Embeddings in Music and Audio

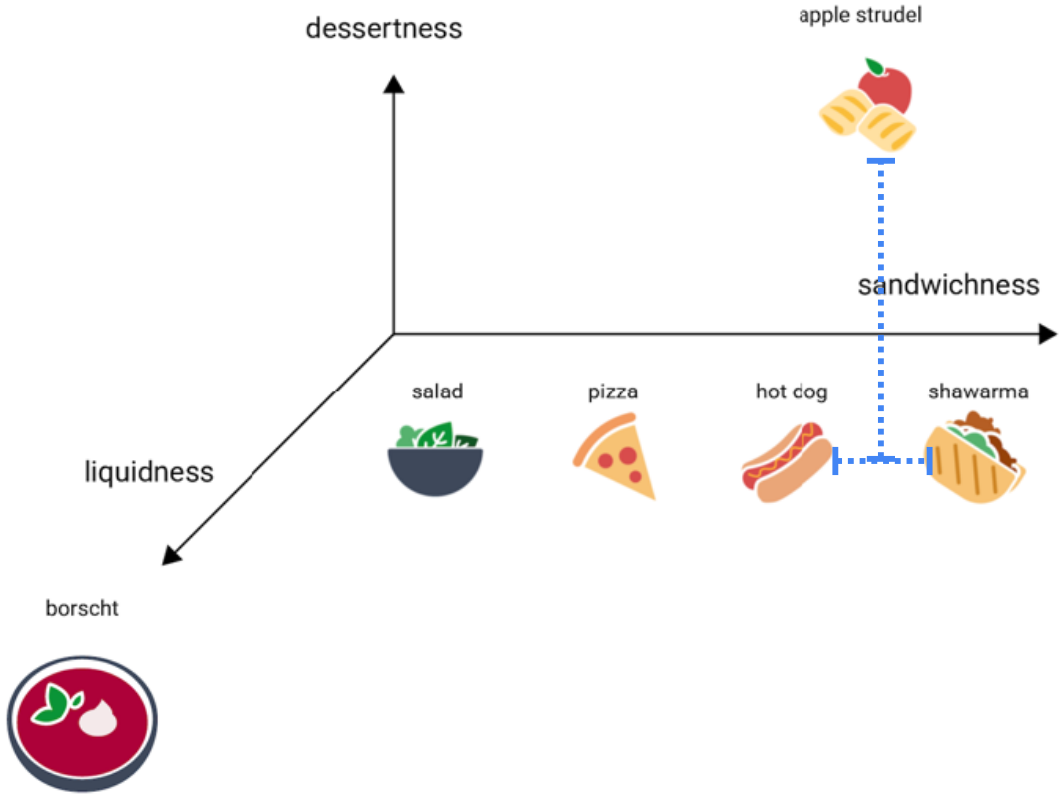
CS352 Winter 2026
Bryan Pardo & Annie Chu

What's an "embedding"?

- A neural network embodies a function $f: X \rightarrow X'$
- X is the input to the net and X' is the set of activations of some layer of the net.
- Colloquially, we refer to X' as an "embedding" of the input to the net.
- Ideally, want an embeddings to have meaningful "groupings"

The point of embedding spaces

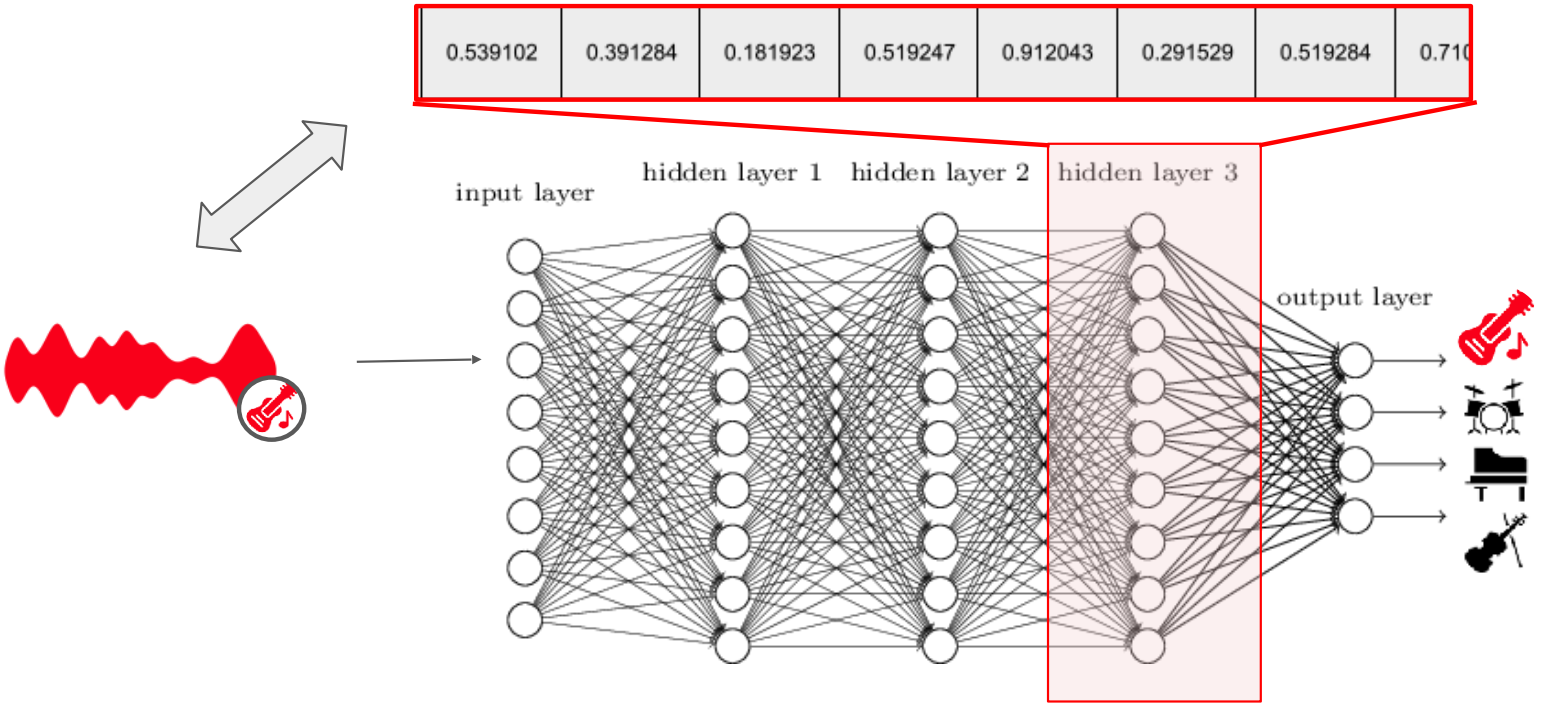
- Low(ish)-dimensional representations data that captures **important relationships between items**
- **S**imilar items should be close together and dissimilar items should be far apart



source: google

Any trained network can be used to make embeddings

Embeddings extracted as general-purpose feature vector



In this example, we'll take embeddings from its penultimate layer

Training an Audio Embedding Network

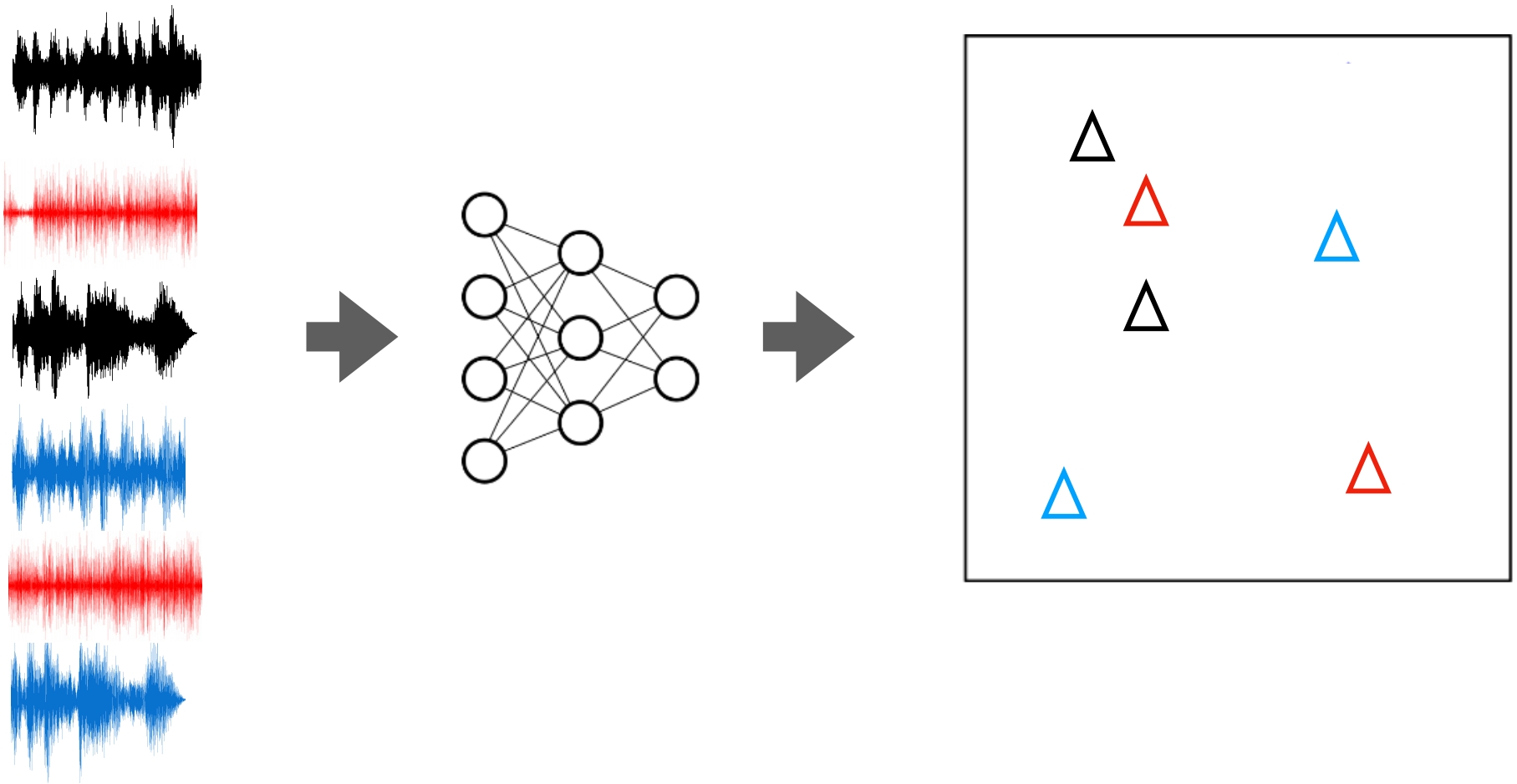
Example: VoiceID

Contrastive Loss

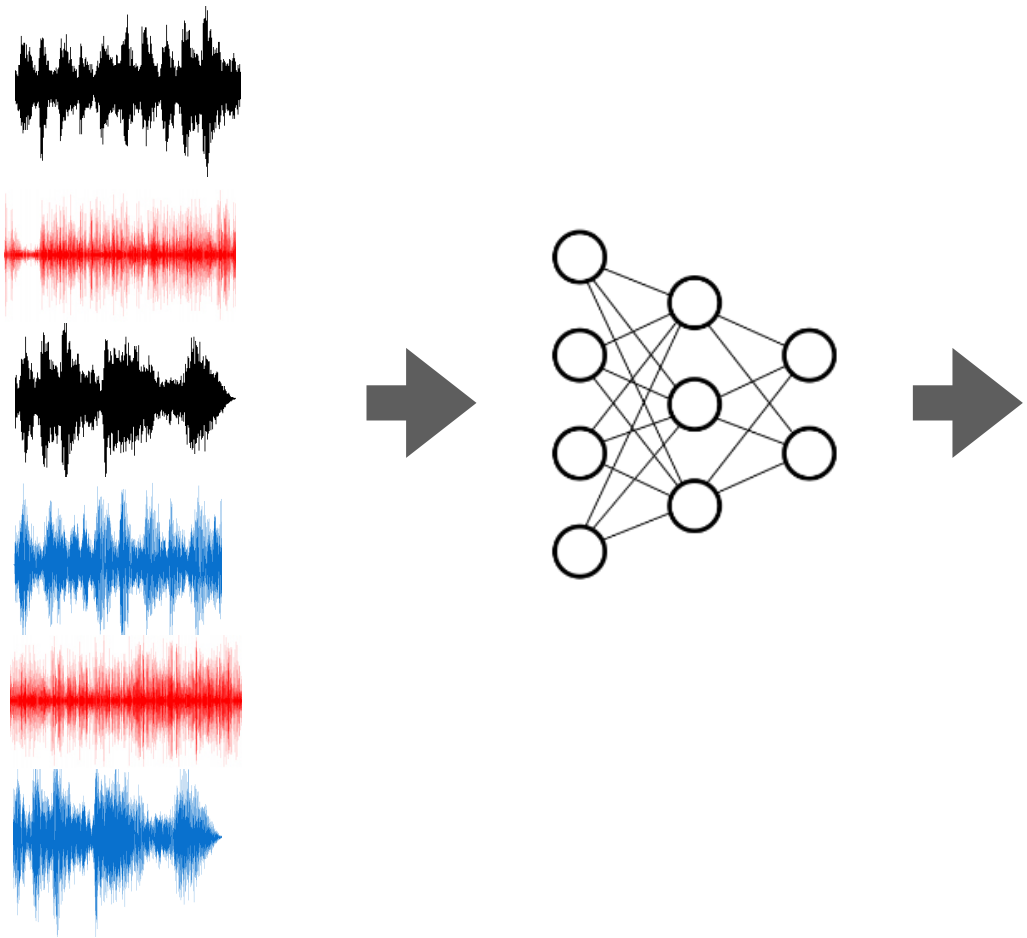
Contrastive Loss: compare 2 (or more) pairs

- I have 3 things: A, B and C.
- I say that A and B should go together, while A and C should not.
- I get the embeddings for A, B, C
- I measure the distances
- We want A & B to be close and A & C to be far.
- I want to increase the CONTRAST in the distances between these two pairs.

Train an embedding net

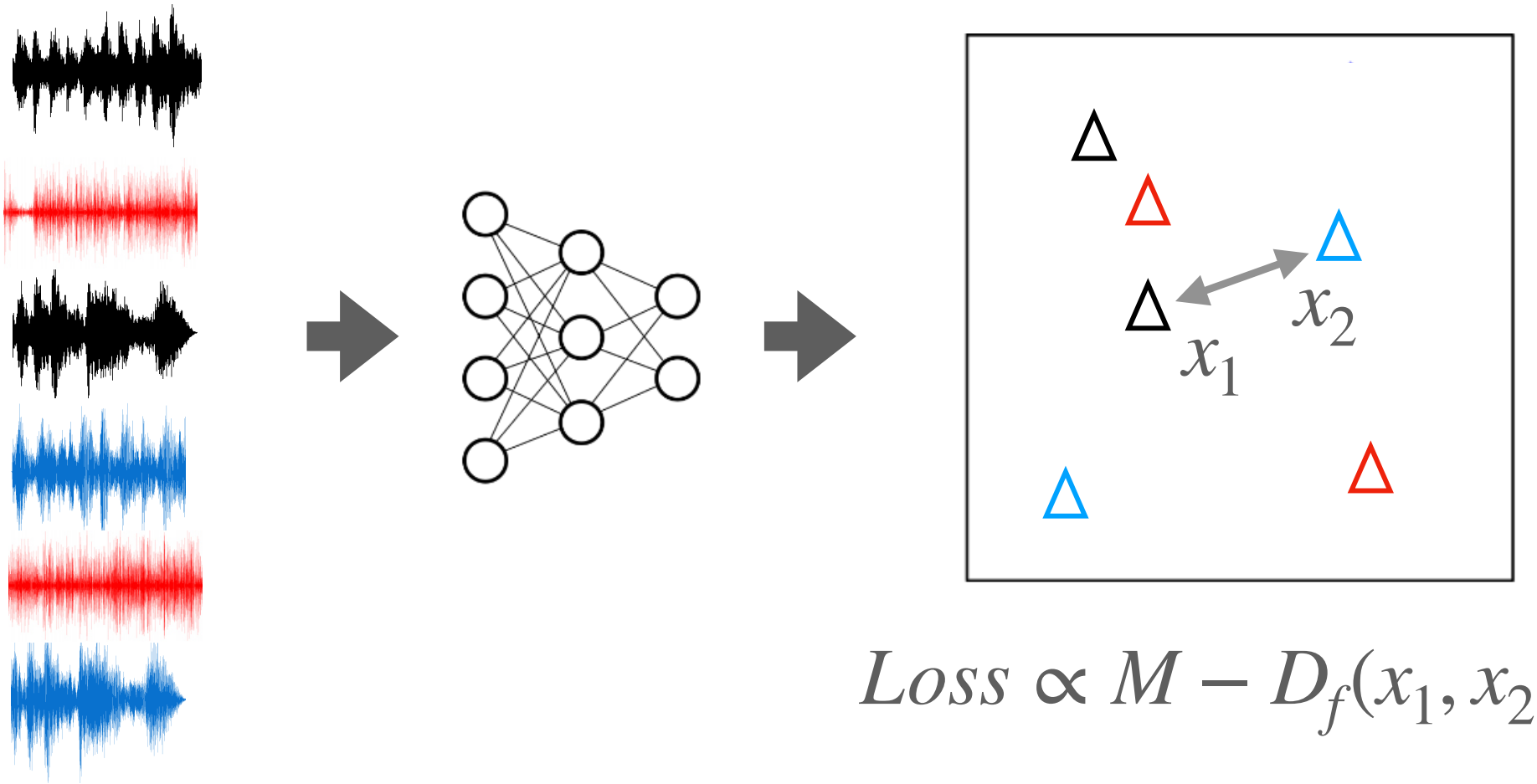


Move things from the same group closer



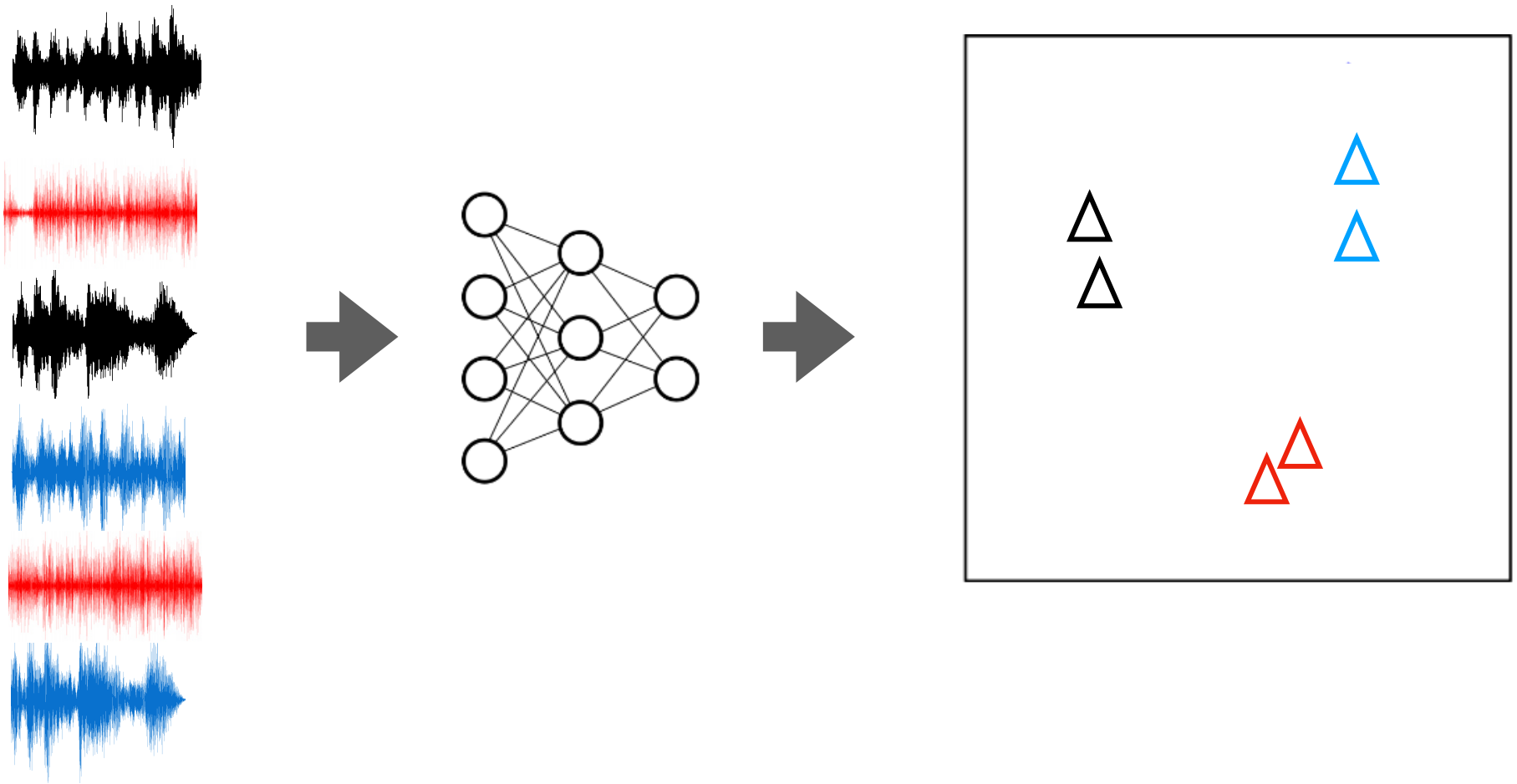
$$Loss \propto D_f(x_1, x_2)$$

Push things from different groups apart



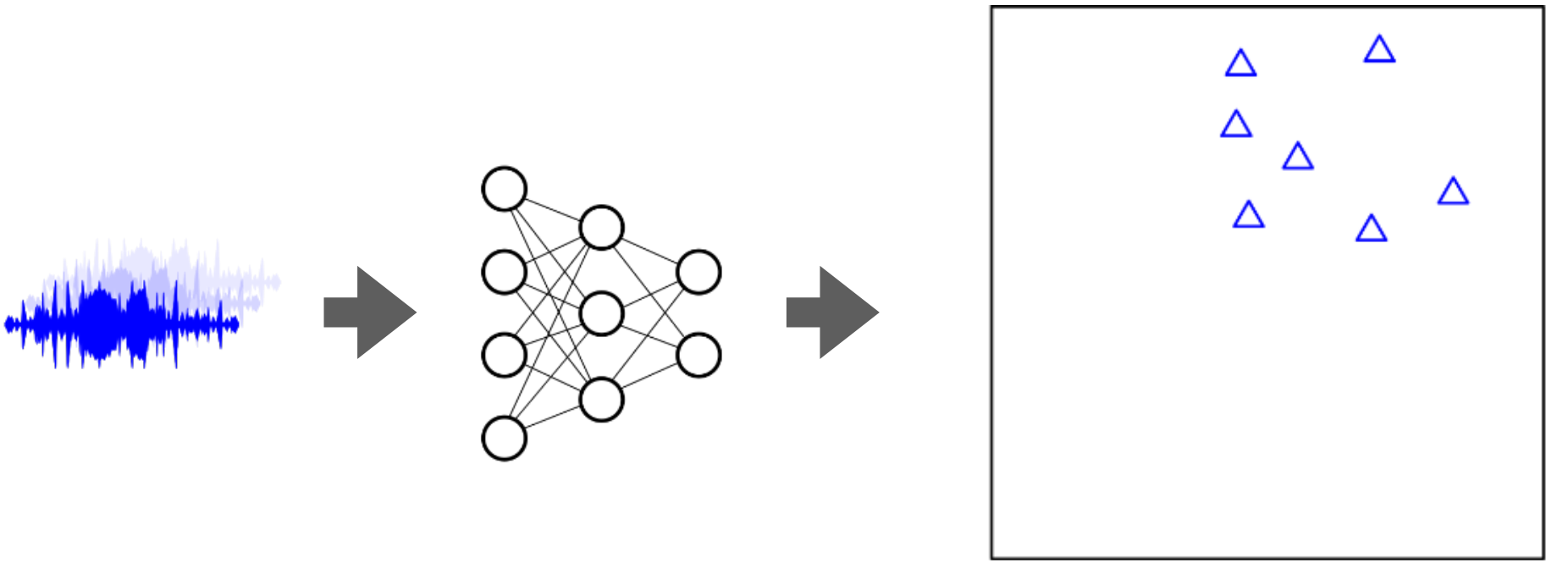
$$Loss \propto M - D_f(x_1, x_2)$$

Train an embedding net



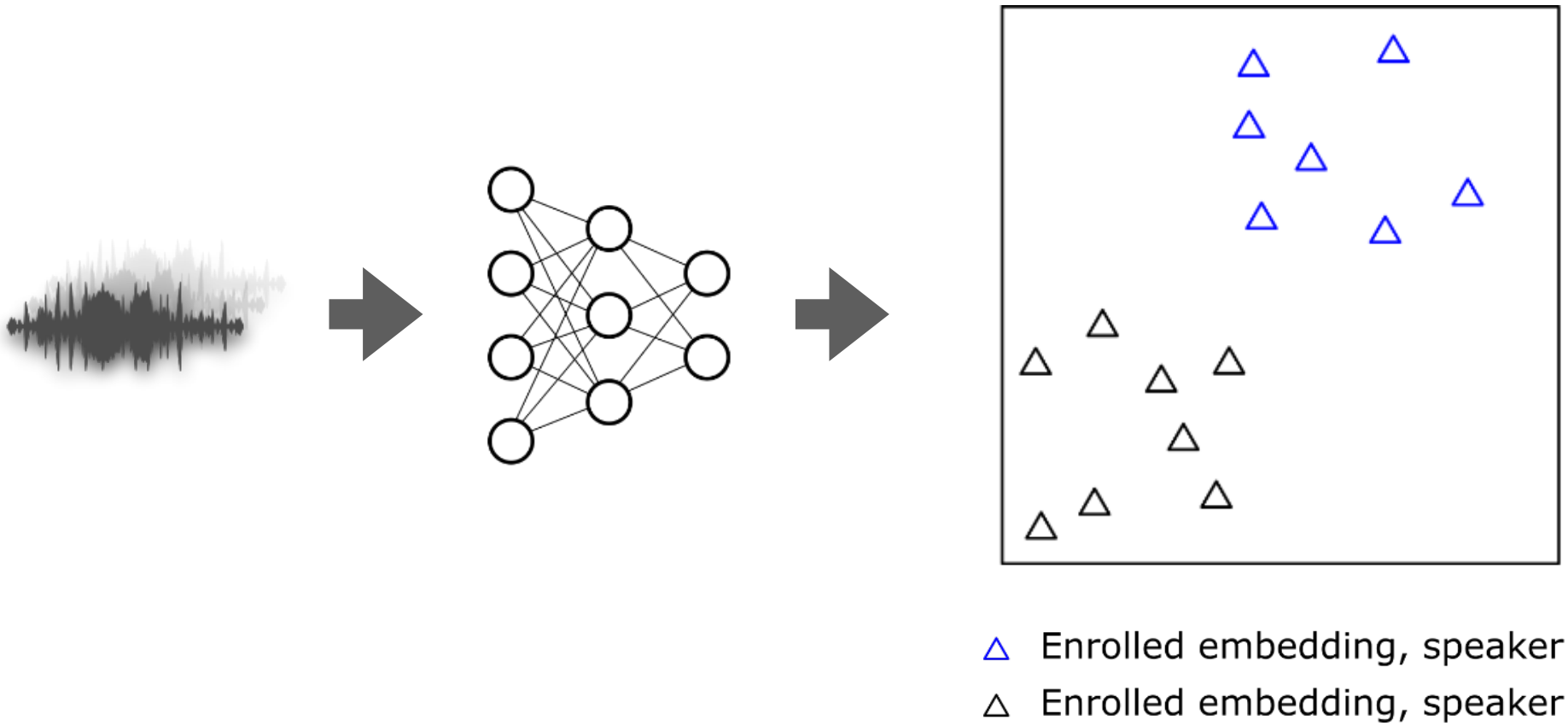
Using the trained embedding

Enroll a voice



△ Enrolled embedding, speaker A

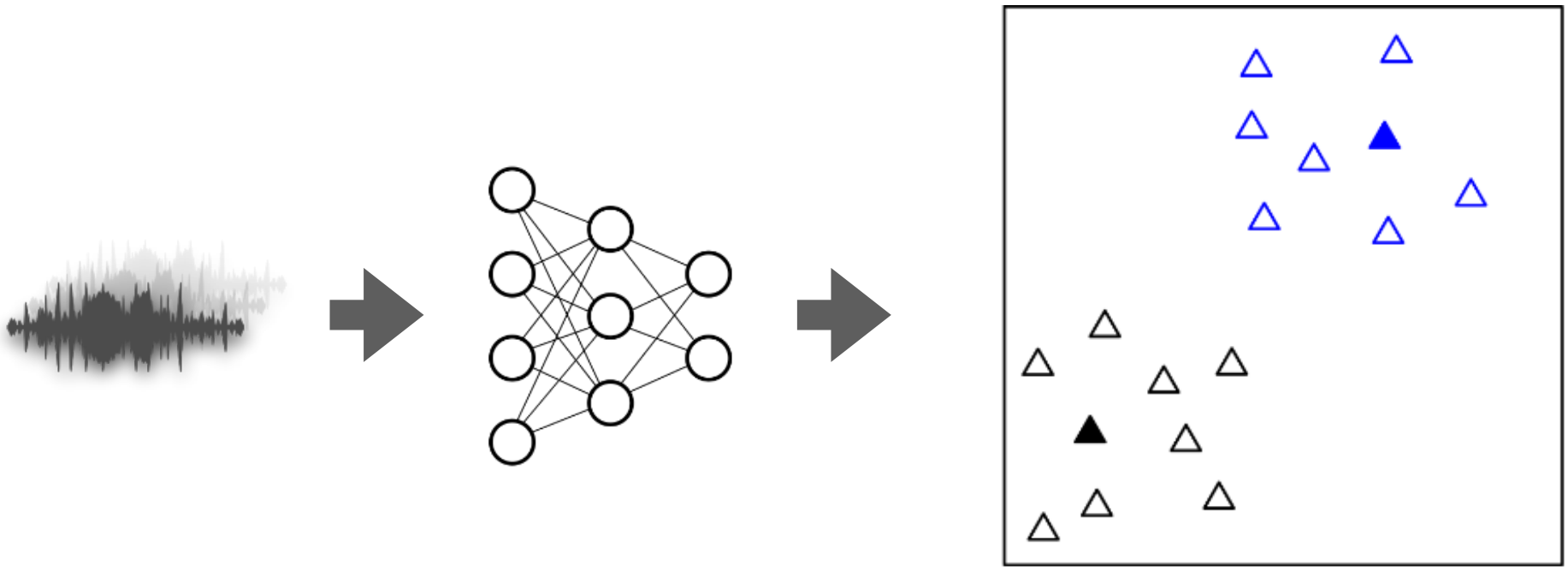
Enroll more voices



△ Enrolled embedding, speaker A

△ Enrolled embedding, speaker B

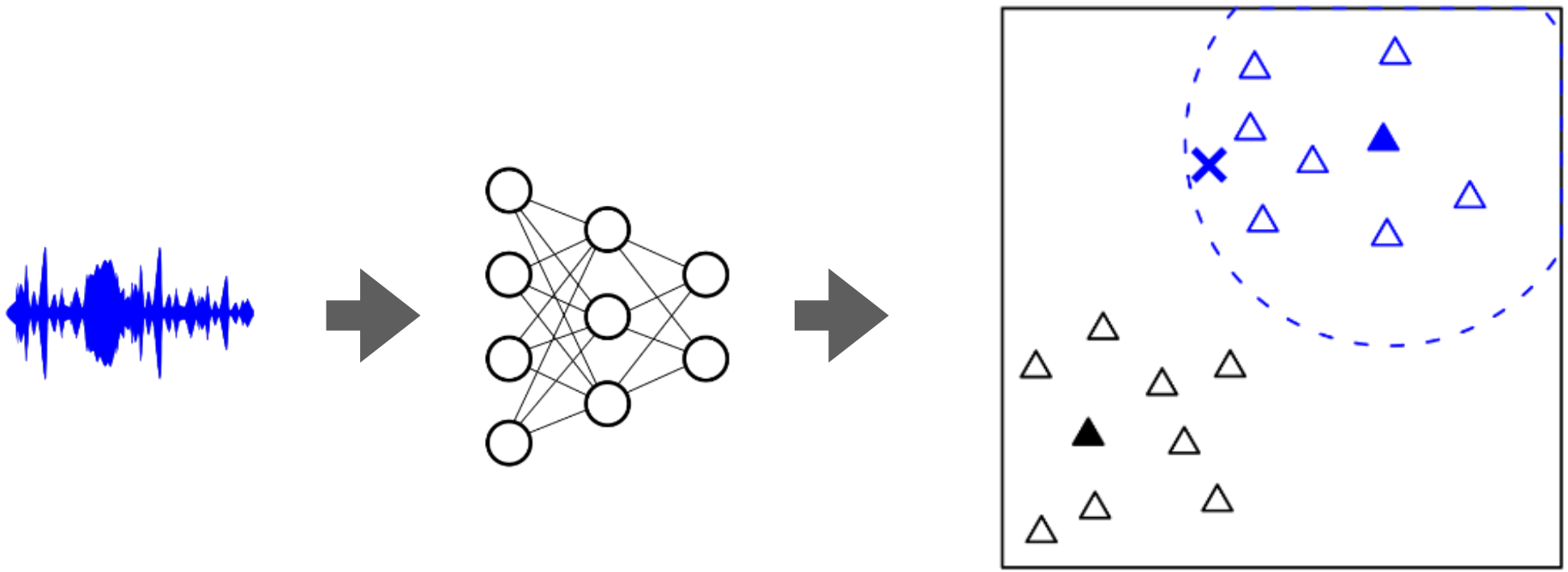
Find centroids



▲ Enrolled embedding centroid, speaker A

▲ Enrolled embedding centroid, speaker B

Use nearest-neighbor search to pick a group



× Query embedding, speaker A

Finding neighbors: Cosine similarity

- Often used with embeddings
- A kind of normalized dot product
- Higher values = more similar

$$S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

What if we remove the normalization?

- Often used with embeddings
- A kind of ~~normalized~~ dot product
- Higher values = more similar
- How would scale affect things?

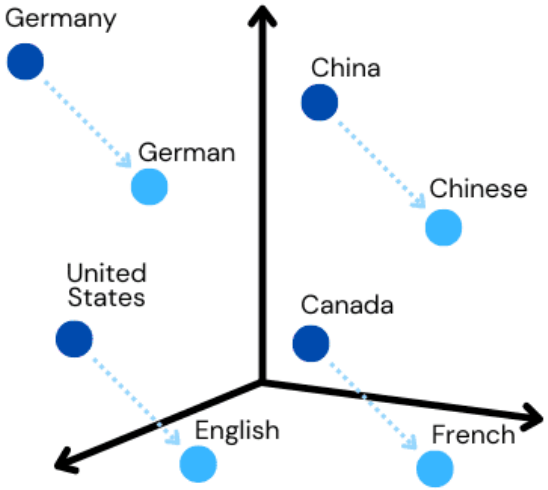
$$S(A, B) = \sum_i A_i B_i$$

Making a text/audio embedding network

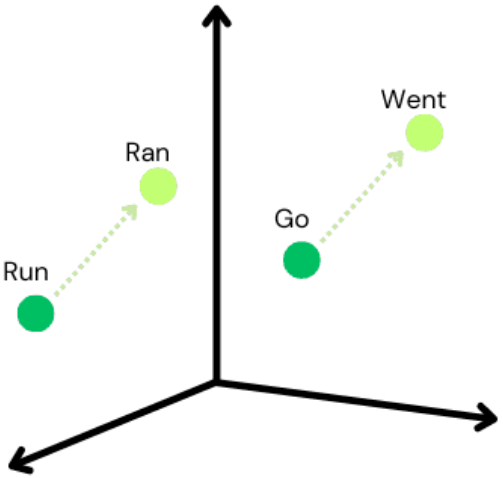
Contrastive Learning Audio Pretraining

B. Elizalde, S. Deshmukh, M. Al Ismail, and H. Wang, "Clap: Learning audio concepts from natural language supervision," in ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing

There are also text embedding spaces



Country - Language



Verb Tense

vector offsets = consistent vector transformations reusable across space

("embedding math")

vectors capture semantic relationships

image source

How can we combine text & audio embeddings?

Here's how we do it

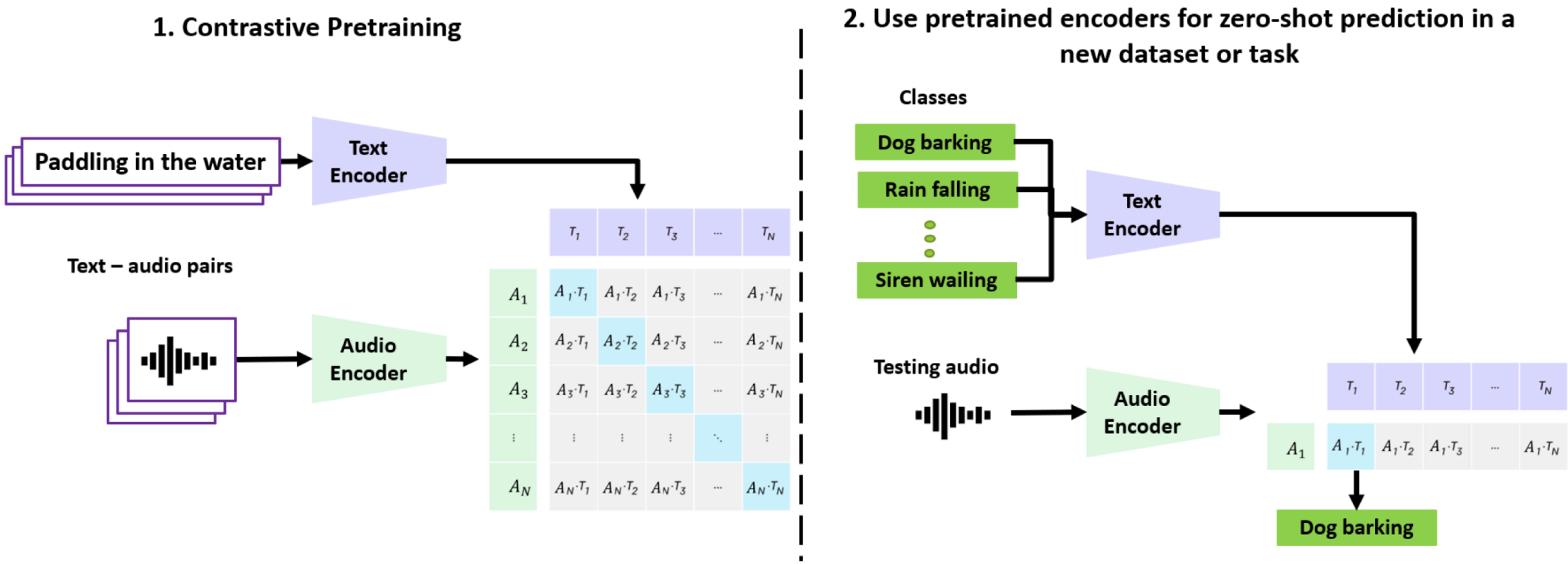
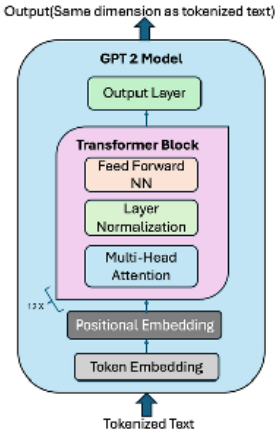


Fig. 1. CLAP 👏jointly trains an audio and a text encoder to learn the (dis)similarity of audio and text pairs in a batch using contrastive learning. At testing time, the pretrained encoders are used to extract audio embeddings from the testing audio and text embeddings from the class labels. Zero-Shot linear classification is achieved by computing cosine similarity between the embeddings.

CLAP's Encoders

GPT2
(transformer)



fine-tuned only

Text – audio pairs

HTSAT
(transformer)

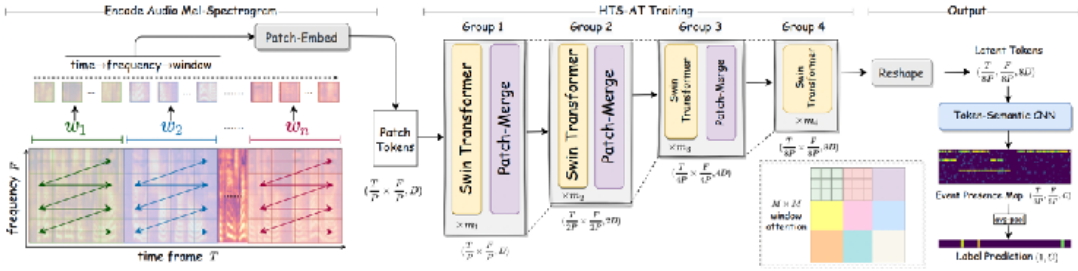


Fig. 1: The model architecture of HTS-AT.

https://arxiv.org/pdf/2202.00874

trained, then fine-tuned on contrastive learning

trained on 22 audio tasks (e.g, classification, retrieval, captioning)

Image from Elizalde, et al. arXiv:2206.04769v1

CLAP Dataset

number of pairs was 119k instead of 128k. The training datasets for the 4.6M collection are: WavCaps [6], AudioSet [2], FSD50K [12], Clotho [13], AudioCaps [14], MACS [15], WavText5k [5], SoundDesc [16], NSynth [17], FMA [18], Mosi [19], Meld [20], Iemocap [21], Mosei [22], MSP-Podcast [23], CochlScene [24], LJspeech [25], EpicKitchen [26], Kinectics700 [27], findsounds.com. Details on GitHub.

4.6M audio-text pairs
mostly general sound, some speech, some music

human-annotated

`Wind and a race car make noise, with a man speaking and the sound of accelerating and tire squealing.`

WavCaps example
(pulled from AudioSet)

`A woman is rapping while a medium engine runs and a cat meows.`

AudioSet Example

inferred captions
(likely from metadata)

genre:
electronic

`This is an [electronic] song`

FMA

filename:
bass_synthetic
_033-047-050

`This is the sound of [synthetic bass]`

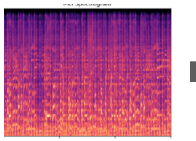
NSynth

Let's take a look at one audio-text pair independently

1. encode paired data in their respective modalities

2) Projection Layer
→ Shared Embedding Space

3) Alignment Objective (e.g, contrastive loss)



Mel-frequency Spectrogram

audio encoder

0.3
0.7
...
0.7

(B, D_audio)

projector
(audio->shared)

0.6
0.5
...
0.9

A_emb_1
(B, **D_shared**)

"paddling in the water"

text encoder

0.2
0.4
...
0.7

(B, D_text)

projector
(text->shared)

0.6
0.5
...
0.9

T_emb_1
(B, **D_shared**)

how similar is the audio embedding to the text embedding? ideally they have the same embedding aka similarity score of 1

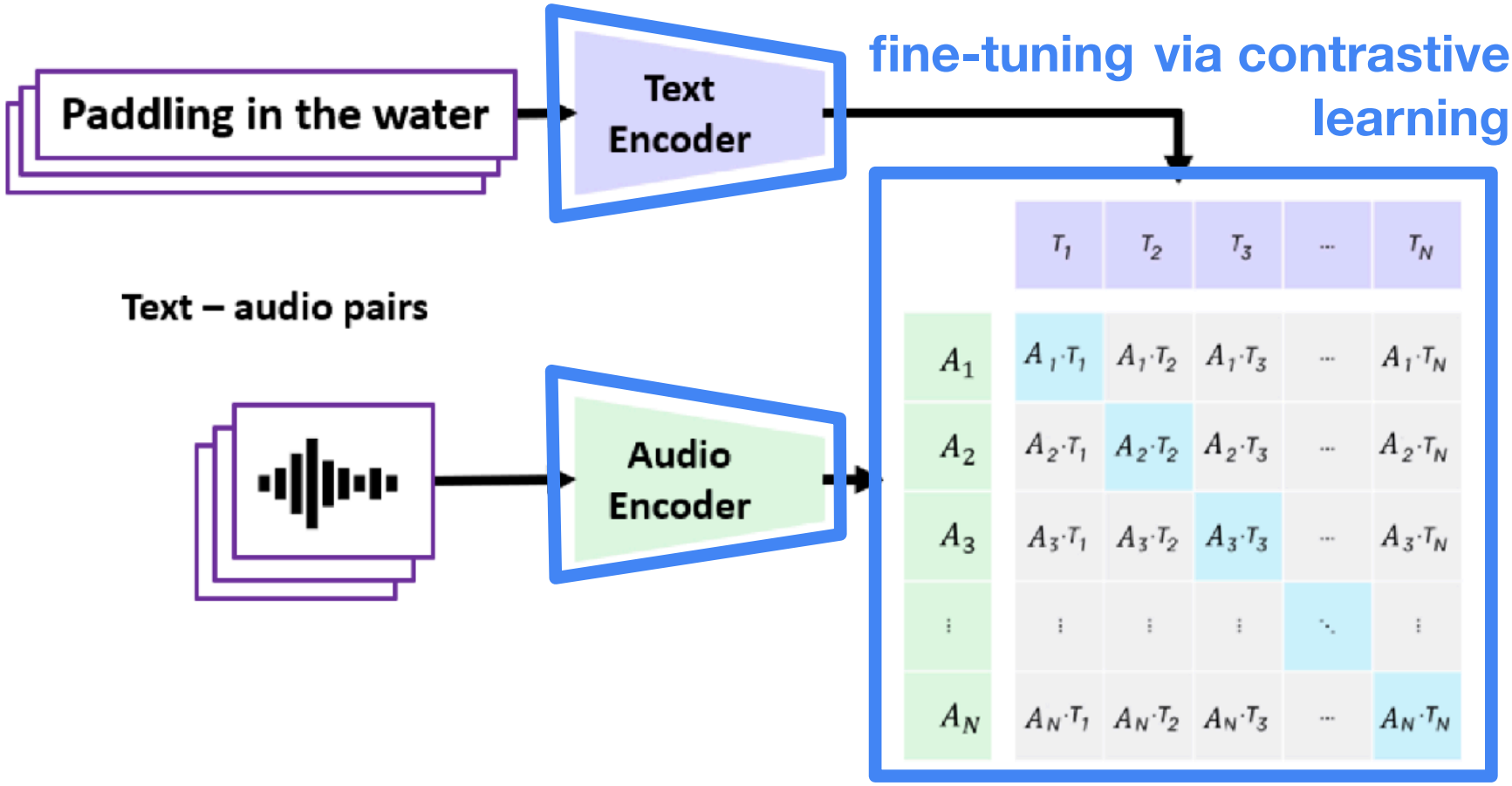
3a. calculate pairwise distance sim

$$sim = cos\_sim(audio\_emb, text\_emb)$$

3b. calculate contrastive loss

$$BCE(sim, 1)$$

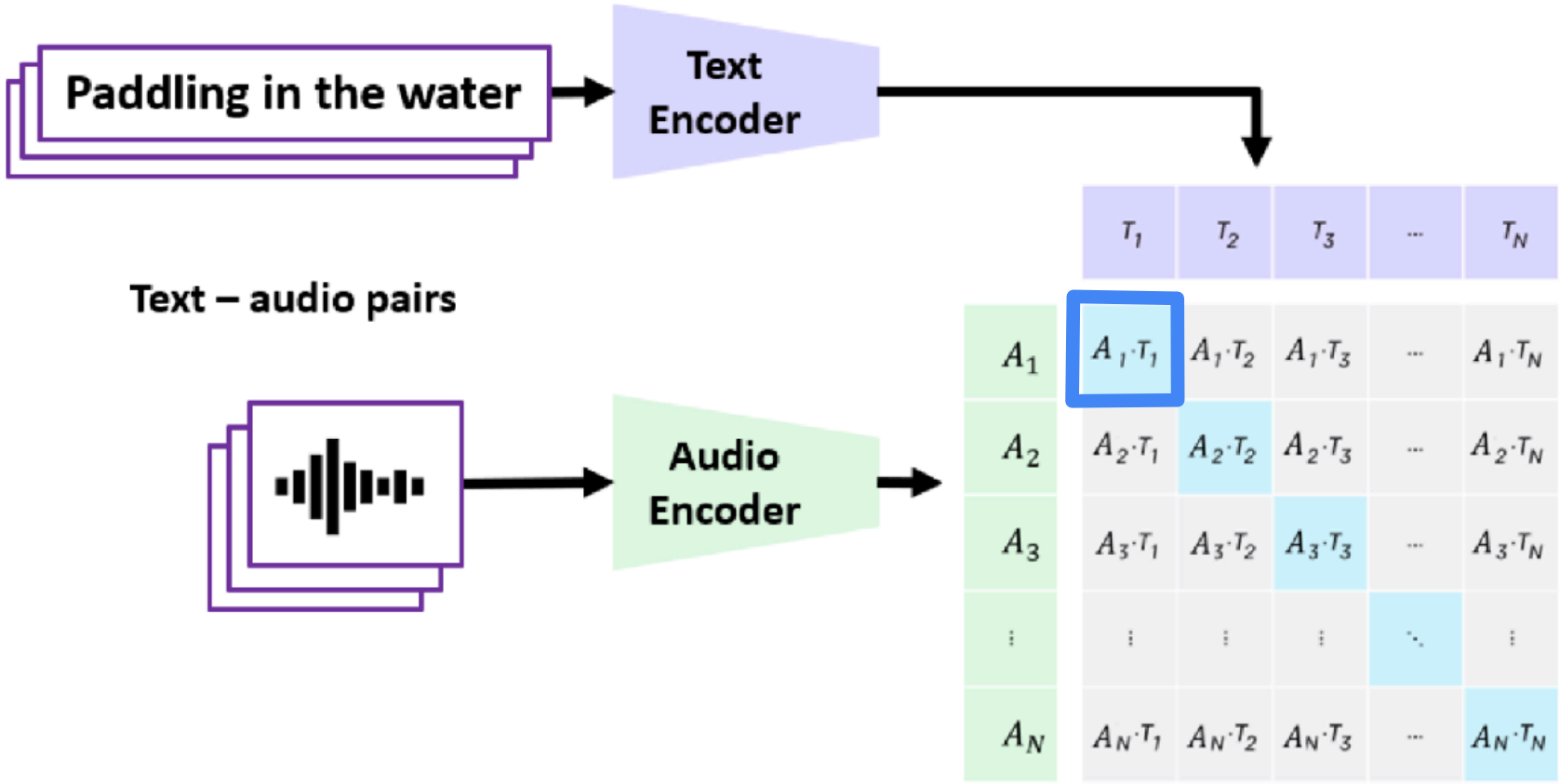
CLAP training—> fine-tuning encoders together



Audio is FreeSound 16770__dobroide__20060307paddling.wav

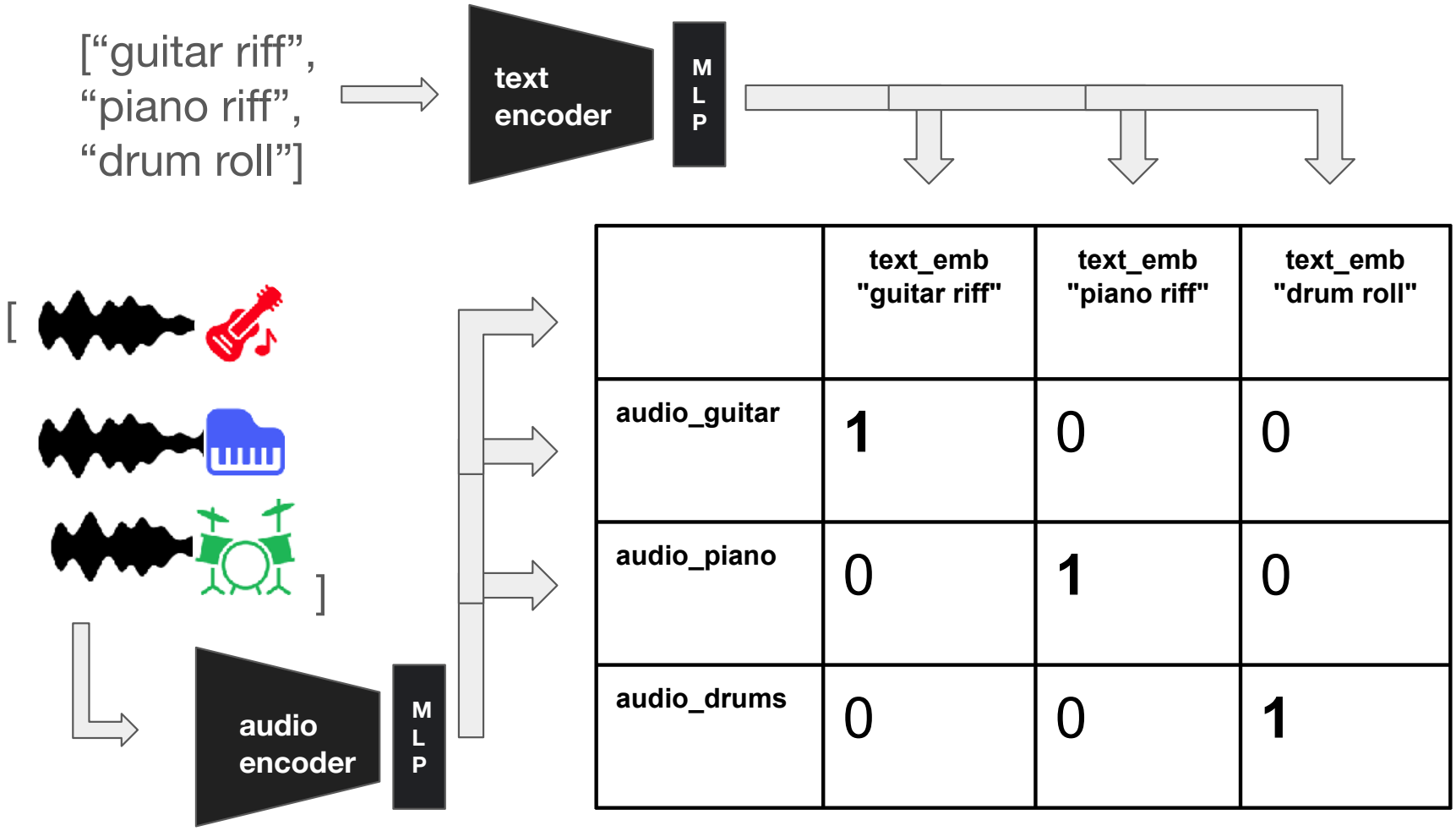
Image from Elizalde, et al. arXiv:2206.04769v1

Comparing one audio/text pair (A1, T1)



What if we do this across a batch of paired examples?

TARGET

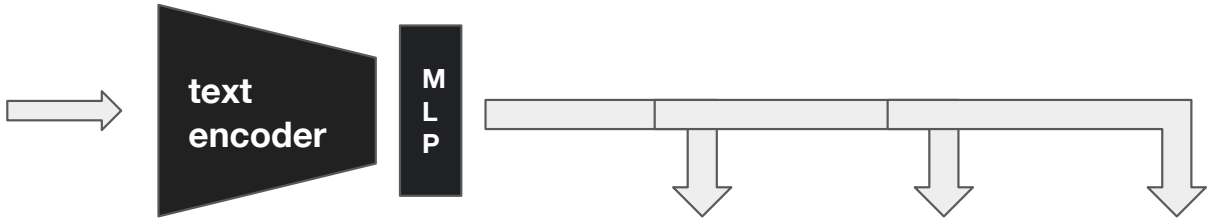


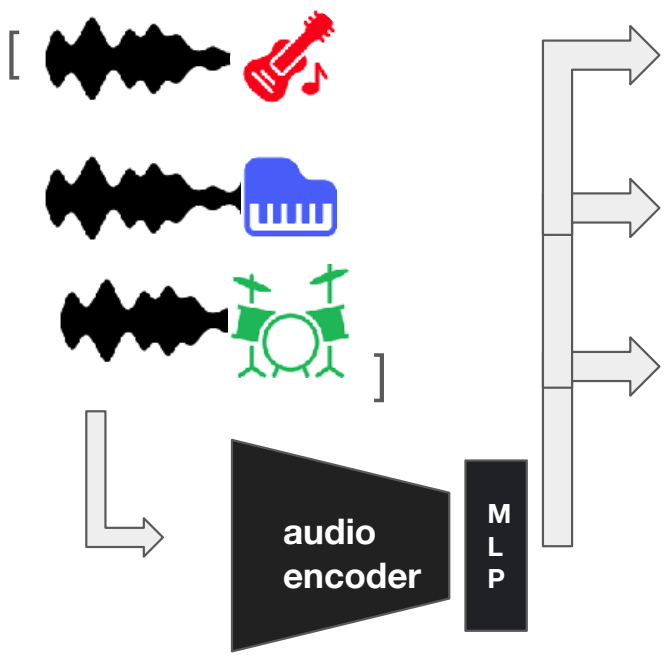
	text_emb "guitar riff"	text_emb "piano riff"	text_emb "drum roll"
audio_guitar	**1**	0	0
audio_piano	0	**1**	0
audio_drums	0	0	**1**

TARGET

Let's just take one row

["guitar riff",
"piano riff",
"drum roll"]



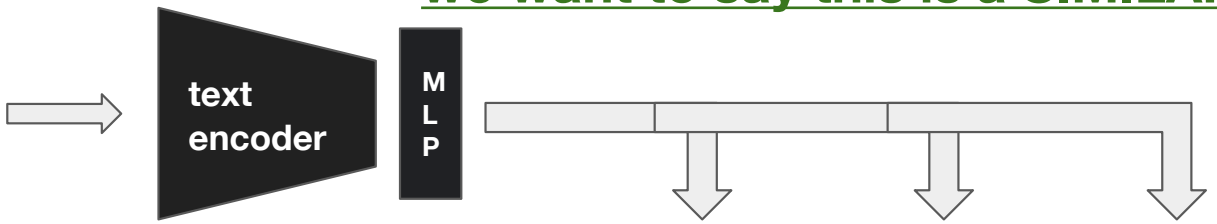


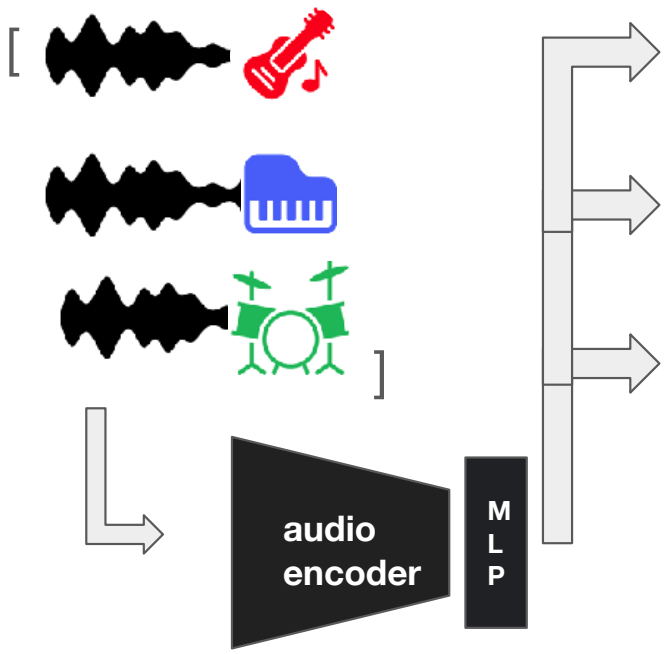
	text_emb "guitar riff"	text_emb "piano riff"	text_emb "drum roll"
audio_guitar	1	0	0
audio_piano	0	1	0
audio_drums	0	0	1

TARGET

we want to say this is a SIMILAR pair

["guitar riff",
"piano riff",
"drum roll"]

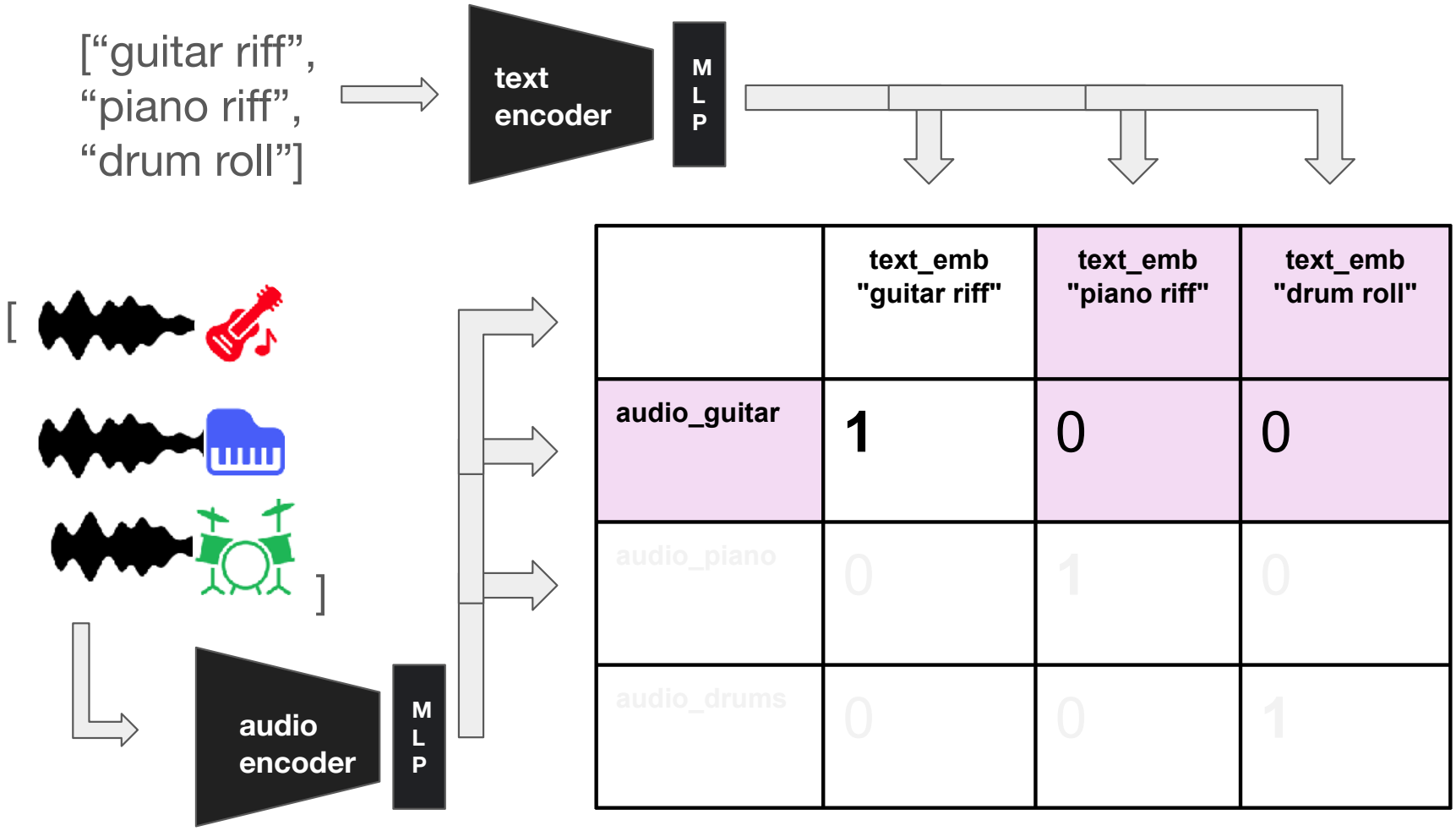




	text_emb "guitar riff"	text_emb "piano riff"	text_emb "drum roll"
audio_guitar	1	0	0
audio_piano	0	1	0
audio_drums	0	0	1

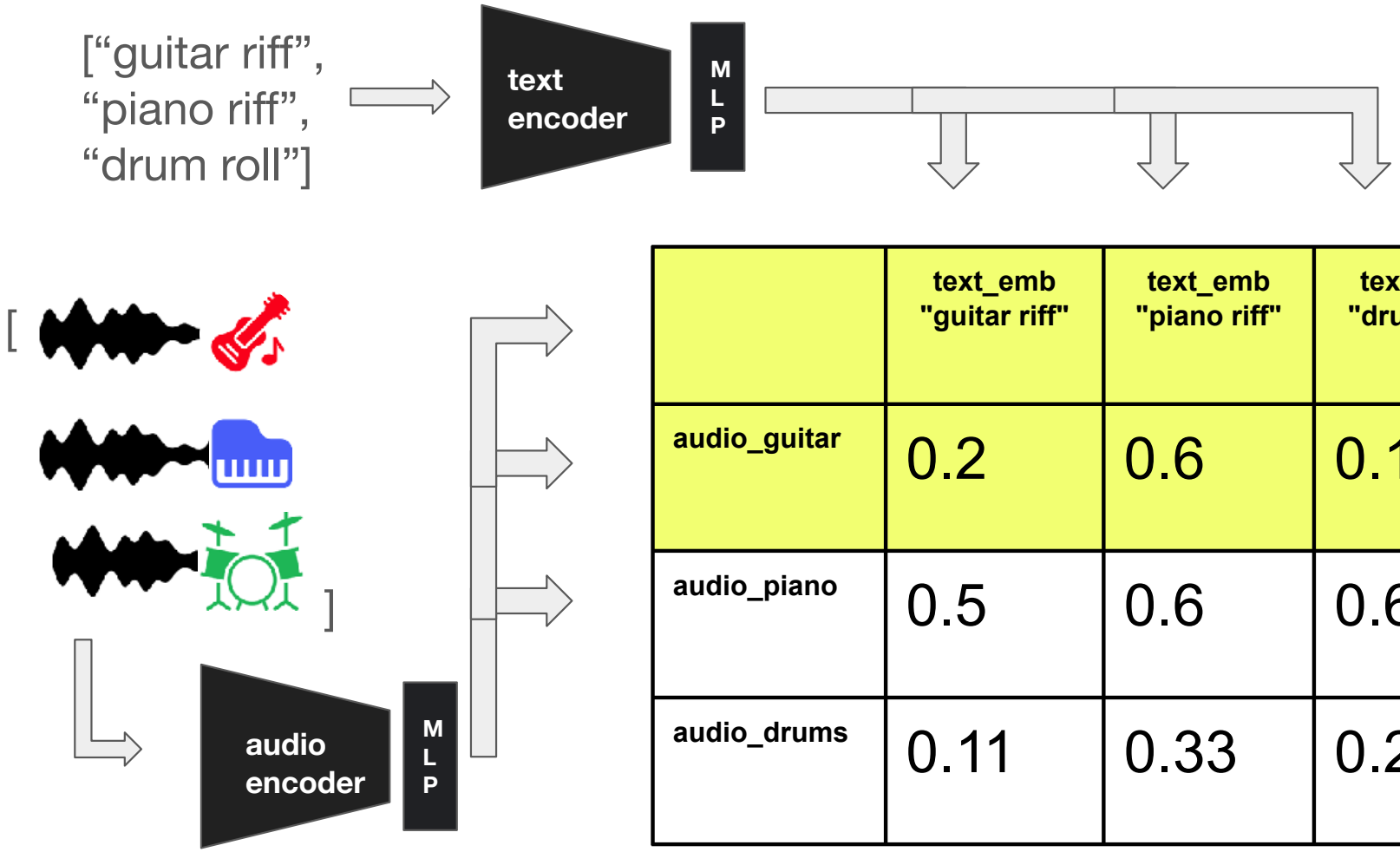
TARGET

we can also say these are DISSIMILAR pairs



	text_emb "guitar riff"	text_emb "piano riff"	text_emb "drum roll"
audio_guitar	1	0	0
audio_piano	0	1	0
audio_drums	0	0	1

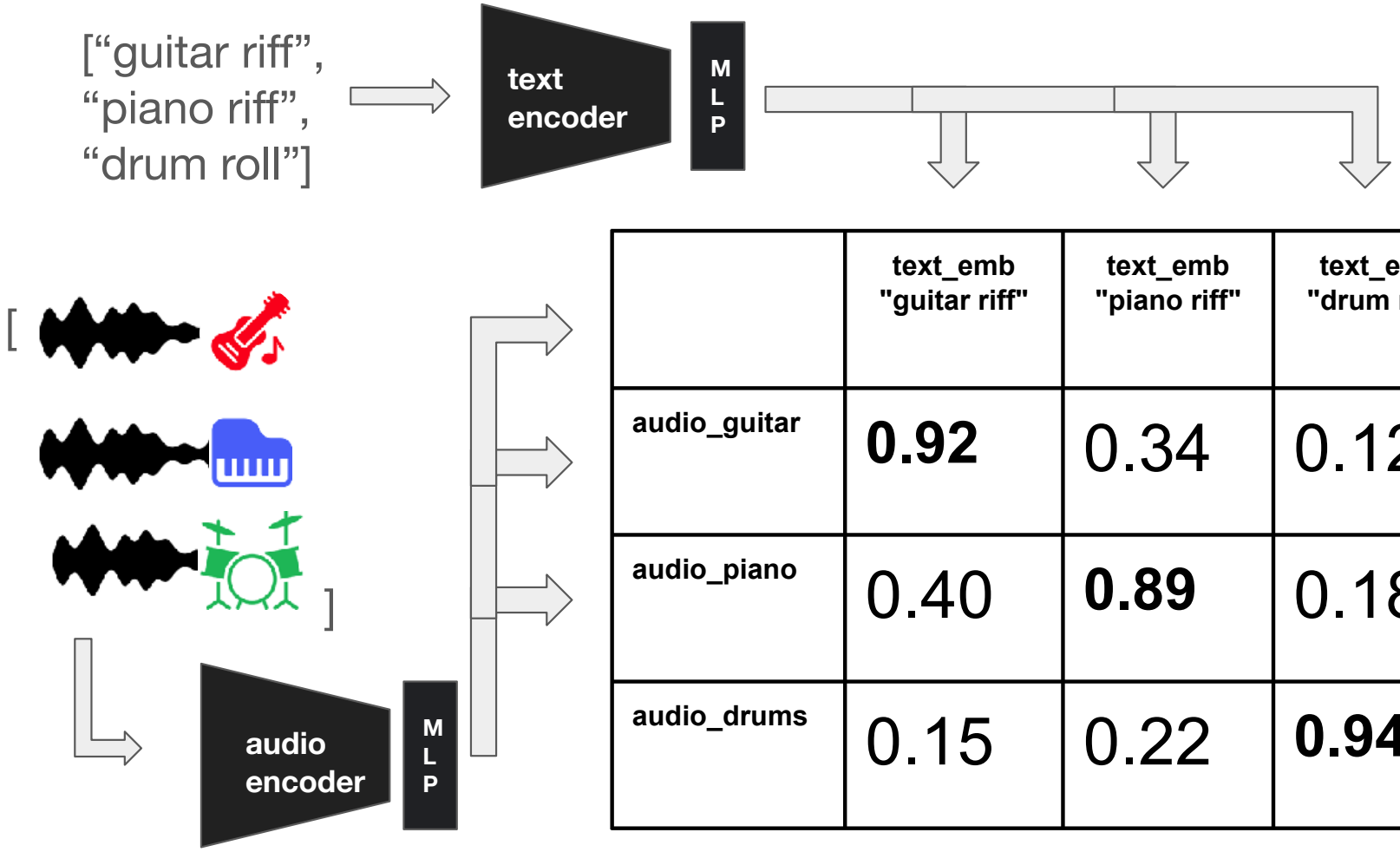
Let's calculate the InfoNCE loss at step 0 for this row (audio_guitar —> all_texts)



	text_emb "guitar riff"	text_emb "piano riff"	text_emb "drum roll"
audio_guitar	0.2	0.6	0.1
audio_piano	0.5	0.6	0.6
audio_drums	0.11	0.33	0.22

After many epochs of training….

AFTER SOME TRAINING (w InfoNCE loss)



	text_emb "guitar riff"	text_emb "piano riff"	text_emb "drum roll"
audio_guitar	**0.92**	0.34	0.12
audio_piano	0.40	**0.89**	0.18
audio_drums	0.15	0.22	**0.94**

Now we have something like this

Content

dog barking

cat meowing

guitar riff

whisper

muffled rumble

coin dropping

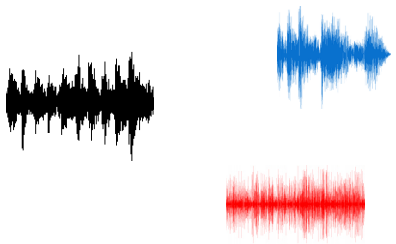
car rumbling

funky

distorted clarinet

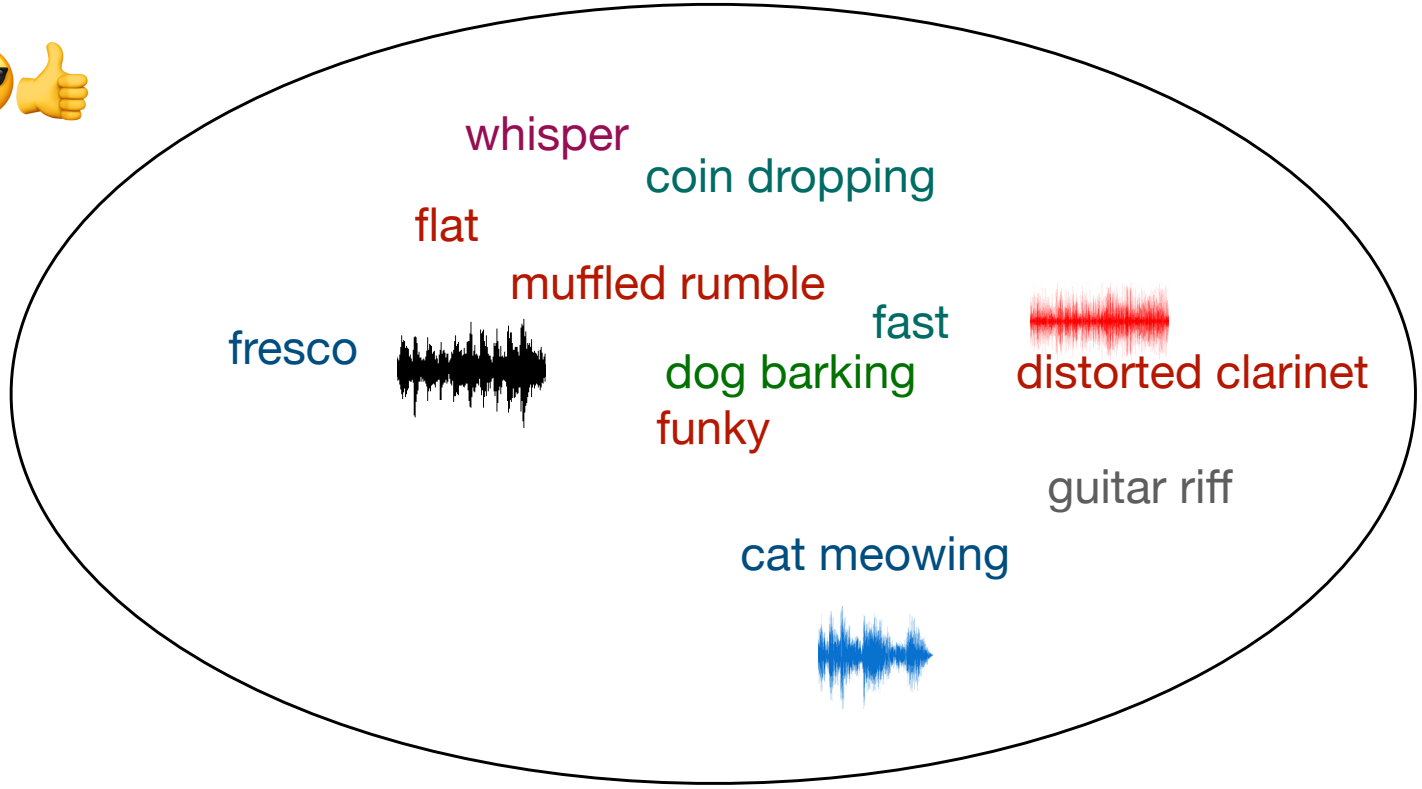
fresco

Sounds



Shared Text-Audio embedding space

nice 😎👍



✅ a joint embedding space aligning audio concepts with corresponding text ✅

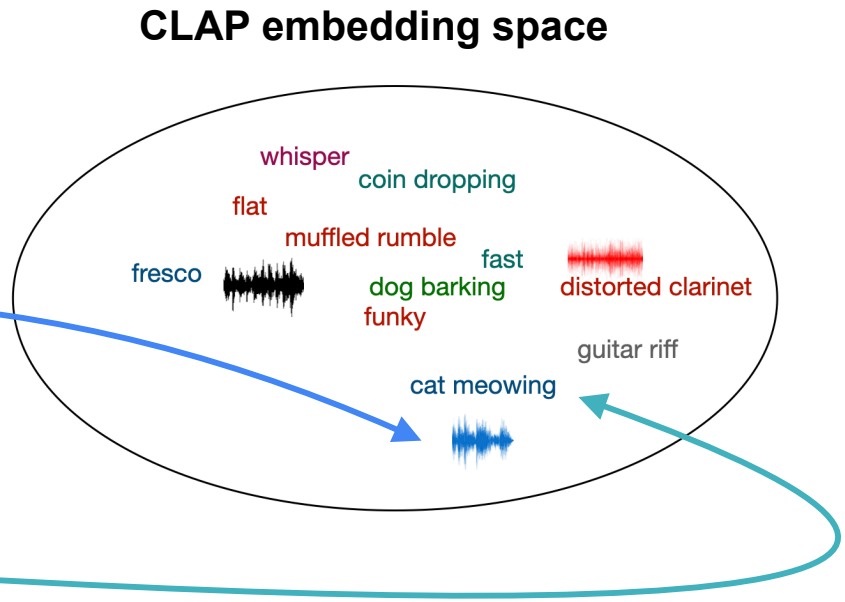
How do we evaluate multimodal embedding models?

Things we want to check

Cross-Modal Alignment (text <> audio)

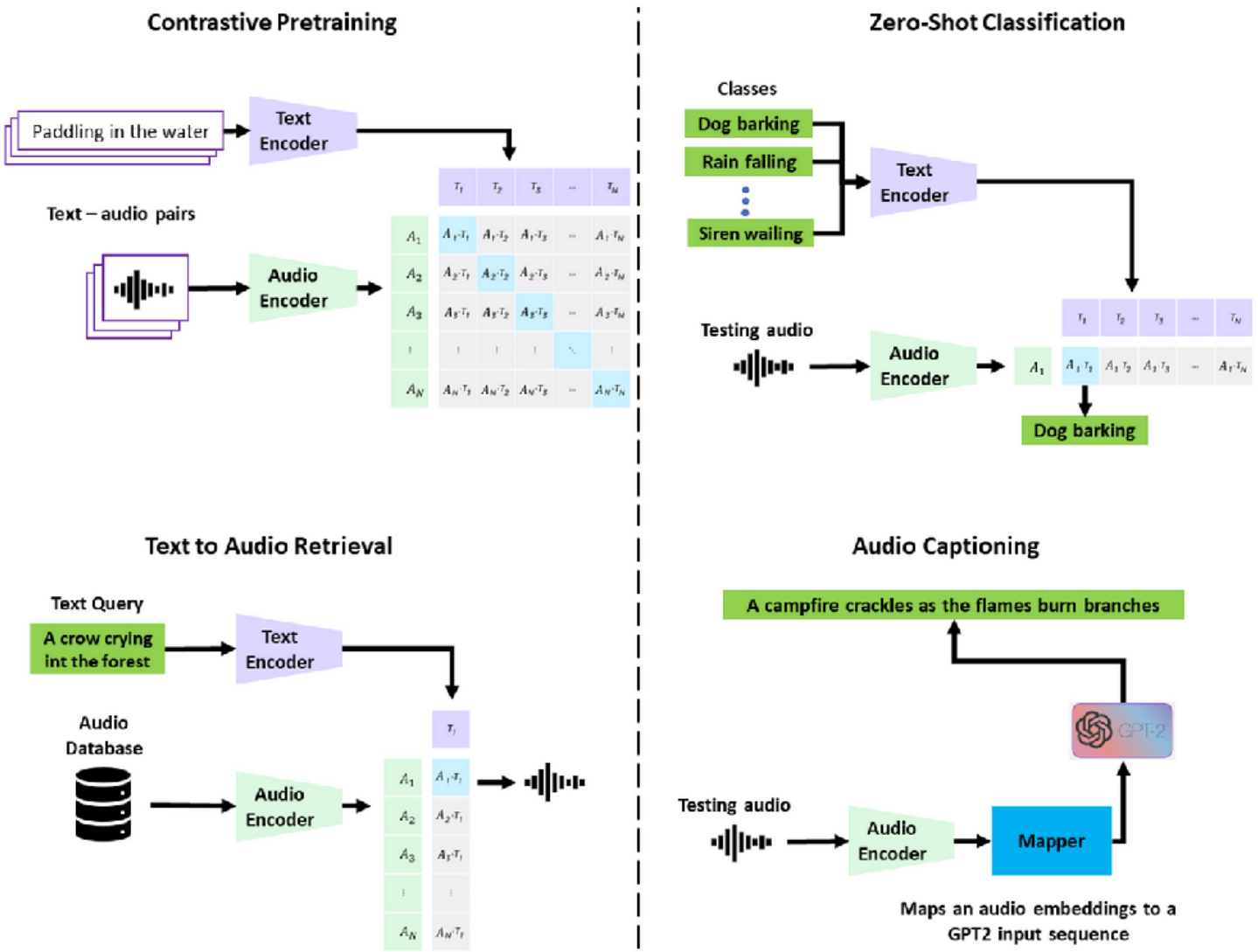
How well do the model's representations of different modalities (e.g., audio and text) align semantically?

Does the **sound of cat meow** have a high similarity score with the text **“cat meowing”**?



we can test this on core downstream tasks

CLAP core downstream tasks



Downstream Tasks

Task Type	Description	Common Metrics
Cross-Modal Retrieval	Match one modality (e.g., audio, image) to another (e.g., text) — e.g., "find the caption for this sound"	Recall@K, Precision@K, Median Rank, mAP
Classification	Predict labels (e.g., “guitar”, “clapping”) using single-modality embeddings with optional fine-tuning	Top-1 Accuracy, F1-score, Precision, Recall
Captioning / Generation	Generate descriptive text from audio, image, or video	BLEU, ROUGE, CIDEr, METEOR, SPICE
Auditory QA (VQA/ AQA)	Answer multiple choice questions based on auditory input and associated text	QA Accuracy, Exact Match, VQA Score
Zero-Shot Learning	Perform tasks with no labeled examples — often via alignment in shared embedding space	Accuracy, F1-score, Recall@K (task-dependent)
Human Evaluation	Collect subjective ratings of match quality, fluency, or semantic correctness	Relevance, Fluency, Preference Scores, Likert Ratings

QA benchmarks

e.g. MuChoMusic

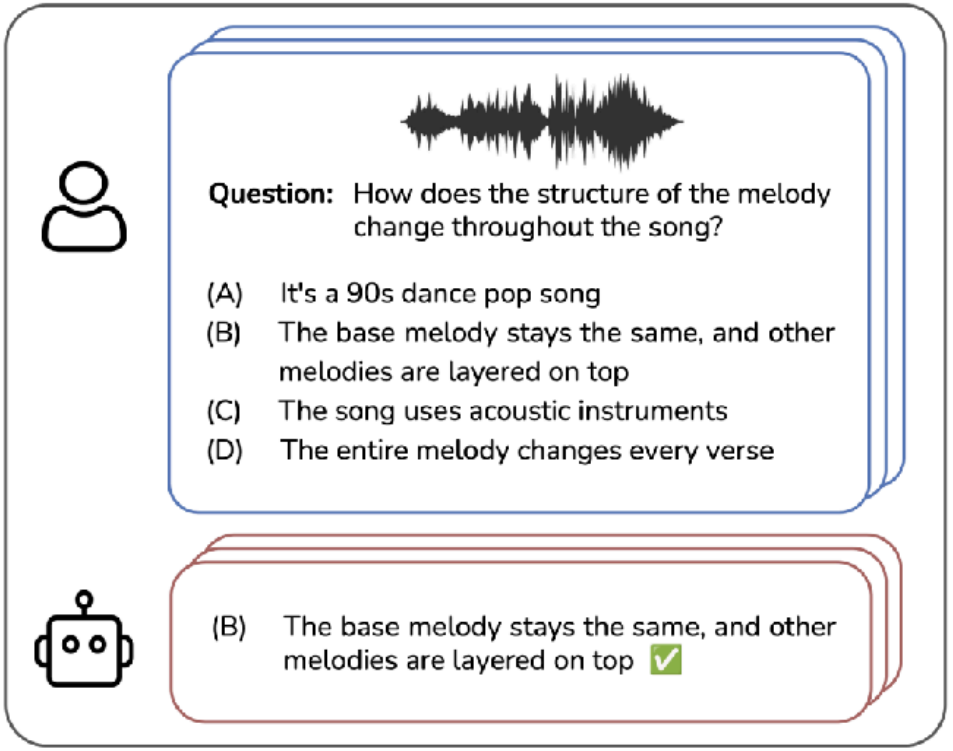


Figure 1. Multiple-choice questions in MuChoMusic have four answer options of different levels of difficulty.

B. Weck, I. Manco, E. Benetos, E. Quinton, G. Fazekas, and D. Bogdanov, "MuchoMusic: Evaluating music understanding in multimodal audio-language models," arXiv preprint arXiv:2408.01337, 2024.

Cross-Modal Retrieval

Task: Given audio, retrieve the matching text, or vice versa

rank-agnostic metrics

quality $\text{Precision@K} = \frac{\text{Number of relevant items in top K}}{K}$

ex. Precision@5: 60% of top 5 retrieved results are relevant

coverage $\text{Recall@K} = \frac{\text{Number of relevant items in top K}}{\text{Total number of relevant items in dataset for the query}}$

ex. Recall@1 = 70% means correct text is top result 70% of the time

both $\text{F1@K} = 2 \times \frac{\text{Precision@K} \times \text{Recall@K}}{\text{Precision@K} + \text{Recall@K}}$

Cross-Modal Retrieval: Precision@K

Task: Given text query, retrieve matching audio clips

Text Query

`"a flock of birds chirping in the morning"`

Rank	Retrieved Audio Label	Relevant (bird-related)?
1	`Birds chirping in forest`	YES
2	`City traffic and sirens`	no
3	`Seagulls near the ocean`	yes
4	`Children playing at a park`	no
5	`Songbirds in early morning`	yes

Out of the results it retrieved, how many were actually relevant?

Relevant items in Top 5: 3
Total retrieved (K): 5

$$\text{Precision@K} = \frac{\text{Number of relevant items in top K}}{K}$$

$$\text{Precision@5} = \frac{3}{5} = 0.6 \text{ or } 60\%$$

Cross-Modal Retrieval: Recall@K

Task: Given text query, retrieve matching audio clips

Text Query

`"a flock of birds chirping in the morning"`

Rank	Retrieved Audio Label	Relevant (bird-related)?
1	`Birds chirping in forest`	YES
2	`City traffic and sirens`	no
3	`Seagulls near the ocean`	yes
4	`Children playing at a park`	no
5	`Songbirds in early morning`	yes

Out of all the relevant items in the dataset, how many did it manage to retrieve?

From ground truth metadata, say we know there are 10 total bird-related audio clips

Relevant items retrieved: 3
Total relevant items in dataset: 10
Total retrieved (K): 5

$$\text{Recall@K} = \frac{\text{Number of relevant items in top K}}{\text{Total number of relevant items in dataset for the query}}$$

$$\text{Recall@5} = \frac{3}{10} = 0.3 \text{ or } 30\%$$

*Only 3 of the 10 possible bird-related audio clips were retrieved in the top 5. So while **Precision@5 was 60%**, **Recall@5 is only 30%** — showing that although our top results were reasonably accurate, the system **missed many other relevant clips** in the dataset.*

Cross-Modal Retrieval: F1@K

Task: Given text query, retrieve matching audio clips

Text Query

`"a flock of birds chirping in the morning"`

Rank	Retrieved Audio Label	Relevant (bird-related)?
1	`Birds chirping in forest`	YES
2	`City traffic and sirens`	no
3	`Seagulls near the ocean`	yes
4	`Children playing at a park`	no
5	`Songbirds in early morning`	yes

How balanced was the system's accuracy and coverage?

$$\text{F1@K} = 2 \cdot \frac{\text{Precision@K} \cdot \text{Recall@K}}{\text{Precision@K} + \text{Recall@K}}$$

$$\text{F1@5} = 2 \cdot \frac{0.6 \cdot 0.3}{0.6 + 0.3} = \frac{0.36}{0.9} = 0.4$$

Cross-Modal Retrieval

Task: Given audio, retrieve the matching text, or vice versa

rank-based metrics

Mean reciprocal rank

$$\text{MRR} = \frac{1}{Q}\sum_{q=1}^{Q}\frac{1}{\text{rank of first relevant item for query } q}$$

what rank was the first relevant item?

avg precision (per query)

$$\text{AP} = \frac{1}{R}\sum_{k=1}^{N}\text{Precision@k}\cdot 1[\text{item at } k \text{ is relevant}]$$

Mean AP

$$\text{mAP} = \frac{1}{Q}\sum_{q=1}^{Q}\text{AP}_q$$

R: Total number of relevant items for the query
N: Total number of returned items (can be all or top-K)
1[·]: 1 if item is relevant, 0 otherwise

do relevant items appear fairly early in the ranked list?

Cross-Modal Retrieval: MRR

Task: Given text query, retrieve matching audio clips

Text Query

`"a flock of birds chirping in the morning"`

Rank	Retrieved Audio Label	Relevant (bird-related)?
1	`Birds chirping in forest`	YES
2	`City traffic and sirens`	no
3	`Seagulls near the ocean`	yes
4	`Children playing at a park`	no
5	`Songbirds in early morning`	yes

How soon was the first relevant item retrieved?

First relevant rank: 1

$$\text{MRR} = \frac{1}{1} = 1.0$$

(If first relevant was at rank 3, MRR = 1/3 ≈ 0.3)

Cross-Modal Retrieval: Average Precision

Task: Given text query, retrieve matching audio clips

Text Query

`"a flock of birds chirping in the morning"`

Rank	Retrieved Audio Label	Relevant (bird-related)?
1	`Birds chirping in forest`	YES
2	`City traffic and sirens`	no
3	`Seagulls near the ocean`	yes
4	`Children playing at a park`	no
5	`Songbirds in early morning`	yes

Where did the relevant results appear in the ranking?

Relevant ranks: 1,3,5

Calculate precision at every relevant position
P@1 = 1/1 = 1
P@3 = 2/3 = 0.67
P@5 = 3/5 = 0.6

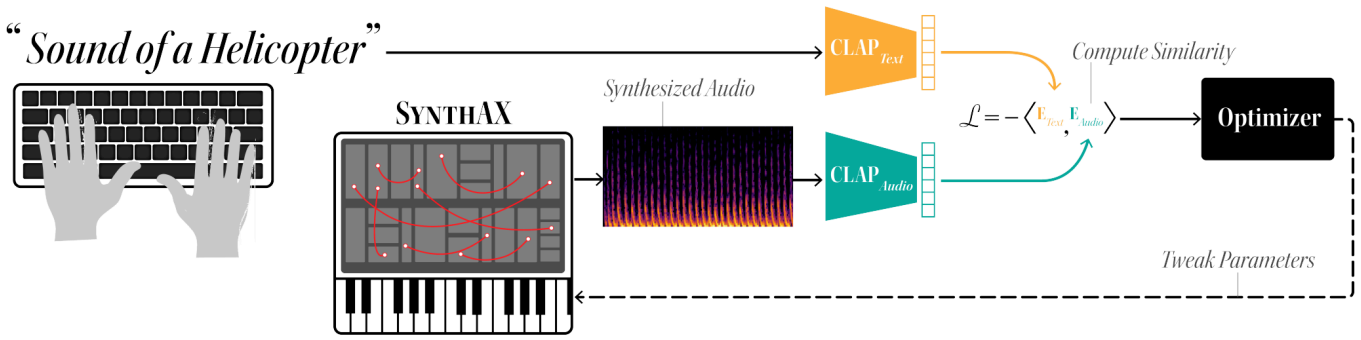
$$\text{AP} = \frac{1.0 + 0.667 + 0.6}{3} \approx 0.756$$

if we had multiple queries, we'd do the same for them then take avg for mAP

How else have people used CLAP? Fun applications

Text-to-Audio Generation: Controlling a Synthesizer

CLAP's embedding space as loss space



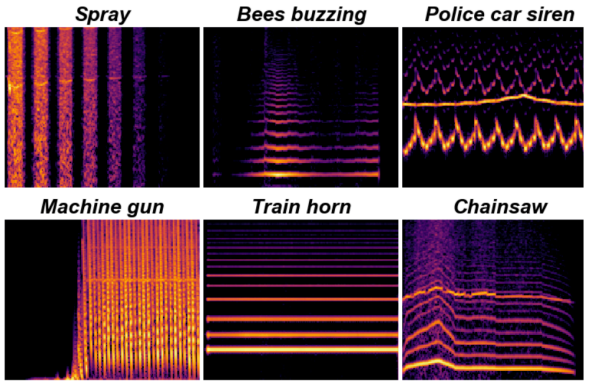


Figure 1. CTAG leverages a virtual modular synthesizer to generate sounds capturing the semantics of user-provided text prompts in a sketch-like way, rather than being acoustically literal. Spectrograms of auditory outputs corresponding to six text prompts showcase the range of sounds this approach can yield, accompanied by a fully interpretable and controllable parameter space.

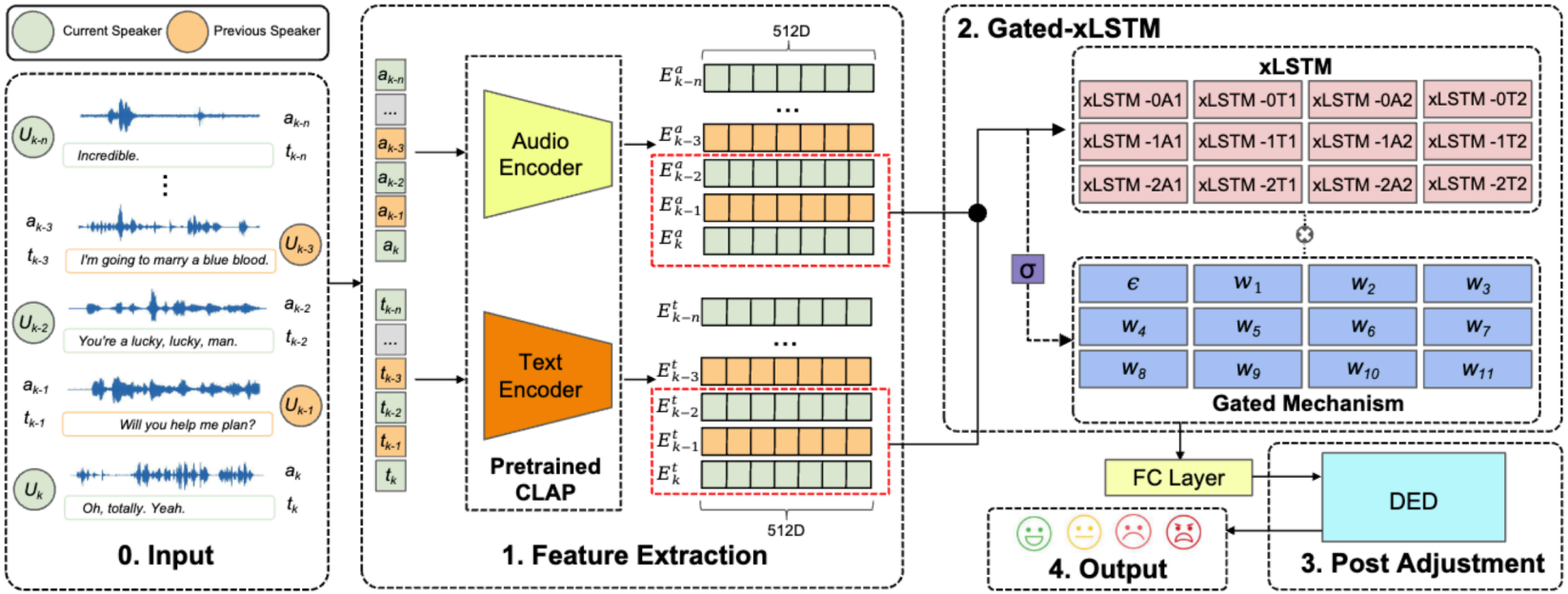
M. Cherep, N. Singh, and J. Shand, "Creative Text-to-Audio Generation via Synthesizer Programming," arXiv preprint arXiv:2406.00294, 2024. [Online]. Available: https://arxiv.org/abs/2406.00294

EXAMPLEs: https://ctag.media.mit.edu/

How else have people used CLAP? Fun applications

Speech emotion recognition

CLAP embeddings as input feature vector



Y. Li, Q. Sun, S. M. Krishna Murthy, E. Alturki, and B. W. Schuller, "GatedxLSTM: A multimodal affective computing approach for emotion recognition in conversations," arXiv preprint arXiv:2503.20919, 2025. [Online]. Available: https://api.semanticscholar.org/CorpusID:277349399

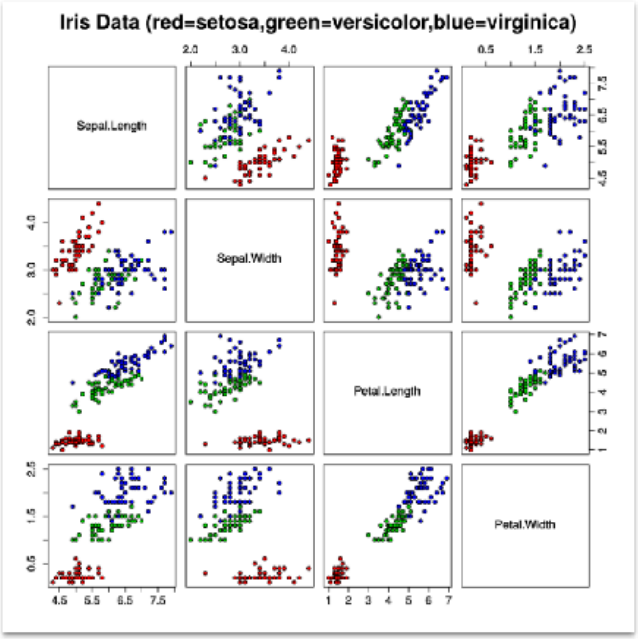
Dimensionality Reduction

CS352 Winter 2026
Bryan Pardo & Annie Chu

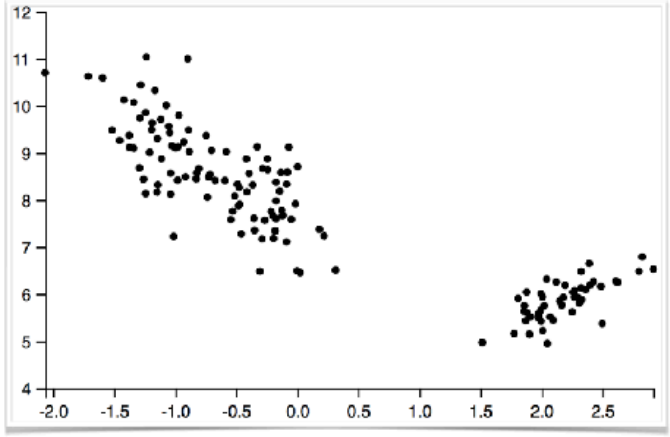
Dimensionality reduction

Goal: Map high dimensional data onto lower-dimensional data in a manner that preserves *distances/similarities*

Original Data (4 dims)



Projection with PCA (2 dims)



Objective: projection should "preserve" relative distances

slide from Bryon Wallace Northeastern

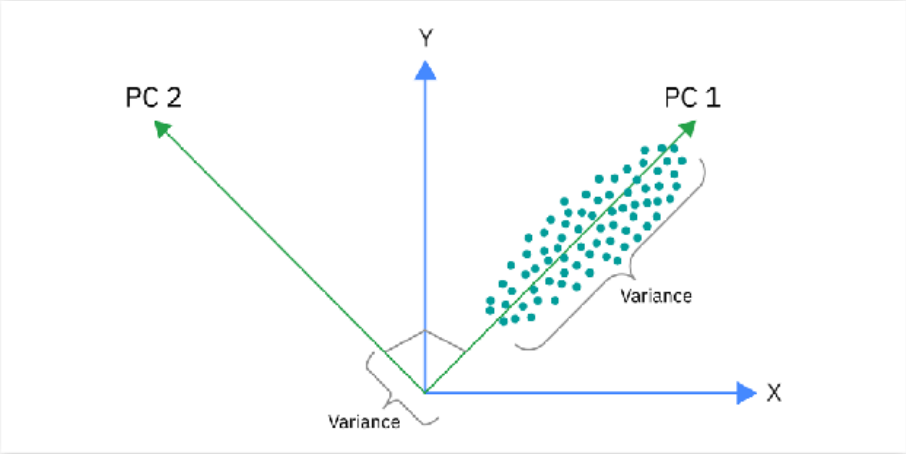
PCA: Principal Component Analysis

TLDR; Linear technique that works to maximize global variance

PCA finds **new axes (called principal components)** along which the **data varies the most**. These axes are linear combinations of the original features.

What it does: Finds the directions with the most variation in data.

How? It uses eigenvectors and eigenvalues of the covariance matrix to find those directions



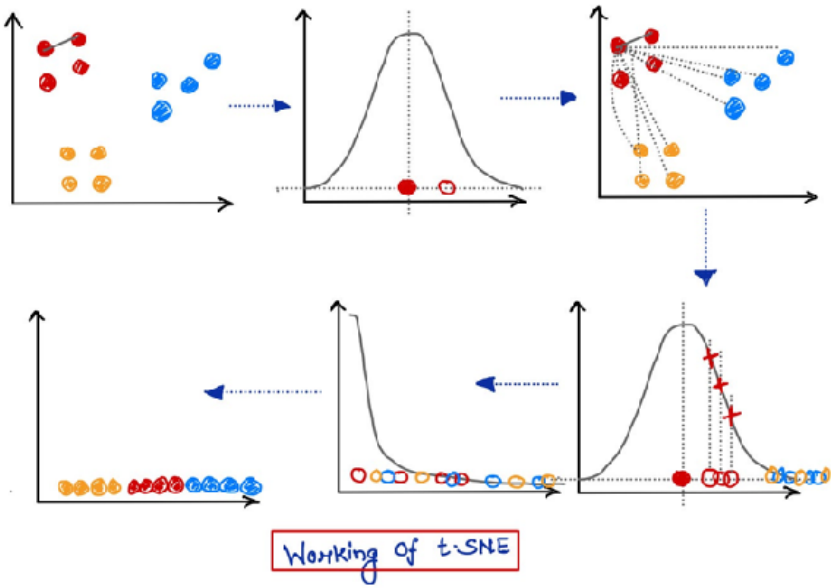
source: ibm

t-Distributed Stochastic Neighbor Embedding (t-SNE)

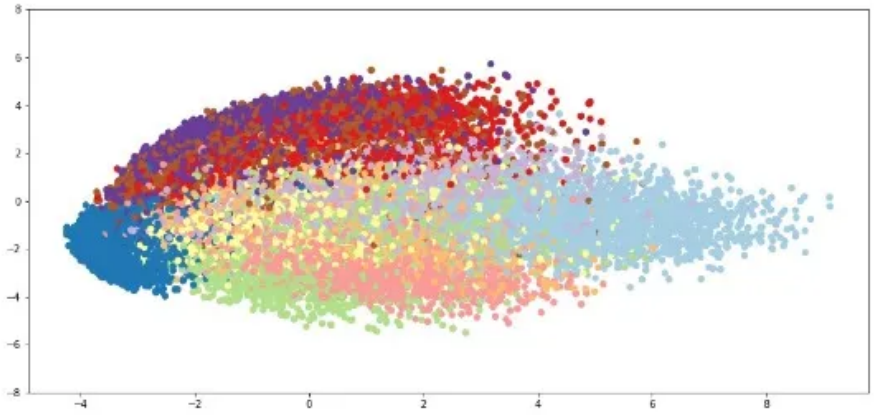
TLDR; Non-linear technique that preserves local structure by modeling pairwise similarities.

t-SNE arranges data in a way that keeps **similar items close together in the low-dimensional space**, making clusters easy to see.

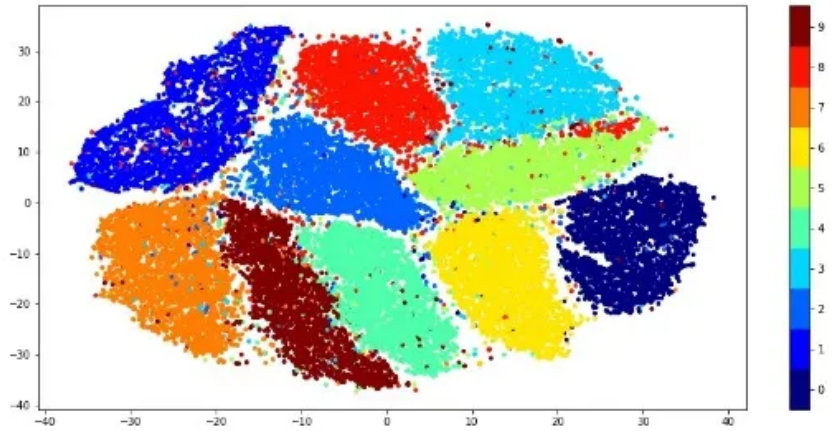
*Computes **pairwise similarities using Gaussian distributions** in high dimensions and **Student t-distributions** in low dimensions, then minimizes distance between distributions via KL divergence*



MNIST - PCA



MNIST - TSNE

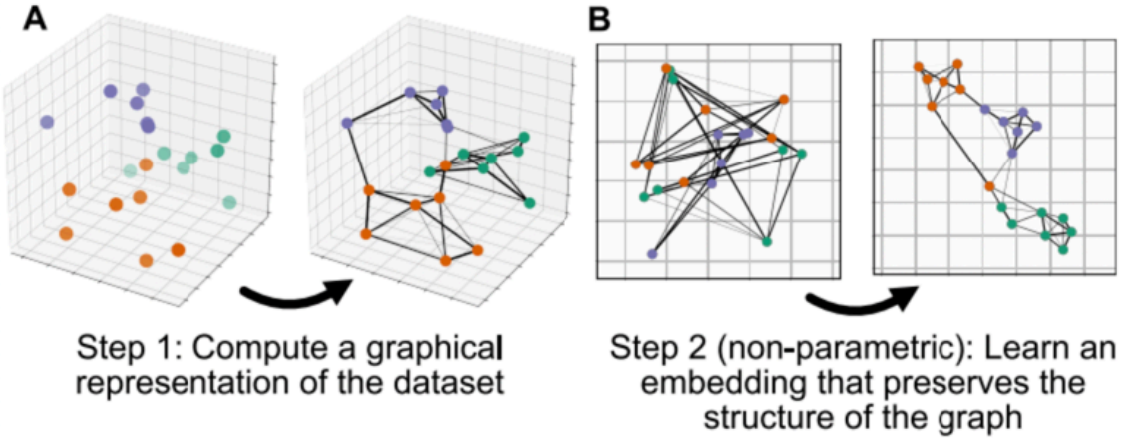


Uniform Manifold Approximation and Projection (UMAP)

TLDR; Non-linear technique that preserves both local + some global structure, scalable for large datasets.

UMAP constructs a **weighted graph** of the data's local structure and then optimizes a low-dimensional layout that preserves those relationships and the overall shape

Build a nearest-neighbor graph of the high-D data model local relationships, then learning a low-D embedding by minimizing a CE loss to aligns the graph structure with a similar graph in the lower-D space



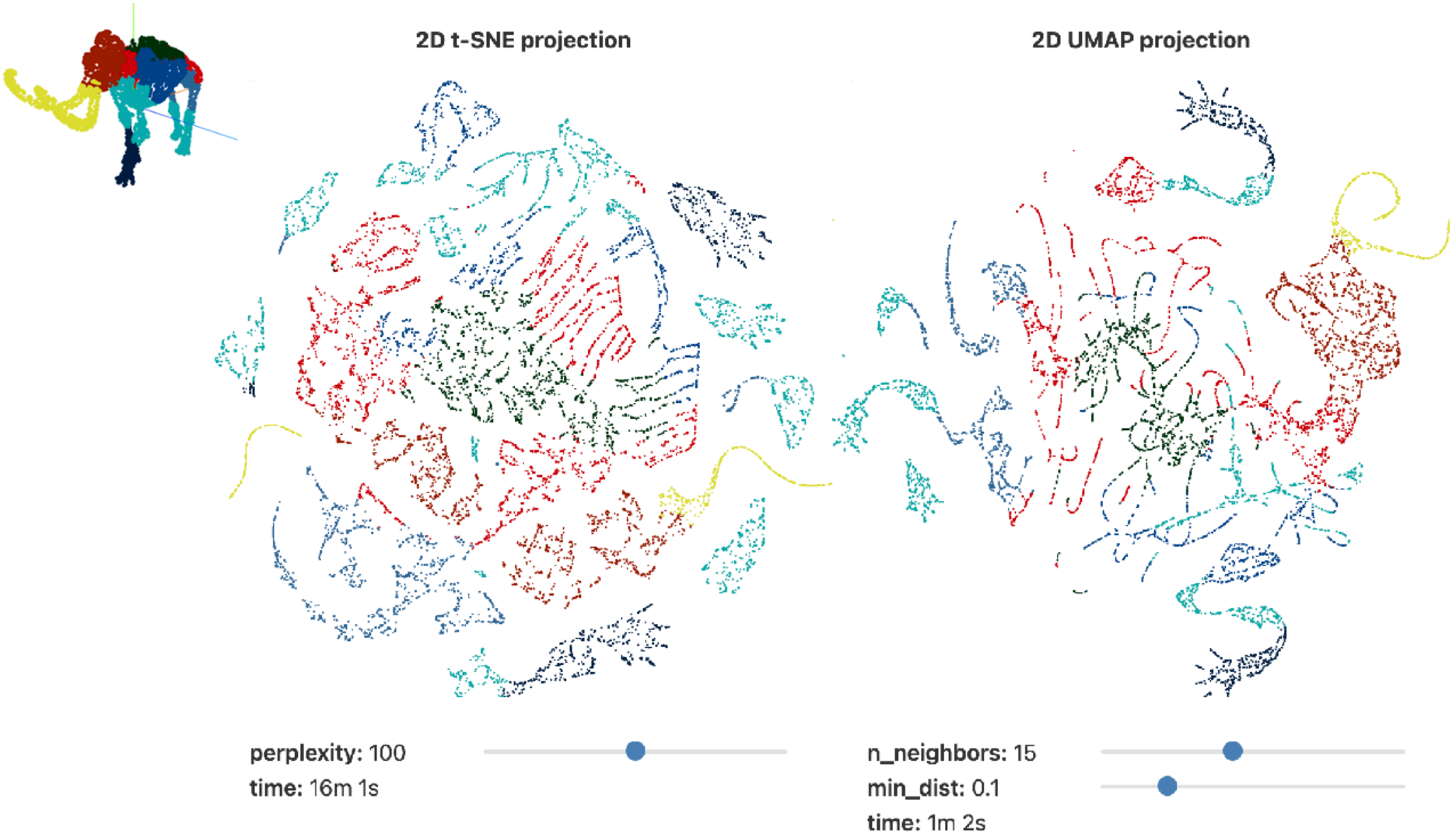


Figure 6: *A comparison between UMAP and t-SNE projections of a 3D woolly mammoth skeleton (50,000 points) into 2 dimensions, with various settings for parameters. Notice how much more global structure is preserved with UMAP, particularly with larger values of* `n_neighbors`.

Summary of PCA vs t-SNE vs UMAP

Feature	PCA	t-SNE	UMAP
Type	Linear	Non-linear	Non-linear
Goal	Maximize global variance	Preserve local structure	Preserve local and global structure
Preserves	Global variance patterns	Local clusters and pairwise similarity	Local neighborhoods and global layout
Interpretability	High	Low	Medium
Speed	Fast	Slow	Fast
Deterministic	Yes	No	Mostly (some stochastic elements)
Best For	Feature compression, initial dimensionality reduction	Visualizing cluster structure in compact space for small-medium datasets	Exploring both local clusters and broader relationships
Limitations	Cannot capture non-linear patterns	Distorts global structure, sensitive to parameters, slow for large datasets	Requires tuning, and still involves some randomness
Example use case	Reduce dimensionality of audio-text embedding space for downstream model input (feature engineering)	Visualize clusters of similar sound-text pairs (e.g., emotion categories, spoken keywords)	Understand large-scale relationships in joint audio-visual-text embeddings

When each might be good to use

Goal	Best Technique
Quick overview, compression, noise filtering	**PCA**
Visualizing clusters (e.g. categories)	**t-SNE**
Maintaining shape + cluster structure	**UMAP**
Handling very large or complex datasets	**UMAP**
Feature engineering for ML models	**PCA** (or UMAP but requires more digging)

Key Parameters

Key Parameter	PCA	t-SNE	UMAP
n_components (dimensionality)	Number of dimensions to retain aka what data to keep (important)	Output dimensionality mainly for visualization (doesn't impact the relationship structure in original space)	Output dimensionality mainly for visualization (doesn't impact the relationship structure in original space)
Neighborhood Size	Not applicable	perplexity – how many neighbors each point considers	n_neighbors – balances local vs. global structure
Cluster Spread	Not applicable	Not directly tunable	min_dist – controls spacing between points in embedding
random_state	Optional, for reproducibility	Affects layout stability	Controls reproducibility