

# Response to ICLR 2023 submission #2483:

## Bayes risk CTC: Controllable CTC alignment in Sequence-to-Sequence tasks

We thank the chair for organizing the peer review stage. We also thank all reviewers for acknowledging our contributions and providing us with valuable suggestions. These suggestions are constructive and do help us to improve. Below we address the concerns of each reviewer in a one-by-one style.

Update: 1) The manuscript and the implementation of BRCTC have been updated according to the reviewers' comments 2) Our response letter is also included in the complementary material.

### Response to reviewer YKWK

**Q1:** The method is compared only against vanilla CTC. However, since the proposed method effectively reduces the length of the encoder output sequence for offline recognition, it would be good to compare it with naive uniform subsampling of encoder outputs with subsampling factors 2 and 3 to achieve the same down-sampling factor.

**A:** Thanks for this question. As instructed, we implement the uniform down-sampling method over the encoder output. Experiments are conducted on Aishell-1 dataset following our hybrid CTC/attention setup in Appendix E. The results are reported in table 1. As suggested in the table, the proposed BRCTC down-sampling method outperforms the uniform down-sampling method on both DSF and CER consistently.

Exp.	DSF	Test CER
Vanilla CTC	-	4.74
Vanilla CTC + uniform down-sampling (factor=2)	0.50	4.82
Vanilla CTC + uniform down-sampling (factor=3)	0.33	4.89
BRCTC down-sampling (risk factor = 10)	0.21	4.75

Table 1: Comparison between uniform down-sampling method and the proposed BRCTC. DSF: down-sampling factor, a.k.a.,  $|\mathbf{h}'|/|\mathbf{h}|$ .

**Q2:** Furthermore, in the case of online recognition, the paper should mention CTC delay constraints from [1], which uses reference alignments obtained with an external DNN-HMM model. Comparing the BRCTC method with the delay constraints method would be nice. This experiment can be implemented with GTN, which the authors used to implement BRCTC, and delay constraint FSA similar to Kaldi TimeEnforcerFst.

**A:** We thank the reviewer for providing us with this pioneering approach. This paper has been cited and mentioned in the *related works* section. As instructed, we have reproduced the delay constraint approach and the results are reported in table 2. Our implementation is also provided in the latest complementary material (our code). The experiments are still conducted on Aishell-1 dataset. The overall latency (DCL + DL + CL) is restricted to roughly below 500 ms.

As shown in that table, we suppose the recognition performance of path constraint CTC and the proposed BRCTC are comparable given the similar overall latency budget (see the markers). Note the delay constraint CTC method requires reference alignment but our BRCTC does not, which should be considered a strength of BRCTC.

$\lambda$	DC(ms)	DCL+DL+CL(ms)	Hardware-Independent			Hardware-Dependent		CER%	Marker
			DCL (ms)	DL (ms)	DCL+DL (ms)	RTF	CL (ms)	Greedy Search	
path constraint CTC									
-	160	422	120	278	398	0.305	24	7.49	*
	240	441	120	297	417	0.305	24	7.41	*
	320	485	120	341	461	0.305	24	7.42	*
	160	479	240	211	451	0.176	28	7.06	**
	240	459	240	191	431	0.176	28	7.03	**
	320	468	240	200	440	0.176	28	7.17	**
BRCTC (ours)									
20	-	315	240	47	287	0.176	28	8.10	
		339	120	195	315	0.305	24	7.97	
		440	480	-80	400	0.128	40	7.23	*
		501	960	-517	443	0.092	58	6.40	**
		570	720	-200	520	0.106	50	6.31	

Table 2: Comparison between BRCTC (ours) and path constraint CTC. All experiments are conducted on Aishell-1 dataset and statistics are based on the test set.  $\lambda$ : the risk factor of BRCTC; DC: delay constraint of the delay constraint CTC; DCL: data collecting latency; CL: computational latency; DL: drift latency; CER: character error rate; RTF: real-time factor. \* and \*\* represent cases for comparison.

**Q3:** Abstract and introduction state that "BRCTC achieves up to 47% inference cost reduction for offline systems without degradation in transcription performance." This statement on its own is misleading. Just changing the loss function does not lead to any speed-up. The speed-up is achieved by architectural changes enabled by BRCTC. Please clarify this in the paper.

**A:** We thank the reviewer for helping us to improve the writing. We suppose the speed-up mainly results from the trimming process. Thus, the revisions are listed below.

In abstract: *Experimentally, the proposed BRCTC, along with a trimming approach, enables us to reduce the inference cost of offline models by up to 47% without performance degradation.*

In introduction: *Experimentally, BRCTC can cooperate with a trimming approach to achieve up to 47% inference cost reduction for offline systems without degradation in transcription performance.*

**Q4:** The use of  $2u$  in eq. 8 is unintuitive at first read and needs to be explained in the main body of the paper. Please explain that  $2u$  is used because  $l_u = l'_{2u}$  as you do in Appendix C.

**A:** We thank this suggestion to help our writing be clearer. We have added the notation  $l_u = l'_{2u}$  in the main text.

**Minor Comments:** We thank the reviewer for checking our paper so carefully. All mentioned minor problems have been fixed accordingly.

# Response to reviewer TGbL

**Q1:** For online experiments, I think CTC-attention training is unnecessary. It's better to show pure CTC training results.

**A:** We thank this comment. As instructed, we have conducted similar online experiments as in section 4.4 with the attention decoder removed. The results are reported in table 3 and figure 1 below. Our observations on this group of experiments are consistent with what is claimed in our paper: 1) with a similar latency budget, the BRCTC online method achieves better performance-latency trade-off; 2) the BRCTC achieves the very low overall latency that cannot be achieved by vanilla CTC.

$\lambda$	DCL+DL+CL(ms)	Hardware-Independent			Hardware-Dependent		CER%	Marker
		DCL (ms)	DL (ms)	DCL+DL (ms)	RTF	CL (ms)	Greedy Search	
	Aishell-1 test - Pure CTC							
0	614	80	513	593	0.263	21	8.14	★★
	529	160	343	503	0.163	26	7.24	★
	512	320	157	477	0.112	35	6.88	★
	613	480	86	566	0.099	47	6.43	★★
	771	640	76	716	0.086	55	6.18	
10	293	80	192	272	0.263	21	9.98	▲
	339	160	153	313	0.163	26	8.32	▲
	347	320	-8	312	0.112	35	8.14	▲
	517	480	-10	470	0.099	47	6.88	★
	630	640	-65	575	0.086	55	6.18	★★

Table 3: Trade-off between the transcription performance and the latency for online CTC models.  $\lambda$ : the risk factor of BRCTC; DCL: data collecting latency; CL: computational latency; DL: drift latency; CER: character error rate; RTF: real-time factor. ★ and ★★ represent cases for comparison. ▲ represents the extremely low latency cases that cannot be achieved by vanilla CTC.

**Q2:** Some are some variants of transducer like (alignment-restricted transducer, pruned Transducer) are also based on modification of forward-backward algorithm to achieve specific properties (latency, computational cost). It's better to cite these papers.

**A:** We thank the reviewer for providing us with these papers and agree they are related. The papers below are cited in the *related works* section.

- J. Mahadeokar et al., "Alignment Restricted Streaming Recurrent Neural Network Transducer," 2021 IEEE Spoken Language Technology Workshop (SLT), 2021, pp. 52-59, doi: 10.1109/SLT48900.2021.9383606.
- Kuang, F., Guo, L., Kang, W., Lin, L., Luo, M., Yao, Z., Povey, D. (2022) Pruned RNN-T for fast, memory-efficient ASR training. Proc. Interspeech 2022, 2068-2072, doi: 10.21437/Interspeech.2022-10340
- Shinohara, Y., Watanabe, S. (2022) Minimum latency training of sequence transducers for streaming end-to-end speech recognition. Proc. Interspeech 2022, 2098-2102, doi: 10.21437/Interspeech.2022-10989
- J. Yu et al., "FastEmit: Low-Latency Streaming ASR with Sequence-Level Emission Regularization," ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 6004-6008, doi: 10.1109/ICASSP39728.2021.9413803.
- Kim, J., Lu, H., Tripathi, A., Zhang, Q., Sak, H. (2021) Reducing Streaming ASR Model Delay with Self Alignment. Proc. Interspeech 2021, 3440-3444, doi: 10.21437/Interspeech.2021-322

**Q3:** For the limitation discussion, the author mentioned the increasing training cost of BRCTC but didn't provide much details.

**A:** We thank this comment. With the identical settings in Appendix E, we compare the training cost of Hybrid CTC/Attention systems on Aishell-1 dataset. The results are reported in table 4. The adoption of BRCTC results in 3.9% computational overhead.

	Attention + Vanilla CTC	Attention + BRCTC
Second / Epoch	883	918 (+3.9%)

Table 4: Training cost comparison between vanilla CTC and BRCTC on Aishell-1 dataset.

# Response to reviewer NBYg

**Q1:** The work does not specify a general condition that the grouping function  $f(\phi)$  has to meet to enable the forward-backward algorithm to be applied. Instead, it only gives an example  $f(\phi)$  which is the largest index of non-blank elements in the path. If it is the only case to which the proposed method can be applied, the main value of the work is to speed up CTC inference. Although accelerating CTC inference is also important, the writing of the paper gives readers the impression that it is a general method to control CTC alignment.

Making CTC output controllable alignment prediction is an interesting topic. The proposed method is novel and technically sound. My main concern is whether the proposed method is applicable to other useful grouping functions except the example given in Sec. 3.2 and if there is a general rule for making a legal grouping function. However, the paper does not include any discussion on it.

**A:** We thank the reviewer for acknowledging our work is novel, well-motivated, technically sound, and well-justified by experiments. We also thank the reviewer for helping us clarify our work. Below we explain why the proposed BRCTC is exactly a general method and has the potential to be widely adopted, besides accelerating CTC inference. Firstly, we discuss the compatibility between the forward-backward algorithm and our BRCTC. Secondly, we explain why the BRCTC is a general and flexible method by providing multiple grouping strategies and risk strategies. We finally show another potential application of BRCTC besides acceleration.

**Compatibility between the forward-backward algorithm and our BRCTC:** We agree with the reviewer that we should clarify when the BRCTC is compatible with the forward-backward algorithm. To our knowledge, the only condition to make the BRCTC compatible with the forward-backward algorithm is about the grouping strategy: the summed posterior of all paths within any path group should be computationally feasible using the forward-backward variables, i.e.,  $\sum_{\pi \in \mathcal{B}^{-1}(\mathbf{1}), f(\pi)=\tau} p(\pi|\mathbf{x})$  can be represented by combining  $\alpha$  and  $\beta$ . Although vanilla CTC only considers a special case (equation 4), the BRCTC formulation can be generalized to other grouping strategies, like equation 8. Owing to the differentiability of  $\alpha$  and  $\beta$  w.r.t the CTC posteriors, the grouping strategies can be designed in a more flexible way without worrying about the gradient computation. For better clarification, the following sentence is added in section 3.1: *Here the only requirement to achieve compatibility between BRCTC and the forward-backward process is that the summed posterior within any group can be fully represented by the forward-backward variables.*

**Flexibility from grouping strategy design:** We agree with the reviewer that we should provide more than one grouping strategy examples to claim the generalization. Actually, in section 3.2, we provide two grouping strategy examples: 1) considering the state occupation, like in equation 4; 2) considering the ending point of a given non-blank token, like in equation 8. The latter has been applied to sections 3.3 and 3.4 and justified by experiments. Although the former is less emphasized in our paper, one of its potential applications will be discussed in the last second paragraph (other potential applications) of this response. Besides the two strategies above, the proposed BRCTC does have other possibilities in grouping strategy design. A simple example is that, with a fixed  $t$ , all paths can be divided into two groups concerning whether  $\pi_t$  is blank or not:  $\sum_{\pi \in \mathcal{B}^{-1}(\mathbf{1}), \pi_t=\emptyset} p(\pi|\mathbf{x})$  and  $\sum_{\pi \in \mathcal{B}^{-1}(\mathbf{1}), \pi_t \neq \emptyset} p(\pi|\mathbf{x})$ . Theoretically, this grouping strategy has the potential to encourage or discourage the blank emission at frame  $t$ . We have revised section 3.2 to clarify there are two grouping strategy examples.

**Flexibility from risk strategy design:** We would like to note that the flexibility of BRCTC is not only on the grouping strategy design  $f(\pi)$  but also on the group risk value design  $r_g(\tau)$ . Even with an identical grouping strategy, the BRCTC can serve different purposes by changing the group risk value designs. The example is that, although the grouping strategy in equation 8 is consistently adopted in both sections 3.3 and 3.4, different goals (down-sampling and latency reduction) are achieved due to the different group risk value designs.

**Other potential applications:** The paragraphs above have discussed the generalization and flexibility of BRCTC from the perspectives of both grouping strategy design and risk strategy design. We agree with the reviewer that discussing more potential directions for BRCTC applications will be beneficial. Thus, we provide another example of BRCTC's potential applications: it has the potential to calibrate the CTC alignment prediction. Although in sections 3.3 and 3.4 we only design the  $r_g(\tau)$  without any external information, this design can be more flexible if the external information (e.g., reference alignment) is provided. Specifically, we may calibrate the CTC alignment by 1) adopting the grouping strategy concerning the state occupation, like in equation 4, and 2) encoding the reference alignment information into the  $r_g(\pi)$ . With this setting, our preliminary experiments suggest that, compared with vanilla CTC, the predicted alignment can achieve better agreement with its DNN-HMM reference (see the figure below).

**Conclusion:** In this response, we have clarified the general condition to make the BRCTC compatible with the forward-backward algorithm. In addition, we provide multiple grouping strategy designs and group risk value designs. Finally, we describe a potential application (CTC alignment calibration) using BRCTC. Concerning the statements above, we are attempting to convince the reviewer that the proposed BRCTC is a general alignment control method rather than a special case study. It is obvious that the design of BRCTC will not be limited to what we mentioned in our paper and in this response, but we regret that we cannot enumerate each of these potential possibilities. Instead, we would like to encourage our readers to design their strategies from their own orientations.

# Response to reviewer GviL

**Q1:** There shouldn't be a 't' in the subscript for 'p' in equation (2)

**A:** We thank the reviewer for pointing out this. This typo has been fixed.

**Q2:** I may be missing something here, but shouldn't the final result in equation (5) be a product over all 't's.

**A:** Thanks for asking this. We believe that summing over all  $t$  in equation 5, with a fixed  $v$ , is not valid. An implicit requirement of this summing process is that each path should be considered once and only once. Firstly, with a fixed  $v$ ,  $\pi_t = l'_v$  does not exclude the possibility of  $\pi_{t+1} = l'_v$ , which means some paths are counted more than once when we are enumerating all  $t$ . Secondly, when  $l'_v$  is a blank symbol, it is not guaranteed that  $l'_v$  will be included in every path, which means some paths are not counted. To explain more, equation 8 is a valid solution because each path groups are exclusive and we only consider the situation when  $l_u = l'_{2u}$  is a non-blank token.

**Q3:** The arg max operation defined in the first paragraph of page 5 is not very clear. There is a boolean operation inside the argmax(.) where elements will be either 0 or 1, what will argmax return in such a case?

**A:** We thank this comment. The  $\tau = f_u(\pi) = \arg \max_t (\pi_t = l_u)$  has been replaced by  $\tau = f_u(\pi) = \arg \max_t \text{ s.t. } \pi_t = l_u$  to exclude this ambiguity.

**Q4:** Typo: page 6 last line, DFS  $\rightarrow$  DSF

**A:** Thanks for reminding us of this. The typo has been fixed.

**Q5:** Downsampling: Using a CNN for subsampling a speech sequence such that the new sequence length is similar to the one obtained by the controlled alignment. The CNN is like a feature extractor from speech. For example, wav2vec2.0 uses 20ms windows, what happens if we use a >20ms window and lower the sequence length? How much performance is degraded and how does it compare with the proposed model?

**A:** Thanks for this question. To explore the effect of a larger CNN down-sampling factor (DSF), the CNN down-sampling factor is changed from 4 (as in our original setup) to 8 (a large number that is rarely used in practice). The experiment is conducted on Aishell-1 dataset and the results are presented in table 5. As suggested in the table: 1) the adoption of larger CNN down-sampling factor results in a slight performance degradation; 2) the adoption of the BRCTC down-sampling method can effectively reduce the encoder output length regardless of the CNN down-sampling factors.

Hint: we still implement the CNN down-sampling using 2D-CNN over the Fbank features. We have not implemented the down-sampling by applying 1D-CNN over the waveform (like in wav2vec 2.0) in order to keep consistency with our initial setup.

CNN stride	CNN DSF	Time per frame in <b>h</b>	BRCTC down-sampling	<b> h </b>	<b> h' </b>	Test CER
{2,2}	4	40ms	False	124.1	-	4.74
			True		29.2	4.75
{2,2,2}	8	80ms	False	61.3	-	4.79
			True		24.4	4.78

Table 5: Comparison of different CNN down-sampling factors

**Q6:** Online Latency reduction: Plantinga et al. [1] have a simple solution for emitting phones sooner where they have an "alignment loss which encourages outputs only when features do not resemble silence." It would also be a good idea to cite the above paper.

**A:** We thank the reviewer for providing us with this paper. We agree this paper is related and have added it into the *related works* section.

**Q7:** Can such an alignment controlling mechanism be used to train RNN transducers which are also very popular and use a quite similar algorithm to compute the loss?

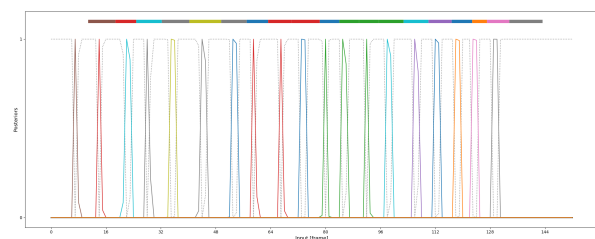
**A:** We thank the reviewer for providing us with this additional inspiration. Indeed, it is possible to generalize the proposed method to the neural transducer since both CTC and transducer are maximizing the summed posterior of all valid paths. We prefer to address this problem in our future work.

**Q8:** What are some other applications apart from the two explored where controlling CTC alignment will be useful?

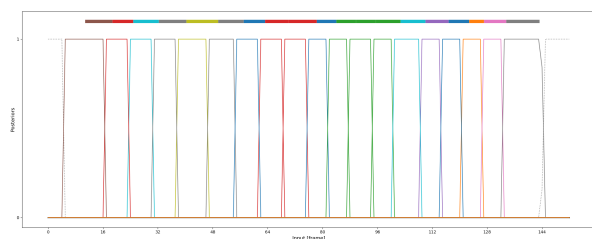
**A:** We thank the reviewer for encouraging us to explore more possible applications of BRCTC.

Although in sections 3.3 and 3.4 we only consider some heuristically designed group risk values  $r_g(\tau)$ , it would be possible to design these values by encoding some external supervision information. An example is to encode the reference alignment: by adopting the occupation condition (like equation 4) as the grouping strategy and adopting group risk values that encode the external reference alignment, we attempt to obtain accuracy in both recognition results and alignment prediction. Our preliminary attempts suggest this direction might be possible: as shown in the figure below, the adoption of BRCTC with a proper risk design can remarkably alleviate the disagreement between the predicted alignment and its reference.

We would also like to remind the reviewer that the BRCTC is proposed as a general framework and the risk design is left customizable. Instead of enumerating all possible formulations, we encourage our users to design the exact Bayes risk function from their own orientations and task-specific needs.



Vanilla CTC



BRCTC with alignment calibration (ours)

Figure 1: An intuitive explanation of alignment calibration through BRCTC. Colored bars are the reference alignment obtained by DNN-HMM systems.