ON THE CHOICE OF THE OPTIMAL TEMPORAL SUPPORT FOR AUDIO CLASSIFICATION WITH PRE-TRAINED EMBEDDINGS

Aurian Quelennec, Michel Olvera, Geoffroy Peeters, Slim Essid

1. Motivation

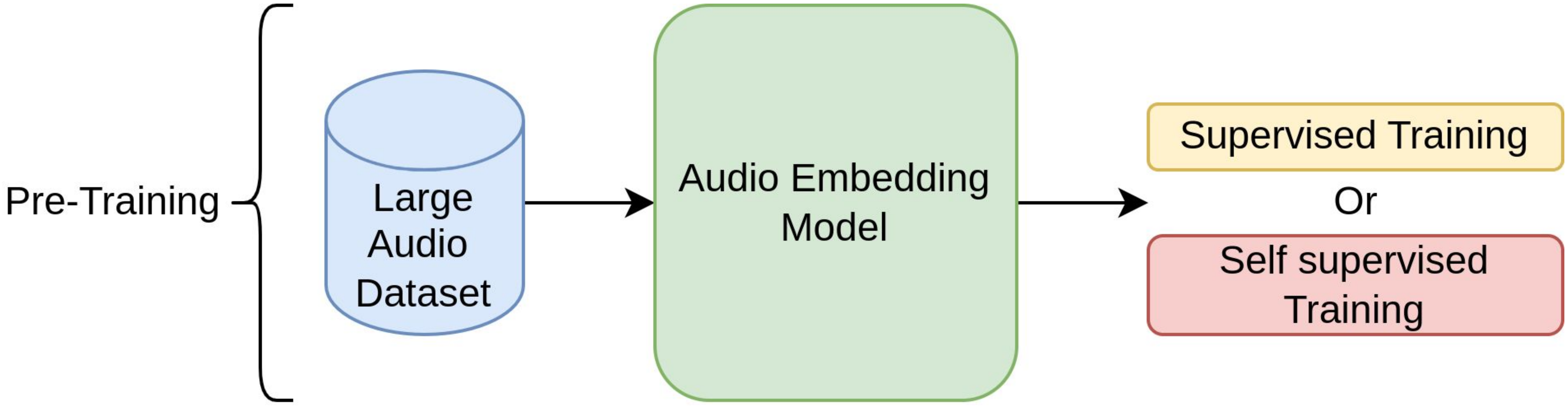
2. Method
    a. General Overview
    b. Technical details

3. Experiment
    a. Models
    b. Datasets
    c. Parameters

4. Results

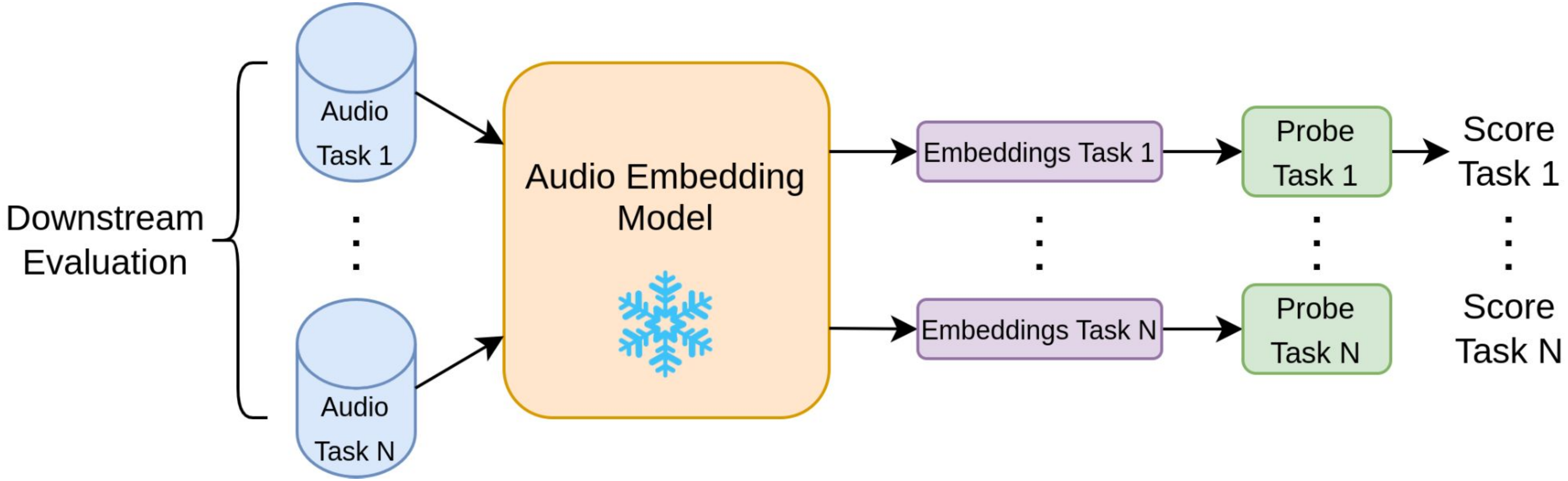
1- Motivation

- Current state-of-the-art audio analysis systems rely on pre-trained embedding models.



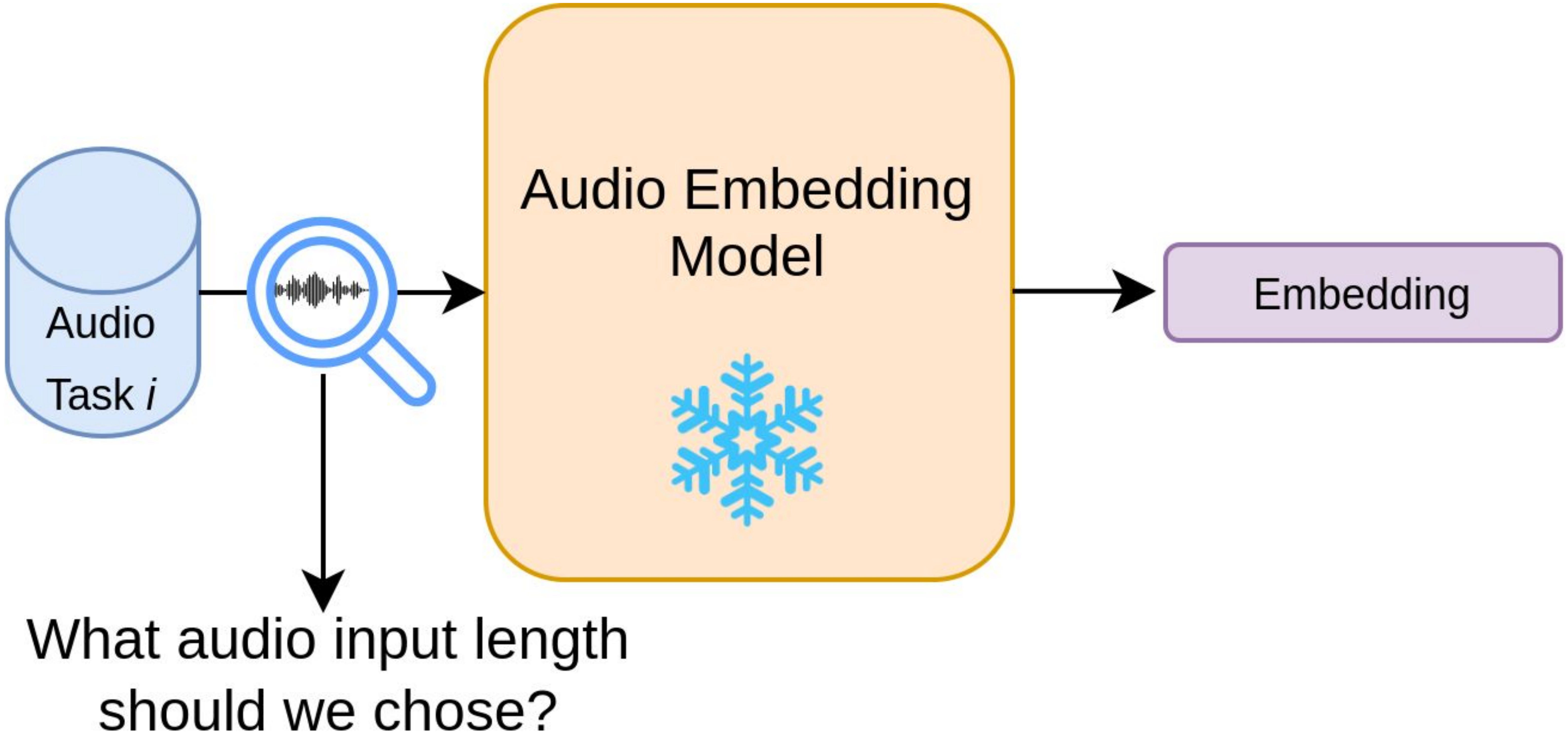
1- Motivation

- When used for a downstream classification task:
    a. Extract the pre-trained embeddings, used as features
    b. Use them to train a simple linear probe



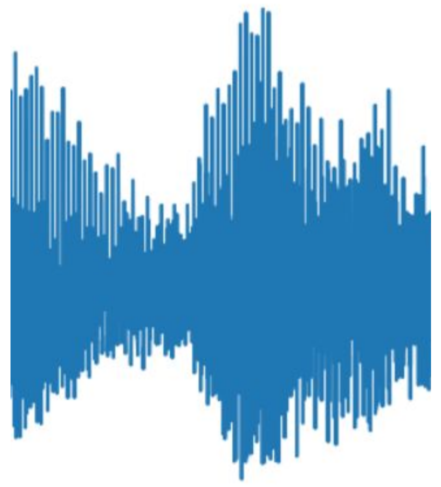
1- Motivation

- One aspect often overlooked in these works is the influence of the duration of audio segment considered to extract an embedding

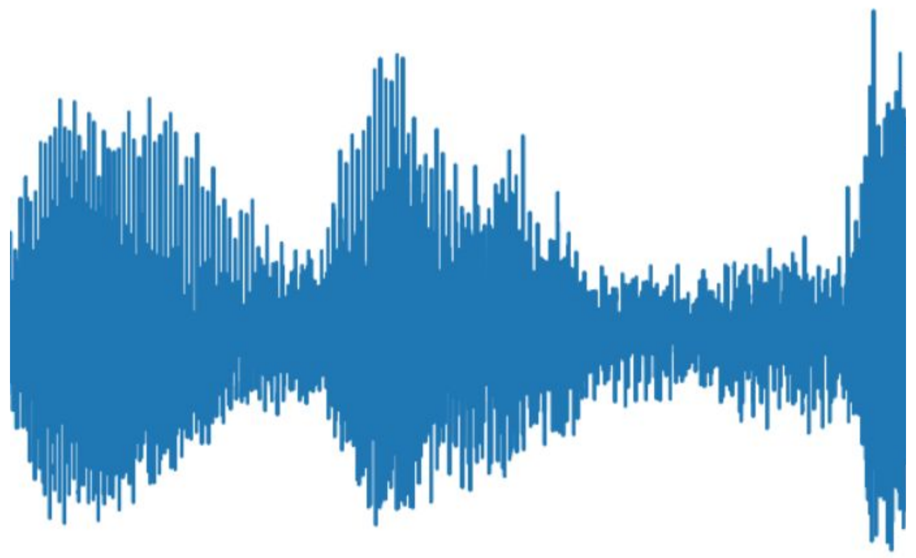


1- Motivation

- Does it have an impact on the downstream tasks’ scores?



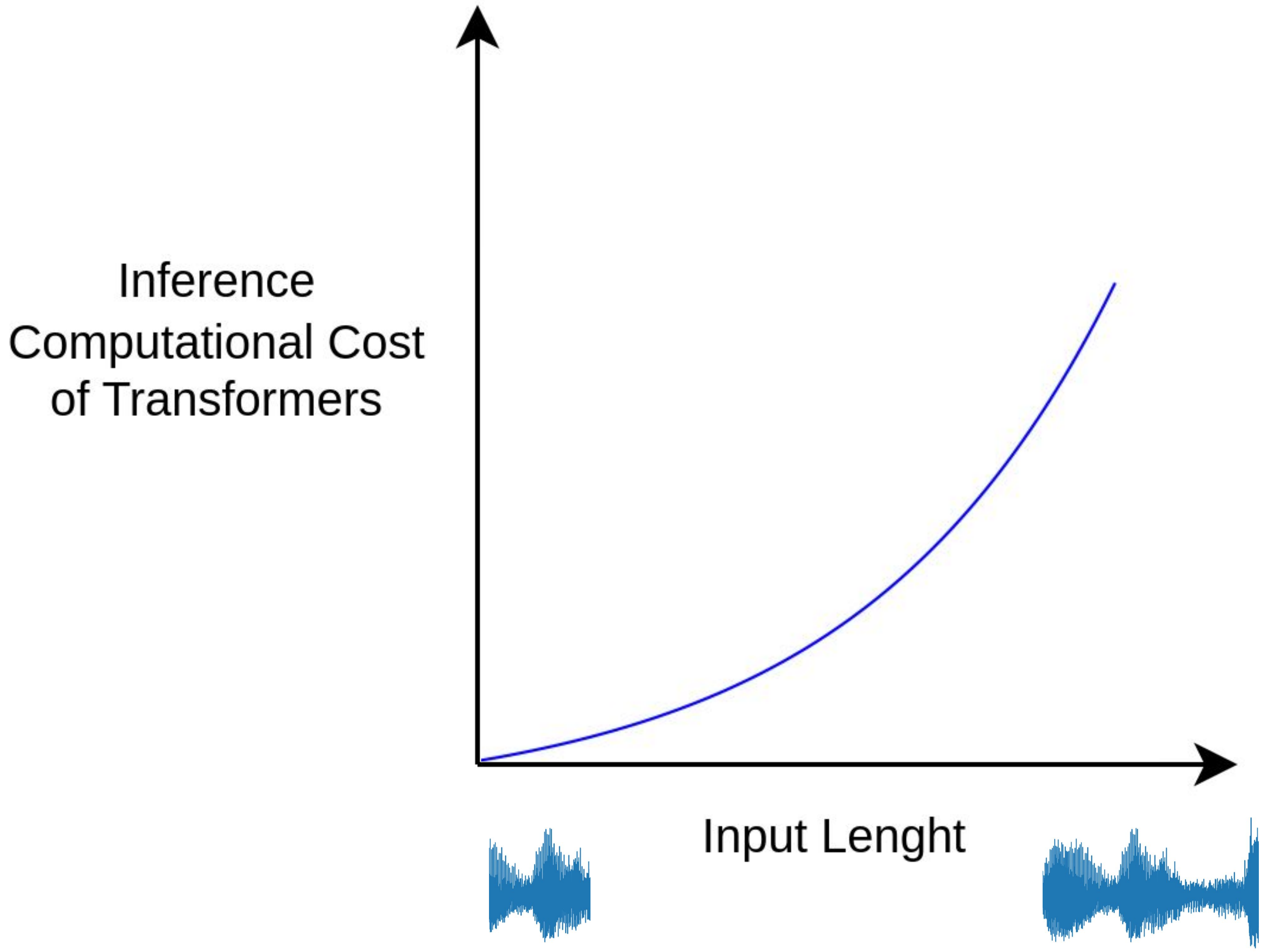
- Does using short audio segments implies bad scores?



- And using long audio segments good scores?

1- Motivation

- Can it help in reducing inference computational cost?



2- a) General Overview

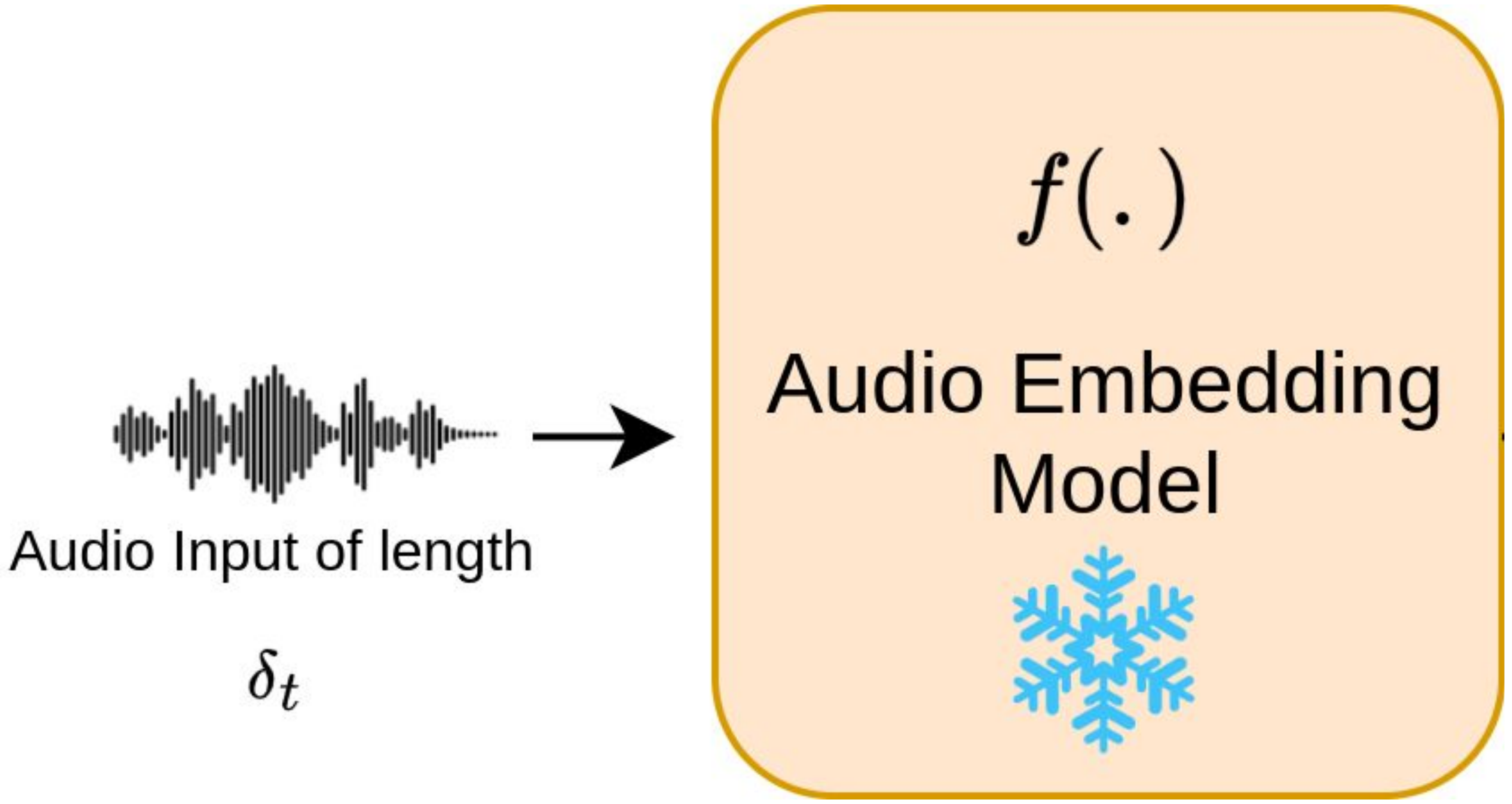
- We refer to the duration of audio input considered to extract an embedding as Temporal Support, denoted by δ_t

Audio Input of length

δ_t

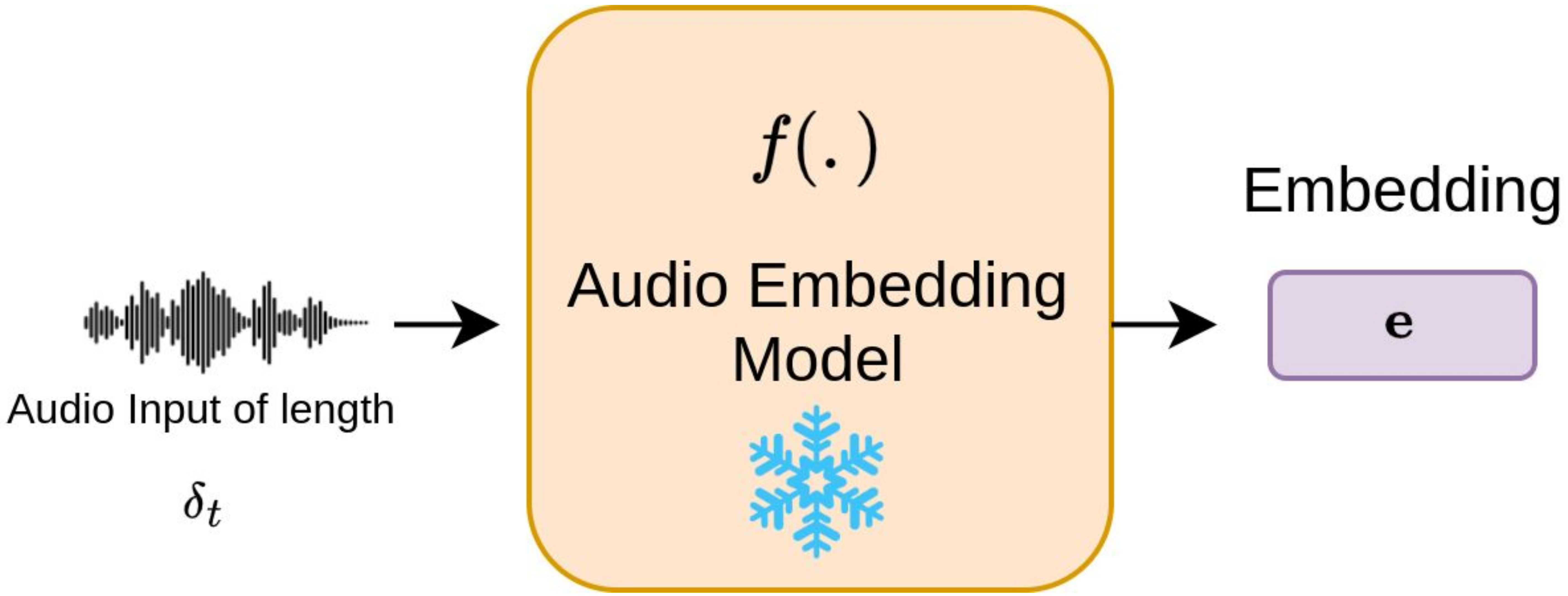
2- a) General Overview

- We use it with a frozen pre-trained model f(.)



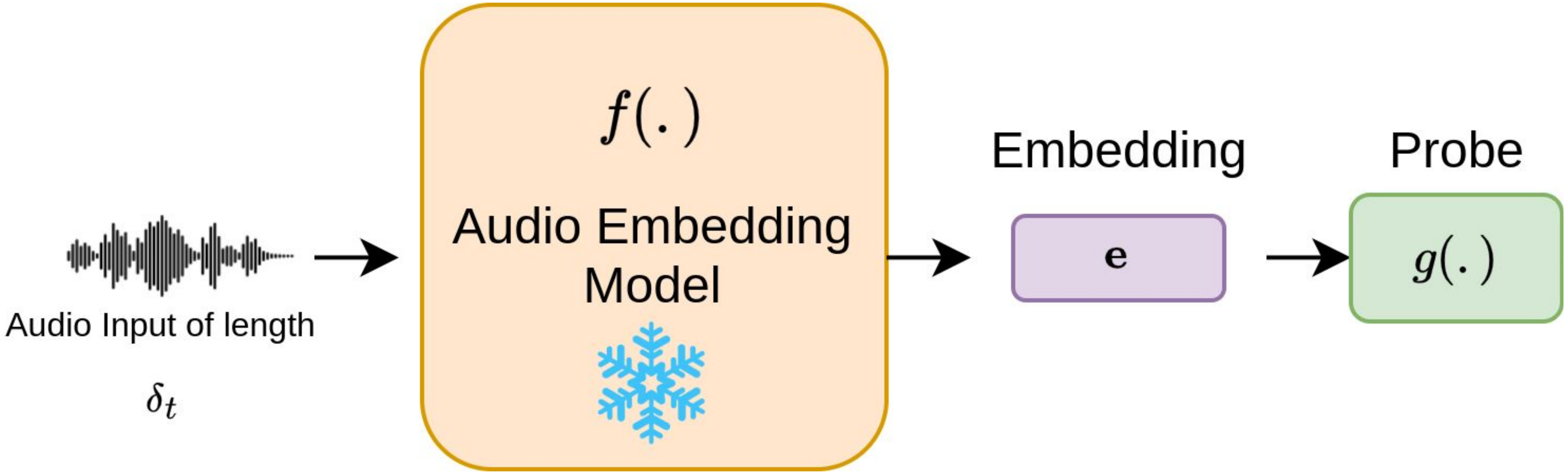
2- a) General Overview

- We obtain an embedding **e**



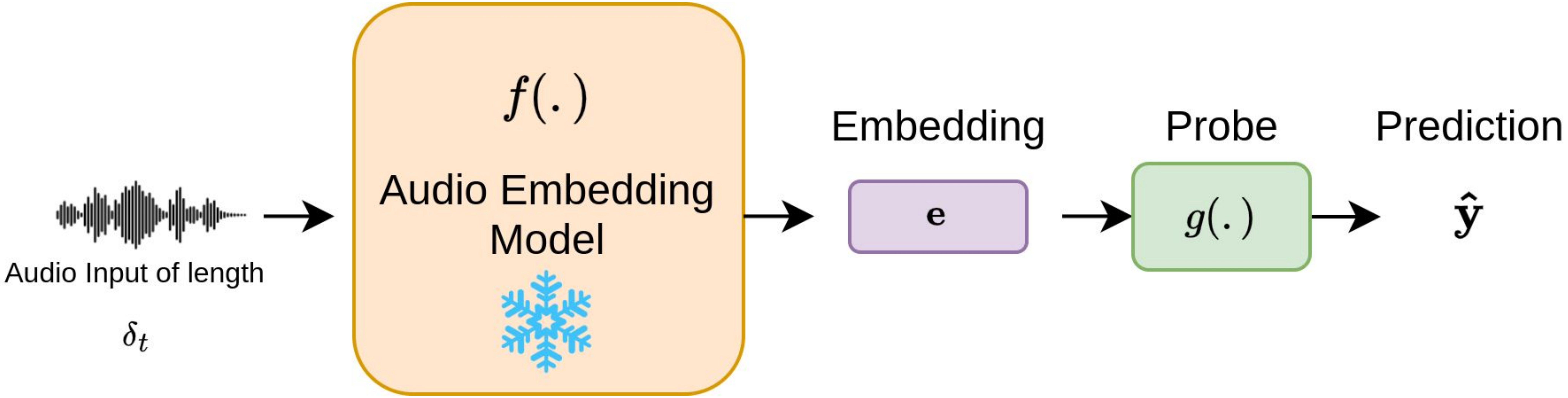
2- a) General Overview

- The embedding **e** is projected with a linear probe $g(.)$



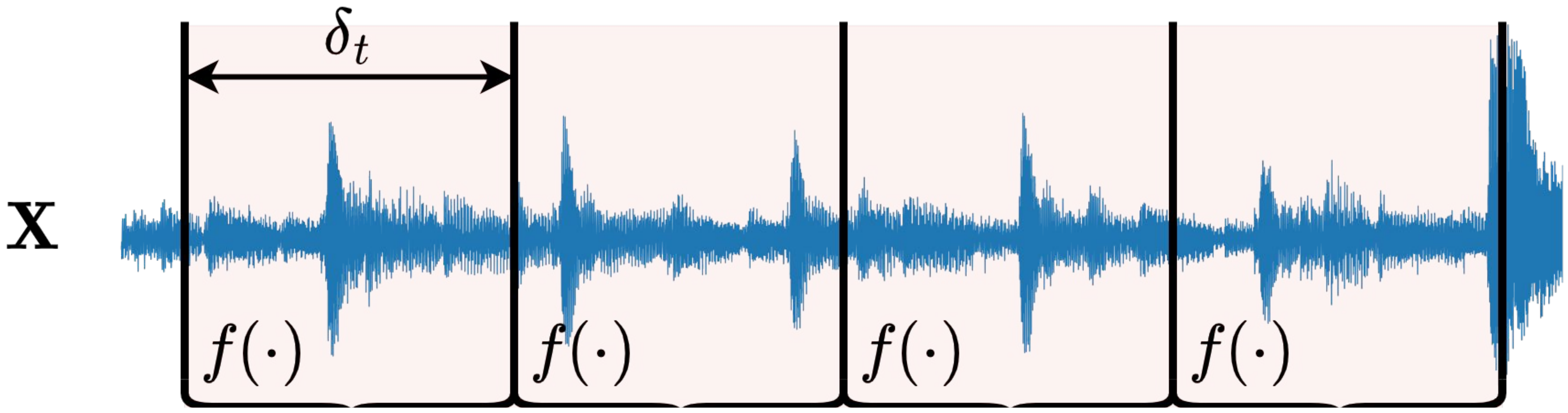
2- a) General Overview

- We thus obtain a local prediction $\hat{\mathbf{y}}$



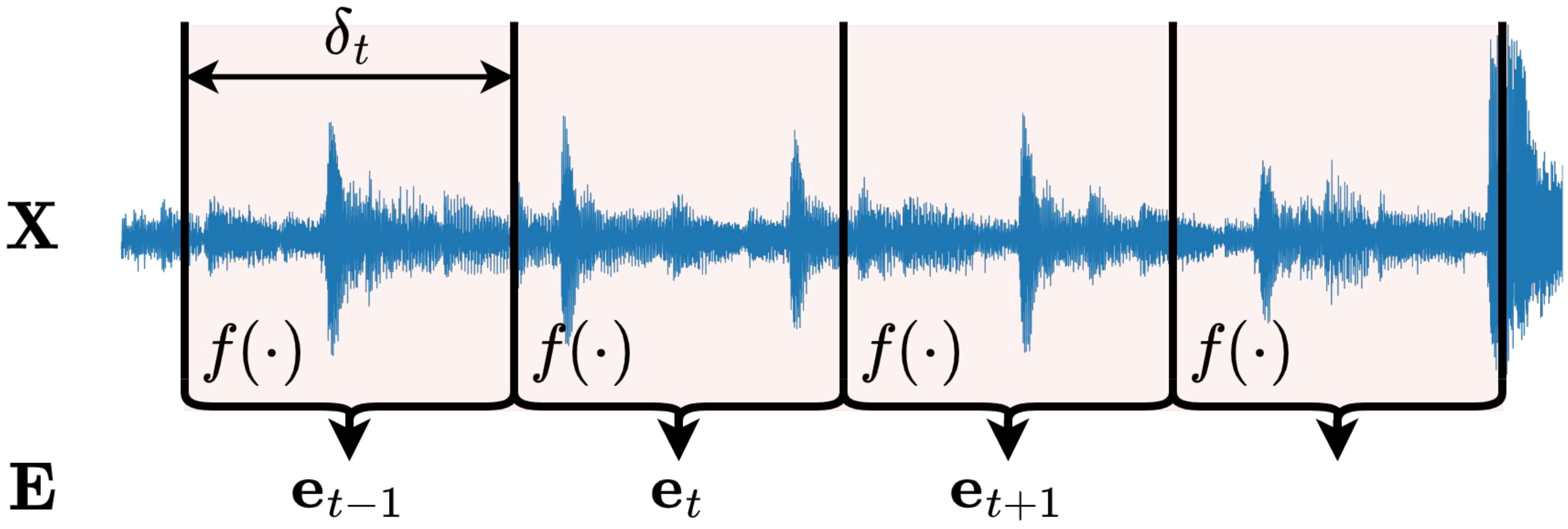
2- b) Technical details

- However in most downstream tasks, for a given audio example $\mathbf{x}$, we have δ_t shorter than the whole audio.



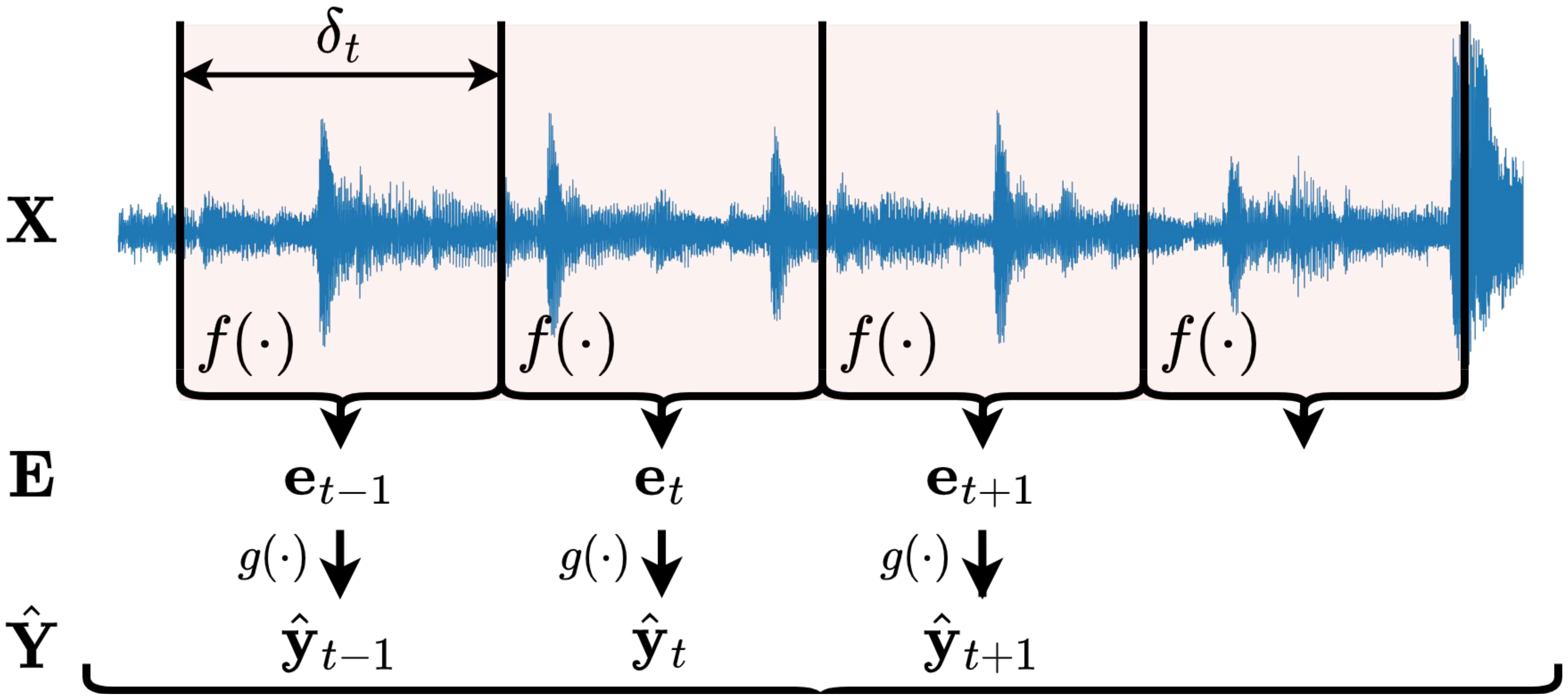
2- b) Detailed principles

- So we denote by $\mathbf{E}$ the embedding sequence extracted over $\mathbf{X}$ with $f(.)$.



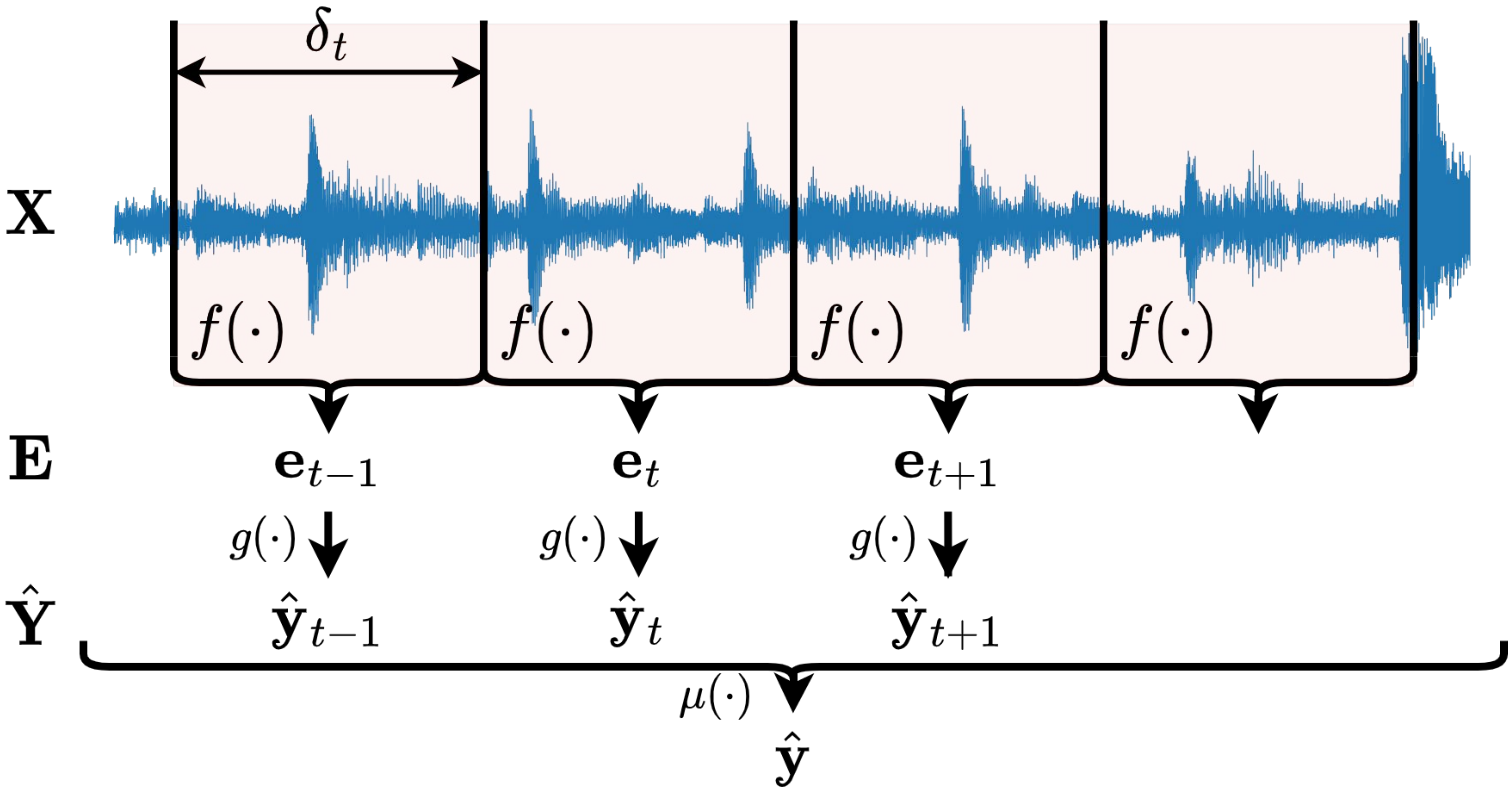
2- b) Detailed principles

- And we denote by $\hat{\mathbf{Y}}$ the prediction sequence derived from $\mathbf{E}$ with $g(.)$.



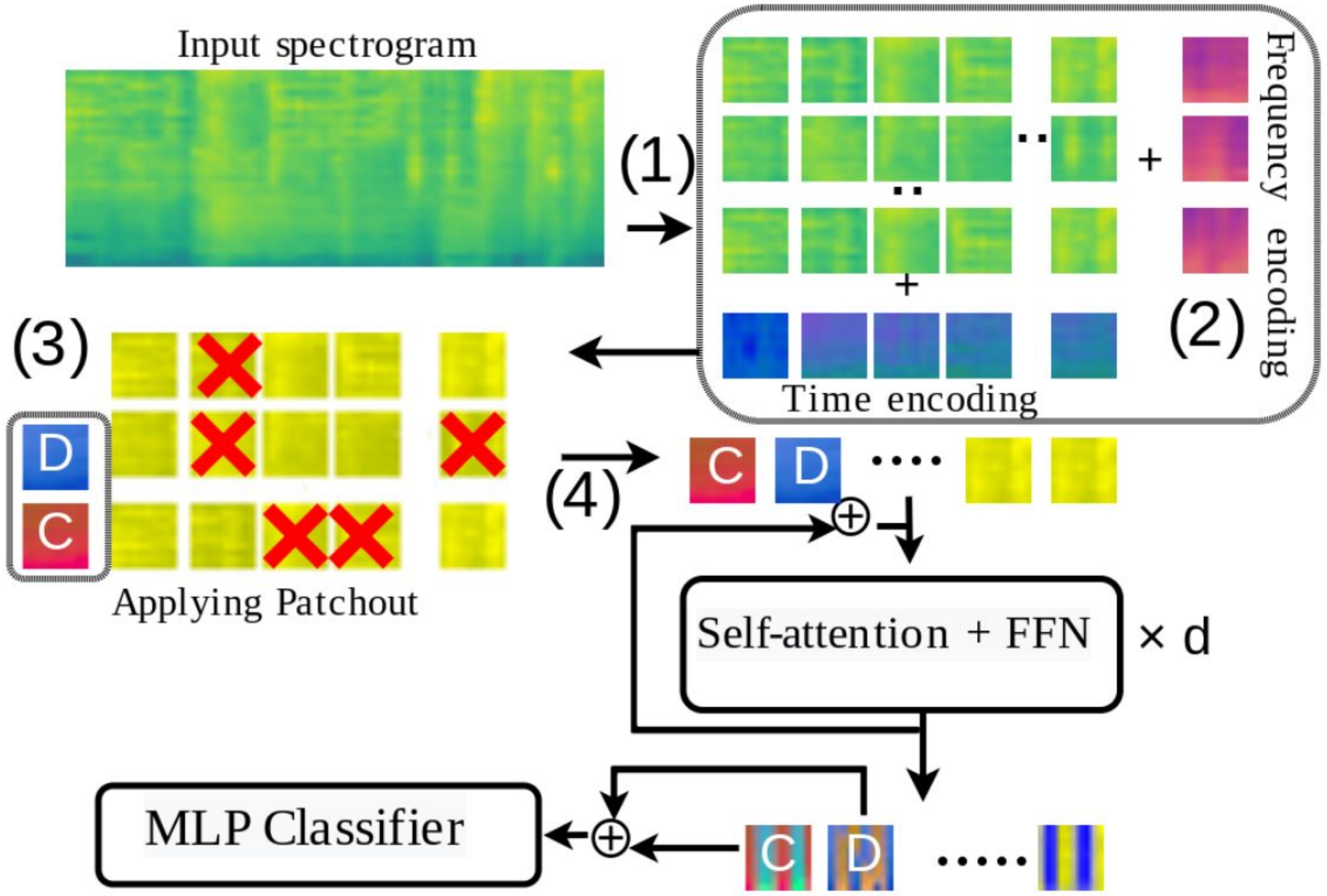
2- b) Detailed principles

- Finally, to obtain a clip-level prediction $\hat{\mathbf{y}}$ we use an aggregation function $\mu(.)$



3- a) Models

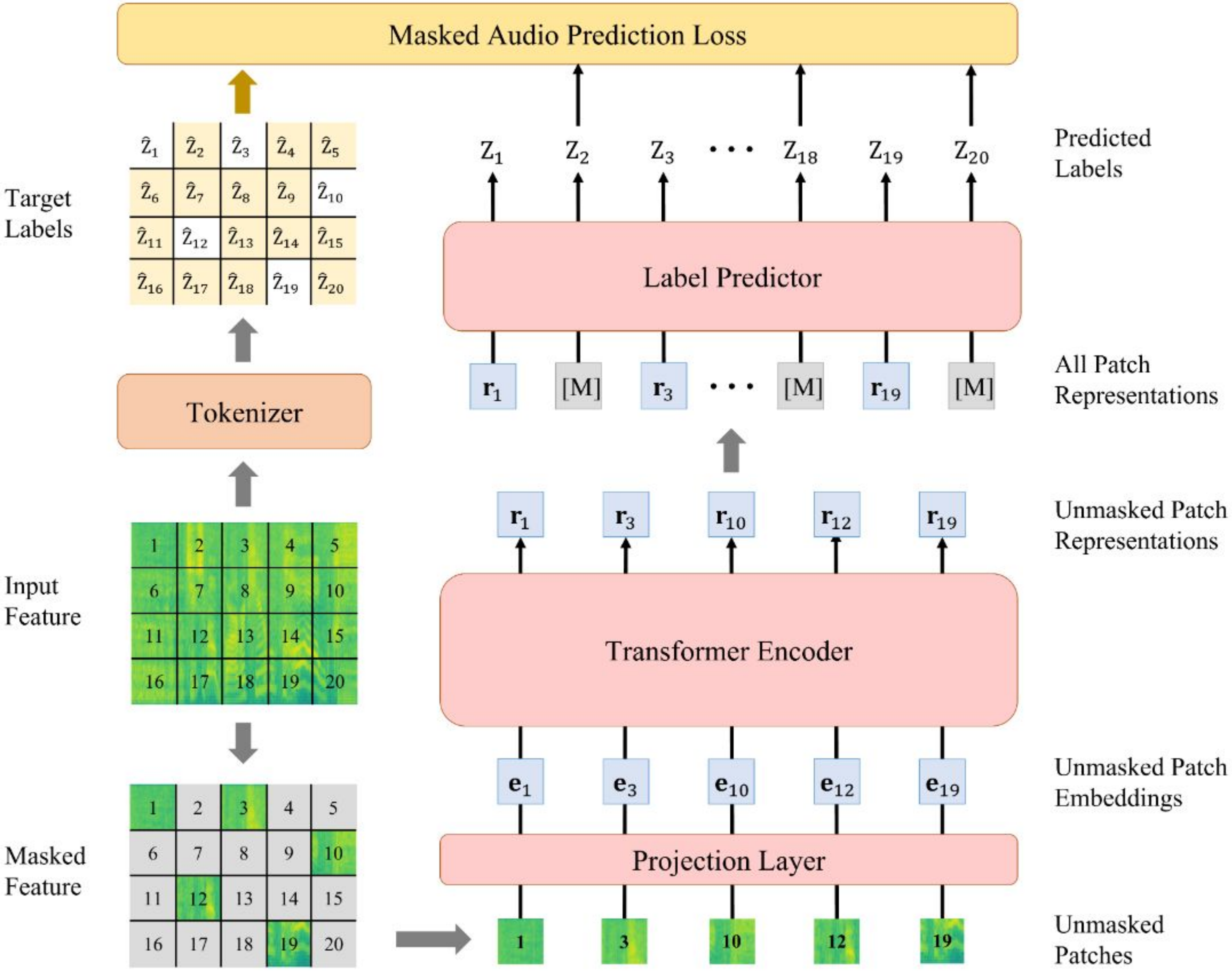
- PaSST [1] :
  - Audio Spectrogram Transformer based on ViT
  - Supervised training
  - Trained with δ_t of 10s



[1] Efficient Training of Audio Transformers with Patchout, *Koutini et al.* Interspeech 2022

3- a) Models

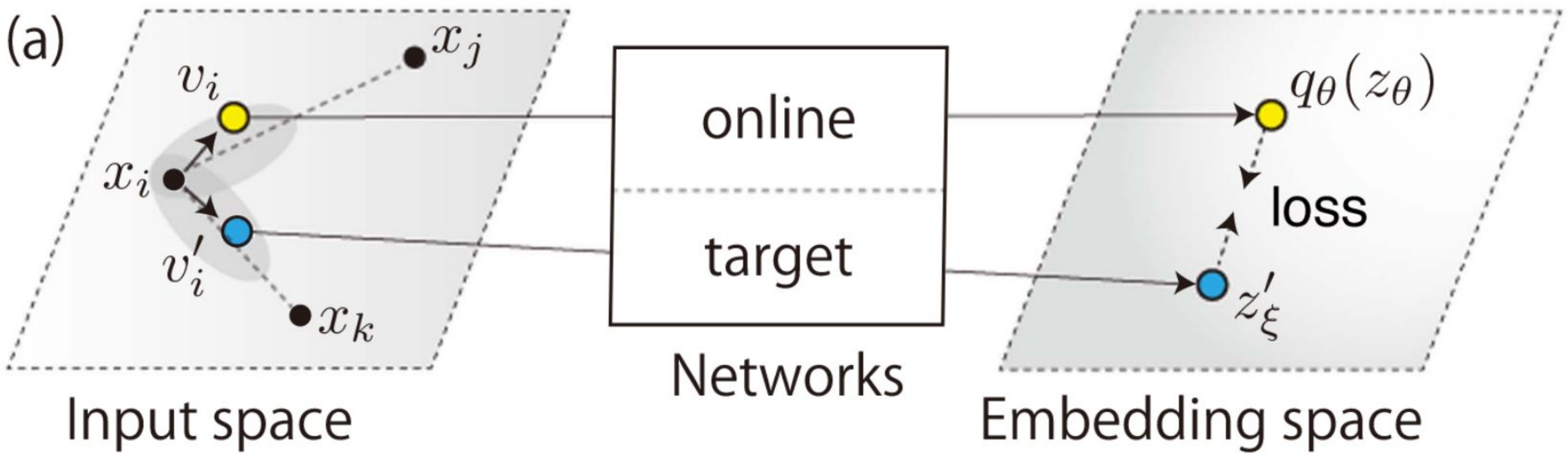
- BEATs [2]:
  - Audio Spectrogram Transformer based
  - Self Supervised Learning iterative training procedure
  - Trained with δ_t of 10s



[2] BEATs: Audio Pre-Training with Acoustic Tokenizers, *Chen et al.* ICML 2023

3- a) Models

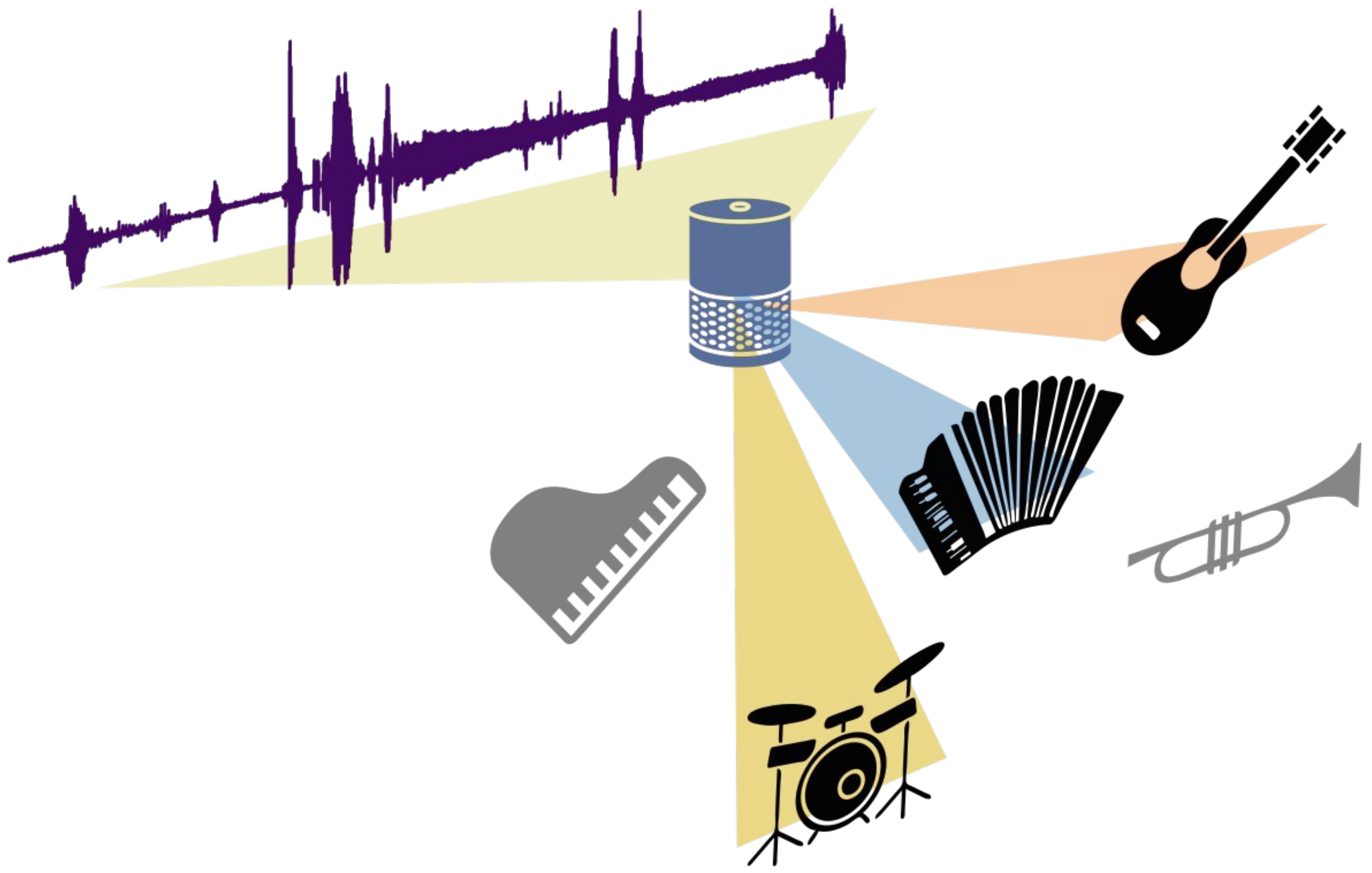
- BYOL-A [3]:
  - CNN based
  - SSL iterative training procedure
  - Trained with δ_t of 1s



[3] BYOL for Audio: Exploring Pre-Trained General-Purpose Audio Representations, *Niizumi et al.* IEEE/ACM Transactions on Audio, Speech and Language Processing 2023

3- b) Datasets

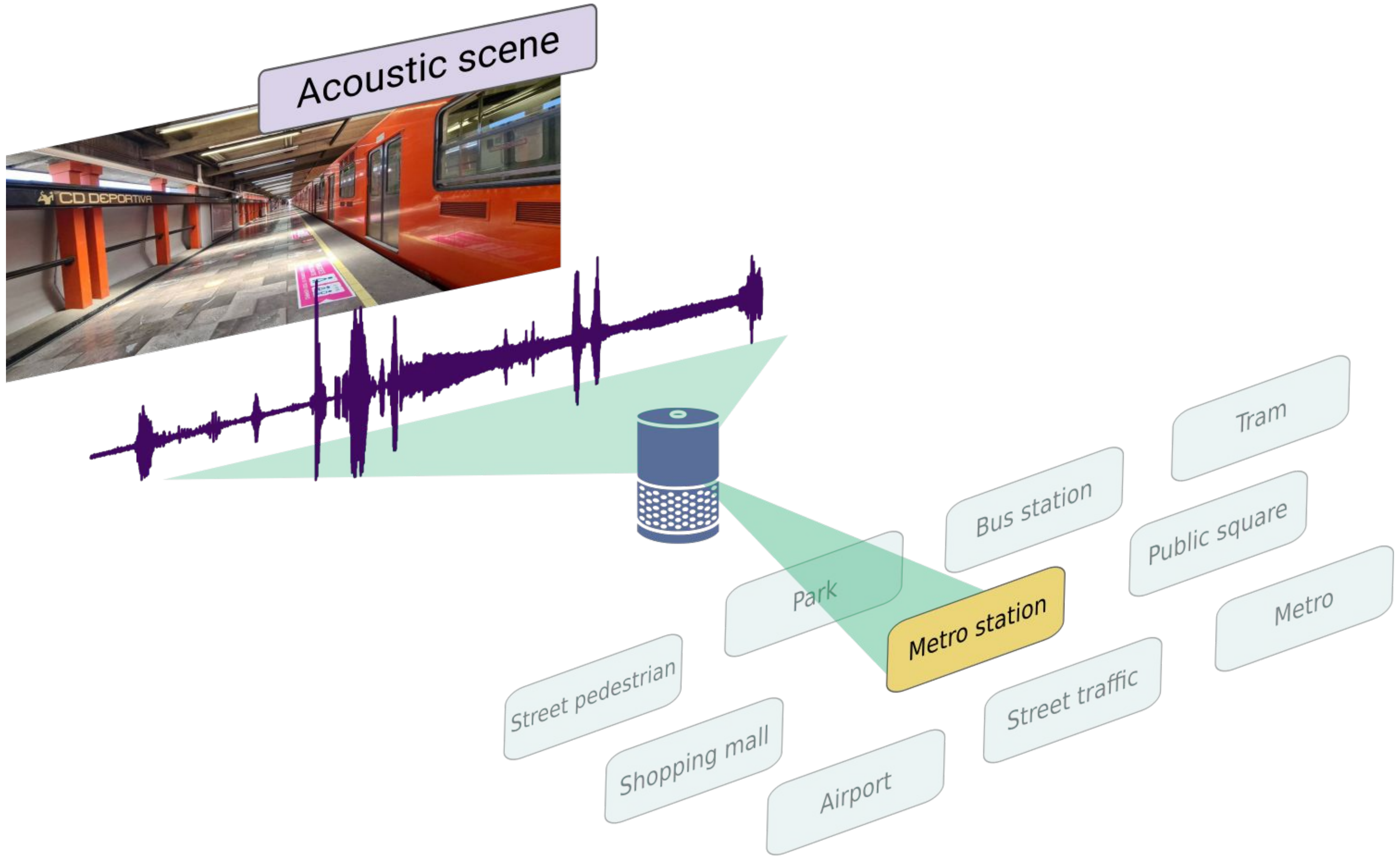
- OpenMIC [4] :
    - Instrument Classification
    - 20 classes, multi-label
    - 20,000 excerpt of 10s



[4] Openmic-2018: An open data-set for multiple instrument recognition, *Humphrey et al.* ISMIR 2018

3- b) Datasets

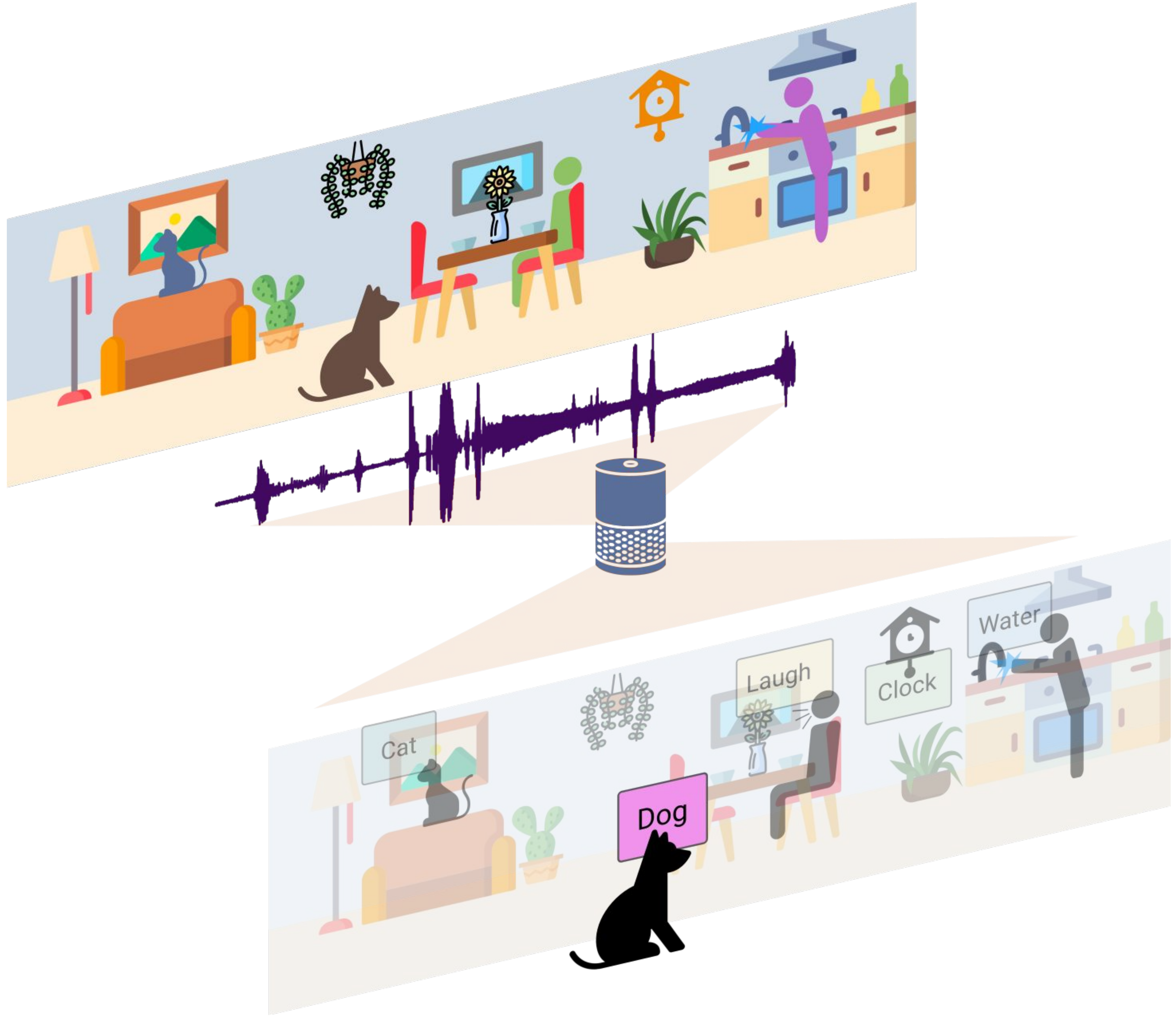
- TAU Urban Acoustic Scenes [5] :
  - Scene classification
  - 10 classes, multi-class
  - 23,040 excerpt of 10s



[5] A multi-device dataset for urban acoustic scene classification, *Mesaros et al.* DCASE 2018

3- b) Datasets

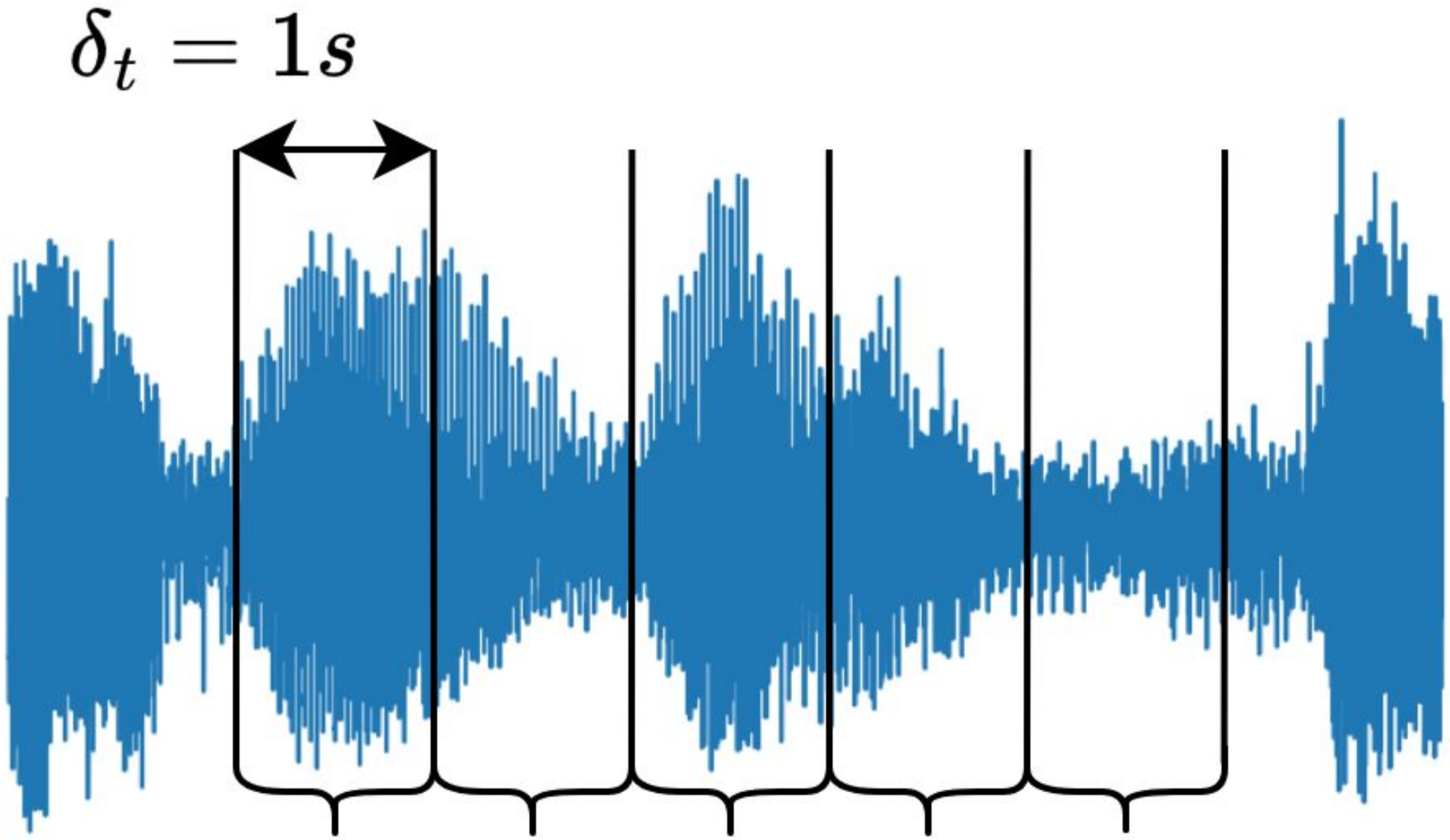
- ESC-50 [6] :
  - Event classification
  - 50 classes, multi-class
  - 2,000 excerpt of 5s



[6] ESC: dataset for environmental sound classification, *Piczak et al.* ACM 2018

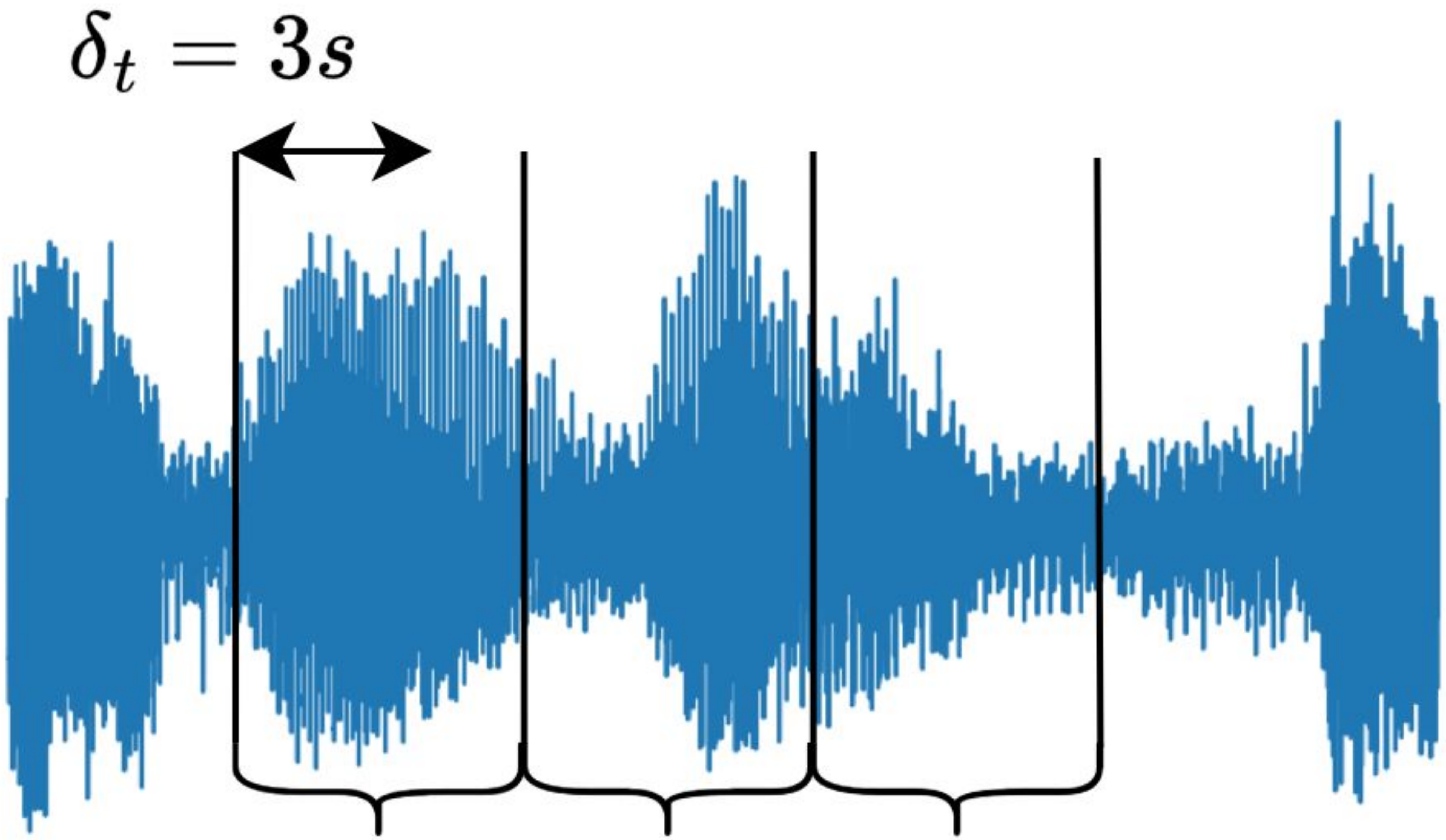
3- c) Parameters

- δ_t :



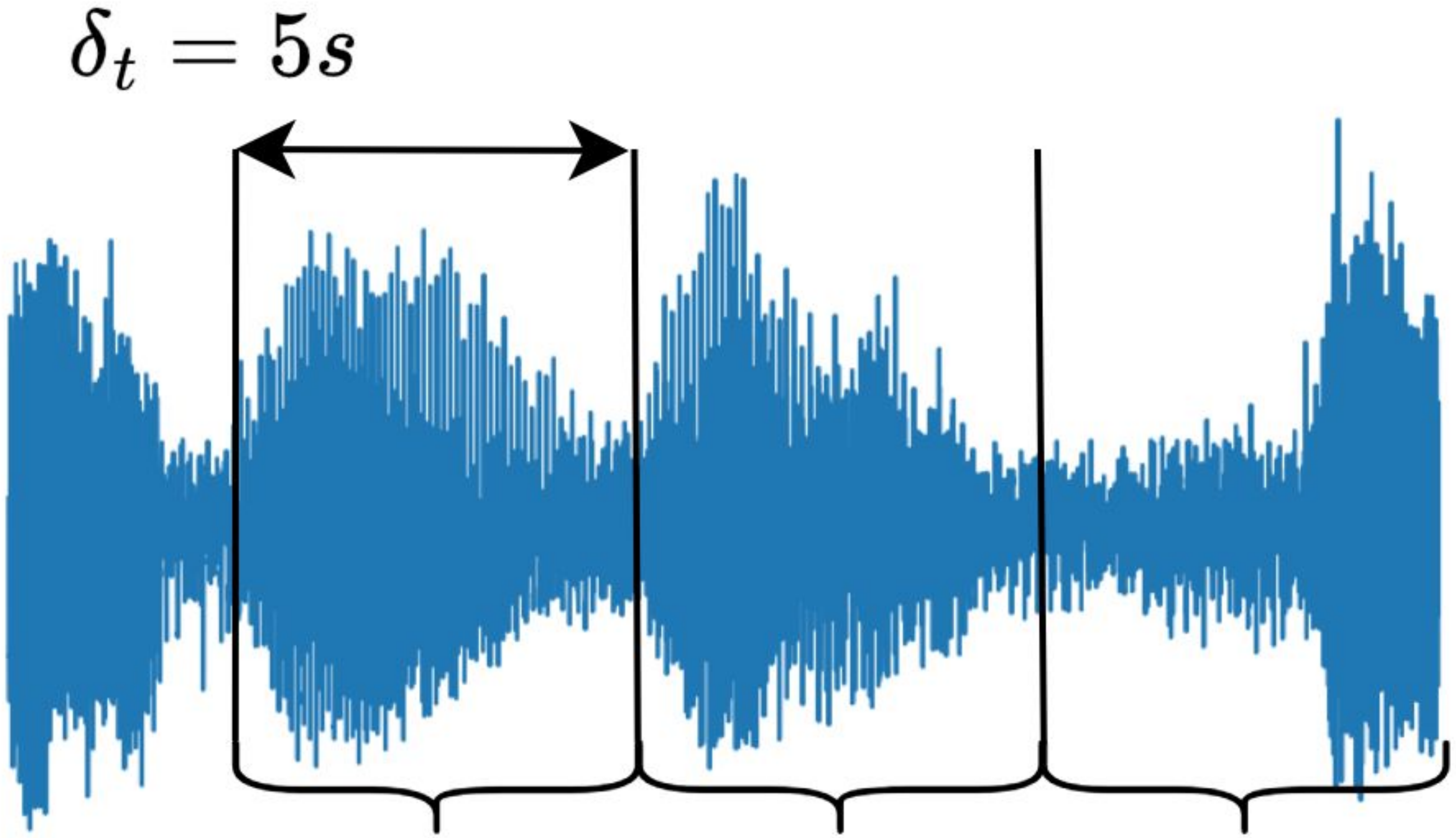
3- c) Parameters

- δ_t :



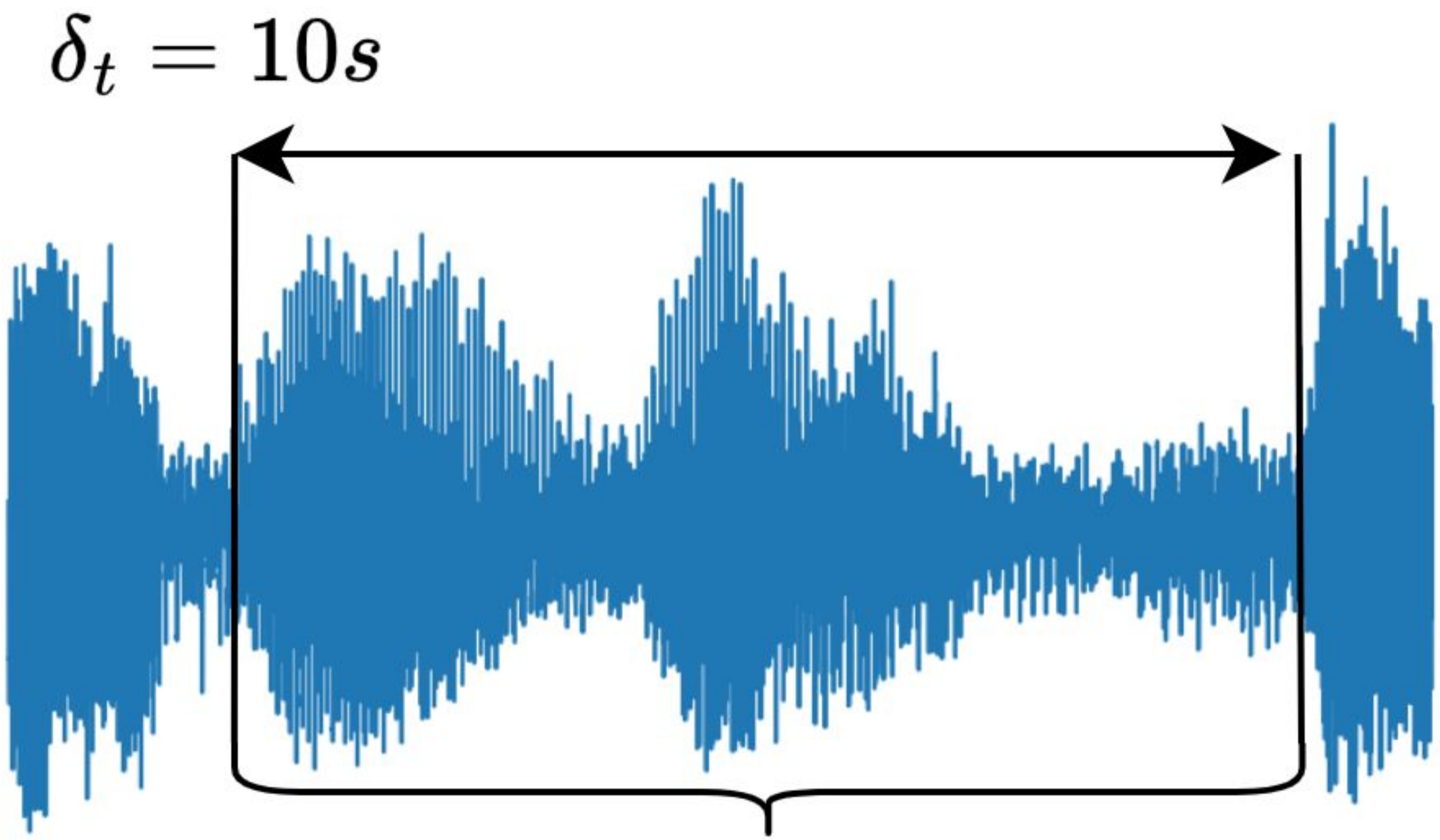
3- c) Parameters

- δ_t :



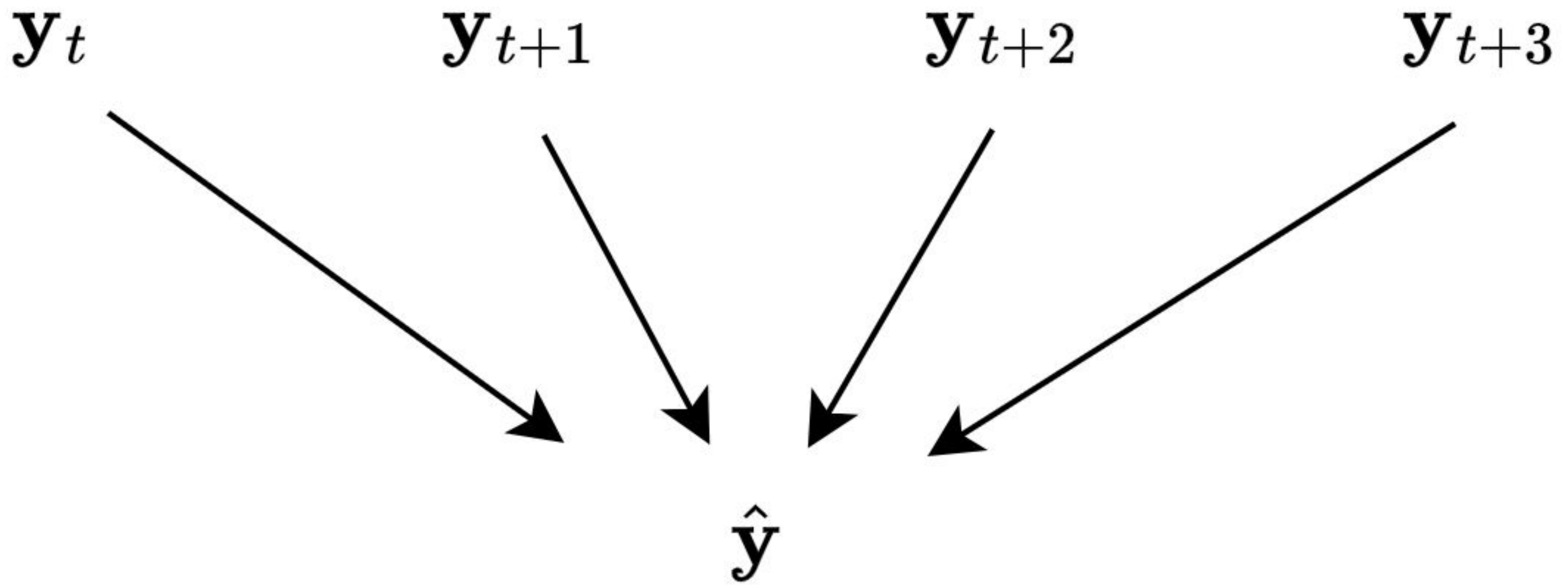
3- c) Parameters

- δ_t :



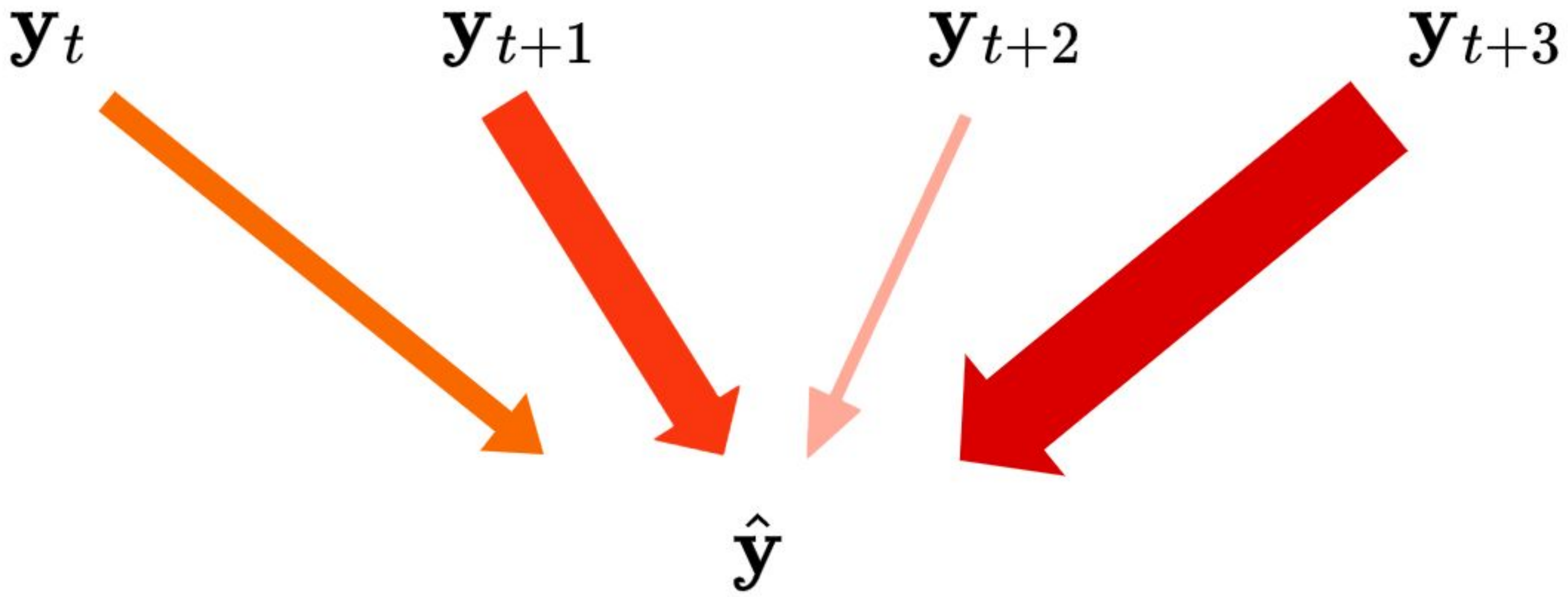
3- c) Parameters

- $\mu_m(\cdot)$ mean pooling: $$\hat{\mathbf{y}} = \frac{1}{N}\sum_{t=0}^{N}\mathbf{y_t}$$

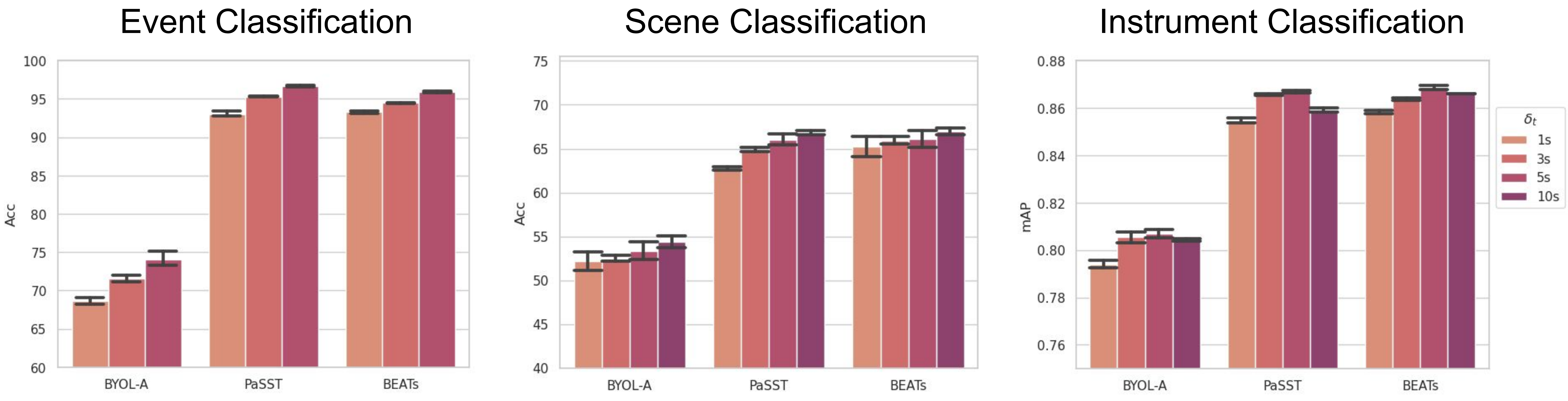


3- c) Parameters

- $\mu_m(\cdot)$ attention pooling: $$\hat{\mathbf{y}} = \sum_{t=0}^{N} w_{\mathbf{y_t}} \mathbf{y_t} \qquad w_{\mathbf{y_t}} = \frac{\sigma(v^\top \mathbf{y_t})}{\sum_{t'=0}^{N} \sigma(v^\top \mathbf{y_{t'}})}$$



4- Results



- A longer δ_t does does not always result in a better score
- Best performances are not necessarily achieved for the δ_t used to train the model.

4- Results

- For Instrument Classification task we reach the SOTA for δ_t of 3s and 5s without fine-tuning

5. Conclusion

31

5- Conclusion

- Using the longest δ_t does not always imply better performances, even if the models were trained with longer audio

- The choice of the optimal δ_t for the best score depends on the model and the dataset

- A smaller δ_t reduces the memory and computational cost of the Transformer models at inference time.

Model	δ_t	OpenMIC		TAU Urban		ESC-50		Emb. Size	#Param. $f(\cdot)$
		$\mu_{\mathrm{m}}(\cdot)$	$\mu_{\mathrm{a}}(\cdot)$	$\mu_{\mathrm{m}}(\cdot)$	$\mu_{\mathrm{a}}(\cdot)$	$\mu_{\mathrm{m}}(\cdot)$	$\mu_{\mathrm{a}}(\cdot)$		
BYOL-A v2	1	0.792 ± 0.001	0.797 ± 0.003	52.5 ± 1.4	50.6 ± 1.7	69.1 ± 1.4	68.7 ± 1.1	3072	6.3M
PaSST	1	0.851 ± 0.001	0.860 ± 0.002	63.3 ± 0.4	62.0 ± 0.5	93.1 ± 0.2	93.0 ± 0.4	768	87M
BEATs	1	0.852 ± 0.001	0.865 ± 0.001	$\mathbf{67.5 \pm 0.2}$	61.0 ± 4.3	93.2 ± 0.1	93.4 ± 0.4	$48 \cdot 768$	90M
BYOL-A v2	3	0.805 ± 0.001	0.804 ± 0.005	53.9 ± 0.9	52.3 ± 0.9	71.2 ± 1.1	72.6 ± 1.0	3072	6.3M
PaSST	3	0.866 ± 0.001	0.865 ± 0.000	65.0 ± 0.4	64.5 ± 0.5	95.7 ± 0.1	95.0 ± 0.1	768	87M
BEATs	3	0.862 ± 0.000	0.866 ± 0.002	66.8 ± 0.2	64.9 ± 1.4	95.4 ± 0.1	93.4 ± 0.3	$144 \cdot 768$	90M
BYOL-A v2	5	0.806 ± 0.002	0.808 ± 0.003	53.8 ± 1.1	53.6 ± 0.9	72.8 ± 1.8	74.0 ± 1.1	3072	6.3M
PaSST	5	0.866 ± 0.001	0.868 ± 0.001	66.5 ± 0.5	65.9 ± 1.0	$\mathbf{96.8 \pm 0.2}$	96.6 ± 0.2	768	87M
BEATs	5	$\mathbf{0.869 \pm 0.002}$	$\mathbf{0.869 \pm 0.001}$	$\mathbf{67.5 \pm 0.2}$	65.4 ± 2.6	96.1 ± 0.0	95.7 ± 0.3	$248 \cdot 768$	90M
BYOL-A v2	10	0.803 ± 0.001	0.805 ± 0.002	52.4 ± 1.5	54.7 ± 0.8	-	-	3072	6.3M
PaSST	10	0.861 ± 0.001	0.857 ± 0.001	66.7 ± 0.5	66.9 ± 0.4	-	-	768	87M
BEATs	10	0.866 ± 0.000	0.867 ± 0.000	$\mathbf{67.5 \pm 0.3}$	67.2 ± 1.1	-	-	$496 \cdot 768$	90M
Results from papers									
ResAtt [23]	10	0.860		-		-		2048	-
PaSST-S [8]	10/5	0.843		75.6		96.8		768	87M
BEATs iter3+[10]	5	-		-		98.1		$248 \cdot 768$	90M