k×1	1.5e-3		0.3		0.1	0.1	–	3.0	3.0	–
A15	16	4k×2	1.0e-3		0.1		–	0.1	–	–	3.0	–
A17	4	20k×1	8.6e-4		0.3		8.0	8.0	1.6	1.0	1.0	1.0
A18	16	2k×4	1.0e-3		0.1		–	–	–	–	–	1.0
A19	4	20k×1	6.0e-4		0.3		8.0	8.0	–	3.0	3.0	1.0
A20	16	2k×4	1.0e-3		0.1		–	8.0	–	–	3.0	1.0
B4	32	40k×1	1.0e-3	10k	0.1	0.2	–	0.1	–	–	0.0	–
B5		40k×1	1.0e-3		0.1			0.1	–		0.0	–
B9		40k×1	1.1e-3		0.1			0.1	–		10.0	–
B10		40k×1	1.0e-3		0.1			0.1	–		10.0	–
B11		20k×2	8.6e-4		0.3			8.0	1.6		0.5	0.5
B12		20k×2	7.0e-4		0.3			8.0	1.6		0.5	0.5
B13		20k×2	1.5e-3		0.3			8.0	–		1.5	1.5
B14		20k×2	7.0e-4		0.3			8.0	–		5.0	1.5
C3, 4	32	2k×30	5.0e-4	10k	0.1	0.2	–	8.0	–	–	10.0	–
C5′				1k			–	–	–	–	–	0.0
C6, 7				1k			–	8.0	–	–	10.0	0.0

Table 8: Training hyperparameters

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

$$\mathcal{L}_{\text{total}} = \sum_{i=1}^{2} \mathcal{L}_{\text{s2t}}(Y|X_i) + \beta \mathcal{L}_{\text{kl}}^{\text{s2t}}(X_1, X_2). \quad (5)$$

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

$$\mathcal{L}_{\text{total}} = \sum_{i=1}^{2} \mathcal{L}_{\text{asr}}(Y_{\text{src}}|X_i) + \gamma \mathcal{L}_{\text{kl}}^{\text{asr}}(X_1, X_2). \quad (6)$$

A.3 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, we used a 24-layer Conformer encoder and stacked a one-layer length adaptor (Li et al., 2021) on it. The length adaptor halved the sequence length. Therefore, the frame rate of every encoder output corresponds to 40ms in both cases. In this case, we set $(d_{\text{model}}, d_{\text{ff}}, N_{\text{head}})$ to $(1024, 4096, 16)$.

Translatotron 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, we doubled these three values. We set the pre-net dimension and reduction factor of the spectrogram decoder to 32 and 3, respectively.

Translatotron2+ 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 the Fisher and CVSS-C corpora. 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 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 Translatotron in all the settings.

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

UnitY We used the same first-pass decoder as Translatotron2+ 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 S2UT in all the settings.

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	MLS (918 hours) (Pratap et al., 2020) Common Voice v7 (290 hours) (Ardila et al., 2020)	Librispeech (960 hours) (Panayotov et al., 2015) TEDLIUM3 (452 hours) (Rousseau et al., 2012)
MT Supervised MT1	CCMatrix (Schwenk et al., 2021)	–
Supervised MT2 (Cascaded S2ST)	OpenSubtitle2018 (Lison et al., 2018) UNCorpus (Ziemski et al., 2016) EUBookshop v2 (Skadiņš et al., 2014) Europarl v10 (Koehn, 2005) Wikipedia v1.0(Wołk and Marasek, 2014) TED2020 v1‘(Reimers and Gurevych, 2020)	
T2U/TTS	CSS100 (23.8 hours) (Park and Mulc, 2019)	LJSpeech (24 hours) (Ito and Johnson, 2017)
Unlabeled text Text mBART	CC100 (Conneau et al., 2020)	
Unlabeled speech Wav2Vec2.0	Libri-Light (60k hours) (Kahn et al., 2020)	VoxPopuli Es (16k hours) (Wang et al., 2021a)
Unit mBART	VoxPopuli En (14k hours) VoxPopuli Es (16k hours) Libri-Light (60k hours)	
mHuBERT	VoxPopuli En (14k hours) VoxPopuli Es (16k hours) VoxPopuli Fr	

Table 9: 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 Unit 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 10: Links to pre-trained self-supervised models and TTS models

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

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

A.4 Training details

We list the training hyperparameters in Table 8.

A.5 Data

We list all datasets we used for the experiments in Table 9.

A.5.1 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.

A.6 Pre-trained models

We list all the pre-trained self-supervised models and TTS models used in our experiments in Table 10.

A.7 Additional results

In this section, we present additional experimental results.

ID	Encoder pre-training	Model	Output unit	(ASR-)BLEU (↑)	
				Speech	Text
E1		Translatotron2+	Phoneme	**50.4**	–
E2			Character	50.2	54.0
E3			Subword	49.2	**54.4**
E1′		Translatotron2+	Phoneme	58.1	–
E2′	✓		Character	58.1	61.5
E3′			Subword	**58.4**	**62.0**
E4		UnitY	Phoneme	49.8	–
E5			Character	48.9	53.7
E6			Subword	**50.5**	**55.4**
E4′		UnitY	Phoneme	54.7	–
E5′	✓		Character	55.0	60.9
E6′			Subword	**55.1**	**61.2**

Table 11: Results of output units for the first-pass decoder in two-pass direct S2ST models on the Fisher Es→En dev set. We use 1k unit for the subword vocabulary.

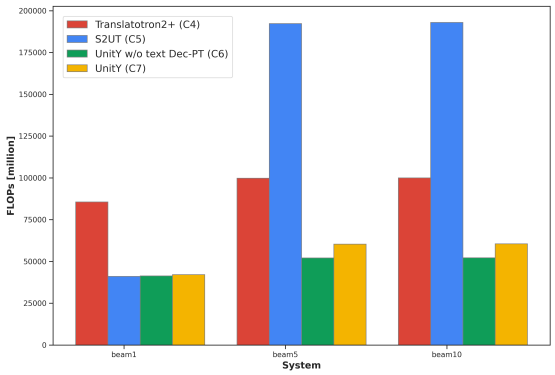

Figure 5: FLOPs of direct S2ST models on multi-domain Es→En corpus. The beam width of two-pass models corresponds to the first-pass decoder.

A.7.1 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 Section 4.4. UnitY achieved 1.65× and 1.16× FLOPs reduction over Translatotron2+ and S2UT models, respectively.

A.7.2 Output unit for first-pass decoder

We show full results of a comparison of output units for the first-pass decoder in two-pass direct S2ST models in Table 11, as discussed in Section 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 Translatotron2+, the best unit differed according to whether we pre-trained the speech encoder or not. However, predicting subwords in the first pass led to the best BLEU score for the text output in all the settings.

A.8 Human evaluation protocol

In this section, we describe metrics used in human evaluation.

Mean translation score We used cross-lingual semantic textual similarity (XSTS) (Licht et al., 2022) 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.

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.