Self-Supervised Learning from Automatically Separated Sound Scenes

Eduardo Fonseca[12], Aren Jansen[1], Daniel P. W. Ellis[1], Scott Wisdom[1], Marco Tagliasacchi[1], John R. Hershey[1], Manoj Plakal[1], Shawn Hershey[1], R. Channing Moore[1], Xavier Serra[2]

[1] Google Research [2] upf. Universitat Pompeu Fabra Barcelona MTG Music Technology Group

WASPAA 2021

Context

- **Task**: learn audio representation from unlabeled data

- **Self-supervised learning**
  - Learn representation from unlabeled data without external supervision
  - **Proxy learning task**:
    - Generate pseudo-labels from patterns in data
    - Learn mapping from inputs to low dimensional representations
  - Use representations for downstream tasks e.g. sound event classification

Self-Supervised Contrastive Representation Learning

- Contrastive learning is learning by comparing
  - We compare **pairs of input 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

Building a Proxy Learning Task

To compare **pairs of positive** examples:

1. **How to generate the pairs of positive examples?**
2. **Once generated, how to compare them?**

How to Generate Pairs of Positive Examples?

- Data augmentation ⟶ differently-augmented **views** of the same input example
- Previously, composition of augmentations:
  - sampling nearby audio frames
  - artificial mixing
  - time/freq masking
  - cropping
  - shifts
  - …

How to Generate Pairs of Positive Examples?

- Data augmentation ⟶ differently-augmented **views** of the same input example
- Previously, composition of augmentations:
  - sampling nearby audio frames
  - artificial mixing
  - time/freq masking
  - cropping
  - shifts
  - …
- Artificial & handcrafted transformations with tunable hyperparameters
- Risk of introducing somewhat unrealistic domain shift?

How to Generate Pairs of Positive Examples?

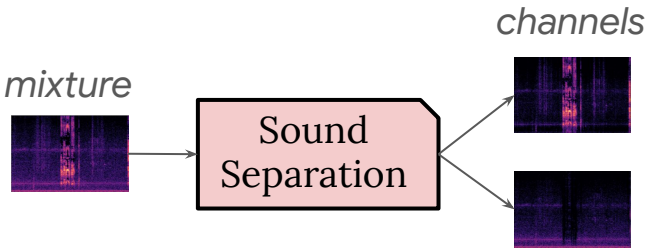
- Real-world sound scenes: time-varying collections of sound sources
  - **Mixture of sound events**

- Association of sound events with mixture and each other is **semantically constrained**
  - Not all classes co-occur naturally

Sound Separation to Generate Views for Contrastive Learning

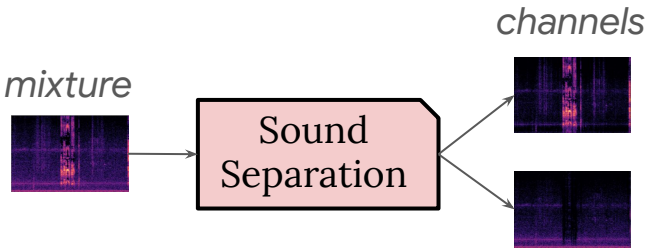
- Decompose sound scene (**mixture**) into M semantically-linked views
  - Simpler **separated channels** share semantics with mixture and with each other

- Unlike previous approaches to generate views, sound separation:
  - input-dependent / ecologically valid views / reduces need for parameter tuning



Sound Separation to Generate Views for Contrastive Learning

- Decompose sound scene (**mixture**) into M semantically-linked views
  - Simpler **separated channels** share semantics with mixture and with each other
- Unlike previous approaches to generate views, sound separation:
  - input-dependent / ecologically valid views / reduces need for parameter tuning
- Comparing ***mixture* vs *channel*** meets recommended guidelines
  - Mutual information between views is reduced
  - Some relevant semantic information is preserved



How to Compare Pairs of Examples?

Two popular proxy tasks:

- **Similarity Maximization (SimCLR)**
  - Maximize the similarity between differently-augmented views
- **Coincidence Prediction**
  - Predict whether a pair of examples occurs within a temporal proximity

Chen et al., **A Simple Framework for Contrastive Learning of Visual Representations**. ICML 2020
Fonseca et al., **Unsupervised Contrastive Learning of Sound Event Representations**, ICASSP 2021
Jansen et al., **Coincidence, Categorization, and Consolidation: Learning to Recognize Sounds with Minimal Supervision**. ICASSP 2020

How to Compare Pairs of Examples?

Two popular proxy tasks:

- **Similarity Maximization (SimCLR)**
  - Maximize the similarity between differently-augmented views
- **Coincidence Prediction**
  - Predict whether a pair of examples occurs within a temporal proximity

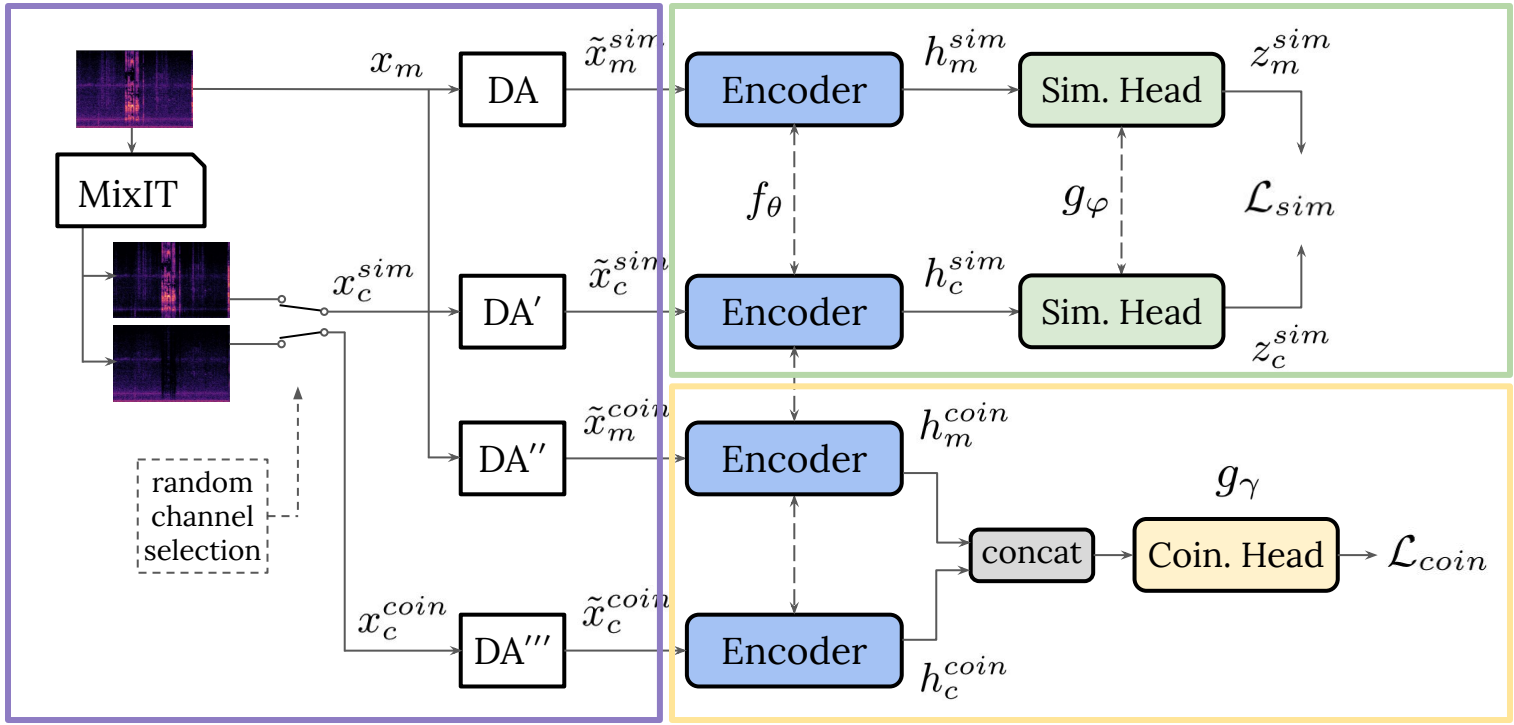
- We propose to **optimize them jointly** as a multi-task objective
- Same goal ⟶ semantically structured embedding space, pursued in different way
  - **SM**: Co-locate representations of positives
  - **CP**: Weaker condition → get a representation that supports coincidence prediction

Chen et al., **A Simple Framework for Contrastive Learning of Visual Representations**. ICML 2020
Fonseca et al., **Unsupervised Contrastive Learning of Sound Event Representations**, ICASSP 2021
Jansen et al., **Coincidence, Categorization, and Consolidation: Learning to Recognize Sounds with Minimal Supervision**. ICASSP 2020

Proposed Approach: Overview

Sound separation augmentation front-end

Similarity maximization

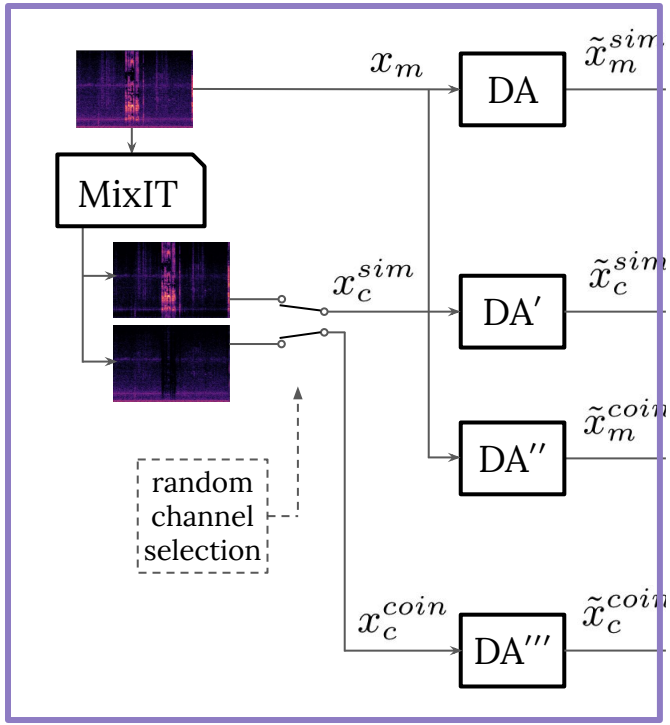


Coincidence prediction

Google Research

Augmentation Front-end

Sound separation
augmentation front-end



MixIT for Unsupervised Sound Separation

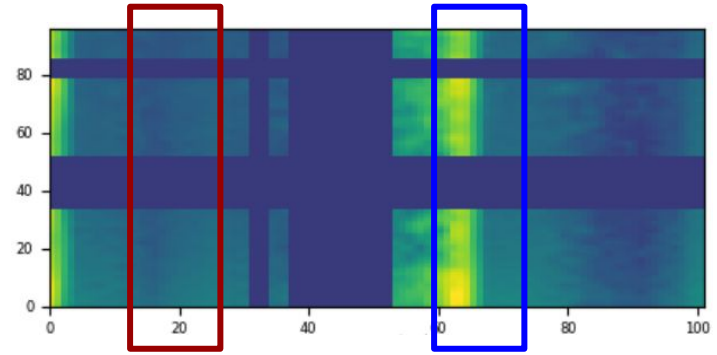
- **Mixture invariant training (MixIT)**
  - Model is tasked to separate mixtures of audio clips
  - Fully **unsupervised**
  - Promising results in Universal Sound Separation

- Separation model:
  - Improved time-domain convolutional network (TDCN++)
    - Similar to Conv-TasNet

Wisdom et al., **Unsupervised Sound Separation Using Mixture Invariant Training**. NeurIPS 2020
Kavalerov et al., **Universal Sound Separation**. WASPAA 2019

Composition of Data Augmentation Methods

- Composing multiple augmentations is important
  - more challenging proxy learning task ⟶ better representation
- We combine sound separation with
  - **Temporal proximity** sampling
  - **SpecAugment**
    - time/freq masking
    - mild time warping

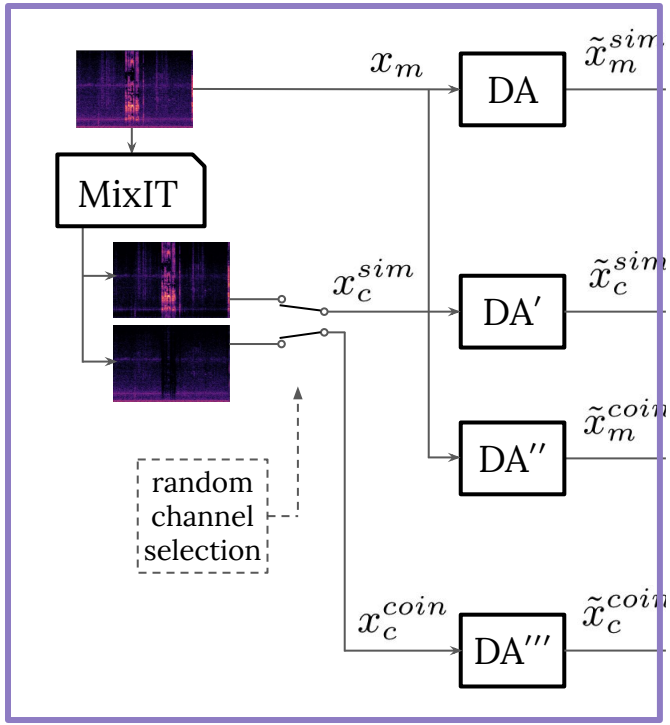


Jansen et al., **Unsupervised learning of semantic audio representations**. ICASSP 2018

Park et al., **SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition**. Interspeech 2019

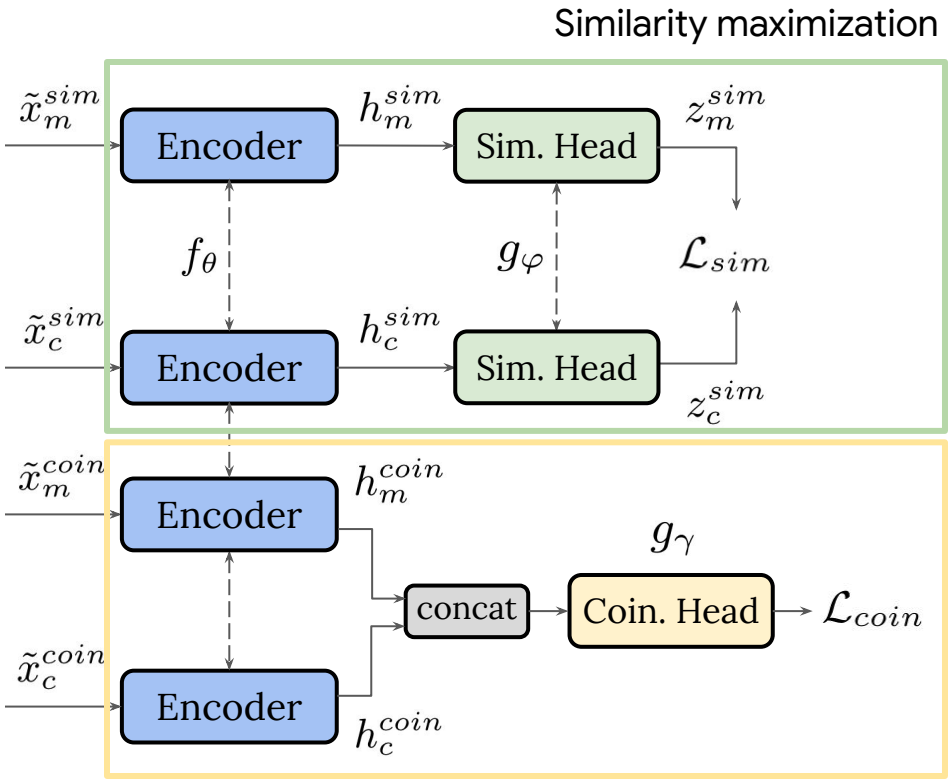
Augmentation front-end

Sound separation
augmentation front-end



- MixIT separation
- Data Augmentation (DA) blocks:
  - Temporal proximity sampling
  - SpecAugment
- Each DA block ⟶ different instance

Proxy Learning Tasks



Proposed Approach: Implementation Details

- **MixIT**: M=2 output channels (more practical than 4)
- One **encoder**: PANN's CNN14 (~75M)
  - Architecture with VGG style
  - Bottleneck layer to 128-d for representation *h*
- Two **heads**:
  - MLP & ReLU

Evaluation

- **Downstream Classification** with shallow model (mAP):
  - Using external AudioSet version
  - Training a shallow network on top of the learned representation
  - MLP w/ one layer + ReLU

- **Query by Example Retrieval** (mAP):
  - Subset of AudioSet with ~100 positive and negative examples
  - Compute cosine distance for all possible pairs (pos, pos) and (pos, neg)
  - Rank distances & compute AP

Evaluation

- **Downstream Classification** with shallow model (mAP):
  - Using external AudioSet version
  - Training a shallow network on top of the learned representation
  - MLP w/ one layer + ReLU
- **Query by Example Retrieval** (mAP):
  - Subset of AudioSet with ~100 positive and negative examples
  - Compute cosine distance for all possible pairs (pos, pos) and (pos, neg)
  - Rank distances & compute AP

Sound Separation for Contrastive Learning

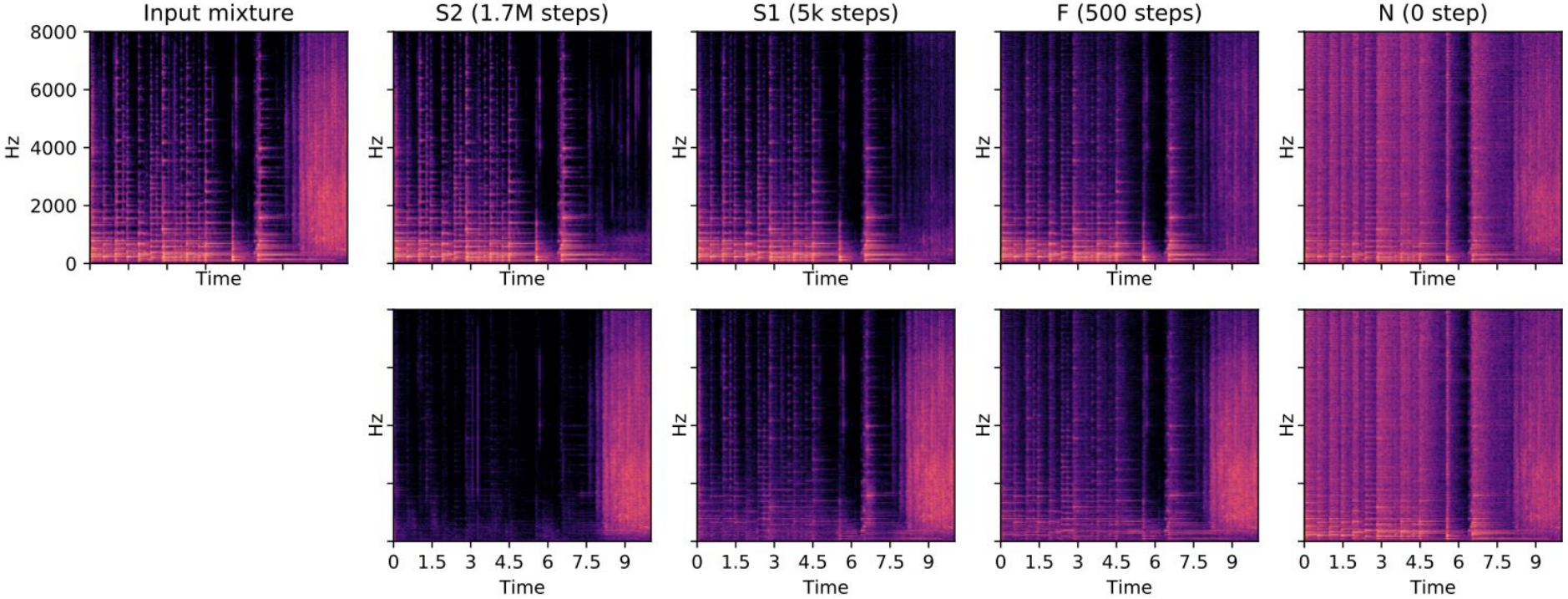
- SSep pre-processing outperforms SpecAugment
- Best: combining both
- Comparing input mixture w/ separated channels
  - better representations than using only the input mixture
- SSep can be successfully combined with other commonly-used augmentations

Table 2: Classification mAP using sound separation (SSep) in the front-end and the *SimCLR* back-end. TP is always applied; SpecAugment (SA) is applied as specified.

Comparison	SSep	SA	mAP
Mix vs mix (baseline)	-	-	0.248
Mix vs mix (baseline)	-	✓	0.265
Mix vs chan	✓	-	0.272
Mix vs chan	✓	✓	**0.282**

How About Using a Separation Model *Before* Convergence?

- Four audio processors: four training checkpoints of a single separation network



How About Using a Separation Model *Before* Convergence?

- All processors provide valid forms of augmentation
- Combining some of them using OR rule can be helpful

Table 4: Classification mAP using different checkpoints of the separation model as learning progresses (top), as well as some combinations (bottom). TP and SpecAugment are applied.

Models	SimCLR
S2 (1.7M)	0.282
S1 (5k)	0.283
F (500)	0.280
N (0)	**0.286**
S2 ∨ F	0.283
S2 ∨ N	**0.297**
S2 ∨ F ∨ N	0.285

Jointly Optimizing Both Proxy Tasks

- Similarity Maximization & Coincidence Prediction
- Small boosts across the board
- Key ingredient:
  - Combination of diverse processing by separation model as learning progresses

Table 4: Classification mAP using different checkpoints of the separation model as learning progresses (top), as well as some combinations (bottom). TP and SpecAugment are applied.

Models	SimCLR	SimCLR & CP
S2 (1.7M)	0.282	0.289
S1 (5k)	0.283	0.293
F (500)	0.280	0.297
N (0)	**0.286**	**0.301**
S2 ∨ F	0.283	0.300
S2 ∨ N	**0.297**	0.306
S2 ∨ F ∨ N	0.285	**0.310**

Comparison with Previous Work

- Proposed framework
  - Outperforms some past and multimodal approaches
  - Competitive with SOTA

Table 5: Comparison with previous work using shallow model classification. MM = Multimodal approach.

Method	d	MM	mAP
Unsupervised triplet [14]	128	-	0.244
C^3 [15]	128	✓	0.285
Separation-based framework (ours)	128	-	0.310
Separation-based framework (ours)	1024	-	0.326
MMV [40]	2048	✓	0.309
Multi-format [19]	2048	-	0.329

[14] Jansen et al., **Unsupervised learning of semantic audio representations**. ICASSP 2018
[15] Jansen et al., **Coincidence, Categorization, and Consolidation: Learning to Recognize Sounds with Minimal Supervision**. ICASSP 2020
[40] Alayrac et al., **Self-supervised multimodal versatile networks.** 2020
[19] Wang & van der Oord. **Multi-Format Contrastive Learning of Audio Representations**. SAS Workshop NeurIPS 2020

Takeaways

- Sound separation → **valid augmentation** to generate views for contrastive learning
- Learning to **associate sound mixtures w/ separated channels** elicits semantic structure in learned representation
- Sound separation allows **combination w/ commonly-used augmentations**
- Transformations by **different checkpoints of the same separation model**
  - Valid augmentations for generating positives
- Benefit in **jointly training similarity maximization and coincidence prediction**

Self-Supervised Learning from Automatically Separated Sound Scenes

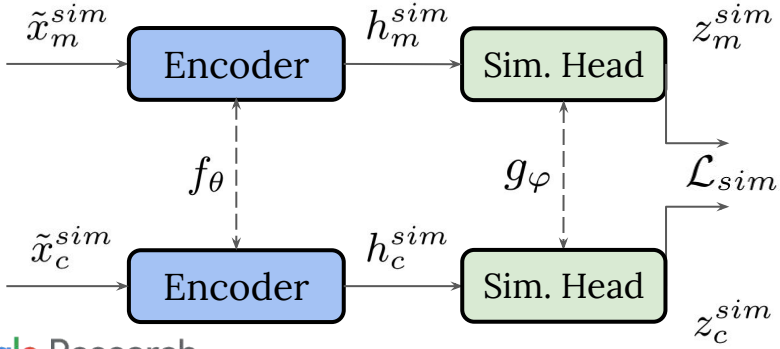
Eduardo Fonseca[12], Aren Jansen[1], Daniel P. W. Ellis[1], Scott Wisdom[1], Marco Tagliasacchi[1], John R. Hershey[1], Manoj Plakal[1], Shawn Hershey[1], R. Channing Moore[1], Xavier Serra[2]

[1] Google Research [2] upf. Universitat Pompeu Fabra Barcelona MTG Music Technology Group

WASPAA 2021

Similarity Maximization (SM)

- **Goal**: maximize agreement between differently-augmented views
- **Encoder**: Extract low-dimensional embeddings h
  - Once training is over, *h* is used for downstream tasks
- **Similarity Head**: Map *h* to metric embedding *z*, where loss is applied (tends to work best)
- Normalized temperature-scaled cross-entropy (**NT-Xent**) **loss**
  - Softmax structure
  - Scoring function: cosine similarity with temperature scaling τ
  - Maximize similarity between views



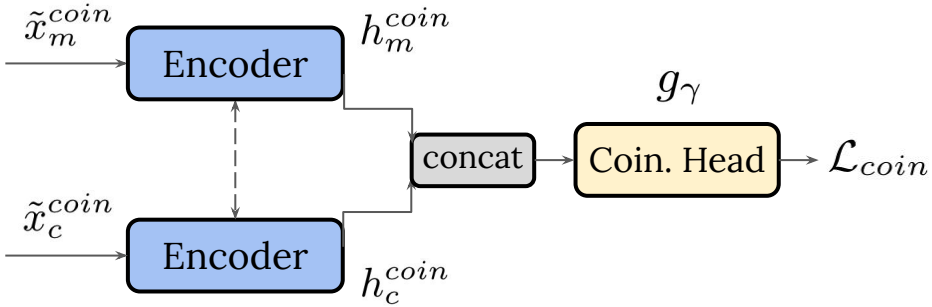
$$\mathcal{L}_{sim_{i,j}} = -\log \frac{\exp\left(\text{sim}(z_i, z_j)/\tau\right)}{\sum_{v=1}^{2N} \mathbb{1}_{v \neq i} \exp\left(\text{sim}(z_i, z_v)/\tau\right)}$$

Google Research

Chen et al., **A Simple Framework for Contrastive Learning of Visual Representations.** ICML 2020

Coincidence Prediction (CP)

- **Based on *slowness prior***: waveforms vary quickly ↔ semantics change slowly
  - Stable representation to explain semantics
  - Representation that would support prediction of coincidence in temporal proximity
- **Encoder**: Extract low-dimensional embeddings *h* & concatenate pairs of embeddings
- **Coincidence Head**: Map $[h_m, h_c]$ to probability that pair is coinciding
  - binary classification task: predict (non)-coincidence
- **Binary cross entropy loss (BCE)**



$$\mathcal{L}_{coin}(X) = -\frac{1}{N}\sum_{i=1}^{N} \log g_\gamma([h_m^{coin,i}, h_c^{coin,i}]) - \frac{1}{N(N-1)} \sum_{\substack{1 \le i,j \le N \\ j \neq i}} \log \left[1 - g_\gamma([h_m^{coin,i}, h_c^{coin,j}])\right]$$

Jansen et al., **Coincidence, Categorization, and Consolidation: Learning to Recognize Sounds with Minimal Supervision**. ICASSP 2020