ICML | 2019
Thirty-sixth International Conference on Machine Learning

Weakly-Supervised Temporal Localization via Occurrence Count Learning

Julien Schroeter
schroeterj1@cardiff.ac.uk

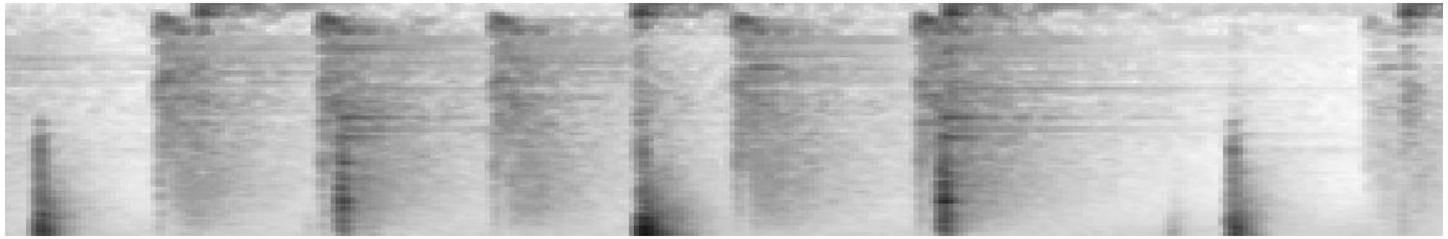
Dr Kirill Sidorov **Prof David Marshall**

CONTEXT
Temporal Localization

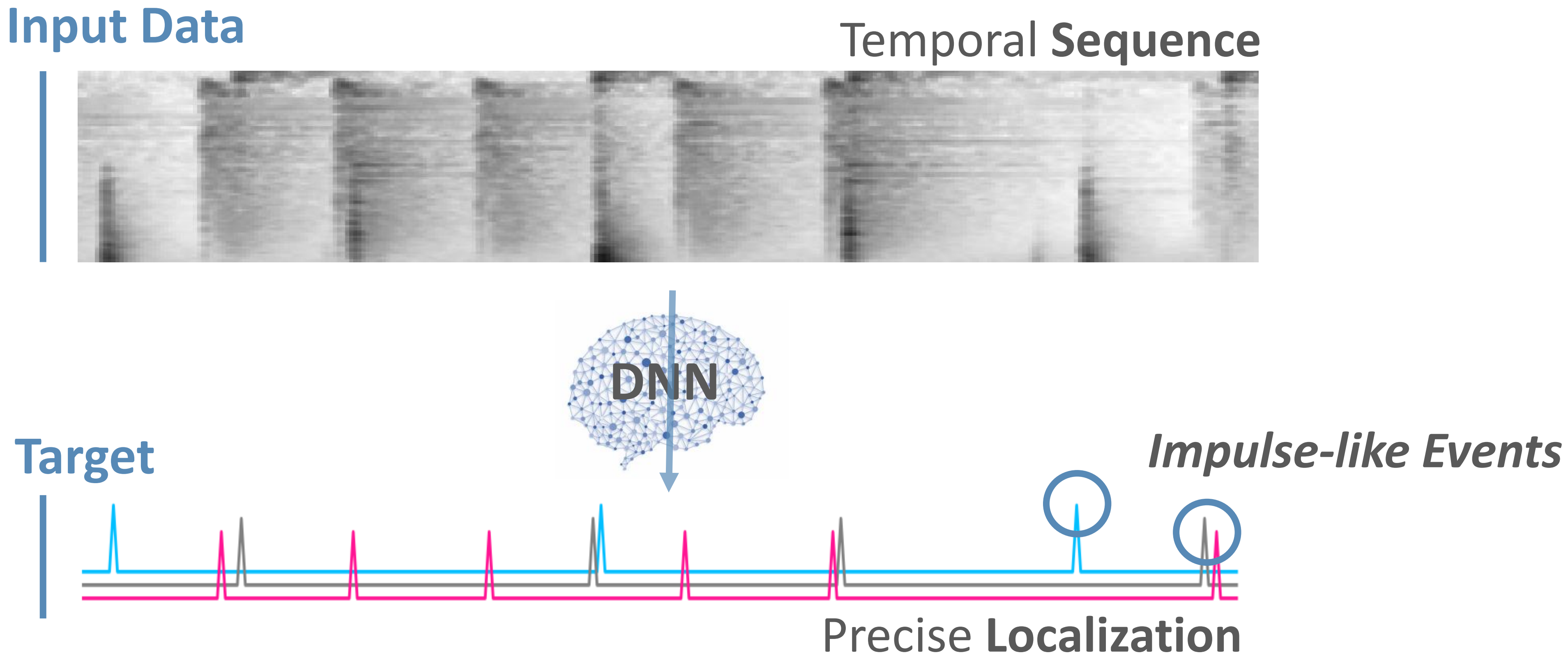
CONTEXT
Temporal Localization

Input Data

Temporal **Sequence**



CONTEXT
Temporal Localization

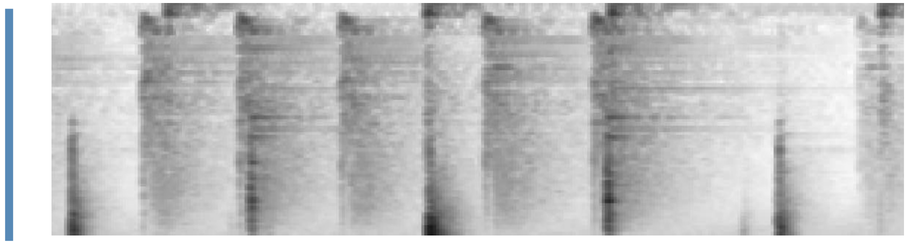


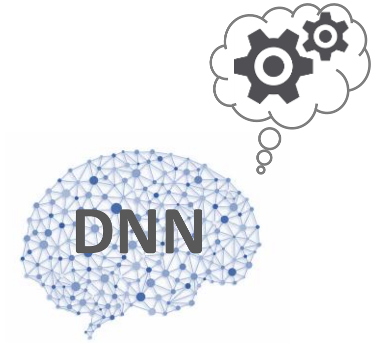
CONTEXT
Temporal Localization

Fully-Supervised

CONTEXT
Temporal Localization

Input Data

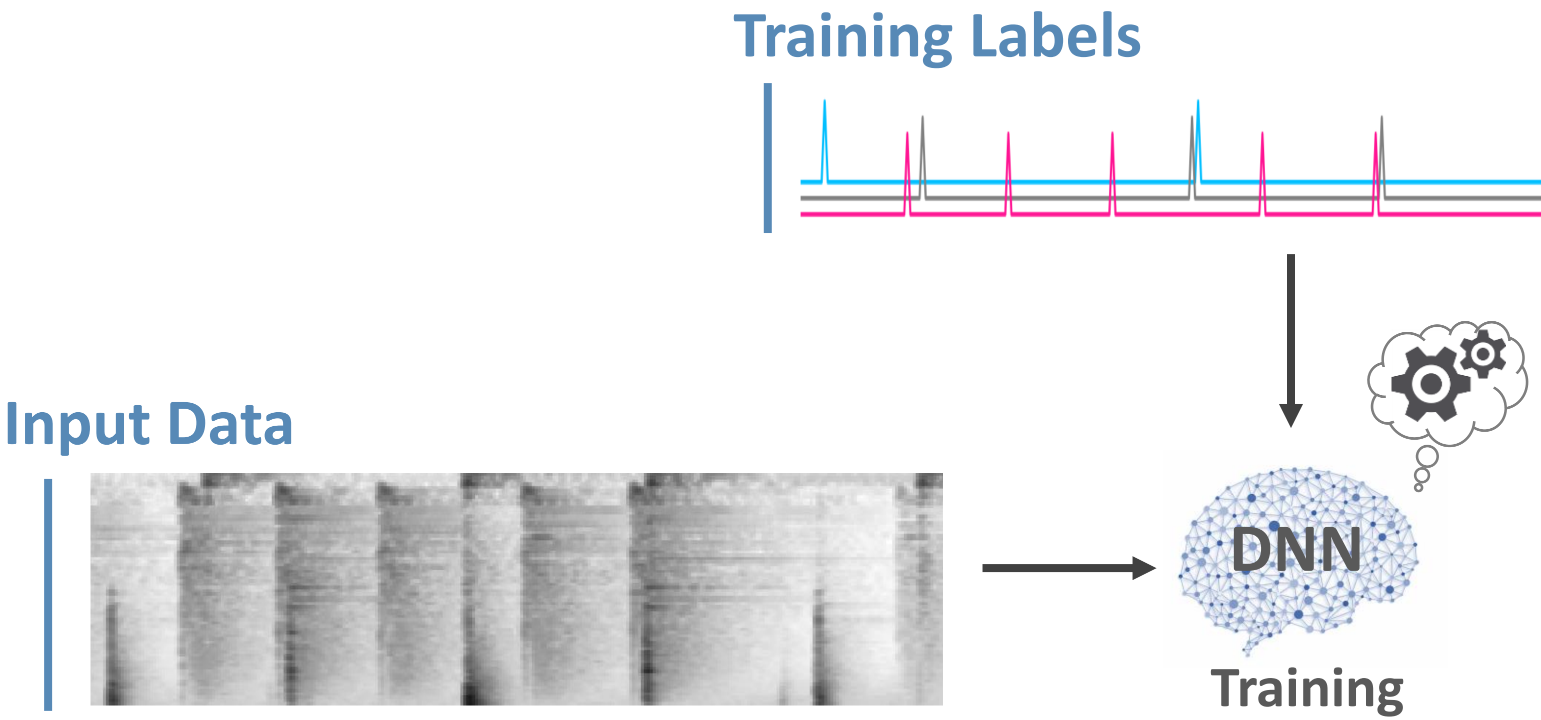




Training

Fully-Supervised

CONTEXT
Temporal Localization



Fully-Supervised

OBJECTIVE

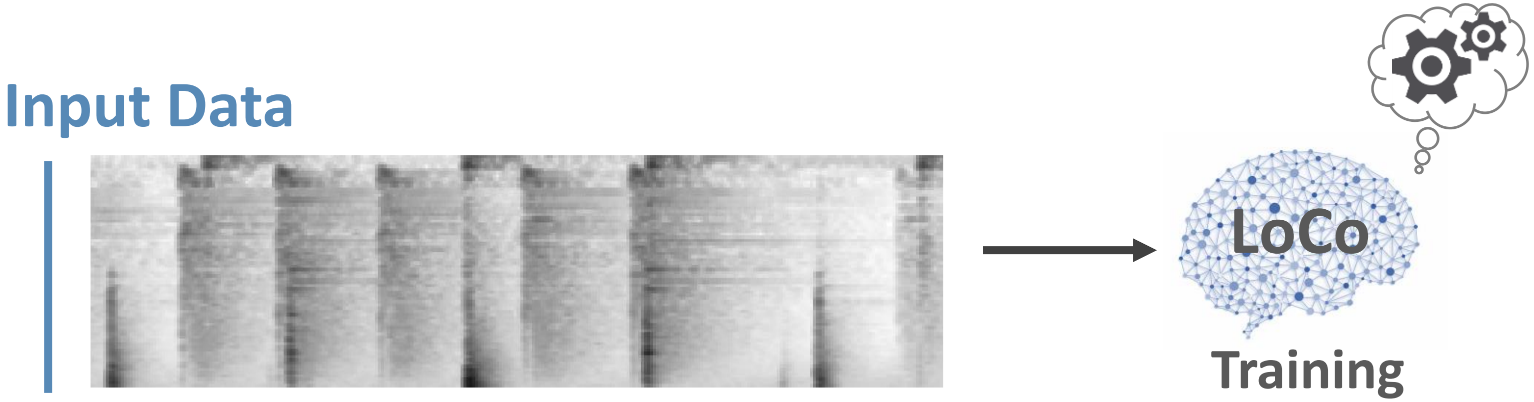
Weakening the annotation requirement

Training

Our approach

OBJECTIVE

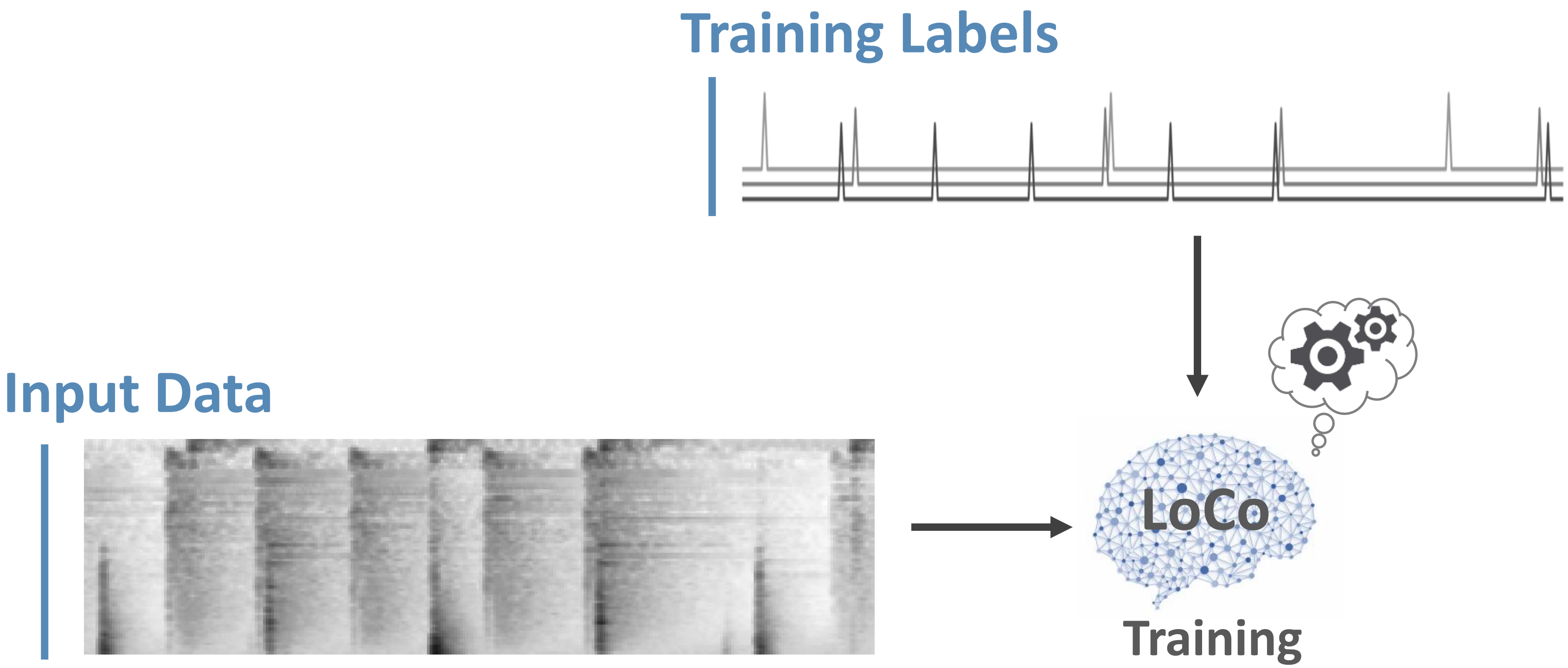
Weakening the annotation requirement



Our approach

OBJECTIVE

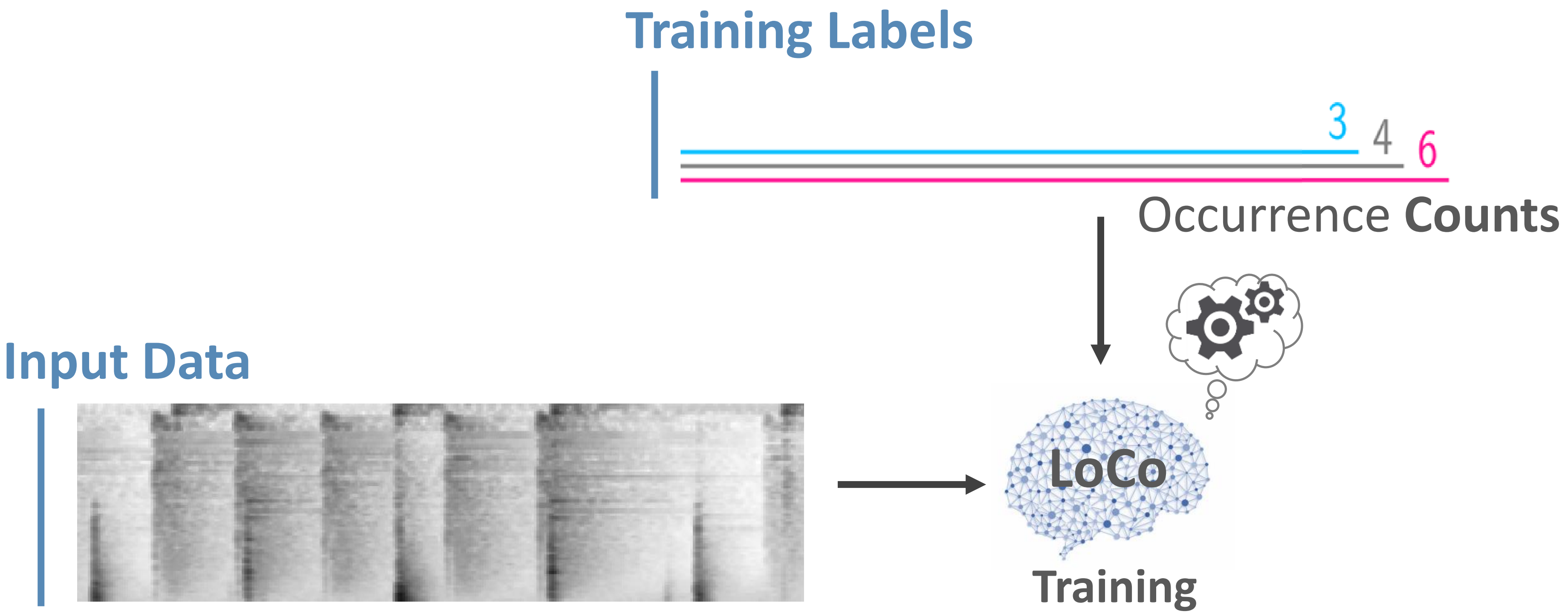
Weakening the annotation requirement



Our approach

OBJECTIVE

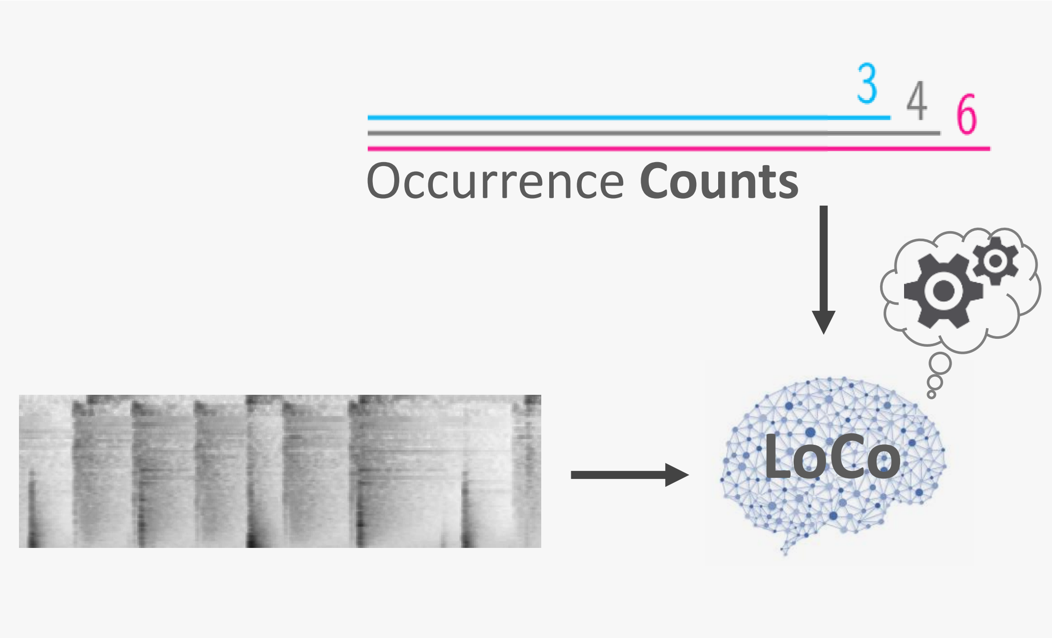
Weakening the annotation requirement



Our approach

OBJECTIVE

Weakening the annotation requirement

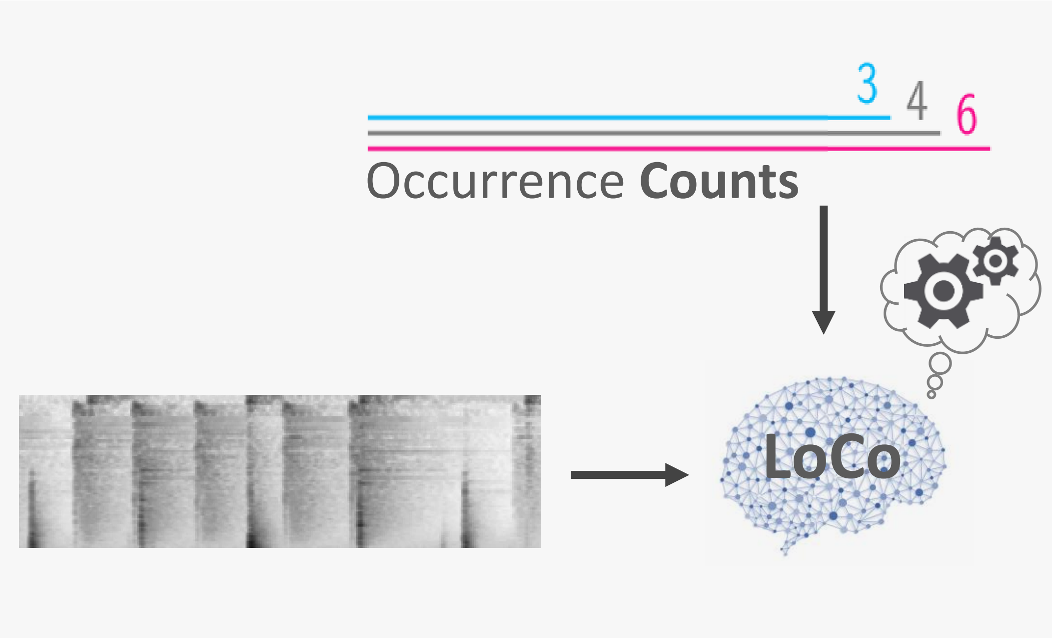


Training

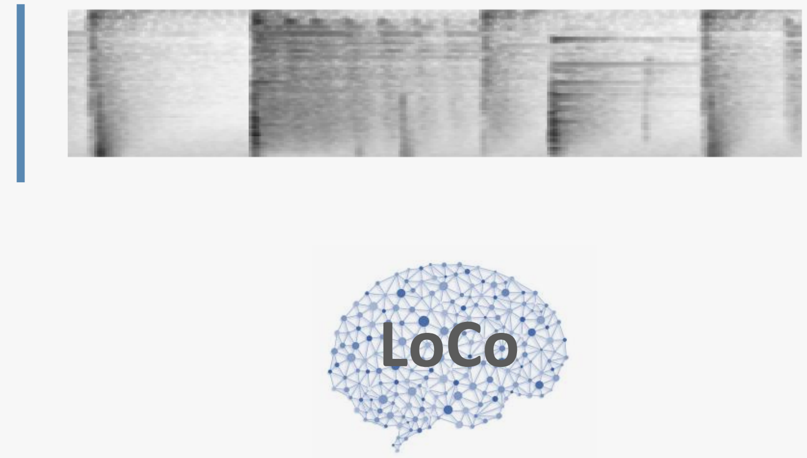
Inference

OBJECTIVE

Weakening the annotation requirement



Input

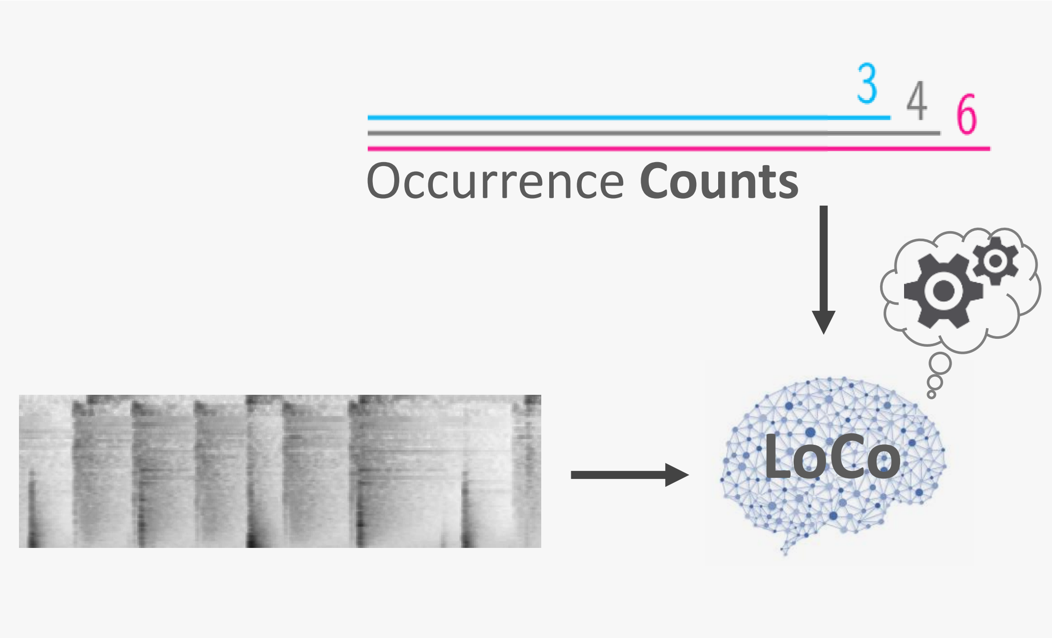


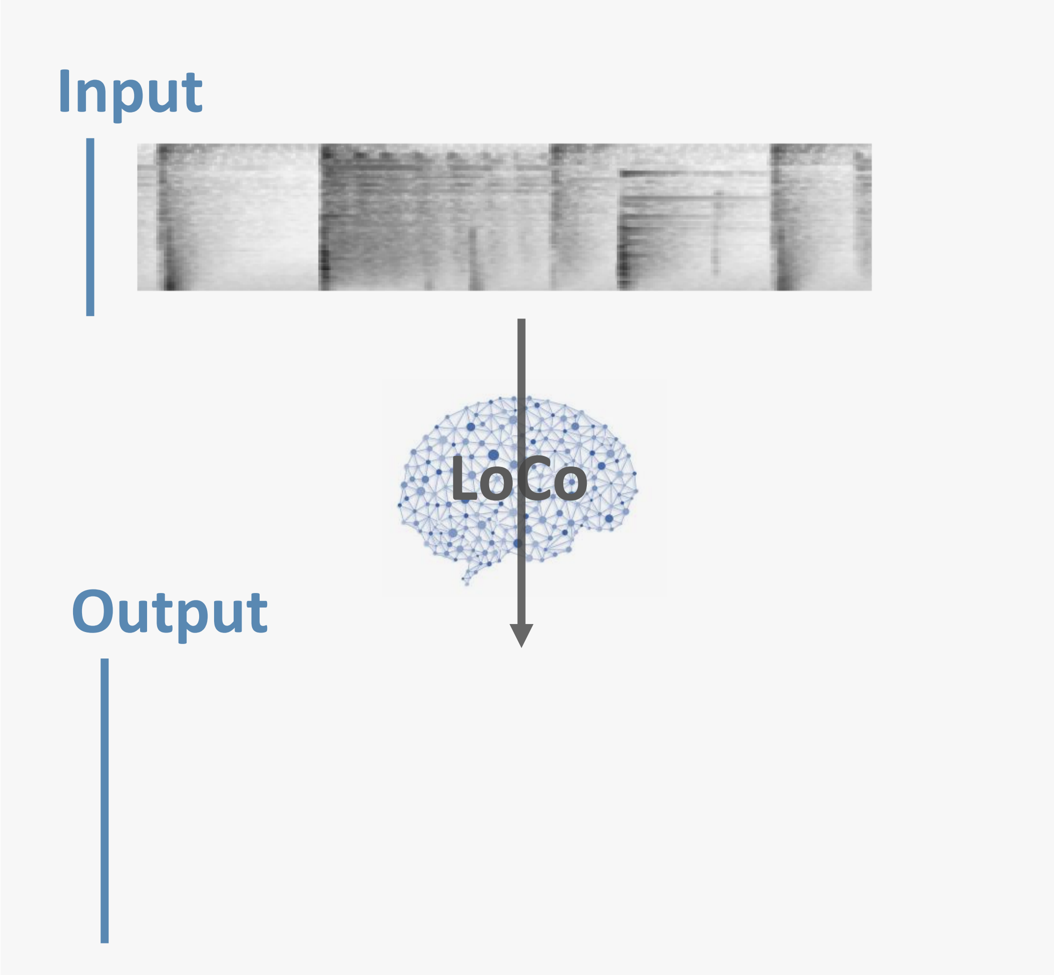
Training

Inference

OBJECTIVE

Weakening the annotation requirement

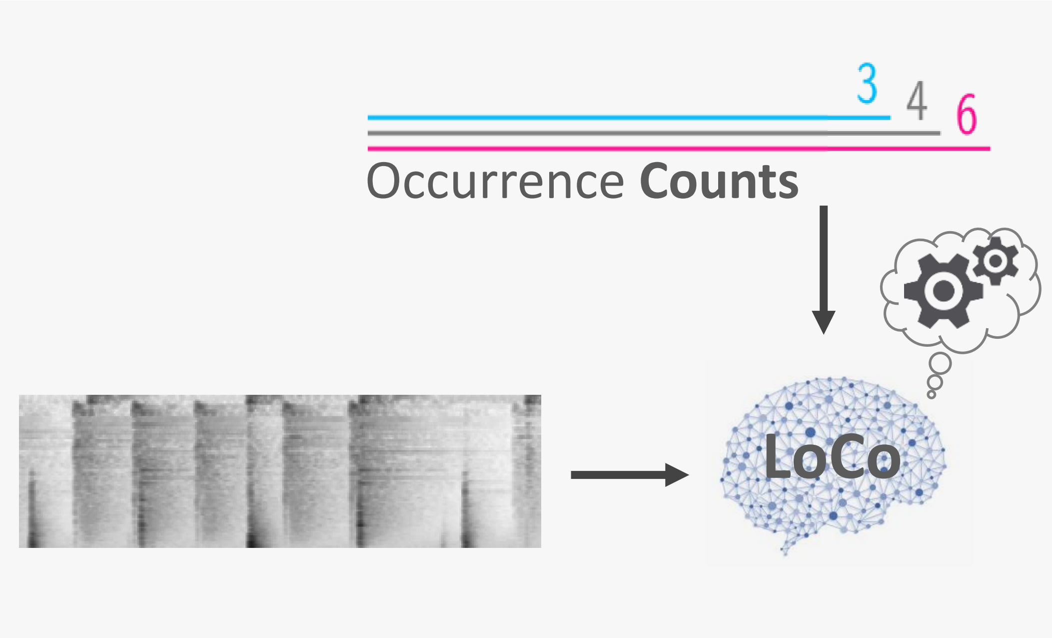




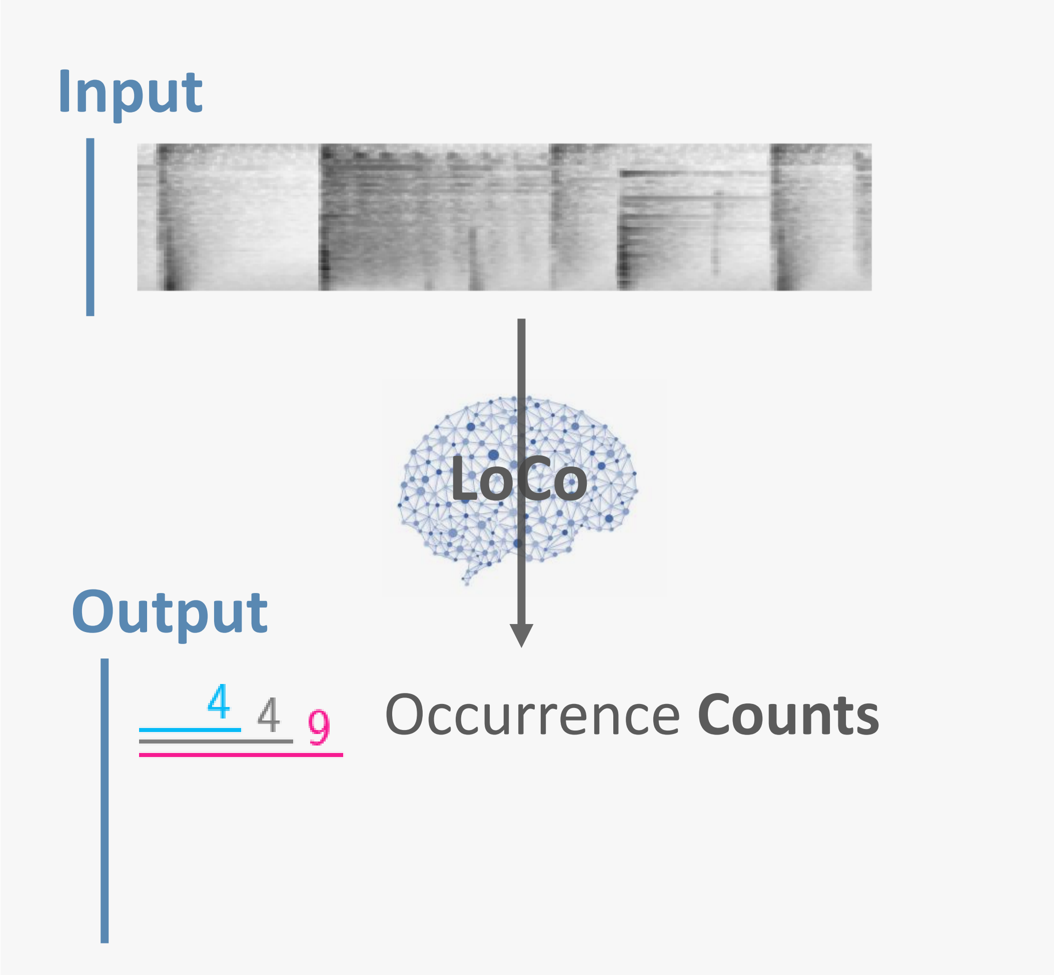
Training

Inference

Objective
Weakening the annotation requirement



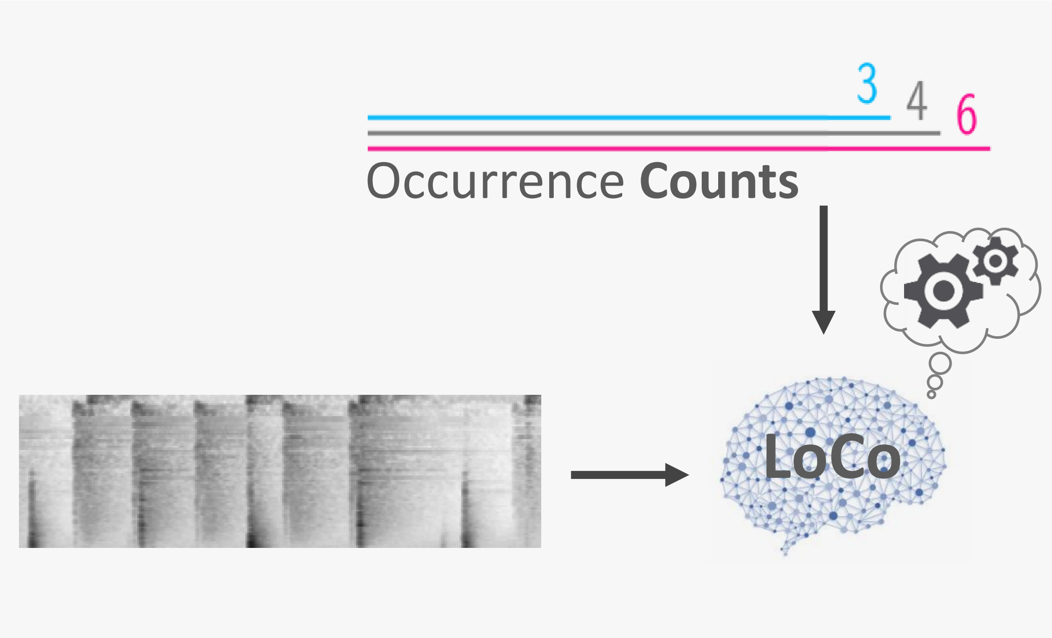
Training

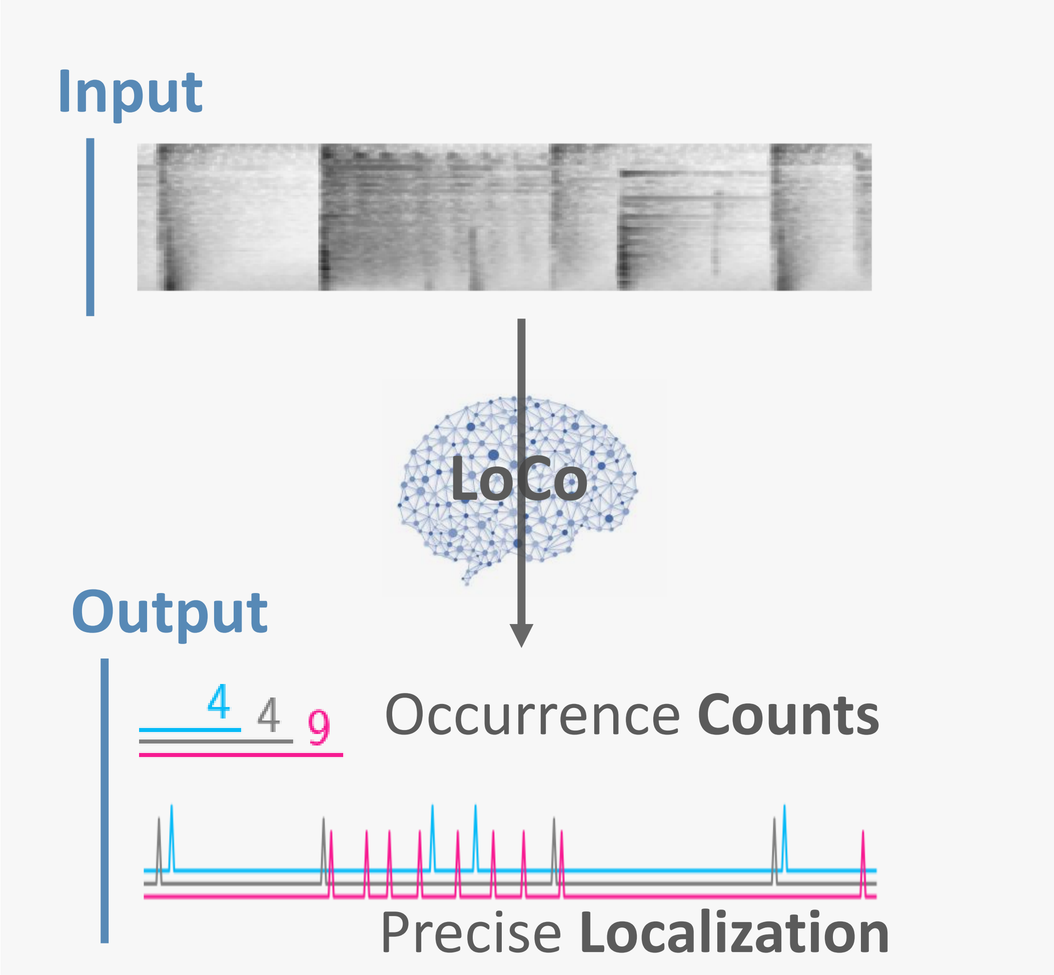


Inference

OBJECTIVE

Weakening the annotation requirement



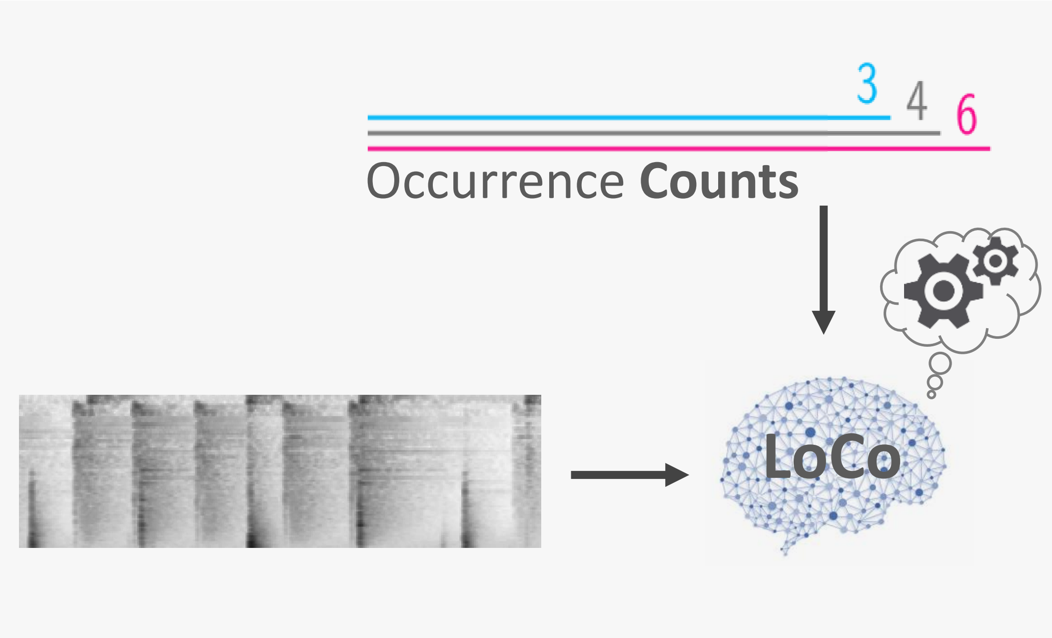


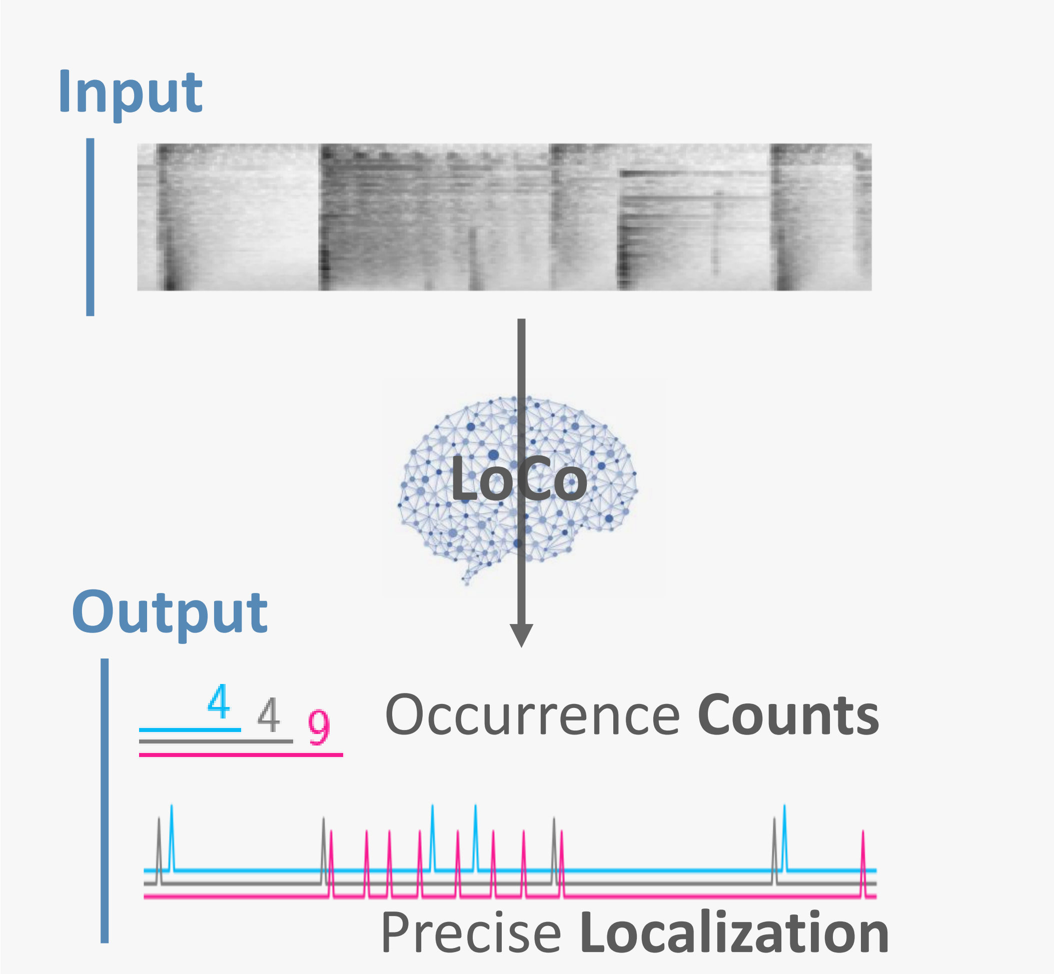
Training

Inference

Objective

Weakening the annotation requirement





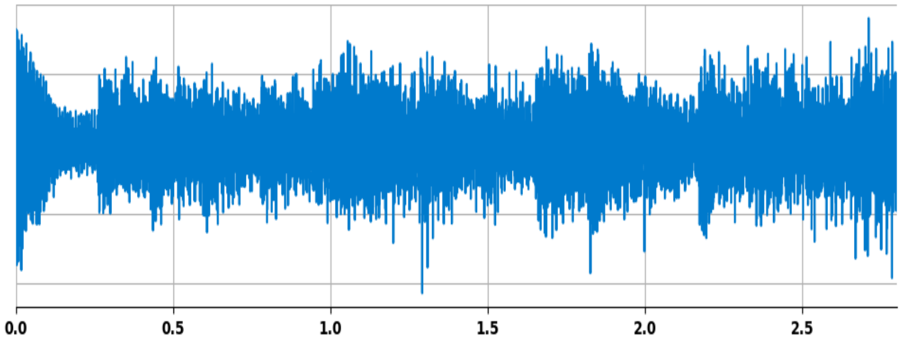
Training — Inference

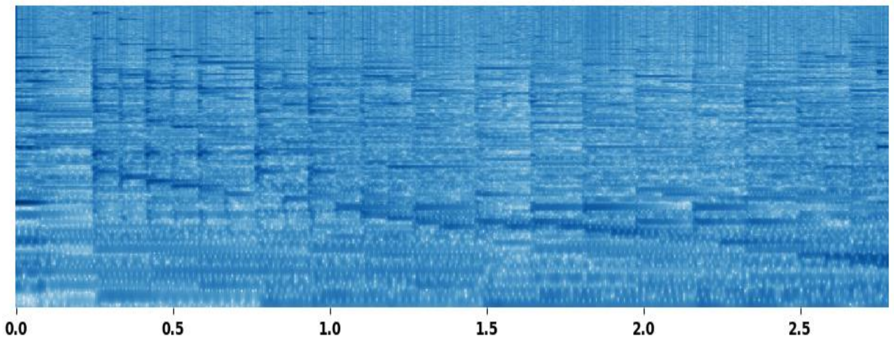
Weakly-Supervised

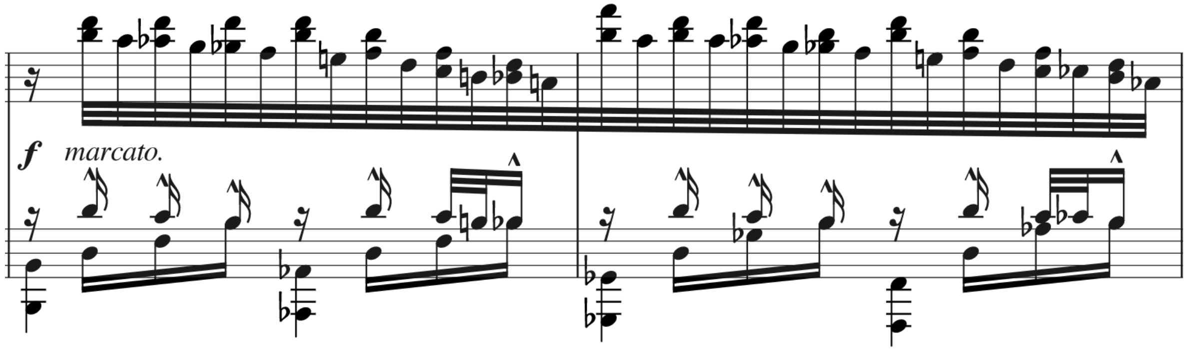
Is it useful?

Objective

Weakening the annotation requirement



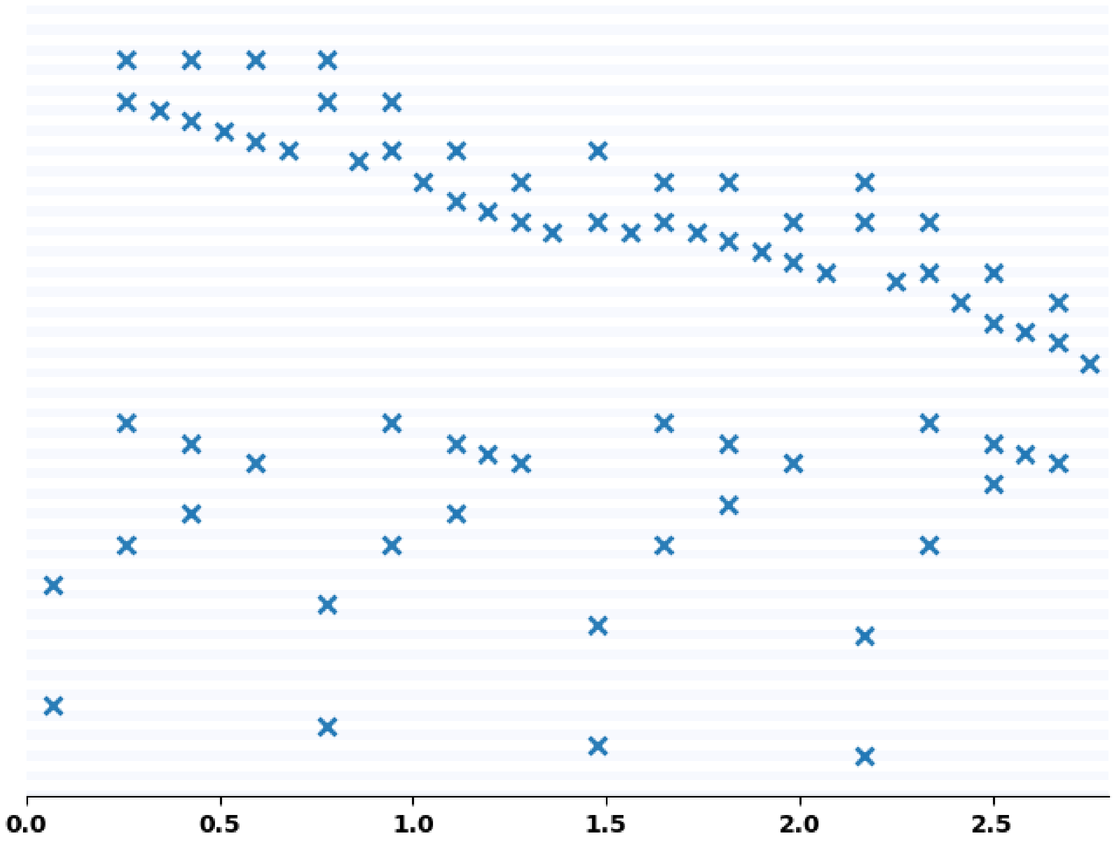




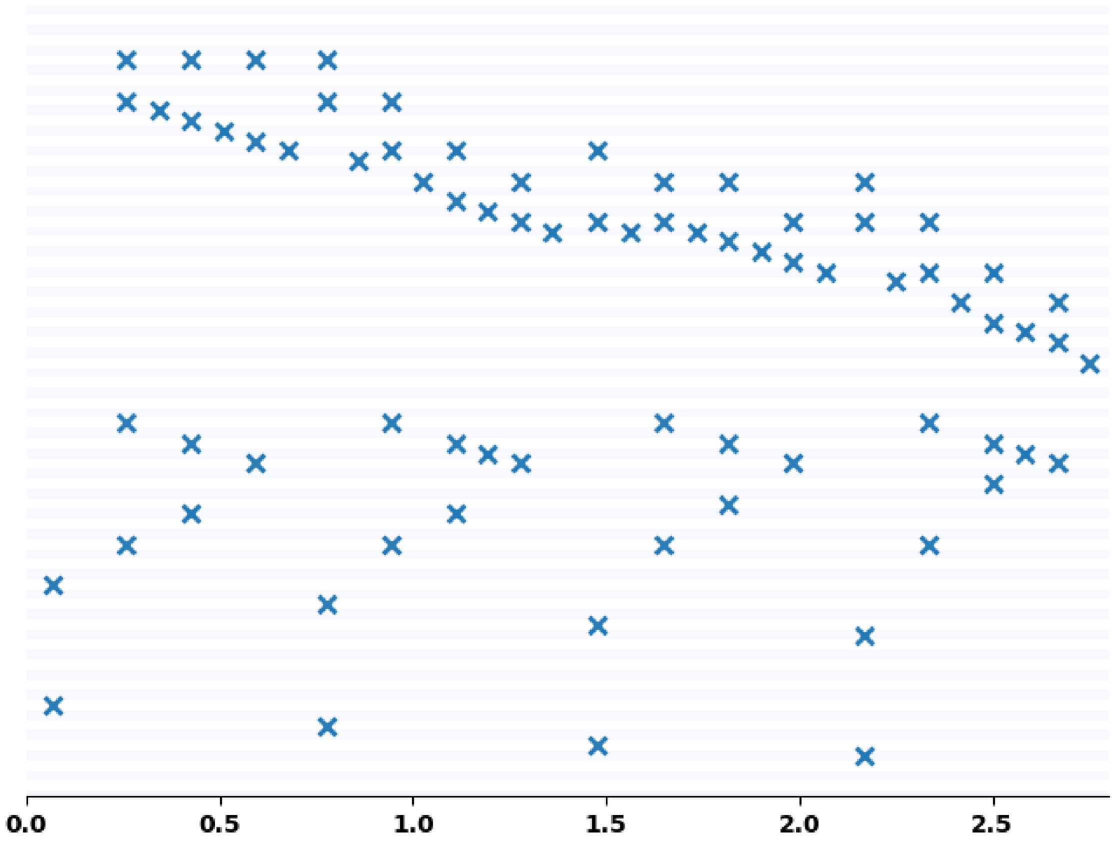
Label Piano Music

Objective

Weakening the annotation requirement



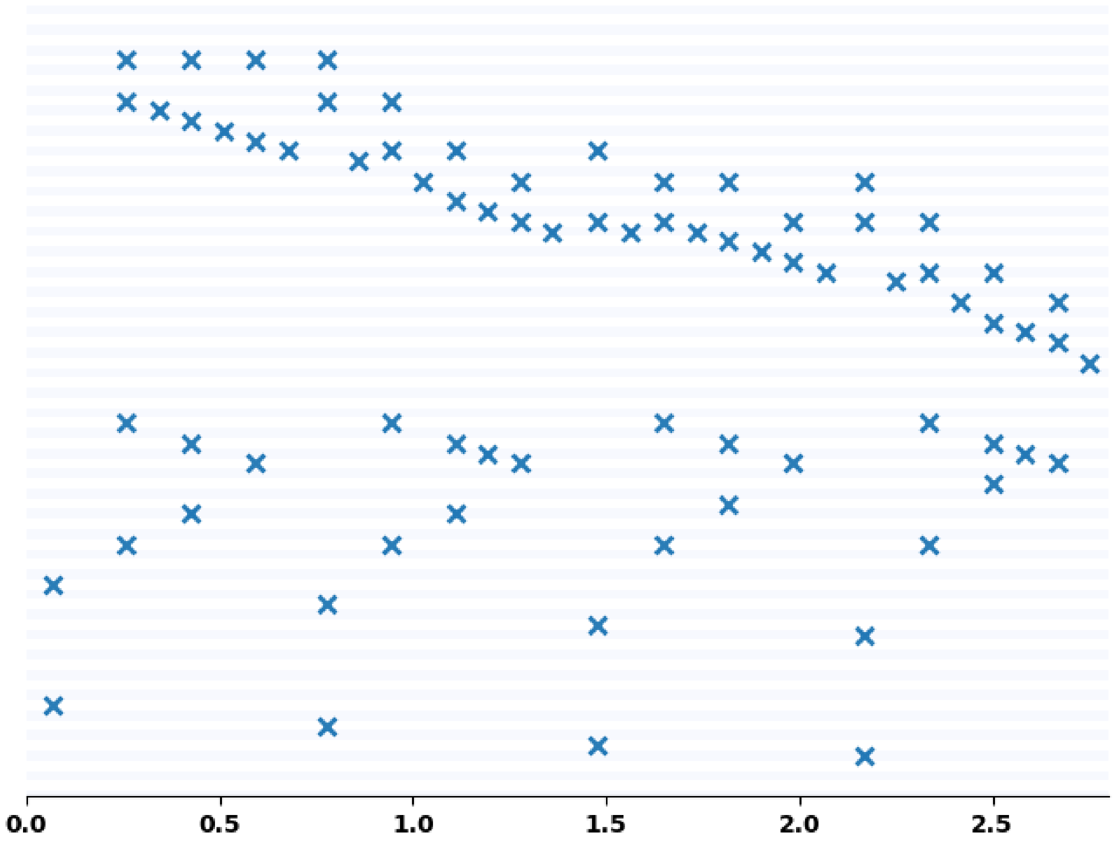
OBJECTIVE
Weakening the annotation requirement



✖ Precise hand-labeling is very **tedious**

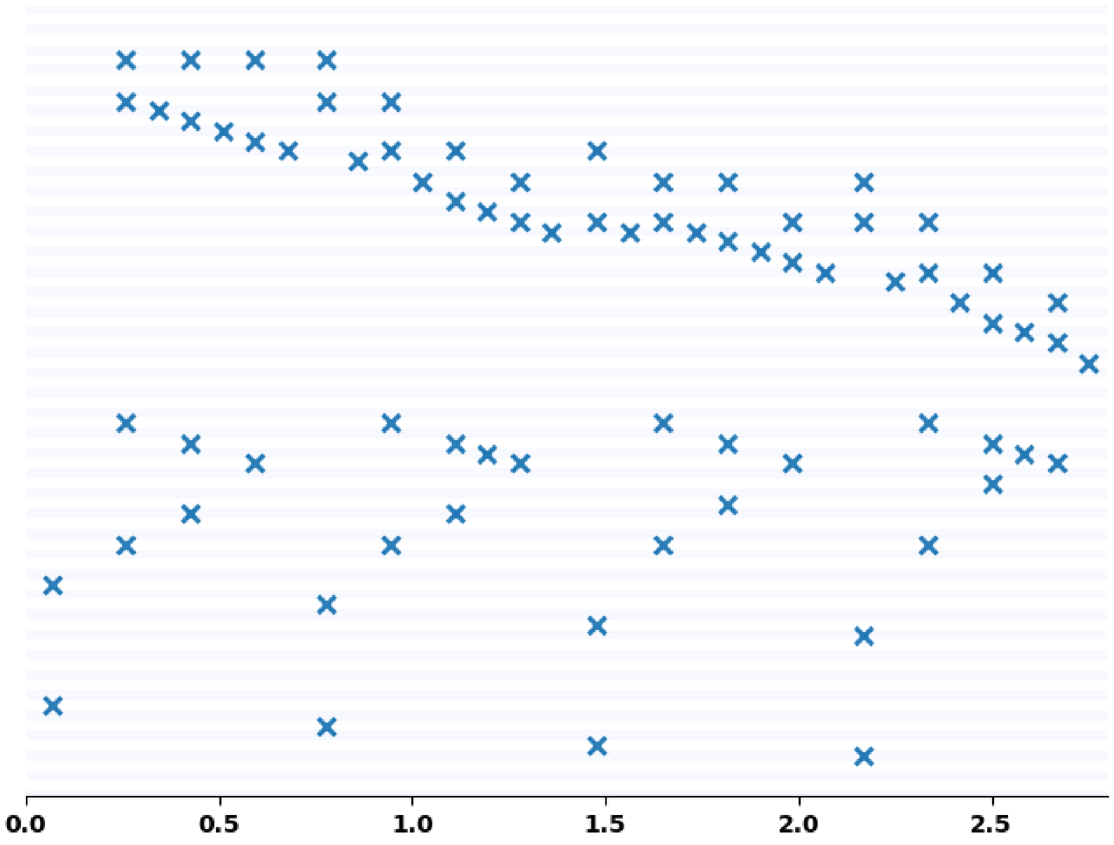
OBJECTIVE

Weakening the annotation requirement



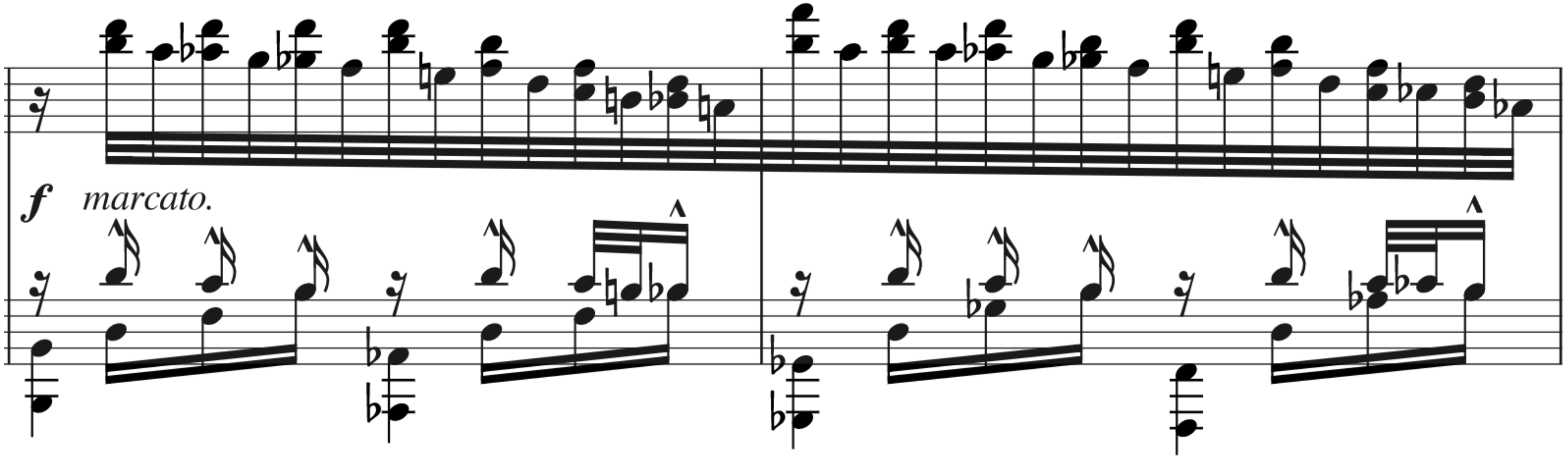
- ✖ Precise hand-labeling is very **tedious**
- ✖ Prone to labeling **inaccuracy**

OBJECTIVE
Weakening the annotation requirement



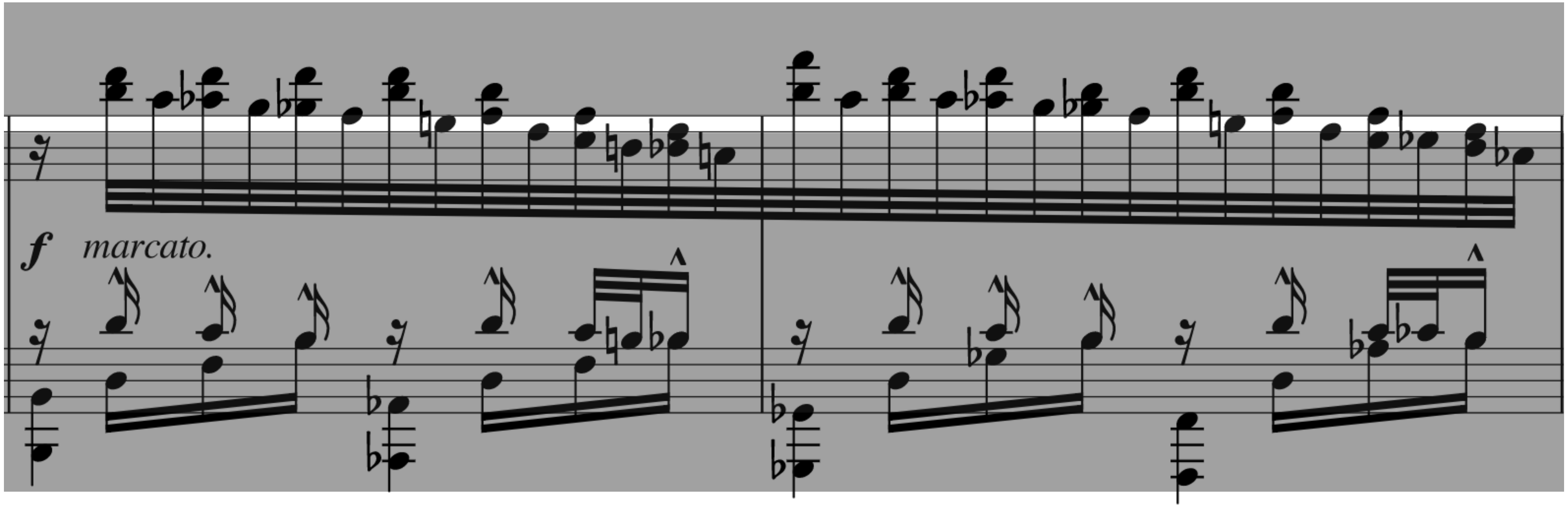
Proposed Approach

Objective
Weakening the annotation requirement



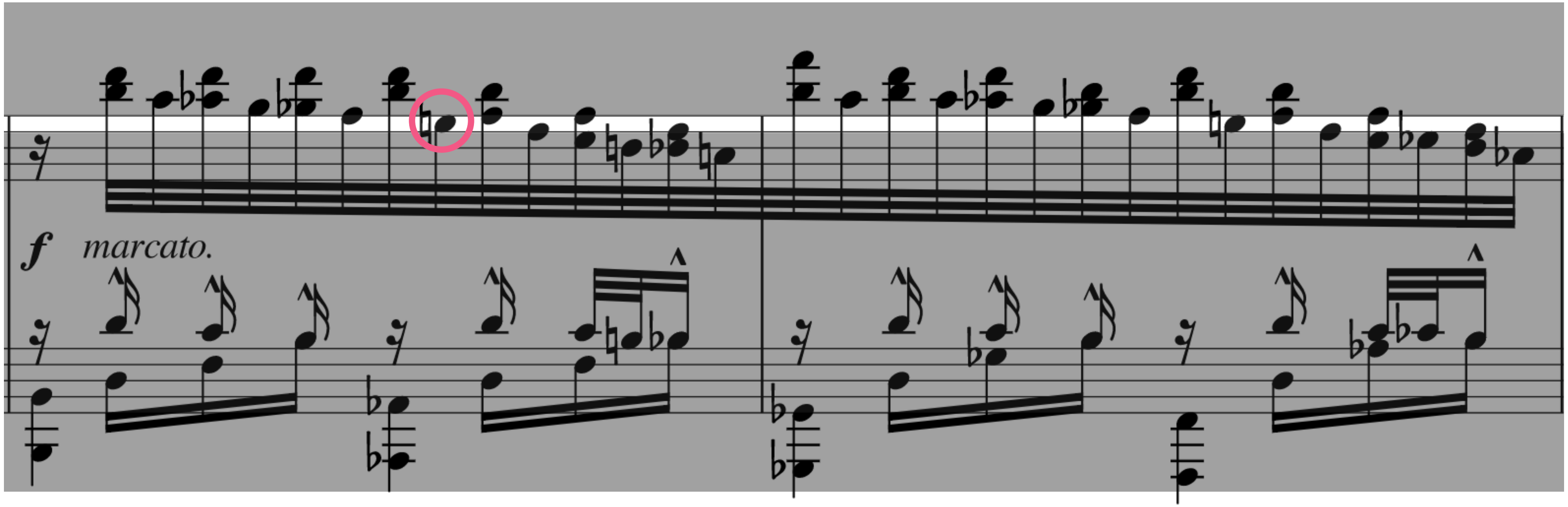
How many notes per pitch?

Objective
Weakening the annotation requirement



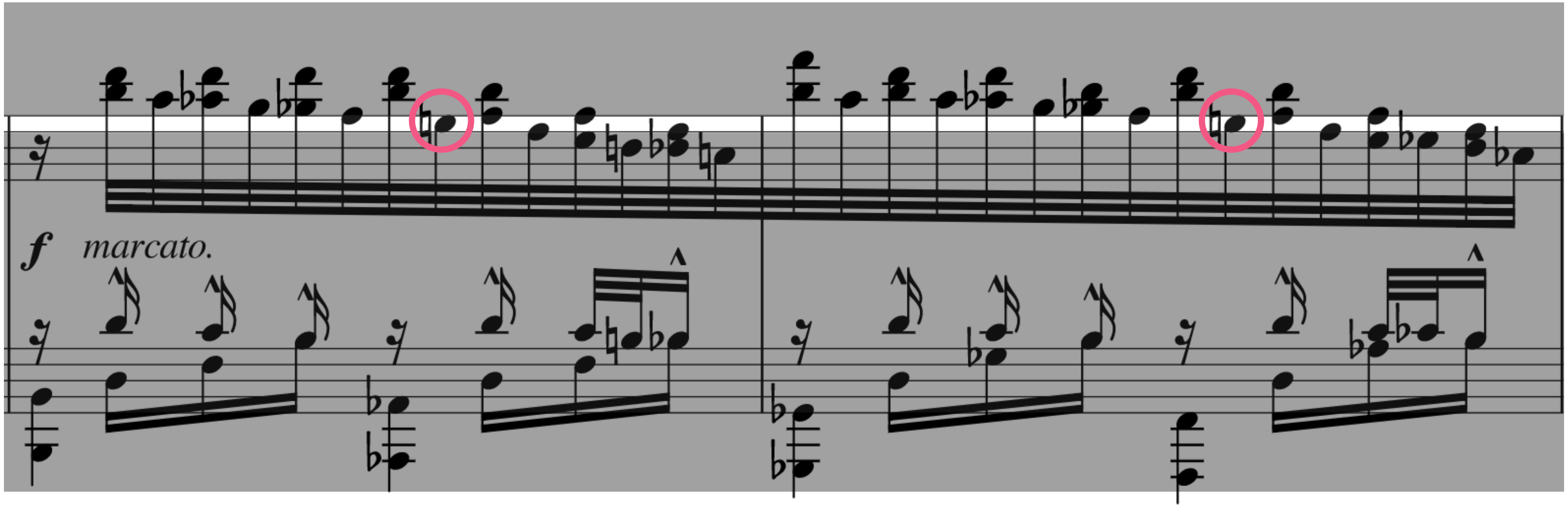
How many notes per pitch?

Objective
Weakening the annotation requirement



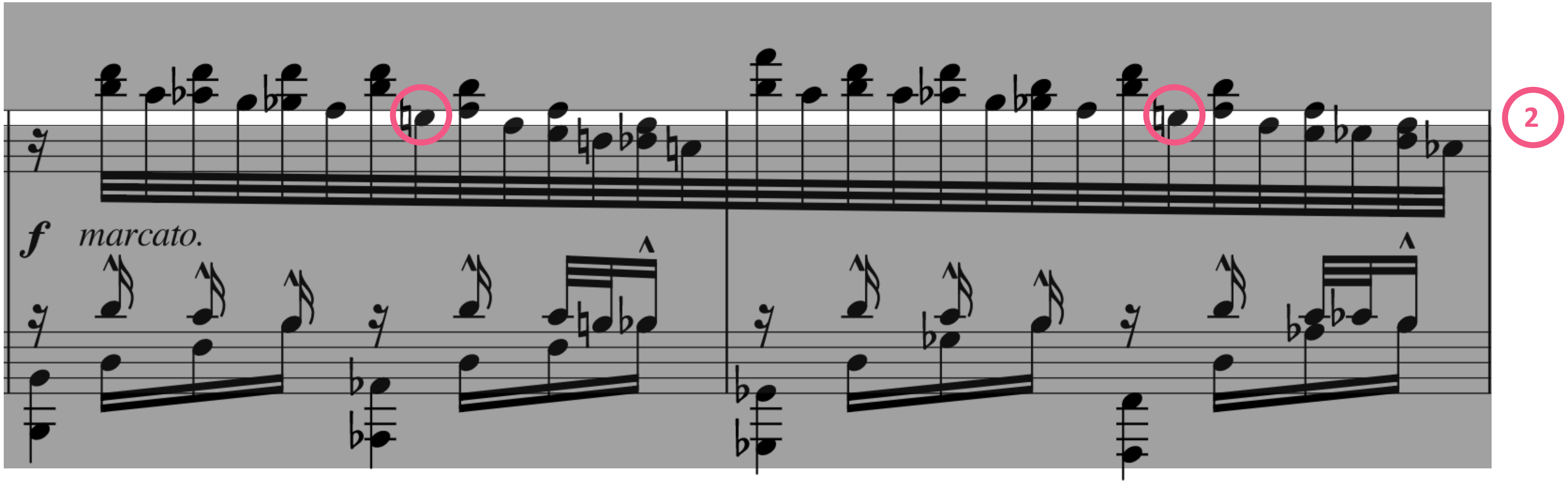
How many notes per pitch?

OBJECTIVE
Weakening the annotation requirement



How many notes per pitch?

OBJECTIVE
Weakening the annotation requirement



How many notes per pitch?

The Model

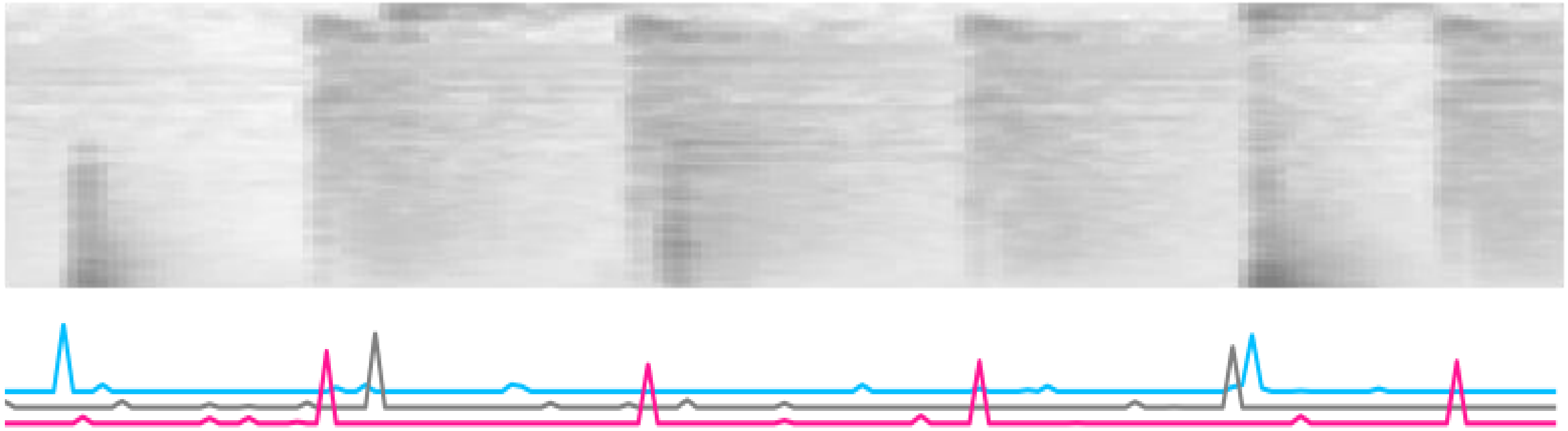
MODEL
Main Idea

Unlike existing methods, in which localization is explicitly achieved by design, our model learns localization **implicitly** as a byproduct of learning to count instances.

Model
Counting Occurrences

$$p_i\left(t\right) = f\left(\left(\mathbf{x}_i(n)\right)_{n=1}^{t}\right)$$

Probability of Event occurrence



MODEL
Counting Occurrences

$$p_i\left(t\right) = f\left(\left(\mathbf{x}_i(n)\right)_{n=1}^{t}\right)$$

$$E_i\left(t\right) = \mathfrak{B}\left(p_i(t)\right), \text{ ind. Bernoulli}$$

Event occurrence

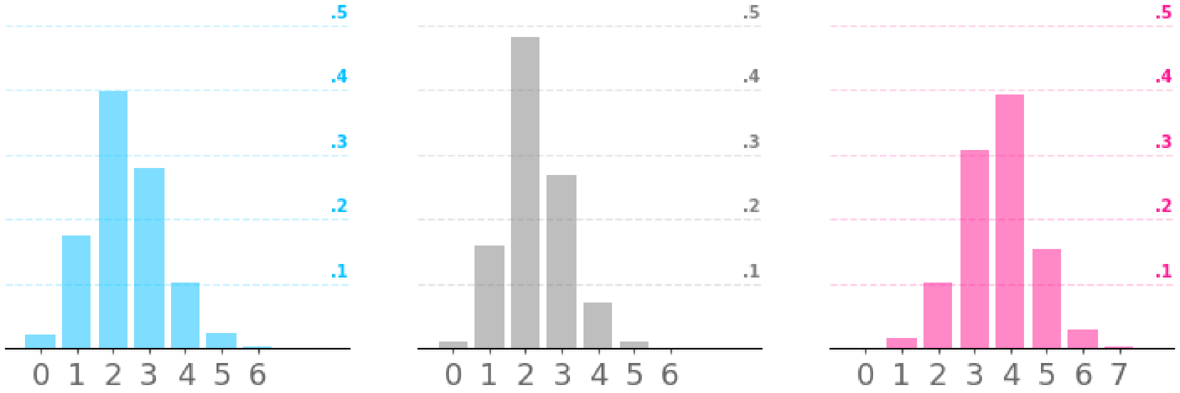
MODEL
Counting Occurrences

$$p_i\left(t\right) = f\left(\left(\mathbf{x}_i(n)\right)_{n=1}^{t}\right)$$

$$E_i\left(t\right) = \mathfrak{B}\left(p_i(t)\right), \text{ind. Bernoulli}$$

$$Y_i = \sum_t E_i(t)$$

Occurrence Count



MODEL
Counting Occurrences

Input Data

$$p_i(t) = f\left(\left(\boxed{\mathbf{x}_i}(n)\right)_{n=1}^{t}\right)$$

$$E_i(t) = \mathfrak{B}\left(p_i(t)\right), \text{ ind. Bernoulli}$$

$$Y_i = \sum_t E_i(t)$$

Model
Counting Occurrences

Input Data

$$p_i(t) = f\left(\left(\boxed{\mathbf{x}_i}(n)\right)_{n=1}^{t}\right)$$

$$E_i(t) = \mathfrak{B}\left(p_i(t)\right), \text{ ind. Bernoulli}$$

$$\boxed{Y_i} = \sum_t E_i(t)$$

Occurrence Count

Model
Counting Occurrences

Input Data

$$p_i(t) = \boxed{f}\left(\left(\boxed{\mathbf{x}_i}(n) \right)_{n=1}^{t} \right)$$

$$E_i(t) = \mathfrak{B}\left(p_i(t)\right), \text{ ind. Bernoulli}$$

$$\boxed{Y_i} = \sum_t E_i(t)$$

Occurrence Count

MODEL

Counting Occurrences

Estimated through RNN (e.g. LSTM)

Input Data

$$p_i(t) = \boxed{f}\left(\left(\boxed{\mathbf{x}_i}(n) \right)_{n=1}^{t} \right)$$

$$E_i(t) = \mathfrak{B}\left(p_i(t)\right), \text{ ind. Bernoulli}$$

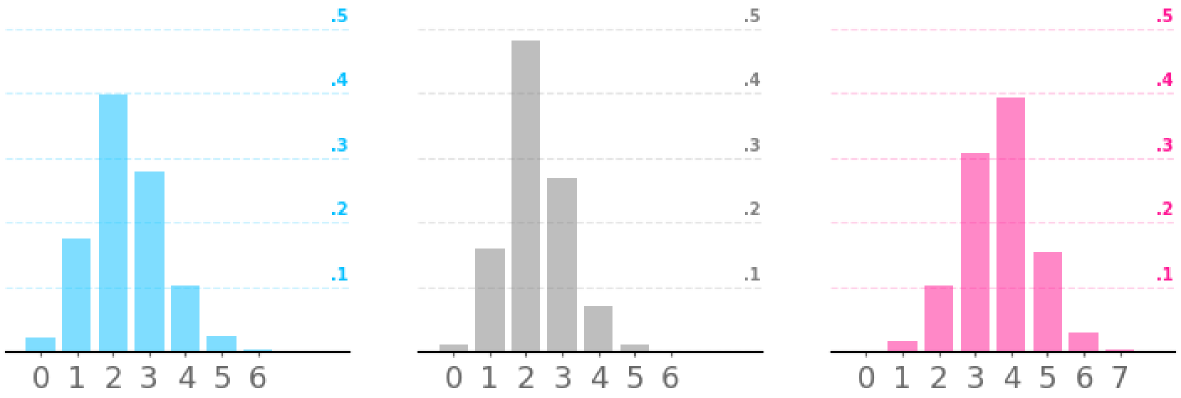
$$\boxed{Y_i} = \sum_t E_i(t)$$

Occurrence Count

Model
Loss

$$\boxed{Y_i} = \sum_t E_i(t)$$

Occurrence Count

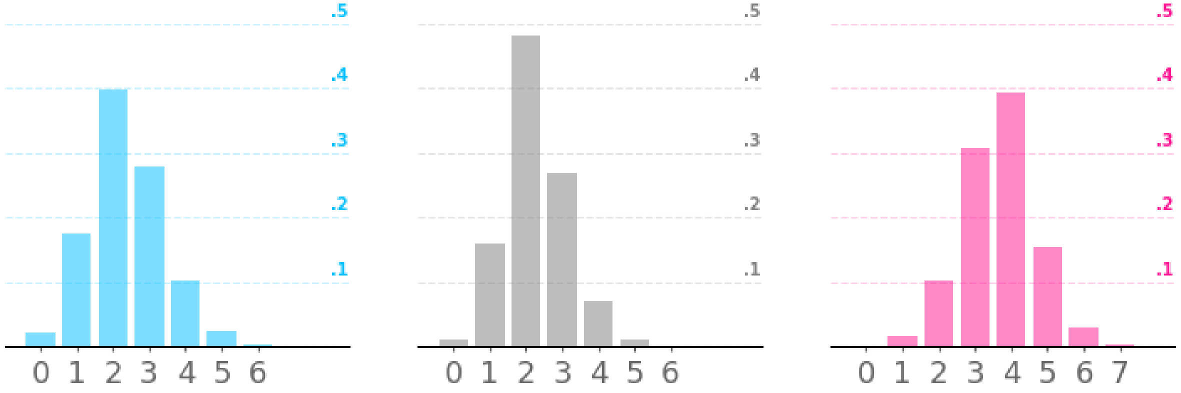


MODEL

Loss

$$\boxed{Y_i} = \sum_t E_i(t)$$

Occurrence Count

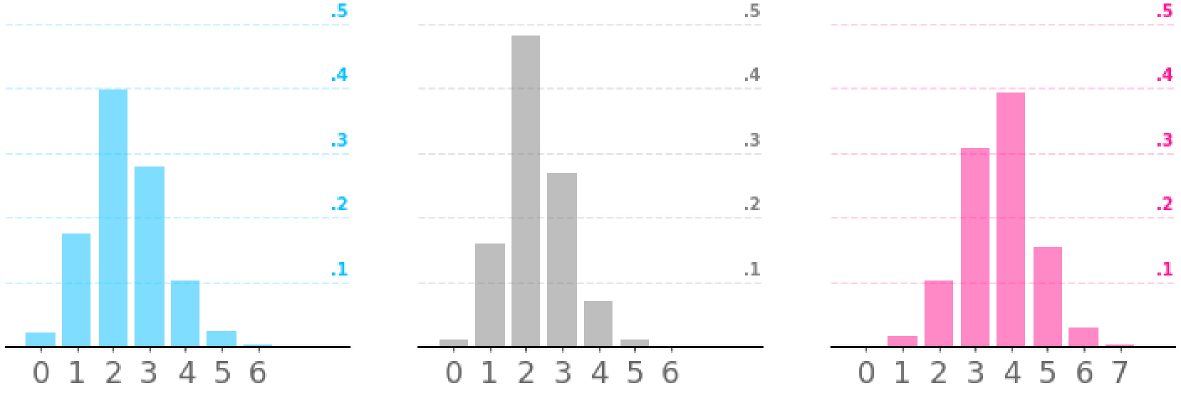


Compare them to true observed counts.

Model
Loss

$$\boxed{Y_i} = \sum_t E_i(t)$$

Occurrence Count



Compare them to true observed counts.

Observed *Count*

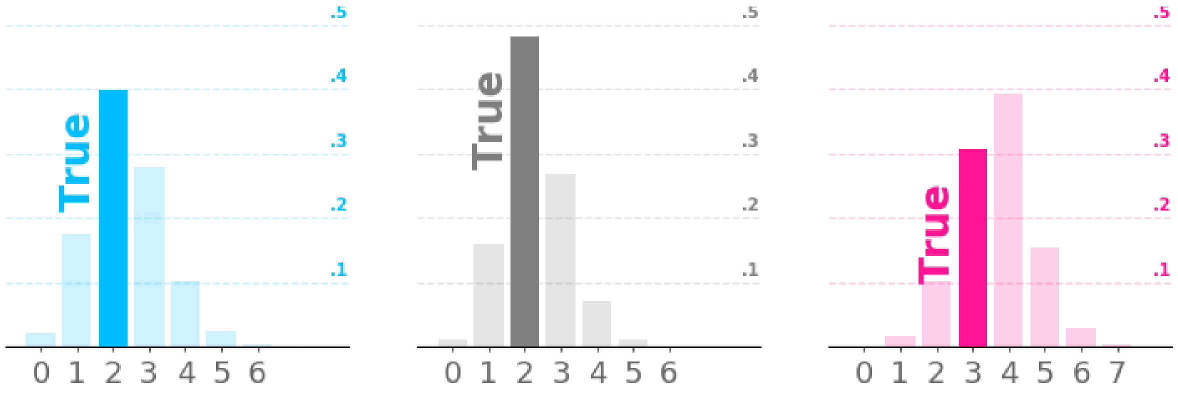
$$L(\theta) = -\sum \log\left(\Pr\left(\boxed{Y_{i,\theta}} = \boxed{y_i} \mid \mathbf{X}_i\right)\right)$$

MODEL

Loss

$$\boxed{Y_i} = \sum_t E_i(t)$$

Occurrence Count



Compare them to true observed counts.

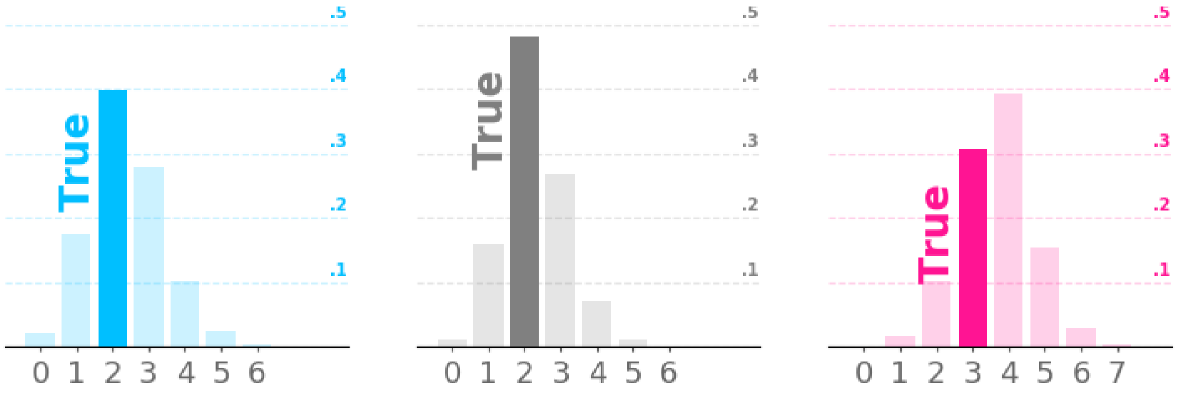
Observed *Count*

$$L(\theta) = -\sum \log\left(\Pr\left(\boxed{Y_{i,\theta}} = \boxed{y_i} \,\middle|\, \mathbf{X}_i\right)\right)$$

Model
Loss

$$\boxed{Y_i} = \sum_t E_i(t)$$

Occurrence Count



Compare them to true observed counts.

Observed Count

$$L(\theta) = -\sum \log\left(\Pr\left(\boxed{Y_{i,\theta}} = \boxed{y_i} \mid \mathbf{X}_i\right)\right)$$

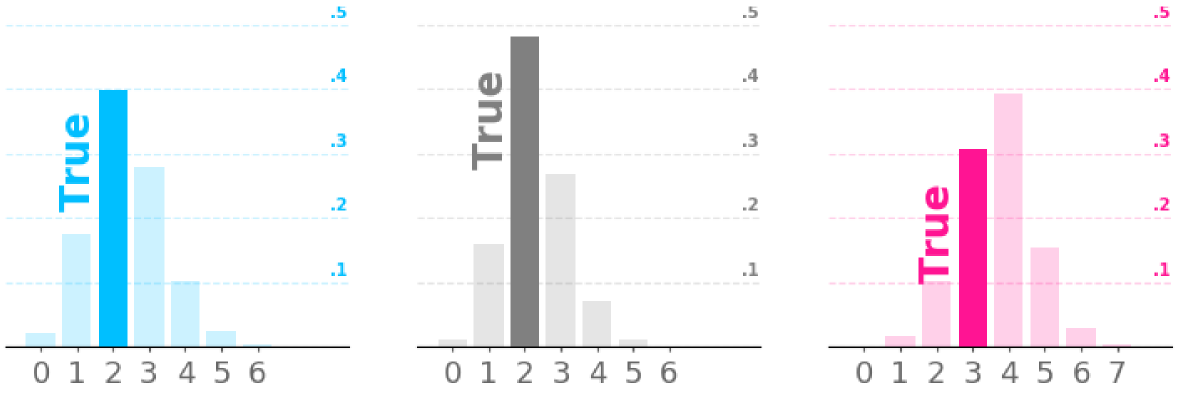
$$L(\theta) = -\log\left(\quad\right) - \log\left(\quad\right) - \log\left(\quad\right).$$

MODEL

Loss

$$\boxed{Y_i} = \sum_t E_i(t)$$

Occurrence Count



Compare them to true observed counts.

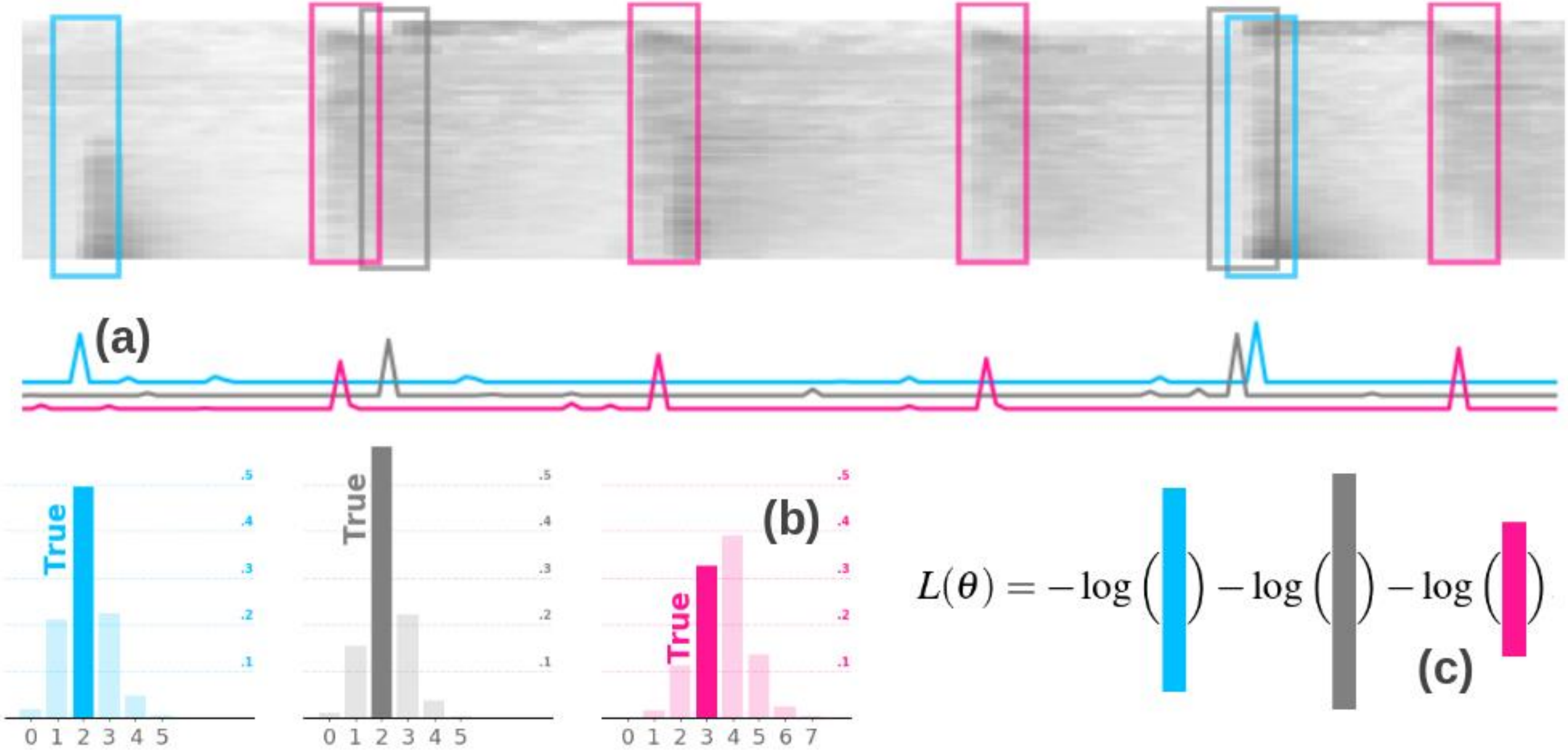
Observed *Count*

$$L(\theta) = -\sum \log\left(\Pr\left(\boxed{Y_{i,\theta}} = \boxed{y_i} \mid \mathbf{X}_i\right)\right)$$

Optimized with standard backpropagation

$$L(\theta) = -\log\left(\quad\right) - \log\left(\quad\right) - \log\left(\quad\right).$$

Model
Full Pipeline



Why does it work?

Model
Poisson Binomial Counts

$$\Pr(Y_{i,\theta} = k \mid \mathbf{X}_i) = \sum_{A \in F_k} \prod_{l \in A} \hat{p}_{i,\theta}(l) \prod_{j \in A^c} (1 - \hat{p}_{i,\theta}(j)),$$

Y follows a Poisson-binomial distribution

MODEL
Recursion

$$\Upsilon_i(k,t) := \Pr(Y_{i,\theta}\,(t) = k)$$

Bin k of count distribution at time t

MODEL
Recursion

$$\Upsilon_i(k,t) := \Pr(Y_{i,\theta}(t) = k)$$

Bin k of count distribution at time t

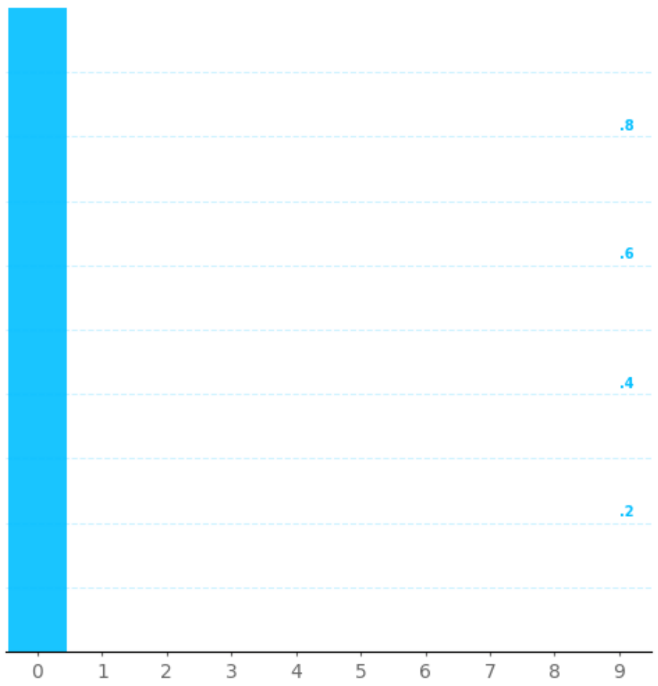
Property 2 (Recursion on k, t)

$$\Upsilon_i(k,t) = \begin{cases} (1-p_i(t))\Upsilon_i(k,t-1) & k=0 \\ (1-p_i(t))\Upsilon_i(k,t-1) + p_i(t)\Upsilon_i(k-1,t-1) & k>0 \end{cases} \tag{9}$$

where $\Upsilon_i(k,0) = \mathbb{1}_{k=0}$.

MODEL
Recursion

$t = 0$

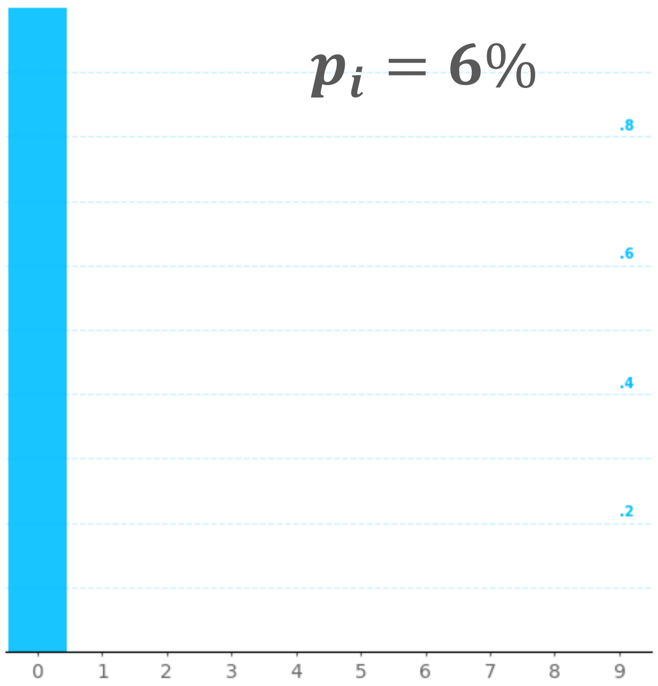


Property 2 (Recursion on k, t)

$$\Upsilon_i(k,t) = \begin{cases} (1-p_i(t))\Upsilon_i(k,t-1) & k=0 \\ (1-p_i(t))\Upsilon_i(k,t-1) + p_i(t)\Upsilon_i(k-1,t-1) & k>0 \end{cases} \tag{9}$$

where $\Upsilon_i(k,0) = \mathbb{1}_{k=0}$.

MODEL
Recursion

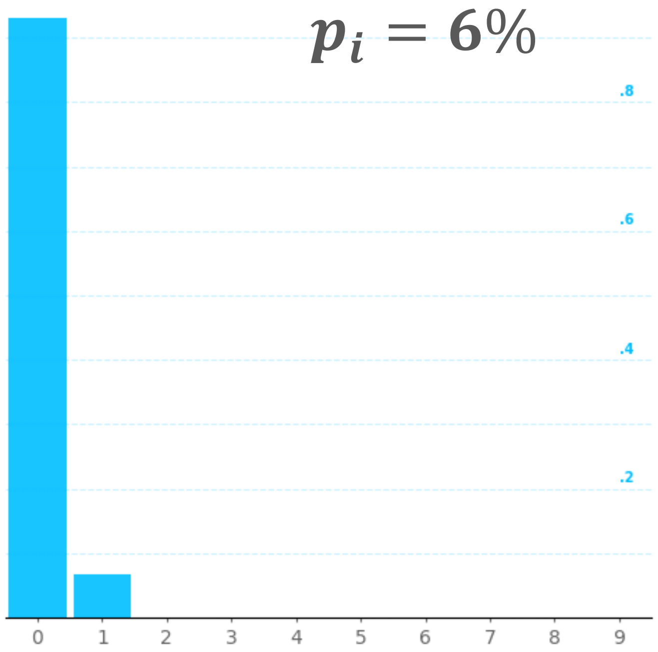


Property 2 (Recursion on k, t)

$$\Upsilon_i(k,t) = \begin{cases} (1-p_i(t))\Upsilon_i(k,t-1) & k=0 \\ (1-p_i(t))\Upsilon_i(k,t-1) + p_i(t)\Upsilon_i(k-1,t-1) & k>0 \end{cases} \tag{9}$$

where $\Upsilon_i(k,0) = \mathbb{1}_{k=0}$.

MODEL
Recursion

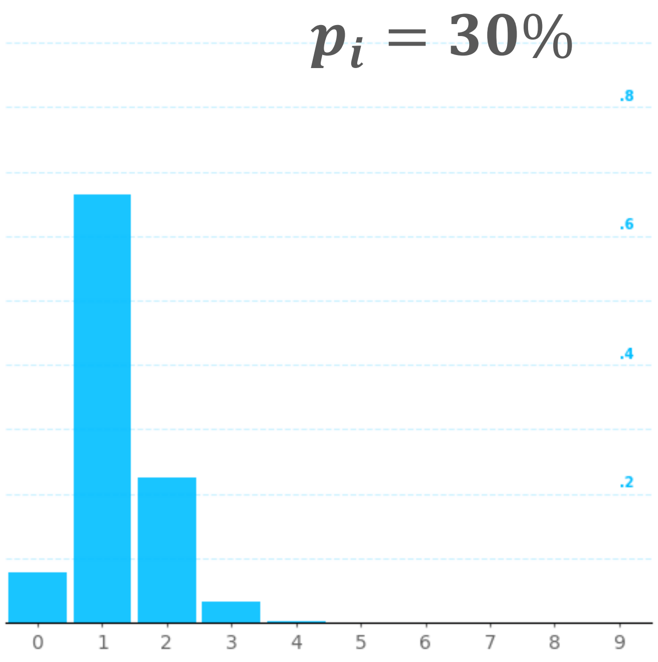


Property 2 (Recursion on k, t)

$$\Upsilon_i(k,t) = \begin{cases} (1-p_i(t))\Upsilon_i(k,t-1) & k=0 \\ (1-p_i(t))\Upsilon_i(k,t-1) + p_i(t)\Upsilon_i(k-1,t-1) & k>0 \end{cases} \tag{9}$$

where $\Upsilon_i(k,0) = \mathbb{1}_{k=0}$.

Model
Recursion

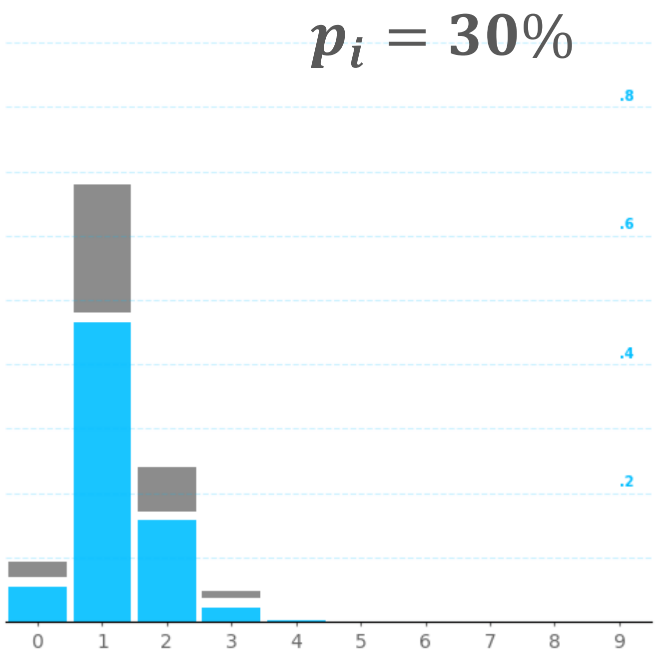


Property 2 (Recursion on k, t)

$$\Upsilon_i(k,t) = \begin{cases} (1-p_i(t))\Upsilon_i(k,t-1) & k=0 \\ (1-p_i(t))\Upsilon_i(k,t-1) + p_i(t)\Upsilon_i(k-1,t-1) & k>0 \end{cases} \tag{9}$$

where $\Upsilon_i(k,0) = \mathbb{1}_{k=0}$.

MODEL
Recursion

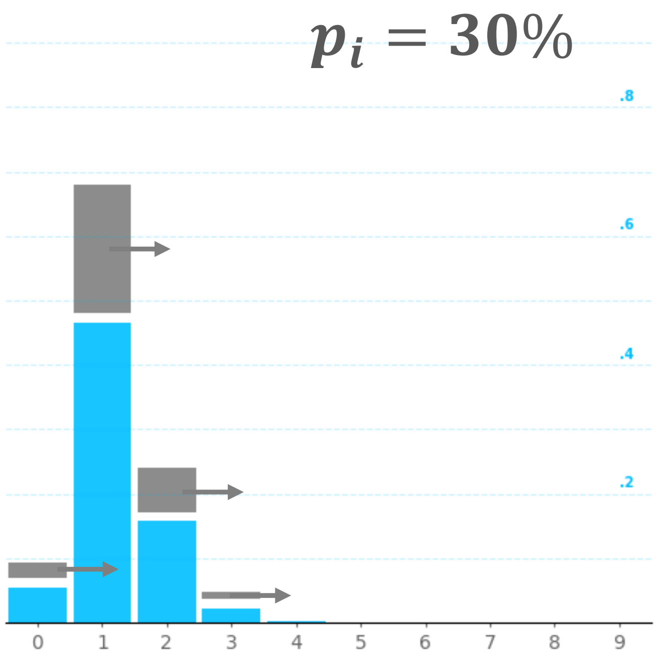


Property 2 (Recursion on k, t)

$$\Upsilon_i(k,t) = \begin{cases} (1-p_i(t))\Upsilon_i(k,t-1) & k=0 \\ (1-p_i(t))\Upsilon_i(k,t-1) + p_i(t)\Upsilon_i(k-1,t-1) & k>0 \end{cases} \tag{9}$$

where $\Upsilon_i(k,0) = \mathbb{1}_{k=0}$.

MODEL
Recursion



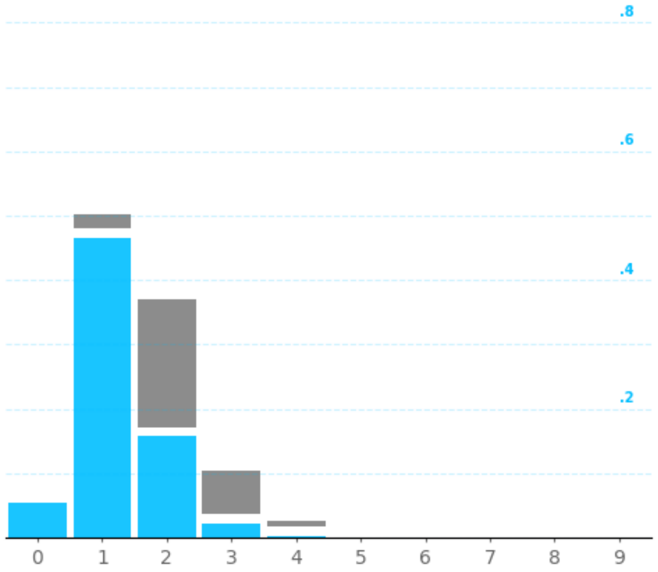
Property 2 (Recursion on k, t)

$$\Upsilon_i(k,t) = \begin{cases} (1-p_i(t))\Upsilon_i(k,t-1) & k=0 \\ (1-p_i(t))\Upsilon_i(k,t-1) + p_i(t)\Upsilon_i(k-1,t-1) & k>0 \end{cases} \tag{9}$$

where $\Upsilon_i(k,0) = \mathbb{1}_{k=0}$.

Model
Recursion

$p_i = 30\%$



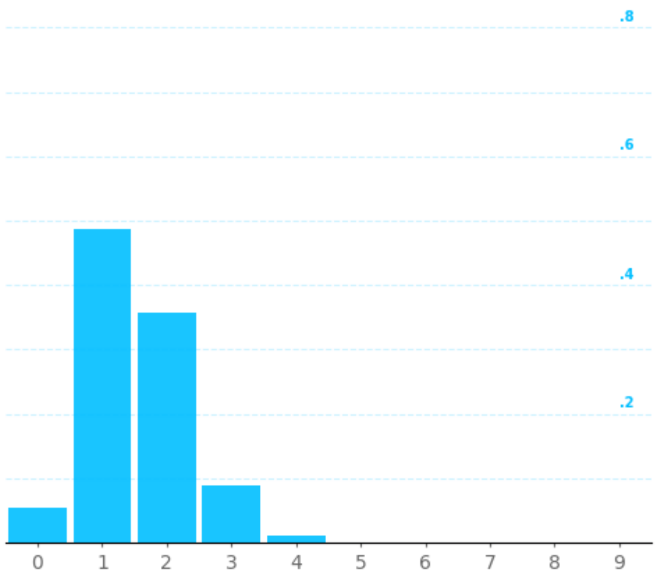
Property 2 (Recursion on k, t)

$$\Upsilon_i(k,t) = \begin{cases} (1-p_i(t))\Upsilon_i(k,t-1) & k=0 \\ (1-p_i(t))\Upsilon_i(k,t-1) + p_i(t)\Upsilon_i(k-1,t-1) & k>0 \end{cases} \quad (9)$$

where $\Upsilon_i(k,0) = \mathbb{1}_{k=0}$.

MODEL
Recursion

$p_i = 30\%$



Property 2 (Recursion on k, t)

$$\Upsilon_i(k,t) = \begin{cases} (1-p_i(t))\Upsilon_i(k,t-1) & k=0 \\ (1-p_i(t))\Upsilon_i(k,t-1) + p_i(t)\Upsilon_i(k-1,t-1) & k>0 \end{cases} \tag{9}$$

where $\Upsilon_i(k,0) = \mathbb{1}_{k=0}$.

Model
No early triggering

Property 1 (Mass shift irreversibility)

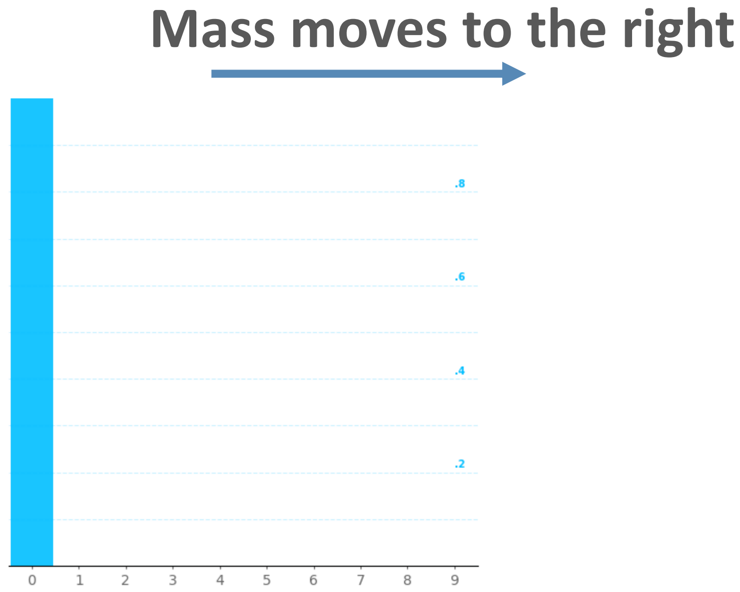
$$\left(Y_{i,\theta}(t)\right)_{t=1}^{T_i} \text{ is monotonically increasing.}$$

Model

No early triggering

Property 1 (Mass shift irreversibility)

$$\left(Y_{i,\theta}(t)\right)_{t=1}^{T_i} \text{ is monotonically increasing.}$$

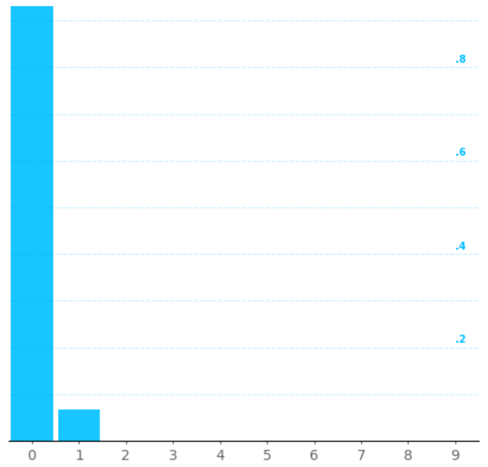


Model
No early triggering

Property 1 (Mass shift irreversibility)

$$\left(Y_{i,\theta}(t)\right)_{t=1}^{T_i} \text{ is monotonically increasing.}$$

Mass moves to the right

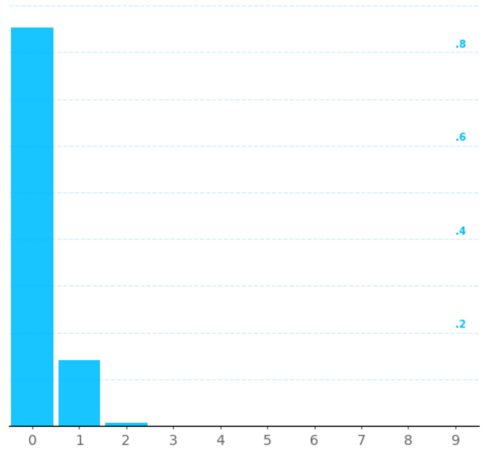


Model
No early triggering

Property 1 (Mass shift irreversibility)

$$\left(Y_{i,\theta}(t)\right)_{t=1}^{T_i} \text{ is monotonically increasing.}$$

Mass moves to the right

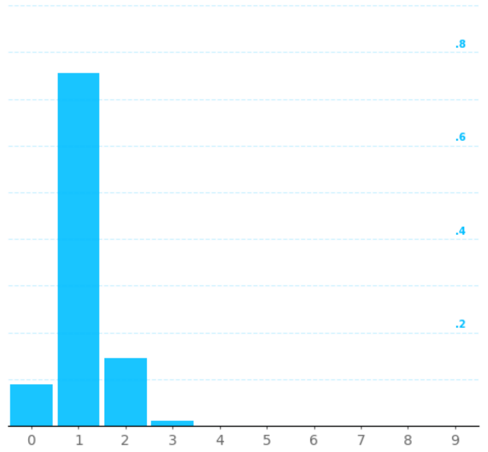


MODEL
No early triggering

Property 1 (Mass shift irreversibility)

$$\left(Y_{i,\theta}(t)\right)_{t=1}^{T_i} \text{ is monotonically increasing.}$$

Mass moves to the right

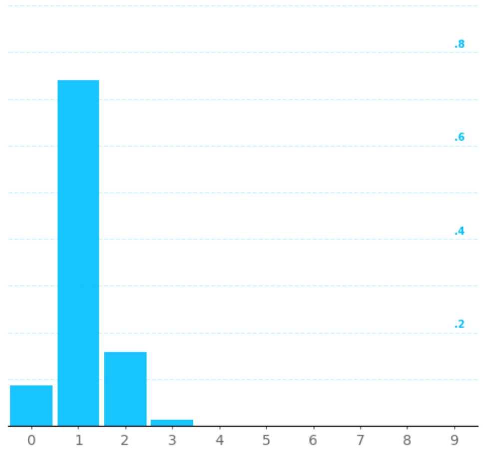


Model
No early triggering

Property 1 (Mass shift irreversibility)

$$\left(Y_{i,\theta}(t)\right)_{t=1}^{T_i} \text{ is monotonically increasing.}$$

Mass moves to the right

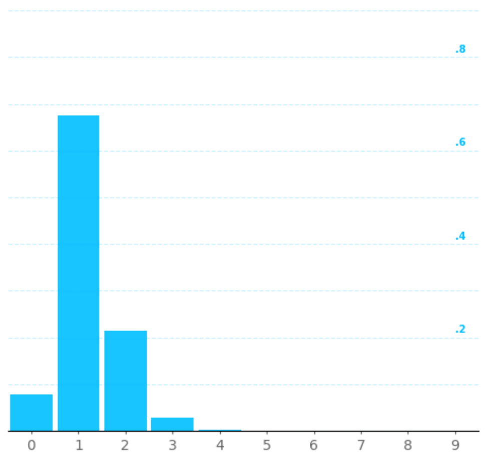


MODEL
No early triggering

Property 1 (Mass shift irreversibility)

$$\left(Y_{i,\theta}(t)\right)_{t=1}^{T_i} \text{ is monotonically increasing.}$$

Mass moves to the right

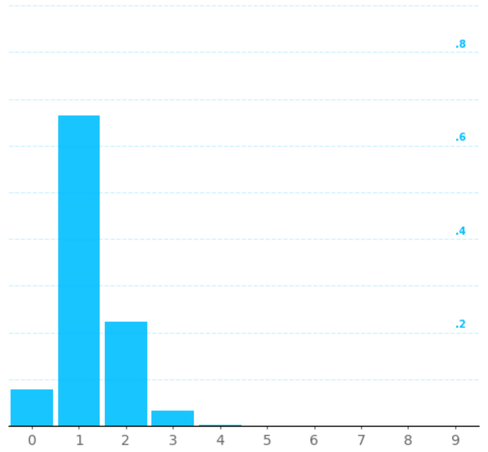


Model
No early triggering

Property 1 (Mass shift irreversibility)

$$\left(Y_{i,\theta}(t)\right)_{t=1}^{T_i} \text{ is monotonically increasing.}$$

Mass moves to the right

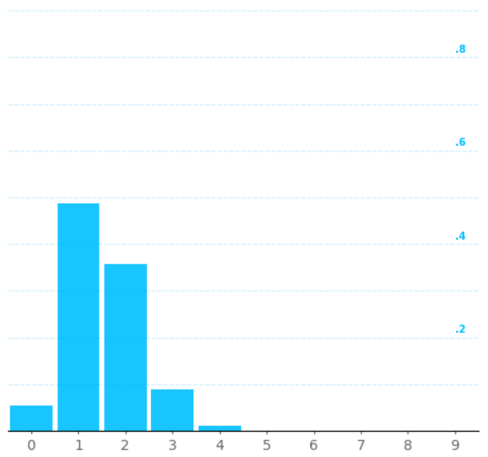


MODEL
No early triggering

Property 1 (Mass shift irreversibility)

$$\left(Y_{i,\theta}(t)\right)_{t=1}^{T_i} \text{ is monotonically increasing.}$$

Mass moves to the right

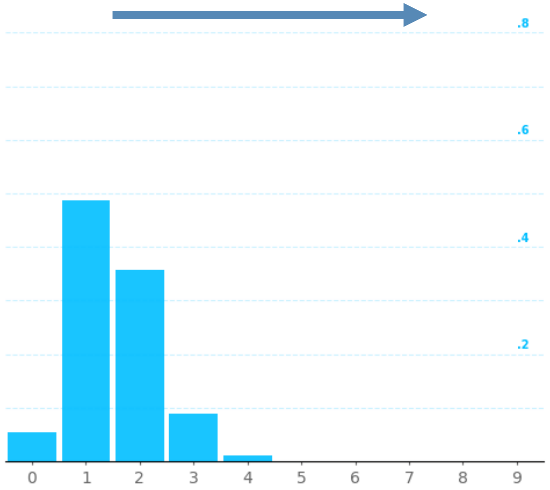


Model
No early triggering

Property 1 (Mass shift irreversibility)

$$\left(Y_{i,\theta}(t)\right)_{t=1}^{T_i} \text{ is monotonically increasing.}$$

Mass moves to the right



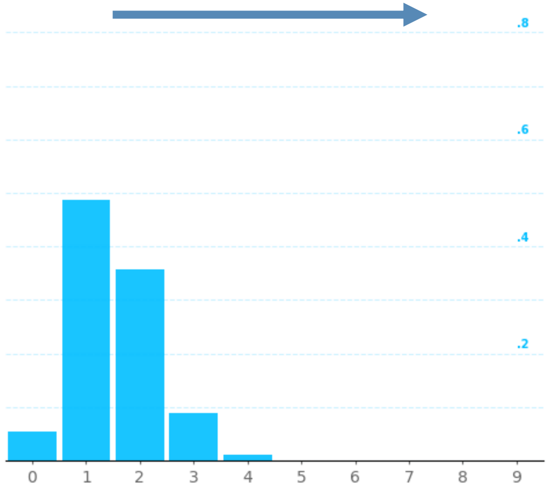
***Consequence:* Mass shifts are irreversible**

Model
No early triggering

Property 1 (Mass shift irreversibility)

$$\left(Y_{i,\theta}(t)\right)_{t=1}^{T_i} \text{ is monotonically increasing.}$$

Mass moves to the right



Consequence:* *Mass shifts are irreversible

- prevents the model from triggering early
- prevents the model from false alarms

Model
Mass Convergence

Lemma 2 (First upper bound)

$$\max_{k} \Upsilon_i(k,t) \leq \frac{1}{2} + \min_{j \leq t} \| \frac{1}{2} - p_i(j) \|$$

MODEL
Mass Convergence

Lemma 2 (First upper bound)

$$\max_k \Upsilon_i(k,t) \leq \frac{1}{2} + \min_{j \leq t} \| \frac{1}{2} - p_i(j) \|$$

$$L(\theta) = -\sum_i \log\left(\Pr\left(Y_{i,\theta} = y_i \mid \mathbf{X}_i\right)\right)$$

Counting Loss

Model
Mass Convergence

Lemma 2 (First upper bound)

$$\max_k \Upsilon_i(k,t) \leq \frac{1}{2} + \min_{j \leq t} \| \frac{1}{2} - p_i(j) \|$$

$$L(\theta) = -\sum_i \log\left(\Pr\left(Y_{i,\theta} = y_i \mid \mathbf{X}_i\right)\right)$$

$$= -\sum_i \log\left(\Upsilon_i(y_i, T_i)\right)$$

Counting Loss

MODEL
Mass Convergence

Lemma 2 (First upper bound)

$$\max_k \Upsilon_i(k,t) \leq \frac{1}{2} + \min_{j \leq t} \| \frac{1}{2} - p_i(j) \|$$

$$L(\theta) = -\sum_i \log\left(\Pr\left(Y_{i,\theta} = y_i \mid \mathbf{X}_i\right)\right)$$

Counting Loss

$$= -\sum_i \log\left(\Upsilon_i(y_i, T_i)\right)$$

$$\boxed{\overset{(11)}{\geq}} -\sum_i \log\left(\frac{1}{2} + \min_{j \leq t} \| \frac{1}{2} - p_i(j) \|\right)$$

MODEL

Mass Convergence

Lemma 2 (First upper bound)

$$\max_k \Upsilon_i(k,t) \leq \frac{1}{2} + \min_{j \leq t} \| \frac{1}{2} - p_i(j) \|$$

Learns to count $\downarrow$

$$L(\theta) = -\sum_i \log\left(\Pr\left(Y_{i,\theta} = y_i \mid \mathbf{X}_i\right)\right)$$

$$= -\sum_i \log\left(\Upsilon_i(y_i, T_i)\right)$$

$$\overset{(11)}{\geq} -\sum_i \log\left(\frac{1}{2} + \min_{j \leq t} \| \frac{1}{2} - p_i(j) \|\right)$$

Counting Loss

Model
Mass Convergence

Lemma 2 (First upper bound)

$$\max_k \Upsilon_i(k,t) \leq \frac{1}{2} + \min_{j \leq t} \| \frac{1}{2} - p_i(j) \|$$

Learns to count

$$L(\theta) = -\sum_i \log\left(\Pr\left(Y_{i,\theta} = y_i \mid \mathbf{X}_i\right)\right)$$

Counting Loss

$$= -\sum_i \log\left(\Upsilon_i(y_i, T_i)\right)$$

Converge towards 0,1 extremes

$$\overset{(11)}{\geq} -\sum_i \log\left(\frac{1}{2} + \min_{j \leq t} \| \frac{1}{2} - \boxed{p_i(j)} \|\right)$$

MODEL

Mass Convergence

Property 3 (Sparse mass concentration) The inequality derived below reveals that, as the loss decreases, small $p_i(\cdot)$ will quickly converge towards zero.

$$
\begin{aligned}
\max_k \Upsilon_i(k,t) &\overset{(8)}{\leq} \min_{l\leq t} \max_k \Upsilon_i(k,l) \overset{\text{ind}}{=} \min_{\sigma,l\leq t} \max_k \Upsilon_{i,\sigma}(k,l) \\
&\overset{\text{Le Cam}}{\leq} \min_{\sigma,l\leq t} \max_k \frac{\lambda_{i,\sigma,l}^k e^{-\lambda_{i,\sigma,l}}}{k!} + 2\sum_{j=1}^{l} p_{i,\sigma}(j)^2 \qquad (13) \\
&\overset{\text{def}}{=} \min_{\sigma,l\leq t} \max_k \frac{\left[\sum_{j=1}^{l} p_{i,\sigma}(j)\right]^k e^{-\left[\sum_{j=1}^{l} p_{i,\sigma}(j)\right]}}{k!} + 2\sum_{j=1}^{l} p_{i,\sigma}(j)^2,
\end{aligned}
$$

Model
Mass Convergence

Property 3 (Sparse mass concentration) The inequality derived below reveals that, as the loss decreases, small $p_i(\cdot)$ will quickly converge towards zero.

$$
\begin{aligned}
\max_k \Upsilon_i(k,t) &\overset{(8)}{\leq} \min_{l \leq t} \max_k \Upsilon_i(k,l) \overset{\text{ind}}{=} \min_{\sigma, l \leq t} \max_k \Upsilon_{i,\sigma}(k,l) \\
&\overset{\text{Le Cam}}{\leq} \min_{\sigma, l \leq t} \max_k \frac{\lambda_{i,\sigma,l}^k e^{-\lambda_{i,\sigma,l}}}{k!} + 2 \sum_{j=1}^{l} p_{i,\sigma}(j)^2 \qquad (13) \\
&\overset{\text{def}}{=} \min_{\sigma, l \leq t} \max_k \frac{\left[\sum_{j=1}^{l} p_{i,\sigma}(j)\right]^k e^{-\left[\sum_{j=1}^{l} p_{i,\sigma}(j)\right]}}{k!} + 2 \sum_{j=1}^{l} p_{i,\sigma}(j)^2,
\end{aligned}
$$

A detection cannot be split into numerous small $p_i(\cdot)$ contributions

Model
Mass Convergence

As the model learns to count event occurrences:

MODEL
Mass Convergence

As the model learns to count event occurrences:

- $p_i(\cdot)$ converge towards 0,1 extremes

MODEL
Mass Convergence

As the model learns to count event occurrences:

- $p_i(\cdot)$ converge towards 0,1 extremes
- A detection cannot be split into numerous small $p_i(\cdot)$ contributions

Model
Mass Convergence

As the model learns to count event occurrences:

- $p_i(\cdot)$ converge towards 0,1 extremes
- A detection cannot be split into numerous small $p_i(\cdot)$ contributions

A single $p_i(\cdot)$ will contain almost all of them mass for an event.

MODEL
Properties

1. Almost **binary** predictions

MODEL
Properties

1. **Almost binary predictions**
2. **No early triggering**

MODEL
Properties

1. Almost **binary** predictions
2. No **early** triggering
3. No systematic **late** bias ← *Not a theoretical property*

Model
Properties

1. **Almost binary predictions**
2. **No early triggering**
3. **No systematic late bias** ⟵ *Not a theoretical property*

↓

Achieved trough an implementation trick:

MODEL
Properties

1. **Almost binary predictions**
2. **No early triggering**
3. **No systematic late bias** ← *Not a theoretical property*

Achieved trough an implementation trick:
Feeding sequences of variable length

Model
Properties

1. Almost binary predictions
2. No early triggering
3. No systematic late bias

MODEL
Properties

1. Almost binary predictions
2. No early triggering
3. No systematic late bias

If the model accurately learns to count occurrences and if the events are detectable, then a coherent localization will emerge naturally.

Experiments

Drum Detection
Experiment Specifications

Detection of three different drum types in drum audio extracts ♫♫

Drum Detection
Experiment Specifications

Detection of three different drum types in drum audio extracts ♫♫

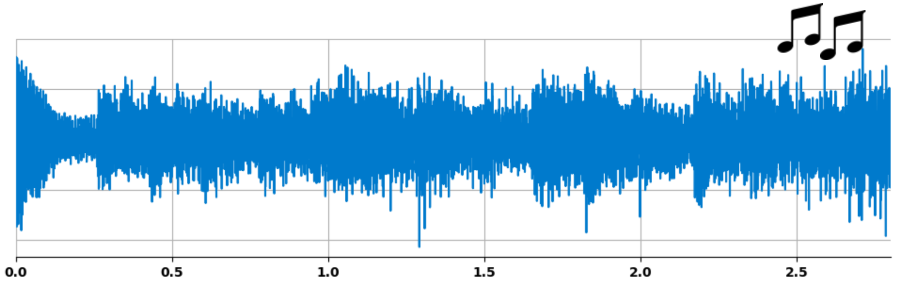
- Tight tolerance of 50ms for a prediction to be correct

Drum Detection
Experiment Specifications

Detection of three different drum types in drum audio extracts ♫♫

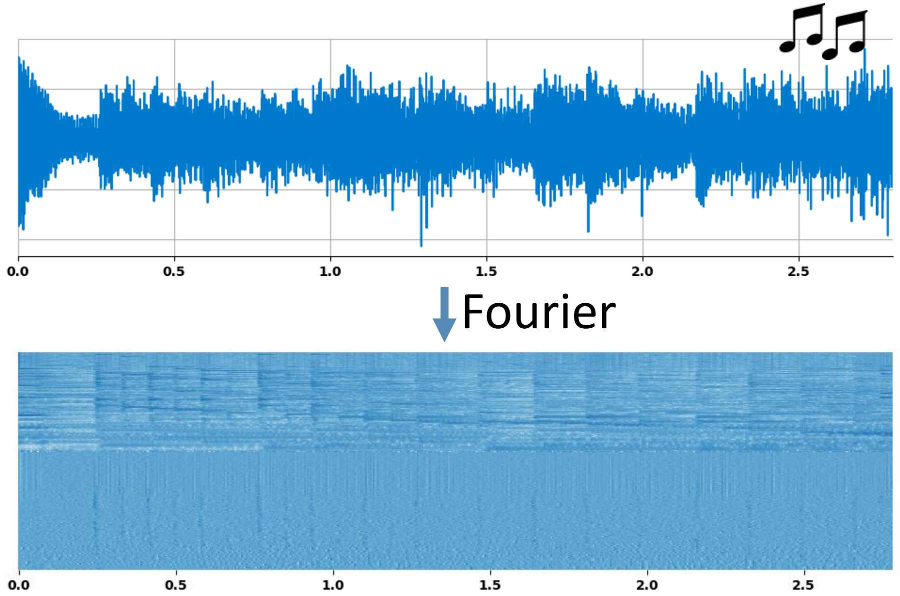
- Tight tolerance of 50ms for a prediction to be correct
- Comparison with fully-supervised benchmark models

Drum Detection
Our approach



Signal

Drum Detection
Our approach

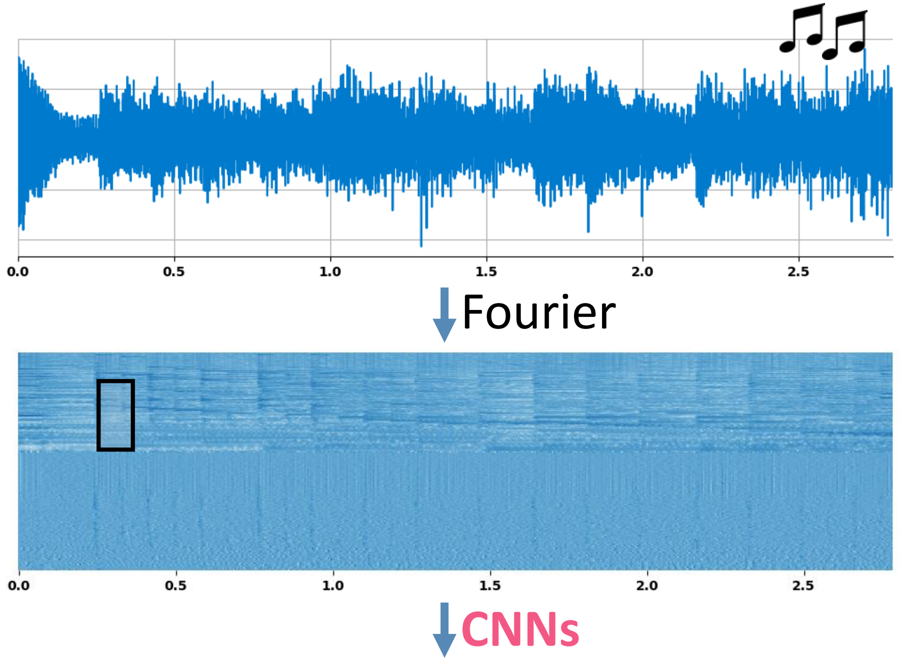


Signal

Mel-spectrogram

1st order derivative

DRUM DETECTION
Our approach



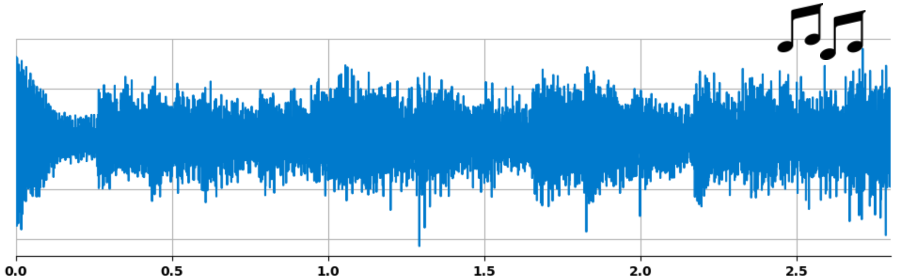
Convolutional Representations

Signal

Mel-spectrogram

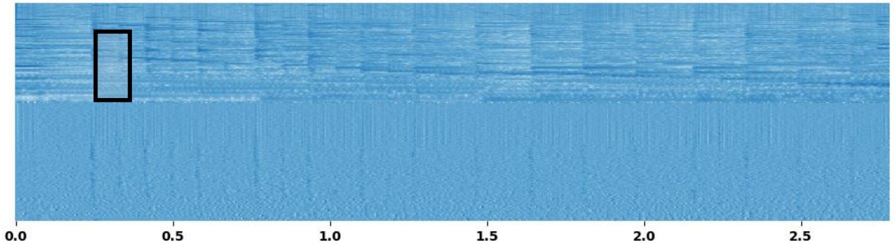
1st order derivative

Drum Detection
Our approach



Signal

↓ Fourier



Mel-spectrogram

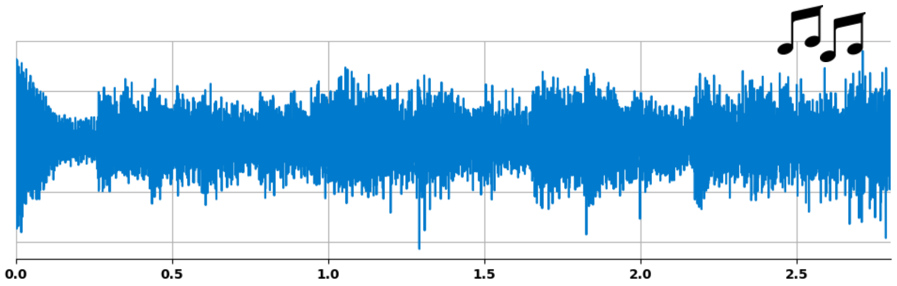
1st order derivative

↓ **CNNs**

Convolutional Representations

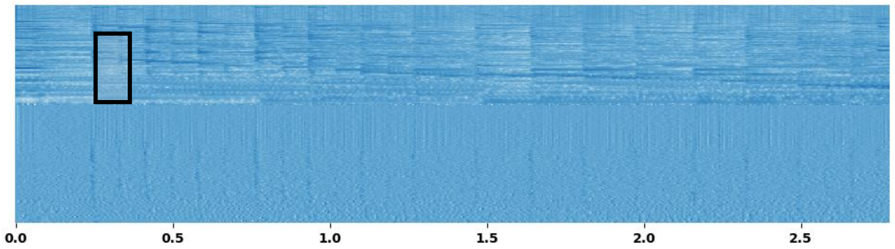
↓ **LSTM**

DRUM DETECTION
Our approach



Signal

↓ Fourier



Mel-spectrogram

1st order derivative

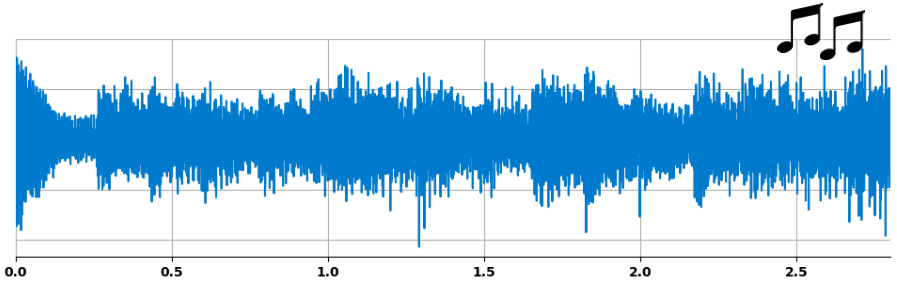
↓ **CNNs**

Convolutional Representations

↓ **LSTM**

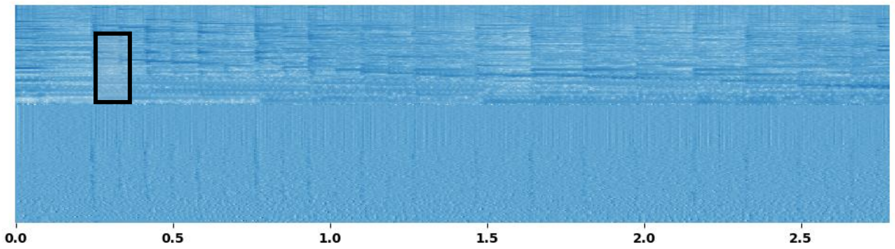
↓ **FCs**

DRUM DETECTION
Our approach



Signal

Fourier



Mel-spectrogram

1^{st} order derivative

CNNs

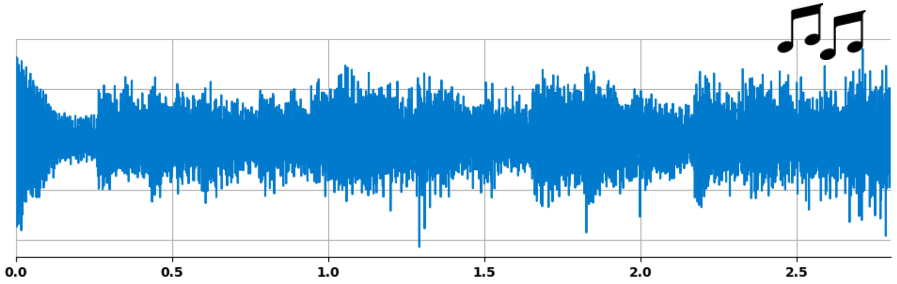
Convolutional Representations

LSTM

FCs

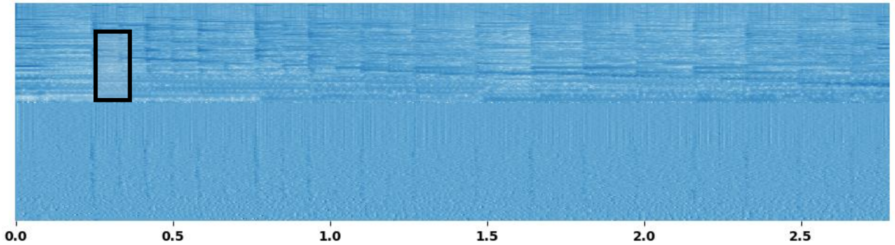
Localization

DRUM DETECTION
Our approach



Signal

↓ Fourier



Mel-spectrogram

1st order derivative

↓ **CNNs**

Convolutional Representations

↓ **LSTM**

↓ **FCs**

Localization

Trained with our loss
(using only occurrence counts)

Drum Detection
Results

	Method	KD	SD	HH	Pre	Rec	F_1
		D-DTP dataset					
Random	RNN	**94.7**	79.5	88.3	84.1	**93.3**	87.5
	tanhB	92.4	84.6	87.1	86.3	92.1	88.0
	ReLUts	91.3	83.8	85.2	83.7	92.3	86.8
	lstmpB	94.4	84.1	91.4	90.8	90.8	**90.0**
	GRUts	94.2	**87.1**	87.7	88.6	92.7	89.7
	ours (LoCo)	*92.3*	*81.2*	***93.0***	***90.9***	*87.1*	*88.9*
Subset	RNN	**91.0**	57.8	82.2	72.8	**88.3**	77.0
	tanhB	82.7	61.6	84.8	74.1	83.8	76.4
	ReLUts	79.4	62.1	80.8	69.6	84.2	74.1
	lstmpB	85.8	**68.8**	83.7	78.3	84.7	**79.4**
	GRUts	87.7	62.3	79.4	73.0	85.2	76.5
	ours (LoCo)	*84.9*	*59.4*	***90.0***	***84.8***	*73.5*	*78.1*

Drum Detection
Results

	Method	KD	SD	HH	Pre	Rec	F_1
		D-DTP dataset					
Random	RNN	**94.7**	79.5	88.3	84.1	**93.3**	87.5
	tanhB	92.4	84.6	87.1	86.3	92.1	88.0
	ReLUts	91.3	83.8	85.2	83.7	92.3	86.8
	lstmpB	94.4	84.1	91.4	90.8	90.8	**90.0**
	GRUts	94.2	**87.1**	87.7	88.6	92.7	89.7
	ours (LoCo)	*92.3*	*81.2*	***93.0***	***90.9***	*87.1*	*88.9*
Subset	RNN	**91.0**	57.8	82.2	72.8	**88.3**	77.0
	tanhB	82.7	61.6	84.8	74.1	83.8	76.4
	ReLUts	79.4	62.1	80.8	69.6	84.2	74.1
	lstmpB	85.8	**68.8**	83.7	78.3	84.7	**79.4**
	GRUts	87.7	62.3	79.4	73.0	85.2	76.5
	ours (LoCo)	*84.9*	*59.4*	***90.0***	***84.8***	*73.5*	*78.1*

State-of-the-art

Drum Detection
Results

	Method	KD	SD	HH	Pre	Rec	F_1
		D-DTP dataset					
Random	RNN	**94.7**	79.5	88.3	84.1	**93.3**	87.5
	tanhB	92.4	84.6	87.1	86.3	92.1	88.0
	ReLUts	91.3	83.8	85.2	83.7	92.3	86.8
	lstmpB	94.4	84.1	91.4	90.8	90.8	**90.0**
	GRUts	94.2	**87.1**	87.7	88.6	92.7	89.7
	ours (LoCo)	*92.3*	*81.2*	***93.0***	***90.9***	*87.1*	*88.9*
Subset	RNN	**91.0**	57.8	82.2	72.8	**88.3**	77.0
	tanhB	82.7	61.6	84.8	74.1	83.8	76.4
	ReLUts	79.4	62.1	80.8	69.6	84.2	74.1
	lstmpB	85.8	**68.8**	83.7	78.3	84.7	**79.4**
	GRUts	87.7	62.3	79.4	73.0	85.2	76.5
	ours (LoCo)	*84.9*	*59.4*	***90.0***	***84.8***	*73.5*	*78.1*

Great Overall F1-Score

State-of-the-art

Drum Detection
Results

Detection of three different drum types in drum audio extracts

Further tests on HH reveal that:

DRUM DETECTION
Results

Detection of three different drum types in drum audio extracts

Further tests on HH reveal that:

- In that setting, the standard deviation is only of **4.35ms** for the distance between true and predicted hits.

Piano Onset Detection
Results

Detection of piano notes in audio extracts ♫♫

PIANO ONSET DETECTION
Results

Detection of piano notes in audio extracts ♫♫

- Complex task with 88 channels

Piano Onset Detection
Results

Detection of piano notes in audio extracts ♫♫

- Complex task with 88 channels
- Tight tolerance of 50ms for a prediction to be considered correct

PIANO ONSET DETECTION
Results

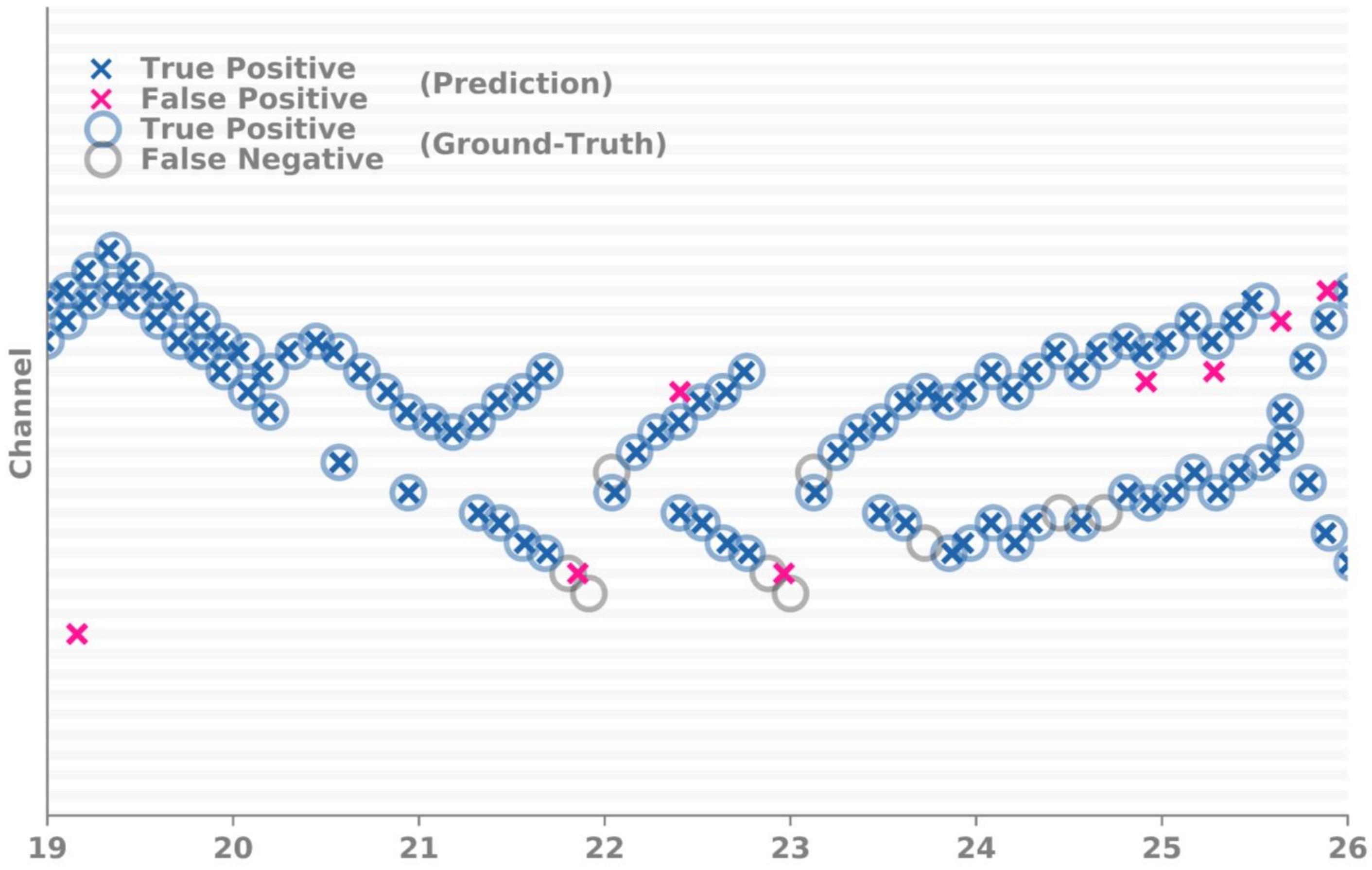
Detection of piano notes in audio extracts 🎶

- Complex task with 88 channels
- Tight tolerance of 50ms for a prediction to be considered correct
- Comparison with fully-supervised benchmark models

PIANO ONSET DETECTION
Results

METHOD	PRE	REC	F_1
SIGTIA ET AL.(2016)	44.97	49.55	46.58
KELZ ET AL.(2016)	44.27	61.29	50.94
HAWTHORNE ET AL.(2017)	**84.24**	**80.67**	**82.29**
ours (LoCo)	*76.22*	*68.61*	*71.99*

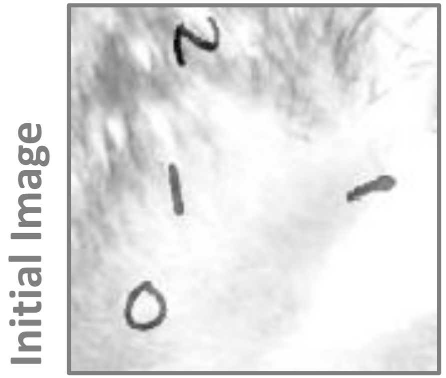
Piano Onset Detection
Results



Digit Detection Experiment

Digit Detection Experiment

Main Idea

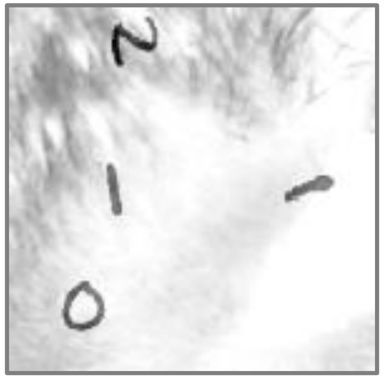


Not a sequence

DIGIT DETECTION EXPERIMENT

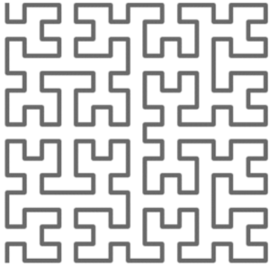
Main Idea

Initial Image



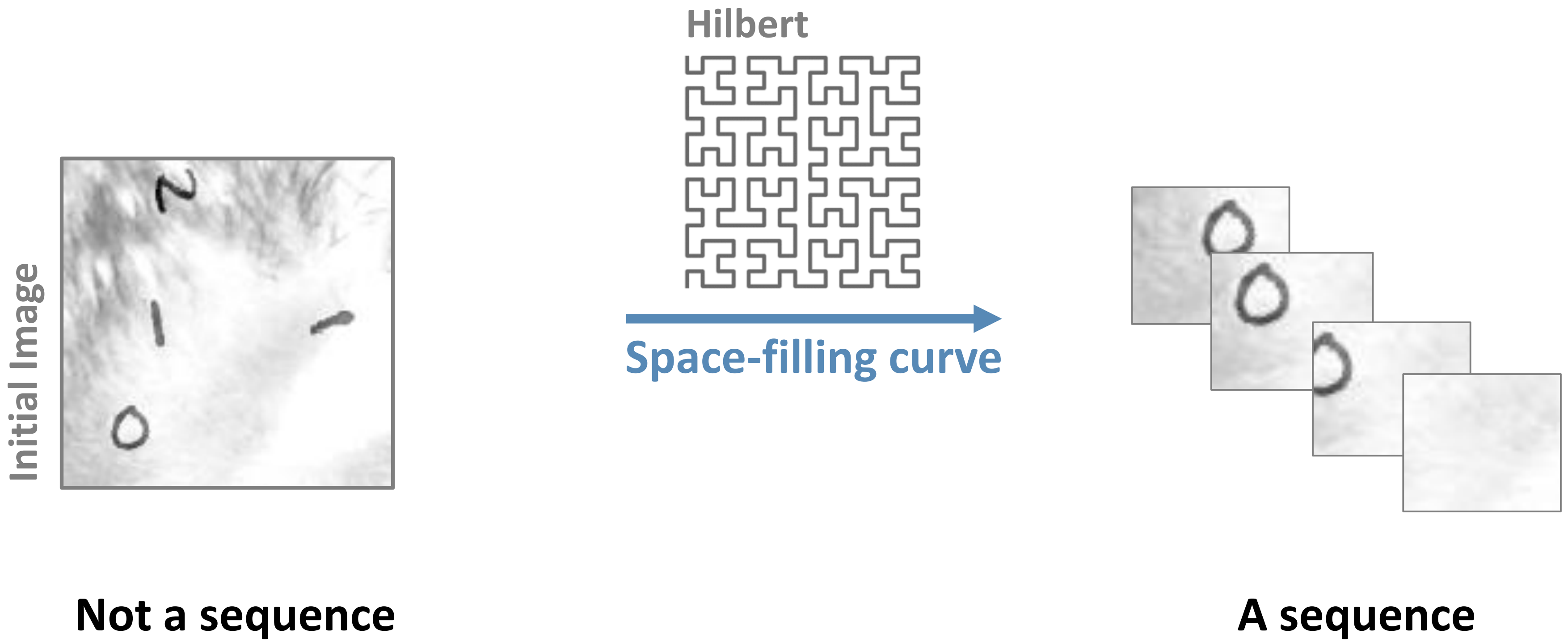
Not a sequence

Hilbert

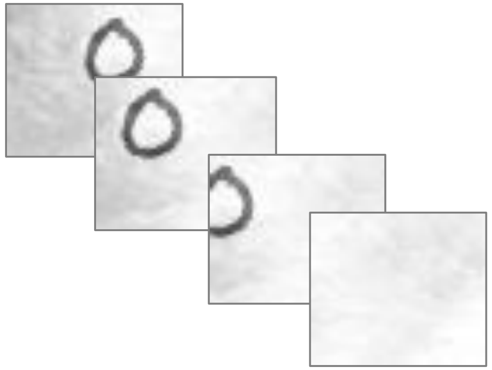


Space-filling curve

Digit Detection Experiment
Main Idea

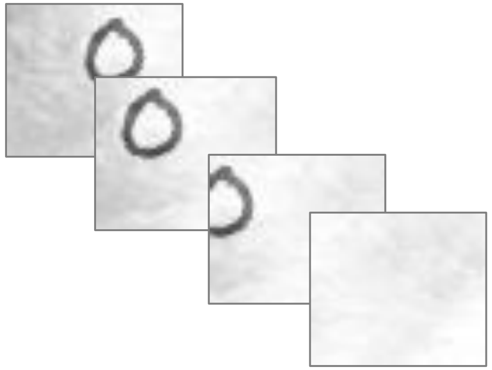


DIGIT DETECTION EXPERIMENT
Main Idea



Input

DIGIT DETECTION EXPERIMENT
Main Idea

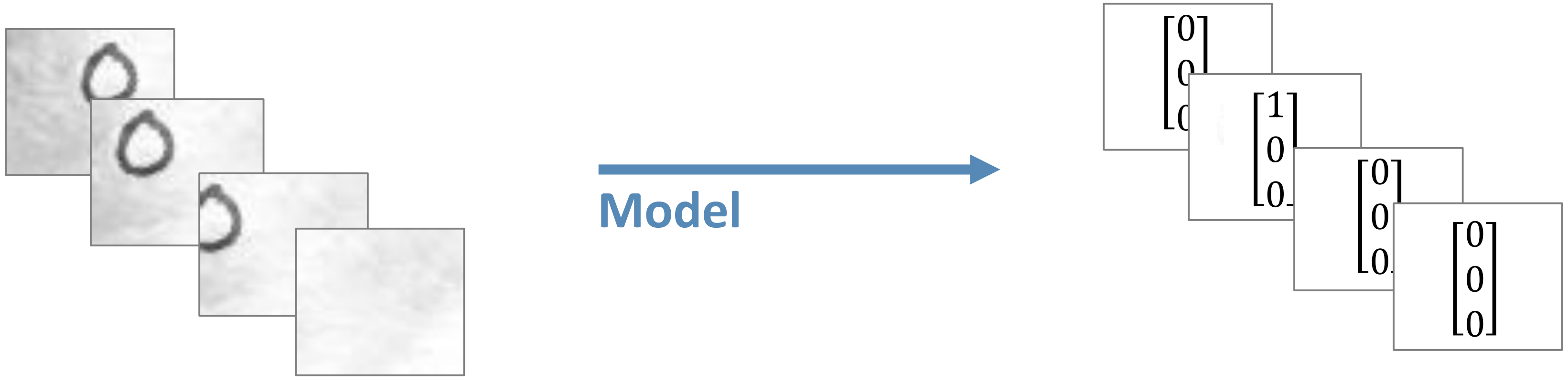


Input

Model

Digit Detection Experiment

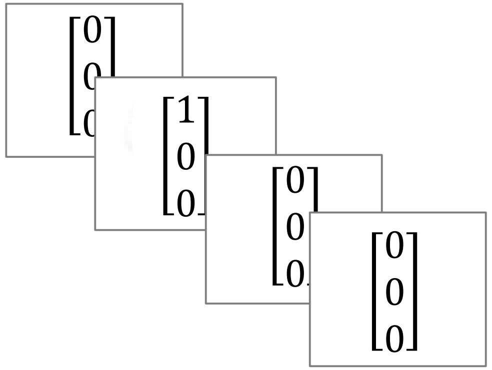
Main Idea



Input **Predictions**

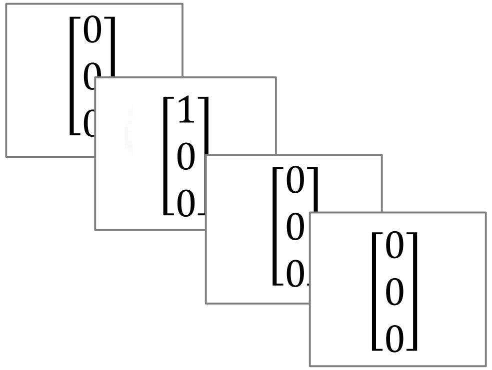
Digit Detection Experiment

Main Idea



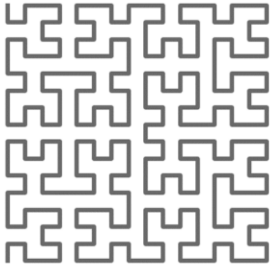
Predictions

DIGIT DETECTION EXPERIMENT
Main Idea



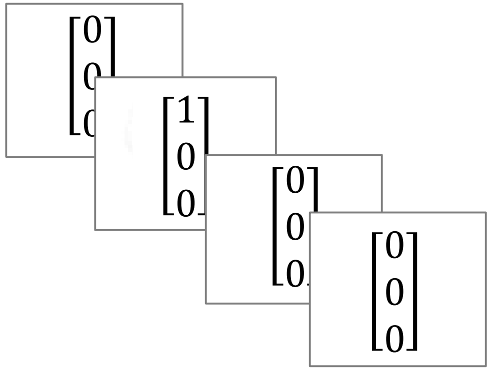
Predictions

Hilbert



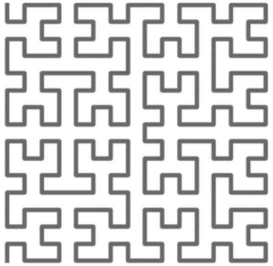
Space-filling curve

DIGIT DETECTION EXPERIMENT
Main Idea



Predictions

Hilbert



Space-filling curve

Object Detection

Digit Detection Experiment
Representations

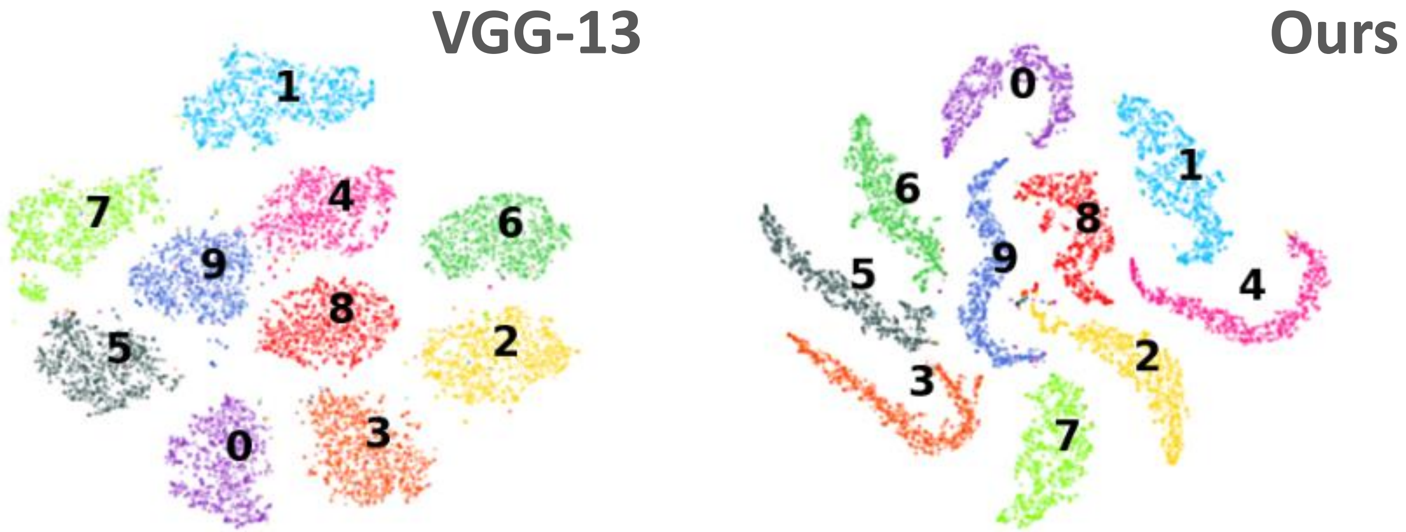
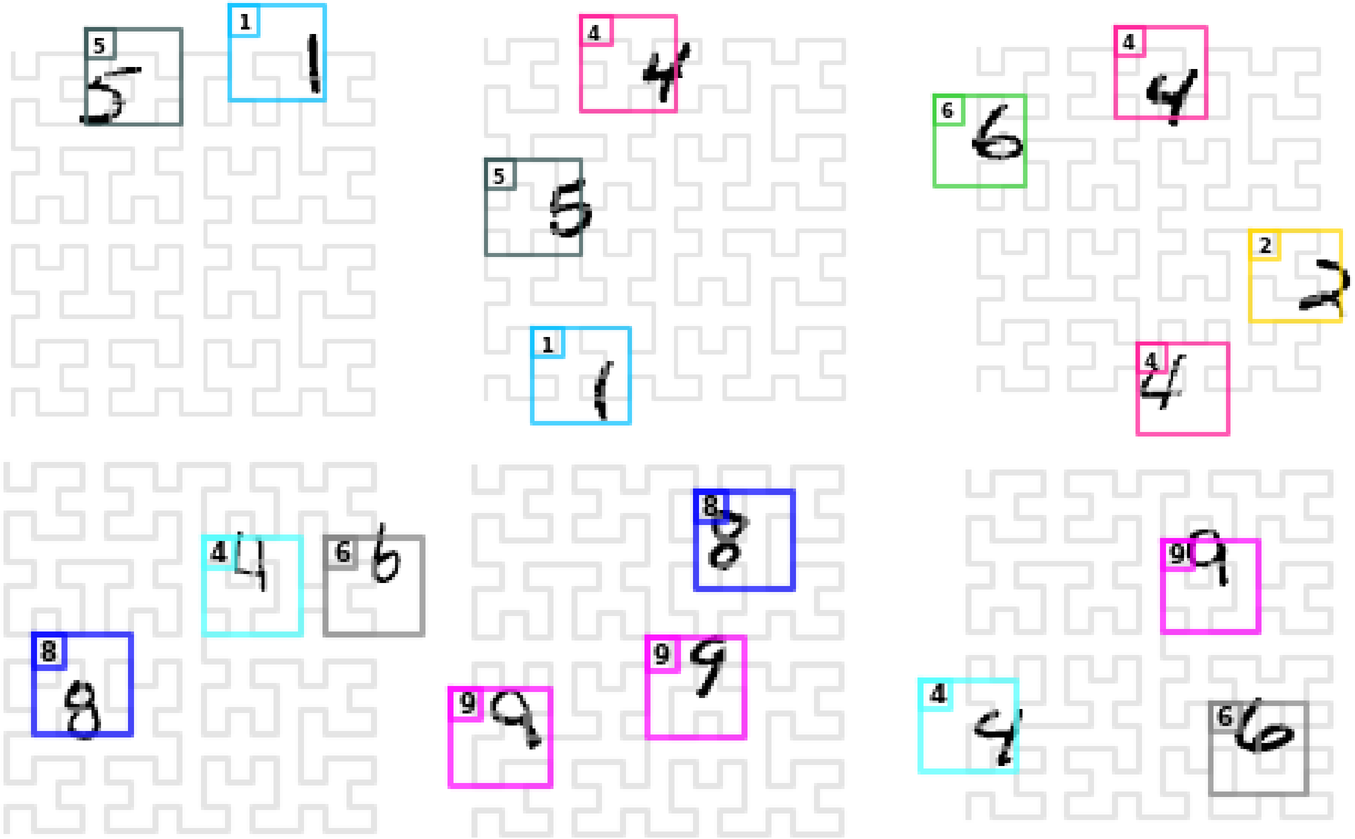


Figure 4. Digit Representations. Comparison of t-SNE digit feature representations resulting from the *fully*-supervised VGG-13 architecture (left) and from our *weakly*-supervised approach (right).

Digit Detection Experiment

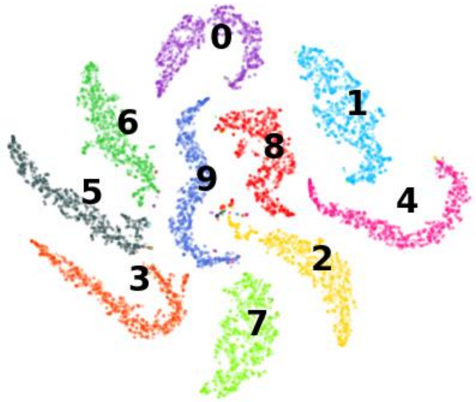
Detection Performance

Mean absolute distance between true and estimated bounding box centers: 9:04 pixels
(approx. step size of the space filling curve)



Digit Detection Experiment
Conclusion

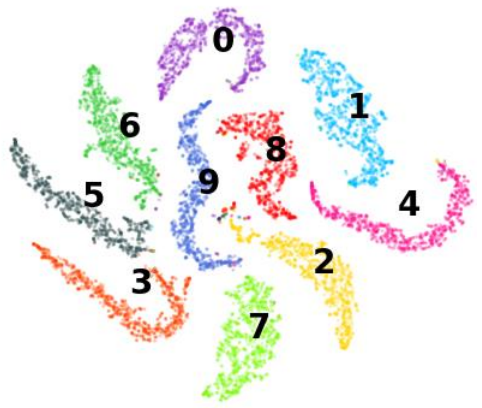
The model learnt:



DIGIT DETECTION EXPERIMENT
Conclusion

The model learnt:

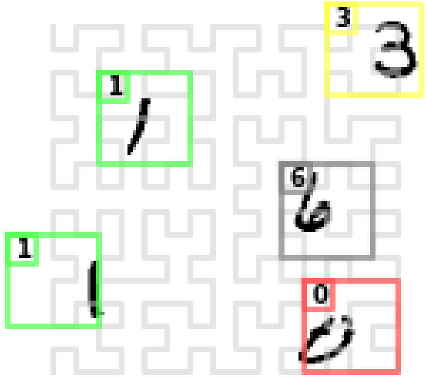
1. Feature representation



Digit Detection Experiment
Conclusion

The model learnt:

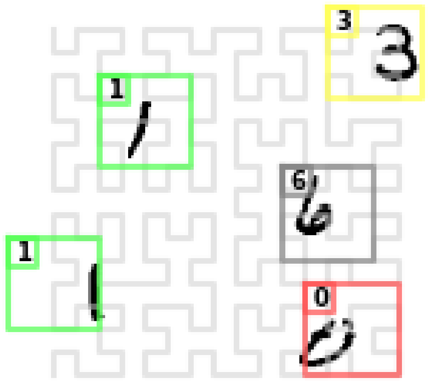
1. Feature representation
2. Space-mapping



Digit Detection Experiment
Conclusion

The model learnt:

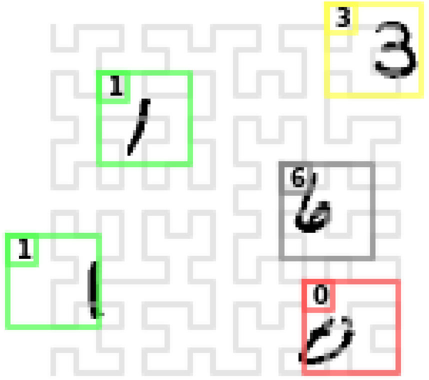
1. Feature representation
2. Space-mapping
3. Object detection



DIGIT DETECTION EXPERIMENT
Conclusion

The model learnt:

1. Feature representation
2. Space-mapping
3. Object detection



Using only occurrence counts as training labels

CONCLUSION

This work shows that implicit model constraints can be used to ensure that accurate localization emerges as a byproduct of learning to count occurrences.

Conclusion

This work shows that implicit model constraints can be used to ensure that accurate localization emerges as a byproduct of learning to count occurrences.

Competitive results against fully-supervised state-of-the-art models.

Questions?

Poster #255