Algorithm 1 Action-Free Guide

Input: states s, returns-to-go $\hat{R}$, time steps t
 # get positional embedding for each time step
$f_t = \text{Embed}_t(t)$
 # compute the state and return-to-go embeddings
$f_s,\ f_{\hat{R}} = \text{Embed}_s(s) + f_t,\ \text{Embed}_R(\hat{R}) + f_t$
 # send to transformer in the order $(s_0, \hat{R}_0, s_1, \hat{R}_1, \ldots)$
$f_{output} = \text{Transformer}(\text{stack}(f_s,\ f_{\hat{R}}))$
 # predict the state change
$\Delta s = \text{Pred}_s(\text{unstack}(f_{output}.\text{states}))$
Output: Δs + s

Embed_t: a single-layer temporal encoder

Embed_t: a single-layer state encoder

Embed_R: a single-layer return-to-go encoder

stack: operation to stack state features f_s and return-to-go features $f_{\hat{R}}$

Pred_s: state decoder converting output state features to the state change Δs

Algorithm 2 Compute Guiding Reward

Input: states $s_{1:t}$, return-to-go $\hat{R}_{1:t}$, policy π, state standard deviation σ_s, environment env, AFDT with context length K
repeat
  # get AFDT's prediction of the next state
  $\widetilde{s}_{t+1} = \text{AFDT}(s_{t-K+1:t}, \hat{R}_{t-K+1,t})$
  # apply the policy in the environment for one step
  $a_t = \pi(s_t)$
  $s_{t+1}, r_e = env.\text{step}(a_t)$
  # compute current guiding reward using Eq.4
  $r_g = -\|\frac{1}{\sigma_s} \odot (\widetilde{s}_{t+1} - s_{t+1})\|_2$
  # update return-to-go (same as DT) and time step
  $\hat{R}_{t+1} = \hat{R}_t$ - r_e
  $t = t + 1$
until Episode is finished

$\tilde{s}_{t+1}$: planned next state from AFDT

r_e: environment reward

r_g: intrinsic guiding reward