Algorithm 1 Noise-Aliased Diffusion Steering via Reinforcement Learning (DSRL-NA)

input: pretrained diffusion policy $\pi_{\text{dp}}^{\mathcal{W}}$, offline data $\mathfrak{D}_{\text{off}}$ and/or online environment $\mathcal{M}$

Initialize replay buffer $\mathfrak{B} \leftarrow \mathfrak{D}_{\text{off}}$, $\mathcal{A}$-critic $Q^{\mathcal{A}}$, latent-noise critic $Q^{\mathcal{W}}$, latent-noise actor $\pi^{\mathcal{W}}$

for $t = 1, \ldots, T$ **do**

  Update $Q^{\mathcal{A}}$: $\min_{Q^{\mathcal{A}}} \mathbb{E}_{(\boldsymbol{s},\boldsymbol{a},r,\boldsymbol{s}')\sim\mathfrak{B},\boldsymbol{a}'\sim\pi_{\text{dp}}^{\mathcal{W}}(\boldsymbol{s}',\pi^{\mathcal{W}}(\boldsymbol{s}'))}\left[(Q^{\mathcal{A}}(\boldsymbol{s},\boldsymbol{a}) - r - \gamma\bar{Q}^{\mathcal{A}}(\boldsymbol{s}',\boldsymbol{a}'))^2\right]$

  Update $Q^{\mathcal{W}}$: $\min_{Q^{\mathcal{W}}} \mathbb{E}_{\boldsymbol{s}\sim\mathfrak{B},\boldsymbol{w}\sim\mathcal{N}(0,I)}\left[(Q^{\mathcal{W}}(\boldsymbol{s},\boldsymbol{w}) - Q^{\mathcal{A}}(\boldsymbol{s},\pi_{\text{dp}}^{\mathcal{W}}(\boldsymbol{s},\boldsymbol{w})))^2\right]$

  Update $\pi^{\mathcal{W}}$: $\max_{\pi^{\mathcal{W}}} \mathbb{E}_{\boldsymbol{s}\sim\mathfrak{B}}\left[Q^{\mathcal{W}}(\boldsymbol{s},\pi^{\mathcal{W}}(\boldsymbol{s}))\right]$

  if access to online environment $\mathcal{M}$ **then**

    Sample latent-noise action $\boldsymbol{w}_t \sim \pi^{\mathcal{W}}(\boldsymbol{s}_t)$ and compute $\boldsymbol{a}_t \leftarrow \pi_{\text{dp}}^{\mathcal{W}}(\boldsymbol{s}_t,\boldsymbol{w}_t)$

    Play $\boldsymbol{a}_t$ in $\mathcal{M}$, observe r_t and next state $\boldsymbol{s}_{t+1}$, and add $(\boldsymbol{s}_t,\boldsymbol{a}_t,r_t,\boldsymbol{s}_{t+1})$ to $\mathfrak{B}$