OWSM-CTC: An Open Encoder-Only Speech Foundation Model for Speech Recognition, Translation, and Language Identification

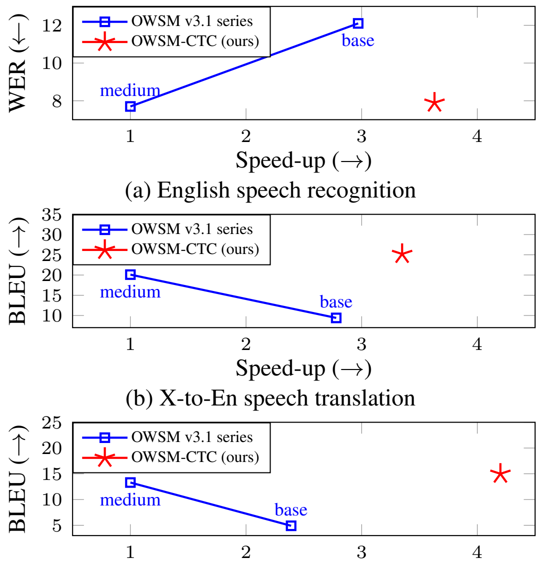
Anonymous ACL submission

Abstract

There has been an increasing interest in large speech models that can perform multiple speech processing tasks in a single model. Such models usually adopt the encoder-decoder or decoder-only architecture due to their popularity and good performance in many domains. However, autoregressive models can be slower during inference compared to non-autoregressive models and also have potential risks of hallucination. Though prior studies observed promising results of non-autoregressive models for certain tasks at small scales, it remains unclear if they can be scaled to speech-to-text generation in diverse languages and tasks. Inspired by the Open Whisper-style Speech Model (OWSM) project, we propose OWSM-CTC, a novel encoder-only speech foundation model based on Connectionist Temporal Classification (CTC). It is trained on 180k hours of public audio data for multilingual automatic speech recognition (ASR), speech translation (ST), and language identification (LID). Compared to encoder-decoder OWSM, our OWSM-CTC achieves competitive results on ASR and up to 25% relative improvement on ST, while it is more robust and 3 to 4 times faster for inference. OWSM-CTC also improves the long-form ASR result with 20x speed-up. We will publicly release our codebase, pre-trained model, and training logs to promote open science in speech foundation models.

1 Introduction

The great success of large language models (LLMs) (OpenAI, 2023; Touvron et al., 2023; Anil et al., 2023b) has sparked a growing interest in developing foundation models in various modalities. Recent studies have explored different approaches towards multilingual and multi-tasking speech foundation models (Radford et al., 2023; Zhang et al., 2023; Pratap et al., 2023; Rubenstein et al., 2023; Barrault et al., 2023; Peng et al., 2023e). OpenAI's Whisper (Radford et al., 2023)



(a) English speech recognition

(b) X-to-En speech translation

(c) En-to-X speech translation

Figure 1: Performance vs. speed for encoder-decoder OWSM v3.1 and our encoder-only OWSM-CTC.

is a series of Transformer encoder-decoder models trained on 680k hours of proprietary labelled audio. Whisper achieves strong results in multilingual automatic speech recognition (ASR), any-to-English speech translation (ST), and spoken language identification (LID). Although it shows the effectiveness of large-scale (weakly) supervised pre-training, the full development pipeline including training data details is not publicly accessible. Recent work releases Open Whisper-style Speech Models (OWSM) (Peng et al., 2023e, 2024) with the aim of reproducing Whisper-style training using public data and open-source toolkits. However, Whisper and OWSM adopt the encoder-decoder architecture, which generates text tokens given speech in an autoregressive manner. They might hallucinate during inference, and the speed can be slow. Other models with a decoder-only architecture like AudioPaLM (Rubenstein et al., 2023) and VioLA (Wang et al., 2023b) would suffer from the same issues due to autoregressive decoding.

Another type of work like Google's USM (Zhang et al., 2023) and Meta's MMS (Pratap et al., 2023) uses non-autoregressive models with Connectionist Temporal Classification (CTC) (Graves et al., 2006), but these CTC-based models are designed for ASR only. Prior studies have also achieved promising results of CTC models for ST only, but they mainly focus on specific language pairs at much smaller scales (Inaguma et al., 2021; Chuang et al., 2021; Xu et al., 2023). Some of them employ additional decoders (Inaguma et al., 2021; Yan et al., 2023) or cross-attention layers (Xu et al., 2023), making the model more complicated.

A natural question now arises: *Can we build a non-autoregressive encoder-only model for speech-to-text generation in diverse languages and multiple tasks like Whisper/OWSM?* This research problem has become increasingly important in the era of LLMs, because large-scale pre-trained speech encoders can serve as an adaptor between the speech and text modalities (Gong et al., 2023; Wang et al., 2023a), providing a promising avenue towards general-purpose multi-modal foundation models (Anil et al., 2023a).

In this work, we propose OWSM-CTC, a novel encoder-only speech foundation model based on multi-task self-conditioned CTC (Nozaki and Komatsu, 2021) to imitate OWSM's multilingual ASR, any-to-any ST, and LID functionalities. Following previous encoder-decoder OWSM v3.1 models (Peng et al., 2024), we train a 1B OWSM-CTC model using 180k hours of public data covering 151 languages. Extensive evaluations show that our OWSM-CTC exhibits strong performance and efficiency. Compared to the 1B OWSM v3.1 medium model, OWSM-CTC achieves comparable performance for ASR and superior performance for various ST directions (up to 25% relative improvement) while being more robust and showing 3 to 4 times inference speed-up. For long-form ASR, OWSM-CTC improves the WER and is 20 times faster due to the batched parallel decoding. OWSM-CTC also outperforms the other models on LID. We will publicly release our codebase, pre-trained model weights, and training logs to facilitate the development of large speech models.

2 Related Work

2.1 Speech foundation models

Attention-based encoder-decoder. OpenAI's Whisper (Radford et al., 2023) adopts the standard Transformer encoder-decoder architecture (Vaswani et al., 2017) and scales the training data to 680k hours of proprietary labelled audio.[1] Despite its strong performance on ASR, ST, and LID, the full development pipeline including training data details and training codebase is not publicly available. A recent project OWSM aims to reproduce Whisper-style training using public data and open-source toolkits to promote transparency and open science in this field (Peng et al., 2023e). The latest OWSM v3.1 models (Peng et al., 2024) employ E-Branchformer (Kim et al., 2023) as the encoder and Transformer as the decoder, which is trained with a joint ASR CTC loss (Kim et al., 2017). Although OWSM has promising results using public corpora, it still follows the encoder-decoder architecture, which can be slow and unstable at inference time.

Decoder-only. Several studies employ decoder-only models for speech-to-text tasks. AudioPaLM (Rubenstein et al., 2023) extends the textual PaLM-2 (Anil et al., 2023b) to support speech understanding and generation tasks including ASR and ST. DOTA (Gupta et al., 2024) is a decoder-only Transformer model trained on 93k hours of public English ASR data, but it does not support other languages or ST. Decoder-only models face the same slowness and robustness issues as encoder-decoder due to autoregressive decoding.

CTC or Transducer. Another line of research proposes to utilize CTC (Graves et al., 2006) or Transducer (Graves, 2012) for ASR. Google's USM (Zhang et al., 2023) provides generic ASR models, which are first pre-trained on 12M hours of unlabelled audio and then fine-tuned on proprietary labelled data with CTC or Transducer. Meta's MMS (Pratap et al., 2023) pre-trains a wav2vec 2.0 model (Baevski et al., 2020) on massively multilingual data and then fine-tunes it with CTC on labelled ASR data covering over 1k languages. These models employ CTC only for ASR. In our OWSM-CTC, we propose a single CTC-based encoder-only model for ASR, ST and LID. Our supported tasks are more similar to Whisper-style models.

2.2 Efficient speech models

Model compression. Various algorithms have been utilized to compress speech models, including knowledge distillation (Chang et al., 2022; Lee et al., 2022; Peng et al., 2023d; Gandhi et al., 2023),

[1] Their latest large-v3 version uses 1M hours of labelled audio and 4M hours of pseudo-labelled audio.

pruning (Lai et al., 2021; Peng et al., 2023a), quantization (Yeh et al., 2023; Ding et al., 2023), and dynamic module execution (Yoon et al., 2022; Peng et al., 2023c; Strimel et al., 2023). These methods are typically applied to pre-trained models and are thus orthogonal to this work. In the future, we will apply compression to further improve efficiency.

Efficient architectures. Better network architectures can also improve efficiency, including attention with linear complexity (Beltagy et al., 2020; Wang et al., 2020b; Tay et al., 2023) and sequence length reduction (Burchi and Vielzeuf, 2021; Kim et al., 2022; Nawrot et al., 2023; Rekesh et al., 2023). In this work, we do not modify the attention, but we use larger downsampling in CNN to reduce the sequence length. More details are in Appendix A.2 and B.1.

2.3 CTC-based speech models

Non-autoregressive models have a faster inference speed than their autoregressive counterparts due to parallel decoding. They have been utilized in machine translation (Gu et al., 2018; Ghazvininejad et al., 2019; Xiao et al., 2023), ASR (Chen et al., 2019; Higuchi et al., 2020; Ng et al., 2021; Chi et al., 2021; Lee and Watanabe, 2021; Nozaki and Komatsu, 2021), and ST (Inaguma et al., 2021; Chuang et al., 2021; Xu et al., 2023).

CTC is originally proposed to label sequences without explicit segmentation (Graves et al., 2006). CTC-based ASR models learn a monotonic alignment between speech features and text tokens. With parallel greedy decoding, they are much faster than autoregressive models. However, the accuracy of CTC is generally inferior due to the conditional independence assumption between output tokens. To address this issue, Intermediate CTC (InterCTC) (Lee and Watanabe, 2021) calculates additional CTC losses using intermediate representations from the encoder. Self-conditioned CTC (Nozaki and Komatsu, 2021) further extends InterCTC by adding back predictions of intermediate CTC layers to the subsequent encoder. These approaches have shown to be highly effective in speech-to-text generation tasks without a decoder (Higuchi et al., 2021).

Although CTC assumes a monotonic alignment between input and output, it is promising for ST due to the reordering capability of self-attention (Inaguma et al., 2021; Chuang et al., 2021).

Conventional CTC models are typically designed for a specific task or language. It remains under-explored whether such approaches can be scaled to multilingual and multi-task scenarios. This work proposes a novel encoder-only speech foundation model based on multi-task self-conditioned CTC. This single model performs well in multilingual ASR, ST and LID.

3 OWSM-CTC

3.1 Overall architecture

Figure 2 shows the architecture of OWSM-CTC. Its main component is a speech encoder, which takes speech features as input and predicts the spoken language as well as the ASR or ST hypothesis using CTC. To mimic Whisper-style models, which condition text generation on an optional text prompt (Radford et al., 2023; Peng et al., 2023e, 2024), we employ a separate Transformer encoder to process the prompt and inject the output to the main model through cross-attention. Then, the model can potentially attend to the text prompt when generating text.

3.2 Speech encoder

For an input waveform, we first extract log Mel filterbanks and then apply a 2D convolution module to downsample the feature sequence along the time dimension. Let $\mathbf{X}_{\text{speech}} \in \mathbb{R}^{T \times d}$ be the downsampled feature sequence of length T and feature size d. To specify the language and task, we prepend two special tokens to the sequence:

$$\mathbf{X} = \text{concat}(\mathbf{e}_{\text{lang}}, \mathbf{e}_{\text{task}}, \mathbf{X}_{\text{speech}}), \tag{1}$$

where $\text{concat}(\cdot)$ is concatenation along time and $\mathbf{e}_{\text{lang}}, \mathbf{e}_{\text{task}} \in \mathbb{R}^{1 \times d}$ are embeddings of special tokens `<lang>` and `<task>`, respectively. $\mathbf{X}$ now has shape $(T+2) \times d$. If the spoken language is known, the true language token will be used as input. Otherwise, a special token `<nolang>` denoting "unknown language" will be used. During training, we randomly replace the true language with `<nolang>` according to probability 0.5 so that either can be used for inference. The task token is `<asr>` for speech recognition and `<st_lang>` for translation to a target language.

Next, we add sinusoidal positional embeddings to $\mathbf{X}$, and apply a stack of N encoder layers:

$$\mathbf{X}^{(0)} = \mathbf{X} + \text{PosEmb}(\mathbf{X}), \tag{2}$$

$$\mathbf{X}^{(l)} = \text{SpeechEnc}^{(l)}(\mathbf{X}^{(l-1)}), \tag{3}$$

where l is a layer index from 1 to N, $\text{PosEmb}(\cdot)$ generates positional embeddings, and

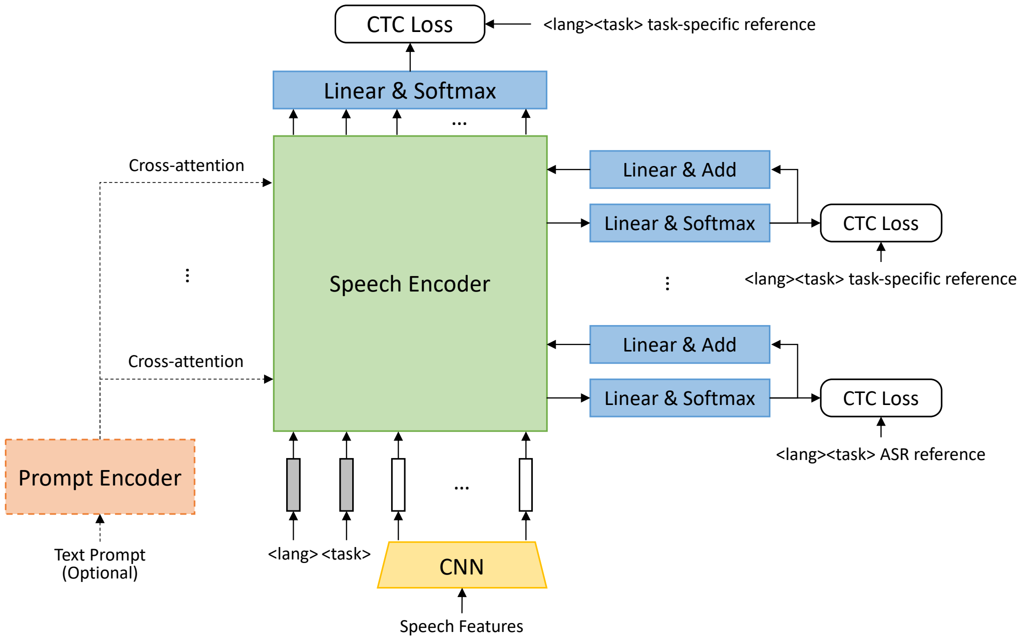


Figure 2: Architecture of our OWSM-CTC. For an input audio, it predicts a language token along with ASR or ST text tokens depending on the task specifier. An optional text prompt can be provided, which mimics Whisper.

$\text{SpeechEnc}^{(l)}(\cdot)$ is the l-th encoder layer. The encoder is E-Branchformer (Kim et al., 2023), an enhanced version of Branchformer (Peng et al., 2022), which shows excellent performance across a wide range of benchmarks (Peng et al., 2023b).

We compute the CTC loss using the final encoder output $\mathbf{X}^{(N)}$ and an augmented reference $\mathbf{y}_{\text{task}}$. To create this reference, we simply prepend `<lang>` and `<task>` to the original groundtruth text of the desired task. Hence, the model will learn to predict the language token in addition to ASR or ST text tokens. This CTC loss is denoted as follows:

$$\mathcal{L}^{(N)} = -\log P_{\text{CTC}}(\mathbf{y}_{\text{task}} \mid \text{softmax}(\mathbf{X}^{(N)}\mathbf{W}_1)), \quad (4)$$

where $\mathbf{W}_1 \in \mathbb{R}^{d \times V}$ is a linear layer and V is the size of the CTC vocabulary.

As discussed in Section 2.3, we apply self-conditioned CTC (Nozaki and Komatsu, 2021) at intermediate layers $\mathcal{S} \subseteq \{1, \dots, N-1\}$ to alleviate the conditional independence assumption of CTC. For any layer $s \in \mathcal{S}$, Equation 3 is replaced by the following operations:

$$\mathbf{A}^{(s)} = \text{SpeechEnc}^{(s)}(\mathbf{X}^{(s-1)}), \quad (5)$$

$$\mathbf{B}^{(s)} = \text{softmax}(\mathbf{A}^{(s)}\mathbf{W}_1), \quad (6)$$

$$\mathbf{X}^{(s)} = \mathbf{A}^{(s)} + \mathbf{B}^{(s)}\mathbf{W}_2, \quad (7)$$

where $\mathbf{W}_2 \in \mathbb{R}^{V \times d}$ is a linear layer. The intermediate CTC loss at layer s is defined as follows:

$$\mathcal{L}^{(s)} = -\log P_{\text{CTC}}(\mathbf{y}^{(s)} \mid \mathbf{B}^{(s)}), \quad (8)$$

where $\mathbf{y}^{(s)}$ is the augmented reference at layer s. Similar to $\mathbf{y}_{\text{task}}$ in Equation 4, we prepend the language and task tokens to the original groundtruth text. Note that the choice of the reference text depends on the task. If the task for the current input is ASR, we simply use the ASR transcript to create $\mathbf{y}^{(s)}$ for all s, which is consistent with conventional ASR models. However, if the task is ST, we empirically find that the model cannot converge if we use the translated text as the reference at all intermediate layers $\mathcal{S}$ (see Appendix B.2 for discussions). Therefore, as shown in Figure 2, we utilize the ASR transcript at the first N_{ASR} layers and the ST text at the remaining N_{ST} layers, where $N_{\text{ASR}} + N_{\text{ST}} = |\mathcal{S}| \leq N - 1$. This design mimics a cascaded system that first performs ASR and then ST, but our entire model is optimized jointly and trained from scratch. In other words, the first N_{ASR} CTC layers always perform ASR regardless of the task token (named "ASR-only CTC"), whereas the other CTC layers are multi-tasking - they can perform ASR or ST according to the task token (named "task-specific or task-dependent CTC").

The overall training loss is an average of the loss terms defined in Equation 4 and Equation 8:

$$\mathcal{L}_{\text{total}} = \frac{1}{1 + |\mathcal{S}|}\left(\mathcal{L}^{(N)} + \sum_{s \in \mathcal{S}} \mathcal{L}^{(s)}\right). \quad (9)$$

3.3 Prompt encoder

Whisper-style models generate text conditioned on an optional text prompt (Radford et al., 2023; Peng

	Params	Time shift	Training data	GPU hours
Whisper (encoder-decoder) (Radford et al., 2023)				
base	74M	20ms	680k hours	unknown
small	244M	20ms	680k hours	unknown
medium	769M	20ms	680k hours	unknown
OWSM v3.1 (encoder-decoder) (Peng et al., 2024)				
base	101M	40ms	180k hours	2.3k
medium	1.02B	40ms	180k hours	24.6k
OWSM-CTC (ours)				
medium	1.01B	80ms	180k hours	19.2k

Table 1: Summary of model size, training data and training cost measured on NVIDIA A100 GPU (40GB).

et al., 2023e, 2024). During training, this prompt is simply the previous sentence in the same audio recording. During inference, it can be provided by the user to potentially adjust the output. For encoder-decoder models like Whisper, the text prompt is a prefix to the autoregressive decoder. For our encoder-only model, we leverage a separate Transformer encoder to process the prompt and inject it to the speech encoder through cross-attention. If no prompt is provided, a special token <na> will be used. Let $\mathbf{X}_{\text{prompt}} \in \mathbb{R}^{T' \times d'}$ be the output of the prompt encoder. We insert a cross-attention layer at a subset of layers $\mathcal{T} \subseteq \{1, \dots, N\}$ of the speech encoder. For any $t \in \mathcal{T}$, the original $\text{SpeechEnc}^{(t)}(\cdot)$ in Equation 3 or Equation 5 becomes $\text{SpeechEncCA}^{(t)}(\cdot, \cdot)$:

$$\mathbf{D}^{(t)} = \text{SpeechEnc}^{(t)}(\mathbf{X}^{(t-1)}), \tag{10}$$

$$\text{SpeechEncCA}^{(t)}(\mathbf{X}^{(t-1)}, \mathbf{X}_{\text{prompt}}) = \mathbf{D}^{(t)} + \text{CrossAtt}(\mathbf{D}^{(t)}, \mathbf{X}_{\text{prompt}}, \mathbf{X}_{\text{prompt}}), \tag{11}$$

where $\text{CrossAtt}(\cdot, \cdot, \cdot)$ is a cross-attention layer with three arguments: query, key, and value.

Our training data is a mixture of public ASR and ST datasets. Some of them provide unsegmented long audio, but the others only release segmented short audio. At training time, if the sample does not have a previous sentence, we will use <na>. Otherwise, we use either <na> or the previous sentence as the prompt according to 0.5 probability. Section 4.6 shows that OWSM-CTC can leverage the prompt's information when necessary.

4 Experiments

4.1 Experimental setups

Table 1 is a brief summary of model size, training data, and training cost.

Data format. Our training data is prepared using scripts publicly released by OWSM v3.1 (Peng et al., 2024). It is a mixture of more than 25 public ASR and ST corpora covering 151 languages and various translation directions. The total audio duration is 180k hours. To create long-form data, consecutive utterances from the same audio recording are concatenated to a duration of no more than 30 seconds. The input audio to the model is always padded to a fixed length of 30 seconds. Appendix A.1 and Table 10 present the training data statistics. The original Whisper-style data contains the start and end timestamps for each utterance. These timestamp tokens are predicted along with normal text tokens during the autoregressive decoding. In OWSM-CTC, we do not include any explicit timestamps since the time-aligned hypothesis can be obtained by forced alignment if desired.

Model architecture. Our speech encoder is a 27-layer E-Branchformer with a hidden size of 1024 and 16 attention heads. Four intermediate layers (6, 12, 15, and 21) are used for self-conditioned CTC. The first three are ASR only, while the others are task-specific. The prompt encoder is a 4-layer Transformer with a hidden size of 512 and 8 attention heads. It is injected into the speech encoder at every third layer. The total model size is 1.01B, which matches the size of the encoder-decoder OWSM v3.1 medium (1.02B). More details about the architecture are in Appendix A.2 (see Table 11).

Implementation. We implement OWSM-CTC in ESPnet (Watanabe et al., 2018) based on PyTorch (Paszke et al., 2019). FlashAttention (Dao et al., 2022) is used to improve training efficiency, but it is not used for inference. The batch size per GPU is 4, and 64 NVIDIA A100 GPUs (40GB) are used with distributed data parallel. The total training time is approximately 300 hours. For optimization, we employ the Adam optimizer (Kingma and Ba, 2015) with the piece-wise linear learning rate schedule (Peng et al., 2024). The peak learning rate is 2e-4. Other training hyperparameters can be found in Appendix A.3 (see Table 12).

Evaluation. We fairly compare our encoder-only OWSM-CTC with the previously released encoder-decoder OWSM v3.1 models (Peng et al., 2024) since they are trained on the same data. We also show the results of Whisper under the same decoding setup for reference, but we note that they are not comparable with ours due to completely different training data. By default, short-form audio without any text prompt is used, but we also evaluate the long-form ASR performance in Section 4.5 and

	Accuracy % (↑)
Whisper (encoder-decoder) (Radford et al., 2023)	
base	47.6
small	53.1
medium	54.8
OWSM v3 (encoder-decoder) (Peng et al., 2023e)	
medium	81.4
OWSM v3.1 (encoder-decoder) (Peng et al., 2024)	
base	41.9
medium	75.6
OWSM-CTC (ours)	
medium	<u>**87.6**</u>

Table 2: Spoken language identification results on the FLEURS test set. **Bold**: the best result. <u>Underlined</u>: our OWSM-CTC outperforms OWSM v3.1 medium.

	CommonVoice en	FLEURS en	LibriSpeech test-clean	LibriSpeech test-other	MLS en	Switchboard eval2000	TEDLIUM	VoxPopuli en	WSJ eval92	Average WER (↓)	Speed-up (↑)
Whisper (encoder-decoder) (Radford et al., 2023)											
base	25.2	12.4	5.1	12.0	13.4	25.7	6.3	10.2	5.0	12.8	2.40x
small	15.7	9.6	3.3	7.7	9.1	22.2	**4.6**	8.5	4.3	9.4	1.46x
medium	**11.9**	**6.4**	2.8	6.5	10.2	19.4	5.1	**7.6**	**2.9**	8.1	0.76x
OWSM v3.1 (encoder-decoder) (Peng et al., 2024)											
base	21.5	14.8	3.6	9.1	12.0	22.9	7.8	12.0	5.3	12.1	2.97x
medium	12.6	9.0	**2.4**	**5.0**	**7.1**	**16.3**	5.1	8.4	3.5	**7.7**	1.00x
OWSM-CTC (ours)											
medium	<u>12.1</u>	9.9	**2.4**	5.2	7.3	16.9	<u>4.9</u>	8.6	4.2	7.9	<u>**3.63x**</u>

Table 3: WER % (↓) of English ASR. Speed-up (↑) is measured using the average decoding time. Whisper is trained on 438k hours of English audio, whereas OWSM v3.1 and our OWSM-CTC are trained on only 73k hours. **Bold**: the best result. <u>Underlined</u>: our OWSM-CTC outperforms OWSM v3.1 medium.

investigate the effect of text prompt in Section 4.6.

4.2 Language identification

Table 2 presents the LID results on the FLEURS test set. Our OWSM-CTC achieves a top-1 accuracy of 87.6%, outperforming the other encoder-decoder models by a large margin. This is likely because spoken LID requires a powerful encoder to extract useful information from the input audio. Our encoder-only model is especially suitable for this type of task.

4.3 Speech recognition

Table 3 presents word error rates (WERs) on multiple English ASR test sets. Following Peng et al. (2023e, 2024), we leverage greedy decoding and apply the Whisper English text normalizer before

	MLS es	MLS fr	MLS de	MLS nl	MLS it	MLS pt	MLS pl	AISHELL-1 (zh)	KsponSpeech clean (ko)	KsponSpeech other (ko)	ReazonSpeech (ja)	Average Error Rate (↓)
data size	11.1	9.8	13.3	2.1	2.6	8.6	4.3	23.4	8.0	8.0	7.1	
Whisper (encoder-decoder) (Radford et al., 2023)												
base	14.5	25.2	19.9	30.9	32.9	23.5	25.2	39.1	27.0	22.9	54.1	28.7
small	9.1	13.6	11.5	18.2	21.3	13.8	12.5	25.1	24.0	15.4	32.5	17.9
medium	**6.1**	**9.7**	**8.1**	**12.2**	**15.6**	**8.9**	**6.8**	15.7	17.6	**12.8**	25.3	**12.6**
data size	2.0	2.5	3.7	1.7	0.7	0.3	0.3	16.3	1.0	1.0	18.9	
OWSM v3.1 (encoder-decoder) (Peng et al., 2024)												
base	18.5	24.2	18.7	28.6	33.7	44.9	49.7	12.2	23.8	26.1	11.2	26.5
medium	9.0	12.1	10.8	18.1	20.2	21.6	25.2	**6.4**	16.7	18.9	**7.9**	15.2
OWSM-CTC (ours)												
medium	10.3	12.9	11.9	20.4	22.1	23.5	31.6	**6.4**	<u>**14.8**</u>	<u>16.5</u>	8.1	16.2

Table 4: Multilingual ASR results. CER % (↓) is shown for Chinese (zh), Korean (ko) and Japanese (ja), while WER % (↓) is shown for the others. Data sizes are in thousand hours. **Bold**: the best result. <u>Underlined</u>: our OWSM-CTC outperforms OWSM v3.1 medium.

scoring. We record the average decoding time across all English test sets on NVIDIA A40 GPU and calculate the relative speed-up. Results show that our non-autoregressive OWSM-CTC generally has comparable WERs with the autoregressive OWSM v3.1 medium (average: 7.9 vs. 7.7), both of which have 1B parameters. However, OWSM-CTC achieves 3.63x speed-up due to parallel decoding. Notably, OWSM-CTC is even faster than OWSM v3.1 base, which has only 100M parameters, and our WERs are much lower (average: 7.9 vs. 12.1). Compared to Whisper models trained on significantly more data, our OWSM-CTC is still competitive in many cases, and our inference is much faster. These results demonstrate that OWSM-CTC achieves an excellent trade-off between recognition accuracy and inference efficiency.

Table 4 shows the results of multilingual ASR. We perform greedy decoding and apply the Whisper basic text normalizer before scoring. Our OWSM-CTC is slightly worse than OWSM v3.1 in terms of the average WER/CER (16.2 vs. 15.2). For European languages in MLS (Pratap et al., 2020), OWSM-CTC generally falls behind. But for East Asian languages like Chinese, Japanese and Korean, OWSM-CTC is on par with or better than OWSM v3.1 medium. This difference might be related to the training data size and tokenization.

4.4 Speech translation

We evaluate the ST performance using the CoVoST-2 (Wang et al., 2020a) test sets. Again, we perform

Src Lang.	de	es	fr	ca	Average (↑)	Speed-up (↑)
data size	4.3	6.7	4.5	0.2		
Whisper (encoder-decoder) (Radford et al., 2023)						
base	11.4	19.2	13.1	9.7	13.4	1.84x
small	25.0	32.8	26.4	21.7	26.5	1.54x
medium	**33.6**	**39.7**	**34.4**	**29.2**	**34.2**	0.84x
data size	0.2	0.1	0.3	0.1		
OWSM v3.1 (encoder-decoder) (Peng et al., 2024)						
base	7.3	10.0	11.1	9.0	9.4	2.78x
medium	17.1	22.3	22.7	18.4	20.1	1.00x
OWSM-CTC (ours)						
medium	<u>21.1</u>	<u>28.2</u>	<u>27.7</u>	<u>23.7</u>	<u>25.2</u>	**<u>3.35x</u>**

Table 5: BLEU (↑) of X-to-En ST on CoVoST-2. Speed-up is measured using average decoding time. Data sizes are in thousand hours. **Bold**: the best result. <u>Underlined</u>: our OWSM-CTC outperforms OWSM v3.1 medium.

greedy decoding and calculate BLEU scores using lowercase without punctuation. For X-to-En translation, we follow OWSM v3.1 (Peng et al., 2024) to report results of directions where the training data size is over 100 hours. For the other low-resource directions, both OWSM v3.1 and our OWSM-CTC do not work in general. For En-to-X translation, we report results in all 15 directions. We calculate the speed-up based on the average decoding time on the NIVIDA A40 GPU.

Table 5 shows the X-to-En results. Notably, our encoder-only OWSM-CTC consistently outperforms the encoder-decoder OWSM v3.1 by a large margin. The average BLEU score is improved from 20.1 to 25.2 (25% relatively). We also achieve 3.35x speed-up for inference.

Table 6 presents En-to-X results. OpenAI Whisper does not support these directions. Similarly, our OWSM-CTC achieves superior performance than OWSM v3.1 in 12 out of 15 translation directions. The average BLEU is improved from 13.3 to 15.0 (13% relatively), and the inference speed-up is 4.20 times.

We have the following observations from the ST results: (1) Our non-autoregressive OWSM-CTC generally achieves 3 to 4 times speed-up compared to the encoder-decoder baseline, which is consistent with ASR. (2) OWSM-CTC even improves the ST performance sometimes by a large margin. One reason is that the autoregressive model suffers from hallucination and error propagation, while the non-autoregressive model is more stable. (3) The BLEU improvement of X-to-En is larger than that of En-to-X, likely because: (i) the OWSM training set contains lots of English ASR data and OWSM-CTC might obtain strong capability of generating English text; (ii) X-to-En has fewer training data than En-to-X, and the encoder-decoder model may need a sufficient amount of training data to achieve good performance for translation.

Our findings reveal that large-scale CTC-based models are also promising for ST in various language pairs, which is consistent with prior investigations at smaller scales (Yan et al., 2023).

4.5 Long-form speech recognition

For long-form ASR, a model takes as input an unsegmented audio recording of arbitrary length and generates the entire transcription without explicit voice activity detection. Whisper and encoder-decoder OWSM can predict start and end timestamps of each utterance within a fixed-length segment. Those timestamps are used to shift the recognition window for chunk-wise long-form ASR. However, this chunk-wise recognition is a sequential process because the location of the next chunk depends on the predicted timestamp in the current chunk. By contrast, our OWSM-CTC performs chunk-wise recognition in a fully parallel manner. We first split the entire audio into overlapped chunks of 30s, where the overlapped region serves as the left and right context.[2] We then perform CTC greedy decoding on batched chunks. The batch size is 32 on a single NVIDIA A40 GPU (48GB). Table 7 shows the WER and speed-up with different context lengths. Our OWSM-CTC achieves lower WERs than the encoder-decoder OWSM v3.1, while being approximately 20 times faster due to the batched parallel decoding. OWSM-CTC is also robust to different context lengths. These observations indicate that CTC-based non-autoregressive models perform very well for long-form ASR, which is consistent with prior findings (Koluguri et al., 2023).

4.6 Effect of text prompt

As described in Figure 2 and Section 3.3, OWSM-CTC can take an additional text prompt as input which might change the output. During training, either a special token <na> or the previous sentence in the same audio is used as the prompt according to a probability of 0.5, which follows the setup of Whisper and OWSM. To verify that OWSM-CTC can utilize information from the prompt when

[2]We follow this tutorial for long-form ASR with CTC: https://github.com/NVIDIA/NeMo/blob/main/tutorials/asr/Streaming_ASR.ipynb

Tgt Lang.	de	ca	zh	fa	et	mn	tr	ar	sv	lv	sl	ta	ja	id	cy	Average (↑)	Speed-up (↑)
data size	14.0	0.4	13.7	0.8	0.4	0.4	0.9	0.9	0.4	0.4	0.4	0.4	1.0	0.4	0.4		
OWSM v3.1 (encoder-decoder) (Peng et al., 2024)																	
base	14.6	7.7	14.5	3.0	1.8	1.0	1.2	1.6	8.1	1.3	0.7	0.0	8.7	5.1	4.5	4.9	2.39x
medium	25.4	19.6	32.1	**10.1**	7.7	4.6	6.5	7.2	20.3	6.4	9.0	0.0	**19.6**	16.1	15.3	13.3	1.00x
OWSM-CTC (ours)																	
medium	**25.5**	**23.0**	**35.1**	10.0	**9.2**	**4.8**	**6.8**	**8.2**	**23.8**	**7.7**	**12.0**	0.0	18.5	**21.0**	**19.4**	**15.0**	**4.20x**

Table 6: BLEU (↑) of En-to-X ST on CoVoST-2. Speed-up is measured using the average decoding time across all 15 directions. Data sizes are in thousand hours. **Bold**: the best result. Underlined: our OWSM-CTC outperforms OWSM v3.1 medium. Note that Whisper (Radford et al., 2023) does not support En-to-X translation.

	Context Length	WER % (↓)	Speed-up (↑)
Whisper (encoder-decoder) (Radford et al., 2023)			
base	-	5.3	1.40x
small	-	4.4	1.62x
medium	-	**3.8**	0.86x
OWSM v3.1 (encoder-decoder) (Peng et al., 2024)			
base	-	9.6	1.40x
medium	-	5.7	1.00x
OWSM-CTC (ours)			
medium	2s	5.4	**22.40x**
	4s	5.2	19.35x
	6s	5.2	16.07x
	8s	5.2	12.09x

Table 7: Long-form ASR results on the TEDLIUM (Hernandez et al., 2018) test set which consists of 11 audio recordings ranging from 6 to 27 minutes. **Bold**: the best result. Underlined: our OWSM-CTC outperforms OWSM v3.1 medium.

Dataset	Previous text as prompt?	WER % (↓)
TEDLIUM dev	No	4.9
	Yes	4.1

Table 8: Using the previous sentence (groundtruth) as a text prompt improves the ASR WER of OWSM-CTC. The optional prompt encoder is defined in Figure 2 and Section 3.3.

Input length	5s	10s	20s
Whisper (encoder-decoder) (Radford et al., 2023)			
large-v3	Fjell	Fusilet	Rekordverk
OWSM v3.1 (encoder-decoder) (Peng et al., 2024)			
medium	thank you	thank you	(Applause)
OWSM-CTC (ours)			
medium	.	(	( )

Table 9: ASR outputs with random noise as input.

necessary, we perform greedy decoding on the TEDLIUM dev set, where the previous sentence of each utterance is available. As shown in Table 8, using the previous sentence as the text prompt reduces the WER from 4.9% to 4.1%. Appendix C provides an example where the previous sentence also affects the output text style.

4.7 Robustness

To investigate the robustness, we first consider random noise as input. Table 9 shows the ASR outputs generated by three models. Encoder-decoder models including Whisper and OWSM v3.1 tend to generate some text that looks meaningful, while our OWSM-CTC only generates some punctuation marks without actual meaning. Note that punctuation marks are typically removed before ASR scoring, so our error rate will be zero.

Another typical issue for autoregressive decoding is that the generation might fall into an infinite loop of a few characters or words until reaching the maximum output length. Table 16 in Appendix D presents two examples from ASR and ST, respectively. Our non-autoregressive model is more robust in such cases.

5 Conclusion

We propose OWSM-CTC, a novel encoder-only speech foundation model built upon 180k hours of public audio data and open-source toolkits. OWSM-CTC employs multi-task self-conditioned CTC for multilingual ASR, any-to-any ST, and LID. We conduct extensive experiments to compare OWSM-CTC with the encoder-decoder OWSM models trained on the same data. We find that OWSM-CTC achieves competitive performance on ASR and superior performance on ST for both X-to-En (25% relative improvement) and En-to-X (13% relative improvement), while being more robust and 3 to 4 times faster at inference time. Additionally, OWSM-CTC improves the long-form ASR WER with 20 times faster inference due to the batched parallel decoding. OWSM-CTC also outperforms the baselines on LID. To promote open research on large speech models, we will publicly release our codebase, pre-trained model weights and training logs.

Limitations

Although our OWSM-CTC is several times faster and has comparable or superior performance than the encoder-decoder OWSM v3.1 in a wide range of benchmarks, it may still generate incorrect ASR or ST outputs due to limited training in certain languages. Care should be taken when using our model for low-resource ASR or ST. Besides, we have only evaluated our model with greedy decoding as it has the fastest inference speed. The non-autoregressive model sometimes makes mistakes in spelling or grammar due to lack of language models.

Broader Impacts and Ethics

Our OWSM-CTC is a novel encoder-only speech foundation model built upon public datasets and open-source toolkits. It achieves very strong performance and efficiency compared to other popular choices. We adhere to the ACL ethics policy and there is no violation of privacy in our experiments. We plan to publicly release all scripts, pre-trained models, and training logs, which can promote transparency and open science. We believe this will benefit the entire speech research community and it can make the latest speech technology available to a broader range of people all over the world.

References

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, and Julian Schrittwieser et al. 2023a. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernández Ábrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan A. Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vladimir Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, and et al. 2023b. Palm 2 technical report. *CoRR*, abs/2305.10403.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 4218–4222. European Language Resources Association.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Jeong-Uk Bang, Seung Yun, Seung-Hi Kim, Mu-Yeol Choi, Min-Kyu Lee, Yeo-Jeong Kim, Dong-Hyun Kim, Jun Park, Young-Jik Lee, and Sang-Hun Kim. 2020. Ksponspeech: Korean spontaneous speech corpus for automatic speech recognition. *Applied Sciences*, 10(19).

Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenthaler, Paul-Ambroise Duquenne, Brian Ellis, Hady Elsahar, Justin Haaheim, John Hoffman, Min-Jae Hwang, Hirofumi Inaguma, Christopher Klaiber, Ilia Kulikov, Pengwei Li, Daniel Licht, Jean Maillard, Ruslan Mavlyutov, Alice Rakotoarison, Kaushik Ram Sadagopan, Abinesh Ramakrishnan, Tuan Tran, Guillaume Wenzek, Yilin Yang, Ethan Ye, Ivan Evtimov, Pierre Fernandez, Cynthia Gao, Prangthip Hansanti, Elahe Kalbassi, Amanda Kallet, Artyom Kozhevnikov, Gabriel Mejia Gonzalez, Robin San Roman, Christophe Touret, Corinne Wong, Carleigh Wood, Bokai Yu, Pierre Andrews, Can Balioglu, Peng-Jen Chen, Marta R. Costa-jussà, Maha Elbayad, Hongyu Gong, Francisco Guzmán, Kevin Heffernan, Somya Jain, Justine Kao, Ann Lee, Xutai Ma, Alexandre Mourachko, Benjamin Peloquin, Juan Pino, Sravya Popuri, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, Anna Sun, Paden Tomasello, Changhan Wang, Jeff Wang, Skyler Wang, and Mary Williamson. 2023. Seamless: Multilingual expressive and streaming speech translation. *CoRR*, abs/2312.05187.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng. 2017. Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline. In *2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*, pages 1–5.

Maxime Burchi and Valentin Vielzeuf. 2021. Efficient conformer: Progressive downsampling and grouped attention for automatic speech recognition. In *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2021, Cartagena, Colombia, December 13-17, 2021*, pages 8–15. IEEE.

Jean Carletta. 2007. Unleashing the killer corpus: experiences in creating the multi-everything AMI meeting corpus. *Lang. Resour. Evaluation*, 41(2):181–190.

Roldano Cattoni, Mattia Antonino Di Gangi, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2021. Must-c: A multilingual corpus for end-to-end speech translation. *Comput. Speech Lang.*, 66:101155.

Heng-Jui Chang, Shu-Wen Yang, and Hung-yi Lee. 2022. Distilhubert: Speech representation learning by layer-wise distillation of hidden-unit bert. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, Virtual and Singapore, 23-27 May 2022*, pages 7087–7091. IEEE.

Guoguo Chen, Shuzhou Chai, Guan-Bo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, Mingjie Jin, Sanjeev Khudanpur, Shinji Watanabe, Shuaijiang Zhao, Wei Zou, Xiangang Li, Xuchen Yao, Yongqing Wang, Zhao You, and Zhiyong Yan. 2021. Gigaspeech: An evolving, multi-domain ASR corpus with 10, 000 hours of transcribed audio. In *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, pages 3670–3674. ISCA.

Nanxin Chen, Shinji Watanabe, Jesús Villalba, and Najim Dehak. 2019. Listen and fill in the missing letters: Non-autoregressive transformer for speech recognition. *CoRR*, abs/1911.04908.

Ethan A. Chi, Julian Salazar, and Katrin Kirchhoff. 2021. Align-refine: Non-autoregressive speech recognition via iterative realignment. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1920–1927, Online. Association for Computational Linguistics.

Shun-Po Chuang, Yung-Sung Chuang, Chih-Chiang Chang, and Hung-yi Lee. 2021. Investigating the reordering capability in CTC-based non-autoregressive end-to-end speech translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1068–1077, Online. Association for Computational Linguistics.

Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. 2023. Fleurs: Few-shot learning evaluation of universal representations of speech. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 798–805.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Shaojin Ding, David Qiu, David Rim, Yanzhang He, Oleg Rybakov, Bo Li, Rohit Prabhavalkar, Weiran Wang, Tara N. Sainath, Shivani Agrawal, Zhonglin Han, Jian Li, and Amir Yazdanbakhsh. 2023. Usmlite: Quantization and sparsity aware fine-tuning for speech recognition with universal speech models. *CoRR*, abs/2312.08553.

Sanchit Gandhi, Patrick von Platen, and Alexander M. Rush. 2023. Distil-whisper: Robust knowledge distillation via large-scale pseudo labelling. *CoRR*, abs/2311.00430.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *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.

John J. Godfrey, Edward Holliman, and Jane McDaniel. 1992. SWITCHBOARD: telephone speech corpus for research and development. In *1992 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '92, San Francisco, California, USA, March 23-26, 1992*, pages 517–520. IEEE Computer Society.

Yuan Gong, Hongyin Luo, Alexander H. Liu, Leonid Karlinsky, and James R. Glass. 2023. Listen, think, and understand. *CoRR*, abs/2305.10790.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *CoRR*, abs/1211.3711.

Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, volume 148 of *ACM International Conference Proceeding Series*, pages 369–376. ACM.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

Ankit Gupta, George Saon, and Brian Kingsbury. 2024. Exploring the limits of decoder-only models trained on public speech recognition corpora. *CoRR*, abs/2402.00235.

François Hernandez, Vincent Nguyen, Sahar Ghannay, Natalia A. Tomashenko, and Yannick Estève. 2018.

TED-LIUM 3: Twice as much data and corpus repartition for experiments on speaker adaptation. In *Speech and Computer - 20th International Conference, SPECOM 2018, Leipzig, Germany, September 18-22, 2018, Proceedings*, volume 11096 of *Lecture Notes in Computer Science*, pages 198–208. Springer.

Yosuke Higuchi, Nanxin Chen, Yuya Fujita, Hirofumi Inaguma, Tatsuya Komatsu, Jaesong Lee, Jumon Nozaki, Tianzi Wang, and Shinji Watanabe. 2021. A comparative study on non-autoregressive modelings for speech-to-text generation. In *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2021, Cartagena, Colombia, December 13-17, 2021*, pages 47–54. IEEE.

Yosuke Higuchi, Shinji Watanabe, Nanxin Chen, Tetsuji Ogawa, and Tetsunori Kobayashi. 2020. Mask CTC: non-autoregressive end-to-end ASR with CTC and mask predict. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 3655–3659. ISCA.

Hirofumi Inaguma, Yosuke Higuchi, Kevin Duh, Tatsuya Kawahara, and Shinji Watanabe. 2021. Orthros: non-autoregressive end-to-end speech translation with dual-decoder. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7503–7507.

Kwangyoun Kim, Felix Wu, Yifan Peng, Jing Pan, Prashant Sridhar, Kyu J. Han, and Shinji Watanabe. 2023. E-branchformer: Branchformer with enhanced merging for speech recognition. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 84–91.

Sehoon Kim, Amir Gholami, Albert E. Shaw, Nicholas Lee, Karttikeya Mangalam, Jitendra Malik, Michael W. Mahoney, and Kurt Keutzer. 2022. Squeezeformer: An efficient transformer for automatic speech recognition. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Suyoun Kim, Takaaki Hori, and Shinji Watanabe. 2017. Joint ctc-attention based end-to-end speech recognition using multi-task learning. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4835–4839.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Nithin Rao Koluguri, Samuel Kriman, Georgy Zelenfroind, Somshubra Majumdar, Dima Rekesh, Vahid Noroozi, Jagadeesh Balam, and Boris Ginsburg. 2023. Investigating end-to-end ASR architectures for long form audio transcription. *CoRR*, abs/2309.09950.

Cheng-I Jeff Lai, Yang Zhang, Alexander H. Liu, Shiyu Chang, Yi-Lun Liao, Yung-Sung Chuang, Kaizhi Qian, Sameer Khurana, David D. Cox, and Jim Glass. 2021. PARP: prune, adjust and re-prune for self-supervised speech recognition. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 21256–21272.

Jaesong Lee and Shinji Watanabe. 2021. Intermediate loss regularization for ctc-based speech recognition. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6224–6228.

Yeonghyeon Lee, Kangwook Jang, Jahyun Goo, Youngmoon Jung, and Hoi Rin Kim. 2022. Fithubert: Going thinner and deeper for knowledge distillation of speech self-supervised models. In *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*, pages 3588–3592. ISCA.

Piotr Nawrot, Jan Chorowski, Adrian Lancucki, and Edoardo Maria Ponti. 2023. Efficient transformers with dynamic token pooling. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6403–6417, Toronto, Canada. Association for Computational Linguistics.

Edwin G. Ng, Chung-Cheng Chiu, Yu Zhang, and William Chan. 2021. Pushing the limits of non-autoregressive speech recognition. In *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, pages 3725–3729. ISCA.

Jumon Nozaki and Tatsuya Komatsu. 2021. Relaxing the conditional independence assumption of ctc-based ASR by conditioning on intermediate predictions. In *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, pages 3735–3739. ISCA.

Patrick K. O'Neill, Vitaly Lavrukhin, Somshubra Majumdar, Vahid Noroozi, Yuekai Zhang, Oleksii Kuchaiev, Jagadeesh Balam, Yuliya Dovzhenko, Keenan Freyberg, Michael D. Shulman, Boris Ginsburg, Shinji Watanabe, and Georg Kucsko. 2021. Spgispeech: 5, 000 hours of transcribed financial audio for fully formatted end-to-end speech recognition. In *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, pages 1434–1438. ISCA.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, pages 5206–5210. IEEE.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

Yifan Peng, Siddharth Dalmia, Ian R. Lane, and Shinji Watanabe. 2022. Branchformer: Parallel mlp-attention architectures to capture local and global context for speech recognition and understanding. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 17627–17643. PMLR.

Yifan Peng, Kwangyoun Kim, Felix Wu, Prashant Sridhar, and Shinji Watanabe. 2023a. Structured pruning of self-supervised pre-trained models for speech recognition and understanding. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Yifan Peng, Kwangyoun Kim, Felix Wu, Brian Yan, Siddhant Arora, William Chen, Jiyang Tang, Suwon Shon, Prashant Sridhar, and Shinji Watanabe. 2023b. A Comparative Study on E-Branchformer vs Conformer in Speech Recognition, Translation, and Understanding Tasks. In *Proc. INTERSPEECH 2023*, pages 2208–2212.

Yifan Peng, Jaesong Lee, and Shinji Watanabe. 2023c. I3D: transformer architectures with input-dependent dynamic depth for speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2023, Rhodes Island, Greece, June 4-10, 2023*, pages 1–5. IEEE.

Yifan Peng, Yui Sudo, Muhammad Shakeel, and Shinji Watanabe. 2023d. Dphubert: Joint distillation and pruning of self-supervised speech models. *CoRR*, abs/2305.17651.

Yifan Peng, Jinchuan Tian, William Chen, Siddhant Arora, Brian Yan, Yui Sudo, Muhammad Shakeel, Kwanghee Choi, Jiatong Shi, Xuankai Chang, Jee-weon Jung, and Shinji Watanabe. 2024. OWSM v3.1: Better and faster open whisper-style speech models based on e-branchformer. *CoRR*, abs/2401.16658.

Yifan Peng, Jinchuan Tian, Brian Yan, Dan Berrebbi, Xuankai Chang, Xinjian Li, Jiatong Shi, Siddhant Arora, William Chen, Roshan Sharma, Wangyou Zhang, Yui Sudo, Muhammad Shakeel, Jee-Weon Jung, Soumi Maiti, and Shinji Watanabe. 2023e. Reproducing whisper-style training using an open-source toolkit and publicly available data. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8.

Matt Post, Gaurav Kumar, Adam Lopez, Damianos Karakos, Chris Callison-Burch, and Sanjeev Khudanpur. 2013. Improved speech-to-text translation with the fisher and callhome Spanish-English speech translation corpus. In *Proceedings of the 10th International Workshop on Spoken Language Translation: Papers*, Heidelberg, Germany.

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, Alexei Baevski, Yossi Adi, Xiaohui Zhang, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. 2023. Scaling speech technology to 1, 000+ languages. *CoRR*, abs/2305.13516.

Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. 2020. MLS: A large-scale multilingual dataset for speech research. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 2757–2761. ISCA.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR.

Dima Rekesh, Nithin Rao Koluguri, Samuel Kriman, Somshubra Majumdar, Vahid Noroozi, He Huang, Oleksii Hrinchuk, Krishna Puvvada, Ankur Kumar, Jagadeesh Balam, and Boris Ginsburg. 2023. Fast conformer with linearly scalable attention for efficient speech recognition. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8.

Paul K. Rubenstein, Chulayuth Asawaroengchai, Duc Dung Nguyen, Ankur Bapna, Zalán Borsos, Félix de Chaumont Quitry, Peter Chen, Dalia El Badawy, Wei Han, Eugene Kharitonov, Hannah Muckenhirn, Dirk Padfield, James Qin, Danny Rozenberg, Tara N. Sainath, Johan Schalkwyk, Matthew Sharifi, Michelle Tadmor Ramanovich, Marco Tagliasacchi, Alexandru Tudor, Mihajlo Velimirovic, Damien Vincent, Jiahui Yu, Yongqiang Wang, Vicky Zayats, Neil Zeghidour, Yu Zhang, Zhishuai Zhang, Lukas Zilka, and Christian Havnø Frank. 2023. Audiopalm: A large language model that can speak and listen. *CoRR*, abs/2306.12925.

Grant Strimel, Yi Xie, Brian John King, Martin Radfar, Ariya Rastrow, and Athanasios Mouchtaris. 2023.

Lookahead when it matters: Adaptive non-causal transformers for streaming neural transducers. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 32654–32676. PMLR.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2023. Efficient transformers: A survey. *ACM Comput. Surv.*, 55(6):109:1–109:28.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

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

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

Mingqiu Wang, Wei Han, Izhak Shafran, Zelin Wu, Chung-Cheng Chiu, Yuan Cao, Nanxin Chen, Yu Zhang, Hagen Soltau, Paul K. Rubenstein, Lukas Zilka, Dian Yu, Golan Pundak, Nikhil Siddhartha, Johan Schalkwyk, and Yonghui Wu. 2023a. Slm: Bridge the thin gap between speech and text foundation models. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020b. Linformer: Self-attention with linear complexity. *CoRR*, abs/2006.04768.

Tianrui Wang, Long Zhou, Ziqiang Zhang, Yu Wu, Shujie Liu, Yashesh Gaur, Zhuo Chen, Jinyu Li, and Furu Wei. 2023b. Viola: Unified codec language models for speech recognition, synthesis, and translation. *CoRR*, abs/2305.16107.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. 2018. Espnet: End-to-end speech processing toolkit. In *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*, pages 2207–2211. ISCA.

Yisheng Xiao, Lijun Wu, Junliang Guo, Juntao Li, Min Zhang, Tao Qin, and Tie-Yan Liu. 2023. A survey on non-autoregressive generation for neural machine translation and beyond. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(10):11407–11427.

Chen Xu, Xiaoqian Liu, Xiaowen Liu, Qingxuan Sun, Yuhao Zhang, Murun Yang, Qianqian Dong, Tom Ko, Mingxuan Wang, Tong Xiao, Anxiang Ma, and Jingbo Zhu. 2023. CTC-based non-autoregressive speech translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13321–13339, Toronto, Canada. Association for Computational Linguistics.

Brian Yan, Siddharth Dalmia, Yosuke Higuchi, Graham Neubig, Florian Metze, Alan W Black, and Shinji Watanabe. 2023. CTC alignments improve autoregressive translation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1623–1639, Dubrovnik, Croatia. Association for Computational Linguistics.

Rong Ye, Chengqi Zhao, Tom Ko, Chutong Meng, Tao Wang, Mingxuan Wang, and Jun Cao. 2022. Gigast: A 10, 000-hour pseudo speech translation corpus. *CoRR*, abs/2204.03939.

Ching-Feng Yeh, Wei-Ning Hsu, Paden Tomasello, and Abdelrahman Mohamed. 2023. Efficient speech representation learning with low-bit quantization. *CoRR*, abs/2301.00652.

Seiji Fujimoto Yue Yin, Daijiro Mori, and S Fujimoto. 2023. Reazonspeech: A free and massive corpus for japanese asr. In *Annual meetings of the Association for Natural Language Processing*.

Ji Won Yoon, Beom Jun Woo, and Nam Soo Kim. 2022. Hubert-ee: Early exiting hubert for efficient speech recognition. *CoRR*, abs/2204.06328.

Binbin Zhang, Hang Lv, Pengcheng Guo, Qijie Shao, Chao Yang, Lei Xie, Xin Xu, Hui Bu, Xiaoyu Chen, Chenchen Zeng, Di Wu, and Zhendong Peng. 2022. Wenetspeech: A 10000+ hours multi-domain mandarin corpus for speech recognition. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6182–6186.

Yu Zhang, Wei Han, James Qin, Yongqiang Wang, Ankur Bapna, Zhehuai Chen, Nanxin Chen, Bo Li, Vera Axelrod, Gary Wang, Zhong Meng, Ke Hu, Andrew Rosenberg, Rohit Prabhavalkar, Daniel S. Park, Parisa Haghani, Jason Riesa, Ginger Perng, Hagen Soltau, Trevor Strohman, Bhuvana Ramabhadran, Tara N. Sainath, Pedro J. Moreno, Chung-Cheng Chiu, Johan Schalkwyk, Françoise Beaufays, and Yonghui Wu. 2023. Google USM: scaling automatic speech recognition beyond 100 languages. *CoRR*, abs/2303.01037.

A Details of Experimental Setups

A.1 Training data

Table 10 summarizes the training data statistics. We prepare the training data mixture using the scripts publicly released by OWSM v3.1 (Peng et al., 2024). This ensures fair comparison between our OWSM-CTC and the previously released encoder-decoder OWSM models.

Our use of the data is consistent with their intended use. These datasets have been widely used in speech research. They do not violate the privacy of creators or users, nor do they contain any offensive content. Specifically, the individual training datasets and licenses are listed below: AIDATATANG (CC BY-NC-ND 4.0)[3], AISHELL-1 (Apache 2.0) (Bu et al., 2017), AMI (CC BY 4.0) (Carletta, 2007), Babel[4], CommonVoice (CC0-1.0) (Ardila et al., 2020), CoVoST2 (CC BY-NC 4.0) (Wang et al., 2020a), Fisher Switchboard (LDC) (Godfrey et al., 1992), Fisher Callhome Spanish (LDC) (Post et al., 2013), FLEURS (CC-BY-4.0) (Conneau et al., 2023), Googlei18n[5], GigaSpeech (Apache 2.0) (Chen et al., 2021), GigaST (CC BY-NC 4.0) (Ye et al., 2022), KsponSpeech (MIT License) (Bang et al., 2020), LibriSpeech (CC BY 4.0) (Panayotov et al., 2015), Multilingual LibriSpeech (CC BY 4.0) (Pratap et al., 2020), MagicData (CC BY-NC-ND 4.0)[6], MuST-C (CC BY NC ND 4.0 International) (Cattoni et al., 2021), SPGISpeech (O'Neill et al., 2021), TEDLIUM3 (CC BY-NC-ND 3.0) (Hernandez et al., 2018), ReazonSpeech (Apache 2.0 / CDLA-Sharing-1.0) (Yin et al., 2023), Russian OpenSTT (CC-BY-NC)[7], VCTK (CC BY 4.0)[8], VoxForge (GPL)[9], VoxPopuli (Attribution-NonCommercial 4.0 International) (Wang et al., 2021), WenetSpeech (Creative Commons Attribution 4.0 International License) (Zhang et al., 2022).

A.2 Model architectures

Table 11 shows the model configurations. Our OWSM-CTC mostly follows the design of OWSM v3.1 medium (Peng et al., 2024), but we only use an encoder. To match the total model size, we increase the number of layers to 27, leading to a total of 1B parameters. Note that the sequence length of the encoder is usually longer than that of the decoder. Hence, the encoder-only model can have a higher computational cost than the encoder-decoder model. To alleviate this issue, we apply a larger downsampling rate in the CNN module to reduce the sequence length. Our final time shift is 80ms, as opposed to 40ms of the encoder-decoder OWSM models. We observe that our training time for a fixed number of updates is roughly the same as that of OWSM v3.1 medium. We also investigated different downsampling strategies at a smaller scale, as discussed in Appendix B.1 and Table 13.

A.3 Training hyperparameters

Table 12 presents the training hyperparameters of OWSM v3.1 and our OWSM-CTC. Again, we follow the previous OWSM v3.1 (Peng et al., 2024) for fair comparison, except that we adopt self-conditioned CTC (Nozaki and Komatsu, 2021) at four intermediate layers (see Section 3.2).

B Small-Scale Ablation Studies

Before the large-scale training using the entire 180k hours of audio data, we also conducted preliminary experiments on MuST-C v2 En-De (Cattoni et al., 2021) to investigate the effect of the CNN

[3] https://www.openslr.org/62/

[4] https://www.iarpa.gov/research-programs/babel

[5] Resources 32, 35, 36, 37, 41, 42, 43, 44, 52, 53, 54, 61, 63, 64, 65, 66, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, and 86 from openslr.org.

[6] https://openslr.org/68/

[7] https://github.com/snakers4/open_stt

[8] https://huggingface.co/datasets/vctk

[9] https://www.voxforge.org/

Model	Unlabelled	English ASR	Other ASR	ST	Languages	Vocabulary Size
Whisper (Radford et al., 2023)						
Initial versions	-	438k hours	117k hours	125k hours	99	52k
large-v3	4M hours	1M hours of labelled in total			100	52k
OWSM v3.1 (Peng et al., 2024)						
	-	73k hours	67k hours	40k hours	151	50k
OWSM-CTC (ours)						
	-	73k hours	67k hours	40k hours	151	50k

Table 10: Details of training data. Our training data is prepared using the scripts released by OWSM v3.1 (Peng et al., 2024).

Model	Params	Encoder	Decoder	Layers	Hidden Size	Attention Heads	Time Shift
Whisper (Radford et al., 2023)							
tiny	39M	Transformer	Transformer	4	384	6	20ms
base	74M	Transformer	Transformer	6	512	8	20ms
small	244M	Transformer	Transformer	12	768	12	20ms
medium	769M	Transformer	Transformer	24	1024	16	20ms
large	1.55B	Transformer	Transformer	32	1280	20	20ms
large-v3	1.55B	Transformer	Transformer	32	1280	20	20ms
OWSM v3.1 (Peng et al., 2024)							
base	101M	E-Branchformer	Transformer	6	384	6	40ms
medium	1.02B	E-Branchformer	Transformer	18	1024	16	40ms
OWSM-CTC (ours)							
medium	1.01B	E-Branchformer	-	27	1024	16	80ms

Table 11: Details of model architectures. Whisper (Radford et al., 2023) and OWSM v3.1 (Peng et al., 2024) are encoder-decoder models, whereas our OWSM-CTC is an encoder-only model. We mostly follow the design of OWSM v3.1 medium, but we increase the number of encoder layers to match the overall model size.

downsampling rate and the choice of the task for intermediate CTC layers. Specifically, we train 24-layer E-Branchformer-CTC models on the combined ASR and ST data from MuST-C v2 En-De. The input is always English audio, but the output can be the English ASR transcript or its German translation depending on the task specifier (see Figure 2).

B.1 Effect of downsampling strategies

Table 13 compares different downsampling strategies while the other configurations are kept the same. The attention is implemented with FlashAttention (Dao et al., 2022). Self-conditioned CTC is applied at three intermediate layers: 6, 12, and 18. The first two CTC layers always perform ASR, while the others are task-dependent. The results show that using 8x downsampling in the CNN module leads to a slight degradation on WER and BLEU but reduces the GPU memory usage by a half. We thus decide to employ 8x downsampling in our large-scale OWSM-CTC, enabling a doubled batch size per GPU. As mentioned in Appendix A.2, with this strategy, we observe a similar training speed compared to the encoder-decoder OWSM model.

B.2 Choice of the CTC task

As discussed in Section 3.2, the intermediate CTC layers can be configured to perform a specific task like ASR or multiple tasks depending on the input task token. Table 14 compares different choices at a small scale using MuST-C v2 En-De. If all CTC layers are task-dependent (i.e., multi-tasking), the model cannot converge when trained from scratch. As more layers are used for ASR only, the ASR WER is improved, but the ST BLEU is slightly decreased. A good trade-off is to use the first half for ASR only and the second half for multi-tasking. Therefore, in our large-scale OWSM-CTC with 27 layers, we configure the 6th, 12th and 15th layers to perform ASR only and the other two CTC layers (i.e., 21st and 27th layers) to be multi-tasking. This design also mimics the conventional cascaded

Model	Batch Size	Total Steps	Warmup Steps	Max Learning Rate	InterCTC Layers $\mathcal{S}$
OWSM v3.1 (Peng et al., 2024)					
base	256	675k	60k	1e-3	-
medium	256	675k	60k	2e-4	-
OWSM-CTC (ours)					
medium	256	600k	60k	2e-4	6, 12, 15, 21

Table 12: Training hyperparameters. We mostly follow the training config of OWSM v3.1 medium (Peng et al., 2024). As described in Section 3.2, we employ self-conditioned CTC at four intermediate layers.

Downsampling Strategy	Params	GPU VRAM (↓)	Speed-up (↑)	ASR WER (↓)	ST BLEU (↑)
4x in CNN	55M	38GB	1.00x	**8.3**	**22.0**
6x in CNN	55M	22GB	1.12x	8.6	21.3
8x in CNN	55M	**19GB**	**1.13x**	8.8	21.5
4x in CNN + 2x in the middle of Encoder	55M	38GB	1.03x	9.7	21.6

Table 13: Comparison of different downsampling strategies on MuST-C v2 En-De. The other configurations such as batch size are kept the same. Using 4x downsampling achieves the best ASR and ST results, while using 8x downsampling significantly reduces the GPU memory usage, which enables a larger batch size per GPU. We employ 8x downsampling in our large-scale OWSM-CTC to reduce training cost.

ASR-Only CTC Layers	Task-Dependent CTC Layers	ASR WER (↓)	ST BLEU (↑)
-	6, 12, 18, 24	diverged	
6	12, 18, 24	9.0	**21.6**
6, 12	18, 24	8.8	21.5
6, 12, 18	24	**8.4**	21.2

Table 14: Effect of the CTC type. This small-scale model has 24 layers with 8x downsampling in CNN. As described in Section 3.2, we employ self-conditioned CTC at some intermediate layers. These CTC layers can perform a single task like ASR or multiple tasks depending on the task specifier. If we allow all CTC layers to perform multiple tasks (ASR and ST), the model cannot converge from scratch. Therefore, we leverage the first few CTC layers for ASR only and the remaining ones for multi-tasking.

system for ST.

C Effect of text prompt

Table 15 is an example from the TEDLIUM dev set, which shows that the text prompt can potentially change the output style. When there is no prompt, the ASR output of OWSM-CTC is in true case with punctuation, and the apostrophes are combined with the previous words. However, when the previous sentence is used as a prompt, the style of the ASR hypothesis becomes more similar to that of the prompt. Specifically, the text is now in lower case without punctuation marks, and the apostrophes are separate from previous words. This style is more consistent with the groundtruth transcript.

Although the above example looks promising for biasing the model's output towards certain directions, we note that this is not guaranteed to work in a zero-shot manner. We have also tried zero-shot contextual biasing, where we provide a few biasing words in the prompt (e.g., person names), but we find that the model may not be able to generate the correct word in many cases. This is mainly because the model is not really trained to perform this type of tasks - we just provide the previous sentence (according to some probability) as the prompt during training, which might not be useful at all; thus, our non-autoregressive model can simply ignore it in most cases. A more practical way to utilize this feature is to fine-tune our pre-trained model using some carefully designed data for contextual biasing. We will explore this in the future.

D Robustness

Table 16 shows that autoregressive decoding sometimes fails to generate the correct output for either ASR or ST, while the non-autoregressive decoding is generally more robust to this type of errors.

Input audio content	Previous sentence	ASR w/o previous	ASR w/ previous
future 's over here wind sun a new energy grid new investments to create high paying jobs repower america it 's time to get real there is an old african proverb that says if you want to go quickly go alone if you want to go far go together we need to go far quickly thank you very much	with one hundred percent clean electricity within ten years a plan to put america back to work make us more secure and help stop global warming finally a solution that 's big enough to solve our problems repower america find out more this is the last one it 's about repowering america one of the fastest ways to cut our dependence on old dirty fuels that are killing our planet	Future's over here. Wind, sun. A new energy grid. New investments to create high-pan jobs. Repower America. It's time to get real. There's an old African proverb that says, "If you want to go quickly, go alone. if you want to go far, go together." We need to go far quickly. Thank you very much. (Applause)	future 's over here wind sun a new energy grid new investments to create high pan jobsrepower america it 's time to get real there 's an old african proverb that says if you want to go quickly go alone if you want to go far go together we need to go far quickly thank you very much

Table 15: Using a previous sentence as the prompt might change the output style. The optional prompt encoder is defined in Figure 2 and Section 3.3.

Groundtruth reference	**OWSM v3.1 output**	**OWSM-CTC output (ours)**
in search of the mythical treasure your grandfather is supposed to have secreted there he laughed and the girl instinctively shuddered with a newborn distrust there was no mirth in the sound	in search of the mythical treasure your grandfather is supposed to have secreted there ha ha ha ha ha ha ha ha ha ha ha ha ha ha ha ha ha ha ...	in search of the mythical treasure your grandfather is supposed to have secreted there he laughed and the girl instinctively shuddered with a new-born distrust there was no mirth in the sound
and with her they began a national tour that took them all around the country	they take a national gira which leads to rerererererererererererere ...	with learn a national tour that leads them to run the entire country

Table 16: Autoregressive decoding sometimes gets trapped in an infinite loop for both ASR (row 1, MLS en) and ST (row 2, CoVoST-2 es-en). Our OWSM-CTC is more robust.