DOI 10.15622/ia.25.1.5

IU. LEZHENIN, N. BOGACH

AN INPUT-SYNCHRONOUS BLOCKWISE DECODING ALGORITHM FOR CTC-AED SPEECH RECOGNITION

Lezhenin I., Bogach N. **An Input-synchronous Blockwise Decoding Algorithm for CTC-AED Speech Recognition.**

Abstract. Automatic speech recognition (ASR) systems for real-life scenarios are required to process audio streams of arbitrary length with stable accuracy under limited computational resources. While the joint connectionist temporal classification (CTC) and attention-based encoder-decoder (AED) model delivers high recognition quality, its vanilla form is unable to meet these requirements. This paper proposes an input-synchronous blockwise decoding algorithm for the joint CTC-AED model. The algorithm processes overlapping blocks of audio synchronously with the input frames, utilizing CTC alignment to determine the proper context from the overlapping part for the AED component. The fixed block length ensures predictable and limited resource consumption and avoids long-form speech generalization issues, while the overlap mitigates WER degradation caused by edge effects. Unlike existing methods, the proposed approach requires neither model architecture modifications nor a special training procedure, while also supporting block overlapping. The word error rate (WER) performance of the algorithm is studied with respect to block size and overlap size.

Keywords: streaming automatic speech recognition (ASR), blockwise decoding, end-to-end, CTC, AED.

1. Introduction. Today, end-to-end (E2E) approaches for automatic speech recognition (ASR) have a prominent position in industrial system development and attract significant research interest. In contrast to a classical ASR system based on HMM-DNN framework [1], where acoustic and language modelling is carried out independently, an E2E system models speech integrally, which provides higher recognition quality.

There are three broad classes of E2E systems: connectionist temporal classification (CTC) [2], recurrent neural network transducer (RNN-T) [3] and attention encoder-decoder (AED) [4]. The CTC model assumes conditional independence between output tokens (language units: phones, graphemes, words). Conversely, RNN-T and AED models treat output tokens as mutually dependent, thereby achieving a higher degree of speech modelling integration. AED provides the better recognition quality than CTC or RNN-T [5] and can be extended to multilanguage [6] or multitask scenarios, e.g., speaker-attributed ASR [7]. However, the highest performance is obtained by combining CTC and AED models [8]. The joint CTC-AED model is the de facto standard for ASR systems with SOTA recognition quality.

A crucial part of any ASR system is the decoding algorithm. There are two approaches for CTC-AED model decoding [9]. The CTC model is decoded synchronously with the input sequence of feature frames. This

algorithm can be extended to CTC-AED by using AED as an additional scorer, similar to an external language model. This type of decoding is often referred to as input-synchronous (i-sync) decoding. The AED model is decoded in an autoregressive manner, i.e., synchronously with the output sequence of language tokens. Similarly, the algorithm extension for CTC-AED uses a CTC model as an additional scorer. It is also referred to as output-synchronous (o-sync) decoding.

The AED model and the succeeding CTC-AED model were initially designed for a non-streaming scenario, where the whole input audio is processed in a single run. These models have the following limitations:

1. To process audio of arbitrary length, unbounded computational resources are required. The computation time and the amount of consumed memory depend on the audio length. The quadratic time and memory complexity [10] of the transformer-based AED implementations make the problem even worse.

2. The result of recognition will be obtained only when processing is finished. There is no defined way to extract an intermediate recognition result tied to a specific time position with controlled latency.

3. AED-based models are sensitive to the lengths of input and output sequences, and may not generalize well to long-form speech recognition [11]. Especially, transformer-based AED tends to overfit to particular input length [12]. The model shows performance degradation when length of input audio does not correspond to the length distribution seen during training.

These effects are not acceptable for most of real-life commercial ASR applications, where computational resources are limited, intermediate results are required, and recognition quality should be stable. These requirements are highly related to the concept of streaming ASR system. Thus, CTC-AED model and corresponding decoding algorithm require streaming modifications to make them more suitable for commercial scenario.

Two approaches to developing streaming CTC-AED systems can be distinguished. These approaches differ in model design perspective.

The first approach involves developing a strategy to limit the input context of the attention module that is between encoder and decoder. Monotonic chunkwise attention (MoChA) [13] is one of the first conventional modifications for the RNN-based AED model. MoChA model scans encoder output to predict a frame where the next output should be generated by the decoder, and then uses a local window to the left of the selected frame to compute attention. MoChA

was also adapted for transformer-based AED [14] as a form of cross-attention. Similar methods were proposed and studied, e.g., stable monotonic chunkwise attention (sMoChA) [15], monotonic truncated attention (MTA) [15, 16], and monotonic multihead attention (MMA) [17]. These methods were initially proposed for AED model, but can be easily extended to joint CTC-AED model for both i-sync and o-sync decoding algorithms. The triggered attention (TA) [18] was developed for CTC-AED model directly and also falls under considered approach. It selects the frames required for next token prediction using CTC output. Since TA needs time alignment of CTC scores, it is a natural extension to i-sync decoding algorithm.

Methods in the first approach allow building streaming versions of CTC-AED with low latency, but they cause significant modification of training and decoding procedures, and also may cause performance degradation due to enforcing monotony and locality of attention.

The second approach is to process input audio in a blockwise manner. The input audio is split into sequential blocks, and then each block is processed by an ASR model. The non-overlapping blockwise o-sync decoding algorithms which do not require any model and training procedure modification were proposed in [19, 20]. It is shown that in o-sync setup the detection of block end requires additional processing, e.g., analysis of token repetitions [19] or counting the expected number of tokens in block using CTC [20]. Other attempts include model training with a special token, that denotes the end of the block [21]. The model is trained on overlapping blocks, which overlap that allows mitigating edge effects, e.g., when a word is split between adjacent blocks.

Methods in the second approach typically cannot provide low latency, which equals the block length. But with a sufficiently large block size, the blockwise decoding provides word error rate (WER) performance close to the non-streaming scenario, since there are no attention monotonicity restrictions within a block. Also, each block is processed uniformly, so these methods are easier to implement and can be efficiently optimized for batch processing.

This paper proposes an i-sync blockwise decoding algorithm for the CTC-AED ASR system. Its main idea is to process overlapping blocks input-synchronously, using CTC alignment to determine the proper context from the overlapping part for the AED model. It allows processing blocks with arbitrary overlap while maintaining correspondence between inference and training for the AED part. Thus, the algorithm avoids the end-of-block detection problem, requires neither model architecture modifications nor special training procedure, and supports overlapping.

The paper is structured as follows. Section 2 describes the proposed blockwise decoding algorithm, and Section 3 presents a study on the WER performance of the decoding algorithm with respect to block size and overlap size.

2. Blockwise i-sync CTC-AED decoding algorithm. This section describes the proposed algorithm. The first part describes the CTC-AED model scoring for the basic non-streaming scenario. The second part proposes scoring procedure modifications for a blockwise streaming scenario. The third part formulates the decoding scheme and the corresponding algorithm. The last part provides an optimization for the proposed scheme.

2.1. Scoring in a non-streaming scenario. In a non-streaming scenario, the CTC-AED model processes a sequence of input features $\mathbf{x}_{1:T} \in \mathbb{R}^{T \times d_x}$. Here, d_x is the dimension of features, and T is the length of input sequences. The encoder, which is part of the CTC-AED model, encodes the input features into a sequence of acoustic embeddings:

$$\mathbf{h}_{1:T'} = \mathcal{E}(\mathbf{x}_{1:T}),\ \mathbf{h}_{1:T'} \in \mathbb{R}^{T' \times d_h},$$

where d_h is the embedding dimension and T' is the length of embedding sequence. Typically, the encoder is an RNN or Transformer neural network, that is prepended with convolution layers. The convolution layers perform downsampling and, thus, $T' = \lceil T/k \rceil$, where k is a downsampling factor. To simplify the notation, let $k = 1$ and $T' = T$.

The acoustic embeddings $\mathbf{h}_{1:T}$ are used by both CTC and AED parts separately to estimate the probability of a given output token sequence $u_{1:l} \in \mathcal{U}^l$, where $\mathcal{U}$ is a token alphabet, e.g., words, phonemes, or BPE tokens, and l is the length of token sequence.

The AED part predicts the probability of the next token u_l conditioned on its history $u_{1:l-1}$ and the entire input sequence of embeddings $\mathbf{h}_{1:T}$:

$$P_{AED}(u_{1:l}|\mathbf{h}_{1:T}) = \prod_{l'=1}^{l} P_{AED}(u_{l'}|u_{1:l'-1}, \mathbf{h}_{1:T}). \tag{1}$$

The probability $P_{AED}(u_{l'}|u_{1:l'-1}, \mathbf{h}_{1:T})$ is produced by the decoder that is typically implemented as an RNN or Transformer neural network with softmax activation in the last layer.

The CTC part estimates the probability token u_l that will be emitted on a given acoustic embedding $\mathbf{h}_t$ without any conditional dependence on other tokens:

$$P_{CTC}(u_{1:l}|\mathbf{h}_{1:t}) = \sum_{a_{1:t}\in\mathcal{B}^{-1}(u_{1:l})} P_{CTC}(a_{1:t}|\mathbf{h}_{1:t})$$
$$= \sum_{a_{1:t}\in\mathcal{B}^{-1}(u_{1:l'})} \prod_{t'=1}^{t} P_{CTC}(a_{t'}|\mathbf{h}_{t'}). \quad (2)$$

In the equation above $a_{1:t} \in \mathcal{U}_\epsilon^t$, where $\mathcal{U}_\epsilon = \mathcal{U} \cup \{\epsilon\}$, is a CTC alignment between the output and input sequences. Let $\mathcal{B} : \mathcal{U}_\epsilon^t \to \mathcal{U}^l$ be the CTC mapping that converts an alignment $a_{1:t}$ to a token sequence $u_{1:l}$. The conversion consists of two steps: 1) remove consecutive duplicates, 2) remove ϵ symbols. The $\mathcal{B}^{-1}(u_{1:l})$ denotes the set of all alignments that correspond to $u_{1:l}$. The probability $P_{CTC}(a_t|\mathbf{h}_t)$ is typically predicted by a simple neural network that consists of a linear layer with softmax activation.

For i-sync decoding, the total probability of a given token sequence $u_{1:l}$ is estimated on each time step t as:

$$P_t(u_{1:l}) = P_{CTC}(u_{1:l}|\mathbf{h}_{1:t}) \cdot P_{AED}(u_{1:l}, \mathbf{h}_{1:T})^\alpha,$$

where α is an empirically chosen scale factor for the AED part. The goal of decoding is to find the best sequence $u_{1:l}$ that yields the highest probability $P_T(u_{1:l})$.

2.2. Scoring in a blockwise streaming scenario. In a blockwise streaming scenario, let the input be a sequence of B overlapping blocks of the length T_b: $\{\mathbf{x}_{b,1:T_b}\}_{b=1}^B$, where $\mathbf{x}_{b,1:T_b} \in \mathbb{R}^{T_b \times d_x}$ is the b-th block. In each block, the first T_c and last T_c frames overlap with adjacent blocks, considered as context. The central $T_b - 2\,T_c$ frames are used for processing. Thus, there is a correspondence: $\mathbf{x}_{b-1,T_b-2T_c+t} = \mathbf{x}_{b,t}$ for $t \in [1, 2\,T_c]$. To provide context and the full length for the last block, the input sequence is padded with zeros on both sides. Figure 1 illustrates padding and splitting procedures.

The encoder processes each input feature block independently producing blocks of embeddings:

$$\mathbf{h}_{b,1:T_b} = \mathcal{E}(\mathbf{x}_{1:T_b}),\ \mathbf{h}_{b,1:T_b} \in \mathbb{R}^{T_b \times d_h}.$$

However, the encoder may have mutable internal state, which allowing it to retain long-term acoustic information.

The CTC model predicts the probability for single embedding at a time. It can be easily extended to blockwise processing by substituting a concatenation into 2. This yields:

$$\mathbf{h}_{1:T} = \left[\mathbf{h}_{1,T_c:T_b-T_c}, \mathbf{h}_{2,T_c:T_b-T_c}, \ldots, \mathbf{h}_{B,T_c:T_b-T_c}\right],$$

$$\begin{aligned} &P_{CTC}(u_{1:l}|\{\mathbf{h}_{b',1:T_b}\}_{b'=1}^{b-1}, \mathbf{h}_{b,1:t}) \\ &= \sum_{a_{1:t}\in\mathcal{B}^{-1}(u_{1:l'})} \prod_{b'=1}^{b-1} \prod_{t'=1+T_c}^{T_b-T_c} P_{CTC}(a_{t'}|\mathbf{h}_{b',t'}) \prod_{t'=1+T_c}^{t} P_{CTC}(a_{t'}|\mathbf{h}_{b,t'}). \end{aligned} \quad (3)$$

For the AED model, in order to match the training procedure we must process each block in its entirety (including context) and provide the correct token history.

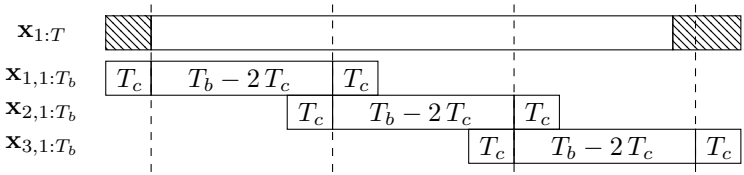

Fig. 1. Splitting input sequence into blocks. Here, the T_c is the context length, T_b is the block length. The hatching denotes the padding added to the input sequence

First, we define the AED probability for an alignment $a_{1:t}$ instead of a token sequence $u_{1:l}$:

$$P_{AED}(a_{1:t}|\mathbf{h}_{1:T}) = \prod_{t'=1}^{t} P_{AED}(a_{t'}|a_{1:t'-1}, \mathbf{h}_{1:T}), \quad (4)$$

where

$$P_{AED}(a_t|a_{1:t-1}, \mathbf{h}_{1:T}) = \begin{cases} P_{AED}(u_l|u_{1:l-1}, \mathbf{h}_{1:T}), \\ \quad \text{where } u_{1:l} = \mathcal{B}(a_{1:t}) & a_t \neq \epsilon,\ a_t \neq a_{t-1} \\ 1 & otherwise \end{cases}.$$

With this definition, it is holds that:

$$P_{AED}(u_{1:l}|\mathbf{h}_{1:T}) = P_{AED}(a_{1:t}|\mathbf{h}_{1:T}) \text{ for } a_{1:t} = \mathcal{A}_t(u_{1:l}),$$

where $\mathcal{A}_t : \mathcal{U}^l \to \mathcal{U}_\epsilon^t$ is an alignment function, that provides a valid CTC alignment $a_{1:t}$ for given tokens $u_{1:l}$, i.e., $\mathcal{A}_t(u_{1:l}) \in \mathcal{B}^{-1}(u_{1:l})$. Thus, the AED probability is tied to a particular decoding time step.

Next, we split $a_{1:T}$ into overlapping blocks $\{a_{b,1:T_b}\}_{b=1}^{B}$ in the same manner as described for the input sequence. For the overlapping part, the following holds: $a_{b-1,T_b-2T_c+t} = a_{b,t}$, $t \in [1, 2\,T_c]$. Substituting the concatenation into 4 yields:

$$a_{1:T} = \left[a_{1,T_c:T_b-T_c}, a_{2,T_c:T_b-T_c}, \ldots, a_{B,T_c:T_b-T_c}\right],$$

$$\begin{aligned}
&P_{AED}(u_{1:l}|\{\mathbf{h}_{b',1:T_b}\}_{b=1}^{b}) \\
&= P_{AED}(\{a_{b',1+T_c:T_b-T_c}\}_{b'=1}^{b-1}, a_{b,1:t}|\{\mathbf{h}_{b',1:T_b}\}_{b=1}^{b}) \\
&= \prod_{b'=1}^{b-1} \prod_{t'=1+T_c}^{T_b-T_c} P_{AED}(a_{b',t'}|a_{b',1:t'-1}, \mathbf{h}_{b',1:T_b}) \\
&\qquad \cdot \prod_{t'=1+T_c}^{t} P_{AED}(a_{b,t'}|a_{b,1:t'-1}, \mathbf{h}_{b,1:T_b}).
\end{aligned} \tag{5}$$

Here, each token is predicted by the AED model using the embeddings and history corresponding to the appropriate block.

The full score is computed at each step as:

$$P_{b,t}(u_{1:l}) = P_{CTC}(u_{1:l}|\{\mathbf{h}_{b',1:T_b}\}_{b'=1}^{b-1}, \mathbf{h}_{b,1:t}) \cdot P_{AED}(u_{1:l}|\{\mathbf{h}_{b',1:T_b}\}_{b=1}^{b})^{\alpha},$$

using Eqs. (3) and (5). These equatations are expressed in a form where the tokens $u_{1:l}$ are estimated using only the first b blocks, forming the foundation for streaming blockwise decoding algorithm.

2.3. Blockwise scoring computation scheme. The CTC decoding algorithm and, therefore, i-sync decoding algorithm are based on the recursive scheme for CTC forward probability calculation [2]. This scheme for blockwise

scenario can be formulated with respect to Eq. (3). Let the forward probability variable be:

$$\gamma_{b,t}(u_{1:l}) = P_{CTC}(u_{1:l}|\{\mathbf{h}_{b',1:T_b}\}_{b'=1}^{b-1}, \mathbf{h}_{b,1:t}).$$

It can be computed recursively using auxiliary variables $\gamma^{\epsilon}_{b,t}(u_{1:l})$ and $\gamma^{\bar{\epsilon}}_{b,t}(u_{1:l})$:

$$\gamma_{b,t}(u_{1:l}) = \gamma^{\epsilon}_{b,t}(u_{1:l}) + \gamma^{\bar{\epsilon}}_{b,t}(u_{1:l}), \tag{6}$$

$$\gamma^{\epsilon}_{b,t}(u_{1:l}) = P_{CTC}(\epsilon|\mathbf{h}_{b,t}) \cdot (\gamma^{\epsilon}_{b,t-1}(u_{1:l}) + \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l})), \tag{7}$$

$$\gamma^{\bar{\epsilon}}_{b,t}(u_{1:l}) = P_{CTC}(u_l|\mathbf{h}_{b,t}) \cdot (\gamma^{\epsilon}_{b,t-1}(u_{1:l-1}) + \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l-1}) + \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l})), \tag{8}$$

where the time index t runs from $1 + T_c$ to $T_b - T_c$ and the block index b is going from 1 to B. This way, each block is processed one by one except contexts. To provide the glue between blocks, we set:

$$\gamma^{\epsilon}_{b,T_c}(u_{1:l}) = \gamma^{\epsilon}_{b-1,T_b-T_c}(u_{1:l}) \text{ and } \gamma^{\bar{\epsilon}}_{b,T_c}(u_{1:l}) = \gamma^{\bar{\epsilon}}_{b-1,T_b-T_c}(u_{1:l}),$$

with the initial conditions:

$$\gamma^{\epsilon}_{1,T_c}(u_{1:0}) = 1 \text{ and } \gamma^{\bar{\epsilon}}_{1,T_c}(u_{1:0}) = 0.$$

An alignment $\mathcal{A}_{b,t}(u_{1:l})$, corresponding to the t-th frame of the b-block, can be constructed via a recursive backtracing procedure. It follows the previous scheme in Eqs. (6)-(8):

$$\mathcal{A}_{b,t}(u_{1:l}) = \begin{cases} \mathcal{A}^{\epsilon}_{b,t}(u_{1:l}), & \gamma^{\epsilon}_{b,t}(u_{1:l}) \geqslant \gamma^{\bar{\epsilon}}_{b,t}(u_{1:l}) \\ \mathcal{A}^{\bar{\epsilon}}_{b,t}(u_{1:l}), & \gamma^{\epsilon}_{b,t}(u_{1:l}) < \gamma^{\bar{\epsilon}}_{b,t}(u_{1:l}) \end{cases}, \tag{9}$$

$$\mathcal{A}^{\epsilon}_{b,t}(u_{1:l}) = \begin{cases} \mathcal{A}^{\epsilon}_{b,t-1}(u_{1:l}) \oplus \epsilon, & \gamma^{\epsilon}_{b,t-1}(u_{1:l}) \geqslant \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) \\ \mathcal{A}^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) \oplus \epsilon, & \gamma^{\epsilon}_{b,t-1}(u_{1:l}) < \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) \end{cases}, \tag{10}$$

$$\mathcal{A}^{\bar{\epsilon}}_{b,t}(u_{1:l}) = \begin{cases} \mathcal{A}^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) \oplus u_l, & \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) \geqslant \gamma^{\epsilon}_{b,t-1}(u_{1:l-1}) \\ & \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) \geqslant \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l-1}) \\ \mathcal{A}^{\epsilon}_{b,t-1}(u_{1:l-1}) \oplus u_l, & \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) < \gamma^{\epsilon}_{b,t-1}(u_{1:l-1}) \text{ and} \\ & \gamma^{\epsilon}_{b,t-1}(u_{1:l-1}) \geqslant \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l-1}) \\ \mathcal{A}^{\bar{\epsilon}}_{b,t-1}(u_{1:l-1}) \oplus u_l, & \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) < \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l-1}) \text{ and} \\ & \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l-1}) \geqslant \gamma^{\epsilon}_{b,t-1}(u_{1:l-1}) \end{cases}, \quad (11)$$

where the symbol $\oplus$ denotes the concatenation of an alignment and a token. Time index t and block index b have the same ranges as in the forward probability scheme. Each alignment is associated with its corresponding forward probability: $A^{\epsilon}_{b,t}(u_{1:l}) \sim \gamma^{\epsilon}_{b,t}(u_{1:l})$ and $A^{\bar{\epsilon}}_{b,t}(u_{1:l}) \sim \gamma^{\bar{\epsilon}}_{b,t}(u_{1:l})$. At each time step, the alignment is extended from the preceding alignment associated with the highest forward probability. The glue between blocks is maintained analogously forward probabilities:

$$\mathcal{A}^{\epsilon}_{b,T_c}(u_{1:l}) = \mathcal{A}^{\epsilon}_{b-1,T_b-T_c}(u_{1:l}) \text{ and } \mathcal{A}^{\bar{\epsilon}}_{b,T_c}(u_{1:l}) = \mathcal{A}^{\bar{\epsilon}}_{b-1,T_b-T_c}(u_{1:l}),$$

as well as the initial values, that are:

$$\mathcal{A}^{\epsilon}_{1,T_c}(u_{1:0}) = \emptyset \text{ and } \mathcal{A}^{\bar{\epsilon}}_{1,T_c}(u_{1:0}) = \emptyset.$$

Combining the time alignment computation scheme in Eqs. (9)-(11) with Eq. (5), we define a recursive scheme for the AED probabilities.

Let

$$\phi_{b,t}(u_{1:l}) = P_{AED}(\{a_{b',1+T_c:T_b-T_c}\}_{b'=1}^{b-1}, a_{b,1:t} | \{\mathbf{h}_{b',1:T_b}\}_{b=1}^{b}),$$

then

$$\phi_{b,t}(u_{1:l}) = \begin{cases} \phi^{\epsilon}_{b,t}(u_{1:l}), & \gamma^{\epsilon}_{b,t}(u_{1:l}) \geqslant \gamma^{\bar{\epsilon}}_{b,t}(u_{1:l}) \\ \phi^{\bar{\epsilon}}_{b,t}(u_{1:l}), & \gamma^{\epsilon}_{b,t}(u_{1:l}) < \gamma^{\bar{\epsilon}}_{b,t}(u_{1:l}) \end{cases}, \quad (12)$$

$$\phi^{\epsilon}_{b,t}(u_{1:l}) = \begin{cases} \phi^{\epsilon}_{b,t-1}(u_{1:l}) \cdot 1, & \gamma^{\epsilon}_{b,t-1}(u_{1:l}) \geqslant \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) \\ \phi^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) \cdot 1, & \gamma^{\epsilon}_{b,t-1}(u_{1:l}) < \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) \end{cases}, \tag{13}$$

$$\phi^{\bar{\epsilon}}_{b,t}(u_{1:l}) = \begin{cases} \phi^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) \cdot 1\,, & \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) \geqslant \gamma^{\epsilon}_{b,t-1}(u_{1:l-1}) \text{ and} \\ & \quad \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) \geqslant \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l-1}) \\ \phi^{\epsilon}_{b,t-1}(u_{1:l-1}) \cdot & \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) < \gamma^{\epsilon}_{b,t-1}(u_{1:l-1}) \text{ and} \\ P_{AED}(u_l | a^{\epsilon}_{b,1:t-1}, \mathbf{h}_{b,1:T_b}) & \gamma^{\epsilon}_{b,t-1}(u_{1:l-1}) \geqslant \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l-1}) \\ \phi^{\bar{\epsilon}}_{b,t-1}(u_{1:l-1}) \cdot & \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) < \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l-1}) \text{ and} \\ P_{AED}(u_l | a^{\bar{\epsilon}}_{b,1:t-1}, \mathbf{h}_{b,1:T_b}) & \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l-1}) \geqslant \gamma^{\epsilon}_{b,t-1}(u_{1:l-1}) \end{cases}, \tag{14}$$

where $a^{\epsilon}_{b,1:t-1}$ and $a^{\bar{\epsilon}}_{b,1:t-1}$ are obtained as the last $t-1$ elements of $\mathcal{A}^{\epsilon}_{b,t-1}(u_{l-1})$ and $\mathcal{A}^{\bar{\epsilon}}_{b,t-1}(u_{l-1})$, respectively. Again, for the glue between blocks

$$\phi^{\epsilon}_{b,T_c}(u_{1:l}) = \phi^{\epsilon}_{b-1,T_b-T_c}(u_{1:l}) \text{ and } \phi^{\bar{\epsilon}}_{b,T_c}(u_{1:l}) = \phi^{\bar{\epsilon}}_{b-1,T_b-T_c}(u_{1:l}),$$

and the initial values are:

$$\phi^{\epsilon}_{1,T_c}(u_{1:0}) = 1 \text{ and } \phi^{\bar{\epsilon}}_{1,T_c}(u_{1:0}) = 1.$$

The final decoding procedure is described in Algorithm 1. It is almost classic i-sync decoding procedure Eqs. (6)-(8). It differs only in special procedure for AED score estimation, which is built on Eqs. (12)-(14). AED score estimation is the proposed novel part of i-sync decoding and is presented in Algorithm 2.

At each step of Algorithm 1, a hypothesis set $\Omega_{b,t}$ is formed by expanding the hypotheses from the previous step. The forward probabilities of the hypotheses are calculation along with expansion. At the end of each step the full score $P_{b,t}(u_{1:l})$ is calculated. After the final step, the best hypothesis, denoted $\tilde{h}$, is obtained.

Algorithm 1. Blockwise input-synchronous decoding

$\Omega_{0,T_c} \leftarrow \{\{\}\}$ ▷ initialize with empty hypothesis
$\gamma^{\epsilon}_{0,T_c}(\{\}) \leftarrow 1$; $\gamma^{\bar{\epsilon}}_{0,T_c}(\{\}) \leftarrow 0$
$\phi^{\epsilon}_{0,T_c}(\{\}) \leftarrow 1$; $\phi^{\bar{\epsilon}}_{0,T_c}(\{\}) \leftarrow 1$
for $b = 1, \ldots, B$ **do**
  for $t = 1 + T_c, \ldots, T_b - T_c$ **do**
    $\Omega_{b,t} \leftarrow \emptyset$
    for $g \in \Omega_{b,t-1}$ **do**
      for $a \in U \cup \{\epsilon\}$ **do**
        if $a = \epsilon$ **then**
          $\gamma^{\epsilon}_{b,t}(g) \overset{+}{\leftarrow} P_{CTC}(a|\mathbf{h}_t) \cdot \left(\gamma^{\epsilon}_{b,t-1}(g) + \gamma^{\bar{\epsilon}}_{b,t-1}(g)\right)$
          $\Omega_{b,t} \overset{\cup}{\leftarrow} \{g\}$
          continue
        end if
        $h \leftarrow g \oplus a$ ▷ build a new hypothesis h by concatenating g and a
        if $a = g_{-1}$ **then** ▷ g_{-1} is the last symbol of g
          $\gamma^{\bar{\epsilon}}_{b,t}(g) \overset{+}{\leftarrow} P_{CTC}(a|\mathbf{h}_t) \cdot \gamma^{\bar{\epsilon}}_{b,t-1}(g)$
          $\gamma^{\bar{\epsilon}}_{b,t}(h) \overset{+}{\leftarrow} P_{CTC}(a|\mathbf{h}_t) \cdot \gamma^{\epsilon}_{b,t-1}(g)$
        else
          $\gamma^{\bar{\epsilon}}_{b,t}(h) \overset{+}{\leftarrow} P_{CTC}(a|\mathbf{h}_t) \cdot \left(\gamma^{\epsilon}_{b,t-1}(g) + \gamma^{\bar{\epsilon}}_{b,t-1}(g)\right)$
        end if
        if $h \notin \Omega_{b,t-1}$ **then**
          $\gamma^{\epsilon}_{b,t}(h) \overset{+}{\leftarrow} P_{CTC}(\epsilon|\mathbf{h}_t) \cdot \left(\gamma^{\epsilon}_{b,t-1}(h) + \gamma^{\bar{\epsilon}}_{b,t-1}(h)\right)$
          $\gamma^{\bar{\epsilon}}_{b,t}(h) \overset{+}{\leftarrow} P_{CTC}(a|\mathbf{h}_t) \cdot \gamma^{\bar{\epsilon}}_{b,t-1}(h)$
        end if
        $\Omega_{b,t} \overset{\cup}{\leftarrow} \{h\}$
      end for
    end for
    for $h \in \Omega_{b,t}$ **do** ▷ score the hypothesis set
      $\phi_{b,t}(h) \leftarrow \text{AEDScoreEstimation}(h)$
      $P_{b,t}(h) \leftarrow \left(\gamma^{\epsilon}_{b,t}(h) + \gamma^{\bar{\epsilon}}_{b,t}(h)\right) \cdot \left(\phi_{b,t}(h)\right)^{\alpha}$
    end for
    $\Omega_{b,t} \leftarrow \text{Pruning}(\Omega_{b,t})$ ▷ prune the hypothesis set
  end for
  for $h \in \Omega_{b,t}$ **do** ▷ initialize values for the next block
    $\gamma^{\epsilon}_{b+1,T_c}(h) \leftarrow \gamma^{\epsilon}_{b,T_b-T_c}(h)$; $\gamma^{\bar{\epsilon}}_{b+1,T_c}(h) \leftarrow \gamma^{\bar{\epsilon}}_{b,T_b-T_c}(h)$
    $\phi^{\epsilon}_{b+1,T_c}(h) \leftarrow \phi^{\epsilon}_{b,T_b-T_c}(h)$; $\phi^{\bar{\epsilon}}_{b+1,T_c}(h) \leftarrow \phi^{\bar{\epsilon}}_{b,T_b-T_c}(h)$
  end for
end for
$\tilde{h} \leftarrow \arg\max_{h \in \Omega_{B,T_b-T_c}} P_{B,T_b-T_c}(h)$

Algorithm 2. AED score estimation for blockwise input-synchronous decoding

if $\gamma^{\epsilon}_{b,t-1}(h) \geqslant \gamma^{\bar{\epsilon}}_{b,t-1}(h)$ **then**
  $\mathcal{A}^{\epsilon}_{b,t}(h) \leftarrow \mathcal{A}^{\epsilon}_{b,t-1}(h) \cdot \epsilon$
  $\phi^{\epsilon}_{b,t}(h) \leftarrow \phi^{\epsilon}_{b,t-1}(h)$
else
  $\mathcal{A}^{\epsilon}_{b,t}(h) \leftarrow \mathcal{A}^{\bar{\epsilon}}_{b,t-1}(h) \cdot \epsilon$
  $\phi^{\epsilon}_{b,t}(h) \leftarrow \phi^{\bar{\epsilon}}_{b,t-1}(h)$
end if
if $\gamma^{\bar{\epsilon}}_{b,t-1}(h) < \gamma^{\epsilon}_{b,t-1}(g)$ or $\gamma^{\bar{\epsilon}}_{b,t-1}(h) < \gamma^{\bar{\epsilon}}_{b,t-1}(g)$ **then**
  if $\gamma^{\epsilon}_{b,t-1}(g) \geqslant \gamma^{\bar{\epsilon}}_{b,t-1}(g)$ **then**
    $\mathcal{A}^{\bar{\epsilon}}_{b,t}(h) \leftarrow \mathcal{A}^{\epsilon}_{b,t-1}(g) \cdot a$
    $a^{\epsilon}_{b,1:t-1} \leftarrow$ last $t-1$ frames of $\mathcal{A}^{\epsilon}_{b,t-1}$
    $\phi^{\bar{\epsilon}}_{b,t}(h) \leftarrow \phi^{\epsilon}_{b,t-1}(g) \cdot P_{AED}(a|a^{\epsilon}_{b,1:t-1}, \mathbf{h}_{b,1:T_b})$
  else
    $\mathcal{A}^{\bar{\epsilon}}_{b,t}(h) \leftarrow \mathcal{A}^{\bar{\epsilon}}_{b,t-1}(g) \cdot a$
    $a^{\bar{\epsilon}}_{b,1:t-1} \leftarrow$ last $t-1$ frames of $\mathcal{A}^{\bar{\epsilon}}_{b,t-1}$
    $\phi^{\bar{\epsilon}}_{b,t}(h) \leftarrow \phi^{\bar{\epsilon}}_{b,t-1}(g) \cdot P_{AED}(a|a^{\bar{\epsilon}}_{b,1:t-1}, \mathbf{h}_{b,1:T_b})$
  end if
else
  $\mathcal{A}^{\bar{\epsilon}}_{b,t}(h) \leftarrow \mathcal{A}^{\bar{\epsilon}}_{b,t-1}(h) \cdot a$
  $\phi^{\epsilon}_{b,t}(h) \leftarrow \phi^{\bar{\epsilon}}_{b,t-1}(h)$
end if
if $\gamma^{\epsilon}_{b,t}(h) \geqslant \gamma^{\bar{\epsilon}}_{b,t}(h)$ **then**
  $\phi_{b,t}(h) \leftarrow \phi^{\epsilon}_{b,t}(h)$
else
  $\phi_{b,t}(h) \leftarrow \phi^{\bar{\epsilon}}_{b,t}(h)$
end if

2.4. An optimization of the blockwise scoring computation scheme.

In a real-world implementation of the proposed scheme for a Transformer-based CTC-AED model, the two model states are stored for each hypothesis h. A model state typically includes a key-value cache for each attention layer in the decoder, requiring a significant amount of memory. This key-value cache contains precomputed values for each token in the history. The first state corresponds to the history in $\mathcal{A}^{\epsilon}_{b,t}(h)$, and the second corresponds to the history in $\mathcal{A}^{\bar{\epsilon}}_{b,t}(h)$ (Eq. (14)). This can be optimized by keeping only one state per hypothesis.

The alignment backtracing can be simplified to maintain only a single alignment per time step, as follows:

$$\mathcal{A}_{b,t}(u_{1:l}) = \begin{cases} \mathcal{A}^{\epsilon}_{b,t}(u_{1:l}), & \gamma^{\epsilon}_{b,t}(u_{1:l}) \geqslant \gamma^{\bar{\epsilon}}_{b,t}(u_{1:l}) \\ \mathcal{A}^{\bar{\epsilon}}_{b,t}(u_{1:l}), & \gamma^{\epsilon}_{b,t}(u_{1:l}) < \gamma^{\bar{\epsilon}}_{b,t}(u_{1:l}) \end{cases}, \tag{15}$$

$$\mathcal{A}^{\epsilon}_{b,t}(u_{1:l}) = \quad \mathcal{A}_{b,t-1}(u_{1:l}) \oplus \epsilon \,, \tag{16}$$

$$\mathcal{A}^{\bar{\epsilon}}_{b,t}(u_{1:l}) = \begin{cases} \mathcal{A}_{b,t-1}(u_{1:l}) \oplus u_l, & \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) \geqslant \gamma^{\epsilon}_{b,t-1}(u_{1:l-1}) \text{ and} \\ & \quad \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) \geqslant \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l-1}) \\ \mathcal{A}_{b,t-1}(u_{1:l-1}) \oplus u_l, & \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) < \gamma^{\epsilon}_{b,t-1}(u_{1:l-1}) \text{ or} \\ & \quad \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) < \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l-1}) \end{cases}. \tag{17}$$

Substituting of Eq. (15) into Eq. (16) yields Eq. (10), therefore, in this respect, the two alignment schemes are the same. Meanwhile, substituting Eq. (15) into Eq. (17) produces a result different from Eq. (11). When $\gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) \geqslant \gamma^{\epsilon}_{b,t-1}(u_{1:l-1})$ and $\gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) \geqslant \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l-1})$ but $\gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) \leqslant \gamma^{\epsilon}_{b,t-1}(u_{1:l})$ the $\mathcal{A}^{\epsilon}_{b,t-1}(u_{1:l})$ is chosen for extension instead of $\mathcal{A}^{\bar{\epsilon}}_{b,t-1}(u_{1:l})$. This situation leads to token duplication but is considered to be very rare.

For this simplified alignment, an approximate AED score computation scheme is defined as follows:

$$\phi_{b,t}(u_{1:l}) = \begin{cases} \phi^{\epsilon}_{b,t}(u_{1:l}), & \gamma^{\epsilon}_{b,t}(u_{1:l}) \geqslant \gamma^{\bar{\epsilon}}_{b,t}(u_{1:l}) \\ \phi^{\bar{\epsilon}}_{b,t}(u_{1:l}), & \gamma^{\epsilon}_{b,t}(u_{1:l}) < \gamma^{\bar{\epsilon}}_{b,t}(u_{1:l}) \end{cases}, \tag{18}$$

$$\phi^{\epsilon}_{b,t}(u_{1:l}) = \begin{cases} \phi^{\epsilon}_{b,t-1}(u_{1:l}) \cdot 1, & \gamma^{\epsilon}_{b,t-1}(u_{1:l}) \geqslant \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) \\ \phi^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) \cdot 1, & \gamma^{\epsilon}_{b,t-1}(u_{1:l}) < \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) \end{cases}, \tag{19}$$

$$\phi^{\bar{\epsilon}}_{b,t}(u_{1:l}) = \begin{cases} \phi^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) \cdot 1\,, & \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) \geqslant \gamma^{\epsilon}_{b,t-1}(u_{1:l-1}) \text{ and} \\ & \quad \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) \geqslant \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l-1}) \\ \phi^{\epsilon}_{b,t-1}(u_{1:l-1}) \cdot & \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) < \gamma^{\epsilon}_{b,t-1}(u_{1:l-1}) \text{ or} \\ P_{AED}(u_l | a_{b,1:t-1}, \mathbf{h}_{b,1:T_b}) & \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l}) < \gamma^{\bar{\epsilon}}_{b,t-1}(u_{1:l-1}) \end{cases}, \tag{20}$$

where $a_{b,1:t-1}$ is obtained as the last $t-1$ frames of $\mathcal{A}_{b,t-1}(u_{1:l-1})$. Within this scheme, only one model state needs to be stored, corresponding to the history in $\mathcal{A}_{b,t}$.

The simplified estimation for AED score, based on Eqs. (18)-(20), is described in Algorithm 3.

Algorithm 3. Simplified AED score estimation for blockwise input-synchronous decoding

if $\gamma^{\epsilon}_{b,t}(h) \geqslant \gamma^{\bar{\epsilon}}_{b,t}(h)$ **then**
  $\mathcal{A}_{b,t}(h) \leftarrow \mathcal{A}_{b,t-1}(h) \cdot \epsilon$
  $\phi_{b,t}(h) \leftarrow \phi_{b,t-1}(h)$
else
  if $\gamma^{\bar{\epsilon}}_{b,t-1}(h) \geqslant \gamma^{\epsilon}_{b,t-1}(g)$ and $\gamma^{\bar{\epsilon}}_{b,t-1}(h) \geqslant \gamma^{\bar{\epsilon}}_{b,t-1}(g)$ **then**
    $\mathcal{A}_{b,t}(h) \leftarrow \mathcal{A}_{b,t-1}(h) \cdot a$
    $\phi_{b,t}(h) \leftarrow \phi_{b,t-1}(h)$
  else
    $\mathcal{A}_{b,t}(h) \leftarrow \mathcal{A}_{b,t-1}(g) \cdot a$
    $a_{b,1:t-1} \leftarrow$ last $t-1$ frames of $\mathcal{A}_{b,t-1}(g)$
    $\phi_{b,t}(h) \leftarrow \phi_{b,t-1}(g) \cdot P_{AED}(a|a_{b,1:t-1}, \mathbf{h}_{b,1:T_b})$
  end if
end if

3. Experiments. This section presents a study of the proposed algorithm in different scenarios and conditions with the previously trained CTC-AED model. The first part describes the data used and the details of the CTC-AED model training. The second part presents the results of algorithm evaluation.

3.1. Experimental setup. For algorithm evaluation, a CTC-AED model was trained in-house using the EspNet toolkit [22]. The model consists of 12 UCONV-Conformer encoder blocks [23] and 6 Transformer decoder blocks [10]. Each block, in both the encoder and decoder, uses 8 attention heads, has a hidden dimension of 240 and a linear layer width of 1024. The total number of parameters in the model is 30 million. The model was trained for 100 epochs using the joint CTC-AED loss [8], the Adam optimizer, and a weight decay of 1e-6. A warm-up schedule [10] with a peak learning rate of 0.002 achieved at 15th epoch was applied. The acoustic features are 80-dimensional mel-scale log filter banks, calculated using a 25ms window and a 10ms shift. The model was trained to predict positional graphemes as target tokens.

The training dataset comprises approximately 540 hours of in-house Russian speech data collected from various domains. The model was evaluated

on six in-house test sets, also from diverse domains, all with a 16 kHz sampling rate. These test sets differ in their source and, consequently, their acoustic environments. The details and statistics of the data used are presented in Table 1. Most utterances in the training datasets are approximately 30 seconds long. The test datasets were selected to provide a variety of input lengths.

Table 1. Domain and amount of experimental data

Dataset	Domain	Num. samples	Duration			
			Total, h	Mean, s	Min, s	Max, s
Train	Mixed	60494	540.12	32.14	1.00	116.47
TS1	Mid-field, noisy	1125	2.15	6.87	3.10	15.00
TS1	Far-field, quiet	180	1.63	32.53	16.86	38.30
TT1	Telephone	415	3.17	27.50	2.14	30.00
TT2	Telephone	30	1.07	128.46	60.84	278.82
TT3	Telephone	104	4.03	139.34	19.00	488.69
TT4	Telephone	86	2.20	92.07	9.47	497.66

For decoding, a C++ implementation of the proposed algorithm was used. It incorporates the optimization described Section 2.4 and is therefore defined by Algorithms 1 and 3. Pruning in Algorithm 1 is implemented by retaining a fixed number of the best hypotheses. The number is referred to as beam size.

3.2. Results of evaluation. The first series of experiments were designated to estimate the model performance in terms of WER in non-streaming scenario. The results are presented in Table 2. A significant difference is observed between short-form datasets (TS1, TS2, TT1) and long-form datasets (TT2, TT3, TT4). For long-form datasets, the best WER is achieved at lower values of α, which suggests that AED predictions are not highly relevant for this case, especially for TT4. This is attributed to the poor ability of the AED model to generalize to audio that is long relative to the training dataset [11].

The second series of experiments estimates the model's WER in the blockwise streaming scenario. The block size was chosen to be 30 seconds, which is consistent with the train dataset. The results are shown in Table 3. A WER reduction of 2-2.5% (absolute) is observed for the long-form test datasets. The proper block size during decoding allows avoiding the long-form generalization issues. The WER on short-form datasets is close to the non-streaming scenario.

In addition, the optimal α values for all datasets are close to each other and grouped around 1.2-1.4. Thus, there is no need to tune α for long-form and short-form scenarios separately.

Table 2. WER in the non-streaming scenario as a function of α. This corresponds to the case, where T_b equals the length of the input audio. The bottom row shows the best WER for the given dataset achieved across all evaluated values of α

α	TS1	TS2	TT1	TT2	TT3	TT4	Mean
0.0	36.64	41.63	33.27	24.04	23.39	41.54	33.42
0.4	33.04	38.68	30.61	**21.87**	**22.30**	**36.76**	30.54
0.6	32.22	38.45	30.09	22.06	23.31	38.48	30.77
0.8	31.85	38.14	29.69	22.91	24.99	39.49	31.18
1.0	31.88	37.80	29.47	23.37	27.10	40.77	31.73
1.2	31.58	**37.77**	**29.32**	25.67	29.79	42.10	32.71
1.4	31.38	37.86	29.32	28.21	32.67	43.94	33.90
1.6	31.46	38.37	29.43	30.24	35.81	45.60	35.15
1.8	**31.30**	39.28	29.66	32.78	38.78	47.38	36.53
best	31.30	37.77	29.32	21.87	22.30	36.76	29.89

Table 3. WER in streaming scenario with the proposed algorithm. $T_b = 30$ s, $T_c = 1$ s. The bottom row shows the best WER achieved across all evaluated values of α

α	TS1	TS2	TT1	TT2	TT3	TT4	Mean
0.00	36.60	42.54	33.45	22.78	22.92	38.98	32.88
0.40	32.68	39.16	30.37	20.77	21.14	36.42	30.09
0.60	31.79	38.73	29.96	20.43	20.86	35.80	29.59
0.80	31.31	38.71	29.67	20.21	20.54	35.36	29.30
1.00	31.29	**38.55**	29.45	19.97	20.36	35.16	29.13
1.20	31.10	38.69	29.39	19.65	20.18	34.99	29.00
1.40	31.01	38.75	**29.25**	**19.36**	20.06	**34.91**	28.89
1.60	**30.75**	39.42	29.32	19.79	19.99	35.14	29.07
1.80	30.84	41.18	29.79	19.57	**19.89**	35.31	29.43
best	30.75	38.55	29.25	19.36	19.89	34.91	28.79

The third series of experiments investigates the dependence of WER on context size T_c and block size T_b. The results are presented in Tables 4 and 5. The model without contexts, i.e., without overlap between blocks, performs, on average, 0.5-0.6 % worse, compared to a model with a context of 1 second. Overlap helps to mitigate negative effects at block edges. Increasing contexts beyond 1 second does not have a noticeable effect on performance. Decreasing the size of a whole block leads to WER performance degradation. However, there is a trade-off between WER and computational cost, as the encoder and decoder have quadratic time and memory complexity with respect to the input length. For example, decreasing the block size from 30 to 20 seconds results in a noticeable speed-up at the cost of a 0.5 % WER degradation.

Table 4. WER for the streaming scenario with the proposed algorithm as a function of T_c (measured in seconds). The block size is fixed at $T_b = 30$ seconds. The AED weight is fixed at $\alpha = 1.2$

T_c	TS1	TS2	TT1	TT2	TT3	TT4	Mean
0.0	31.58	38.94	29.32	20.94	20.71	36.27	29.63
0.4	31.24	38.76	29.02	20.11	20.20	35.19	29.09
1.0	31.10	38.69	29.39	**19.65**	20.18	34.99	29.00
2.0	**31.09**	**38.65**	**29.14**	19.73	**19.85**	**34.89**	**28.89**

Table 5. WER for the streaming scenario with the proposed algorithm as a function of T_b (measured in seconds). The block context size is fixed at $T_c = 1$ second. The AED weight is fixed at $\alpha = 1.2$

T_b	TS1	TS2	TT1	TT2	TT3	TT4	Mean
30.00	**31.10**	**38.69**	**29.39**	**19.65**	**20.18**	34.99	**29.00**
26.00	31.14	38.96	29.44	19.87	20.23	**34.92**	29.09
20.00	31.69	39.25	29.80	20.51	20.06	35.12	29.41
0.0	31.58	38.94	29.32	20.94	20.71	36.27	29.63
8.00	33.25	40.75	31.34	28.34	20.98	36.95	31.94

Table 6 demonstrates WER, achieved with the proposed algorithm in different scenarios, in comparison with the baseline non-streaming o-sync[1] and i-sync[2] decoding algorithms from EspNet toolkit.

Table 6. Comparison of the proposed algorithm (in different modes) with the non-streaming decoding algorithms from EspNet in terms of WER. The AED weight is fixed at $\alpha = 1.2$

Decoding	TS1	TS2	TT1	TT2	TT3	TT4	Mean
EspNet O-Sync, non-streaming	39.58	38.94	31.14	29.41	33.421	44.70	36.20
EspNet I-Sync, non-streaming	34.79	38.34	29.40	26.79	30.37	44.91	34.10
Proposed, non-streaming ($T_b = \inf,\ T_c = 0$ s)	31.58	37.77	29.32	25.67	29.79	42.10	32.71
Proposed, streaming ($T_b = 30$ s, $T_c = 0$ s)	31.30	38.92	29.32	20.51	20.50	36.06	29.47
Proposed, streaming ($T_b = 30$ s, $T_c = 1$ s)	31.10	38.69	29.39	19.65	20.18	34.99	29.00

[1]https://github.com/espnet/espnet/blob/v.202412/espnet/nets/beam_search.py
[2]https://github.com/espnet/espnet/blob/v.202412/espnet/nets/beam_search_timesync.py

All algorithms were run with similar settings and the same beam size of 15. In the non-streaming scenario, the proposed algorithm is close to the EspNet i-sync decoding, since they follow the same decoding scheme and differ only in implementation details. This also confirms the correctness of the studied algorithm's implementation.

4. Conclusion. This paper proposes a novel blockwise input-synchronous (i-sync) decoding algorithm for the CTC-AED model. In contrast to alternative streaming algorithms, it inherently supports arbitary-length overlap between blocks and requires no modifications to the CTC-AED architecture or its training procedure. The presented experiments show that the proposed algorithm is able to retain WER performance close to non-streaming scenario on short-form test datasets and performs significantly better on long-form test datasets, thereby avoiding the length generalization problem. It is also shown that overlap has a positive effect on WER, leading to an average decrease of 0.5% and up to 2% in a long-form scenario.

References

1. Trentin E., Gori M. A survey of hybrid ANN/HMM models for automatic speech recognition. Neurocomputing. 2001. vol. 37. no. 1-4. pp. 91–126. DOI: 10.1016/S0925-2312(00)00308-8.
2. Graves A., Fernandez S., Gomez F., Schmidhuber J. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. Proceedings of the 23rd international conference on Machine learning. 2006. pp. 369–376. DOI: 10.1145/1143844.1143891.
3. Graves A., Mohamed A.-r., Hinton G. Speech recognition with deep recurrent neural networks. IEEE International Conference on Acoustics, Speech and Signal Processing. 2013. pp. 6645–6649. DOI: 10.1109/ICASSP.2013.6638947.
4. Chorowski J.K., Bahdanau D., Serdyuk D., Cho K., Bengio Y. Attention-based models for speech recognition. Advances in neural information processing systems. 2015. vol. 28.
5. Prabhavalkar R., Rao K., Sainath T.N., Li B., Johnson L., Jaitly N. A comparison of sequence-to-sequence models for speech recognition. Proceedings of Interspeech. 2017. pp. 939–943. DOI: 10.21437/Interspeech.2017-233.
6. Li B., Pang R., Sainath T.N., et al. Scaling end-to-end models for large-scale multilingual ASR. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). 2021. pp. 1011–1018. DOI: 10.1109/ASRU51503.2021.9687871.
7. Kanda N., Ye G., Gaur Y., Wang X., Meng Z., Chen Z., Yoshioka T. End-to-end speaker-attributed ASR with transformer. Proceedings of Interspeech. 2021. pp. 4413–4417. DOI: 10.21437/Interspeech.2021-101.
8. Watanabe S., Hori T., Kim S., Hershey J.R., Hayashi T. Hybrid CTC/attention architecture for end-to-end speech recognition. IEEE Journal of Selected Topics in Signal Processing. 2017. vol. 11. no. 8. pp. 1240–1253. DOI: 10.1109/JSTSP.2017.2763455.
9. Yan B., Dalmia S., Higuchi Y., Neubig G., Metze F., Black A.W., Watanabe S. CTC alignments improve autoregressive translation. Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics. 2023. pp. 1623–1639. DOI: 10.18653/v1/2023.eacl-main.119.
10. Vaswani A., Shazeer N., Parmar N., et al. Attention is all you need. Advances in Neural Information Processing Systems (NIPS 2017). 2017. vol. 30.

11. Chiu C.-C., Han W., Zhang Y., et al. A comparison of end-to-end models for long-form speech recognition. IEEE Automatic Speech Recognition and UnderstandingWorkshop (ASRU). 2019. pp. 889–896. DOI: 10.1109/ASRU46091.2019.9003854.
12. Varis D., Bojar O. Sequence length is a domain: Length-based overfitting in transformer models. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. 2021. pp. 8246–8257. DOI: 10.18653/v1/2021.emnlp-main.650.
13. Chiu C.-C., Raffel C. Monotonic chunkwise attention. arXiv preprint arXiv:1712.05382. 2017.
14. Tsunoo E., Kashiwagi Y., Kumakura T., Watanabe S. Towards online end-to-end transformer automatic speech recognition. arXiv preprint arXiv: 1910.11871. 2019. DOI: 10.48550/arXiv.1910.11871.
15. Miao H., Cheng G., Zhang P., Yan Y. Online hybrid CTC/attention end-to-end automatic speech recognition architecture. IEEE/ACM Transactions on Audio, Speech, and Language Processing. 2020. vol. 28. pp. 1452–1465. DOI: 10.1109/TASLP.2020.2987752.
16. Miao H., Cheng G., Gao C., Zhang P., Yan Y. Transformer-based online CTC/attention end-to-end speech recognition architecture. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020). 2020. pp. 6084–6088. DOI: 10.1109/ICASSP40776.2020.9053165.
17. Inaguma H., Mimura M., Kawahara T. Enhancing monotonic multihead attention for streaming ASR. Proceedings of Interspeech. 2020. pp. 2137–2141. DOI: 10.21437/Interspeech.2020-1780.
18. Moritz N., Hori T., Le J. Streaming automatic speech recognition with the transformer model. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020). 2020. pp. 6074–6078. DOI: 10.1109/ICASSP40776.2020.9054476.
19. Tsunoo E., Kashiwagi Y., Watanabe S. Streaming transformer asr with blockwise synchronous beam search. IEEE Spoken Language Technology Workshop (SLT). 2021. pp. 22–29. DOI: 10.1109/SLT48900.2021.9383517.
20. Tsunoo E., Narisetty C., Hentschel M., Kashiwagi Y., Watanabe S. Run-and-back stitch search: Novel block synchronous decoding for streaming encoder-decoder ASR. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2022). 2022. pp. 8287–8291. DOI: 10.1109/ICASSP43922.2022.9747800.
21. Zeineldeen M., Zeyer A., Schluter R., Ney H. Chunked attention-based encoderdecoder model for streaming speech recognition. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2024-2024). 2024. pp. 11331–11335. DOI: 10.1109/ICASSP48485.2024.10446035.
22. Watanabe S., Hori T., Karita S., Hayashi, T., Nishitoba, J., Unno, Y., Enrique Yalta Soplin, N., Heymann, J., Wiesner, M., Chen, N., Renduchintala, A., Ochiai, T. ESPnet: End-to-end speech processing toolkit. Proceedings of Interspeech. 2018. pp. 2207–2211. DOI: 10.21437/Interspeech.2018-1456.
23. Andrusenko A., Nasretdinov R., Romanenko A. Uconv-conformer: High reduction of input sequence length for end-to-end speech recognition. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2023-2023). 2023. pp. 1–5.

Lezhenin Iurii — PhD student, Speech and NLP Research Group, Institute of Computer Science, Peter the Great St. Petersburg Polytechnic University (SPbPU); Leading researcher, ASR team, Speech Technology Center LTD. Research interests: machine learning, digital signal processing (DSP), speech processing. The number of publications — 24. yurij.lezhenin@gmail.com; 29B, Polytechnicheskaya St., 195220, St. Petersburg, Russia; office phone: +7(905)265-9788.

Bogach Natalia — Ph.D., Associate professor, Speech and NLP Research Group, Institute of Computer Science, Peter the Great St. Petersburg Polytechnic University (SPbPU). Research interests: digital signal processing (DSP), speech processing, linguistics. The number of publications — 29. n.v.bogach@gmail.com; 29B, Polytechnicheskaya St., 195220, St. Petersburg, Russia; office phone: +7(905)265-9788.

УДК 004.934 **DOI** 10.15622/ia.25.1.5

Ю.И. Леженин, Н.В. Богач

БЛОЧНЫЙ АЛГОРИТМ ДЕКОДИРОВАНИЯ С СИНХРОНИЗАЦИЕЙ ПО ВХОДУ ДЛЯ CTC-AED СИСТЕМ РАСПОЗНАВАНИЯ РЕЧИ

Леженин Ю.И., Богач Н.В. **Блочный алгоритм декодирования с синхронизацией по входу для CTC-AED систем распознавания речи.**

Аннотация. Для работы в реальных условиях от систем автоматического распознавания речи требуется обеспечивать стабильную точность распознавания при обработке входного аудиопотока произвольной длины в условиях ограниченных вычислительных ресурсов. Объединенная модель из коннекционисткой темпоральной классификации (connectionist temporal classification, CTC) и кодировщик-декодировщика с механизмом внимания (attention-based encoder-decoder, AED) обеспечивают высокое качество распознавания, но исходная версия модели не удовлетворяет данным требованиям. В данной статье предлагается алгоритм блочного декодирования с синхронизацией по входу для совместной модели CTC-AED. Алгоритм обрабатывает перекрывающиеся блоки аудио синхронно относительно входной последовательности признаков, используя CTC-выравнивание для определения соответствующего контекста на перекрывающемся участке для AED декодировщика. Фиксированная длина блока обеспечивает предсказуемое и ограниченное потребление ресурсов и позволяет избежать проблем с обобщением на длинных речевых сегментах, в то время как перекрытие блоков снижает ухудшение качества распознавания, вызванное краевыми эффектами на границах блоков. В отличие от других алгоритмов декодирования для CTC-AED, предложенный алгоритм не требует ни модификации архитектуры модели, ни специальной процедуры обучения, и, в то же время, поддерживает перекрытие блоков. В работе также исследуется производительность предложенного алгоритма с точки зрения доли словесных ошибок (word error rate, WER) в зависимости от размера блока и размера перекрытия.

Ключевые слова: потоковое распознавание речи, блочное декодирование, сквозные модели, AED, CTC.

Литература

1. Trentin E., Gori M. A survey of hybrid ANN/HMM models for automatic speech recognition. Neurocomputing. 2001. vol. 37. no. 1-4. pp. 91–126. DOI: 10.1016/S0925-2312(00)00308-8.
2. Graves A., Fernandez S., Gomez F., Schmidhuber J. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. Proceedings of the 23rd international conference on Machine learning. 2006. pp. 369–376. DOI: 10.1145/1143844.1143891.
3. Graves A., Mohamed A.-r., Hinton G. Speech recognition with deep recurrent neural networks. IEEE International Conference on Acoustics, Speech and Signal Processing. 2013. pp. 6645–6649. DOI: 10.1109/ICASSP.2013.6638947.
4. Chorowski J.K., Bahdanau D., Serdyuk D., Cho K., Bengio Y. Attention-based models for speech recognition. Advances in neural information processing systems. 2015. vol. 28.
5. Prabhavalkar R., Rao K., Sainath T.N., Li B., Johnson L., Jaitly N. A comparison of sequence-to-sequence models for speech recognition. Proceedings of Interspeech. 2017. pp. 939–943. DOI: 10.21437/Interspeech.2017-233.

6. Li B., Pang R., Sainath T.N., et al. Scaling end-to-end models for large-scale multilingual ASR. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). 2021. pp. 1011–1018. DOI: 10.1109/ASRU51503.2021.9687871.
7. Kanda N., Ye G., Gaur Y., Wang X., Meng Z., Chen Z., Yoshioka T. End-to-end speaker-attributed ASR with transformer. Proceedings of Interspeech. 2021. pp. 4413–4417. DOI: 10.21437/Interspeech.2021-101.
8. Watanabe S., Hori T., Kim S., Hershey J.R., Hayashi T. Hybrid CTC/attention architecture for end-to-end speech recognition. IEEE Journal of Selected Topics in Signal Processing. 2017. vol. 11. no. 8. pp. 1240–1253. DOI: 10.1109/JSTSP.2017.2763455.
9. Yan B., Dalmia S., Higuchi Y., Neubig G., Metze F., Black A.W., Watanabe S. CTC alignments improve autoregressive translation. Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics. 2023. pp. 1623–1639. DOI: 10.18653/v1/2023.eacl-main.119.
10. Vaswani A., Shazeer N., Parmar N., et al. Attention is all you need. Advances in Neural Information Processing Systems (NIPS 2017). 2017. vol. 30.
11. Chiu C.-C., Han W., Zhang Y., et al. A comparison of end-to-end models for long-form speech recognition. IEEE Automatic Speech Recognition and UnderstandingWorkshop (ASRU). 2019. pp. 889–896. DOI: 10.1109/ASRU46091.2019.9003854.
12. Varis D., Bojar O. Sequence length is a domain: Length-based overfitting in transformer models. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. 2021. pp. 8246–8257. DOI: 10.18653/v1/2021.emnlp-main.650.
13. Chiu C.-C., Raffel C. Monotonic chunkwise attention. arXiv preprint arXiv:1712.05382. 2017.
14. Tsunoo E., Kashiwagi Y., Kumakura T., Watanabe S. Towards online end-to-end transformer automatic speech recognition. arXiv preprint arXiv: 1910.11871. 2019. DOI: 10.48550/arXiv.1910.11871.
15. Miao H., Cheng G., Zhang P., Yan Y. Online hybrid CTC/attention end-to-end automatic speech recognition architecture. IEEE/ACM Transactions on Audio, Speech, and Language Processing. 2020. vol. 28. pp. 1452–1465. DOI: 10.1109/TASLP.2020.2987752.
16. Miao H., Cheng G., Gao C., Zhang P., Yan Y. Transformer-based online CTC/attention end-to-end speech recognition architecture. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020). 2020. pp. 6084–6088. DOI: 10.1109/ICASSP40776.2020.9053165.
17. Inaguma H., Mimura M., Kawahara T. Enhancing monotonic multihead attention for streaming ASR. Proceedings of Interspeech. 2020. pp. 2137–2141. DOI: 10.21437/Interspeech.2020-1780.
18. Moritz N., Hori T., Le J. Streaming automatic speech recognition with the transformer model. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020). 2020. pp. 6074–6078. DOI: 10.1109/ICASSP40776.2020.9054476.
19. Tsunoo E., Kashiwagi Y., Watanabe S. Streaming transformer asr with blockwise synchronous beam search. IEEE Spoken Language Technology Workshop (SLT). 2021. pp. 22–29. DOI: 10.1109/SLT48900.2021.9383517.
20. Tsunoo E., Narisetty C., Hentschel M., Kashiwagi Y., Watanabe S. Run-and-back stitch search: Novel block synchronous decoding for streaming encoder-decoder ASR. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2022). 2022. pp. 8287–8291. DOI: 10.1109/ICASSP43922.2022.9747800.
21. Zeineldeen M., Zeyer A., Schluter R., Ney H. Chunked attention-based encoderdecoder model for streaming speech recognition. IEEE International Conference on Acoustics,

Speech and Signal Processing (ICASSP 2024-2024). 2024. pp. 11331–11335. DOI: 10.1109/ICASSP48485.2024.10446035.

22. Watanabe S., Hori T., Karita S., Hayashi, T., Nishitoba, J., Unno, Y., Enrique Yalta Soplin, N., Heymann, J., Wiesner, M., Chen, N., Renduchintala, A., Ochiai, T. ESPnet: End-to-end speech processing toolkit. Proceedings of Interspeech. 2018. pp. 2207–2211. DOI: 10.21437/Interspeech.2018-1456.

23. Andrusenko A., Nasretdinov R., Romanenko A. Uconv-conformer: High reduction of input sequence length for end-to-end speech recognition. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2023-2023). 2023. pp. 1–5.

Леженин Юрий Игоревич — аспирант, научная группа речевых технологий и обработки естественного языка, институт компьютерных наук и кибербезопасности, Санкт-Петербургский политехнический университет Петра Великого (СПбПУ); ведущий научный сотрудник, команда распознавания речи, ООО «Центр Речевых Технологий». Область научных интересов: машинное обучение, цифровая обработка сигналов, обработка речи. Число научных публикаций — 24. yurij.lezhenin@gmail.com; улица Политехническая, 29В, 195220, Санкт-Петербург, Россия; р.т.: +7(905)265-9788.

Богач Наталья Владимировна — канд. техн. наук, доцент, научная группа речевых технологий и обработки естественного языка, институт компьютерных наук и кибербезопасности, Санкт-Петербургский политехнический университет Петра Великого (СПбПУ). Область научных интересов: цифровая обработка сигналов, обработка речи, лингвистика. Число научных публикаций — 29. n.v.bogach@gmail.com; улица Политехническая, 29В, 195220, Санкт-Петербург, Россия; р.т.: +7(905)265-9788.