REINFORCEMENT LEARNING Project Report

Deep Q-Network

강화학습 프로젝트

발표 주제

Bandit DQN: 프레임 단위 정밀 Sound Event Detection을 위한 Deep Q-Network 구현

팀 / 발표자

팀 메키 / 김태곤 / 120250319

소속 / 학과

서강대학교 / 컴퓨터공학과
CVIP Lab

Github : https://github.com/TSATOA/RL_Project.git

INDEX TABLE OF CONTENTS

목차

SECTION 01 INTRODUCTION

프로젝트 주제 및 목표

프로젝트 주제 선정

Bandit DQN: 프레임 단위 정밀 Sound Event Detection을 위한 Deep Q-Network 구현

Sound Event Detection에서는 오래전부터 임계값 설정이 데이터별로 달라지는 고질적 문제가 존재한다.

이에 본 프로젝트는 이 문제를 선택·보상 구조로 재해석하고 **강화학습 기반 DQN**을 적용하여 프레임 단위에서 **최적 임계값을 자동으로 학습**하고자 한다.

주요 달성 목표

강화학습 알고리즘 구현 능력 함양

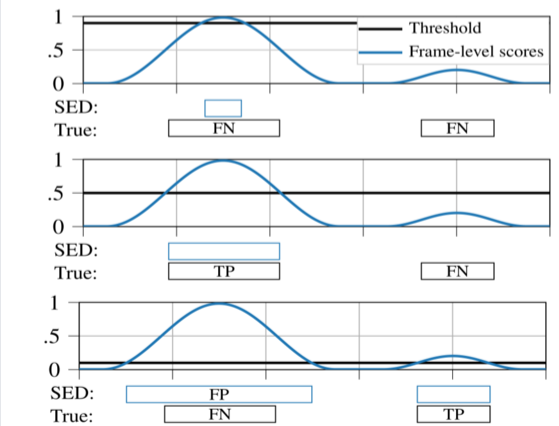
Deep Q-Network 구조를 Sound Event Detection 문제에 맞게 직접 설계하고 구현한다.

탐색·보상 구조 설계 및 실험

프레임 단위 임계값 선택 문제를 위해 State–Action–Reward 구조를 설정하고 이에 맞는 탐색 방식을 적용한다.

데이터 기반 성능 평가 및 시각화

프레임 단위 Sound Event Detection에 강화학습을 적용하여 임계값 선택 성능을 정량적으로 평가 및 시각화한다.



Threshold에 따라 달라지는 Detection 결과

SECTION 02 Problem Background & DATA

연구 배경 및 데이터셋 설명

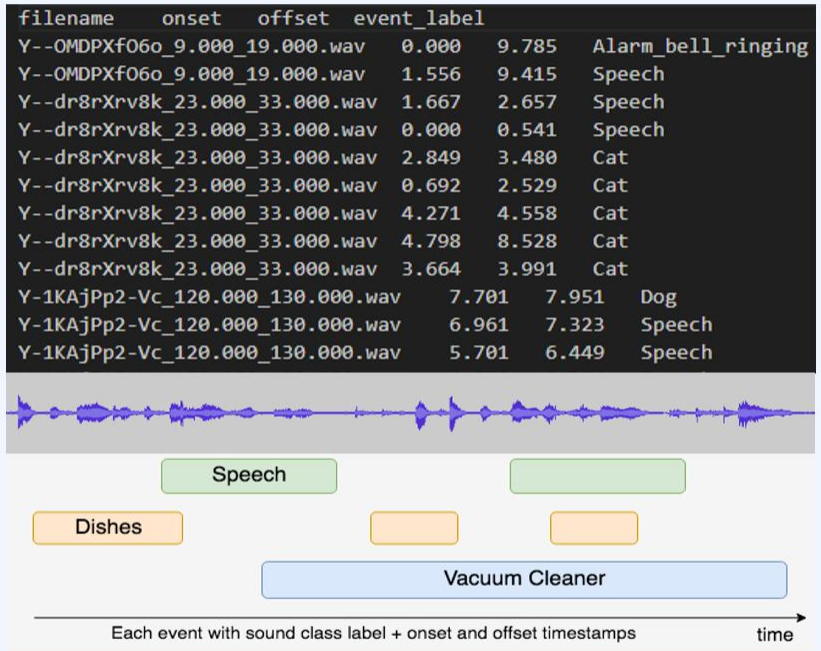
연구 배경(Background)

- **프레임 기반 Sound Event Detection의 구조적 한계**
  Sound Event Detection(SED) 모델은 일반적으로 프레임별 존재 확률을 산출하지만,이 확률을 실제 이벤트로 변환하기 위해서는 반드시 임계값(Threshold) 설정이 필요
- **기존 Threshold 방식의 비효율성과 불안정성**
  고정 Threshold는 특정 구간에서는 과탐(과한 예측), 다른 구간에서는 미탐(놓침) 발생
  또한 입력된 소리 이벤트의 종류에 맞는 Threshold 설정 필요

➡ **프레임별 최적 Threshold 동적 조정 및 소리 이벤트 존재 구간 미세 조정 필요**

데이터셋 구성 (Dataset)

- **DCASE* 2024 Task4 Strong Label* Dataset 활용 :**
  Sound Event Detection 학습을 위해 DCASE 챌린지에서 공식 제공하는 10개 클래스의 유튜브 기반 Audioset Strong Dataset(총 1,370개 오디오) 사용
- **DCASE 2024 Task4 Synthetic Strong Label Dataset 활용 :**
  DCASE에서 제공하는 합성(Synthetic) 데이터 10,000개(Train)를 추가로 활용하였으며, 이는 동일하게 10개 클래스로 구성되어 실제 환경의 다양한 소리 조합 모사
- **평가용 데이터셋 구성 :**
  모델 성능 평가는 Synthetic 데이터 2,500개(Val)를 사용하여 정량적 측정 수행
  (Threshold 선택 정확도, 프레임 단위 이벤트 검출 성능 등)



DCCASE 챌린지에서 제공하는 Strong Label Dataset

DCASE 챌린지란?
환경 음향 및 소리 인식 기술을 평가하기 위한 국제적 벤치마크 대회이다.
Strong Label이란?
이벤트의 시작·종료 시점과 클래스 정보를 모두 포함하는 프레임 단위 라벨이다.

SECTION 03 DATA PIPELINE

데이터 Preprocessing

정밀한 Sound Event Detection을 위해 오디오 신호로부터 **RMS·DASM 기반의 프레임 단위 특징** 추출
이를 활용하여 확률 계산 및 **Baseline Threshold**를 구성하여 학습할 수 있는 형태로 전처리 수행
반복 학습의 효율성과 일관성을 위해 모든 특징을 NPZ 형태로 캐싱하여 사용

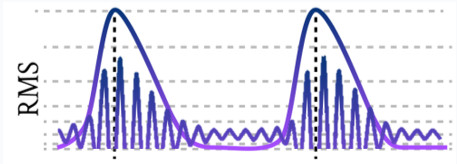
1

RMS* 계산

입력된 오디오를 바탕으로 RMS 계산

음향 이벤트의 존재 여부를 비교적 안정적으로 반영

특정 **이벤트 존재** 여부를 **구분하지 못함**



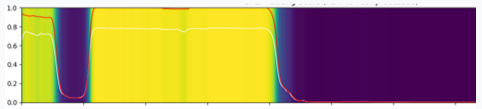
2

DASM* Score계산

입력된 오디오와 **텍스트에 대한 이벤트** 기반 Score 계산

단순히 소리의 유무가 아닌 이벤트 검출

SED 모델 특성 상 **오검출 발생 가능**



3

RMS x DASM

오디오 이벤트에 대한 **물리적 강도**와 **의미적 중요도**의 결합

Noise·배경음 같은 RMS만 높은 구간 제거

두 신호의 단점을 서로 보완

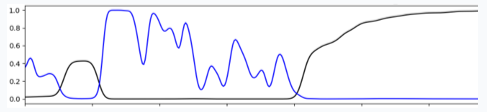
4

Dynamic Threshold 계산

(1-DASM)으로 통해 배경음을 반영한 초기 **Baseline Threshold** 계산

강화학습 기반 Threshold 미세 조정

최종 (RMSxDASM)과 Threshold를 통해 Event 존재 구간 검출



**구체적인 과정은 부록1 참고

RMS란? 오디오 신호의 프레임별 에너지 크기를 나타내는 지표로, 소리의 물리적 세기를 정량화 한 값이다.
DASM이란? Sound Event Detection 모델로, 입력 쿼리에 대한 의미 기반 점수를 출력하는 모델이다. Open-Vocabulary Sound Event Detection with Multi-Modal Queries(ACM MM 2025)

SECTION 04 SAR DEFINITION

State, Action, Reward 설계

State (상태)

에이전트가 관측하는 환경 정보

RMS : 에너지 기반 loudness
D_raw : 원본 DASM 스코어
D_adj : 보정된 DASM 스코어
Prob : RMS와 DASM 기반 확률값
T_base : 현재 적용된 Threshold 값
Binary_pred : 현 프레임 이벤트 판단 결과(0/1)

기준 프레임 ± 10프레임의 정보 제공 → 총 21프레임
6개의 feature x 21프레임 = **126차원의 State**

프레임 단위 이벤트는 **국소적 문맥**(temporal context)이 중요하기 때문에 주변 구간을 함께 입력

Action (행동)

에이전트가 선택하는 의사결정

Action Space : [-0.2, -0.18, ⋯, 0, 0.02, ⋯, 0.2]
Threshold = T_base + Action

-0.2 ~ +0.2 범위의 **21개 Discrete Action Space**
기존 Threshold 기준 보정값 조정

과도한 변화는 학습이 불안정해지므로 ± 0.2로 제한

미세 조정 중심의 Action 설계로 Threshold를 부드럽게 이동시키며 안정적인 학습을 유도

Reward (보상)

행동에 대한 환경의 피드백

맞았다 → 맞았다 : **+0.2**
맞았다 → 틀렸다 : **-1.0**
틀렸다 → 틀렸다 : **-0.2**
틀렸다 → 맞았다 : **+1.0**

잘못된 예측을 **고쳤을 때 큰 보상**(+1.0)
기존에 잘 맞던 예측을 **망치면 큰 패널티**(-1.0)
변화 없는 경우는 작은 보상/패널티로 안정적 탐색 유도

올바른 방향의 Threshold 조정을 적극적으로 강화하고 불필요한 변화는 억제

SECTION 05 ALGORITHM

Bandit DQN 알고리즘

핵심 개념 (Core Concept)

DQN : 상태(State)와 행동(Action)의 가치를 Q-Value로 근사하기 위해 **Neural Network**를 사용하여 복잡한 환경에서도 최적 정책을 학습할 수 있도록 하는 가치 기반(Value-based) 강화학습 알고리즘 기법

Q-Learning의 업데이트 규칙을 신경망이 직접 근사하기 때문에, 연속적인 상태나 고차원 입력에서도 안정적인 학습 가능

왜 DQN을 선택했는가?

- **독립적 의사결정의 효율성**: Threshold 조정은 매 프레임 독립적인 Bandit 의사결정 문제. 장기 시퀀스가 필요한 정책 기반 알고리즘보다 Q-value 기반의 DQN이 효율적
- **문제 단순화 (Bandit Formulation)**: γ(감가율)를 극한으로 줄여 미래 보상 대신 프레임 단위의 즉시 보상만 학습하는 Bandit 형태로 문제 단순화 가능

Q-Network 구성

SECTION 06 CONFIGURATION

Hyperparameter 설정

TCN-BiGRU-Transformer 하이브리드 구조의 Q-Network를 기반 DQN 구성
Bandit 문제 특성에 맞춰 Gamma=0으로 설정하여 즉시 보상을 최적화

CATEGORY	PARAMETER	VALUE	DESCRIPTION
Network	Hidden Dim	128	Block 공통 hidden 크기(Transformer는 256)
	Seq Length	21 frames	state는 21-frame window
	In Channels	6 channels	RMS, D_raw, D_adj, Prob, T, Binary
Learning	Learning Rate	1e-4	고정 LR (Optimizer: Adam)
	Batch Size	32	replay sampling batch
	Gamma	0	Bandit 문제
Exploration	Epsilon Decay	1.0 → 0.05	linear decay
	Decay Steps	20,000 steps	20000까지 선형 감소
Memory	Buffer Size	50,000	GPU 기반 deque
	Train Start	> 2,000 samples	버퍼 최소 확보 후 시작
	저장 조건	prob - th ≤ 0.2	Boundary-based Learning*
Action	Type	Discrete (21)	threshold 변환값 선택
	Range	-0.2 ~ +0.2	threshold 조정 step (0.02)

Training Strategy

Boundary-based Learning: 예측 확률이 Threshold의 ±0.2 안에 있는 샘플만 선택적으로 Replay Buffer에 저장
즉, **Action의 결과로 정답이 바뀔 수 있는 경우만 선택**

SECTION 07 EXPERIMENTAL SETUP

실험 셋업: 환경 및 평가 지표

실험 실행 환경 (Execution Environment)

Framework

Python 3.10.18, PyTorch 2.0.0, CUDA 11.4 기반 Docker 환경 구성

Hardware

NVIDIA RTX 3090 (24GB), Dual Intel Xeon Gold 6226R (총 32코어)

Dataset

> **Train**: DCASE 2024 Task4 Strong & Synthetic Strong Dataset (총 11,370개 오디오)
>
> **Eval**: DCASE 2024 Task4 Synthetic Validation Set (2,500개 오디오)

평가 지표 (Evaluation Metrics)

Accuracy

(TP + TN)/ (TP + TN + FP + FN)
전체 프레임 중 모델이 정확히 예측한 프레임의 비율을 나타내는 평가 지표

IoU (Intersection over Union)

TP / (TP + FP + FN)
예측 구간과 실제 이벤트 구간이 겹치는 비율을 통해 검출된 이벤트의 위치 정확도를 평가

Frame-level F1 Score

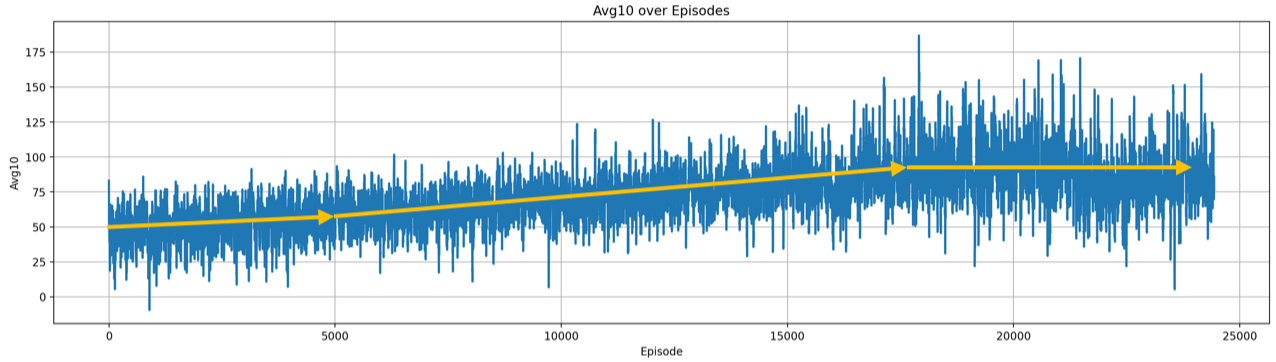
(2 · Precision · Recall) / (Precision + Recall)
각 프레임 단위 예측과 정답의 일치 여부를 기반으로 정밀도와 재현율의 조화 평균으로 평가

SECTION 08 EXPERIMENTAL RESULTS

실험 결과: 정량적 성능 분석

학습 결과 : Avg10 over Episodes

Average Reward (Moving Avg)



DQN 적용 전후 성능 비교 (Before vs After)

EVALUATION METRIC	BEFORE DQN (BASELINE)	AFTER DQN (PROPOSED)	IMPROVEMENT
Accuracy 프레임 단위 정확도	78.28%	**82.07%**	+3.79%
IoU Intersection over Union	63.30%	**64.22%**	+0.92%
F1 Score Frame-level Harmonic Mean	72.75%	**73.68%**	+0.93%

학습 경향 분석

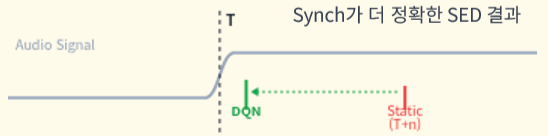
- **초기 탐색** : 0~5k 구간에서는 높은 Epsilon으로 인한 보상 수치 낮음
- **안정적 학습** : 약 5k episode부터 보상 수치가 점진적 증가
- **최적화 달성** : 초기 대비 최종 구간 높은 보상 유지하며 최적 Policy 도달

성능 향상 분석

- **Accuracy, IoU, F1 Score**가 모두 소폭 **상승**
- **상승폭이 제한적인 이유 :**
  DQN의 목적이 전체 프레임을 다시 판정하는 것이 아니라, **소리 변화가 발생하는 구간(Onset/Offset)**에서의 threshold 오차를 국부적으로 보정하는 데 있음.

  즉, 모델 본연의 Base Threshold로도 안정적으로 검출되는 대부분의 프레임은 그대로 유지하며, **오류 발생 가능성이 높은 Transition Zone만 정밀하게 보정**하도록 설계(더 정밀한 Onset 시점 검출)

Onset 구간 정밀 조정



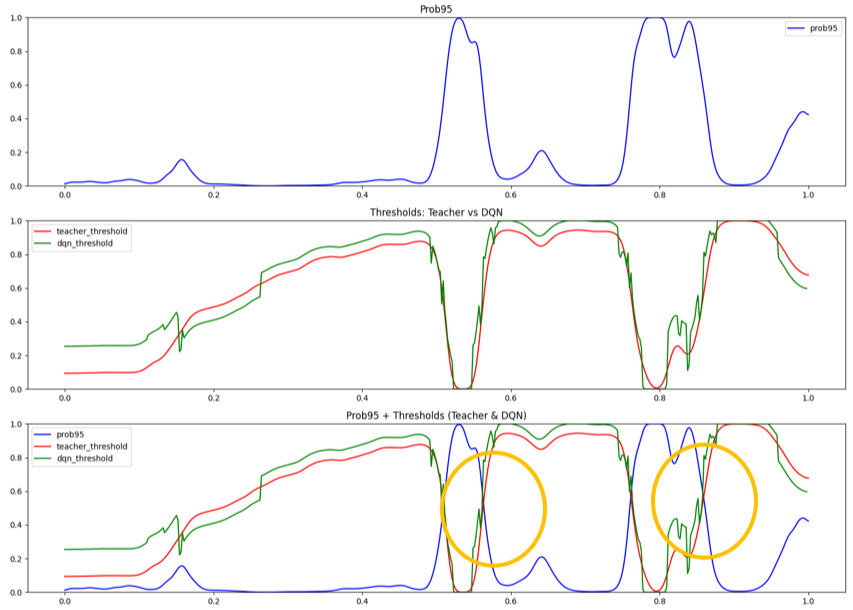
기존 검출된 Onset 지점 대비 더 **정확한 Synch Timing**

SECTION 08 EXPERIMENTAL RESULTS

실험 결과: 실험 결과 시각화

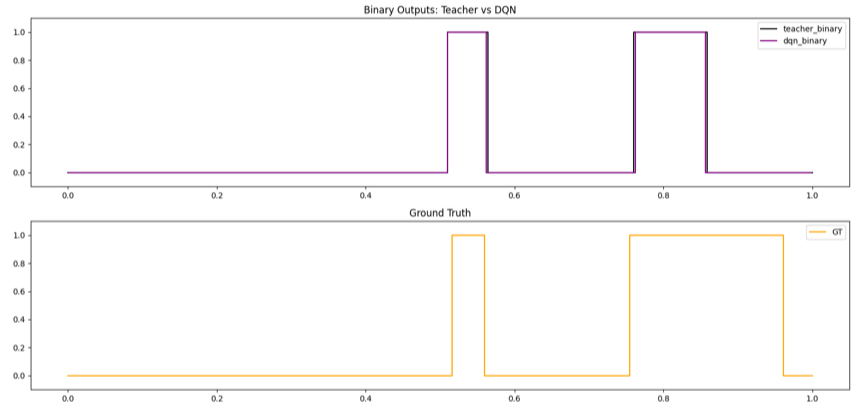
소리 존재 확률과 DQN 적용 전/후 Threshold 변화

Prob95 Base_T DQN_T



DQN 적용 전 후 정답 구간의 변화

기존 예측 DQN 추가 예측 GT



그래프 및 시각화 상세 설명

1. 소리 확률과 Threshold (좌측)

소리 **존재 확률**과 **기존 Threshold, DQN이 조정한 Threshold**의 변화를 시간 축에서 비교

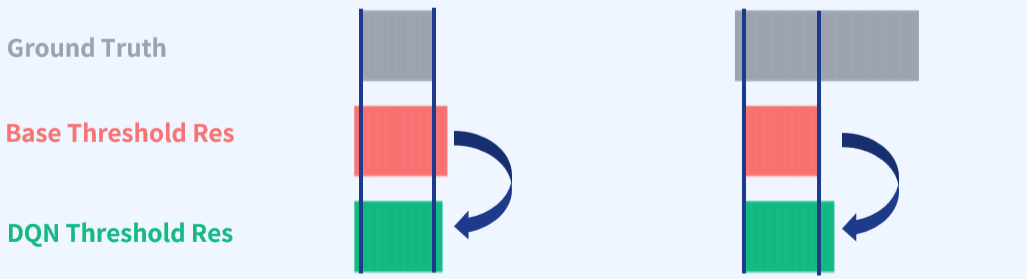
DQN은 확률 변화 패턴에 맞추어 **Threshold를 미세 조정**하여 에러 영역을 보완함

2. Threshold를 이용한 SED 결과 (우측)

기존 Threshold 기반 예측과 DQN이 조정한 Threshold 기반 예측 비교

DQN 적용 후 이벤트 검출 범위가 **GT에 더 일치하도록 개선**되는 구간 확인 가능

SED 결과 예측 변화 비교



기존 더 넓은 영역의 event 구간을 탐지
DQN 적용을 통해 좀 더 정답 방향으로 이동

기존 더 좁은 영역 Event 구간 탐지
DQN 적용을 통해 좀 더 많이 포함된 결과 예측

SECTION 09 DISCUSSION & CONCLUSION

토의 및 결론

실험 결과 고찰

KEY FINDINGS

DQN 알고리즘의 유효성 검증

DQN이 확률 변화 패턴을 학습해 기존 Base Threshold 대비
더 정답에 가까운 조정 정책 형성 확인
Accuracy, IoU, F1 Score 등 주요 성능 지표가 전반적으로 향상

Threshold 정밀 조정을 통한 Onset/Offset 구간의 보정 효과

변화가 발생하는 Transition Zone에서만 Threshold를 조정하는 정책을
학습하여 **과다/과소 검출 문제 완화**
이벤트의 시작과 종료 위치가 **GT에 더 근접하도록 보정**되는 효과 확인

학습 안정성 및 탐색 전략의 효과

초기 높은 Epsilon으로 인한 불안정 구간을 지나, 약 5k episode 이후 보상이
점진적으로 증가하며 **안정적으로 수렴**
탐색–활용 균형이 잘 유지되면서 최종적으로 일관된 Threshold 조정 정책을 학습

보완 및 개선사항

FUTURE WORKS

Reward 설계 및 Sampling 전략의 고도화

현재 Reward는 단순화된 형태로 설계되어 있어 정답 구간 근처에서의 세밀한 보상
차이가 충분히 반영되지 못한다는 한계 존재
→ Temporal consistency reward 등 더 **정밀한 Reward 설계 필요**

부족한 학습 데이터

Frame단위 오디오 라벨 데이터 확보에는 **많은 시간과 비용** 발생
→ 더 많은 라벨과 상황에 맞는 정밀 데이터 확보 필요

기존보다 발전된 모델 도입

DQN 자체 구조적 한계 존재
얕은 깊이로 간단하게 구성된 Q-Network로 인한 모델 성능 저하
→ 더 좋은 학습 환경에서 좀 더 **고도화된 모델 설계 필요**

SECTION 10 Appendix

부록 1

데이터 Preprocessing 과정 수식화

(0) Input Audio

$x(t) \in \mathbb{R}^L$

(1) RMS Energy (frame-based)

$$RMS(i) = \sqrt{\frac{1}{N}\sum_{n=0}^{N-1} x^2(iH+n)}$$

$$r_{\log}(i) = \log(1 + RMS(i))$$

$$P_{95}^{r} = \text{percentile}(r_{\log}, 95)$$

$$r_{\text{clip}}(i) = \min(r_{\log}(i), P_{95}^{r})$$

$$r_{\text{norm}}(i) = \frac{r_{\text{clip}}(i) - \min(r_{\text{clip}})}{\max(r_{\text{clip}}) - \min(r_{\text{clip}}) + \epsilon}$$

(1-1) Interpolation (RMS length → DASM length)

$$r(i) = \text{Interp}(r_{\text{norm}},\ |D_{\text{adj}}|)$$

(2) DASM Strong Score Post-processing

$$S_q(i) = f_{\text{DASM}}(x, q)$$

$$\tilde{S}_q(i) = \text{MedianFilter}(S_q(i),\ k_1)$$

$$\tilde{S}_{q,\text{med}}(i) = \text{MedianFilter}(\tilde{S}_q(i),\ k_2)$$

$$\tilde{S}_{q,\text{pow}}(i) = (\tilde{S}_q(i))^{2.2}$$

$$D_{\text{mix}}(i) = 0.7 \cdot \frac{\tilde{S}_{q,\text{med}}(i) - \min}{\max - \min + \epsilon} + 0.3 \cdot \frac{\tilde{S}_{q,\text{pow}}(i) - \min}{\max - \min + \epsilon}$$

$$P_{95}^{D} = \text{percentile}(D_{\text{mix}}, 95)$$

$$D_{\text{clip}}(i) = \min(D_{\text{mix}}(i), P_{95}^{D})$$

$$D_{\text{adj}}(i) = \frac{D_{\text{clip}}(i) - \min(D_{\text{clip}})}{\max(D_{\text{clip}}) - \min(D_{\text{clip}}) + \epsilon}$$

(3) RMS–DASM Fusion (soft probability)

$$P_{\text{raw}}(i) = r(i) \cdot D_{\text{adj}}(i)$$

$$P_{95}^{P} = \text{percentile}(P_{\text{raw}}, 95)$$

$$P_{\text{clip}}(i) = \min(P_{\text{raw}}(i), P_{95}^{P})$$

$$P_{\text{norm}}(i) = \frac{P_{\text{clip}}(i) - \min}{\max - \min + \epsilon}$$

$$P(i) = G_{\sigma=2} * P_{\text{norm}}(i)$$

(4) Local Background Modeling (RMS-based)

$$r_{\text{bg}}(i) = \text{MedianFilter}(r(i),\ w = 200)$$

$$r_{\text{bg,s}}(i) = G_{\sigma=3} * r_{\text{bg}}(i)$$

(5) Semantic Anti-mask Weight

$$w_{\text{sem}}(i) = (1 - D_{\text{adj}}(i))^{\beta}$$

$$w_{\text{sem,s}}(i) = G_{\sigma=3} * w_{\text{sem}}(i)$$

(6) Dynamic Threshold

$$T_{\text{raw}}(i) = \alpha \cdot r_{\text{bg,s}}(i) \cdot w_{\text{sem,s}}(i)$$

$$T_{\text{norm}}(i) = \frac{T_{\text{raw}}(i) - \min(T_{\text{raw}})}{\max(T_{\text{raw}}) - \min(T_{\text{raw}}) + \epsilon}$$

(7) Final Binary Label (Pseudo Label)

$$y(i) = \begin{cases} 1, & P(i) > T_{\text{norm}}(i) \\ 0, & \text{otherwise} \end{cases}$$

DQN 학습 전 State, T_base 수집 과정

1. 입력된 오디오로부터 물리적 세기 RMS 계산
   물리적 세기를 사람의 청각 인지 기준인 Log-Scale로 변환
   과도한 값 방지를 위해 95% 클리핑 후 정규화 수행
2. SED모델인 DASM에 오디오와 Text Query 입력(weak label 기반)
   Median Filter 및 mix 계산 적용
   동일하게 95% 클리핑 후 정규화
3. RMS와 DASM의 곱을 통해 각 프레임 별 소리 존재 확률 산출
   Gaussian Smoothing으로 시간적 연속성 보정
4. $((1 - DASM)^{\beta}$ 를 통해 해당 이벤트의 소리가 없는 배경 구간 추출
   이를 기반으로 Dynamic Threshold 계산
5. 계산된 확률과 T_base를 비교하여 프레임별 소리 존재 여부 결정

SECTION 10 Appendix

부록 2

학습 진행 중

```
root@faa7a39a4f89:/workspace/Transformer4SED# python DQN.py
Training:   0%|                                                  | 13/28592 [00:04<3:47:38,  2.09it/s, eps=0.999, reward=33.4, avg10=49.3]
train start
train start
train start
train start
Training:  49%|_____________________________                     | 13897/28592 [7:12:50<12:31:38,  3.07s/it, eps=0.340, reward=96.6, avg10=72.6]
```

학습 과정의 진척 상황을 확인하기 위해 tmux 세션에서 출력된 실시간 학습 로그 확인

DQN 학습이 실제 서버에서 약 20시간 이상 연속으로 진행된 로그 확인

성능 평가 결과

w/o DQN

```
Evaluating (npz binary): 100%| | 5570/5570 [00:54<00:00, 103.15it/s]

========== FINAL RESULTS (npz binary only) ==========
Accuracy : 0.7828
IoU      : 0.6330
F1 Score : 0.7275
================================================
```

w/ DQN

```
Evaluating: 100%| | 5570/5570 [4:30:02<00:00,  2.91s/it]

========== FINAL RESULTS (env mode) ==========
Accuracy : 0.8207
IoU      : 0.6422
F1 Score : 0.7368
==============================================
```