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:

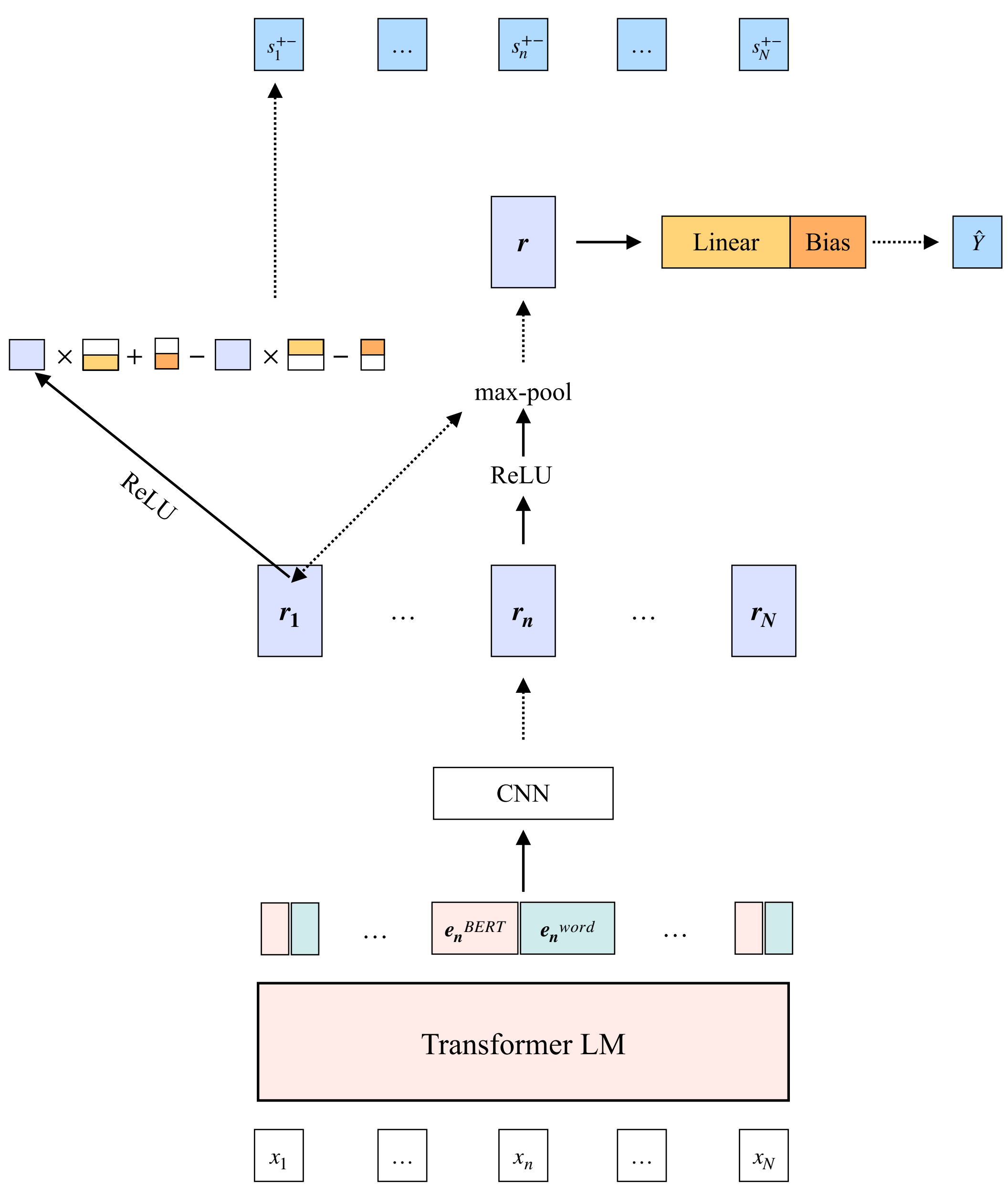
    - Label changes: $y_n' = y_n + \Delta_n$

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

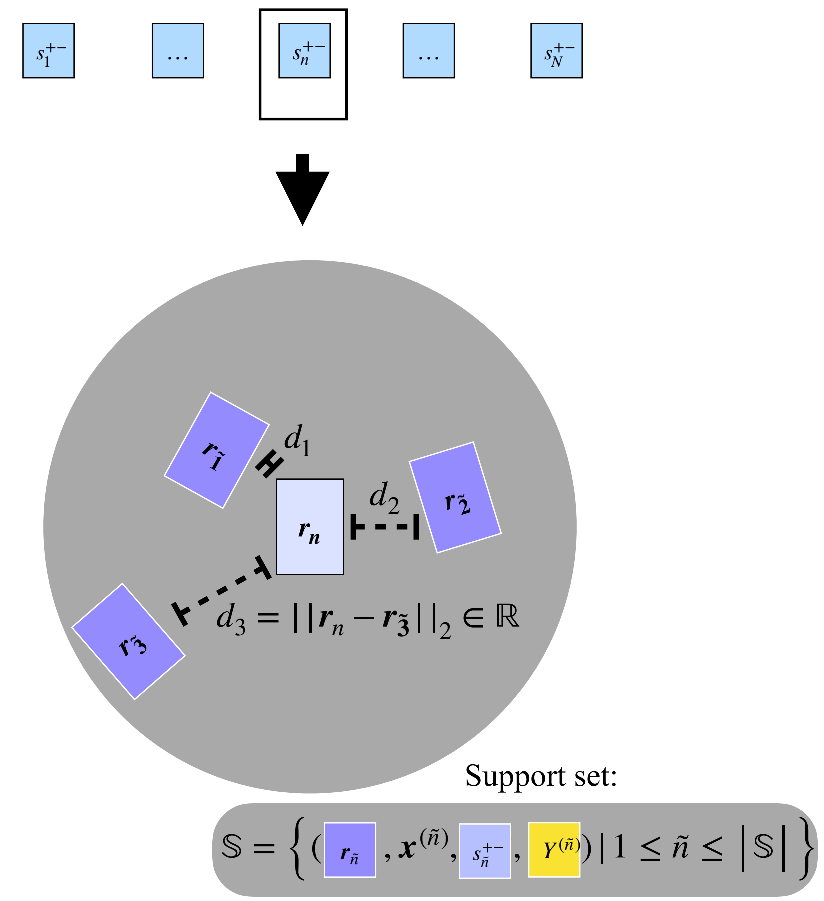
    - New lightweight models over representations (e.g., using data additions): $c' : \mathbb{R}^M \rightarrow \mathbb{Y}'$

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

Sequence Labeling via a Convolutional Decomposition



K-NN Approximation



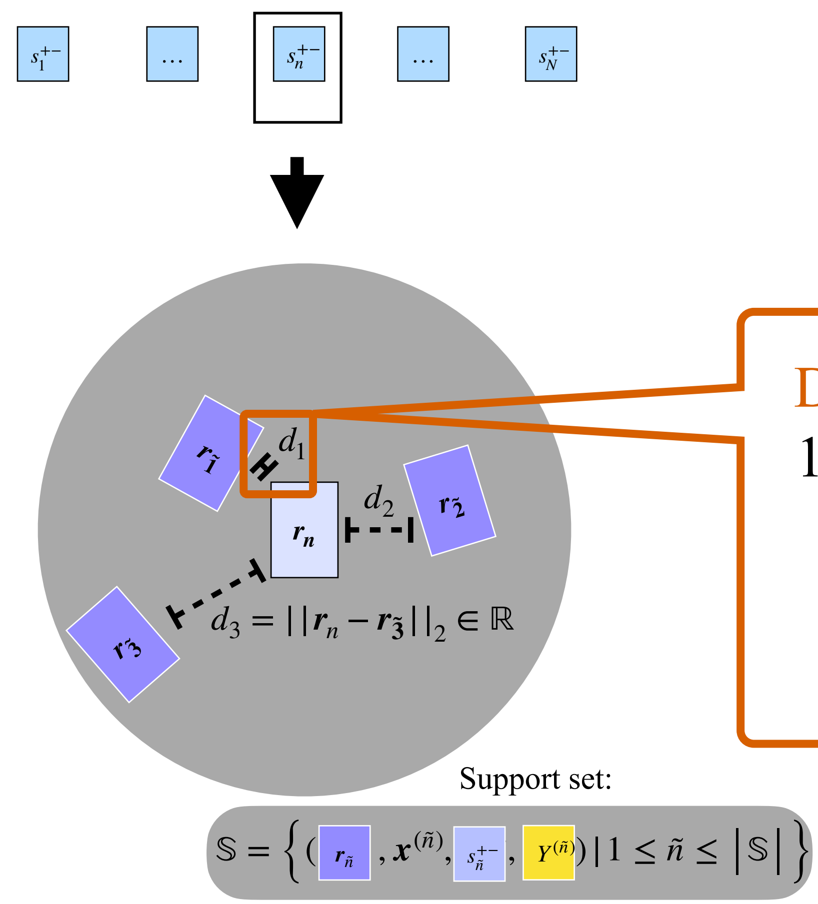
$$\boxed{s_n^{+-}} \approx \beta + w_1 \cdot \left(\tanh(\boxed{s_{\tilde 1}^{+-}}) + \gamma \cdot \boxed{Y^{(\tilde 1)}}\right) + w_2 \cdot \left(\tanh(\boxed{s_{\tilde 2}^{+-}}) + \gamma \cdot \boxed{Y^{(\tilde 2)}}\right) + w_3 \cdot \left(\tanh(\boxed{s_{\tilde 3}^{+-}}) + \gamma \cdot \boxed{Y^{(\tilde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

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

Allen Schmaltz

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

K-NN Approximation



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

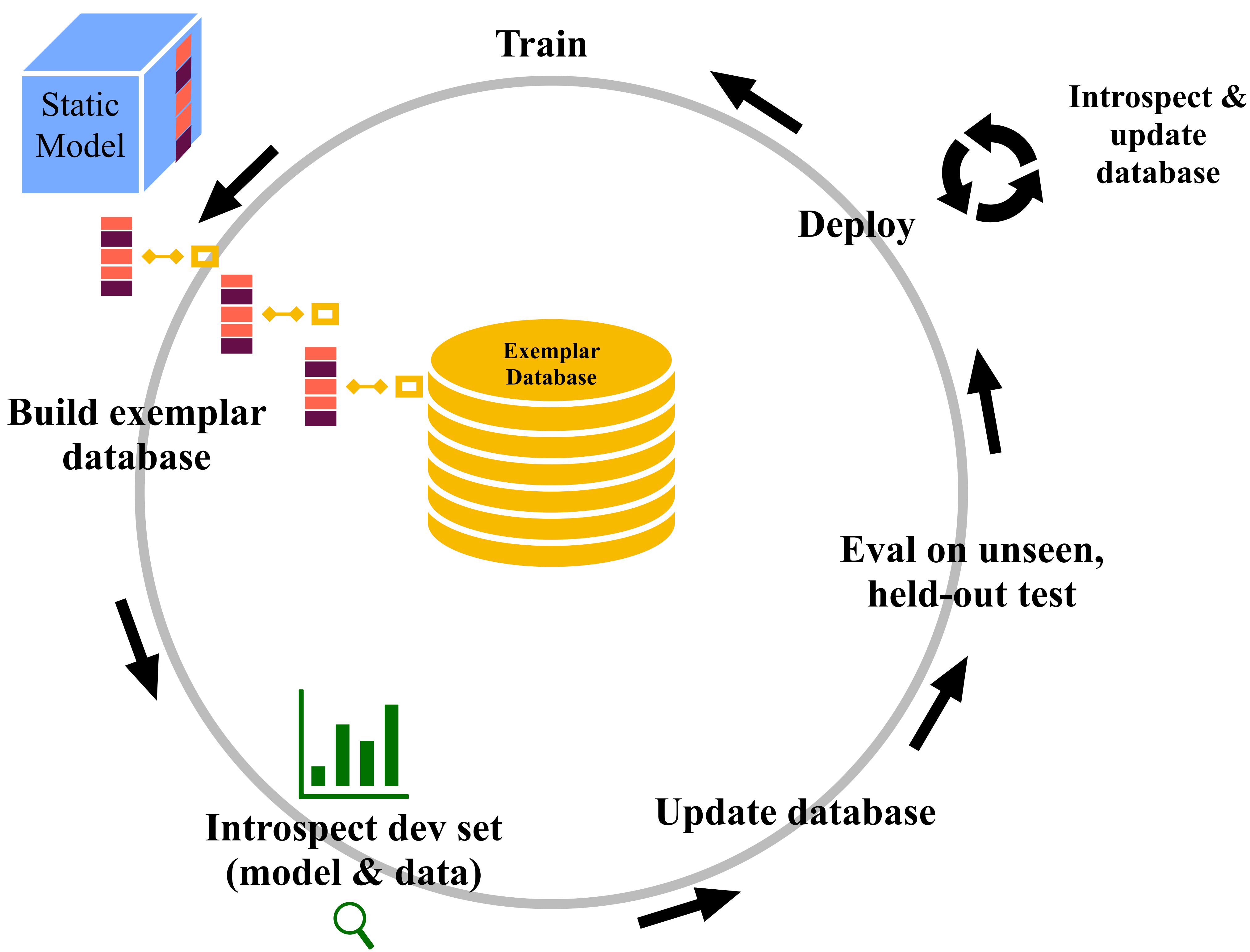
Model uncertainty: This bounded value reaches its min/max when $\tanh(s_k^{+-})$ & $Y^{(k)}$ (or y_k, with token-level labels) agree, for all k (assuming $\gamma > 0$).

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

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

Allen Schmaltz

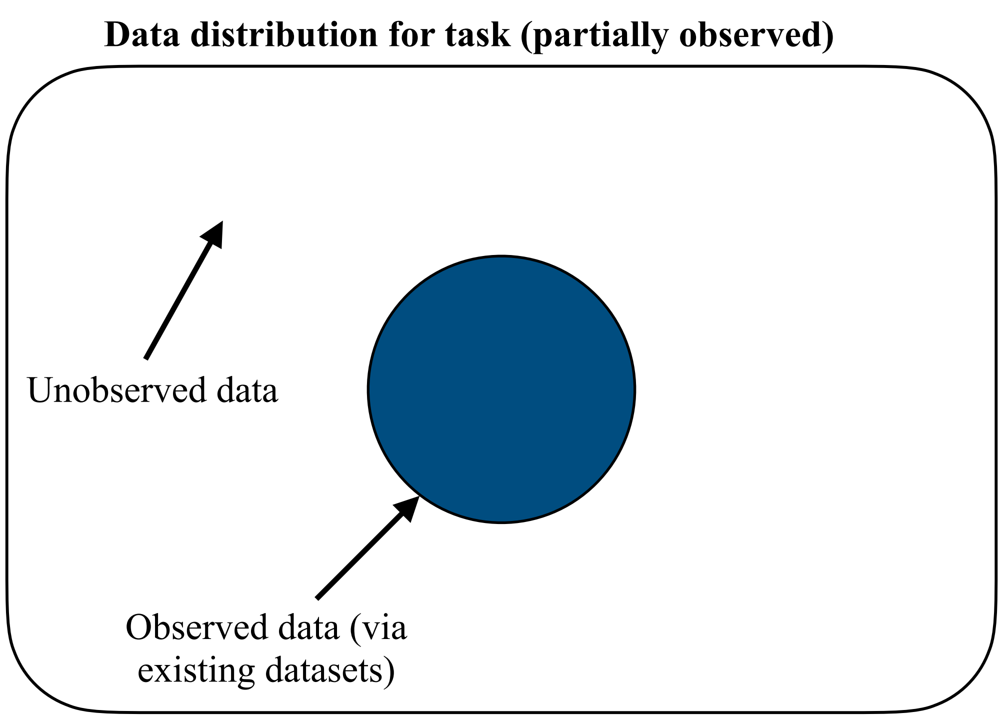
Exemplar Auditing Lifecycle

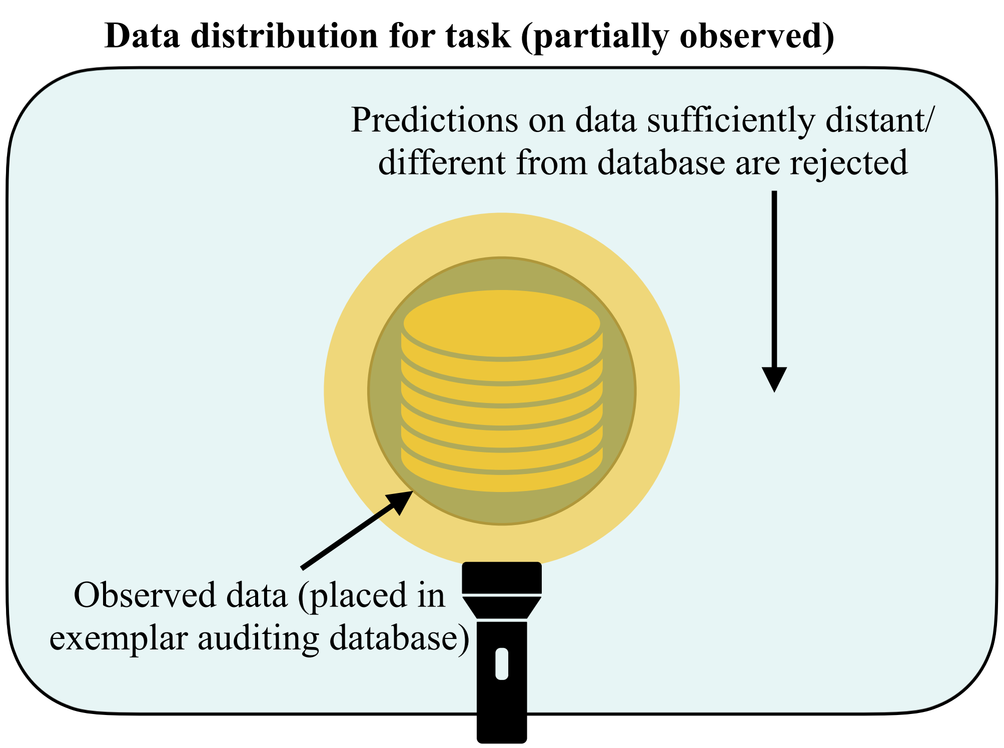


Allen Schmaltz

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.





Implementations

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

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

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

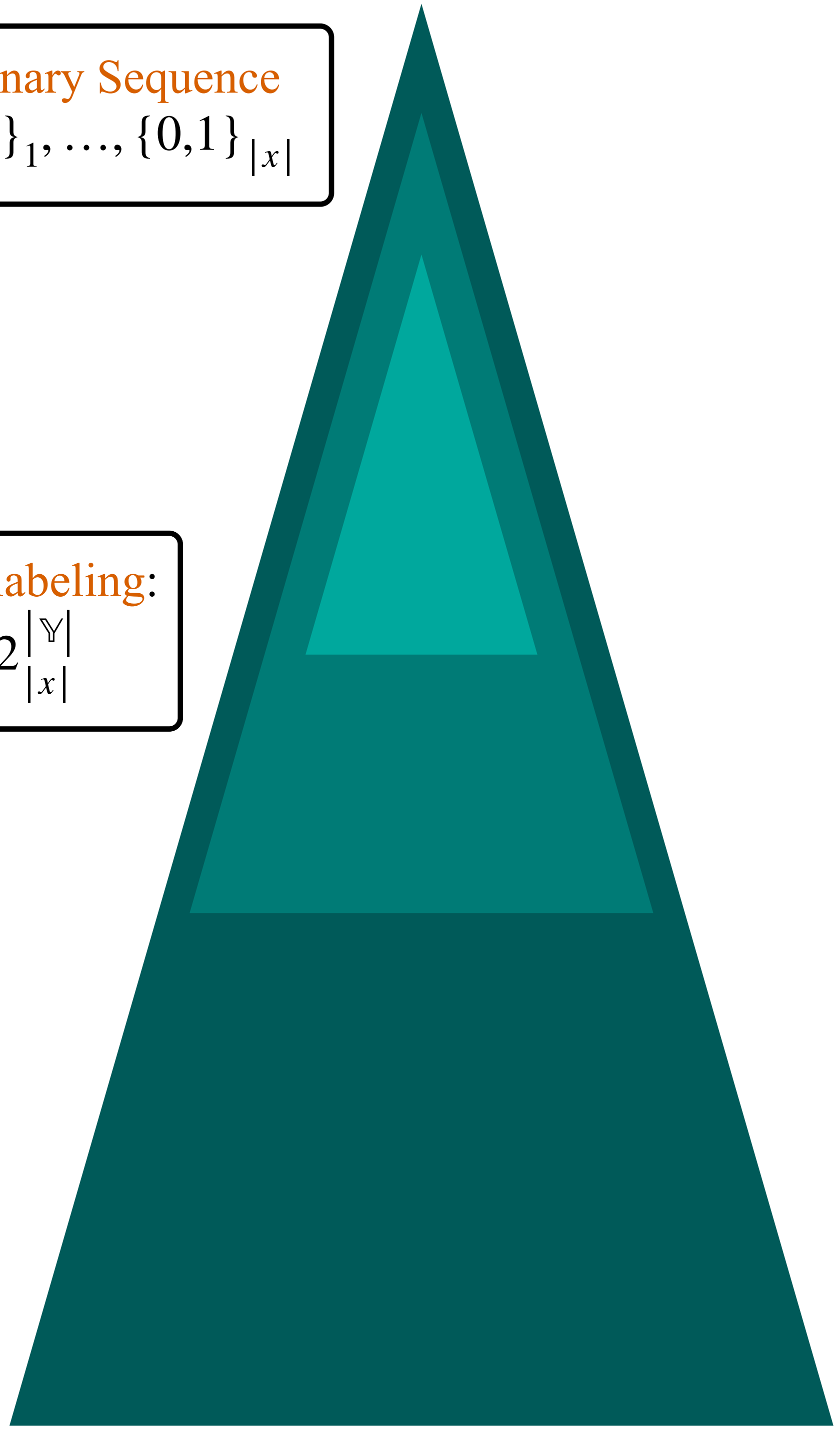
- Multi-label classification: $f : \mathbb{X} \to 2^{|\mathbb{Y}|}$

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

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

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

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



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 $g\left(q\right) = r_q \in \mathbb{R}^M \longleftrightarrow g\left(s_1\right) = r_{s_1} \in \mathbb{R}^M$

$$\vdots$$

$$g\left(s_{|\mathbb{D}|}\right) = r_{s_{|\mathbb{D}|}} \in \mathbb{R}^M$$

$r_{s_1}, \ldots, r_{s_{|\mathbb{D}|}}$ 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 $r_q \longleftrightarrow g\left((q, s'_1)\right) = r_{(q,s'_1)} \in \mathbb{R}^M$

$$\vdots$$

$$g\left((q, s'_K)\right) = r_{(q,s'_K)} \in \mathbb{R}^M$$

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

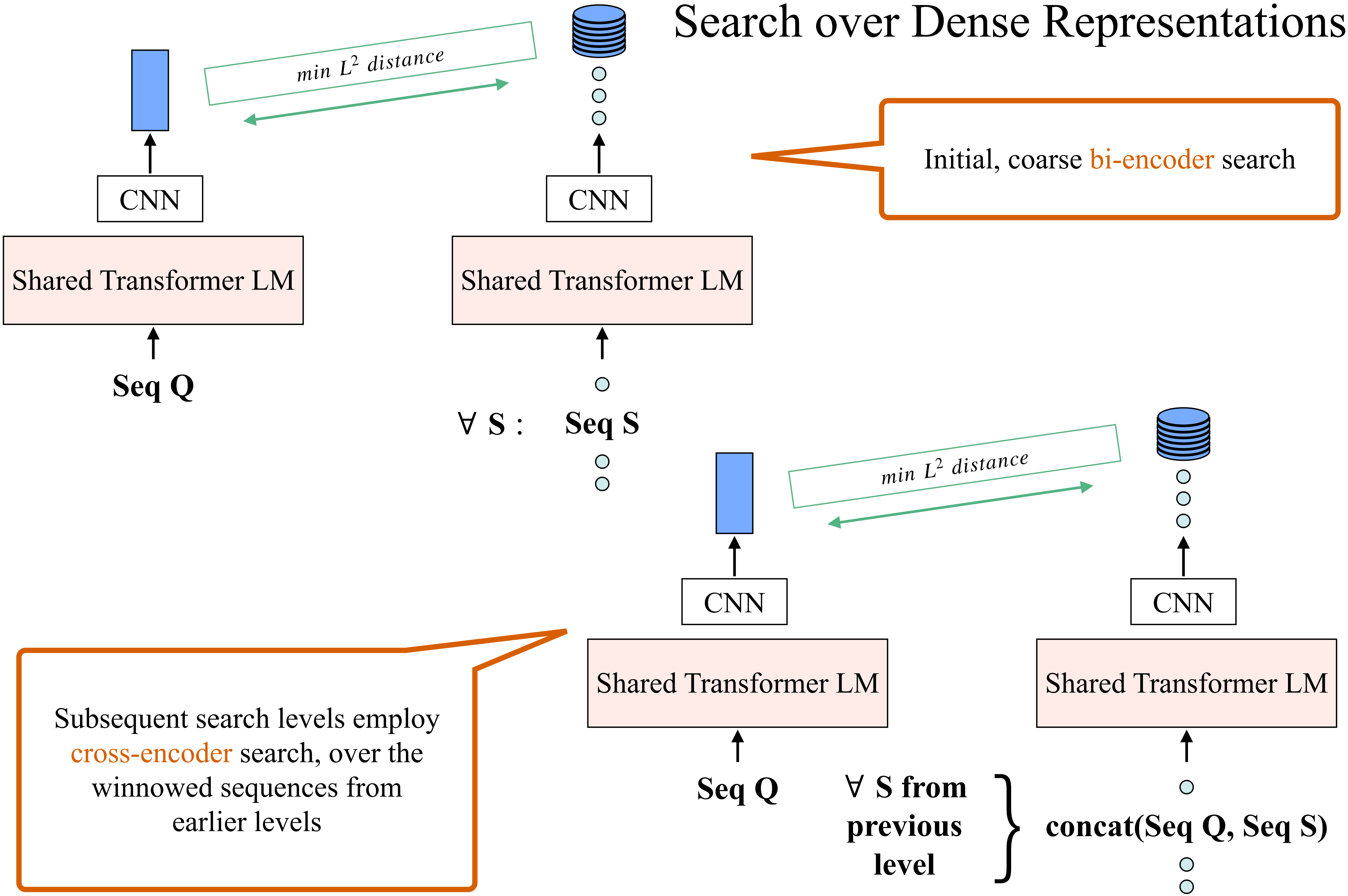
$$\hat{y} = \arg\min_{y \in \{\text{Supports, Refutes, Unverifiable}\}} || r_q - r_{(y,q,s''_1,\ldots,s''_Z)} ||_2$$

$\hat{y}$ is the label prediction

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

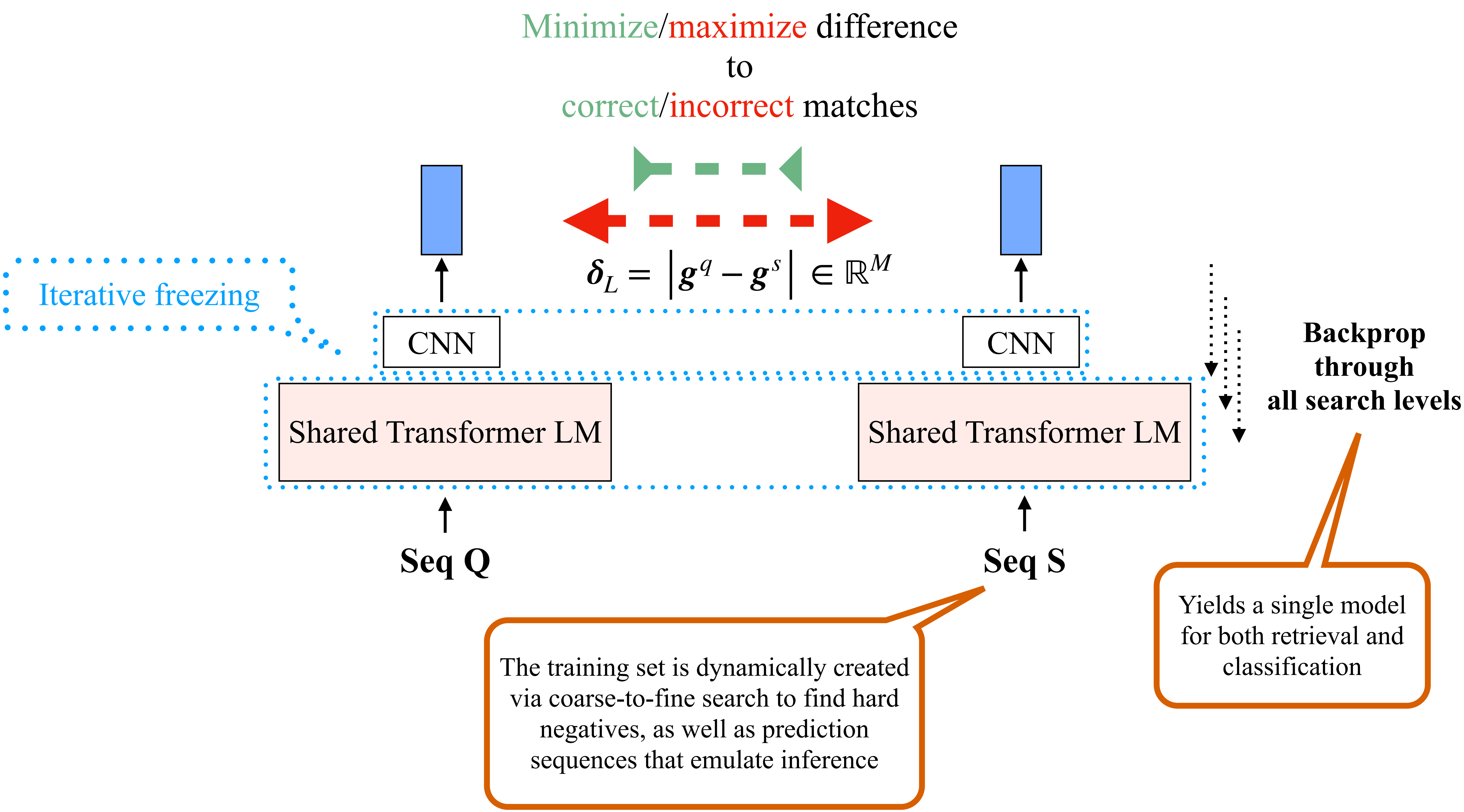
Allen Schmaltz

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

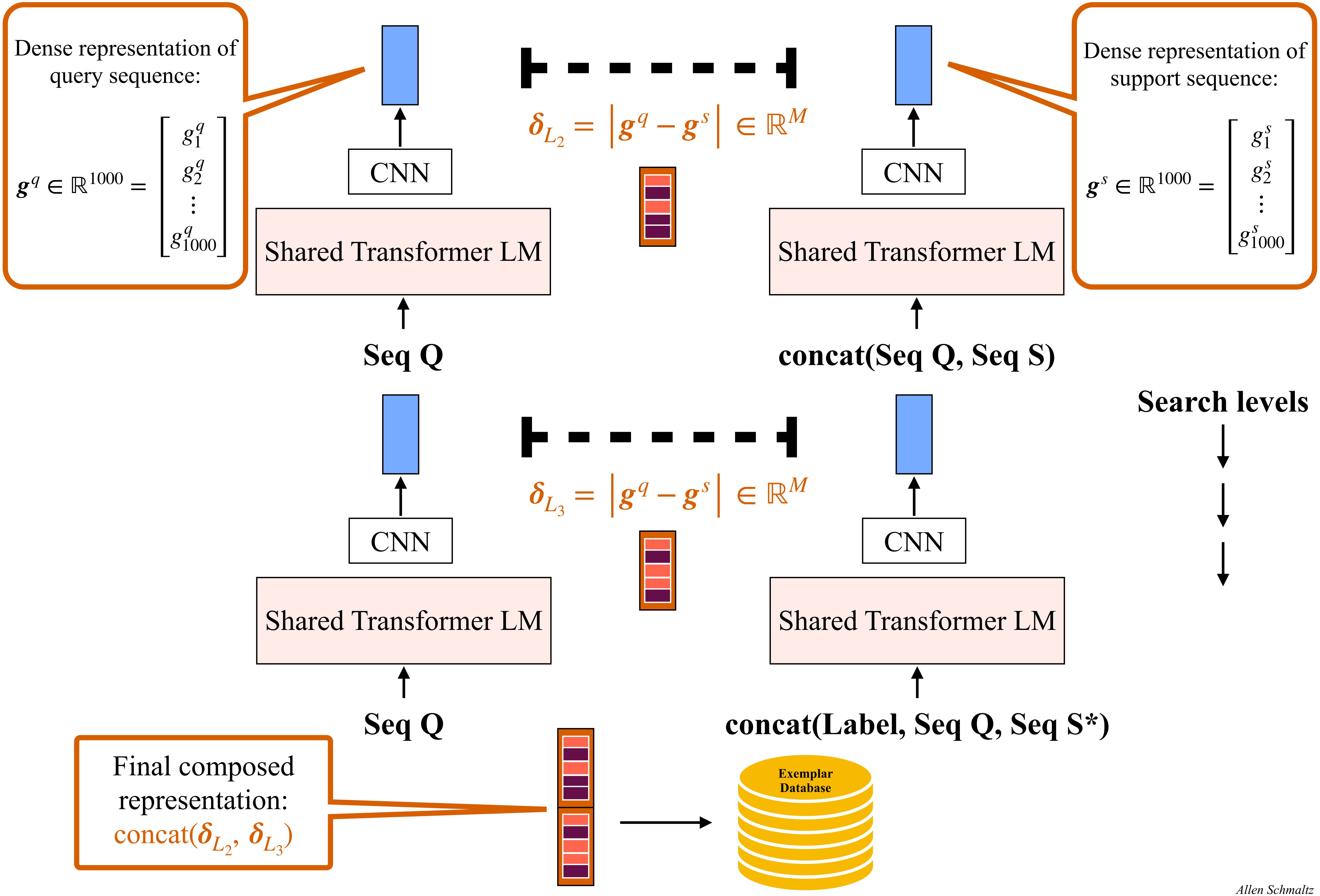


Allen Schmaltz

Joint Retrieval and Classification Training

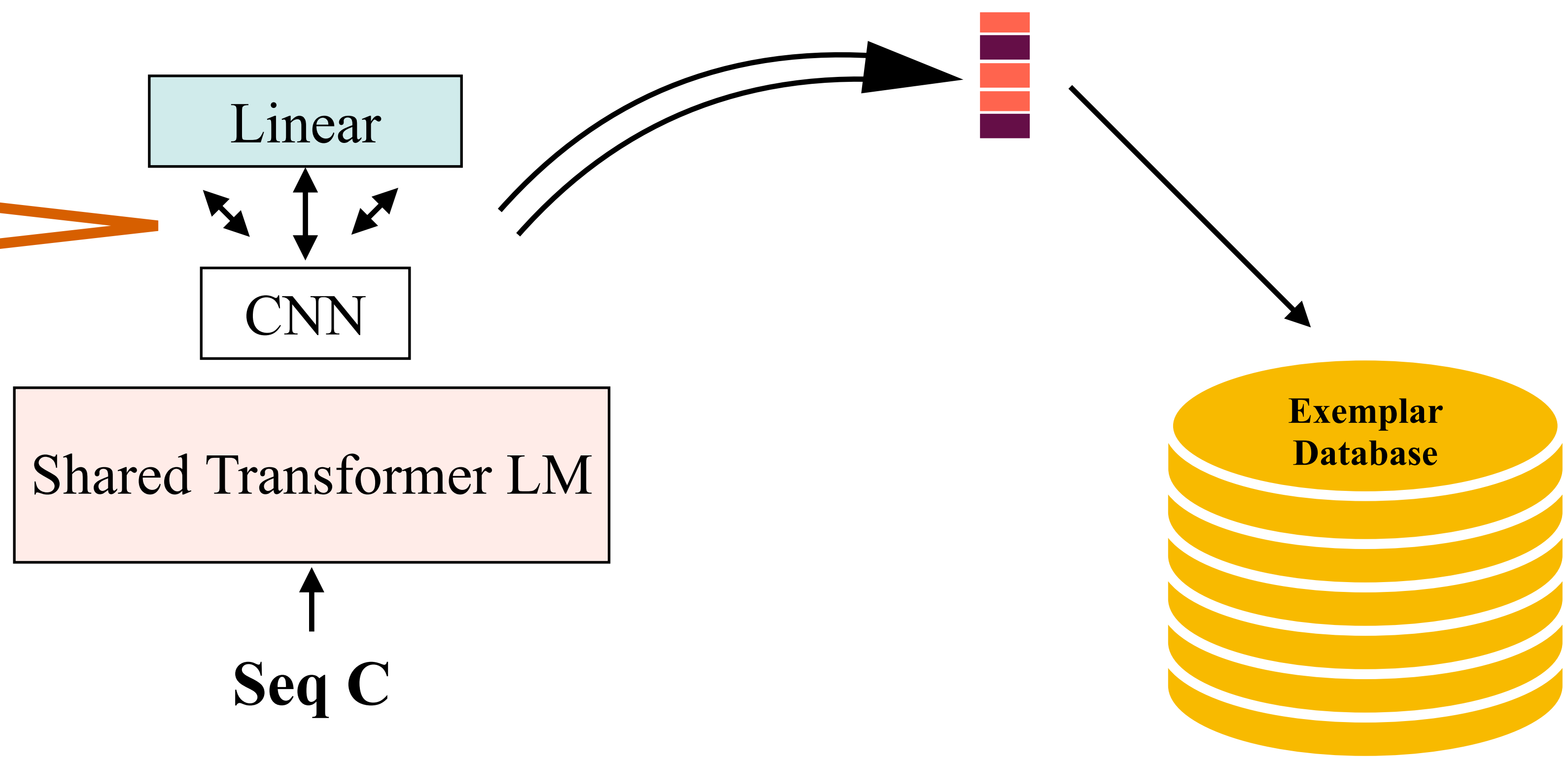


Multi-Sequence Representation Composition for Exemplar Auditing

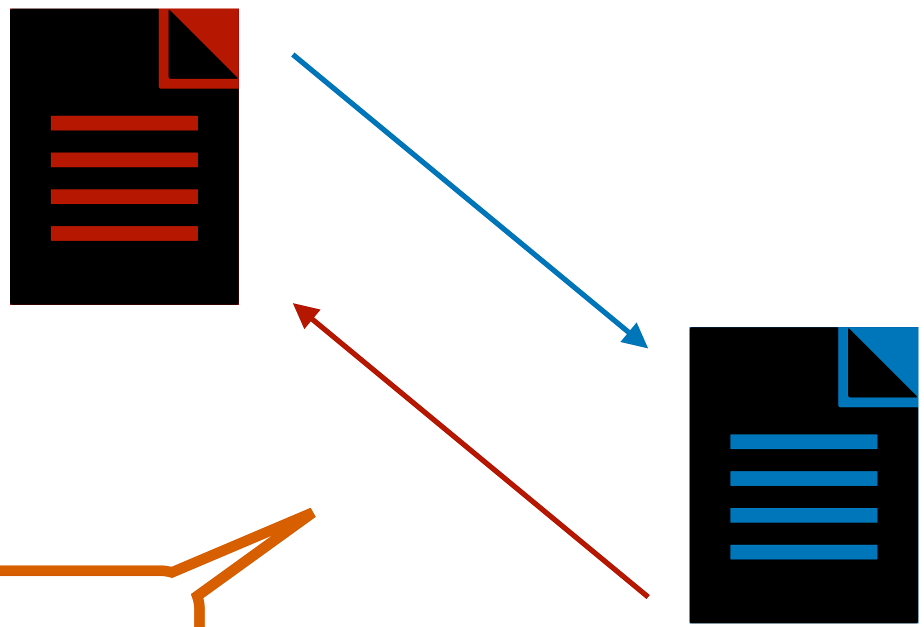


Token-Level Representations for Exemplar Auditing

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)



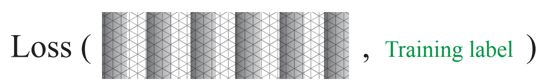
Extractive, Comparative (Feature-wise) Summarization



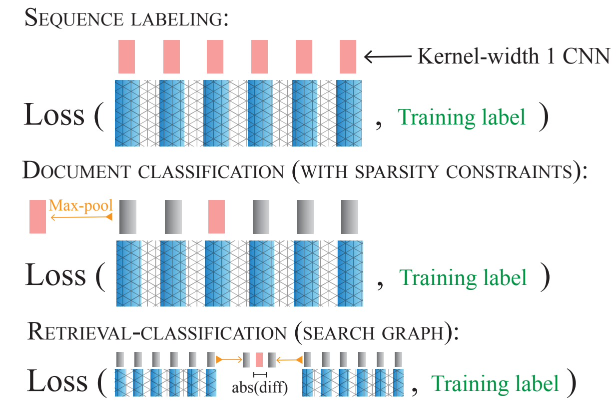
With facility over features, relating a global prediction to individual sequence elements, we can readily score, examine, & compare salient subsequences across correct & incorrect predictions for each class

Uncertainty Quantification

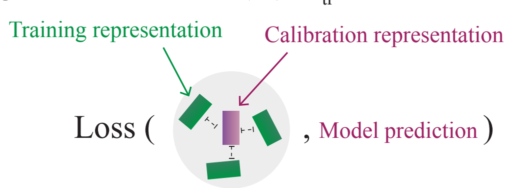
1. (Pre-) Train (& fine-tune) deep network, as usual.



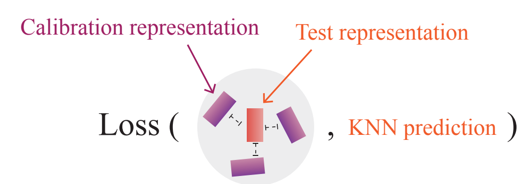
2. Freeze network. Add & train a memory layer for TASK. Extract exemplar representations.



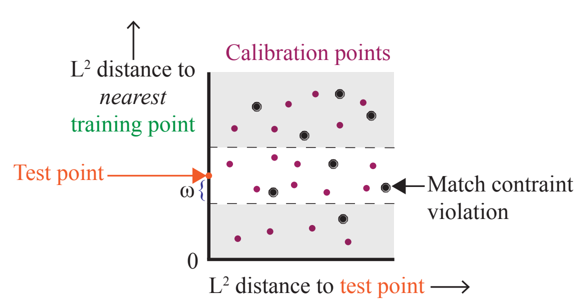
3. Train a KNN-based model approximation over exemplar representations from the memory layer, relating a new instance to training instances (predictions and ground-truth labels): $f(x)^{\mathrm{KNN}}_{\mathrm{tr}}$



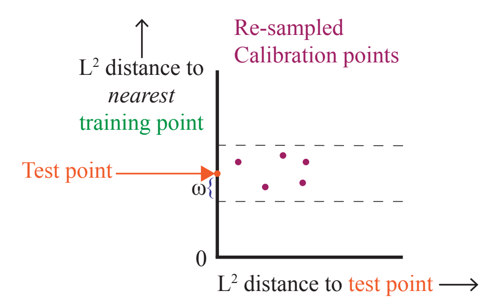
4. Train another KNN-based model approximation, relating a new test instance to representations and *KNN predictions* over the calibration set: $f(x)^{\overline{\mathrm{KNN}}}_{\mathrm{ca}}$



5. Calculate unique quantile thresholds *for each label for each test point* from the constrained set of calibration points within the distance band.



6. Optionally, re-sample the calibration set to be more similar to the test distribution. Repeat Step 5.

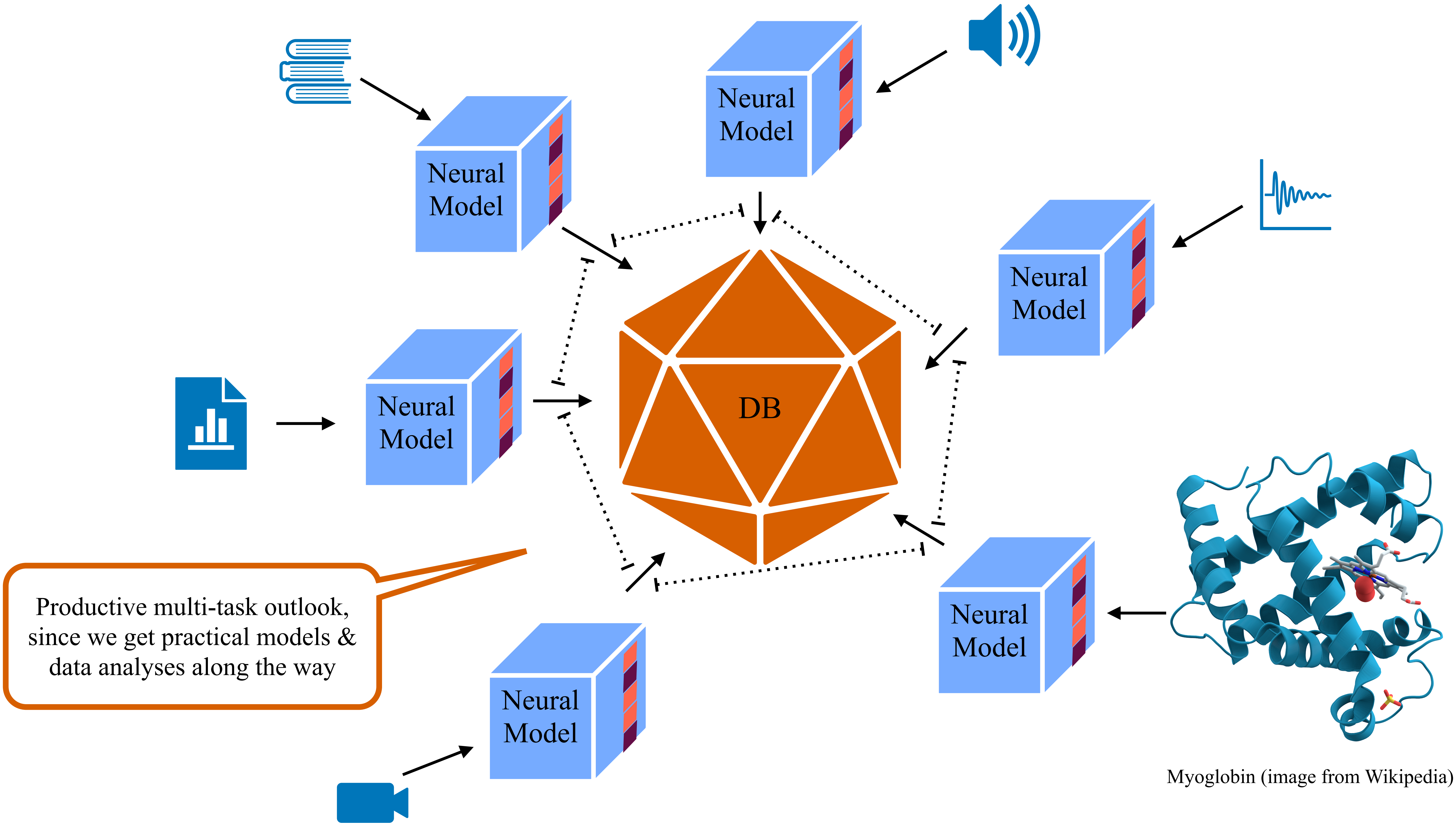


7. Optionally, condition on prediction set membership. Additional heuristics screen unreliable cases. (See text.)

ADMIT: A general framework for constructing, constraining, and analyzing point predictions and distribution-free prediction sets for deep neural networks.

Allen Schmaltz and Danielle Rasooly

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



Myoglobin (image from Wikipedia)

Allen Schmaltz