

# DDBot: Differentiable Physics-based Digging Robot for Unknown Granular - Supplementary Materials

Xintong Yang<sup>1</sup>, Minglun Wei<sup>1</sup>, Yu-Kun Lai<sup>2</sup>, Ze Ji<sup>1</sup>

## S.I. GRADIENTS OF THE SKILL PARAMETERS FROM ACTIONS THROUGH THE SKILL-TO-ACTION MAPPING

This subsection presents the gradient chain of the skill-to-action mapping function, which is defined mathematically by Eqs. 8 to 22. We are concerned of the gradients  $\frac{\partial \bar{a}(\theta)}{\partial \theta}$ . The following derives the gradient of each parameter from phases 1 to 4, assuming the calculation of a rounded number of timesteps is adopted. For phases 1 and 4, the alternative approach of using a unrounded number of timesteps in per-action displacement calculation is also presented. Please also refer to Table I for math notations.

The first parameter,  $\theta_{displace}$ , is involved in computing only the first phase of the skill actions. Thus, with  $T_1 = 0$  ( $\theta_{displace} = \theta_{rotate} = 0$ ),  $\frac{\partial \bar{a}}{\partial \theta_{displace}} = \frac{\partial \bar{a}[0:T_1]}{\partial \theta_{displace}} = 0$ . Then, because  $a_t$  is the same within each phase, for any timestep  $t$  that satisfies  $T_1 > t > 0$ , we have:

$$\left. \frac{\partial \bar{a}[0:T_1]}{\partial \theta_{displace}} \right|_{rounded\_T_1} = \sum_{t=0}^{T_1} \frac{\partial a_t}{\partial \theta_{displace}} = T_1 \cdot \frac{\partial a_t}{\partial \Delta x_1} \cdot \frac{\partial \Delta x_1}{\partial d_1} \cdot \frac{\partial d_1}{\partial \theta_{displace}} \quad \text{Thus, we have}$$

$$\begin{aligned} \frac{\partial a_t}{\partial \Delta x_1} &= \frac{da_t[0]}{d\Delta x_1} = 1, \quad \frac{\partial \Delta x_1}{\partial d_1} = \frac{d\Delta x_1}{dd_1} = \frac{1}{T_1}, \\ \frac{\partial d_1}{\partial \theta_{displace}} &= \frac{dd_1}{d\theta_{displace}} = 0.12 \end{aligned}$$

Thus, the gradient of  $\theta_{displace}$  with the rounded number of timesteps being used in skill-to-action mapping is:

$$\left. \frac{\partial \bar{a}[0:T_1]}{\partial \theta_{displace}} \right|_{rounded\_T_1} = 0.12$$

If  $T_1$  is replaced with the unrounded  $T_1^{float}$  to compute the per-step action, for any timestep  $t$  that satisfies  $T_1 > t > 0$ , we have:

$$\begin{aligned} \left. \frac{\partial \bar{a}[0:T_1]}{\partial \theta_{displace}} \right|_{unrounded\_T_1} &= \sum_{t=0}^{T_1} \frac{\partial a_t}{\partial \theta_{displace}} \\ &= T_1 \cdot \left[ \frac{\partial a_t}{\partial \Delta x_1} \cdot \frac{\partial \Delta x_1}{\partial \theta_{displace}} + \frac{\partial a_t}{\partial \Delta r x_1} \cdot \frac{\partial \Delta r x_1}{\partial \theta_{displace}} \right] \\ &= T_1 \cdot \left[ \frac{\partial a_t}{\partial \Delta x_1} \cdot \left( \frac{\partial \Delta x_1}{\partial d_1} \cdot \frac{\partial d_1}{\partial \theta_{displace}} + \frac{\partial \Delta x_1}{\partial T_1^{float}} \cdot \frac{\partial T_1^{float}}{\partial \theta_{displace}} \right) \right. \\ &\quad \left. + \frac{\partial a_t}{\partial \Delta r x_1} \cdot \frac{\partial \Delta r x_1}{\partial T_1^{float}} \cdot \frac{\partial T_1^{float}}{\partial \theta_{displace}} \right] \end{aligned}$$

where,

<sup>1</sup>Xintong Yang, Minglun Wei and Ze Ji are with the School of Engineering, Cardiff University, Cardiff, United Kingdom. {yangx66, weim9, jiz1}@cardif.ac.uk

<sup>2</sup>Yu-Kun Lai is with the School of Computer Science and Informatics, Cardiff University, Cardiff, United Kingdom. laiy4@cardif.ac.uk

$$\begin{aligned} \frac{\partial \Delta x_1}{\partial d_1} &= \frac{d\Delta x_1}{dd_1} = \frac{1}{T_1^{float}}, \\ \frac{\partial \Delta x_1}{\partial T_1^{float}} &= \frac{d\Delta x_1}{dT_1^{float}} = \frac{-d_1}{(T_1^{float})^2}, \\ \frac{a_t}{\Delta r x_1} &= 1, \quad \frac{\partial \Delta r x_1}{\partial T_1^{float}} = \frac{d\Delta r x_1}{dT_1^{float}} = \frac{-\varphi_1}{(T_1^{float})^2} \\ \frac{\partial T_1^{float}}{\partial \theta_{displace}} &= \begin{cases} \frac{\partial T_1^{float}}{\partial d_1} \cdot \frac{\partial d_1}{\partial \theta_{displace}}, & T_1^{float} \geq T_{\varphi_1}^{float} > 0 \\ \frac{\partial T_1^{float}}{\partial \varphi_1} = 0, & 0 < T_1^{float} < T_{\varphi_1}^{float} \end{cases} \end{aligned}$$

where,

$$\frac{\partial T_1^{float}}{\partial d_1} = \frac{1}{v_l \cdot \Delta t}, \quad \frac{\partial |d_1|}{\partial \theta_{displace}} = \begin{cases} 0.12, & \theta_{displace} > 0 \\ -0.12, & \theta_{displace} < 0 \\ 0, & \theta_{displace} = 0 \end{cases}$$

$$\begin{aligned} \left. \frac{\partial \bar{a}[0:T_1]}{\partial \theta_{displace}} \right|_{unrounded\_T_1} &= \begin{cases} T_1 \cdot \left[ 1 \cdot \left( \frac{1}{T_1^{float}} \cdot 0.12 + \frac{-d_1}{(T_1^{float})^2} \cdot \frac{1}{v_l \cdot \Delta t} \cdot 0.12 \right) \right. \\ \quad \left. + 1 \cdot \frac{-\varphi_1}{(T_1^{float})^2} \cdot \frac{1}{v_l \cdot \Delta t} \cdot 0.12 \right], & T_1^{float} \geq T_{\varphi_1}^{float} > 0, \theta_{displace} > 0 \\ T_1 \cdot \left[ 1 \cdot \left( \frac{1}{T_1^{float}} \cdot 0.12 + \frac{-d_1}{(T_1^{float})^2} \cdot \frac{1}{v_l \cdot \Delta t} \cdot -0.12 \right) \right. \\ \quad \left. + 1 \cdot \frac{-\varphi_1}{(T_1^{float})^2} \cdot \frac{1}{v_l \cdot \Delta t} \cdot -0.12 \right], & T_1^{float} \geq T_{\varphi_1}^{float} > 0, \theta_{displace} < 0 \\ T_1 \cdot 1 \cdot \frac{1}{T_1^{float}} \cdot 0.12, & 0 < T_1^{float} < T_{\varphi_1}^{float} \\ T_1 \cdot \frac{-\pi \cdot \theta_{rotate} \cdot v_l \cdot \Delta t}{0.36 \cdot \theta_{displace}^2}, & T_1^{float} \geq T_{\varphi_1}^{float} > 0, \theta_{displace} > 0 \\ T_1 \cdot \left( \frac{2 \cdot v_l \cdot \Delta t}{\theta_{displace}} + \frac{\pi \cdot \theta_{rotate} \cdot v_l \cdot \Delta t}{0.36 \cdot \theta_{displace}^2} \right), & T_1^{float} \geq T_{\varphi_1}^{float} > 0, \theta_{displace} < 0 \\ T_1 \cdot \frac{0.12 \cdot v_w \cdot \Delta t}{|\theta_{rotate}| \cdot \pi / 3}, & 0 < T_1^{float} < T_{\varphi_1}^{float} \end{cases} \end{aligned}$$

Secondly, as  $\theta_{rotate}$  is involved in the computation of skill phases 1, 2 and 4, its gradient is  $\frac{\partial \bar{a}}{\partial \theta_{rotate}} = \frac{\partial \bar{a}[0:T_1]}{\partial \theta_{rotate}} + \frac{\partial \bar{a}[T_1:T_1+T_2]}{\partial \theta_{rotate}} + \frac{\partial \bar{a}[T_1+T_2+T_3:T_1+T_2+T_3+T_4]}{\partial \theta_{rotate}}$ . Again, with  $T_1 = 0$  ( $\theta_{displace} = \theta_{rotate} = 0$ ), we have  $\frac{\partial \bar{a}[0:T_1]}{\partial \theta_{rotate}} = 0$ . Then, for any timestep  $t$  that satisfies  $T_1 > t > 0$ , we have:

$$\left. \frac{\partial \bar{a}[0:T_1]}{\partial \theta_{rotate}} \right|_{rounded\_T_1} = T_1 \cdot \frac{\partial a_t}{\partial \Delta r x_1} \cdot \frac{\partial \Delta r x_1}{\partial \varphi_1} \cdot \frac{\partial \varphi_1}{\partial \theta_{rotate}} = \frac{\pi}{3}$$

Again, if  $T_1$  is replaced with the unrounded value, we have

$$\begin{aligned} & \left. \frac{\partial \bar{a}[0 : T_1]}{\partial \theta_{rotate}} \right|_{unrounded\_T_1} \\ &= T_1 \cdot \left[ \frac{\partial a_t}{\partial \Delta r x_1} \cdot \left( \frac{\partial \Delta r x_1}{\partial \varphi_1} \cdot \frac{\partial \varphi_1}{\partial \theta_{rotate}} + \frac{\partial \Delta r x_1}{\partial T_1^{float}} \cdot \frac{\partial T_1^{float}}{\partial \theta_{rotate}} \right) \right. \\ & \quad \left. + \frac{\partial a_t}{\partial \Delta x_1} \cdot \frac{\partial \Delta x_1}{\partial T_1^{float}} \cdot \frac{\partial T_1^{float}}{\partial \theta_{rotate}} \right] \end{aligned}$$

where,

$$\begin{aligned} \frac{\partial \Delta r x_1}{\partial \varphi_1} &= \frac{d \Delta r x_1}{d \varphi_1} = \frac{1}{T_1^{float}}, \\ \frac{\partial T_1^{float}}{\partial \theta_{rotate}} &= \begin{cases} \frac{\partial T_1^{float}}{\partial \theta_{rotate}} = 0, & T_{d_1}^{float} \geq T_{\varphi_1}^{float} > 0 \\ \frac{\partial T_{d_1}^{float}}{\partial |\varphi_1|} \cdot \frac{\partial |\varphi_1|}{\partial \theta_{rotate}}, & 0 < T_{d_1}^{float} < T_{\varphi_1}^{float} \end{cases}, \\ \frac{\partial T_{d_1}^{float}}{\partial |\varphi_1|} &= \frac{1}{v_w \cdot \Delta t}, \quad \frac{\partial |\varphi_1|}{\partial \theta_{rotate}} = \begin{cases} \pi/3, & \theta_{rotate} > 0 \\ -\pi/3, & \theta_{rotate} < 0 \\ 0, & \theta_{rotate} = 0 \end{cases} \end{aligned}$$

Thus,

$$\begin{aligned} & \left. \frac{\partial \bar{a}[0 : T_1]}{\partial \theta_{rotate}} \right|_{unrounded\_T_1} \\ &= \begin{cases} T_1 \cdot \left[ 1 \cdot \left( \frac{1}{T_{\varphi_1}^{float}} \cdot \frac{\pi}{3} + \frac{-\varphi_1}{(T_1^{float})^2} \cdot \frac{1}{v_w \cdot \Delta t} \cdot \frac{\pi}{3} \right) \right. \\ \quad \left. + 1 \cdot \frac{-d_1}{(T_{\varphi_1}^{float})^2} \cdot \frac{\pi/3}{v_w \cdot \Delta t} \right], & T_{\varphi_1}^{float} > T_{d_1}^{float} > 0, \theta_{rotate} > 0 \\ T_1 \cdot \left[ 1 \cdot \left( \frac{1}{T_{\varphi_1}^{float}} \cdot \frac{\pi}{3} + \frac{-\varphi_1}{(T_1^{float})^2} \cdot \frac{1}{v_w \cdot \Delta t} \cdot \left( -\frac{\pi}{3} \right) \right) \right. \\ \quad \left. + 1 \cdot \frac{-d_1}{(T_{\varphi_1}^{float})^2} \cdot \frac{-\pi/3}{v_w \cdot \Delta t} \right], & T_{\varphi_1}^{float} > T_{d_1}^{float} > 0, \theta_{rotate} < 0 \\ T_1 \cdot 1 \cdot \frac{1}{T_{d_1}^{float}} \cdot \frac{\pi}{3}, & T_{\varphi_1}^{float} \leq T_{d_1}^{float} \end{cases} \\ &= \begin{cases} T_1 \cdot \frac{-0.36 \cdot \theta_{displace} \cdot v_w \cdot \Delta t}{\theta_{rotate}^2 \cdot \pi}, & T_{\varphi_1}^{float} > T_{d_1}^{float} > 0, \theta_{rotate} > 0 \\ T_1 \cdot \left( \frac{2 \cdot v_w \cdot \Delta t}{\theta_{rotate}} + \frac{0.36 \cdot \theta_{displace} \cdot v_w \cdot \Delta t}{\theta_{rotate}^2 \cdot \pi} \right), & T_{\varphi_1}^{float} > T_{d_1}^{float} > 0, \theta_{rotate} < 0 \\ T_1 \cdot \frac{\frac{\pi}{3} \cdot v_l \cdot \Delta t}{|\theta_{displace}| \cdot 0.12}, & 0 < T_{\varphi_1}^{float} \leq T_{d_1}^{float} \end{cases} \end{aligned}$$

Note that we assume  $T_1^{float} = T_{d_1}^{float}$  when  $T_{d_1}^{float} = T_{\varphi_1}^{float}$  for the  $max()$  operation (Eq. 10).

The next item of the gradient of  $\theta_{rotate}$  is  $\frac{\partial \bar{a}[T_1 : T_1 + T_2]}{\partial \theta_{rotate}}$ , where  $T_2$  is kept an integer. Again, it is zero with  $T_2 = 0$ , and for any timestep  $t$  that satisfies  $T_1 + T_2 > t > T_1$ , we have:

$$\begin{aligned} & \left. \frac{\partial \bar{a}[T_1 : T_1 + T_2]}{\partial \theta_{rotate}} \right|_{rounded\_T_2} = T_2 \cdot \left[ \frac{\partial a_t}{\partial \Delta x_2} \cdot \frac{\partial \Delta x_2}{\partial \varphi_1} \cdot \frac{\partial \varphi_1}{\partial \theta_{rotate}} \right. \\ & \quad \left. + \frac{\partial a_t}{\partial \Delta z_2} \cdot \frac{\partial \Delta z_2}{\partial \varphi_1} \cdot \frac{\partial \varphi_1}{\partial \theta_{rotate}} \right] \\ &= T_2 \cdot \frac{-d_2}{T_2} \cdot \frac{\pi}{3} \cdot \left( \sin(\varphi_1 + \frac{\pi}{2}) + \cos(\varphi_1 + \frac{\pi}{2}) \right) \\ &= -d_2 \cdot \frac{\pi}{3} \cdot \left( \sin(\varphi_1 + \frac{\pi}{2}) + \cos(\varphi_1 + \frac{\pi}{2}) \right) \end{aligned}$$

where  $d_2 = (\theta_{insert\_dist} + 1) \cdot 0.03$  and  $\varphi_1 = \theta_{rotate} \cdot \frac{\pi}{3}$ . Note that phases 2 and 3 are not suitable for the unrounded number replacement of the number of timesteps due to the zero gradient issue.

The third part of the gradient of  $\theta_{rotate}$  comes from skill phase 4. Because  $T_4 > T_{lift}^{float} > 0$ , for any timestep  $t$  that satisfies  $T_1 + T_2 + T_3 + T_4 > t > T_1 + T_2 + T_3$ , we have

$$\begin{aligned} & \left. \frac{\partial \bar{a}[T_1 + T_2 + T_3 : T_1 + T_2 + T_3 + T_4]}{\partial \theta_{rotate}} \right|_{rounded\_T_4} \\ &= T_4 \cdot \frac{\partial a_t}{\partial \Delta r x_4} \cdot \frac{\partial \Delta r x_4}{\partial \varphi_1} \cdot \frac{\partial \varphi_1}{\partial \theta_{rotate}} \\ &= T_4 \cdot 1 \cdot -\frac{1}{T_4} \cdot \frac{\pi}{3} = -\frac{\pi}{3} \end{aligned}$$

If  $T_4$  is replaced with the unrounded number, we have

$$\begin{aligned} & \left. \frac{\partial \bar{a}[T_1 + T_2 + T_3 : T_1 + T_2 + T_3 + T_4]}{\partial \theta_{rotate}} \right|_{unrounded\_T_4} \\ &= T_4 \cdot \left[ \frac{\partial a_t}{\partial \Delta r x_4} \cdot \left( \frac{\partial \Delta r x_4}{\partial \varphi_1} \cdot \frac{\partial \varphi_1}{\partial \theta_{rotate}} + \frac{\partial \Delta r x_4}{\partial T_4^{float}} \cdot \frac{\partial T_4^{float}}{\partial \theta_{rotate}} \right) \right. \\ & \quad \left. + \frac{\partial a_t}{\partial \Delta z_4} \cdot \frac{\partial \Delta z_4}{\partial T_4^{float}} \cdot \frac{\partial T_4^{float}}{\partial \theta_{rotate}} \right] \end{aligned}$$

where,

$$\begin{aligned} \frac{\partial \Delta r x_4}{\partial \varphi_1} &= \frac{d \Delta r x_4}{d \varphi_1} = -\frac{1}{T_4^{float}}, \\ \frac{\partial \Delta r x_4}{\partial T_4^{float}} &= \frac{d \Delta r x_4}{d T_4^{float}} = \frac{-\varphi_1}{(T_4^{float})^2} \\ \frac{\partial T_4^{float}}{\partial \theta_{rotate}} &= \begin{cases} \frac{\partial T_{d_4}^{float}}{\partial \theta_{rotate}} = 0, & T_{lift}^{float} > T_{\varphi_1}^{float} > 0 \\ \frac{\partial T_{\varphi_1}^{float}}{\partial |\varphi_1|} \cdot \frac{\partial |\varphi_1|}{\partial \theta_{rotate}}, & 0 < T_{lift}^{float} \leq T_{\varphi_1}^{float} \end{cases} \end{aligned}$$

Thus, we have

$$\begin{aligned} & \left. \frac{\partial \bar{a}[T_1 + T_2 + T_3 : T_1 + T_2 + T_3 + T_4]}{\partial \theta_{rotate}} \right|_{unrounded\_T_4} \\ &= \begin{cases} T_4 \cdot \frac{-3 \cdot d_4 \cdot v_w \cdot \Delta t}{\theta_{rotate}^2 \cdot \pi}, & T_{\varphi_1}^{float} \geq T_{lift}^{float} > 0, \theta_{rotate} > 0 \\ T_4 \cdot \left( \frac{-2 \cdot v_w \cdot \Delta t}{\theta_{rotate}} + \frac{3 \cdot d_4 \cdot v_w \cdot \Delta t}{\theta_{rotate}^2 \cdot \pi} \right), & T_{\varphi_1}^{float} \geq T_{lift}^{float} > 0, \theta_{rotate} < 0 \\ T_4 \cdot \frac{-\pi \cdot v_l \cdot \Delta t}{3 \cdot d_4}, & 0 < T_{\varphi_1}^{float} < T_{lift}^{float} \end{cases} \end{aligned}$$

Note  $d_4$  is a given constant.

Thirdly,  $\frac{\partial \bar{a}}{\partial \theta_{insert\_dist}} = \frac{\partial \bar{a}[T_1 : T_1 + T_2]}{\partial \theta_{insert\_dist}}$ . Again, with  $T_2 > 0$  ( $\theta_{insert\_dist} > -1$  and  $T_2$  remains an integer), for any timestep  $t$  that satisfies  $T_1 + T_2 > t > T_1$ , we have:

$$\begin{aligned} & \left. \frac{\partial \bar{a}[T_1 : T_1 + T_2]}{\partial \theta_{insert\_dist}} \right|_{rounded\_T_2} \\ &= T_2 \cdot \left[ \frac{\partial a_t}{\partial \Delta x_2} \cdot \frac{\partial \Delta x_2}{\partial d_2} \cdot \frac{\partial d_2}{\partial \theta_{insert\_dist}} + \frac{\partial a_t}{\partial \Delta z_2} \cdot \frac{\partial \Delta z_2}{\partial d_2} \cdot \frac{\partial d_2}{\partial \theta_{insert\_dist}} \right] \\ &= T_2 \cdot \left[ 1 \cdot \frac{\cos \varphi_2}{T_2} \cdot 0.03 + 1 \cdot \frac{-\sin \varphi_2}{T_2} \cdot 0.03 \right] \\ &= 0.03 \cdot (\cos \varphi_2 - \sin \varphi_2) \end{aligned}$$

Fourthly, we derive the gradient of  $\theta_{push\_dist}$ . Again, with  $T_3 = 0$ , we have  $\frac{\partial \bar{a}}{\partial \theta_{push\_dist}} = \frac{\partial \bar{a}[T_1 + T_2 : T_1 + T_2 + T_3]}{\partial \theta_{push\_dist}} = 0$ , while when  $T_3 > 0$  ( $T_3$  remains an integer), for any timestep  $t$  that satisfies  $T_1 + T_2 + T_3 > t > T_1 + T_2$ , we have:

$$\begin{aligned} & \left. \frac{\partial \bar{a}[T_1 + T_2 : T_1 + T_2 + T_3]}{\partial \theta_{push\_dist}} \right|_{rounded\_T_3} \\ &= T_3 \cdot \left[ \frac{\partial a_t}{\partial \Delta x_3} \cdot \frac{\partial \Delta x_3}{\partial d_3} \cdot \frac{\partial d_3}{\partial \theta_{push\_dist}} + \frac{\partial a_t}{\partial \Delta z_3} \cdot \frac{\partial \Delta z_3}{\partial d_3} \cdot \frac{\partial d_3}{\partial \theta_{push\_dist}} \right] \\ &= T_3 \cdot \left[ 1 \cdot \frac{\cos(\varphi_3)}{T_3} \cdot 0.1 + 1 \cdot \frac{\sin(\varphi_3)}{T_3} \cdot 0.1 \right] \\ &= 0.1 \cdot [\cos(\varphi_3) + \sin(\varphi_3)] \end{aligned}$$

Lastly, we derive the gradient of  $\theta_{push\_angle}$ . With  $T_3 = 0$ , we have  $\frac{\partial \bar{a}}{\partial \theta_{push\_angle}} = \frac{\partial \bar{a}[T_1+T_2:T_1+T_2+T_3]}{\partial \theta_{push\_angle}} = 0$ , while when  $T_3 > 0$  ( $T_3$  remains an integer), for any timestep  $t$  that satisfies  $T_1 + T_2 + T_3 > t > T_1 + T_2$ , we have:

$$\begin{aligned} & \left. \frac{\partial \bar{a}[T_1 + T_2 : T_1 + T_2 + T_3]}{\partial \theta_{push\_angle}} \right|_{rounded\_T_3} \\ &= T_3 \cdot \left[ \frac{\partial a_t}{\partial \Delta x_3} \cdot \frac{\partial \Delta x_3}{\partial \varphi_3} \cdot \frac{\partial \varphi_3}{\partial \theta_{push\_angle}} + \frac{\partial a_t}{\partial \Delta z_3} \cdot \frac{\partial \Delta z_3}{\partial \varphi_3} \cdot \frac{\partial \varphi_3}{\partial \theta_{push\_angle}} \right] \\ &= T_3 \cdot \left[ 1 \cdot \frac{d_3}{T_3} \cdot -\sin(\varphi_3) \cdot \frac{\pi}{3} + 1 \cdot \frac{d_3}{T_3} \cdot \cos(\varphi_3) \cdot \frac{\pi}{3} \right] \\ &= \frac{\pi \cdot d_3}{3} [\cos(\varphi_3) - \sin(\varphi_3)] \end{aligned}$$

where  $d_3 = (\theta_{push\_dist} + 1) \cdot 0.1 + 0.04$  and  $\varphi_3 = (\theta_{push\_angle} + 3) \cdot \pi/3$ .

## S.II. DIFFERENTIABILITY OF THE CONSTITUTIVE MODELS

We employ the Saint Venant–Kirchhoff (SVK) model as our elastic energy density function (constitutive model) and the Drucker-Prager (DP) following [1]. In our work, we require the gradients of key physical properties. To verify the differentiability of this process, we manually calculate the partial derivatives of the input variables (i.e., the Lamé constants of the soil) with respect to the output of the elastoplasticity model (Eqs. 3 to 7). In the DP plasticity model, for cases without projection and projection onto the cone tip, we have  $\frac{\partial S'_p}{\partial \mu} = \frac{\partial S'_p}{\partial \lambda} = O_{dim}$ , where  $O_{dim}$  denotes the zero matrix of dimension  $dim$ . In contrast, for projection onto the cone surface:

$$\frac{\partial S'_p}{\partial \mu} = \frac{\partial S'_p}{\partial \delta_{\gamma_p}} \frac{\partial \delta_{\gamma_p}}{\partial \mu} \quad \frac{\partial S'_p}{\partial \lambda} = \frac{\partial S'_p}{\partial \delta_{\gamma_p}} \frac{\partial \delta_{\gamma_p}}{\partial \lambda} \quad (S.1)$$

where the partial derivative of  $S'_p$  with respect to  $\delta_{\gamma_p}$  is:

$$\frac{\partial S'_p}{\partial \delta_{\gamma_p}} = -\exp(\epsilon_p - \delta_{\gamma_p} \frac{\hat{\epsilon}_p}{\|\hat{\epsilon}_p\|}) \frac{\hat{\epsilon}_p}{\|\hat{\epsilon}_p\|} \quad (S.2)$$

and the partial derivatives of  $\delta_{\gamma_p}$  with respect to  $\mu$  and  $\lambda$  are:

$$\frac{\partial \delta_{\gamma_p}}{\partial \mu} = \frac{-\lambda dim \text{tr}(\epsilon_p) \alpha_f}{2\mu^2} \quad \frac{\partial \delta_{\gamma_p}}{\partial \lambda} = \frac{dim \text{tr}(\epsilon_p) \alpha_f}{2\mu} \quad (S.3)$$

The partial derivatives of the updated deformation gradient  $F'_p$  and its determinant  $\det(F'_p)$  with respect to  $S'_p$  are given by:

$$\frac{\partial F'_p}{\partial S'_p} = U'_p V'^{rT} \quad \frac{\partial \det(F'_p)}{\partial S'_p} = \det(F'_p) F'^{-T}_p \frac{\partial F'_p}{\partial S'_p} \quad (S.4)$$

And the partial derivatives of the Cauchy stress  $\sigma_p$  with respect to  $S'_p$  is:

$$\begin{aligned} \frac{\partial \sigma_p}{\partial S'_p} &= \frac{\partial}{\partial S'_p} \left( \frac{1}{\det(F'_p)} \hat{P}_p F'^{rT}_p \right) \\ &= -\frac{1}{\det(F'_p)^2} \frac{\partial \det(F'_p)}{\partial S'_p} \hat{P}_p F'^{rT}_p \\ &\quad + \frac{1}{\det(F'_p)} \left( \frac{\partial \hat{P}_p}{\partial S'_p} F'^{rT}_p + \hat{P}_p \frac{\partial F'^{rT}_p}{\partial S'_p} \right) \end{aligned} \quad (S.5)$$

where the partial derivative of  $\hat{P}_p$  with respect to  $S'_p$  is:

$$\frac{\partial \hat{P}_p}{\partial S'_p} = U'_p \left( \frac{2\mu(I_{dim} - \ln S'_p)}{(S'_p)^2} + \frac{\lambda(I_{dim} - \text{tr}(\ln S'_p))}{(S'_p)^2} \right) V'^{rT}_p \quad (S.6)$$

Therefore, we can obtain the partial derivative of the Cauchy stress  $\sigma_p$  with respect to  $\lambda$  and  $\mu$ :

$$\frac{\partial \sigma_p}{\partial \lambda} = \frac{\partial \sigma_p}{\partial S'_p} \frac{\partial S'_p}{\partial \lambda} \quad \frac{\partial \sigma_p}{\partial \mu} = \frac{\partial \sigma_p}{\partial S'_p} \frac{\partial S'_p}{\partial \mu} \quad (S.7)$$

These derivatives confirm the differentiability of our constitutive model.

## S.III. GOAL-CONDITIONED SOFT ACTOR-CRITIC

Goal-conditioned Reinforcement learning (GRL) algorithms optimise for a policy  $\pi(\bar{a}|o)$  to take actions that maximise the future goal-conditioned return  $R = \sum^T \mathbb{E}_{\bar{a} \sim \pi, o' \sim f(o, \bar{a})} [r(o', \hat{g})]$ , where  $T = 1$  with a single temporal-abstracted action  $\bar{a} = \theta$  [2], [3]. The soft actor-critic (SAC) method is an off-policy stochastic actor-critic RL method that maintains a so-called soft Q function,  $q(o, \hat{g}, \bar{a})$ , which estimates the entropy-augmented future expected return, and a stochastic policy that produces the mean and standard deviations of the action distribution (Gaussian) that maximising the following objective with data batches drawn from a replay buffer  $\mathcal{D}$  [4]:

$$\begin{aligned} J(\pi) &= \mathbb{E}_{o, \hat{g} \sim \mathcal{D}} \mathbb{E}_{\bar{a} \sim \pi} \left[ \min \{ q_1^\pi(o, \hat{g}, \bar{a}), q_2^\pi(o, \hat{g}, \bar{a}) \} \right. \\ &\quad \left. - \alpha_{sac} \log \pi(\bar{a}|o, \hat{g}) \right] \end{aligned}$$

where  $\alpha_{sac}$  is a weight that controls the importance of maximising the policy entropy during policy optimisation, automatically updated during runtime following [4]. To reduce value overestimation, we maintain two q networks following [5] and take the minimal value in policy optimisation. The objective of both critic networks in this one-step POGCMDP is to minimise the following mean-square error:

$$J(q^\pi) = \mathbb{E}_{o, \hat{g}, \bar{a}, r \sim \mathcal{D}} (q^\pi(o, \hat{g}, \bar{a}) - r)^2$$

As indicated in [3], goal-relabelling helps to improve sample efficiency for goal-conditioned RL. Therefore, for every interaction experience  $(o, \hat{g}, \bar{a}, r, o')$  collected during training, we duplicate the experience, replace the desired goal with the achieved goal (i.e., the resultant observation  $o' \rightarrow \hat{g}$ ), recompute the goal-conditioned reward, and append the relabelled experience into the replay buffer.

**Implementation details:** To process point cloud data, we employ the PointNet network architecture for a shared encoder for all q and policy networks [6]. For the q networks, the encoded feature is passed to two MLP layers of size 1024 and 512 with batch normalisation and ReLU activation, with dropout (0.4) applied to the second layer, and then passed to the last MLP layer of size 256, outputting the q value. For the policy network, the encoded feature is passed to three MLP layers of sizes 1024, 256 and 256 with ReLU activation and then two separate MLP layers of size 256 that output

the mean (activated by the hyperbolic tangent function) and the logarithm of the standard deviation of a Gaussian policy distribution of the action's dimension. The parameters of the PointNet encoder are only updated through the critic loss function, while the policy objective is used to update the MLP layers only.

#### S.IV. ADDITIONAL FIGURES OF GRADIENTS AND LOSS LANDSCAPES

Figure S.1 shows how the gradients of the key MLS-MPM variables, physics parameters, and actions evolve from the last global timestep to the initial timestep as well as how they evolve in the substep scale for 100 substeps (1 global timestep = 20 substeps). These gradients are generated by simulating the trajectory produced by the skill prior and using the EMD loss to compute the loss with a soil-digging manipulation target.

It is clear from the red lines (with no gradient operations) that the scales of the gradients grow sharply while back-propagating within 100 substeps (5 global steps) and eventually result in *NaNs* or numerical infinities. On the other hand, with the three gradient operations, these gradients can be kept within a reasonable scale ( $[-5, 5]$ ), as shown by the orange, blue, and green lines, corresponding to the clipping, dynamic scaling, and normalisation operations, respectively. This indicates the effectiveness of the three techniques in constraining the gradient scales of the key MLS-MPM variables, thus ensuring stable gradients for optimising the physics parameters and actions.

Figure S.2 shows the loss landscapes and the approximate gradients of the system identification task over Poisson's ratio and sand friction angle dimensions (top) as well as the second, third and fourth skill parameters for the first digging task (bottom). These results are also obtained with sampling resolutions 20, 40 and 60. Similar to the analyses in subsection III-F, it can be seen that HMD tends to produce smoother loss landscapes, indicated by the observation that the HMD gradients are less noisy.

#### S.V. TRAINING STATISTICS OF THE REINFORCEMENT LEARNING AGENT

Figures S.3, S.4, and S.5 present the detailed statistics of the goal-conditioned SAC agent on the three soil digging tasks. In three tasks, the increased actor losses (estimated Q values), the decreased critic losses and the stabilised policy entropies indicate that the algorithms have converged. The EMD and height map losses, however, indicate that the algorithm could only find good solutions in task 1 while it failed at the other two tasks (this was also observed in Figure 10a).

#### REFERENCES

- [1] G. Klár, T. Gast, A. Pradhana, *et al.*, "Drucker-Prager elastoplasticity for sand animation," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 1–12, 2016.
- [2] R. S. Sutton, "Reinforcement learning: An introduction," *A Bradford Book*, 2018.

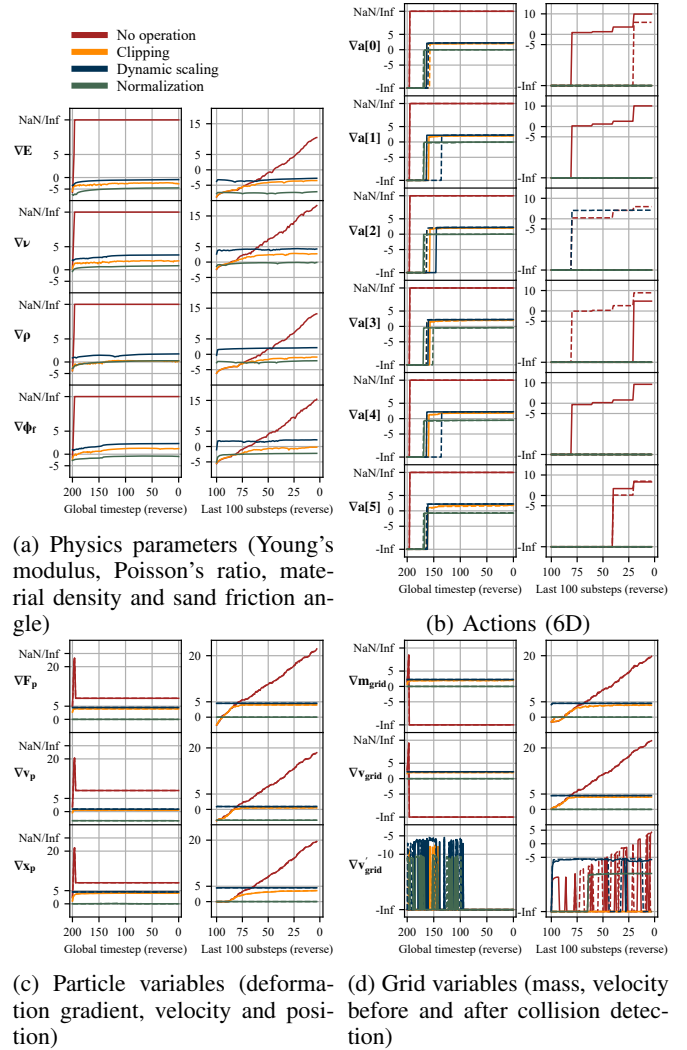
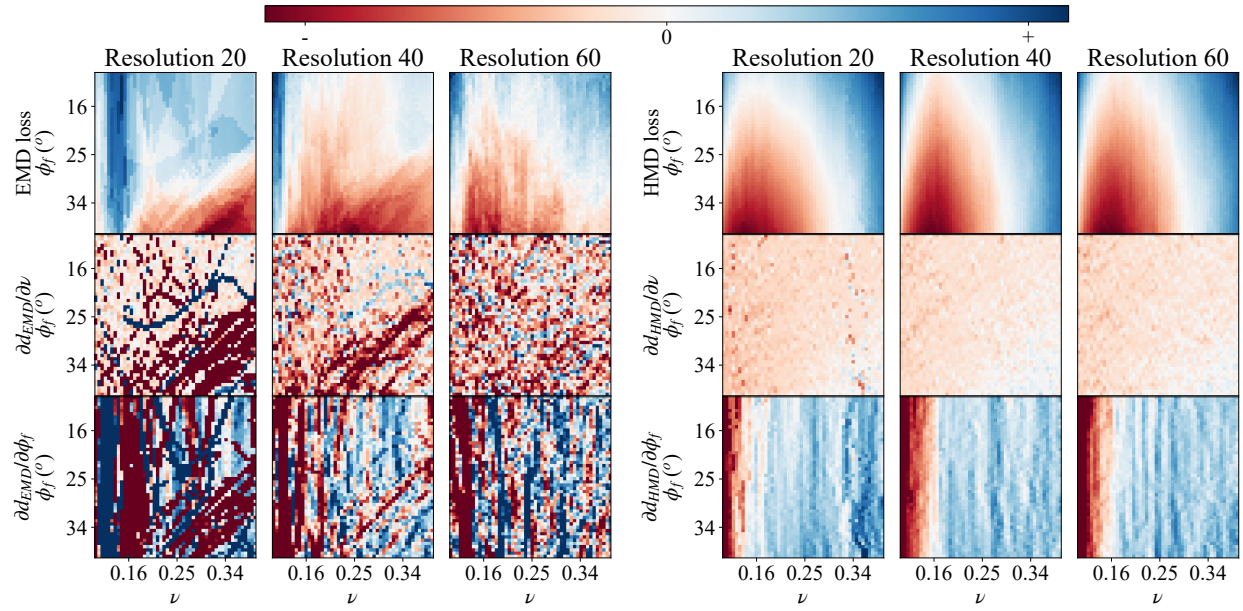
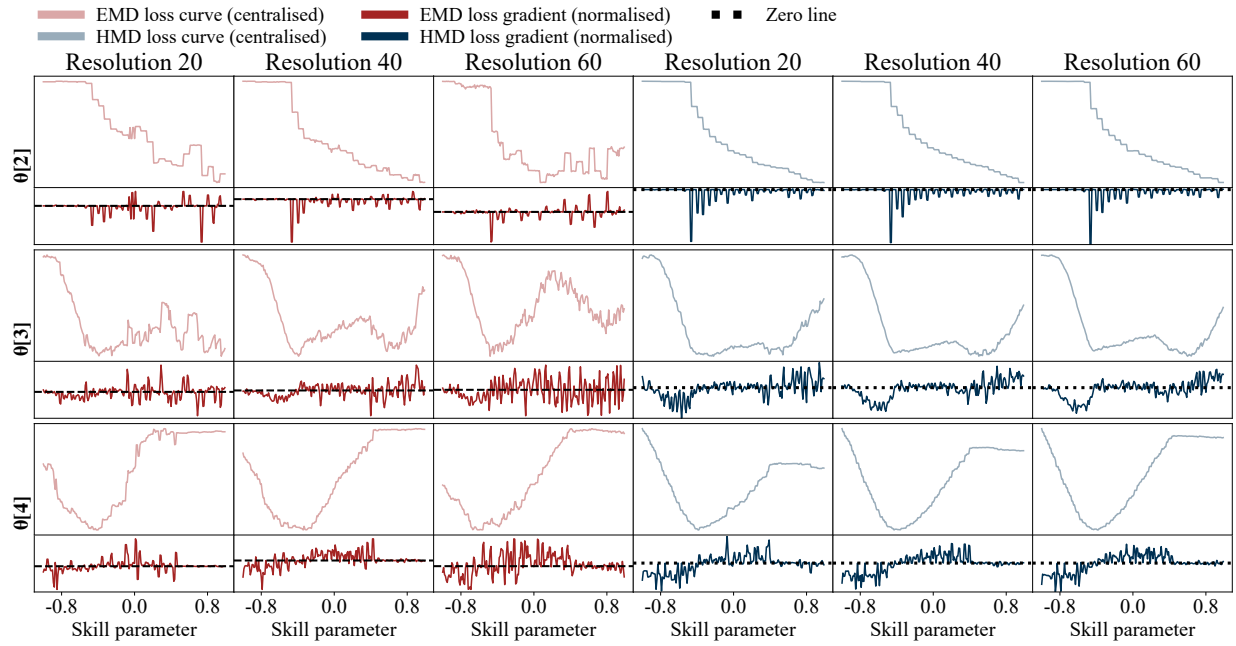


Fig. S.1: The log scale (base 10) of the gradients of the key MLS-MPM variables, physics parameters, and actions from the last global timestep to the initial timestep (left side of each subfigure) as well as in the substep scale for 100 substeps (right side of each subfigure). A global timestep = 20 substeps. Solid lines represent the maximal value and dash lines the minimal value of the gradient vector.

- [3] M. Andrychowicz, F. Wolski, A. Ray, *et al.*, "Hindsight experience replay," *Advances in neural information processing systems*, vol. 30, 2017.
- [4] T. Haarnoja, A. Zhou, K. Hartikainen, *et al.*, "Soft actor-critic algorithms and applications," *arXiv preprint arXiv:1812.05905*, 2018.
- [5] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International conference on machine learning*, PMLR, 2018, pp. 1587–1596.
- [6] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.



(a) System identification task. On Poisson's ratio and sand friction angle.



(b) Skill optimisation task. On skill parameter 2, 3 and 4.

Fig. S.2: Loss landscapes and their gradient distributions.

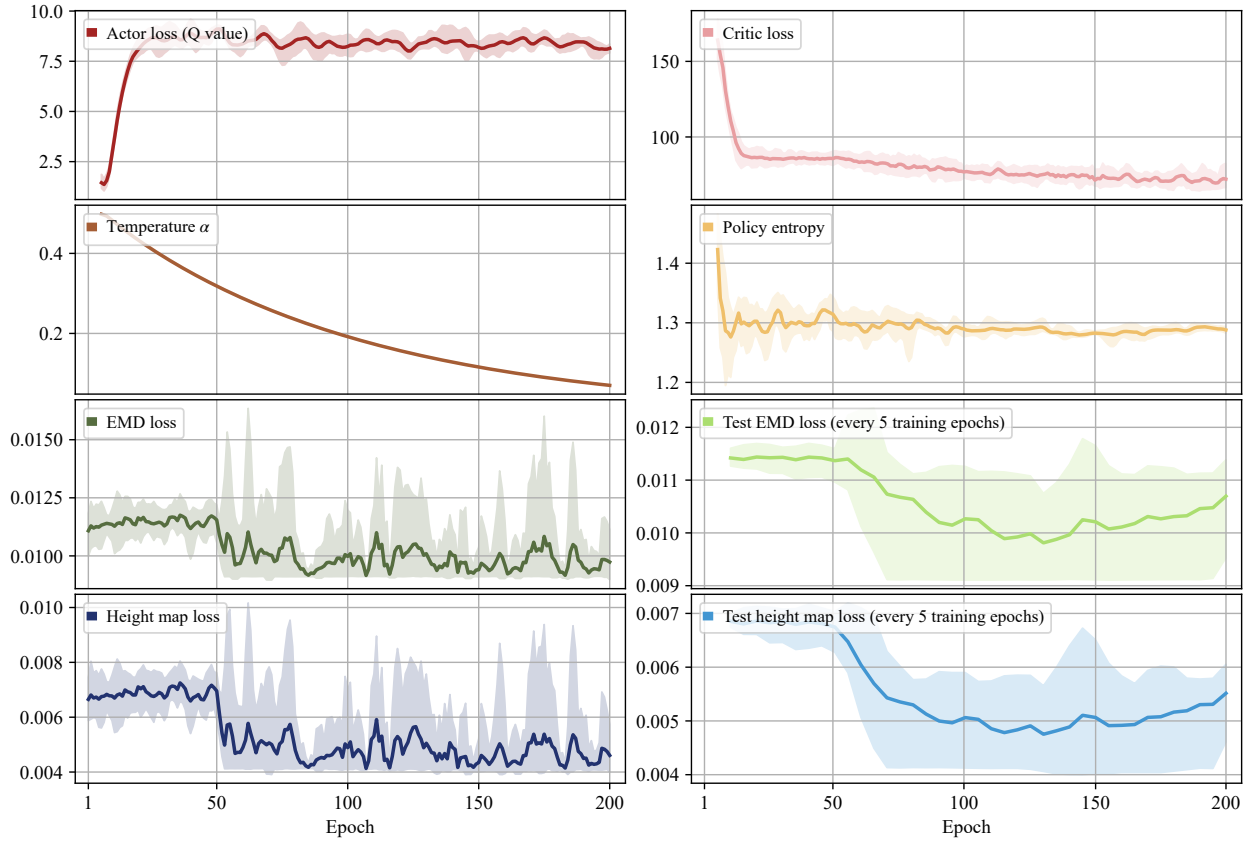


Fig. S.3: Goal-conditioned SAC training statistics on soil task 1.

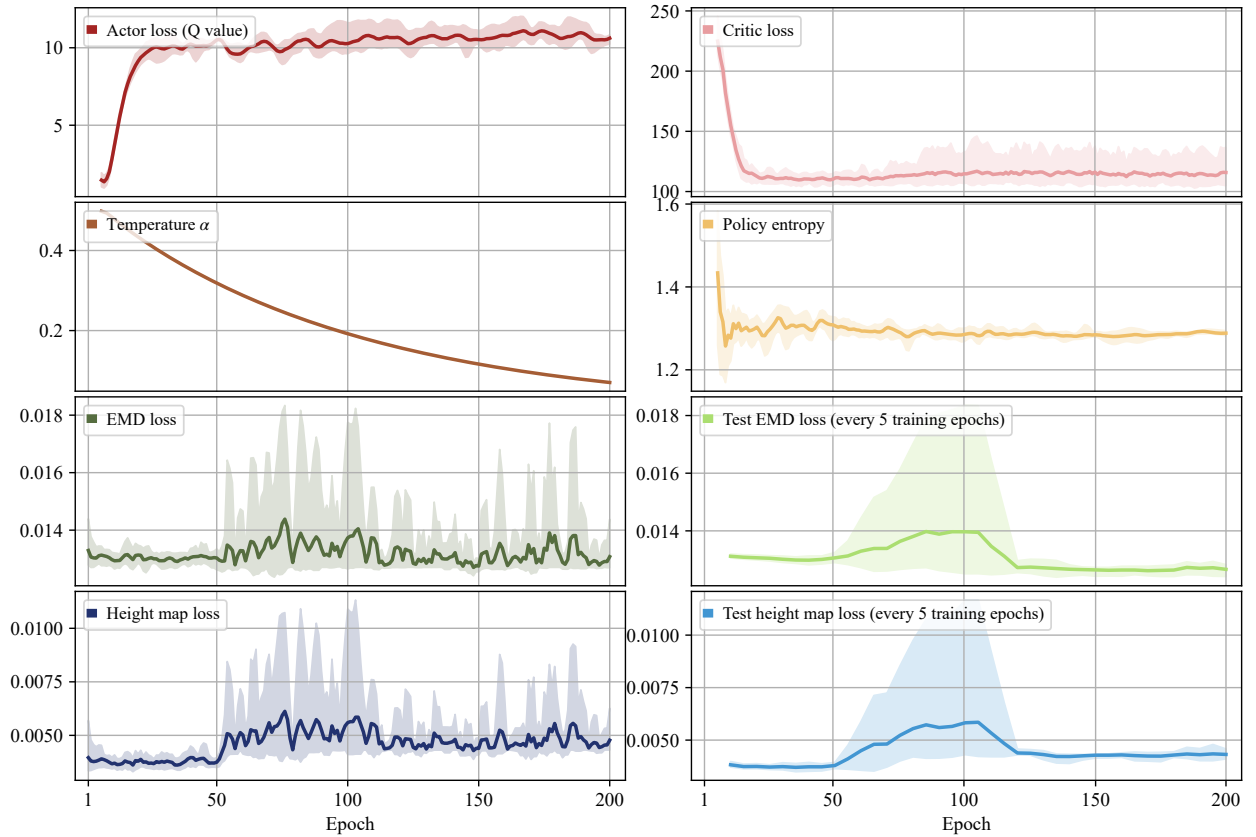


Fig. S.4: Goal-conditioned SAC training statistics on soil task 2.

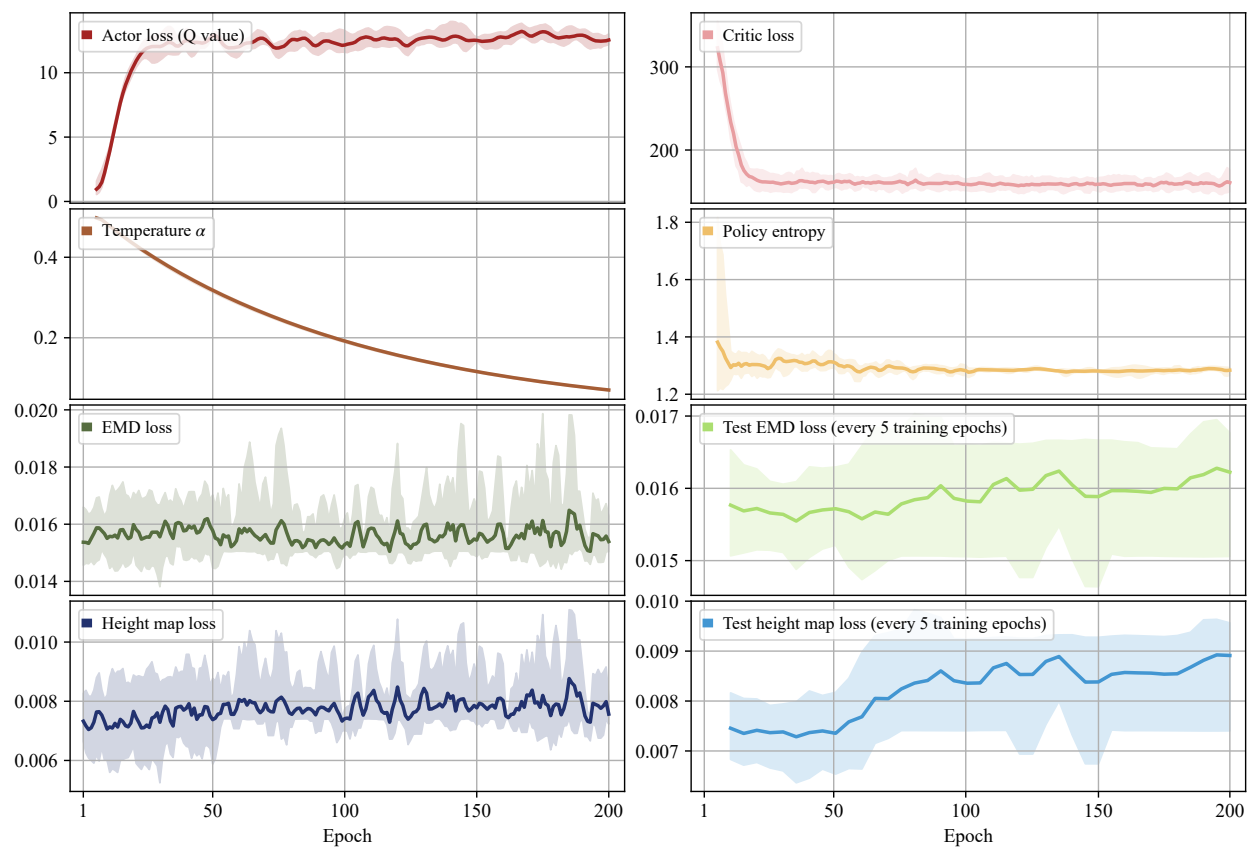


Fig. S.5: Goal-conditioned SAC training statistics on soil task 3.