Duplex Diffusion Models Improve Speech-to-Speech Translation

Xianchao Wu
NVIDIA
xianchaow@nvidia.com, wuxianchao@gmail.com

Abstract

Speech-to-speech translation is a typical sequence-to-sequence learning task that naturally has two directions. How to effectively leverage bidirectional supervision signals to produce high-fidelity audio for both directions? Existing approaches either train two separate models or a multitask-learned model with low efficiency and inferior performance. In this paper, we propose a duplex diffusion model that applies diffusion probabilistic models to both sides of a reversible duplex Conformer, so that either end can simultaneously input and output a distinct language's speech. Our model enables reversible speech translation by simply flipping the input and output ends. Experiments show that our model achieves the first success of reversible speech translation with significant improvements of ASR-BLEU scores compared with a list of state-of-the-art baselines.

1 Introduction

Direct speech-to-speech translation (S2ST) (Lee et al., 2021; Inaguma et al., 2022), transforming a source language's speech to the target language's speech, is essential for online international communications and is friendly to numerous languages that do not have their own writing systems or textual vocabularies. S2ST circumvents a cascaded architecture (Lavie et al., 1997; Nakamura et al., 2006; Wahlster, 2000) of combining automatic speech recognition (ASR) of the source speech, textual source-to-target machine translation (MT), and target text-to-speech (TTS) synthesis where multiple types of datasets are required, error propagates, latency is high, and unavailable for thousands of (spoken) languages who do not have a writing system.

For S2ST, speech-to-speech parallel data is required, and it is costly to collect a comparable size dataset with textual counterparts. To alleviate the data scarcity problem, self-supervised pre-training and data augmentation techniques were used by Popuri et al. (2022), and unsupervised and weakly-supervised speech and text data under Translatotron 2 (Jia et al., 2021) were leveraged by Jia et al. (2022a). Techniques such as multi-task learning (Weiss et al., 2017), pseudo labeling (Pino et al., 2020), and knowledge distillation (Inaguma et al., 2021) have also been adapted and achieved promising results.

From S2ST architecture's point of view, Inaguma et al. (2022) describes four categories, (1) Translatotron (Jia et al., 2019) style which includes a speech encoder and a spectrogram decoder, (2) Translatotron2+ (Jia et al., 2021) style which inserts a first-pass text decoder followed by a TTS encoder between the two modules of Translatotron, (3) speech-to-unit translation (S2UT) (Lee et al., 2021) that uses discrete clustered units of the target language speech instead of spectrogram, and (4) UnitY (Inaguma et al., 2022) that inserts a first-pass text decoder followed by a text-to-unit (T2U) encoder between the two modules in S2UT.

In this paper, following the motivations of textual duplex machine translation (Zheng et al., 2021), we leverage S2ST's two directions: effectively utilizing supervision signals from both directions is estimated to both relieve the pain of data scarcity and bring novel architectures of training and inferencing. Existing architectures (e.g., Translatotron1/2, S2UT, and UnitY) either train two separate models or a multitask-learned model with low efficiency and inferior performance. In contrast, we propose a *duplex diffusion model* that applies diffusion probabilistic models to both sides of a *reversible duplex Conformer*, so that either end can simultaneously input and output a distinct language's speech. Our model enables reversible speech translation by simply flipping the input and output ends. Experiments show that our model achieves the first success of reversible speech translation with significant improvements of ASR-BLEU scores compared with a list of strong baselines.

Findings of the Association for Computational Linguistics: ACL 2023, pages 8035–8047
July 9-14, 2023 ©2023 Association for Computational Linguistics

Our contributions are concluded as follows:

- a novel *reversible duplex Conformer* that extends the widely used Conformer (Gulati et al., 2020) architecture from ASR to S2ST, with reversible and symmetrical forward/reverse building blocks;
- a novel *duplex diffusion model* that jointly train one reversible duplex Conformer in diffusion ways to fit two translation directions;
- significantly better or comparable ASR-BLEU scores are achieved by comparing with a list of state-of-the-art baselines including Translatotron, Translatotron2, S2UT, and UnitY.

2 Backgrounds

2.1 REDER

REDER, **RE**versible **D**uplex Transform**ER**, was proposed by Zheng et al. (2021) for reversible textual machine translation through duplex sequence-to-sequence (seq2seq) learning. A neural network with a parameter set θ is *duplex* when it satisfies the following conditions. First, the network has two ends, each end can take one language as its input or output. Second, the network defines a forward mapping function $f_\theta^{\rightarrow} : \mathcal{X} \mapsto \mathcal{Y}$, and a backward (reverse) mapping function $f_\theta^{\leftarrow} : \mathcal{Y} \mapsto \mathcal{X}$, that satisfies two reversibilities: $f_\theta^{\leftarrow} = (f_\theta^{\rightarrow})^{-1}$ and $f_\theta^{\rightarrow} = (f_\theta^{\leftarrow})^{-1}$. Third, the network satisfies the cycle consistencies: $\forall \boldsymbol{x} \in \mathcal{X} : f_\theta^{\leftarrow}(f_\theta^{\rightarrow}(\boldsymbol{x})) = \boldsymbol{x}$ and $\forall \boldsymbol{y} \in \mathcal{Y} : f_\theta^{\rightarrow}(f_\theta^{\leftarrow}(\boldsymbol{y})) = \boldsymbol{y}$.

However, building duplex seq2seq networks is non-trivial and must satisfy the following constraints, **reversibility** and **homogeneity**. First, vanilla encoder-decoder network, such as frequently used Transformer (Vaswani et al., 2017) and its variants, is irreversible. It is not feasible for the output end of the decoder side to take in input signals to exhibit the encoding functionality and vice versa. Second, the natural network architectures of the non-autoregressive encoder and the autoregressive decoder are heterogeneous. Therefore, REDER, leverages reversible Transformer layers (Gomez et al., 2017a) and fully non-autoregressive modeling without explicit encoder and decoder division, is designed to solve these two challenges. As reported in (Zheng et al., 2021), REDER worked in a duplex way that better exploited the bidirectional supervisions for achieving better downstream reversible machine translation tasks' performance.

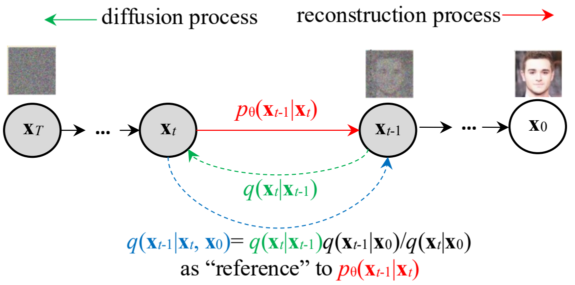
diffusion process
reconstruction process
$\mathbf{x}_T$
$\mathbf{x}_t$
$p_\theta(\mathbf{x}_{t\text{-}1}|\mathbf{x}_t)$
$\mathbf{x}_{t\text{-}1}$
$\mathbf{x}_0$
$q(\mathbf{x}_t|\mathbf{x}_{t\text{-}1})$
$q(\mathbf{x}_{t\text{-}1}|\mathbf{x}_t, \mathbf{x}_0)= q(\mathbf{x}_t|\mathbf{x}_{t\text{-}1})q(\mathbf{x}_{t\text{-}1}|\mathbf{x}_0)/q(\mathbf{x}_t|\mathbf{x}_0)$
as “reference” to $p_\theta(\mathbf{x}_{t\text{-}1}|\mathbf{x}_t)$

Figure 1: The Markov chain of forward diffusion (backward reconstruction) process of generating a sample by step-by-step adding (removing) noise. Image adapted from (Ho et al., 2020).

The architecture of REDER is a stack of L reversible duplex transformer layers where the 1-st to $L/2$-th layers are mirror of the $(L/2+1)$-th to L-th layers to ensure the whole model being symmetric. In particular, each layer contains a multi-head self-attention (MHSA) module and a feed-forward network (FFN) module with a novel reversible design to ensure duplex behavior, where the input and output tensors of such a layer are split into two halves, $\mathbf{H}_{l-1} = [\mathbf{H}_{l-1}^{(1)}; \mathbf{H}_{l-1}^{(2)}]$ and $\mathbf{H}_l = [\mathbf{H}_l^{(1)}; \mathbf{H}_l^{(2)}]$, respectively. Formally, the *regular form* of the l-th layer $\mathcal{F}_l$ performs as follow:

$$[\mathbf{H}_l^{(1)}; \mathbf{H}_l^{(2)}] = \mathcal{F}_l([\mathbf{H}_{l-1}^{(1)}; \mathbf{H}_{l-1}^{(2)}]), \tag{1}$$

$$\mathbf{H}_l^{(1)} = \mathbf{H}_{l-1}^{(1)} + \text{MHSA}(\mathbf{H}_{l-1}^{(2)}), \tag{2}$$

$$\mathbf{H}_l^{(2)} = \mathbf{H}_{l-1}^{(2)} + \text{FFN}(\mathbf{H}_l^{(1)}). \tag{3}$$

The *reverse form* $\mathcal{F}_l^{-1}$ can be computed by subtracting the residuals:

$$[\mathbf{H}_{l-1}^{(1)}; \mathbf{H}_{l-1}^{(2)}] = \mathcal{F}_l^{-1}([\mathbf{H}_l^{(1)}; \mathbf{H}_l^{(2)}]), \tag{4}$$

$$\mathbf{H}_{l-1}^{(2)} = \mathbf{H}_l^{(2)} - \text{FFN}(\mathbf{H}_l^{(1)}), \tag{5}$$

$$\mathbf{H}_{l-1}^{(1)} = \mathbf{H}_l^{(1)} - \text{MHSA}(\mathbf{H}_{l-1}^{(2)}). \tag{6}$$

2.2 DDPM

We briefly introduce the diffusion and reconstruction processes in **D**enoising **D**iffusion **P**robabilistic **M**odels (DDPM). Given a data point $\mathbf{x}_0$ sampled from a real data distribution $q(\mathbf{x})$ ($\mathbf{x}_0 \sim q(\mathbf{x})$), Ho et al. (2020) define a *forward diffusion process* in which small amount of Gaussian noise is added to sample $\mathbf{x}_0$ in T steps to obtain a sequence of noisy samples $\mathbf{x}_0, ..., \mathbf{x}_T$. A predefined (hyper-parameter) variance schedule $\{\beta_t \in (0,1)\}_{t=1}^{T}$ controls the step sizes:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}); \tag{7}$$

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}). \tag{8}$$

When $T \to \infty$, $\mathbf{x}_T$ is equivalent to following an isotropic Gaussian distribution. Note that, there are no trainable parameters used in this forward diffusion process.

Let $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, we can express an arbitrary step t's diffused sample $\mathbf{x}_t$ by the initial data sample $\mathbf{x}_0$:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t. \tag{9}$$

Here, noise $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \mathbf{I})$ shares the same shape with $\mathbf{x}_0$ and $\mathbf{x}_t$.

In order to reconstruct from a Gaussian noise input $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$, we need to learn a model p_θ to approximate the conditional probabilities to run the *reverse diffusion (reconstruction) process*:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t));$$

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t). \tag{10}$$

Note that the reverse conditional probability is tractable by first applying Bayes' rule to three Gaussian distributions and then completing the "quadratic component" in the $\exp(\cdot)$ function:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t\mathbf{I}) \tag{11}$$

$$= q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} \tag{12}$$

$$\propto \exp(-\frac{1}{2\tilde{\beta}_t}(\mathbf{x}_{t-1} - \tilde{\boldsymbol{\mu}}_t)^2). \tag{13}$$

Here, variance $\tilde{\beta}_t$ is a scalar and mean $\tilde{\boldsymbol{\mu}}_t$ depends on $\mathbf{x}_t$ and noise $\boldsymbol{\epsilon}_t$:

$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t; \tag{14}$$

$$\tilde{\boldsymbol{\mu}}_t = \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_t). \tag{15}$$

Intuitively, $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ acts as a *reference* to learn $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$. We can use the variational lower bound (VLB) to optimize the negative log-likelihood:

$$-\log p_\theta(\mathbf{x}_0) \leq -\log p_\theta(\mathbf{x}_0) + D_{\mathrm{KL}}(q(\mathbf{x}_{1:T}|\mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0)). \tag{16}$$

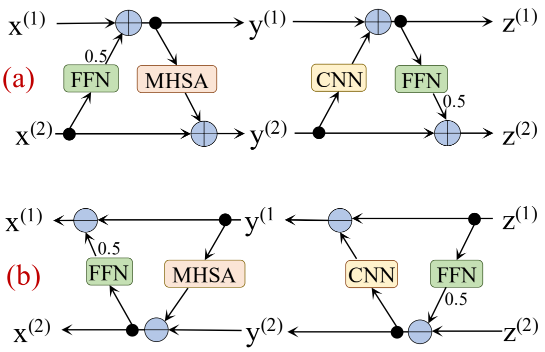

Figure 2: Forward (a) and reverse (b) building blocks for one layer in our reversible duplex Conformer.

Using the definitions of $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$ in Equation 8 and $p_\theta(\mathbf{x}_{0:T})$ in Equation 10, a loss item L_t ($1 \leq t \leq T-1$) is expressed by:

$$\mathcal{L}_t = D_{\mathrm{KL}}(q(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})) \tag{17}$$

$$= \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}_t}\left[\frac{\parallel \tilde{\boldsymbol{\mu}}_t - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t) \parallel^2}{2 \parallel \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t) \parallel_2^2}\right].$$

We further reparameterize the Gaussian noise term instead to predict $\boldsymbol{\epsilon}_t$ from time step t's input $\mathbf{x}_t$ and use a simplified objective that ignores the weighting term:

$$\mathcal{L}_t^{\mathrm{simple}} = \mathbb{E}_{t\sim[1,T], \mathbf{x}_0, \boldsymbol{\epsilon}_t}\left[\parallel \boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \parallel^2\right] \tag{18}$$

3 Reversible Duplex Conformer

In this paper, we extend the widely used Conformer (Gulati et al., 2020) architecture for encoding the speech signals into dense and compact representations of both ends. Conformer has achieved impressive results in supervised ASR by leveraging transformer's capturing of content-based *global* interactions and convolutional neural network's exploiting of *local* features. In Conformer, two macaron-like FFN layers with half-step residual connections sandwich the MHSA and convolution (CNN) modules followed by a post layer normalization. Besides supervised ASR, Conformer has also been successfully used in self-supervised Wav2Vec (Schneider et al., 2019; Baevski et al., 2020) pretraining for downstream application tasks' finetuning.

3.1 Forward and Reverse Building Blocks

Following (Gomez et al., 2017b; Zheng et al., 2021), we split the l-th layer's (left-end) input tensor into two parts, $\mathbf{H}_{l-1} = [\mathbf{x}^{(1)}; \mathbf{x}^{(2)}]$. The (right-end) output tensor is split in the same way, $\mathbf{H}_l = [\mathbf{z}^{(1)}; \mathbf{z}^{(2)}]$. Thus, the forward target of this REDER-style Conformer layer is $[\mathbf{x}^{(1)}; \mathbf{x}^{(2)}] \mapsto [\mathbf{z}^{(1)}; \mathbf{z}^{(2)}]$.

We introduce two intermediate tensors, $\mathbf{y}^{(1)}$ and $\mathbf{y}^{(2)}$, for intuitive understanding and mathematical convenient. Both Conformer's four sub-modules (two FFNs, one MHSA and one CNN) and four residual connections are kept in our reversible duplex Conformer.

Figure 2 depicts the forward (a) and reverse (b) building blocks for one layer in our proposed reversible duplex Conformer. In Figure 2, the reverse block is a mirror of the forward block with symmetrical network connections and subtract residual connections. The forward block can be formally expressed as follows:

$$\mathbf{y}^{(1)} = \mathbf{x}^{(1)} + 0.5 \times \mathrm{FFN}(\mathbf{x}^{(2)}); \tag{19}$$

$$\mathbf{y}^{(2)} = \mathbf{x}^{(2)} + \mathrm{MHSA}(\mathbf{y}^{(1)}); \tag{20}$$

$$\mathbf{z}^{(1)} = \mathbf{y}^{(1)} + \mathrm{CNN}(\mathbf{y}^{(2)}); \tag{21}$$

$$\mathbf{z}^{(2)} = \mathbf{y}^{(2)} + 0.5 \times \mathrm{FFN}(\mathbf{z}^{(1)}). \tag{22}$$

Symmetrically, the reverse block is expressed by:

$$\mathbf{y}^{(2)} = \mathbf{z}^{(2)} - 0.5 \times \mathrm{FFN}(\mathbf{z}^{(1)}); \tag{23}$$

$$\mathbf{y}^{(1)} = \mathbf{z}^{(1)} - \mathrm{CNN}(\mathbf{y}^{(2)}); \tag{24}$$

$$\mathbf{x}^{(2)} = \mathbf{y}^{(2)} - \mathrm{MHSA}(\mathbf{y}^{(1)}); \tag{25}$$

$$\mathbf{x}^{(1)} = \mathbf{y}^{(1)} - 0.5 \times \mathrm{FFN}(\mathbf{x}^{(2)}). \tag{26}$$

We employ Layer Normalization (LN) (Ba et al., 2016) at the beginning of each module, i.e., PreLN (Xiong et al., 2020). The FFN module processes the input tensor $\mathbf{x}$ by six components:

$$\mathrm{FFN}(\mathbf{x}) = p_2 \circ \boldsymbol{W}_2 \circ p_1 \circ \mathrm{SiLU} \circ \boldsymbol{W}_1 \circ \mathrm{LN}(\mathbf{x}).$$

Here, $\circ$ means a layer takes $\circ$'s right-hand-side network's output (e.g., $\mathrm{LN}(\mathbf{x})$) as the input of $\circ$'s left-hand-side network (e.g., $\boldsymbol{W}_1$ to perform $\boldsymbol{W}_1(\mathrm{LN}(\mathbf{x}))$). $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are two linear layers that preforms $h \mapsto 4h$ and $4h \mapsto h$ linear projections, respectively. Two dropout layers p_1 and p_2 are used. The Sigmoid Linear Unit (SiLU) (Elfwing et al., 2017) activation function is inserted between the two linear layers. The MHSA module contains three components:

$$\mathrm{MHSA} = p \circ \mathrm{Attention} \circ \mathrm{LN}(\mathbf{x}).$$

We use multi-head attention with relative positional embedding (Shaw et al., 2018) for the "Attention" component. Note that, the attention module is extendable to cross-attention cases where a source sequence's encoded representation acts as memory (i.e., key and value) to the target sequence. Finally, the CNN module utilizes two types of convolutions, pointwise (PW) and 1D depthwise (DW), to capture local-range dependencies of the input speech. The idea of employing attention for global context modeling and convolution for local context modeling is also inspired by the long-short range attention mechanism used in the lite transformer (Wu et al., 2020). Formally,

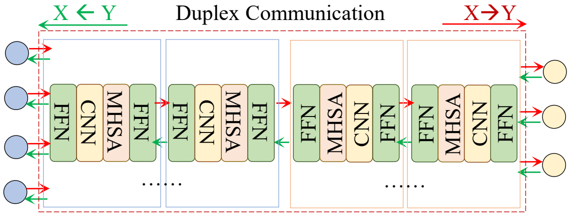

Figure 3: Symmetric architecture using reversible duplex Conformer building blocks for duplex speech-to-speech translation.

$$\begin{aligned}\mathrm{CNN}(\mathbf{x}) = {} & p \circ \mathrm{PW}_2 \circ \mathrm{Swish} \circ \mathrm{BN} \\ & \circ \mathrm{DW} \circ \mathrm{Glu} \circ \mathrm{PW}_1 \circ \mathrm{LN}(\mathbf{x}).\end{aligned}$$

Here, BN stands for batch normalization. Two types of activation functions, Glu (Dauphin et al., 2016) and Swish (Ramachandran et al., 2017), are inserted between convolution networks.

3.2 Symmetric Network Architecture

As depicted in Figure 3, the forward and reverse building blocks are arranged symmetrically in the whole architecture to achieve homogeneous computations. Specifically, in the L building blocks, the 1-st to $L/2$-th layers are set to be reverse blocks whereas the $(L/2+1)$-th to L-th layers be the regular forward form:

$$\begin{aligned}f_\theta^{\rightarrow}(\mathbf{x}) = {} & \mathcal{F}_L \circ \cdots \circ \mathcal{F}_{L/2+1} \\ & \circ \mathcal{F}_{L/2}^{-1} \circ \cdots \circ \mathcal{F}_1^{-1}(\mathbf{x}); \\ f_\theta^{\leftarrow}(\mathbf{z}) = {} & \mathcal{F}_1 \circ \cdots \circ \mathcal{F}_{L/2} \\ & \circ \mathcal{F}_{L/2+1}^{-1} \circ \cdots \circ \mathcal{F}_L^{-1}(\mathbf{z}).\end{aligned}$$

This design makes our reversible duplex Conformer to be homogeneous: the forward computational operation chain reads as a palindrome string $< fcmf \cdots fcmf | fmcf \cdots fmcf >$ and so does the reverse chain, where f, m, c denotes FFN, MHSA and CNN, respectively.

Algorithm 1: Duplex Diffusion Model (DDM) Training Algorithm - One Step

```
Given: $\mathbf{x}$, $\mathbf{y}$, $\mathcal{E}_\mathbf{x}$, $\mathcal{E}_\mathbf{y}$
$\mathbf{x}_0 = \mathcal{E}_\mathbf{x}(\mathbf{x})$ ▷ encode by pretrained
  wav2vec models
$\mathbf{y}_0 = \mathcal{E}_\mathbf{y}(\mathbf{y})$ ▷ encode by pretrained
  wav2vec models
$t \sim \text{Uniform}(1, ..., T)$
$\epsilon_\mathbf{x} \sim \mathcal{N}_\mathbf{x}(\mathbf{0}, \mathbf{I}), \epsilon_\mathbf{y} \sim \mathcal{N}_\mathbf{y}(\mathbf{0}, \mathbf{I})$
$\mathbf{x}_t = \sqrt{\bar{\alpha}_{t,\mathbf{x}}}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_{t,\mathbf{x}}}\epsilon_\mathbf{x}$
$\mathbf{y}_t = \sqrt{\bar{\alpha}_{t,\mathbf{y}}}\mathbf{y}_0 + \sqrt{1 - \bar{\alpha}_{t,\mathbf{y}}}\epsilon_\mathbf{y}$
$\epsilon^\theta_\mathbf{x} = \overleftarrow{\mathcal{M}_\theta}(\mathbf{x}_t, t, \mathbf{y}_0)$ ▷ reverse, given $\mathbf{y}_0$
$\epsilon^\theta_\mathbf{y} = \overrightarrow{\mathcal{M}_\theta}(\mathbf{y}_t, t, \mathbf{x}_0)$ ▷ forward, given $\mathbf{x}_0$
$\mathcal{L}_{\text{DDM}} = \lambda_1 \parallel \epsilon_\mathbf{x} - \epsilon^\theta_\mathbf{x} \parallel^2 + \lambda_2 \parallel \epsilon_\mathbf{y} - \epsilon^\theta_\mathbf{y} \parallel^2$
```

There are several selections of input types of the source and target ends in Figure 3. Popuri et al. (2022) explores self-supervised pretrained models such as (1) wav2vec2 (Baevski et al., 2020) to encode the source speech and (2) Unit mBART (Liu et al., 2020a) to encode the target discrete units (Lee et al., 2021), and then translate source speech into target clustered units through fine-tuning. The generated discrete unit sequence is then sent to an independently trained "text"-to-speech (TTS) model to obtain the final waves.

In this paper, we follow the usage of discrete units that are generated by first using pretrained HuBERT (Hsu et al., 2021) to encode the target speech and then perform k-means clustering. Then, we use the DiffWave (Kong et al., 2020b) vocoder to generate the final waves.

4 Duplex Diffusion Model

Cycle consistency has been utilized in textual neural machine translation (Zheng et al., 2021) and image-to-image translation (Su et al., 2022). In this paper, we propose a duplex diffusion model that alternatively optimizes both directions by two diffusion processes.

The training algorithm is described in Algorithm 1. Generally, we borrow DDPM (Ho et al., 2020)'s architecture and extend it to a duplex scenario where sequences of two ends are diffused alternatively during training. At the beginning, the source sequence $\mathbf{x}$ and target sequence $\mathbf{y}$ are encoded into dense representations by pretrained wav2vec models $\mathcal{E}_\mathbf{x}, \mathcal{E}_\mathbf{y}$ through self-supervised learning on monolingual datasets, respectively. Then, time t and two normal Gaussian noise signals $\epsilon_\mathbf{x}, \epsilon_\mathbf{y}$ are sampled. Note that the lengths of the source and target sequences are diverse. We pre-define two variance schedules $\{\beta_{t,\mathbf{x}} \in (0, 1)\}_{t=1}^T$ and $\{\beta_{t,\mathbf{y}} \in (0, 1)\}_{t=1}^T$, for the source and target languages, respectively. Thus, we have $\alpha_{t,\mathbf{x}} = 1 - \beta_{t,\mathbf{x}}$, $\bar{\alpha}_{t,\mathbf{x}} = \prod_{i=1}^t \alpha_{i,\mathbf{x}}$, $\alpha_{t,\mathbf{y}} = 1 - \beta_{t,\mathbf{y}}$ and $\bar{\alpha}_{t,\mathbf{y}} = \prod_{i=1}^t \alpha_{i,\mathbf{x}}$, as used in Algorithm 1.

The variance schedules, initial sequence representations and normal Gaussian noises work together to give us diffused representations, $\mathbf{x}_t$ and $\mathbf{y}_t$, respectively. They are then sent to the reversible duplex Conformer architecture $\mathcal{M}_\theta$ (Figure 3) to predict the noises.

Originally in Figure 3, we are intended to produce $\mathbf{x}_0$ from $\mathbf{y}_0$ in the reverse process of $\overleftarrow{\mathcal{M}_\theta}$. Now, we have two additional inputs, t and $\mathbf{x}_t$. The output also changes from predicting $\mathbf{x}_0$ to estimating $\epsilon^\theta_\mathbf{x}$ which shares the same shape with $\mathbf{x}_0$.

We thus have two ways to organize the network $\overleftarrow{\mathcal{M}_\theta}$: (1) reuse Figure 3's architecture and predicting $\epsilon^\theta_\mathbf{x}$ from $\mathbf{y}_0$ by taking $\mathbf{x}_t$ as the "memory" which acts as key and value in the cross-attention network in Conformer, or (2) follow traditional stable diffusion models (Rombach et al., 2021) and predict $\epsilon^\theta_\mathbf{x}$ from $\mathbf{x}_t$ by taking $\mathbf{y}_0$ as the conditional "memory". That is, in the MHSA function, we set query $\mathbf{x}_t$ to be and key/value to be $\mathbf{y}_0$, MHSA$(q = \mathbf{x}_t, k = \mathbf{y}_0, v = \mathbf{y}_0)$, so that the identical lengths of $q = \mathbf{x}_t$ and $\epsilon^\theta_\mathbf{x}$ are ensured. Note that, in the second choice used in our experiments, we are not limited to use a reversible duplex Conformer, i.e., any transformer architecture with cross-attention are applicable. These two options still hold during inferencing from given $\mathbf{y}_0$, T, and $\mathbf{x}_T$ to iteratively reconstruct $\mathbf{x}_0$.

Since the lengths of the source and target sequences are diverse, we follow textual duplex translation (Zheng et al., 2021) and double the source end's length by a upsampling convolutional network.

We only describe the reverse process $\overleftarrow{\mathcal{M}_\theta}$ and the forward process $\overrightarrow{\mathcal{M}_\theta}$ shares the similar strategies. To achieve a full cycle consistency, predicting the target Gaussian noise from source sequence by taking target noisy sequence as the "conditional memory" is more appropriate in current scenario setting so that both translation directions are achieved in one duplex diffusion model.

After the reverse and forward processes, we can compute the MSE losses of between the two pairs of reference and predicted noises. They are interpolated together by hyper-parameter weights λ_1 (=0.5) and λ_2 (=0.5) to the final loss $\mathcal{L}_{\text{DDM}}$ to be optimized.

5 Training

Our reversible duplex Conformer is largely inspired by REDER (Zheng et al., 2021). The novel parts are that (1) we select and reconstruct convolution-enhanced Conformer (Gulati et al., 2020) to synthetically capture global information by attentions and local context by convolutions, and (2) we extend from textual duplex machine translation to (dense) duplex speech-to-speech translation. When training our reversible duplex Conformer, we borrow and adapt the losses that are used in REDER to fit our scenario.

In REDER, three types of losses were used. The first loss is to model the variable-length of source and target sequences by a latent alignment approach, i.e., the Connectionist Temporal Classification (CTC) (Graves et al., 2006). Starting from the conditional independence assumption, CTC is capable of efficiently (by dynamic programming) finding all valid (yet monotonic) alignments $\mathbf{a}$ which derives from the target $\mathbf{y}$ by allowing consecutive repetitions and inserting blank tokens. The CTC loss is defined by:

$$\mathcal{L}_{\text{CTC}} = -\log p_{\text{CTC}}(\mathbf{y}|\mathbf{x};\theta) = -\log \sum_{\mathbf{a}} p_\theta(\mathbf{a}|\mathbf{x}).$$

We adapt this loss for speech translation when the target are clustered unit sequences. We use the MSE loss instead when the target is a sequence of mel-spectrogram. Also, to ensure the source sequence is always longer than the target sequence, we upsample the source sequences by convolutional layers before sending them to the reversible duplex Conformer.

The second loss measures the layer-wise forward-backward agreement (fba, measured by cosine similarity) of between the forward l-th layer's representation $\overrightarrow{\mathbf{H}}_l = \mathcal{F}_l(\overrightarrow{\mathbf{H}}_{l-1})$ and the reverse representation $\overleftarrow{\mathbf{H}}_l = \mathcal{F}_l(\overleftarrow{\mathbf{H}}_{l+1})$. Thus,

$$\mathcal{L}_{\text{fba}}(\mathbf{y}|\mathbf{x};\theta) = \frac{1}{L}\sum_{l=1}^{L}\left\{1 - \cos(\overrightarrow{\mathbf{H}}_l, \mathsf{sg}(\overleftarrow{\mathbf{H}}_l))\right\},$$

where sg denotes the stop-gradient operation.

The third loss explicitly describe the cycle consistency of a pair of seq2seq tasks, i.e., we minimize the distance between the original $\mathbf{x}$ and its reconstruction $f_\theta^{\leftarrow}(f_\theta^{\rightarrow}(\mathbf{x}))$ by,

$$\mathcal{L}_{\text{cc}}(\mathbf{x};\theta) = \texttt{distance}(\mathbf{x}, f_\theta^{\leftarrow}(f_\theta^{\rightarrow}(\mathbf{x}))).$$

For speech translation, the source sequence can be expressed by mel-spectrogram or clustered units so that MSE loss or CTC loss can be applied to them, respectively. Finally, these three types of losses are doubled to two directions and interpolated together for the final loss. That is, when predicting discrete units, the final loss function is:

$$\begin{aligned}\mathcal{L}_{\text{unit}} = {} & w_1 * \mathcal{L}_{\text{CTC}}(\mathbf{y}|\mathbf{x}) + w_2 * \mathcal{L}_{\text{CTC}}(\mathbf{x}|\mathbf{y}) \\ & + w_3 * \mathcal{L}_{\text{fba}}(\mathbf{y}|\mathbf{x}) + w_4 * \mathcal{L}_{\text{fba}}(\mathbf{x}|\mathbf{y}) \\ & + w_5 * \mathcal{L}_{\text{cc}}(\mathbf{y}) + w_6 * \mathcal{L}_{\text{cc}}(\mathbf{x}).\end{aligned}$$

When predicting mel-spectrograms, the final loss function is:

$$\begin{aligned}\mathcal{L}_{\text{mel}} = {} & w_1 * \mathcal{L}_{\text{MSE}}(\mathbf{y}|\mathbf{x}) + w_2 * \mathcal{L}_{\text{MSE}}(\mathbf{x}|\mathbf{y}) \\ & + w_3 * \mathcal{L}_{\text{fba}}(\mathbf{y}|\mathbf{x}) + w_4 * \mathcal{L}_{\text{fba}}(\mathbf{x}|\mathbf{y}) \\ & + w_5 * \mathcal{L}_{\text{cc}}(\mathbf{y}) + w_6 * \mathcal{L}_{\text{cc}}(\mathbf{x}).\end{aligned}$$

We reuse the default hyper-parameter values described in REDER (Zheng et al., 2021) for setting weights w_1 to w_6.

In our experiments, we first train the reversible duplex Conformer architecture by a predefined K_1 (=200,000) iterations and then apply the duplex diffusion training algorithm as shown in Algorithm 1. After another predefined K_2 (=200,000) iterations, we fix the diffusion processes and focus on updating the reversible duplex Conformer only so that traditional search algorithms such as beam search can be used for seeking target hypotheses.

6 Experimental Setups

6.1 Data

To compare with state-of-the-art baselines' reported results, we align with UnitY (Inaguma et al., 2022) and use three S2ST datasets: (1) Fisher Es→En (Post et al., 2013) with 170-hour Spanish (Es) conversational telephone speech with textual transcriptions in Es and En. The English speech is synthesized by a single-female-speaker TTS model. (2) CVSS-C (Jia et al., 2022b), a public multilingual S2ST corpus from CoVoST2 (Wang et al., 2020). Again, a single-female-speaker TTS model is employed to synthesize the target speech. (3) Multi-domain En↔Es corpora (Popuri et al., 2022). We follow (Inaguma et al., 2022) to collect 1983-hour source speech for En→Es and 1404-hour source speech for Es→En.

6.2 Pre-training and Pre-processing

We use the pretrained wav2vec2.0 (Baevski et al., 2020) with a 24-layer Conformer (Gulati et al., 2020) self-trained on the Libri-Light dataset (Kahn et al., 2019), HuBERT (Hsu et al., 2021), mHuBERT (Popuri et al., 2022), and mBART (Liu et al., 2020b) given in Table 9 of (Inaguma et al., 2022).

For acoustic feature extraction, discrete unit extraction (100 clusters) and text normalization (e.g., for evaluation score computing), we follow (Popuri et al., 2022; Inaguma et al., 2022).

6.3 Vocoder

Instead of using the HiFi-GAN vocoder (Kong et al., 2020a; Polyak et al., 2021) which converts mel-spectrograms or discrete units to waveform sequences for TTS and direct speech-to-spectrogram/unit models, we borrow a comparable diffusion based vocoder, DiffWave (Kong et al., 2020b), for reconstructing waveforms from spectrogram or unit sequences.

6.4 Training and Decoding Configurations

We implement our models based on the Fairseq toolkit[1] (Ott et al., 2019). All our models are optimized with a mixed precision training for footprint saving. Our reversible duplex Conformer uses the settings of Conformer-Large with 135.1M parameters (Gulati et al., 2020). The two diffusion variance schedules used in our duplex diffusion model follow stable diffusion (Rombach et al., 2021). We use a NVIDIA DGX-A100*8 workstation to perform the training with a total of 2,500 GPU hours.

During inferencing, we set the beam size to be 10 which aligns with most of the baselines for fair comparison. Other configurations not mentioned here follow their default settings in their open-source repositories.

6.5 Evaluation

We use a pre-trained ASR model to transcribe the target speech into texts and then calculate 4-gram BLEU scores (Papineni et al., 2002), denoted as ASR-BLEU. The target languages' ASR models are fine-tuned from pretrained wav2vec2.0 (Baevski et al., 2020) models with the CTC objective (Graves et al., 2006) when we taking discrete unit sequences as the prediction target. The same criterion has been used in (Inaguma et al., 2022).

Model	dev	dev2	test
ASR-MT-TTS	42.1	43.5	43.9
S2TT-TTS, C	47.8	48.9	48.3
S2TT-TTS, C-w2v2	51.0	52.2	52.1
S2Sp-Tn, C	43.9	44.4	43.8
S2Sp-Tn, C-w2v2	45.5	47.6	46.3
S2Sp-Tn2+, C	50.4	51.1	50.8
S2Sp-Tn2+, C-w2v2	58.4	59.5	58.6
S2Sp-RDC (Ours)	46.1	47.3	47.0
S2Sp-RDC, w2v2	50.7	51.5	51.0
S2Sp-DDM (Ours)	52.4	55.1	54.8
S2Sp-DDM, w2v2	**58.9**	**59.8**	**59.1**
S2U, C	46.2	47.6	47.4
S2U, C-w2v2	53.4	53.9	53.7
UnitY, C	50.5	51.6	51.4
UnitY, C-w2v2	55.1	56.5	55.9
S2U-RDC (Ours)	48.1	49.0	48.5
S2U-RDC, w2v2	50.8	52.1	51.8
S2U-DDM (Ours)	52.2	53.6	53.1
S2U-DDM, w2v2	**56.3**	**58.0**	**57.4**

Table 1: ASR-BLEU (%) on the Fisher Es→En corpus. S2Sp = speech-to-spectrogram, S2U = speech-to-unit, Tn = Translatotron, C = Conformer, RDC = reversible duplex Conformer (Section 3), DDM = duplex diffusion model (Section 4), and w2v2 = wav2vec2.0.

7 Experimental Results

7.1 Fisher Es→ En

In Table 1, we compare the ASR-BLEU scores of our systems (RDC and DDM) with three cascaded systems, four speech-to-spectrogram baselines which are variants of Translatotron (Jia et al., 2019, 2021), and four speech-to-unit baselines which are variants of (Lee et al., 2021) and UnitY (Inaguma et al., 2022). Baseline results are originally listed in (Inaguma et al., 2022).

We use RDC to denote our reversible duplex Conformer architecture that are trained in a similar way with textual REDER (Zheng et al., 2021). Our DDM further "boost" the quality of pretrained RDC models by bidirectional diffusion processes and can be recognized as an integration of the diffusion framework with RDC. Of the three categories, S2Sp and S2U achieved significantly better ($p < 0.01$) ASR-BLEU scores than the three traditional cascaded systems. In the S2Sp paradigm, our "S2Sp-DDM, w2v2" model achieves comparable results with the best baseline "S2Sp-Tn2+, C-w2v2". In the S2U paradigm, our model "S2U-DDM, w2v2" achieves significantly better ($p < 0.05$) re-

[1] https://github.com/facebookresearch/fairseq

Model	Avg.	High	Mid	Low
S2TT-TTS, ASR	12.7	30.7	18.3	4.4
S2TT-TTS, w2v-b	13.2	21.3	16.1	9.2
S2Sp-Tn2, w2v-b	17.9	32.5	22.9	10.9
S2Sp-Tn2+, w2v-b	20.8	31.6	25.4	**15.4**
S2Sp-RDC (Ours)	18.2	32.4	22.1	10.2
S2Sp-DDM (Ours)	**22.1**	**33.5**	**27.4**	15.2
S2U, w2v-b	20.8	31.6	25.4	15.4
UnitY, w2v-b	24.5	34.6	28.9	19.3
S2U-RDC (Ours)	22.1	32.5	27.1	17.8
S2U-DDM (Ours)	**24.9**	**35.2**	**30.2**	**20.4**

Table 2: ASR-BLEU (%) on the CVSS-C corpus. ASR = ASR pretraining, w2v-b = wav2vec BERT.

Model En→Es	E-ST	MuST-C
ASR-MT-TTS	36.8	30.8
S2TT-TTS	36.4	33.4
S2Sp-Tn2+	35.6	33.5
S2Sp-Tn2+, mB	36.9	**34.3**
S2Sp-RDC (Ours)	35.1	32.7
S2Sp-DDM (Ours)	**37.2**	**34.3**
UnitY	35.1	33.7
UnitY, mB	35.3	34.1
S2U-RDC (Ours)	34.7	32.6
S2U-DDM (Ours)	**35.8**	**34.5**
Model Es→En	**CoVoST-2**	**E-ST**
ASR-MT-TTS	32.9	34.2
S2TT-TTS	37.2	34.0
S2Sp-Tn2+	37.0	23.4
S2Sp-Tn2+, mB	**37.2**	23.7
S2Sp-RDC (Ours)	34.5	30.6
S2Sp-DDM (Ours)	37.1	**32.8**
UnitY	35.4	30.8
UnitY, mB	36.4	33.1
S2U-RDC (Ours)	35.1	31.2
S2U-DDM (Ours)	**36.7**	**34.0**

Table 3: ASR-BLEU (%) on the multi-domain En↔Es tasks. mB = mBART, E-ST=Europarl-ST.

sults than the best baseline "UnitY, C-w2v2", with 1.2%, 1.5% and 1.5% absolute ASR-BLEU points. These reflects that our proposed duplex seq2seq learning can be boosted by the bidirectional diffusion processes to better capture the translation distributions of among the source and target sides. In addition, wav2vec2.0 acts as an essential component for all the model variants.

Table 1 also lists four variants of our models for ablation study. When we only use S2U-RDC, it performs better than the S2U+Conformer baseline. However, this advantage disappears when w2v2 is further employed to these two variants. S2U-RDC also performs relatively worse than UnitY which employs two pass decoding of basing on texts and units whereas our S2U-RDC uses units only. These reflect that, (1) additional textual information brings better results than duplex training, (2) diffusion processes can partly "hedge" the benefits from two-pass decoding used in UnitY and enhance the performance of duplex translations.

7.2 CVSS-C

The CVSS-C corpus's ASR-BLEU scores of six baselines from three categories and our models are listed in Table 2. We observe almost the same tendencies with the result comparisons in the Fisher task (Table 1). The best baseline is still the two-pass UnitY model enhanced by a pretrained wav2vec-BERT model. Our S2U-DDM model improves UnitY by 0.4% ASR-BLEU points on average, comparable yet not significant.

7.3 Multi-domain En↔Es

The bidirectional multi-domain En↔Es results are listed in Table 3. We again compare with six state-of-the-art baselines in three categories. On both directions, our model variants meet the best performances on the two test sets. We notice that the baselines perform less stable under the Europarl-ST corpus with ASR-BLEU ranges from 23.4% to 34.2%. In the S2Sp scenario, both our RDC and DDM variants perform significantly better ($p < 0.01$) than the two baselines. Our S2U-DDM variant performs significantly better ($p < 0.05$) than UnitY and is comparable to the best cascaded system. Note that we only require one run training for bidirectional translations.

7.4 Inference Speed

We use a NVIDIA DGX-A100*8 workstation to perform the inferencing comparison without additional engineering optimization. We randomly select 500 utterances from the multi-domain Es→En dev set. For end-to-end S2ST inferencing, our final RDC with one-pass decoding achieved 1.72× decoding speed-ups over the best-performance UnitY (Inaguma et al., 2022) baseline which requires a two-pass text+unit decoding.

7.5 Human Evaluation

Finally, we performed an audio-only human evaluation to evaluate the translation quality and acceptances of the best baseline UnitY and our DDM. For direct comparison, we use the mTEDx test with 989 samples. We obtained a mean translation quality score of 4.202(/5.0) which is comparable to UnitY's 4.197 and an acceptable ratio of 92.89% which is also comparable to UnitY's 92.94%.

8 Conclusion

Aiming at effectively leveraging bidirectional supervision signals of speech-to-speech translation (S2ST), we have proposed two models for duplex S2ST, a reversible duplex Conformer and a duplex diffusion model. We compare with cascaded S2ST models, single/multi-pass speech-to-spectrogram/unit models and report significantly better or comparable ASR-BLEU and human-evaluated scores, with fewer training time and faster inference speed.

9 Limitations

Our duplex diffusion model and reversible duplex Conformer architecture do not explicitly take reordering as an essential challenge. However, depicts the language pairs described in the experiments, there are languages such as English and Japanese which shares subject-verb-object (SVO) and subject-object-verb (SOV) word orders. These limit the scalability of our proposed methods and external pre-ordering (Zhao et al., 2018; Wu et al., 2011) or post-ordering (Goto et al., 2013) techniques on clustered units of speech should be taken into consideration in the future work.

Large-scale unlabeled speech data is required to train self-supervised wav2vec2.0 or HuBERT models. However, this is frequently not easy to collect. Moreover, it is even more difficult to collect paired speech-to-speech data and existing TTS models for generating speech from text are still under developing. These are pre-conditions of applying our proposed approaches.

Finally, we still need to train pair-by-pair for S2ST which is quadratic to the number of languages. Our approach is less effective than textual multilingual machine translation architecture in which linear number of translation models are required and achieved comparable results than pairwise baselines. Multilingual S2ST requires novel training architectures and inferencing algorithms.

10 Ethics Statement

Our target is to build direct speech-to-speech translation systems with a duplex idea of training both directions in one run. We try our best to reuse existing pretrained wav2vec2.0, HuBERT, mHuBERT and mBART models to save energy consuming. In one run, we require much less GPU-hours for obtaining S2ST models for both direction usages. However, compared with textual duplex MT systems, pre-processing of speech signals still requires much higher costing of GPU-hours and as listed in our limitation section (Section 9), smarter ways of multilingual S2ST architectures are preferred in the future to reduce the cost of energy from current quadratic to linear number of models.

Generally, S2ST circumvents traditional cascaded systems which concatenate ASR, MT and TTS with high latency and high requirements of datasets. There are 3,000 around languages in the world who do not have their own writing systems or textual vocabularies. Through our duplex S2ST models, we hope to be friendly to these languages so that more and more languages can be covered.

References

Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *ArXiv*, abs/1607.06450.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460. Curran Associates, Inc.

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2016. Language modeling with gated convolutional networks. *CoRR*, abs/1612.08083.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. 2017. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *CoRR*, abs/1702.03118.

Aidan N. Gomez, Mengye Ren, Raquel Urtasun, and Roger B. Grosse. 2017a. The reversible residual network: Backpropagation without storing activations. *CoRR*, abs/1707.04585.

Aidan N Gomez, Mengye Ren, Raquel Urtasun, and Roger B Grosse. 2017b. The reversible residual network: Backpropagation without storing activations. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Isao Goto, Masao Utiyama, and Eiichiro Sumita. 2013. Post-ordering by parsing with itg for japanese-english statistical machine translation. *ACM Transactions on Asian Language Information Processing*, 12(4).

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 369–376, New York, NY, USA. Association for Computing Machinery.

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 *Proc. Interspeech 2020*, pages 5036–5040.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *CoRR*, abs/2006.11239.

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. *CoRR*, abs/2106.07447.

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

Hirofumi Inaguma, Sravya Popuri, Ilia Kulikov, Peng-Jen Chen, Changhan Wang, Yu-An Chung, Yun Tang, Ann Lee, Shinji Watanabe, and Juan Pino. 2022. Unity: Two-pass direct speech-to-speech translation with discrete units. *ArXiv*, abs/2212.08055.

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 *Interspeech*.

Ye Jia, Michelle Tadmor Ramanovich, Tal Remez, and Roi Pomerantz. 2021. Translatotron 2: High-quality direct speech-to-speech translation with voice preservation. In *International Conference on Machine Learning*.

Ye Jia, Michelle Tadmor Ramanovich, Quan Wang, and Heiga Zen. 2022b. CVSS corpus and massively multilingual speech-to-speech translation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6691–6703, Marseille, France. European Language Resources Association.

Ye Jia, Ron J. Weiss, Fadi Biadsy, Wolfgang Macherey, Melvin Johnson, Z. Chen, and Yonghui Wu. 2019. Direct speech-to-speech translation with a sequence-to-sequence model. In *Interspeech*.

Jacob Kahn, Morgane Rivière, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, Pierre-Emmanuel Mazaré, Julien Karadayi, Vitaliy Liptchinsky, Ronan Collobert, Christian Fuegen, Tatiana Likhomanenko, Gabriel Synnaeve, Armand Joulin, Abdelrahman Mohamed, and Emmanuel Dupoux. 2019. Libri-light: A benchmark for ASR with limited or no supervision. *CoRR*, abs/1912.07875.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020a. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *ArXiv*, abs/2010.05646.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. 2020b. Diffwave: A versatile diffusion model for audio synthesis. *ArXiv*, abs/2009.09761.

A. Lavie, A. Waibel, L. Levin, M. Finke, D. Gates, M. Gavalda, T. Zeppenfeld, and Puming Zhan. 1997. Janus-iii: speech-to-speech translation in multiple languages. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 99–102 vol.1.

Ann Lee, Peng-Jen Chen, Changhan Wang, Jiatao Gu, Xutai Ma, Adam Polyak, Yossi Adi, Qing He, Yun Tang, Juan Miguel Pino, and Wei-Ning Hsu. 2021. Direct speech-to-speech translation with discrete units. *CoRR*, abs/2107.05604.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020a. Multilingual denoising pre-training for neural machine translation. *CoRR*, abs/2001.08210.

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

S. Nakamura, K. Markov, H. Nakaiwa, G. Kikui, H. Kawai, T. Jitsuhiro, J.-S. Zhang, H. Yamamoto, E. Sumita, and S. Yamamoto. 2006. The atr multilingual speech-to-speech translation system. *IEEE 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. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Juan Miguel Pino, Qiantong Xu, Xutai Ma, Mohammad Javad Dousti, and Yun Tang. 2020. Self-training for end-to-end speech translation. In *Interspeech*.

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. *CoRR*, abs/2104.00355.

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. *arXiv preprint arXiv:2204.02967*.

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 the 10th International Workshop on Spoken Language Translation: Papers*, Heidelberg, Germany.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. 2017. Searching for activation functions. *CoRR*, abs/1710.05941.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2021. High-resolution image synthesis with latent diffusion models. *CoRR*, abs/2112.10752.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised Pre-Training for Speech Recognition. In *Proc. Interspeech 2019*, pages 3465–3469.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*.

Xu Su, Jiaming Song, Chenlin Meng, and Stefano Ermon. 2022. Dual diffusion implicit bridges for image-to-image translation. *ArXiv*, abs/2203.08382.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wolfgang Wahlster, editor. 2000. *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer, Berlin.

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

Ron J. Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Z. Chen. 2017. Sequence-to-sequence models can directly translate foreign speech. In *Interspeech*.

Xianchao Wu, Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2011. Extracting preordering rules from predicate-argument structures. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 29–37, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Zhanghao Wu, Zhijian Liu, Ji Lin, Yujun Lin, and Song Han. 2020. Lite transformer with long-short range attention.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. 2020. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Yang Zhao, Jiajun Zhang, and Chengqing Zong. 2018. Exploiting pre-ordering for neural machine translation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Zaixiang Zheng, Hao Zhou, Shujian Huang, Jiajun Chen, Jingjing Xu, and Lei Li. 2021. Duplex sequence-to-sequence learning for reversible machine translation. In *Advances in Neural Information Processing Systems*, volume 34, pages 21070–21084. Curran Associates, Inc.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
9

☑ A2. Did you discuss any potential risks of your work?
10

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

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

B ☒ Did you use or create scientific artifacts?

Left blank.

□ B1. Did you cite the creators of artifacts you used?
No response.

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

□ 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)?
No response.

□ 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?
No response.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

□ 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.
No response.

C ☑ Did you run computational experiments?

7

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
7

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?
6

☑ 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?
7

☑ 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.)?
7

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?
7

□ 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.?
Not applicable. Left blank.

□ 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)?
Not applicable. Left blank.

☑ 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?
6

☑ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
6

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.