Unity RL ztest1.0 技术报告（in detail）

1. 游戏设计初衷

1.1 游戏设计初衷

> 将基于RL训练的具身智能模型作为游戏NPC，在不断优化迭代玩家性能和具身智能模型性能的 **“对抗过程”** 中，即能增加了游戏的趣味性（物理交互的真实感以及不可预测性）；也能为**具身智能算法迭代**提供了很好的环境

1.2 《人机追逐对抗》 (解谜/恐怖类) 游戏设计初步想法

核心玩法：

- 玩家是一个人类，被一群这种“走路姿势诡异”的机器人追杀。
  开发重点：
- RL具身智能模型训练：灵活适应复杂地形
- 自主寻路：需要写一个简单的脚本，根据 NavMesh 计算路径，然后算出 vr/wr 给机器人，让它自动追玩家。
- 环境-氛围：灯光调暗；给机器人加两只发红光的眼睛；走路音效（包含远近）
- 技能设计-考验机器人NPC的RL能力训练；例如：“玩家技能-投放障碍物道具”
- 环境布局优化-考验机器人NPC的RL能力训练（老建筑内；有破坏的墙角可以爬出去，有的房间有不止一个出口；有点房间单出口，但是较大且障碍物多）

1.3 《人机追逐对抗 0.0》设计

- Unity-RL：NPC设计基于OpenLoong大体型机器人，**实现双足机器人从“离散步态切换”到“连续平滑过渡”的能力，并可以自由控制机器人的行走速度**
- 环境设计：简单长廊+几个拐角+楼梯或简单杂物
- 玩家/摄像头放置：放置可以”wasd方向控制+space跳跃+鼠标视角变化“玩家+摄像头

2. Baseline框架解读

> 参考https://github.com/loongOpen/Unity-RL-Playground的Playground总动员例程
>
> - 原**Baseline**已经实现 **离散步态（walk 或 run）** 的能力（两个神经网络）
> - **《人机追逐对抗 0.0》** 的目标是用 **一个神经网络实现“连续平滑过渡”** 的能力

2.1 核心控制公式——关键变量

原Baseline控制策略采用 **“基于参考轨迹的残差控制（Reference-Guided Residual Control）”** 架构。
核心控制公式 (The Final Control Formula)：

$$u_{total} = \underbrace{(k_b \cdot u)}_{\text{RL策略 (大脑)}} + \underbrace{(d_h \cdot u_f + d_0)}_{\text{参考轨迹 (脊髓)}}$$

- **输入**：
  1. `continuousActions[i]`：来自 RL 策略网络（大脑）的原始输出。
  2. `currentGait`：当前的步态参数（脊髓设定）。
- **输出**：
  - `utotal[]`：**最终关节指令。**融合了 RL 修正量和正弦波参考量，直接发送给 ArticulationBody 的 xDrive.target。

1. 参考轨迹相关核心参数—— MoB 的核心
这组参数决定了参考轨迹的“形状”和“风格”，也是实现跑走切换的关键。

> **其中，更重点关注以下3个超参数：**

```
public struct GaitParams
{
    public float T1; // 周期 (Walk~40, Run~25)
    public float dh; // 抬腿幅度 (Walk~10, Run~40)
    public float d0; // 身体姿态偏置 (Walk~0, Run~20)
}
```

- `d0` **(Offset / Posture Bias)**:
  - **定义**：正弦波的零点偏置，即关节的**静态平衡位置**。
  - **物理意义**：决定了机器人的**基础姿态**。
  - **示例**：对于 Tinker 机器人，直立状态会导致关节过伸，因此默认需要 `d0=20`（屈膝 20 度）作为物理零点。
    - **Walk (走)**：`d0` 较小（接近基础姿态），站姿较高。
    - **Run (跑)**：`d0` 较大（如下蹲 15 度），降低重心以获得更好的蹬地爆发力。
- `dh` **(Dynamic Height / Amplitude)**:
  - **定义**：正弦波的振幅。
  - **物理意义**：决定了腿部的**运动幅度**（抬腿多高、跨步多大）。
  - **示例**：`dh=20` 为标准步幅，`dh=35` 为高抬腿大跨步。
- `uf1` / `uf2` **(Unit Factors)**:
  - **定义**：归一化后的正弦波信号，值域 `[0, 1]`。
  - **计算公式**：$uf = \frac{-\cos(2\pi \cdot \frac{tp}{T1}) + 1}{2}$
  - **作用**：将线性的时间 `tp` 转化为平滑的钟形曲线，驱动关节柔和运动。
  - `T1` **(Period / Cycle Duration)**:
    - **定义**：半个步态周期所包含的物理帧数。
    - **物理意义**：**频率的倒数**。`T1` 越小，动作越快。
    - **数值范围**:
      - `T1 = 20`：极速跑（每 20 帧迈一步）。
      - `T1 = 40`：悠闲慢走（每 40 帧迈一步）。
  - `tp` **(Time Phase Counter)**:
    - **定义**：当前的相位计数器。
    - **逻辑**：在 `FixedUpdate` 中每帧 `+1`。
      - 当 `0 < tp <= T1`：处于 **左腿驱动相**（左腿动，右腿支撑）。
      - 当 `T1 < tp <= 2*T1`：处于 **右腿驱动相**（右腿动，左腿支撑）。
      - 当 `tp > 2*T1`：重置为 0，开始新循环。

> **代码逻辑上确实没有“腾空相”，但物理结果上会出现“腾空”。**
>
> **控制信号（Signal）角度——确实没有腾空相。**
>
> - `0 ~ T1`: 左腿有信号。
> - `T1 ~ 2*T1`: 右腿有信号。
> - **信号是无缝衔接的**。代码中并没有一段 `T_flight` 时间让 `uf1` 和 `uf2` 同时为 0。
>
> **动力学角度——动力学涌现 (Emergent Dynamics)，因惯性而腾空。**
> 当 `T1` 很小（频率快）且 `dh` 很大（蹬地猛）时：
>
> 1. **蓄力**：正弦波前半段，腿弯曲。
> 2. **爆发**：正弦波后半段，腿猛烈伸直（蹬地）。
> 3. **起飞**：由于蹬地的**冲量（Impulse）**非常大，产生的垂直速度 V_y 足以让机器人在地心引力把它拉下来之前，飞在空中。
> 4. **滞空**：在空中时，虽然代码已经切换到了“右腿相”，并且右腿可能已经在尝试往下踩了，但因为身体飞得太高，脚够不着地 —— **这就是腾空相。**

> **比喻：**
> 这就好比你骑自行车。你的脚是一圈一圈不停蹬的（信号连续），但如果你骑得足够快并冲过一个小坡，车子就会腾空（物理结果）。你不需要停止蹬车才能腾空。
>
> **为什么 Run 需要 `d0` (蹲姿)?**
> 这也解释了为什么 Run 模式下 `d0` 要设为 15（蹲下）：
>
> - 如果你站得很直（`d0=0`），腿已经伸直了，就没有“伸长空间”来蹬地了。
> - 只有先蹲下（`d0=15`），像弹簧一样压缩，当正弦波让腿伸直时，才能输出爆发力，从而**制造出代码里没有写、但物理上存在的腾空相**。

2. RL策略相关参数

$$u_{total-RL}[i] = kb[i] * u[i] + kb1[i] * ut[i] + kb2[i] * utt[i]$$

- `u[]`：RL Residual Action / 神经网络残差动作。**对应传统PID控制中的P项**
  - **定义**：由强化学习策略网络（Policy Network）根据当前观测实时计算出的原始控制信号。
  - **物理本质**：它是对基础正弦波轨迹的**动态修正项**。当基础步态（Walk/Run）无法应对复杂地形或扰动时，RL 通过输出非零的 u 值，在基础角度上叠加一个微调力矩，从而实现动态平衡。
  - **信号处理**：代码中对 u 进行了低通滤波（系数 kk=0.9），以过滤高频抖动，确保电机输出平滑，模拟生物肌肉的粘滞特性。
- `kb[]`：**增益 (Gain)**。用于放大 RL 的输出信号（如设为 30）。
  - *注：在 Debug 过程中，如果发现 RL“对抗”正弦波（导致跛脚），通常需要降低此值。*
- `ut[]`, `utt[]`, `kb1[]`, `kb2[]`： 对应 PID 控制中的 **一重积分项（I项，稳态误差 Steady-state Error），二重积分项（II项）**以及对应的增益系数

```
        for (int i = 0; i < ActionNum; i++)
        {
            u[i] = u[i] * kk + (1 - kk) * continuousActions[i];
            ut[i] += u[i];
            utt[i] += ut[i];
            utotal[i] = kb[i] * u[i] + kb1[i] * ut[i] + kb2[i] * utt[i];
            if (fixbody) utotal[i] = 0;
        }
```

> **双足：忽略 `ut[]`, `utt[]`；轮式考虑**
>
> - **积分项（I）擅长通过累积历史偏差来消除稳态误差（提供对抗恒定阻力的“后劲”）；**
> - **双足 (Biped)：仅用 P项 (`u`)**。核心诉求是 **极速响应**，动态平衡需要 0.02s 级的瞬时修正，无法容忍积分项带来的相位滞后，误差修补全靠 NN 的高频实时计算。
> - **轮腿 (LegWheeled)：引入 I/II项 (`ut/utt`)**。核心诉求是 **稳态维持**，利用积分项消除静差以克服恒定的地面摩擦力，确保轮子保持平滑的恒定转速。

传统PID控制视角			**动力学视角**
变量/参数	控制理论对应	数学运算	物理作用
u	**P (Proportional)**	$f(x)$	**主要发力**。神经网络说“去这里”，电机就用力往那里转。
ut	**I (Integral)**	$\int f(x)dt$	**消除误差**。如果没这就位，就持续加力（双足中禁用）。
utt	**II (Double Integral)**	$\iint f(x)dt2$	**位移/惯性维持**。极其迟钝（双足中禁用）。
damping	**D (Derivative)**	$f\prime(x)$	**刹车/稳定**。防止动作太猛产生震荡。**代码中固定为 100，RL无法控制。**

> **低通滤波（Low-pass Filter） 处理（平滑处理）——这一帧的动作 = 90% 的上一帧动作 + 10% 的新想法**
> 在 OnActionReceived 中，u 并不是直接等于神经网络的输出，而是经过了一个低通滤波。其中 `continuousActions[i]` 是神经网络的输出

2.2 全生命周期数据流 (Lifecycle & Data Flow)

Unity-cs主要包含以下组件：

```
    public override void Initialize()
    public override void OnEpisodeBegin()
    public override void CollectObservations(VectorSensor sensor)
    public override void OnActionReceived(ActionBuffers actionBuffers)
    void FixedUpdate()
```

系统的运行顺序：1-2-3-2-3-.....

1. 启动与初始化 (Initialize)

- **时机**：脚本加载时执行一次。
- **操作**：构建身体（获取关节引用），统计自由度 (`ActionNum`)，初始化默认步态参数。

2. 场景重置 (OnEpisodeBegin)

- **时机**：训练开始或机器人摔倒/超时。
- **操作**：
  1. **物理归位**：重置位置与速度。
  2. ~~**任务采样**：调用 `RandomizeGaitCommand()` ，随机生成本回合的战术目标，这是 MoB 训练多样性的源头。~~

3. 物理步进循环 (Physics Step Loop)

这是以 50Hz 运行的主循环，单帧数据流向如下：

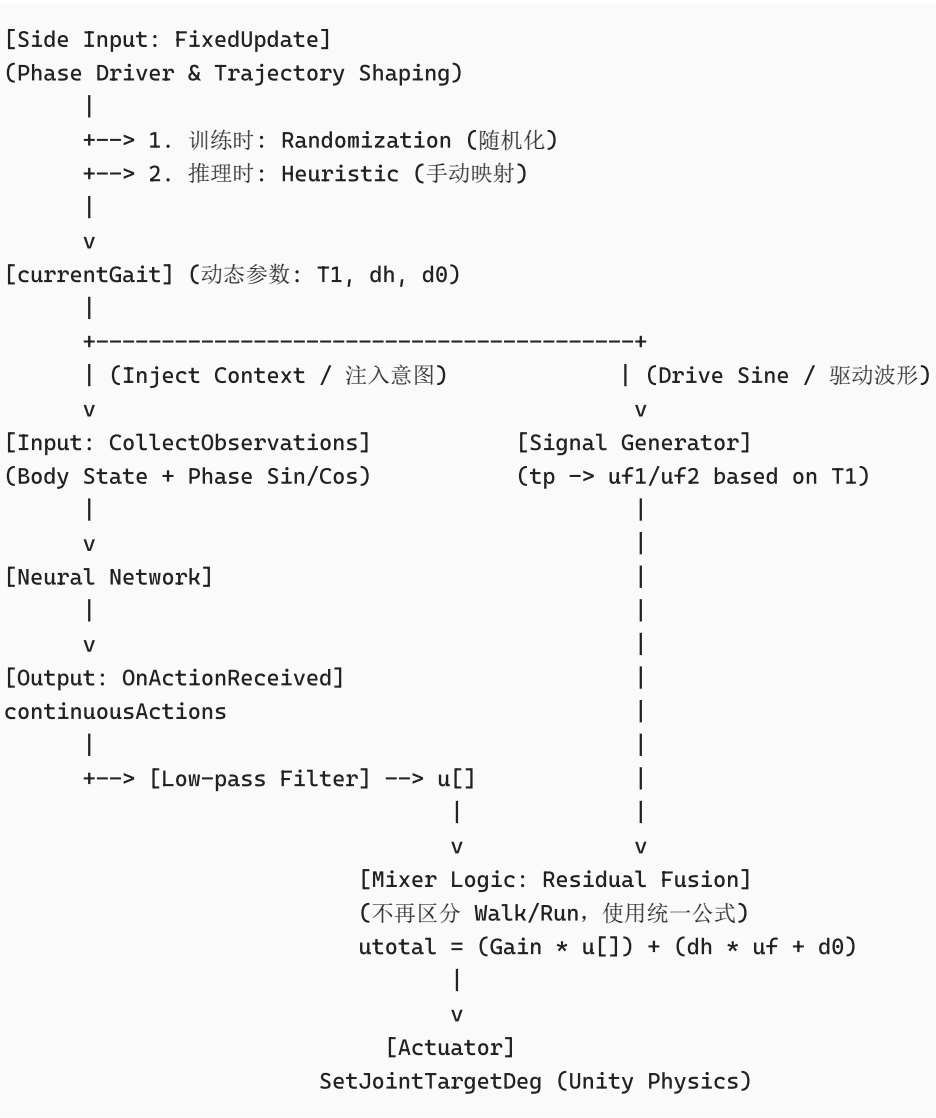
- **Step A: 脊髓节律 (FixedUpdate 前半部分)**
  - **时钟推进**：tp++。
  - **波形生成**：根据当前 T1 计算参考轨迹信号 uf1 / uf2。
  - ~~**参数演化**：(训练时) 对 GaitParams 进行平滑插值。~~
- **Step B: 感知 (CollectObservations)**
  - **操作**：读取物理状态（速度/角度） + **注入相位 (`sin/cos`) 与步态指令 (T1, dh, d0)**。
  - **流向**：打包数据，发送给 ML-Agents Python 端。
- **Step C: 大脑决策 (ML-Agents 内部黑盒)**
  - **Input**：Step B 收集的 Observation 向量。
  - **Process**：策略网络 (Policy Network) 前向传播。
  - **Output**：continuousActions[i] (原始动作指令)。
- **Step D: 运动混合与下发 (OnActionReceived)**
  - **信号处理**：低通滤波 $u_t = 0.9u_t - 1 + 0.1a_{raw} u_t = 0.9u_t - 1 + 0.1a_{raw}$
  - **指令融合**：执行核心公式 $u_{total} = (kb \cdot u) + (dh \cdot uf + d0) u_{total} = (kb \cdot u) + (dh \cdot uf + d0)$
  - **下发执行**：调用 SetJointTargetDeg 将最终指令发送给电机 (xDrive.target)。
- **Step E: 物理模拟 (Unity Physics Engine)**
  - **黑盒计算**：物理引擎结合重力、摩擦力、惯性与电机指令，计算出机器人**下一帧的真实状态**（产生腾空等动力学现象）。
- **Step F: 结果评估与反馈 (FixedUpdate 后半部分)**
  - **计算奖励**：基于物理模拟后的真实状态计算 Reward。
  - **闭环反馈**：将 Reward 发回给 RL 算法；(训练模式下) Python 端利用积累的 Reward 和 Observation 序列更新神经网络参数。

3. 技术报告：MoB 架构下的代码重构与实现

> 为了实现双足机器人从“离散步态切换”到“连续平滑过渡”的能力，我们将原有的有限状态机（FSM）逻辑重构为基于 **Multiplicity of Behavior (MoB)** 的参数化控制架构。

MoB 架构数据流图

相比原版，主要变化在于：**“侧路输入”现在不仅驱动物理信号，还注入了感知层**；混合逻辑从**“状态选择”变成了“统一公式”**。



图解说明 (写在报告中的解释)

1. **参数源头 (`Side Input`)**：
   - 不再是静态的 `if-else`。
   - 引入了 `currentGait` 作为**单一事实来源 (Single Source of Truth)**。它既决定了物理层面的参考轨迹，也决定了感知层面的任务指令。
2. **双路注入 (Dual Injection)**：
   - **一路向右（脊髓）**：`currentGait` 直接驱动正弦波发生器，产生基础步态信号。
   - **一路向左（大脑）**：`currentGait` 被注入到 `CollectObservations`，让神经网络“看见”当前的频率 (`T1`) 和幅度 (`dh`) 要求。
3. **统一混合 (Unified Mixer)**：
   - 原版的 `(根据 Walk/Run 选择参数)` 被移除。
   - 取而代之的是**连续变化的参数计算**，实现了动作的无缝平滑过渡。

3.1 数据结构重构：从“状态枚举”到“参数向量”

原状 (Baseline)：使用 `enum StyleB { walk, run }` 进行硬编码切换。变量 `T1`，`dh`，`d0` 散落在类中，通过 `if-else` 赋值。
修改 (MoB)：引入结构体封装，将步态视为一个连续的**参数向量**。

- **代码实现**：

```
// [New] 定义步态参数结构体
[System.Serializable]
public struct GaitParams
{
    public float T1; // 周期/频率
    public float dh; // 振幅/抬腿高度
    public float d0; // 偏置/姿态高度
}

// [New] 引入当前状态与目标状态，用于插值
[SerializeField] private GaitParams currentGait; // 当前执行目标（平滑/过渡）
public GaitParams targetGait; // 遥控指令目标
[SerializeField] private int changeGaitTimer = 0; // 训练计时器
```

- 在 `FixedUpdate()` 中：

```
            // 参数平滑插值，避免物理突变
            currentGait.T1 = Mathf.Lerp(currentGait.T1, targetGait.T1, 0.1f);
            currentGait.dh = Mathf.Lerp(currentGait.dh, targetGait.dh, 0.1f);
            currentGait.d0 = Mathf.Lerp(currentGait.d0, targetGait.d0, 0.1f);
```

3.2 感知空间增强：注入“相位”与“意图”

原状 (Baseline)：神经网络仅观测物理状态（速度、角度）。它是“盲目”的，不知道当前脚本希望它快跑还是慢走，全靠脚本的物理强制力拖拽。
修改 (MoB)：在 `CollectObservations` 中显式注入**相位信息**和**任务上下文**，使 RL 大脑与脊髓节律同步。

- **代码实现**：

```
public override void CollectObservations(VectorSensor sensor)
{
    // ... 原有物理观测 ...

    // [New 1] 相位编码 (Phase Encoding)
    // 使用 Sin/Cos 解决 0 与 2π 的不连续问题，告诉 NN 当前腿摆在什么位置
    float phase = (float)tp / (2f * Mathf.Max(1f, currentGait.T1));
    sensor.AddObservation(Mathf.Sin(2 * Mathf.PI * phase));
    sensor.AddObservation(Mathf.Cos(2 * Mathf.PI * phase));

    // [New 2] 任务上下文 (Context/Command)
    // 告诉 NN 当前的目标风格（是跑还是走？）
    sensor.AddObservation(currentGait.T1);
    sensor.AddObservation(currentGait.dh);
    sensor.AddObservation(currentGait.d0);
}
```

> **注意**：需同步在 Unity Inspector 中增加 `Observation Space Size` (+5)。

3.3 控制逻辑统一：移除“硬编码状态机”

原状 (Baseline)：

```
if (Walk) { T1=30; utotal[...] += ... }
if (Run)  { T1=20; utotal[...] += ... } // 逻辑割裂，无法混合
```

修改 (MoB)：统一使用**同一套残差公式**，通过 `currentGait` 变量驱动。无论走还是跑，物理本质都是正弦波驱动，区别仅在于参数值不同。

- **代码实现 (`OnActionReceived`)：**

```
// [Refactor] 移除所有 if (Walk/Run) 分支，使用统一逻辑
// 动态获取参数
T1 = (int)currentGait.T1;
dh = currentGait.dh;
d0 = currentGait.d0;

// 执行统一的混合公式 (Residual Fusion)
// idx[0] 为髋关节
utotal[idx[0]] += (dh * uf1 + d0) * Mathf.Sign(idx[0]);
// ... 其他关节同理 ...

// [Crucial] 移除“自杀逻辑”
// Delete: if (target != last) EndEpisode();
// 允许在同一回合内动态改变参数，训练过渡能力
```

3.4 训练策略升级：域随机化 (Domain Randomization)

原状 (Baseline)：静态训练。设为 Walk 就一直练 Walk。
修改 (MoB)：在训练过程中，动态、随机地改变步态参数(T1,dh,d0)，迫使 Agent 适应参数的动态变化。

> 课程学习训练：从慢走 (40,10,0) 到快跑 (25,40,20) 的所有参数组合可能组合

```
        if (train)
        {
            changeGaitTimer++;
            if (changeGaitTimer > 300) // 每3秒变一次
            {
                changeGaitTimer = 0;
                RandomizeGaitCommand_Smoothed();
            }
            // 平滑插值
            currentGait.T1 = Mathf.Lerp(currentGait.T1, targetGait.T1, 0.1f);
            currentGait.dh = Mathf.Lerp(currentGait.dh, targetGait.dh, 0.1f);
            currentGait.d0 = Mathf.Lerp(currentGait.d0, targetGait.d0, 0.1f);
        }
```

```
    void RandomizeGaitCommand_Smoothed()
    {
        int step = Academy.Instance.StepCount;
        // 计算课程进度 (0.0 到 1.0)，在 200万步时完成
        float progress = Mathf.Clamp01((float)step / 2000000f);
        // 动态范围：随着训练深入，难度逐渐加大
        float minT1 = Mathf.Lerp(38f, 24f, progress);
        float maxDh = Mathf.Lerp(14f, 41f, progress); // 从小碎步(14) 扩展到 大跨步(41)
        float maxD0 = Mathf.Lerp(5f, 20f, progress);  // 从直立(5) 扩展到 蹲姿(20)
        targetGait.T1 = Random.Range(minT1, 41f);
        targetGait.dh = Random.Range(9f, maxDh);
        targetGait.d0 = Random.Range(0f, maxD0);
    }
```

> 测试：

```
            changeGaitTimer++;
            // 起点：慢走
            const float startT1 = 40f;
            const float startDh = 10f;
            const float startD0 = 0f;
```

```
            // 终点：快跑
            const float endT1 = 25f;
            const float endDh = 40f;
            const float endD0 = 20f;

            // 每 ... 前进 10%
            float progress = Mathf.Clamp01(changeGaitTimer / 3000f);
            targetGait.T1 = Mathf.Lerp(startT1, endT1, progress);
            targetGait.dh = Mathf.Lerp(startDh, endDh, progress);
            targetGait.d0 = Mathf.Lerp(startD0, endD0, progress);

            currentGait.T1 = Mathf.Lerp(currentGait.T1, targetGait.T1, 0.1f);
            currentGait.dh = Mathf.Lerp(currentGait.dh, targetGait.dh, 0.1f);
            currentGait.d0 = Mathf.Lerp(currentGait.d0, targetGait.d0, 0.1f);
```

总结

通过上述修改，我们从根本上改变了 Agent 的学习目标：

- **Before**：学习“如何在 Walk 状态下不倒” 和 “如何在 Run 状态下不倒”。
- **After**：学习**“如何根据输入的 T1/dh/d0 指令，配合正弦波信号，在任何步态参数下保持动态平衡”**。

这使得训练出的模型（Policy）能够通过简单地调节 `currentGait` 变量，实现从慢走到快跑的无缝丝滑过渡。

4. 实现过程

其他设置：

1. Unity界面，观察数量+5（改成38）
2. Unity界面，maxstep修改成5000并在cs文件把 `FixedUpdate()` 中 `tt>=1000` 注释掉。（原代码中step=1000后机器人会重置）

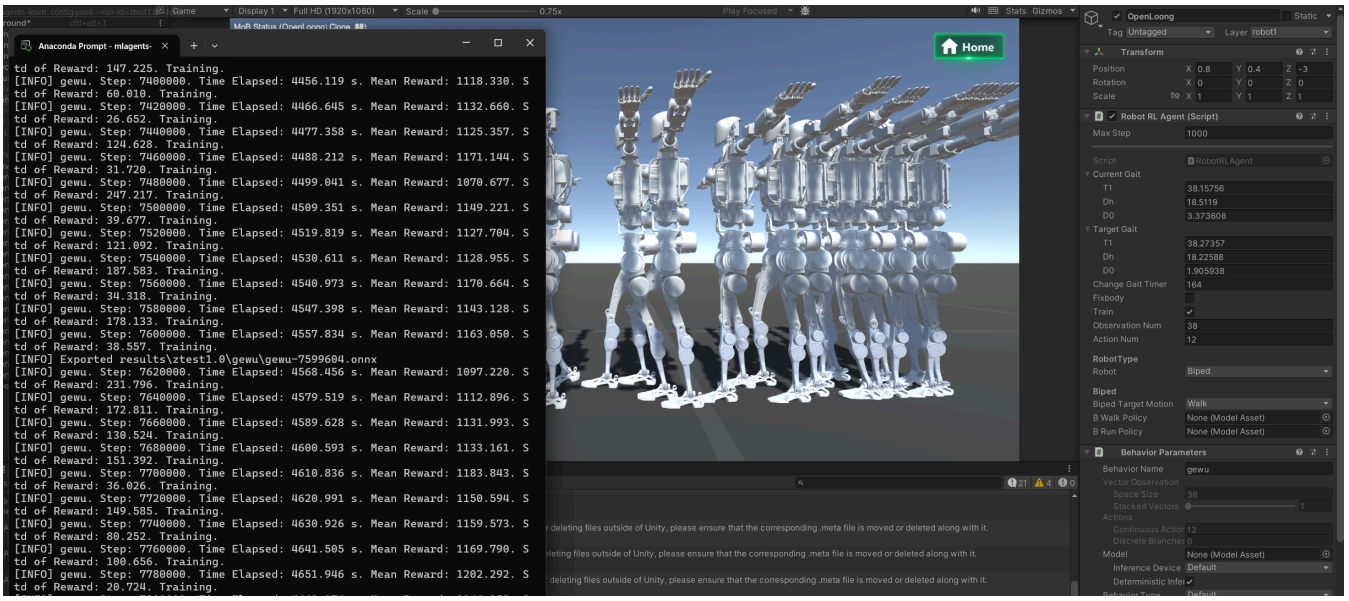
```
if (Mathf.Abs(EulerTrans(body.eulerAngles[0])) > 20f || Mathf.Abs(EulerTrans(body.eulerAngles[2]))
> 20f ) // || tt>=1000
```

3. Unity界面，训练场地增大10倍（不然一定距离后机器人会掉下去）
4. 训练时，cs文件内，机器人克隆数量修改成20（根据cpu性能决定）

训练过程

fixbody 测试底层正弦波（见附件视频）

训练700wstep时，已经可以平稳行走（可能不用700w就可以了，中间睡了一觉没看到...）
此时（T1，Dh，D0）的取值为（38.15,18.51,3.37）



此时，删除源代码中的步数限制，修改maxstep为5000

一开始reward还在提升

```
[INFO] gewu. Step: 9000000. Time Elapsed: 53.768 s. No episode was completed since last summary. Training.
[INFO] gewu. Step: 9020000. Time Elapsed: 67.342 s. Mean Reward: 6799.005. Std of Reward: 352.663. Training.
[INFO] gewu. Step: 9040000. Time Elapsed: 74.776 s. Mean Reward: 6853.167. Std of Reward: 35.943. Training.
[INFO] gewu. Step: 9060000. Time Elapsed: 92.243 s. Mean Reward: 6969.277. Std of Reward: 0.000. Training.
[INFO] gewu. Step: 9080000. Time Elapsed: 102.709 s. No episode was completed since last summary. Training.
[INFO] gewu. Step: 9100000. Time Elapsed: 113.101 s. No episode was completed since last summary. Training.
[INFO] gewu. Step: 9120000. Time Elapsed: 119.611 s. Mean Reward: 7280.069. Std of Reward: 75.030. Training.
[INFO] gewu. Step: 9140000. Time Elapsed: 133.614 s. Mean Reward: 7259.951. Std of Reward: 35.708. Training.
[INFO] gewu. Step: 9160000. Time Elapsed: 144.125 s. Mean Reward: 7564.449. Std of Reward: 0.000. Training.
[INFO] gewu. Step: 9180000. Time Elapsed: 154.557 s. No episode was completed since last summary. Training.
[INFO] gewu. Step: 9200000. Time Elapsed: 165.058 s. Mean Reward: 5531.707. Std of Reward: 0.000. Training.
[INFO] Exported results\ztest1.0\gewu\gewu-9199080.onnx
[INFO] gewu. Step: 9220000. Time Elapsed: 175.584 s. Mean Reward: 7667.285. Std of Reward: 91.259. Training.
[INFO] gewu. Step: 9240000. Time Elapsed: 182.511 s. Mean Reward: 7687.806. Std of Reward: 35.576. Training.
[INFO] gewu. Step: 9260000. Time Elapsed: 199.288 s. Mean Reward: 7756.780. Std of Reward: 0.000. Training.
[INFO] gewu. Step: 9280000. Time Elapsed: 212.415 s. No episode was completed since last summary. Training.
[INFO] gewu. Step: 9300000. Time Elapsed: 223.340 s. Mean Reward: 7061.743. Std of Reward: 798.611. Training.
[INFO] gewu. Step: 9320000. Time Elapsed: 233.874 s. Mean Reward: 8004.603. Std of Reward: 44.871. Training.
[INFO] gewu. Step: 9340000. Time Elapsed: 245.249 s. Mean Reward: 4511.321. Std of Reward: 3606.991. Training.
[INFO] gewu. Step: 9360000. Time Elapsed: 255.875 s. Mean Reward: 4324.568. Std of Reward: 3340.112. Training.
[INFO] gewu. Step: 9380000. Time Elapsed: 262.523 s. No episode was completed since last summary. Training.
[INFO] gewu. Step: 9400000. Time Elapsed: 275.363 s. Mean Reward: 8135.579. Std of Reward: 0.000. Training.
[INFO] gewu. Step: 9420000. Time Elapsed: 285.979 s. Mean Reward: 8254.636. Std of Reward: 50.240. Training.
[INFO] gewu. Step: 9440000. Time Elapsed: 296.510 s. Mean Reward: 8251.211. Std of Reward: 7.516. Training.
[INFO] gewu. Step: 9460000. Time Elapsed: 307.063 s. Mean Reward: 8331.047. Std of Reward: 22.749. Training.
[INFO] gewu. Step: 9480000. Time Elapsed: 317.543 s. No episode was completed since last summary. Training.
[INFO] gewu. Step: 9500000. Time Elapsed: 327.947 s. Mean Reward: 8492.319. Std of Reward: 0.000. Training.
[INFO] gewu. Step: 9520000. Time Elapsed: 338.183 s. Mean Reward: 8551.509. Std of Reward: 47.751. Training.
[INFO] gewu. Step: 9540000. Time Elapsed: 348.794 s. Mean Reward: 8588.429. Std of Reward: 37.391. Training.
[INFO] gewu. Step: 9560000. Time Elapsed: 359.167 s. Mean Reward: 7247.954. Std of Reward: 2470.574. Training.
[INFO] gewu. Step: 9580000. Time Elapsed: 365.659 s. No episode was completed since last summary. Training.
[INFO] gewu. Step: 9600000. Time Elapsed: 377.984 s. Mean Reward: 8740.586. Std of Reward: 0.000. Training.
[INFO] Exported results\ztest1.0\gewu\gewu-9599284.onnx
[INFO] gewu. Step: 9620000. Time Elapsed: 388.337 s. Mean Reward: 8852.071. Std of Reward: 33.114. Training.
[INFO] gewu. Step: 9640000. Time Elapsed: 399.743 s. Mean Reward: 3929.574. Std of Reward: 4088.117. Training.
[INFO] gewu. Step: 9660000. Time Elapsed: 410.343 s. Mean Reward: 8909.518. Std of Reward: 38.178. Training.
[INFO] gewu. Step: 9680000. Time Elapsed: 420.733 s. No episode was completed since last summary. Training.
[INFO] gewu. Step: 9700000. Time Elapsed: 431.058 s. Mean Reward: 9052.534. Std of Reward: 0.000. Training.
[INFO] gewu. Step: 9720000. Time Elapsed: 441.476 s. Mean Reward: 9025.901. Std of Reward: 21.754. Training.
[INFO] gewu. Step: 9740000. Time Elapsed: 451.873 s. Mean Reward: 9010.826. Std of Reward: 11.766. Training.
[INFO] gewu. Step: 9760000. Time Elapsed: 462.915 s. Mean Reward: 7026.510. Std of Reward: 2753.876. Training.
[INFO] gewu. Step: 9780000. Time Elapsed: 473.321 s. No episode was completed since last summary. Training.
[INFO] gewu. Step: 9800000. Time Elapsed: 485.124 s. Mean Reward: 9005.274. Std of Reward: 0.000. Training.
[INFO] gewu. Step: 9820000. Time Elapsed: 495.489 s. Mean Reward: 9013.756. Std of Reward: 100.764. Training.
[INFO] gewu. Step: 9840000. Time Elapsed: 501.922 s. Mean Reward: 9029.551. Std of Reward: 39.373. Training.
[INFO] gewu. Step: 9860000. Time Elapsed: 512.315 s. Mean Reward: 9016.672. Std of Reward: 73.591. Training.
[INFO] gewu. Step: 9880000. Time Elapsed: 522.814 s. No episode was completed since last summary. Training.
```

后来reward一直在震荡

```
[INFO] gewu. Step: 10260000. Time Elapsed: 726.433 s. Mean Reward: 5989.549. Std of Reward: 4225.803. Training.
[INFO] gewu. Step: 10280000. Time Elapsed: 732.579 s. No episode was completed since last summary. Training.
[INFO] gewu. Step: 10300000. Time Elapsed: 742.731 s. Mean Reward: 8859.429. Std of Reward: 76.345. Training.
[INFO] gewu. Step: 10320000. Time Elapsed: 753.364 s. Mean Reward: 8646.715. Std of Reward: 482.198. Training.
[INFO] gewu. Step: 10340000. Time Elapsed: 763.762 s. Mean Reward: 6339.404. Std of Reward: 3725.447. Training.
[INFO] gewu. Step: 10360000. Time Elapsed: 774.273 s. Mean Reward: 8787.036. Std of Reward: 6.038. Training.
[INFO] gewu. Step: 10380000. Time Elapsed: 784.445 s. Mean Reward: 8694.826. Std of Reward: 0.000. Training.
[INFO] gewu. Step: 10400000. Time Elapsed: 794.842 s. Mean Reward: 8419.134. Std of Reward: 595.608. Training.
[INFO] Exported results\ztest1.0\gewu\gewu-10399029.onnx
[INFO] gewu. Step: 10420000. Time Elapsed: 805.616 s. Mean Reward: 8642.537. Std of Reward: 117.025. Training.
[INFO] gewu. Step: 10440000. Time Elapsed: 815.807 s. Mean Reward: 8118.068. Std of Reward: 723.883. Training.
[INFO] gewu. Step: 10460000. Time Elapsed: 826.083 s. No episode was completed since last summary. Training.
[INFO] gewu. Step: 10480000. Time Elapsed: 836.193 s. Mean Reward: 8384.656. Std of Reward: 436.974. Training.
[INFO] gewu. Step: 10500000. Time Elapsed: 845.737 s. Mean Reward: 8575.158. Std of Reward: 131.641. Training.
[INFO] gewu. Step: 10520000. Time Elapsed: 856.103 s. Mean Reward: 8578.365. Std of Reward: 133.616. Training.
[INFO] gewu. Step: 10540000. Time Elapsed: 866.750 s. Mean Reward: 7053.823. Std of Reward: 2577.212. Training.
[INFO] gewu. Step: 10560000. Time Elapsed: 876.978 s. No episode was completed since last summary. Training.
[INFO] gewu. Step: 10580000. Time Elapsed: 887.403 s. Mean Reward: 7423.811. Std of Reward: 1548.951. Training.
[INFO] gewu. Step: 10600000. Time Elapsed: 898.305 s. Mean Reward: 8190.464. Std of Reward: 768.723. Training.
[INFO] gewu. Step: 10620000. Time Elapsed: 909.149 s. Mean Reward: 8510.784. Std of Reward: 49.510. Training.
[INFO] gewu. Step: 10640000. Time Elapsed: 919.584 s. Mean Reward: 8370.254. Std of Reward: 0.000. Training.
[INFO] gewu. Step: 10660000. Time Elapsed: 925.363 s. Mean Reward: 8476.953. Std of Reward: 1.707. Training.
[INFO] gewu. Step: 10680000. Time Elapsed: 935.721 s. Mean Reward: 7825.118. Std of Reward: 1871.182. Training.
[INFO] gewu. Step: 10700000. Time Elapsed: 946.877 s. Mean Reward: 8529.773. Std of Reward: 55.589. Training.
[INFO] gewu. Step: 10720000. Time Elapsed: 957.601 s. Mean Reward: 8495.993. Std of Reward: 101.629. Training.
[INFO] gewu. Step: 10740000. Time Elapsed: 968.021 s. Mean Reward: 4725.000. Std of Reward: 0.000. Training.
[INFO] gewu. Step: 10760000. Time Elapsed: 978.096 s. Mean Reward: 8347.758. Std of Reward: 152.332. Training.
[INFO] gewu. Step: 10780000. Time Elapsed: 988.897 s. Mean Reward: 8393.204. Std of Reward: 77.880. Training.
[INFO] gewu. Step: 10800000. Time Elapsed: 998.874 s. Mean Reward: 7202.482. Std of Reward: 1867.077. Training.
[INFO] Exported results\ztest1.0\gewu\gewu-10799972.onnx
[INFO] gewu. Step: 10820000. Time Elapsed: 1008.802 s. Mean Reward: 7025.290. Std of Reward: 1295.464. Training.
[INFO] gewu. Step: 10840000. Time Elapsed: 1019.158 s. Mean Reward: 6492.049. Std of Reward: 1881.482. Training.
[INFO] gewu. Step: 10860000. Time Elapsed: 1029.570 s. Mean Reward: 8257.318. Std of Reward: 93.070. Training.
[INFO] gewu. Step: 10880000. Time Elapsed: 1040.775 s. Mean Reward: 7272.925. Std of Reward: 2490.185. Training.
[INFO] gewu. Step: 10900000. Time Elapsed: 1051.112 s. Mean Reward: 8364.204. Std of Reward: 95.182. Training.
[INFO] gewu. Step: 10920000. Time Elapsed: 1061.461 s. Mean Reward: 8382.539. Std of Reward: 48.061. Training.
[INFO] gewu. Step: 10940000. Time Elapsed: 1071.811 s. Mean Reward: 8334.292. Std of Reward: 100.701. Training.
[INFO] gewu. Step: 10960000. Time Elapsed: 1082.414 s. Mean Reward: 6884.312. Std of Reward: 2567.693. Training.
[INFO] gewu. Step: 10980000. Time Elapsed: 1093.615 s. Mean Reward: 8281.877. Std of Reward: 22.148. Training.
[INFO] gewu. Step: 11000000. Time Elapsed: 1104.046 s. Mean Reward: 8284.510. Std of Reward: 29.649. Training.
[INFO] gewu. Step: 11020000. Time Elapsed: 1114.375 s. Mean Reward: 8219.717. Std of Reward: 26.920. Training.
[INFO] gewu. Step: 11040000. Time Elapsed: 1123.480 s. Mean Reward: 8290.493. Std of Reward: 33.288. Training.
[INFO] gewu. Step: 11060000. Time Elapsed: 1134.120 s. Mean Reward: 6091.167. Std of Reward: 2788.203. Training.
[INFO] gewu. Step: 11080000. Time Elapsed: 1140.510 s. Mean Reward: 8237.737. Std of Reward: 0.000. Training.
[INFO] gewu. Step: 11100000. Time Elapsed: 1150.735 s. Mean Reward: 5404.010. Std of Reward: 3024.406. Training.
[INFO] gewu. Step: 11120000. Time Elapsed: 1161.852 s. Mean Reward: 8251.549. Std of Reward: 150.886. Training.
[INFO] gewu. Step: 11140000. Time Elapsed: 1172.220 s. Mean Reward: 8232.226. Std of Reward: 73.128. Training.
```

2000w时甚至不如1000w的效果好

```
[INFO] gewu. Step: 19660000. Time Elapsed: 6332.471 s. Mean Reward: 6516.613. Std of Reward: 1358.130. Training.
[INFO] gewu. Step: 19680000. Time Elapsed: 6345.273 s. Mean Reward: 2363.978. Std of Reward: 8626.309. Training.
[INFO] gewu. Step: 19700000. Time Elapsed: 6356.802 s. Mean Reward: 4047.751. Std of Reward: 3425.321. Training.
[INFO] gewu. Step: 19720000. Time Elapsed: 6369.393 s. Mean Reward: 6410.186. Std of Reward: 1965.984. Training.
[INFO] gewu. Step: 19740000. Time Elapsed: 6382.580 s. Mean Reward: 5756.792. Std of Reward: 2901.428. Training.
[INFO] gewu. Step: 19760000. Time Elapsed: 6394.787 s. Mean Reward: 5963.594. Std of Reward: 1900.832. Training.
[INFO] gewu. Step: 19780000. Time Elapsed: 6406.860 s. Mean Reward: 5814.098. Std of Reward: 2242.048. Training.
[INFO] gewu. Step: 19800000. Time Elapsed: 6419.641 s. Mean Reward: 7241.996. Std of Reward: 382.071. Training.
[INFO] gewu. Step: 19820000. Time Elapsed: 6432.084 s. Mean Reward: 6860.187. Std of Reward: 1017.515. Training.
[INFO] gewu. Step: 19840000. Time Elapsed: 6444.789 s. Mean Reward: 7453.436. Std of Reward: 48.440. Training.
[INFO] gewu. Step: 19860000. Time Elapsed: 6451.835 s. Mean Reward: 7269.826. Std of Reward: 343.084. Training.
[INFO] gewu. Step: 19880000. Time Elapsed: 6464.941 s. Mean Reward: 6433.353. Std of Reward: 2038.925. Training.
[INFO] gewu. Step: 19900000. Time Elapsed: 6478.352 s. Mean Reward: 5535.952. Std of Reward: 2544.286. Training.
[INFO] gewu. Step: 19920000. Time Elapsed: 6489.706 s. Mean Reward: 6537.033. Std of Reward: 2476.751. Training.
[INFO] gewu. Step: 19940000. Time Elapsed: 6502.408 s. Mean Reward: 6895.301. Std of Reward: 1316.098. Training.
[INFO] gewu. Step: 19960000. Time Elapsed: 6514.967 s. Mean Reward: 7514.332. Std of Reward: 219.757. Training.
[INFO] gewu. Step: 19980000. Time Elapsed: 6527.230 s. Mean Reward: 5960.319. Std of Reward: 2848.877. Training.
[INFO] gewu. Step: 20000000. Time Elapsed: 6539.863 s. Mean Reward: 5621.945. Std of Reward: 2913.801. Training.
[INFO] Exported results\ztest1.0\gewu\gewu-19999403.onnx
[INFO] Exported results\ztest1.0\gewu\gewu-20000403.onnx
```

最终训练2000w的神经网络测试（见附件视频）。在(T1,dh,d0)分别于（40，32），（10，25），（0，10）区间内变化时，表现还好。但步频和步幅上去之后就摔了。

5. 后续优化

我猜测上述实验不成功的原因：

1.