Exemplar Auditing for Multi-Label Biomedical Text Classification

Allen Schmaltz
PhD (Computer Science)

Department of Epidemiology, Harvard University

April 27, 2020

About

- I am an AI/Natural Language Processing (NLP) researcher working in the area of Epidemiology and Public Health at Harvard, with Dr. Andrew Beam.

- My current research focuses on building out the `“exemplar auditing”` framework for AI/ML/data analysis, which I introduce here.

- We are also actively working on medical QA—that is an intermediate point toward the long-term (aspirational) goal of more general automated medical reasoning.

Overview

- This is a chalk talk without chalk…on video…
    - I'll post these slides after the talk in case you want to step through the worked examples again.
- I'm going to build up intuition for the approach, and hopefully, with that intuition you can apply it to your own problems/tasks.
- To build up intuition, we'll first need some machinery that you maybe haven't seen before to produce local feature detections from document level labels from a convolutional neural network (CNN).
- I'll introduce the approach and we'll walk through an example, and then we can open for QA.

Motivation

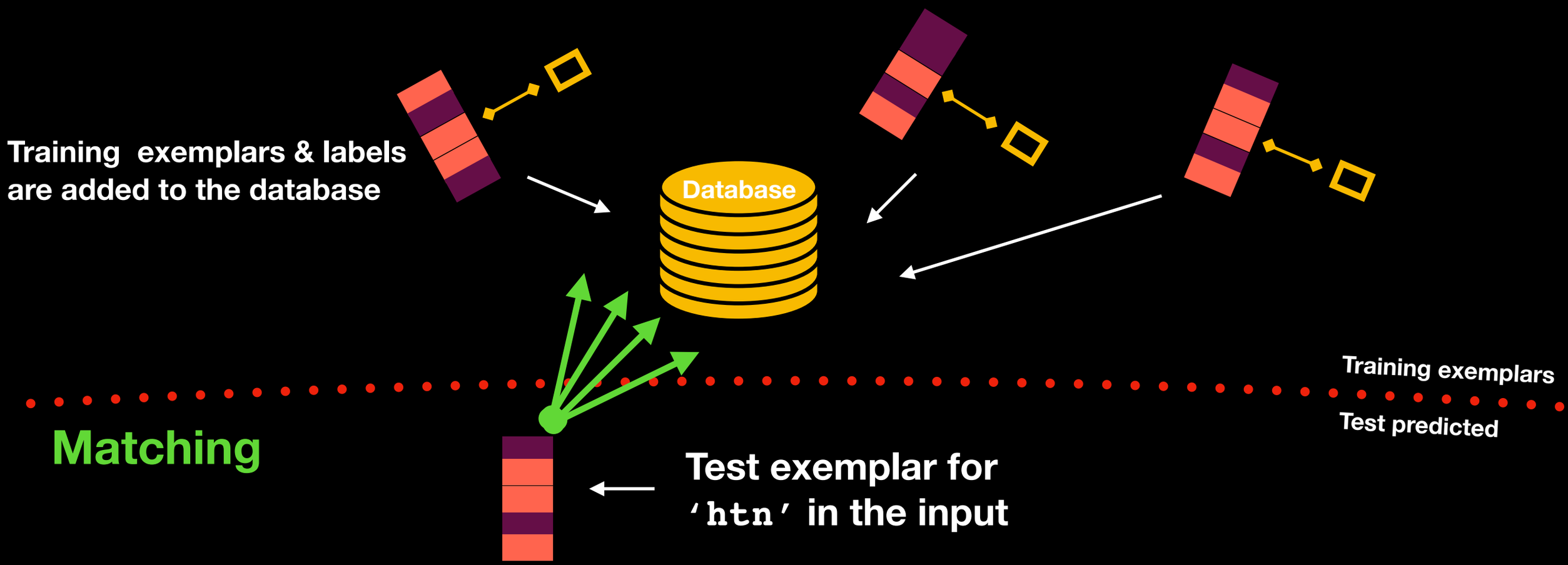
- Many practical applications of AI in medicine must make use of coarse-grained labels, even though the identification of fine-grained features is the desired goal.

- E.g., we seek particular patient characteristics for a given disease/ procedure, but only have coarse-grained billing labels
    - Relevant when input is large-scale, high-dimensional

- Additionally, to the extent we do uncover such fine-grained patterns, we need (some semblance of) verification based on the labels we do have

- Our approach begins to address these issues
    - Important connections (similarities and differences) with metric learning, prototypical networks, matching networks, and more

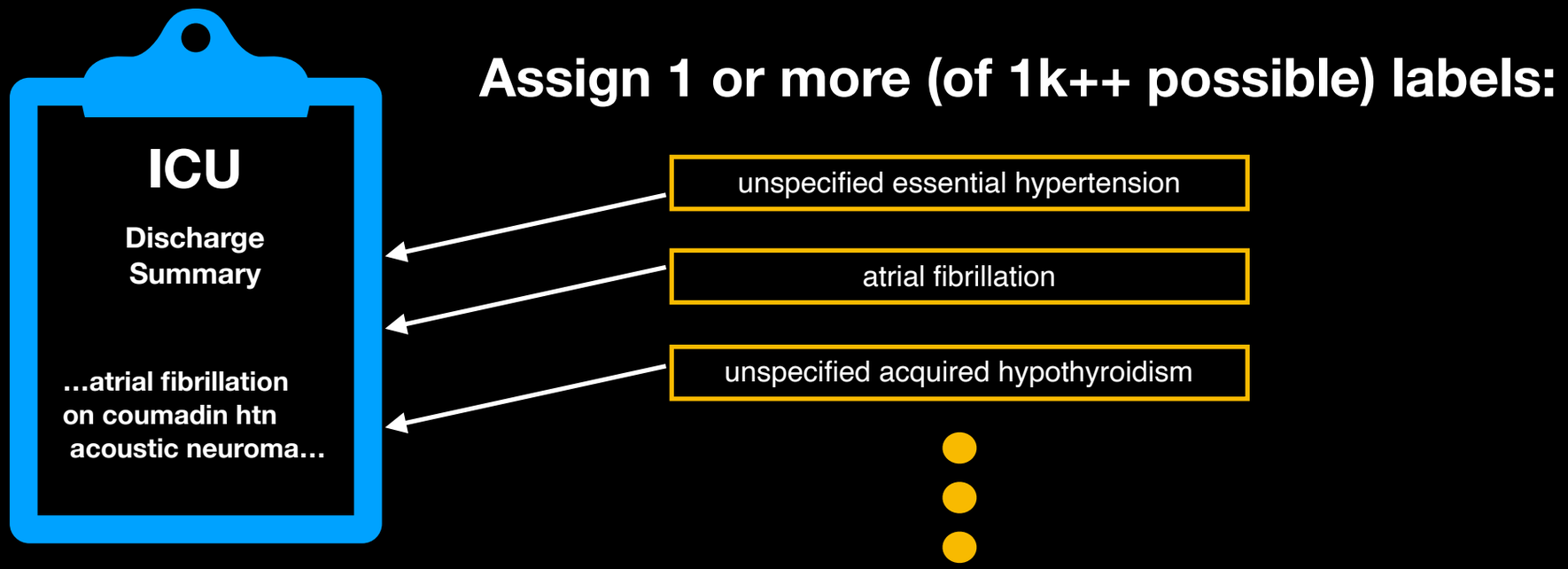
Preview



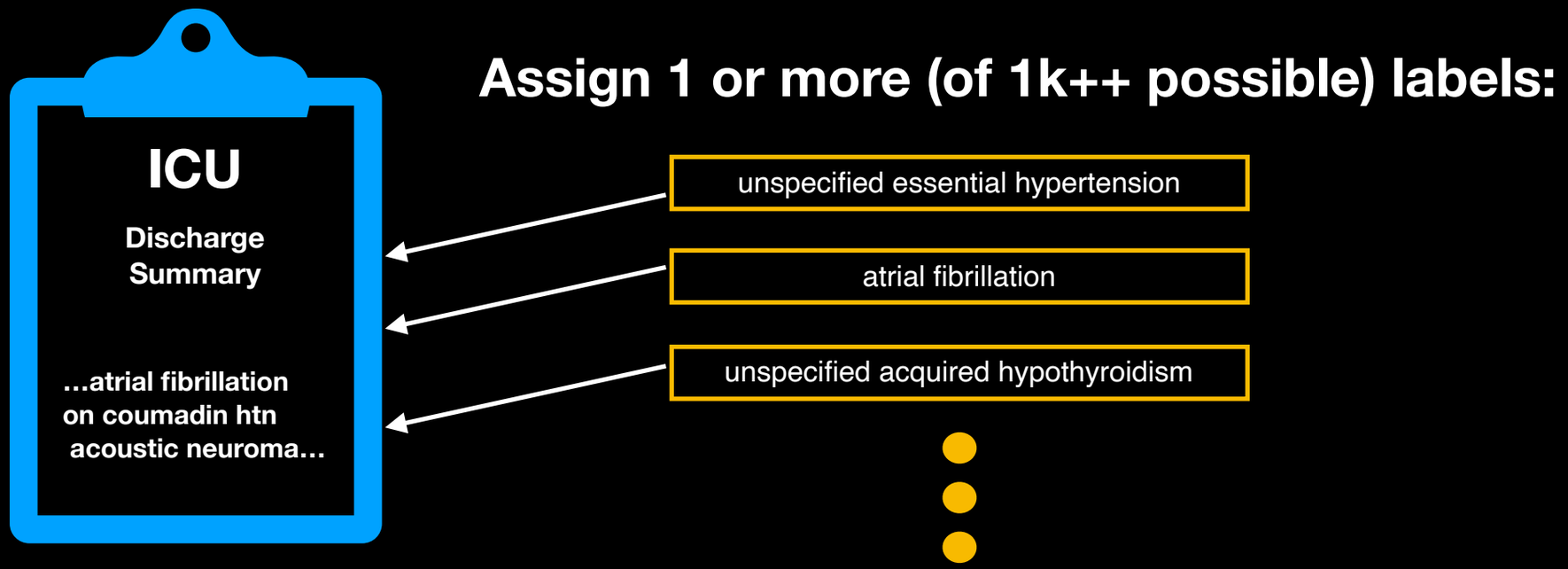
- Eventually, we're going to get here, but we have some background to build up first to understand this…

Task: Multi-label Classification

Task: Multi-label Classification

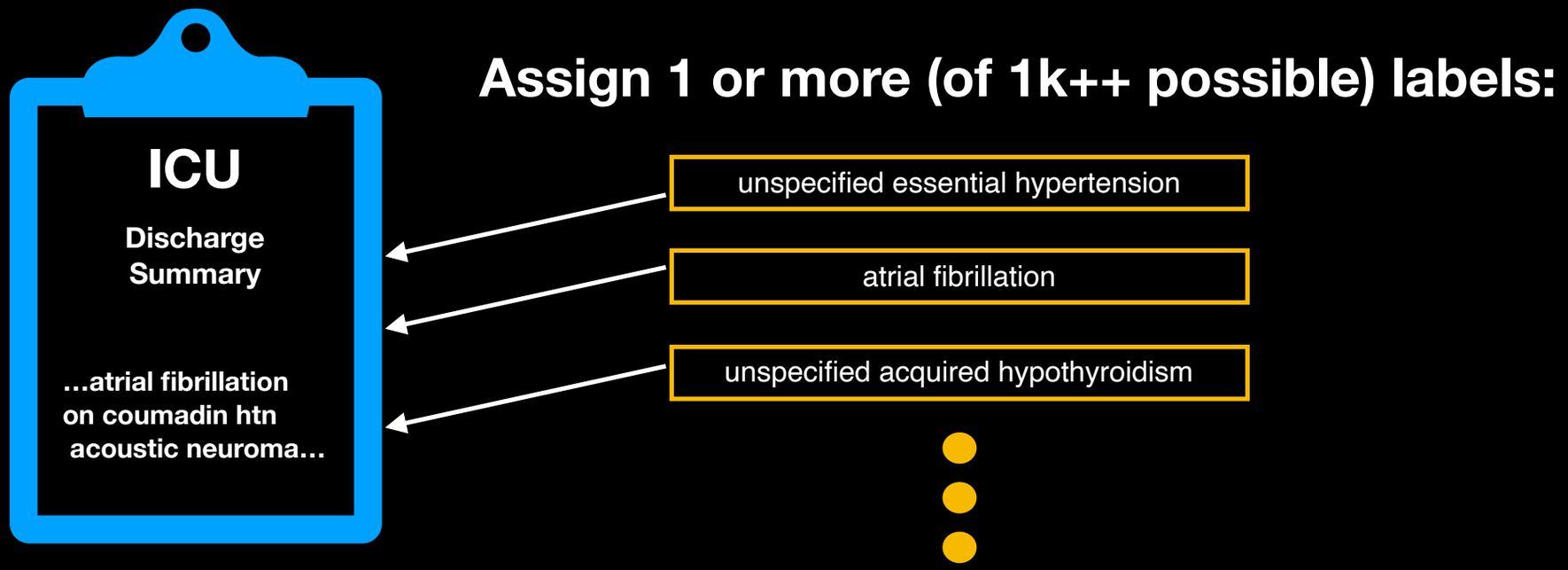


Task: Multi-label Classification



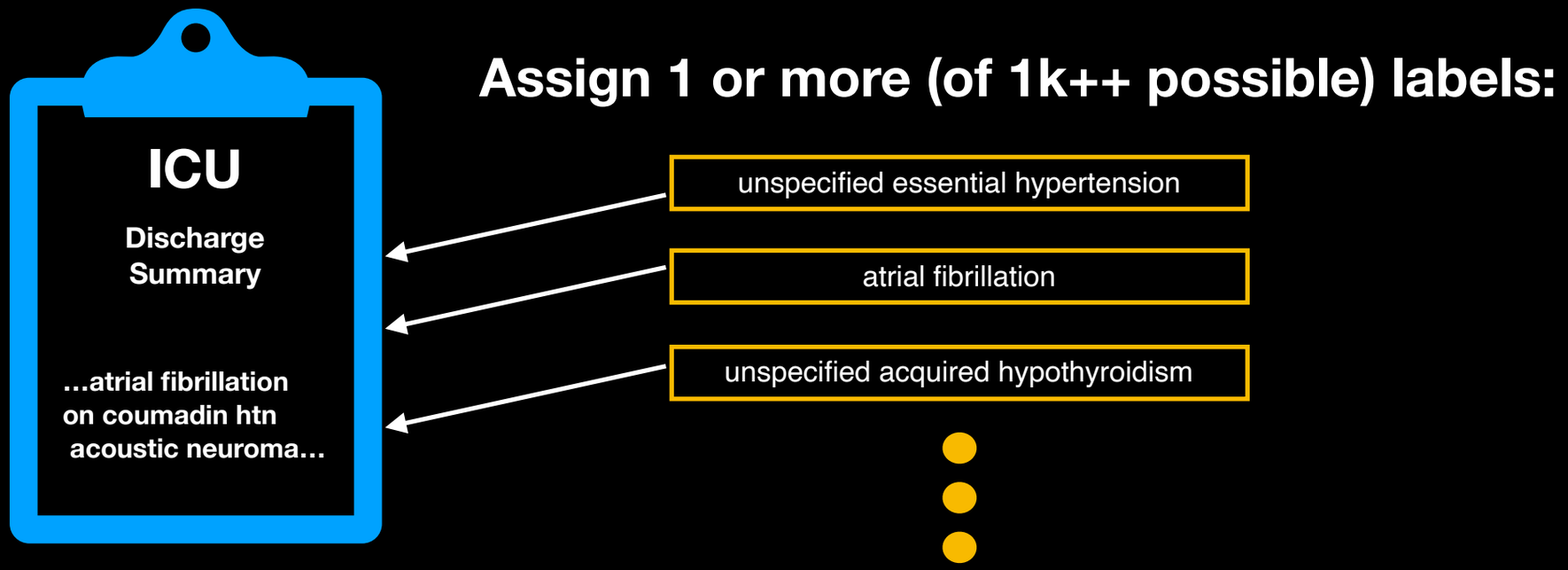
- We aim to assign ICD-9 codes (diseases, procedures) to an electronic health record (EHR) text report

Task: Multi-label Classification



- We aim to assign ICD-9 codes (diseases, procedures) to an electronic health record (EHR) text report
- Hard
  - Multiple, varying numbers of labels per document, from 8000+ possible labels
  - Documents are long and noisy
  - Potential ambiguity in labels themselves

Task: Multi-label Classification



- We aim to assign ICD-9 codes (diseases, procedures) to an electronic health record (EHR) text report
- Hard
  - Multiple, varying numbers of labels per document, from 8000+ possible labels
  - Documents are long and noisy
  - Potential ambiguity in labels themselves
- Importantly, we want to introspect the model predictions—to assess the model and data, and to serve as a possible decision support tool

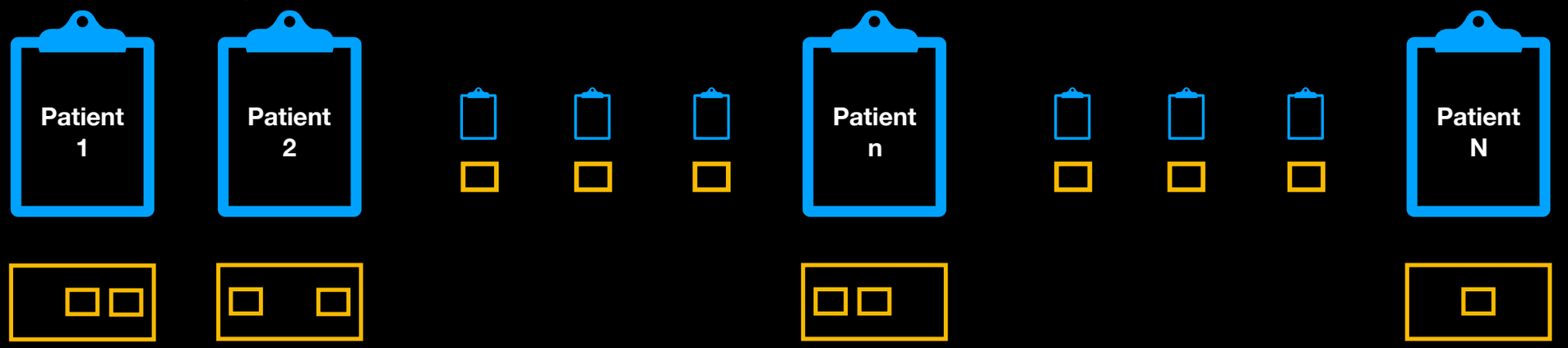
Task: Setup

Just to be clear…
before we make it confusing…

Task: Setup

Just to be clear…
before we make it confusing…

For Training

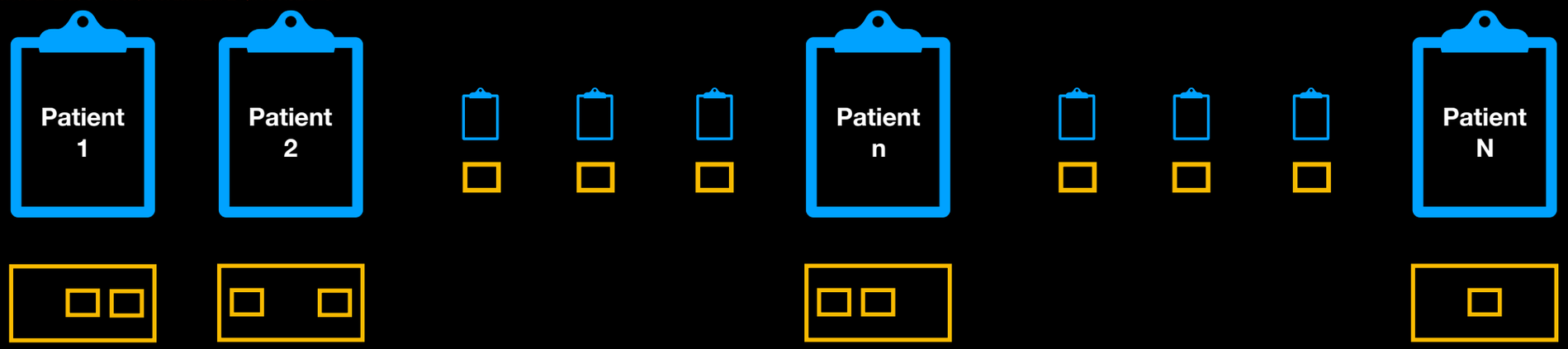


Ground-truth Label Sets

Task: Setup

Just to be clear…
before we make it confusing…

For Training

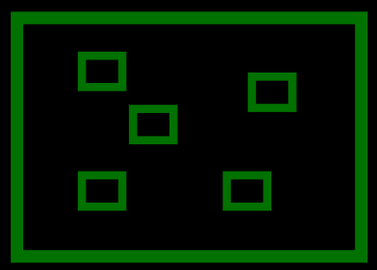


Ground-truth Label Sets

Test/inference Time

New patient ICU discharge summary not seen in training

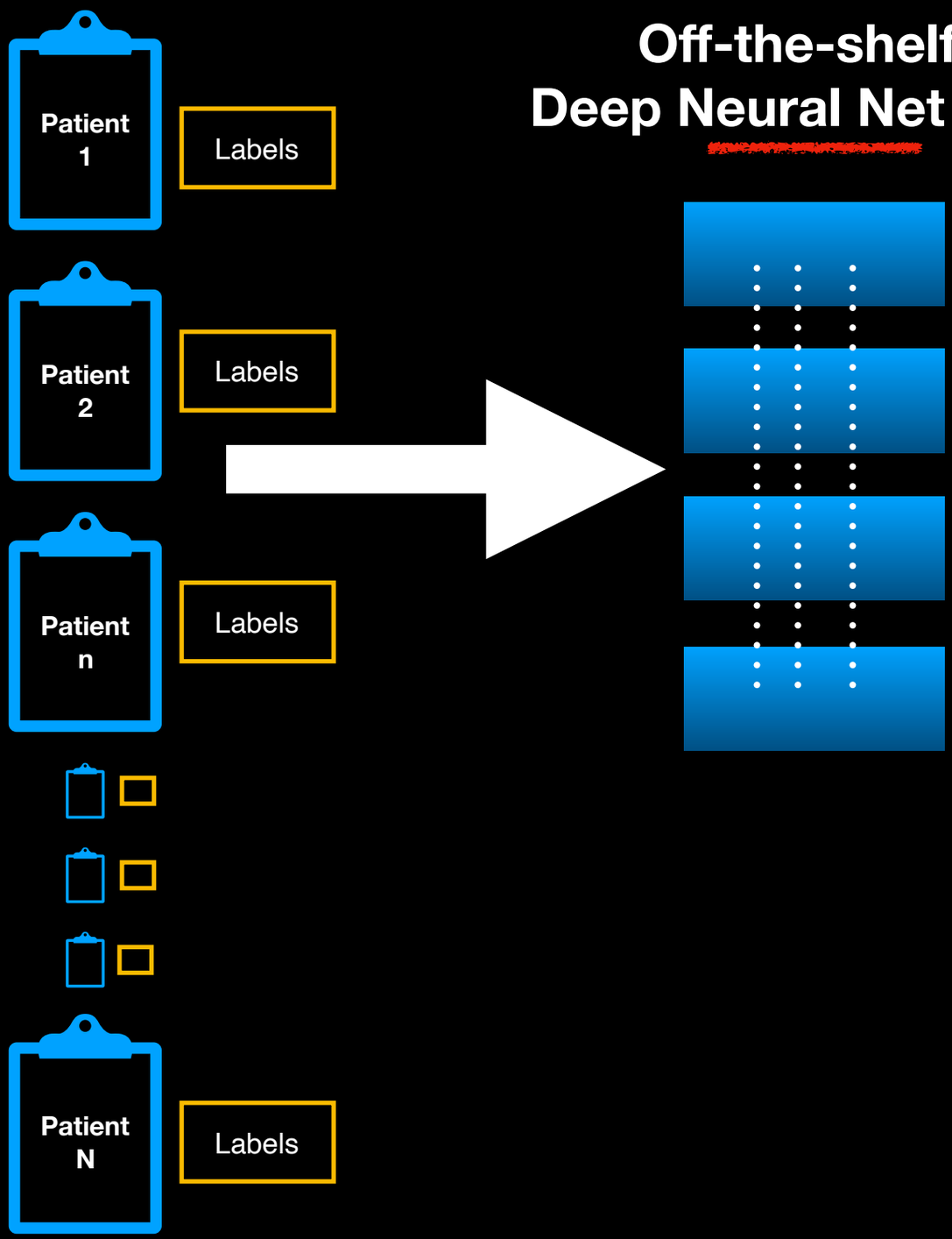
Predicted Label Set



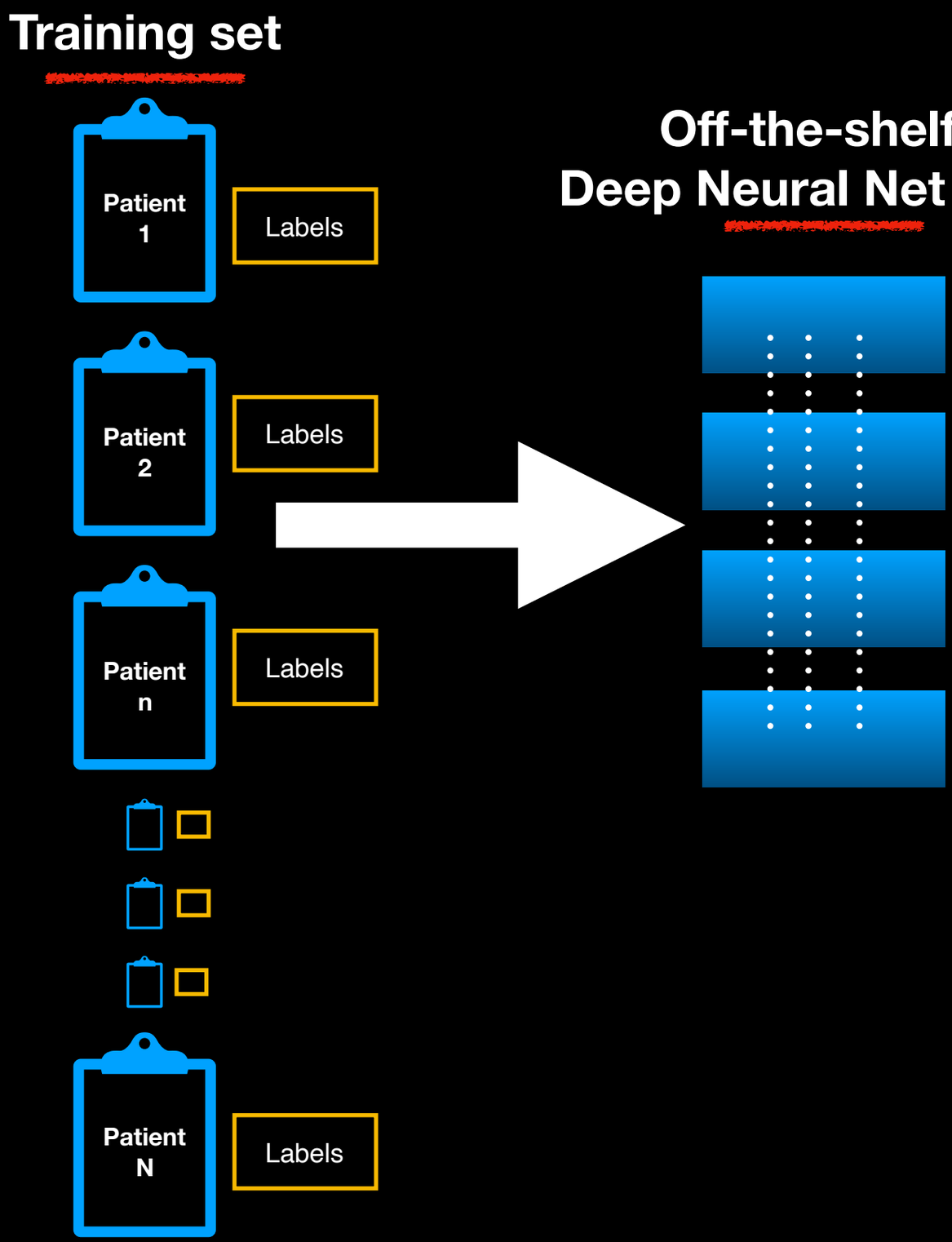
I.e., standard classification setting

Possible Approach #1: Deep Neural Net

Training set



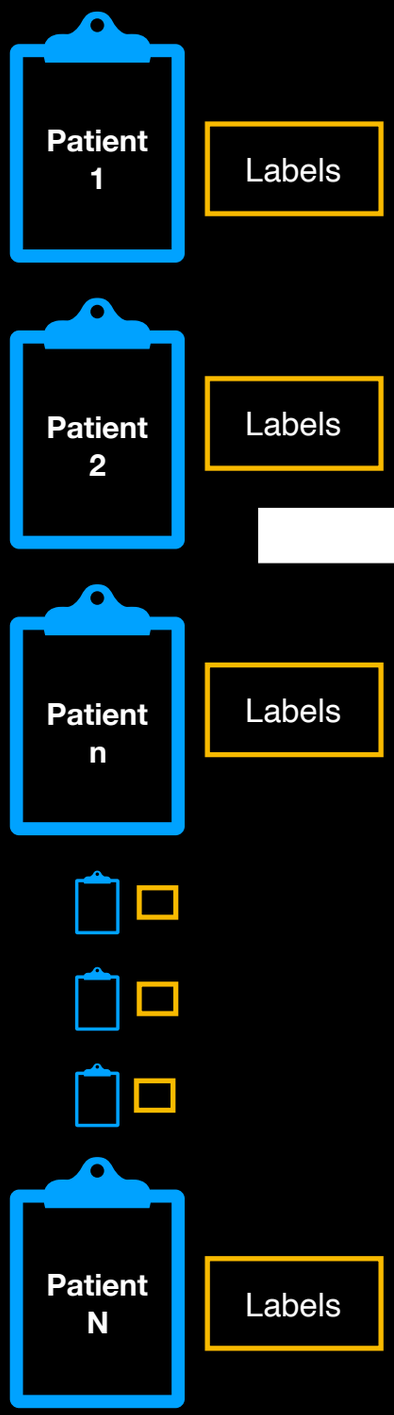
Possible Approach #1: Deep Neural Net



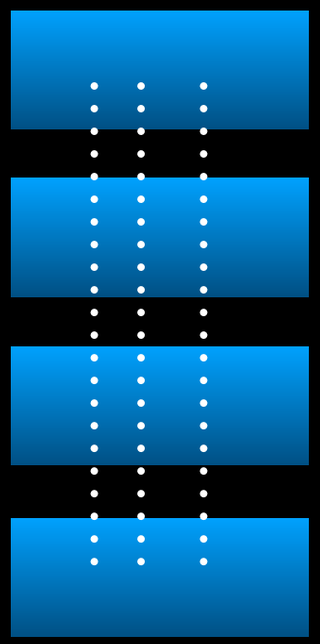
- **Classification effectiveness**

Possible Approach #1: Deep Neural Net

Training set



Off-the-shelf Deep Neural Net XYZ

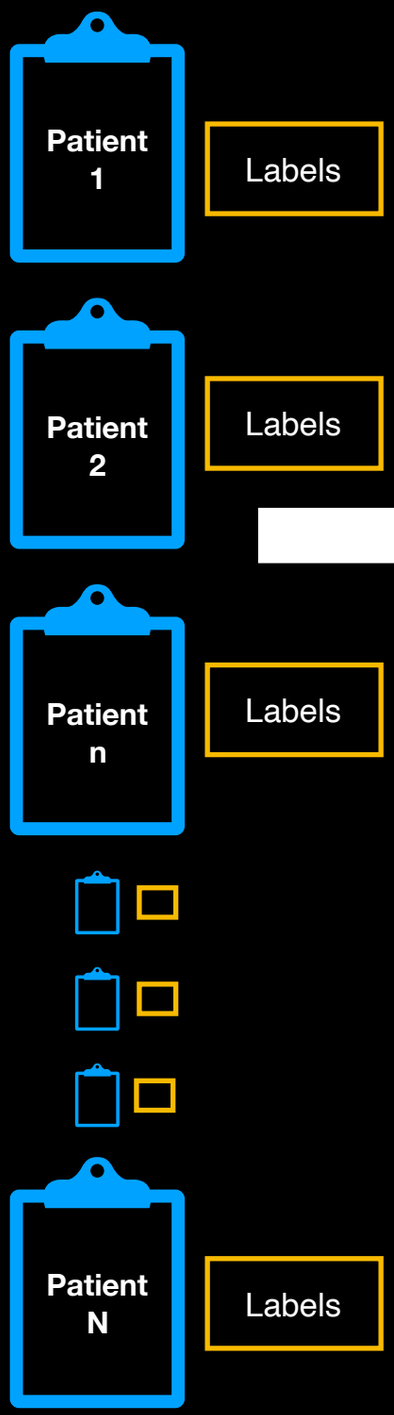


Lottery Tickets

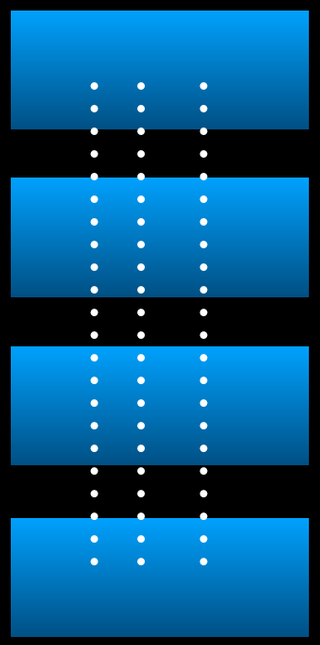
- **Classification effectiveness**

Possible Approach #1: Deep Neural Net

Training set



Off-the-shelf Deep Neural Net XYZ

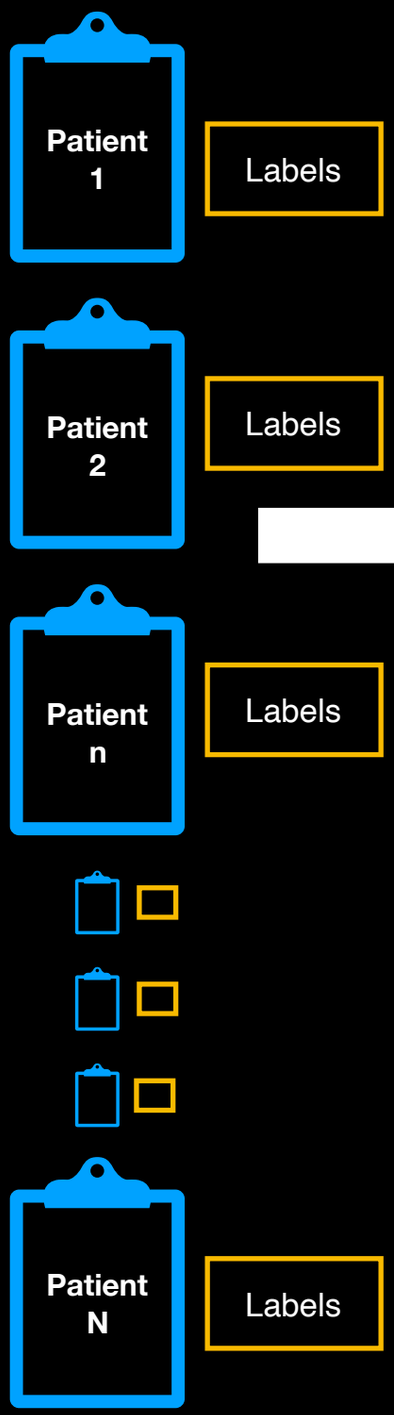


Lottery Tickets

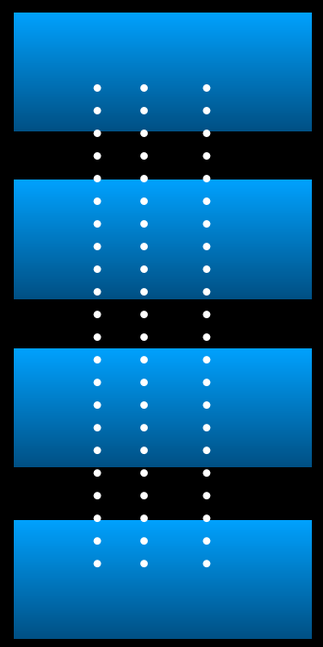
- **Classification effectiveness** ✓

Possible Approach #1: Deep Neural Net

Training set



Off-the-shelf Deep Neural Net XYZ

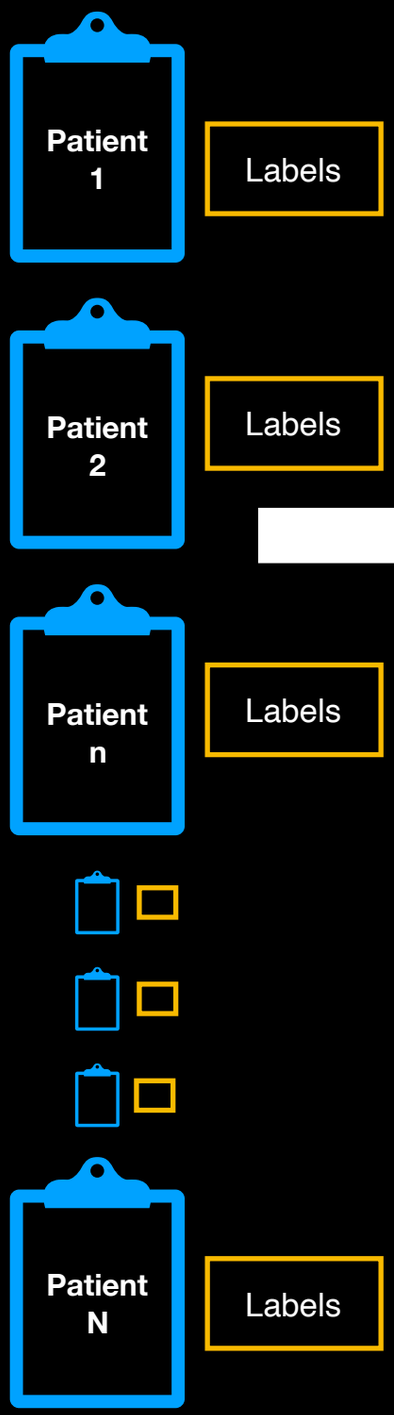


Lottery Tickets

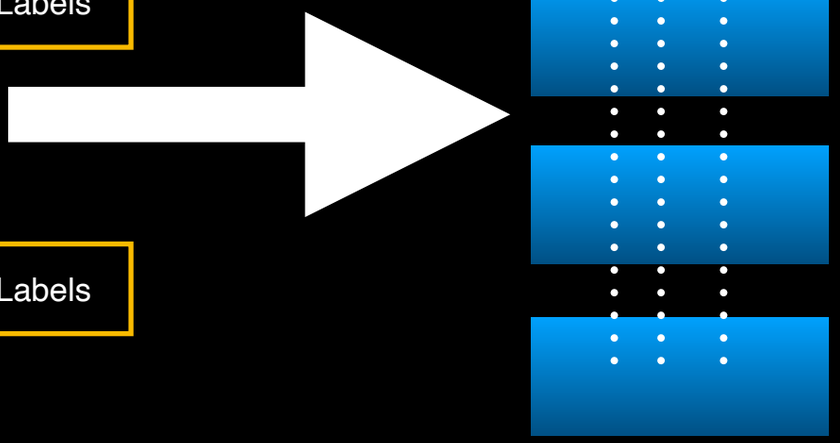
- **Classification effectiveness** ✓
- **Interpretability**

Possible Approach #1: Deep Neural Net

Training set



Off-the-shelf Deep Neural Net XYZ

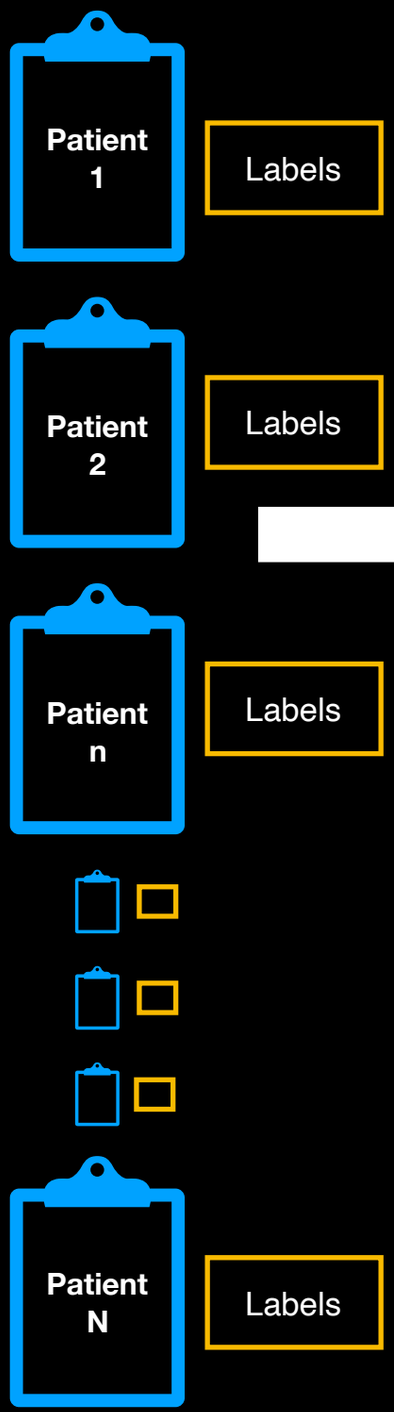


Lottery Tickets

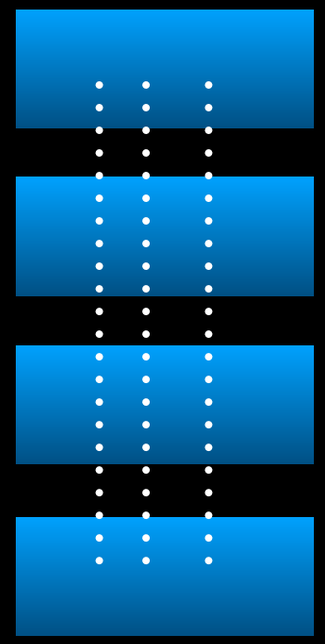
- **Classification effectiveness** ✓
- **Interpretability** ✗

Possible Approach #1: Deep Neural Net

Training set



Off-the-shelf Deep Neural Net XYZ

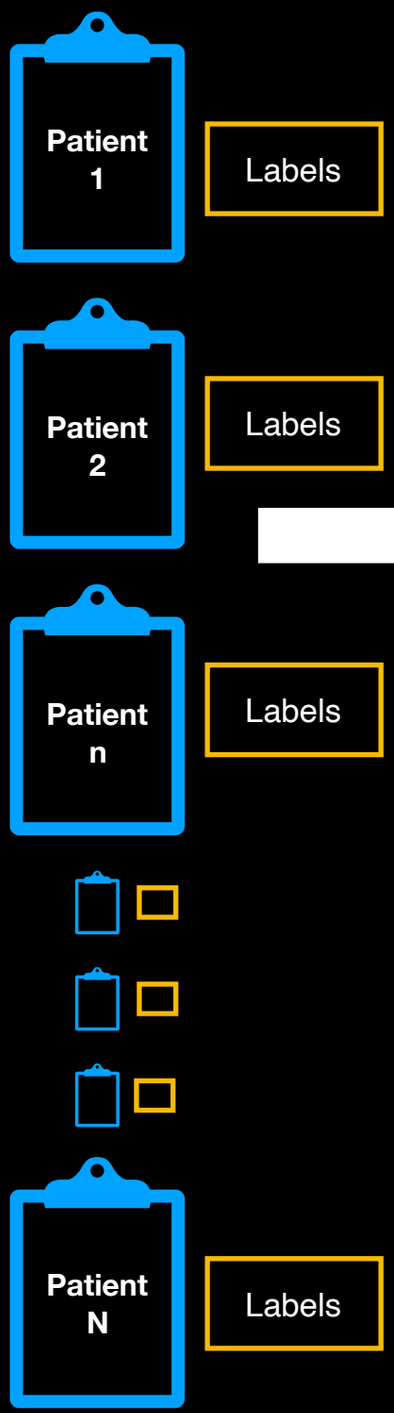


Lottery Tickets

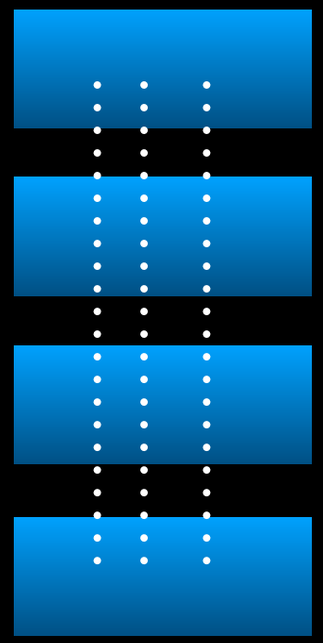
- **Classification effectiveness** ✓
- **Interpretability** ✗
- **Aid for decision making at lower granularities/resolutions**

Possible Approach #1: Deep Neural Net

Training set



Off-the-shelf Deep Neural Net XYZ

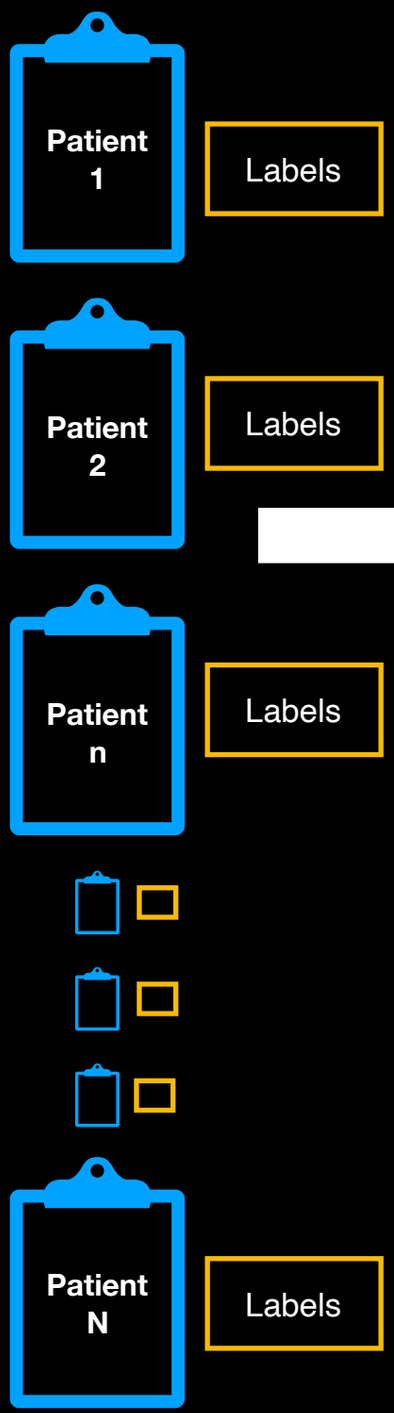


Lottery Tickets

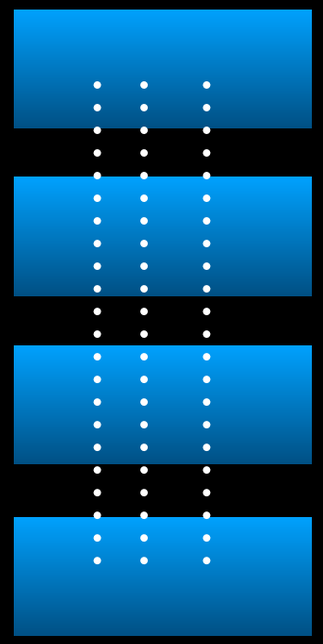
- **Classification effectiveness** ✓
- **Interpretability** ✗
- **Aid for decision making at lower granularities/resolutions** ✗

Possible Approach #1: Deep Neural Net

Training set



Off-the-shelf Deep Neural Net XYZ

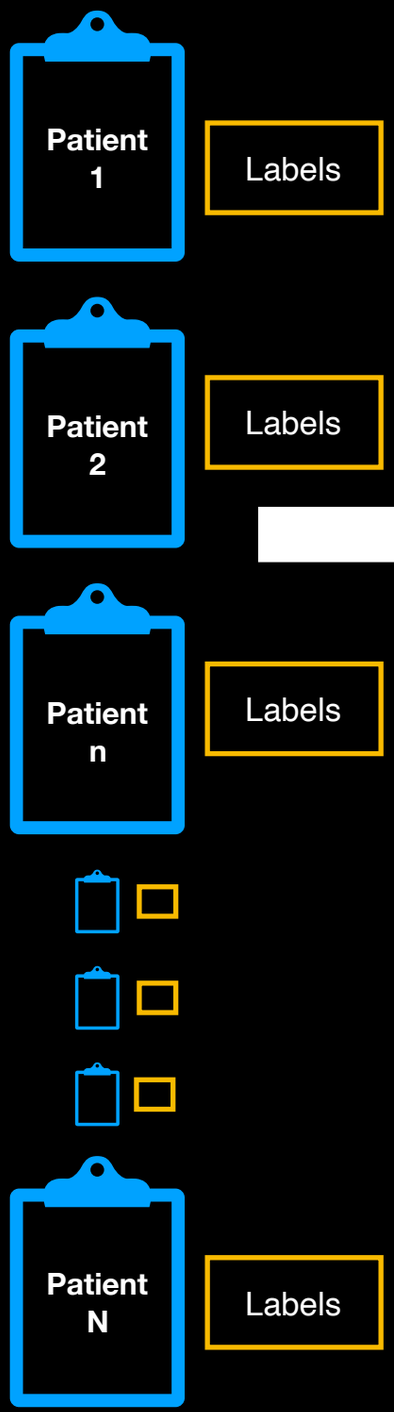


Lottery Tickets

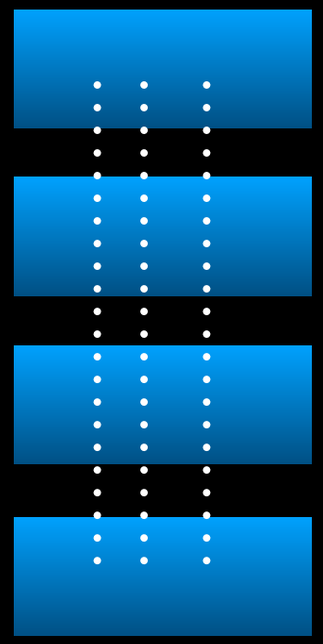
- **Classification effectiveness** ✓
- **Interpretability** ✗
- **Aid for decision making at lower granularities/resolutions** ✗
- **Verify findings with ground-truth training labels**

Possible Approach #1: Deep Neural Net

Training set



Off-the-shelf Deep Neural Net XYZ

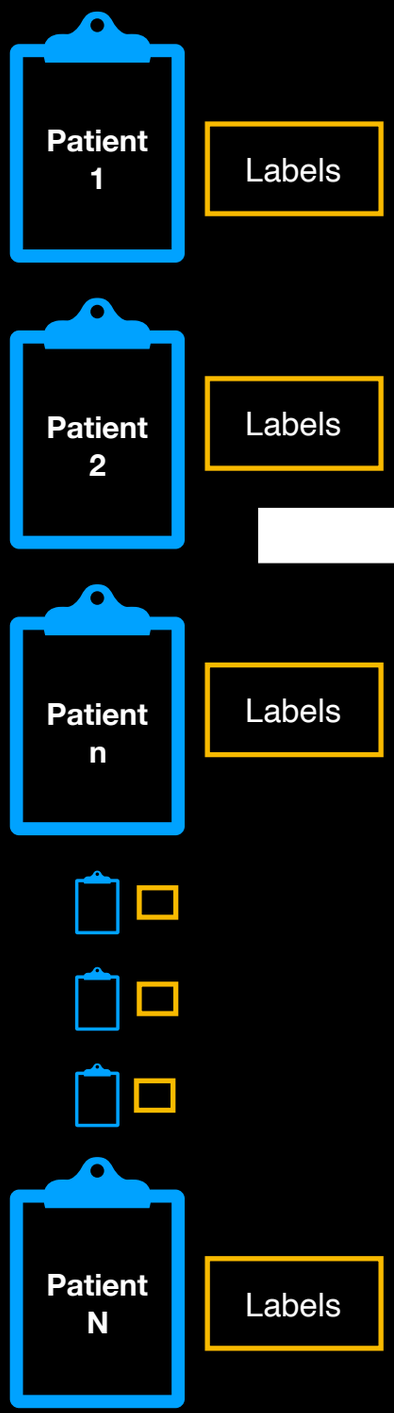


Lottery Tickets

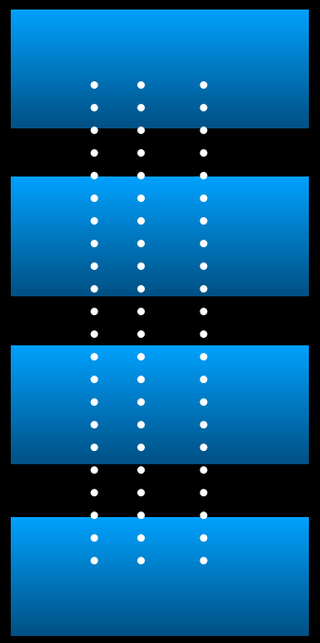
- **Classification effectiveness** ✓
- **Interpretability** ✗
- **Aid for decision making at lower granularities/resolutions** ✗
- **Verify findings with ground-truth training labels** ✗

Possible Approach #1: Deep Neural Net

Training set



Off-the-shelf Deep Neural Net XYZ

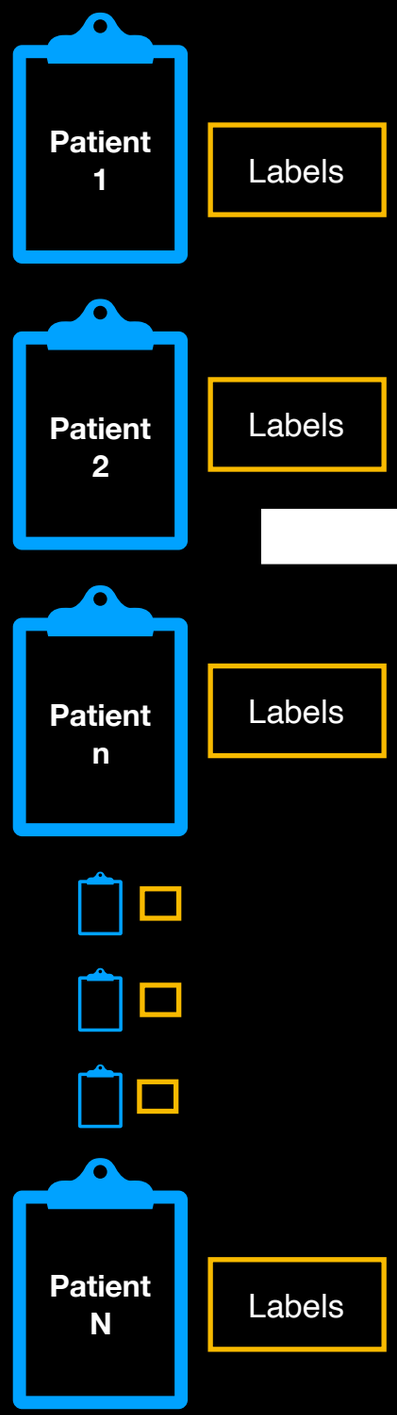


Lottery Tickets

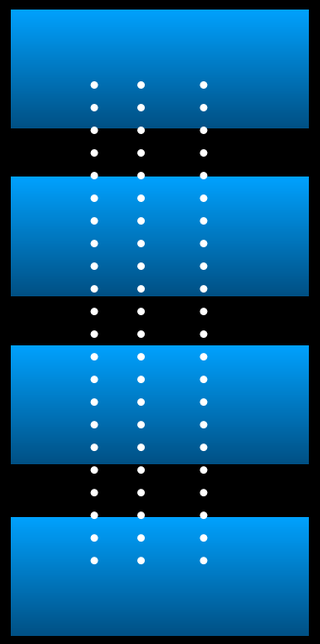
- **Classification effectiveness** ✓
- **Interpretability** ✗
- **Aid for decision making at lower granularities/resolutions** ✗
- **Verify findings with ground-truth training labels** ✗
- **Parameter identification**

Possible Approach #1: Deep Neural Net

Training set



Off-the-shelf Deep Neural Net XYZ

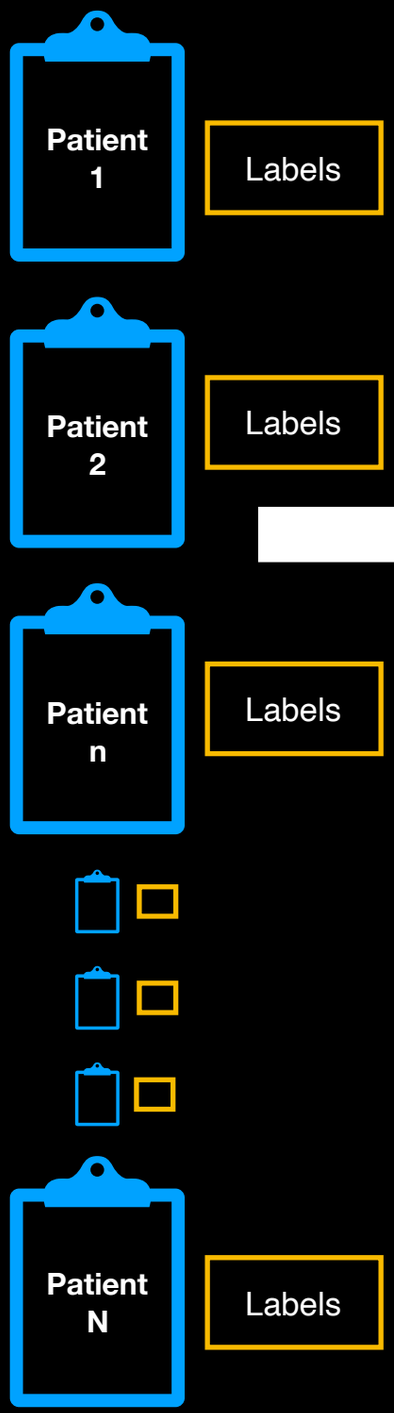


Lottery Tickets

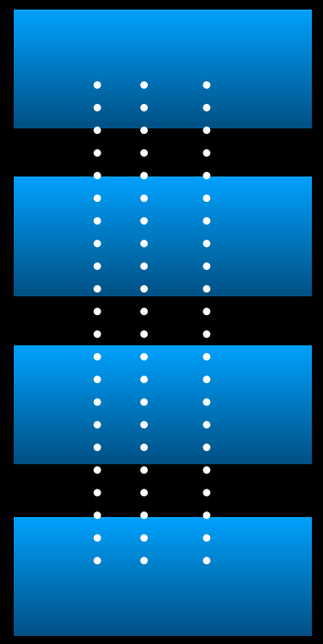
- **Classification effectiveness** ✓
- **Interpretability** ✗
- **Aid for decision making at lower granularities/resolutions** ✗
- **Verify findings with ground-truth training labels** ✗
- **Parameter identification** ✗

Possible Approach #1: Deep Neural Net

Training set



Off-the-shelf Deep Neural Net XYZ

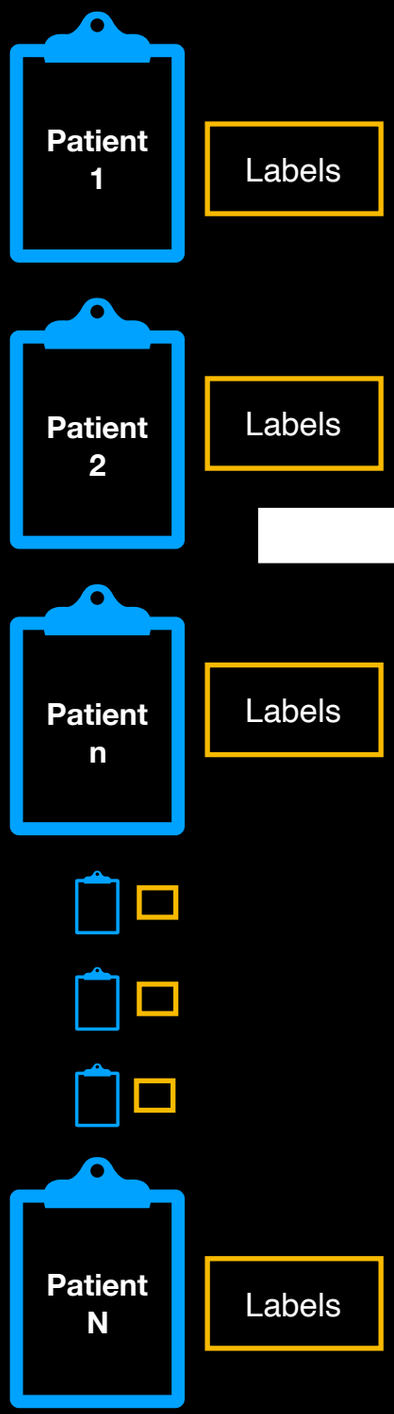


Lottery Tickets

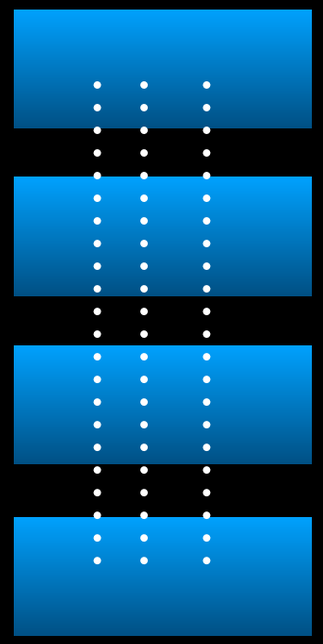
- **Classification effectiveness** ✓
- **Interpretability** ✗
- **Aid for decision making at lower granularities/resolutions** ✗
- **Verify findings with ground-truth training labels** ✗
- **Parameter identification** ✗
- **Robustness to adversarial attacks**

Possible Approach #1: Deep Neural Net

Training set



Off-the-shelf Deep Neural Net XYZ

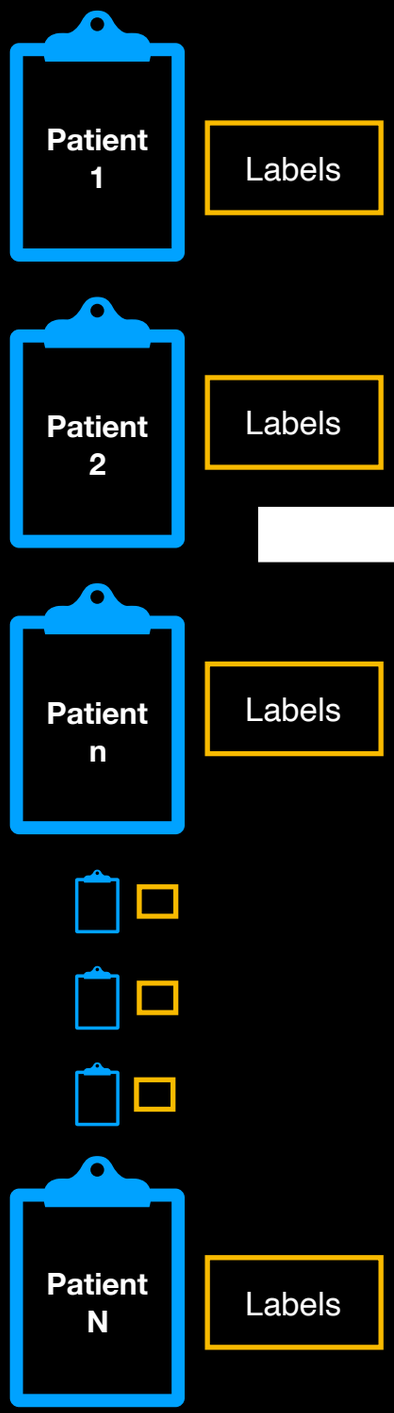


Lottery Tickets

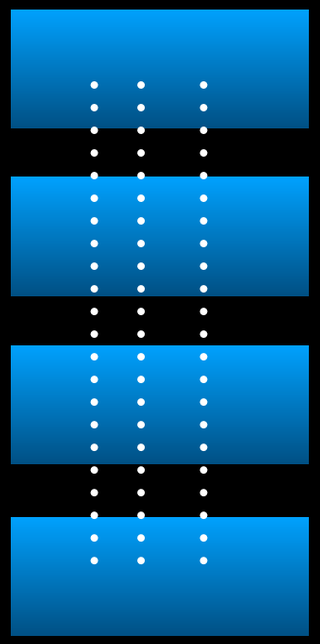
- **Classification effectiveness** ✓
- **Interpretability** ✗
- **Aid for decision making at lower granularities/resolutions** ✗
- **Verify findings with ground-truth training labels** ✗
- **Parameter identification** ✗
- **Robustness to adversarial attacks** ✗

Possible Approach #1: Deep Neural Net

Training set



Off-the-shelf Deep Neural Net XYZ

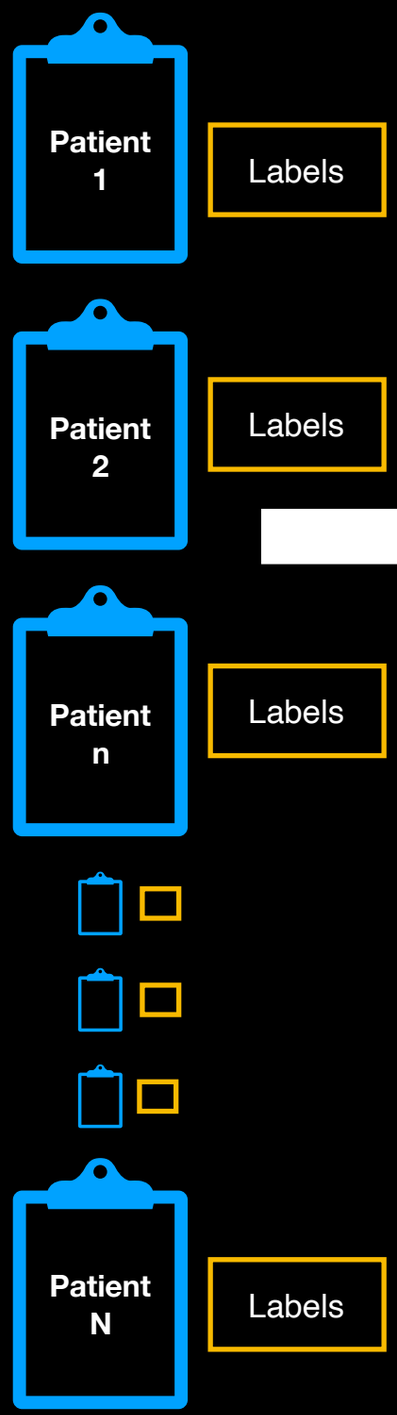


Lottery Tickets

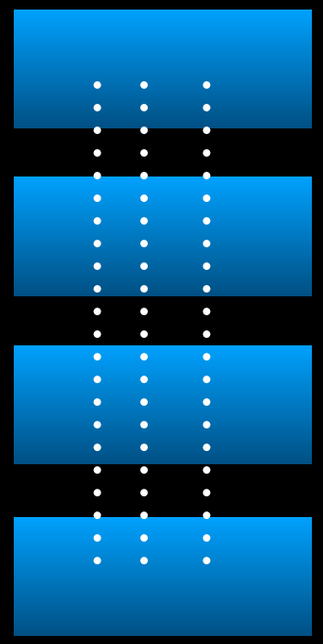
- **Classification effectiveness** ✓
- **Interpretability** ✗
- **Aid for decision making at lower granularities/resolutions** ✗
- **Verify findings with ground-truth training labels** ✗
- **Parameter identification** ✗
- **Robustness to adversarial attacks** ✗
- **Efficiency (space/time)**

Possible Approach #1: Deep Neural Net

Training set



Off-the-shelf Deep Neural Net XYZ

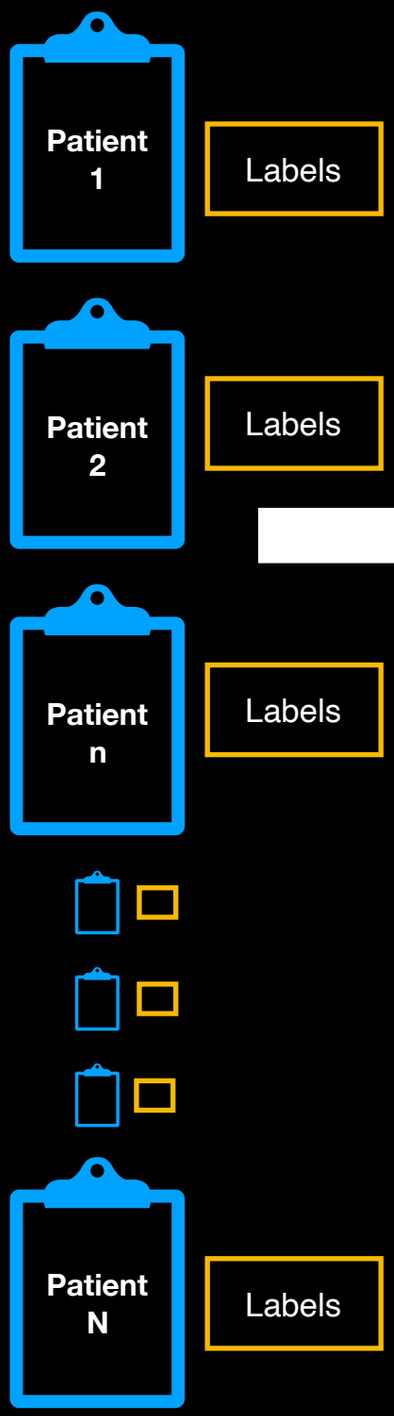


Lottery Tickets

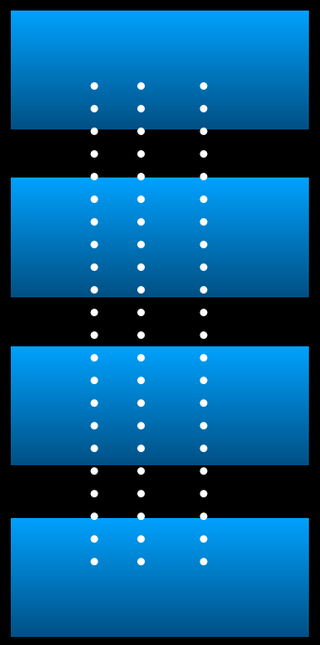
- **Classification effectiveness** ✓
- **Interpretability** ✗
- **Aid for decision making at lower granularities/resolutions** ✗
- **Verify findings with ground-truth training labels** ✗
- **Parameter identification** ✗
- **Robustness to adversarial attacks** ✗
- **Efficiency (space/time)** ?

Possible Approach #2: Neural Net with Attention

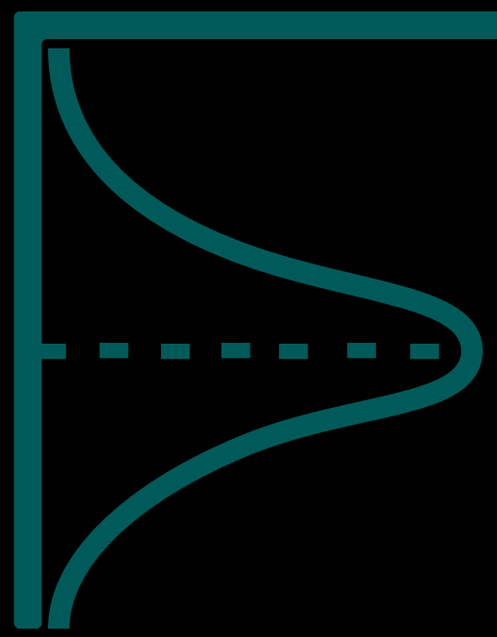
Training set



Off-the-shelf Deep Neural Net XYZ

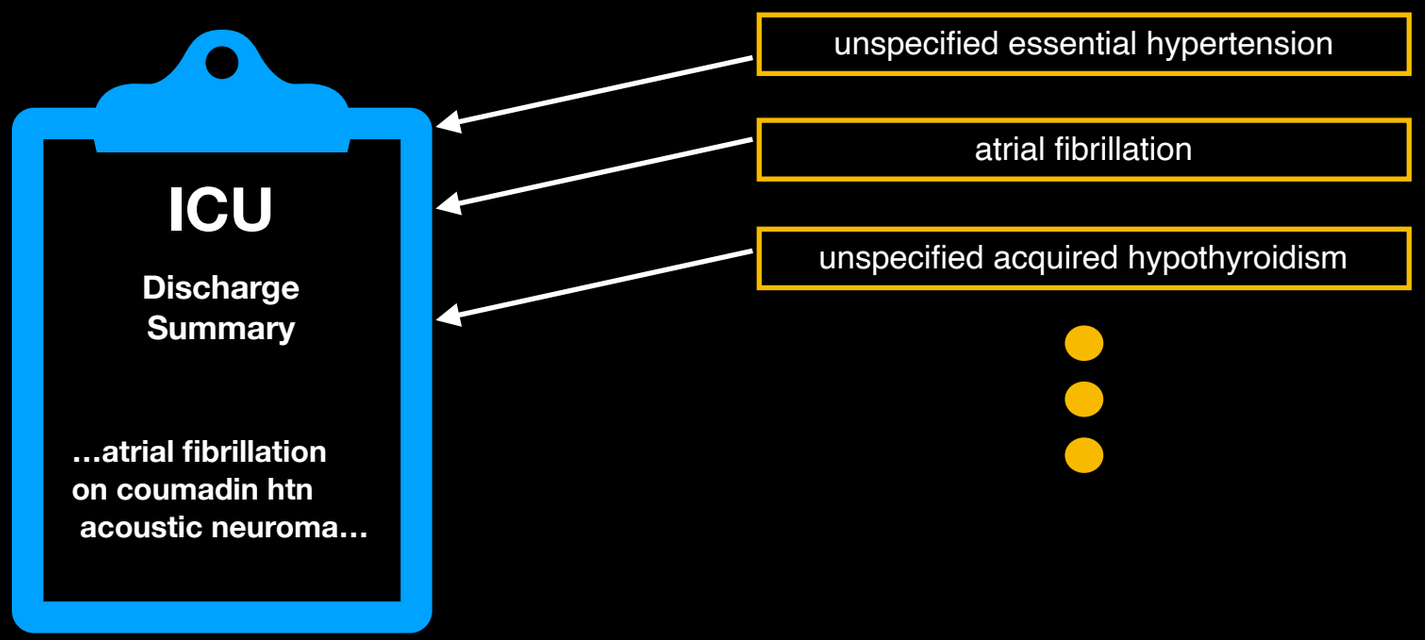


Distribution/scores over hidden states/input/words



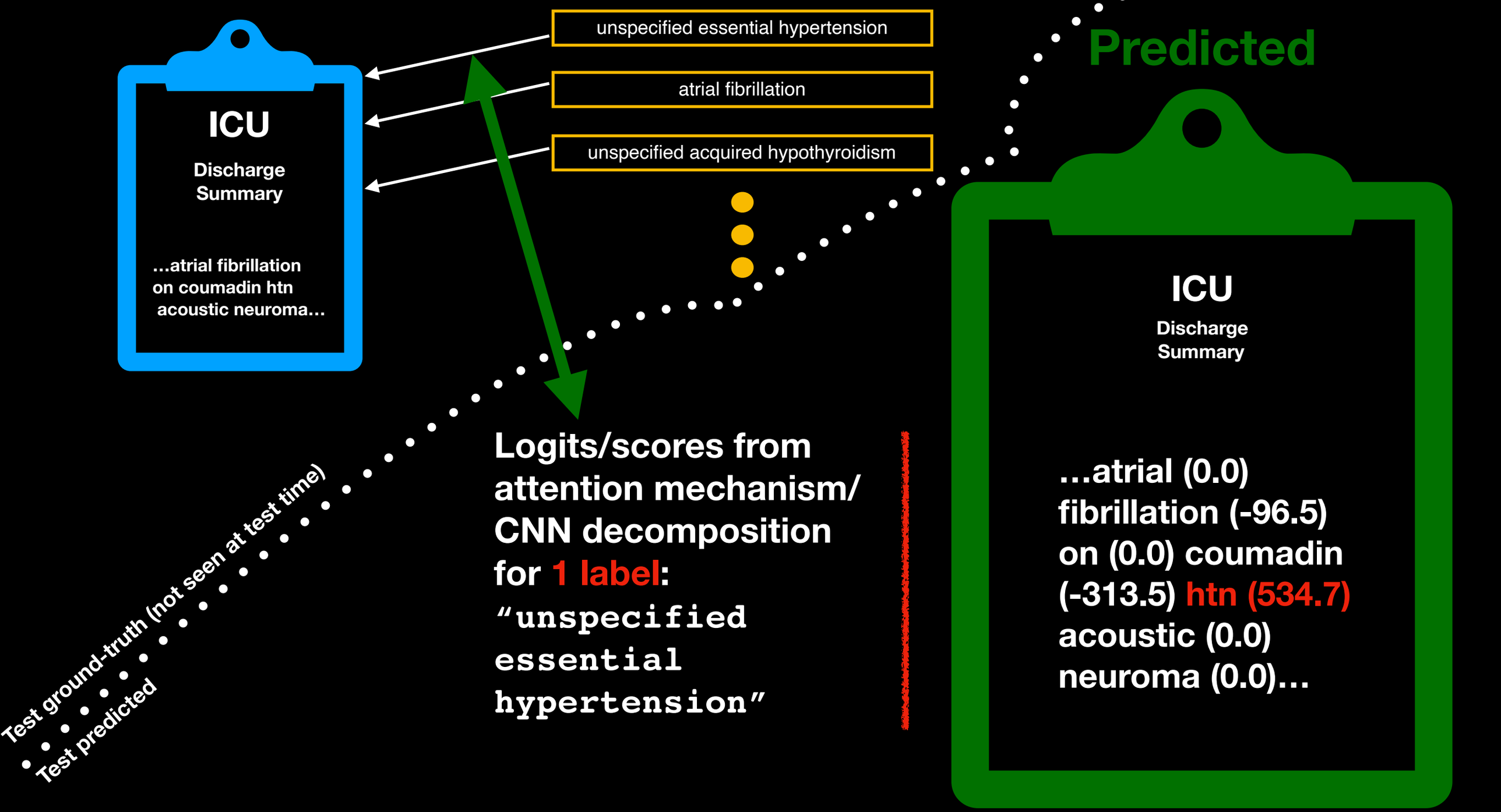
Possible Approach #2: Neural Net with Attention (Cont.)

- Additional layer that relates document labels to words in the EHR (distribution/scores over words *for each label for each EHR*)

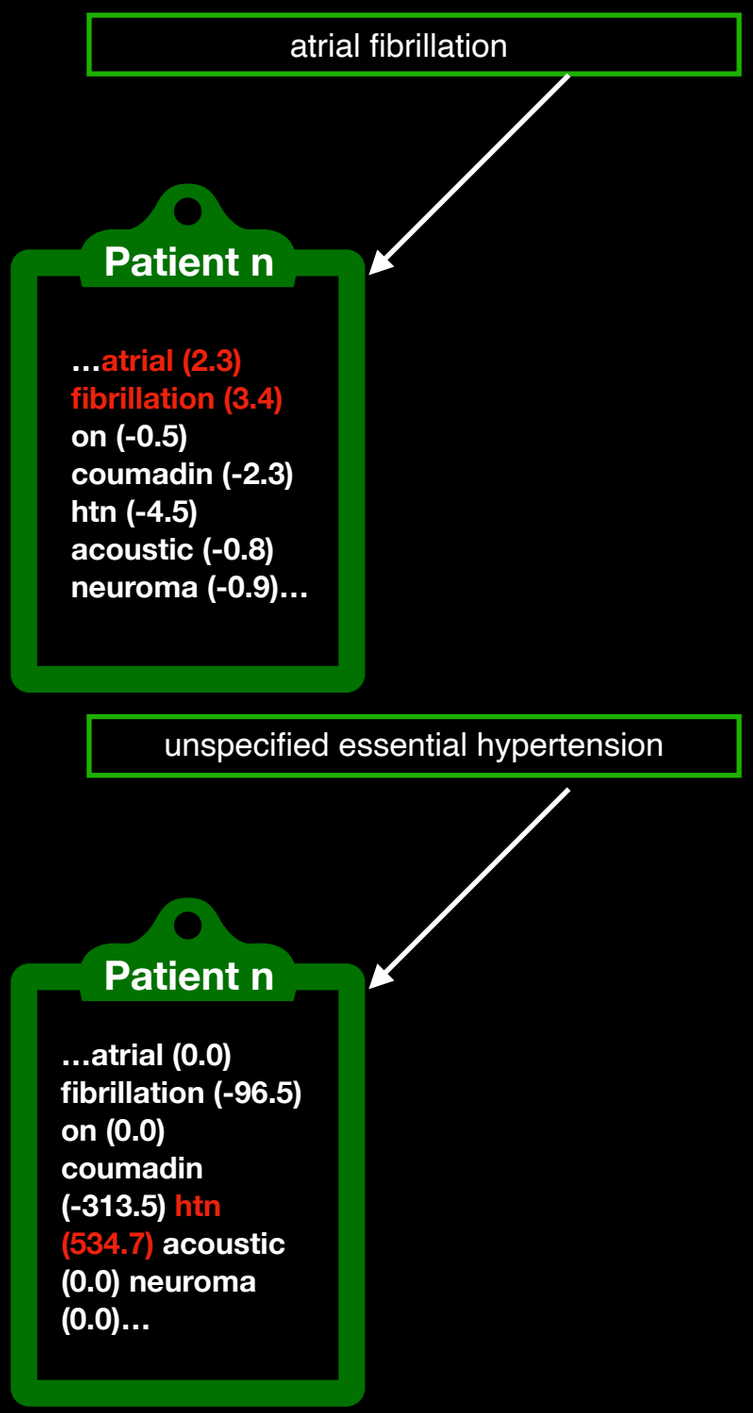


Possible Approach #2: Neural Net with Attention (Cont.)

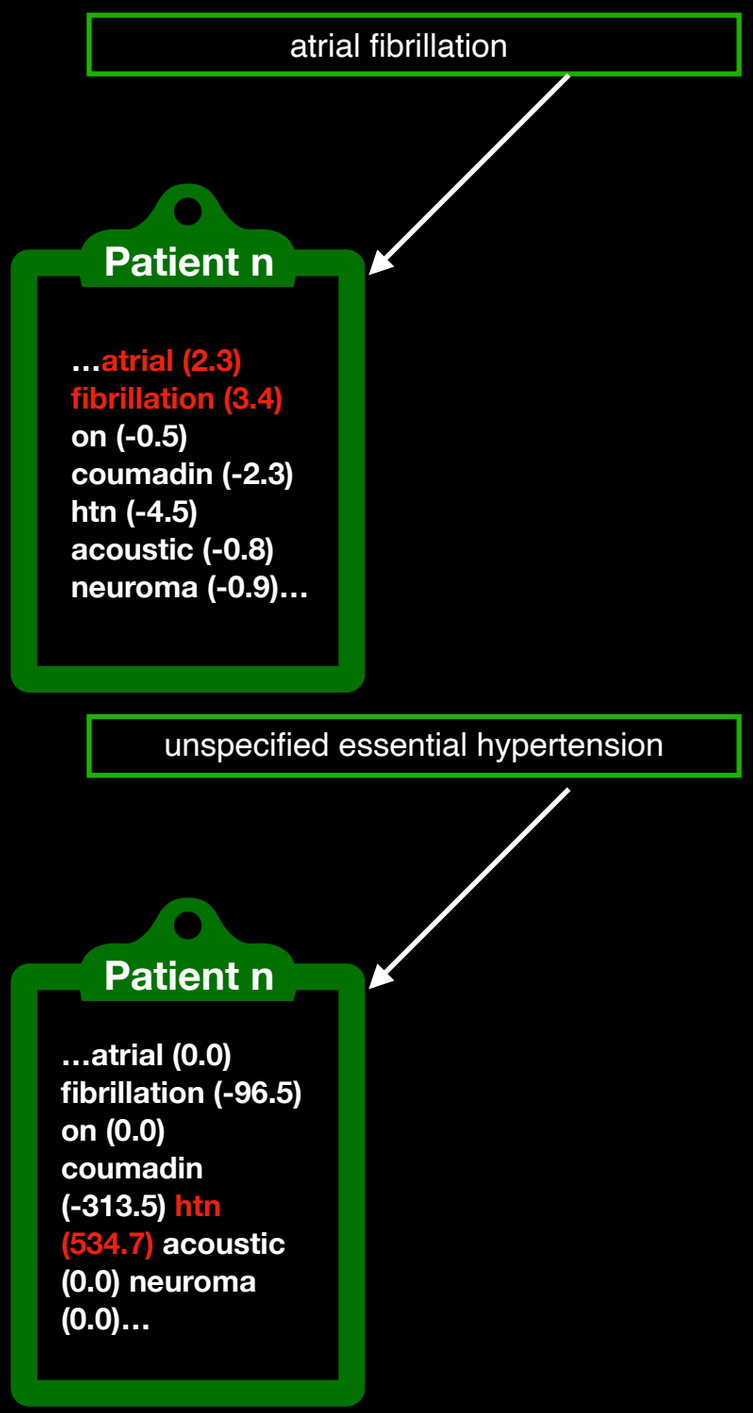
- Additional layer that relates document labels to words in the EHR (distribution/scores over words *for each label for each EHR*)



Possible Approach #2: Neural Attention (Cont.)

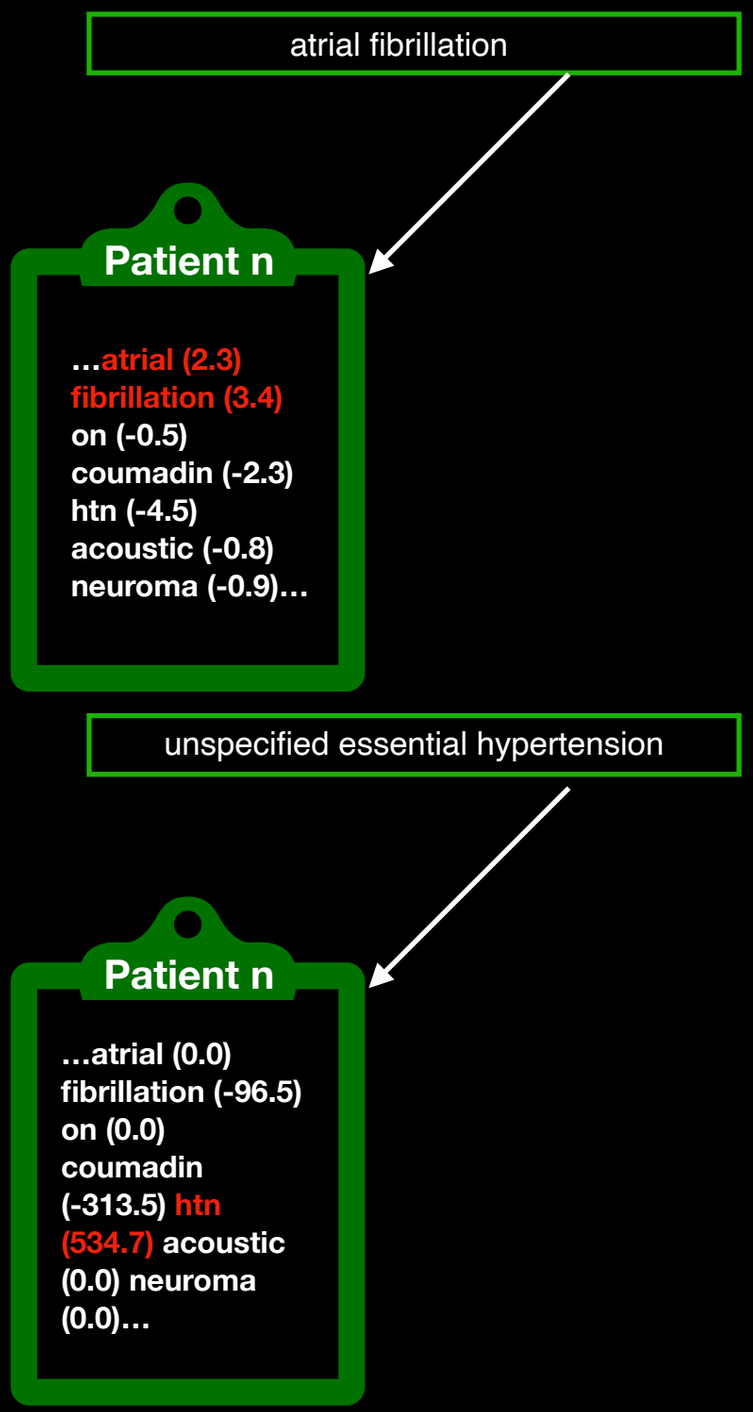


Possible Approach #2: Neural Attention (Cont.)



- **Classification effectiveness**

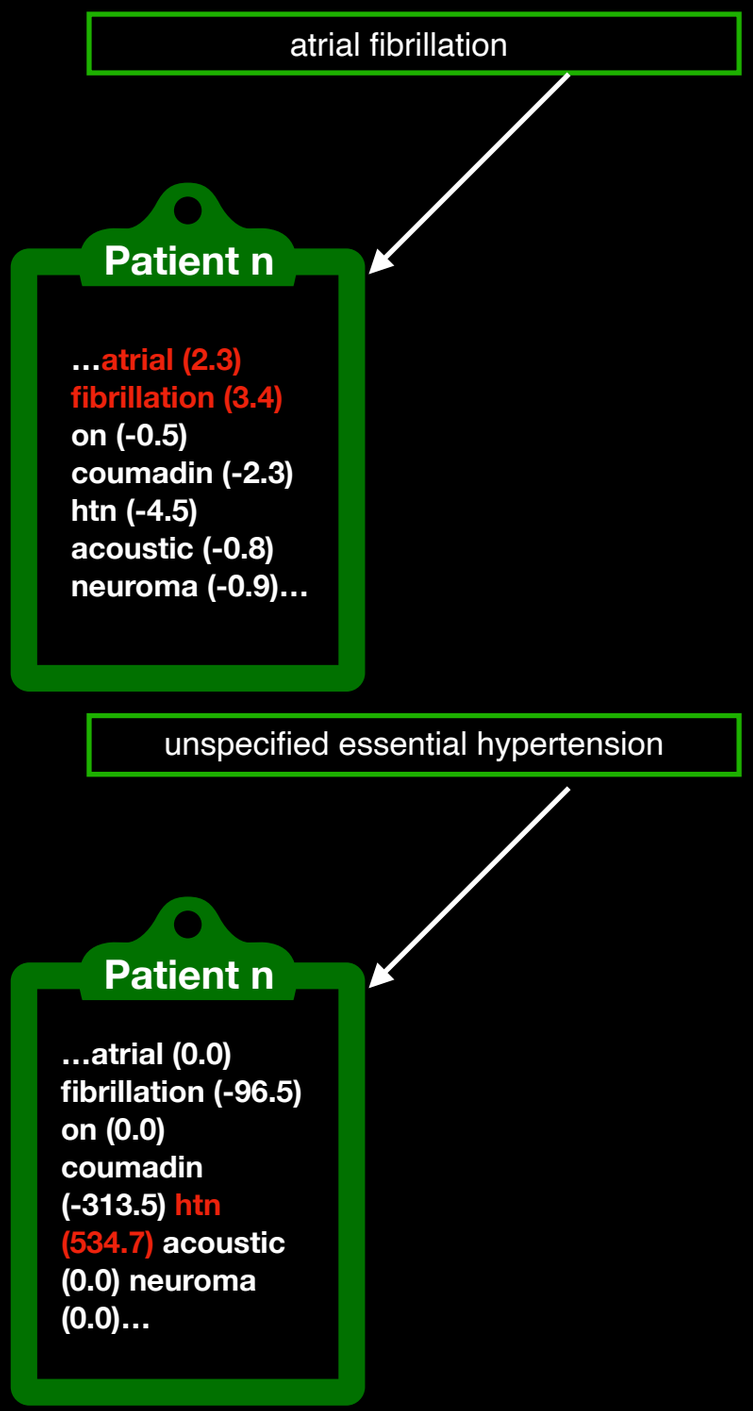
Possible Approach #2: Neural Attention (Cont.)



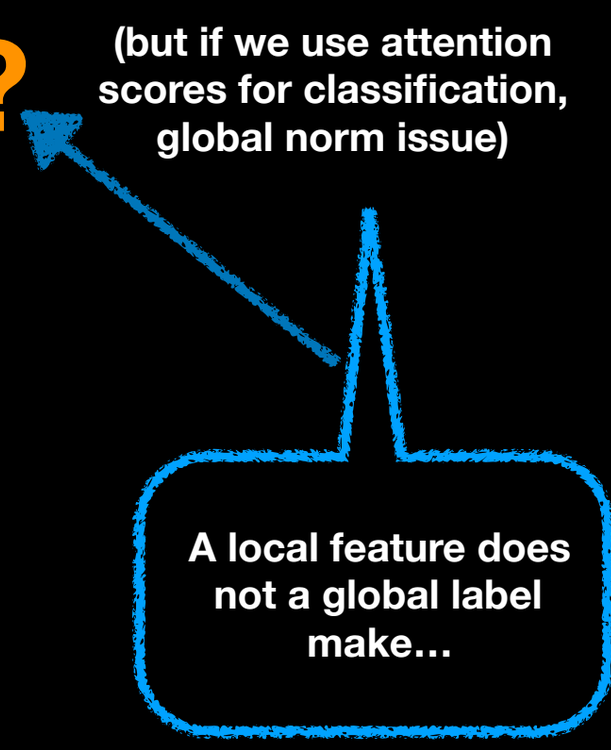
- **Classification effectiveness ✓ / ?**

(but if we use attention scores for classification, global norm issue)

Possible Approach #2: Neural Attention (Cont.)



- **Classification effectiveness ✓ / ?**
- **Interpretability**



Possible Approach #2: Neural Attention (Cont.)

atrial fibrillation

Patient n

…atrial (2.3) fibrillation (3.4) on (-0.5) coumadin (-2.3) htn (-4.5) acoustic (-0.8) neuroma (-0.9)…

unspecified essential hypertension

Patient n

…atrial (0.0) fibrillation (-96.5) on (0.0) coumadin (-313.5) htn (534.7) acoustic (0.0) neuroma (0.0)…

- **Classification effectiveness** ✓ / ?
- **Interpretability** ?

(but if we use attention scores for classification, global norm issue)

A local feature does not a global label make…

Possible Approach #2: Neural Attention (Cont.)

atrial fibrillation

Patient n

…atrial (2.3) fibrillation (3.4) on (-0.5) coumadin (-2.3) htn (-4.5) acoustic (-0.8) neuroma (-0.9)…

unspecified essential hypertension

Patient n

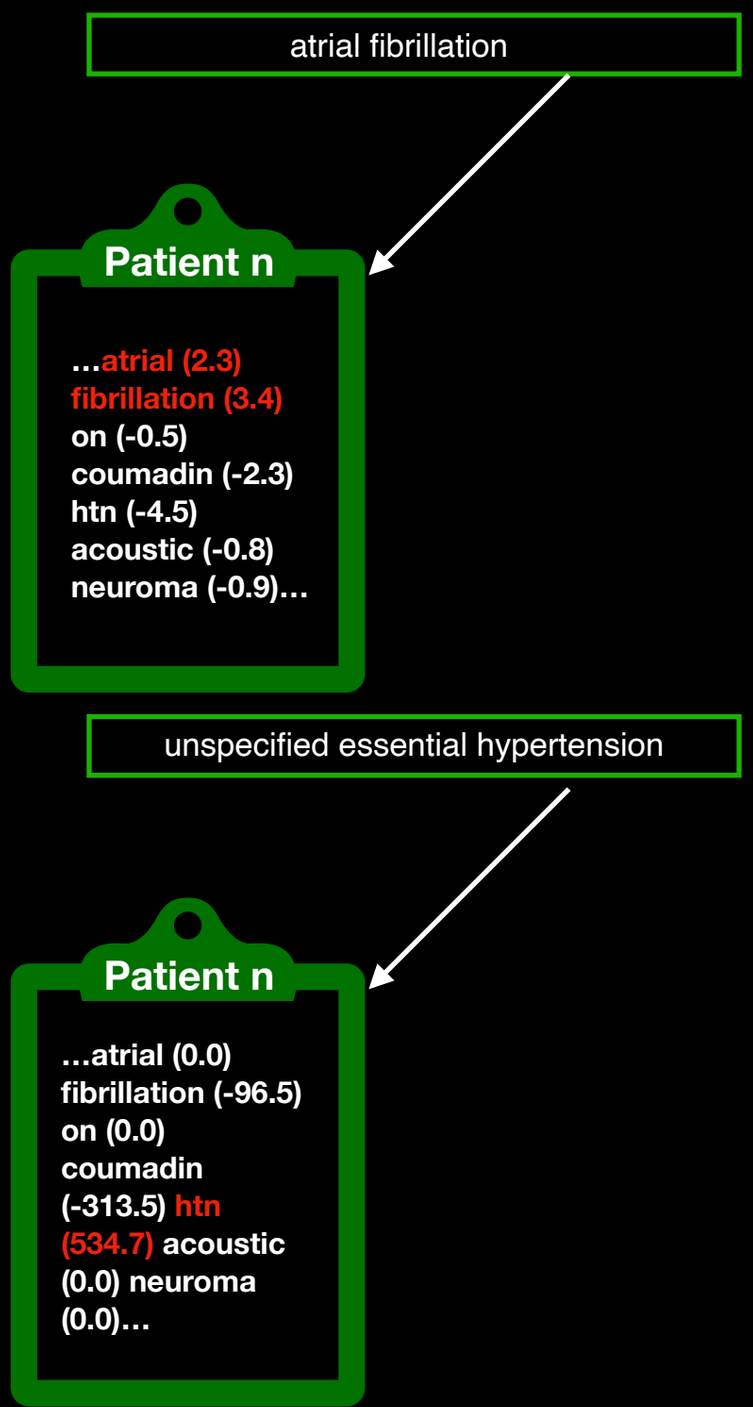
…atrial (0.0) fibrillation (-96.5) on (0.0) coumadin (-313.5) htn (534.7) acoustic (0.0) neuroma (0.0)…

- **Classification effectiveness ✓ / ?**
- **Interpretability ?**
- **Aid for decision making at lower granularities/resolutions**

(but if we use attention scores for classification, global norm issue)

A local feature does not a global label make…

Possible Approach #2: Neural Attention (Cont.)

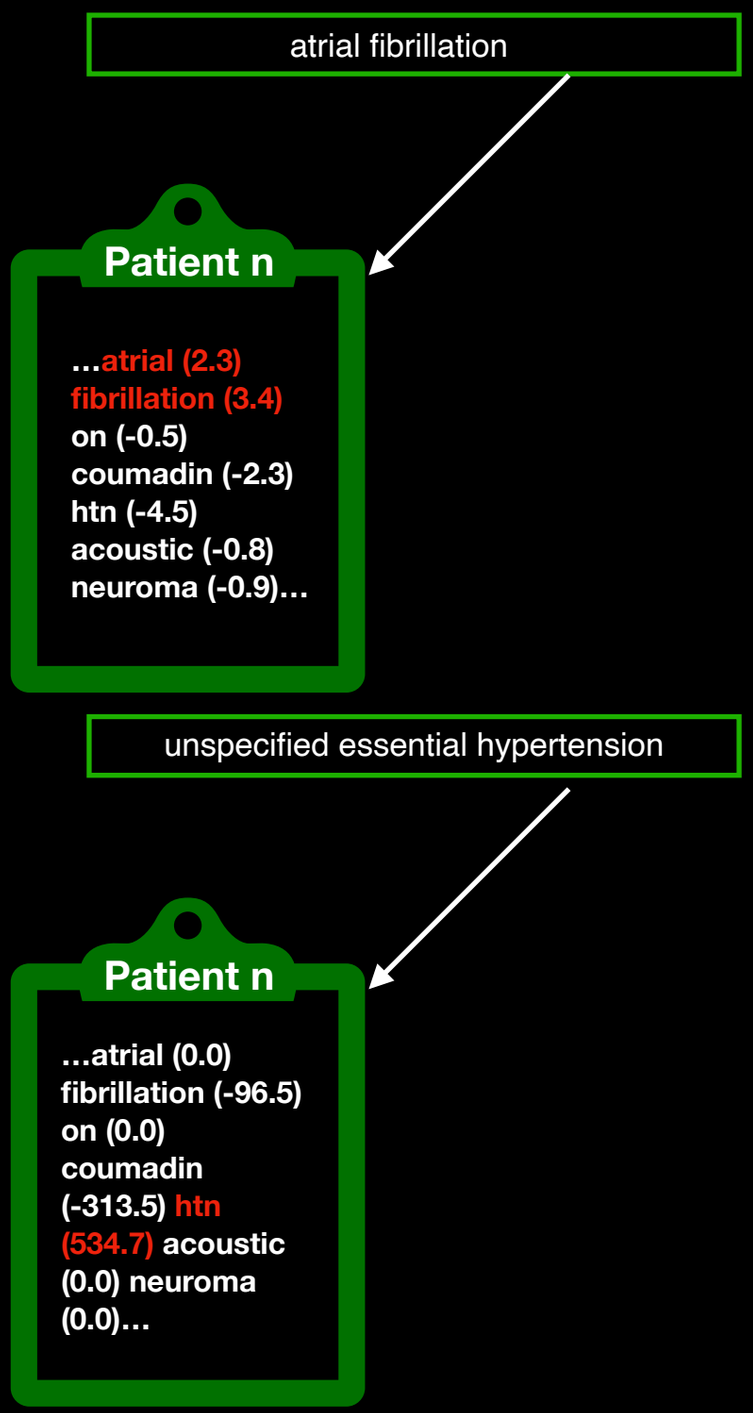


- **Classification effectiveness** ✓ / ?
- **Interpretability** ?
- **Aid for decision making at lower granularities/resolutions** ✓

(but if we use attention scores for classification, global norm issue)

A local feature does not a global label make…

Possible Approach #2: Neural Attention (Cont.)

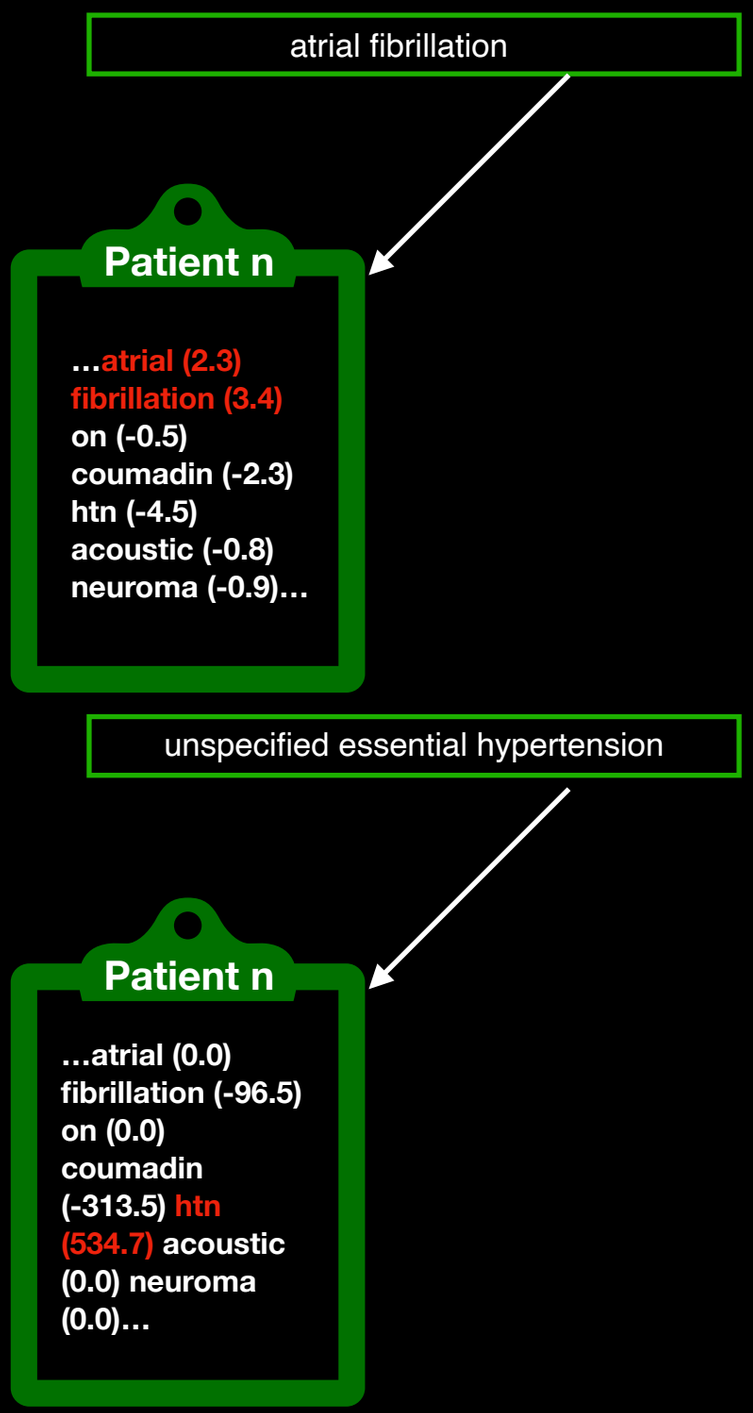


- **Classification effectiveness ✓ / ?**
- **Interpretability ?**
- **Aid for decision making at lower granularities/resolutions ✓**
- **Verify findings with ground-truth training labels**

(but if we use attention scores for classification, global norm issue)

A local feature does not a global label make…

Possible Approach #2: Neural Attention (Cont.)

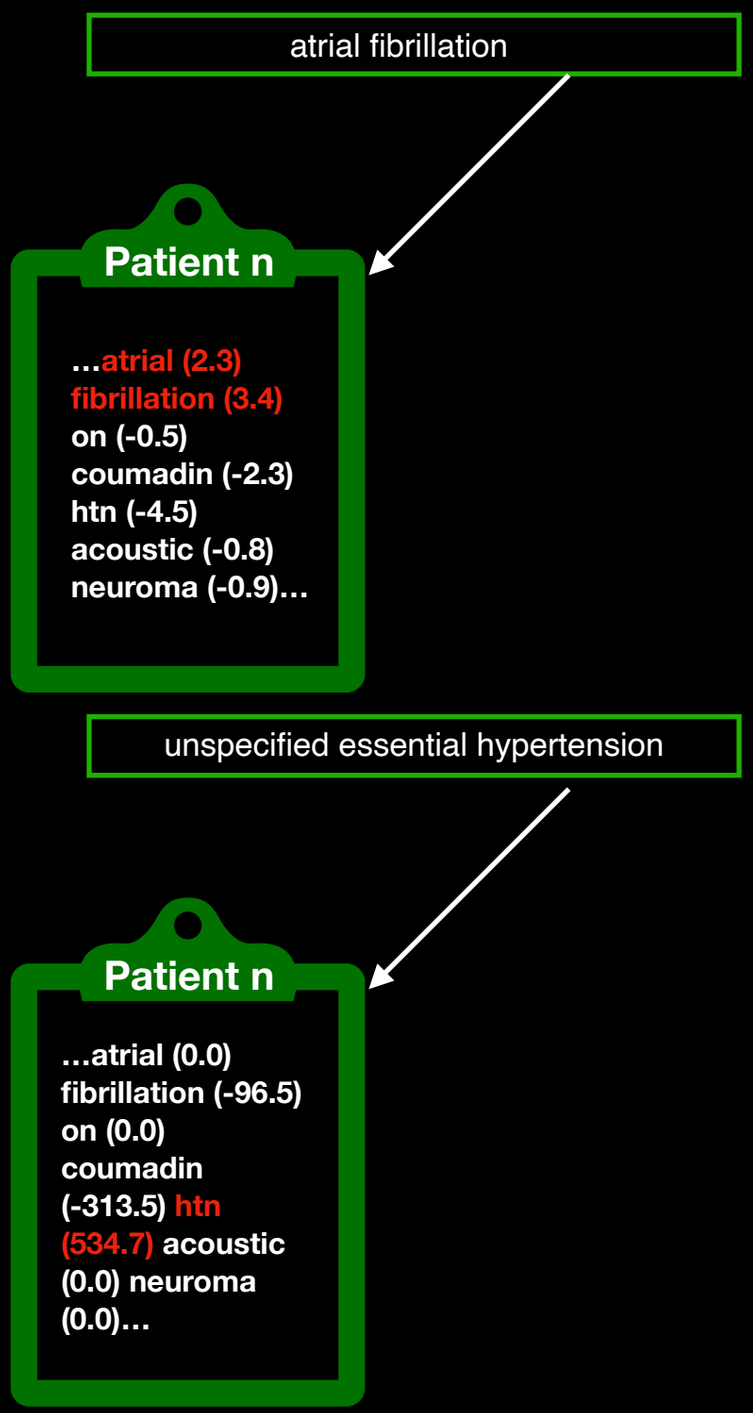


- **Classification effectiveness ✓ / ?**
- **Interpretability ?**
- **Aid for decision making at lower granularities/resolutions ✓**
- **Verify findings with ground-truth training labels ✗**

(but if we use attention scores for classification, global norm issue)

A local feature does not a global label make…

Possible Approach #2: Neural Attention (Cont.)

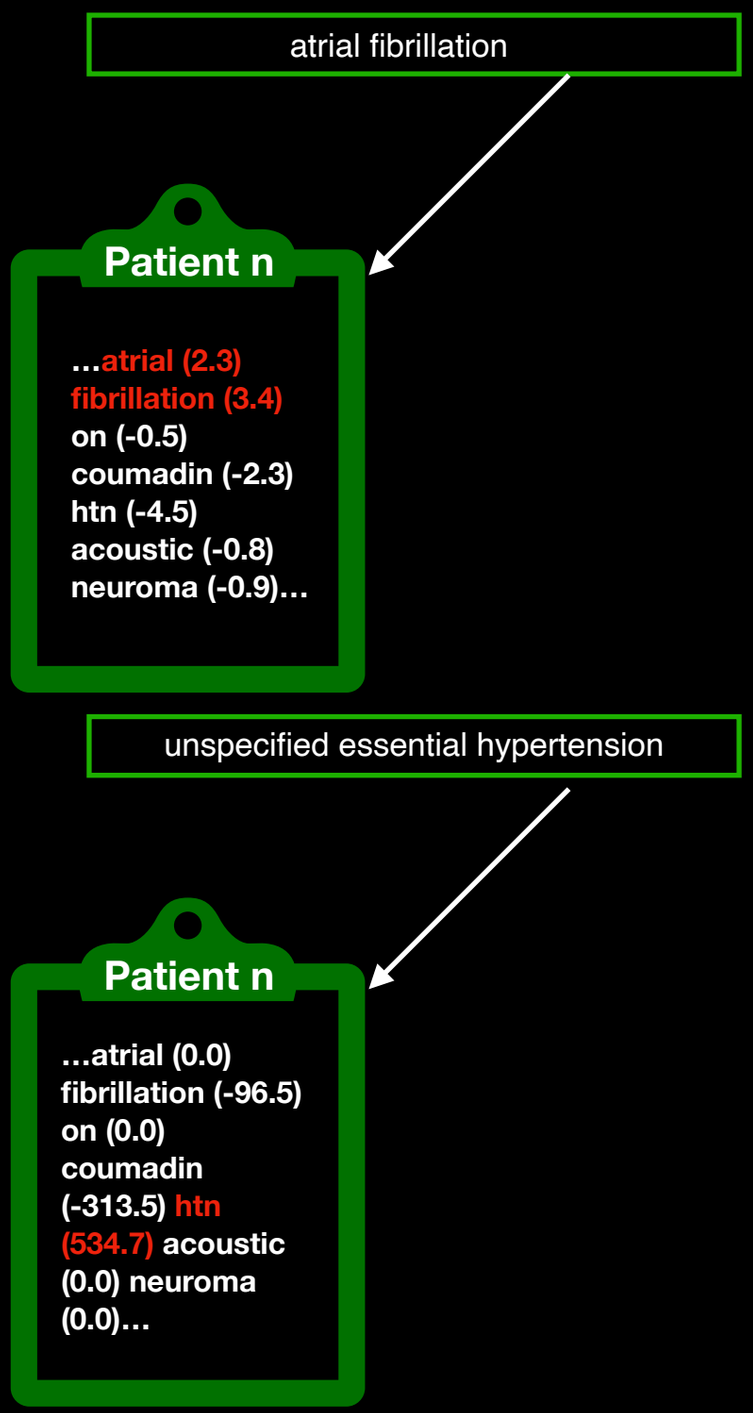


- **Classification effectiveness** ✓ / ?
- **Interpretability** ?
- **Aid for decision making at lower granularities/resolutions** ✓
- **Verify findings with ground-truth training labels** ✗
- **Parameter identification**

(but if we use attention scores for classification, global norm issue)

A local feature does not a global label make…

Possible Approach #2: Neural Attention (Cont.)

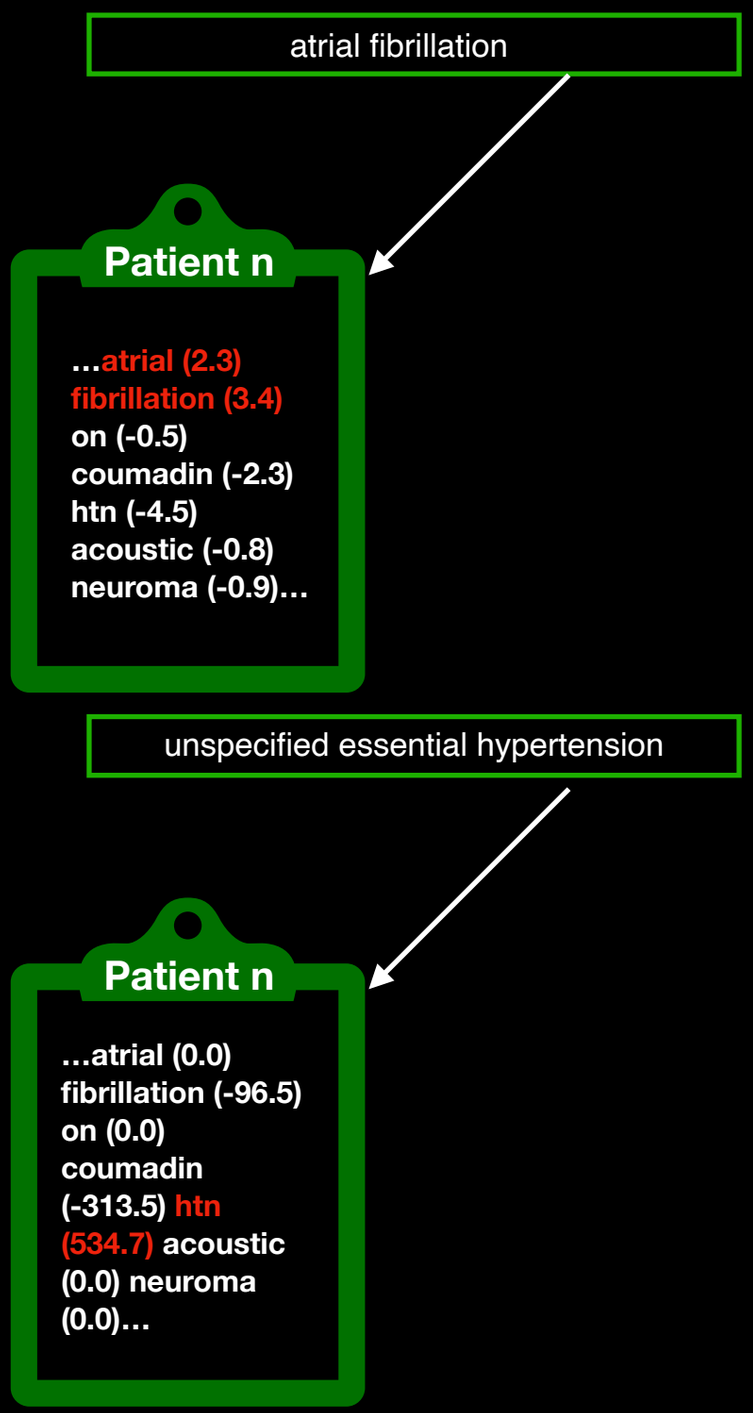


- **Classification effectiveness** ✓ / ?
- **Interpretability** ?
- **Aid for decision making at lower granularities/resolutions** ✓
- **Verify findings with ground-truth training labels** ✗
- **Parameter identification** ✗

(but if we use attention scores for classification, global norm issue)

A local feature does not a global label make…

Possible Approach #2: Neural Attention (Cont.)

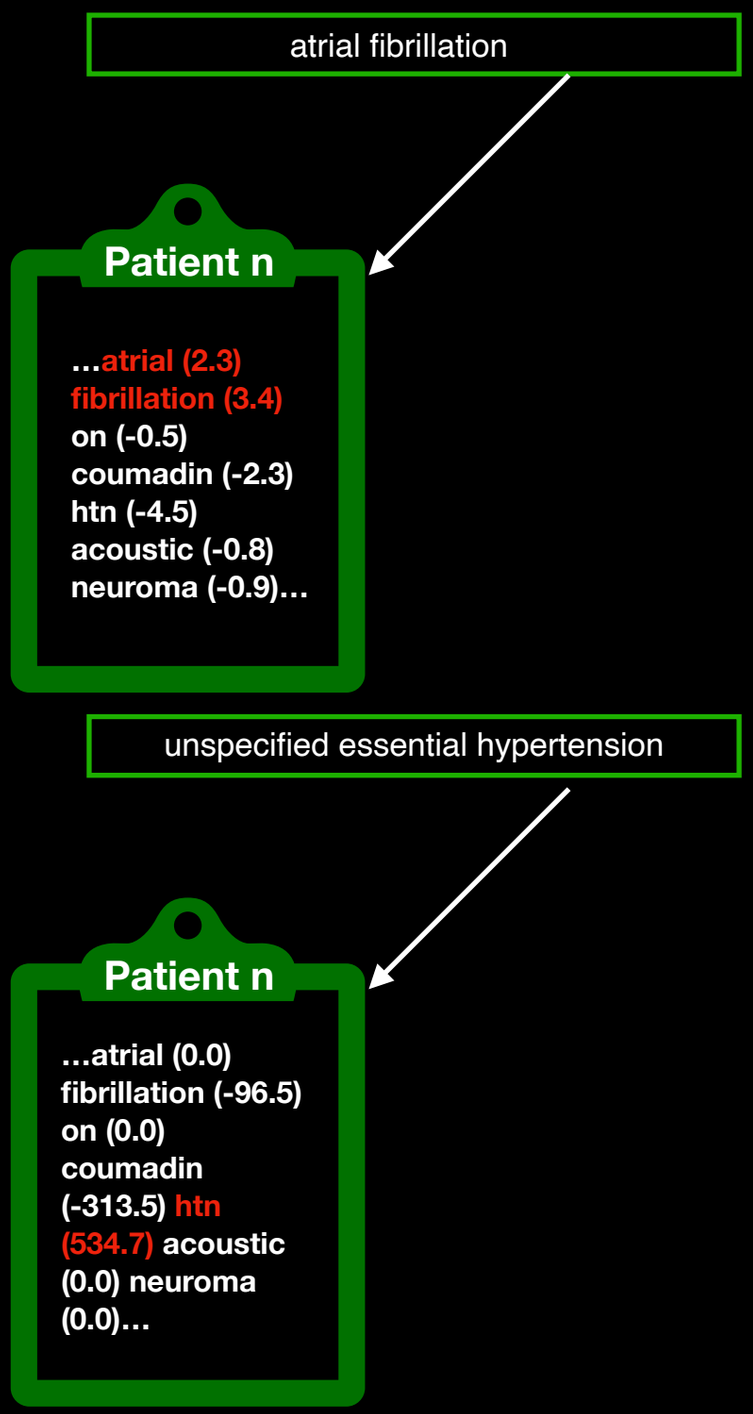


- **Classification effectiveness** ✓ / ?
- **Interpretability** ?
- **Aid for decision making at lower granularities/resolutions** ✓
- **Verify findings with ground-truth training labels** ✗
- **Parameter identification** ✗
- **Robustness to adversarial attacks**

(but if we use attention scores for classification, global norm issue)

A local feature does not a global label make…

Possible Approach #2: Neural Attention (Cont.)

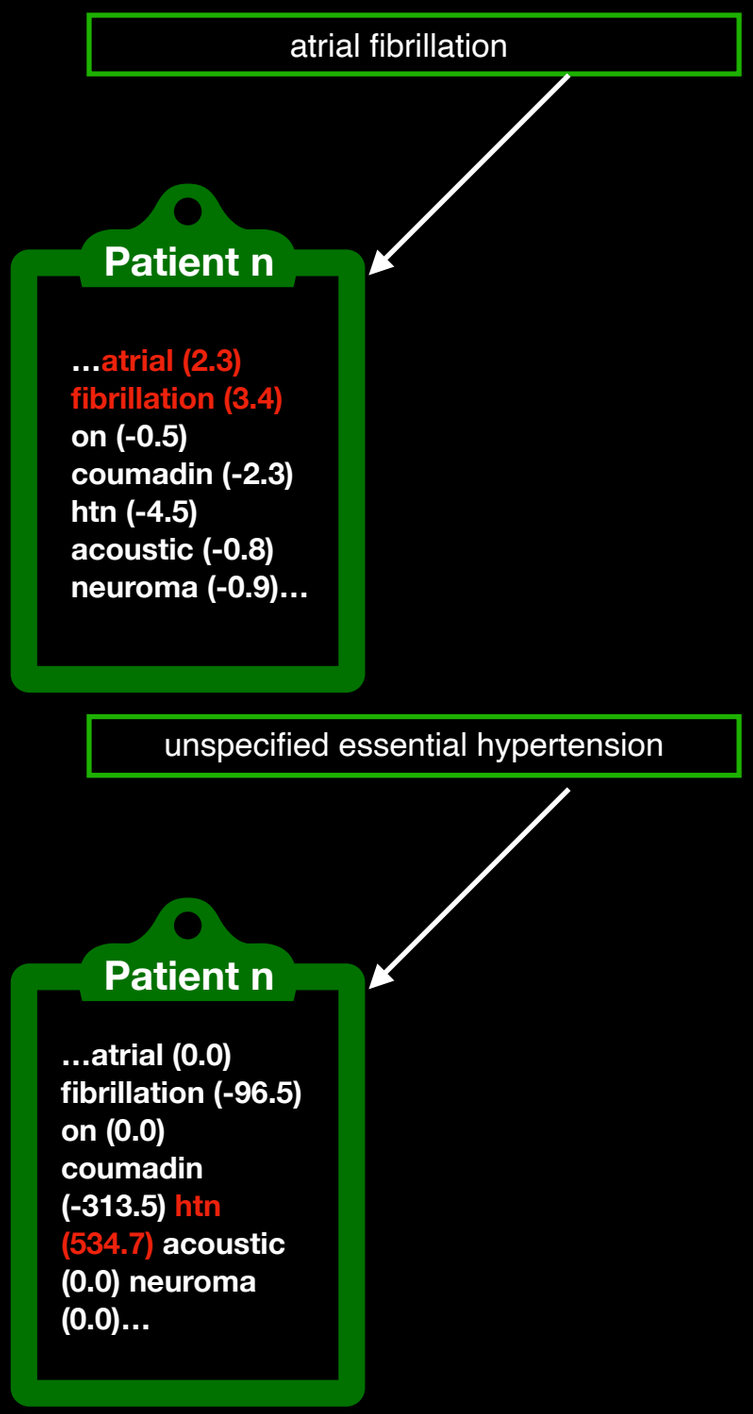


- **Classification effectiveness** ✓ / ?
- **Interpretability** ?
- **Aid for decision making at lower granularities/resolutions** ✓
- **Verify findings with ground-truth training labels** ✗
- **Parameter identification** ✗
- **Robustness to adversarial attacks** ✗

(but if we use attention scores for classification, global norm issue)

A local feature does not a global label make…

Possible Approach #2: Neural Attention (Cont.)

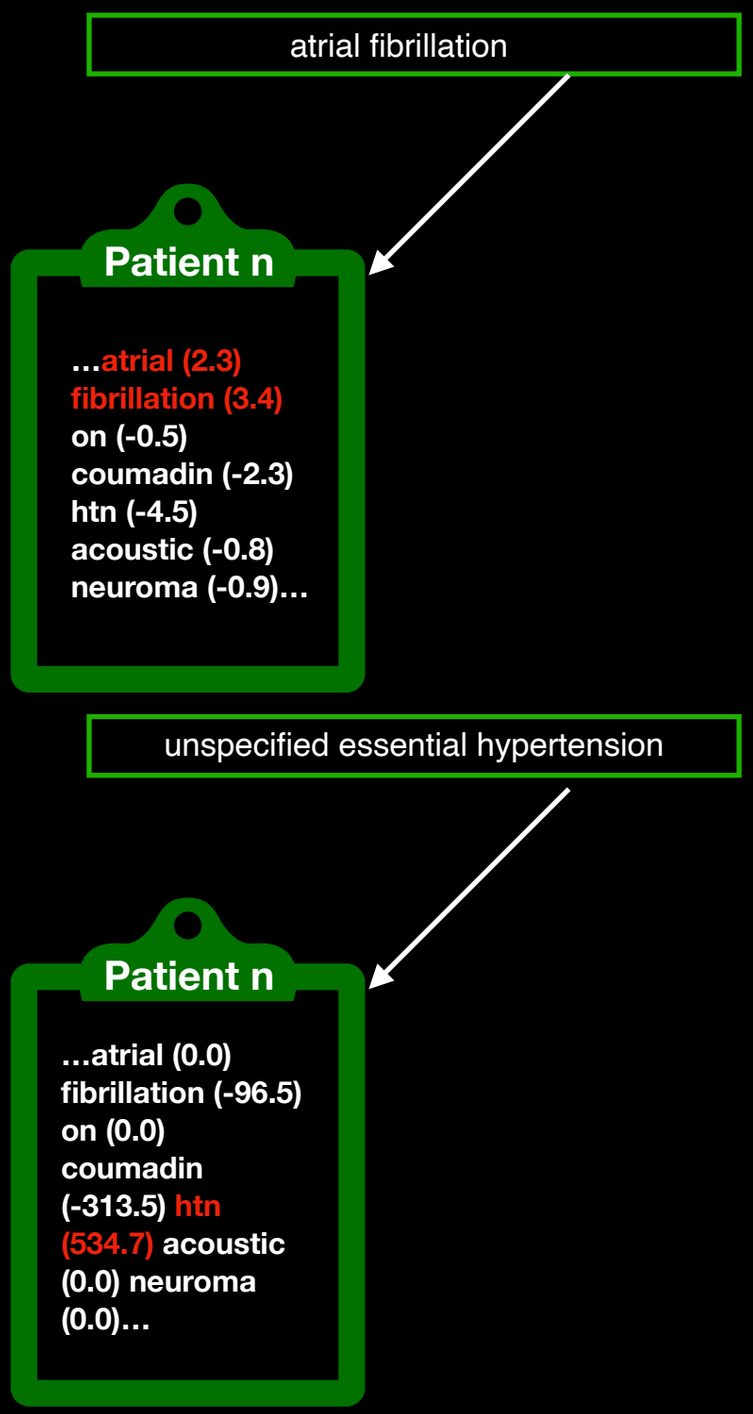


- **Classification effectiveness** ✓ / ?
- **Interpretability** ?
- **Aid for decision making at lower granularities/resolutions** ✓
- **Verify findings with ground-truth training labels** ✗
- **Parameter identification** ✗
- **Robustness to adversarial attacks** ✗
- **Efficiency (space/time)**

(but if we use attention scores for classification, global norm issue)

A local feature does not a global label make…

Possible Approach #2: Neural Attention (Cont.)

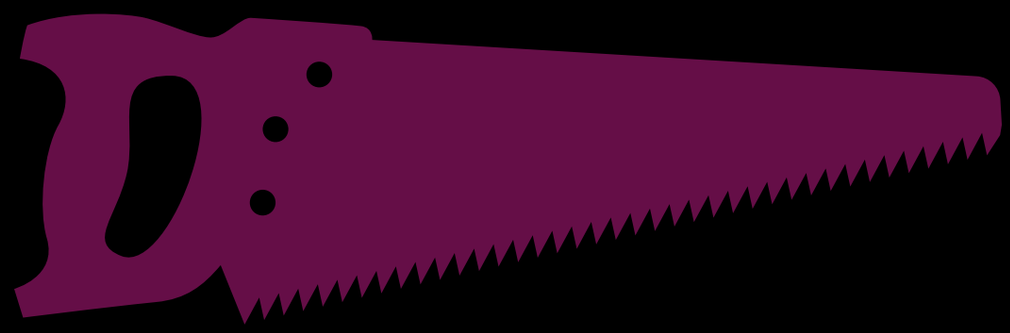


- **Classification effectiveness** ✓ / ?
- **Interpretability** ?
- **Aid for decision making at lower granularities/resolutions** ✓
- **Verify findings with ground-truth training labels** ✗
- **Parameter identification** ✗
- **Robustness to adversarial attacks** ✗
- **Efficiency (space/time)** ?

(but if we use attention scores for classification, global norm issue)

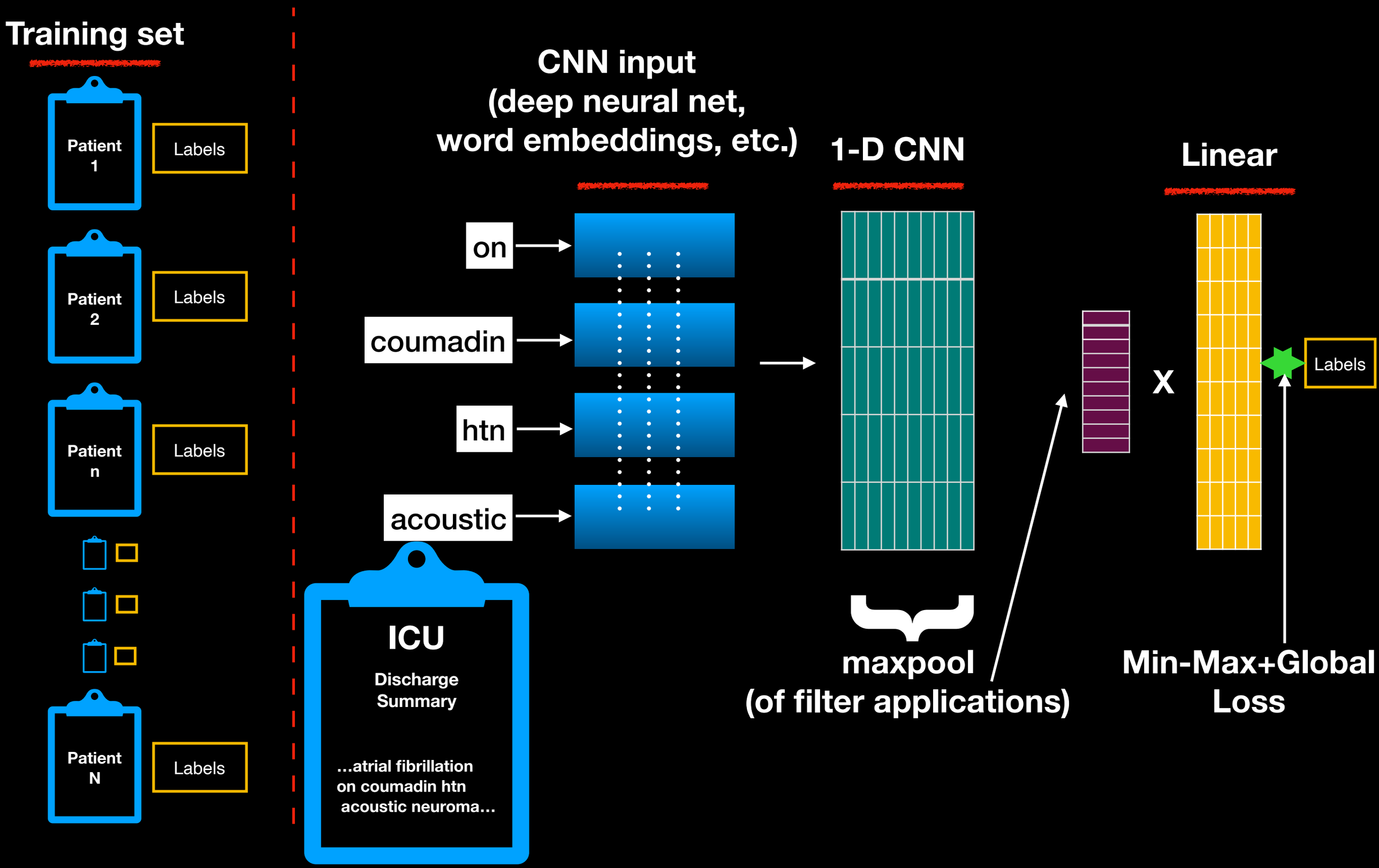
A local feature does not a global label make…

Our Solution



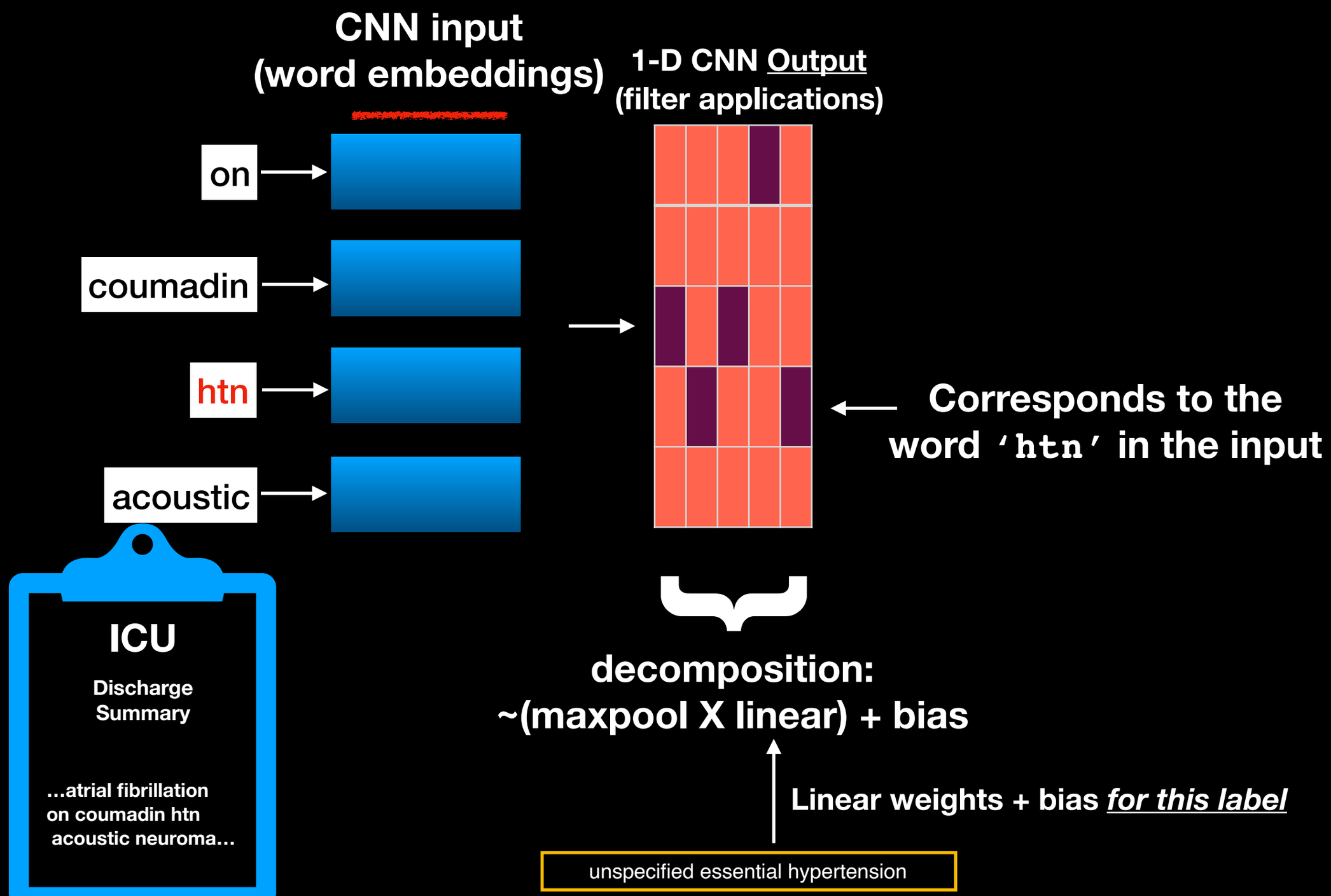
- ~“Cut the neural network into (relevant) pieces and then match the pieces from test with those from training (for which we have gold labels)”
  - CNN Decomposition (our “attention” mechanism)
    - BLADE: “lancet”
  - Exemplar auditing
    - The mapping approach: “suture to bind train & test features”

Our Solution (high-level intuition)



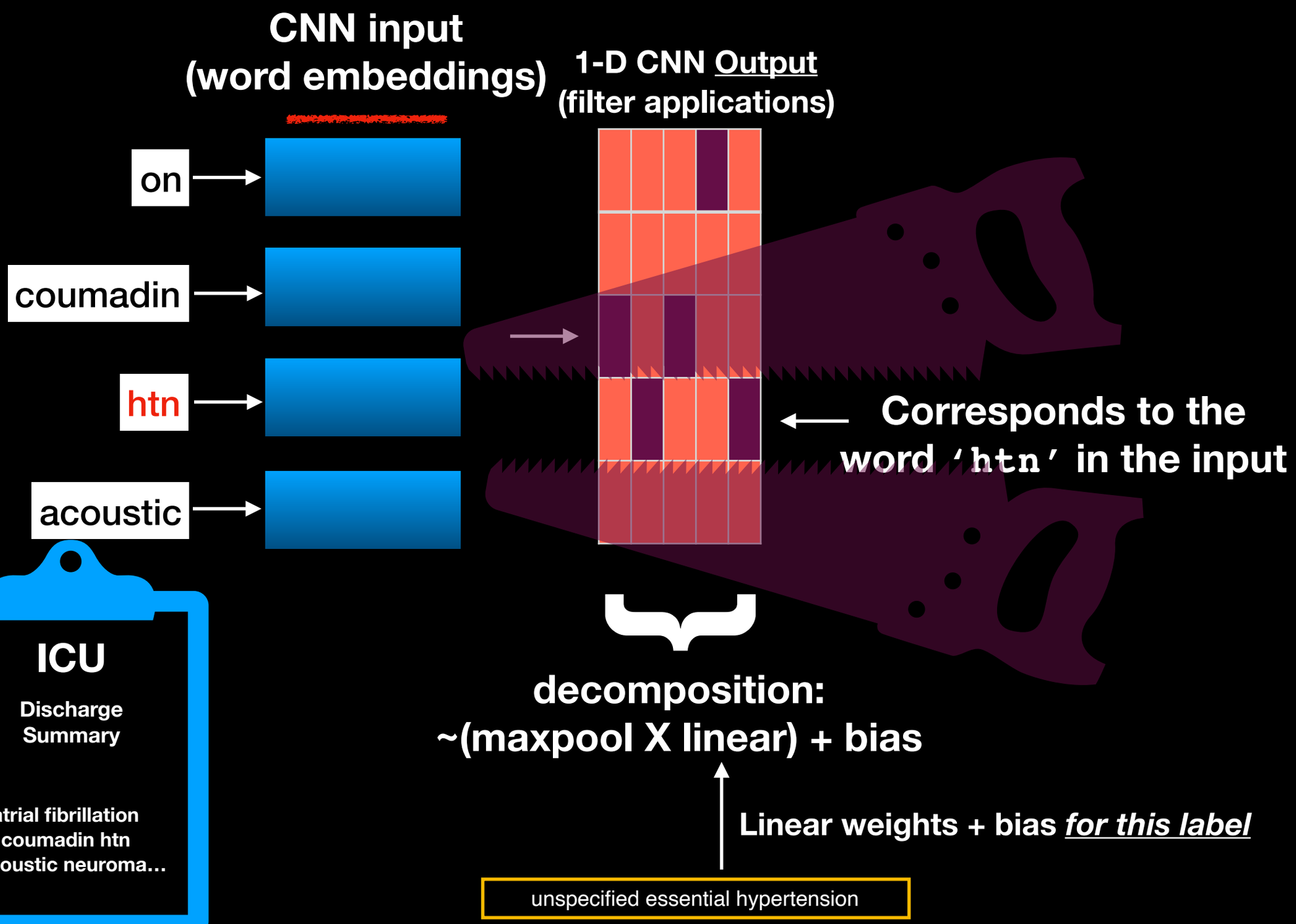
Our Solution (high-level intuition) Cont.

Step 1 (of 3): Use the CNN Decomposition to identify relevant features (*for each label*)



Our Solution (high-level intuition) Cont.

Step 2 (of 3): Match relevant features (*for each label*) from test and training

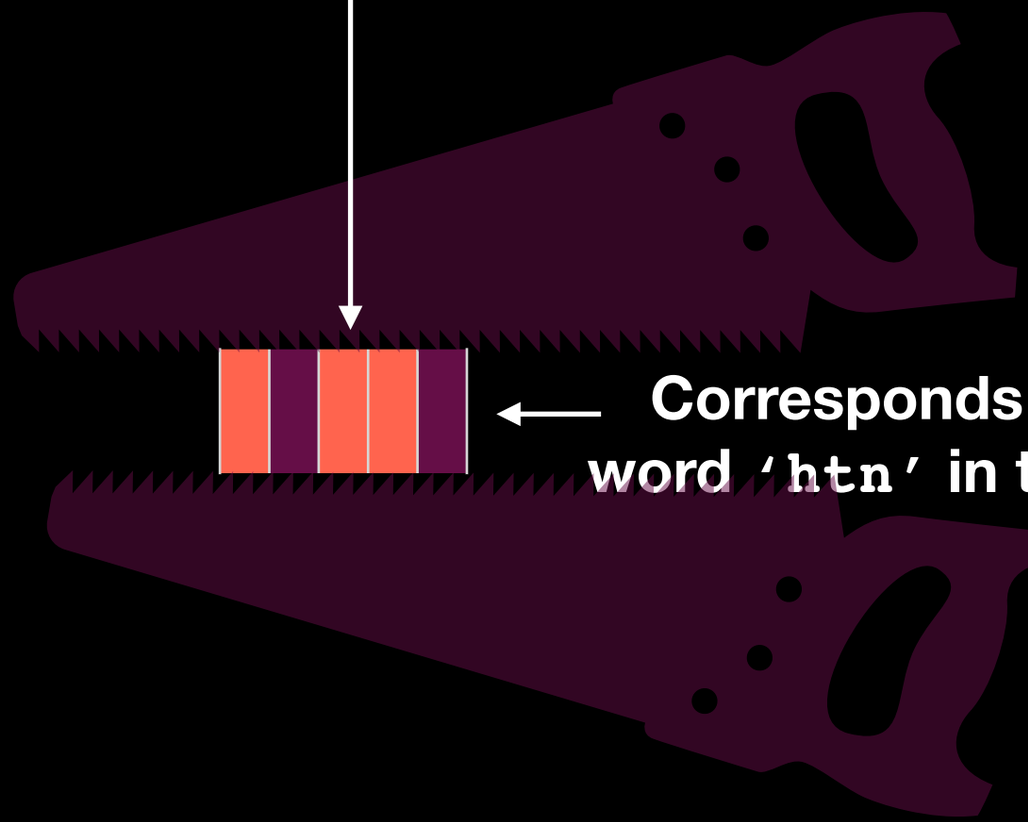


Our Solution (high-level intuition) Cont.

Step 2 (of 3): Match relevant features (*for each label*) from test and training

“Exemplar vector”

(summary of the network for this feature *for this label*)



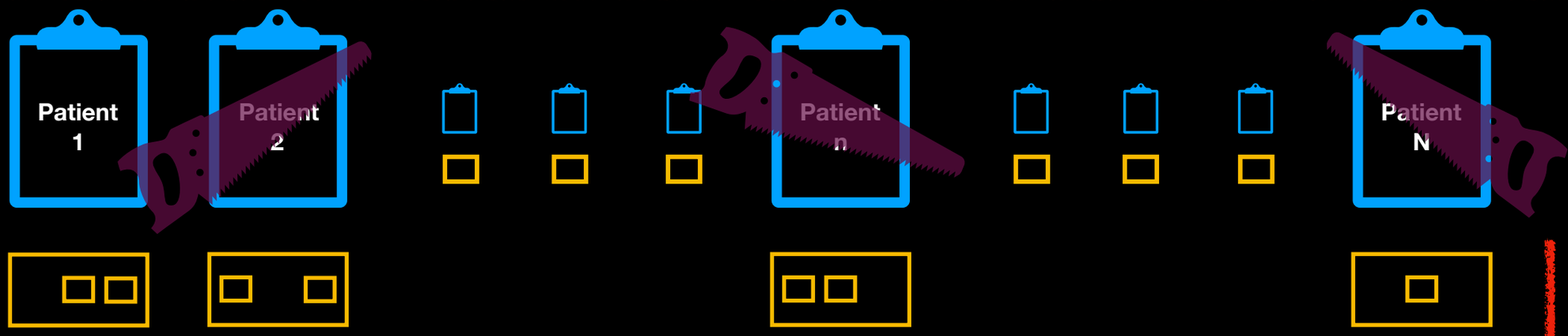
Corresponds to the word ‘htn’ in the input

unspecified essential hypertension

Our Solution (high-level intuition) Cont.

Step 2 (of 3) Cont.: Match relevant features (*for each label*) from test and training

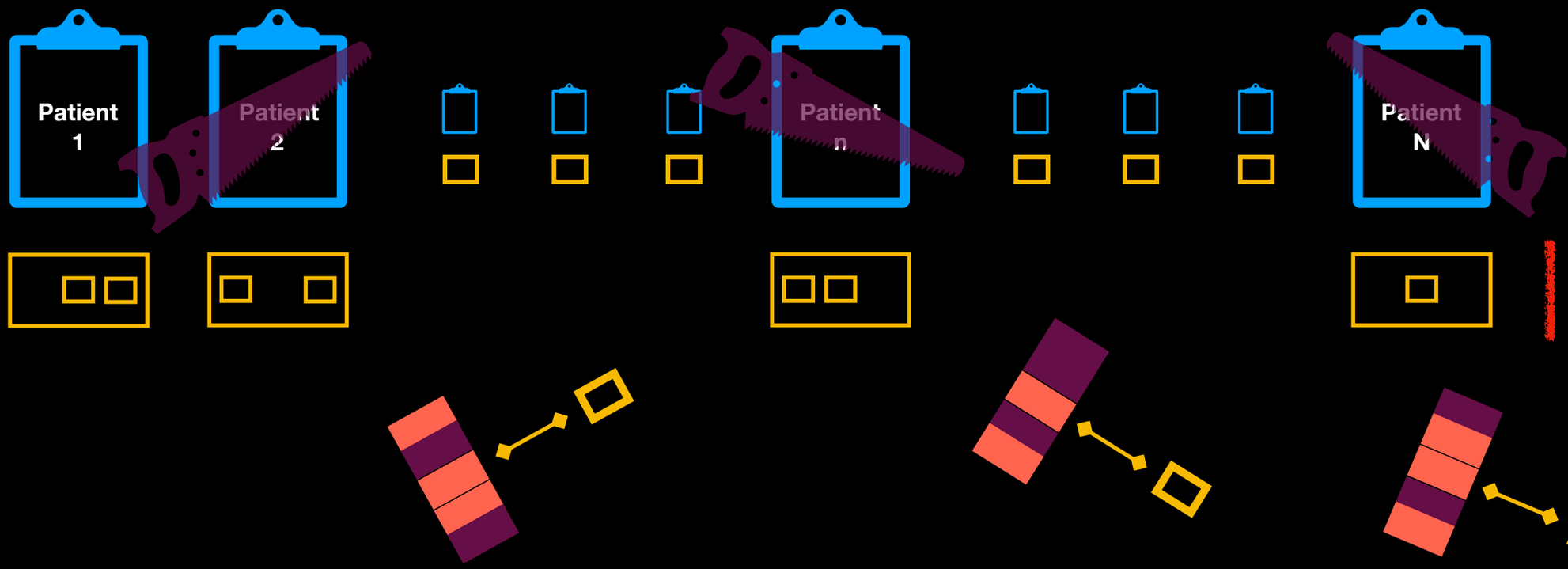
In same manner, cut exemplars from training (but also *save associated ground-truth labels*)



Our Solution (high-level intuition) Cont.

Step 2 (of 3) Cont.: Match relevant features (*for each label*) from test and training

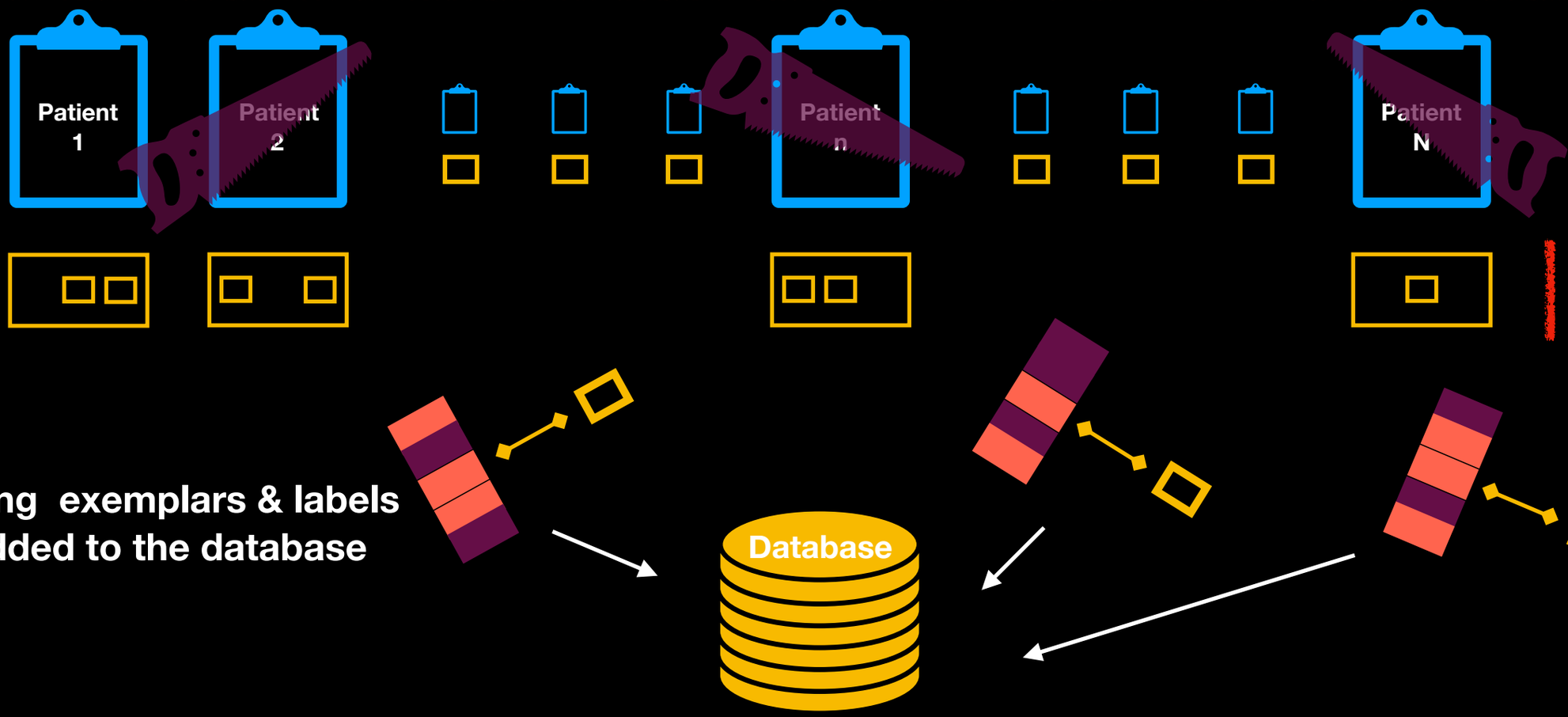
In same manner, cut exemplars from training (but also *save associated ground-truth labels*)



Our Solution (high-level intuition) Cont.

Step 2 (of 3) Cont.: Match relevant features (*for each label*) from test and training

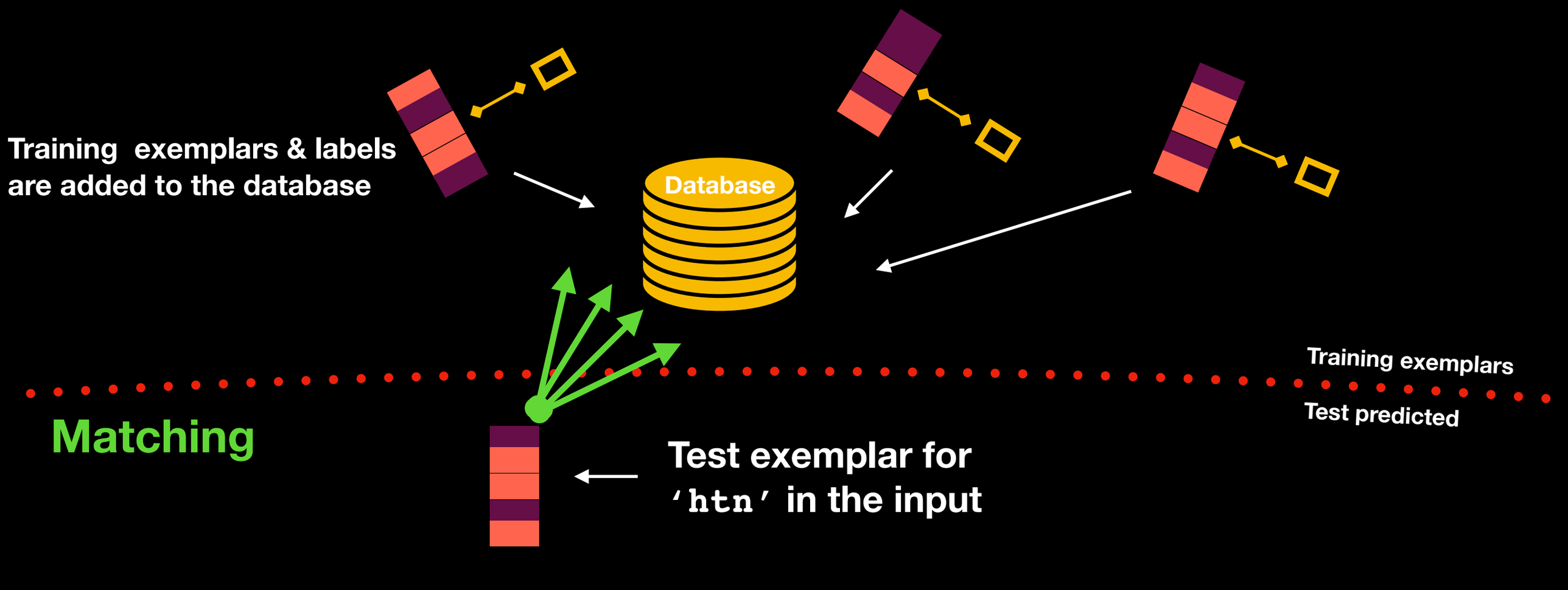
In same manner, cut exemplars from training (but also *save associated ground-truth labels*)



Our Solution (high-level intuition) Cont.

Step 2 (of 3) Cont.: Match relevant features (*for each label*) from test and training

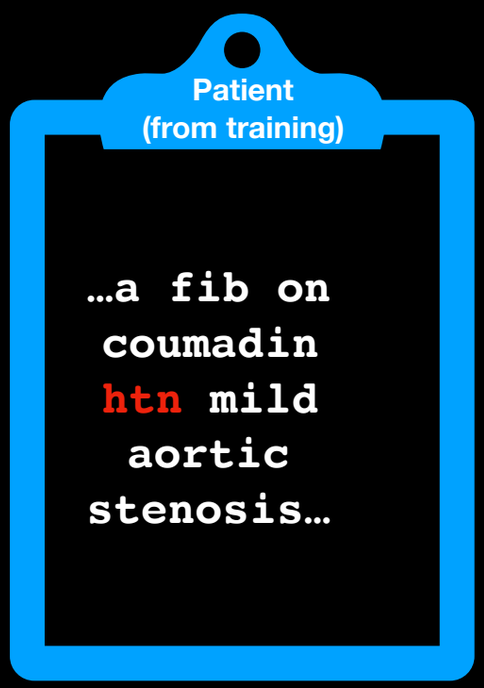
In same manner, cut exemplars from training (but also *save associated ground-truth labels*)



Our Solution (high-level intuition) Cont.

Step 2 (of 3) Cont.: Match relevant features (*for each label*) from test and training

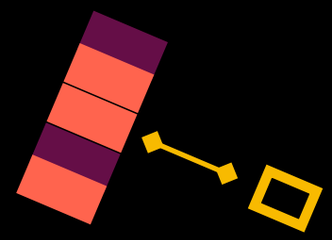
Match test exemplar vector with the training exemplar that minimizes the Euclidean distance



Ground-truth label associated with the training exemplar (and patient)

unspecified essential hypertension

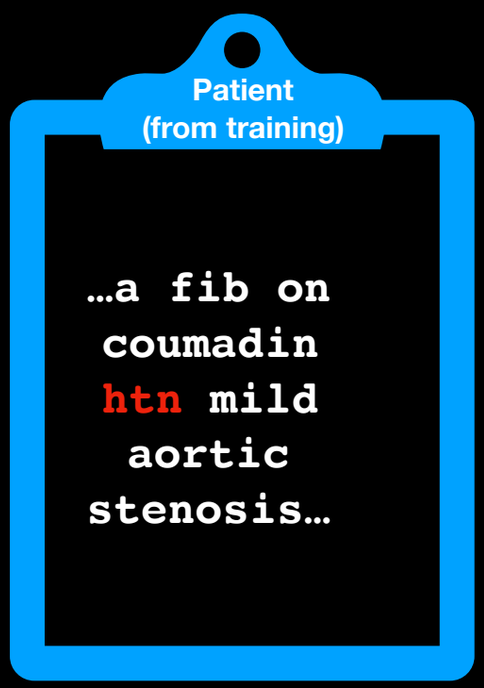
Training exemplar (here, also for `'htn'` in the training input)



Our Solution (high-level intuition) Cont.

Step 2 (of 3) Cont.: Match relevant features (*for each label*) from test and training

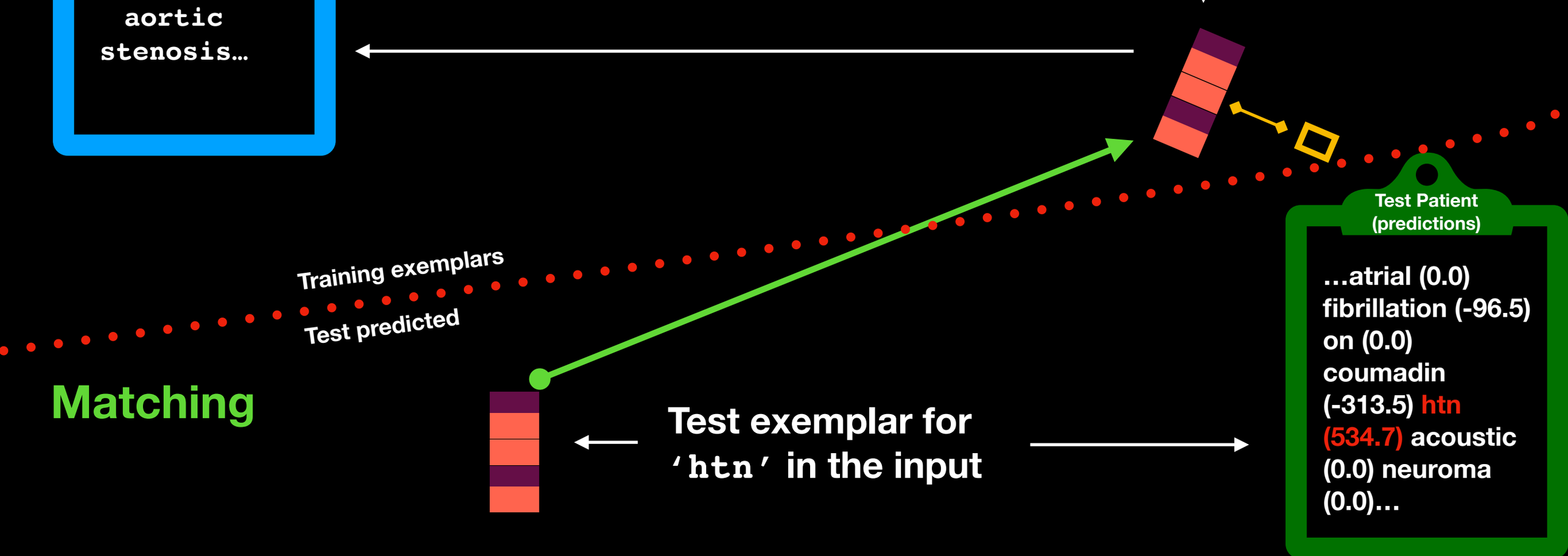
Match test exemplar vector with the training exemplar that minimizes the Euclidean distance



Ground-truth label associated with the training exemplar (and patient)

unspecified essential hypertension

Training exemplar (here, also for `'htn'` in the training input)



Our Solution (high-level intuition) Cont.

Step 3 (of 3) Cont.: Examine distances to nearest True Positives (TP), False Negatives (FN), False Positives (FP), and True Negatives (TN) from training

Match test exemplar vector with the training exemplar that minimizes the Euclidean distance

Euclidean distance between two vectors ($\mathbf{a}$, $\mathbf{b}$):

$$\mathbf{a} \in \mathbb{R}^N = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} \qquad \mathbf{b} \in \mathbb{R}^N = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix}$$

$$distance(\mathbf{a}, \mathbf{b}) = \|\mathbf{a} - \mathbf{b}\|_2 = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \cdots + (a_N - b_N)^2}$$

Our Solution (high-level intuition) Cont.

Step 3 (of 3) Cont.: Examine distances to nearest True Positives (TP), False Negatives (FN), False Positives (FP), and True Negatives (TN) from training

Match test exemplar vector with the training exemplar that minimizes the Euclidean distance

Find:

$$\underset{\mathbf{v}^{train}}{argmin} \|\mathbf{v}^{test} - \mathbf{v}^{train}\|_2$$

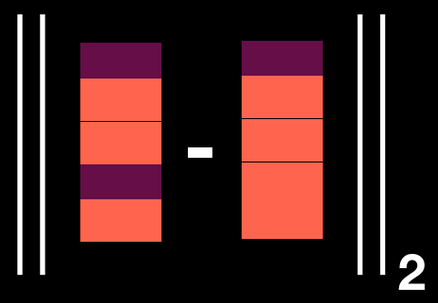
Our Solution (high-level intuition) Cont.

Step 3 (of 3) Cont.: Examine distances to nearest True Positives (TP), False Negatives (FN), False Positives (FP), and True Negatives (TN) from training

Match test exemplar vector with the training exemplar that minimizes the Euclidean distance

Find:

$$\underset{\mathbf{v}^{train}}{argmin} \|\mathbf{v}^{test} - \mathbf{v}^{train}\|_2$$



Repeat for each relevant TP, FN, FP, & TN from training

Our Solution (high-level intuition) Cont.

Step 3 (of 3) Cont.: Examine distances to nearest True Positives (TP), False Negatives (FN), False Positives (FP), and True Negatives (TN) from training

Normalize *negative* Euclidean distances with a softmax

Softmax function applied to a vector (c):

$$\mathbf{c} \in \mathbb{R}^N = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix}$$

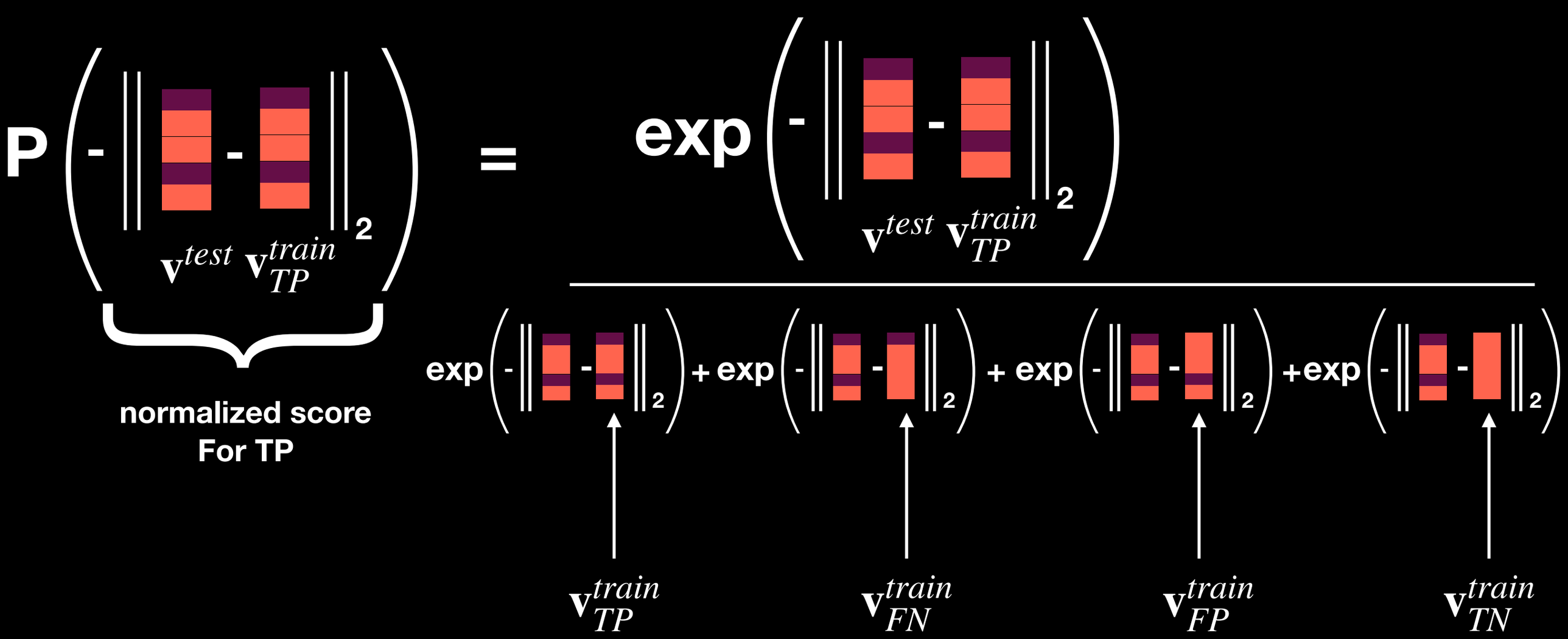
$$softmax(c_i) = \frac{e^{c_i}}{\sum_{n=1}^{N} e^{c_n}}$$

Here, our vector contains 4 values: the distances to each TP, FN, FP, & TN from training

Our Solution (high-level intuition) Cont.

Step 3 (of 3) Cont.: Examine distances to nearest True Positives (TP), False Negatives (FN), False Positives (FP), and True Negatives (TN) from training

Normalize *negative* Euclidean distances with a softmax



Our Solution (high-level intuition) Cont.

Step 3 (of 3) Cont.: Examine distances to nearest True Positives (TP), False Negatives (FN), False Positives (FP), and True Negatives (TN) from training

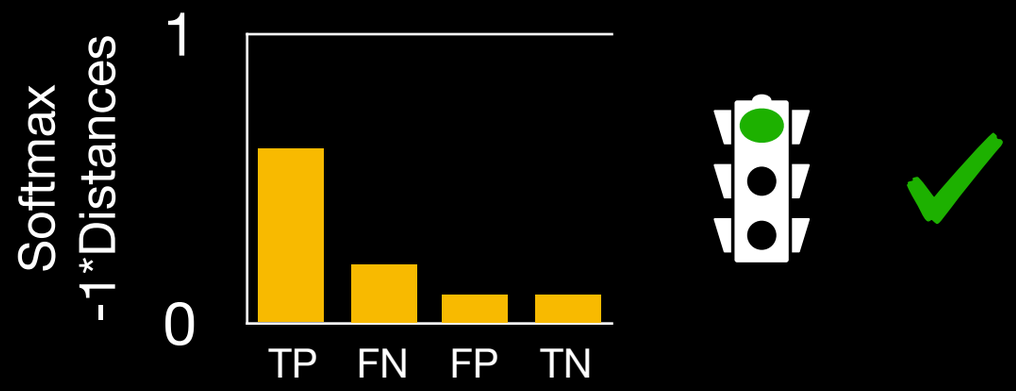
Exemplar Auditing: Illustrative Cases

Our Solution (high-level intuition) Cont.

Step 3 (of 3) Cont.: Examine distances to nearest True Positives (TP), False Negatives (FN), False Positives (FP), and True Negatives (TN) from training

Exemplar Auditing: Illustrative Cases

Closest exemplar is a true positive

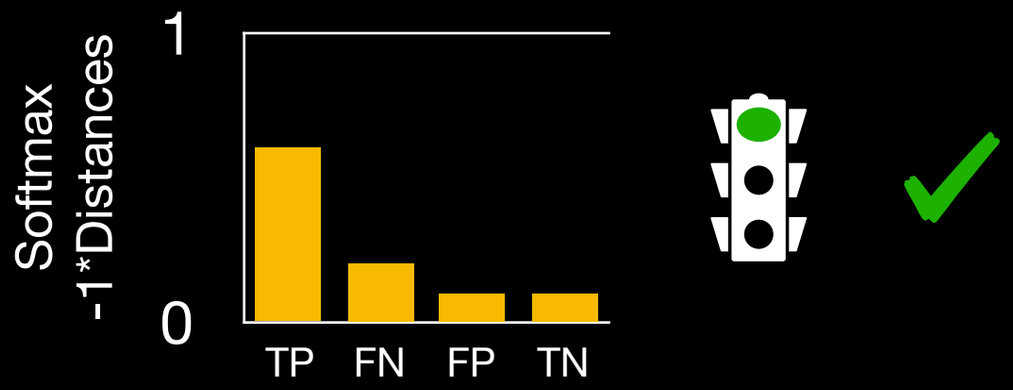


Our Solution (high-level intuition) Cont.

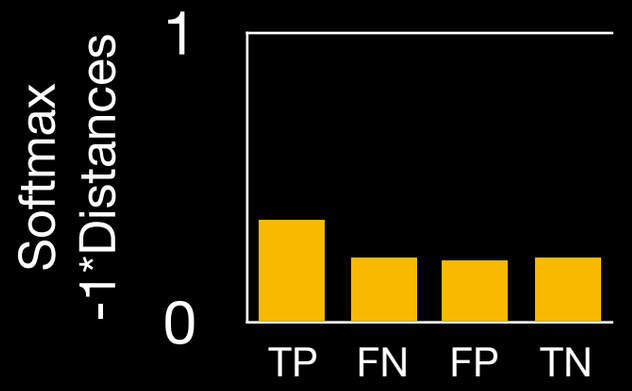
Step 3 (of 3) Cont.: Examine distances to nearest True Positives (TP), False Negatives (FN), False Positives (FP), and True Negatives (TN) from training

Exemplar Auditing: Illustrative Cases

Closest exemplar is a true positive



Closest exemplar is a true positive, BUT distribution is diffuse

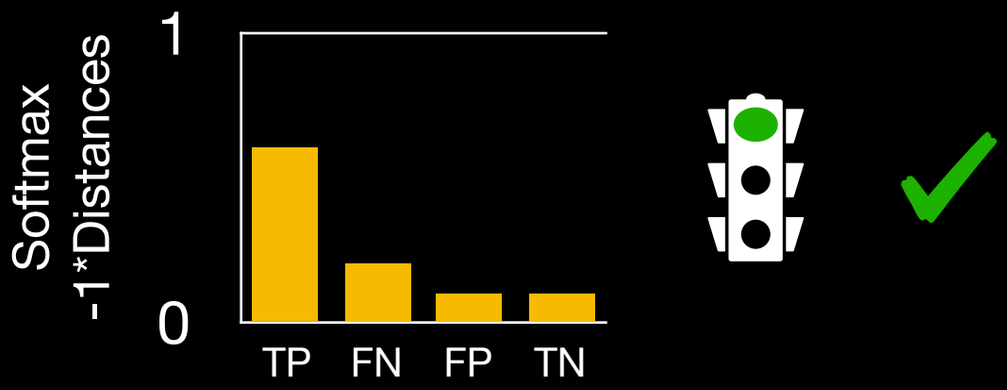


Our Solution (high-level intuition) Cont.

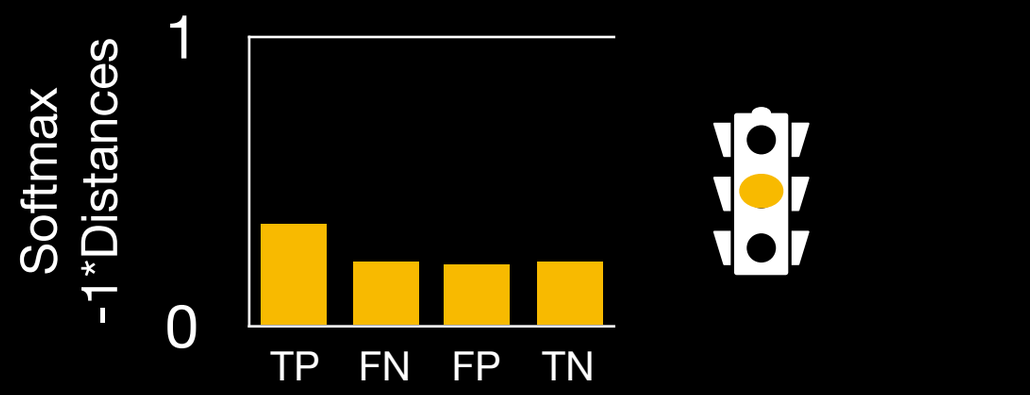
Step 3 (of 3) Cont.: Examine distances to nearest True Positives (TP), False Negatives (FN), False Positives (FP), and True Negatives (TN) from training

Exemplar Auditing: Illustrative Cases

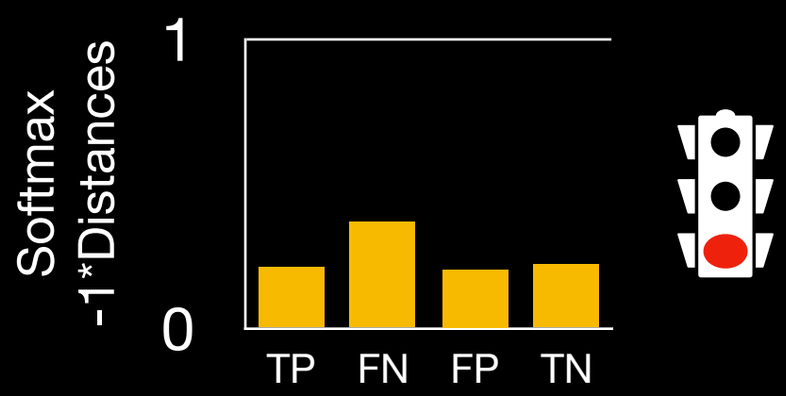
Closest exemplar is a true positive

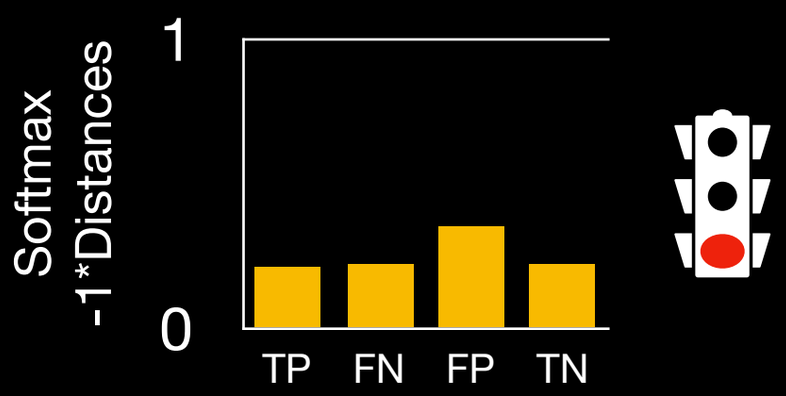


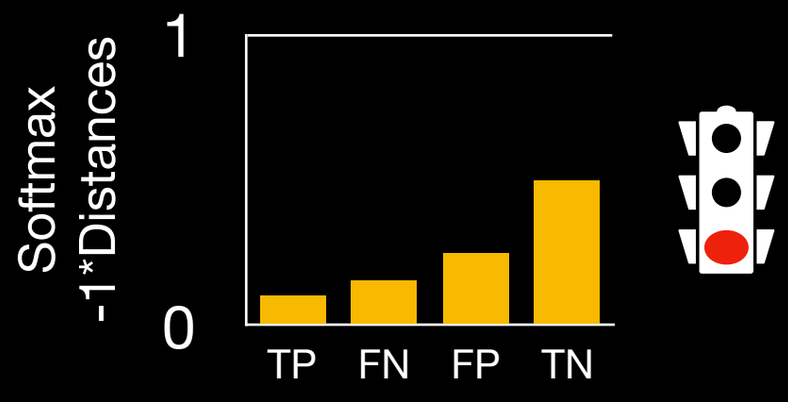
Closest exemplar is a true positive, BUT distribution is diffuse



Closest exemplar is a false negative, false positive, or true negative







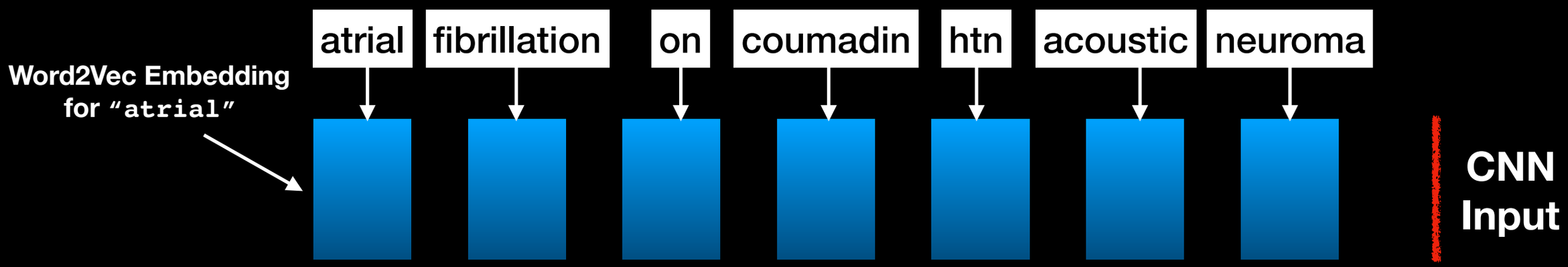
Details*

- *Actually, for simplicity, I'm going to leave out some non-trivial details, among others:
  - Exact details of fine-tuning (min-max + global norm)
  - Top k sampling
  - Label learning schedule
  - Filter widths > 1 & multiple filter widths
  - Reducing the exemplar search space

Detour:
CNN Walkthrough

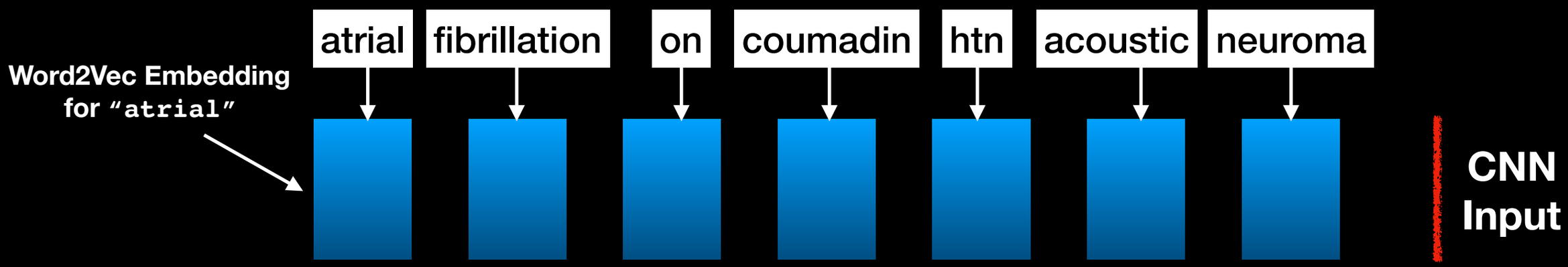
CNN Walkthrough

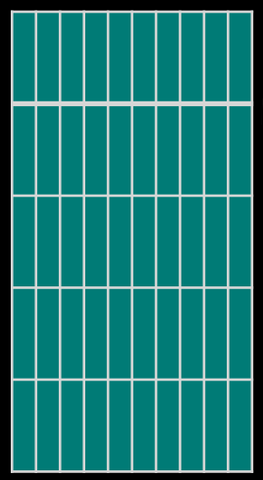
Here, input is just 7 words



CNN Walkthrough

Here, input is just 7 words

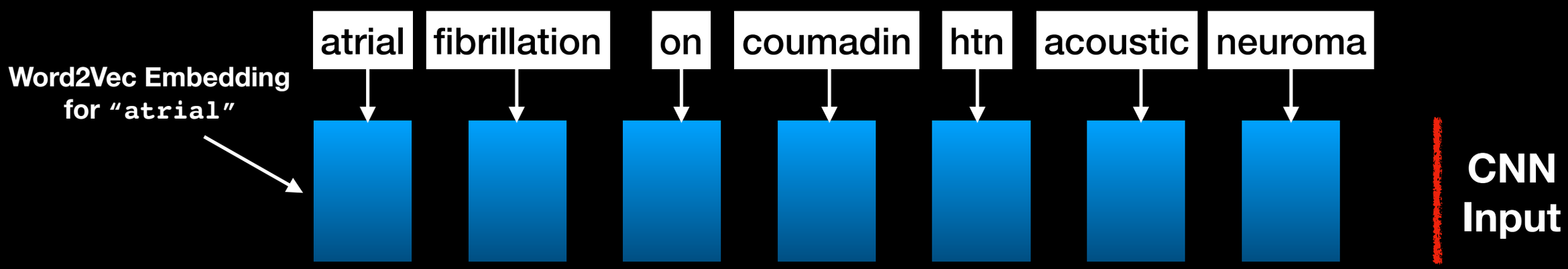


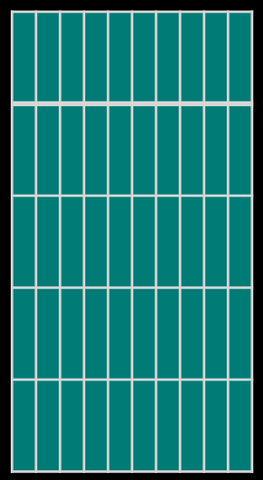


5 CNN filters of width 1 (i.e., covers entire input but no overlap across words; in practice, we also use width > 1 & multiple widths)

CNN Walkthrough

Here, input is just 7 words

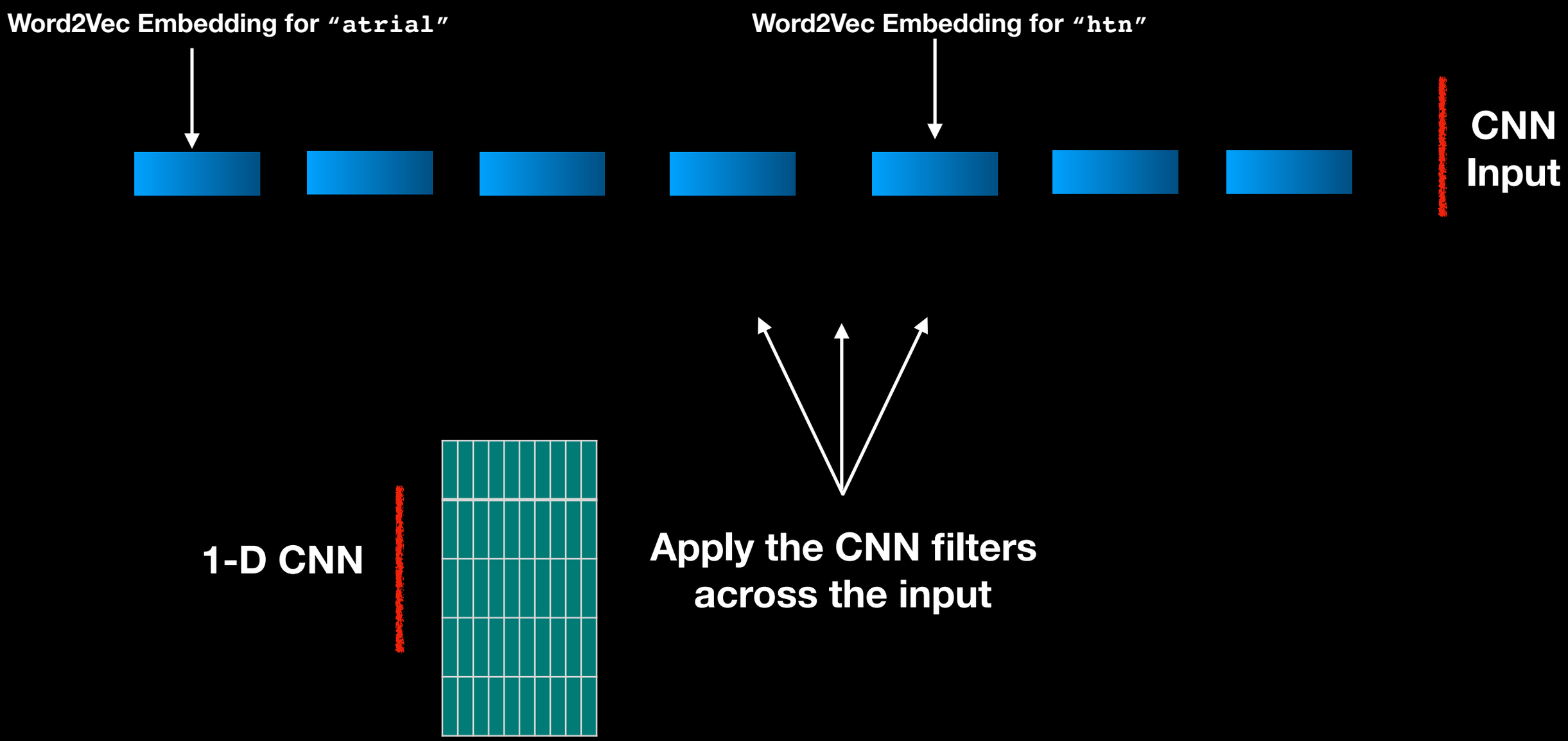




5 CNN filters of width 1 (i.e., covers entire input but no overlap across words; in practice, we also use width > 1 & multiple widths)

…These same weights are applied to all inputs.

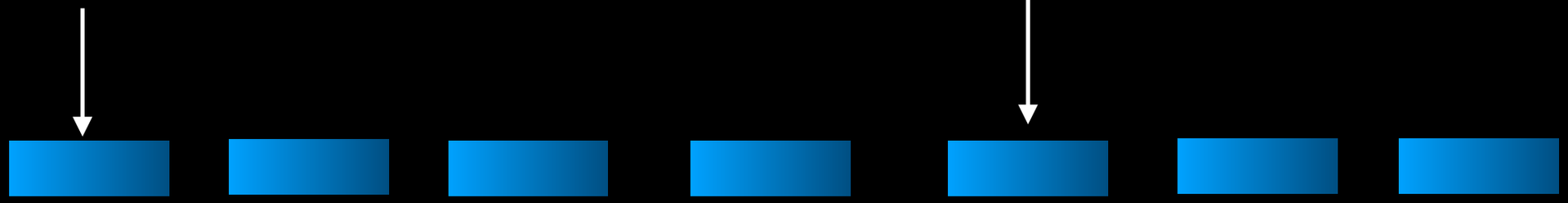
Application of CNN filter maps *for one label*



Application of CNN filter maps *for one label*

Word2Vec Embedding for `"atrial"`

Word2Vec Embedding for `"htn"`



CNN Input

After applying the filters to the input we have a matrix of 5 x 7 floats

(here, we had 5 filters, each of width 1, so we get 1 float for each input word embedding for each filter):

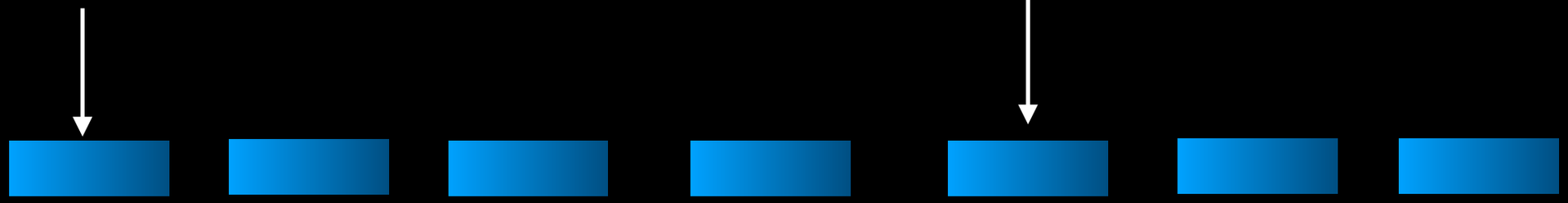
-8.9	-6.9	1.4	-1.2	9.6	-8.6	-2.6
5.1	9.7	1.1	1.1	-1.2	-7.5	-7.5
4.1	3.4	9.4	23.3	2.3	7.4	2.3
3.1	5.2	4.1	5.1	6.1	5.1	5.1
5.1	2.1	3.5	8.1	5.1	7.1	5.1

CNN Output

Application of CNN filter maps *for one label*

Word2Vec Embedding for `"atrial"`

Word2Vec Embedding for `"htn"`



CNN Input

After applying the filters to the input we have a matrix of 5 x 7 floats

(here, we had 5 filters, each of width 1, so we get 1 float for each input word embedding for each filter):

0	0	1.4	0	9.6	0	0
5.1	9.7	1.1	1.1	0	0	0
4.1	3.4	9.4	23.3	2.3	7.4	2.3
3.1	5.2	4.1	5.1	6.1	5.1	5.1
5.1	2.1	3.5	8.1	5.1	7.1	5.1

CNN Output

Apply ReLu

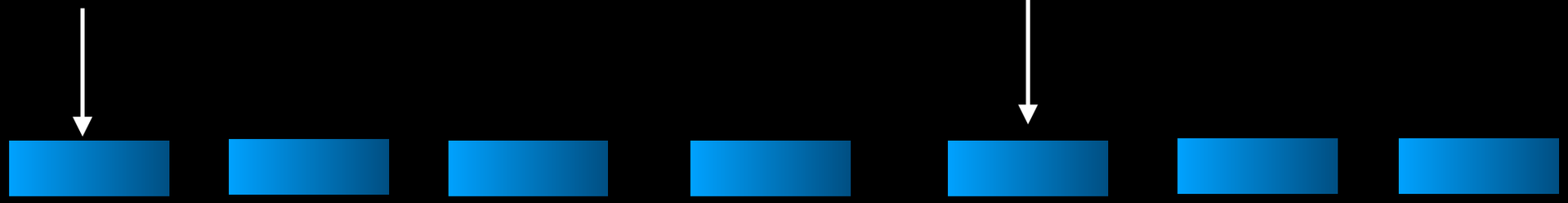
"ReLu" :

$g(x) = max(0,x)$

Application of CNN filter maps *for one label*

Word2Vec Embedding for `"atrial"`

Word2Vec Embedding for `"htn"`



CNN Input

After applying the filters to the input we have a matrix of 5 x 7 floats

(here, we had 5 filters, each of width 1, so we get 1 float for each input word embedding for each filter):

0	0	1.4	0	9.6	0	0
5.1	9.7	1.1	1.1	0	0	0
4.1	3.4	9.4	23.3	2.3	7.4	2.3
3.1	5.2	4.1	5.1	6.1	5.1	5.1
5.1	2.1	3.5	8.1	5.1	7.1	5.1

CNN Output

Apply ReLu & max pool

Results in

9.6
9.7
23.3
6.1
8.1

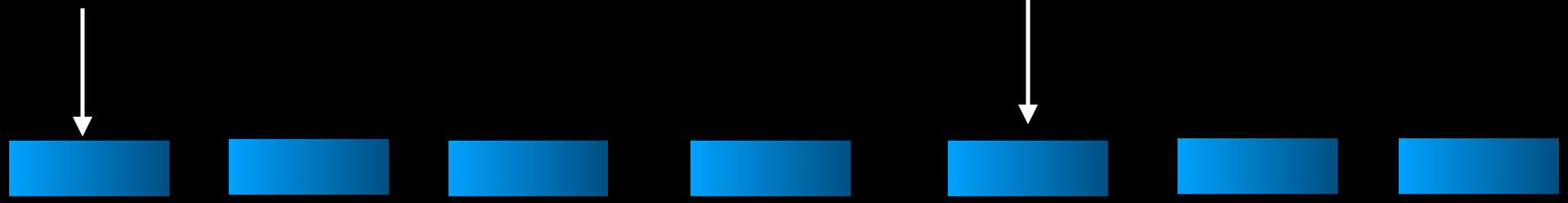
"ReLu" :

$g(x) = max(0,x)$

Application of CNN filter maps *for one label*

Word2Vec Embedding for "atrial"

Word2Vec Embedding for "htn"



CNN Input

After applying the filters to the input we have a matrix of 5 x 7 floats

(here, we had 5 filters, each of width 1, so we get 1 float for each input word embedding for each filter):

0	0	1.4	0	9.6	0	0
5.1	9.7	1.1	1.1	0	0	0
4.1	3.4	9.4	23.3	2.3	7.4	2.3
3.1	5.2	4.1	5.1	6.1	5.1	5.1
5.1	2.1	3.5	8.1	5.1	7.1	5.1

CNN Output

Apply ReLu & max pool

Results in

9.6
9.7
23.3
6.1
8.1

Linear layer

11	12	13	14	15
-51	22	23	24	25

Bias

0.2
-0.3

additional weights & biases for other labels

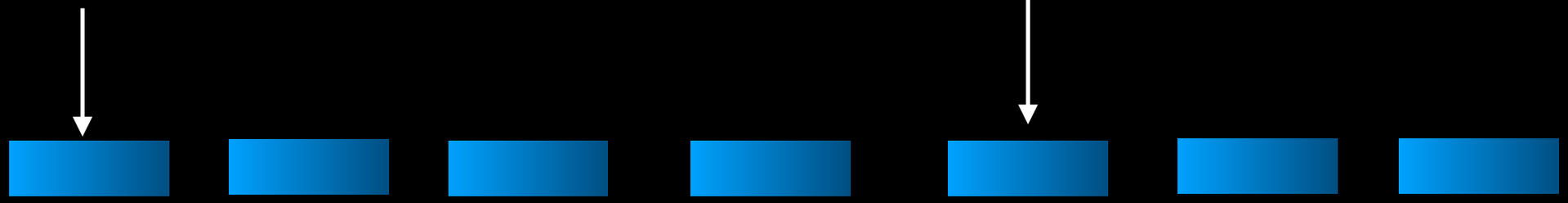
"ReLu" :

$g(x) = max(0,x)$

Application of CNN filter maps *for one label*

Word2Vec Embedding for "atrial"

Word2Vec Embedding for "htn"



CNN Input

After applying the filters to the input we have a matrix of 5 x 7 floats

(here, we had 5 filters, each of width 1, so we get 1 float for each input word embedding for each filter):

0	0	1.4	0	9.6	0	0
5.1	9.7	1.1	1.1	0	0	0
4.1	3.4	9.4	23.3	2.3	7.4	2.3
3.1	5.2	4.1	5.1	6.1	5.1	5.1
5.1	2.1	3.5	8.1	5.1	7.1	5.1

CNN Output

Apply ReLu & max pool

Results in

9.6
9.7
23.3
6.1
8.1

"ReLu" :

$g(x) = max(0,x)$

Linear layer

x

11	12	13	14	15
-51	22	23	24	25

additional weights & biases for other labels

Bias

+

0.2
-0.3

"on" state weights

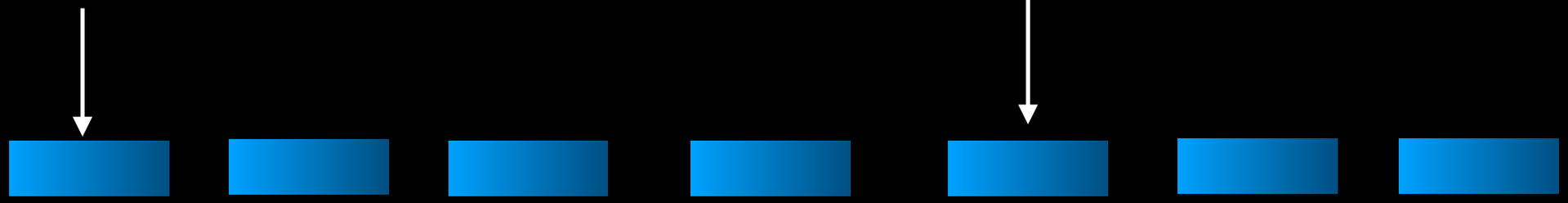
"off" state weights

$\in \mathbb{R}^{2 \cdot \{|labels|\}}$

Application of CNN filter maps *for one label*

Word2Vec Embedding for "atrial"

Word2Vec Embedding for "htn"



CNN Input

After applying the filters to the input we have a matrix of 5 x 7 floats

(here, we had 5 filters, each of width 1, so we get 1 float for each input word embedding for each filter):

0	0	1.4	0	9.6	0	0
5.1	9.7	1.1	1.1	0	0	0
4.1	3.4	9.4	23.3	2.3	7.4	2.3
3.1	5.2	4.1	5.1	6.1	5.1	5.1
5.1	2.1	3.5	8.1	5.1	7.1	5.1

CNN Output

Apply ReLu & max pool

Results in

sigmoid

9.6
9.7
23.3
6.1
8.1

"ReLu" :

$g(x) = max(0,x)$

x

Linear layer

11	12	13	14	15
-51	22	23	24	25

additional weights & biases for other labels

+

Bias

0.2
-0.3

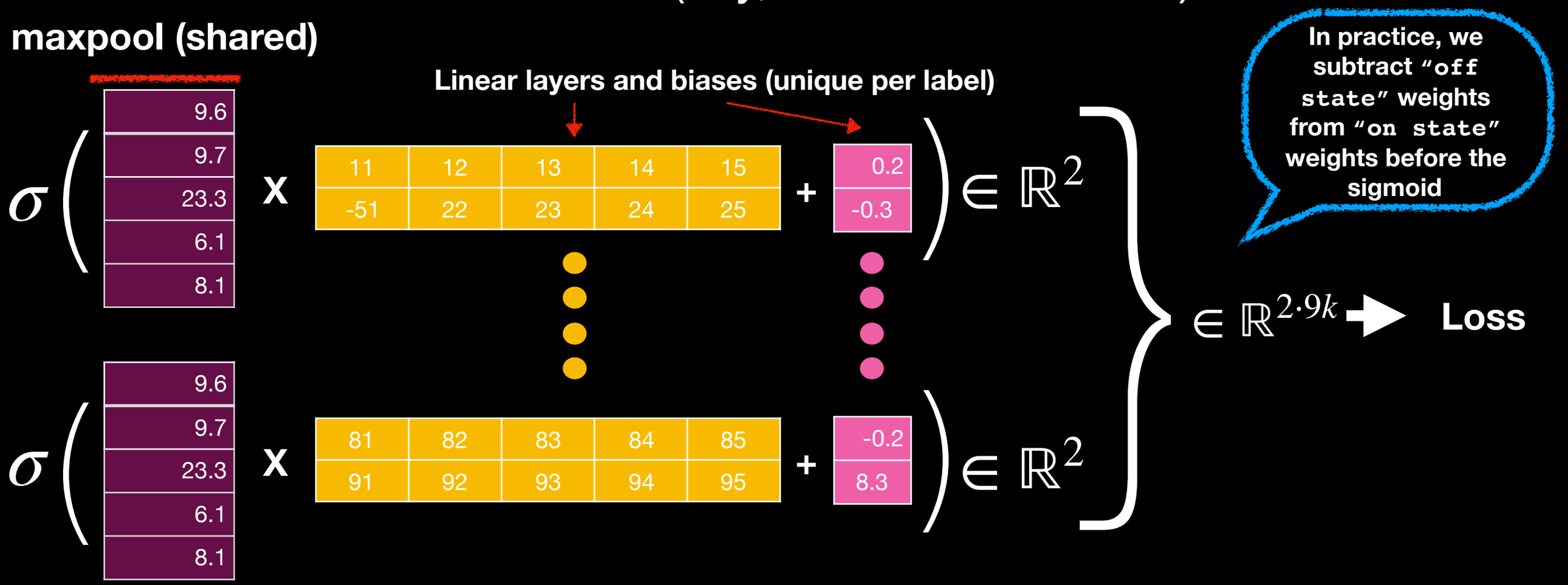
"on" state weights

"off" state weights

$\in \mathbb{R}^{2 \cdot \{|labels|\}}$

Initial Document-Level Training

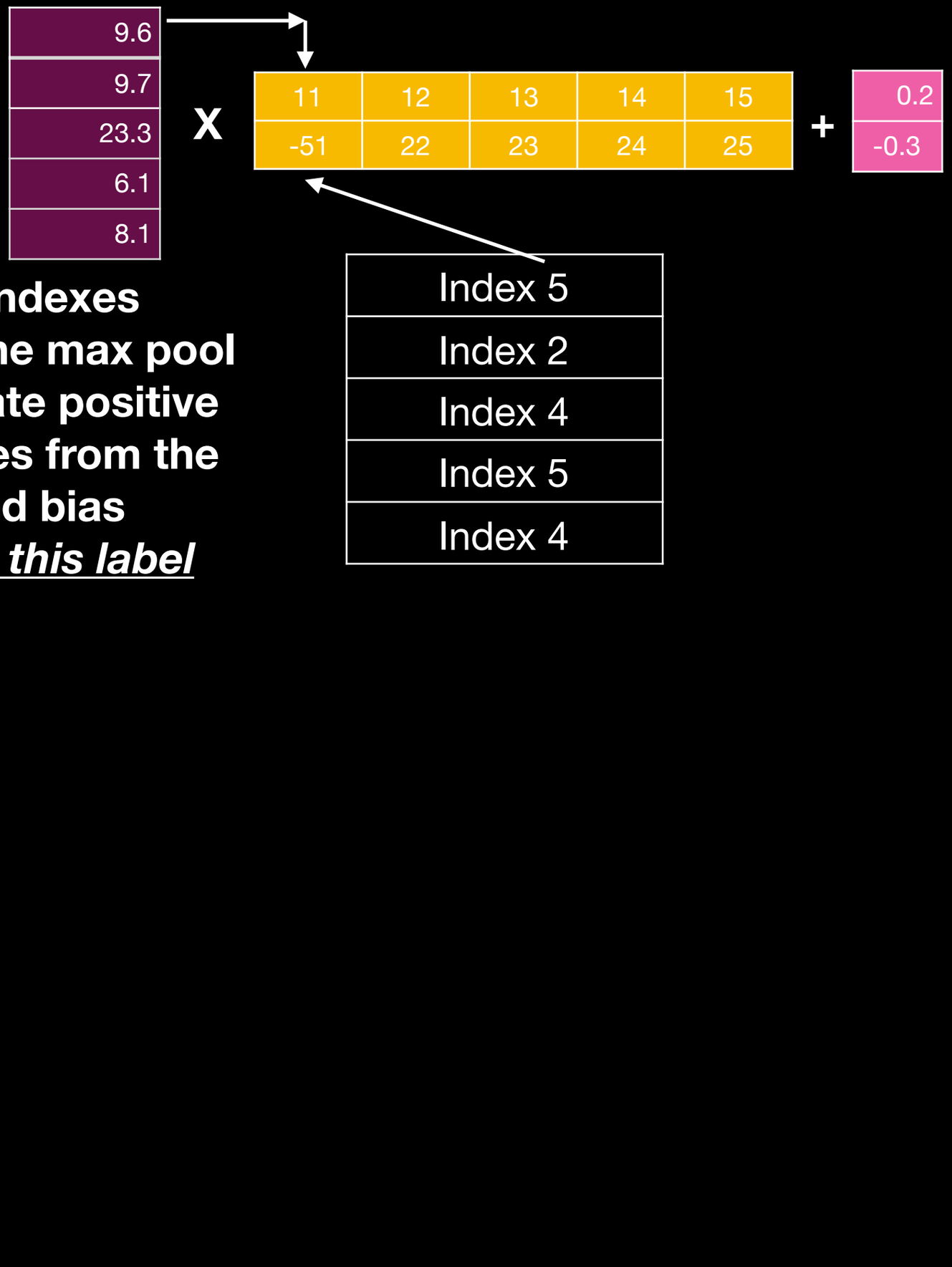
- Train with the standard binary cross-entropy loss, with document-level labels (say, we have 9k labels)



- Once trained, how do we get word-level labels?

CNN Decomposition Example

CNN Decomposition (for word-level scores *for one label*)



We retain the indexes corresponding to the max pool results and calculate positive and negative scores from the linear layer and bias *corresponding to this label*

CNN Decomposition (for word-level scores *for one label*)

We retain the indexes corresponding to the max pool results and calculate positive and negative scores from the linear layer and bias *corresponding to this label*

For example, for the word `"htn"` at index 5:

atrial fibrillation on coumadin htn acoustic neuroma

CNN Decomposition (for word-level scores *for one label*)

We retain the indexes corresponding to the max pool results and calculate positive and negative scores from the linear layer and bias *corresponding to this label*

For example, for the word "`htn`" at index 5:

atrial | fibrillation | on | coumadin | htn | acoustic | neuroma

$s_5^+ =$ 9.6 * 11 + 6.1 * 14 + 0.2 = 191.2

CNN Decomposition (for word-level scores *for one label*)

We retain the indexes corresponding to the max pool results and calculate positive and negative scores from the linear layer and bias *corresponding to this label*

For example, for the word "`htn`" at index 5:

atrial | fibrillation | on | coumadin | htn | acoustic | neuroma

$s_5^+ =$ 9.6 * 11 + 6.1 * 14 + 0.2 = 191.2

$s_5^- =$ 9.6 * -51 + 6.1 * 24 + -0.3 = -343.5

CNN Decomposition (for word-level scores *for one label*)

We retain the indexes corresponding to the max pool results and calculate positive and negative scores from the linear layer and bias *corresponding to this label*

For example, for the word "`htn`" at index 5:

atrial | fibrillation | on | coumadin | htn | acoustic | neuroma

$s_5^+ =$ 9.6 * 11 + 6.1 * 14 + 0.2 = 191.2

$s_5^- =$ 9.6 * -51 + 6.1 * 24 + -0.3 = -343.5

$$s_5^{+-} = s_5^+ - s_5^- = 534.7$$

(With width > 1 filters, we sum over all relevant application indexes & concatenate results for differing widths)

CNN Fine-tuning Example

Document-Level Fine-tuning

- Fine-tune CNN with min-max+global BCE loss
- Toy example with *1 label*: unspecified essential hypertension

Document-Level Fine-tuning

- Fine-tune CNN with min-max+global BCE loss

- Toy example with *1 label*: unspecified essential hypertension

$$s_5^{+-} = s_5^{+} - s_5^{-} = 534.7$$

atrial	fibrillation	on	coumadin	htn	acoustic	neuroma
0.0	-96.5	0.0	-313.5	534.7	0.0	0.0
s_1^{+-}	s_2^{+-}	s_3^{+-}	s_4^{+-}	s_5^{+-}	s_6^{+-}	s_7^{+-}

Document-Level Fine-tuning

- Fine-tune CNN with min-max+global BCE loss

- Toy example with *1 label*: unspecified essential hypertension

$$s_5^{+-} = s_5^{+} - s_5^{-} = 534.7$$

atrial	fibrillation	on	coumadin	htn	acoustic	neuroma
0.0	-96.5	0.0	-313.5	534.7	0.0	0.0
s_1^{+-}	s_2^{+-}	s_3^{+-}	s_4^{+-}	s_5^{+-}	s_6^{+-}	s_7^{+-}

$$L_{min}^{c} = -\log(1 - \sigma(s_{min}^{+-})) = -\log(1 - \sigma(-313.5))$$

Document-Level Fine-tuning

- Fine-tune CNN with min-max+global BCE loss

- Toy example with *1 label*: unspecified essential hypertension

$$s_5^{+-} = s_5^{+} - s_5^{-} = 534.7$$

atrial	fibrillation	on	coumadin	htn	acoustic	neuroma
0.0	-96.5	0.0	-313.5	534.7	0.0	0.0
s_1^{+-}	s_2^{+-}	s_3^{+-}	s_4^{+-}	s_5^{+-}	s_6^{+-}	s_7^{+-}

$$L_{min}^{c} = -\log(1 - \sigma(s_{min}^{+-})) = -\log(1 - \sigma(-313.5))$$

$$L_{max}^{c} = -Y \cdot \log \sigma(s_{max}^{+-}) - (1 - Y) \cdot \log(1 - \sigma(s_{max}^{+-})) = -1 \cdot \log \sigma(534.7)$$

Document-Level Fine-tuning

- Fine-tune CNN with min-max+global BCE loss

- Toy example with *1 label*: unspecified essential hypertension

$$s_5^{+-} = s_5^{+} - s_5^{-} = 534.7$$

atrial	fibrillation	on	coumadin	htn	acoustic	neuroma
0.0	-96.5	0.0	-313.5	534.7	0.0	0.0

$$s_1^{+-} \quad s_2^{+-} \quad s_3^{+-} \quad s_4^{+-} \quad s_5^{+-} \quad s_6^{+-} \quad s_7^{+-}$$

$$L_{min}^{c} = -\log(1 - \sigma(s_{min}^{+-})) = -\log(1 - \sigma(-313.5))$$

$$L_{max}^{c} = -Y \cdot \log \sigma(s_{max}^{+-}) - (1 - Y) \cdot \log(1 - \sigma(s_{max}^{+-})) = -1 \cdot \log \sigma(534.7)$$

$$o' = \left(\begin{bmatrix} 9.6 \\ 9.7 \\ 23.3 \\ 6.1 \\ 8.1 \end{bmatrix} \times \begin{bmatrix} 11 & 12 & 13 & 14 & 15 \end{bmatrix} + 0.2 - \begin{bmatrix} 9.6 \\ 9.7 \\ 23.3 \\ 6.1 \\ 8.1 \end{bmatrix} \times \begin{bmatrix} -51 & 22 & 23 & 24 & 25 \end{bmatrix} - (-0.3) \right) = 123.7$$

Document-Level Fine-tuning

- Fine-tune CNN with min-max+global BCE loss

- Toy example with *1 label*: unspecified essential hypertension

$$s_5^{+-} = s_5^+ - s_5^- = 534.7$$

atrial	fibrillation	on	coumadin	htn	acoustic	neuroma
0.0	-96.5	0.0	-313.5	534.7	0.0	0.0

$$s_1^{+-} \quad s_2^{+-} \quad s_3^{+-} \quad s_4^{+-} \quad s_5^{+-} \quad s_6^{+-} \quad s_7^{+-}$$

$$L_{min}^c = -\log(1-\sigma(s_{min}^{+-})) = -\log(1-\sigma(-313.5))$$

$$L_{max}^c = -Y \cdot \log\sigma(s_{max}^{+-}) - (1-Y)\cdot\log(1-\sigma(s_{max}^{+-})) = -1\cdot\log\sigma(534.7)$$

In practice, o' comes from a 2nd fully-connected layer

$$o' = \left(\begin{bmatrix} 9.6 \\ 9.7 \\ 23.3 \\ 6.1 \\ 8.1 \end{bmatrix} \times \begin{bmatrix} 11 & 12 & 13 & 14 & 15 \end{bmatrix} + 0.2 - \begin{bmatrix} 9.6 \\ 9.7 \\ 23.3 \\ 6.1 \\ 8.1 \end{bmatrix} \times \begin{bmatrix} -51 & 22 & 23 & 24 & 25 \end{bmatrix} - (-0.3) \right) = 123.7$$

Document-Level Fine-tuning

- Fine-tune CNN with min-max+global BCE loss

- Toy example with *1 label*: unspecified essential hypertension

$$s_5^{+-} = s_5^{+} - s_5^{-} = 534.7$$

atrial	fibrillation	on	coumadin	htn	acoustic	neuroma
0.0	-96.5	0.0	-313.5	534.7	0.0	0.0

$$s_1^{+-} \quad s_2^{+-} \quad s_3^{+-} \quad s_4^{+-} \quad s_5^{+-} \quad s_6^{+-} \quad s_7^{+-}$$

In practice, o' comes from a 2nd fully-connected layer

$$L_{min}^{c} = -\log(1 - \sigma(s_{min}^{+-})) = -\log(1 - \sigma(-313.5))$$

$$L_{max}^{c} = -Y \cdot \log \sigma(s_{max}^{+-}) - (1 - Y) \cdot \log(1 - \sigma(s_{max}^{+-})) = -1 \cdot \log \sigma(534.7)$$

$$o' = \left(\begin{bmatrix} 9.6 \\ 9.7 \\ 23.3 \\ 6.1 \\ 8.1 \end{bmatrix} \times \begin{bmatrix} 11 & 12 & 13 & 14 & 15 \end{bmatrix} + 0.2 - \begin{bmatrix} 9.6 \\ 9.7 \\ 23.3 \\ 6.1 \\ 8.1 \end{bmatrix} \times \begin{bmatrix} -51 & 22 & 23 & 24 & 25 \end{bmatrix} - -0.3 \right) = 123.7$$

$$L_{combined} = -Y \cdot \log \sigma(o' + s_{max}^{+-}) - (1 - Y) \cdot \log(1 - \sigma(o' + s_{max}^{+-})) = -1 \cdot \log \sigma(123.7 + 534.7)$$

Recap

- We've trained our network via two steps
  - Initial (standard) document-level training
  - Fine-tuning with a sparsity-encouraging loss
- From the trained model, we can now back-out word-level scores with the document-level predictions in the same manner as done for fine-tuning
- We use those word-level scores to determine where to "cut" the network
  - The resulting vectors summarize the relevant features

Exemplar vectors

maxpool	atrial	fibrillation	on	coumadin	htn	acoustic	neuroma
9.6	-8.9	-6.9	1.4	-1.2	9.6	-8.6	-2.6
9.7	5.1	9.7	1.1	1.1	-1.2	-7.5	-7.5
23.3	4.1	3.4	9.4	23.3	2.3	7.4	2.3
6.1	3.1	5.2	4.1	5.1	6.1	5.1	5.1
8.1	5.1	2.1	3.5	8.1	5.1	7.1	5.1

CNN Output

$(filter\ width = 1)$

$\mathbf{v}^{maxpool}$

$\mathbf{v}_5$

Exemplar vectors

maxpool	atrial	fibrillation	on	coumadin	htn	acoustic	neuroma
9.6	-8.9	-6.9	1.4	-1.2	9.6	-8.6	-2.6
9.7	5.1	9.7	1.1	1.1	-1.2	-7.5	-7.5
23.3	4.1	3.4	9.4	23.3	2.3	7.4	2.3
6.1	3.1	5.2	4.1	5.1	6.1	5.1	5.1
8.1	5.1	2.1	3.5	8.1	5.1	7.1	5.1

CNN Output

$(filter\ width = 1)$

$\mathbf{v}^{maxpool}$ ← *maxpool*

$\mathbf{v}_5$ ← htn

$$\mathbf{v}_5^{test} = \mathbf{v}_5, \mathbf{v}^{maxpool} =$$

9.6
-1.2
2.3
6.1
5.1
9.6
9.7
23.3
6.1
8.1

← **concatenate**

Exemplar vectors

maxpool	atrial	fibrillation	on	coumadin	htn	acoustic	neuroma
9.6	-8.9	-6.9	1.4	-1.2	9.6	-8.6	-2.6
9.7	5.1	9.7	1.1	1.1	-1.2	-7.5	-7.5
23.3	4.1	3.4	9.4	23.3	2.3	7.4	2.3
6.1	3.1	5.2	4.1	5.1	6.1	5.1	5.1
8.1	5.1	2.1	3.5	8.1	5.1	7.1	5.1

CNN Output

$(filter\ width = 1)$

$\mathbf{v}^{maxpool}$

$\mathbf{v}_5$

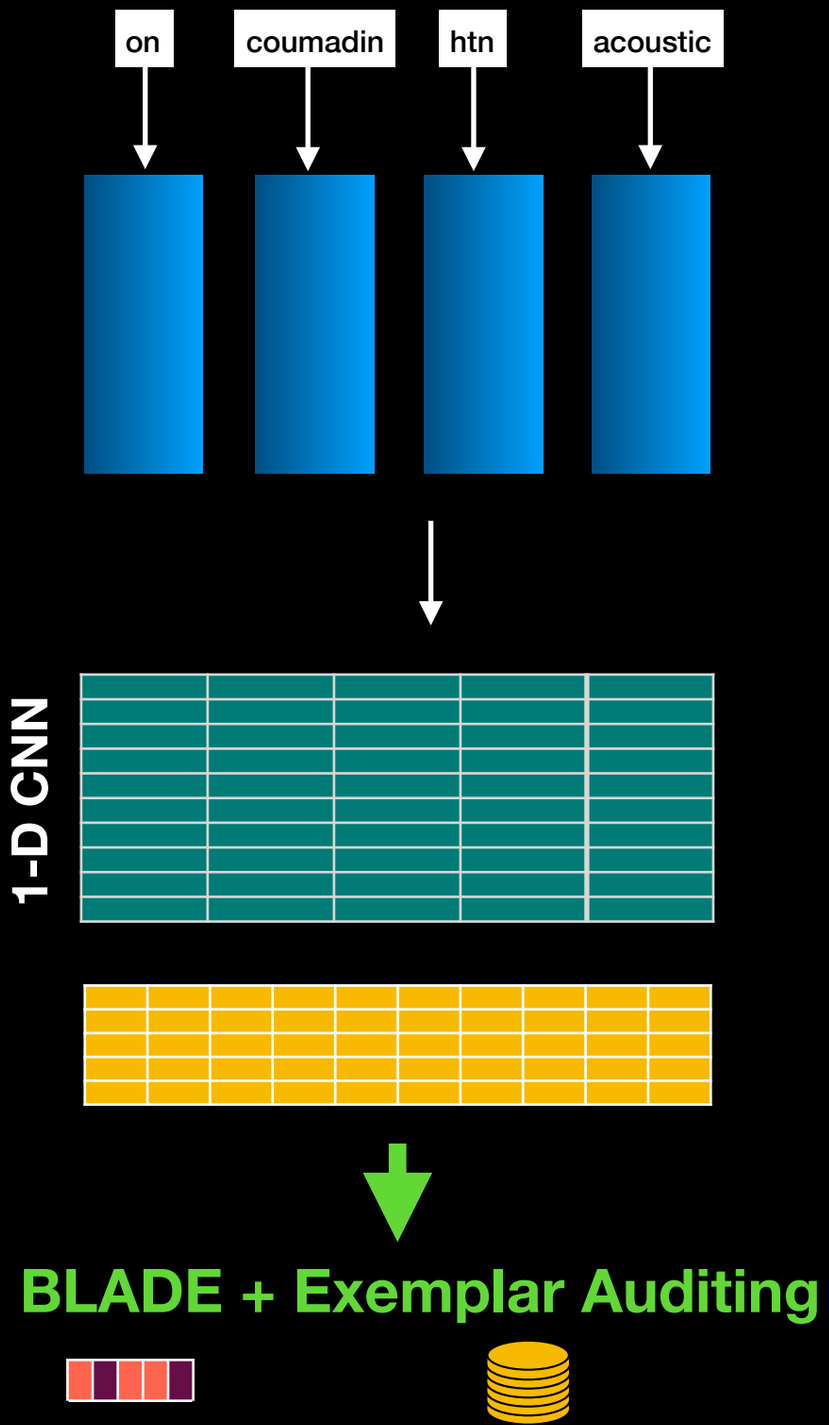
9.6
-1.2
2.3
6.1
5.1
9.6
9.7
23.3
6.1
8.1

$\mathbf{v}_5^{test} = \mathbf{v}_5, \mathbf{v}^{maxpool} =$ ← **concatenate**

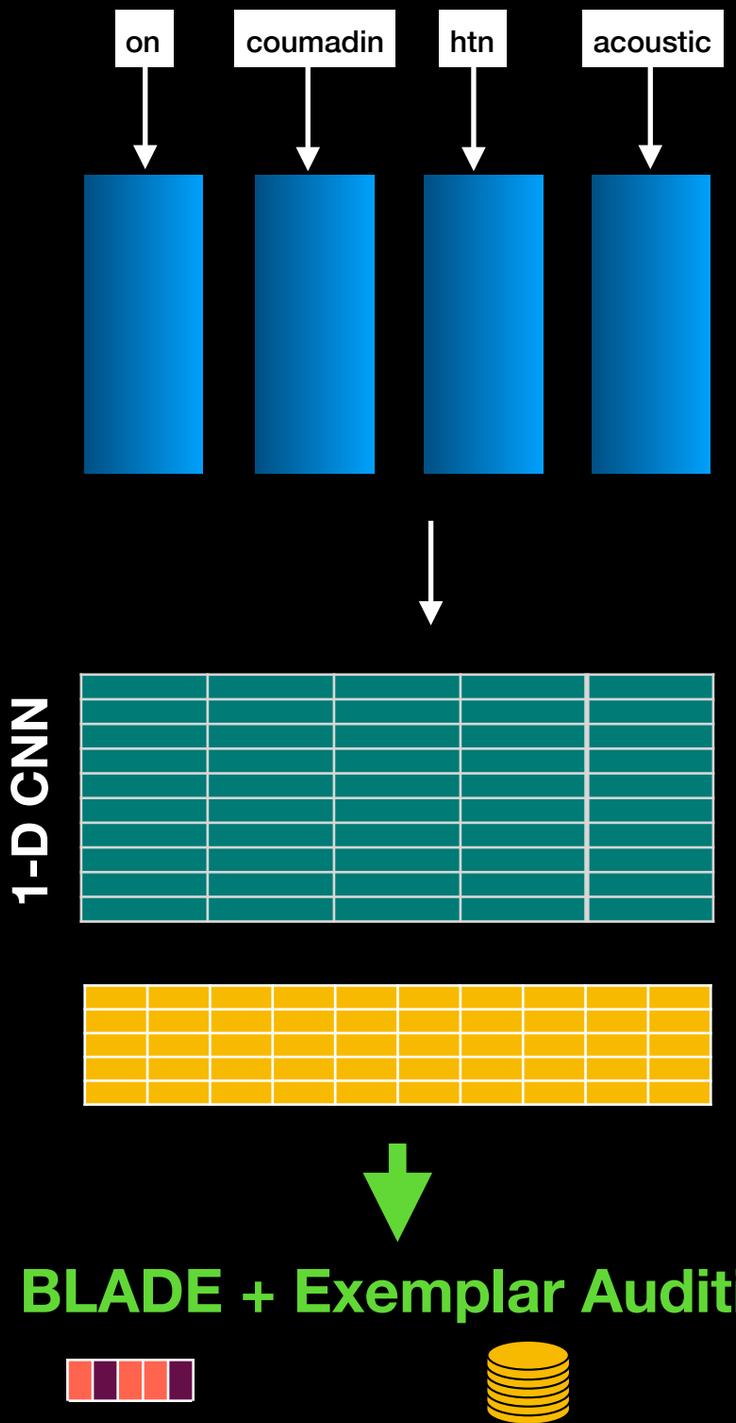
Find:

$$\underset{\mathbf{v}^{train}}{argmin} \|\mathbf{v}_5^{test} - \mathbf{v}^{train}\|_2$$

Our Solution (Summary)

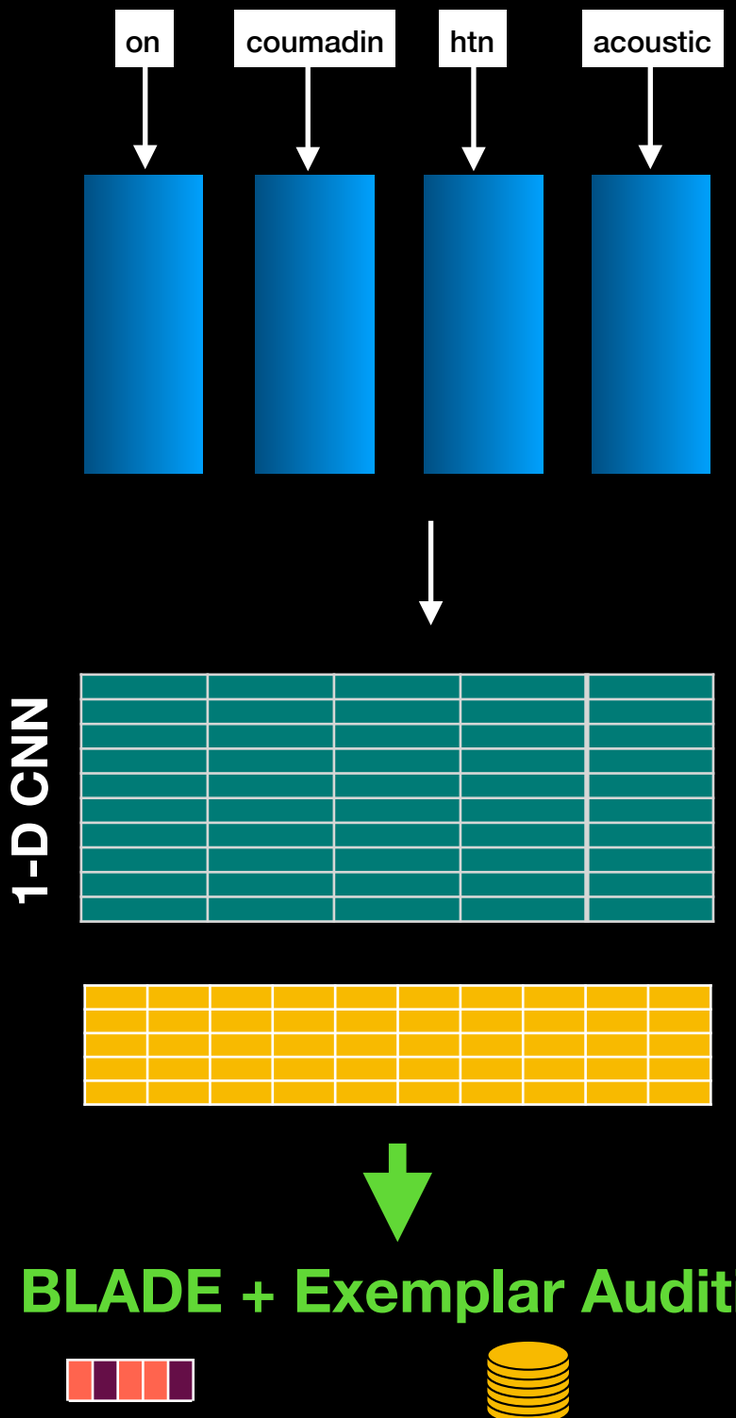


Our Solution (Summary)



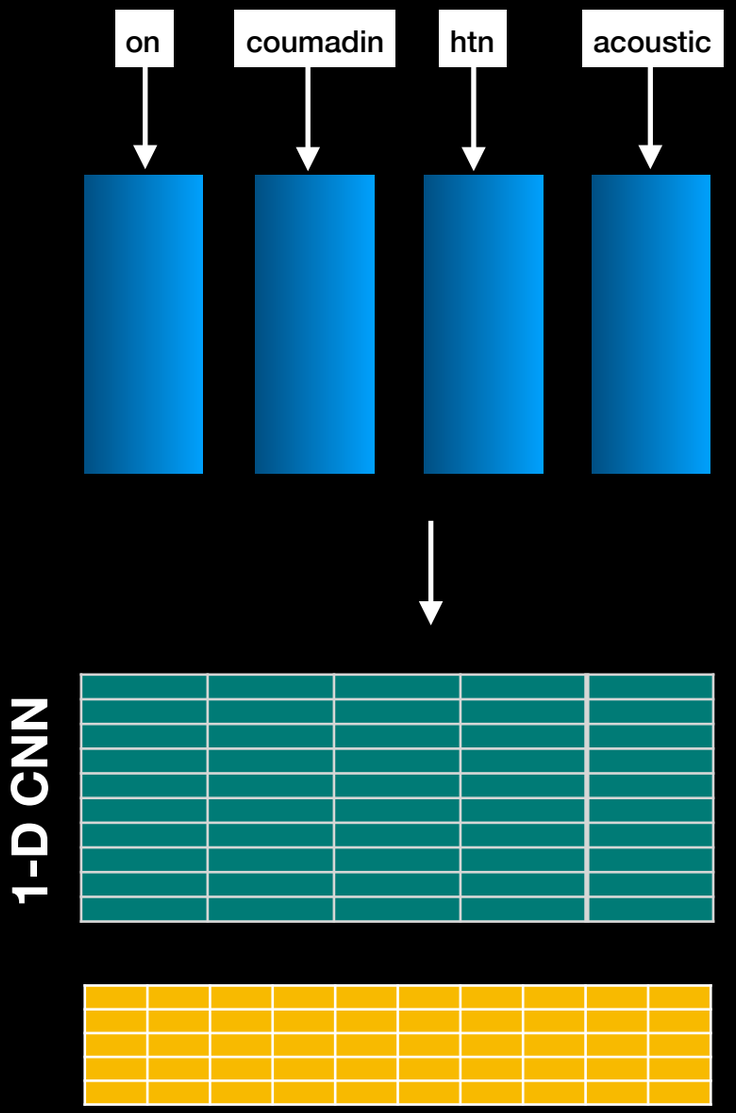
- **Classification effectiveness**

Our Solution (Summary)



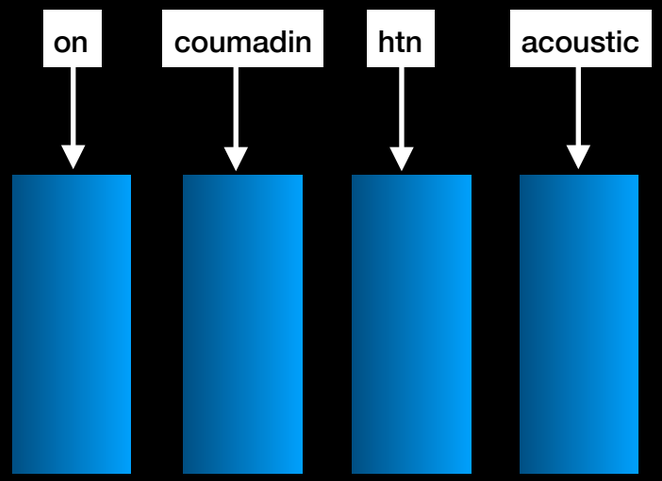
- **Classification effectiveness** ✓

Our Solution (Summary)



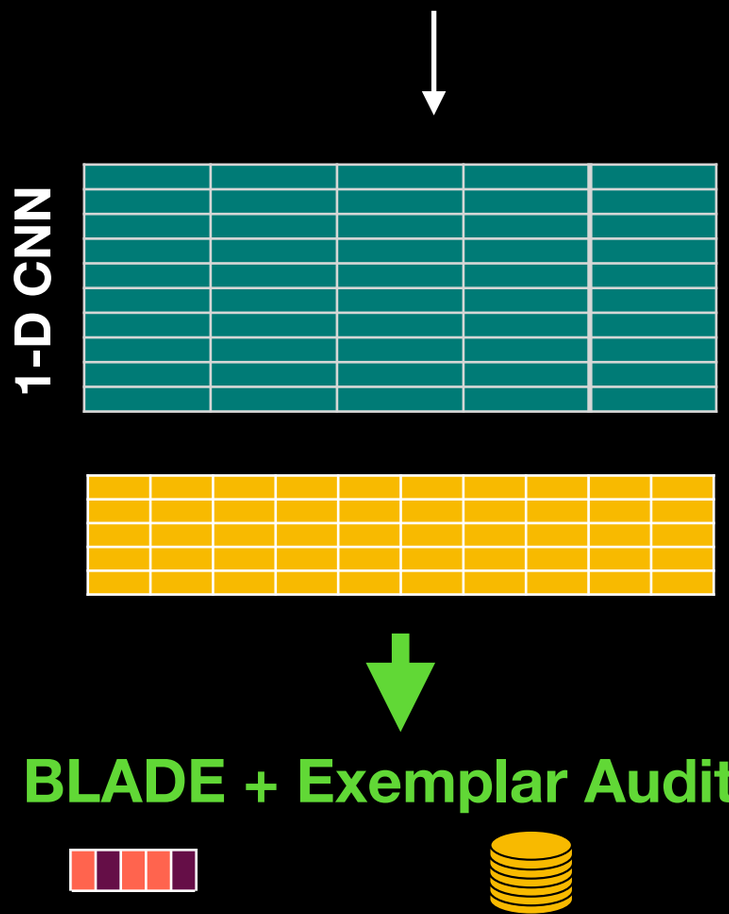
- **Classification effectiveness** ✓
- **Interpretability**

Our Solution (Summary)

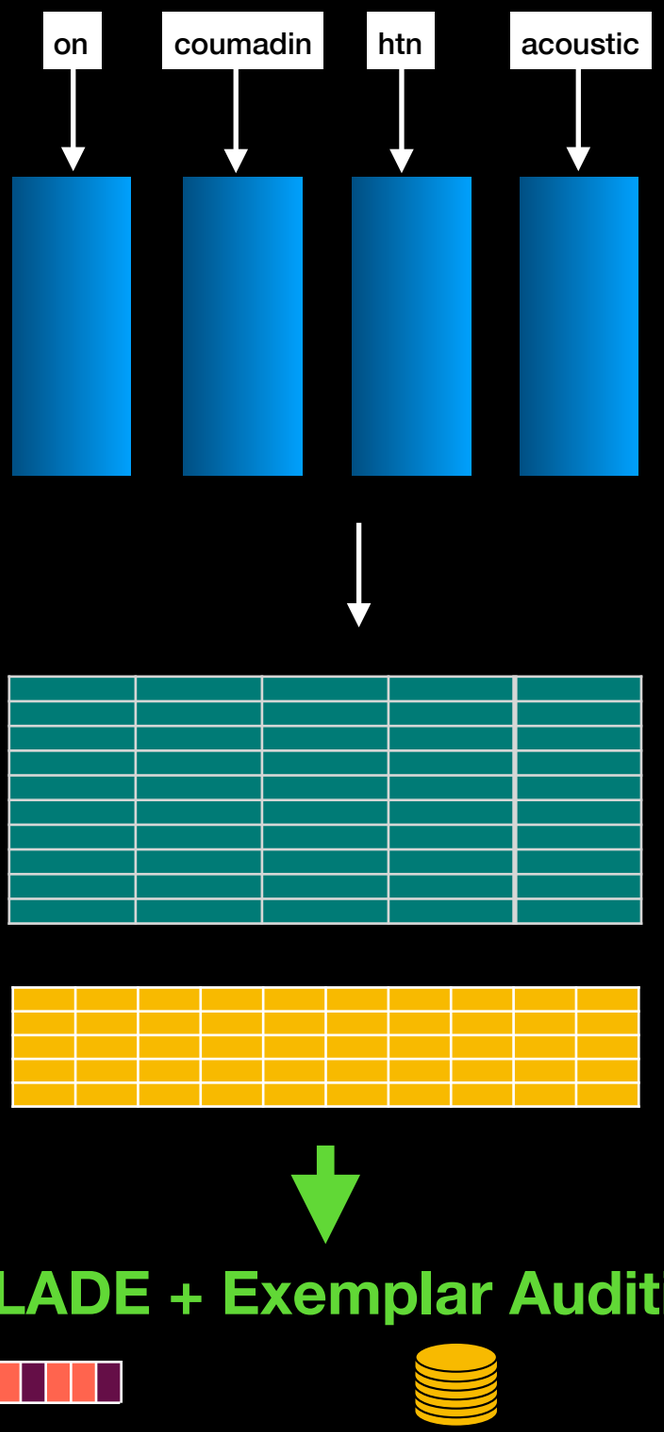


- **Classification effectiveness** ✓
- **Interpretability** ? / ✓

(Depends on def. & task, but yes insofar that we have an approach for resolving the global norm issue)

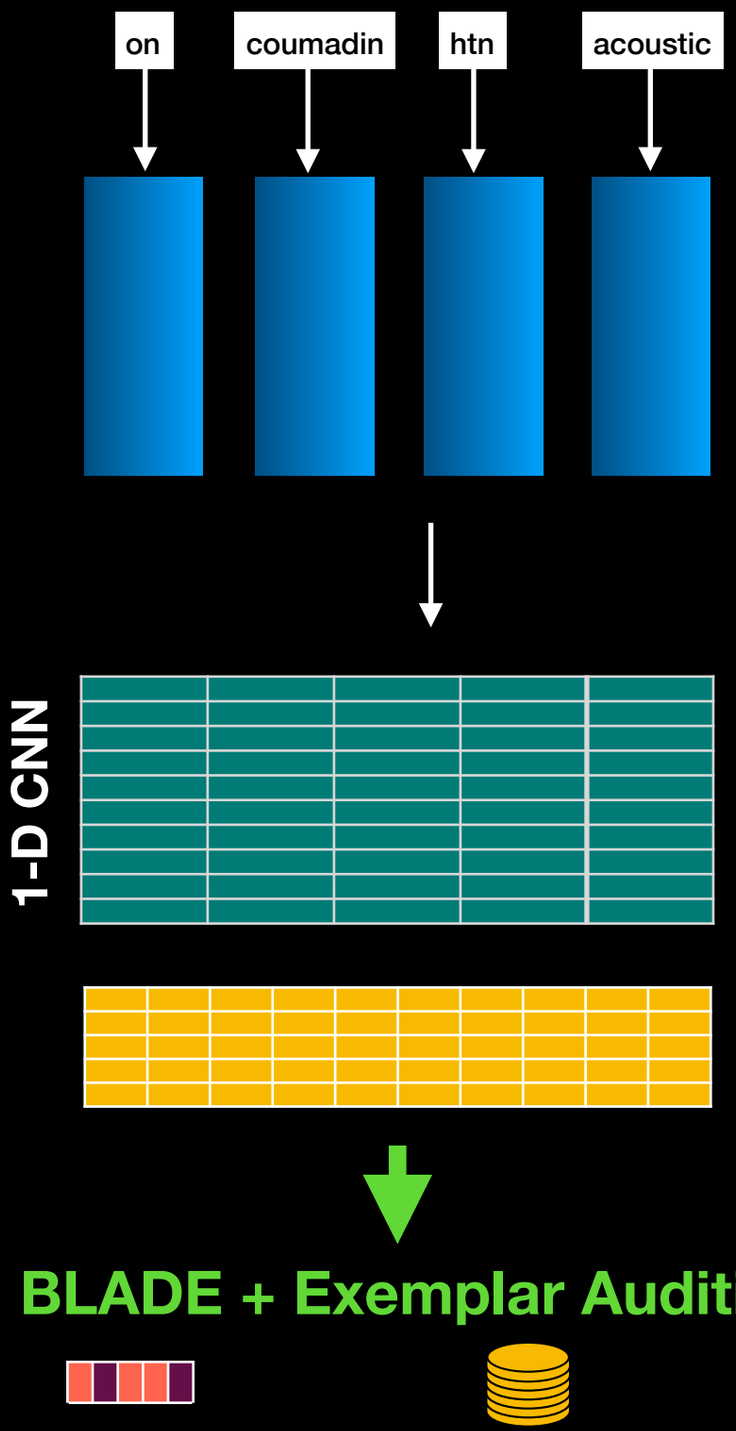


Our Solution (Summary)



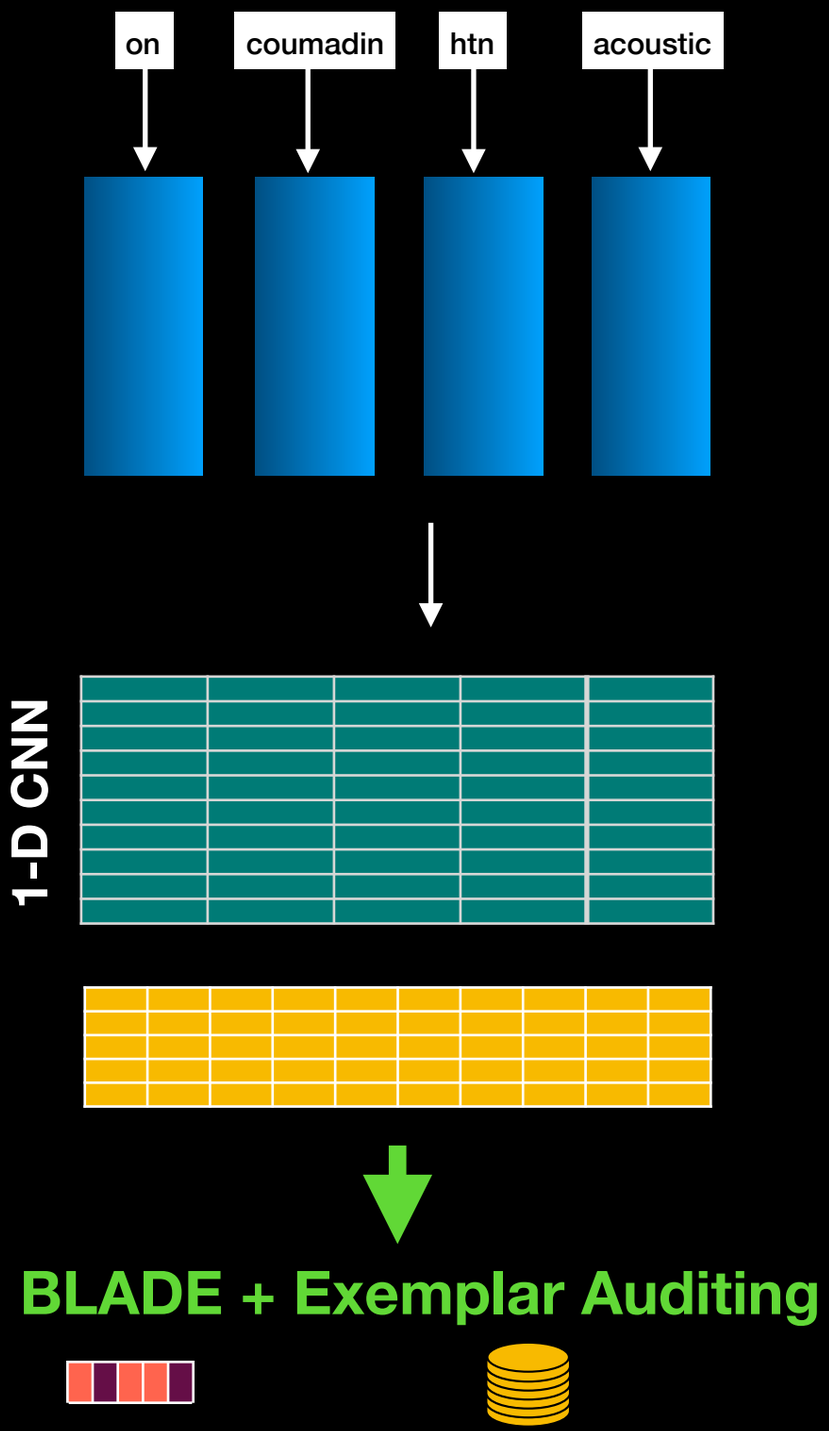
- **Classification effectiveness** ✓
- **Interpretability** ? / ✓ (Depends on def. & task, but yes insofar that we have an approach for resolving the global norm issue)
- **Aid for decision making at lower granularities/resolutions**

Our Solution (Summary)



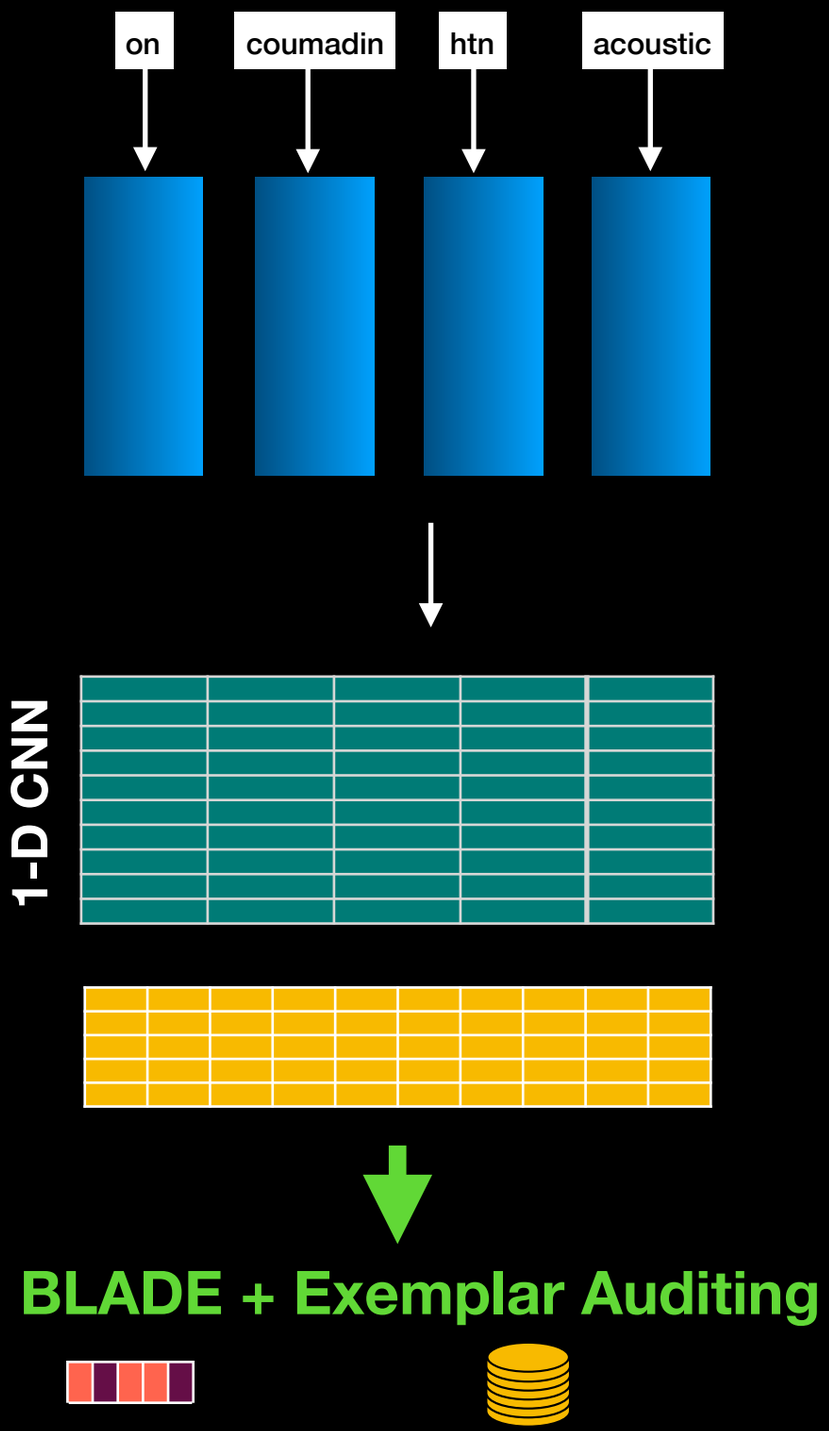
- **Classification effectiveness** ✓
- **Interpretability** ? / ✓ (Depends on def. & task, but yes insofar that we have an approach for resolving the global norm issue)
- **Aid for decision making at lower granularities/resolutions** ✓

Our Solution (Summary)



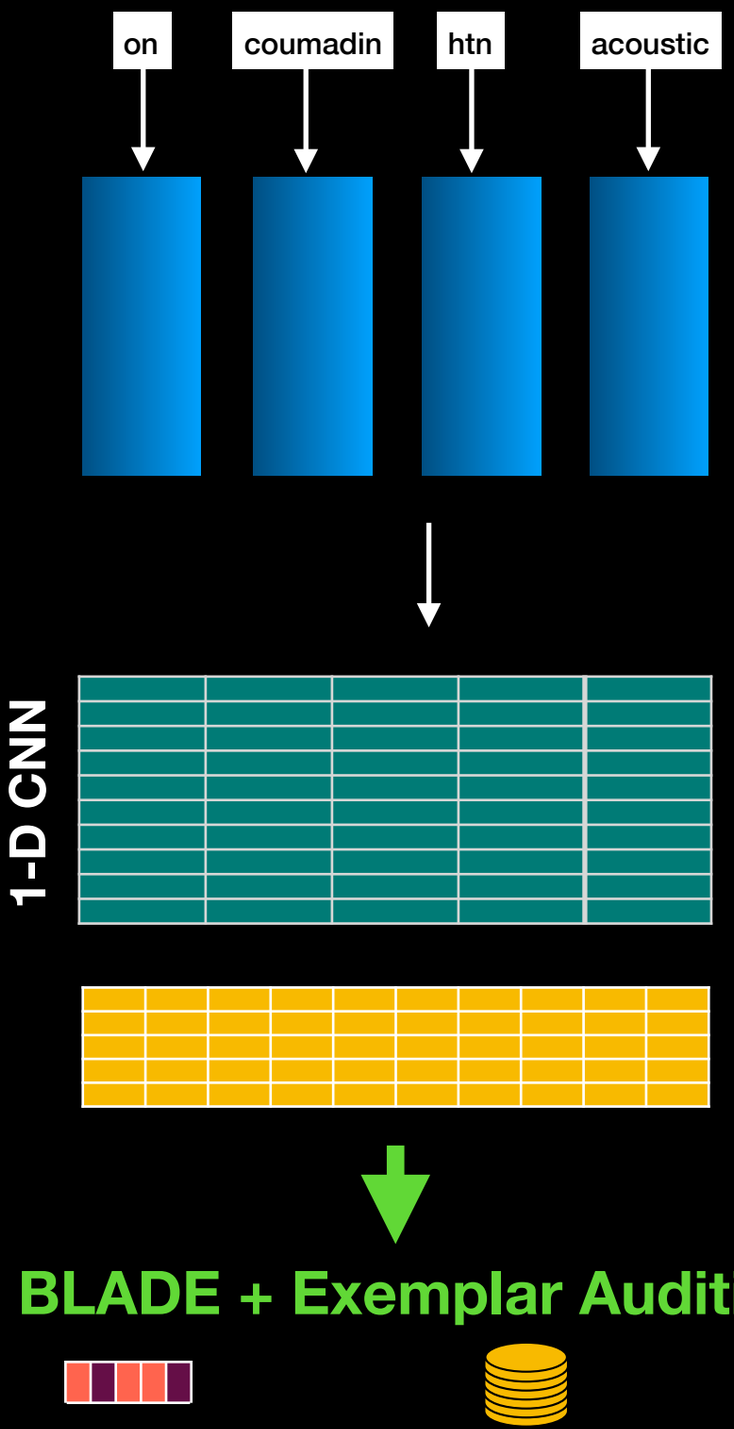
- **Classification effectiveness** ✓
- **Interpretability** ? / ✓ (Depends on def. & task, but yes insofar that we have an approach for resolving the global norm issue)
- **Aid for decision making at lower granularities/resolutions** ✓
- **Verify findings with ground-truth training labels**

Our Solution (Summary)



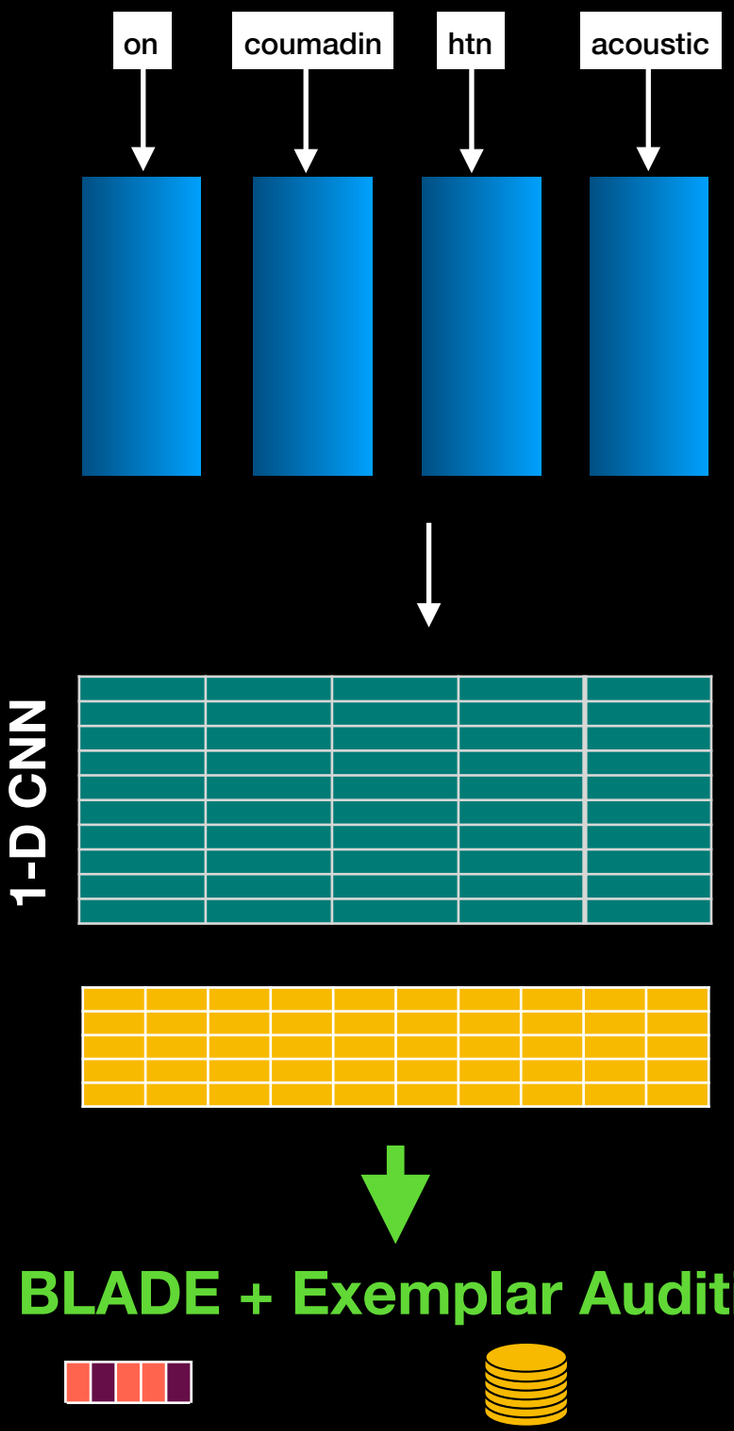
- **Classification effectiveness** ✓
- **Interpretability** ? / ✓ (Depends on def. & task, but yes insofar that we have an approach for resolving the global norm issue)
- **Aid for decision making at lower granularities/resolutions** ✓
- **Verify findings with ground-truth training labels** ✓

Our Solution (Summary)



- **Classification effectiveness** ✓
- **Interpretability** ? / ✓ (Depends on def. & task, but yes insofar that we have an approach for resolving the global norm issue)
- **Aid for decision making at lower granularities/resolutions** ✓
- **Verify findings with ground-truth training labels** ✓
- **Parameter identification**

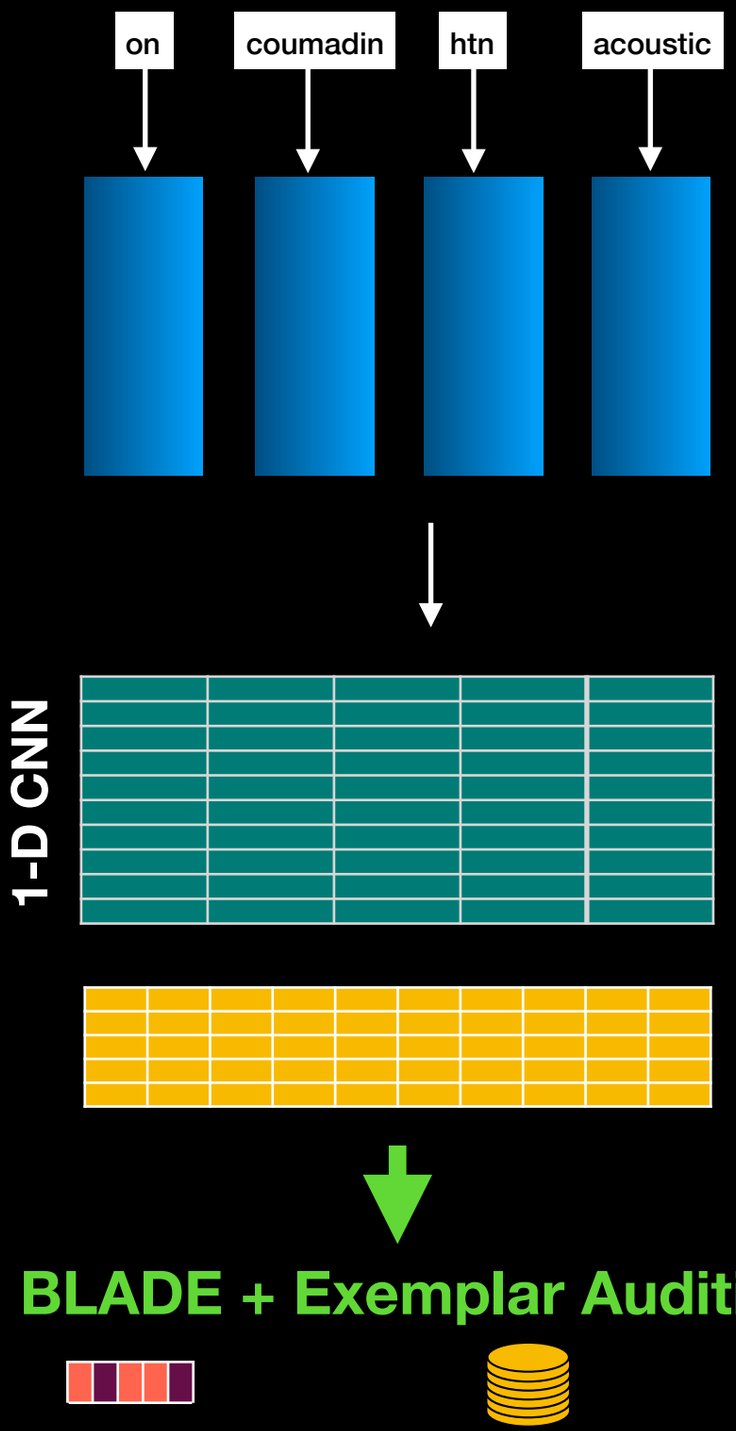
Our Solution (Summary)



- **Classification effectiveness** ✓
- **Interpretability** ? / ✓

  (Depends on def. & task, but yes insofar that we have an approach for resolving the global norm issue)
- **Aid for decision making at lower granularities/resolutions** ✓
- **Verify findings with ground-truth training labels** ✓
- **Parameter identification** ✗

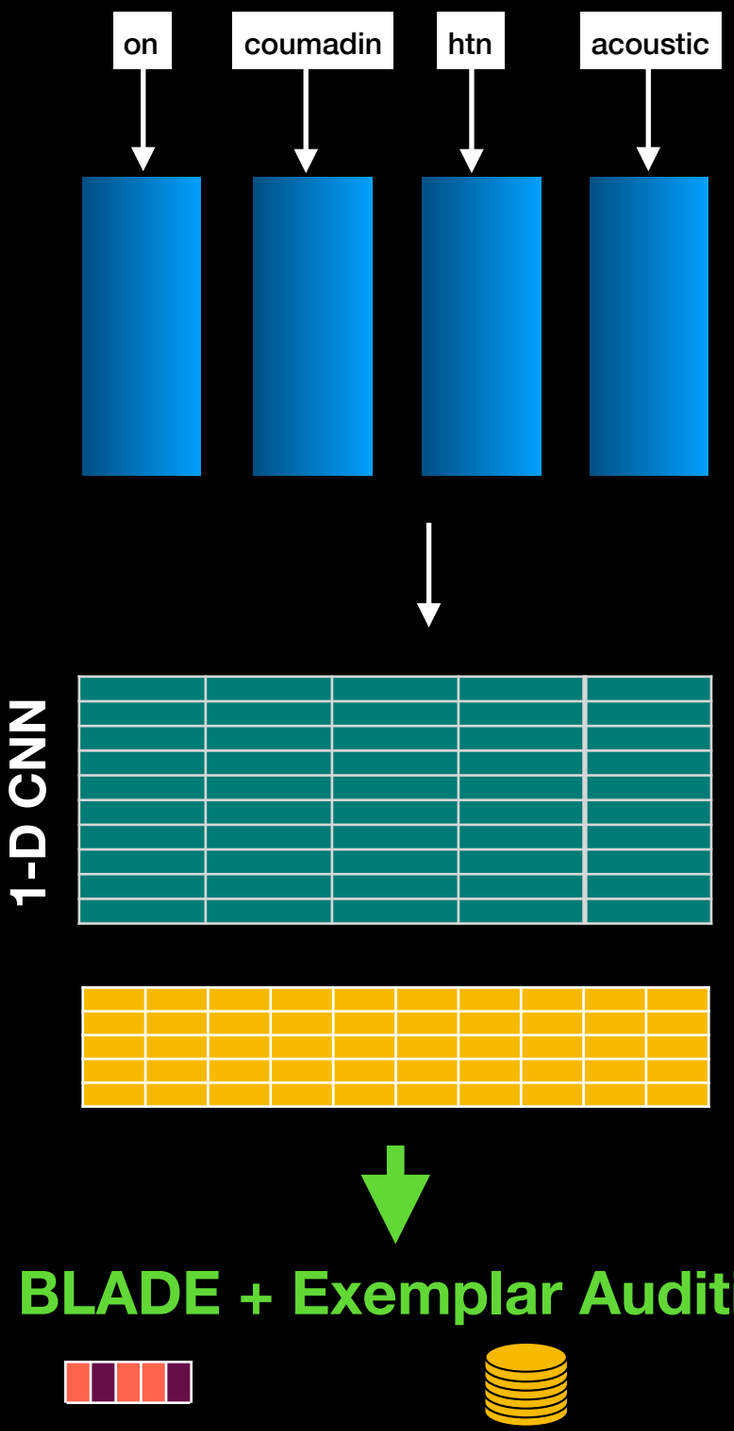
Our Solution (Summary)



- **Classification effectiveness** ✓
- **Interpretability** ? / ✓

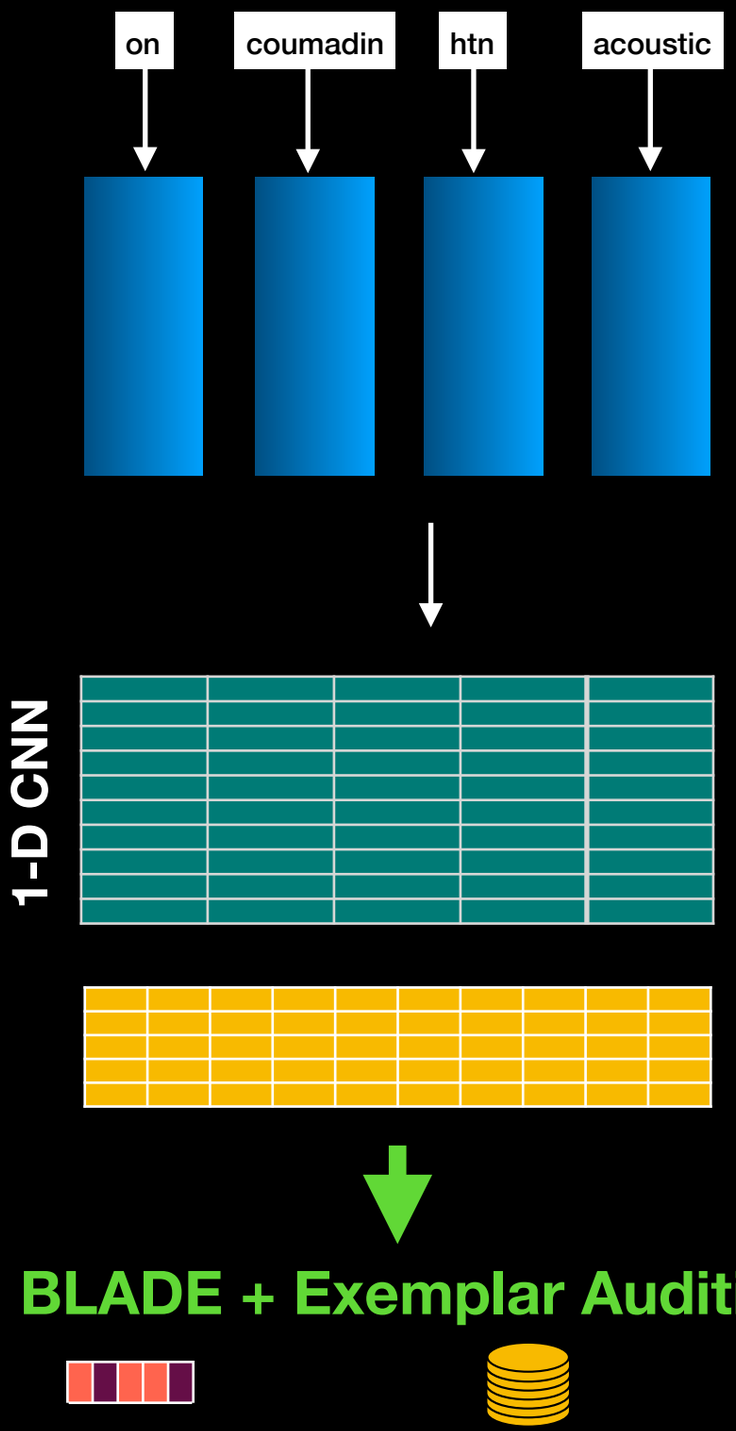
  (Depends on def. & task, but yes insofar that we have an approach for resolving the global norm issue)
- **Aid for decision making at lower granularities/resolutions** ✓
- **Verify findings with ground-truth training labels** ✓
- **Parameter identification** ✗
- **Robustness to adversarial attacks**

Our Solution (Summary)



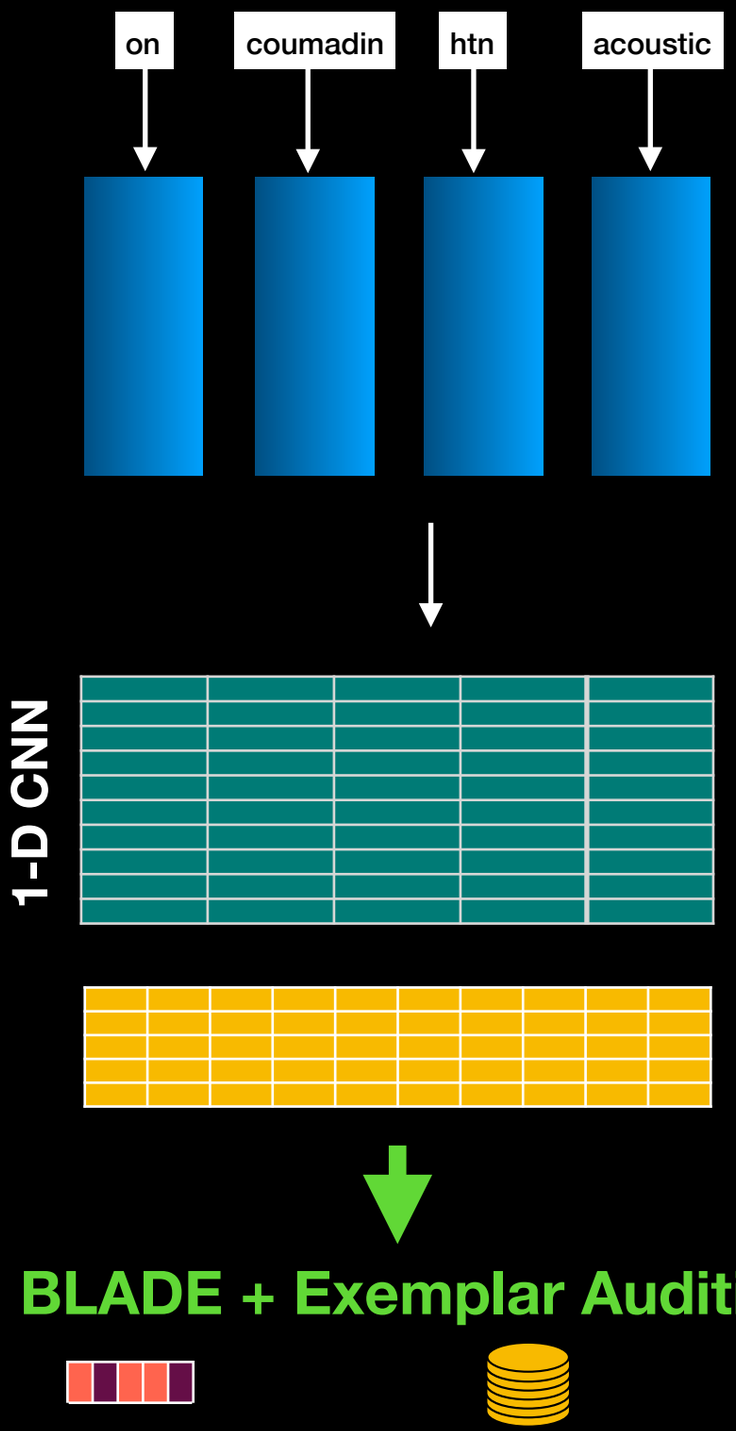
- **Classification effectiveness** ✓
- **Interpretability** ? / ✓ (Depends on def. & task, but yes insofar that we have an approach for resolving the global norm issue)
- **Aid for decision making at lower granularities/resolutions** ✓
- **Verify findings with ground-truth training labels** ✓
- **Parameter identification** ✗
- **Robustness to adversarial attacks** ? (Maybe) ✗ (esp. with training set access)

Our Solution (Summary)



- **Classification effectiveness** ✓
- **Interpretability** ? / ✓ (Depends on def. & task, but yes insofar that we have an approach for resolving the global norm issue)
- **Aid for decision making at lower granularities/resolutions** ✓
- **Verify findings with ground-truth training labels** ✓
- **Parameter identification** ✗
- **Robustness to adversarial attacks** ? (Maybe) ✗ (esp. with training set access)
- **Efficiency (space/time)**

Our Solution (Summary)



- **Classification effectiveness** ✓
- **Interpretability** ? / ✓ (Depends on def. & task, but yes insofar that we have an approach for resolving the global norm issue)
- **Aid for decision making at lower granularities/resolutions** ✓
- **Verify findings with ground-truth training labels** ✓
- **Parameter identification** ✗
- **Robustness to adversarial attacks** ? (Maybe) ✗ (esp. with training set access)
- **Efficiency (space/time)** ✗

Current Work

- Extension to additional tasks (and input modalities)
  - Tabular data/etc.
  - Symbolic extensions
    - Can we induce (stochastic) rule sets?
- QA

Questions?