GROVE: A Generalized Reward for Learning Open-Vocabulary Physical Skill

Jieming Cui[1,2,*], Tengyu Liu[2,*], Ziyu Meng[3], Jiale Yu[4], Ran Song[3], Wei Zhang[3], Yixin Zhu[1,✉], Siyuan Huang[2,✉]

[1] Institute for Artificial Intelligence, Peking University, [2] State Key Laboratory of General Artificial Intelligence, BIGAI
[3] School of Control Science and Engineering, Shandong University, [4] Department of Automation, Tsinghua University

https://jiemingcui.github.io/grove/

CVPR Nashville JUNE 11-15, 2025

VLM-based Reward Design

ANY Text Description I

Iterate text enhancement

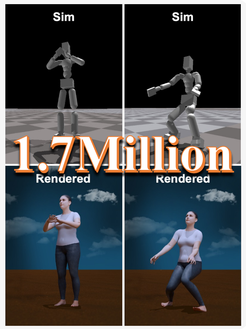
1. Describe one specific human pose
2. Ensure the verb and noun relate to a human limb.
3. Make the description fluent, concise, and unambiguous. ...

$$R_V(s, I) = \frac{CLIP(I) \cdot Pose2CLIP(s)}{\| CLIP(I) \|_2 \cdot \| Pose2CLIP(s) \|_2}$$

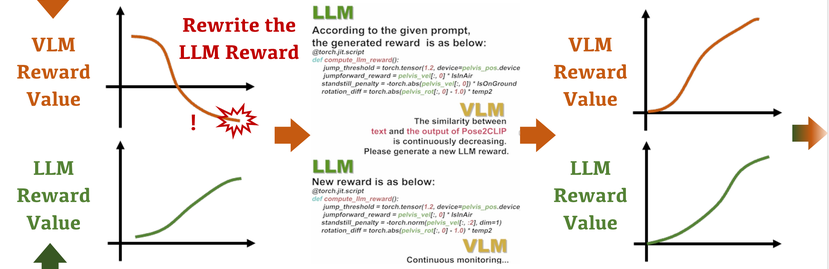
✓ We use the value of VLM-based reward as Fitness Score F of Reward Design Problem module

Agent State S
position & velocity
73d * history

Image Feature
512d
(CLIP VIT)



Reward Design Problem (RDP)



LLM-based Reward Design

1. You are a reward engineer trying to write reward functions to solve RL tasks ... 2. In the simulator, we define Z as the up-axis. 3. The reward function signature is ... 4. Make sure that the code is compatible with TorchScript (e.g., use torch tensor instead of numpy array) 5. The reward code´s input variables must contain only attributes of the provided environment class definition.

```
SMPL_BONE_ORDER_NAMES =
    "pelvis",
    "torso",
    "head",
    "right_upper_arm",
    "right_lower_arm",
    "right_hand",
    "left_upper_arm",
```

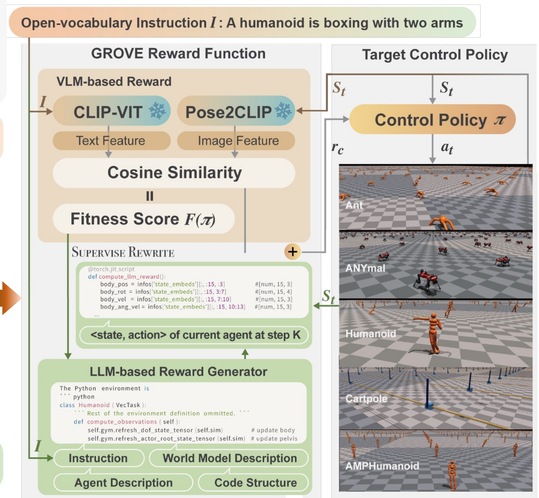
```
@torch.jit.script
def compute_llm_reward():
    body_pos = infos["state_embeds"][:,
        :21, :3]  # [num, 21, 3]
    body_rot = infos["state_embeds"][:,
        :21, 3:7]  # [num, 21, 4]
    body_vel = infos["state_embeds"][:,
        :21, 7:10]  # [num, 21, 3]
```

```
pelvis_pos, pelvis_rot, pelvis_vel,
    pelvis_ang_vel = body_pos[:, 0, :],
    body_rot[:, 0, :], body_vel[:, 0,
    :], body_ang_vel[:, 0, :]
torso_pos, torso_rot, torso_vel,
    torso_ang_vel = body_pos[:, 1, :],
    body_rot[:, 1, :], body_vel[:, 1,
    :], body_ang_vel[:, 1, :]
head_pos, head_rot, head_vel,
    head_ang_vel = body_pos[:, 2, :],
    body_rot[:, 2, :], body_vel[:, 2,
    :], body_ang_vel[:, 2, :]
...
```

Contributions

- ✓ GROVE, a generalized reward framework that combines LLM-generated precise constraints with VLM-based semantic evaluation.
- ✓ Pose2CLIP, a lightweight pose-to-semantic feature mapper that bridges the domain gap between simulation and natural images.
- ✓ Generate natural motions for arbitrary embodiments from open-vocabulary instructions.

Training Model and Main Results



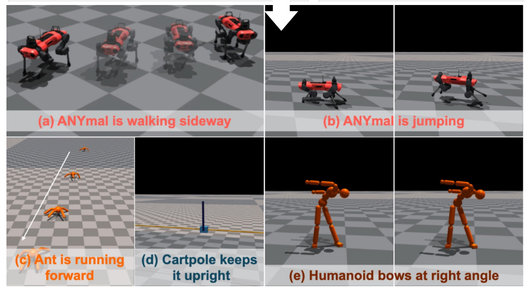
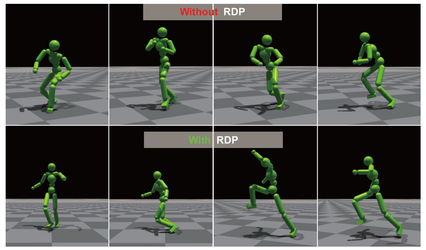


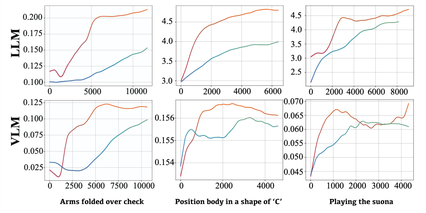
Table 1. **Quantitative evaluation of learned humanoid skills using the GROVE reward.**

	Completion ↑	Natural ↑	Smoothness↓	Physics↑	CLIP_S↑
AvatarCLIP [10]	1.621	4.793	0.411	5.821	22.105
TMR [31]	3.793	5.966	**0.374**	7.684	18.885
MoMask [8]	3.621	6.724	0.492	7.863	22.372
MotionGPT [13]	5.207	6.552	0.885	7.000	23.142
AnySkill [7]	6.108	5.938	0.486	8.168	23.925
Ours (+ CALM) [41]	**7.924**	**6.793**	0.488	**8.452**	**28.998**

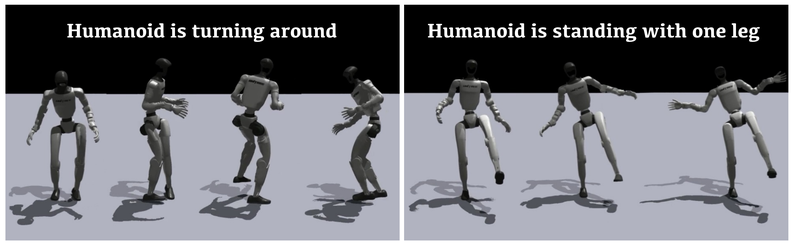
Table 4. **Quantitative results of the ablation study.**

	Completion ↑	Natural↑	Smoothness↓	CLIP_S↑	Training Time↓
VLM only [7]	6.108	5.938	0.486	23.925	59min
VLM + LLM	6.954	6.292	**0.457**	24.088	47min
LLM only w/o RDP [24]	6.622	5.907	0.582	22.977	20min
LLM only with RDP	6.650	6.177	0.599	24.099	19min
Pose2CLIP only	6.785	5.954	0.553	24.547	16min
Pose2CLIP + LLM w/o RDP	7.269	6.738	0.475	25.331	**7min**
Pose2CLIP + LLM	**7.924**	**6.793**	0.488	**28.998**	**7min**





Results on Humanoid Agent and Robot



GROVE enables a humanoid robot to act on open-vocabulary instructions.
Stay tuned for our upcoming real-world deployments!