


Deep Networks as *hidden* Metric Learners

- N training instances: $x_1, \dots, x_n, \dots, x_N$
- Ground truth training labels: $y_1, \dots, y_n, \dots, y_N$
- 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: $f(x_{N+1}) \approx \sum_{n=1}^N y_n \cdot \alpha_n \cdot ||r_n - r_{N+1}||_2$  I.e., a test prediction is approx. a distance-weighting (between **“exemplar”** representations) over the training set

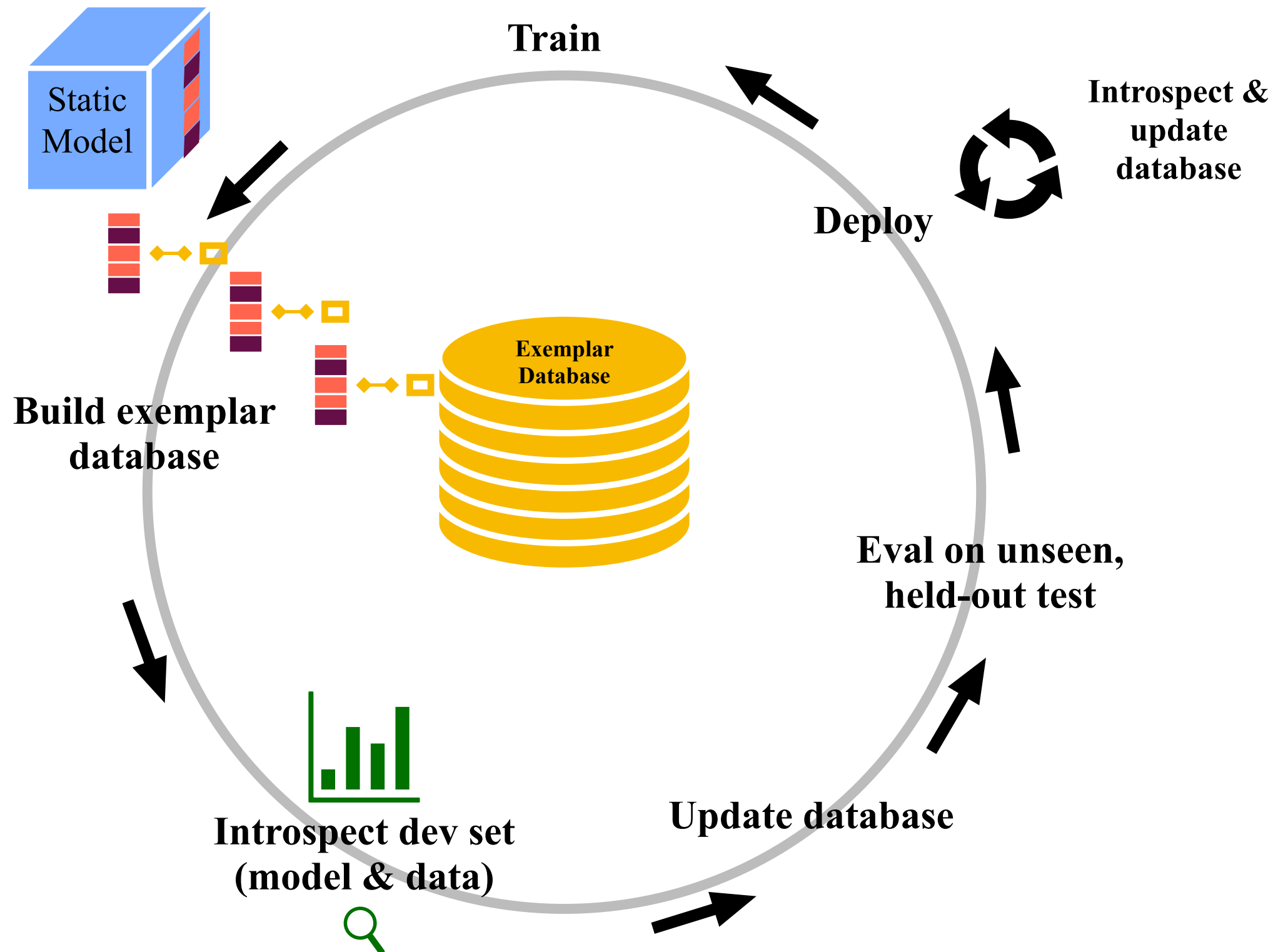
- Enables **interpretable/introspectable** decision rules & various analyses (hence, **“auditing”**): E.g., only admit true positive (TP) matches:
 $\hat{y}_{N+1} = f(x_{N+1}) \cdot [f(x_{N+1}) = f(x_n) \wedge f(x_n) = y_n] + NULL \cdot [f(x_{N+1}) \neq f(x_n) \vee f(x_n) \neq y_n]$, where $n = \arg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

- Enables **updatability/adaptability**:

- Label changes: $y'_n = y_n + \Delta_n$
- Data additions (a.k.a., continual/lifelong learning):
 $\mathbb{D}^N = \{(x_1, y_1), \dots, (x_N, y_N)\}$ becomes $\mathbb{D}^{N'} = \{(x_1, y_1), \dots, (x_N, y_N), \dots, (x_{N'}, y_{N'})\}$
- New lightweight models over representations (e.g., using data additions): $c': \mathbb{R}^M \rightarrow \mathbb{Y}'$

We use such training set matching as a post-processing mask (with decision rules) over $f(x_{N+i})$, but in principle, with the model presented here, we could directly train against this (via the similarity loss)

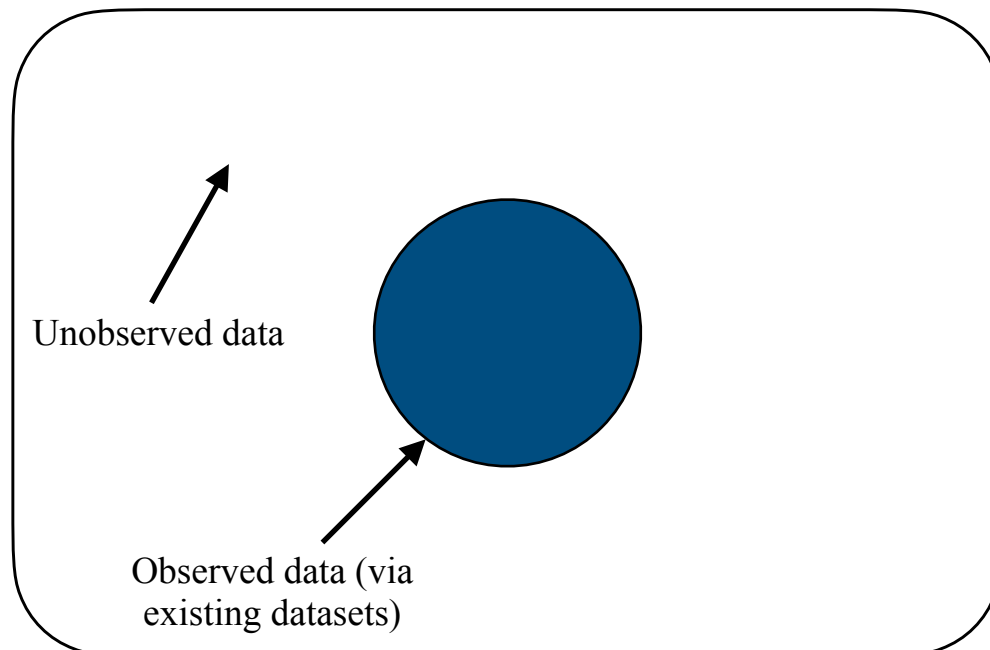
Exemplar Auditing Lifecycle



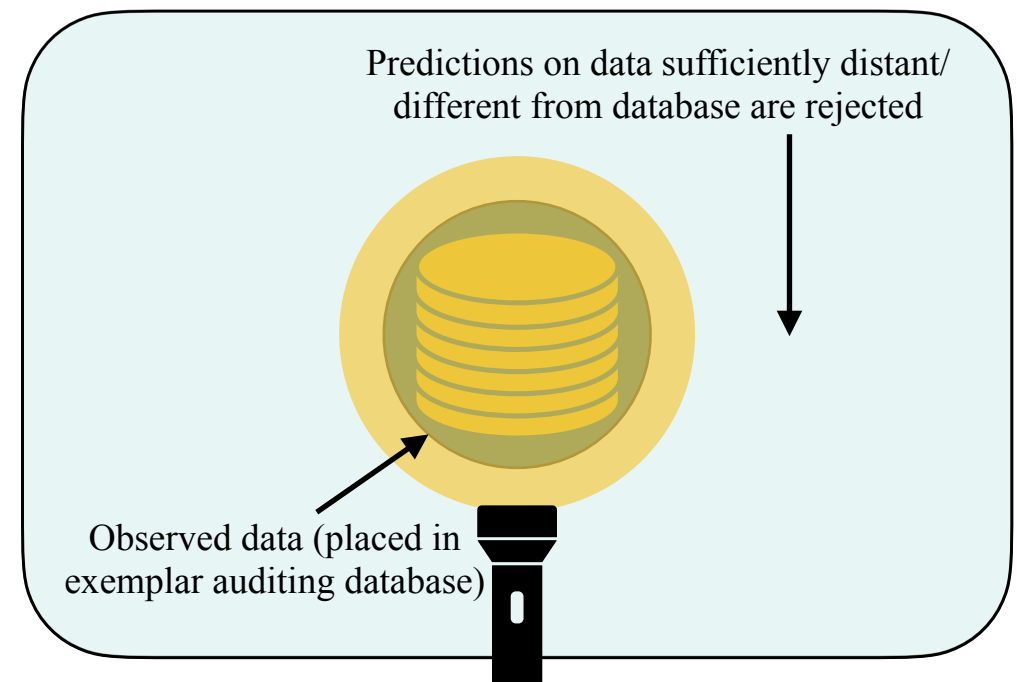
Out-of-Domain Settings

- 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)



Memory Matching Model

- 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

$q = \text{Query sequence}$

$s = \text{Support sequence}$

A Wikipedia sentence

$s_i \in \mathbb{D}$

Set of K nearest Wikipedia sentences

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

Set of Z nearest Wikipedia sentences from Search Level 2

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

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

$$g(s_1) = r_{s_1} \in \mathbb{R}^M$$

\vdots

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

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

Search Level 2

$$r_q \longleftrightarrow g((q, s'_1)) = r_{(q,s'_1)} \in \mathbb{R}^M$$

\vdots

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

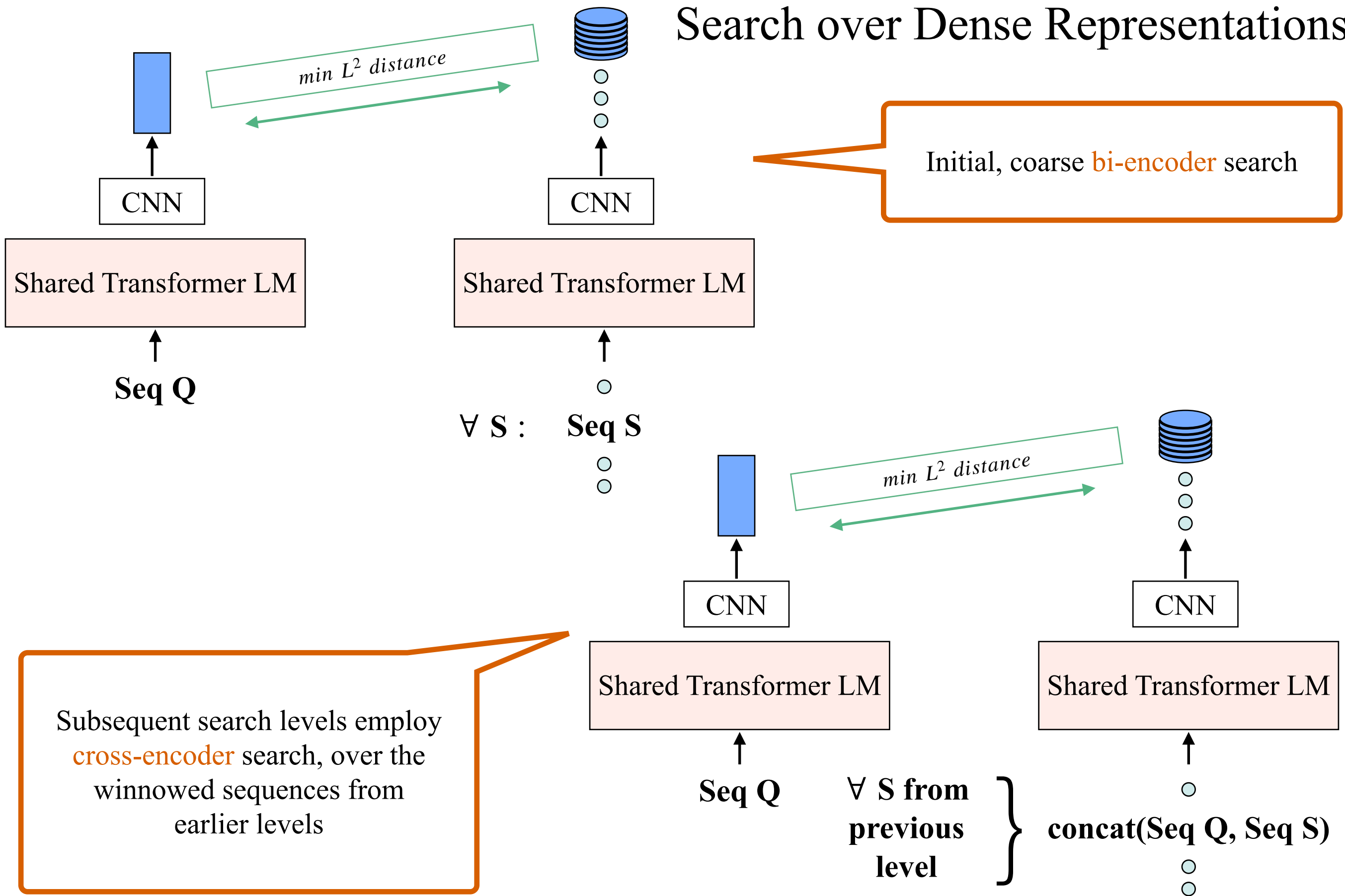
Search Level 3

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

\hat{y} is the label prediction

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

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



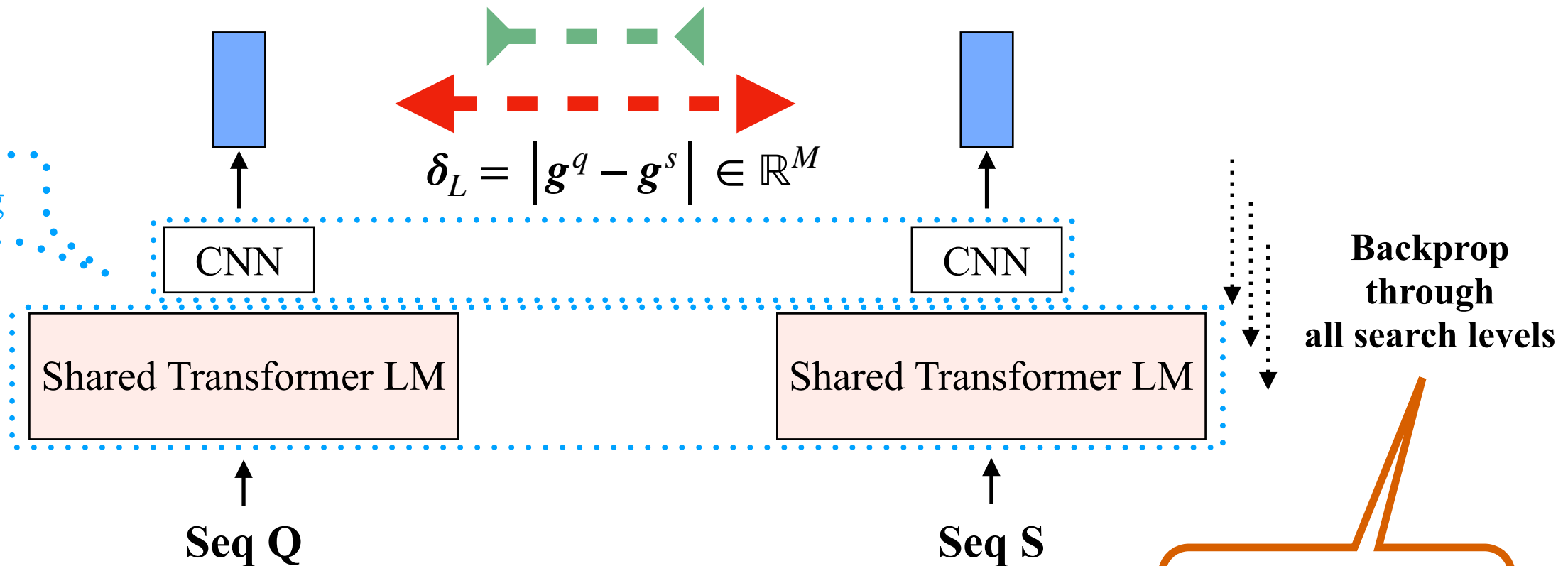
Joint Retrieval and Classification Training

Minimize/maximize difference
to
correct/incorrect matches



$$\delta_L = |g^q - g^s| \in \mathbb{R}^M$$

Iterative freezing



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

Multi-Sequence Representation Composition for Exemplar Auditing

