Speech Self-Supervised Learning Using Diffusion Model Synthetic Data

Heting Gao [1] **Kaizhi Qian** [2] **Junrui Ni** [1] **Chuang Gan** [2] **Mark Hasegawa-Johnson** [1] **Shiyu Chang** [3] **Yang Zhang** [2]

Abstract

While self-supervised learning (SSL) in speech has greatly reduced the reliance of speech processing systems on annotated corpora, the success of SSL still hinges on the availability of a large-scale unannotated corpus, which is still often impractical for many low-resource languages or under privacy concerns. Some existing work seeks to alleviate the problem by data augmentation, but most works are confined to introducing perturbations to real speech and do not introduce new variations in speech prosody, speakers, and speech content, which are important for SSL. Motivated by the recent finding that diffusion models have superior capabilities for modeling data distributions, we propose DIFFS4L, a pretraining scheme that augments the limited unannotated data with synthetic data with different levels of variations, generated by a diffusion model trained on the limited unannotated data. Finally, an SSL model is pre-trained on the real and the synthetic speech. Our experiments show that DIFFS4L can significantly improve the performance of SSL models, such as reducing the WER of the HuBERT pretrained model by 6.26 percentage points in the English ASR task. Notably, we find that the synthetic speech with all levels of variations, i.e. new prosody, new speakers, and even new content (despite the new content being mostly babble), accounts for significant performance improvement. The code is available at `github.com/Hertin/DiffS4L`.

1. Introduction

Self-supervised learning (SSL) in speech has greatly reduced the reliance of speech processing systems on large-scale annotated corpora. By pretraining a speech representation network on a large-scale unannotated dataset, SSL models only require a relatively small annotated dataset for finetuning, which has significantly improved the efficiency and feasibility of speech processing, particularly for low-resource languages. However, the success of such methods still hinges on the availability of a large-scale unannotated corpus. For example, the training of HuBERT (Hsu et al., 2021), one of the most widely-used speech pretraining models, typically requires that the unannotated corpus contains at least 1,000 hours of speech. If the dataset size drops to 100 hours, it tends to perform significantly worse. Yet, in many scenarios, obtaining such a large-scale dataset is impractical due to various constraints, e.g., low-resource languages, privacy concerns, *etc*. There have been many research attempts that perform data augmentation on training/pre-training data, but most of them perform perturbations to the real speech data, such as adding noise, and do not introduce many new speech variations, such as prosody, speaker, and content, which are important for SSL.

In situations where the pretraining dataset is limited, it becomes crucial to maximize the amount of information captured from the dataset to achieve the best performance in downstream tasks. Data augmentation can be regarded as introducing new variations/combinations in speech, which can supply new information to SSL. This raises the question – do existing SSL techniques have a high enough information efficiency? Could there be additional information that SSL models fail to capture, which would otherwise contribute to a better performance in downstream tasks? If so, could we design a data augmentation technique that can supply the overlooked information?

On the other hand, generative models are also often considered models that capture distributional information about data. Recently, diffusion models (Ho et al., 2020; Song et al., 2021), with their superior performance in computer vision, have quickly attracted wide research attention. Researchers have found that compared to other generative models, diffusion models can generate samples with much better global coherence (Li et al., 2022b) and local details (Dhariwal & Nichol, 2021), an indication that diffusion models may be able to capture more complete information from a limited dataset that could complement those learnable by existing

[1] University of Illinois at Urbana-Champaign, IL, USA [2] MIT-IBM Watson AI Lab, MA, USA [3] University of California, Santa Barbara, CA, USA. Correspondence to: Heting Gao <hgao17@illinois.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

SSL methods.

Motivated by this, in this paper, we conduct an extensive exploration of using synthetic data generated by diffusion models to improve the performance of existing SSL methods in a low-resource setting. In particular, we propose a Synthetic Speech Self-Superised Learning algorithm called DIFFS4L. DIFFS4L introduces a diffusion model, which learns from a given small pretraining dataset and then expands it into a much larger synthetic dataset. The new dataset contains synthetic speech utterances with different levels of variations from the original annotated speech, including speech with 1) novel prosody, 2) novel speakers, and 3) novel content compared with the original speech. Finally, the synthetic dataset is used to pretrain SSL models using existing algorithms. Since the diffusion model only has access to the information in the original real dataset, the entire process can be viewed as restructuring and recreating the information in the original pretraining dataset into a more digestible form for existing SSL methods.

Our experiments on DIFFS4L reveal many interesting findings. DIFFS4L can significantly improve the performance of existing SSL algorithms over models pretrained on the real data alone across low-resource and high-resource scenarios. In English ASR, for example, with 100 hours of real data, DIFFS4L can reduce the WER by 6.26 percentage points for HuBERT pretrained models, which is a 26.4% relative improvement. For low-resource language, we show that by further pretraining multi-lingual pretraining models, XLSR, on 100 hours of real speech plus 860 hours of DIFFS4L augmented speech in the low-resource language before fine-tuning it on ASR in that language, the WER can be improved by 1-3 percentage points compared to directly finetuning XLSR on low-resource ASR. Notably, the babbles generated by diffusion models, which are complete nonsense to humans, can account for a significant portion of the performance improvement, while babbles generated by other generative models, such as WaveNet (van den Oord et al., 2016), only deteriorate the performance. These findings suggest the information in pretraining datasets has been under-utilized, and diffusion models are very effective in capturing the information that has been overlooked by existing SSL training methods and other generative models.

2. Related Work

Data Augmentation with Synthetic Data Training neural networks with synthetic data to improve performance has been extensively studied in various computer vision tasks such as visual representation learning (Baradad Jurjo et al., 2021; Jahanian et al., 2021; Wu et al., 2022; Kataoka et al., 2022), image classification (Gan et al., 2021; Mikami et al., 2021), object detection (Peng et al., 2015; Prakash et al., 2019; Chattopadhyay et al., 2022), anomaly detection (Tsai & Wang, 2022), semantic segmentation (Ros et al., 2016; Wang et al., 2020), action recognition (De Souza et al., 2017; Varol et al., 2021), visual reasoning (Johnson et al., 2017), and embodied perception (Kolve et al., 2017; Savva et al., 2019; Xia et al., 2018). Recently this direction is also studied in NLP tasks such as machine translation (Downey et al., 2022) and language model pretraining (Yao et al., 2022) and finetuning (Steinert-Threlkeld et al., 2022).

Augmenting datasets with synthetic data has been shown effective in improving speech processing systems. One research direction modifies speech waveforms by adding random noise (Amodei et al., 2015), warping spectrogram, masking blocks of spectrograms in frequency and time domains (Park et al., 2019), modifying pitch and adding reverberation (Kharitonov et al., 2020), and disentangling speaker information from speech content (Qian et al., 2022).

Another line of research augments the dataset using speech data generated from speech synthesizers and reports improvement on speech translations (Zhao et al., 2022), fake audio detection (Li et al., 2022a), and speech recognition (Hayashi et al., 2018; Mimura et al., 2018; Li et al., 2018; Rossenbach et al., 2020; Violeta et al., 2022; Jin et al., 2022; Krug et al., 2022; Zevallos et al., 2022), etc. Zheng et al. (2021) use synthetic data to improve the recognition of out-of-vocabulary words in ASR systems. Zhao et al. (2022) generate synthetic training data by retrieving and stitching clips from a spoken vocabulary bank. Li et al. (2018) train a TACOTRON-2 (Shen et al., 2018) conditioned on Global Style Tokens (Wang et al., 2018) to generate speech with different speaking styles. Jin et al. (2022) use a GAN-based generator conditioned on dysarthric speech characteristics to generate synthetic speech for dysarthric ASR. Krug et al. (2022) generate articulatory speech for phoneme recognition. These works improve traditional task-specific speech systems by generating additional paired speech and text data while our work aims to improve general-purpose self-supervised speech representations without additional text data that benefits downstream ASR and other speech-related tasks.

Denoising Diffusion Probabilistic Models for Speech Denoising diffusion probabilistic models (DDPMs) have recently demonstrated great power in image synthesis (Ho et al., 2020; Dhariwal & Nichol, 2021) and image impainting (Lugmayr et al., 2022) tasks. Recently various DDPM-based vocoders and text-to-speech (TTS) synthesizers have been proposed (Chen et al., 2021a;b; Kong et al., 2020b; Lam et al., 2022; Huang et al., 2022a;b) and achieved high quality. WAVEGRAD (Chen et al., 2021a) and DIFFWAVE (Kong et al., 2020b) are two concurrent works that study the DDPM-based vocoder to synthesize audio waveform from spectrograms; WAVEGRAD uses a neural architecture inspired by GAN-TTS (Bińkowski et al., 2019) and DIF-

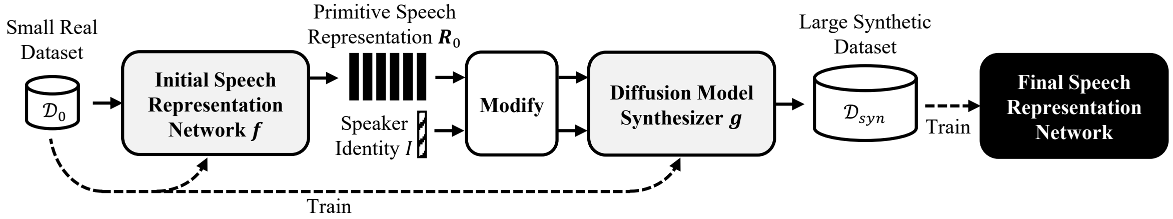

Small Real Dataset
$\mathcal{D}_0$
Initial Speech Representation Network $\boldsymbol{f}$
Primitive Speech Representation $\boldsymbol{R}_0$
Speaker Identity I
Modify
Diffusion Model Synthesizer $\boldsymbol{g}$
Large Synthetic Dataset
$\mathcal{D}_{syn}$
Train
Final Speech Representation Network
Train

Figure 1. The algorithm overview. Solid arrows represent the data flow that generates the synthetic dataset. Dashed arrows mark the dataset on which each network is trained.

fWave inspired by WaveNet. FastDiff (Huang et al., 2022a) and ProDiff (Huang et al., 2022b) are end-to-end TTS systems that use FastSpeech (Ren et al., 2020), a transformer-based TTS encoder, to extract text feature to condition the DDPM and adopts the noise scheduling algorithm proposed in BBDM (Lam et al., 2022) to shorten the sampling steps for fast speech synthesis. Our work utilizes a DDPM-based unit-to-speech synthesizer that is adapted from the DDPM synthesizer in Huang et al. (2022a) by expanding its conditioning on discrete speech units extracted from true speech to allow for fine-grained control over the synthesized content.

3. The DiffS4L Algorithm

In this section, we will formally introduce our proposed DiffS4L algorithm to improve the speech self-supervised learning problem under a low-resource scenario.

3.1. The Algorithm Overview

Assume that we have an unannotated corpus for pre-training an SSL model, denoted as $\mathcal{D}_l$ potentially with limited data size. Denote a speech utterance from this corpus as $\boldsymbol{X}$. For example, assume that the speech dataset contains an utterance saying ‘How are you?’, then $\boldsymbol{X}$ represents the speech waveform or spectrogram of this utterance, depending on what input feature the SSL model takes. The goal of DiffS4L is to generate multiple synthetic speech utterances based on each of the real speech utterances $\boldsymbol{X}$. The resulting synthetic speech forms a much larger dataset $\mathcal{D}_{syn}$, which is then used to pre-train the speech representation network. As shown in Figure 1, the algorithm consists of four steps.

Step 1: Use $\mathcal{D}_0$ to train an initial speech representation network $\boldsymbol{f}_0(\cdot)$, which can produce a primitive speech representation, denoted as $\boldsymbol{R}_0 = \boldsymbol{f}_0(\boldsymbol{X})$.

Step 2: Use $\mathcal{D}_0$ to train a diffusion-model-based speech synthesizer $\boldsymbol{g}(\cdot)$, which generates synthetic speech $\tilde{\boldsymbol{X}}$ conditional on the partially masked primitive speech representation $\boldsymbol{R}_0$ and speaker identity, denoted as $\boldsymbol{I}$, *i.e.*, $\tilde{\boldsymbol{X}} = \boldsymbol{g}(\boldsymbol{R}_0, \boldsymbol{I})$. The speaker identity $\boldsymbol{I}$ can be obtained in two ways. If $\mathcal{D}_0$ comes with speaker labels, then $\boldsymbol{I}$ can be a one-hot embedding of the speaker label, or a learned speaker embedding corresponding to the speaker label. If $\mathcal{D}_0$ does not have speaker labels, then I can be produced by feeding the seed utterance to a pre-trained speaker encoding network, such as GE2E (Wan et al., 2018), which would produce a speaker embedding.

Step 3: For each utterance $\boldsymbol{X}$ in $\mathcal{D}_0$, manipulate its speech representation $\boldsymbol{R}_0$ and speaker identity I, and then fed to the speech synthesizer to generate utterances with different levels of variations. Denote the resulting dataset as D_{syn}.

Step 4: Use D_{syn} to train a new speech representation network.

It is worth noting that the diffusion model only has access to the original pretraining dataset $\mathcal{D}_0$ during training and generation, so the synthetic dataset $\mathcal{D}_{syn}$ would contain no more information than $\mathcal{D}_0$, but may restructure and recreate it in a way that is more beneficial for SSL with existing methods. The following subsection will provide more details on steps 1-3, respectively.

3.2. Primitive Speech Representation

In our setting, the size of $\mathcal{D}_0$ is very small. We adopt the Wav2vec2.0 (Baevski et al., 2020) for our primitive speech representation learning because it has stable performance in low-resource scenarios. Note that the algorithm used to train the final speech representation network (step 4) need not be the same as the one for the primitive speech representation learning. After the Wav2vec2.0 is trained, we elicit the 5th-layer feature and quantize it into 500 classes using k-means, which becomes the primitive speech representation $\boldsymbol{R}_0$ for the subsequent steps. The resulting discrete speech representation is shown to preserve content information while obscuring speaker identity (Polyak et al., 2021) and pitch information (Choi et al., 2021). A discussion on choosing the number of clusters is provided in Appendix A.

3.3. Diffusion-Model-Based Speech Synthesizer

Diffusion models refer to a family of generative models that denoise from noise signals into clean signals through multiple denoising steps. In this work, we adopt the canon-

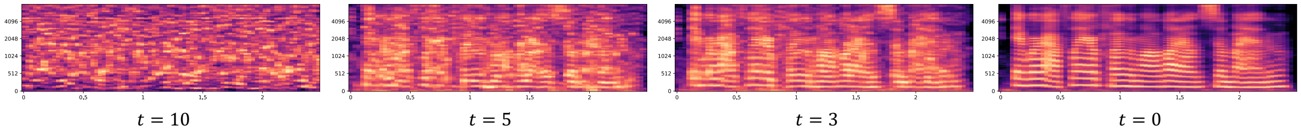


Figure 2. The intermediate denoising spectrograms of a 20-step DDPM denoising process. As t decreases to zero, the spectrograms transform from white noise to a speech spectrogram.

ical denoising diffusion probabilistic model (DDPM) (Ho et al., 2020) to generate a speech spectrogram. Specifically, DDPM introduces a set of intermediate variables forming a Markov process, denoted as $\boldsymbol{X}_{0:T}$, where $\boldsymbol{X}_0$ is the original speech spectrogram, and $\boldsymbol{X}_t$ is corrupted from $\boldsymbol{X}_{t-1}$ with Gaussian noise:

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

where β_t is a hyperparameters. It can be shown that with a proper β_t schedule, $\boldsymbol{X}_T$ is very close to standard Gaussian noise. To generate $\boldsymbol{X}_0$, we randomly sample $\boldsymbol{X}_T$ from the standard Gaussian distribution, and sequentially recover $\boldsymbol{X}_{T-1}$ through $\boldsymbol{X}_0$, as visualized in Figure 2, via the following denoising process:

$$p_\theta(\boldsymbol{X}_{t-1}|\boldsymbol{X}_t, \boldsymbol{C}) = \mathcal{N}(\boldsymbol{X}_{t-1}; \boldsymbol{\mu}_\theta(\boldsymbol{X}_t, t, \boldsymbol{C}), \sigma_t\boldsymbol{I}), \tag{2}$$

where $\boldsymbol{\mu}_\theta$ is produced by a (reparameterized) denoising network, and σ_t can be computed from β_t.

$\boldsymbol{C}$ represents the conditioning information for the denoising network. In this paper, we introduce two models with different levels of conditioning: a *fully-conditional model* and a *partially-conditional model*. For the fully-conditional model, the denoising network is conditioned upon the entire primitive speech representation $\boldsymbol{R}_0$, so that the diffusion model will generate speech that follows the content depicted in $\boldsymbol{R}_0$. For the partially-conditional model, the denoising network is still conditioned upon $\boldsymbol{R}_0$, but with a consecutive span of 80% of the frames masked out. In this case, the diffusion model will follow the content in $\boldsymbol{R}_0$ only where it is unmasked, and try to generate novel content that fits into the given context at the remaining frames. These two models are both crucial in generating synthetic data with different levels of variations.

Besides $\boldsymbol{R}_0$, both models are also conditional on speaker labels I, which can be either one-hot vectors or speaker embeddings produced by a pre-trained speaker embedding network, depending on whether D_0 comes with speaker labels. We will compare different conditioning settings in Section 4.

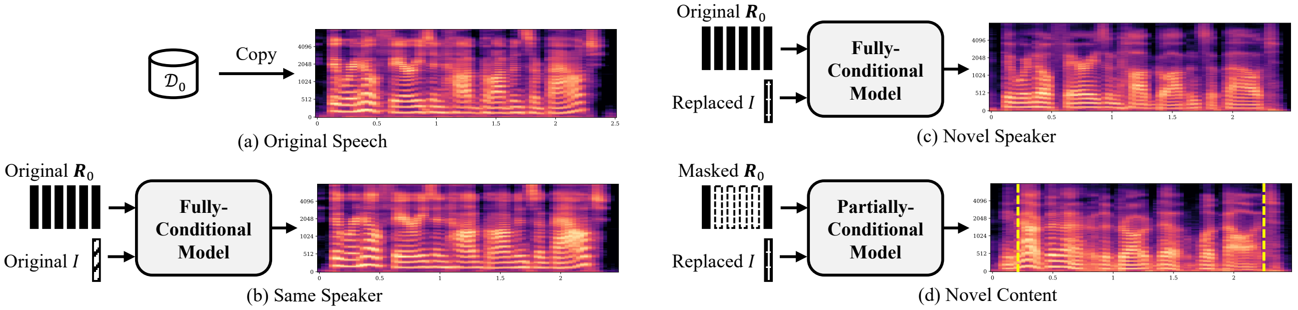
To convert the spectrograms into speech waveforms, we adopt a HifiGAN (Kong et al., 2020a), which is also trained only on the small dataset $\mathcal{D}_0$.

3.4. Synthetic Speech Generation

The synthetic speech generation uses the original speech dataset $\mathcal{D}_0$ as seeds. Specifically, we first draw a speech utterance from $\mathcal{D}_0$ as the seed speech, eliciting its primitive speech representation $\boldsymbol{R}_0$ and speaker identity I, and then generate a synthetic utterance by feeding a modified version of these conditioning variables to the diffusion model synthesizer. When designing the modification schemes for the conditioning variables, we primarily consider the tradeoff between novelty and naturalness – if the generated speech is identical to the original utterance, we can achieve maximum naturalness but introduces no new information to the dataset; if the generated speech is a complete babble, we can introduce maximum novelty but may significantly compromise naturalness. Therefore, we introduce the following four different levels of novelty, as shown in Figure 3:

- **Original Speech (O):** The seed speech is directly copied to the synthetic dataset without modification, as shown in Figure 3(a). No resynthesis is involved for this level.
- **Same Speaker (SS):** $\boldsymbol{R}_0$ and I are fed as is to the fully-conditional diffusion model. The resulting synthetic speech is almost the same as the seed speech. However, since $\boldsymbol{R}_0$ tends to obscure the pitch information, the synthetic speech will be in a different intonation, as shown in Figure 3(b).
- **Novel Speaker (NS):** $\boldsymbol{R}_0$ is still fed as is to the fully-conditional diffusion model, but I is replaced with a different speaker ID. As a result, the synthetic would still have the same content, but in a different voice and intonation, as shown in Figure 3(c).
- **Novel Content (NC):** We mask out a consecutive span of 80% frames in $\boldsymbol{R}_0$ and replace I before feeding them to the partially-conditional diffusion model. As shown in Figure 3(d), the synthetic speech is almost completely different from the seed speech in terms of content, speaker, and prosody, except for the content information in the 20% unmasked frames. The utterances are almost nonsensical babbles to human listeners. We are thus interested in seeing whether utterances at this high level of randomness could still contribute to SSL.

As we will show, all four levels of the speech are beneficial for the subsequent speech pretraining and thus should all



(a) Original Speech

(b) Same Speaker

(c) Novel Speaker

(d) Novel Content

Figure 3. An example of synthetic utterances at different levels of variations. The transcription of the original utterance is 'There were no ferries and hobgoblins about'. The yellow dashed lines on the spectrogram in (d) mark the boundaries of the masks on $\boldsymbol{R}_0$.

be included into $\mathcal{D}_{syn}$ with appropriate ratios. We have included additional spectrograms in Appendix B, as well as some generated audio files in the supplemental materials. We also perform 1) an ASR experiment, which verifies that the content in NC speech is different from the original speech, while that in NS is the same; and 2) an automatic speaker verification (ASV) experiment, which verifies that the speakers in NC and NS speech are different from the original speakers. The results are listed in Appendix C.

4. Experiments

In this section, we will present our experimental results on training different SSL models integrating DIFFS4L. Some additional experimental results are presented in the appendix.

4.1. Configurations

Pretraining Dataset For the experiments in English, the methods to be evaluated are pretrained on `Librispeech-960` dataset (Panayotov et al., 2015). We consider two settings, the *low-resource setting* and the *high-resource setting*. For the low-resource setting, the seed dataset $\mathcal{D}_0$ for training Steps 1 and 2 contains only 100 hours of real speech from the `train-clean-100` subset. The synthetic dataset $\mathcal{D}_{syn}$ contains 1) 100 hours of real speech; 2) 430 hours of SS/NS speech, which is generated by replacing the speaker ID with a uniformly randomly chosen one from all the speakers in $\mathcal{D}_0$ can be the same as the original speaker); and 3) 430 hours of NC speech. We deliberately make the total hours of speech in $\mathcal{D}_{syn}$ equal to 960 so that we can compare to the common setting with 960 hours of real speech. In the following, we will use $x + y + z$ notation to represent the hours of real speech (x), SS/NS speech (y), and NC speech (z) respectively. So the above $\mathcal{D}_{syn}$ composition is represented as $100 + 430 + 430$. For the high-resource setting, $\mathcal{D}_0$ contains all 960 hours of real speech from `Librispeech-960` and the dataset composition of $\mathcal{D}_{syn}$ is $960 + 960 + 480$. We explore other dataset compositions in Section 4.6.

Evaluation Tasks We consider two sets of tasks, *automatic speech recognition (ASR)* and *the SUPERB benchmark* (Yang et al., 2021). For ASR, we use the 'base_10h' configuration file in FAIRSEQ for WAV2VEC2.0 and HUBERT fine-tuning on a 10-hour limited supervision dataset. We follow the same finetuning procedure as in Baevski et al. (2020) and Hsu et al. (2021) where we add a linear projection layer on top and finetune with the CTC loss. For SUPERB, which is a collection of speech-processing tasks, we evaluate our models on KS (keyword spotting), IC (intent classification), SID (speaker identification), ER (emotion recognition), Qbe (query by example spoken term detection), SF (slot filling), ASV (automatic speaker verification) and SD (speaker diarization), We did not include the ASR and PR (phoneme recognition), because they overlap with the first task.

Evaluation Models For both the high-resource (960h real) and low-resource (100h real) settings, we compare the following four models:

- WAV2VEC2/HUBERT-DIFFS4L: WAV2VEC2.0 (Baevski et al., 2020) and HUBERT (Hsu et al., 2021) pretrained on the synthetic dataset produced by the proposed DiffS4L procedure;
- WAV2VEC2/HUBERT-REAL: WAV2VEC2.0 and HUBERT pretrained on real speech only.

In addition, for the low-resource setting, we add three models for better comparison:

- WAV2VEC2/HUBERT-ONEHOT: In DIFFS4L Models, we use the pretrained GE2E speaker embedding (Wan et al., 2018). To study whether this would leak information of more real speech data, we replace it with one-hot speaker embedding.
- WAV2VEC2-AUG: Wav2vec2.0 pretrained on 100-hour real data augmented by adding reverberation, Gaussian noise, and modifying the pitch of the speech samples (Sriram et al., 2022).

Table 1. Main results on (a) English automatic speech recognition and (b) SUPERB benchmark. The bolded results show the best performance among all but the topline models.

	English ASR		SUPERB								
			KS	IC	SID	ER	QbE	SF		ASV	SD
Task/Metric	CER↓	WER↓	ACC↑	ACC↑	ACC↑	ACC↑	MTWV↑	F1↑	CER↓	EER↓	DER↓
High-Resource Setting (960-hour real speech)											
Wav2vec2-Real	3.18	10.49	96.23	92.35	66.20	60.55	0.0233	87.64	25.37	6.67	6.65
HuBERT-Real	3.03	10.30	96.30	98.26	**66.27**	60.74	0.0736	88.53	25.20	5.80	6.30
Wav2vec2-DiffS4L	2.98	9.93	96.17	94.73	65.79	61.29	0.0630	88.50	24.71	6.60	6.63
HuBERT-DiffS4L	**2.95**	**9.87**	**96.47**	**98.50**	64.36	**61.40**	**0.0766**	**88.93**	**24.03**	**5.78**	**6.26**
Low-Resource Setting (100-hour real speech)											
Wav2vec2-Real	7.37	23.48	91.92	88.64	47.68	58.99	0.0311	81.31	37.06	8.78	8.45
HuBERT-Real	7.43	23.71	91.82	78.43	**57.53**	61.84	0.0419	78.87	40.69	8.91	8.53
Wav2vec2-Aug	6.92	22.06	92.18	92.83	48.65	58.34	0.0377	81.99	36.39	8.37	8.84
Wav2vec2-DiffS4L	**5.19**	16.67	93.57	91.01	45.41	59.86	0.0331	83.13	33.60	8.02	7.14
Wav2vec2-OneHot	**5.19**	**16.65**	93.23	91.41	48.94	61.64	0.0364	83.00	34.64	8.14	7.28
HuBERT-DiffS4L	5.33	17.45	**94.68**	**95.94**	44.22	62.02	**0.0469**	**84.61**	32.68	**7.42**	**7.09**
HuBERT-OneHot	5.36	17.47	94.26	95.89	44.25	**62.60**	0.0445	83.98	**32.33**	7.64	7.44

Implementation Details The entire training pipeline is constructed based on two existing code repositories: Fairseq (Ott et al., 2019) and ProDiff (Huang et al., 2022b). The code and configuration files are uploaded to GitHub[1]. We follow the same procedure as in Baevski et al. (2020) and Hsu et al. (2021) to pretrain all the Wav2vec2.0 and HuBERT models using Fairseq. All the Wav2vec2.0/HuBERT models are trained for 400k updates with a learning rate of 5×10^{-4}. We use the base model of Wav2vec2.0 and HuBERT, which contain 12 Transformer layers and 95M parameters. For HuBERT, we adopt two rounds of training; the first round uses a K-Means teacher of 500 clusters on the 80-bin mel-spectrogram and the second round uses a K-Means teacher of 500 clusters on the HuBERT feature from the first round.

The speech synthesizer is based on the code of the ProDiff-TTS model implemented in ProDiff, which consists of a Fastspeech2 encoder and a DDPM. We remove the Energy Predictor, Pitch Predictor in the Fastspeech2 encoder, and replace the Duration Predictor that aligns the text with mel-spectrogram with an upsampling network that resamples the HuBERT units from 50Hz to 62.5Hz to match the length of mel-spectrogram. The DDPM models a 20-time-step forward and reverse Gaussian diffusion process on the mel-spectrogram, conditioned on the FastSpeech2 encoder outputs. Both the fully- and partially-conditional diffusion models are trained for 200k iterations. To convert the mel-spectrogram into a speech waveform, we apply the HifiGAN vocoder (Kong et al., 2020a), which is trained on the same real dataset $\mathcal{D}_0$ for 1M iterations.

More implementation details are included in Appendix D.

4.2. Main Results

Table 1 reports the character error rate (CER) and word error rate (WER) of the ASR task and the performance on the SUPERB tasks. There are four key observations. First, in both low-resource and high-resource scenarios, pretraining on DiffS4L-synthetic data consistently improve the performance of almost all downstream tasks, compared to pretraining on the real speech portion only. Second, performance improvement is particularly significant in low-resource scenarios. This confirms that DiffS4L more thoroughly utilizes the information in the same real speech dataset that is otherwise overlooked by SSL models. Third, Wav2vec2.0-based systems perform slightly better in ASR tasks, whereas HuBERT-based systems do better in SUPERB tasks. Additional results in Appendix E and F further verify that the performance advantage is consistent with and without language models, across different sizes of the finetuning dataset, and with different choices of diffusion models. Appendix G presents the results of DiffS4L trained with just 20 hours of real speech, which further demonstrate the advantage of DiffS4L in low-resource settings.

Second, models utilizing one-hot speaker embeddings demonstrate similar performance to those using GE2E embeddings, confirming that the performance advantage of DiffS4L does *not* come from the leakage of the GE2E pretraining dataset. To be fair, DiffS4L does need speaker information or labels whereas the baseline pretraining methods do not. It is worth emphasizing, though, that DiffS4L only converts the speaker identity to the seen speakers in the original real dataset. It does not introduce new speakers.

Third, DiffS4L systems consistently outperform Wav2vec2-Aug, suggesting that DiffS4L better capture

[1] https://github.com/Hertin/DiffS4L

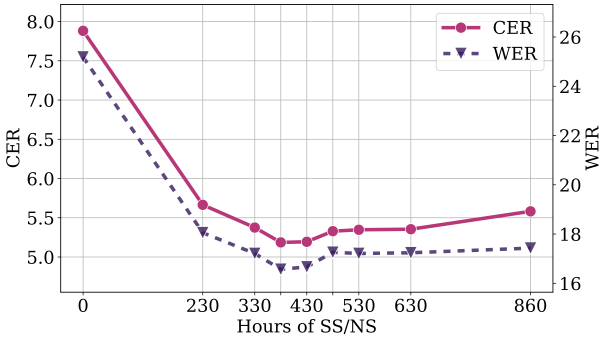


Figure 4. CER/WER across data compositions by varying the ratios of SS/NS and NC speech, ranging from 100+860+0 to 100+0+860.

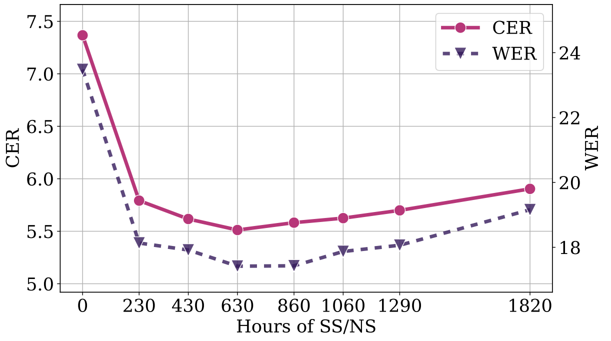


Figure 5. Performance over different synthetic dataset sizes, 100+x+0, where x ranges from 0 to 1820.

the information and variations in the real speech than signal-processing-based augmentation. Since Wav2vec2-Aug is designed for Wav2vec2.0 and does not have a HuBERT counterpart implementation, we self-implemented a HuBERT version denoted as HuBERT-Aug and test it on English ASR task under low-resource setting. The resulting CER/WER is $7.012/22.716$, which is worse than that of HuBERT-DiffS4L .

Finally, comparing wav2vec2-Real in 960-hour setting and wav2vec-DiffS4L in 100-hour setting which augments the dataset to 960 hours, we would like to note that the DiffS4L models trained on the synthetic dataset still underperform those trained on the same amount of the real speech data.

4.3. Extension to Other Languages

To test whether the performance improvement of DiffS4L can generalize to other languages, we select all the seven non-English languages from the `Mulingual LibriSpeech` (`MLS`) dataset (Pratap et al., 2020) and six languages from the `Commonvoice` dataset (Ardila et al., 2019). The languages in the `Commonvoice` dataset are chosen based on the criterion that they have just over 100 hours of validated data in the dataset. For each language in `MLS`, we sample 100 hours from training split for pretaining and use the limited supervision subset for finetuning. Both cases use the provided dev and test split for validation and testing. For each language in `CommonVoice`, we create a 100-hour split for pretraining, and a 10-hour split for finetuning. The provided dev and test split are used for validation and testing, respectively. We only evaluate the Wav2vec2.0 systems due to the substantial time cost for pretraining and due to our observation that the relative improvements in both Wav2vec2.0 and HuBERT are similar. Also, since most of these languages do not have 960 hours of data, we cannot compute the topline results, so we show only the baseline and DiffS4L models.

Table 2 demonstrates a consistent performance advantage of DiffS4L across all the languages. In particular, DiffS4L can reduce the CER by an average of 2.6 percentage points, and WER by an average of 8.3 percentage points, which is a significant improvement for ASR. Notice that these languages are from different language families and each has very unique phonetic, lexical, and syntactic structures, so these results show that the diffusion models can successfully capture various structures in all these languages. Additional results in Appendix H show that the performance gain is consistent across different dataset partitions and compositions.

4.4. Extension to Large Multi-lingual Pretraining

So far, all our experiments are performed on models pre-trained in at most 960 hours of English only. We would like to find out whether DiffS4L is still useful if the pre-trained model sees even more data in many languages. To this end, we select a multilingual pre-trained model XLSR-128 (Babu et al., 2021), which is pretrained on 128 languages. We then use the six low-resource languages from `Commonvoice` for finetuning. For each language, we derive two other pre-trained models, one by further pre-training the XLSR models on 100 hours of the low-resource data, and the other by further pre-training on 100 hours of low-resource data plus the DiffS4L-augmented data. All three pre-trained models are then finetuned on the ASR task with 10 hours of labeled data, and the results are reported in Table 4, which shows a clear advantage of DiffS4L despite the abundance of pre-training data.

Appendix I presents additional multilingual experiments using XLSR-53 (Conneau et al., 2020) pretrained on 53 languages and Appendix J presents cross-lingual pretaining experiments using Wav2vec2 models pretrained on 60k-hour English corpora and the results of both yield a similar conclusion. Appendix K provides the training schemes in total numbers of hours of pretraining and finetuning resources

Table 2. ASR results (CER/WER) on selected languages from `MLS` and `CommonVoice`. The languages are (from left to right, top to bottom) English, German, Spanish, French, Italian, Dutch, Polish, Portuguese, Bashki, Central Kurdish, Welsh, Meadow Mari, Swahili, and Tamil.

Languages	EN	DE	ES	FR	IT	NL	PL
Wav2vec-100R	7.4/23.5	8.3/30.4	7.1/27.2	16.2/45.5	8.3/35.1	17.8/50.9	11.4/44.2
Wav2vec-DiffS4L	**5.2/16.8**	**6.4/23.3**	**4.5/16.7**	**11.9/34.8**	**6.2/27.2**	**14.7/44.8**	**7.1/31.0**
Languages	PO	BA	CKB	CY	MHR	SW	TA
Wav2vec-100R	13.8/45.8	10.2/43.8	7.2/39.0	20.6/62.1	10.7/45.4	8.8/31.5	9.2/47.2
Wav2vec-DiffS4L	**8.9/37.1**	**8.9/37.1**	**6.7/29.7**	**16.7/52.3**	**9.4/37.5**	**7.0/25.9**	**7.5/41.0**

Table 3. ASR Performance on improving multilingual XLSR-128 models.

Model	BA	CKB	CY	MHR	SW	TA
XLSR-128	6.69/31.28	4.62/24.46	11.05/41.11	7.51/33.31	6.04/24.49	6.94/41.26
XLSR-128-100R	6.45/30.28	4.59/25.00	10.81/40.68	7.09/31.66	5.81/24.48	6.81/41.09
XLSR-128-DiffS4L	**6.32/29.77**	**4.29/21.69**	**10.44/39.31**	**6.91/30.10**	**5.73/24.20**	**6.79/40.86**

Table 4. ASR Performance of WavLM, DiffS4L and WavLM-DiffS4L pretrained on 100 and 960 hours of real speech.

	100H Real	960H Real
WavLM	5.88/18.38	2.98/9.92
HuBERT-DiffS4L	5.19/16.65	2.95/9.87
WavLM-DiffS4L	**4.31/13.69**	**2.89/9.62**

for multilingual experiments.

4.5. Combination with Other Augmentation Techniques

Since DiffS4L is a generic approach that can be applied to many SSL procedures, it can be combined with approaches that consider other speech variations, such as noises and recording conditions. We introduce an experiment that combines DiffS4L with WavLM (Chen et al., 2021c), an SSL approach considering additive interference. The combined model is denoted as WavLM-DiffS4L. The results are in Table 4. From the table, there are two observations. First, DiffS4L can outperform WavLM in both low- and high-resource scenarios, and the performance gap is greater as the resource gets more scarce. Second, when the two approaches are combined, we achieve an even more significant performance gain, which indicates that DiffS4L can indeed complement the conventional additive data augmentation and further improve the performance.

4.6. Dataset Composition

In the low-resource setting, the dataset composition is fixed to 100+430+430 (recall the three numbers are the hours of real speech, SS/NS speech, and NC speech respectively). To better understand the contribution of each component, we perform an ablation study where we change the dataset composition. To keep our computation tractable, we only perform experiments on Wave2vec2.0 and on the English ASR tasks in all the remaining ablation studies.

In our first experiment, we fix the total hours of the dataset to 960 and fix hours of real data to 100, but we vary the ratio of the SS/NS and NC from 100+860+0 to 100+0+860. The results are shown in Figure 4. There are two important observations. First, the performance curve exhibits a U-shape, with the lowest CER and WER achieved when both SS/NS and NC are of comparable amounts. This indicates that both the recombination of speaker information and the innovation of content plays a crucial role in improving the performance of SSL models. In particular, note that NC data is essentially nonsensical babbles reflecting the limited knowledge of phone transitions learned by the diffusion models from the small real dataset, and that one of the purposes of SSL models is also to learn the phone transition structures. The fact that the nonsensical babble can still help the SSL performance implies that the existing SSL algorithms cannot effectively utilize all the phone transition information in the original real dataset.

Our second observation of Figure 4 is that comparing the two extreme cases, the performance without the SS/NS data (the left endpoint) is worse than that without the NC data (the right endpoint). Recall that SS/NS data are generated conditional on the true content information and therefore are of high quality, whereas NC data generally sound messier and noisier. This observation may be ascribed to the quality differences in the synthetic data.

Now that we have verified the contribution of synthesizing novel content, we will investigate the effect of synthesizing novel speaker combinations in the next experiment. In par-

Table 5. English ASR performance of Wav2vec pretrained on DiffS4L-generated data versus that on WaveNet-generated data.

Composition	DiffS4L	WaveNet
100+860+0	5.58/17.43	6.07/18.95
100+430+430	5.19/16.67	6.71/21.90
100+0+860	7.88/24.91	12.57/39.02

Table 6. Performance over different augmentation schemes.

Model	English ASR
Wav2vec-DiffS4L	5.19/16.67
Wav2vec-SS	6.91/21.69
Wav2vec-NoReal	18.26/52.79

ticular, we start with the standard dataset composition, *i.e.* 100+430+430, but instead, we do not replace the speaker in any of the synthesis types; hence there is no longer NS data and the NC data has reduced speaker variations. The result, shown in Table 6 (Wav2vec-SS), shows a marked performance degradation (1.7 percentage points in CER and 5.0 percentage points in WER) compared to the standard dataset composition, which verifies that the novel speaker combination is crucial to the performance.

Finally, to test the contribution of including the original dataset, we remove the real data and expand the synthetic data proportionally to 960 hours, *i.e.* 0+480+480. The result, as shown in Table 6 (Wav2vec-NoReal), shows an even larger performance degradation. In fact, we find that without the real data, the SSL training is hard to converge. This shows that including the real data is essential for successful SSL training with synthetic data.

4.7. Dataset Size

Since we have verified that synthetic data improve SSL training, a natural follow-up question is whether the more synthetic data the better. To answer this question, we fix the real data to 100 hours and NC data to 0 hours but vary the hours of SS/NS data, *i.e.*, 100+x+0, with x ranging from 0 to 1820. Figure 5 shows the corresponding Wav2vec results on English ASR. As shown, the performance does not always improve as the amount of synthetic data increases. When synthetic data is small, increasing synthetic data can drastically improve performance. However, as synthetic data continues to increase, the performance gradually saturates and then starts to degrade, with the optimal performance achieved at around 630 hours. Combining the previous results, we can conclude that although adding synthetic data can inject new knowledge and variations, adding too much can dilute the contribution of the real data, which have been shown essential for the training, and hence will negatively impact the performance.

4.8. Comparison with WaveNet

To generate the NC babbles, we randomly select 3 seconds of real speech as the prompt and use the partially conditional WaveNet to generate the subsequent waveforms conditional on I. We then pretrain Wav2vec2.0 using three synthetic data compositions, 100+860+0, 100+430+430, and 100+0+860, and compare the English ASR results with the diffusion model counterparts, as shown in Table 5. As shown, both WaveNet results are worse than the corresponding diffusion model ones, which suggests that WaveNet-generated speech may have a lower overall quality. Appendix L compares the distributional distance of DiffS4L- and WaveNet-generated speech to the source real speech and provides additional evidence for this hypothesis. More importantly, unlike the case of diffusion models, where an adequate amount of NC babble improves performance, WaveNet-generated NC babbles are always detrimental to performance. This comparison underlines the unique advantage of the diffusion model in generating babble that better captures the inherent structure in speech.

5. Conclusion

In this study, we examined SSL from an information efficiency perspective and found that performance can be greatly improved by utilizing the information present in the pretraining dataset, particularly in low-resource settings. We discovered that synthetic data is an effective way to extract information and enhance SSL performance. Specifically, diffusion models were found to be particularly capable of capturing complex structures in speech that traditional pretraining methods cannot; thus even synthetic babbles contain valuable information for SSL training. DiffS4L opens the door to a new approach to speech SSL. One limitation of DiffS4L is that it is a time-consuming process, as it involves training of multiple networks sequentially. As a next step, we plan to investigate more efficient methods of information sharing between diffusion models and SSL models to reduce the need for synthetic data generation and prolonged pretraining.

Impact Statement

This paper presents work whose goal is to improve the performance of speech processing systems, including speech recognition, speaker identification, *etc.*, particularly for low-resource languages. Therefore, this work is expected to benefit low-resource languages by extending the AI processing capabilities to these languages. There is no significant negative impact associated with this work.

References

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Chen, J., Chrzanowski, M., Coates, A., Diamos, G. F., Elsen, E., Engel, J., Fan, L. J., Fougner, C., Hannun, A. Y., Jun, B., Han, T. X., LeGresley, P., Li, X., Lin, L., Narang, S., Ng, A., Ozair, S., Prenger, R. J., Qian, S., Raiman, J., Satheesh, S., Seetapun, D., Sengupta, S., Sriram, A., Wang, C.-J., Wang, Y., Wang, Z., Xiao, B., Xie, Y., Yogatama, D., Zhan, J., and Zhu, Z. Deep speech 2 : End-to-end speech recognition in english and mandarin. In *International Conference on Machine Learning*, 2015.

Ardila, R., Branson, M., Davis, K., Henretty, M., Kohler, M., Meyer, J., Morais, R., Saunders, L., Tyers, F. M., and Weber, G. Common voice: A massively-multilingual speech corpus. *arXiv preprint arXiv:1912.06670*, 2019.

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. Xls-r: Self-supervised cross-lingual speech representation learning at scale. *ArXiv*, abs/2111.09296, 2021. URL https://api.semanticscholar.org/CorpusID:244270531.

Baevski, A., Zhou, H., rahman Mohamed, A., and Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. *ArXiv*, abs/2006.11477, 2020.

Baradad Jurjo, M., Wulff, J., Wang, T., Isola, P., and Torralba, A. Learning to see by looking at noise. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 2556–2569. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/14f2ebeab937ca128186e7ba876faef9-Paper.pdf.

Bińkowski, M., Donahue, J., Dieleman, S., Clark, A., Elsen, E., Casagrande, N., Cobo, L. C., and Simonyan, K. High fidelity speech synthesis with adversarial networks. In *International Conference on Learning Representations*, 2019.

Chattopadhyay, P., Sarangmath, K., Vijaykumar, V., and Hoffman, J. Pasta: Proportional amplitude spectrum training augmentation for syn-to-real domain generalization. *arXiv preprint arXiv:2212.00979*, 2022.

Chen, N., Zhang, Y., Zen, H., Weiss, R. J., Norouzi, M., and Chan, W. Wavegrad: Estimating gradients for waveform generation. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=NsMLjcFaO8O.

Chen, N., Zhang, Y., Zen, H., Weiss, R. J., Norouzi, M., Dehak, N., and Chan, W. Wavegrad 2: Iterative refinement for text-to-speech synthesis. In *Interspeech*, 2021b.

Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., Li, J., Kanda, N., Yoshioka, T., Xiao, X., Wu, J., Zhou, L., Ren, S., Qian, Y., Qian, Y., Zeng, M., and Wei, F. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16:1505–1518, 2021c. URL https://api.semanticscholar.org/CorpusID:239885872.

Choi, H.-S., Lee, J., Kim, W. S., Lee, J. H., Heo, H., and Lee, K. Neural analysis and synthesis: Reconstructing speech from self-supervised representations. *ArXiv*, abs/2110.14513, 2021. URL https://api.semanticscholar.org/CorpusID:239998228.

Conneau, A., Baevski, A., Collobert, R., rahman Mohamed, A., and Auli, M. Unsupervised cross-lingual representation learning for speech recognition. In *Interspeech*, 2020. URL https://api.semanticscholar.org/CorpusID:220055837.

De Souza, C. R., Gaidon, A., Cabon, Y., and Lopez, A. M. Procedural generation of videos to train deep action recognition networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2594–2604. IEEE Computer Society, 2017.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf.

Downey, C., Liu, L., Zhou, X., and Steinert-Threlkeld, S. Learning to translate by learning to communicate. *ArXiv*, abs/2207.07025, 2022.

Dunbar, E., Karadayi, J., Bernard, M., Cao, X.-N., Algayres, R., Ondel, L., Besacier, L., Sakti, S., and Dupoux, E. The zero resource speech challenge 2020: Discovering discrete subword and word units. In *Interspeech*, 2020.

Gan, C., Schwartz, J., Alter, S., Schrimpf, M., Traer, J., De Freitas, J., Kubilius, J., Bhandwaldar, A., Haber, N., Sano, M., et al. Threedworld: A platform for interactive multi-modal physical simulation. In *Annual Conference on Neural Information Processing Systems*, 2021.

Hayashi, T., Watanabe, S., Zhang, Y., Toda, T., Hori, T., Astudillo, R., and Takeda, K. Back-translation-style data augmentation for end-to-end asr. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pp. 426–433. IEEE, 2018.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., and Mohamed, A. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021.

Huang, R., Lam, M. W. Y., Wang, J., Su, D., Yu, D., Ren, Y., and Zhao, Z. Fastdiff: A fast conditional diffusion model for high-quality speech synthesis. In *International Joint Conference on Artificial Intelligence*, 2022a.

Huang, R., Zhao, Z., Liu, H., Liu, J., Cui, C., and Ren, Y. Prodiff: Progressive fast diffusion model for high-quality text-to-speech. *Proceedings of the 30th ACM International Conference on Multimedia*, 2022b.

Jahanian, A., Puig, X., Tian, Y., and Isola, P. Generative models as a data source for multiview representation learning. In *International Conference on Learning Representations*, 2021.

Jin, Z., Xie, X., Geng, M., Wang, T., Hu, S., Deng, J., Li, G., and Liu, X. Adversarial data augmentation using vae-gan for disordered speech recognition. *arXiv preprint arXiv:2211.01646*, 2022.

Johnson, J., Hariharan, B., Van Der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., and Girshick, R. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2901–2910, 2017.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. In *Proc. NeurIPS*, 2022.

Kataoka, H., Hayamizu, R., Yamada, R., Nakashima, K., Takashima, S., Zhang, X., Martinez-Noriega, E. J., Inoue, N., and Yokota, R. Replacing labeled real-image datasets with auto-generated contours. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21232–21241, 2022.

Kharitonov, E., Rivière, M., Synnaeve, G., Wolf, L., Mazar'e, P.-E., Douze, M., and Dupoux, E. Data augmenting contrastive learning of speech representations in the time domain. *2021 IEEE Spoken Language Technology Workshop (SLT)*, pp. 215–222, 2020.

Kolve, E., Mottaghi, R., Han, W., VanderBilt, E., Weihs, L., Herrasti, A., Gordon, D., Zhu, Y., Gupta, A., and Farhadi, A. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv*, 2017.

Kong, J., Kim, J., and Bae, J. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in Neural Information Processing Systems*, 33:17022–17033, 2020a.

Kong, Z., Ping, W., Huang, J., Zhao, K., and Catanzaro, B. Diffwave: A versatile diffusion model for audio synthesis. *ArXiv*, abs/2009.09761, 2020b.

Krug, P. K., Birkholz, P., Gerazov, B., van Niekerk, D. R., Xu, A., and Xu, Y. Articulatory Synthesis for Data Augmentation in Phoneme Recognition. In *Proc. Interspeech 2022*, pp. 1228–1232, 2022. doi: 10.21437/Interspeech.2022-10874.

Lakhotia, K., Kharitonov, E., Hsu, W.-N., Adi, Y., Polyak, A., Bolte, B., Nguyen, T.-A., Copet, J., Baevski, A., Mohamed, A., et al. On generative spoken language modeling from raw audio. *Transactions of the Association for Computational Linguistics*, 9:1336–1354, 2021.

Lam, M. W. Y., Wang, J., Su, D., and Yu, D. BDDM: bilateral denoising diffusion models for fast and high-quality speech synthesis. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=L7wzpQttNO.

Li, J., Gadde, R. T., Ginsburg, B., and Lavrukhin, V. Training neural speech recognition systems with synthetic speech augmentation. *ArXiv*, abs/1811.00707, 2018.

Li, K., Li, S., Lu, X., Akagi, M., Liu, M., Zhang, L., Zeng, C., Wang, L., Dang, J., and Unoki, M. Data augmentation using mcadams-coefficient-based speaker anonymization for fake audio detection. In *Proc. INTERSPEECH*, 2022a.

Li, X. L., Thickstun, J., Gulrajani, I., Liang, P., and Hashimoto, T. B. Diffusion-lm improves controllable text generation. *arXiv preprint arXiv:2205.14217*, 2022b.

Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., and Van Gool, L. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11461–11471, 2022.

Mikami, H., Fukumizu, K., Murai, S., Suzuki, S., Kikuchi, Y., Suzuki, T., Maeda, S.-i., and Hayashi, K. A scaling law for synthetic-to-real transfer: How much is your pre-training effective? *arXiv preprint arXiv:2108.11018*, 2021.

Mimura, M., Ueno, S., Inaguma, H., Sakai, S., and Kawahara, T. Leveraging sequence-to-sequence speech synthesis for enhancing acoustic-to-word speech recognition. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pp. 477–484. IEEE, 2018.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

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

Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., and Le, Q. V. Specaugment: A simple data augmentation method for automatic speech recognition. *ArXiv*, abs/1904.08779, 2019.

Peng, X., Sun, B., Ali, K., and Saenko, K. Learning deep object detectors from 3d models. In *Proceedings of the IEEE international conference on computer vision*, pp. 1278–1286, 2015.

Polyak, A., Adi, Y., Copet, J., Kharitonov, E., Lakhotia, K., Hsu, W.-N., Mohamed, A., and Dupoux, E. Speech resynthesis from discrete disentangled self-supervised representations. *arXiv preprint arXiv:2104.00355*, 2021.

Prakash, A., Boochoon, S., Brophy, M., Acuna, D., Cameracci, E., State, G., Shapira, O., and Birchfield, S. Structured domain randomization: Bridging the reality gap by context-aware synthetic data. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 7249–7255. IEEE, 2019.

Pratap, V., Xu, Q., Sriram, A., Synnaeve, G., and Collobert, R. Mls: A large-scale multilingual dataset for speech research. *ArXiv*, abs/2012.03411, 2020.

Qian, K., Zhang, Y., Gao, H., Ni, J., Lai, C.-I., Cox, D., Hasegawa-Johnson, M. A., and Chang, S. Contentvec: An improved self-supervised speech representation by disentangling speakers. In *International Conference on Machine Learning*, 2022.

Ren, Y., Hu, C., Tan, X., Qin, T., Zhao, S., Zhao, Z., and Liu, T.-Y. Fastspeech 2: Fast and high-quality end-to-end text to speech. In *International Conference on Learning Representations*, 2020.

Ros, G., Sellart, L., Materzynska, J., Vazquez, D., and Lopez, A. M. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3234–3243, 2016.

Rossenbach, N., Zeyer, A., Schlüter, R., and Ney, H. Generating synthetic audio data for attention-based speech recognition systems. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7069–7073. IEEE, 2020.

Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9339–9347, 2019.

Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerrv-Ryan, R., et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 4779–4783. IEEE, 2018.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

Sriram, A., Auli, M., and Baevski, A. Wav2vec-aug: Improved self-supervised training with limited data. *ArXiv*, abs/2206.13654, 2022.

Steinert-Threlkeld, S., Zhou, X., Liu, Z., and Downey, C. Emergent communication fine-tuning (ec-ft) for pretrained language models. In *Emergent Communication Workshop at ICLR 2022*, 2022.

Su, K., Qian, K., Shlizerman, E., Torralba, A., and Gan, C. Physics-driven diffusion models for impact sound synthesis from videos. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9749–9759, 2023. URL https://api.semanticscholar.org/CorpusID:257805229.

Tsai, M.-C. and Wang, S.-D. Self-supervised image anomaly detection and localization with synthetic anomalies. *Available at SSRN 4264542*, 2022.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. In *9th ISCA Speech Synthesis Workshop*, pp. 125–125, 2016.

Varol, G., Laptev, I., Schmid, C., and Zisserman, A. Synthetic humans for action recognition from unseen viewpoints. *Int. J. Comput. Vision*, 129(7): 2264–2287, jul 2021. ISSN 0920-5691. doi: 10.1007/s11263-021-01467-7. URL https://doi.org/10.1007/s11263-021-01467-7.

Violeta, L. P., Ma, D., Huang, W.-C., and Toda, T. Intermediate fine-tuning using imperfect synthetic speech for improving electrolaryngeal speech recognition. *arXiv preprint arXiv:2211.01079*, 2022.

Wan, L., Wang, Q., Papir, A., and Moreno, I. L. Generalized end-to-end loss for speaker verification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4879–4883. IEEE, 2018.

Wang, Y., Stanton, D., Zhang, Y., Ryan, R.-S., Battenberg, E., Shor, J., Xiao, Y., Jia, Y., Ren, F., and Saurous, R. A. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In *International Conference on Machine Learning*, pp. 5180–5189. PMLR, 2018.

Wang, Z., Yu, M., Wei, Y., Feris, R., Xiong, J., Hwu, W.-m., Huang, T. S., and Shi, H. Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12635–12644, 2020.

Wu, Y., Wang, Z., Zeng, D., Shi, Y., and Hu, J. Synthetic data can also teach: Synthesizing effective data for unsupervised visual representation learning, 2022. URL https://arxiv.org/abs/2202.06464.

Xia, F., Zamir, A. R., He, Z., Sax, A., Malik, J., and Savarese, S. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9068–9079, 2018.

Yang, S., Chi, P.-H., Chuang, Y.-S., Lai, C.-I. J., Lakhotia, K., Lin, Y. Y., Liu, A. T., Shi, J., Chang, X., Lin, G.-T., Huang, T.-H., Tseng, W.-C., tik Lee, K., Liu, D.-R., Huang, Z., Dong, S., Li, S.-W., Watanabe, S., Mohamed, A., and yi Lee, H. SUPERB: Speech Processing Universal PERformance Benchmark. In *Proc. Interspeech 2021*, pp. 1194–1198, 2021. doi: 10.21437/Interspeech.2021-1775.

Yao, S., Yu, M., Zhang, Y., Narasimhan, K. R., Tenenbaum, J. B., and Gan, C. Linking emergent and natural languages via corpus transfer. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=49A1Y6tRhaq.

Zevallos, R., Bel, N., Cámbara, G., Farrús, M., and Luque, J. Data augmentation for low-resource quechua asr improvement. *arXiv preprint arXiv:2207.06872*, 2022.

Zhao, J., Haffar, G., and Shareghi, E. Generating synthetic speech from spokenvocab for speech translation. *arXiv preprint arXiv:2210.08174*, 2022.

Zheng, X., Liu, Y., Gunceler, D., and Willett, D. Using synthetic audio to improve the recognition of out-of-vocabulary words in end-to-end asr systems. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5674–5678. IEEE, 2021.

A. Number of Clusters for Speech Units

Table 7. ASR performance across the number of clusters for speech units.

#Clusters	100	200	300	500
CER/WER	32.9/51.0	27.9/43.7	28.3/42.2	**23.0/34.1**

In the preliminary experiment, we train a WaveNet conditioned on the speech units of 100, 200, 300 and 500 clusters, to synthesize the English speech. We then measure the quality of synthesized speech using a Wav2vec2-CTC model. The results are show in Table 7. The 500-cluster speech units yield the best ASR performance, indicating the 500-cluster units better capture the speech information.

In addition, we perform the ABX test from Zero Resource Speech Challenges 2020 (Dunbar et al., 2020; Lakhotia et al., 2021) on the 500-cluster units and get the ABX within/across speaker score of 7.87/10.29, which is not too far away from the 200-cluster 'hubert_l6' units reported in Lakhotia et al. (2021), which has an ABX score of 5.99/7.31.

B. More Example Spectrograms of the Synthetic Speech

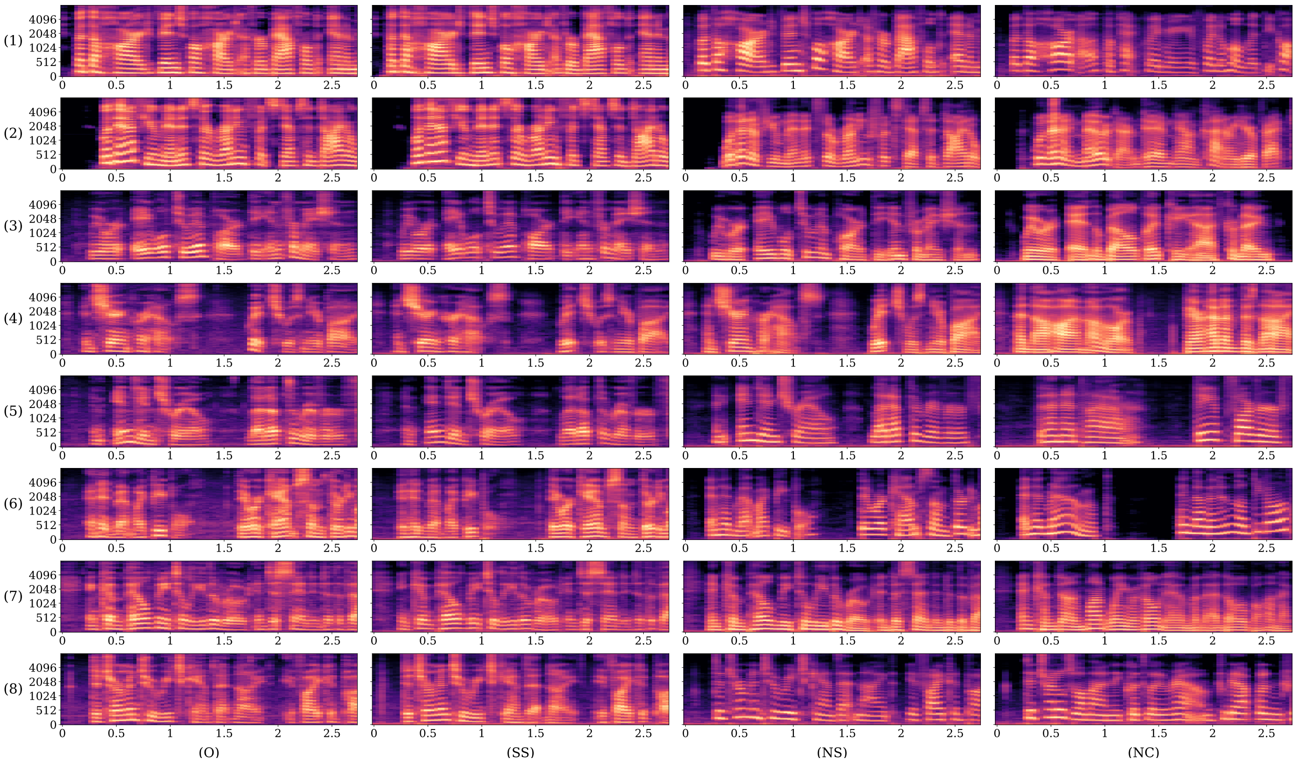


Figure 6. Eight examples of synthetic utterances at different levels of variation. Each row is one utterance and each column is one variation, original (O), same speaker (SS), new speaker (NS) and new content (NC) from left to right.

Figure 6 shows the spectrograms of some generated speech at different novelty levels. Each row represents the generated audios from one seed speech. Each column represents a novelty level. As can be observed, as the novelty level progresses from SS to NC, the generated speech becomes increasingly dissimilar to the original speech. In particular, the NC speech barely preserves any structures in the original speech and is very close to babbles. We have also included some generated audios in the supplemental materials. We encourage the readers to listen to these audios to get a more direct sense of the quality and stochasticity of the synthetic speech.

C. Verification of Novel Content and Speaker Information in NC and NS Speech

For the speaker information, previous works (Polyak et al., 2021) have established that by having a discrete $\boldsymbol{R}_0$, the resulting synthetic speech will follow the speaker control by $\boldsymbol{I}$. If $\boldsymbol{I}$ is the same as the original speech, so would the voice in the synthetic speech; if $\boldsymbol{I}$ is a different speaker, the voice would be in a different speaker as well. For the content information, $\boldsymbol{R}_0$ provides the content information of the source speech if $\boldsymbol{R}_0$ is masked, the synthesizer would have no access to the content information in the source speech, and thus the synthetic content would be random.

We perform ASR and ASV experiments to test whether the speaker and content are different from the original speech. Specifically, to test whether the content is the same as the source speech, we perform ASR on the synthetic speech against the transcription of the source speech. If the WER is high, that means the content in the synthetic speech is different from the source speech.

Table 8. ASR error rates for NC and NS speech.

ASR	NC	NS
CER/WER	66.5/84.9	5.1/12.7

Table 8 shows the CER/WER results. As can be seen, NC has a very high CER/WER, indicating that its content is significantly different from the original speech. On the other hand, NS, which only changes the speaker embedding while retaining the conditioning on speech representation $\boldsymbol{R}_0$, has a low CER/WER, indicating that it has almost the same content as the original speech.

Next, to test if the generated speech follows the control of the speaker embedding, we perform a speaker classification experiment on the generating speech, using the speaker whose speaker embedding is conditioned upon to generate the speech as the ground-truth labels. If the speaker classification accuracy is high, that means the voice of synthetic speech follows the control of the speaker embedding.

Table 9. ASV Accuracy for NC and NS speech.

ASV	NC	NS
ACCURACY	88.7	89.3

Table 9 shows the speaker classification accuracy results. As can be seen, both accuracies are very high, confirming that the voice follows the control of the speaker embedding. Consequently, if we feed the same speaker embedding as the original speech, the synthesized speech will be in the same voice. If we feed a different speaker embedding, the synthesized speech will be in a different voice.

D. Additional Implementation Details

The entire training pipeline is constructed based on two existing code repositories: FAIRSEQ[2] (Ott et al., 2019), PRODIFF[3] (Huang et al., 2022b).

Pretraining SSL models We use FAIRSEQ (Ott et al., 2019) to pretrain all the speech SSL models. In particular, We use the same hyperparameter as specified in the 'wav2vec2_base_librispeech' and 'hubert_base_librispeech.yaml' configuration file in FAIRSEQ to pretrain WAV2VEC2 and HUBERT respectively. The training of WAV2VEC2 models requires 64 Tesla V100-SXM2-32GB GPUs and that of HUBERT models requires 32 GPUs. The pretraining dataset for both models is the 100-hour seed dataset, $\mathcal{D}_0$ for the initial speech representation network and is the augmented dataset $\mathcal{D}_{syn}$ for the final speech representation network as described in Sec 4.6 Dataset Composition. The SSL models are trained for 400k updates with a learning rate of 5×10^{-4}. Each batch contains 1.4M audio samples. The checkpoint with the best validation loss is selected for downstream tasks.

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

[3] https://github.com/Rongjiehuang/ProDiff

Finetuning SSL models We use the 'base_10h' configuration file in FAIRSEQ for WAV2VEC2.0 and HUBERT fine-tuning on a 10-hour limited supervision dataset. We follow the same finetuning procedure as in Baevski et al. (2020) and Hsu et al. (2021) where we add a linear projection layer on top and finetune with the CTC loss. The model is trained for 40k updates on two V100-SXM2-32GB GPUs with each batch containing 3.2M audio samples and a learning rate of 5×10^{-5}. The checkpoint with the best CER on the validation set is selected for further evaluation.

Training Diffusion Speech Synthesizer The speech synthesizer consists of a FASTSPEECH2 encoder and a 20-step DDPM model. The FASTSPEECH2 encoder contains 4 Transformer encoder layers each with 4 heads. Using the initial speech representation network, we extracted the speech units from $\mathcal{D}_0$ and substitute them for the text inputs. We replace the Duration Predictor with an upsampling network consisting of a transposed convolution with a kernel size of 9, a stride of 5, and a padding of 2, followed by a convolution layer with a kernel size of 8, a stride of 5, and a padding of 2, that resamples the HUBERT units from 50Hz to 62.5Hz to match the length of 80-bin mel-spectrogram. The FASTSPEECH2 encoder encodes the speech units into hidden embeddings, which are combined with the broadcasted speaker embeddings to condition the training and inference of the DDPM model. The speech synthesizer is trained for 200k iterations using one V100-SXM2-32GB GPU with a batch size of 64 and a learning rate of 1. The synthesizer is optimized for the weighted sum of $\mathcal{L}_1$ reconstruction loss and structural similarity index (SSIM) loss (Huang et al., 2022b) with the weight being 0.5 for each loss. We use adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^{-9}$ and inverse square root scheduler with 2000 warmup updates.

We use a HIFIGAN vocoder[4] (Kong et al., 2020a) to convert mel-spectrogram to waveform. The vocoder is trained on the same seed dataset $\mathcal{D}_0$ for 1M iteration using four V100-SXM2-32GB GPU.

E. Full ASR results on Librispeech English

We provide the full ASR results on Librispeech English dataset in Table 10, including the CER/WER evaluated on `dev-clean`, `dev-other`, `test-clean` and `test-other` subset of `LibriSpeech-960` dataset. The experiments are labeled as 'EN-X-Y', where 'X' denotes the number of hours of untranscribed real speech for pretraining and 'Y' denotes the number of hours of transcribed real speech for finetuning. We use the 10-hour limited supervision set from LibriLight for Y=10 and the 'train-clean-100' subset from LibriSpeech for Y=100. We additionally provide the results of WAV2VEC-AUG in EN-100-10 for comparison with WAV2VEC2.0 and WAV2VEC-DIFFS4L.

Language Models It has been widely known that introducing language models will rectify the ASR results, and thus tend to obscure the performance gap between different ASR algorithms. We therefore would like to see whether DIFFS4L is still helpful in the presence of a language model. To this end, we introduce a 4-gram language model to the English ASR task. As can be observed from the rows marked with '4-GRAM' in Table 13, not only does the performance advantage persist when the 4-gram language model is introduced, but also the gap is largely the same as that without the language model. These results verify the robustness of DIFFS4L regardless of the use of the language models.

Size of Finetuning Dataset To study the impact of the size of the finetuning dataset on performance, we finetune SSL models on the `train-clean-100` subset of `LibriSpeech-960` dataset and compare the results to those obtained from the 10-hour supervision set of LirbriLight. We observe that in the 100-hour low-resource setting (EN-100-100) WAV2VEC-DIFFS4L systems still have a relatively large gain compared to the baseline WAV2VEC-REAL. In the high-resource setting (EN-960-100) where there is a sufficient amount of labeled speech, the gain diminishes.

F. Additional Experiment using EDM

We experiment with another diffusion model EDM (Karras et al., 2022) instead of DDPM to generate synthetic data. The dataset configurations and evaluation procedures are exactly the same as described in Section 4 except that the diffusion process is changed. The architecture of the speech synthesizer remains the same while the diffusion training and inference pipeline follow the official implementation of EDM[5]. We keep the default hyperparameters of the original EDM implementation except for the data standard deviation, which is calculated from our training data. The diffusion model is trained for 300k iterations on eight V100-SXM2-32GB GPU with a batch size of 32 per GPU and a learning rate of

[4] `https://github.com/jik876/hifi-gan`

[5] `https://github.com/NVlabs/edm`

Table 10. Full ASR results on Librispeech English dataset, including the CER/WER of Wav2vec2.0 model pretrained on 100/960 hours and fine-tuned on 10/100 hours. Results of Wav2vec-Aug are included for EN-100-10 experiment for a comparison with Wav2vec2.0 and Wav2vec-DiffS4L.

		Dev-clean		Dev-other		Test-clean		Test-other	
Model	LM	CER	WER	CER	WER	CER	WER	CER	WER
EN-100-10									
Wav2vec-Real	None	7.13	22.17	15.06	37.57	7.17	22.62	15.74	39.24
	4-gram	9.79	19.91	18.45	36.05	9.80	20.20	19.40	37.71
Wav2vec-Aug	None	6.92	22.06	14.83	37.17	6.95	22.48	15.64	39.01
	4-gram	9.24	19.36	18.28	36.02	9.47	19.64	19.22	37.35
Wav2vec-SS/NS	None	5.58	17.43	12.84	32.58	5.59	17.78	13.31	33.99
	4-gram	7.84	15.41	15.92	31.26	7.91	15.74	16.47	32.54
Wav2vec-DiffS4L	None	5.19	16.67	11.85	30.03	5.31	17.39	12.17	31.27
	4-gram	7.70	15.00	14.99	28.73	7.57	15.01	15.40	29.91
EN-960-10									
Wav2vec-Real	None	3.18	10.49	6.69	18.03	3.07	10.39	6.64	18.53
	4-gram	5.17	9.14	9.35	16.98	5.1	9.06	9.31	17.43
Wav2vec-DiffS4L	None	2.98	9.93	6.31	17.19	3.03	10.14	6.27	17.55
	4-gram	4.97	8.42	8.80	15.91	5.08	8.76	8.92	16.36
EN-100-100									
Wav2vec-Real	None	4.44	13.95	14.43	34.47	4.56	14.60	15.50	36.90
	4-gram	6.42	12.25	17.62	33.44	6.67	12.87	18.81	35.75
Wav2vec-DiffS4L	None	2.93	9.56	10.51	25.94	3.03	9.98	10.74	26.77
	4-gram	4.81	8.22	13.32	24.76	5.02	8.69	13.75	25.91
EN-960-100									
Wav2vec-Real	None	1.65	5.60	5.03	13.62	1.65	5.74	4.76	13.4
	4-gram	3.33	4.60	7.33	12.77	3.43	5.07	6.98	12.49
Wav2vec-DiffS4L	None	1.61	5.58	4.80	12.91	1.63	5.66	4.63	12.93
	4-gram	3.33	4.60	7.19	12.26	3.41	5.01	7.02	12.16

5×10^{-4}. We use adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$ and inverse square root scheduler with 32000 warmup updates. The sampling process for data generation uses 18 steps.

The results of Wav2vec-DiffS4LEDM and HuBERT-DiffS4LEDM trained using synthetic data are shown in Table 11. We get similar ASR and SUPERB performances as using DDPM, suggesting that the diffusion models consistently generate babble that better captures the inherent speech structure.

G. Additional 20-Hour Pretraining Experiments

We conduct experiments on English ASR with the pre-training data further reduced to 20 hours. We used DiffS4L to augment the data to 200 hours. The results are shown in Table 12. As can be observed, DiffS4L does significantly improve the performance with only 20 hours of pre-training data available. Also, the performance advantage tends to be more significant when there is less data.

H. Full ASR results on MLS and CommonVoice

We provide the full ASR results on `MLS` and `Commonvoice` dataset in Table 13. To better examine the robustness of DiffS4L under different settings, we perform some additional experiments on the `MLS` ASR task (Wav2vec-SS/NS in Table 13).

Additional Test Set The `MLS` and `Commonvoice` datasets come with a dev set and a test set for each language, both of which can be utilized as test sets to evaluate the ASR performance. In the main paper, we reported the dev set performance.

Table 11. Results of EDM on (a) English automatic speech recognition and (b) SUPERB benchmark. The Wav2vec-960R and HuBERT-960R are topline models.

	(a) English		(b) SUPERB								
	ASR		KS	IC	SID	ER	QbE	SF		ASV	SD
Task/Metric	CER↓	WER↓	ACC↑	ACC↑	ACC↑	ACC↑	MTWV↑	F1↑	CER↓	EER↓	DER↓
Wav2vec-DiffS4LEDM	5.20	16.81	92.99	93.94	47.28	61.24	0.0327	81.66	35.15	7.88	7.30
HuBERT-DiffS4LEDM	5.21	17.03	94.55	95.94	43.78	61.80	0.0501	82.68	34.32	7.42	7.26

Table 12. ASR Performance of Wav2vec2-Real and Wav2vec2-DiffS4L pretrained on 20, 100 and 960 hours of real speech.

Model	20-Hour Real	100-Hour Real	960-Hour Real
Wav2vec2-Real	22.794/62.525	7.37/23.48	3.18/10.49
Wav2vec2-DiffS4L	**13.425/41.067**	**5.19/16.67**	**2.98/9.93**

Here, we include the results on the test set to show the statistical significance of the performance advantage of DiffS4L. As shown in the columns under 'Test' in Table 13, DiffS4L maintains a consistent advantage over the baseline, which is trained on the 100 hours of real speech alone, and the performance gaps are similar to that in the dev set. These results confirm the significance of the benefit induced by DiffS4L-generated data.

Dataset Compositions In the main paper, we only examined the effect of varying dataset compositions for English ASR. In this section, we extend the experiment to different languages by introducing the Wav2vec-SS/NS, we were trained on the 100+860+0 dataset composition, *i.e.*, without NC speech. As can be observed from the rows marked with 'Wav2vec-SS/NS' in Table 13, the performance always deteriorates when NC speech is removed. This is a rather impressive finding because different languages have different structures, some of which are easier to capture than others. The fact that NC speech is able to improve the performance for *all* these languages indicates that the diffusion model can successfully capture all the different types of structural information.

I. Additional Multi-lingual Pretraining Experiments

Following the same setting as in Section 4.4, we conduct additional experiments on another multilingual pre-trained model XLSR-53 (Conneau et al., 2020), which is pretrained on 53 languages. The XLSR-53 model is finetuned on two low-resource languages: Bashki and Central Kurdish. The results are shown in Table 14, which exhibit the same trend as the XLSR-128 results.

J. Scaling Up to 10k-100k Hours of Pretraining Data

We would like to investigate, in the cross-lingual setting, where there is abundant high-resource language data but only a small amount of low-resource language data (both annotated and unannotated), whether DiffS4L can still improve the performance when the amount of high-resource language is large. To study this, we conducted an experiment where we further scaled up the data for the high-resource language. We adopt the LV-60k model in Baevski et al. (2020), which is pretrained on 60k hours of English. We choose the same two languages as in Section I. For each low-resource language, we train three models: 1) Wav2vec2L-LV60k, derived by directly finetuning the LV-60k on the low-resource language ASR with 10 hours of transcribed data; 2) Wav2vec2L-LV60k-100R, derived by further pre-training the LV-60k with 100 hours of low-resource language before fine-tuning on 10 hours of ASR data in the low-resource language; 3) Wav2vec2L-LV60k–DiffS4L, derived by further pretraining the LV-60k on the 960-hour augmented data for the low-resource language using DiffS4L , before fine-tuning on 10 hours of ASR data in the low-resource language. Table 15 shows the performance of these three models, and DiffS4L augmented models continue to demonstrate superior performance.

Table 13. ASR performance of Wav2vec pretrained on DiffS4L-generate data on LibriSpeech, MLS and CommonVoice dataset

			Dev		Test	
Lang	Model	LM	CER	WER	CER	WER
EN	Wav2vec-100R	None	7.13	22.17	7.17	22.62
	Wav2vec-SS/NS	None	5.58	17.43	5.59	17.78
	Wav2vec-DiffS4L	None	5.19	16.67	5.31	17.39
DE	Wav2vec-100R	None	8.33	30.44	9.93	33.83
	Wav2vec-SS/NS	None	6.67	24.48	7.96	27.45
	Wav2vec-DiffS4L	None	6.37	23.27	7.55	26.11
ES	Wav2vec-100R	None	7.10	27.22	7.08	27.33
	Wav2vec-SS/NS	None	6.20	23.46	6.29	23.44
	Wav2vec-DiffS4L	None	4.49	16.65	4.48	16.83
FR	Wav2vec-100R	None	16.16	45.50	14.49	41.84
	Wav2vec-SS/NS	None	12.12	35.80	10.61	31.65
	Wav2vec-DiffS4L	None	11.91	34.77	10.61	31.13
IT	Wav2vec-100R	None	8.33	35.08	7.80	33.62
	Wav2vec-SS/NS	None	8.09	34.39	7.35	32.10
	Wav2vec-DiffS4L	None	6.24	27.22	5.54	24.43
NL	Wav2vec-100R	None	17.83	50.92	11.55	39.09
	Wav2vec-SS/NS	None	15.31	46.78	9.49	33.85
	Wav2vec-DiffS4L	None	14.69	44.83	9.37	33.25
PL	Wav2vec-100R	None	11.42	44.22	9.92	43.20
	Wav2vec-SS/NS	None	7.80	32.75	7.67	35.72
	Wav2vec-DiffS4L	None	7.14	30.95	7.56	34.90
PO	Wav2vec-100R	None	13.83	45.75	16.48	50.92
	Wav2vec-SS/NS	None	10.37	35.17	12.45	40.16
	Wav2vec-DiffS4L	None	9.88	34.60	11.96	39.78
BA	Wav2vec-100R	None	10.16	43.81	11.82	47.99
	Wav2vec-DiffS4L	None	8.90	37.07	9.12	37.09
CKB	Wav2vec-100R	None	7.23	39.04	7.75	40.86
	Wav2vec-DiffS4L	None	6.71	29.70	6.48	26.65
CY	Wav2vec-100R	None	20.58	62.05	17.25	49.37
	Wav2vec-DiffS4L	None	16.70	52.28	12.48	37.45
MHR	Wav2vec-100R	None	10.74	45.41	12.91	49.43
	Wav2vec-DiffS4L	None	9.44	37.52	10.04	39.19
SW	Wav2vec-100R	None	8.80	31.54	8.83	29.71
	Wav2vec-DiffS4L	None	6.99	25.92	7.55	24.55
TA	Wav2vec-100R	None	9.16	47.20	11.19	54.07
	Wav2vec-DiffS4L	None	7.51	40.98	8.58	45.17

Table 14. ASR Performance on improving multilingual XLSR-53 models.

Model	ba	ckb
XLSR-53	6.98/32.54	5.29/26.99
XLSR-53-100R	6.94/31.91	5.05/26.41
XLSR-53-DiffS4L	**6.61/30.11**	**4.71/24.55**

K. Training Schemes of Multilingual Experiments

We provide the total number of hours of pretraining and finetuning resources for multilingual experiments (Section 4.3, 4.4, I and J) in Table 16.

Table 15. ASR Performance on improving LV-60K models in cross-lingual setting.

LANG	WAV2VEC2L-LV60K	WAV2VEC2L-LV60K-100R	WAV2VEC2L-LV60K–DIFFS4L
BA	9.21/41.01	7.43/33.97	**7.38/33.67**
CKB	6.85/35.57	5.81/30.60	**5.55/26.94**

Table 16. Total number of hours of pretraining and finetuning resources for multilingual experiments.

MODEL	PRETRAINING	FURTHER PRETRAINING	FINETUNING
WAV2VEC-100R (SEC. 4.3)	100	N/A	10
WAV2VEC-DIFFS4L (SEC. 4.3)	100+860	N/A	10
XLSR-128 (SEC. 4.4)	436K	N/A	10
XLSR-128-100R (SEC. 4.4)	436K	100	10
XLSR-128-DIFFS4L (SEC. 4.4)	436K	100+860	10
XLSR-53 (SEC. I)	56K	N/A	10
XLSR-53-100R (SEC. I)	56K	100	10
XLSR-53-DIFFS4L (SEC. I)	56K	100+860	10
WAV2VEC2L-LV60K (SEC. J)	60K	N/A	10
WAV2VEC2L-LV60K-100R (SEC. J)	60K	100	10
WAV2VEC2L-LV60K-DIFFS4L (SEC. J)	60K	100+860	10

L. Further Analysis of Generated Speech

Table 17. FID of synthetic speech with different levels of variation generated by DIFFS4L and WAVENET.

	DIFFS4L-SS	DIFFS4L-NS	DIFFS4L-NC	WAVENET-NS	WAVENET-NC
FID↓	0.12	0.13	0.34	0.65	0.69

Table 18. PESQ and STOI of synthetic speech generated by DIFFS4L and WAVENET.

	DIFFS4L	WAVENET
PESQ↑	0.66	0.34
STOI↑	1.09	1.06

Distributional analysis of the synthetic speech To investigate the benefit brought by DIFFS4L, we conduct experiments investigating how the distribution changes as we increase the level of variations from real speech to random babble. Specifically, we follow Su et al. (2023) and compute the FID score between the 100 hours of real speech and: 1) DIFFS4L -SS, which only contains the reconstruction with the same speakers (but with novel prosody), 2) DIFFS4L-NS, which contains reconstruction with new speakers, 3) DIFFS4L-NC, which contains synthetic speech with novel content (babble), 4) WAVENET-NS, WAVENET generation with novel speakers, and 5) WAVENET-NC, WAVENET-generated babble. To compute the FID, we replaced the feature extractor in Su et al. (2023) with the HUBERT feature with mean-pooling, because the original feature extractor was tailored for sound classification. Table 17 shows the results.

There are two observations. First, as the level of variations increases, the FID increases, implying an increasing distributional discrepancy with the 100-hour real speech. In particular, the increase brought by introducing novel content is most significant. Second, even when the novel content is introduced, the FID score for DIFFS4L is still much lower than the WAVENET-generated audio, implying that DIFFS4L can generate speech that can follow the ground truth speech distribution much better.

Quality of Synthetic Speech Also, to better compare DIFFS4L and WAVENET synthesis, we conduct experiments to compare the audio quality of DIFFS4L- and WAVENET-generated speech in terms of two objective metrics, PESQ and STOI, and the results are shown in Table 18. These results confirm that DIFFS4L has a better speech quality than WAVENET.

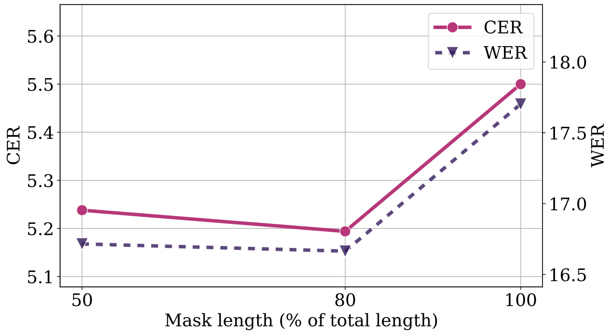


Figure 7. Performance over different masking ratios when synthesizing NC speech.

M. Masking Length

Recall that the NC data is generated by conditioning on $\boldsymbol{R}_0$ with 80% frames masked out, as shown in Figure 3(d). We would like to investigate whether the masking length has an impact on the performance. We thus retrain two partially-conditional diffusion models, one with 50% masking length and the other with 100% (which becomes totally unconditional). We then generate two synthetic datasets, whose compositions are both 100+430+430, but whose NC data are generated with 50% and 100% masking length, respectively. The corresponding WAVE2VEC English ASR results are shown in Figure 7. As shown, there are only slight differences in the performance, with the optimal achieved by 80% masking length. We conjecture that two factors influence the performance when changing the mask length. One is the amount of novel content, which increases as masking length increases; the other is the quality of generated speech, which decreases as masking length increases. Therefore, pushing the mask length to both extremes negatively impact the performance.