Deep Networks as *hidden* Metric Learners

- N training instances: $x_1, \ldots, x_n, \ldots, x_N$

- Ground truth training labels: $y_1, \ldots, y_n, \ldots, y_N$

- Seek a function, $f: \mathbb{X} \rightarrow \mathbb{Y}$, to predict $\hat{y}_{N+1}$ for a new, unseen instance x_{N+1}, with minimal *distance* between $\hat{y}_{N+1}$ and y_{N+1}

- New view: Back-out a metric learner from the parametric deep network:
  $f = c \circ g$, where $g: \mathbb{X} \rightarrow \mathbb{R}^M$, $c: \mathbb{R}^M \rightarrow \mathbb{Y}$, and $r \in \mathbb{R}^M$ is a dense representation of the input under the parametric model

- Sense in which: $f\left(x_{N+1}\right) \approx \beta + \sum_{n=1}^{N} \left(\tanh(f(x_n)) + \gamma \cdot y_n\right) \cdot w\left(||r_n - r_{N+1}||_2\right)$

  I.e., a test prediction is approx. a distance-weighting (between "***exemplar***" representations) over the training set (model predictions & associated labels)

  $w(\cdot)$ is a function of the distance between representations (Relatable to instance-based learning, kernel methods, …)

- Enables interpretable/introspectable decision rules & various analyses (hence, "***auditing***"): E.g., only admit true positive (TP) matches:
  $\hat{y}_{N+1} = f(x_{N+1}) \cdot \left[f(x_{N+1}) = f(x_n) \wedge f(x_n) = y_n\right] + NULL \cdot \left[f(x_{N+1}) \neq f(x_n) \vee f(x_n) \neq y_n\right]$, where $n = \underset{n \in \{1,\ldots,N\}}{\arg\min} ||r_n - r_{N+1}||_2$

- Enables updatability/adaptability:

  - Label changes: $y_n' = y_n + \Delta_n$

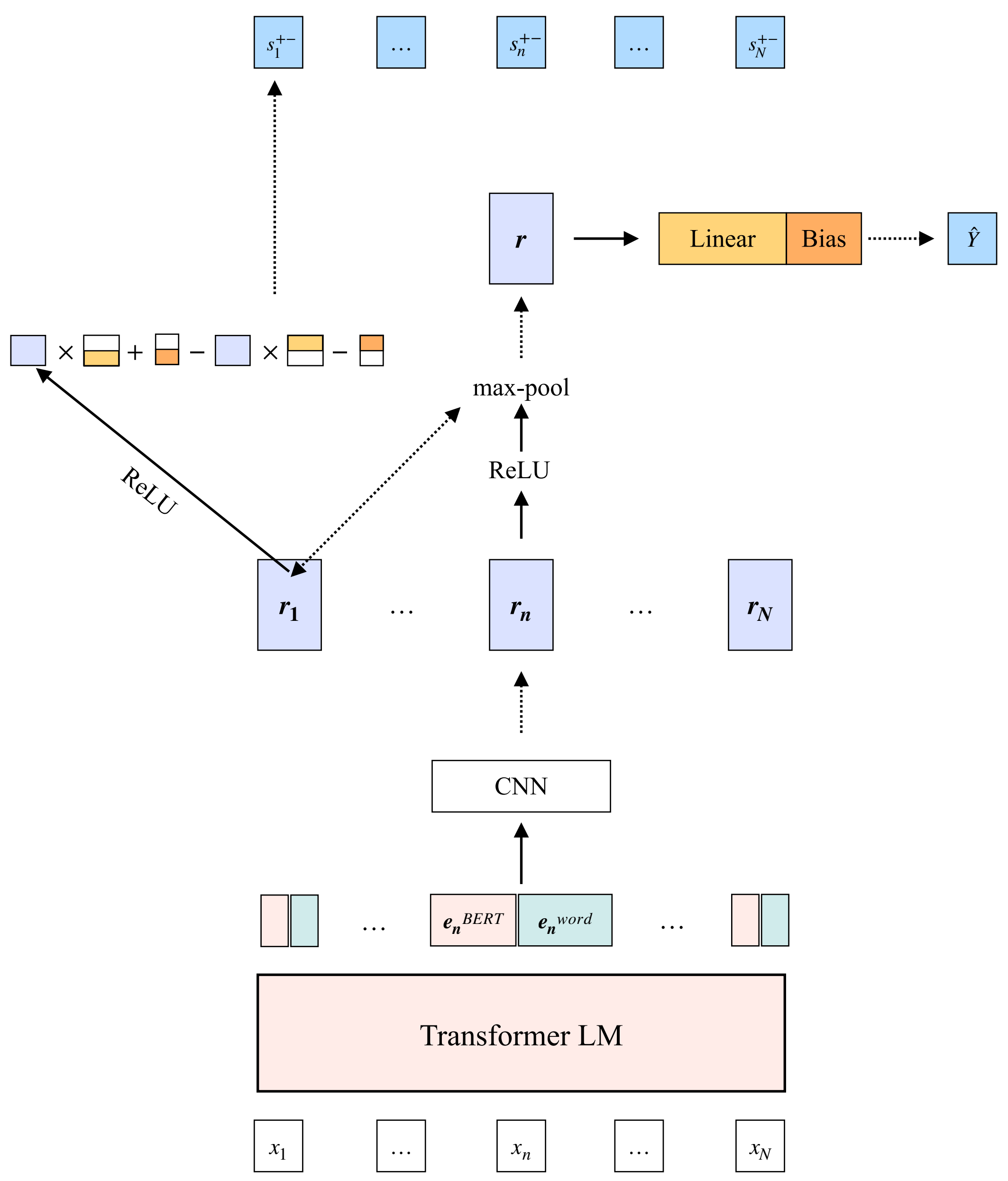
  - Data additions (a.k.a., continual/lifelong learning):
    $\mathbb{D}^N = \left\{(x_1, y_1), \ldots, (x_N, y_N)\right\}$ becomes $\mathbb{D}^{N'} = \left\{(x_1, y_1), \ldots, (x_N, y_N), \ldots, (x_{N'}, y_{N'})\right\}$

  - New lightweight models over representations (e.g., using data additions): $c': \mathbb{R}^M \rightarrow \mathbb{Y}'$

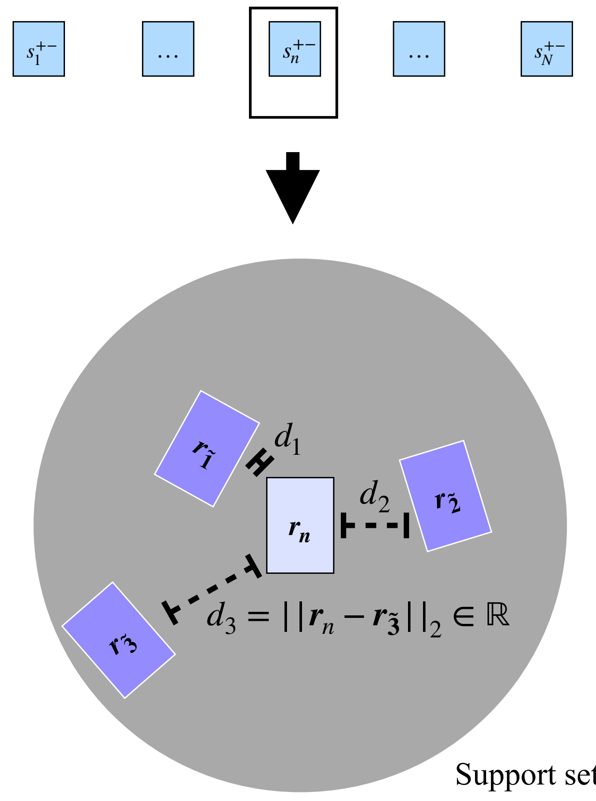
Allen Schmaltz

Horizontal (across the input) & *Vertical* (across the support set) Model Decompositions

Sequence Labeling via a Convolutional Decomposition



K-NN Approximation



s_1^{+-}
…
s_n^{+-}
…
s_N^{+-}
$\boldsymbol{r}_{\tilde{1}}$
d_1
$\boldsymbol{r}_n$
d_2
$\boldsymbol{r}_{\tilde{2}}$
$\boldsymbol{r}_{\tilde{3}}$
$d_3 = ||\boldsymbol{r}_n - \boldsymbol{r}_{\tilde{3}}||_2 \in \mathbb{R}$

Support set:

$$\mathbb{S} = \left\{ \left(\boldsymbol{r}_{\tilde{n}}, \boldsymbol{x}^{(\tilde{n})}, s_{\tilde{n}}^{+-}, Y^{(\tilde{n})} \right) \mid 1 \le \tilde{n} \le \left|\mathbb{S}\right| \right\}$$

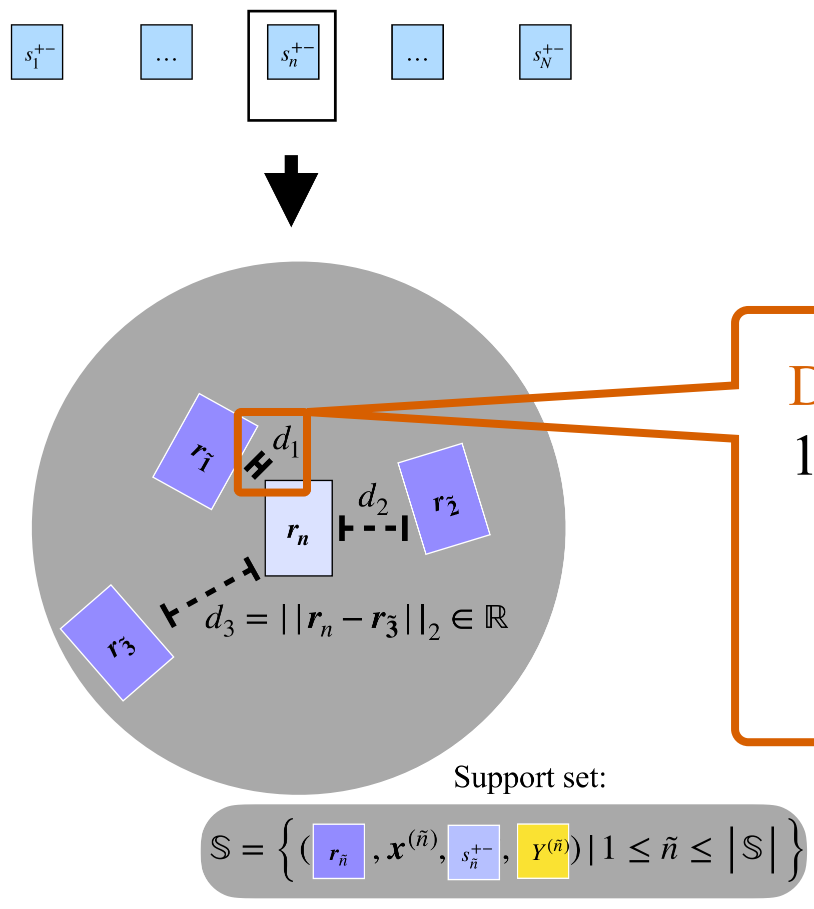
$$\begin{aligned} s_n^{+-} \approx \beta &+ w_1 \cdot \left(\tanh(s_{\tilde{1}}^{+-}) + \gamma \cdot Y^{(\tilde{1})}\right) \\ &+ w_2 \cdot \left(\tanh(s_{\tilde{2}}^{+-}) + \gamma \cdot Y^{(\tilde{2})}\right) \\ &+ w_3 \cdot \left(\tanh(s_{\tilde{3}}^{+-}) + \gamma \cdot Y^{(\tilde{3})}\right) \end{aligned}$$

$$w_k = \frac{\exp(-d_k/\tau)}{\sum_{k'=1}^{3} \exp(-d_{k'}/\tau)}$$

Allen Schmaltz

Leveraging Model Approximations for Prediction Reliability Heuristics & Screening Input Dissimilar to the Support Set

K-NN Approximation



s_1^{+-}
…
s_n^{+-}
…
s_N^{+-}
$\boldsymbol{r}_{\tilde{1}}$
d_1
d_2
$\boldsymbol{r}_{\tilde{2}}$
$\boldsymbol{r}_n$
$\boldsymbol{r}_{\tilde{3}}$
$d_3 = ||\boldsymbol{r}_n - \boldsymbol{r}_{\tilde{3}}||_2 \in \mathbb{R}$
Support set:
$\mathbb{S} = \left\{ (\boldsymbol{r}_{\tilde{n}}, \boldsymbol{x}^{(\tilde{n})}, s_{\tilde{n}}^{+-}, Y^{(\tilde{n})}) \mid 1 \leq \tilde{n} \leq |\mathbb{S}| \right\}$

Data uncertainty: Distance to 1st match (d_1), an exogenous factor, captures uncertainty w.r.t. data (training data compared to test data).

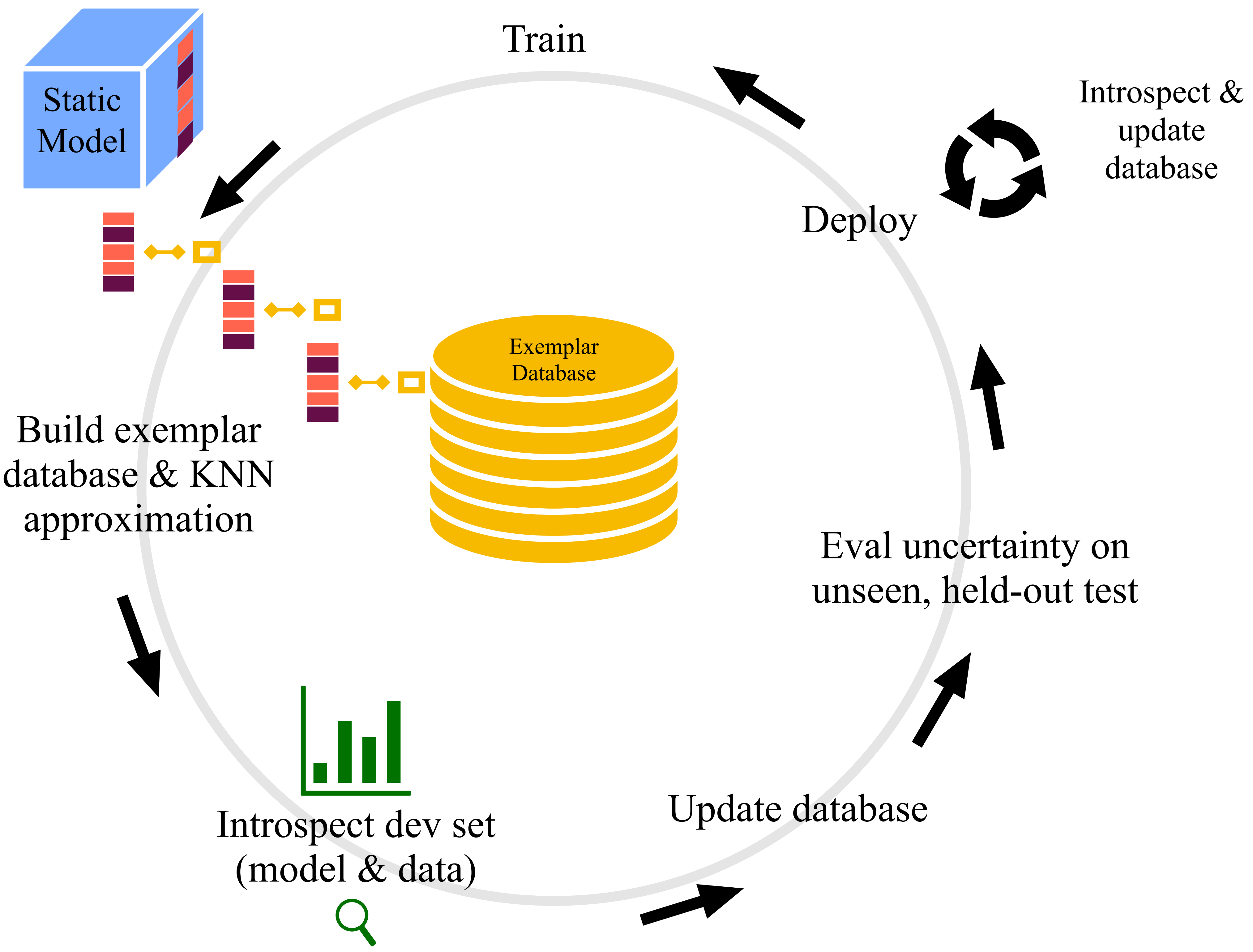
Model uncertainty: This bounded value reaches its min/max when $\tanh(s_k^{+-})$ & $Y^{(k)}$ (or y_k, with token-level labels) agree, for all k (assuming $\gamma > 0$).

$$
\begin{aligned}
s_n^{+-} \approx \boxed{\begin{aligned} \beta &+ w_1 \cdot (\tanh(s_{\tilde{1}}^{+-}) + \gamma \cdot Y^{(\tilde{1})}) \\ &+ w_2 \cdot (\tanh(s_{\tilde{2}}^{+-}) + \gamma \cdot Y^{(\tilde{2})}) \\ &+ w_3 \cdot (\tanh(s_{\tilde{3}}^{+-}) + \gamma \cdot Y^{(\tilde{3})}) \end{aligned}}
\end{aligned}
$$

$$
w_k = \frac{\exp(-d_k/\tau)}{\sum_{k'=1}^{3} \exp(-d_{k'}/\tau)}
$$

Allen Schmaltz

Exemplar Auditing Lifecycle



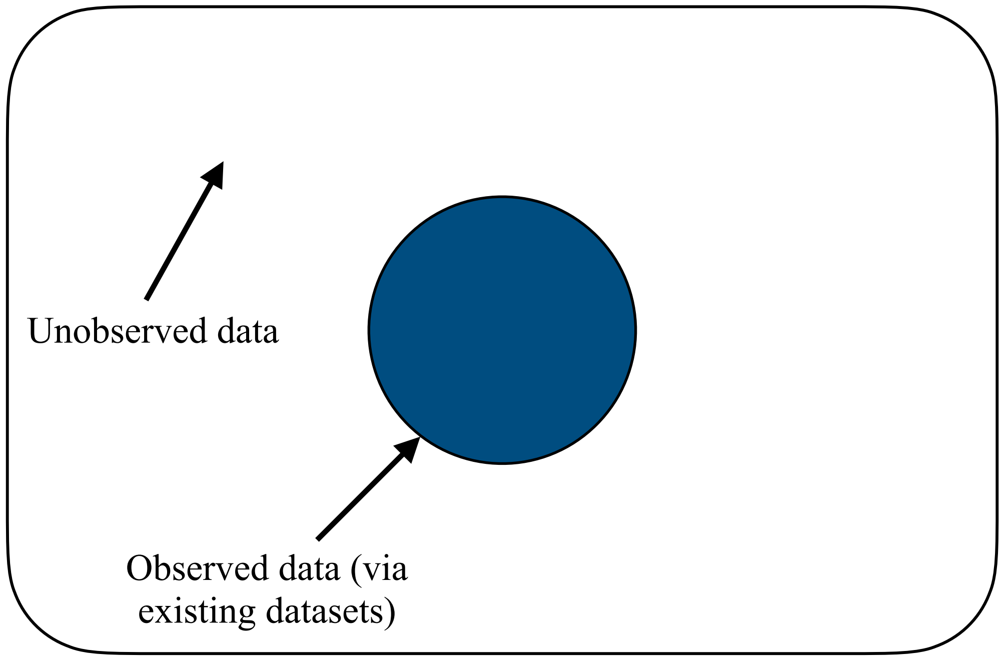
Static Model
Train
Introspect & update database
Deploy
Exemplar Database
Build exemplar database & KNN approximation
Eval uncertainty on unseen, held-out test
Introspect dev set (model & data)
Update database

Allen Schmaltz

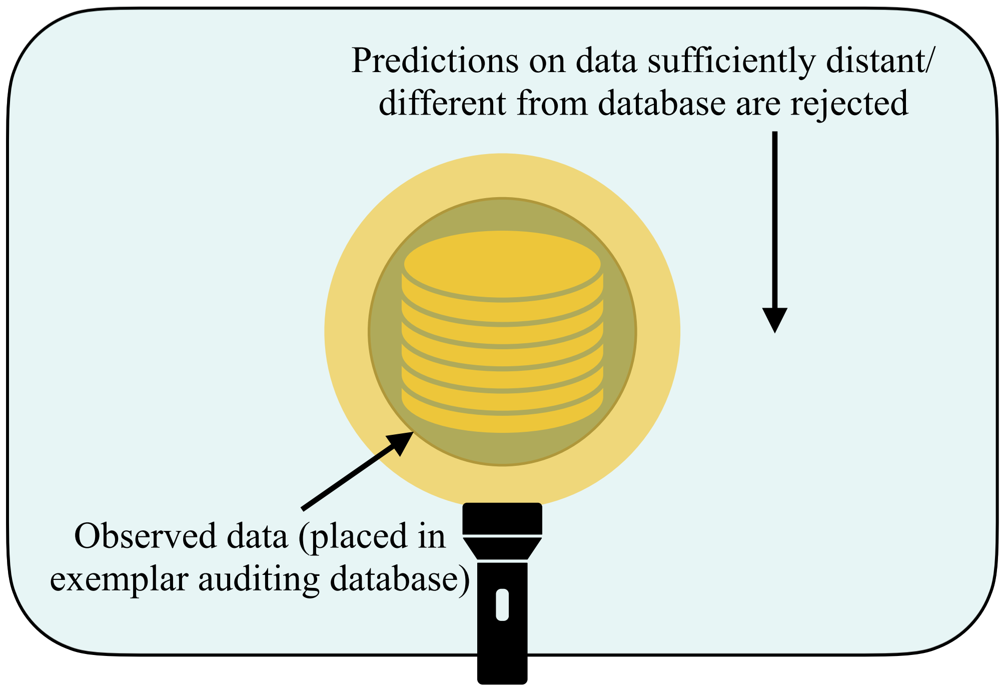
Out-of-Domain Settings

- Pre-train with as much data as possible
- Add as much data as possible to the database, including data not seen in training
    - Corral the in-domain space, around the ball of the observed data
    - Never predict over out-of-domain data in high-risk settings. Instead: Rearrange the deployment to handle non-admitted predictions.

Data distribution for task (partially observed)



Data distribution for task (partially observed)


Predictions on data sufficiently distant/ different from database are rejected
Observed data (placed in exemplar auditing database)

Implementations

- Binary classification: $f : \mathbb{X} \rightarrow \{0,1\}$

Unique side effect: Binary Sequence labeling: $f : \mathbb{X} \rightarrow \{0,1\}_1, \ldots, \{0,1\}_{|x|}$

  - "Detecting Local Insights from Global Labels: Supervised & Zero-Shot Sequence Labeling via a Convolutional Decomposition"

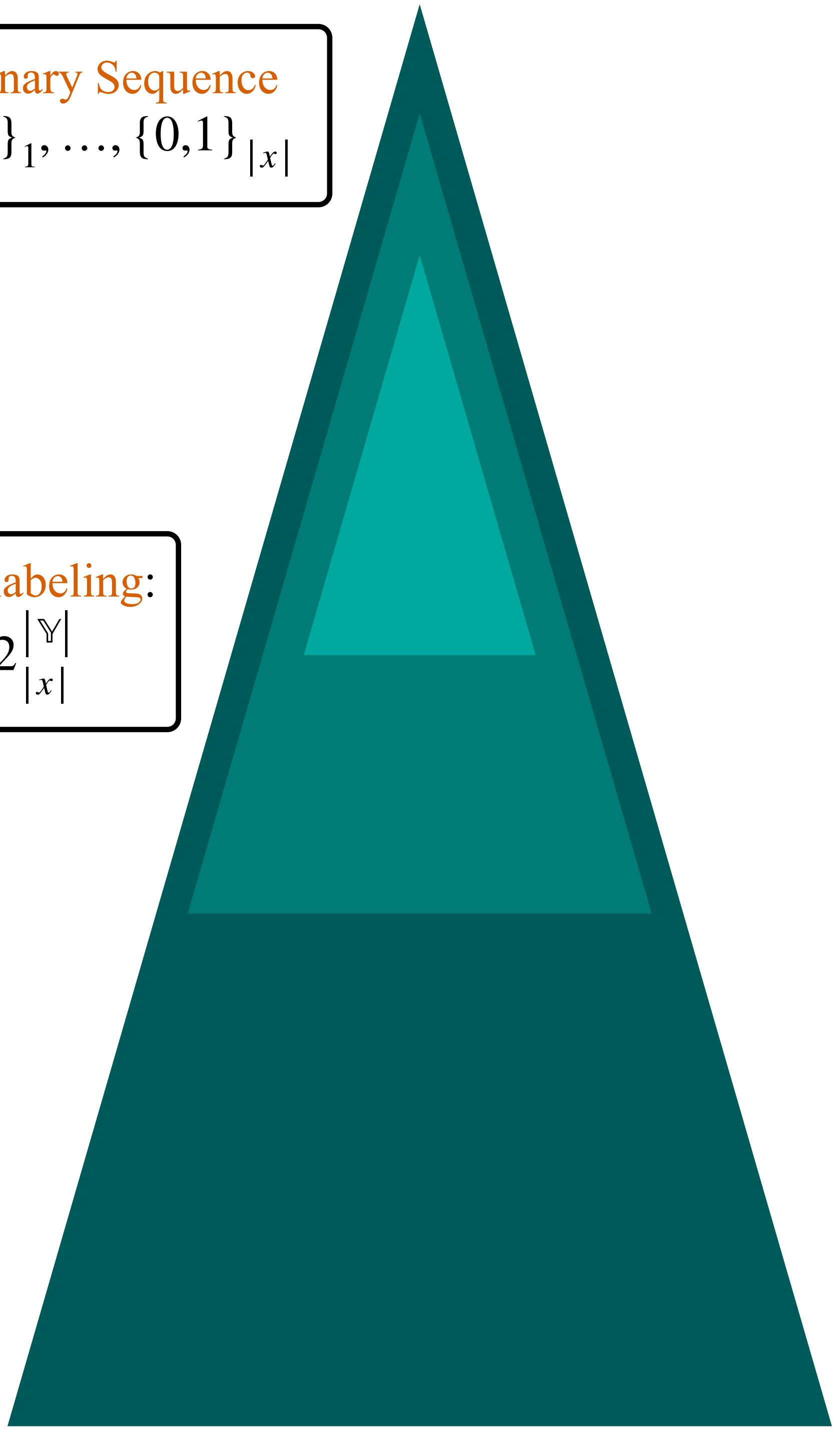
- Multi-label classification: $f : \mathbb{X} \rightarrow 2^{|\mathbb{Y}|}$

Multi-label sequence labeling: $f : \mathbb{X} \rightarrow 2_1^{|\mathbb{Y}|}, \ldots, 2_{|x|}^{|\mathbb{Y}|}$

  - "Exemplar Auditing for Multi-Label Biomedical Text Classification"

- Retrieval-classification: $f : \mathbb{X} \times \mathscr{D} \rightarrow \left\langle \{0,1,2\},\ 2^{|\mathbb{D}|} \right\rangle$

  - "Coarse-to-Fine Memory Matching for Joint Retrieval and Classification"



Allen Schmaltz

Memory Matching Search

- Approach (*high-level*): Run the same shared network, g, over all of Wikipedia, $\mathbb{D}$, caching the representations, & then perform search by matching the query representation with progressively built-up support sequences

q = Query sequence

s = Support sequence

A Wikipedia sentence

Search Level 1 $g(q) = r_q \in \mathbb{R}^M$

$$\begin{tikzcd} g(q) = r_q \in \mathbb{R}^M \arrow[r, leftrightarrow] \arrow[dr, leftrightarrow] & g(s_1) = r_{s_1} \in \mathbb{R}^M \\ & g\left(s_{|\mathbb{D}|}\right) = r_{s_{|\mathbb{D}|}} \in \mathbb{R}^M \end{tikzcd}$$

$\vdots$

$r_{s_1}, \ldots, r_{s_{|\mathbb{D}|}}$ can be cached

$s_i \in \mathbb{D}$

Set of K nearest Wikipedia sentences

$$s'_k \in \arg K \min_{s_i} ||r_q - r_{s_i}||_2$$

Search Level 2

$$\begin{tikzcd} r_q \arrow[r, leftrightarrow] \arrow[dr, leftrightarrow] & g((q, s'_1)) = r_{(q,s'_1)} \in \mathbb{R}^M \\ & g((q, s'_K)) = r_{(q,s'_K)} \in \mathbb{R}^M \end{tikzcd}$$

$\vdots$

Set of Z nearest Wikipedia sentences from Search Level 2

$$s''_z \in \arg Z \min_{s'_k} ||r_q - r_{(q,s'_k)}||_2$$

Search Level 3

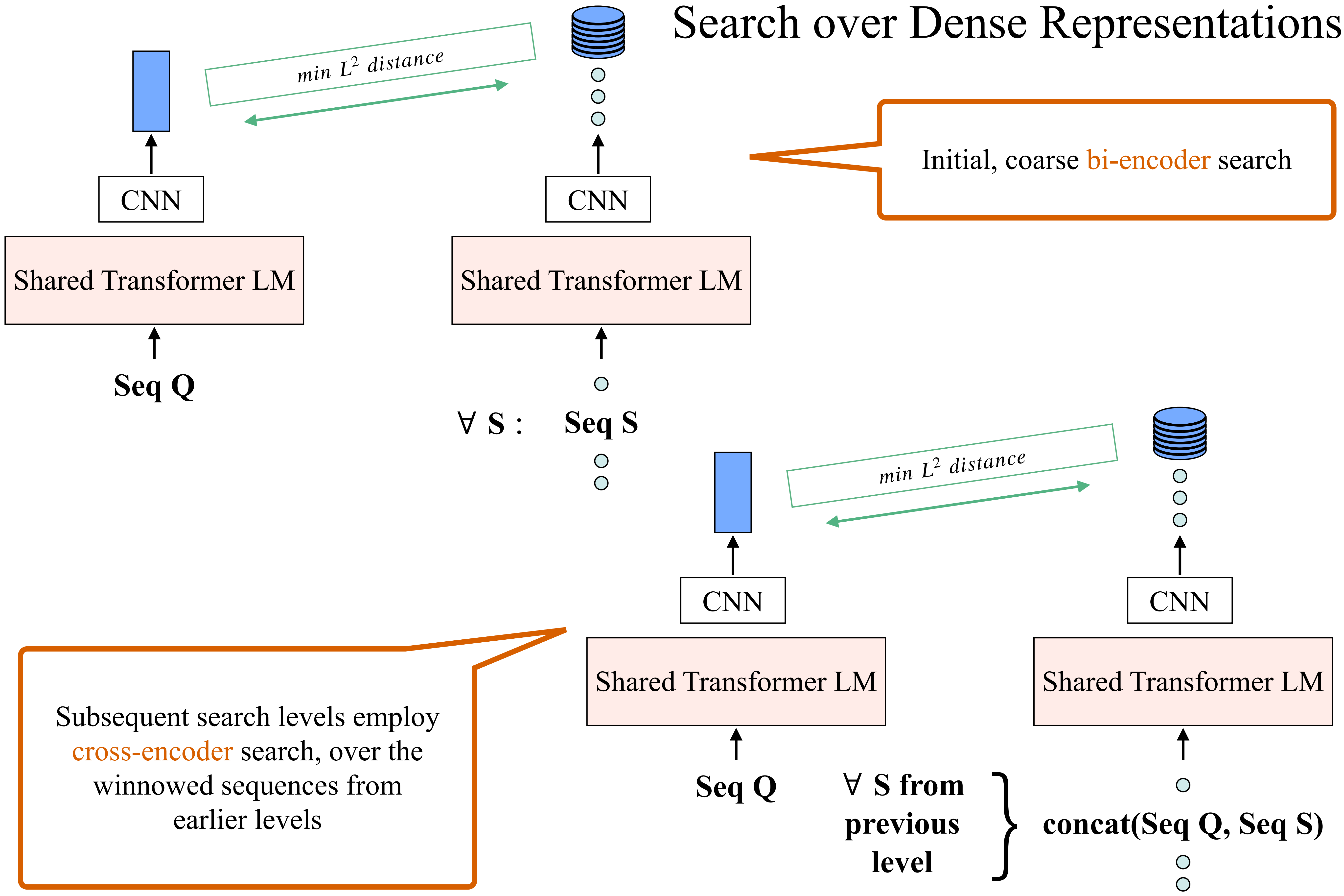
$$\hat{y} = \arg\min_{y \in \{\text{Supports, Refutes, Unverifiable}\}} ||r_q - r_{(y,q,s''_1,\ldots,s''_Z)}||_2$$

$\hat{y}$ is the label prediction

$\{s''_1, \ldots, s''_Z\}$ is the set of Wikipedia support sentences

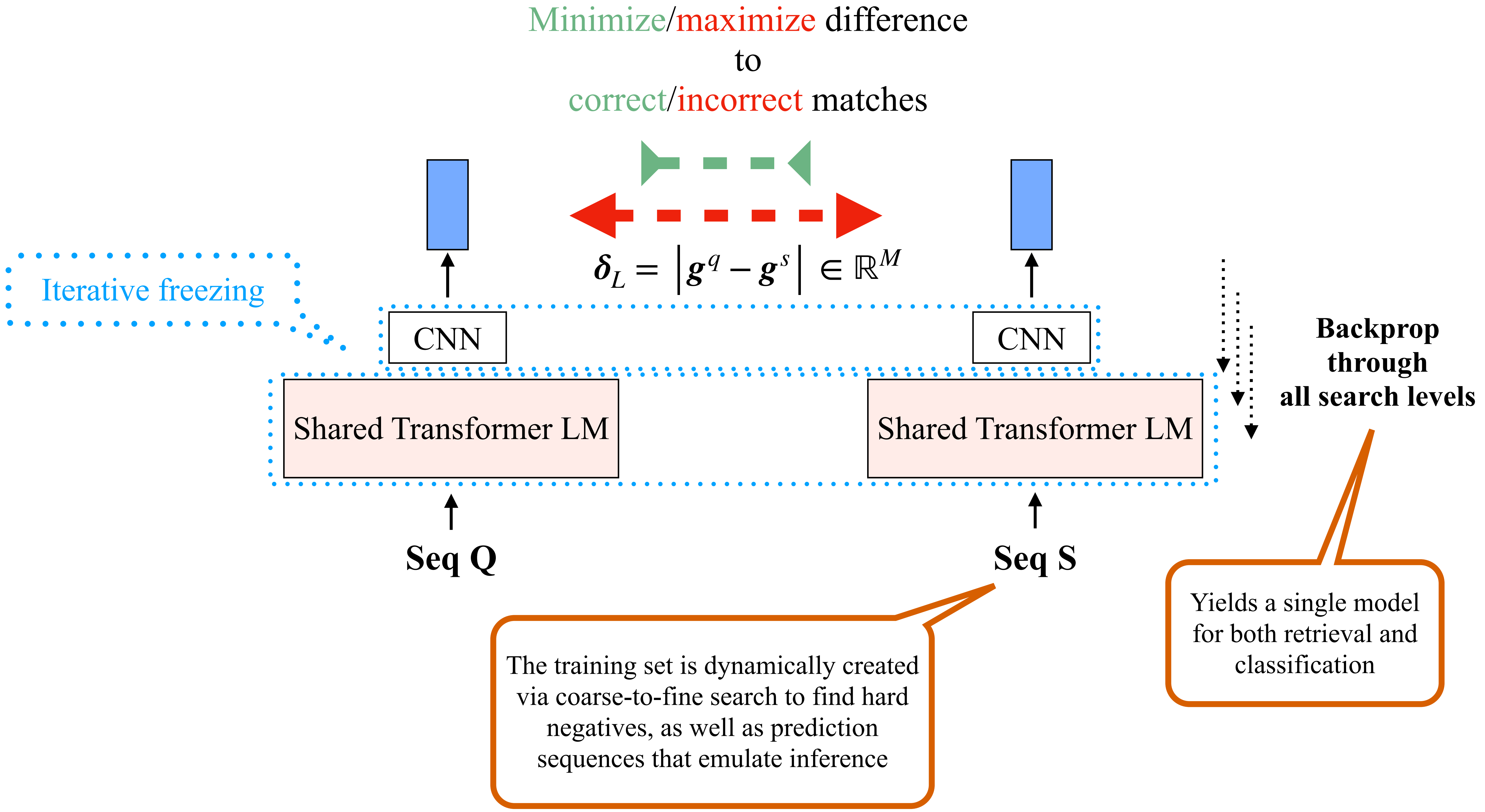
Allen Schmaltz

An End-to-End Retrieval-Classification Model via a Coarse-to-Fine Search over Dense Representations



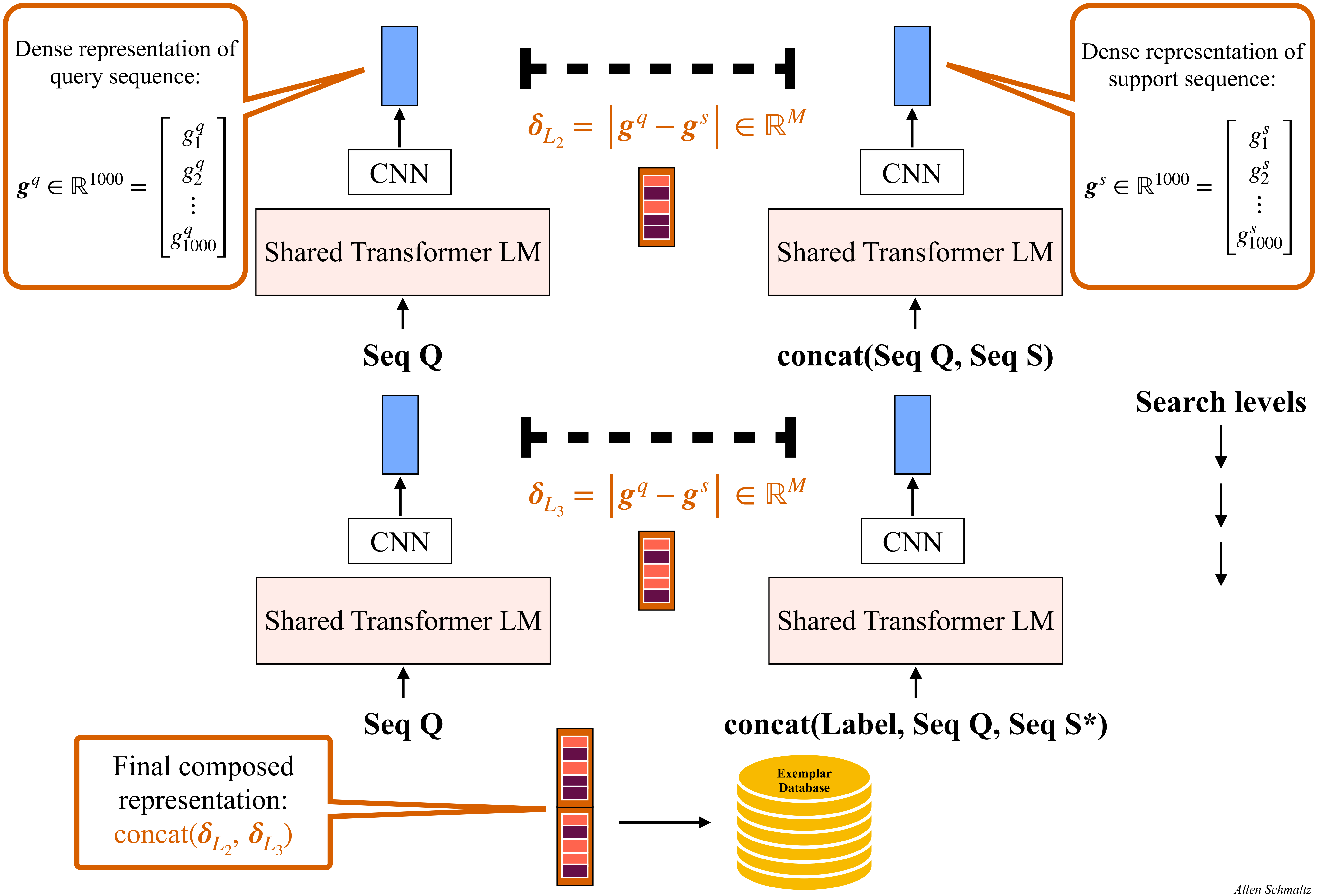
Allen Schmaltz

Joint Retrieval and Classification Training


Minimize/maximize difference
to
correct/incorrect matches
$\delta_L = \left| g^q - g^s \right| \in \mathbb{R}^M$
Iterative freezing
CNN
CNN
Shared Transformer LM
Shared Transformer LM
Backprop through all search levels
Seq Q
Seq S
The training set is dynamically created via coarse-to-fine search to find hard negatives, as well as prediction sequences that emulate inference
Yields a single model for both retrieval and classification

Allen Schmaltz

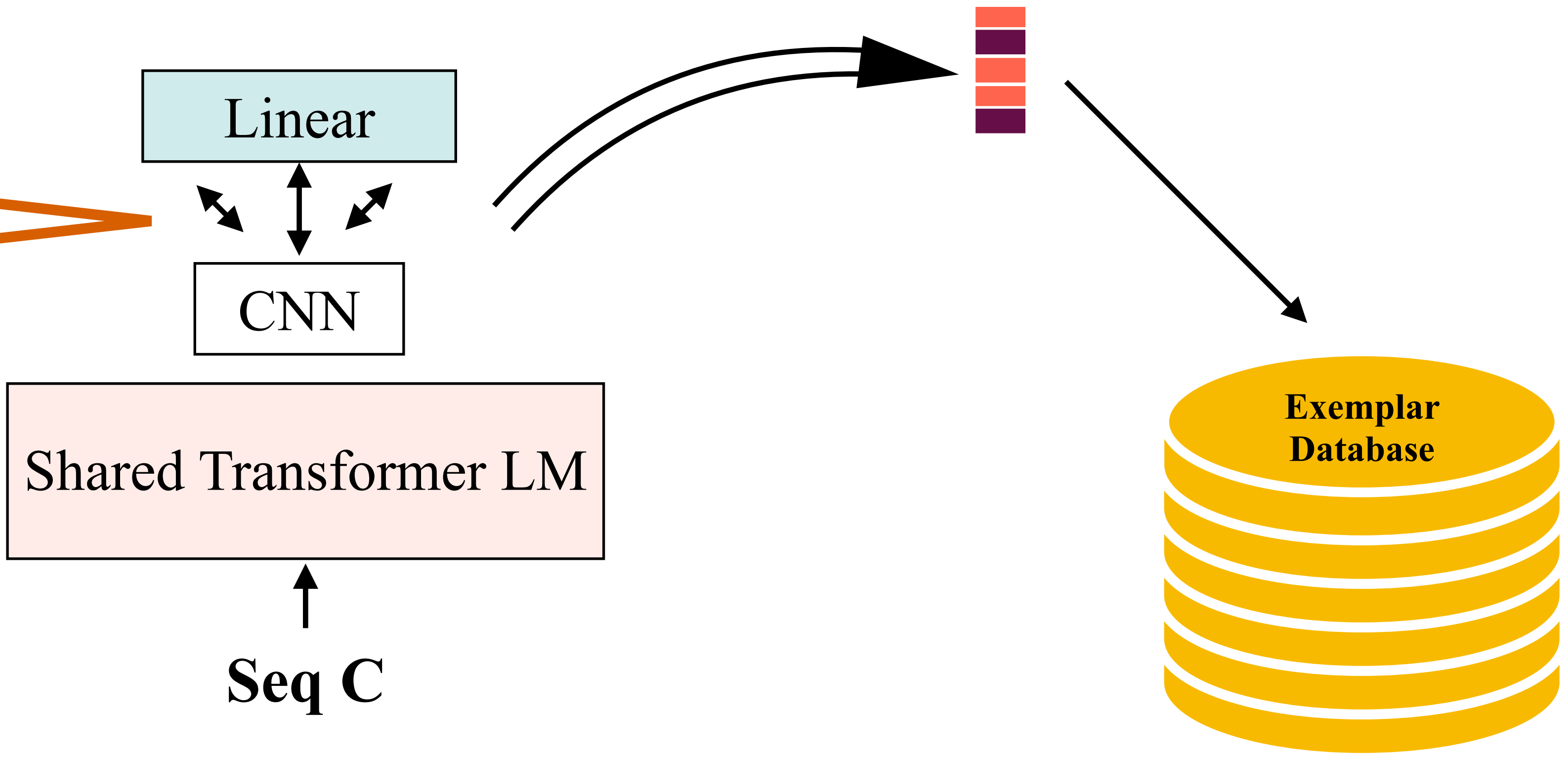
Multi-Sequence Representation Composition for Exemplar Auditing


Dense representation of query sequence:
$\boldsymbol{g}^q \in \mathbb{R}^{1000} = \begin{bmatrix} g_1^q \\ g_2^q \\ \vdots \\ g_{1000}^q \end{bmatrix}$
$\boldsymbol{\delta}_{L_2} = \left| \boldsymbol{g}^q - \boldsymbol{g}^s \right| \in \mathbb{R}^M$
Dense representation of support sequence:
$\boldsymbol{g}^s \in \mathbb{R}^{1000} = \begin{bmatrix} g_1^s \\ g_2^s \\ \vdots \\ g_{1000}^s \end{bmatrix}$
CNN
CNN
Shared Transformer LM
Shared Transformer LM
Seq Q
concat(Seq Q, Seq S)
Search levels
$\boldsymbol{\delta}_{L_3} = \left| \boldsymbol{g}^q - \boldsymbol{g}^s \right| \in \mathbb{R}^M$
CNN
CNN
Shared Transformer LM
Shared Transformer LM
Seq Q
concat(Label, Seq Q, Seq S*)
Final composed representation: concat($\boldsymbol{\delta}_{L_2}$, $\boldsymbol{\delta}_{L_3}$)
Exemplar Database

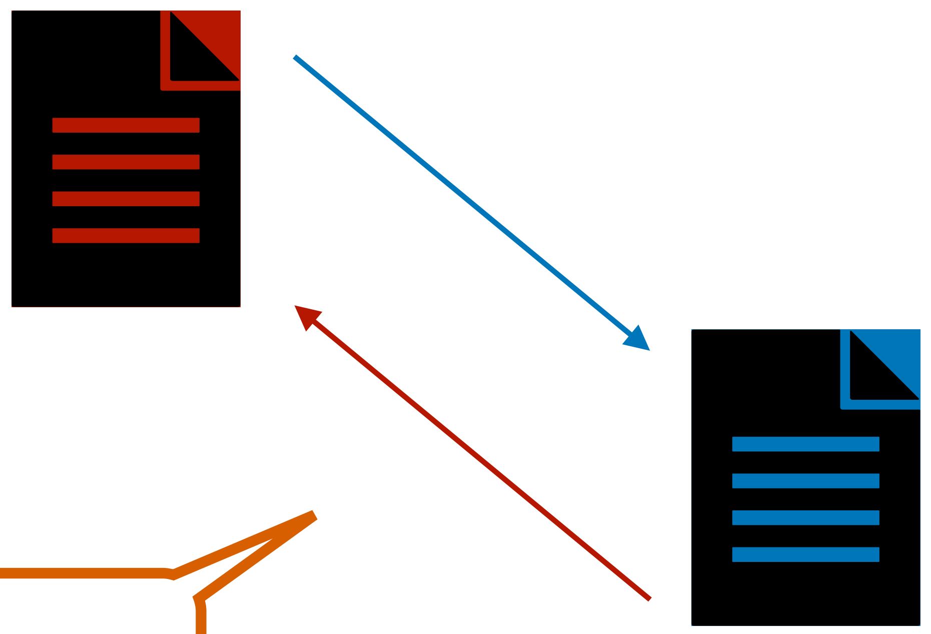
Allen Schmaltz

Token-Level Representations for Exemplar Auditing

Identify the dense representation of a token-level feature using [multi-] binary labeling via a convolutional decomposition (optionally, with priors to encourage/discourage particular features)

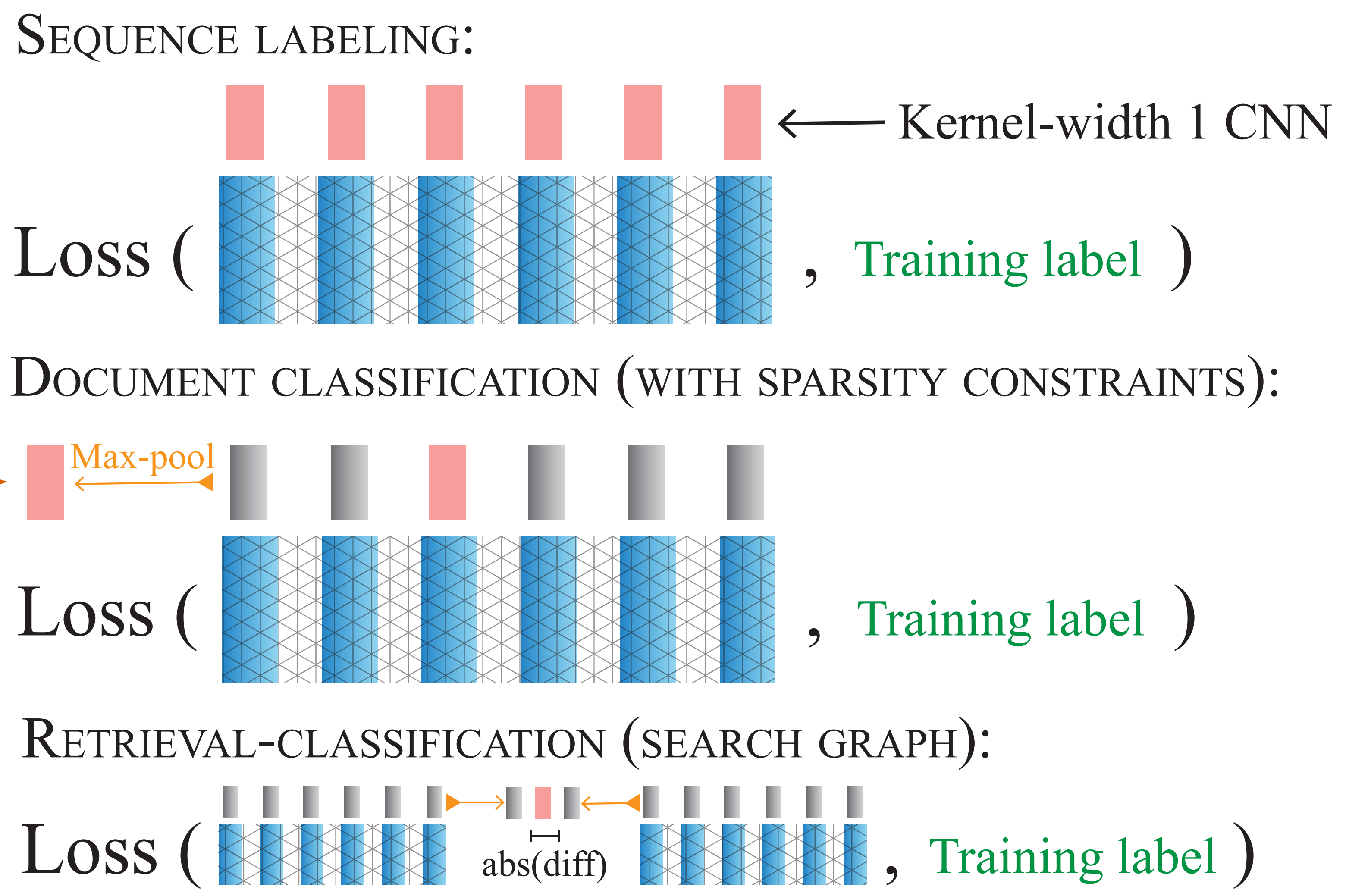


Extractive, Comparative (Feature-wise) Summarization



With facility over features, relating a global prediction to individual sequence elements, we can readily score, examine, & compare salient subsequences across correct & incorrect predictions for each class

In summary, exemplar representations (& model approximations) can be effectively constructed across input modalities/tasks, at a resolution suitable for the task



Semi-supervised feature detection can be useful as part of an analysis pipeline with mechanisms for matching into the support set and uncertainty quantification.

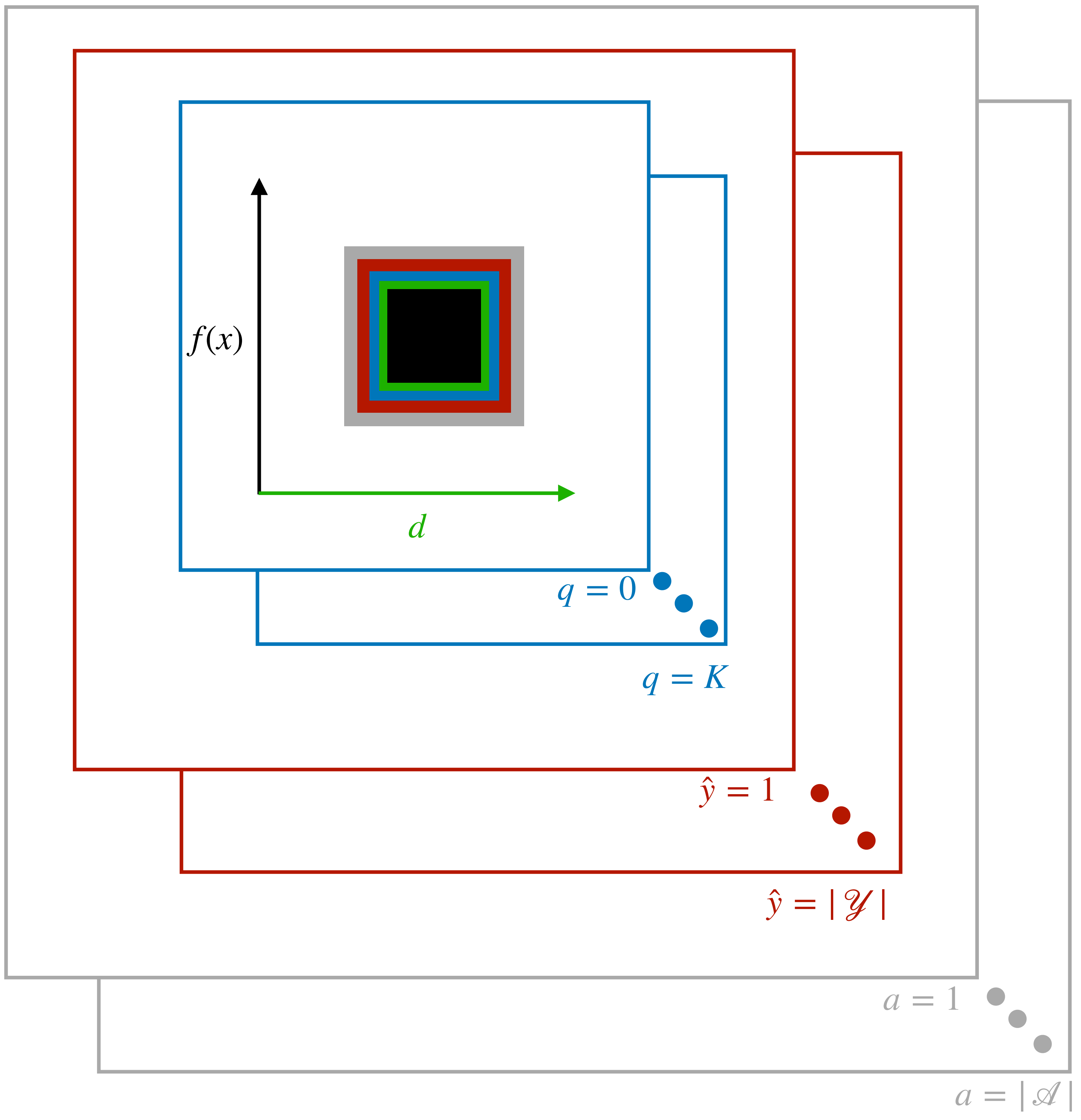
Note: To reiterate, *retrieval* is distinct from the matching of the exemplar representations and KNN approximations. These two mechanisms can be used in conjunction, but serve distinct roles. An end-to-end dense model can be constructed that has a retrieval component for classification (e.g., retrieving relevant Wikipedia documents in a bi- and/or cross-encoded manner); the exemplar representations and KNN approximations are then used for interpretability and uncertainty quantification (as with VENN-ADMIT Predictors) of that underlying retrieval model.

Allen Schmaltz

A template for analyzing high-dimensional data with neural networks

(constraining the black box)

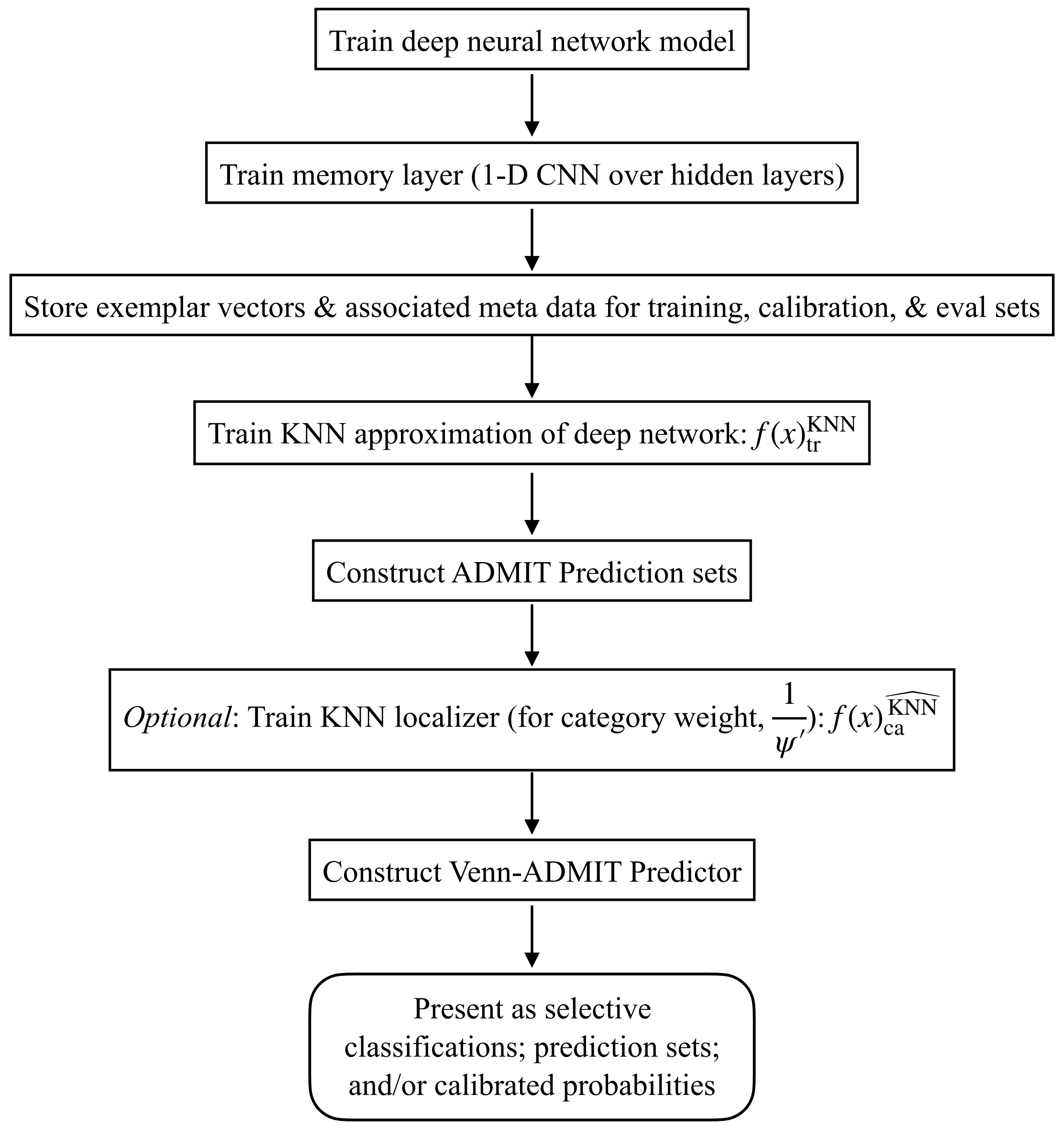
- Model output: $f(x)$
- Distance to nearest training instance: d
- Count of consecutive matches of nearest training instances with the same sign (true label + predictions): $q \in \{0,\ldots,K\}$
- Predicted label: $\hat{y} \in \{1,\ldots,|\mathscr{Y}|\}$
- Known attributes (if available): $a \in \mathscr{A}$



The key signals for analyzing high-dimensional data with neural networks (e.g., large language models). With these constraints, we can then divide the data into partitions over which we can reliably calculate uncertainty, relating new, unseen test points to the points with known labels (e.g., from calibration).

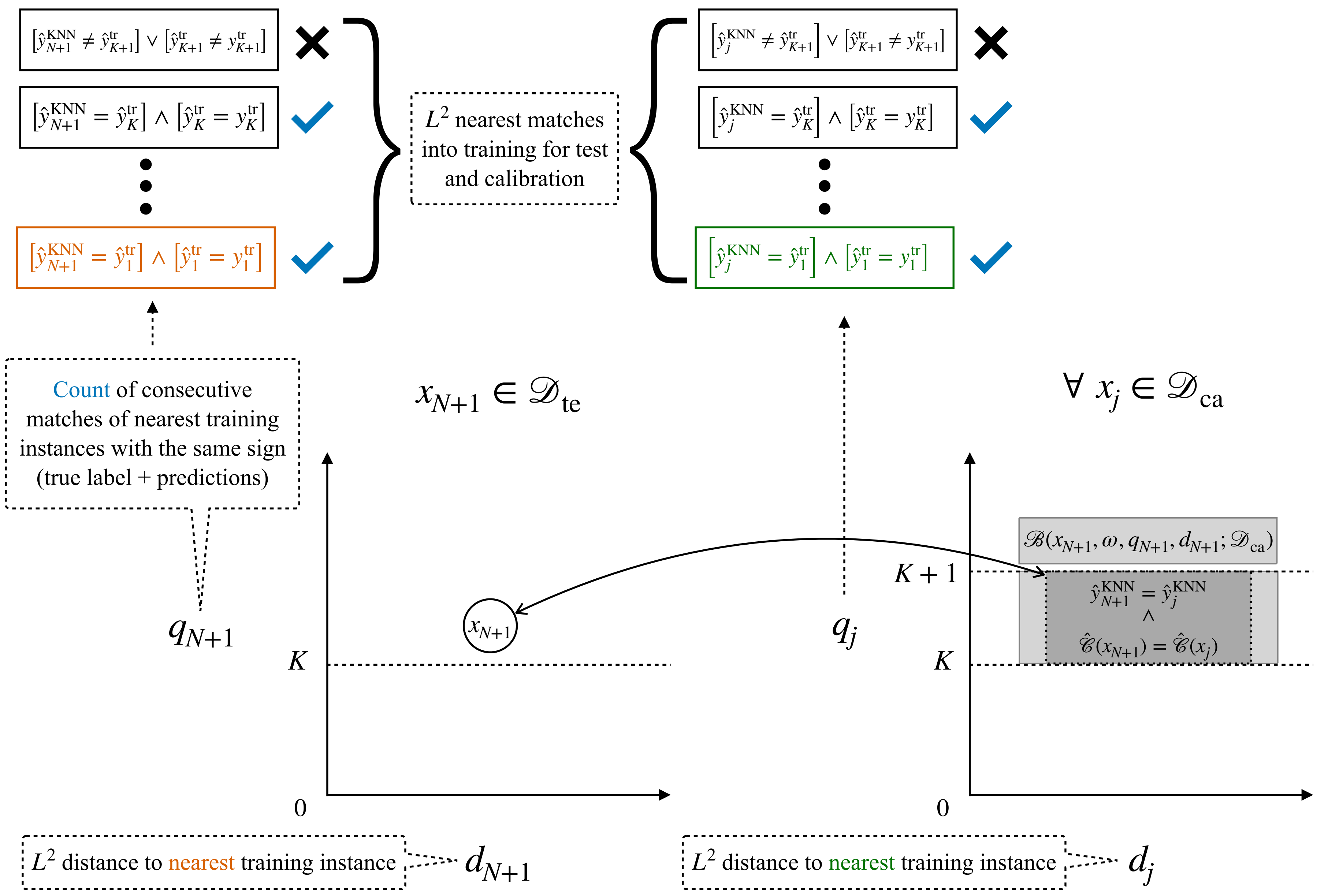
Allen Schmaltz

Uncertainty Quantification: VENN-ADMIT Predictor Overview



Train deep neural network model
Train memory layer (1-D CNN over hidden layers)
Store exemplar vectors & associated meta data for training, calibration, & eval sets
Train KNN approximation of deep network: $f(x)_{\text{tr}}^{\text{KNN}}$
Construct ADMIT Prediction sets
Optional: Train KNN localizer (for category weight, $\frac{1}{\psi'}$): $f(x)_{\text{ca}}^{\widehat{\text{KNN}}}$
Construct Venn-ADMIT Predictor
Present as selective classifications; prediction sets; and/or calibrated probabilities

Uncertainty Quantification: Visualization of a Category Assignment with the VENN-ADMIT Taxonomy

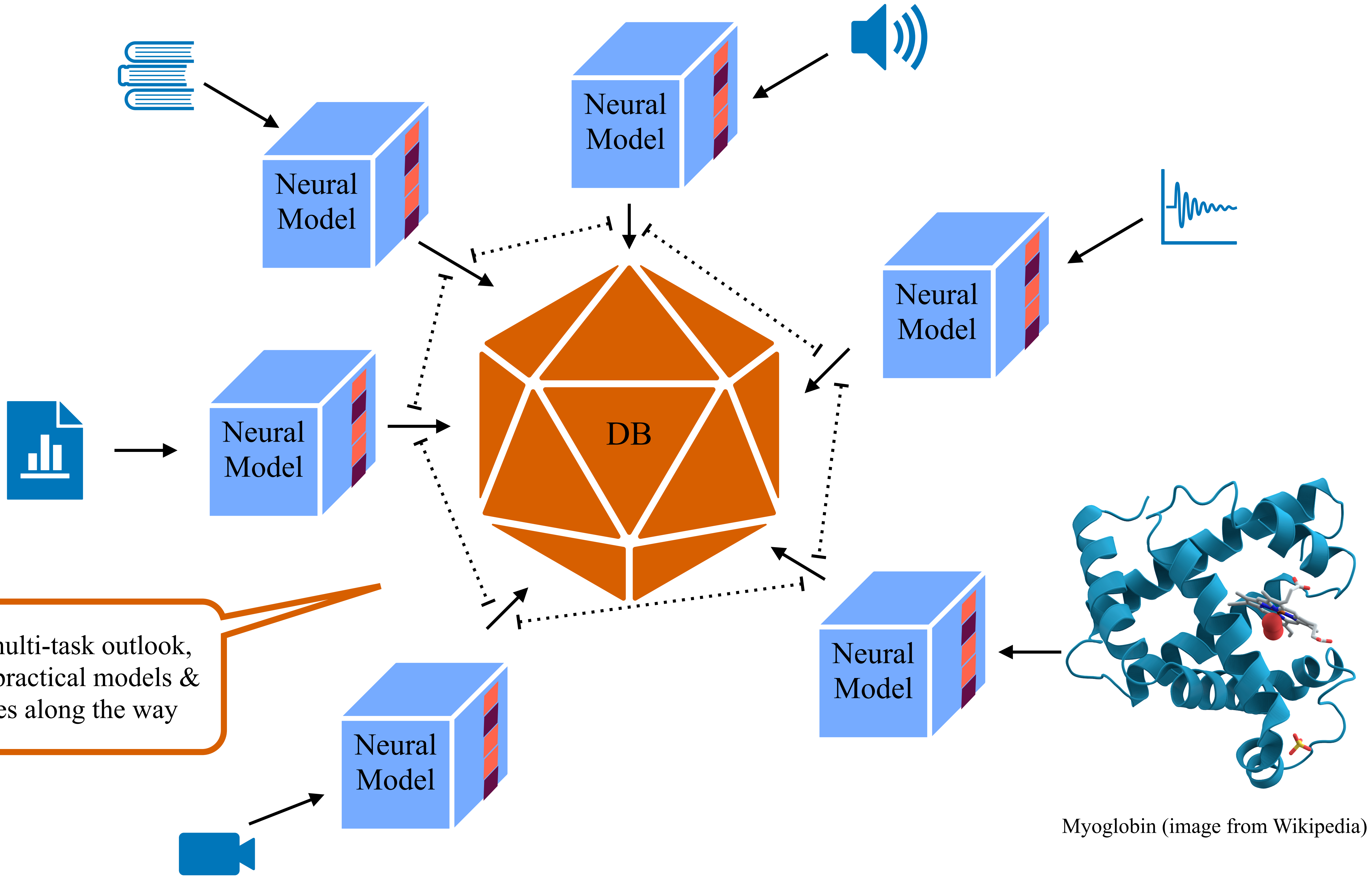

$[\hat{y}^{\mathrm{KNN}}_{N+1} \neq \hat{y}^{\mathrm{tr}}_{K+1}] \vee [\hat{y}^{\mathrm{tr}}_{K+1} \neq y^{\mathrm{tr}}_{K+1}]$ ✗
$[\hat{y}^{\mathrm{KNN}}_{N+1} = \hat{y}^{\mathrm{tr}}_{K}] \wedge [\hat{y}^{\mathrm{tr}}_{K} = y^{\mathrm{tr}}_{K}]$ ✓
$[\hat{y}^{\mathrm{KNN}}_{N+1} = \hat{y}^{\mathrm{tr}}_{1}] \wedge [\hat{y}^{\mathrm{tr}}_{1} = y^{\mathrm{tr}}_{1}]$ ✓
L^2 nearest matches into training for test and calibration
$[\hat{y}^{\mathrm{KNN}}_{j} \neq \hat{y}^{\mathrm{tr}}_{K+1}] \vee [\hat{y}^{\mathrm{tr}}_{K+1} \neq y^{\mathrm{tr}}_{K+1}]$ ✗
$[\hat{y}^{\mathrm{KNN}}_{j} = \hat{y}^{\mathrm{tr}}_{K}] \wedge [\hat{y}^{\mathrm{tr}}_{K} = y^{\mathrm{tr}}_{K}]$ ✓
$[\hat{y}^{\mathrm{KNN}}_{j} = \hat{y}^{\mathrm{tr}}_{1}] \wedge [\hat{y}^{\mathrm{tr}}_{1} = y^{\mathrm{tr}}_{1}]$ ✓
Count of consecutive matches of nearest training instances with the same sign (true label + predictions)
$x_{N+1} \in \mathcal{D}_{\mathrm{te}}$
$\forall\ x_j \in \mathcal{D}_{\mathrm{ca}}$
q_{N+1}
K
0
x_{N+1}
q_j
$\mathcal{B}(x_{N+1}, \omega, q_{N+1}, d_{N+1}; \mathcal{D}_{\mathrm{ca}})$
$K+1$
$\hat{y}^{\mathrm{KNN}}_{N+1} = \hat{y}^{\mathrm{KNN}}_{j}$ $\wedge$ $\hat{\mathscr{C}}(x_{N+1}) = \hat{\mathscr{C}}(x_j)$
K
0
L^2 distance to nearest training instance
d_{N+1}
L^2 distance to nearest training instance
d_j

More reliable predictions and prediction sets: higher q, lower d

Model behavior in the most reliable data partitions is remarkably stable across covariate shifts, providing a degree of uncertainty quantification robustness not typically otherwise observed with neural networks.

Allen Schmaltz

Prospective Outlook: Interlocking distance constraints across input modalities and tasks via a single, shared model and a dense database…



Productive multi-task outlook, since we get practical models & data analyses along the way

Myoglobin (image from Wikipedia)

Allen Schmaltz