IKFST: IOO and KOO Algorithms for Accelerated and Precise WFST-based End-to-End Automatic Speech Recognition

Anonymous ACL submission

Abstract

End-to-end automatic speech recognition has become the dominant paradigm in both academia and industry. To enhance recognition performance, the Weighted Finite-State Transducer (WFST) is widely adopted to integrate acoustic and language models through static graph composition, providing robust decoding and effective error correction. However, WFST decoding relies on a frame-by-frame autoregressive search over CTC posterior probabilities, which severely limits inference efficiency. Motivated by establishing a more principled compatibility between WFST decoding and CTC modeling, we systematically study the two fundamental components of CTC outputs, namely blank and non-blank frames, and identify a key insight: blank frames primarily encode positional information, while non-blank frames carry semantic content. Building on this observation, we introduce Keep-Only-One and Insert-Only-One, two decoding algorithms that explicitly exploit the structural roles of blank and non-blank frames to achieve significantly faster WFST-based inference without compromising recognition accuracy. Experiments on large-scale in-house, AISHELL-1, and LibriSpeech datasets demonstrate state-of-the-art recognition accuracy with substantially reduced decoding latency, enabling truly efficient and high-performance WFST decoding in modern speech recognition systems.

1 Introduction

Recent advances in deep neural networks have substantially improved automatic speech recognition (ASR) (Sainath et al., 2015; Dong et al., 2018; Wu et al., 2023b; Zhang et al., 2024a; Zhou et al., 2024). Traditional ASR systems rely on multi-stage pipelines with expert-designed components (Juang and Rabiner, 1991), while end-to-end (E2E) models simplify this process by directly mapping speech to text (Prabhavalkar et al., 2024). E2E ASR methods mainly include CTC (Graves et al., 2006; Amodei et al., 2016; Yao et al., 2025), RNN-T (Graves, 2012; Graves and Jaitly, 2014), and attention-based encoder–decoder models (Chan et al., 2016; Zhang et al., 2023a,b), as well as recent hybrid formulations such as CTC/RNN-T and CTC/AED (Watanabe et al., 2017; Kim et al., 2017; Zhang et al., 2024b). In these systems, external language models are commonly integrated during inference to enhance linguistic consistency and domain adaptation, typically through N-best re-ranking or non-autoregressive re-scoring methods (Chorowski and Jaitly, 2017; Kannan et al., 2018; Sainath et al., 2021; Yao et al., 2021).

Among various decoding frameworks, the Weighted Finite-State Transducer (WFST) has long served as a foundational component in large-scale ASR systems, owing to its strong theoretical grounding, flexible topology composition, and robust decoding capability (Mohri et al., 2000; Hori and Nakamura, 2022; Lv et al., 2021). Within this framework, CTC models provide frame-level posterior probabilities, which are then decoded by a modified WFST through Viterbi beam search (Miao et al., 2015; Laptev et al., 2022; Povey et al., 2016). Despite its effectiveness and widespread adoption, directly performing WFST decoding over the full posterior sequence remains computationally prohibitive, posing a critical challenge to efficient and scalable deployment. To alleviate the aforementioned issues, Chen et al (Chen et al., 2016) proposed the Lattice-based Phone Synchronous Decoding (LSD) algorithm. Speech-LLaMA (Wu et al., 2023a) proposed an averaging strategy to preserve this latent information. PolyVoice (Dong et al., 2023) systematically discards all blank frames. Building on similar insights, the Spike Window Decoding (SWD) algorithm further refined this concept by selectively incorporating a limited neighborhood of blank frames surrounding high-probability non-blank spikes (Zhang et al., 2025).

From the collective insights of prior research, a unifying observation arises: even after pruning a significant number of blank frames, the model can deliver competitive or even superior recognition results in the WFST decoding. This observation motivates a deeper investigation into the intrinsic positional and semantic characteristics of blank and non-blank frames, aiming to uncover the minimal yet sufficient frame representation for optimal decoding. Building on this perspective, we introduce two complementary algorithms: Insert-Only-One (IOO) and Keep-Only-One (KOO). The IOO algorithm operates by first discarding all probabilistic, model-learned blank frames and then strategically inserting a deterministic, user-defined blank frame between adjacent non-blank frames thereby preserving crucial transitional cues while simultaneously eliminating superfluous temporal redundancy. In parallel, the KOO algorithm addresses redundancy within the non-blank domain by selectively retaining only one representative spike per activation cluster, effectively compressing the posterior sequence without degrading its semantic fidelity. Notably, the IOO algorithm is broadly applicable, enhancing the performance of both CTC-FST and AED-FST decoding.

We conduct an extensive evaluation of the proposed KOO and IOO algorithms across diverse datasets to thoroughly assess their effectiveness and generalization. Experiments are performed on the widely adopted AISHELL-1 Mandarin dataset (Bu et al., 2017), the English Librispeech dataset (Panayotov et al., 2015), as well as on a large-scale 65,000-hours In-House dataset. We first construct a CTC/AED hybrid acoustic model combined with a GPU-accelerated WFST decoding framework (Daniel and Kaldewey, 2023), forming a robust foundation that achieves state-of-the-art (SOTA) recognition accuracy. Building on this, the proposed methods demonstrate significant gains in both inference speed and recognition performance. These results validate the approach's capability to deliver efficient, high-accuracy decoding across diverse linguistic contexts and data scales.

2 Related Work

2.1 CTC-based ASR Model

CTC is a widely adopted objective for end-to-end ASR due to its ability to train models without requiring frame-level alignments. For a given an acoustic feature sequence X, we define Y as the corresponding label sequence, which has a length of L. the encoder produces a sequence of hidden representations:

$$H_{encoder} = \text{Encoder}(X). \tag{1}$$

CTC introduces an intermediate alignment sequence by allowing blank symbols and repeated tokens, enabling flexible many-to-one mappings between acoustic frames and output labels. Let $\mathcal{B}(\cdot)$ denote the mapping that removes blanks and repeated symbols; the CTC objective is:

$$\mathcal{L}_{CTC} = -\log \sum_{Z \in \mathcal{B}^{-1}(Y)} p(Z \mid H_{encoder}), \tag{2}$$

where the probability of an alignment path Z is modeled under conditional independence assumptions across time steps.

2.2 Hybrid CTC/AED Algorithm

CTC is frequently combined with an attention-based encoder–decoder (AED). The AED decoder conditions on both encoder representations and previously generated tokens:

$$H_{decoder} = \text{Decoder}(H_{encoder}, Y), \tag{3}$$

and is trained using cross-entropy:

$$\mathcal{L}_{AED} = \text{CrossEntropy}(H_{decoder}, Y). \tag{4}$$

The hybrid objective interpolates the two losses:

$$\mathcal{L} = \alpha\, \mathcal{L}_{CTC} + (1 - \alpha)\, \mathcal{L}_{AED}, \tag{5}$$

where α is a hyper-parameter used to adjust the weight ratio between the encoder and decoder, with its value ranging from $[0, 1]$, which is routinely configured to 0.1 for the rest of the study.

2.3 WFST based decoding algorithm

We integrate a WFST-based decoding module on the encoder side and construct a static decoding graph following the standard TLG composition framework. The overall graph is obtained through the sequential composition of the token, lexicon, and grammar transducers:

$$TLG = T \circ \min(\det(L \circ G)), \tag{6}$$

where L and G denote the lexicon FST and the grammar FST, respectively. The operators $\det$,

Figure 1: (a) represents a search performed using dense frames; (b)-(d) and (e)-(f) illustrate the use of the IOO algorithm on blank frames and non-blank frames, respectively; (g) preserves the frame with the highest posterior probability, whereas (h) retains the frame with the lowest posterior probability; (i) denotes the application of the KOO algorithm based on probability; (k) indicates the integration of AED-FST with the IOO algorithm.

min, and $\circ$ correspond to determinization, minimization, and WFST composition. It is worth noting that we adopt the GPU-accelerated WFST decoder and the compact token transducer T proposed in (Daniel and Kaldewey, 2023), which substantially reduces the state-space complexity of graph construction—from second-order exponential to linear, while preserving competitive decoding accuracy.

3 Methodology

In this study, we investigate the integration of CTC posterior probabilities with FST to achieve optimal trade-offs between inference speed and recognition accuracy. We systematically analyze the distinct characteristics of blank and non-blank outputs within the CTC posterior, and based on this, propose two novel algorithms: Insert-Only-One (IOO) and Keep-Only-One (KOO). By combining these two algorithms, we demonstrate a significant improvement in both processing speed and recognition performance. Notably, using the IOO algorithm is an effective way to mitigate the severe drop in accuracy for AED-FST.

3.1 IOO and KOO Algorithms

In the CTC framework, the encoder produces a sequence of logits $H_{encoder}$, which represent the raw, unnormalized token scores at each timestep. To obtain the corresponding posterior logits P, the softmax function is applied over the vocabulary dimension:

$$P = \text{SoftMax}(H_{encoder}). \tag{7}$$

This yields a probability distribution over all possible tokens which including the blank symbol at every frame. As illustrated in Fig. 1 (a), the posterior distribution produced by the CTC model exhibits a pronounced spike-like pattern, characterized by a large number of high-probability blank predictions interspersed with token activations (represented in the figure using two example token classes A and B). When the decoding process is performed in a dense manner, that is, by providing the complete CTC posterior sequence directly to the FST, the decoding efficiency degrades substantially. This degradation is caused by the overwhelming number of blank-dominated frames, which greatly expand

the effective search space and introduce significant computational overhead during FST traversal.

Therefore, the IOO algorithm is introduced to mitigate this issue. As illustrated in Fig.1 (b), the IOO algorithm performs a controlled insertion of customized blank frames. This strategy preserves the desirable temporal separation provided by blank symbols while simultaneously alleviating the decoding inefficiency caused by densely occurring blank regions in the original CTC posterior sequence. Specifically, given the CTC posterior sequence P, the IOO algorithm removes all original blank frames and replaces each position containing one or more consecutive blank predictions with a single customized blank distribution. Each customized blank frame is defined as a one-hot probability vector in which the blank token (indexed as 0) is assigned a posterior probability of 1.0, while all remaining token probabilities are set to 0.0. Formally, the inserted blank frame is given by:

$$p_{blk} = [1.0,\ 0.0,\ 0.0,\ \ldots,\ 0.0] \in \mathbb{R}^{|\mathcal{V}|}, \quad (8)$$

where $|\mathcal{V}|$ denotes the size of vocabulary. This replacement not only eliminates the inefficiency associated with densely occurring blank segments but also introduces a consistent and deterministic blank representation that improves the stability of FST decoding. The final IOO-enhanced posterior sequence is denoted as P_{IOO}.

Complementary to the proposed IOO algorithm, which optimizes redundant blank frames, we further introduce the KOO algorithm to refine the non-blank components of the CTC posterior sequence. Given the frame-level probability sequence P_{IOO}, the KOO algorithm identifies non-blank regions directly along the original temporal axis. Let the predicted token at frame t be denoted by $c_t = \arg\max_{v\in\mathcal{V}} P_{t,v}$. After that, the non-blank blocks B are then constructed by grouping consecutive timesteps that share the same non-blank prediction:

$$B_k = \left\{ t_k^{\text{start}}, t_k^{\text{start}}+1, \ldots, t_k^{\text{end}} \right\}, \quad (9)$$

$$\text{s.t. } c_t = c_k \neq \text{blank},\ \forall t \in B_k, \quad (10)$$

where the index k enumerates the non-blank blocks in temporal order, and t_k^{start} corresponds to the first timestep belonging to the k-th block. Importantly, KOO must operate on the full sequence $(c_1, c_2, ..., c_T)$ without removing blank frames, discarding blank frames prematurely could artificially merge two non-adjacent frames and produce incorrect temporal adjacency.

As shown in the Fig.1 (g) and Fig.1 (h), for each block B_k, KOO selects a single representative frame according to either the maximum- or minimum-probability strategy:

$$f^* = \begin{cases} \arg\max\limits_{t\in B_k} P_{t,c_k}, & \text{max strategy}, \\ \arg\min\limits_{t\in B_k} P_{t,c_k}, & \text{min strategy}. \end{cases} \quad (11)$$

Collecting the logits associated with the selected indices produces the refined non-blank sequence:

$$L_{\text{nb}} = \left[P_{f_1^*}, P_{f_2^*}, \ldots, P_{f_k^*} \right] \in \mathbb{R}^{K\times|\mathcal{V}|}. \quad (12)$$

The resulting sequence L_{nb} preserves the temporal structure of the original CTC output while eliminating the cumulative errors introduced by repeated non-blank frames within the FST decoding process. By retaining only a single representative frame for each non-blank region, the sequence enables a more accurate and reliable decoding outcome.

When applying KOO to the IOO-processed probability sequence P_{IOO}, the final posterior P_{final} combines positional information from IOO and semantic information from KOO, which preserves K non-blank frames. Since the IOO algorithm inserts at most one customized blank frame after each retained non-blank frame, the length of the final sequence satisfies:

$$|P_{\text{final}}| \leq 2K + 1. \quad (13)$$

Because K is significantly smaller than the original number of timesteps T in the CTC posterior sequence, the number of frames that participate in the subsequent FST decoding is drastically reduced. As a result, the total number of FST search steps decreases proportionally, leading to substantially improved decoding efficiency. The pseudo code for the proposed IOO and KOO procedures for CTC-FST is provided in Algorithm 1 (Appendix A) .

3.2 AED-IOO Integration

Although AED-based speech recognition models have recently emerged as a dominant paradigm and have achieved state-of-the-art performance on many benchmarks, our experiments reveal an important limitation: directly feeding the raw AED posterior sequence into an FST-based decoder yields unsatisfactory results. We believe that, despite the absence of an explicit blank concept in

AED models, the FST search process still relies on a form of temporal separation between adjacent frames to operate effectively.

Motivated by this observation, we extend the IOO procedure to AED by inserting a customized blank token after every decoder-generated frame. The resulting output sequence is illustrated in Fig.1 (k). In practical terms, this corresponds to augmenting the AED posterior sequence with a deterministic blank distribution—identical in form to the custom blank used on the CTC side. Empirically, this modification consistently improves recognition accuracy, demonstrating that the introduction of blank-induced temporal separation provides a beneficial regularization effect for FST decoding, even in AED frameworks that do not natively employ blank symbols. The pseudo code for the proposed IOO algorithm for AED-FST is provided in Algorithm 2 (Appendix B) .

3.3 TLG graph optimization

The conventional approach for constructing CTC-based TLG decoding graphs is summarized in Eq. 6. However, when applied to large-scale language models, this construction process often results in an exceedingly large TLG graph, which in turn leads to substantial memory consumption, increased storage requirements, and a more costly inference procedure. To address these challenges, the present work incorporates a weight-pushing step into the graph-building pipeline, positioned between the $\det$ and $\min$ operations. By shifting the path weights toward earlier states, this operation effectively enables early pruning of low-probability paths without sacrificing decoding accuracy. As a result, both the effective search space and the runtime efficiency are significantly improved. The resulting static graph construction procedure is formalized as follows:

$$TLG = T \circ \min(\text{pushing}(\det(L \circ G))) \quad (14)$$

4 Experiments

4.1 Dataset

We conduct experiments on LibriSpeech, AISHELL-1, and large-scale In-House datasets. LibriSpeech is a 960-hour English corpus, while AISHELL-1 is a 178-hour Mandarin corpus recorded at a sampling rate of 16 kHz. Our in-house dataset is also a Mandarin speech corpus, recorded at 8 kHz. It contains approximately 65,000 hours of labeled training data, and the test set consists of around 6,000 randomly sampled short utterances from both inbound and outbound telephone calls.

4.2 Experimental settings

4.2.1 Acoustic model settings

The acoustic model constructed in this work is a multi-task Hybrid CTC/AED structure, in which the encoder and decoder models are based on the Zipformer and transformer models, respectively. On the all datasets, the encoder follows the large size Zipformer in (Yao et al., 2023), the decoder side is based on the transformer standard model, which has 6 transformer blocks. The final size of the acoustic model is 0.22B parameters, and the key encoder-related configurations are summarized in Table 3 (Appendix C) .

4.2.2 Language model settings

For the AISHELL-1 and LibriSpeech datasets, the language models are constructed exclusively from the text in their respective training sets. Specifically, 5-gram language models are trained using the SRILM toolkit[1]. On the In-House dataset, we use about 43 million pieces of Mandarin dialog text data to construct the language model. The operations composition, $\det$, $\min$ and weight pushing introduced in Section 3 are implemented using the Openfst tool [2].

4.2.3 Training, inference and evaluation

During the training stage, 80-dimensional filter banks are extracted as speech features, with a frame length of 25 ms and a frame shift of 10 ms. To augment the data, a speech speed perturbation (Ko et al., 2015) is used, using perturbation coefficients of 0.9, 1.0, and 1.1. Furthermore, the SpecAugment (Park et al., 2019) strategy is also used to enhance the robustness of the model. All models are trained on 16 NVIDIA Tesla H200 GPUs with mixed precision training. For inference, all our computations are consistently performed on a Tesla T4 GPU (16GB) with a 16-core CPU and 32GB of RAM. For the inference stage, recognition performance is evaluated using the standard Character Error Rate (CER), computed by measuring the Levenshtein distance (Levenshtein, 1966) between the predicted sequence and the corresponding ground-truth transcription. To further accelerate decoding, a batch size of 5 is employed for all experiments.

[1] https://www.sri.com/platform/srilm

[2] https://www.openfst.org/twiki/bin/view/FST/WebHome

Table 1: Experiment results on the In-House dataset. The arrows indicate whether higher or lower values are preferable.

ID	Model	Decoding type	CER (%) ↓	Speed up ↑
A1	CTC Zipformer	CTC Greedy Search	3.27	3.45 ×
A2	AED Zipformer	AED Greedy Search	3.21	0.87 ×
B1	A1 + 5-gram	Dense CTC FST	3.09	1.00 ×
B2	A2 + 5-gram	Dense AED FST	3.56	0.51 ×
C	B1 + LSD (Chen et al., 2016)	0.99 *blank* threshold	3.12	1.18 ×
D	B1 + Speech-LLaMA (Wu et al., 2023a)	Averaging	3.10	1.29 ×
E	B1 + PolyVoice (Dong et al., 2023)	Discarding	4.71	1.99 ×
F	B1 + SWD (Zhang et al., 2025)	$\{0, \pm 1\}$	3.08	1.47 ×
G1	B1 + IOO-B	$\{0, 1\}$	3.06	1.65 ×
G2	B1 + IOO-B	$\{0, 1, 2\}$	3.06	1.54 ×
G3	G1 + IOO-NB	$\{*\}$	3.78	1.72 ×
G4	G1 + IOO-NB	$\mathrm{Max}\{*\}$	3.75	1.75×
G5	G1 + KOO	$\mathrm{Min}\{*\}$	3.06	1.67 ×
G6	G1 + KOO	$\mathrm{Max}\{*\}$	**3.05**	1.69 ×
H	B2 + IOO-B	$\{0, 1\}$	3.13	0.43 ×

4.3 Experimental results

4.3.1 Experiment of the In-House dataset.

Table 1 presents a quantitative evaluation of the proposed IOO and KOO algorithms on the In-House dataset, benchmarked against standard decoding baselines and representative heuristic acceleration methods. Rows A1 and A2 report greedy decoding results for the Zipformer model with CTC and AED outputs, respectively, defining the performance bounds under simple greedy search. Rows B1 and B2 show the corresponding results obtained with 5-gram Dense WFST decoding, serving as the primary accuracy–latency references. In particular, Row B1 (CTC Zipformer + 5-gram) establishes the dense baseline, achieving a CER of 3.09% and a normalized decoding speed of $1.00\times$, against which all subsequent acceleration results are measured.

Experiments C focus on evaluating the impact of a label synchronization-based algorithm when integrated with the TLG decoding graph, which using a 0.99 blank threshold. The LSD algorithm is applied under the hypothesis that as the frame discard threshold increases, recognition accuracy will approach that of the vanilla dense system, but with a corresponding reduction in inference speed. Experiments D and E involve averaging the blank frames between the neighbouring non-blank frames and discarding all the blank frames, respectively. Although these methods do enhance inference speed, they come with a more substantial trade-off in recognition accuracy.

Starting from the B1 baseline, G1 demonstrates that IOO-B accelerates decoding by 1.65× without any loss in recognition accuracy. Inserting additional blank frames in G2 offers no improvement and slightly reduces speed, while aggressive simplification in G3, replacing all non-blank frames with deterministic one-hot distributions, causes substantial accuracy degradation. A more conservative strategy in G4, retaining the highest-posterior frame within clusters of identical tokens, mitigates accuracy loss while still improving speed.

Experiments G5 and G6 validate the KOO algorithm: retaining only a single non-blank frame per token achieves significant acceleration without degrading accuracy, with the highest-posterior frame yielding the best performance. These results indicate that, for CTC-based WFST decoding, preserving appropriate blank frames and retaining only the most informative non-blank frames maintains recognition accuracy while maximizing computational efficiency.

Experiment H further confirms the generality of IOO on AED outputs, effectively mitigating the accuracy degradation observed with direct WFST decoding. Overall, these findings corroborate the key insight that blank frames encode positional information, whereas non-blank frames carry semantic content.

We further explore the impact of decoding hyperparameters on FST performance. The analyses in Fig. 2 indicate that Beam values must balance pruning and search breadth to maintain accuracy and speed, while Lattice Beam can be fixed without affecting performance. Max Active Tokens requires careful tuning to avoid search collapse or excessive memory usage, with around 5000 tokens provid-

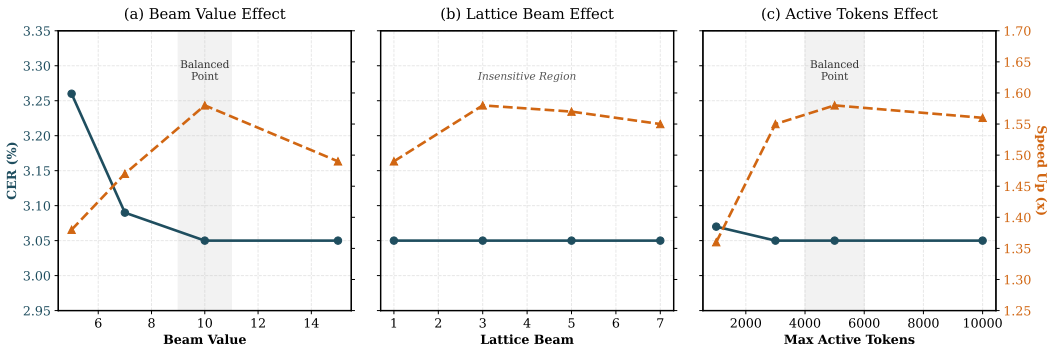

Figure 2: Sensitivity analysis of decoding parameters. The dual-axis plots illustrate the impact of (a) Beam Value, (b) Lattice Beam, and (c) Max Active Tokens on character error rate (CER, solid teal lines) and decoding speed-up (dashed orange lines). Shaded regions indicate the optimal operating points identified in the study, confirming that Lattice Beam is largely insensitive while Beam and Active Tokens require balanced tuning.

ing an effective operating point. Importantly, these results confirm that the IOO–KOO framework's improvements are not contingent on specific parameter choices: a single customized blank per blank region and the highest-probability non-blank frame per spike suffice to preserve accuracy while substantially accelerating decoding.

4.3.2 Experiment of the open-source datasets.

Table 2 present the experimental results on AISHELL-1 and LibriSpeech datasets. Across both datasets, our IOO–KOO framework consistently improves decoding efficiency while preserving recognition accuracy. On AISHELL-1, applying IOO or KOO to the 5-gram CTC system (B1) yields further CER reductions—from 3.73%/3.94% to 3.71%/3.93% (G1) and 3.70%/3.92% (G6), while achieving over 2.3 times decoding speedup. On LibriSpeech dataset, similar trends are observed. Relative to the dense CTC baseline B1 (1.95%/3.94%), both IOO (G1) and KOO (G6) maintain comparable accuracy and deliver more than 2 times speedup. Overall, the results across Mandarin and English corpora confirm that IOO–KOO is robust to language type and corpus scale. The framework consistently delivers substantial decoding acceleration while achieving equal or even improved recognition accuracy, underscoring its practical value for real-world FST-based ASR deployment.

5 Discussion and Analysis

IOO for positional information We hypothesize that blank frames in CTC-style formulations serve a dual functional role during FST decoding: beyond absorbing non-semantic acoustic variations, they implicitly encode positional information along the temporal axis, thereby constraining the relative ordering and spacing of non-blank tokens. This hypothesis is supported by several empirical observations. First, in CTC-based models, token sequences such as "A–B–<blk>–<blk>–<blk>", "<blk>–<blk>–<blk>–A–B", and "<blk>–A–<blk>–B–<blk>" are all semantically equivalent to the target sequence "AB", reflecting the alignment-invariant nature of CTC. However, our experimental results show that aggressively discarding blank frames only marginally affects performance, whereas their complete removal leads to a severe degradation in recognition accuracy. Moreover, although AED-based models typically suffer substantial accuracy loss under conventional FST decoding, their performance is significantly improved when integrated with the proposed IOO algorithm, which preserves essential blank-frame structure during decoding. These findings indicate that blank frames act as an implicit form of positional encoding in FST decoding, anchoring non-blank tokens in time and stabilizing the decoding process. Consequently, inserting at least one blank frame between any two adjacent non-blank frames is crucial for achieving performance comparable to Dense-FST decoding. This insight provides a principled explanation for the necessity of blank frames: they are not merely placeholders for non-semantic acoustic content, but also carry indispensable positional information that governs temporal alignment and decoding robustness.

KOO for semantic information The KOO algorithm improves FST decoding by selectively dis-

Table 2: Experiment results on the AISHELL-1 dataset. The results of the open-source dataset are evaluated using CER, while the LibriSpeech dataset is assessed using WER.

Model	AISHELL-1			LibriSpeech		
	Dev (%) ↓	Test (%) ↓	Speed up ↑	Test-clean (%) ↓	Test-other (%) ↓	Speed up ↑
CR-CTC (Yao et al., 2025)	**3.69**	3.98	-	**1.88**	3.95	-
CIF-Transducer (Zhang et al., 2024b)	4.1	4.3	-	-	-	-
Zipformer (Yao et al., 2023)	4.03	4.28	-	1.96	4.08	-
Branchformer (Peng et al., 2022)	4.19	4.43	-	2.4	5.3	-
E-Branchformer (Kim et al., 2023)	4.2	4.5	-	2.4	4.6	-
Paraformer (Gao et al., 2022)	4.7	5.1	-	-	-	-
Conformer (Gulati et al., 2020)	4.5	4.9	-	2.1	4.3	-
A1	3.98	4.19	3.63 ×	1.99	4.08	3.20 ×
A2	3.97	4.15	0.90 ×	2.94	4.15	0.89 ×
B1	3.73	3.94	1.00 ×	1.95	3.94	1.00 ×
B2	6.33	6.76	0.68 ×	3.77	6.98	0.55 ×
G1	3.71	3.93	2.37 ×	1.94	3.95	2.07 ×
G6	3.70	**3.92**	2.36 ×	1.94	**3.94**	2.06 ×
H	3.77	4.03	0.55 ×	2.01	4.18	0.49 ×

carding semantically redundant non-blank frames, thereby suppressing irrelevant acoustic variations while preserving essential information. This targeted pruning yields higher-quality inputs for subsequent decoding and leads to consistent gains in both efficiency and recognition accuracy. The effectiveness of KOO is supported by three key observations. First, the performance improvements initially observed in Table 1 (G5 and G6) are consistently reproduced in Table 2 (G6), demonstrating the robustness of the proposed method. Second, as the pruning threshold increases from 0.8 to 0.99 (Appendix D), recognition accuracy improves monotonically, reaching a relative gain of 3.06% in G10, while still maintaining over a 1.6× inference speedup. Finally, KOO is theoretically well aligned with the many-to-one mapping property of CTC. During training, adjacent identical non-blank frames are naturally merged within the CTC lattice, and during decoding, sequences such as "A–A–<blk>–<blk>", "<blk>–<blk>–A–A", and "<blk>–A–A–<blk>" are all semantically equivalent to the target label "A". This intrinsic equivalence provides a principled justification for pruning redundant non-blank frames without degrading decoding correctness.

KOO's hidden gem To the best of our knowledge, existing End-to-End ASR technologies employing a TLG graph for decoding suffer from a latent issue during the construction of the static Token.fst graph (Miao et al., 2015; Laptev et al., 2022): the probabilities of consecutive non-blank frames become excessively high due to self-loop operation at the non-initial states. To counteract the path probability inflation caused by consecutive identical non-blank frames, the KOO algorithm exclusively selects the single frame with the highest posterior probability during the FST decoding process. Tables 1 and 2 (G1 vs. G6) further validate the effectiveness of KOO in resloving the path probability inflation. These findings introduce a novel and effective paradigm for leveraging CTC/AED posterior probabilities during FST-based decoding, offering new insights for further optimization.

6 Conclusion

In this paper, we thoroughly explore the spiking behavior of CTC/AED outputs and propose the conjecture that blank frames provide positional information, while non-blank frames carry semantic information beneficial to the model. Building on this, we present two complementary algorithms to enhance both inference speed and recognition accuracy of CTC/AED-based E2E ASR systems named IOO and KOO algorithms. By reconstructing the sequence of blank and non-blank frames, our method enables a more efficient integration with WFSTs, drastically reducing the number of decoding frames. Additionally, the problem of severe accuracy degradation in AED-FST can be mitigated by using the IOO algorithm. Futhermore, we introduce a weight pushing optimization between the det and min steps, improving TLG search efficiency through early pruning. The experimental results on In-House, AISHELL-1 and Librispeech datasets confirm that the IOO and KOO algorithms can significantly enhance inference speed while even improving recognition accuracy.

Limitations

While this study demonstrates significant improvements in decoding efficiency and accuracy, the evaluation is currently concentrated on standard benchmarks such as AISHELL-1 and LibriSpeech. Consequently, the robustness of the proposed algorithms across a broader spectrum of low-resource languages and complex acoustic environments remains to be fully explored.

References

Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jing Bai, Eric Battenberg, and 1 others. 2016. Deep speech 2: End-to-end speech recognition in english and mandarin. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 173–182.

Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng. 2017. Aishell-1: An opensource mandarin speech corpus and a speech recognition baseline. In *IEEE Proceedings of the 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.

William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4960–4964.

Zhehuai Chen, Yan Zhuang, Yanmin Qian, and Kai Yu. 2016. Phone synchronous speech recognition with ctc lattices. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(1):90–101.

Jan Chorowski and Navdeep Jaitly. 2017. Towards better decoding and language model integration in sequence to sequence models. In *ISCA Annual Conference of the International Speech Communication Association (Interspeech)*, pages 523–527.

Galvez Daniel and Tim Kaldewey. 2023. Gpu-accelerated wfst beam search decoder for ctc-based speech recognition. In *IEEE Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–7.

Linhao Dong, Shuang Xu, and Bo Xu. 2018. Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5884–5888.

Qian Dong, Ziyue Huang, Qiao Tian, Chen Xu, Tom Ko, and 1 others. 2023. Polyvoice: Language models for speech to speech translation. *arXiv preprint arXiv:2306.02982*.

Zhifu Gao, Shiliang Zhang, Ian McLoughlin, and Zhijie Yan. 2022. Paraformer: Fast and accurate parallel transformer for non-autoregressive end-to-end speech recognition. In *ISCA Annual Conference of the International Speech Communication Association (Interspeech)*, pages 2063–2067.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*.

Alex Graves, Santiago Fernandez, Faustino Gomez, and Jurgen Schmidhuber. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pages 369–376.

Alex Graves and Navdeep Jaitly. 2014. Towards end-to-end speech recognition with recurrent neural networks. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pages 1764–1772.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, and 1 others. 2020. Conformer: Convolution-augmented transformer for speech recognition. In *ISCA Annual Conference of the International Speech Communication Association (Interspeech)*, pages 5036–5040.

Takaaki Hori and Atsushi Nakamura. 2022. *Speech recognition algorithms using weighted finite-state transducers*. Springer Nature.

Biing-Hwang Juang and Lawrence R Rabiner. 1991. Hidden markov models for speech recognition. *Technometrics*, 33:251–272.

Anjuli Kannan, Yonghui Wu, Patrick Nguyen, and 1 others. 2018. An analysis of incorporating an external language model into a sequence-to-sequence model. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Kwangyoun Kim, Felix Wu, Yifan Peng, Jing Pan, Prashant Sridhar, and 1 others. 2023. E-branchformer: Branchformer with enhanced merging for speech recognition. In *IEEE Proceedings of the Spoken Language Technology Workshop (SLT)*, pages 84–91.

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

Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. 2015. Audio augmentation for speech recognition. In *ISCA Annual Conference of the International Speech Communication Association (Interspeech)*, pages 3586–3589.

Aleksandr Laptev, Somshubra Majumdar, and Boris Ginsburg. 2022. Ctc variations through new wfst topologie. In *ISCA Annual Conference of the International Speech Communication Association (Interspeech)*, pages 1041–1045.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics doklady*, 10(8):707–710.

Hang Lv, Zhehuai Chen, Hainan Xu, Daniel Povey, Lei Xie, and Sanjeev Khudanpur. 2021. An asynchronous wfst-based decoder for automatic speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6019–6023.

Yajie Miao, Mohammad Gowayyed, and Florian Metze. 2015. Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding. In *IEEE Proceedings of the Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 167–174.

Mehryar Mohri, Fernando CN Pereira, and Michael D Riley. 2000. The design principles of a weighted finite-state transducer library. *Theoretical Computer Science*, 231(1):17–32.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.

Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, and 1 others. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. In *ISCA Annual Conference of the International Speech Communication Association (Interspeech)*, pages 2613–2617.

Yifan Peng, Siddharth Dalmia, Ian Lane, and Shinji Watanabe. 2022. Branchformer: Parallel mlp-attention architectures to capture local and global context for speech recognition and understanding. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 17627–17643.

Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahremani, Vimal Manohar, and 1 others. 2016. Purely sequence-trained neural networks for asr based on latticefree mmi. In *ISCA Annual Conference of the International Speech Communication Association (Interspeech)*, pages 2751–2755.

Rohit Prabhavalkar, Takaaki Hori, Tara N Sainath, Ralf Schluter, and Shinji Watanabe. 2024. End-to-end speech recognition: A survey. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:325–351.

Tara N Sainath, Yanzhang He, Arun Narayanan, Rami Botros, Ruoming Pang, and 1 others. 2021. An efficient streaming non-recurrent on-device end-to-end model with improvements to rare-word modeling. In *ISCA Annual Conference of the International Speech Communication Association (Interspeech)*, pages 1777–1781.

Tara N Sainath, Brian Kingsbury, George Saon, Hagen Soltau, Abdel-rahman Mohamed, and 1 others. 2015. Deep convolutional neural networks for large-scale speech tasks. *Neural Networks*, 64:39–48.

Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi. 2017. Hybrid ctc/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1240–1253.

Jian Wu, Yashesh Gaur, Zhuo Chen, Long Zhou, and Yujia Zhu. 2023a. On decoder-only architecture for speech-to-text and large language model integration. In *IEEE Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8.

Jian Wu, Yashesh Gaur, Zhuo Chen, Long Zhou, Yujia Zhu, and 1 others. 2023b. On decoder-only architecture for speech-to-text and large language model integration. In *IEEE Proceedings of the Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8.

Zhao Yao, Long Guo, Xing Yang, Wei Kang, Fangjun Kuang, and 1 others. 2023. Zipformer: A faster and better encoder for automatic speech recognition. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*.

Zhao Yao, Wei Kang, Xing Yang, Fangjun Kuang, Long Guo, and 1 others. 2025. Cr-ctc: Consistency regularization on ctc for improved speech recognition. In *Proceedings of the Fourteenth International Conference on Learning Representations (ICLR)*.

Zhuo-You Yao, Di Wu, Xiong Wang, Bin-Bin Zhang, Fan Yu, and 1 others. 2021. Wenet: Production oriented streaming and non-streaming end-to-end speech recognition toolkit. In *ISCA Annual Conference of the International Speech Communication Association (Interspeech)*, pages 4054–4058.

Tian-Hao Zhang, Qi Liu, Xinyuan Qian, Song-Lu Chen, Feng Chen, and Xu-Cheng Yin. 2023a. Self-convolution for automatic speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Tian-Hao Zhang, Xinyuan Qian, Feng Chen, and Xu-Cheng Yin. 2024a. Transmitted and aggregated self-attention for automatic speech recognition. In *ISCA Annual Conference of the International Speech Communication Association (Interspeech)*, pages 227–231.

Tian-Hao Zhang, Han Qin, Zhi-Hao Lai, Song-Lu Chen, Qi Liu, and Xu-Cheng Yin. 2023b. Rethinking speech recognition with a multimodal perspective via

acoustic and semantic cooperative decoding. In *ISCA Annual Conference of the International Speech Communication Association (Interspeech)*, pages 914–918.

Tian-Hao Zhang, Dong Zhou, Guo Zhong, Jiatong Zhou, and Bo Li. 2024b. Cif-t: A novel cifbased transducer architecture for automatic speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10531–10535.

Wei Zhang, Tian-Hao Zhang, Cheng Luo, Hang Zhou, Chao Yang, and 1 others. 2025. Breaking through the spike: Spike window decoding for accelerated and precise automatic speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Jiatong Zhou, Sheng Zhao, Yang Liu, Wei Zeng, Yujia Chen, and 1 others. 2024. knn-ctc: Enhancing asr via retrieval of ctc pseudo labels. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11006–11010.

A IOO for CTC-FST

Algorithm 1 IOO and KOO Algorithms for CTC

Require: Posterior probability $P \in \mathbb{R}^{T \times V}$
Ensure: Processed sequence H_{final}

function PROCESSPOSTERIORSCTC(P)
  $B \leftarrow$ Queue ▷ *Init frames block queue*
  $i, j, k \leftarrow 0$
  $koo \leftarrow$ false ▷ *Init koo setting*
  while $i < T$ **do**
    $p_c \leftarrow -1$ ▷ *Init previous c with -1*
    $c, p \leftarrow \text{argmax}(P[i]), \max(P[i])$
    if $c \neq$ p_c **then**
      $p_c \leftarrow c$
      $b \leftarrow [(P[i], p)]$ ▷ *Init sub block b*
      $j \leftarrow i + 1$
      while $j < T \wedge \text{argmax}(P[j]) = c$ **do**
        b.append$((P[j], \max(P[j])))$
        $j \leftarrow j + 1$
      B.push(c, b) ▷ *Push b to queue B*
      $i \leftarrow j$ ▷ *Jump to next block*
    else
      $i \leftarrow i + 1$
  $H_{final} \leftarrow []$
  $v_{blk} \leftarrow [1.0, 0, \ldots, 0]$ ▷ *One-hot blank vector*
  H_{final}.append(v_{blk}) ▷ *First customized frame*
  while B is not empty **do**
    $b \leftarrow B$.pop
    if $k = 0 \wedge b$.key $= 0$ **then**
      $k \leftarrow k + 1$
      continue
    if b.key $= 0$ **then**
      Phase 1: Insert-Only-One (IOO)
      H_{final}.append(v_{blk}) ▷ *With IOO*
    else if not koo **then**
      H_{final}.append$(b$.value$)$ ▷ *Without KOO*
    else
      Phase 2: Keep-Only-One (KOO)
      $f^* \leftarrow$ SelectMaxProbFrame$(b$.value$)$
      H_{final}.append(f^*)
  return H_{final}

B IOO for AED-FST

Algorithm 2 IOO Algorithm for AED

Require: Posterior probability $P \in \mathbb{R}^{T \times V}$
Ensure: Processed sequence H_{final}

function PROCESSPOSTERIORSAED(P)
  $i \leftarrow 0$
  $H_{final} \leftarrow []$
  $v_{blk} \leftarrow [1.0, 0, \ldots, 0]$ ▷ *One-hot blank vector*
  H_{final}.append(v_{blk})
  while $i < T$ **do**
    H_{final}.append$(P[i])$
    H_{final}.append(v_{blk})
    $i \leftarrow i + 1$
  return H_{final}

C Encoder Configurations

Table 3: Zipformer configurations used in our experiments.

Parameter	Values
Layers num	2, 2, 4, 6, 4, 2
FFN dim	512, 768, 1536, 2048, 1536, 768
Encoder dim	192, 256, 512, 768, 512, 256
Encoder-unmasked dim	192, 192, 256, 320, 256, 192

D KOO for Non-Blank Frames

Table 4: Results of thresholds of non-blank frames on the In-House dataset.

ID	Non-balnk threshold	CER (%) ↓	Speed up ↑
G3	$\{*\}$	3.78	1.72 ×
G7	$\{* \geq 0.8\}$	3.77	1.73 ×
G8	$\{* \geq 0.90\}$	3.29	1.62 ×
G9	$\{* \geq 0.95\}$	3.13	1.64 ×
G10	$\{* \geq 0.99\}$	3.06	1.61 ×