CS329X: Human Centered NLP

Human in the Loop

Diyi Yang

Stanford CS

Overview

Reasons we need human feedback

How models can take feedback

How humans can give feedback

Many slides credit to Sherry Wu

Why do we need human feedback?

A misalignment between this **fine-tuning objective** (maximizing the likelihood of human-written text) and **what we care about** (generating high-quality outputs as determined by humans).

The objective function mixes important errors (making up facts) and unimportant errors (selecting the precise word from a set of synonyms)

Models are incentivized to place probability mass on all human demonstrations, including those that are low-quality.

Some common objectives for human feedback…

A misalignment between this **fine-tuning objective** (maximizing the likelihood of human-written text) and **what we care about** (generating high-quality outputs as determined by humans).

Make model output more aligned with our values:

Model performance, robustness and generalizability (aligned with our expectations on model behaviors)

Fairness (aligned with our societal values)

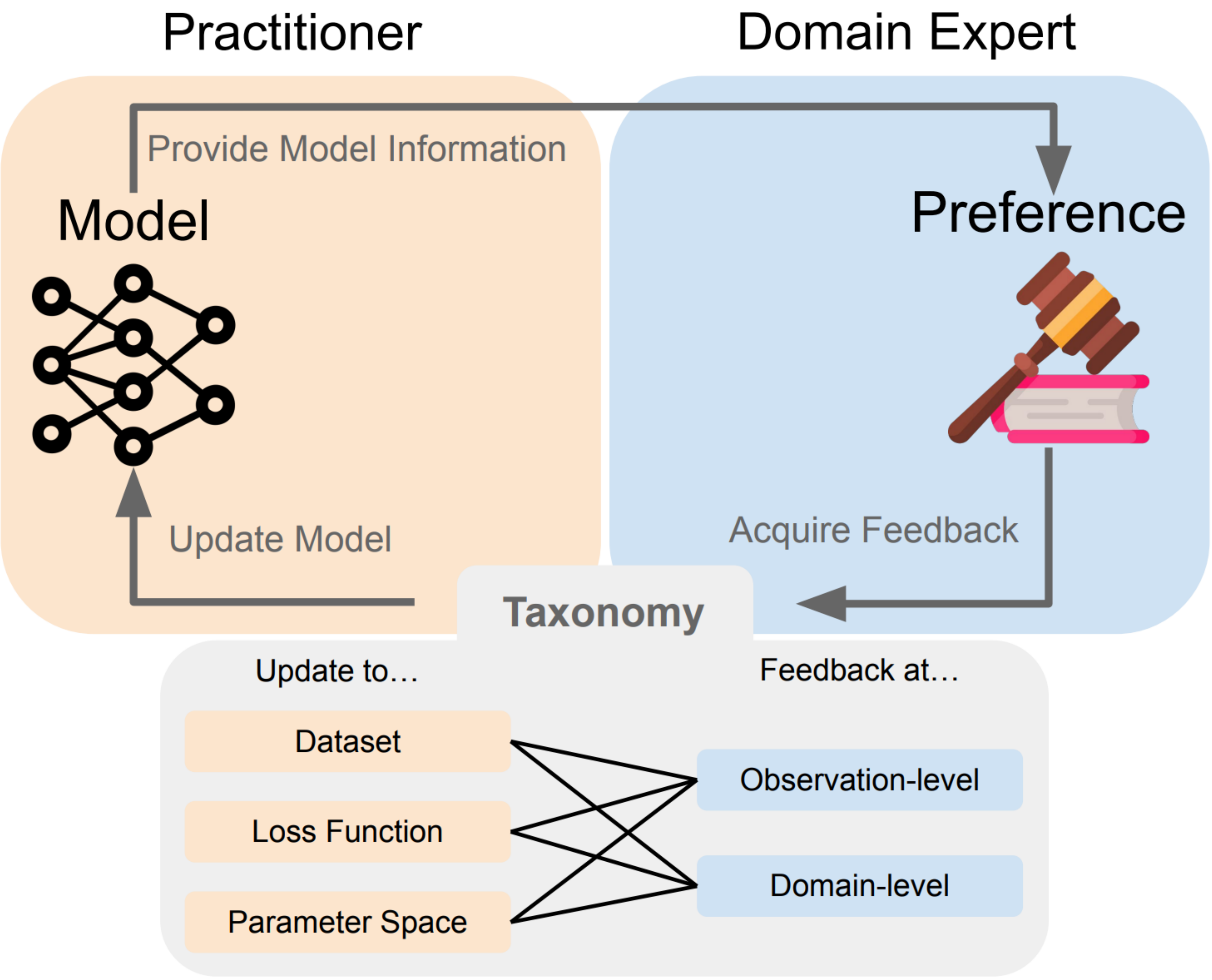
Explainability (aligned with our rationales)

Personal beliefs

What feedback can you imagine giving to a model?

Many forms, but might depends on what the model can take!

Keys of Human-in-the-loop NLP



Allow humans to ***easily provide feedback****.*

Turn nontechnical, human preferences into usable model updates.

Build models to ***effectively take the feedback****.*

Chen, Valerie, et al. "Perspectives on Incorporating Expert Feedback into Model Updates." *ArXiv* (2022).

Human in the loop NLP has a "long" history

Interactive Text Classification

Document-level interaction

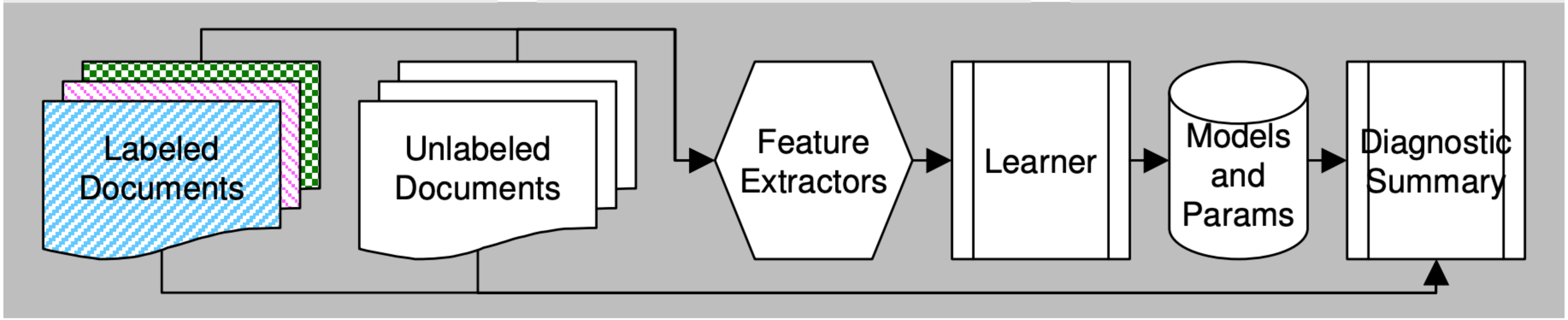
- Suggest documents to label
- Suggest labels for documents
- Check label consistency

Word-level interaction

- Suggest influential terms
- Accept engineered features
- Add/drop features

Model+data exploration

- Term by class parameters
- Drill down into documents
- Accuracy summaries



Godbole, Shantanu, Abhay Harpale, Sunita Sarawagi, and Soumen Chakrabarti. "Document classification through interactive supervision of document and term labels." In European Conference on Principles of Data Mining and Knowledge Discovery, pp. 185-196. Springer, Berlin, Heidelberg, 2004.

Human in the loop NLP has a “long” history

Pat ate the cake on the table that I ***baked*** last night.

Parser: I baked **table**

Human understanding: I baked **cake**

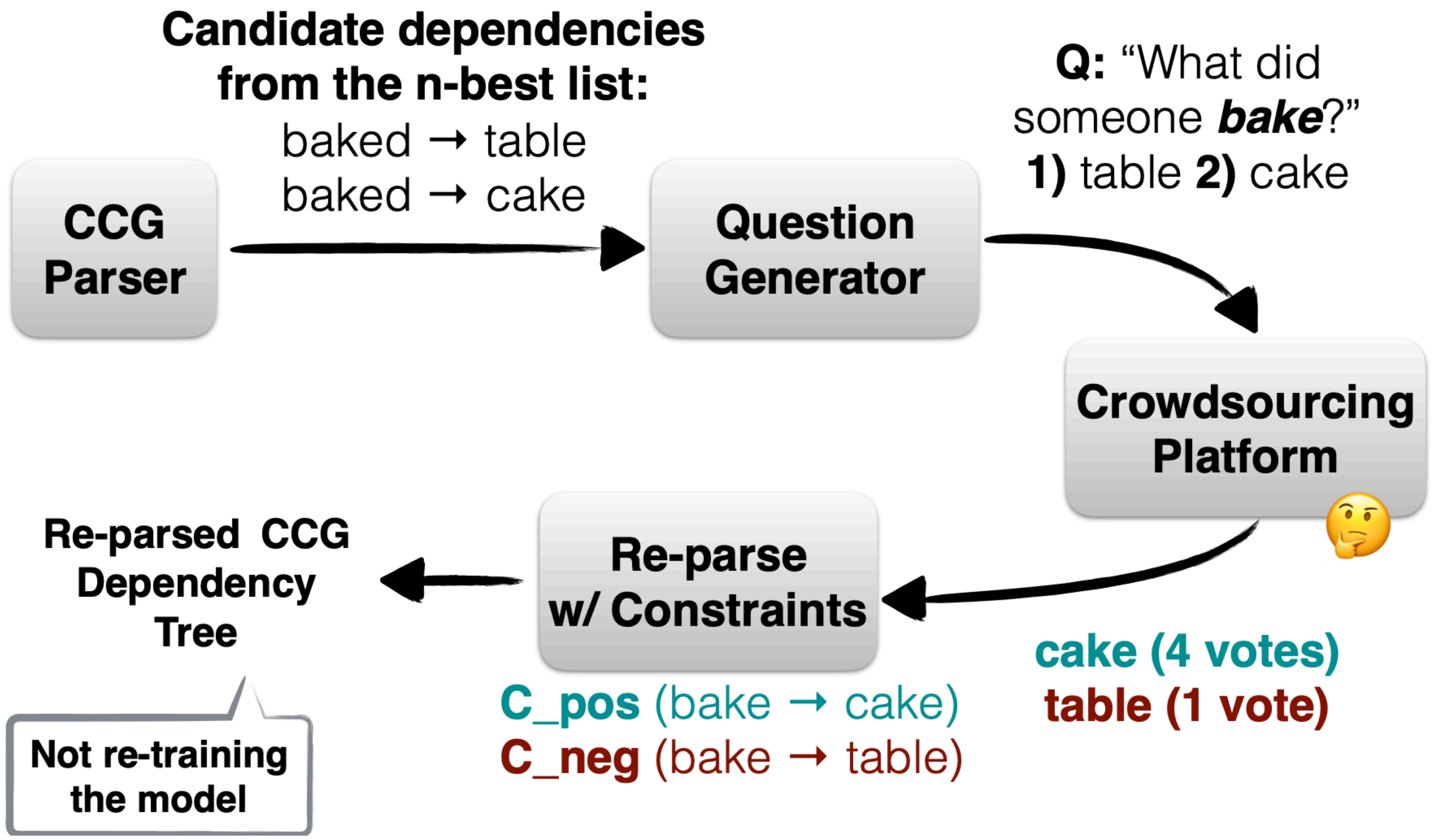
Human-in-the-Loop Parsing

Luheng He, Julian Michael, *Mike Lewis, Luke Zettlemoyer

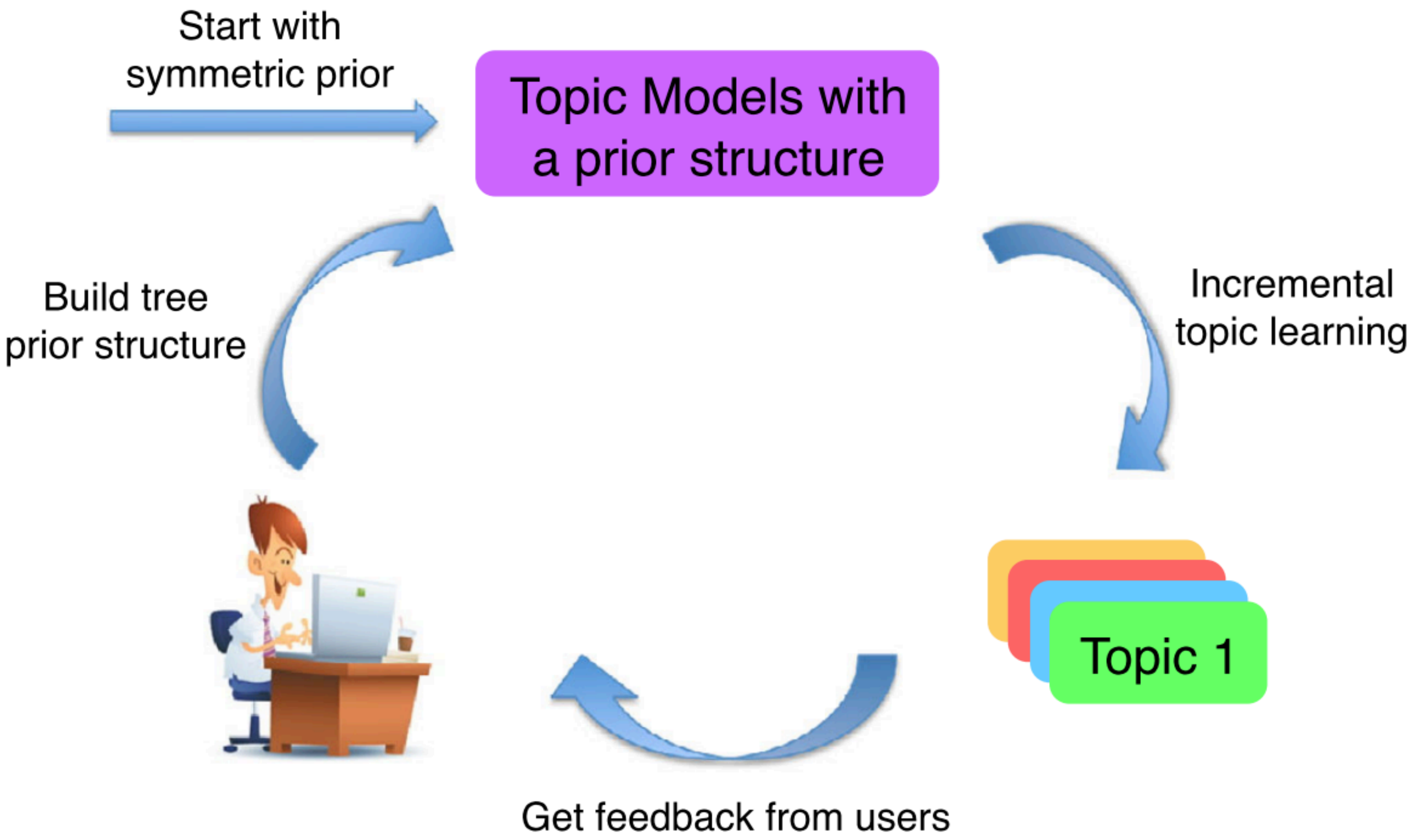
University of Washington

He, Luheng, Julian Michael, Mike Lewis, and Luke Zettlemoyer. "Human-in-the-loop parsing." In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 2337-2342. 2016.

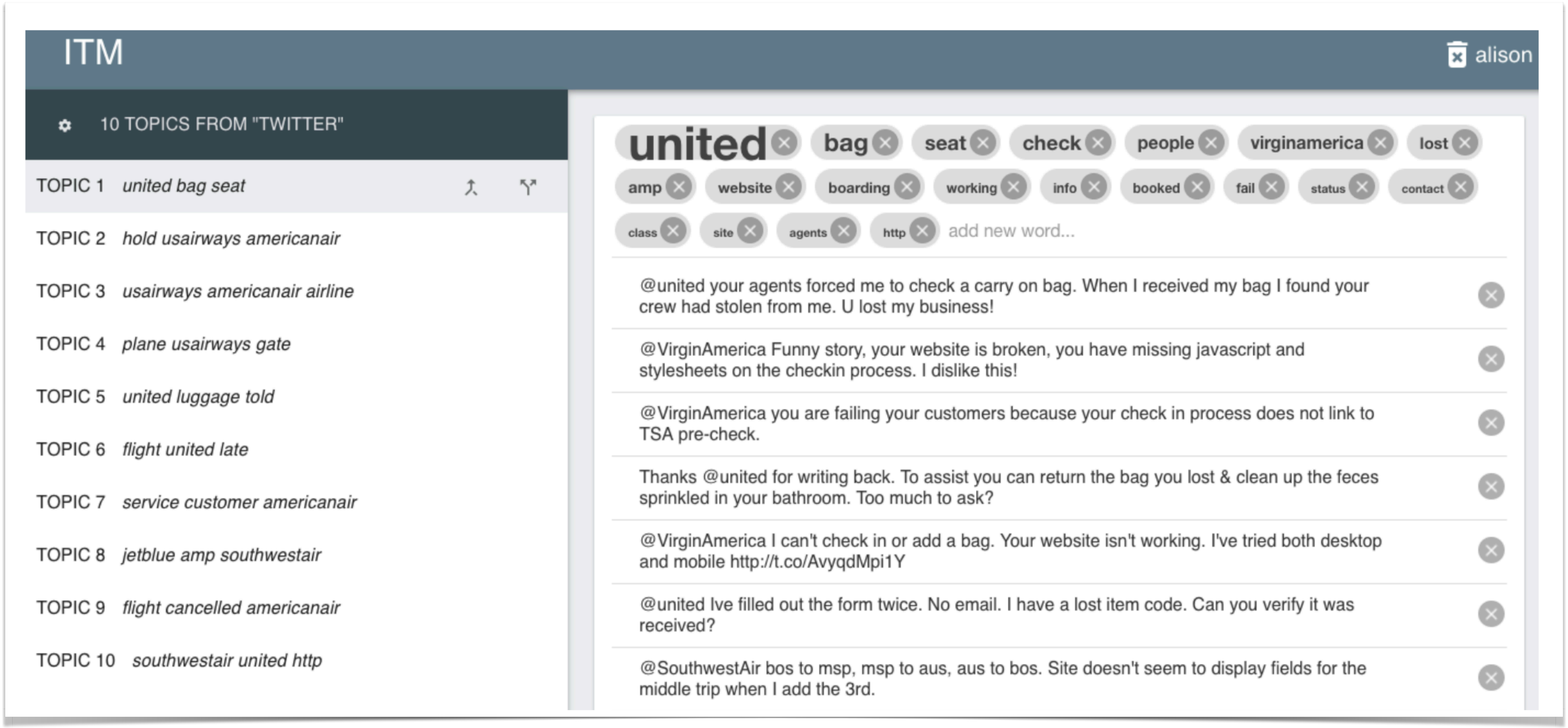
Human in the loop NLP has a “long” history



Human in the loop NLP has a "long" history

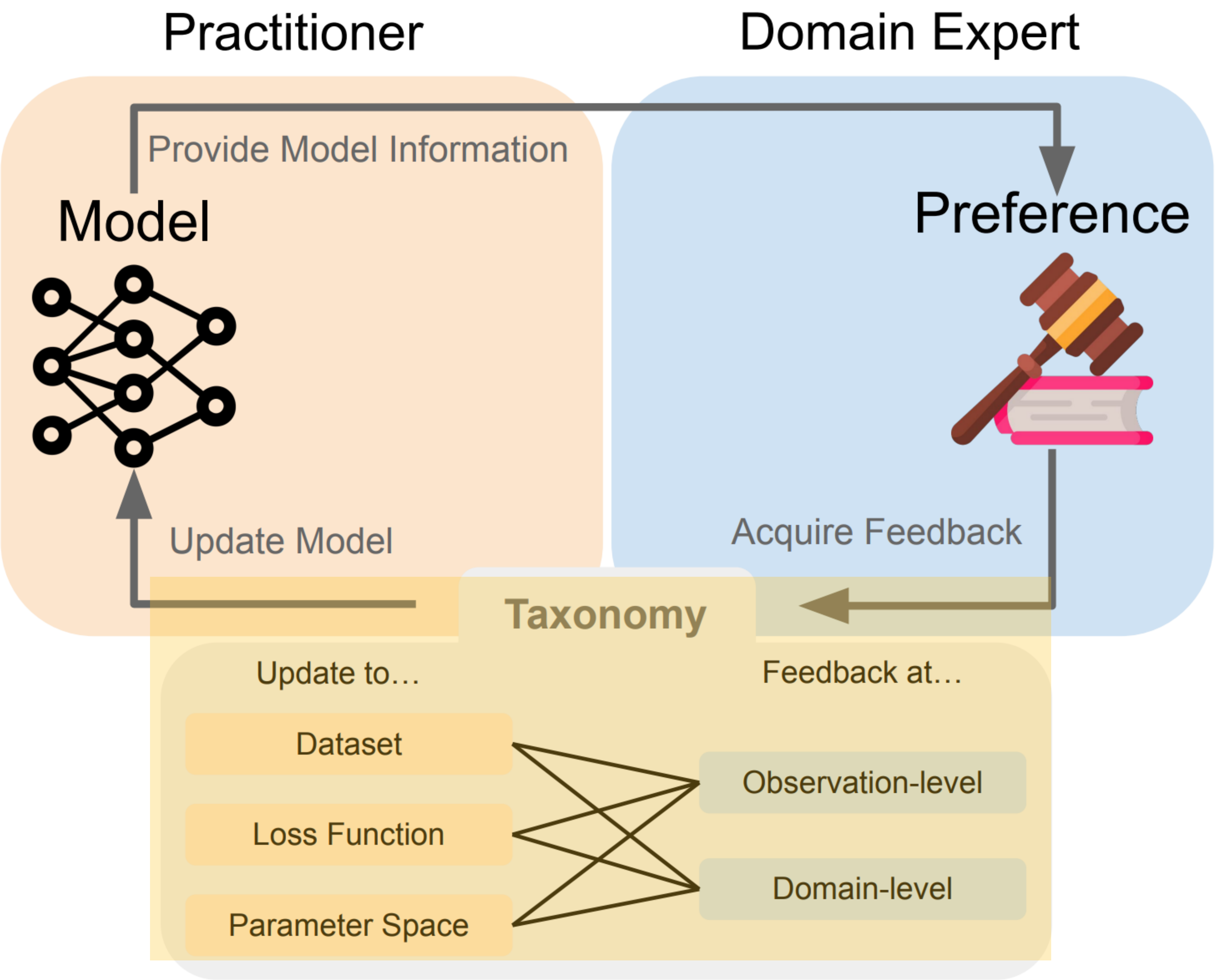


Interactive Topic Modeling: start with a vanilla LDA with symmetric prior, get the initial topics. Then repeat the following process till users are satisfied: show users topics, get feedback from users, encode the feedback into a tree prior, update topics with tree-based LDA



User interface for the HL-TM tool. A list of topics (left) are represented by topics' first three topic words. Selecting a topic reveals more detail (right): the top 20 words and top 40 documents. Hovering or clicking on a word highlights it within the documents. Users can refine the model using simple mechanisms: click "x" next to words or documents to remove them, select and drag words to re-order them, type new words from the vocabulary into the input box and press "enter" to add them, select a word and click the trash can to add it to the stop words list, or click "split" and "merge" (to the right of the topic words) to enter into split and merge modes.

Keys of Human-in-the-loop NLP



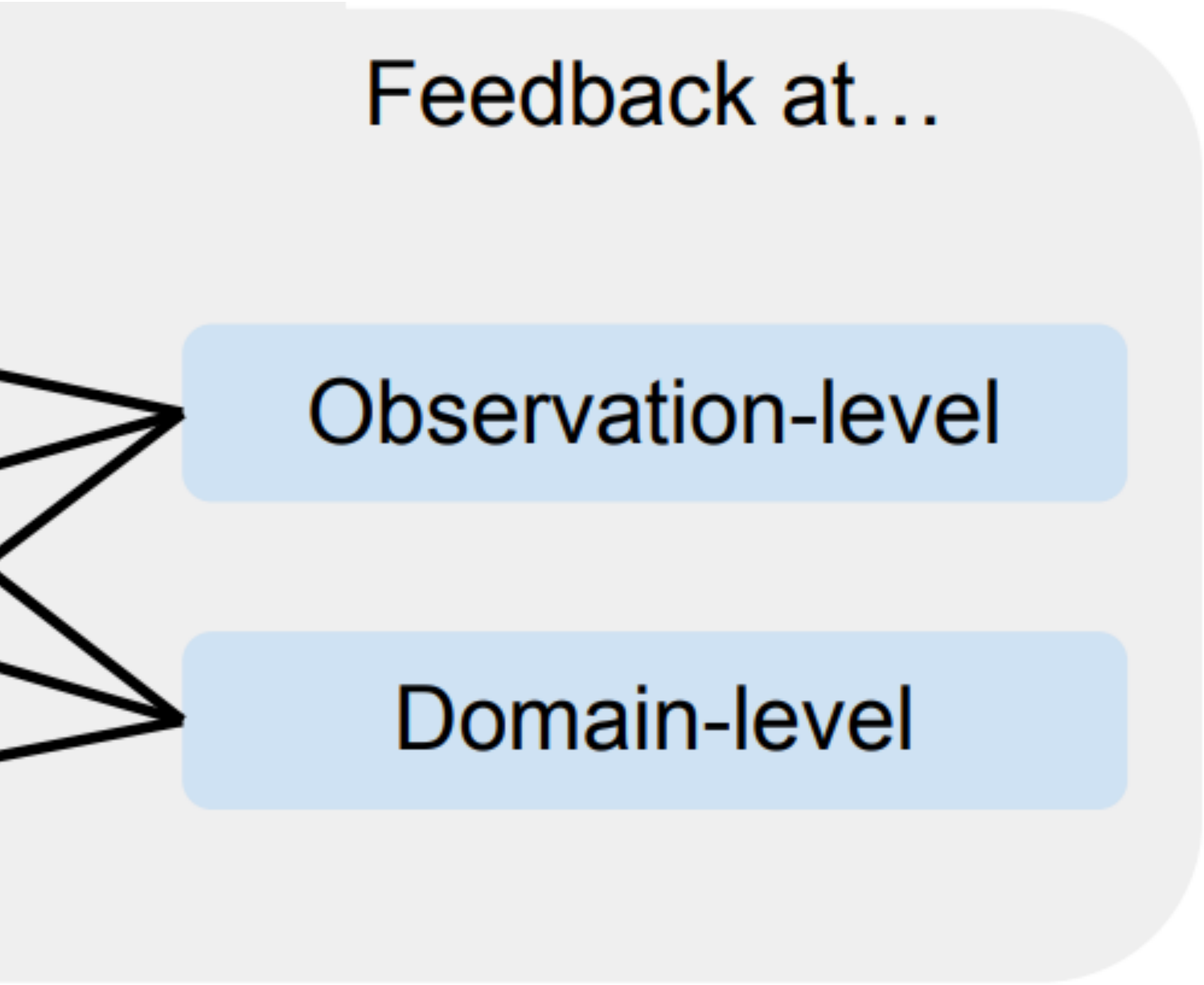
*Allow humans to **easily provide feedback**.*

Turn nontechnical, human preferences into usable model updates.

*Build models to **effectively take the feedback**.*

Chen, Valerie, et al. "Perspectives on Incorporating Expert Feedback into Model Updates." *ArXiv* (2022).

Taxonomy: Levels of domain expert feedback



Observation-level feedback (local)
Infer preferences from human judgements on each data points
e.g., radiologist provide gold annotations on X-ray scans

Domain-level feedback (global)
Provide explicit feedback on the entire task.
e.g., radiologist provide high-level descriptions about the region of interest in X-Rays

Taxonomy: Types of Model Updates

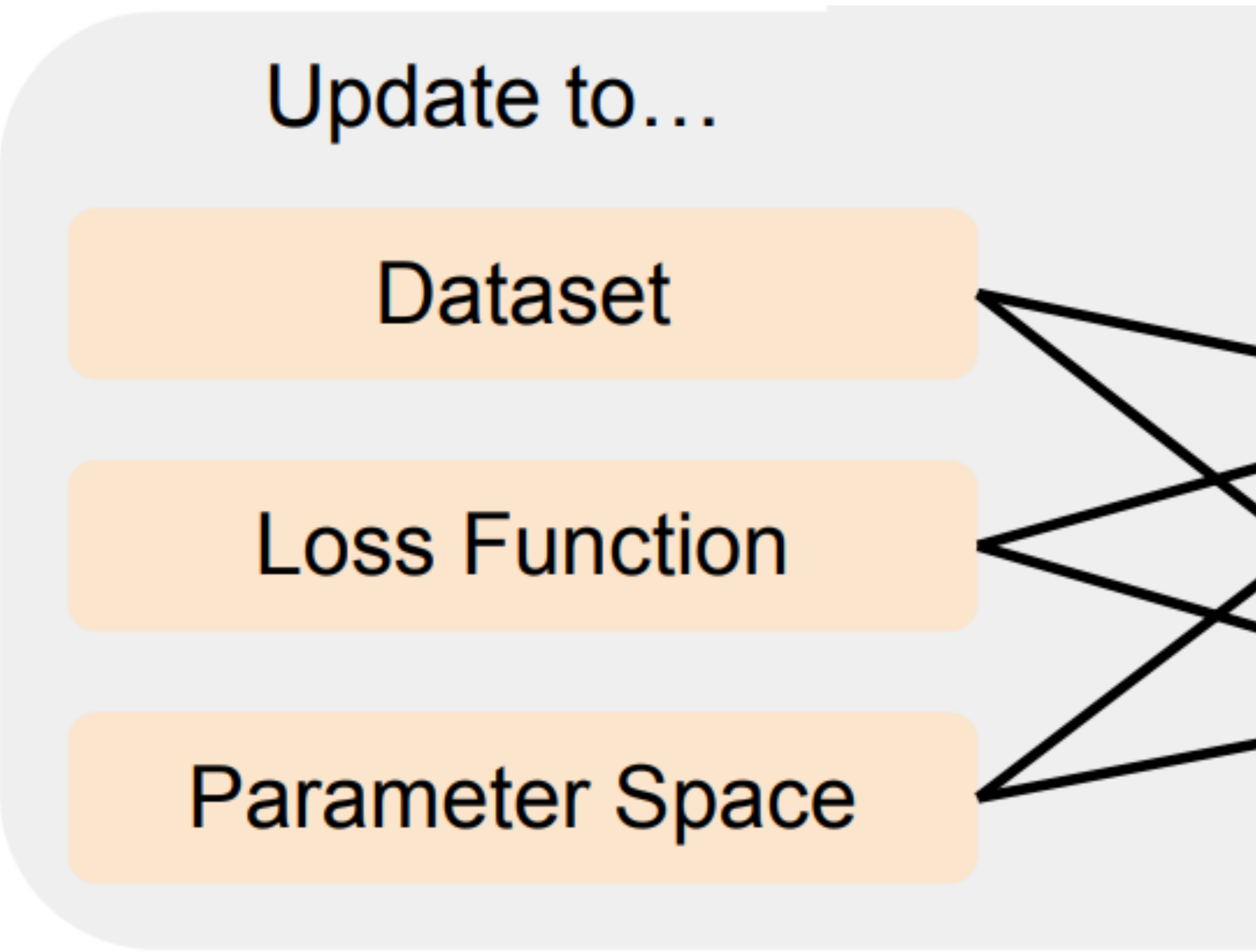
The supervised learning setting

By minimizing a objective function

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \sum_{(x,y) \in D} L(x, y; \theta)$$

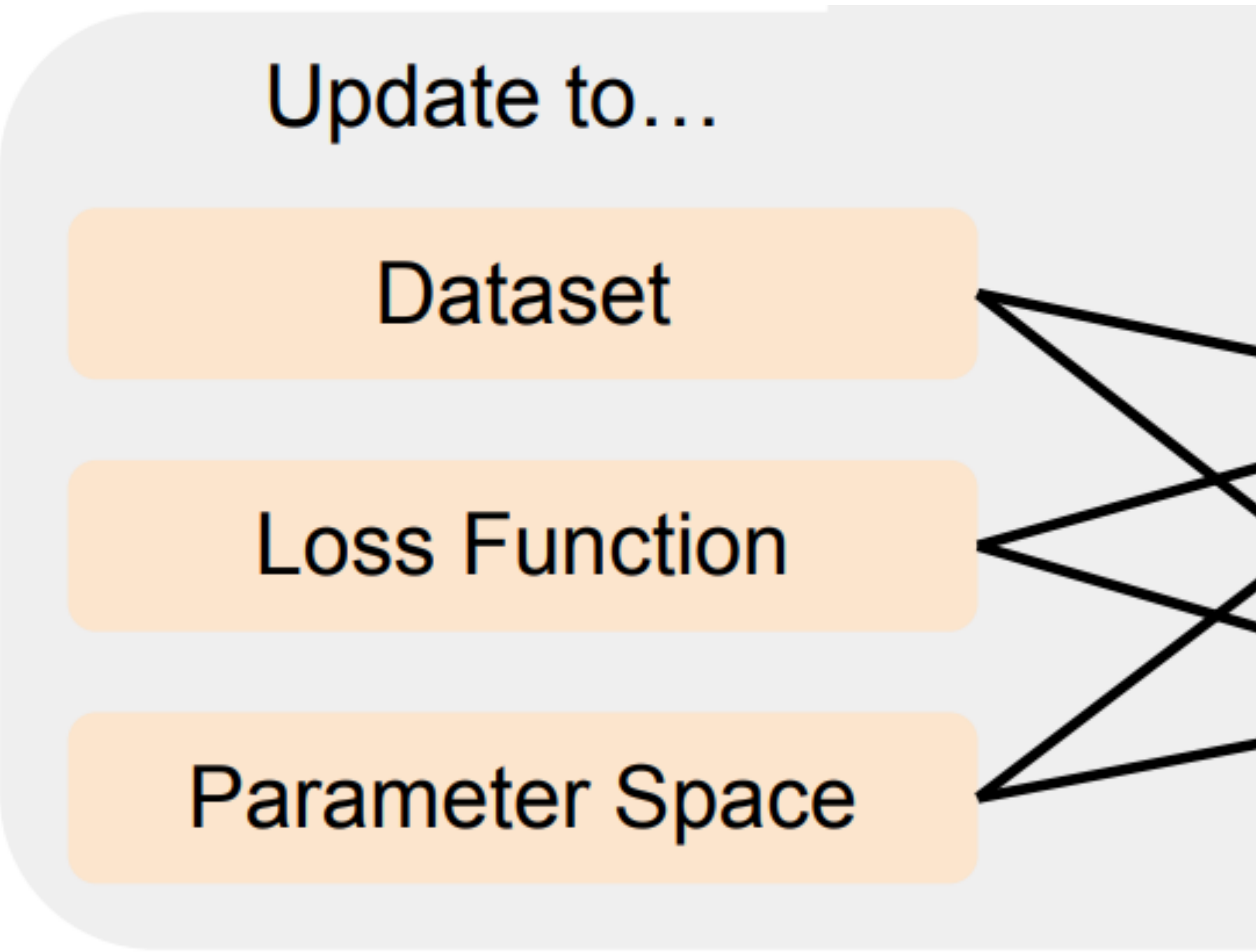
Learn a model parametrized by $\theta \in \Theta$

On a dataset D



Taxonomy: Types of Model Updates

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \sum_{(x,y) \in D} L(x, y; \theta)$$



Dataset updates. change the dataset

e.g., add / remov appropriate datapoints

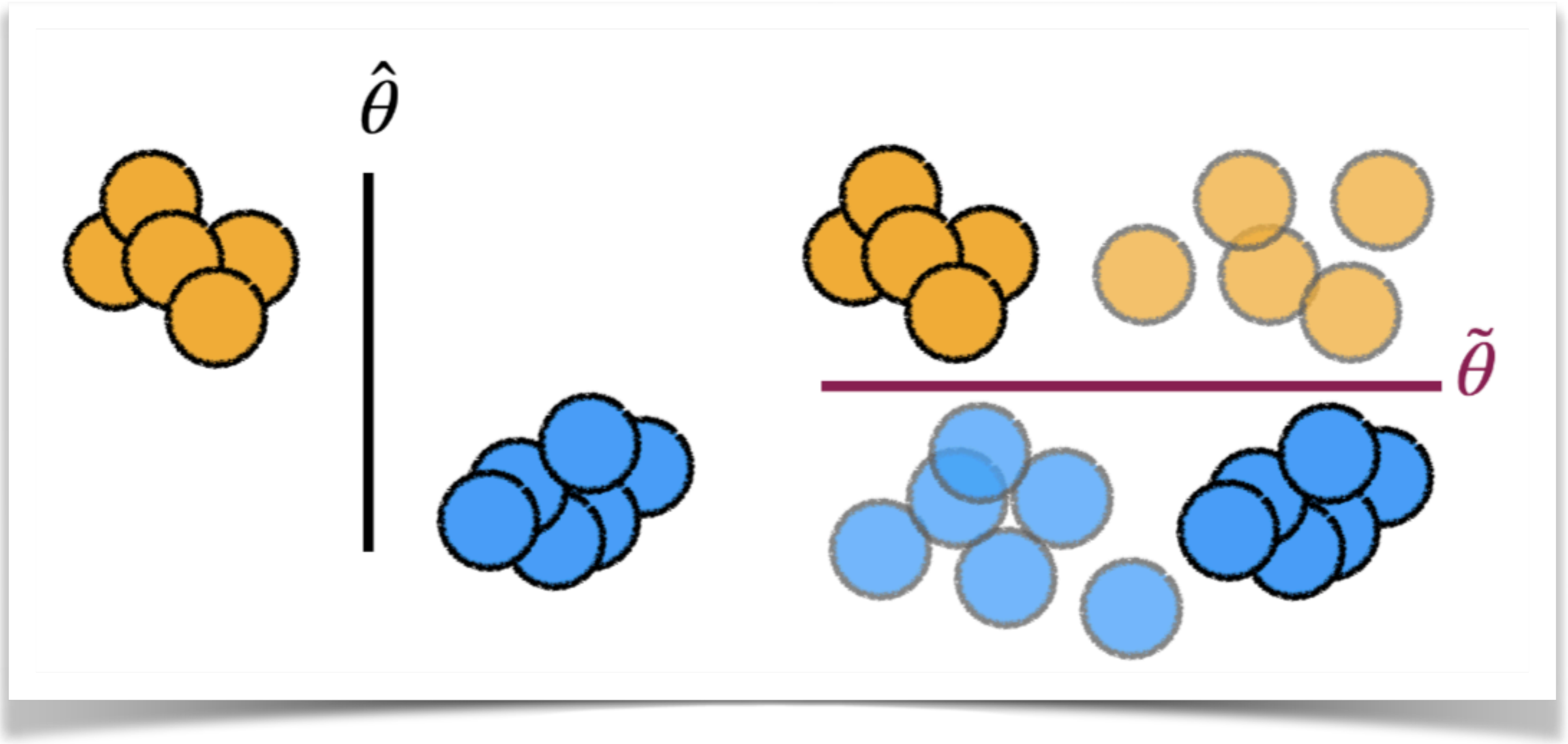
Loss function updates. add a constraint to the optimization objective

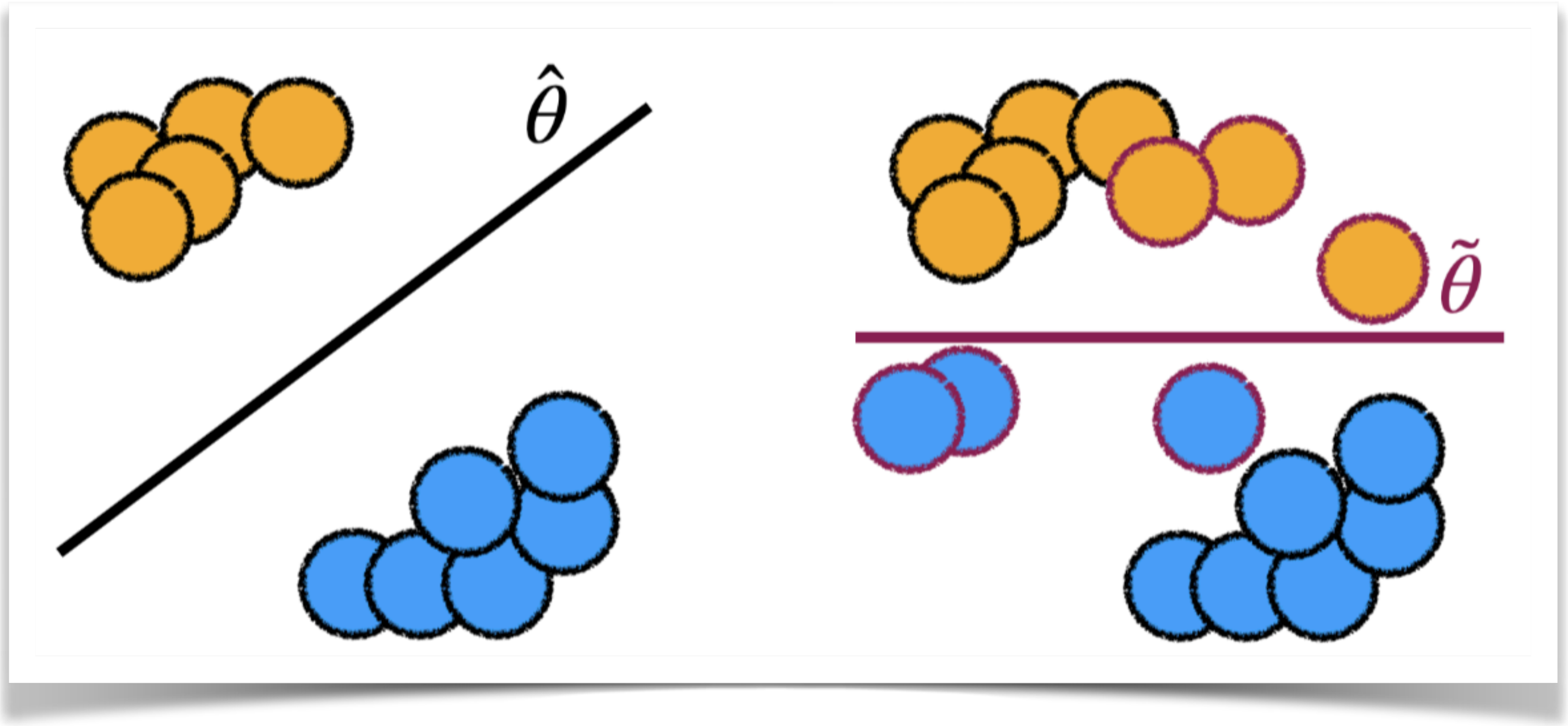
e.g., add a regularizer that penalizes the model for not satisfying this condition

Parameter space updates. Change the model parameters

e.g., optimize over a subspace of parameters

Update Datasets (aka Data curation)





Global: systematically add data points

Data augmentation

Resampling

Local: Iteratively add data points

Active learning

model-assisted adversarial labeling

Global data update: Weak supervision

Weak supervision: Use imperfect or noisy sources of supervision to train models.
Snorkel key idea: data is key, but data collection is too expensive.
We should try using ***noisy sources of signal***, specified at ***higher-levels of abstraction***, to rapidly generate training sets.

Write labeling functions to express domain expertise.

snorkel

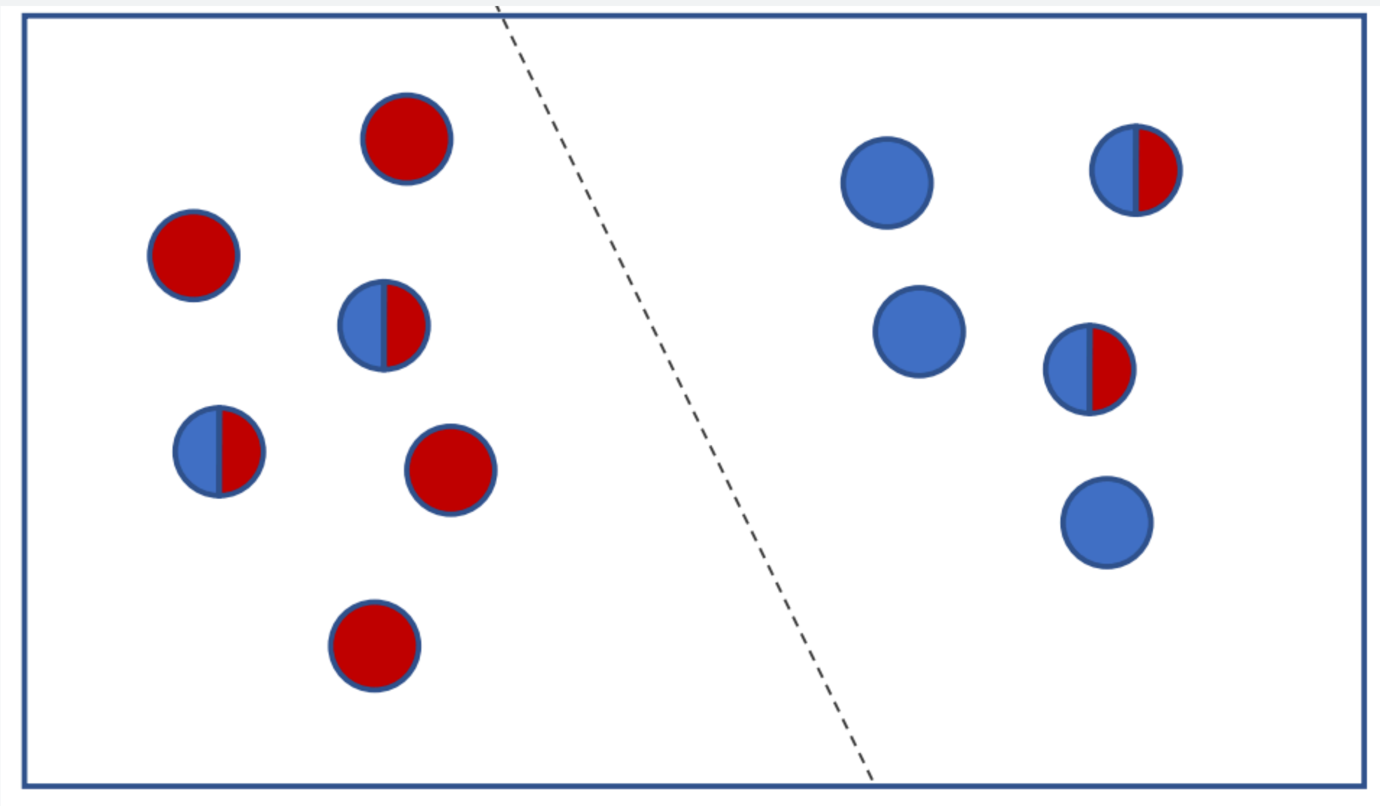
"We find that **Chemical A** likely does **not** cause **Disease X**."

```
def labeling_function_1(x):
    if re.find(r'not', x.between):
        return False
```

Ratner, Alexander, et al. "Snorkel: Rapid training data creation with weak supervision." *VLDB* 2017.
Slides adjusted from Alex Ratner's presentation

Global data update: Weak supervision

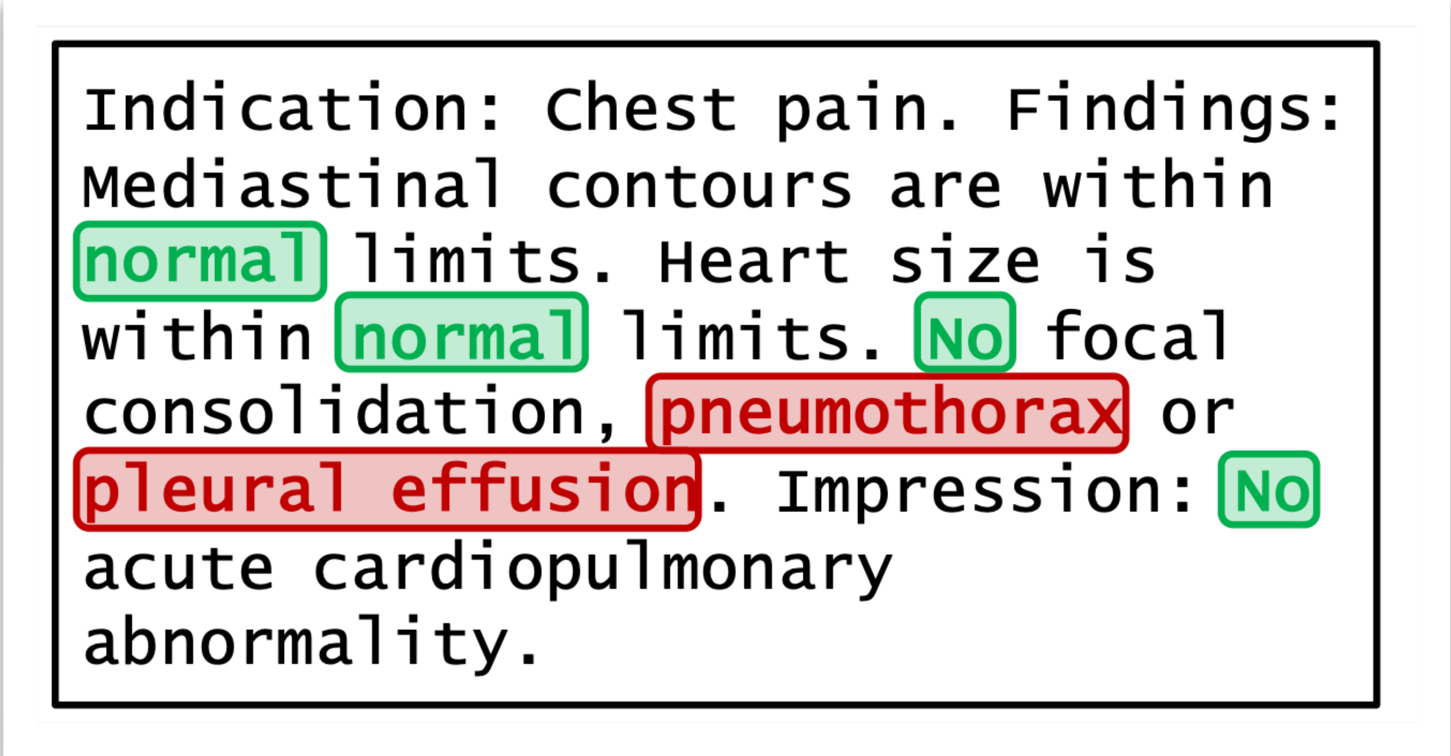
Input: LFs, Unlabeled data



Use LFs to produce
Noisy, conflicting labels

```
def LF_pneumothorax(c):
  if re.search(r'pneumo.*', c.report.text):
      return "ABNORMAL"

def LF_pleural_effusion(c):
  if "pleural effusion" in c.report.text:
      return "ABNORMAL"
```

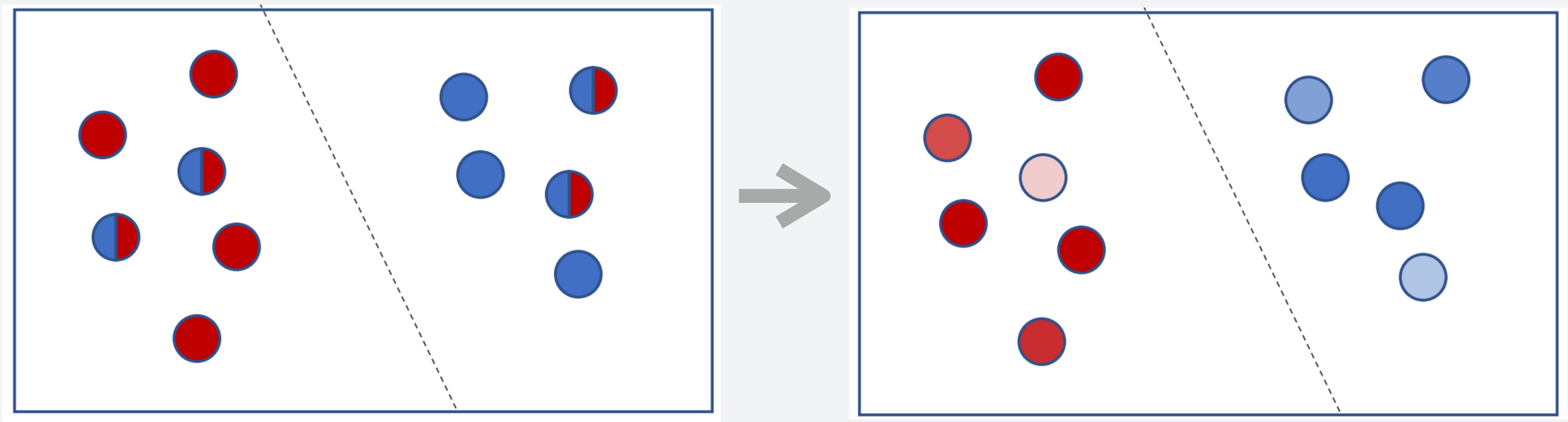


Indication: Chest pain. Findings: Mediastinal contours are within normal limits. Heart size is within normal limits. No focal consolidation, pneumothorax or pleural effusion. Impression: No acute cardiopulmonary abnormality.

Global data update: Weak supervision

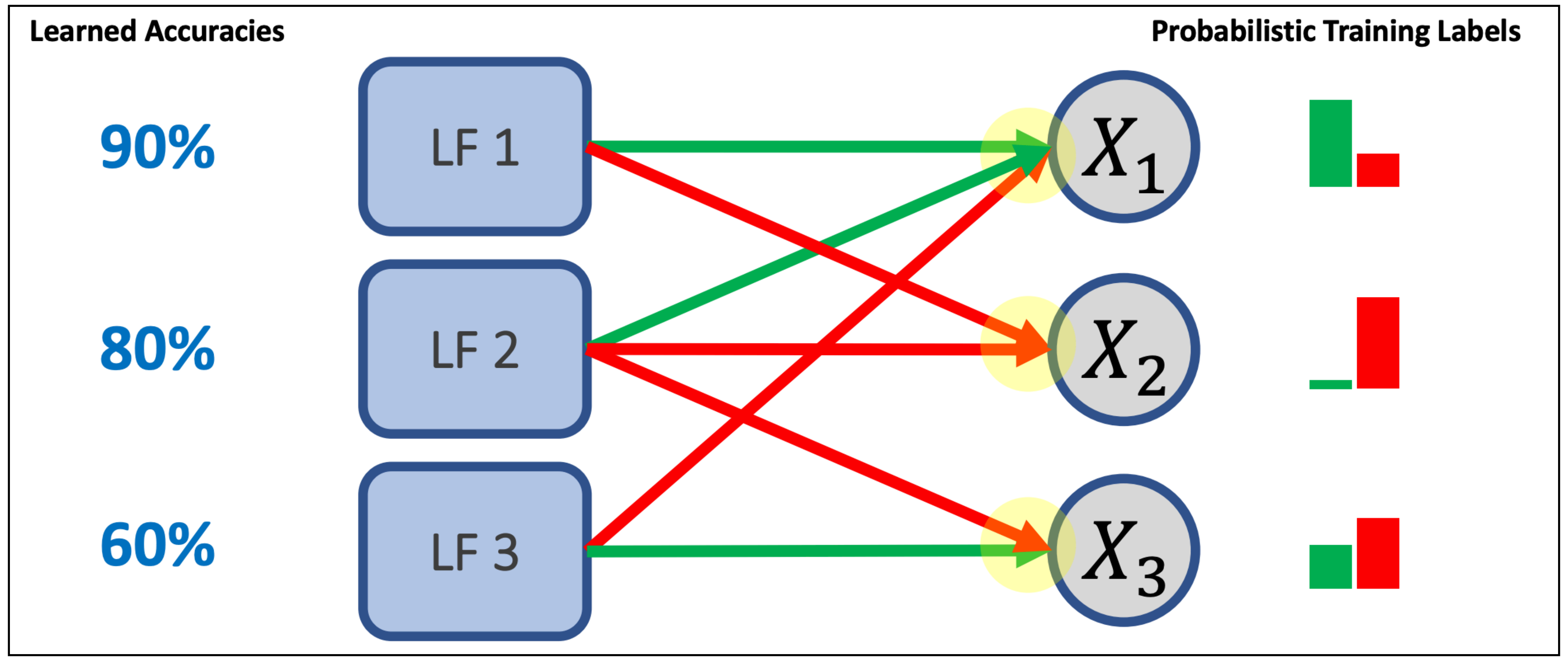
Input: LFs, Unlabeled data

Label Model



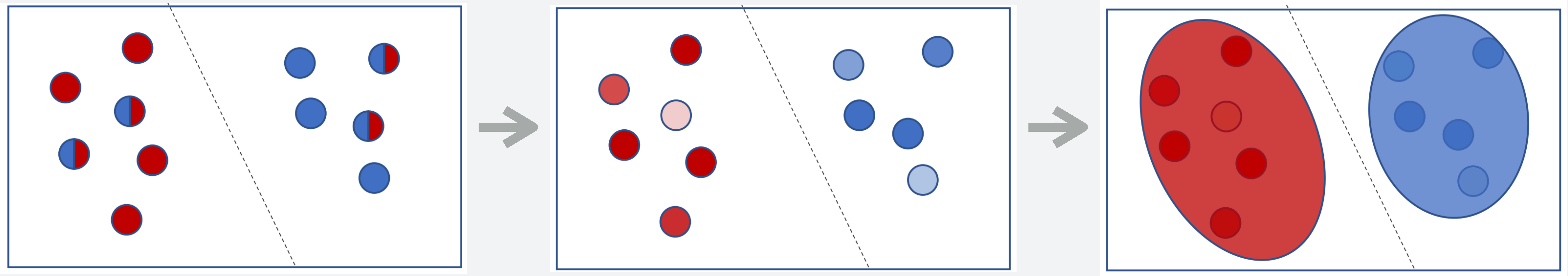
Use LFs to produce
Noisy, conflicting labels

Resolve conflicts,
re-weight & combine



Global data update: Weak supervision

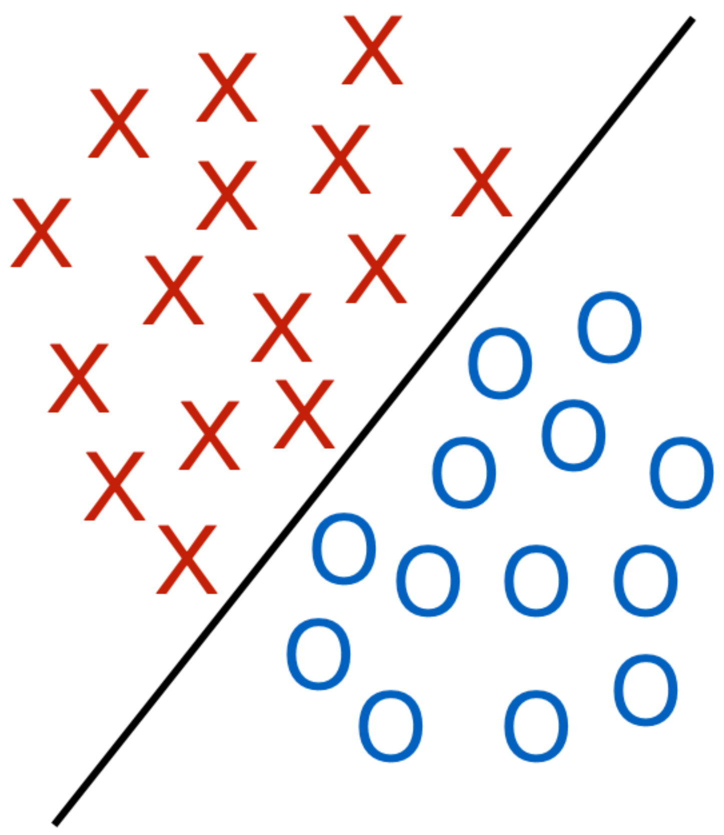
Input: LFs, Unlabeled data	Label Model	End Model



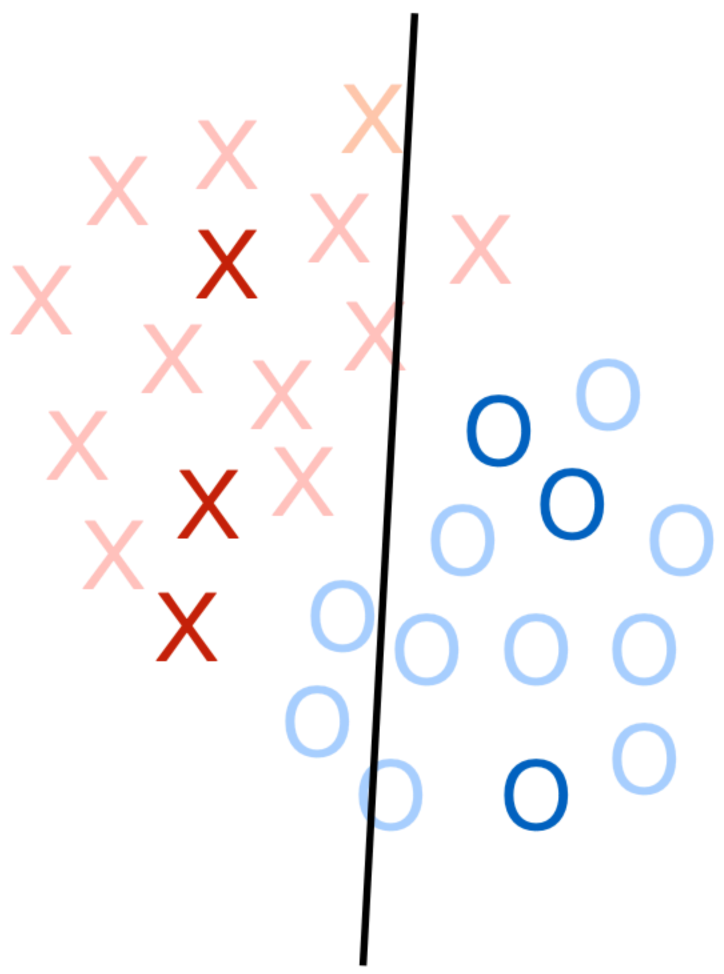
Use LFs to produce Noisy, conflicting labels	Resolve conflicts, re-weight & combine	Generalize beyond labeling functions

Local data update: Active learning

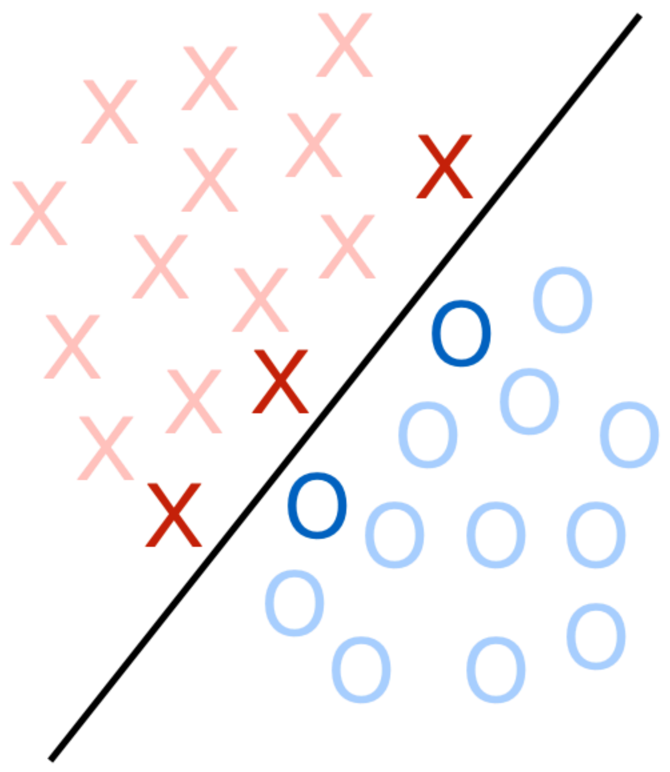
Active learning: Proactively select which data points we want to use to learn from, rather than passively accepting all data points available.



Groundtruth



Less effective data

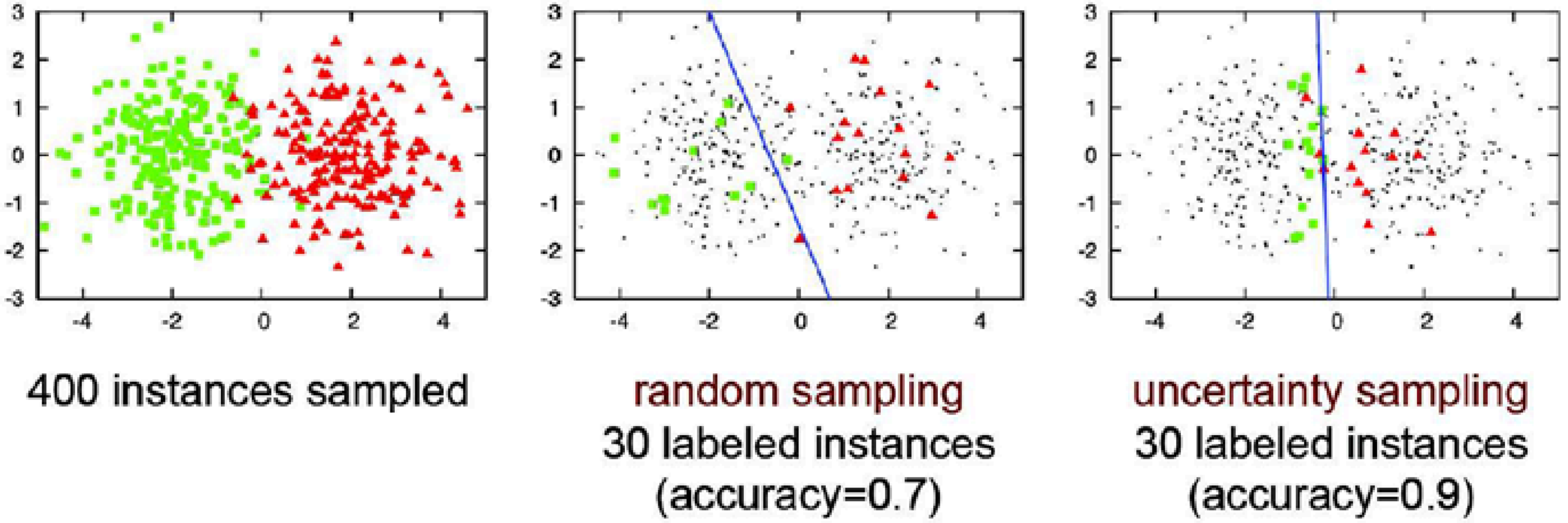


More effective data

Intuition: If we have limited labeling budget, some data points are more useful for learning the true decision boundary than others.

Local data update: Active learning

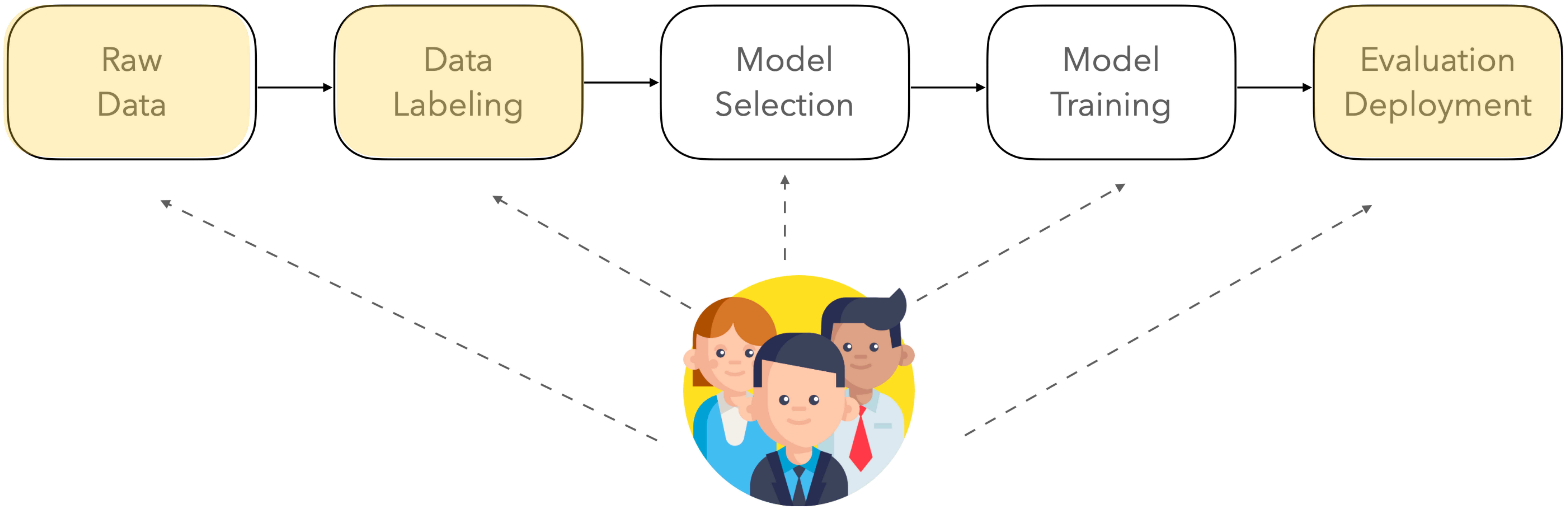
Active learning: Proactively select which data points we want to use to learn from, rather than passively accepting all data points available.



There are multiple ways to estimate “usefulness”, e.g. **uncertainty**.

We provide this form of feedback…

Mostly at places where we have data.



Wang, Zijie J., et al. "Putting humans in the natural language processing loop: A survey." *HCI+NLP Workshop* (2021).

Local vs. Global feedback

As you will also see in other examples…

Global feedback tends to be

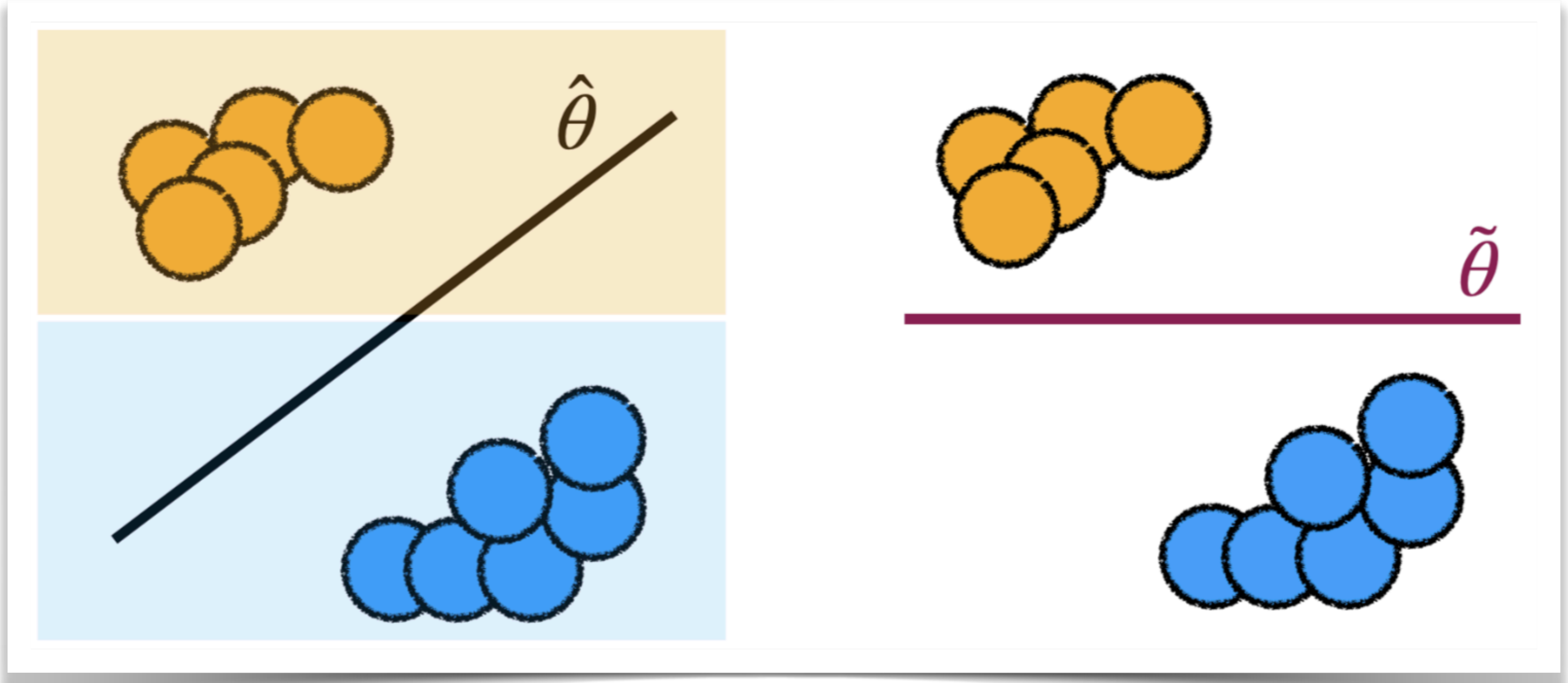
- **More explicit.** requires you to specify what you want
- **More "intrusive"** & has **larger impacts.** e.g.,you can use LF on 10k+ data
- *Be cautious about making large but not thoughtful changes!*

Local feedback tends to be

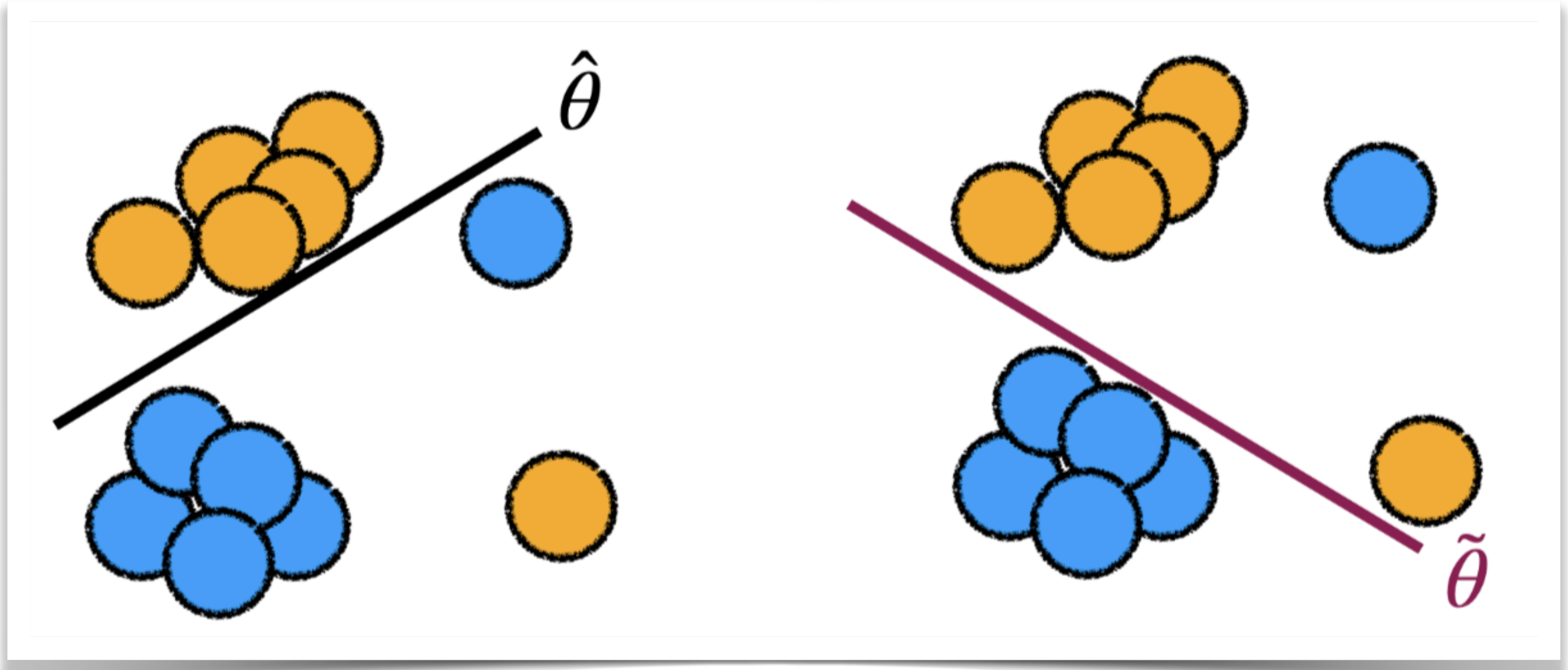
- **More implicit.** Goals are *inferred – which means can be wrong!*
- **Less impactful.** Goals are inferred from a set of smaller tweaks, e.g., you only label 100 examples in active earning!
- *Be cautious about making too trivial or counter-intuitive tweaks!*

Update Loss Function (aka model regularization)

Basically, change the way model is optimized, by adding constraints to the optimization objective.



Global: Explicitly add regularization to specifies model behavior,



Local: infer constraints from expert feedback on individual points(e.g. yellow is a more severe error)

Global loss func update: Unlikelihood training

Penalize undesirable generations (e.g. not following control, repeating previous context)

Prefix	*... starboard engines and was going to crash . " We 're going in ,"*
$\mathcal{L}_{\text{MLE}}$	he said . " We 're going to crash . We 're going to crash . We 're going to crash . We 're going to crash . We 're going to crash . We 're going to crash . We 're going to crash . We 're going to
$\mathcal{L}_{\text{UL-token+seq}}$	Hood said . " I 'm going to make sure we 're going to get back to the water . " The order to abandon ship was given by Admiral Beatty , who ordered the remaining two battlecruisers to turn away . At 18 : 25 , Hood turned his

General language model training objective

$$\mathcal{L}^t_{ULE} = \mathcal{L}^t_{MLE} + \alpha\mathcal{L}^t_{UL}$$

Another objective that lower the likelihood of undesired tokens **C**

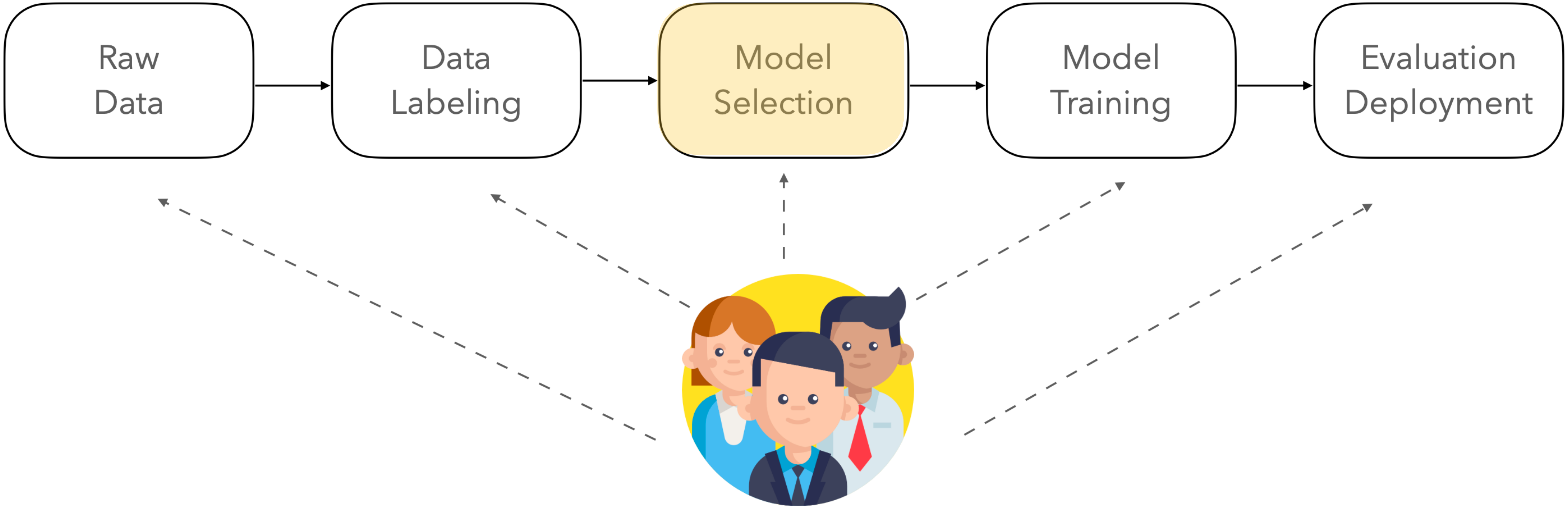
$$\mathcal{L}^t_{UL} = -\sum_{y_{neg} \in \mathcal{C}} \log(1 - P(y_{neg} \mid \{y^*\}_{<t}))$$

e.g. if C is previously seen text, then less repetition and more diversity

Welleck, Sean, et al. "Neural text generation with unlikelihood training." *ICLR* (2019).

We provide this form of feedback…

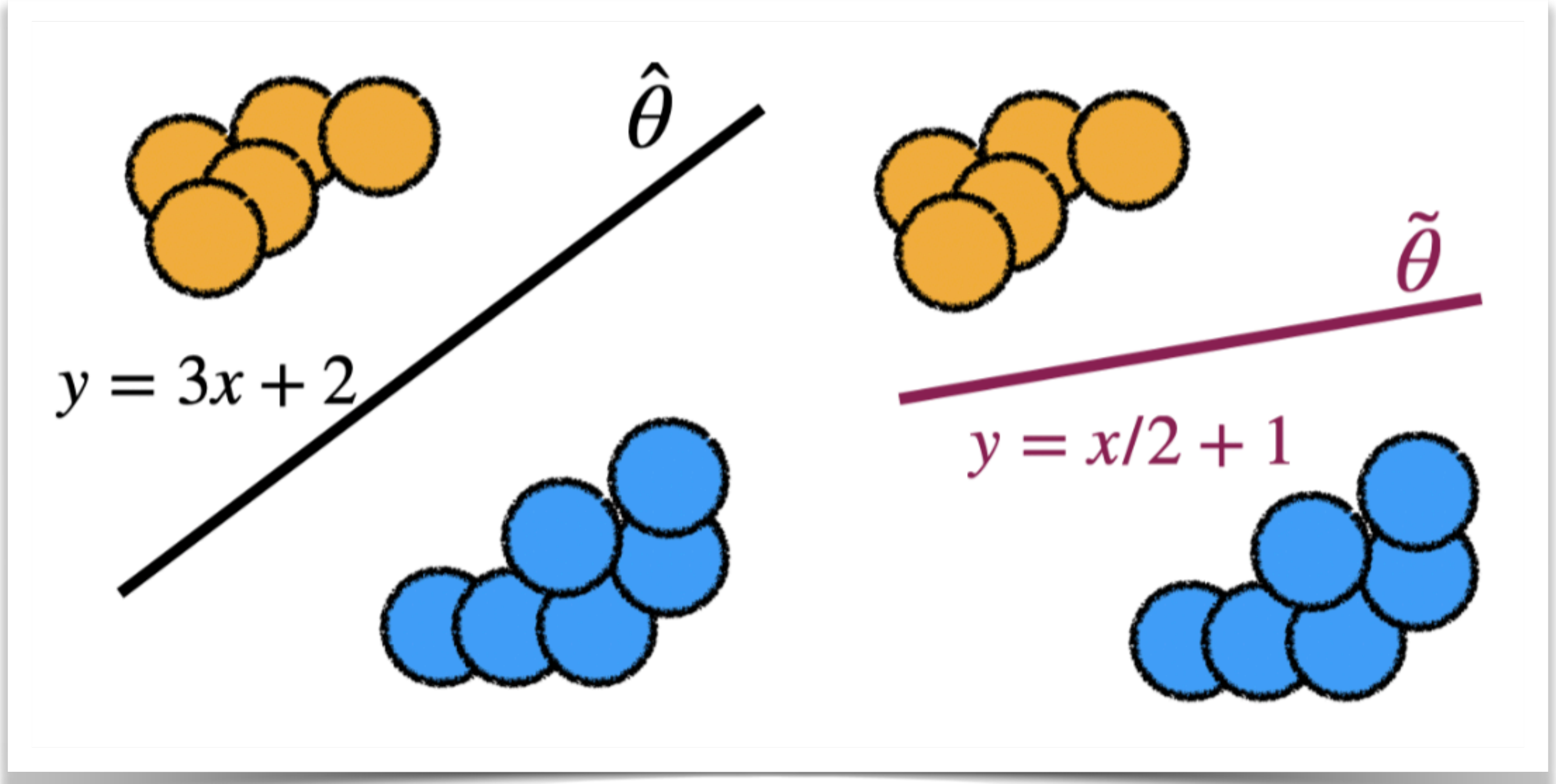
Mostly when we decide what model structure to use.



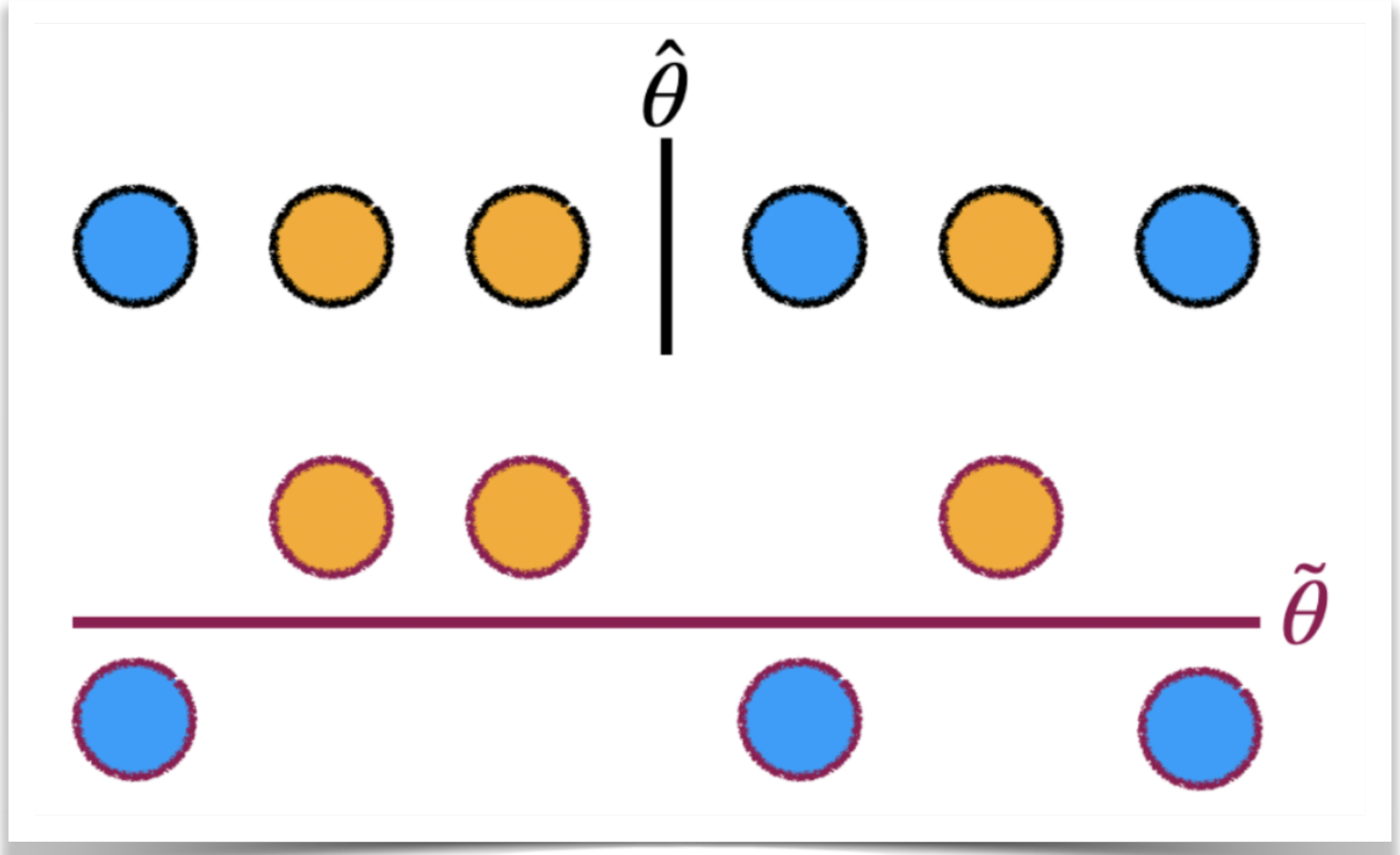
Wang, Zijie J., et al. "Putting humans in the natural language processing loop: A survey." *HCI+NLP Workshop* (2021).

Update Parameter Space (aka model editing)

Basically, directly change the parameters in the model so it uses the information in each data point differently from when it's unedited.



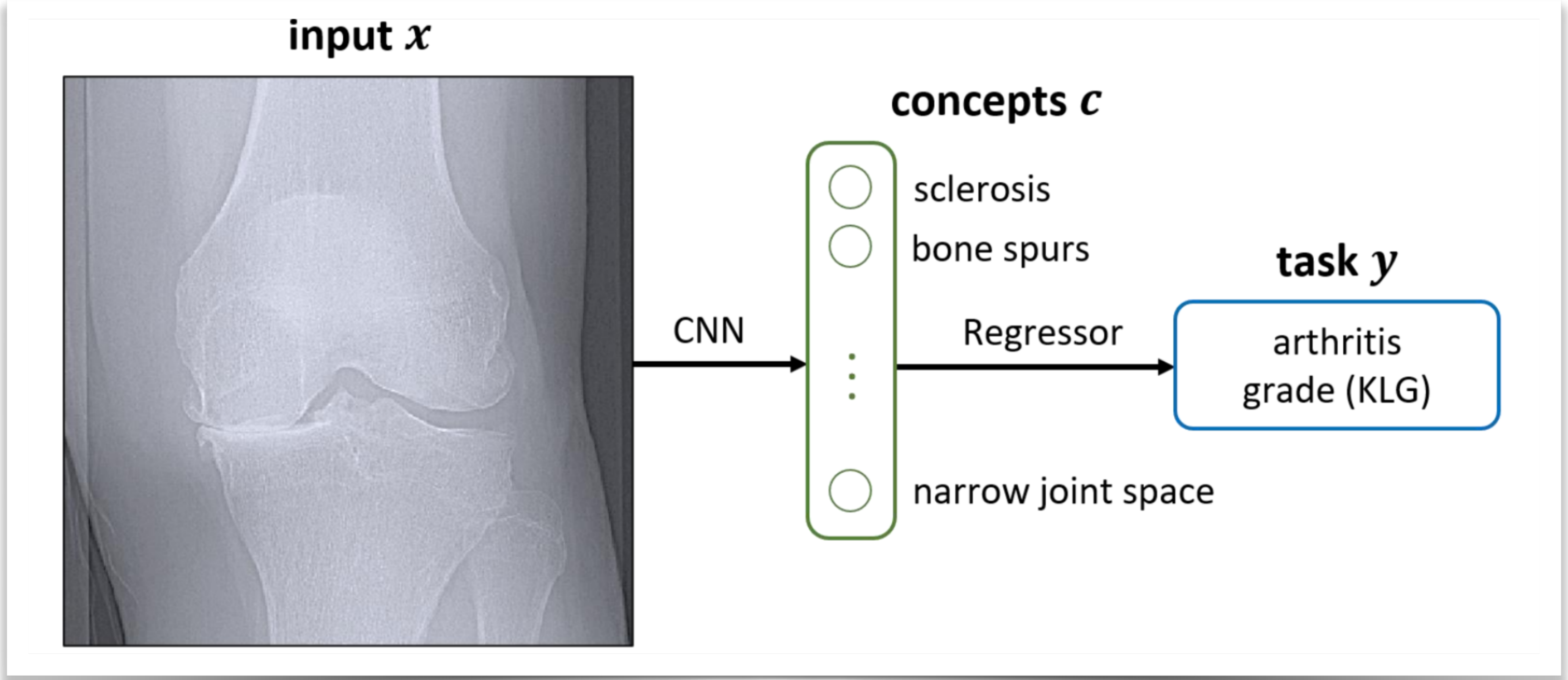
Global: Explicitly edit model parameters

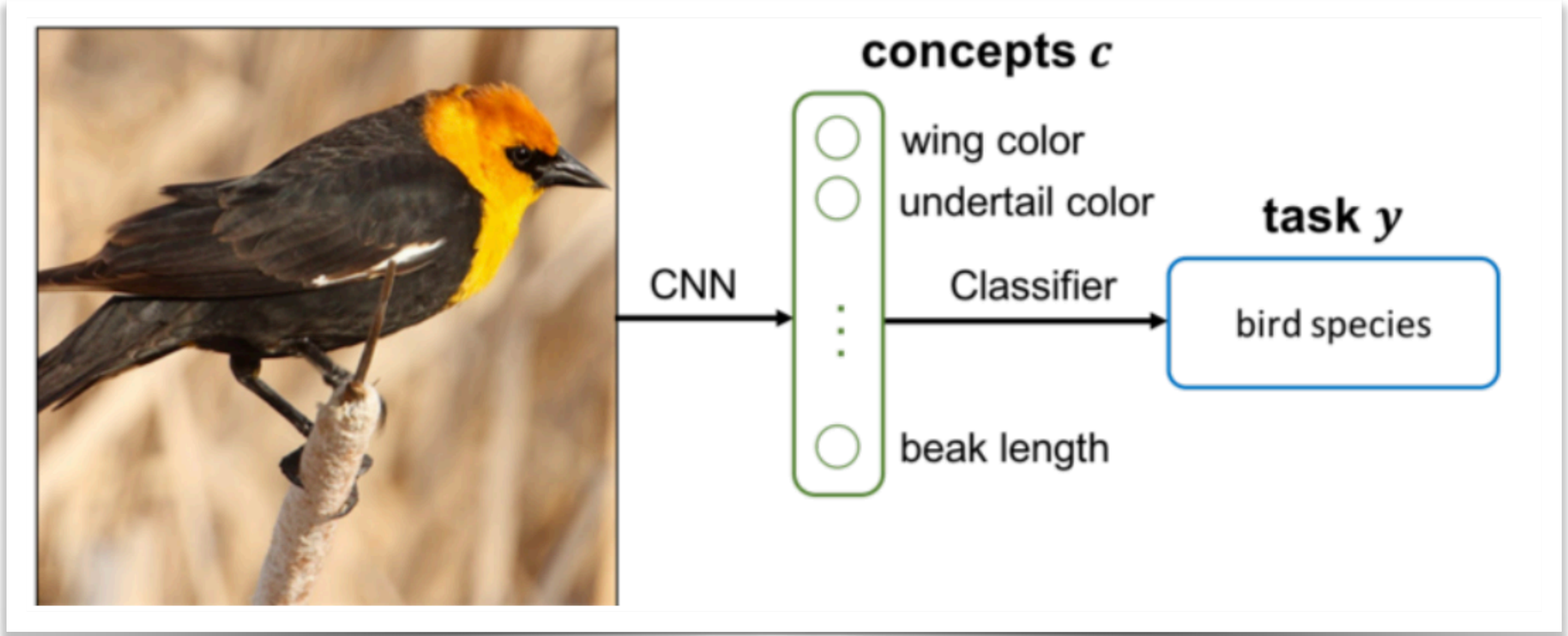


Local: change the feature space (then the weights of those features become 0)

Global Parameter Space update: Concept Bottleneck Model

Train model to explicitly use human-provided concepts.

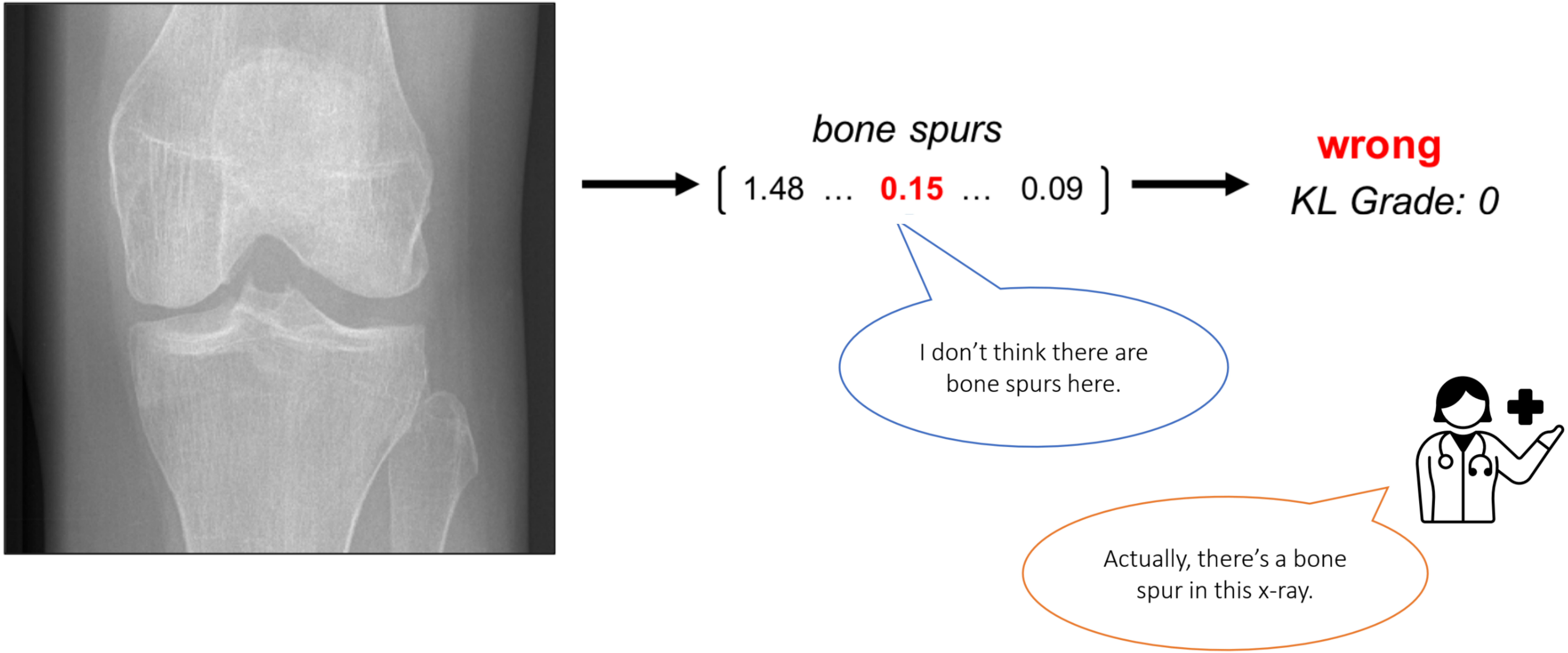




Koh, Pang Wei, et al. "Concept bottleneck models." *International Conference on Machine Learning*. PMLR, 2020.

Global Parameter Space update: Concept Bottleneck Model

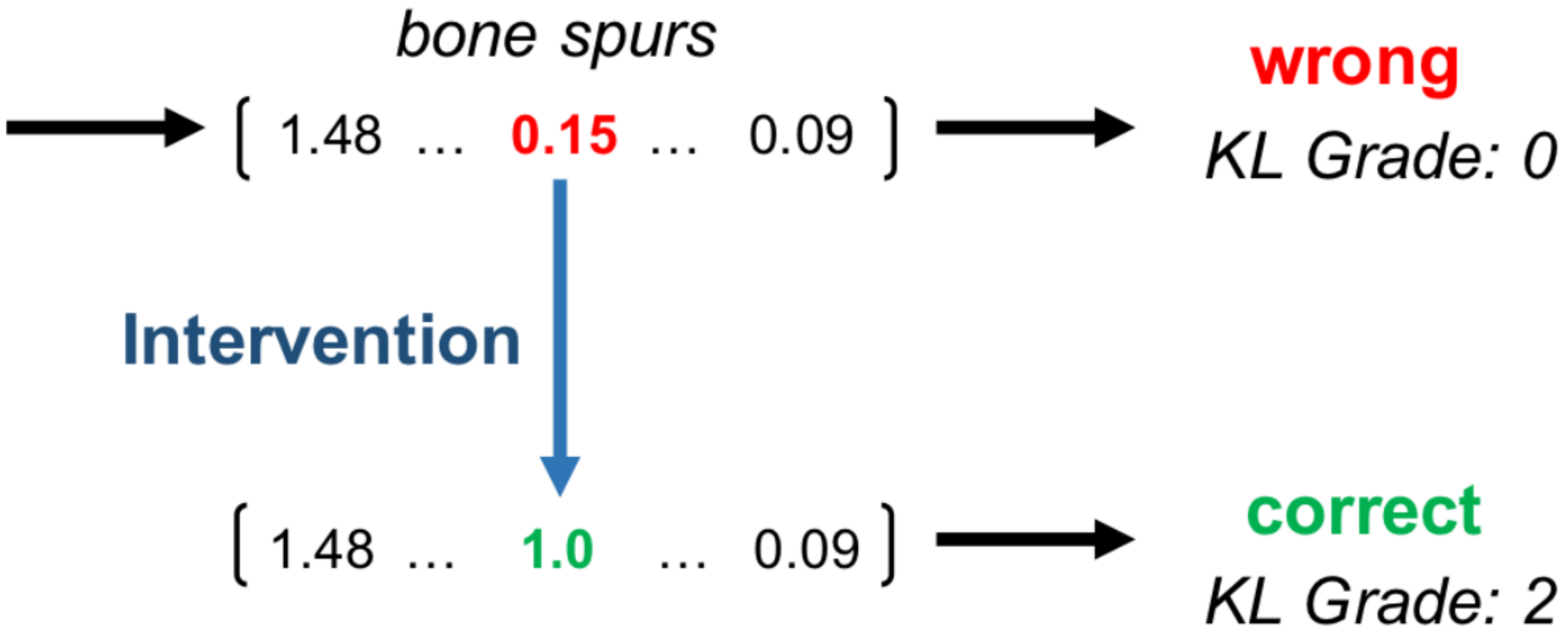
Concept bottlenecks enable interventions.



Koh, Pang Wei, et al. "Concept bottleneck models." *International Conference on Machine Learning*. PMLR, 2020.

Global Parameter Space update: Concept Bottleneck Model

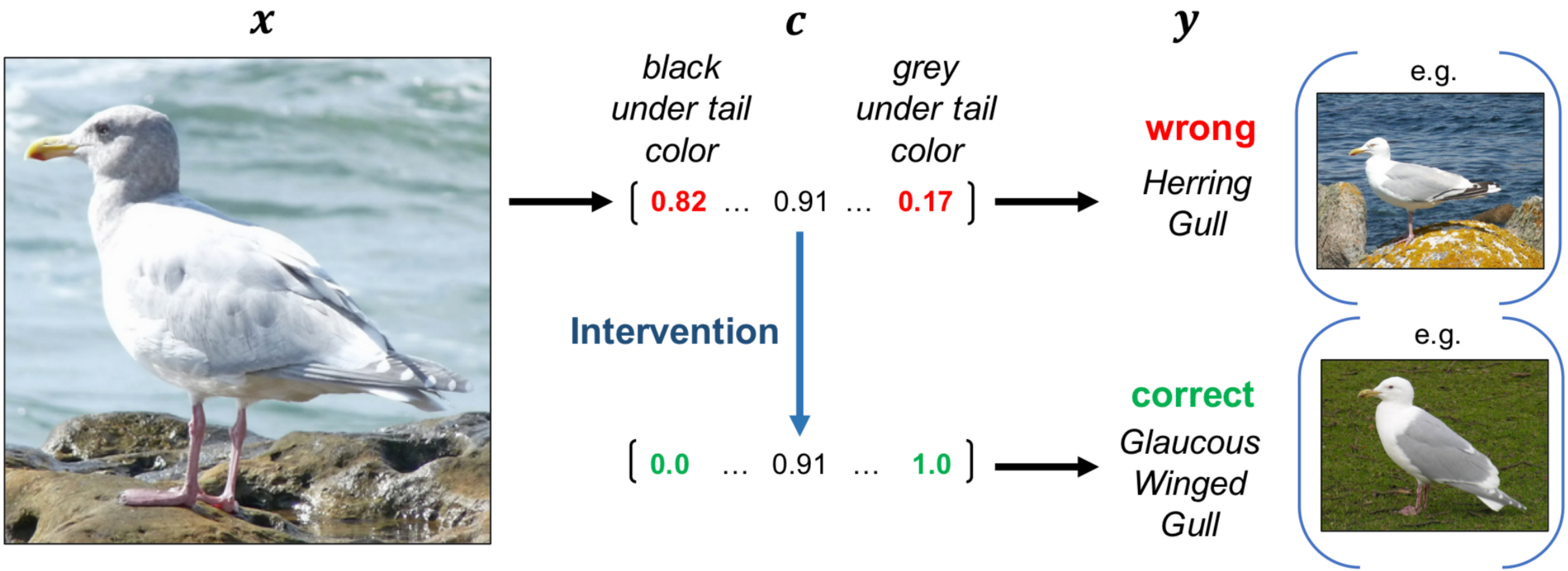
Concept bottlenecks enable interventions.



Koh, Pang Wei, et al. "Concept bottleneck models." *International Conference on Machine Learning*. PMLR, 2020.

Global Parameter Space update: Concept Bottleneck Model

Concept bottlenecks enable interventions.



Koh, Pang Wei, et al. "Concept bottleneck models." *International Conference on Machine Learning*. PMLR, 2020.

Local Parameter Space Update: feature engineering/patching

Dynamically fix model bugs by specifying feature/label space using natural language patches.

	Original Model	Regex patching	few-shot finetuning	language patching
2 stars, but our waitress Wendy was really nice	✗	✓	✓	✓
Two stars for the place, but the ambience is great	✗	✗	✓	✓
The restaurant was noisy, but tacos were bomb	✗	✓	✓	✓
The authorities found a bomb in the restaurant	✓	✗	✗	✓

Regex Patch

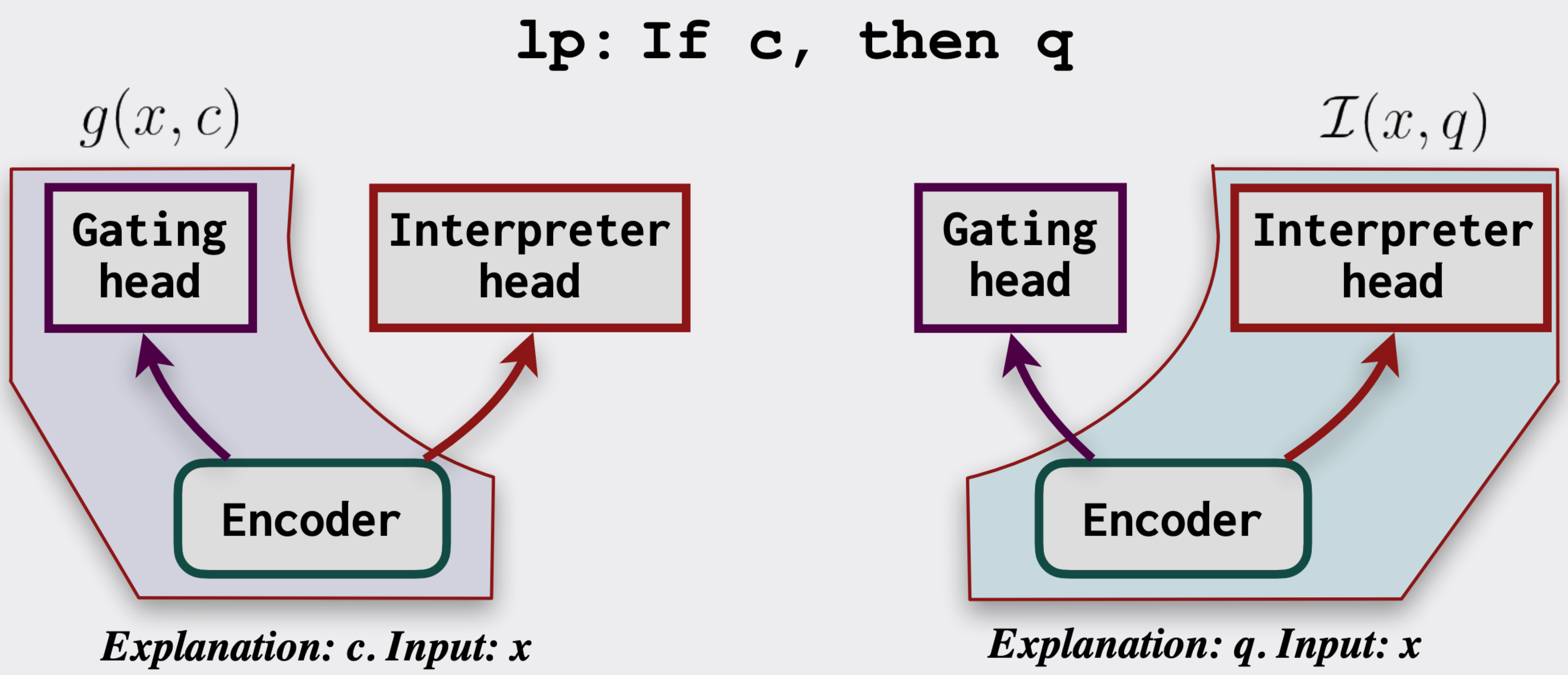
```
def patch_1(x):
  if '2 star' in x:
    return negative
  else:
    return model(x)
def patch_2(x):
  if ' bomb ' in x:
    x = x.replace('bomb', 'good')
  return model(x)
```

Language Patch

```
If food is described as
bomb, then food is good

If review gives 2 stars,
then label is negative

If review gives 4 or 5
stars, then label is
positive
```



$$\mathtt{Fix}(f, x, \mathrm{lp}) = g(x, c) \cdot \mathcal{I}(x, q) + [1 - g(x, c)] \cdot f(x)$$

does patch apply?

incorporating patch info

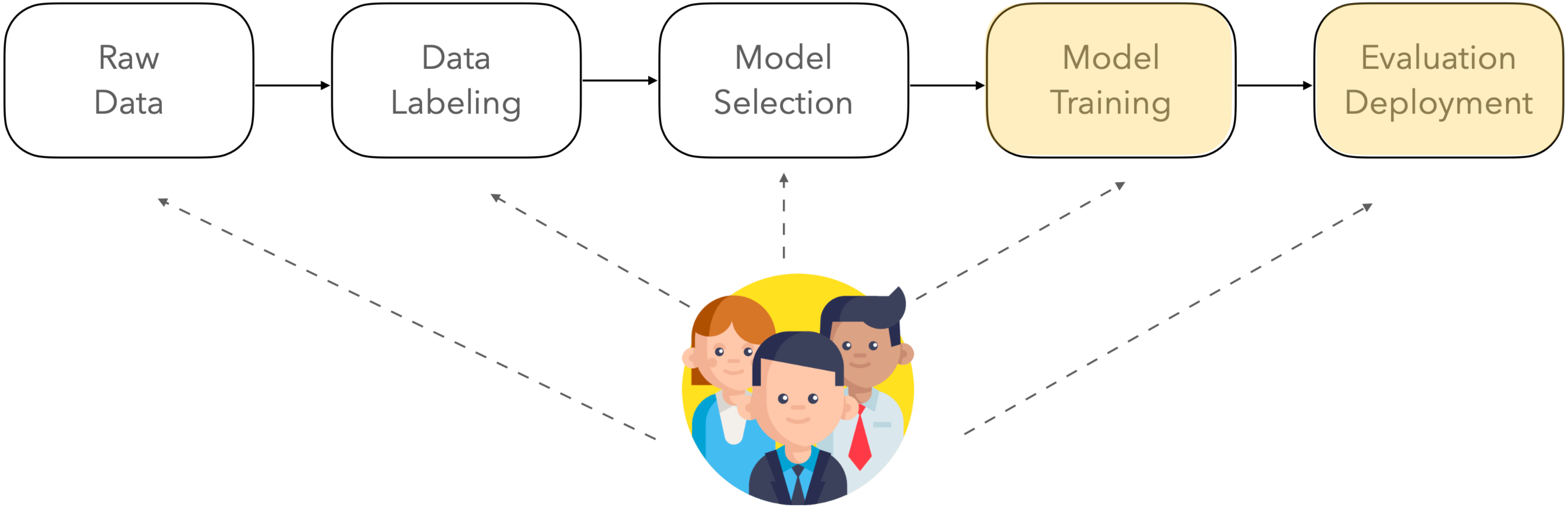
original output

Note that this is a more explicit form of local feedback!

Murty, Shikhar, et al. "Fixing model bugs with natural language patches." *EMNLP 2022.*

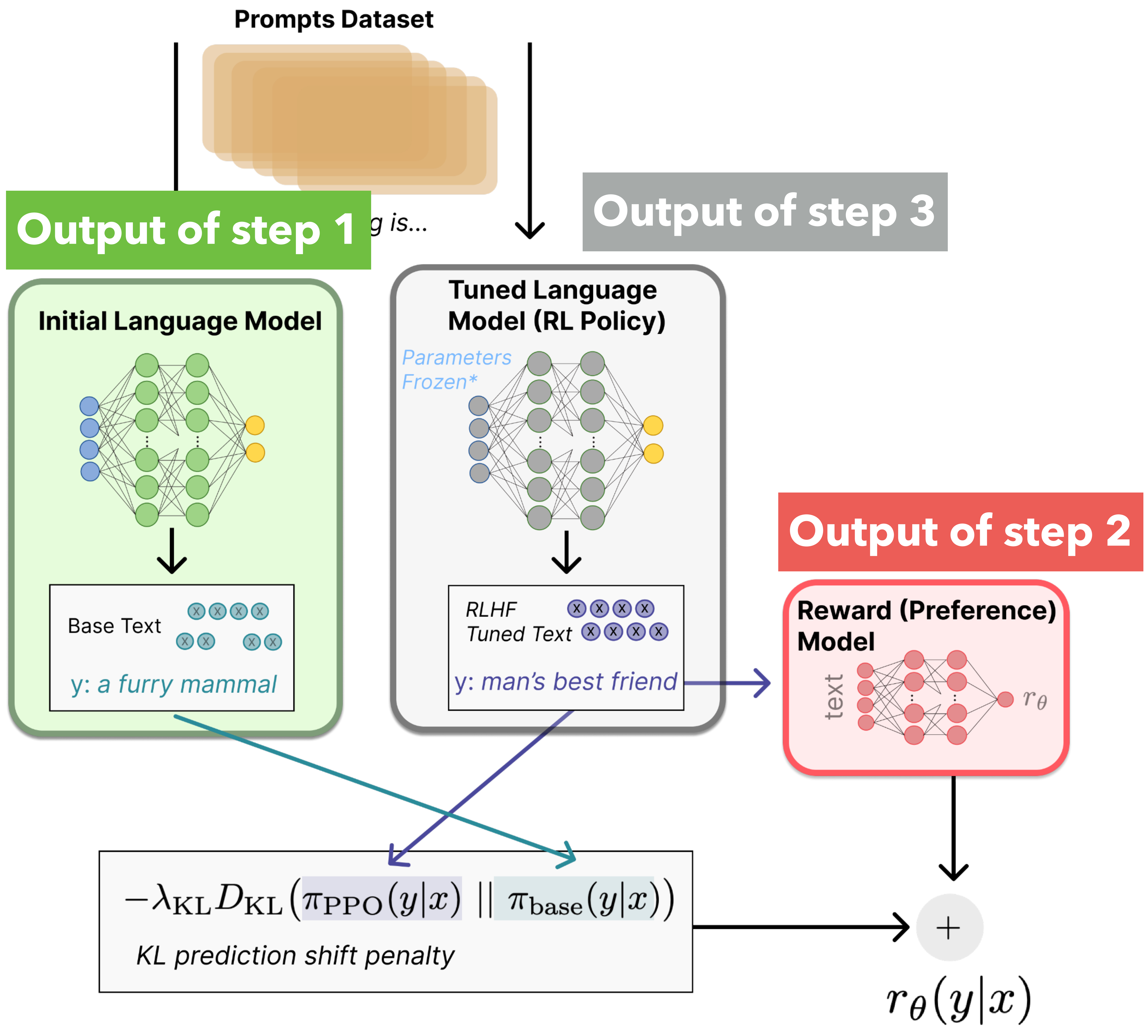
We provide this form of feedback…

During and after the model is trained.



Wang, Zijie J., et al. "Putting humans in the natural language processing loop: A survey." *HCI+NLP Workshop* (2021).

Reinforcement Learning from Human Feedback



Nathan Lambert: Intro to Reinforcement Learning from Human Feedback

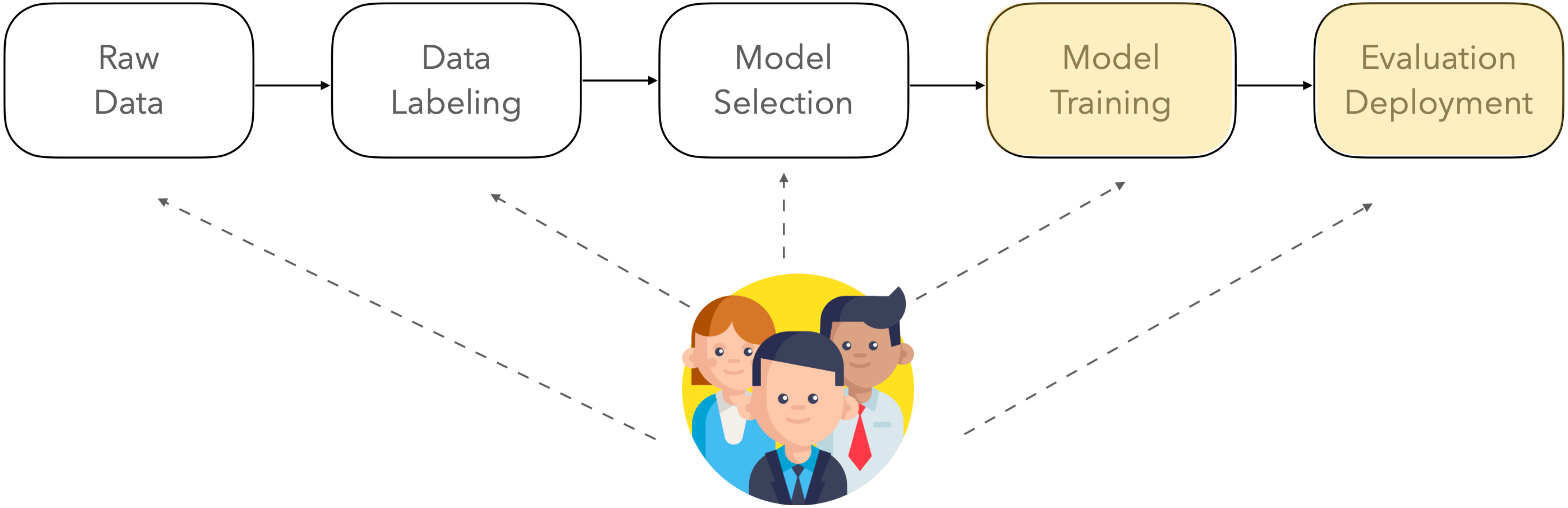
Feedback-Update Taxonomy

	Dataset Update	Loss Function Update	Parameter Space Update
Domain	**Dataset modification** Augmentation Preprocessing **Data generation from constraint** Fairness, weak supervision **Use unlabeled data** **Check synthetic data**	**Constraint specification** Fairness, Interpretability **Resource constraints**	**Model editing** Rules, Weights **Model selection** Prior update, Complexity
Observation	**Active data collection** Add data, Relabel data, Reweight data, collect expert labels **Passive observation**	**Constraint elicitation** Metric learning, Human representations **Collecting contextual information** Generative factors, concept representations, Feature attributions	**Feature modification** Add/remove features, Engineering features

Chen, Valerie, et al. "Perspectives on Incorporating Expert Feedback into Model Updates." *ArXiv* (2022).

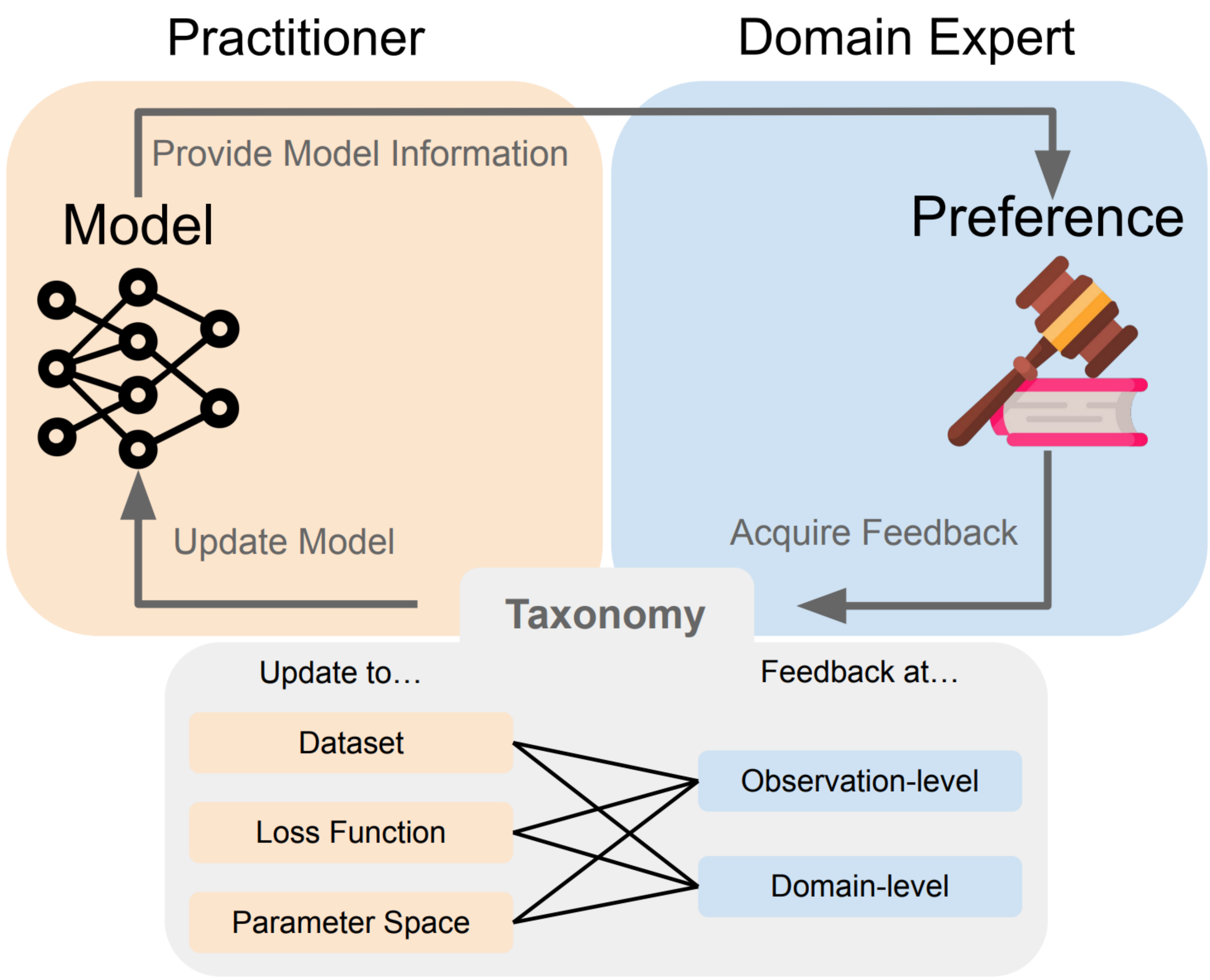
We provide this form of feedback…

During and after the model is trained.



Wang, Zijie J., et al. "Putting humans in the natural language processing loop: A survey." *HCI+NLP Workshop* (2021).

Keys of Human-in-the-loop NLP



*Allow humans to **easily provide feedback**.*

Turn nontechnical, human preferences into usable model updates.

*Build models to **effectively take the feedback**.*

Chen, Valerie, et al. "Perspectives on Incorporating Expert Feedback into Model Updates." *ArXiv* (2022).

What are some forms of feedback?

Label additional data points.

Edit data points.

Change data weights.

Binary/Scaled user feedback.

Natural language feedback.

Code language feedback.

Define, add, remove feature spaces.

Directly change the objective function.

Directly change the model parameter.

…

Which kinds of feedback do you prefer to provide?

Label additional data points.

Edit data points.

Change data weights.

Binary/Scaled user feedback.

Natural language feedback.

Code language feedback.

Define, add, remove feature spaces.

Directly change the objective function.

Directly change the model parameter.

…

Trade-offs: Human-friendly vs. Model friendly

Models need feedback that they can respond to.
Update objective function is more effective.
Labeling is not as much unless large scale.

Humans prefer easier-to-provide feedback,
non-experts maybe:
NL feedback > labeling > model manipulation
Experts maybe the reverse:
Because they know more about feedback effectiveness and reliable-ness.

Label additional data points.

Edit data points.

Change data weights.

Binary/Scaled user feedback.

Natural language feedback.

Code language feedback.

Define, add, remove feature spaces.

Directly change the objective function.

Directly change the model parameter.

…

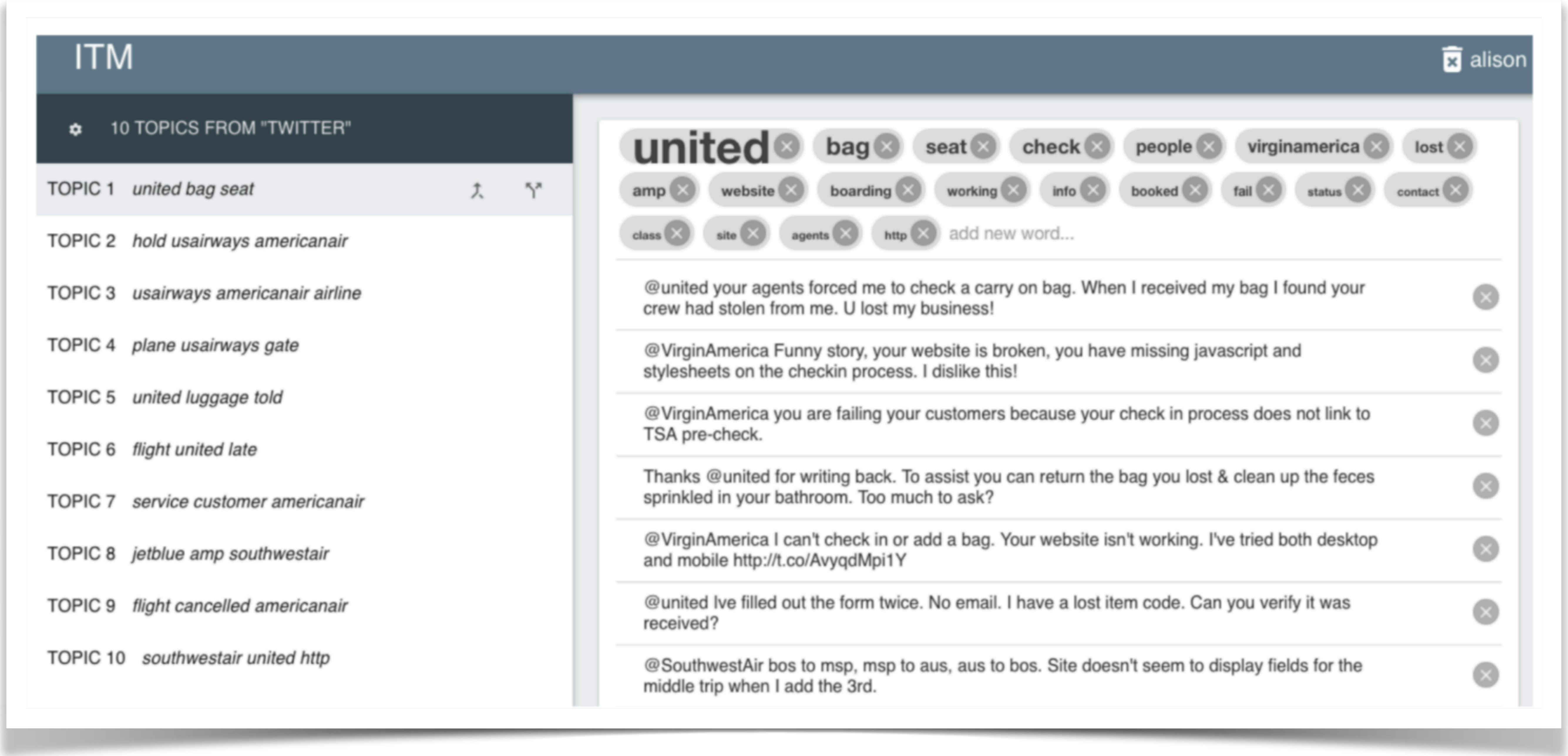
Interaction Medium

Graphical user interface:

Graphic icons, visual indicators

Visualize the blackbox NLP model

Provide users more accurate control



Natural language interface:

Users interact via natural language

Explicit feedback or implicit ones

Intuitive as it simulates a conversation

Hu, Yuening, et al. "Interactive topic modeling." *Machine learning* 95 (2014): 423-469.

Hancock, Braden, et al. "Learning from dialogue after deployment: Feed yourself, chatbot!." *arXiv preprint arXiv:1901.05415* (2019).

What are some challenges in HITL NLP?

Humans can only provide limited amount of feedback.

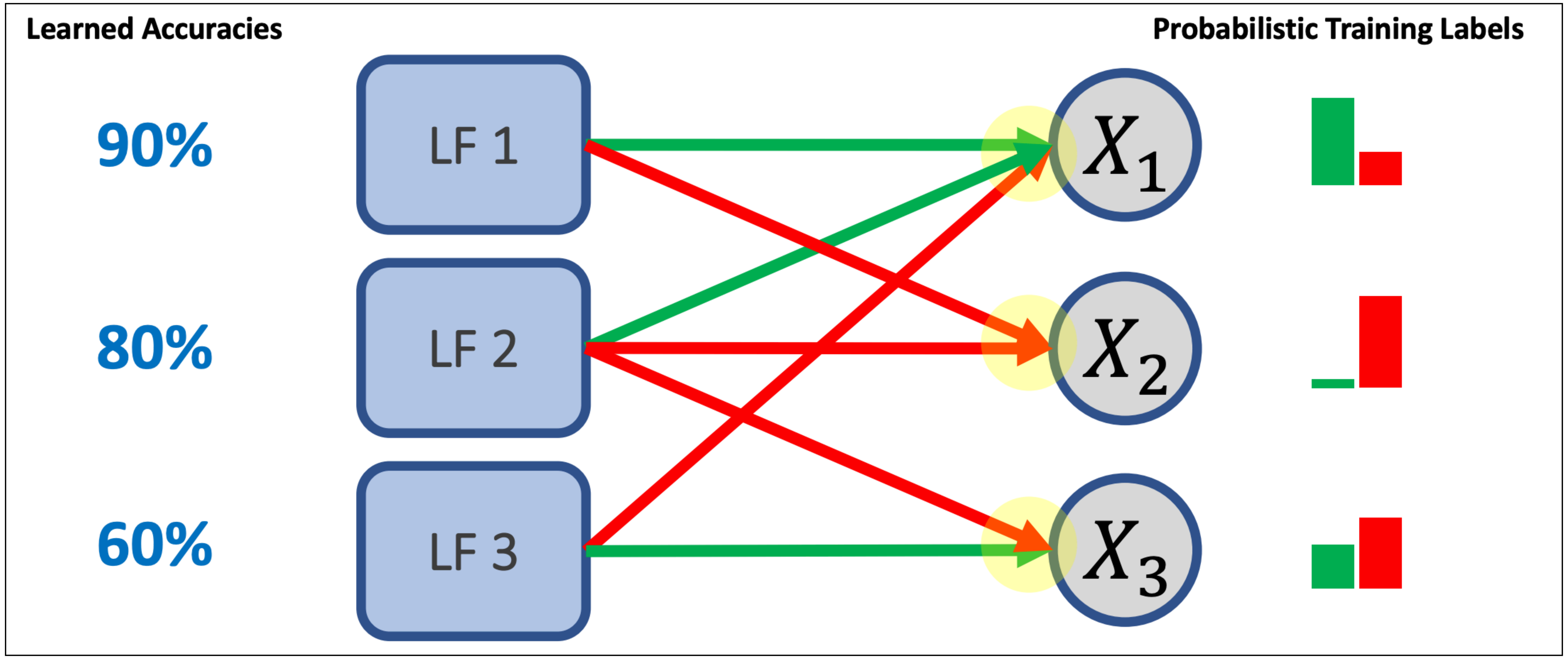
Need to avoid cognitive overload

This is also why sometimes we may prefer local feedback, because global feedback would require a high-level understanding on the task/model which is harder to get.

What are some challenges in HITL NLP?

Humans are not oracle, and make mistakes.

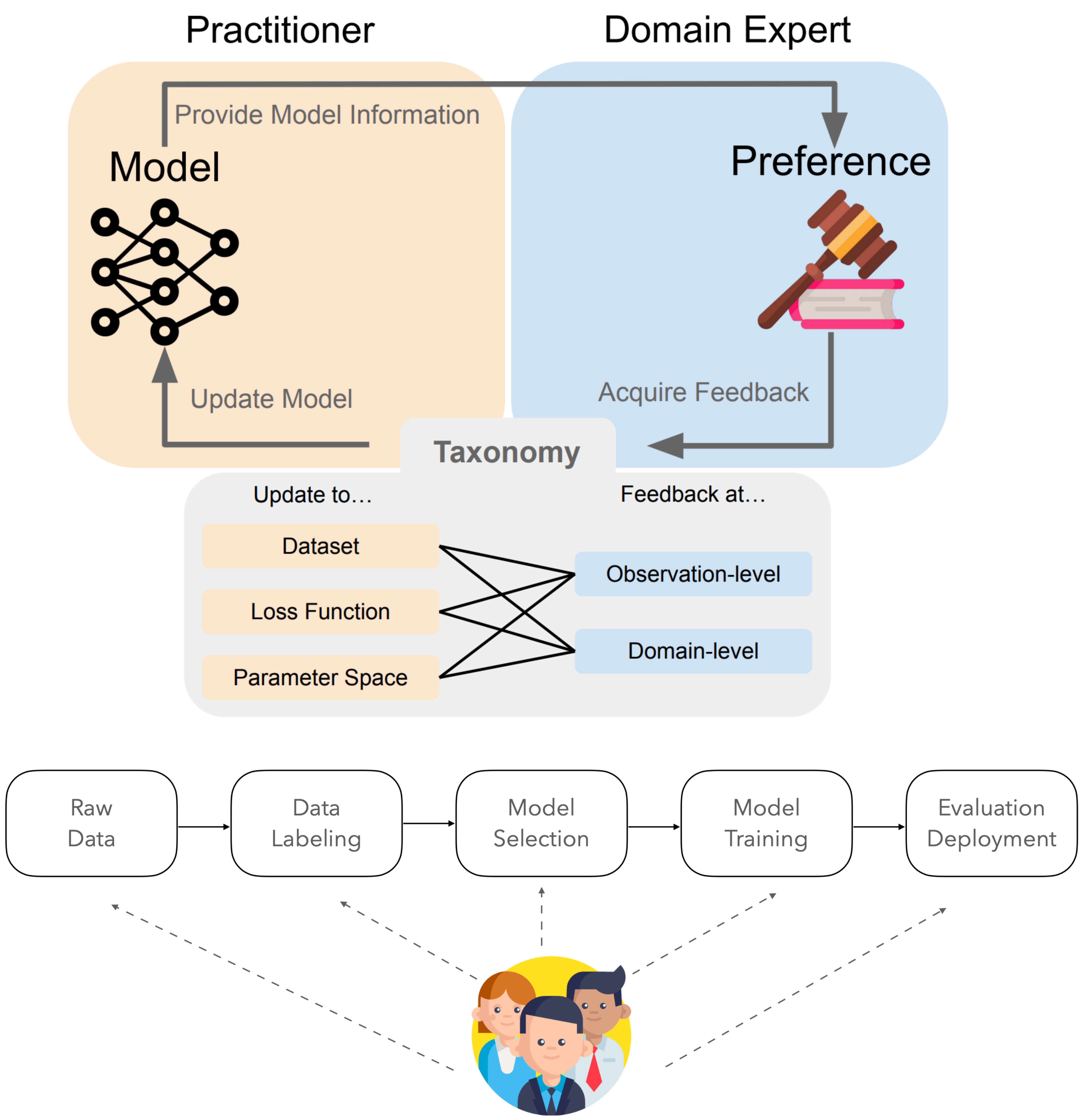
Need to deal with noisy inputs, like what Snorkel is doing.



Other Open Thoughts

- As human feedback can be subjective, who should HITL systems collect feedback from? Is there any expertise levels required to perform the task?
- How to present what the model has learned and what feedback is need? How to visualize the model change after learning from user feedback?
- How to dynamically choose the most helpful feedback to collect? How to guide users to provide useful feedback?
- How to evaluate the collected human feedback as it can be noisy and even misleading?
- How to open-source tools and share user study protocols when publishing new HITL NLP work?

Exercise: Let’s build a better email assistant



Let’s divide into two groups:

HCI: share human insights

NLP: pick which solution to use