Paper ID: 4255

Unsupervised Contrastive Learning of Sound Event Representations

Eduardo Fonseca[1]*, Diego Ortego[2]*, Kevin McGuinness[2], Noel E. O'Connor[2] and Xavier Serra[1]

[1]Music Technology Group, Universitat Pompeu Fabra, Barcelona

[2]Insight Centre for Data Analytics, Dublin City University

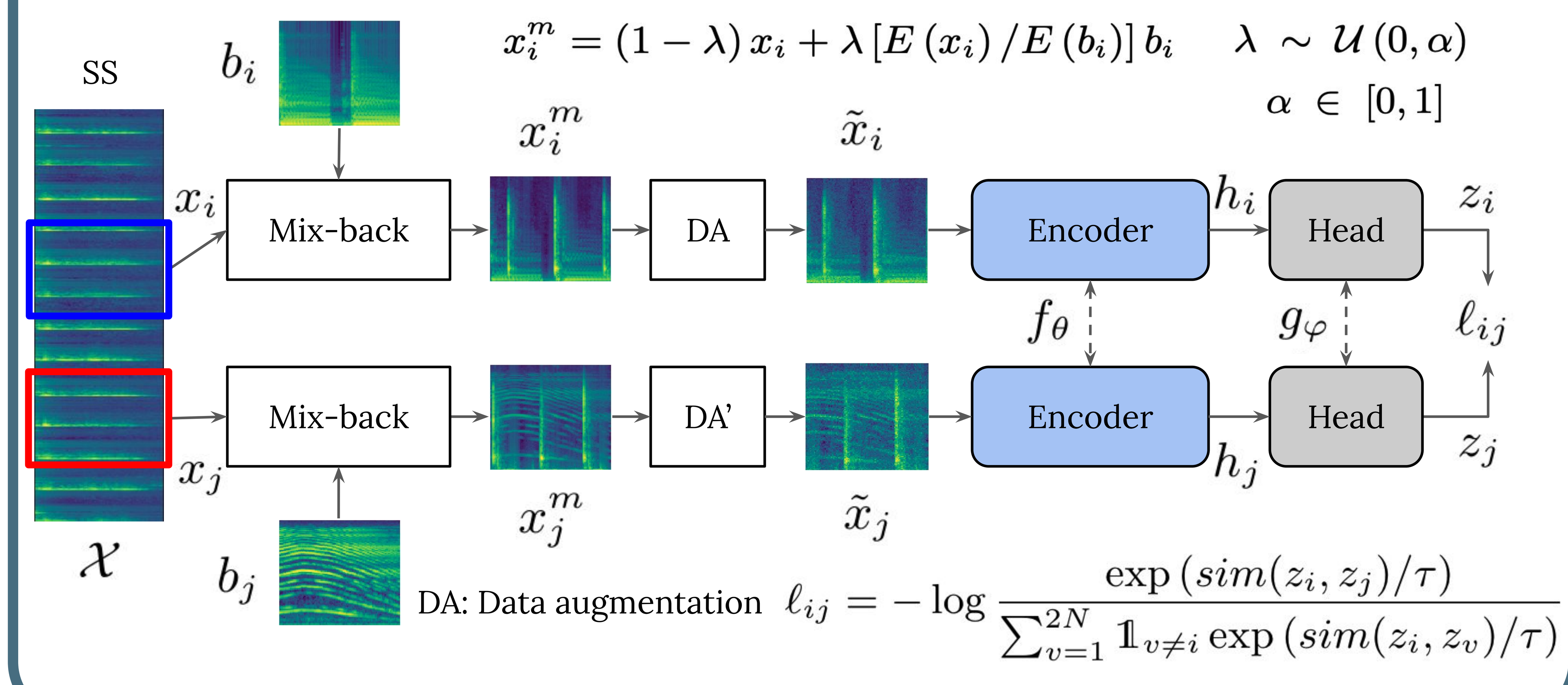
*Equal contribution

ICASSP 2021 TORONTO Canada June 6-11, 2021 Metro Toronto Convention Centre

Context

- **Task**: learn sound event representation in unsupervised fashion
- **Motivation**: common scenario in sound event research
  - few manually labeled / **abundant unlabeled data**
- **Self-supervised learning**:
  - learn representation from data w/o explicit labels
  - generate pseudo-labels, ŷ, from the data itself
  - design **proxy task** → useful representations emerge
- **Contrastive learning** is learning by comparing pairs of examples:
  - **positive** pairs of **similar** inputs
  - **negative** pairs of **unrelated** inputs
- Goal is an embedding space where representations ...
  - of **similar** examples → **close** together
  - of **dissimilar** examples → **further** away

Proposed Approach https://github.com/edufonseca/uclser20



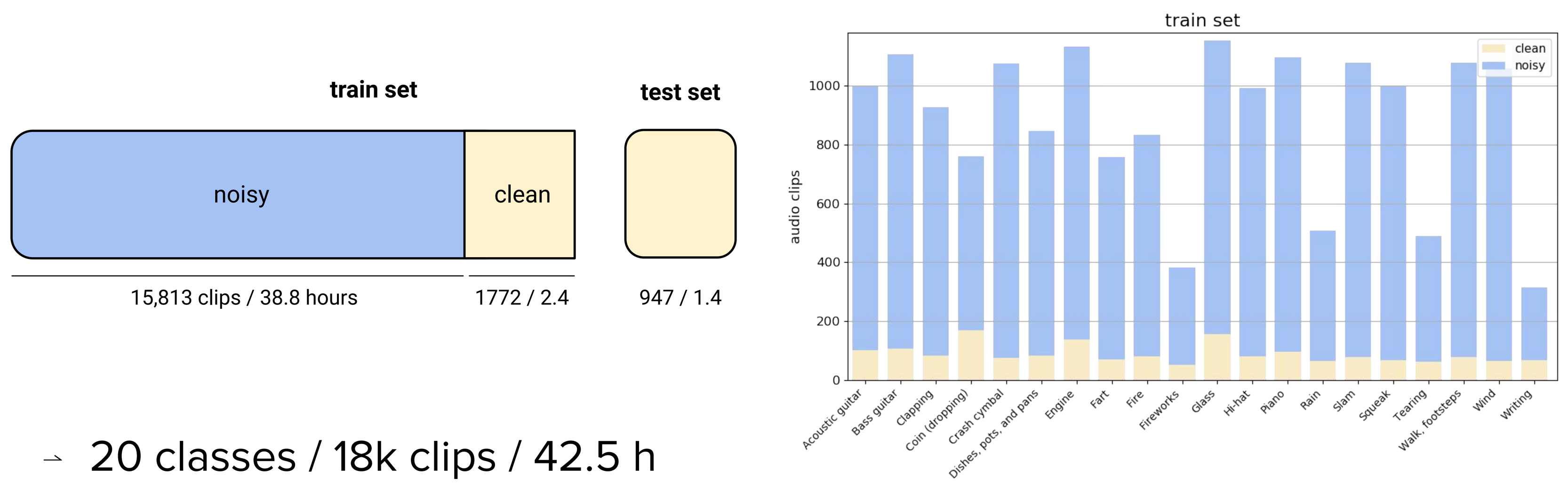
System Description

- **Proxy task**: maximize similarity between differently augmented **views** of sound events, inspired by SimCLR [1]

1. **Sampling TF patches** (aka Temporal Proximity)
   - sample two patches (101x96) at random within audio clip spectrogram
   - temporal coherence among neighbouring patches → natural data augmentation
2. ***Mix-back***: *Mix* incoming patch with a *back*ground patch
   - reduce mutual information while keeping semantics
   - energy adjustment ensures that x_i is always dominant over b_i
3. **Stochastic Data Augmentation**
   - directly over TF patches
   - simple for on-the-fly computation
   - **random resized cropping (RRC), compression, Gaussian noise addition, specAugment [4]**, random time/frequency shifts, Gaussian blurring
   - hyper-parameters randomly sampled from a distribution for each patch

- **Convolutional encoder**
  - extract low-dimensional embeddings *h*
  - once the training is over, *h* is used for downstream tasks
  - ResNet-18 / VGG-like / CRNN after removing classification layer
- **Projection Head**
  - map h to L2-normalized metric embedding *z*, where loss is applied
  - MLP w/ one hidden layer + BNorm + ReLU
- **Normalized temperature-scaled cross-entropy (NT-Xent) loss [1]**
  - softmax structure
  - scoring function: cosine similarity with temperature scaling τ
  - maximize similarity between differently augmented views

Evaluation using FSDnoisy18k [2]



- 20 classes / 18k clips / 42.5 h
- singly-labeled data → accuracy as metric
- proportion train_noisy / train_clean = 90% / 10%
- per-class varying degree of label noise
- www.eduardofonseca.net/FSDnoisy18k/

Two stages:

1. **Unsupervised representation learning**
   - train on *train_noisy* / validate on *train_clean* using labels in **kNN eval**
     - pairwise cosine similarity on z
     - prediction by majority voting across k=200 neighbours
2. **Evaluation of the representation** using supervised tasks (w/ labels)
   - **Linear Eval**: train **additional linear classifier** on top of embeddings
     - train on train_noisy / validate on train_clean
   - **End-to-end Fine Tuning**: **fine-tune model** on two downstream tasks after initializing with pre-trained weights:
     1. train on train_noisy / validate on train_clean
     2. train on train_clean (allow 15% for validation)

Results

Sampling patches

- **best**: sampling at random
- worst: using same patch
- overlapping patches → detrimental
- results accord with [3]
- effective

Table 1. kNN val accuracy for several ways of sampling TF patches.

Sampling method	kNN	Sampling method	kNN
Sampling at random	**70.1**	$d = 125$	67.9
$d = 0$ (same patch)	51.1	$d = 200$	69.9
$d = 25$	61.5	$d = 300$	68.5
$d = 75$	65.1	$d = 400$	69.7

Mix-back

- mixing patches with unrelated backgrounds helps
- adjusting the energy is also beneficial
- prevent aggressive transforms & keeping semantics

Mix-back setting (α)	kNN
w/ E adjustment (0.05)	**70.1**
w/o E adjustment (0.02)	66.2
w/o mix-back	63.3

Data Augmentation

1. Explore DAs applied individually
   - random resized cropping: stretch & freq transposition
   - SpecAugment (time/freq masking) [4]
2. Explore DA compositions based on RRC
   - RRC + compression + Gaussian noise addition
   - RRC + SpecAugment
   - more exhaustive exploration → better results

DA policy	kNN
RRC + comp + noise	**70.1**
RRC + comp	69.6
RRC + specAugment	70.0
RRC	69.0
specAugment [20]	68.0
w/o DA	60.1

Evaluation of learned representations

- **Supervised baselines:** CRNN ≅ VGG-like > ResNet-18
- **Linear Eval:**
  - ResNet-18 is top: larger capacity is better for unsupervised contrastive learning
  - exceeds supervised performance
  - VGG-like & CRNN: recover most of supervised perf

Table 3. Test accuracy for linear eval & two downstream tasks.

Model	Linear	Larger noisy set		Small clean set	
(weights in M)	-	random*	p-t	random	p-t
ResNet-18 (11)	**74.3**	65.4	**78.2**	56.5	**77.9**
VGG-like (0.3)	70.0	70.6	**72.8**	61.1	**72.3**
CRNN (1)	64.4	72.0	**74.2**	58.7	**69.1**

- **Fine tuning**
  - our method is best always
  - ResNet-18
    - worst from scratch
    - top with unsup pre-training
  - Greater improvements in "smaller clean" task
  - Pre-trained performance → little degradation between tasks: why?
    - "smaller clean" task: fine tune on **unseen clean** data (albeit small)
    - "larger noisy" task: fine tune on **same** data used for unsupervised learning (now affected by **label noise**)

References

[1] Chen et al., A Simple Framework for Contrastive Learning of Visual Representations. ICML 2020
[2] Fonseca et al. Learning Sound Event Classifiers from Web Audio with Noisy Labels. ICASSP 2019
[3] Tian et al., What Makes for Good Views for Contrastive Learning? NeurIPS 2020
[4] Park et al., SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. InterSpeech 2019