Current primary challenges

To develop a generalist robot policies with 3D spatial intelligence

- Cameras, sensors not aligned due to different mount location on the robot
- Different robots have different action movement characteristics to accomplish diverse tasks, due to different degrees of freedom, motion controllers, workspace configurations

Solutions

- Convert the input 2D images into 3d representation
- Use adaptive grids method to directly output action tokens into the 3d cubes

How to achieve

Convert 2d image into 3d space

1. Use ZoeMap to convert the 2d image in to depth map
2. Use back projection π^{-1} to map all pixels in the 2d image into 3d space(3d coordinates)
3. Use SigLIP to get the semantic meaning from the visual representations $X \in \mathbb{R}^{d \times h \times w}$, h and w are the coordinates and d represents the property of the pixel, then calculate the 3D postion as well $P \in \mathbb{R}^{3 \times h \times w}$
4. The egocentric 3D positions P are then encoded into 3D position embeddings $P' \in \mathbb{R}^{d \times h \times w}$ through a sinusoidal function $\gamma(\cdot)$ following by a learnable MLP.
5. The 3d spatial representation $O_{3d} \in \mathbb{R}^{d \times h \times w}$ is by adding P' and X

$$O_{3d} = X + P' = X + MLP(\gamma(P)) \tag{1}$$

Implementation of Adaptive Grids
Setups: The goal here is to translate continuous robot actions to discrete

grids(cubes) that are represented as tokenized classes for prediction. For a single-arm robot the actions consist of seven dimensions for movement a = {x, y, z, roll, pitch, yaw, grip}. Split into three parts:

$$a = \{a_{\text{trans}}, a_{\text{rot}}, a_{\text{grip}}\}$$

where $a_{\text{trans}} = \{\phi, \theta, r\}$ represent the translation movements ΔT, $a_{\text{rot}} = \{\text{roll}, \text{pitch}, \text{yaw}\}$ denotes rotation movement ΔR and $a_{\text{grip}} = \{\text{grip}\}$ consists of two discrete tokens open, close.

Implementations:

1. Normalize each action variable into [-1, 1], and use a Gaussian distribution $\mathcal{N}(\mu^a, \Sigma^a)$

2. Split the continuous actions into M intervals $G_{i=1,...,M} = \{(a_1 = -1, a_2), ..., [a_{M-1}, a_M = 1]\}$ with equal probability $\frac{1}{M}$

3. $$a_2, \ldots, a_M = \arg \min_{a_2,\ldots,a_M} \int_{a_i}^{a_{i+1}} f(x)dx - \frac{1}{M}, \quad i = 1, \ldots, M \qquad (2)$$

4. $M_{\text{trans}} = M_\phi \cdot M_\theta \cdot M_r$, $M_{\text{rot}} = M_{\text{roll}} \cdot M_{\text{yaw}} \cdot M_{\text{yaw}}$. The 3 output tokens we need to predict will be:

   $$E_a = \{E_{\text{trans}}, E_{\text{rot}}, E_{\text{grip}}\}$$

   where $E_{\text{trans}} \in \mathbb{R}^{d \times M_{\text{trans}}}$, $E_{\text{rot}} \in \mathbb{R}^{d \times M_{\text{rot}}}$, $E_{\text{grip}} \in \mathbb{R}^{d \times 2}$. (The d represents the vector that embed a movement type, how to use this: suppose a movement type A matches to the translation in E_{trans} #500 columns, then we will probably use an another lookup table that track what actual translation values(ϕ, θ, r numerical values) on column #500 but the paper glossed over this details)

Training

Pre-training

- Train SpatialVLA from Paligemma2 backbone, our job here is to train two parts of parameters, first comes from the MLP layer from equation (1). The second is to train the Embed matrix E_a. The prediction flow is:

1. $O_{3d} \rightarrow$ Paligemma2 $\rightarrow$ translation token prediction.
2. $O_{3d} + d_{\text{trans}} \rightarrow$ Paligemma2 $\rightarrow$ rotation token prediction.
3. $O_{3d} + d_{\text{trans}} + d_{\text{rot}} \rightarrow$ Paligemma2 $\rightarrow$ gripper token prediction.

Post-trainning

Spatial Embedding Adaption: Based on different robots setups we define new adaptive grids G_{new} and use trilinear interpolation with pre-trained action tokens E_a.