Algorithm 1: ICRL with Random Selection

Input: Question Q, rounds n, samples per round k
Output: Final answer distribution $\hat{P}$

Initialization
  Context $\mathcal{C} \leftarrow \{Q\}$;
for $t \leftarrow 1$ **to** n **do**
  Generate k chains $\{A_{tj}\}_{j=1}^{k}$ using prompt $\mathcal{C}$;
  Extract numeric answers $\{a_{tj}\}_{j=1}^{k}$;
  $a^\star \leftarrow \text{majority\_vote}(\{a_{tj}\})$;
  Randomly pick $j^{\text{rand}} \sim \text{Unif}\{1, \ldots, k\}$;
  if $a_{tj^{\text{rand}}} = a^\star$ **then**
    $r \leftarrow 1$
  else
    $r \leftarrow 0$
  Append $(A_{tj^{\text{rand}}}, r)$ to $\mathcal{C}$;
$\hat{P} \leftarrow \text{infer\_distribution}(Q, \mathcal{C})$;
return $\hat{P}$

Algorithm 2: Entropy-Minimisation ICRL (buffer size m)

Input: Question Q, rounds n, samples per round k, buffer limit m
Output: Final answer distribution $\hat{P}$

Context $\mathcal{C} \leftarrow \{Q\}$; Buffer $M \leftarrow \emptyset$;
for $t \leftarrow 1$ **to** n **do**
  Generate k candidate chains $\{A_{tj}\}_{j=1}^{k}$ with prompt $\mathcal{C}$;
  Extract numeric answers $\{a_{tj}\}$ and compute majority $a^\star$;
  foreach $j = 1 \ldots k$ **do**
    $r_{tj} \leftarrow \mathbf{1}[a_{tj} = a^\star]$;
    $\Delta H_{tj} \leftarrow H(\mathcal{C}) - H\big(\mathcal{C} \cup (A_{tj}, r_{tj})\big)$;
  Select $j^{\text{best}} \leftarrow \arg\max_j \Delta H_{tj}$;
  $(A_{\text{new}}, r_{\text{new}}) \leftarrow (A_{tj^{\text{best}}}, r_{tj^{\text{best}}})$;
  if $|M| < m$ **then**
    $M \leftarrow M \cup \{(A_{\text{new}}, r_{\text{new}})\}$;
  else
    foreach $(A_i, r_i) \in M$ **do**
      $\Delta H_i \leftarrow H(\mathcal{C}) - H\big(\mathcal{C} \setminus (A_i, r_i) \cup (A_{\text{new}}, r_{\text{new}})\big)$;
    $i^{\text{weak}} \leftarrow \arg\min_i \Delta H_i$; replace $(A_{i^{\text{weak}}}, r_{i^{\text{weak}}})$ with $(A_{\text{new}}, r_{\text{new}})$;
  Update context $\mathcal{C} \leftarrow \{Q\} \cup M$;
$\hat{P} \leftarrow \text{infer\_distribution}(Q, \mathcal{C})$;
return $\hat{P}$