Algorithm 1 `Main: Training.`

Initialize environment (OpenAI Gym format).
Initialize Learner policy $\pi^{(0)}$.
Initialize Expert policy $\pi^{(E)}$.
Initialize probabilities $[\beta^{(n)}]_{n=1}^{N}$.
Initialize $i = 0$.
while not done **do**
  #Imitation Learning Loop.
  Rollout $\mathcal{T} \leftarrow \texttt{Rollout}(\pi^{(i)}, \pi^{(E)}, [\beta^{(n)}]_{n=1}^{N}, \cdots)$.
  $\pi^{(i+1)} \leftarrow \texttt{Learn}(\pi^{(i)}, \pi^{(E)}, \mathcal{T}, \cdots)$, where `Learn` is replaced with `PPO`, standard backpropagation, etc.
  $i \leftarrow i + 1$.
end while
Return $\pi^{(i)}$.

Algorithm 2 `Main: Evaluation.`

Initialize environment (OpenAI Gym format).
Initialize environment state $\mathbf{s}_0$.
Get $\pi^{(i)}$ from `Main: Training`.
Initialize $j = 0$.
while not done **do**
  Sample $\mathbf{a}_j \sim \pi^{(i)}(\cdot|\mathbf{s}_j)$.
  Update $\mathbf{s}_{j+1} = \texttt{step}(\mathbf{s}_j, \mathbf{a}_j)$
  $j \leftarrow j + 1$.
end while
Return trajectory $\{(\mathbf{s}_j, \mathbf{a}_j)\}$.

Algorithm 3 `Rollout.`

Main Inputs: Learner policy $\pi^{(i)}$, Expert policy $\pi^{(E)}$, $[\beta^{(n)}]_{n=1}^{N}$.
for $n = 1, \cdots, N$ num. trajectories **do**
  Initialize trajectory $\mathbf{s}_0^{(n)}$.
  Initialize $j = 0$, empty set $\mathcal{T} = \{\}$.
  while trajectory not done **do**
    Get Learner action $\mathbf{a}_j^{(n,i)} \sim \pi^{(i)}(\cdot|\mathbf{s}_j^{(n)})$.
    Get Expert action $\mathbf{a}_j^{(n,E)} \sim \pi^{(E)}(\cdot|\mathbf{s}_j^{(n)})$.
    #Case 1
    Set $\mathbf{a}_j^{(n)} = \mathbf{a}_j^{(n,E)}$ w.p. $\beta^{(n)}$, else $\mathbf{a}_j^{(n)} = \mathbf{a}_j^{(n,i)}$.
    #Case 2
    Set $\mathbf{a}_j^{(n)} = \beta^{(n)}\mathbf{a}_j^{(n,E)} + (1 - \beta^{(n)})\mathbf{a}_j^{(n,i)}$.
  end while
  Save rollout $\mathcal{T} \leftarrow \{(\mathbf{s}_j^{(n)}, \mathbf{a}_j^{(n)})\}_j$.
end for
Return rollouts $\mathcal{T} \triangleq \{(\mathbf{s}_j^{(n)}, \mathbf{a}_j^{(n)})\}_{j,n}$.