Algorithm 2 Sample Collection Traversal

function COLLECTSAMPLES($h, p, \theta_0, \theta_1, B^r_p, B^s$)
  if h is terminal **then**
    return $u_p(h)$ ▷ Return the traverser's payoff
  else if $P(h) = p$ **then** ▷ If it's the traverser's turn to act
    $\sigma(I) \leftarrow$ CALCULATE-STRATEGY($I(h), \theta_p$) ▷ Compute infoset action probabilities
    $v \leftarrow 0$
    for $a \in A(h)$ **do**
      $v(a) \leftarrow$ COLLECTSAMPLES($h \cdot a, p, \theta_0, \theta_1, B^r_p, B^s$) ▷ Traverse each action
      $v \leftarrow v + \sigma(I, a) \cdot v(a)$ ▷ Update the expected value
    for $a \in A(h)$ **do**
      $\tilde{d}(I, a) \leftarrow v(a) - v$
    Add $\{(I, \tilde{d}(I), t)\}$ to B^r_p ▷ Add vector of action advantages to buffer
  else if $P(h) = 1 - p$ **then** ▷ If it's the opponent's turn to act
    $\sigma(I) \leftarrow$ CALCULATE-STRATEGY($I(h), \theta_{1-p}$) ▷ Compute infoset action probabilities
    Add $\{(I, \sigma(I), t)\}$ to B^s ▷ Add vector of action probabilities to buffer
    $a \sim \sigma(I)$ ▷ Sample an action from the probability distribution
    return COLLECTSAMPLES($h \cdot a, p, \theta_0, \theta_1, B^r_p, B^s$)
  else ▷ h is a chance node
    $a \sim \sigma(h)$ ▷ Sample a chance outcome
    return COLLECTSAMPLES($h \cdot a, p, \theta_0, \theta_1, B^r_p, B^s$)

Algorithm 3 Infoset Strategy Computation

function CALCULATE-STRATEGY(I, θ_p) ▷ Calculates strategy based on predicted advantages
  $\mathit{sum} \leftarrow 0$
  $\hat{D}(I) \leftarrow f(I|\theta_p)$ ▷ Estimate vector of advantages
  for $a \in A(I)$ **do**
    $\mathit{sum} \leftarrow \mathit{sum} + \max\{0, \hat{D}(I, a)\}$
  if $\mathit{sum} > 0$ **then** ▷ Apply Regret Matching
    for $a \in A(I)$ **do**
      $\sigma(I, a) \leftarrow \frac{\max\{0, \hat{D}(I,a)\}}{\mathit{sum}}$
  else ▷ Choose the highest-advantage action
    for $a \in A(I)$ **do**
      $\sigma(I, a) \leftarrow 0$
    $\sigma(I, \mathrm{argmax}_a\{\hat{D}(I, a)\}) = 1$
  return $\sigma(I_i)$

Algorithm 4 Network Training

function TRAINNETWORK(B, S)
  Initialize θ randomly.
  for $b = 1..N_{train}$ **do**
    for $i = 1..N_{batch}$ **do**
      $(I_i, y_i, t_i) \sim B$ ▷ sample an infoset, action pair from the buffer
      $\hat{z} \leftarrow f(I_i|\theta)$ ▷ predict regret or strategy vector
      if S **then**
        for $a \in A$ **do**
          $\hat{y}_{i,a} \leftarrow \frac{e^{\hat{z}_a}}{\sum_{a'} e^{\hat{z}_{a'}}}$ ▷ apply softmax if computing strategy vector
      else
        $\hat{y}_i \leftarrow \hat{z}$
    $\mathcal{L} \leftarrow \sum_0^{N_{batch}} t_i (y_i - \hat{y}_i)^2$
    $\theta \leftarrow \mathrm{StepAdam}(\theta, \nabla_\theta \mathcal{L})$
  return θ