[THU'24] RDT-1B

1. Link: https://rdt-robotics.github.io/rdt-robotics/
2. Arthurs and institution: Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, Jun Zhu from THU

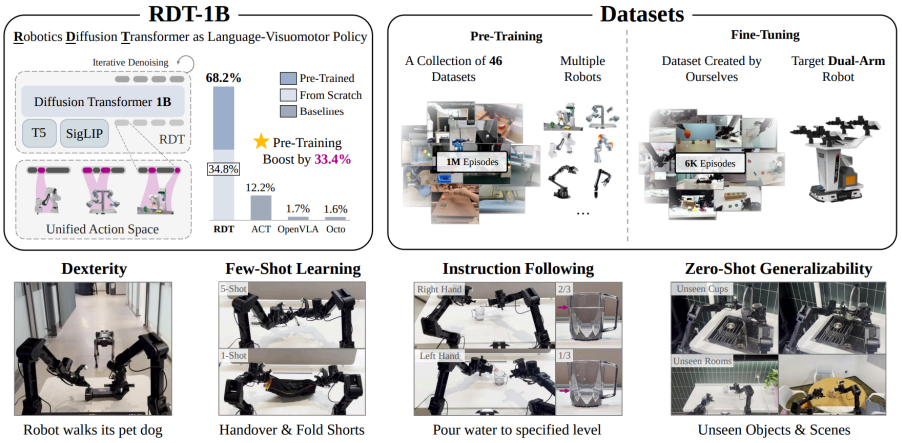


Figure 1: **Overview of Robotics Diffusion Transformer with 1B-Parameters (RDT-1B)**, a language-conditioned visuomotor policy for bimanual manipulation,with state-of-the-art generalizability to unseen scenarios (See App. H for metric calculation details).

TL;DR We present Robotics Diffusion Transformer with 1.2B parameters (RDT-1B), the largest diffusion-based foundation model for robotic manipulation with bimanual capability. **Todos**

3. Read DiT (diffusion transformer)
4. Read the pre-train paper， Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018

Thoughts and critisims

1. The authors provides a very detailed data recipe and training procedure, which helps a lot.
2. through the fine-tuning, some tasks are not usefull at all.
3. the idea of Unified Action Space is interesting, but the arthurs it not clear about its goal, neither knows how to test the effectiveness of the module. Can we elaborate it a bit? I believe a good robot state encoder could be tested if we examine given 2 configurations of robots, the encoder should outputs same features if their eef pose are the same.

Related works

VLA

Directly predict actions --> discretization of action spaces--> quantization errors and uncoordinated behaviors --> diffusion models

heterogeneous data

physical structure and the action space can vary greatly across different robots

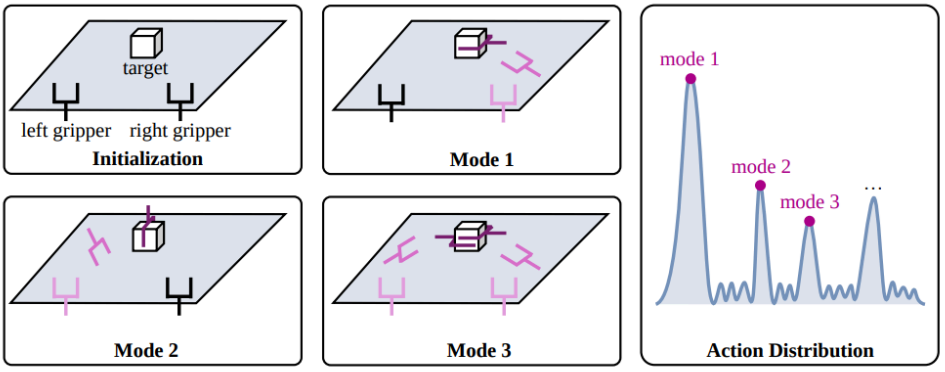
1. restrict themselves to a subset of robots with similar action spaces
2. retain a subset of inputs sharing the same structure

Contributions

Key concepts

Multimodality of actions

Given a task description, the robot has multiple choices for doing that.



(b) Illustration of multi-modality

1. Encoding of Heterogeneous Multi-Modal Inputs
    1. low-dim inputs
        1. proprioception, the action chunk, and the control frequency
        2. use MLPs with Fourier features to capture the high frequency changes in low-dim space
    2. image
        1. image-text-aligned pre-trained vision encoder, SigLIP
    3. language: T5-XXL (Raffel et al., 2020)

RDT

1. overview

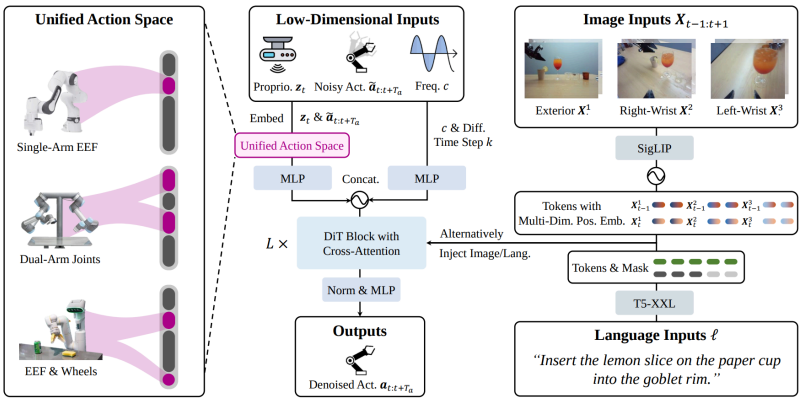


Figure 3: **RDT framework.** Heterogeneous action spaces of various robots are embedded into a unified action space for multi-robot training. **Inputs:** proprioception $\boldsymbol{z}_t$, noisy action chunk $\tilde{\boldsymbol{a}}_{t:t+T_a}$, control frequency c, and diffusion time step k, acting as denoising inputs; image inputs ($T_{\text{img}} = 2$ and $\boldsymbol{X}_\cdot = \{\boldsymbol{X}_\cdot^1, \boldsymbol{X}_\cdot^2, \boldsymbol{X}_\cdot^3\}$ denotes a set of images from exterior, right-wrist, and left wrist cameras) and language inputs, acting as conditions. **Outputs:** denoised action chunk $\boldsymbol{a}_{t:t+T_a}$.

2. sampling

When making a decision with diffusion policies, we first sample a totally noisy action $\boldsymbol{a}_t^K \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and then perform $K \in \mathbb{N}^+$ denoising steps to denoise it to a clean action sample $\boldsymbol{a}_t^0$ from $p(\boldsymbol{a}_t|\ell, \boldsymbol{o}_t)$:

$$\boldsymbol{a}_t^{k-1} = \frac{\sqrt{\bar{\alpha}^{k-1}}\beta^k}{1-\bar{\alpha}^k}\boldsymbol{a}_t^0 + \frac{\sqrt{\alpha^k}(1-\bar{\alpha}^{k-1})}{1-\bar{\alpha}^k}\boldsymbol{a}_t^k + \sigma^k \boldsymbol{z}, \quad k = K, \dots, 1, \tag{1}$$

where $\{\alpha^k\}_{k=1}^K, \{\sigma^k\}_{k=1}^K$ are scalar coefficients pre-defined by a noise schedule (Nichol & Dhariwal, 2021). Here, $\beta^k := 1 - \alpha^k$, and $\bar{\alpha}^{k-1} := \prod_{i=1}^{k-1} \alpha^i$, $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ if $k > 1$, else $\bar{\alpha}^{k-1} = 1, \boldsymbol{z} = \mathbf{0}$.

1.

3. use 2 mlps in decoder to improve non-linear robot actions
4. use cross-attention to accommodate conditions of images and languages

Implementation details

Action chunking

1. cumalative error in action prediction
2. robot action drifting
3. replace a_t by $a_{t:t+T}$

ve model $p(\boldsymbol{a}_{t:t+T_a}|\ell, \boldsymbol{o}_t)$, where $\boldsymbol{a}_{t:t+T_a} := (\boldsymbol{a}_t, \dots, \boldsymbol{a}_{t+T_a-1})$

4. predict multiple action in one-shot

Action encoding

1. encode state and action chunks into token space by shared MLP

Index Range	Element Index	Mapped Physical Quantity
[0, 10)	0–9	Right arm joint positions
[10, 15)	10–14	Right gripper joint positions
[15, 25)	15–24	Right arm joint velocities
[25, 30)	25–29	Right gripper joint velocities
[30, 33)	30–32	Right end effector positions
[33, 39)	33–38	Right end effector 6D pose
[39, 42)	39–41	Right end effector velocities
[42, 45)	42–44	Right end effector angular velocities
[45, 50)	45–49	Reserved
[50, 60)	50–59	Left arm joint positions
[60, 65)	60–64	Left gripper joint positions
[65, 75)	65–74	Left arm joint velocities
[75, 80)	75–79	Left gripper joint velocities
[80, 83)	80–82	Left end effector positions
[83, 89)	83–88	Left end effector 6D pose
[89, 92)	89–91	Left end effector velocities
[92, 95)	92–94	Left end effector angular velocities
[95, 100)	95–99	Reserved
[100, 102)	100–101	Base linear velocities
[102, 103)	102	Base angular velocities
[103, 128)	103–127	Reserved

Table 4: **Description of the unified action space vector.** For single-arm robot cases, its arm is mapped to the "right" arm. For a robot arm with only 6 DoF, its joint positions will be filled in the first 6 of the 10 corresponding positions. The same is true for other physical quantities.

2. encode frequency and diffusion step into token space by 2 mlp repectively
3. we concatenate the action and proprioception with a 0-1 vector indicating whether each dimension is padded before encoding them into the token space
    1. This can supplement the missing availability information and eliminate confusion.
4. add positional embedding

Data

1. pretrain data: 21T, 1M Trajs

Pre-Training Dataset	Sample Percentage (%)
RT-1 Dataset (Brohan et al., 2022)	9.00
TACO Dataset (Rosete-Beas et al., 2022)	1.99
JACO Play Dataset (Dass et al., 2023)	1.10
Cable Routing Dataset (Luo et al., 2023)	0.27
NYU Door Opening (Pari et al., 2021)	0.33
Viola (Zhu et al., 2022a)	0.40
Berkeley UR5 (Chen et al.)	1.06
TOTO (Zhou et al., 2023)	1.06
Kuka (Kalashnikov et al., 2018)	1.66
Language Table (Lynch et al., 2022)	3.32
Columbia Cairlab Pusht Real (Chi et al., 2023)	0.40
Stanford Kuka Multimodal Dataset (Lee et al., 2019)	1.83
Stanford Hydra Dataset (Belkhale et al., 2023)	0.80
Austin Buds Dataset (Zhu et al., 2022b)	0.23
Maniskill Dataset (Gu et al., 2023)	5.78
Furniture Bench Dataset (Heo et al., 2023)	2.36
UCSD Kitchen Dataset (Yan et al., 2023)	0.40
UCSD Pick And Place Dataset (Feng et al., 2023)	1.23
Austin Sailor Dataset (Nasiriany et al., 2022)	0.50
Austin Sirius Dataset (Liu et al., 2023)	0.80
BC Z (Jang et al., 2021)	6.91
UTokyo PR2 Opening Fridge (Oh et al., 2023)	0.30
UTokyo PR2 Tabletop Manipulation (Oh et al., 2023)	0.50
UTokyo Xarm Pick And Place (Matsushima et al., 2023)	0.33
UTokyo Xarm Bimanual (Matsushima et al., 2023)	0.03
Berkeley MVP (Radosavovic et al., 2022)	0.73
Berkeley RPT (Radosavovic et al., 2022)	1.00
KAIST Nonprehensile (Kim et al., 2023)	0.46
Tokyo U LSMO (Osa, 2022)	0.23
DLR Sara Grid Clamp (Padalkar et al., 2023)	0.03
Robocook (Shi et al., 2023)	1.66
Imperialcollege Sawyer Wrist Cam (RethinkRobotics)	0.43
Iamlab CMU Pickup Insert (Saxena et al., 2023)	0.83
UTAustin Mutex (Shah et al., 2023b)	1.29
Fanuc Manipulation (Zhu et al., 2023)	0.66
Play Fusion (Chen et al., 2023)	0.80
DROID (Khazatsky et al., 2024)	10.06
FMB (Luo et al., 2024)	1.39
Dobb·E (Shafiullah et al., 2023)	1.20
QUT Dexterous Manipulation (Federico Ceola, 2023)	0.46
Aloha Dataset (Zhao et al., 2023)	4.98
Mobile Aloha Dataset (Fu et al., 2024)	4.98
RoboSet (Kumar et al., 2024)	4.48
RH20T (Fang et al., 2023)	10.99
Calvin Dataset (Mees et al., 2022)	3.32
BridgeData V2 (Walke et al., 2023)	7.44

Table 5: The pre-training datasets and their corresponding weights.

    1. each with sampling weight

2. fine tuning data: 6K+ trajectories
    1. 300+ challenging tasks, from pick-and-place to plugging cables, even including writing math equations
    2. 100+ objects with rigid and non-rigid bodies of various sizes and textures
    3. 15+ different rooms with different lighting conditions.
3. pre-processing

    1. cleaning
        1. exclude re-petitive, failed episodes
        2. remove blank images
        3. exclude erroneously recorded velocities
        4. filter out overly short trajectories.
        5. Overlength trajectories will be downsampled to avoid unfairness.
    2. language
        1. remove illegal chars and extra spaces
        2. Capitalizing the beginging of sentence, adding period
    3. state
        1. unifying the action units
4. data augmentation
    1. image
        1. color jittering, image corruption
    2. state
        1. add Gaussian Noise with SNR=40dB
    3. intruction
        1. use GPT4

Hardware

ALOHA dual-arm robot from agilex

Software

1. real-time inference
    1. DPM-Solver++ (Lu et al., 2022)
    2. sample an action chunk from 100 steps to 5 steps
    3. 6 hz action chunk, 381 hz action per sec.
    4. on RTX 4090 24GB GPU.

Training

1. model: 1.2B
2. pretrain: 48 H100 for a month, 1M iterations in total
3. fine-tuning: 3 days, 130K steps
4. pytorch+deepspeed
5. TensorFlowDataset(TFD)
6. tricks
    1. use MSE to supervise tranining process
    2. remove static parts at begining during finetuning
    3. sampling angmented instruction
    4. didn't use classifier free guidence

Experiments

Questions

1. Can RDT zero-shot generalize to unseen objects and scenes?
2. How effective is RDT's zero-shot instruction-following capability for unseen modalities?
3. Can RDT facilitate few-shot learning for previously unseen skills?
4. Is RDT capable of completing tasks that require delicate operations?
5. Are large model sizes, extensive data, and diffusion modeling helpful for RDT's performance?

Scenes

1. 7 challenging tasks. 5-350 demos per task for fine-tuning

TASK NAME	DIMENSION	EXPLANATION
Wash Cup	Unseen Object ($\mathcal{Q}$**1**)	To wash one seen and two unseen cups with the faucet
Pour Water	Unseen Scene ($\mathcal{Q}$**1**)	To pour water into the cup in three unseen rooms
Pour Water-L-1/3	Instruction Following ($\mathcal{Q}$**2**)	To pour water into the cup with the **left hand** until **one-third** full
Pour Water-R-2/3	Instruction Following ($\mathcal{Q}$**2**)	To pour water into the cup with the **right hand** until **two-thirds** full
Handover	5-Shot Learning ($\mathcal{Q}$**3**)	To move the marker to the box, where handover is needed due to far distance
Fold Shorts	1-Shot Learning ($\mathcal{Q}$**3**)	To fold the shorts in half horizontally
Robot Dog	Dexterity ($\mathcal{Q}$**4**)	To push the joystick straight to control the robot dog to walk in a straight line