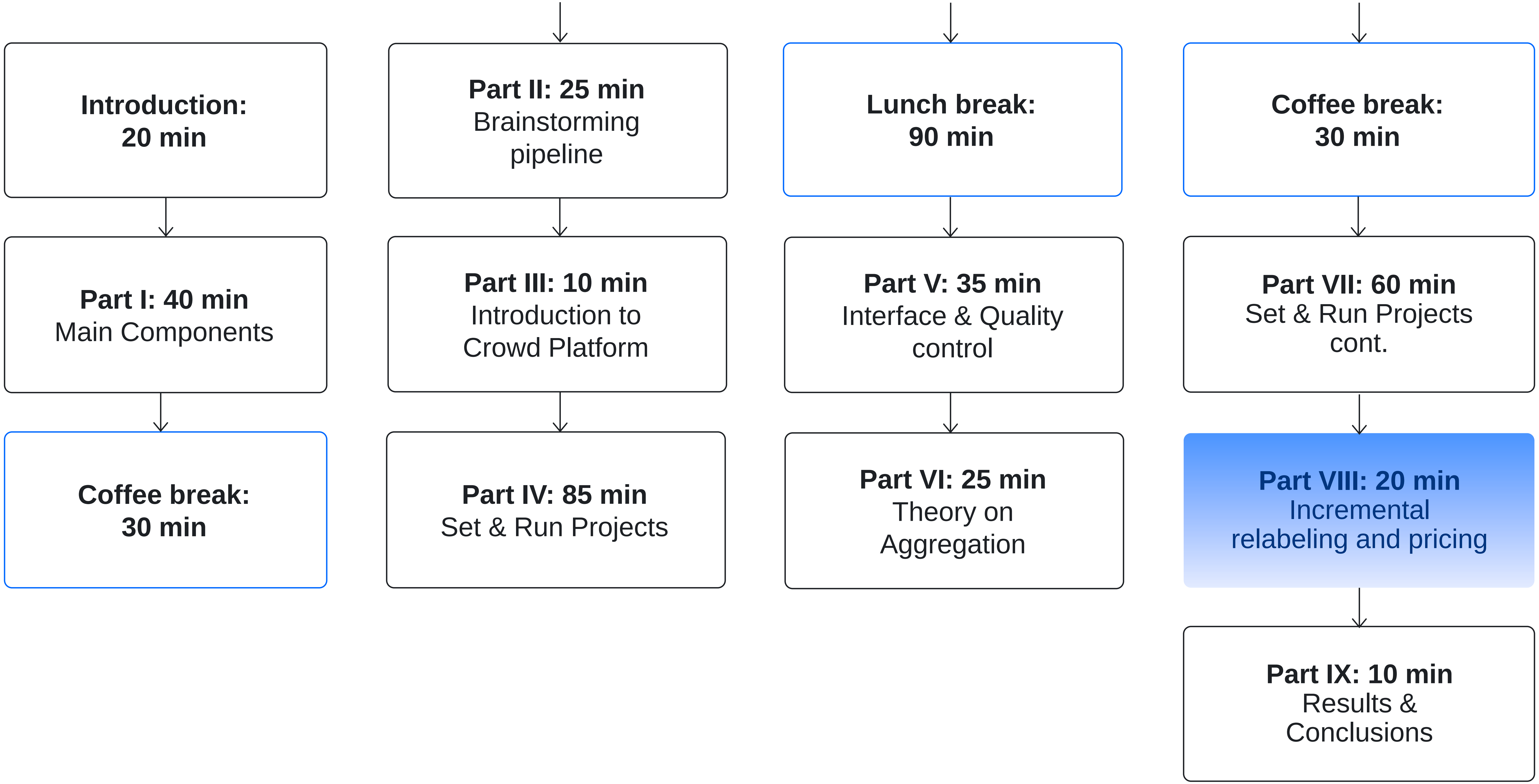
Part VIII

Theory on incremental relabelling and pricing

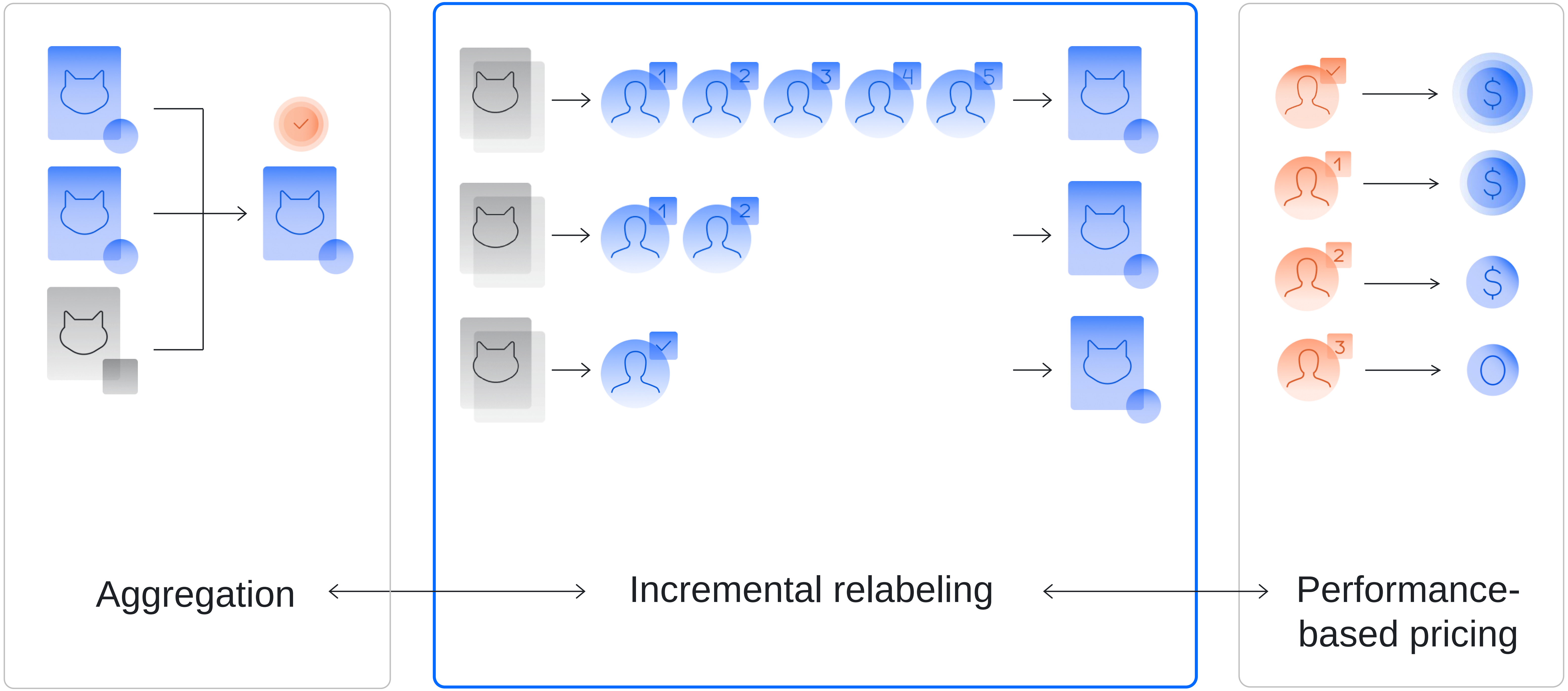
Valentina Fedorova,
Research analyst

Toloka

Tutorial schedule



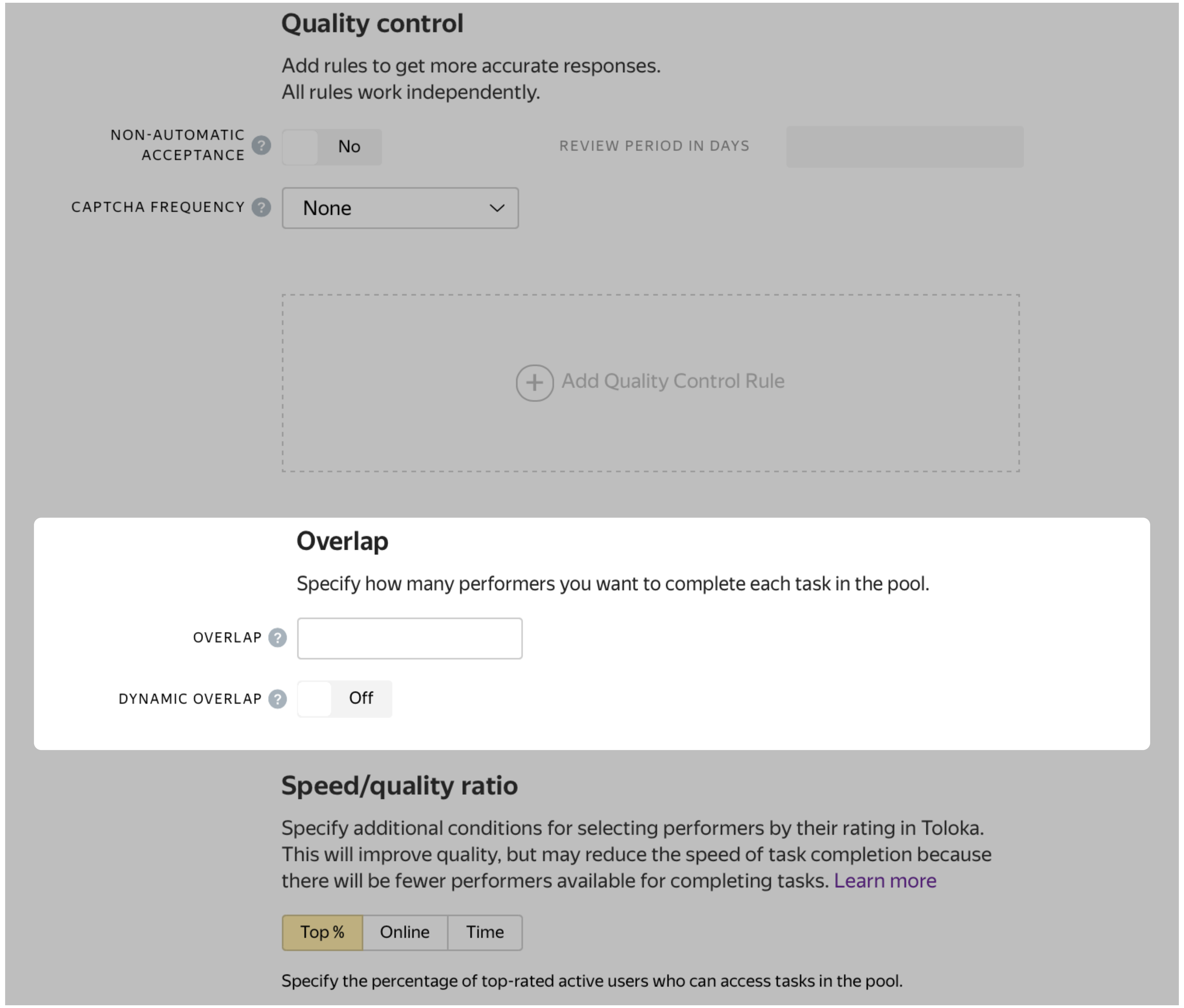
Key components of labeling with crowds



Incremental relabeling

aka dynamic overlap

Pool settings: dynamic overlap



Incremental relabeling problem

Obtain aggregated labels of a desired quality level using a fewer number of noisy labels

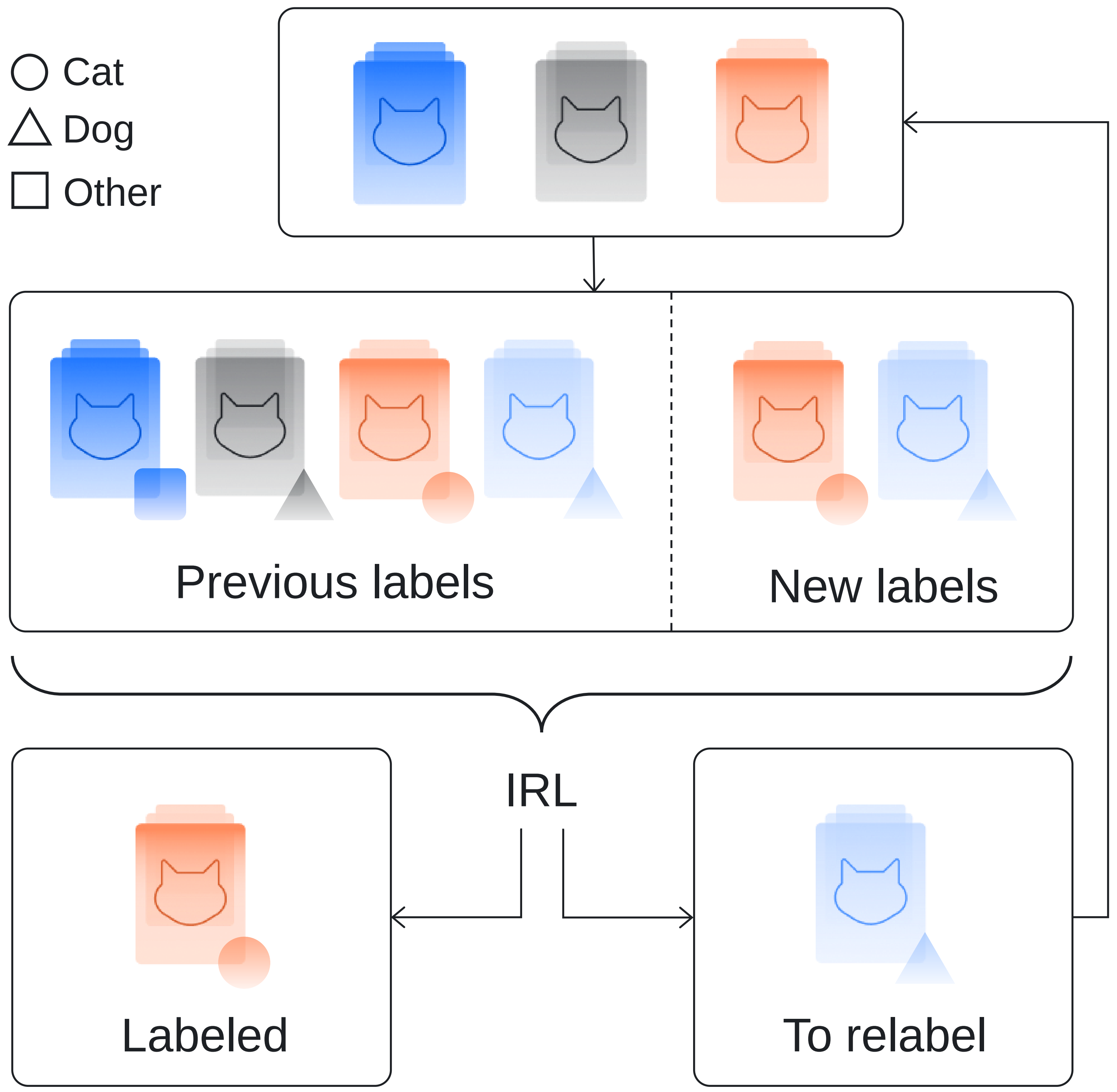
Incremental relabeling scheme (IRL)

Request 1 label for each object

In real time IRL algorithm receives:
(1) previously accumulated labels
(2) new labels

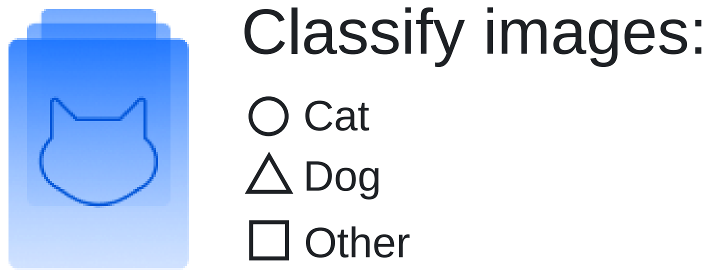
Decides:
(1) which objects are labeled
(2) which objects to relabel

Repeat until all tasks are labeled

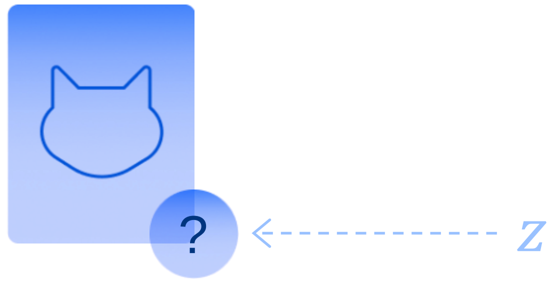


Notations

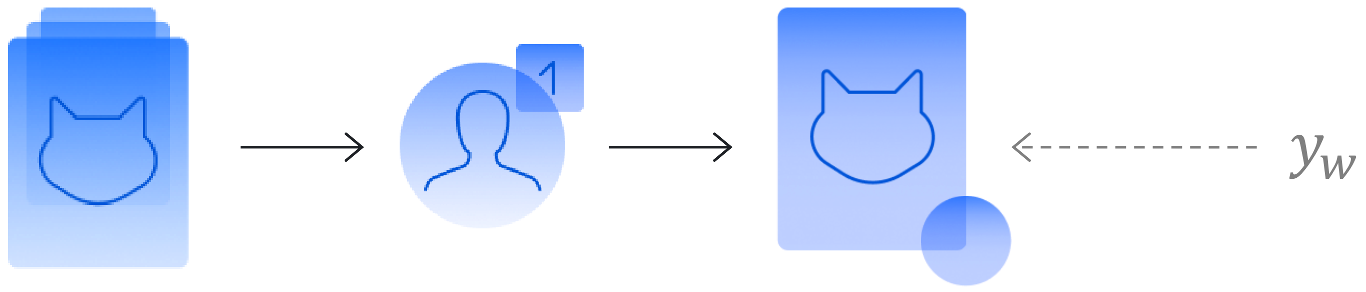
- Consider one object



- $z \in \{1, \dots, K\}$ — latent true label



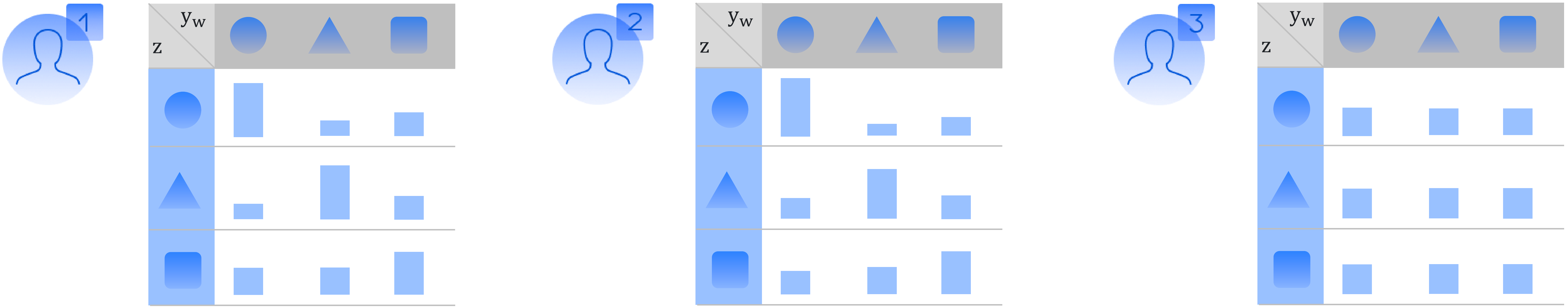
- $y_w \in \{1, \dots, K\}$ — observed noisy label from worker w:



Notations

- Noisy label model for worker w:

$$M_w \in [0,1]^{K\times K}\text{: } \Pr(Y_w = k | Z = c) = M_w[c, k]$$



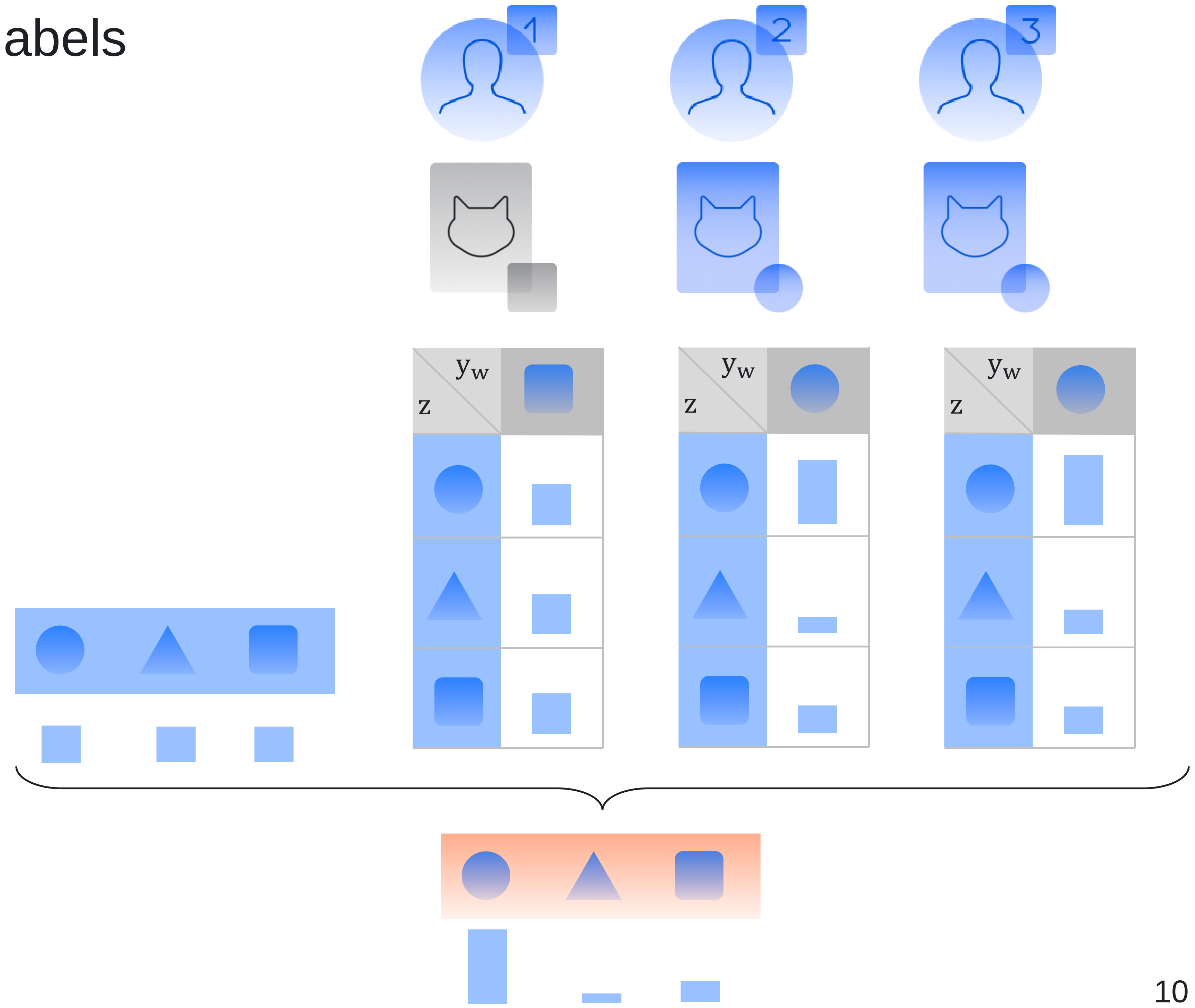
- Prior distribution: $\Pr(Z = k) = p_k$

Posterior distribution

- $\{y_{w_1}, \dots, y_{w_n}\}$ — accumulated noisy labels for the object

- Using Bayes rule:

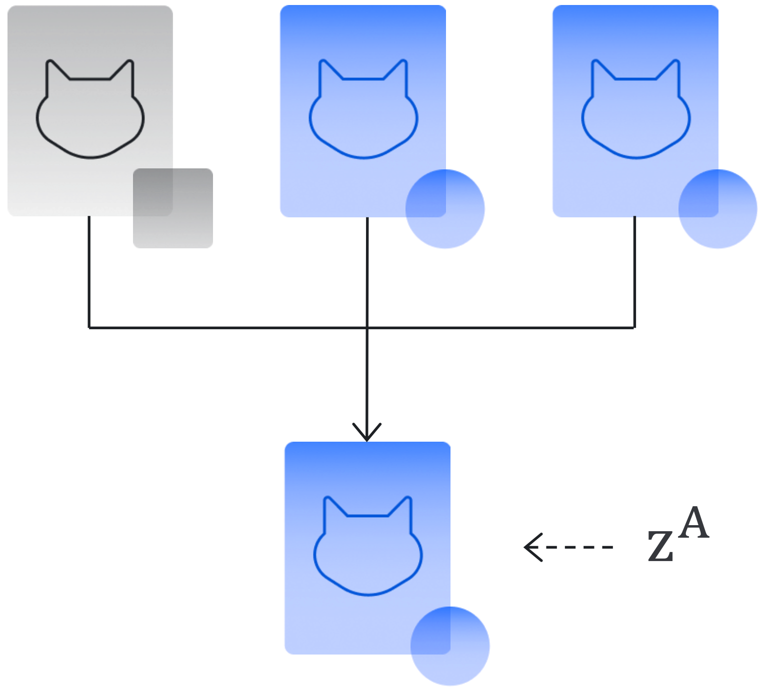
$$\Pr\left(Z = k \middle| \{y_{w_1}, \dots, y_{w_n}\}\right) = \frac{\Pr(Z = k)\Pr\left(\{y_{w_1}, \dots, y_{w_n}\} \middle| Z = k\right)}{\Pr\left(\{y_{w_1}, \dots, y_{w_n}\}\right)} = \frac{p_k \prod_{i=1}^{n} M_{w_i}\left[k, y_{w_i}\right]}{\sum_{t=1}^{K} p_t \prod_{i=1}^{n} M_{w_i}\left[t, y_{w_i}\right]}$$



Expected accuracy of aggregated labels

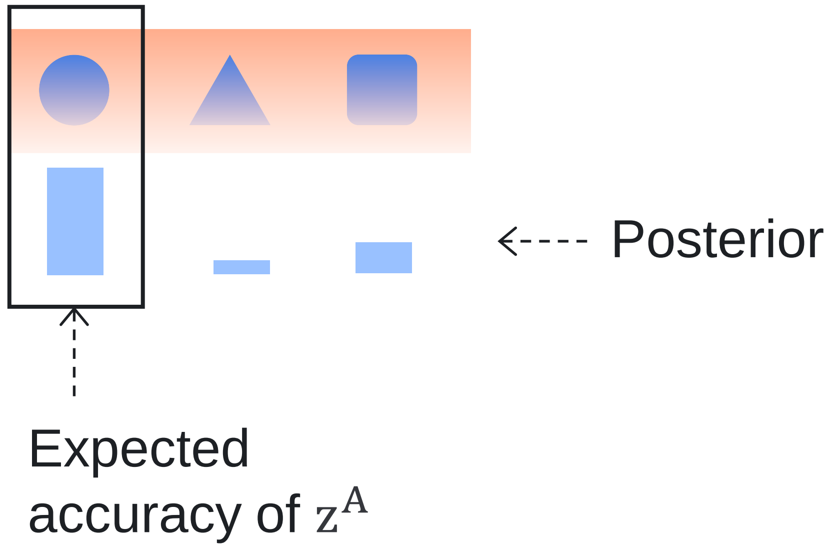
- Let A be an aggregation model, e.g. MV, DS, GLAD,…
- Denote aggregated label $z^A = A(\{y_{w_1}, \dots, y_{w_n}\})$
- Expected accuracy of aggregated labels given noisy labels is

$$E\left(\delta(z = z^A) \middle| \{y_{w_1}, \dots, y_{w_n}\}\right) = \Pr\left(z = z^A \middle| \{y_{w_1}, \dots, y_{w_n}\}\right)$$



- Stop labeling if $E\left(\delta(z = z^A) \middle| \{y_{w_1}, \dots, y_{w_n}\}\right) \geq C$

Parameter



Sheng VS, Provost F, Ipeirotis PG. Get another label? improving data quality and data mining using multiple, noisy labelers. KDD 2008

Incremental relabeling algorithm

Input: $\bigcup_{t=1}^{T-1} Y^t$ — previous labels till step T

Y^T — new labels

Output: R — objects to relabel

For each object j with a label in Y^T: ← Object with a new label

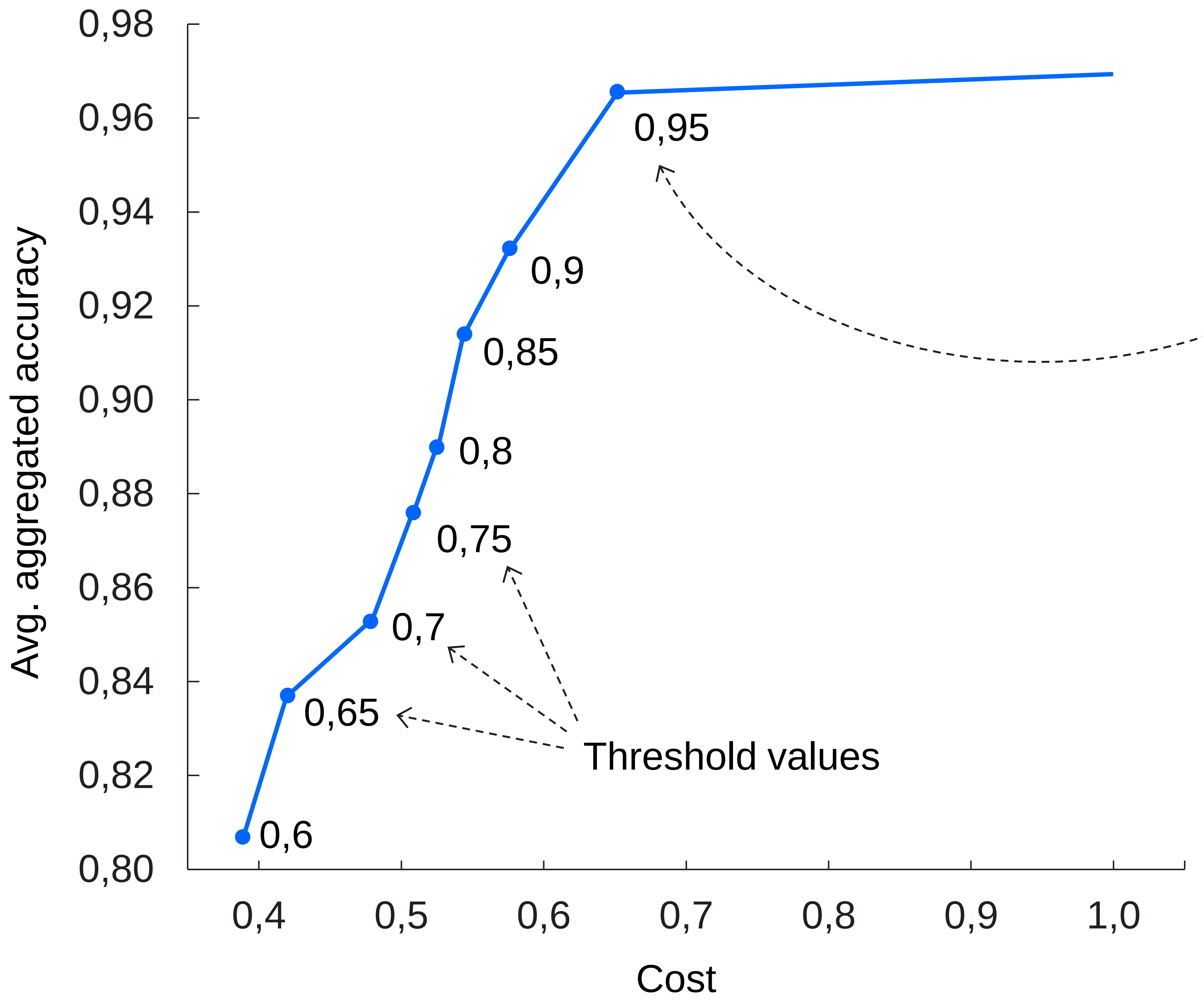
$z_j^M = M(\bigcup_{t=1}^{T} Y^t)$ ← Current aggregated label

$c_j = E(z_j = z_j^M | \bigcup_{t=1}^{T} Y^t)$ ← Expected accuracy for the current aggregated label

If $c_j < c$, then $R = R \cup j$

Parameter: c — threshold for expected accuracy

Threshold in IRL: cost – accuracy trade-off



- Optimal threshold $c = 0.95$
- A higher c does not increase accuracy
- Saving ≈ 35% of noisy labels

How to obtain a cost-accuracy plot

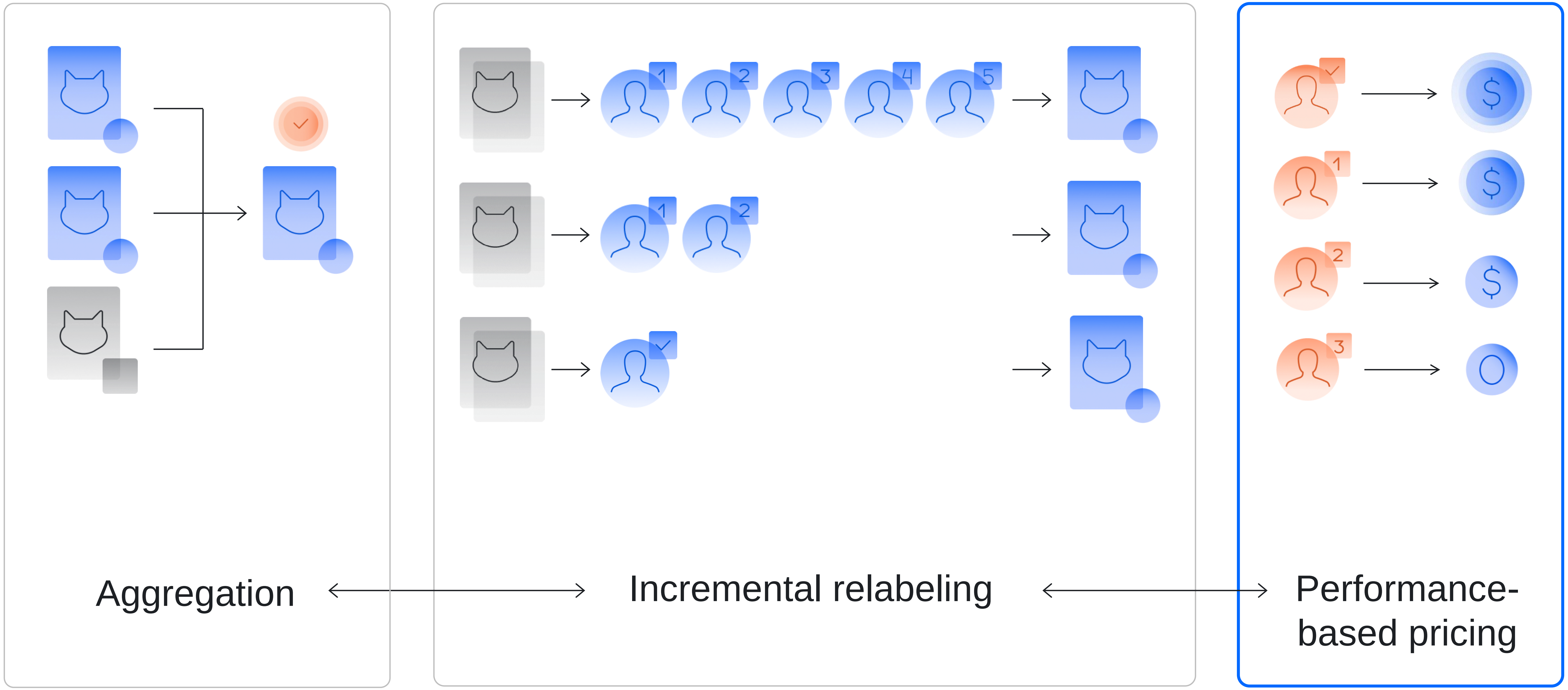
Data for the plot:

- Label a pool of objects with a redundant overlap (e.g., 10)
- Obtain ground truth labels for the objects (e.g., expert labels or MV labels)

Simulate IRL with different thresholds using the data:

- For each threshold c from a grid $0 < c_0 < \ldots < c_m \leq 1$
- Repeat N times:
  1. Shuffle noisy labels and fix the order of labels
  2. Draw labels sequentially and test the IRL condition after each label
  3. Once the IRL condition for an object is met, discard unused labels for the object
  4. When all objects are labelled calculate
     - accuracy of aggregated labels
     - cost as the fraction of used noisy labels
- Average N values of aggregated accuracy and N values of cost for each value of threshold c

Key components of labeling with crowds

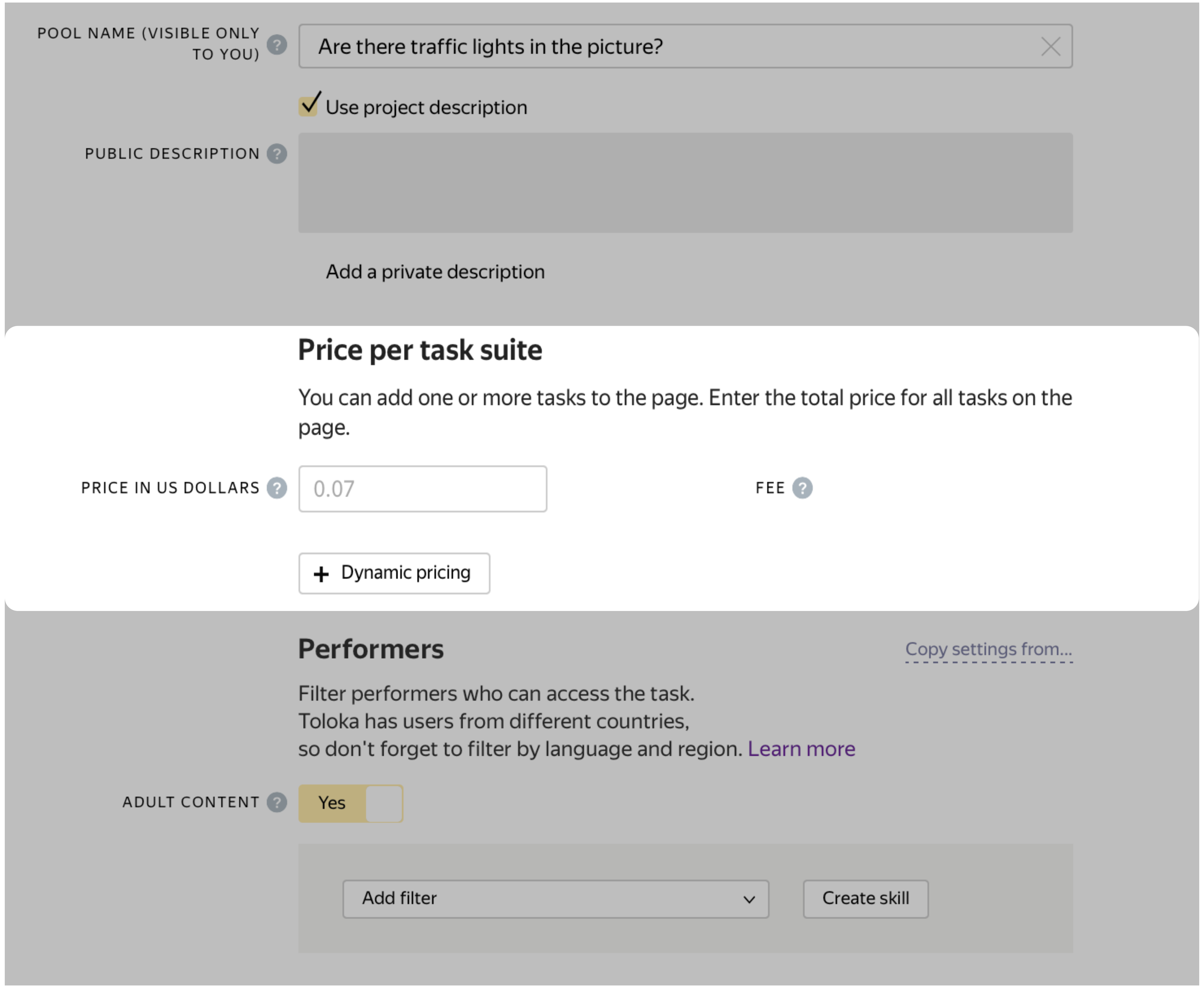


Aggregation ⟷ Incremental relabeling ⟷ Performance-based pricing

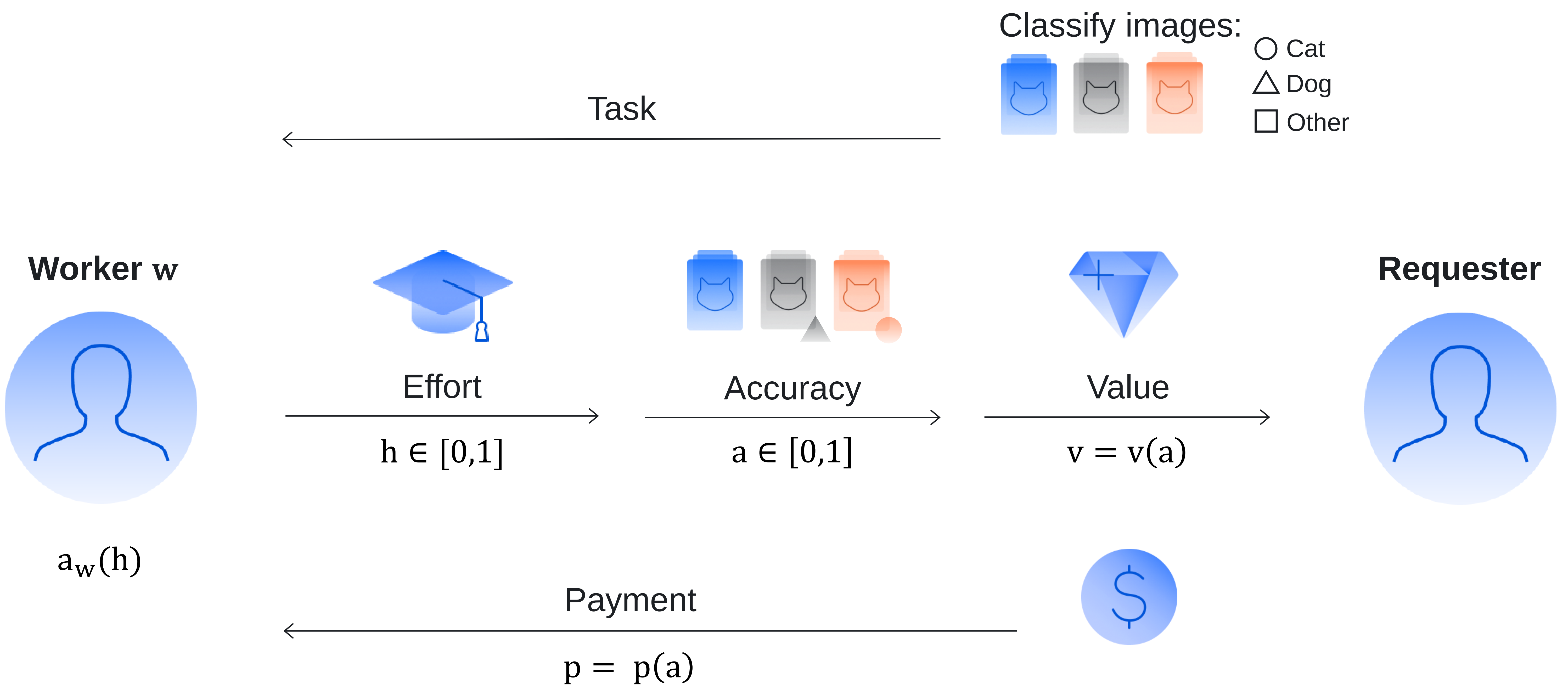
Performance-based pricing

aka dynamic pricing

Pool settings: dynamic pricing

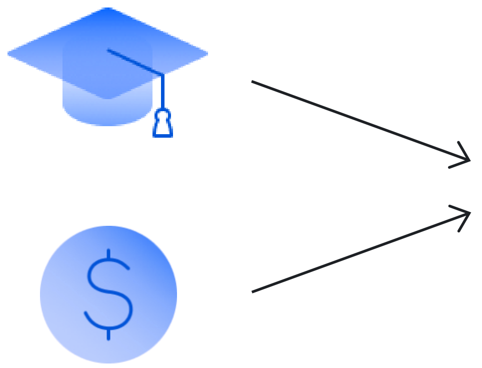


Labeling as a game: notation



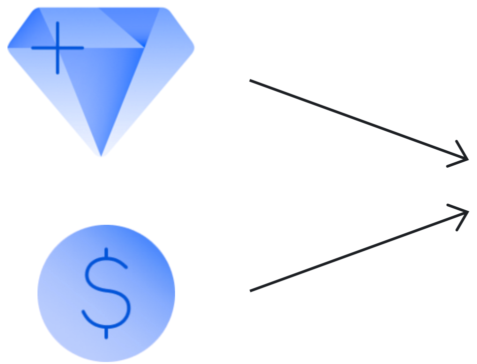
Labeling as a game: formalization

- Each worker w chooses a level of effort h for labeling object to maximize earnings per unit of spent effort:



$$\frac{p\left(a_w(h)\right)}{h} \to \max_{h \geq 0}$$

- The requester chooses a pricing p(a) to minimize payments per unit of obtained value



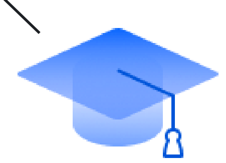
$$\frac{v(a\,)}{p(a)} \to \max_{a \in [0,1]}$$

Labeling as a game: incentive compatible pricing

- Assume $a_w(h)$ is a linear function of h:

$$a_w(h) = c_1 h + c_0$$

Accuracy



Theorem: The requester and workers maximize their utility simultaneously if the pricing $p(a)$ for each label is proportional to its accuracy a

Performance-based pricing in practice: settings

- Price p for the level of accuracy a_0 : $\Pr(\hat{z} = z) \geq a_0$ E.g.:

$p = 0.3\$$ $a_0 = 0.99$

- $\hat{q}_w = \Pr(y^w = z)$ — estimated quality level of worker w, e.g. the fraction of correct labels for golden set (GS):

5 correct GS among 10 $\hat{q}_w = 0.5$

16 correct GS among 20 $\hat{q}_w = 0.8$

100 correct GS among 100 $\hat{q}_w = 1$

Wang, Ipeirotis, and Provost, Quality-Based Pricing for Crowdsourced Workers, 2013

Performance-based pricing in practice: settings

- Aggregation $\hat{z}_j^{wMV} = \arg\max_{y=1,\dots,K} \sum_{w \in W_j} \hat{q}_w \delta(y = y_j^w)$

- IRL algorithm is based on the expected accuracy of $\hat{z}_j^{wMV}$

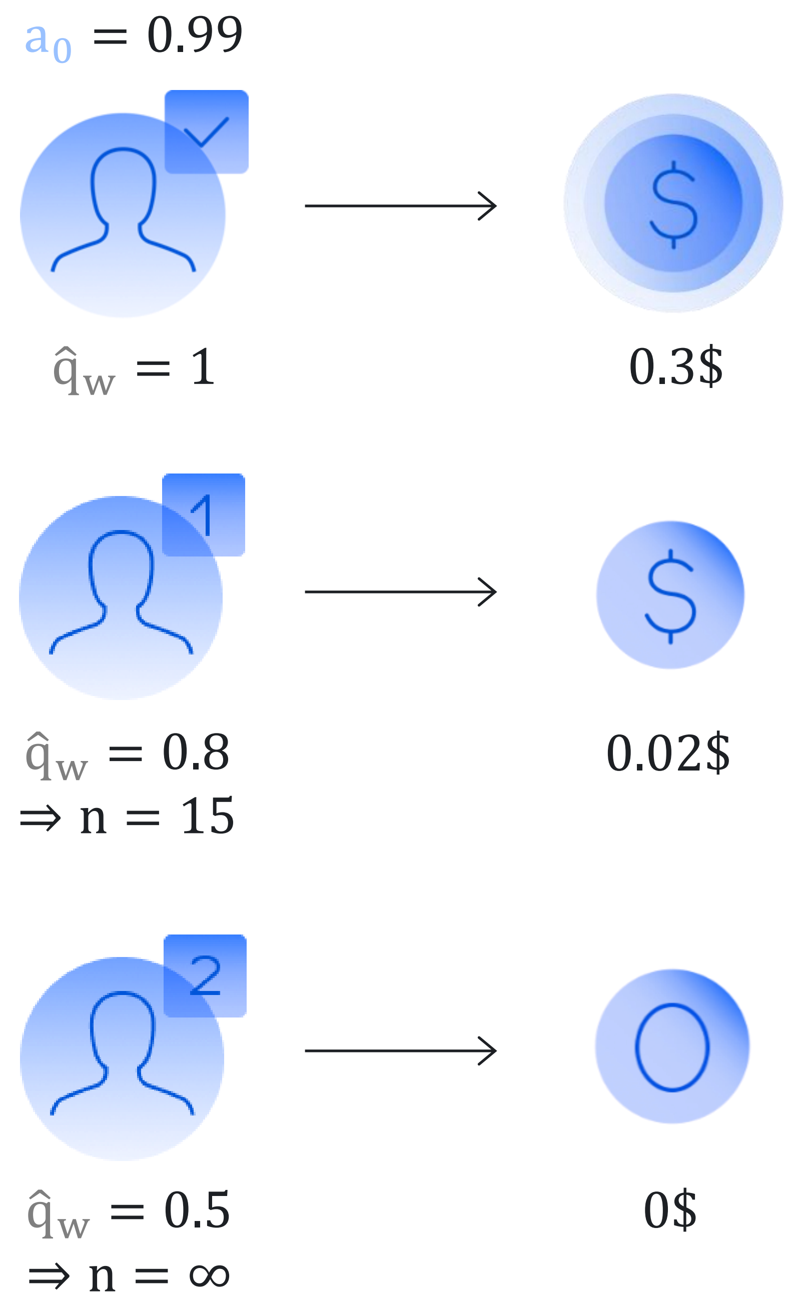
Performance-based pricing in practice

- Pricing rules

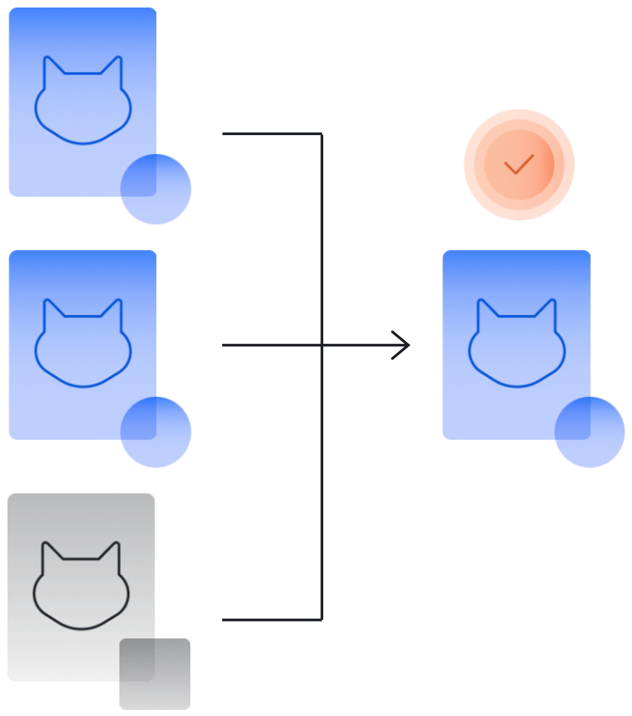
1. If $\hat{q}_w \geq a_0$, then the price is p
2. Else find n:

$$\underbrace{\sum_{k=0}^{n/2}\binom{n}{k}\hat{q}_w^{\,n-k}(1-\hat{q}_w)^k}_{\text{Expected accuracy for MV}} \geq a_0$$

The price is p/n

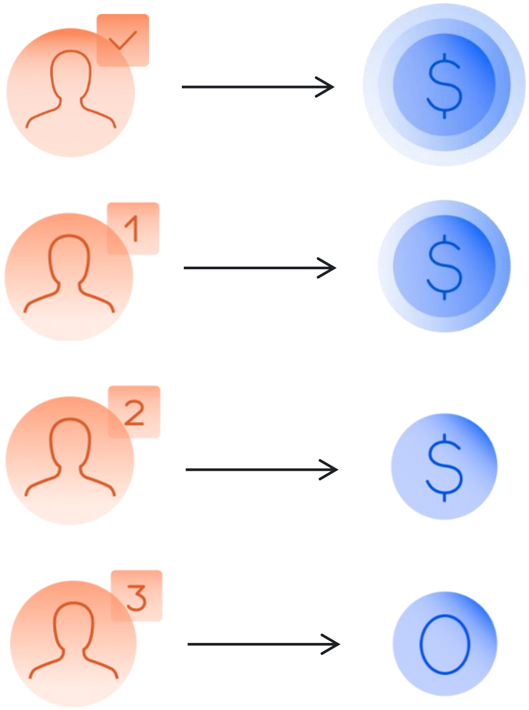


Key components of labeling with crowds



Aggregation

Incremental relabeling



Performance-based pricing