A. Algorithmic Baselines

A.1. PPO

Proximal policy optimization (PPO) [30] is an on-policy algorithm that is designed to improve the stability and sample efficiency of policy gradient methods, which uses a clipped surrogate objective function to avoid large policy updates.

The policy loss is defined as:

$$L_\pi(\boldsymbol{\theta}) = -\mathbb{E}_{\tau\sim\pi}\left[\min\left(\rho_t(\boldsymbol{\theta})A_t,\right.\right. \\ \left.\left.\text{clip}\left(\rho_t(\boldsymbol{\theta}), 1-\epsilon, 1+\epsilon\right)A_t\right)\right], \tag{7}$$

where

$$\rho_t(\boldsymbol{\theta}) = \frac{\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\pi_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}, \tag{8}$$

and ϵ is a clipping range coefficient.

Meanwhile, the value network is trained to minimize the error between the predicted return and a target of discounted returns computed with generalized advantage estimation (GAE) [29]:

$$L_V(\boldsymbol{\phi}) = \mathbb{E}_{\tau\sim\pi}\left[\left(V_{\boldsymbol{\phi}}(\boldsymbol{s}) - V_t^{\text{target}}\right)^2\right]. \tag{9}$$

A.2. VAE

Variational autoencoders (VAE) [10] are reconstruction-based methods that encode observations $\boldsymbol{o}$ into latent variables $\boldsymbol{z}$ while enforcing a prior distribution, balancing reconstruction fidelity and regularization. The loss function of VAE is defined as

$$L_{\text{VAE}} = -\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{o})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{o}|\boldsymbol{z})] \\ + D_{\text{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{o})\|p_{\boldsymbol{\theta}}(\boldsymbol{z})), \tag{10}$$

where $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{o})$ is the encoder, $p_{\boldsymbol{\theta}}(\boldsymbol{o}|\boldsymbol{z})$ is the decoder, and D_{KL} is the Kullback–Leibler (KL) divergence.

A.3. SPR

SPR [33] is a dynamics modeling method that learns predictive latent representations by enforcing multi-step consistency between predicted and encoded future states. The loss function of SPR is defined as

$$L_{\text{SPR}} = \sum_{k=1}^{K} \| f_{\boldsymbol{\theta}}^{(k)}\left(\boldsymbol{z}_t, \boldsymbol{a}_{t:t+k-1}\right) \\ - \text{sg}(g_{\boldsymbol{\phi}}(\boldsymbol{o}_{t+k}))\|_2^2, \tag{11}$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operation. Here, $f_{\boldsymbol{\theta}}$ is the online dynamics model, such that $\boldsymbol{z}_{t+1} = f_{\boldsymbol{\theta}}(\boldsymbol{z}_t, \boldsymbol{a}_t)$. $g_{\boldsymbol{\phi}}$ is the target dynamics model whose parameters are an exponential moving average (EMA) of the online dynamics model parameters.

A.4. SimSiam

SimSiam [1] is a simple self-supervised learning method based on a Siamese network architecture, designed to learn meaningful representations without the need for negative sample pairs, large batches, or momentum encoders. The architecture consists of two identical networks that process two augmented views of the same input image. A key feature of SimSiam is the use of a stop-gradient operation, which prevents the network from collapsing by ensuring that gradients do not propagate to one of the branches. The objective is to maximize the similarity between the representations of the two views, which is achieved using the negative cosine similarity loss function.

The loss function used in SimSiam is given by:

$$L_{\text{SimSiam}} = \frac{1}{2}\left[-\frac{f_{\boldsymbol{\theta}}(\boldsymbol{x_1})\cdot f_{\boldsymbol{\theta}}(\boldsymbol{x_2})}{\|f_{\boldsymbol{\theta}}(\boldsymbol{x_1})\|_2\|f_{\boldsymbol{\theta}}(\boldsymbol{x_2})\|_2}\right] \tag{12}$$

where $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ represents the encoder network's output for the augmented view $\boldsymbol{x}$.

B. Benchmark Design

Table 1. Key reward terms utilized in *LimX-Oli-31dof-Velocity* task.

Term	Formulation	Weight
Linear velocity tracking	$\exp\left(-\frac{\|\boldsymbol{v}_{xy}-\boldsymbol{v}_{xy}^{\text{cmd}}\|^2}{2\sigma^2}\right)$	1.0
Angular velocity tracking	$\exp\left(-\frac{(\omega_z-\omega_z^{\text{cmd}})^2}{2\sigma^2}\right)$	0.5
Base height	$(h-h^\star)^2$	0.5
Linear velocity (z)	$\|v_z\|^2$	-2e-3
Angular velocity (x,y)	$\|\boldsymbol{\omega}_{xy}\|^2$	-0.15
Action smoothness	$\|\boldsymbol{a}_t - 2\boldsymbol{a}_{t-1} - \boldsymbol{a}_{t-2}\|^2$	-2.5e-3
Joint velocity	$\|\dot{\boldsymbol{q}}\|^2$	-1e-3
Joint acceleration	$\|\ddot{\boldsymbol{q}}\|^2$	-5e-7
Joint deviation	$\sum_j \|q_j - q_j^{\text{def}}\|$	-0.1
Joint power	$\|\boldsymbol{\tau}\|\|\dot{\boldsymbol{q}}\|^T$	-2.5e-7
Joint torque	$\|\boldsymbol{\tau}\|_2^2$	-4.0e-7
Joint position limits	$\sum_j \Delta_j$	-0.2
Joint velocity limits	$\sum_j \dot{q}_j$	-0.025

Let $\boldsymbol{q}$ denote the joint positions, $\dot{\boldsymbol{q}}$ the joint velocities, $\ddot{\boldsymbol{q}}$ the joint accelerations, $\boldsymbol{\tau}$ the joint torque, $\boldsymbol{v}_{xy}$ the base linear velocity, $\boldsymbol{\omega}_{xy}$ the base angular velocity on the xy-axis, h^* the expected base height, δ_1, δ_2 the roll and pitch of the waist joint, and Δ the the absolute value of the difference between the joint position and the soft limits. The following tables illustrate the reward terms and components of the proprioceptive states and the privileged states of the two tasks.

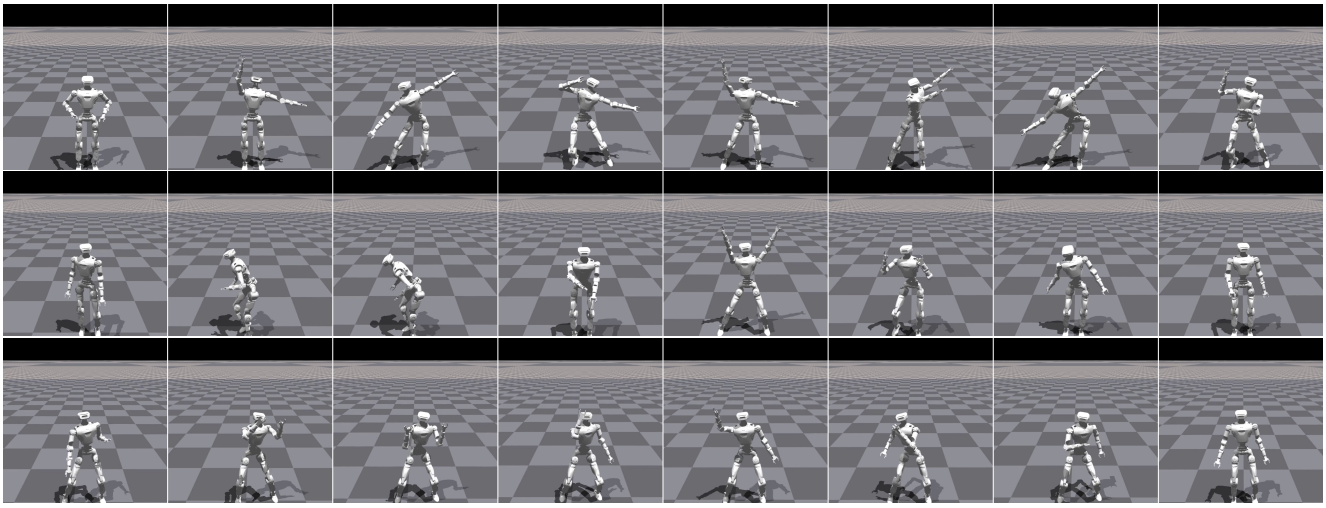

Figure 12. Example screenshots of the motion capture data.

Table 2. The details of the proprioceptive state and privileged state of the *LimX-Oli-31dof-Velocity* task. Here, we stack 5 consecutive proprioceptive states as the input of the policy encoder to ensure robustness.

Proprioceptive State	**Privileged State**
base_ang_vel (3x5) projected_gravity (3x5) velocity_commands (3x5) joint_pos (31x5) joint_vel (31x5) actions (31x5) gait (5)	base_lin_vel (3) base_ang_vel (3) projected_gravity (3) velocity_commands (3) joint_pos (31) joint_vel (31) actions (31) gait (5)

B.1. Velocity Tracking Task

B.2. Motion Imitation Task

C. Experimental Setup

C.1. PPO

PPO [30] is selected as the backbone RL algorithm for all the SRL methods. Table 5 illustrates the network architectures of the policy network and value network, and Table 6 lists the hyperparameters used for the two humanoid WBC tasks. Notably, these configurations remain fixed for all the experiments to isolate the effects of SRL methods.

C.2. PPO+PvP

For PvP, we utilize the root linear velocity relative to the world coordinate system as privileged information for contrastive learning, while the root orientation information is also involved in the motion imitation task. Accordingly, we

Table 3. Key reward terms utilized in *LimX-Oli-31dof-Mimic* task.

Term	**Formulation**	**Weight**
Position tracking	$\exp\left(-\frac{\|\boldsymbol{q}-\boldsymbol{q}^{\text{ref}}\|^2}{2\sigma^2}\right)$	2.0
Feet distance tracking	$\exp\left(-\frac{\|d-d^{\text{ref}}\|^2}{\sigma}\right)$	0.5
Waist pitch orientation tracking	$\exp\left(-\sum_{i=1}^{2}\|\delta_i-\delta_i^{\text{ref}}\|\right)$	0.5
Action rate	$\|\boldsymbol{a}_t-\boldsymbol{a}_{t-1}\|^2$	-0.001
Joint velocity	$\|\dot{\boldsymbol{q}}\|^2$	-0.5e-3
Joint acceleration	$\|\ddot{\boldsymbol{q}}\|^2$	-1.0e-7
Joint Torque	$\|\boldsymbol{\tau}\|_2^2$	-1.0e-5
Joint position limits	$\sum_j \Delta_j$	-1.0
Joint torque limits	$\sum_j \tau_j$	-0.01
Joint velocity limits	$\sum_j \dot{q}_j$	-0.2

Table 4. The details of the proprioceptive state and privileged state of the *LimX-Oli-31dof-Mimic* task.

Proprioceptive State	**Privileged State**
	base_lin_vel (3)
	base_ang_vel (3)
	base_pos_z (1)
base_ang_vel (3)	body_mass (40)
projected_gravity (3)	base_quat (6)
joint_pos (31)	projected_gravity (3)
joint_vel (31)	velocity_commands (3)
actions (31)	joint_pos (31)
mimic reference (69)	joint_vel (31)
	actions (31)
	previous actions (31)
	mimic reference (69)

Table 5. The architectures of the policy and value network, which remain fixed for all the experiments. Here, "O. D." represents "On-demand".

Part	Policy Network	Value Network
Encoder	Linear(O. D., 512) ELU() Linear(512, 256) ELU() Linear(256, 128)	Linear(O. D., 512) ELU() Linear(512, 256) ELU() Linear(256, 128)
Head	Linear(128, 128) ELU() Linear(128, 31)	Linear(256, 128) ELU() Linear(128, 1)

Table 6. The PPO hyperparameters for the two tasks, which remain fixed for all experiments.

Hyperparameter	Value
Reward normalization	Yes
LSTM	No
Maximum Episodes	30000
Episode steps	32
Number of workers	1
Environments per worker	4096
Optimizer	Adam
Learning rate	1e-3
Learning rate scheduler	Adaptive
GAE coefficient	0.95
Action entropy coefficient	0.01
Value loss coefficient	1.0
Value clip range	0.2
Max gradient norm	0.5
Number of mini-batches	4
Number of learning epochs	5
Desired KL divergence	0.01
Discount factor	0.99

attach the zero mask to the proprioceptive state in the whole training to align its dimension with the privileged state. For the loss coefficient, we run an initial hyperparameter search over $\{0.1, 0.5, 1.0\}$ and use 0.5 as the baseline setting.

C.3. PPO+SimSiam

For SimSiam [1], we run an initial hyperparameter search over the loss coefficient $\{0.1, 0.5, 1.0\}$ and the data augmentation operation {random_masking, gaussian_noise, random_amplitude_scaling, identity_mapping}. Then the loss coefficient of 0.5 and random_masking, identity_mapping operation are used as the baseline settings. This is because the proprioceptive state is subjected to the domain randomization in the simulator, which can also be considered data augmentation.

C.4. PPO+SPR

For SPR [33], we run an initial hyperparameter search over the loss coefficient $\{0.1, 0.5, 1.0\}$, the data augmentation operation {random_masking, gaussian_noise, random_amplitude_scaling, identity_mapping}, the number of prediction steps $\{1, 5, 10, 15\}$, and whether to use a average loss. Then the loss coefficient of 0.5, gaussian_noise operation, and the number of prediction steps of 5 are used as the baseline settings.

C.5. PPO+VAE

For VAE [10], we simply run an initial hyperparameter search over the loss coefficient $\{0.1, 0.5, 1.0\}$ and use 0.1 as the baseline setting.

D. Ablation Studies

D.1. Comparison with Teacher-Student Distillation

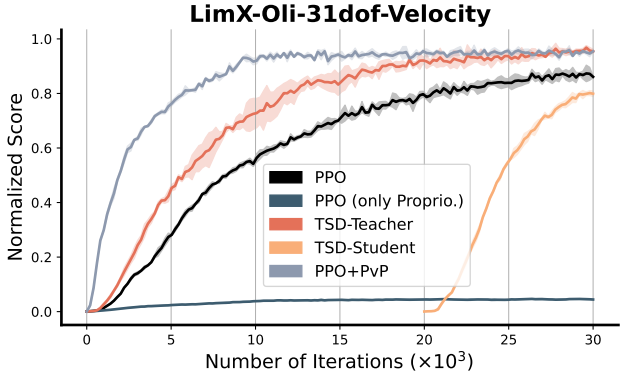

Figure 13. Training progress comparison between the teacher-student distillation method and PvP in the *LimX-Oli-31dof-Velocity* task. The solid line and shaded region denote the mean and standard deviation, respectively.

Both teacher-student distillation (TSD) [12, 17] and PvP aim to leverage privileged information to guide representation learning, thereby reducing the complexity of learning for high-dimensional humanoid control. However, TSD has several critical limitations. The student's performance ceiling is strictly constrained by the teacher's quality and inductive biases, and any sub-optimality or observation mismatches in the teacher are directly inherited by the student. As shown in the experiments below, the student fails to match the teacher's performance and exhibits a significant gap after distillation. Moreover, strict alignment objectives can over-regularize the student and suppress exploration, causing the student to collapse toward conservative or averaged behavior rather than discovering alternative (potentially better) solutions. Finally, the teacher pre-training stage is often compute-intensive, limiting reproducibility and rapid iteration as tasks scale up.

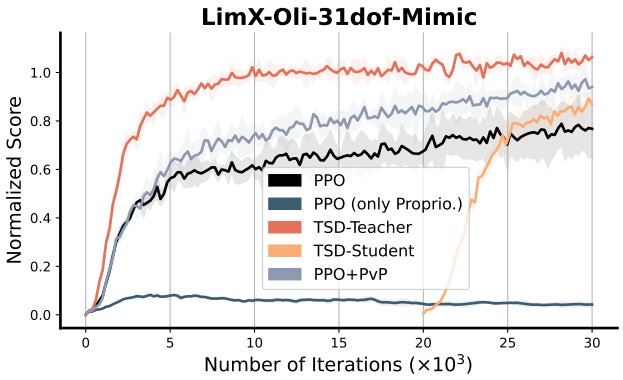

Figure 14. Training progress comparison between the teacher-student distillation method and PvP in the *LimX-Oli-31dof-Mimic* task. The solid line and shaded region denote the mean and standard deviation, respectively.

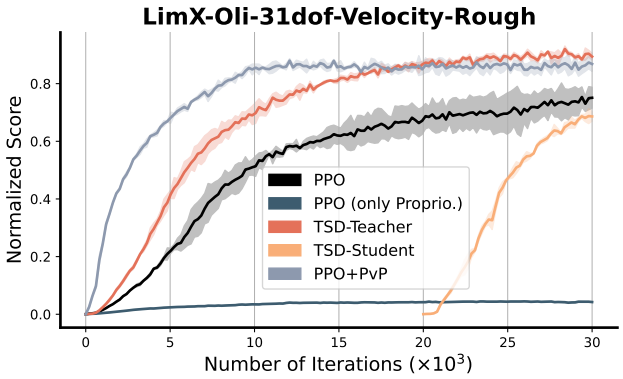

Figure 15. Training progress comparison between the teacher-student distillation method and PvP in the *LimX-Oli-31dof-Velocity-Rough* task. The solid line and shaded region denote the mean and standard deviation, respectively.

In contrast, PvP facilitates a synergy between representation learning and policy optimization. Unlike the disjointed two-stage pipeline of TSD, PvP integrates contrastive representation learning directly into the RL framework. This enables the latent space to evolve alongside the policy's exploration, rather than merely replicating a static teacher. By optimizing a contrastive objective that captures structural invariants, the representation and policy mutually enhance each other. The policy benefits from a more discriminative latent space, while the representation adaptively prioritizes task-relevant features discovered during training.

To demonstrate PvP's advantage over the TSD method, we conduct ablation experiments on the two designed humanoid WBC tasks. As illustrated in Figure 13 and Figure 14, there is a significant performance gap between the teacher and student policies, and consistently outperforms the student policy. Notably, the PPO agent that relies solely on proprioceptive state information fails to learn in both tasks, highlighting the need for privileged information to support learning.

D.2. Evaluation on More Diverse Tasks

To evaluate PvP on more diverse scenarios and robot platforms, we introduce two new tasks: *LimX-Oli-31dof-Velocity-Rough* and *Unitree-G1-29dof-Velocity*. The former is a variant of the previously defined velocity-tracking task that introduces rough terrain, while the latter uses a different robot platform. As illustrated in Figure 15 and Figure 16, PvP excels in the two tasks in terms of final performance and time cost.

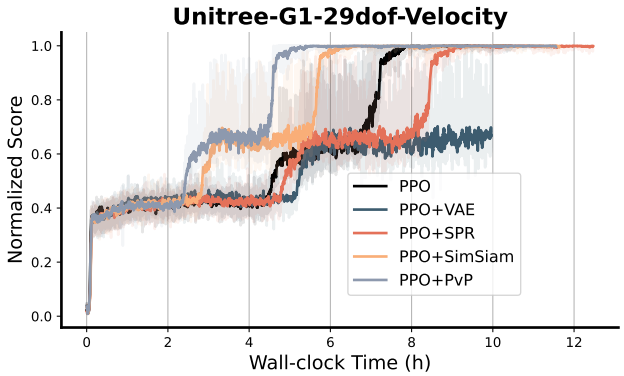

Figure 16. Training progress comparison between the vanilla PPO agent and its combination with four SRL methods on the *Unitree-G1-29dof-Velocity* task. The solid line and shaded region denote the mean and standard deviation, respectively.

D.3. Impact of the λ on PvP

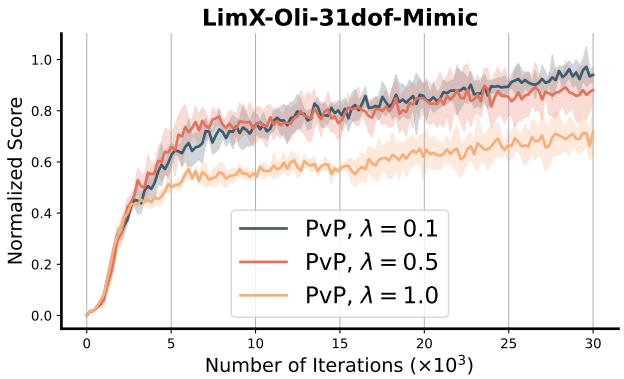

Figure 17. Performance comparison of the PvP method with different weighting coefficients on the *LimX-Oli-31dof-Mimic* task. The solid line and shaded region denote the mean and standard deviation, respectively.

Furthermore, we investigate the impact of the weighting coefficients on the performance of our PvP method. As shown in Figure 17, $\lambda = 0.1$ achieves the best final performance and sample efficiency, indicating that PvP benefits from a relatively smaller λ.