# DIFFNORM: Self-Supervised Normalization for Non-autoregressive Speech-to-speech Translation

**Weiting Tan  Jingyu Zhang  Lingfeng Shen  Daniel Khashabi  Philipp Koehn**
Department of Computer Science
Johns Hopkins University
{wtan12, jzhan237, lshen30, danielk, phi}@jhu.edu

## Abstract

Non-autoregressive Transformers (NATs) are recently applied in direct speech-to-speech translation systems, which convert speech across different languages without intermediate text data. Although NATs generate high-quality outputs and offer faster inference than autoregressive models, they tend to produce incoherent and repetitive results due to complex data distribution (e.g., acoustic and linguistic variations in speech). In this work, we introduce DIFFNORM, a diffusion-based normalization strategy that simplifies data distributions for training NAT models. After training with a self-supervised noise estimation objective, DIFFNORM constructs normalized target data by denoising synthetically corrupted speech features. Additionally, we propose to regularize NATs with classifier-free guidance, improving model robustness and translation quality by randomly dropping out source information during training. Our strategies result in a notable improvement of about $+7$ ASR-BLEU for English-Spanish (En-Es) and $+2$ ASR-BLEU for English-French (En-Fr) translations on the CVSS benchmark, while attaining over $14\times$ speedup for En-Es and $5\times$ speedup for En-Fr translations compared to autoregressive baselines.[1]

## 1 Introduction

Speech-to-speech translation (S2ST) systems are essential to bridge communication gaps and have wide application potential. We focus on non-autoregressive modeling for direct *speech-to-speech* translation, converting source speech to the target without intermediate text data. Such direct S2ST systems [25, 23, 22, 32, 31, 21, 20] can preserve non-linguistic information, avoid error propagation from cascaded systems [29, 34] (e.g., a combination of speech recognition and machine translation systems), and achieve faster inference speed.

Non-autoregressive Transformers (NAT) [21, 20] has played a central role in current S2ST work [31, 32, 21]. NAT translates source waveforms into target **speech units** via parallel decoding, achieving performance comparable to or better than autoregressive models while greatly reducing inference time. This process is often referred to as *speech-to-unit* (S2UT) translation [31]. The predicted speech units are then converted to target waveforms via a unit vocoder in the *unit-to-speech* synthesis stage [32, 38]. However, NATs suffer from incoherent and repetitive generations, referred to as the **multi-modality problem** [12]. This issue stems from NAT's assumption of conditional independence during parallel decoding, worsened by the complex and multi-modal nature of training data distribution.

In this work, we propose DIFFNORM, a **self-supervised** speech normalization strategy that alleviates the multi-modality problem of NAT models by simplifying the target distribution. Instead of distilling training data from an autoregressive model [10] or utilizing perturbed speech to train a normalizer [32, 21], we rely on the denoising objective of Denoising Diffusion Probabilistic Models [15, DDPM] to normalize target speech units. DIFFNORM inject synthetic noise to speech features and use diffusion

[1] Code available at: https://github.com/steventan0110/DiffNorm.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).


https://doi.org/10.52202/079017-1048

model to gradually recover the feature, obtaining a simplified and more consistent data distribution that obscures non-crucial details. As the denoising objective is learned in a self-supervised manner over latent speech representations, it eliminates the necessity for transcription data [32] or manually crafted perturbation functions [21]. As illustrated in Fig. 1, applying DIFFNORM to the target data obtains normalized speech units that lead to better NAT training for speech-to-unit translation.

Besides using DIFFNORM as a data-centric strategy to mitigate the multi-modality problem, we also propose a regularization strategy to enhance the NAT model's robustness and generalizability when facing complex data distribution (e.g., linguistic diversity and acoustic variation [21, 20]). Inspired by classifier-free guidance [17, CG], during training, we occasionally drop out source information and replace it with a "null" representation, compelling the models to generate coherent units without conditioning on the source data. During iterative parallel decoding of the NAT model, we obtain higher-quality translation by mixing conditional and unconditional generation. Ultimately, combining DIFFNORM and CG results in our top-performing state-of-the-art system, achieving approximately $+7$ and $+2$ ASR-BLEU increment for En-Es and En-Fr translation compared to previous non-autoregressive systems on the CVSS [24] benchmark.

In conclusion, we alleviate the multi-modality problem by proposing (1) diffusion-based normalization and (2) regularization with classifier-free guidance. To the best of our knowledge, we are the first to adapt diffusion models and classifier-free guidance into speech-to-speech translation and NAT modeling. Our methods obtain notable improvement compared to previous systems and maintain fast inference speed inherent in non-autoregressive modeling, achieving speedups of $14\times$ for En-Es and $5\times$ for En-Fr compared to autoregressive baselines.

## 2 Problem formulation and overview

We aim to develop a direct (textless) speech-to-speech translation system that transduces a source speech $\boldsymbol{x} = (x_1, \cdots, x_N)$ into target speech. We follow Lee et al. [31] to reduce speech-to-speech translation into two sub-tasks: speech-to-unit translation and unit-to-speech synthesis. In this work, we focus on speech-to-unit translation and follow prior work [31, 32, 21] to use the same unit-to-speech component for a fair comparison. To generate speech units for the target language, we first extract speech feature $\boldsymbol{h} = (h_1, \cdots, h_M) \in \mathbb{R}^{M \times H}$, where each feature has $H$ dimensions. Subsequently, a K-means clustering model is trained on extracted features and used to generate speech units $\boldsymbol{y} = (y_1, \cdots, y_M) \in \mathbb{R}^{M \times 1}$. Once the source speech $\boldsymbol{x}$ and target speech unit $\boldsymbol{y}$ are prepared, we train sequence-to-sequence models to translate from source speech into target units. In practice, we follow Lee et al. [32] to use a multilingual-HuBERT (mHuBERT) model to extract $H = 768$ dimensional speech features $\boldsymbol{h}$. Then we use a 1000-cluster K-means model to predict speech units given the encoded features.[2] For speech-to-unit translation, we follow Huang et al. [21] to adopt Conditional Masked Language Modeling [10, CMLM], a kind of non-autoregressive transformer (NAT).

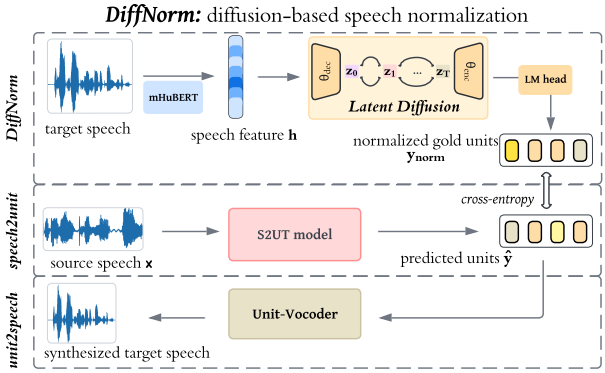


Figure 1: Overview of our proposed system. We first normalize the target speech units with the denoising process from the latent diffusion model. Then speech-to-unit (S2UT) model is trained to predict normalized units, which are converted into waveform from an off-the-shelf unit-vocoder.

To mitigate NAT models' multi-modality problem, we propose DIFFNORM (section §3), which denoises synthetically corrupted speech features to construct normalized speech units $\boldsymbol{y}_{\text{norm}}$. As shown in Fig. 1, such normalized units are then used to train the S2UT (CMLM) model, which generates better-translated units for the unit vocoder to synthesize target speech.

[2] We take the mHuBERT and K-means models off-the-shelf to produce units consistent with prior work. See github.com/facebookresearch/fairseq/tree/main/examples/speech_to_speech

https://doi.org/10.52202/079017-1048

Additionally, we propose to incorporate classifier-free guidance [17, CG], a widely adopted strategy for diffusion-based image generation, as a regularization method to improve the non-autoregressive speech-to-unit system (section §4). As shown in Fig. 3, by forcing NAT models to unmask target speech units without conditioning on source information, the model becomes more robust, generating coherent speech units that result in higher-quality translations. During inference, we follow classifier-free guidance to mix conditional and unconditional generation for the NAT model's iterative decoding, further enhancing its translation quality.

## 3 DIFFNORM: denoising diffusion models for speech normalization

Previous normalization strategy [32, 21] on speech-to-unit translation relies on connectionist temporal classification (CTC) fine-tuning [4] using the HuBERT [19] model. For example, Lee et al. [32] rely on single-speaker data from VoxPopuli [58] to produce normalized (speaker-invariant) speech units for HuBERT-CTC fine-tuning. On the other hand, Huang et al. [21] generate acoustic-agnostic units by perturbing rhythm, pitch, and energy information with (manually) pre-defined transformation functions. We propose to normalize speech units using self-supervised denoising objectives from Denoising Diffusion Probabilistic Models [15, DDPM]. We believe DDPM is more suitable for speech normalization because: (1) It only requires monolingual speech data, without the need of transcriptions to create a text-to-unit model as in [32]. (2) It learns to denoise features in a high-dimensional space rather than relying on hand-designed perturbation as in [21]. DIFFNORM consists of a variational auto-encoder (VAE) and a diffusion model. Since vanilla VAE [27] and DDPM [15] are applied to image generation, we modify them to support token generation and incorporate multitasking objectives. The VAE model reduces the dimension of the speech feature, mapping feature $\boldsymbol{h}$ into lower dimensional latent $\boldsymbol{z}$. The diffusion model (visualized in Fig. 2) is then trained to denoise on the latent representation space, aiming to recover the original latent $\boldsymbol{z}_0$ from the standard Gaussian noise $\boldsymbol{z}_T$.[3] In the subsequent sections, we begin by outlining the architecture of our VAE model (§3.1). Then, we delve into how the diffusion model is trained using the latent variables encoded by the VAE (§3.2).

### 3.1 Variational auto-encoders for latent speech representation

Since speech features encoded by mHuBERT have a high dimension ($H = 768$), we compress the feature into lower-dimension latents for the diffusion model. The VAE model consists of an encoder, a decoder, and a language modeling head. Following our problem formulation (§2), we first prepare a sequence of target speech features $\boldsymbol{h} = (h_1, \cdots, h_M) \in \mathbb{R}^{M \times H}$ and their corresponding speech units $\boldsymbol{y} = (y_1, \cdots, y_M)$. Our VAE model's encoder will map the feature into lower dimension latent $\boldsymbol{z} = f(\boldsymbol{h}; \theta_{\text{enc}}) \in \mathbb{R}^{M \times Z}$ where $Z < H$ is the pre-defined latent dimension. Then VAE model's decoder reconstructs the speech feature $\hat{\boldsymbol{h}} = f(\boldsymbol{z}; \theta_{\text{dec}})$ and we apply the language modeling head to convert the reconstructed feature into a distribution over speech units' vocabulary $\boldsymbol{v} = f(\hat{\boldsymbol{h}}; \theta_{\text{lm}}) \in \mathbb{R}^{M \times V}$. Following prior work [21, 32], we cluster speech features into $V = 1000$ units. The training objective is a weighted combination of reconstruction loss ($\mathcal{L}_{\text{recon}}$), negative log-likelihood (NLL) loss ($\mathcal{L}_{\text{nll}}$), and a Kullback–Leibler (KL) divergence term resulted from the Gaussian constraint [27] to regularize the representation space:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{recon}} + \lambda_2 \mathcal{L}_{\text{nll}} + \lambda_3 \mathcal{L}_{\text{kl}}, \tag{1}$$

where the reconstruction loss is defined as $\mathcal{L}_{\text{recon}}(\hat{\boldsymbol{h}}, \boldsymbol{h}) = ||\boldsymbol{h} - \hat{\boldsymbol{h}}||^2$, and the NLL loss is computed as the cross-entropy between ground-truth units $\boldsymbol{y}$ and the predicted vocabulary distribution $\boldsymbol{v}$: $\mathcal{L}_{\text{nll}}(\boldsymbol{v}, \boldsymbol{y}) = -\sum_{i=1}^{M} \boldsymbol{y}_i \log \boldsymbol{v}_i$, where $\boldsymbol{y}_i$ is the one-hot version of $y_i$. Lastly, $\mathcal{L}_{\text{kl}}$ can be solved analytically when we assume Gaussian distribution for both prior and posterior approximation of latent $\boldsymbol{z}$ (more details in [27] and Appendix B). Though the Gaussian constraint has a small weight $\lambda_3$, we show in the ablation study (§6.2) that regularizing latent distribution is critical for the diffusion model. For details on the architecture of our VAE model, we direct readers to Appendix B.

### 3.2 Diffusion model for denoising latent speech representation

**Training** Once the VAE model is trained, we encode speech feature as the first step feature $\boldsymbol{z}_0 = f(\boldsymbol{h}; \theta_{\text{enc}})$ for the diffusion model. Diffusion models consists of a (1) forward process that

[3] The subscript $z_0$ is added to indicate the time step for diffusion process and $\boldsymbol{z}_0$ and $\boldsymbol{z}$ are equivalent.


https://doi.org/10.52202/079017-1048

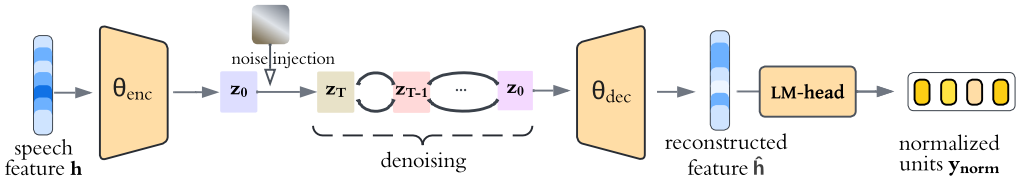


Figure 2: Visualization of our latent diffusion model's denoising process for speech normalization. The clean latent $\boldsymbol{z}_0$ is synthetically noised (into $\boldsymbol{z}_T$) and the reverse diffusion process gradually denoise it to generate normalized speech units.

gradually transforms $\boldsymbol{z}_0$ into a standard Gaussian distribution, and a (2) reverse process that denoise and recovers the original feature $\boldsymbol{z}_0$. Following DDPM [15], with a pre-defined noise scheduler (let $\beta_t \in (0,1)$ be the scaling of noise variance, define $\alpha_t = 1 - \beta_t$ and denote $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_t$ as the noise level for time $t$), the forward and reverse process can be written as:

$$q(\boldsymbol{z}_t|\boldsymbol{z}_0) = \mathcal{N}(\boldsymbol{z}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{z}_0, (1-\bar{\alpha}_t)\boldsymbol{\epsilon}) \quad p_\theta(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t) = \mathcal{N}(\boldsymbol{z}_{t-1}; \boldsymbol{\mu}_\theta(\boldsymbol{z}_t, t), \boldsymbol{\sigma}^2\mathbf{I}) \tag{2}$$

where mean $\boldsymbol{\mu}_\theta(\boldsymbol{z}_t, t)$ and variance $\boldsymbol{\sigma}^2$ are parameterized by trainable models, typically based on U-Net [44] or Transformer [57]. We follow DDPM to train models using the re-weighted noise estimation objective:

$$\mathcal{L}_{\text{noise}}(\theta, t) = \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{\epsilon}, t}||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{z}_0, t)||^2 \tag{3}$$

where the network learns to predict the injected Gaussian noise $\boldsymbol{\epsilon}$ from the forward process. Besides noise estimation, we also train the model with auxiliary reconstruction and NLL loss using VAE model's decoder and language modeling head. Our training procedure is summarized in Alg. 1.

**Algorithm 1** Latent Diffusion Model Training

1: **Input**: speech feature $\boldsymbol{h}$, speech units $\boldsymbol{y}$.
2: **Pre-compute**: $\beta_t, \alpha_t, \bar{\alpha}_t, t \in [1, T]$
3: **while** not converged :
4: $\quad \boldsymbol{z}_0 = f(\boldsymbol{h}; \theta_{\text{enc}})$
5: $\quad t \sim \mathcal{U}[1, T], \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$
6: $\quad \boldsymbol{z}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{z}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$
7: $\quad \hat{\boldsymbol{\epsilon}} = \boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t)$
8: $\quad \mathcal{L}_{\text{noise}} = ||\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}||^2$
9: $\quad \hat{\boldsymbol{z}}_0 = (\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\epsilon}})/\sqrt{\bar{\alpha}_t}$
10: $\quad \hat{\boldsymbol{h}} = f(\hat{\boldsymbol{z}}_0; \theta_{\text{dec}}), \mathcal{L}_{\text{recon}} = ||\boldsymbol{h} - \hat{\boldsymbol{h}}||^2$
11: $\quad \boldsymbol{v} = f(\hat{\boldsymbol{h}}; \theta_{\text{lm}}), \mathcal{L}_{\text{nll}} = \text{NLL}(\boldsymbol{v}, \boldsymbol{y})$
12: $\quad$ Take gradient descent step with loss
13: $\quad\quad \mathcal{L} = \gamma_1\mathcal{L}_{\text{noise}} + \gamma_2\mathcal{L}_{\text{recon}} + \gamma_3\mathcal{L}_{\text{nll}}$

**Algorithm 2** Normalized Units Construction

1: **Input**: speech feature $\boldsymbol{h}$, start time $T$
2: **Pre-compute**: $\beta_t, \alpha_t, \bar{\alpha}_t, t \in [1, T]$
3: $\boldsymbol{z}_0 = f(\boldsymbol{h}; \theta_{\text{enc}})$
4: $\boldsymbol{z}_T = \sqrt{\bar{\alpha}_T}\boldsymbol{z}_0 + \sqrt{1-\bar{\alpha}_T}\boldsymbol{\epsilon}$
5: $t = T$
6: **while** $t > 0$ :
7: $\quad \hat{\boldsymbol{\epsilon}} = \boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t)$
8: $\quad \hat{\boldsymbol{z}}_0 = (\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\epsilon}})/\sqrt{\bar{\alpha}_t}$
9: $\quad \boldsymbol{z}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{\boldsymbol{z}}_0 + \sqrt{1-\bar{\alpha}_{t-1}} \cdot \hat{\boldsymbol{\epsilon}}$
10: $\quad t = t - 1$
11: $\hat{\boldsymbol{h}} = f(\boldsymbol{z}_0; \theta_{\text{dec}})$
12: $\boldsymbol{y}_{\text{norm}} = \text{argmax} f(\hat{\boldsymbol{h}}; \theta_{\text{lm}})$
13: **return** denoised units $\boldsymbol{y}_{\text{norm}}$

As shown in Alg. 1, we randomly sample the current timestep $t \in [1, T]$ and compute corresponding scheduling parameters $\beta_t, \alpha_t, \bar{\alpha}_t$[4]. The training process involves the regular DDPM objective that injects Gaussian noise for the model $\boldsymbol{\epsilon}_\theta$ to perform noise estimation (Alg. 1, line 6-8). Additionally, we generate the pseudo latent $\hat{\boldsymbol{z}}_0$ by reversing the noise injection process (line 9), which is then decoded by the VAE into speech features and units for reconstruction loss (line 10) and NLL loss (line 11). Finally, the objective is a weighted sum of noise estimation loss, reconstruction loss, and NLL loss:

$$\mathcal{L} = \gamma_1\mathcal{L}_{\text{noise}} + \gamma_2\mathcal{L}_{\text{recon}} + \gamma_3\mathcal{L}_{\text{nll}} \tag{4}$$

In our analysis (§6.2), we show that adding NLL and reconstruction loss is indeed helpful in improving the diffusion model's reconstruction quality. Lastly, to parameterize our diffusion model for noise estimation, we modify Diffusion Transformer [37, DiT] to suit our task. For details of our architecture and hyper-parameters, we refer readers to Appendix B.

**Inference** For speech normalization, we choose a start time $T \in [0, 200]$ that decides the amount of noise to inject for the diffusion model to recover. Given the start time $T$, we follow Denoising Diffusion Implicit Models [48, DDIM] sampler to reverse noised latent $\boldsymbol{z}_T$ back to $\boldsymbol{z}_0$. Then, our VAE model converts $\boldsymbol{z}_0$ back to speech units with its decoder and language modeling head. Our inference procedure is summarized in Alg. 2 and visualized in Fig. 2.

[4] We follow the cosine schedule proposed by Nichol and Dhariwal [35] with a maximum timestep of $T = 200$

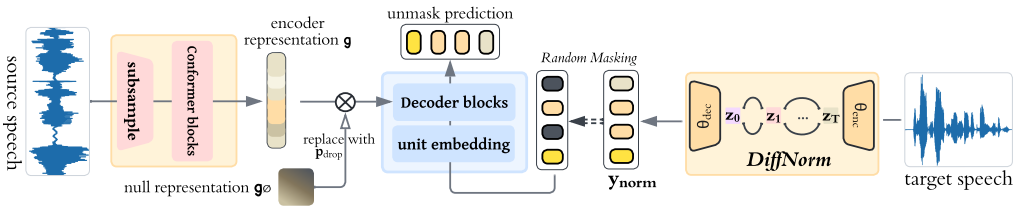


Figure 3: Visualization of CMLM for speech-to-unit translation where the model is trained with the unmasking objective to recover $\boldsymbol{y}_{\text{norm}}$. When classifier-free guidance is used, with probability $p_{\text{drop}}$, we replace the encoded source speech $\boldsymbol{g}$ by a "null" representation $\boldsymbol{g}_{\emptyset}$.

Note that since our diffusion model is used to normalize speech units as a data preprocessing strategy, inference speed is not a concern. Therefore we use step size $\Delta t = 1$ for DDIM sampling throughout our experiments. However, the inference speed could be easily improved by setting a larger step size.

## 4 Classifier-free guidance for non-autoregressive transformer

In §3, we proposed DIFFNORM to normalize speech units and obtained speech-unit pairs $(\boldsymbol{x}, \boldsymbol{y}_{\text{norm}})$ that benefit NAT training. In this section, we propose to adapt classifier-free guidance [17] to regularize NAT models (visualized in Fig. 3), further enhancing NAT-based S2UT model's translation quality.

**Training** We largely adhere to standard CMLM training [21] but introduce a small dropout probability $p_{\text{drop}} = 0.15$ for the encoder representations. Specifically, we parameterize the encoder and decoder of the CMLM as $\phi_{\text{enc}}$ and $\phi_{\text{dec}}$, respectively. The encoder processes the source speech into a representation $\boldsymbol{g} = f(\boldsymbol{x}; \phi_{\text{enc}})$. The decoder then predicts the vocabulary distribution $\boldsymbol{v} = f(\hat{\boldsymbol{y}}|\boldsymbol{g}; \phi_{\text{dec}})$ from $\boldsymbol{g}$ and the randomly masked target speech units $\hat{\boldsymbol{y}}$. The total amount of masked token is uniformed sampled: $n \sim \mathcal{U}[1, M]$; subsequently, $n$ of $M$ tokens from the target units $\boldsymbol{y}$ are randomly masked to form noisy target units $\hat{\boldsymbol{y}}$. CMLM also trains a length predictor that estimates the output length given input sequences and we refer readers to [21] for more details.

Using the predicted distribution $\boldsymbol{v}$, we compute the NLL loss against the ground truth units $\boldsymbol{y}$ at the masked positions to train the model. If dropout is applied, the decoder receives a "null" representation $\boldsymbol{g}_{\emptyset}$—a randomly-initialized learnable vector—forcing it to rely solely on the target information to unmask units. For more details, we refer readers to Appendix C and previous paper [10, 21].

**Inference** CMLM generates tokens through iterative unmasking. Given the input sequence $\boldsymbol{x}$, CMLM first predicts the length of output sequence $M$ and initialize a sequence of $M$ masked tokens $\hat{\boldsymbol{y}}_0 = ([\text{mask}]_1, \cdots, [\text{mask}]_M)$. Given the total number of iterations $T$[5] and current iteration $t \in [1, T]$, CMLM decodes all tokens $y_i, i \in [1, M]$ in parallel using their probabilities:

$$y_i^t = \underset{w}{\text{argmax}} \mathbb{P}(y_i = w|\boldsymbol{x}, \hat{\boldsymbol{y}}_{t-1}; \phi_{\text{dec}}) \quad p_i^t = \log \mathbb{P}(y_i^t|\boldsymbol{x}, \hat{\boldsymbol{y}}_{t-1}; \phi_{\text{dec}}) \tag{5}$$

From the generated tokens $\hat{\boldsymbol{y}}_t = (y_1^t, \cdots, y_M^t)$, we replace $k = M \cdot \frac{T-t}{T}$ tokens with the lowest probabilities $p_i^t$ with mask tokens. This re-masked sequence $\hat{\boldsymbol{y}}_t$ is then inputted into the CMLM for further decoding iterations. This process continues until the final step $t = T$, at which point no tokens are re-masked. With our proposed regularization from classifier-free guidance, we compute two vocabulary distribution in each iteration step $t$:

$$p_{\text{orig}}^t = \mathbb{P}(\cdot|\boldsymbol{g}, \hat{\boldsymbol{y}}_{t-1}; \phi_{\text{dec}}) \quad p_{\text{uncond}}^t = \mathbb{P}(\cdot|\boldsymbol{g}_{\emptyset}, \hat{\boldsymbol{y}}_{t-1}; \phi_{\text{dec}}) \tag{6}$$

where $p_{\text{orig}}^t$ is the same as vanilla CMLM and $p_{\text{uncond}}^t$ is the unconditional probability obtained with the "null" representation. Then we select and re-mask tokens using the adjusted probability

$$p^t = \omega(p_{\text{orig}}^t - p_{\text{uncond}}^t) + p_{\text{orig}}^t \tag{7}$$

where the hyper-parameter $\omega$ is used to control the degree to push away from the unconditional distribution. In the special case of $\omega = 0$, only conditional distribution is used during iterative decoding, resulting in the same generation process as standard CMLM. Through our intensive analysis

[5] $T$ was used in §3 to denote total timesteps for diffusion models. We abuse the notation and re-use it here to represent the number of decoding iterations for CMLM.


https://doi.org/10.52202/079017-1048

(Appendix F), we determine that a relatively small $\omega$ (e.g., $\omega = 0.5$) gives the best result, especially when the number of decoding iterations is large (i.e., $T > 10$). Notably, even when $\omega = 0$ — which corresponds to the traditional CMLM decoding method — CMLMs regularized with classifier-free guidance during training can consistently outperform their standard counterparts. This observation further validates the effectiveness of our proposed regularization.

# 5 Experiments

## 5.1 Experimental setup

**Dataset** We perform experiments using the established CVSS-C datasets [24], which are created from CoVoST2 by employing advanced *text-to-speech* models to synthesize translation texts into speech [59]. CVSS-C comprises aligned speech in multiple languages along with their respective transcriptions. Our methods are evaluated on two language pairs: English-Spanish (En-Es) and English-French (En-Fr), with detailed data statistics provided in Table 1. As our focus is on direct speech-to-speech translation, transcriptions are solely utilized for evaluation purposes. Utilizing the speech data from CVSS, we preprocess the target speech to generate speech units using the mHuBERT and K-means model, as described in our problem formulation (§2).

| *Split* | *En-Es* | | *En-Fr* | |
|---|---|---|---|---|
| | Size | Length | Size | Length |
| Train | 79,012 | 256 | 207,364 | 228 |
| Valid | 13,212 | 296 | 14,759 | 264 |
| Test | 13,216 | 308 | 14,759 | 283 |

Table 1: Data statistics for CVSS benchmarks. Length is the average number of speech units of the target speech.

**Evaluation** To evaluate the performance of various speech-to-speech translation systems, we adopt the standard methodology established in previous studies [32, 31, 21] to compute the ASR-BLEU score. Firstly, our speech-to-unit translation system generates speech units based on the input speech, which are then transformed into speech waveforms using a unit-vocoder. The unit-vocoder is built upon the HifiGAN architecture [28] with customized objectives, detailed in Appendix D. Once we have the waveforms, we employ an ASR model to transcribe them and calculate the BLEU score against the reference transcriptions. This ASR model is fine-tuned based on the Wav2Vec2.0 [5] model with the CTC objective. We direct readers to the original paper [7] for more details. Both the unit-vocoder and ASR models are sourced from an off-the-shelf repository, ensuring consistency in evaluation methodology with previous studies [32, 21].[6]

**Normalized dataset construction** We adhere to the procedure outlined in Alg. 2 to generate normalized speech units. By manipulating start time $T$, we control the level of noise in the latent $\boldsymbol{z}_T = \sqrt{\bar{\alpha}_t}\boldsymbol{z}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$ for the diffusion model, balancing between the reconstruction quality and normalization effect. We explore different levels of noise injection and choose $T = 100$ for En-Es and $T = 120$ for En-Fr (further details in §6.1). Hence, we adopt these settings to construct our normalized dataset, CVSS-NORM.

**Compared systems** In this section, we provide brief descriptions of evaluated speech-to-unit models. For autoregressive models, we evaluate the Transformer model trained following Lee et al. [31]. We also assess the Conformer model trained similar to the Transformer, but with its encoder replaced by a Conformer-Encoder [14]. Lastly, the Norm Transformer shares the same architecture as the Transformer, but it is trained on normalized speech units that are speaker-invariant. The normalized dataset is constructed following the strategy proposed by Lee et al. [32].

For non-autoregressive systems, we train the Conditional Masked Language Model (CMLM) following Huang et al. [21], using a Conformer-based encoder and a Transformer decoder. The CMLM + Bilateral Perturbation (BiP) system retains the same architecture as CMLM but is trained on normalized speech units constructed with BiP [21].

For our improved systems, we train CMLM + DiffNorm, which shares the same architecture as CMLM but is trained on CVSS-NORM, the normalized dataset obtained through DiffNorm. Additionally, we train the CMLM + CG model that incorporates classifier-free guidance introduced in §4. Finally, the CMLM + DiffNorm + CG system uses the architecture of CMLM + CG and is trained on our normalized dataset CVSS-NORM.

[6] Repository available at:github.com/facebookresearch/fairseq/examples/speech_to_speech/asr_bleu.

### 5.2 Results

Table 2 summarizes the (ASR-BLEU) performances of various S2UT systems. Note that, for non-autoregressive methods, Table 2 shows their results obtained with 15 decoding iterations. For more details on the inference speedup achieved through non-autoregressive modeling, we plot Fig. 4 to provide details on the "quality vs. latency" trade-off. Observations from Table 2 are:

**DIFFNORM greatly enhances translation quality compared to systems using original speech units** (model 6 vs. 4; 8 vs. 7). For example, CMLM with DIFFNORM achieves about +7 BLEU score on En-Es compared to CMLM trained on original units. **DIFFNORM also outperforms previous normalization strategies** (model 6 v.s. 2, 5), validating the superior quality of normalized speech units obtained through our methods. Besides normalization, classifier-free guidance effectively improves speech-to-unit quality as a regularization strategy, leading to better translation quality (model 7 v.s. 4; 8 v.s. 6). Finally, **combining both classifier-free guidance and DIFFNORM (model 8) results in the best overall system, outperforming both autoregressive and non-autoregressive baselines.**

| *ID* | *System* | *Quality* ↑ | | | *Inference Speed* ↑ | |
|---|---|---|---|---|---|---|
| | | En-Es | En-Fr | Fr-En* | Speed | Speedup |
| **Autoregressive** | | | | | | |
| 1 | Transformer† [31] | 10.07 | 15.28 | 15.44 | 870 | 1.00× |
| 2 | Norm Transformer† [32] | 12.98 | 15.93 | 15.81 | 870 | 1.00× |
| 3 | Conformer† | 13.75 | 17.07 | 18.02 | 895 | 1.02× |
| **Non-autoregressive Model** | | | | | | |
| 4 | CMLM | 12.58 | 15.62 | 16.95 | 4651 | 5.34× |
| 5 | CMLM + BiP†[21] | 12.62 | 16.97 | 18.03 | | |
| **Our Improved Non-autoregressive Model** | | | | | | |
| 6 | CMLM + DIFFNORM | 18.96 | 17.27 | **19.53** | 4651 | 5.34× |
| 7 | CMLM + CG‡ | 17.06 | 16.89 | _ | | |
| 8 | CMLM + DIFFNORM + CG‡ | **19.49** | **17.54** | _ | | |

Table 2: Comparison of speech-to-speech models evaluated by quality (ASR-BLEU) and speed (units/seconds).*: Fr-En experiments are added during author response period and we leave model 7,8 for future work. Results with † are taken from the prior work [21]. ‡ We use $w = 0.5$ for CG. **Our NAT models achieve superior translation quality while maintaining their fast inference speed**.

**Decoding speed** In Fig. 4, we illustrate the "quality-latency" trade-off of various non-autoregressive speech-to-unit systems. Quality is measured using ASR-BLEU, while latency is determined by the relative speedup over the autoregressive system, calculated as "generated units/second". For instance, the first marker on the line plot represents 15 decoding iterations, resulting in a speedup of 5.34× compared to the autoregressive baseline. Additionally, we include the performance of the Conformer-based autoregressive model as a horizontal dashed line. We observe that our improved system consistently outperforms the baseline CMLM model [21] and achieves better performance than the autoregressive model with more than 14× speedup for En-Es and 5× speedup for En-Fr.

## 6 Analysis

### 6.1 Effect of synthetic noise injection

In this section, we explore the effects of varying degrees of noise injection on DIFFNORM. Recall (from Alg. 2) that synthetic noise is injected into the latent representation as $\boldsymbol{z}_T = \sqrt{\bar{\alpha}_T}\boldsymbol{z}_0 + \sqrt{1-\bar{\alpha}_T}\boldsymbol{\epsilon}$. Hence, adjusting the start time $T$ allows us to control the degree of noise injection.

**Setup and Evaluation** We perturb the degree of noise injection by varying $T$, and assess the *reconstruction quality* of normalized speech units and *downstream performance* of CMLM models trained with different normalized units. For reconstruction quality, we measure the accuracy (**Acc-Rec**) of reconstructed units and ASR-BLEU (**BL-Rec**) of synthesized speech when using original


https://doi.org/10.52202/079017-1048

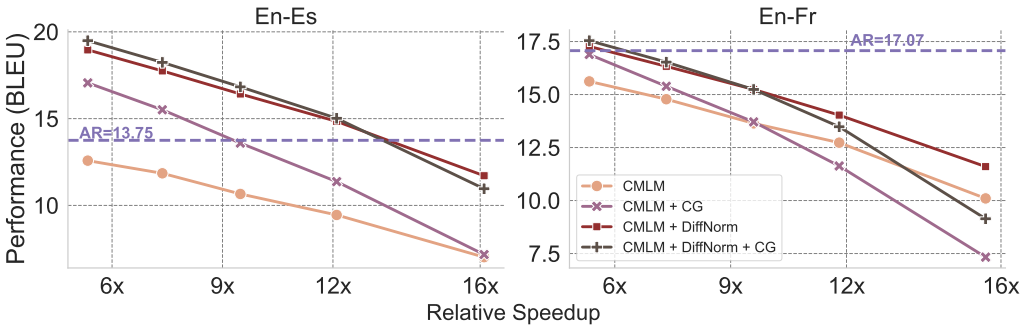


Figure 4: Trade-off between quality (ASR-BLEU) and latency for varying numbers of decoding iterations. Five markers correspond to {15, 10, 7, 5, 3} decoding iterations. Decreasing the number of iterations results in a decline in model performance, traded off for faster speedup. With DIFFNORM and CG, **our S2UT model achieves a better quality-latency trade-off** than CMLM and outperforms a strong autoregressive baseline with large speedups.

units/speech as the reference. For downstream performance, we report the downstream ASR-BLEU of translated speech produced by CMLM models trained with the particular noise setup (**BL-Dn**).[7]

**Results** Shown in Table 5, increasing $T$ leads to more noise and a decline in reconstruction quality, which aligns with expectations that higher noise levels pose greater challenges for the diffusion model in restoring original features accurately. Interestingly, we find from Table 3 that when $T$ is too small or large (e.g., $T = 50$ or $T = 150$), the normalized units do not result in an ideal downstream system. We hypothesize that there is barely any normalization effect when $T$ is too small; and when $T$ is too large, the reconstructed units are no longer semantically correct. This phenomenon is visualized in Fig. 5, where we plot the log-mel spectrograms of the reconstructed speech. As more noise is introduced (larger $T$), the reconstructed speech becomes smoother (e.g., portions of blank speech are filled in by diffusion), exhibiting a more pronounced deviation from the original speech.

| ***Start Step*** | ***Scheduler Parameters*** | | | ***En-Es*** | | | ***En-Fr*** | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\beta_t$ | $\sqrt{\bar{\alpha}_t}$ | $\sqrt{1-\bar{\alpha}_t}$ | **Acc-Rec** ↑ | **BL-Rec** ↑ | **BL-Dn** ↑ | **Acc-Rec** ↑ | **BL-Rec** ↑ | **BL-Dn** ↑ |
| original units | N/A | N/A | N/A | 100 | 48.7 | 12.58 | 100 | 40.64 | 15.62 |
| $T = 50$ | 0.007 | 0.917 | 0.398 | **89.4** | **48.5** | 18.56 | **91.1** | **40.38** | 17.12 |
| $T = 100$ | 0.016 | 0.697 | 0.717 | 81.2 | 47.63 | **18.96** | 83.0 | 39.9 | 16.69 |
| $T = 120$ | 0.022 | 0.577 | 0.816 | 75.3 | 46.25 | 16.7 | 77.7 | 39.03 | **17.27** |
| $T = 150$ | 0.038 | 0.373 | 0.928 | 57.8 | 31.43 | 14.77 | 60.0 | 28.3 | 14.03 |

Table 3: For different start steps $T$, we show corresponding noise scheduling parameter values, reconstruction quality (**-Rec** columns), and downstream translation quality (**-Dn** column). **Noise injection that perturbs about 20% of units (i.e., 80% Acc-Rec) results in the best downstream S2UT performance (highlighted in bold text)**.

From the Table 3, we observed a substantial enhancement in En-Es translation performance with $T = 50$. Consequently, we conducted additional experiments for En-Es translation with start times $T = 10$ and $T = 30$, detailed in Table 4. The results show that at a minimal start time of $T = 10$, the reconstruction accuracy reaches 93.8%, yielding a downstream ASR-BLEU score of 15.98. This score significantly surpasses the baseline CMLM result of 12.58. The high accuracy indicates that about 6% of tokens vary post-reconstruction, likely due to the VAE model's regularization effect since the diffusion model does not cause notable reconstruction deviations at such a small $T$. This suggests that the Spanish speech dataset might have considerable acoustic variations and noise, which the VAE model can partially mitigate. As $T$ increases, the diffusion model further refines the representation, leading to improved downstream results, as evidenced by the rise in ASR-BLEU score from from

| ***Start Step*** | **Acc-Rec** | **BL-Rec** |
|---|---|---|
| $T = 10$ | 93.8 | 15.98 |
| $T = 30$ | 91.6 | 17.92 |

Table 4: Reconstruction and downstream performance with small noise injection.

[7]ASR-BLEU is computed by synthesizing speech from the reconstructed units and transcribing it using the ASR model, as detailed in §5.1.

https://doi.org/10.52202/079017-1048

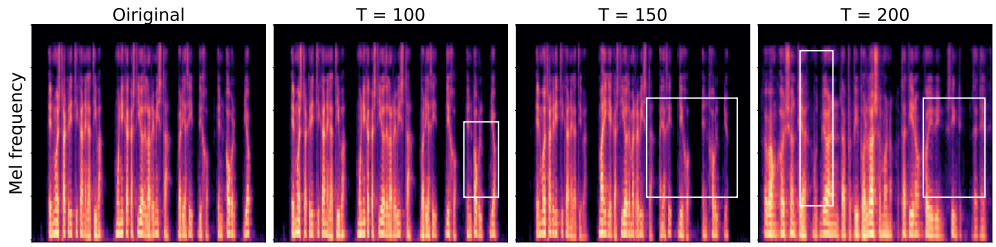


Figure 5: Visualization of reconstructed speech's log-mel spectrograms. Noticeable divergence from the original speech is highlighted in the white bounding boxes.

$T = 10$ to $T = 100$. This ablation study underscores that **the optimal choice of T is affected by the dataset quality**.

### 6.2 Ablation on training objectives for DIFFNORM

DIFFNORM requires an auto-encoder to map speech features into lower-dimension latents and then train denoising diffusion models using the encoded latents. In this section, we perform ablation experiments to investigate the effect of (1) latent dimension (2) Gaussian constraints (3) multitask objective for diffusion models. Firstly, to investigate the effect of latent dimension, we train auto-encoders that encode speech features into 16, 32, and 128 dimensions. As shown in Table 5, it comes as no surprise that a larger latent dimension yields superior reconstruction results, evidenced by higher accuracy across systems trained with varying objectives. Next, we turn our attention to the importance of applying Gaussian constraints to the auto-encoder's latent representation space. Our results (comparing with and without **KL**) reveal that incorporating Gaussian constraints is crucial. Latent spaces not regularized by $\mathcal{L}_{\text{kl}}$ lead to significantly poorer performance. Finally, we explore the effectiveness of our proposed multitasking objective for the diffusion model by training another vanilla DDPM solely with $\mathcal{L}_{\text{noise}}$ (no **Multitask**). We find that the diffusion model's reconstruction capability indeed improves when employing the multitasking objective, particularly when denoising from a more noisy latent representation (e.g., $T = 150$). Through our ablation analysis, we identify the optimal setup for our DiffNorm model, which involves (1) mapping speech features to 128 dimensions, (2) regularizing latent space by Gaussian constraints, and (3) utilizing the multitask objective for diffusion training.

| Start Step | Objectives | | Latent Dimension | | |
|---|---|---|---|---|---|
| | KL | Multitask | 16 | 32 | 128 |
| $T = 50$ | ✗ | ✓ | 51.4 | 69.1 | 85.8 |
| | ✓ | ✗ | 80.9 | 86.4 | 89.3 |
| | ✓ | ✓ | **80.9** | **86.5** | **89.4** |
| $T = 100$ | ✗ | ✓ | 13.3 | 32.5 | 73.1 |
| | ✓ | ✗ | 64.5 | 76 | 80.8 |
| | ✓ | ✓ | **65.3** | **76.2** | **81.2** |
| $T = 150$ | ✗ | ✓ | 2.9 | 5.3 | 34.6 |
| | ✓ | ✗ | 19.8 | 38.2 | 56.8 |
| | ✓ | ✓ | **20.2** | **40** | **57.8** |

Table 5: Accuray of reconstructed speech units. **KL**: when applied, the latent space is regularized to be Gaussian [27]. **Multitask**: when not applied, the latent diffusion model is trained only with $\mathcal{L}_{\text{noise}}$.

## 7 Related work

**Direct speech-to-speech translation (S2ST)** Direct S2ST aims to directly translate speech into another language without cascaded systems that rely on transcriptions or translations. Translatotron [25] and Translatotron 2 [23] are among the first systems for direct S2ST, which uses sequence-to-sequence model to map source speech into spectrograms. Then, a spectrogram decoder is used to synthesize the target language's speech. Instead of transducing speech into spectrogram, Tjandra et al. [53], Zhang et al. [62] utilize Vector-Quantized Variational Auto-Encoder (VQ-VAE) [56] to discretize target speech and convert *speech-to-speech* translation into a *speech-to-unit* task. More recently, Lee et al. [31] improved the speech-to-unit models by obtaining such units with a k-means clustering model trained on self-supervised representation from HuBERT [19]. To convert units back to speech, Lee et al. [31] follows Polyak et al. [38] to train a unit-vocoder based on HifiGAN [2].

**Speech normalization** Speech representation are typically extracted from pre-trained encoders [19, 3, 40, 5] and can be compressed or adapted in different speech-to-speech/text tasks [61, 63, 50, 52, 51]. Inspired by previous work on speech enhancement [54, 1], Lee et al. [32] propose to


https://doi.org/10.52202/079017-1048

normalize speech by synthesizing speaker-invariant waveforms through text-to-speech (TTS) systems [60, 46, 42, 41]. Huang et al. [21] propose to normalize speech with Bilateral Perturbation that focus on the rhythm, pitch, and energy information. Different from previous normalization methods that requires transcription [32] or manually designed perturbation [21], our DIFFNORM strategy leverages diffusion models. **Diffusion models** [15, 48, 49] have achieved remarkable generative ability to produce high-quality images [8, 16, 43, 37, 45] and audio [33, 39, 47]. Using self-supervised denoising objectives [15], we train effective denoising models capable of normalizing speech for training better speech-to-unit systems.

**Non-autoregressive speech-to-speech translation** For sequence-to-sequence modeling, autoregressive [18, 6, 57] and non-autoregressive [12, 13, 10, 20] models have been widely explored. Lee et al. [31] reduce speech-to-speech into speech-to-unit task and follow the widely-used modeling strategy, Transformer [57], to predict speech units. Later, non-autoregressive models [21, 9] have been explored and Huang et al. [21] is the first to use a non-autoregressive transformer model, Conditional Masked Language Model [10, CMLM], for speech-to-unit task. Fang et al. [9], on the other hand, adopts Directed Acyclic Transformer [20] for speech-to-unit translation.

## 8 Conclusion

We improve speech-to-unit translation system through (1) corpus distillation by constructing normalized speech units with DIFFNORM and (2) regularization with classifier-free guidance. Our improved non-autoregressive (NAR) models, greatly outperform previous NAR models, yielding an increase of approximately $+7$ and $+2$ BLEU points for En-Es and En-Fr translation, respectively. Notably, DIFFNORM and classifier-free guidance maintain the inference speed advantages inherent in NAR models. Consequently, our approach obtains better performance to autoregressive baselines while achieving over $14\times$ speedup for En-Es and $5\times$ speedup for En-Fr translations.

## Acknowledgement

We express our profound appreciation to anonymous reviewers for their helpful suggestions. We also thank Xiuyu Li for his valuable suggestions, which greatly enriched our work. Lastly, we thank JHU + Amazon Initiative for Interactive AI for sponsoring the work.

## References

[1] Adiga, N., Pantazis, Y., Tsiaras, V., and Stylianou, Y. (2019). Speech Enhancement for Noise-Robust Speech Synthesis Using Wasserstein GAN. In *Proc. Interspeech 2019*, pages 1821–1825.

[2] Andreev, P., Alanov, A., Ivanov, O., and Vetrov, D. (2023). Hifi++: A unified framework for bandwidth extension and speech enhancement. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

[3] Babu, A., Wang, C., Tjandra, A., Lakhotia, K., Xu, Q., Goyal, N., Singh, K., von Platen, P., Saraf, Y., Pino, J. M., Baevski, A., Conneau, A., and Auli, M. (2021). Xls-r: Self-supervised cross-lingual speech representation learning at scale. In *Interspeech*.

[4] Baevski, A., Auli, M., and rahman Mohamed, A. (2019). Effectiveness of self-supervised pre-training for speech recognition. *ArXiv*, abs/1911.03912.

[5] Baevski, A., Zhou, H., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations.

[6] Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

[7] Chen, P.-J., Tran, K., Yang, Y., Du, J., Kao, J., Chung, Y.-A., Tomasello, P., Duquenne, P.-A., Schwenk, H., Gong, H., Inaguma, H., Popuri, S., Wang, C., Pino, J., Hsu, W.-N., and Lee, A. (2023). Speech-to-speech translation for a real-world unwritten language. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4969–4983, Toronto, Canada. Association for Computational Linguistics.

[8] Dhariwal, P. and Nichol, A. (2021). Diffusion models beat gans on image synthesis.

[9] Fang, Q., Zhou, Y., and Feng, Y. (2023). DASpeech: Directed acyclic transformer for fast and high-quality speech-to-speech translation. In *Advances in Neural Information Processing Systems*.

[10] Ghazvininejad, M., Levy, O., Liu, Y., and Zettlemoyer, L. (2019). Mask-predict: Parallel decoding of conditional masked language models. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6112–6121, Hong Kong, China. Association for Computational Linguistics.

[11] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks.

[12] Gu, J., Bradbury, J., Xiong, C., Li, V. O., and Socher, R. (2018). Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

[13] Gu, J., Wang, C., and Zhao, J. (2019). Levenshtein transformer. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

[14] Gulati, A., Qin, J., Chiu, C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., and Pang, R. (2020). Conformer: Convolution-augmented transformer for speech recognition. *CoRR*, abs/2005.08100.

[15] Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc.

[16] Ho, J., Saharia, C., Chan, W., Fleet, D. J., Norouzi, M., and Salimans, T. (2021). Cascaded diffusion models for high fidelity image generation.

[17] Ho, J. and Salimans, T. (2022). Classifier-free diffusion guidance.

[18] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[19] Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., and Mohamed, A. (2021). Hubert: Self-supervised speech representation learning by masked prediction of hidden units.

[20] Huang, F., Zhou, H., Liu, Y., Li, H., and Huang, M. (2022). Directed acyclic transformer for non-autoregressive machine translation. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S., editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 9410–9428. PMLR.

[21] Huang, R., Liu, J., Liu, H., Ren, Y., Zhang, L., He, J., and Zhao, Z. (2023). Transpeech: Speech-to-speech translation with bilateral perturbation. In *The Eleventh International Conference on Learning Representations*.

[22] Inaguma, H., Popuri, S., Kulikov, I., Chen, P.-J., Wang, C., Chung, Y.-A., Tang, Y., Lee, A., Watanabe, S., and Pino, J. (2023). UnitY: Two-pass direct speech-to-speech translation with discrete units. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15655–15680, Toronto, Canada. Association for Computational Linguistics.

[23] Jia, Y., Ramanovich, M. T., Remez, T., and Pomerantz, R. (2022a). Translatotron 2: High-quality direct speech-to-speech translation with voice preservation.

[24] Jia, Y., Ramanovich, M. T., Wang, Q., and Zen, H. (2022b). Cvss corpus and massively multilingual speech-to-speech translation.

[25] Jia, Y., Weiss, R. J., Biadsy, F., Macherey, W., Johnson, M., Chen, Z., and Wu, Y. (2019). Direct speech-to-speech translation with a sequence-to-sequence model. *ArXiv*, abs/1904.06037.

[26] Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization.

 https://doi.org/10.52202/079017-1048

[27] Kingma, D. P. and Welling, M. (2022). Auto-encoding variational bayes.

[28] Kong, J., Kim, J., and Bae, J. (2020). Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis.

[29] Lavie, A., Waibel, A., Levin, L., Finke, M., Gates, D., Gavalda, M., Zeppenfeld, T., and Zhan, P. (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.

[30] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[31] Lee, A., Chen, P.-J., Wang, C., Gu, J., Popuri, S., Ma, X., Polyak, A., Adi, Y., He, Q., Tang, Y., Pino, J., and Hsu, W.-N. (2022a). Direct speech-to-speech translation with discrete units. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3327–3339, Dublin, Ireland. Association for Computational Linguistics.

[32] Lee, A., Gong, H., Duquenne, P.-A., Schwenk, H., Chen, P.-J., Wang, C., Popuri, S., Adi, Y., Pino, J., Gu, J., and Hsu, W.-N. (2022b). Textless speech-to-speech translation on real data. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V., editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 860–872, Seattle, United States. Association for Computational Linguistics.

[33] Liu, H., Chen, Z., Yuan, Y., Mei, X., Liu, X., Mandic, D., Wang, W., and Plumbley, M. D. (2023). AudioLDM: Text-to-audio generation with latent diffusion models. *Proceedings of the International Conference on Machine Learning*.

[34] Nakamura, S., Markov, K., Nakaiwa, H., Kikui, G., Kawai, H., Jitsuhiro, T., Zhang, J.-S., Yamamoto, H., Sumita, E., and Yamamoto, S. (2006). The atr multilingual speech-to-speech translation system. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(2):365–376.

[35] Nichol, A. Q. and Dhariwal, P. (2021). Improved denoising diffusion probabilistic models. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8162–8171. PMLR.

[36] Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. (2019). fairseq: A fast, extensible toolkit for sequence modeling. In Ammar, W., Louis, A., and Mostafazadeh, N., editors, *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.

[37] Peebles, W. and Xie, S. (2023). Scalable diffusion models with transformers.

[38] Polyak, A., Adi, Y., Copet, J., Kharitonov, E., Lakhotia, K., Hsu, W.-N., Mohamed, A., and Dupoux, E. (2021). Speech resynthesis from discrete disentangled self-supervised representations.

[39] Popov, V., Vovk, I., Gogoryan, V., Sadekova, T., and Kudinov, M. (2021). Grad-tts: A diffusion probabilistic model for text-to-speech. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8599–8608. PMLR.

[40] Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. (2022). Robust speech recognition via large-scale weak supervision.

[41] Ren, Y., Hu, C., Tan, X., Qin, T., Zhao, S., Zhao, Z., and Liu, T.-Y. (2022). Fastspeech 2: Fast and high-quality end-to-end text to speech.

[42] Ren, Y., Ruan, Y., Tan, X., Qin, T., Zhao, S., Zhao, Z., and Liu, T.-Y. (2019). Fastspeech: Fast, robust and controllable text to speech.

[43] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models.

[44] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation.

[45] Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., Salimans, T., Ho, J., Fleet, D. J., and Norouzi, M. (2022). Photorealistic text-to-image diffusion models with deep language understanding.

[46] Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerry-Ryan, R., Saurous, R. A., Agiomyrgiannakis, Y., and Wu, Y. (2018). Natural tts synthesis by conditioning wavenet on mel spectrogram predictions.

[47] Shen, K., Ju, Z., Tan, X., Liu, Y., Leng, Y., He, L., Qin, T., Zhao, S., and Bian, J. (2023). Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers.

[48] Song, J., Meng, C., and Ermon, S. (2021a). Denoising diffusion implicit models. In *International Conference on Learning Representations*.

[49] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021b). Score-based generative modeling through stochastic differential equations.

[50] Tan, W., Chen, Y., Chen, T., Qin, G., Xu, H., Zhang, H. C., Durme, B. V., and Koehn, P. (2024a). Streaming sequence transduction through dynamic compression.

[51] Tan, W., Inaguma, H., Dong, N., Tomasello, P., and Ma, X. (2024b). Ssr: Alignment-aware modality connector for speech language models.

[52] Tang, C., Yu, W., Sun, G., Chen, X., Tan, T., Li, W., Lu, L., Ma, Z., and Zhang, C. (2024). Salmonn: Towards generic hearing abilities for large language models.

[53] Tjandra, A., Sakti, S., and Nakamura, S. (2019). Speech-to-speech translation between untranscribed unknown languages. *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 593–600.

[54] Valentini-Botinhao, C., Wang, X., Takaki, S., and Yamagishi, J. (2016). Speech Enhancement for a Noise-Robust Text-to-Speech Synthesis System Using Deep Recurrent Neural Networks. In *Proc. Interspeech 2016*, pages 352–356.

[55] van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio.

[56] van den Oord, A., Vinyals, O., and kavukcuoglu, k. (2017). Neural discrete representation learning. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[57] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[58] Wang, C., Riviere, M., Lee, A., Wu, A., Talnikar, C., Haziza, D., Williamson, M., Pino, J., and Dupoux, E. (2021). VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 993–1003, Online. Association for Computational Linguistics.

[59] Wang, C., Wu, A., and Pino, J. (2020). Covost 2 and massively multilingual speech-to-text translation.

[60] Wang, Y., Skerry-Ryan, R. J., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., Le, Q. V., Agiomyrgiannakis, Y., Clark, R. A. J., and Saurous, R. A. (2017). Tacotron: Towards end-to-end speech synthesis. In *Interspeech*.

 https://doi.org/10.52202/079017-1048

[61] Yu, W., Tang, C., Sun, G., Chen, X., Tan, T., Li, W., Lu, L., Ma, Z., and Zhang, C. (2023). Connecting speech encoder and large language model for asr.

[62] Zhang, C., Tan, X., Ren, Y., Qin, T., Zhang, K., and Liu, T.-Y. (2020). Uwspeech: Speech to speech translation for unwritten languages.

[63] Zhang, D., Li, S., Zhang, X., Zhan, J., Wang, P., Zhou, Y., and Qiu, X. (2023). Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities.

# Supplementary Material

| Appendix Sections | Contents |
|---|---|
| Appendix A | Qualitative Examples of Translated Speech |
| Appendix B | Architecture and Hyper-parameters of Diffusion Model for DIFFNORM |
| Appendix C | Architecture and Hyper-parameters of Non-autoregressive Models |
| Appendix D | Unit-to-Speech Synthesis |
| Appendix E | Limitations |
| Appendix F | Full Results for Classifier-free Guidance |
| Appendix G | Experiments Results Table |

## A Qualitative Examples

From §5.2 we demonstrate the effectiveness of DiffNorm and classifier-free guidance by evaluating ASR-BLEU scores. In this section, we provide example transcriptions of generated Spanish and French speech in Table 6 and Table 7. Compared to the vanilla CMLM model that has erroneous or incomplete translation, our method results in higher-quality translation.

| Model | Generated Translation | English Version |
|---|---|---|
| Reference | este trabajo fue posible en cooperación con otros como alphonse milne-edwards y leon vaillant. | This work was made possible in cooperation with others such as Alphonse Milne-Edwards and leon vaillant |
| CMLM | esta obra fue posible para lugar con otros como alphon y leol vigelante | This work was possible to place with others such as Alphon and Leol Vigelante |
| CMLM + CG | este hora fue hizo por la cooperación con otros con walphonxe mill y leon fallent | This time was made by cooperation with others with Walphonxe Mill and Leon Fallent |
| CMLM + DiffNorm | esta obra fue posible la cooperación con otros como al von milo edwad y leon vallente | This work was possible through cooperation with others such as Al von Milo Edwad and Leon valente |
| CMLM + DiffNorm + CG | este trabajo fue posible por la cooperación con otros como alphonso edos y león valente | This work was possible through the cooperation with others such as Alphonso Edos and León Valente |

Table 6: Example transcription of translated Spanish speech from different systems.

## B Diffusion Model Hyperparameters

### B.1 Variational Auto-Encoder

**Architecture**: Our Auto-encoder consists of an encoder, a decoder, and a language modeling head. The encoder architecture follows WaveNet [55] that combines Convolutional Neural Networks [30] with residual connections. In practice, we use 2 stacks of WaveNet Residual Blocks, where each block has 3 layers. Our decoder uses the same architecture as the encoder to revert the latent dimension back

| Model | Generated Translation | English Version |
|---|---|---|
| Reference | mais la principauté est envahie et le prince-évêque joseph-clément de bavière doit s'exiler | but the principality is invaded and the prince-bishop Joseph-Clement of Bavaria must go into exile |
| CMLM | mais la précicipauté et le prince joseph clémont de bavier est fisé en exil | but the precicipality and prince joseph clemont of bavier is exiled |
| CMLM + CG | mais la principauté est lebséve josefh crémend de bavière et contré en exil | but the principality is lebséve josefh cremend of bavaria and countered in exile |
| CMLM + DiffNorm | la princepôuté est envahie et le prince-évêque joser clémant de bavière est contr | the princepôuté is invaded and the prince-bishop joser clemant of bavaria is opposed |
| CMLM + DiffNorm + CG | mais la principauté est envahie et le prend s évêque joseph de bavier et borgé en exile | but the principality is invaded and takes bishop joseph of bavaria and borgé into exile |

Table 7: Example transcription of translated French speech from different systems.

to the original dimension. Additionally, we apply a Transformer [57] encoder on top of the decoded feature to further enhance model capacity. The Transformer-encoder has the same configuration as the vanilla Transformer-base model (6 layers, 8 attention heads, etc.). Lastly, to decode speech units from the feature, we use a language modeling head which is parameterized by a feedforward network that converts feature dimension ($H = 768$) into vocabulary size ($V = 1000$).

**Training**: To train the VAE model, as described in §3.1, we use a combination of three objectives

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{recon}} + \lambda_2 \mathcal{L}_{\text{nll}} + \lambda_3 \mathcal{L}_{\text{kl}} \tag{8}$$

where $\lambda_1 = 100, \lambda_2 = 1, \lambda_3 = 0.001$. Now we provide more details about $\mathcal{L}_{\text{kl}}$: in practice, we follow [27] to model the latent $\boldsymbol{z}$ by estimating its mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}$ using the encoder. With the re-parameterization trick, the latent is sampled as $\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \cdot \boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$. By constraining the mean and variance to follow a Gaussian prior, Kingma and Welling [27] showed that

$$\mathcal{L}_{\text{kl}} = \frac{1}{2} \sum_{j=1}^{J} (1 + \log((\sigma_j)^2) - (\mu_j)^2 - (\sigma_j)^2) \tag{9}$$

We implement our VAE model on Fairseq [36]. For optimization, we use the Adam [26] optimizer with betas $(0.9, 0.98)$ and we apply gradient clipping by setting –clip-norm=2.0. During training, we apply dropout with a probability of 0.1. We train the VAE model using a learning rate of 5e-4 with distributed data-parallel (DDP) on 4 A100 GPUs, where we set the maximum batch token to be 15000.

### B.2 Latent Diffusion Model

**Architecture**: We modified DiT [37] to design a Transformer-based architecture for noise estimation. Since our input latent is a sequence of encoded speech feature $\boldsymbol{z} \in \mathbb{R}^{M \times Z}$, we first apply a 1D Convolution to convert the latent to our model's hidden dimension (set to 512). Then we also encode the timestep with a learnable Sinusoidal Positional Embedding [57] to obtain the time embedding.

Subsequently, we feed the latent and time embedding to our modified WaveNet Block, which applies an affine transformation of the latent using the time embedding, similar to the "Scale and Shift" operation in DiT's adaptive layer norm (adaLN) block. For our diffusion model, we use 4 stacks of such modified WaveNet Block, each containing 8 layers. The output feature from WaveNet is then passed through a Transformer-Encoder (12 layers with 8 attention heads) to further enhance model capacity. Lastly, a projection layer parameterized by the feedforward network is used to compress the transformed feature to the latent dimension, which is then used for noise estimation in equation (3).

**Training**: As described in §3.2, we train the diffusion model with a multitask objective:

$$\mathcal{L} = \gamma_1 \mathcal{L}_{\text{noise}} + \gamma_2 \mathcal{L}_{\text{recon}} + \gamma_3 \mathcal{L}_{\text{nll}} \tag{10}$$

and we empirically select $\gamma_1 = 1, \gamma_2 = 0.25, \gamma_3 = 0.005$. For optimization, our latent diffusion model has the same setting as our VAE model (with Adam optimizer, clip-norm equals 2.0, and dropout with 0.1 probability). We train the diffusion model using a learning rate of 1e-4 with DDP on 4 A100 GPUs, where we set the maximum batch token to be 12000. The model is warmup-ed by 10000 steps.


https://doi.org/10.52202/079017-1048

## C Non-autoregressive Transformer Details

### C.1 Background on Non-Autoregressive Transformers

Non-autoregressive Transformers [12, 13, 10, *inter alia.*] is a family of models that transduce sequences. In this work, we follow the formalization from the widely-used NAT: Conditional Masked Language Model [10, CMLM]. CMLM consists of an encoder $\phi_{\text{enc}}$ that represents the input as high-dimensional features and a decoder $\phi_{\text{dec}}$ that generates tokens conditioning on the encoded features. Different from auto-regressive Transformers [57], CMLM's decoder does not apply causal masking and is trained with an unmasking objective.

**CMLM Training** Following formulation from §2, given the input speech $\boldsymbol{x} = (x_1, \cdots, x_N)$ and target speech units $\boldsymbol{y} = (y_1, \cdots, y_M)$, CMLM first mask the target units into $\hat{\boldsymbol{y}}$. The total amount of masked token is uniformed sampled: $n \sim \mathcal{U}[1, M]$; subsequently, $n$ of $M$ tokens from the target units $\boldsymbol{y}$ are randomly masked to form noisy target units $\hat{\boldsymbol{y}}$. Then, the decoder generates a distribution over vocabulary conditioning on the encoder representation and noisy target units $\boldsymbol{v} = f(\hat{\boldsymbol{y}}|\boldsymbol{g}; \phi_{\text{dec}})$ where $\boldsymbol{g} = f(\boldsymbol{x}; \phi_{\text{enc}})$ is the encoder representation. Cross-entropy loss is then computed for the masked positions to update model parameters. CMLM also trains a length predictor that estimates the output length given input sequences and we refer readers to [21] for more details.

**CMLM Mask-Predict**: For CMLM inference, an iterative decoding scheme is used through unmasking speech units and re-masking low-confident positions. Given the input sequence $\boldsymbol{x}$, CMLM first predicts the length of output sequence $M$ and initialize a sequence of $M$ masked tokens $\hat{\boldsymbol{y}}_0 = ([\text{mask}]_1, \cdots, [\text{mask}]_M)$. Given the total number of iterations $T$ and current iteration $t \in [1, T]$, CMLM decodes tokens across all positions and also computes their log-probabilities:

$$y_i^t = \underset{w}{\text{argmax}} \mathbb{P}(y_i = w|\boldsymbol{x}, \hat{\boldsymbol{y}}_{t-1}; \phi_{\text{dec}}) \quad p_i^t = \log \mathbb{P}(y_i^t|\boldsymbol{x}, \hat{\boldsymbol{y}}_{t-1}; \phi_{\text{dec}}) \tag{11}$$

From the generated tokens $\hat{\boldsymbol{y}}_t = (y_1^t, \cdots, y_M^t)$, we replace $k = M \cdot \frac{T-t}{T}$ tokens with the lowest probabilities $p_i^t$ with mask tokens. This re-masked sequence $\hat{\boldsymbol{y}}_t$ is then inputted into the CMLM for further decoding iterations. This process continues until the final step $t = T$, at which point no tokens are re-masked.

### C.2 Architecture

We follow the same architecture and implementation from Huang et al. [21], where a Conformer-based encoder is used to obtain representation from the source speech and a Transformer-decoder (with no causal masking) is used to generate units. The Conformer-encoder has 12 layers with 512 hidden dimensions and 9 attention heads. The encoder also contains a CNN-based sub-sampler that has an effective stride size of 320 (i.e., it reduces the length of speech input by 320×). The Transformer-decoder has 6 layers with 8 attention heads and uses a fixed positional embedding for speech units. When classifier-free guidance is used, we randomly dropout encoding from the Conformer model with a probability of 15% as described in §4.

### C.3 Hyper-parameter for Model Training

We train the model with a learning rate of 5e-4, using 10000 warmup steps We apply Adam optimizer with betas $(0.9, 0.98)$ and we clip the gradient by setting –clip-norm=10. We apply dropout with 0.1 probability and we use label smoothing of ratio 0.2 when computing the NLL loss. We train the model with DDP on 4 A100 GPUs, with a maximum batch tokens of 40,000.

## D Unit-to-Speech Synthesis

We follow prior work [32, 38, 41] to convert predicted units (from S2UT model) into speech waveforms. Specifically, given units $\boldsymbol{y}$, we use the HifiGAN-based unit-vocoder from [38] to synthesize speech. The unit-vocoder is trained using a generator $\mathcal{G}$ and discriminator network $\mathcal{D}$. During training, the generator is updated with reconstruction loss based on the mel-spectrogram of the true and synthesized waveforms. Additionally, adversarial loss and feature-matching loss are added to enhance the fidelity of generated speech. To train the discriminator, Polyak et al. [38] follows the minmax objective from

GAN [11]. After training, the discriminator is discarded and only the generator is used to synthesize waveforms.

We follow [32] to use a unit-vocoder that is also trained with a duration predictor so that it can synthesize speech with reduced speech units. For more details, we refer readers to the discussion from the original paper [32, 41], as well as the open-sourced repository: `github.com/facebookresearch/fairseq/examples/speech_to_speech`

## E Limitations and Broader Impacts

**Limitations**: The major limitation of the work is the data pre-processing cost from DIFFNORM, as our method requires inferencing diffusion model on all training samples to obtain better speech units. With the CVSS benchmark, both En-Es and En-Fr have less than 1M pairs, which can be processed easily with our resource (4 A100 GPUs). However, when the dataset scales up, the corpus distillation cost could be large.

**Broader Impacts**: Our system helps bridge the communication gap between users of different languages. However, as our system involves speech synthesis, it also has a risk of being used to mimic a particular speaker.

## F Ablation on Classifier-free Guidance

In this section, we compare non-autoregressive transformers trained with and without classifier-free guidance (CG). We train CMLM models with classifier-free guidance and evaluate them with $5, 10, 15$ iterations of decoding. From Table 8, we observe that CG improves the quality of translation whether the model is trained on original or normalized speech units. We find CG brings more improvement on the original units than the normalized units. This happens because normalized speech units are more conformed and already result in a large improvement in their translation quality, making the regularization effect from CG less obvious. Nevertheless, the best-performing system is achieved with both DIFFNORM and CG.

Comparing different hyperparameters $w$, we find a small value like $w = 0.5$ or $w = 1$ brings the most improvement empirically, and such improvements are more noticeable when the number of decoding iterations is larger. For example, comparing the results under 5 and 15 iterations, we find $w = 0$ gives better results when the number of iterations is small while $w = 0.5$ obtains the best performance with 15 iterations.

| Model | En-Es | | | En-Fr | | |
|---|---|---|---|---|---|---|
| | 5 | 10 | 15 | 5 | 10 | 15 |
| CMLM | 9.45 | 11.85 | 12.58 | **12.73** | 14.78 | 15.62 |
| CMLM + CG (w=0) | **12.22** | **15.58** | 16.62 | 12.13 | 15.23 | 16.65 |
| CMLM + CG (w=0.5) | 11.37 | 15.51 | **17.06** | 11.63 | **15.39** | **16.89** |
| CMLM + CG (w=1) | 11.27 | 15.4 | 16.99 | 11.52 | 15.44 | 16.65 |
| CMLM + CG (w=2) | 10.98 | 14.97 | 16.57 | 11.12 | 15.04 | 16.50 |
| CMLM + CG (w=3) | 10.63 | 14.7 | 16.22 | 10.80 | 14.71 | 16.17 |
| CMLM + DiffNorm | 14.84 | 17.75 | 18.96 | **14.03** | 16.33 | 17.27 |
| CMLM + DiffNorm + CG (w=0) | 14.69 | 17.57 | 18.63 | 13.03 | 16.05 | 17.13 |
| CMLM + DiffNorm + CG (w=0.5) | **15.02** | **18.24** | **19.49** | 13.48 | **16.53** | **17.54** |
| CMLM + DiffNorm + CG (w=1) | 14.77 | 18.18 | 19.37 | 13.29 | 16.27 | 17.48 |
| CMLM + DiffNorm + CG (w=2) | 14.03 | 17.65 | 18.9 | 12.8 | 15.97 | 17.15 |
| CMLM + DiffNorm + CG (w=3) | 13.45 | 17.09 | 18.5 | 12.23 | 15.44 | 16.52 |

Table 8: Speech-to-speech translation performance of CMLM models with different CG hyperparameters.


https://doi.org/10.52202/079017-1048

## G Experiment Result Tables

| Model | Number of Iterations | | | | |
|---|---|---|---|---|---|
| | 3 | 5 | 7 | 19 | 15 |
| CMLM | 7.02 | 9.45 | 10.66 | 11.85 | 12.58 |
| + CG | 7.17 | 11.37 | 13.59 | 15.51 | 17.06 |
| + DiffNorm | 11.71 | 14.84 | 16.42 | 17.75 | 18.96 |
| + DiffNorm + CG | 10.96 | 15.02 | 16.83 | 18.24 | 19.49 |

Table 9: Experimental Results of different En-Es speech-to-unit translation systems.

| Model | Number of Iterations | | | | |
|---|---|---|---|---|---|
| | 3 | 5 | 7 | 19 | 15 |
| CMLM | 10.1 | 12.73 | 13.64 | 14.78 | 15.62 |
| CMLM + CG | 7.33 | 11.63 | 13.71 | 15.39 | 16.89 |
| CMLM + DiffNorm | 11.6 | 14.03 | 15.24 | 16.33 | 17.27 |
| CMLM + DiffNorm + CG | 9.14 | 13.48 | 15.24 | 16.53 | 17.54 |

Table 10: Experimental Results of different En-Fr speech-to-unit translation systems.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We propose strategies to improve speech-to-speech translation which is covered in our abstract and introduction sections.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Appendix E for our discussion on limitations.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.

- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: There is no theoretical proof involved in this paper.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide details of our implementation (architecture and training hyper-parameters) in Appendix B and Appendix C. We will also release our code and the dataset we used (CVSS) is publicly available at https://github.com/google-research-datasets/cvss.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may


https://doi.org/10.52202/079017-1048

be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The data is publicly available at: https://github.com/google-research-datasets/cvss. The code is publicly accessible at: https://github.com/steventan0110/DiffNorm

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

Justification: We provide all our implementation detail in Appendix B and Appendix C.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We follow the standard evaluation setting, using the same dev and test set as prior work, and evaluate our method in two language directions (English-Spanish and English-French). Our results are consistent over languages, and our ablation studies that introduce additional experiments also confirm our method's effectiveness. Lastly, we improve upon the baseline with a notable $+7$ BLEU increment for En-Es and $+2$ BLEU increment for En-Fr over a test set that has more than 10,000 samples. With the evidence above, we believe our improvement is solid and alleviates the need to re-run experiments for significance tests. We believe our practice is standard and follows from previous work on speech-to-speech translation.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: As noted in Appendix B and Appendix C, we use 4 A100 GPUs with Distributed Data-Parallel setting throughout our experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.


https://doi.org/10.52202/079017-1048

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We follow the Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See Appendix E for our discussions.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our work focuses on direct speech-to-speech translation and we do not believe our model has a high risk for misuse as it does not support voice cloning or other high-risk applications.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite and describe the used dataset or models throughout the paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We do not have new assets in this paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?


https://doi.org/10.52202/079017-1048

Answer: [NA]

Justification: No human study is involved in this research.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: No human subjects or crowdsourcing are involved.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.