中国人民大学
RENMIN UNIVERSITY OF CHINA

数字信号处理

大作业展示

张鑫恺、张联诚、高剑章、徐十一

高瓴人工智能学院

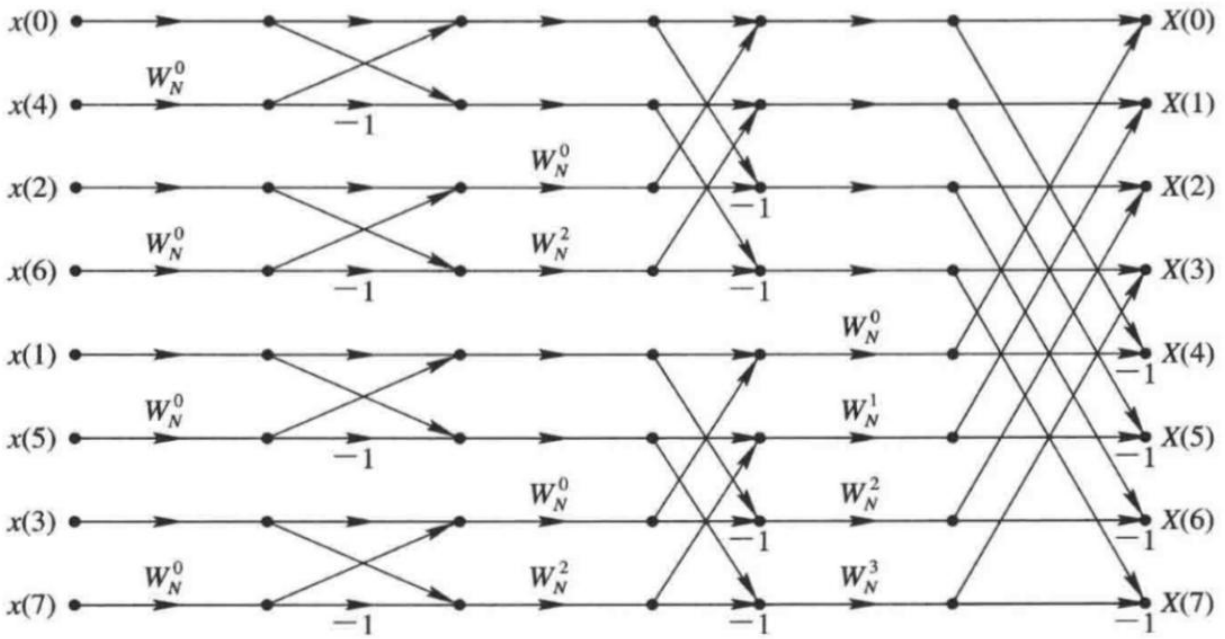
内　容

CONTENTS

算法

FFT

- N为2的幂次：蝴蝶变换
  - 位反转（bit-reverse）
    - 相当于每一位标识两组
    - 位表示等价于二叉树



序号	0	1	2	3	4	5	6	7
二进制	000	001	010	011	100	101	110	111
位翻转	000	100	010	110	001	101	011	111
排序	0	4	2	6	1	5	3	7

算法

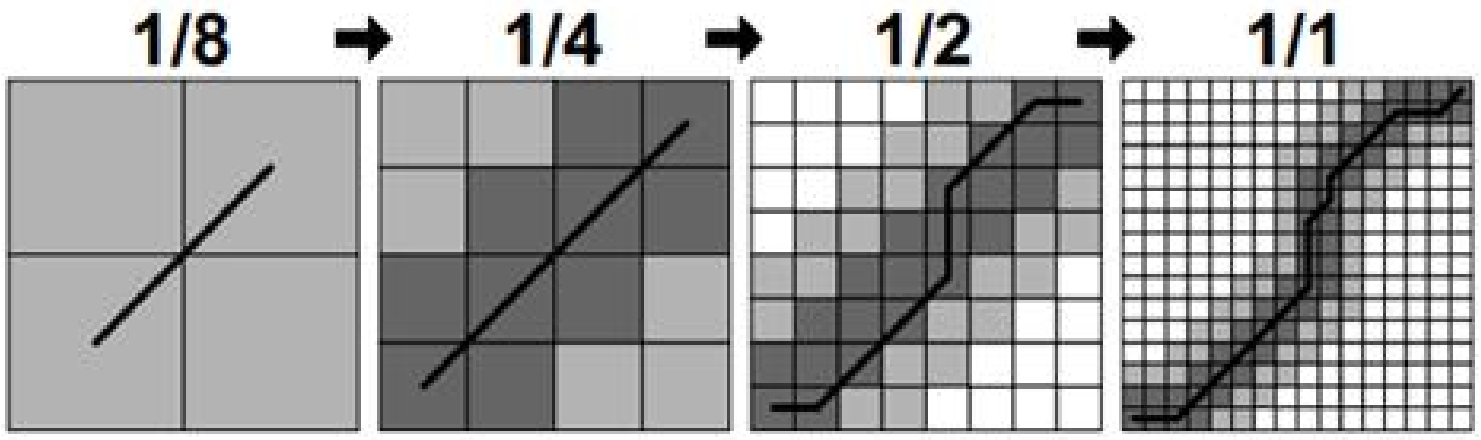
FFT

- N不为2幂次：Cooley-Tukey 混合基算法
  - 对N进行因数分解：N=PQ
    - 对P和Q执行FFT算法
  - N很接近2^k (e.g. 2^k+1)时，pad到2^(k+1)既耗空间又耗时间
  - *雷德算法：N为质数时利用整数模n乘法群和原根的性质简化运算

算法

DTW

- 动态时间规整算法，用于计算两个音频之间的相似度
- 时间复杂度为O(N^2) （如果不等长是O(MN)
  - 改进：FastDTW算法
    - 分级架构
      - 粗化
      - 投影
      - 细化
  - 时间复杂度为O(N)



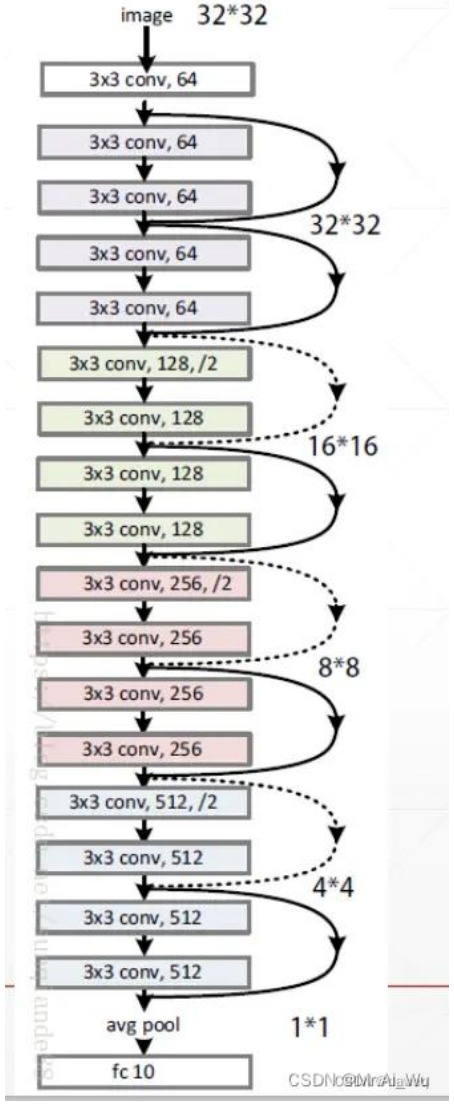
模型总览

- ResNet
- Conformer
- AST: Audio Spectogram Transformer
- BEATs: Audio Pre-Training with Acoustic Tokenizers

模型 (Backbone)

ResNet

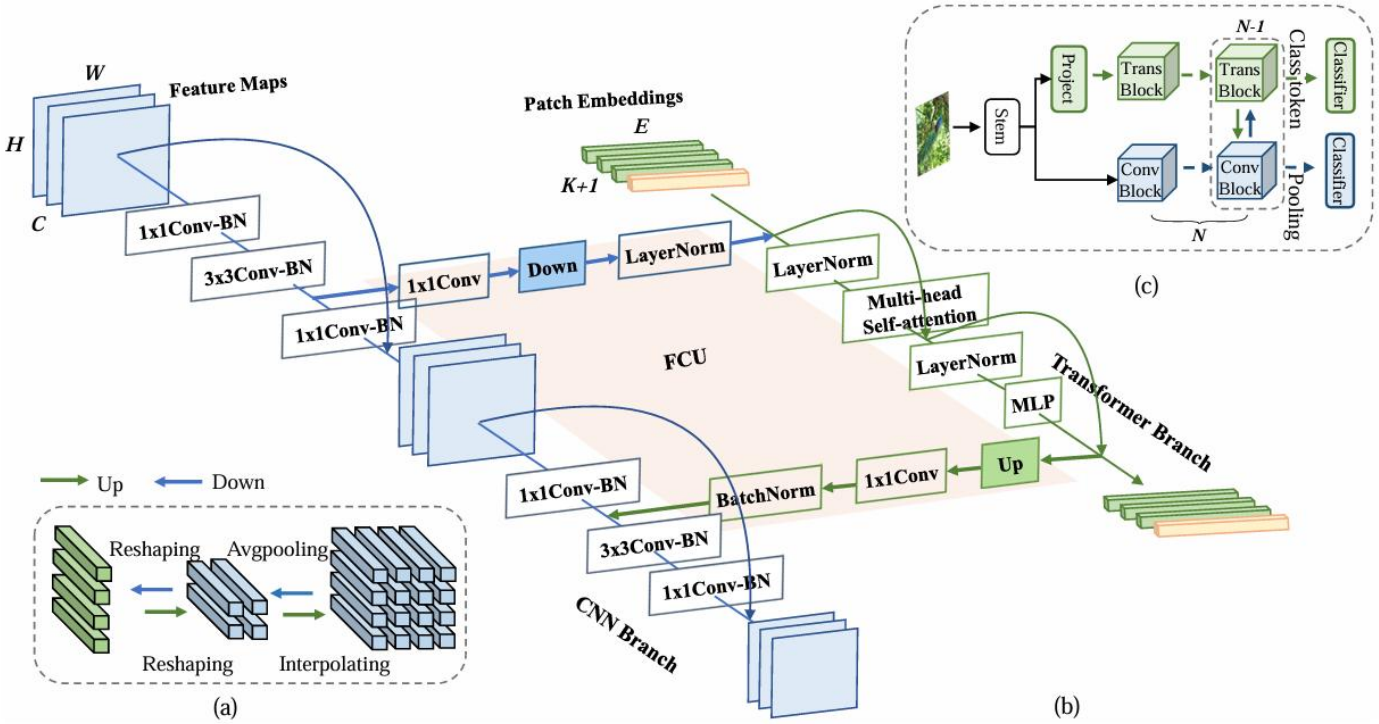
- ImageNet-1K数据集上预训练
- 图像→声音多模态迁移



模型 (Backbone)

Conformer

- Convolution + Transformer
  - CNN提取局部信息
    - 考虑高频特征
  - Transformer提取全局信息
    - 考虑低频特征

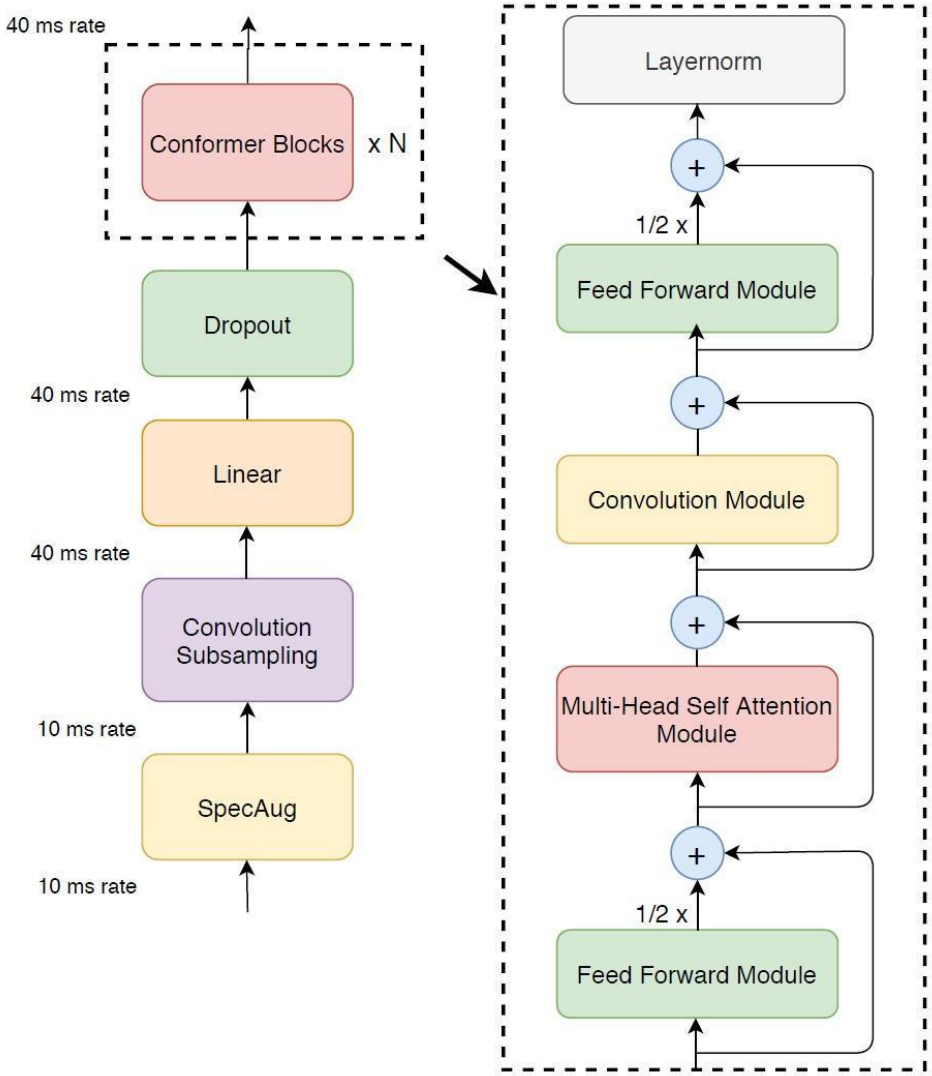


模型 (Backbone)

Conformer

- Macaron-like FFN
  - 两个FFN各1/2

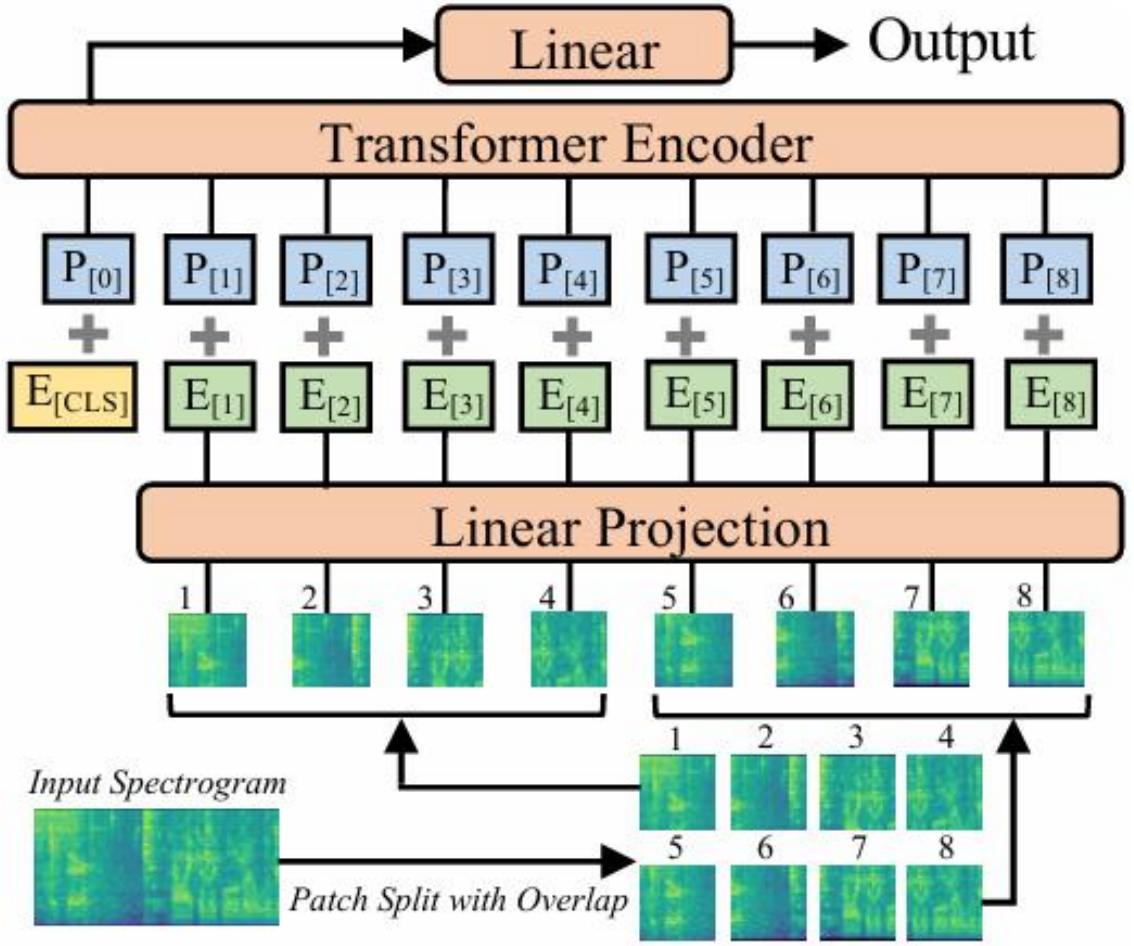
$$\tilde{x}_i = x_i + \frac{1}{2}\mathrm{FFN}(x_i)$$
$$x'_i = \tilde{x}_i + \mathrm{MHSA}(\tilde{x}_i)$$
$$x''_i = x'_i + \mathrm{Conv}(x'_i)$$
$$y_i = \mathrm{Layernorm}(x''_i + \frac{1}{2}\mathrm{FFN}(x''_i))$$



模型

AST

- Audio Spectrogram Transformer
- 多模态知识迁移
  - ImageNet上预训练的ViT
- Learnable Positional Encoding



模型

BEATs

- Audio Pre-Training with Acoustic Tokenizers
- 类似GAN的架构
  - 交替优化
    - Acoustic Tokenizer
    - Self-Supervised Learning (SSL) Model
- 本次实验采用AudioSet-2M上预训练的模型

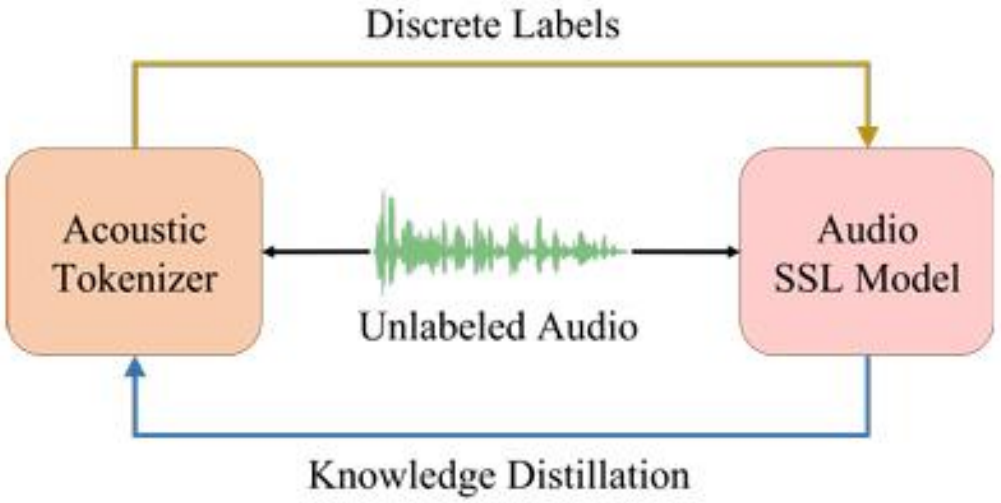


Figure 1: Iterative audio pre-training of BEATs.

模型

BEATs

- Acoustic Tokenizer
- 总体思想
  - 字典学习
  - Nearest Neigbor
  - Self-Distillation
  - 冷启动/热启动

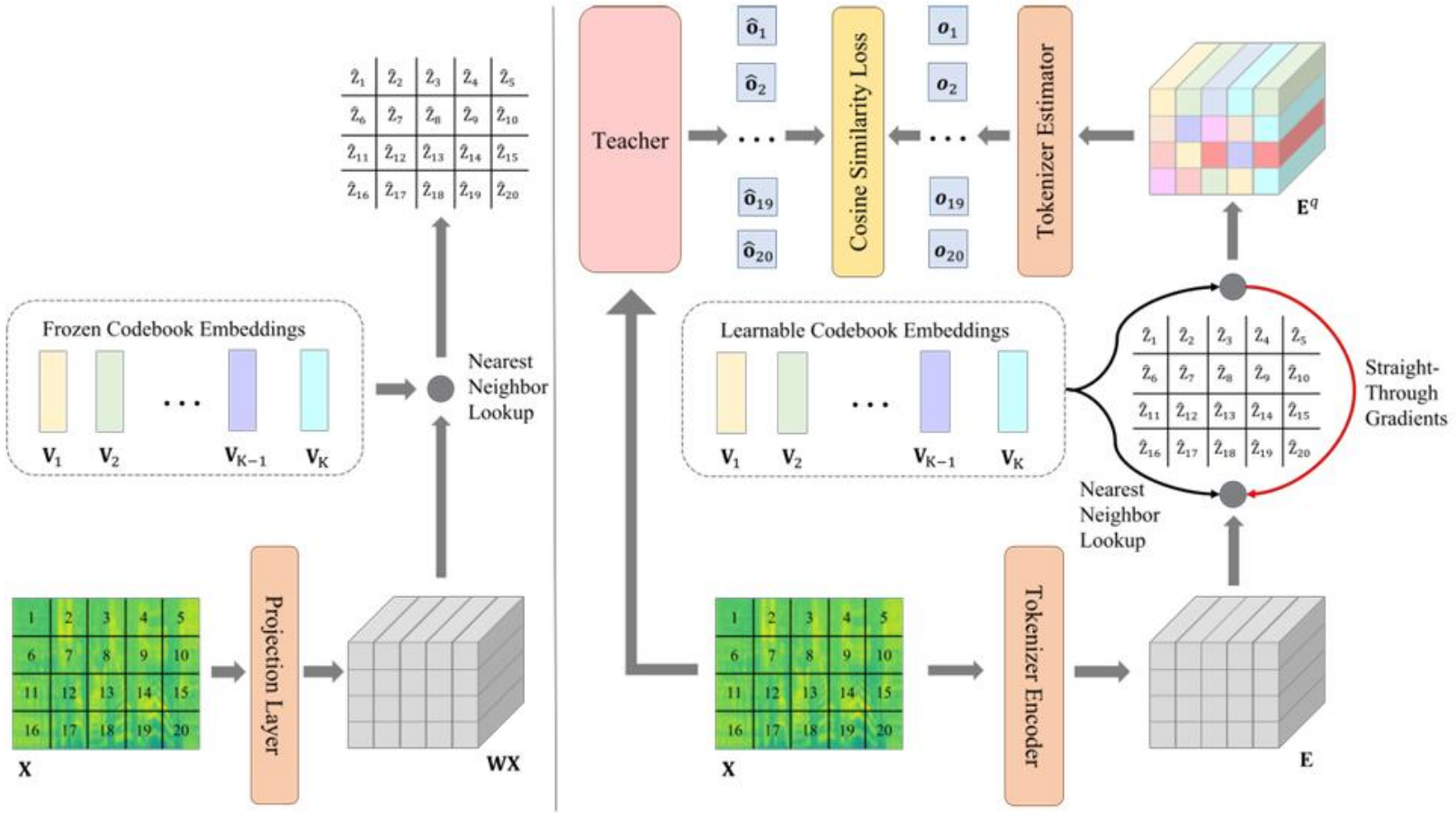


Figure 2: Acoustic tokenizers for discrete label generation.

模型

BEATs

- Audio SSL Model
- 总体思想
  - 类似Masked AE
  - Pretrain
  - Finetune

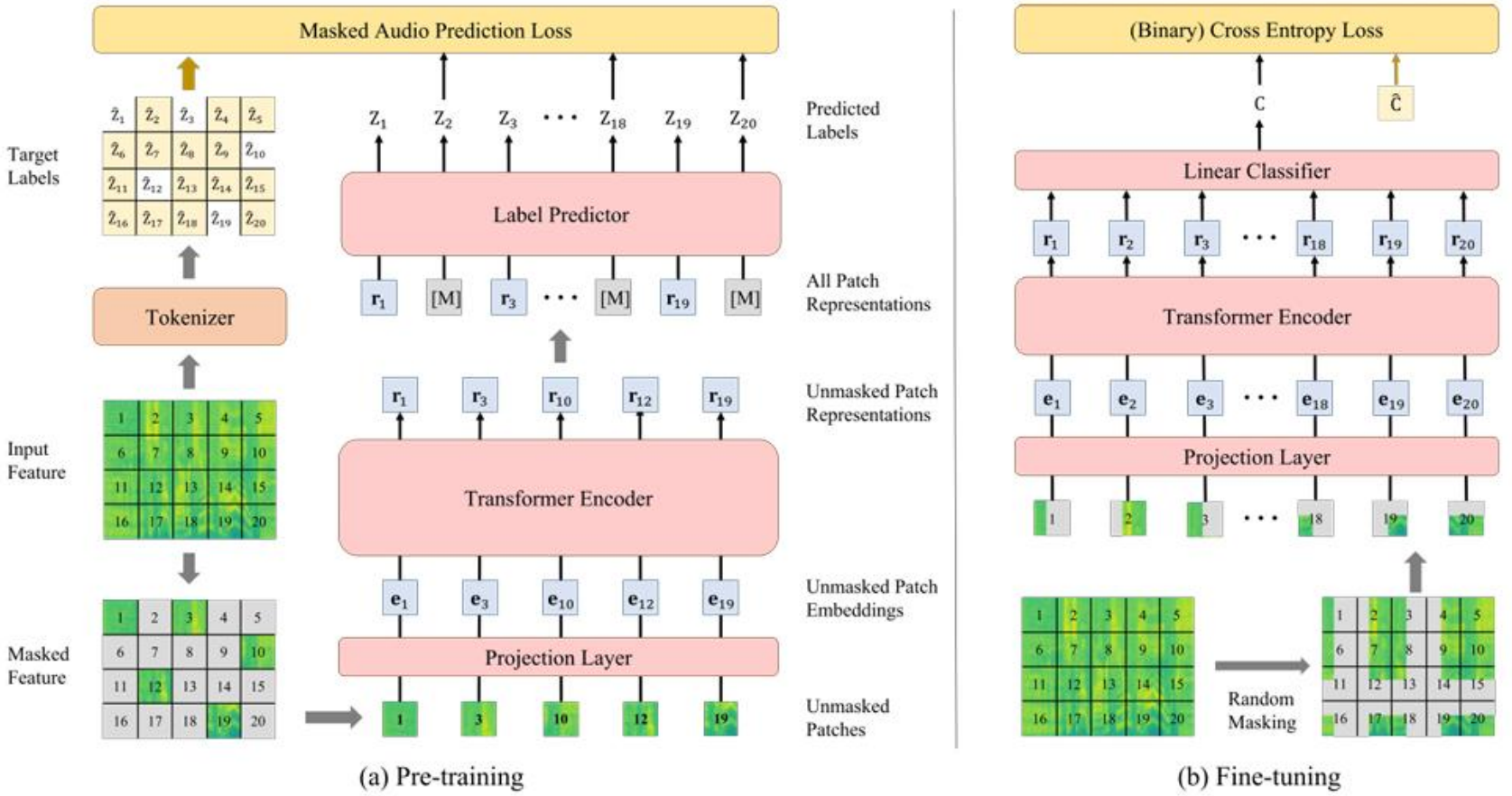


Figure 3: Overview of audio SSL model pre-training and fine-tuning.

实验设计

- **交叉验证**
  - 前四fold用于训练，最后一个fold用于测试
    - 4-fold 交叉验证

- **参数初始化**
  - Xavier Initialization
  - Kaiming Initialization

实验设计

- **数据增强**
  - Time Mask & Frequency Mask
  - 同label线性插值
  - **STFT线性性**
    - **先转换为STFT再数据增强**
      - 先增强：O(KNlogN) (K为增强的样本数量)
      - 先STFT：O(K)

分类

- 基础思想
  - 对audio做变换
    - STFT
    - Mel-Spectogram
    - MFCC
  - 通过backbone得到表征，最后映射到一个num_classes大小的分布上。

分类实验结果

- 预训练模型表现最佳
  - AST表现最优，其次是BEATs，随后是ResNet18的表现最好
  - 而未预训练的模型的表现均不太理想
    - AST（w.o.) 远差于 AST (pretrain)
    - Conformer比AST好
      - 可能是用了声音信号的卷积特性

算法	Accuracy
ResNet18	83.25%
Conformer	54.25%
AST(w.o.)	38.00%
AST(pretrain)	89.50%
BEATs	89.25%

表 5: 不同模型的分类 accuracy

分类不同帧长和帧移实验结果

算法	(1024,512)	(2048,512)	(4096,1024)	(8192,2048)	(2048,1024)	(2048,2048)
ResNet18	-	83.25%	75.50%	62.00%	77.50%	70.25%
AST(pretrain)	89.50%	87.50%	83.75 %	80.00%	88.25%	80.75%

表 4: 不同窗口的 accuracy

- 在一个相对最优的组合下，增大帧长或帧移均会使分类的结果有着不同程度的下降
- 最佳组合的出现并不是单峰的，而是可能存在多个十分接近的峰值点

检索

- 总体思想
  - 对比学习
  - Supervised Contrastive Learning (SCL)
  - 如何得到表征 ：

    用前述backbone作用于经过STFT变换后的频谱

    频谱是双通道的，取了STFT的实部和虚部

检索

- Supervised Contrastive Learning
- 用scaled dot-product
  - cosine similarity得到的loss有和batch size相关的上下界，可能影响参数选择

Eqs. 2 and 3 present the two most straightforward ways to generalize Eq. 1 to incorporate supervision.

$$\mathcal{L}_{out}^{sup} = \sum_{i \in I} \mathcal{L}_{out,i}^{sup} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_p / \tau\right)}{\sum_{a \in A(i)} \exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_a / \tau\right)} \tag{2}$$

$$\mathcal{L}_{in}^{sup} = \sum_{i \in I} \mathcal{L}_{in,i}^{sup} = \sum_{i \in I} -\log \left\{ \frac{1}{|P(i)|} \sum_{p \in P(i)} \frac{\exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_p / \tau\right)}{\sum_{a \in A(i)} \exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_a / \tau\right)} \right\} \tag{3}$$

检索实验结果

算法	利用神经网络	利用标签	利用预训练	MRR@20	Recall@10	Recall@20
DTW	✗	✗	✗	0.45	23.28%	19.23%
Beats	✓	✓	✓	0.86	64.55%	51.31%
SimCLR(resnet18)	✓	✗	✗	0.58	33.30%	27.14%
SCL(resnet18)	✓	✓	✗	0.48	47.42%	47.44%
SCL(resnet18 pretrain)	✓	✓	✓	0.70	70.87%	70.87%
SCL(conformer)	✓	✓	✗	0.59	58.02%	57.64%
SCL(AST)	✓	✓	✓	0.87	81.85%	80.85%

表 1: 不同算法的检索指标

- 未使用机器学习的DTW算法效果最差
  - DTW算法由于使用STFT维度太高，受硬件限制，因此使用MFCC作为特征
- 利用标签的有监督学习效果优于自监督学习（可能是因为数据量太小）。
- 利用预训练参数初始化也有利于提升效果
- AST的表现仍然最佳

检索实验结果

算法	(2048,512)	(4096,1024)	(8192,2048)	(2048,1024)	(2048,2048)
SCL-ResNet(stft)	47.44%	41.53%	34.00%	44.81%	34.55%
SCL-ResNet(mel)	60.75%	54.30%	46.56%	52.29%	49.59%
SCL-ResNet(mfcc)	58.54%	59.13%	57.97%	61.78%	51.80%

表 2: 不同窗口的 Top20 召回率

- 纵向对比，总体来说，MFCC的效果最佳，mel其次，STFT较差
- Mel和MFCC的效果都好于简单STFT，但在帧移小，特征丰富时Mel效果好于MFCC，帧移大，特征少时MFCC效果好于Mel

检索实验结果

- 利用在**分类任务上微调得到的BEATs**模型的Encoder部分编码声音向量

(帧长, 帧移)	Recall@10	Recall@20	MRR@20
(1024, 512)(pretrain)	64.55%	51.31%	0.86
(1024, 512)	88.80%	88.95%	0.89
(2048, 512)	84.23%	83.64%	0.86
(1024, 512)	80.80%	80.03%	0.82
(1024, 1024)	87.90%	87.58%	0.89

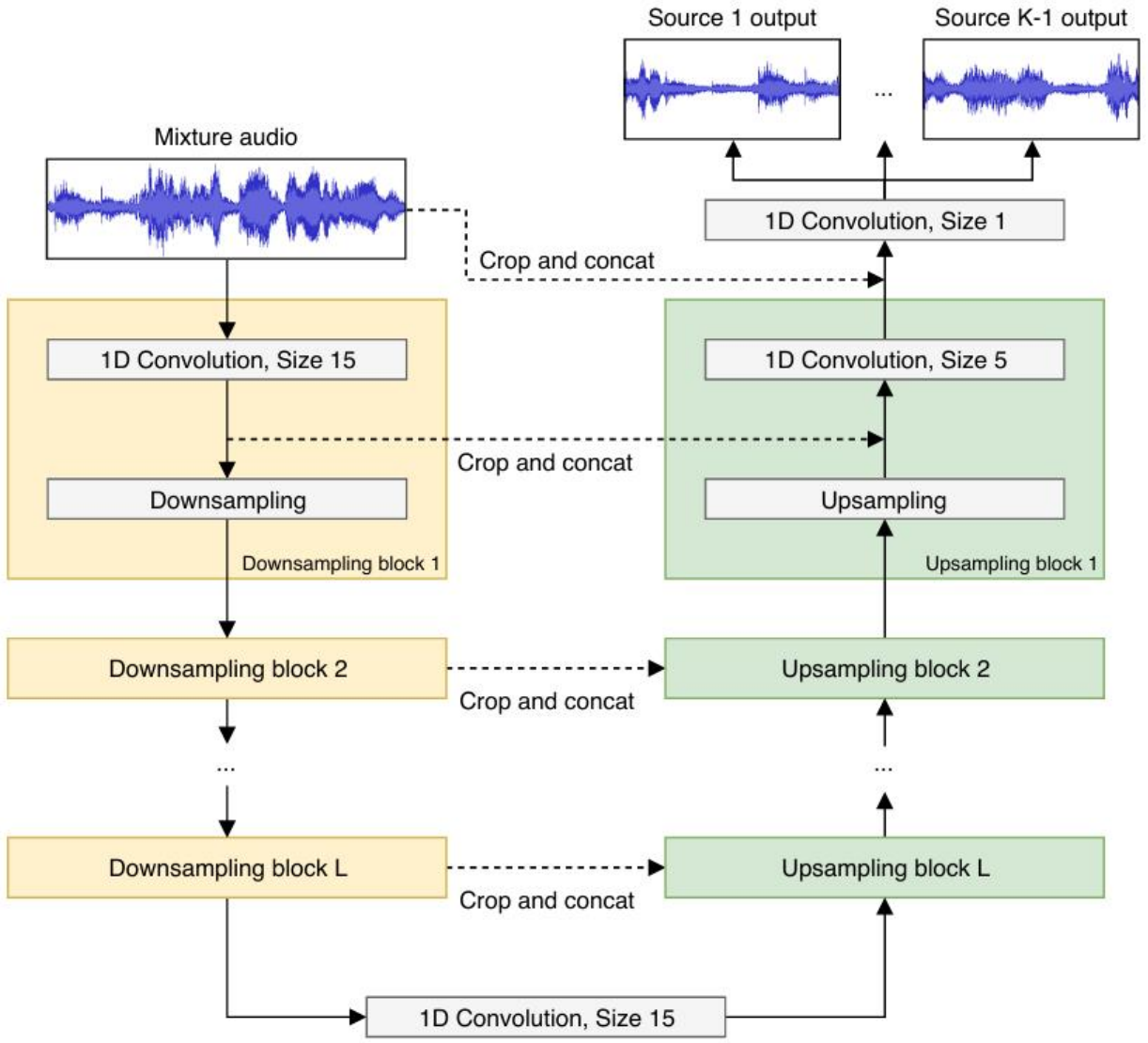
表 6: BEATs 的 Recall 和 MRR

- 在检索时可能模型某种程度上也利用到了latent的分类信息

拓展：音源分离

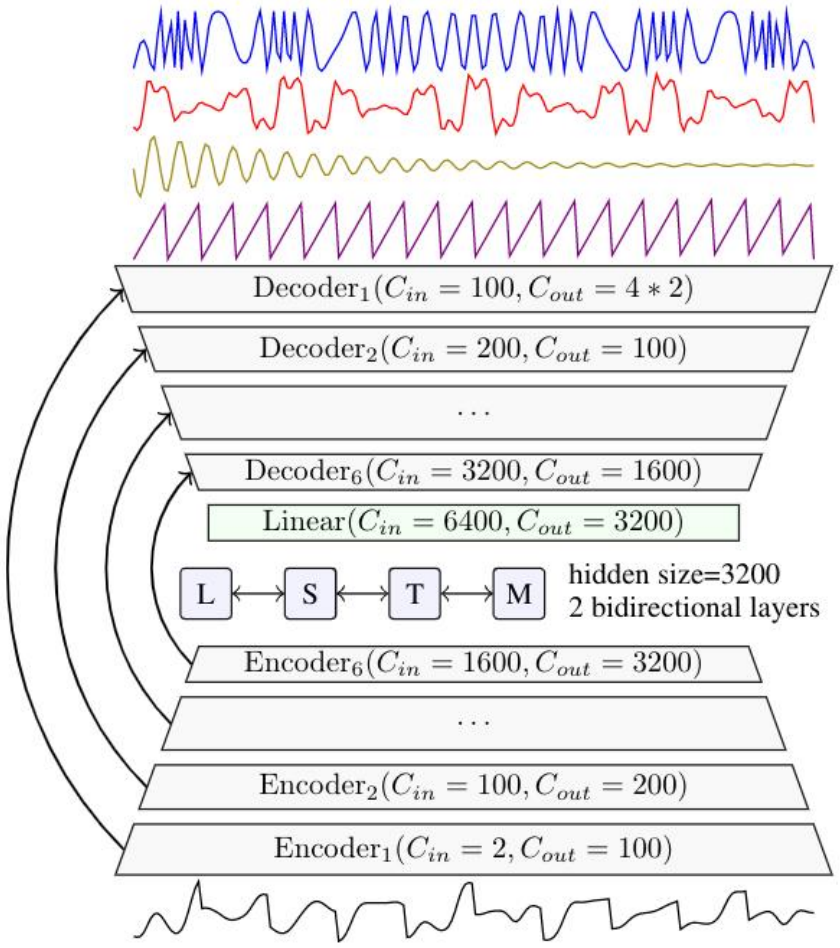
Wave-UNet

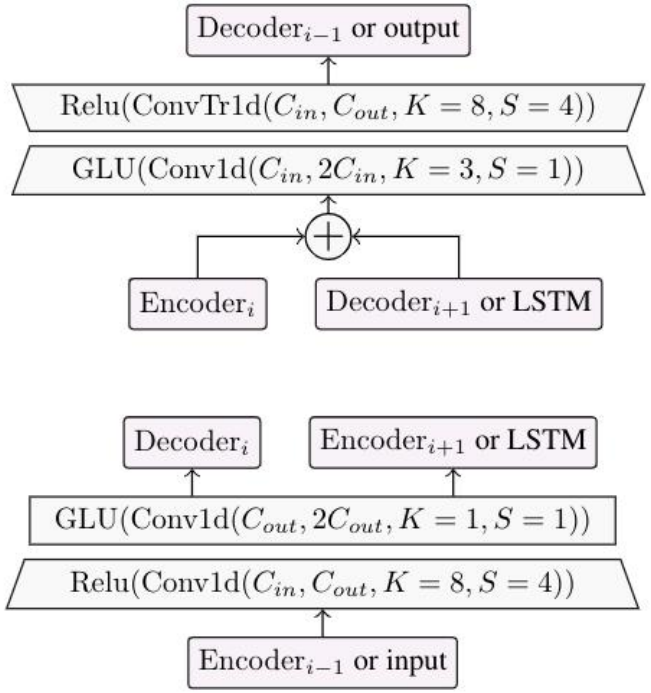
- 端到端音源分离
- UNet架构
- Permutation Invariant Training (PIT)
  - 音源的排列不变性



拓展：音源分离

- **Demucs**
- 结合时域和频域信息
- 改进版UNet





CSDN @小殊小殊

拓展：音源分离

实验结果

- 猫狗声音分离
  - 数据集太小了
    - 原始混合音频
    - （理论上应该是）分离出来的犬吠音频
- 在别的数据集上pretrain的结果
  - 音乐分离
    - 原始音乐
    - 人声
    - 鼓声

实验局限和未来工作

- 受限于时间和设备限制
  - 消融实验没有跑得很完整
- 时间信息太长的问题
  - 简单降采样可能导致信息丢失比较验证
  - 沿时域 or 频域 PCA
  - 分割时间片采用MoE架构
- 音源分离
  - 可以在更大的数据集上再完善

中国人民大学
RENMIN UNIVERSITY OF CHINA

数字信号处理

大作业展示

张鑫恺、张联诚、高剑章、徐十一

高瓴人工智能学院