

## 1. 주제

초저지연 엣지 추론을 위한 공동 토큰 융합 아키텍처 (JTFA)

### 분반, 팀, 학번, 이름

1반 4팀 20222582 전인성

## 2. 요약

엣지 분산 추론에서는 평균 지연보다 예측 불가능한 실시간 엣지 분산 추론에서는 평균 지연보다 p99 꼬리 지연의 안정화가 서비스 품질을 좌우한다. 기존의 중간 특징맵 전송은 대역 요구가 크고 지터/재전송에 취약해 동기화 대기와 변동성을 키운다.

본 과제는 입력을 저차원 토큰( $d=16$ )으로 요약해 전송하고, Fusion 노드에서 윈도우 집계(K-of-N)와 가중합/게이팅으로 결합하는 JTFA를 제안/구현한다.

Raspberry Pi 기반 테스트베드에서  $\approx 5.5 \text{ kbps}$  토큰 전송만으로 p50 유지와 p99 안정화 가능성을 관찰했으며, 모드별(plain/PSMix/HMix) 인코딩은 동일 조건에서 대역을 크게 절감한다.

본 제안은 처리량-지연-대역의 파레토 개선을 목표로 하며, 향후 동적 토큰률·K-of-N 적응과 무선 손실 대응(FEC/중복)으로 확장한다.

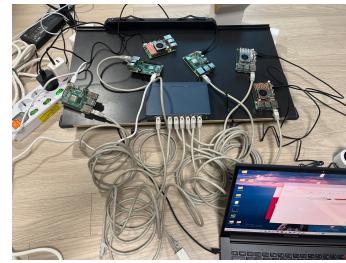
본 제안을 실생활로 연결해본다면 여러 카메라가 보낸 저차원 토큰을 중앙에서 게이팅 가중합으로 결합하되, 시간창 T 안에 도착한 K개만으로도 즉시 예측합니다. 이렇게 해서 대역을 수십 배 줄이면서( $p \approx 5.5 \text{ kbps}$ ) p99 꼬리를 낮추고, p50·정확도·처리량 손실을 최소화하는 것이 목표입니다.

참고 : “ $\Delta = p99 - p50$ , tail factor =  $p99/p50$ ”

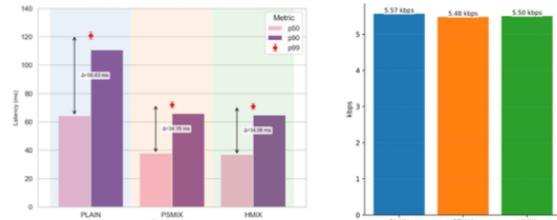
## 3. 대표 그림

```
C:\Users\wjsgl>nmap -sS 192.168.137.0/24
Starting Nmap 7.98 ( https://nmap.org ) at 2025-10-03 14:44
Nmap scan report for 192.168.137.64
Host is up (0.0010s latency).
MAC Address: 2C:CF:FF (Raspberry Pi (Trading))
Nmap scan report for 192.168.137.109
Host is up (0.0010s latency).
MAC Address: DC:A6:32 (Raspberry Pi Trading)
Nmap scan report for 192.168.137.132
Host is up (0.0010s latency).
MAC Address: B8:AE:9E (Raspberry Pi (Trading))
Nmap scan report for 192.168.137.176
Host is up (0.0009s latency).
MAC Address: DC:A6:32:C9:33:0E (Raspberry Pi Trading)
Nmap scan report for 192.168.137.178
Host is up (0.0009s latency).
MAC Address: DC:A6:32:C9:34:94 (Raspberry Pi Trading)
Nmap scan report for 192.168.137.1
Host is up (0.0009s latency).
Nmap done: 256 IP addresses (6 hosts up) scanned in 3.58 s
```

<그림 1 - NMap으로 각 노드 연결상황 확인>



<그림 2 - 실제 실험 환경>



<그림 3 - 인코딩 모드별 지연·통신 특성  
 $d=16$  토큰( $\approx 5.5 \text{ kbps}$ )로 p50 유지·p99 안정화 가능성 >

## 6. 결론

결론: 토큰 기반 통신과 윈도우 가중 융합은 대역을 수십 배 절감하면서도 p99 꼬리 안정화를 달성할 수 있는 유효한 경로다. Raspberry Pi 테스트베드에서  $\approx 5.5 \text{ kbps}$  조건으로 p50 유지 및 p99 안정화 가능성을 확인했다. 본 과제는 분산 추론의 지연·대역 파레토 개선을 지향하며, 다음 단계로 동적 토큰률·K-of-N 적응과 무선 환경 강건화(FEC/중복)를 진행한다.

#### 4. 서론

실시간 엣지 컴퓨팅은 현장에서 생성되는 데이터를 자체 없이 처리해야 하며, 특히 꼬리 지연(p99)의 제어가 서비스 안정성의 관건이다. 분산 추론 환경에서 중간 특징 맵을 교환하면 전송량이 커지고, 불규칙한 jitter와 재전송, 동기화 대기가 겹치며 p99 변동이 커지는 문제가 반복적으로 관찰된다. 반대로, 단일 장치 내부의 토큰 축소나 모델 경량화는 계산량을 줄이는 데 효과적이지만, 다장치 협업 시 발생하는 통신 병목과 융합 대기 문제를 직접 해소하지는 못한다. 이 사이에서 경량 통신과 견고한 융합을 동시에 달성하는 체계가 필요하다.

이 공백을 메우기 위해 본 논문은 Joint Token Fusion Architecture(JTFA)를 제안한다. 각 Producer는 입력으로부터 저차원 토큰을 독립적으로 생성해 통신의 기본 단위로 사용하고, Fusion 노드는 수신된 토큰을 계이팅 기반의 소프트맥스 가중 혼합으로 결합해 최종 결정을 만든다. 이 가중 혼합은 멀티헤드 어텐션(MHA)에서 영감을 받았으나 동일한 구조는 아니다(단일 헤드, 토큰 간 self/cross 상호작용 없음, 간단한 선형 스코어링). W/T/K 정책을 통해 p99 안정화와 처리량 사이의 목표점을 상황에 맞게 조정할 수 있다.

본 연구는 다음 연구 질문을 검증한다.

첫째, 인코딩 방식이  $p50 \cdot p90 \cdot p99$ 와 꼬리 폭 ( $\Delta = p99 - p50$ )을 줄이면서 bytes/decision과 kbps를 과도하게 늘리지 않을 수 있는가.

둘째, 보수적인 W/T 설정이 p99와 변동성을 안정화하는 대신 fillratio · 처리량을 얼마나 희생시키는가?

셋째, 완화된 W/T 설정이 처리량과 함께 안정성을 높이는 대신 p99 변동성을 얼마나 키우는가.

넷째, 동일 차원 · 동일 토큰률에서 정확도 - 대역폭의 형태가 인코딩에 따라 어떻게 달라지는가.

이 질문들은 3장에서 A1 - A4 시나리오로 단계적으로 평가한다.

본 논문의 기여는 네 가지로 요약된다.

(1) 다장치 협업 추론에서 꼬리 지연의 구조적 원인을 겨냥한 토큰 기반 통신 + 가중 혼합 융합 아키텍처(JTFA)를 제안한다 (운영 다이얼: W/T/K).

(2)  $p50 \cdot p90 \cdot p99$ ,  $\Delta$ , bytes/decision(kbps)을 핵심으로 하는 측정 프레임을 정립하고 실험 절차를 제시한다.

(3) Raspberry Pi 테스트베드에서 인코딩 · 정책 · 정확도-대역폭 관계를 단계 검증한다.

(4) 정확도 - 대역 - 지연 사이의 트레이드오프를 제시하고, 차원/양자화/토큰률 스윕과 적응 제어로 확장 가능한 향후 연구 경로를 개괄한다.

참고로, 원시 특징/특징맵 오프로딩의 대역 요구는 이론적으로 JTFA와 큰 스케일 차이를 보인다. 동일 결정률과 동일 뷰 수(동일 fps) 조건에서,  $d=16$  토큰의 실측 대역은 약 50 kbps인 반면 펜얼티밋 벡터 ( $D=1280$ )는 약  $12 \times (\text{int8}) \sim 49 \times (\text{fp32})$ , 마지막 특성맵( $7 \times 7 \times 1280$ )은 약  $600 \times (\text{int8}) \sim 2400 \times (\text{fp32})$  더 큰 대역을 요구한다. 유선 1 GbE에서도 전송 자체는 가능하지만 큐잉/버퍼링/동시성/OS 스택 오버헤드로 p99 변동이 커질 수 있고, 무선에서는 충돌/재전송 특성상 tail 악화가 구조적으로 불가피하다. 따라서 본 논문은 동급 대역( $\approx 5.5$  kbps)에서의 지연 안정화 효과를 공정하게 보기 위해 토큰 기반 모드(PLAIN/PSMix/HMix)에 초점을 둔다.

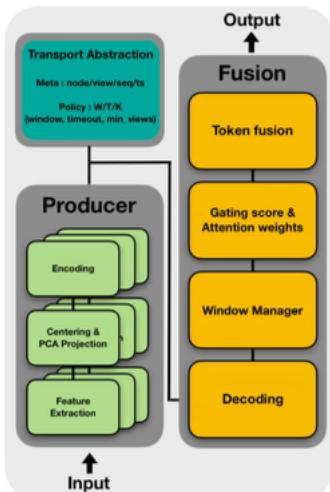
본 논문의 구성은 다음과 같다. 2장은 방법론, 3장은 테스트베드 · 정책 기본값 · 측정 지표를 구분해 실험 설계와 결과를 보고한다. 4장은 결론과 향후 연구를 제시한다.

## 5. 본론

### 5.1 시스템 개요(그림 1개 이상)

구성: *Producer* $\times N \rightarrow (UDP\cdot msgpack) \rightarrow Fusion(Window T, K-of-N, Gating/Weighted Sum)  $\rightarrow$  Linear Head(예측)$

처리 흐름: 입력 프레임  $\rightarrow$  백본 특징  $\rightarrow$  Tokenizer( $d=16$ )  $\rightarrow$  전송  $\rightarrow$  Fusion에서 시간창 내 도착 토큰 결합  $\rightarrow$  예측/로그 기록.



<그림 1 : JTFA의 구조적 흐름>

### 5.2 필요한 기술 요소 설명

통신: 비연결형 UDP, 논블로킹 select, 버퍼 드레인, 패킷 유실 허용 설계.

토큰화 모드: plain(샘플링), psmix(치환·부호 혼합, seed 재현), hmix(FWHT 기반 혼합, n2 메타 포함).

융합: Window(T), K-of-N(지연 큰 노드 비차단), 게이팅/가중합(품질/신뢰도 반영).

평가 지표: p50/p90/p99, tail factor(p99/p50), 대역(kbps), 정확도(선형 헤드).

로깅 스키마: timestamp, cause, m, fill\_ratio, bytes, lat\_mean, p50, p90, p99, tail\_factor, pred, gt, correct, acc\_running.

### 5.3 구현 방법 및 개발 방향

버전 v1(완료): producer.py, fusion.py 최초 파이프라인, CSV 로깅.

v2(안정화): 수신 버퍼 드레인, 차원 검증, tail\_factor 컬럼 추가, README/데모.

v3(평가 확장): A1–A3 시각화 스크립트(지연/대역/정확도), A4 정확도-대역 트레이드오프.

오픈소스 배포: Public GitHub, LICENSE 명시,

## 7. 출처

- [1] Y. Kang et al., "**Neurosurgeon: Collaborative Intelligence Between the Cloud and Mobile Edge**," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2017.
- [2] R. Xu et al., "**V2X-ViT: Vehicle-to-Everything Cooperative Perception with Vision Transformer**," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.
- [3] E. T. B. de Oliveira et al., "**A Survey on Tail Latency in Edge Computing**," *ACM Computing Surveys*, 2022.
- [4] Y. Rao et al., "**DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification**," in *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.
- [5] D. Bolya et al., "**Token Merging: Your ViT But Faster**," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.
- [6] A. Vaswani et al., "**Attention Is All You Need**," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.