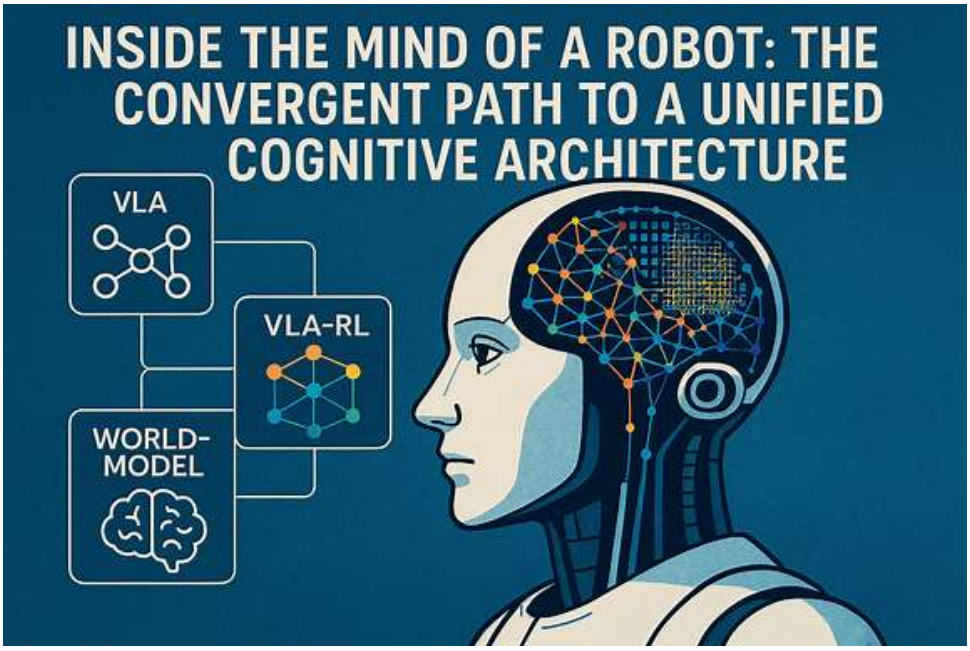


机器人的心智内部：通往统一认知架构的融合之路

Inside the Mind of a Robot: The Convergent Path to a Unified Cognitive Architecture

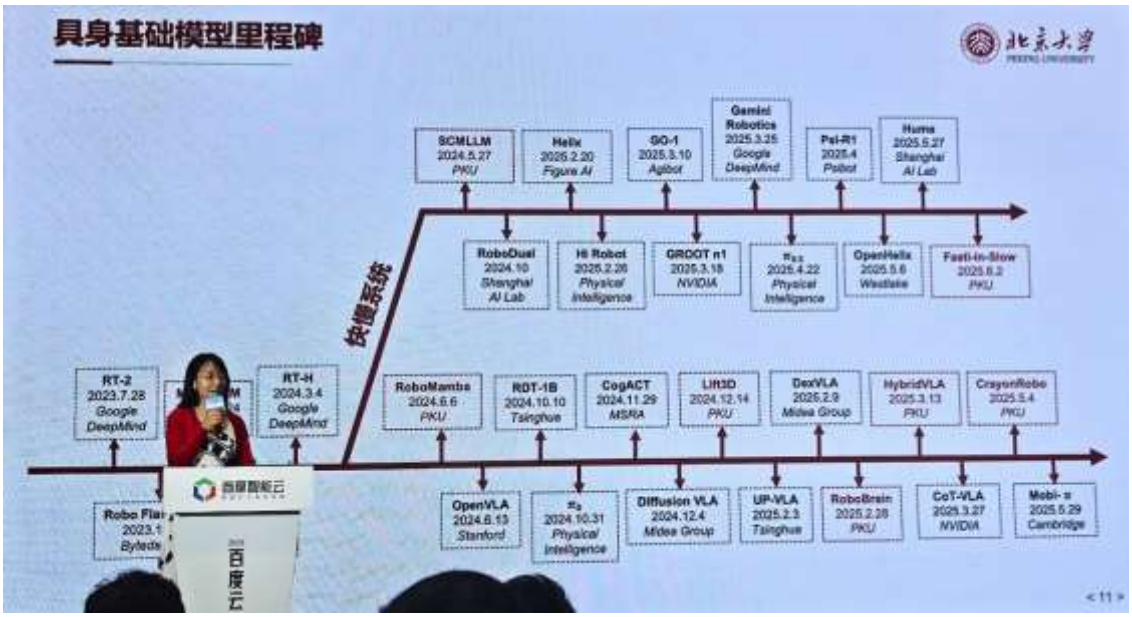
Author : Aslan Tarik

机器人学的基础范式正经历一场历史性的变革。我们正在果断地告别那些脆弱、人工构建的流程系统，迈入一个由学习驱动的端到端系统时代——在这个新时代中，感知、语言、预测与控制被无缝整合。这一转变的动力并非源于渐进式的优化，而是来自一场“伟大的融合”：基础人工智能的突破正迅速与先进的机器人硬件深度结合。

The foundational paradigm of robotics is undergoing a historic transformation. We are moving decisively away from brittle, hand-engineered pipelines and into an era of learned, end-to-end systems that seamlessly couple perception, language, prediction, and control. This shift is not driven by incremental improvement, but by a **Great Convergence**: the rapid merger of foundational AI breakthroughs with advanced robotics hardware.

自 2022 这一分水岭年份以来，这场融合的步伐显著加快。大型语言模型（LLMs）与扩散模型的崛起，为语义推理提供了关键的“脑”，而对人形与四足机器人平台的投资，则构建了所需的“身体”。如今，进入 2024 年及未来，我们正目睹具身基础模型的爆炸性涌现——这些人工智能系统不仅在文本与图像上接受训练，更在物理、行为与因果关系中汲取经验。

Since the watershed year of 2022, the pace of this convergence has accelerated significantly. The rise of large language models (LLMs) and diffusion models has provided the critical "brain" for semantic reasoning, while investments in humanoid and quadruped robotic platforms have built the necessary "body." Now, as we head into 2024 and beyond, we are witnessing an explosion of embodied foundational models—AI systems trained not only on text and images but also on experience with physics, behavior, and causal relationships.



本文将深入剖析推动这场革命的核心架构组件：

- **视觉-语言-动作（VLA）模型**：充当机器人的高级推理皮层，将像素输入与语言指令（如“整理这个房间”）转化为结构化的动作序列。
- **世界模型**：作为内部的前额皮层，用于进行心理模拟，使机器人能够预测自身行为的结果，并通过内部推演规划复杂任务。
- **VLA-RL（强化学习）**：提供专门技能优化机制，通过实践微调通用型 VLA 的策略，以实现稳健的现实世界表现。
- **大脑 + 小脑架构**：提供一种仿生式的实时控制蓝图，将大脑中的高层“做什么”规划与小脑控制器中低层、毫秒级的“如何做”执行相分离。

关键在于，这些组件并非彼此竞争的替代方案，而是正在被整合为统一认知堆栈的互补层级。像 RoboBrain 2.0 这样的框架正是这一演进的典范：没有任何单一模型能够解决全部问题。未来的方向是构建一个“机器人智能互联网”——一个由视觉、语言、动作与模拟等专用模型组成的生态系统，共同构成一个有层次的统一心智。

这种融合带来的影响是深远的。对开发者而言，它意味着可以基于强大的预训练原语构建系统，而不必从零开始。对管理者与战略制定者而言，它预示着具备理解、适应与安全行动能力的通用型机器人，已不再是遥不可及的幻想，而是清晰可行的工程路线图。如今的竞赛不再只是打造更好的马达，而是要编码出机器人在现实世界中运作所需的常识与物理直觉。

本文将在接下来的章节中详细解析上述每一个组件，为开发者提供技术基础，为决策者提供战略视角，最终清晰描绘出这些模块如何在 RoboBrain 2.0 等系统中融合，真正将通用型机器人带入我们的家庭与工作场所。

This article deconstructs the core architectural components powering this revolution:

- **Visual-Language-Action (VLA) Models** act as the robot's **high-level reasoning cortex**, translating pixel inputs and language commands like "Tidy this room" into structured action sequences.
- **World Models** serve as an internal **prefrontal cortex for mental simulation**, enabling the robot to predict the outcomes of its actions and plan complex tasks through internal deliberation.
- **VLA-RL (Reinforcement Learning)** provides the mechanism for **specialized skill optimization**, fine-tuning a generalist VLA's policies through practice to achieve robust, real-world performance.
- **Brain + Cerebellum Architectures** offer a bio-inspired **blueprint for real-time control**, separating high-level "what" planning in the brain from low-level,毫秒级 "how" execution in the cerebellar controller.

Critically, these components are not competing alternatives. They are complementary layers now being integrated into unified cognitive stacks. Frameworks like **RoboBrain 2.0** exemplify this next step: no single model is the solution. Instead, the future lies in an **"internet of robot intelligence"**—an ecosystem where specialized models (vision, language, action, simulation) are composed into a cohesive, hierarchical mind.

The impact of this convergence is profound. For developers, it means building with powerful pre-trained primitives rather than from scratch. For managers and strategists, it signals that general-purpose robots capable of understanding, adapting, and acting safely in human spaces are no longer a distant fantasy but a tangible engineering roadmap. The race is no longer just about building a better motor; it is about encoding the common sense and physical intuition necessary to operate in our world. This article explores the architectures that will get us there.

The following sections will deconstruct each of these components in detail, providing a technical foundation for developers and a strategic overview for decision-makers, ultimately painting a clear picture of how these pieces converge in systems like RoboBrain 2.0 to finally bring capable, general-purpose robots into our homes and workplaces.

机器人如何做出决策?

可以把机器人的决策系统想象成它的“个人作战手册”。这本手册包含了在各种情境下该如何行动的全部指令。这个手册的正式术语是“策略”（policy）。

简单来说，策略就是机器人的行动计划——一套指导它如何从当前位置到达目标位置的行为准则。

这本作战手册主要有两种编写方式：

1. **严格型手册**：每种具体情境都有唯一的应对动作。（比如看到红灯？立刻停车。）
2. **灵活型手册**：某些情境下可能有多个可行选项，每个选项成功的概率不同。（比如遇到障碍物？左转成功的概率是 80%，右转是 20%。）

机器人如何学习自己的手册?

机器人通过练习来学习，就像人类学习玩电子游戏一样。它尝试不同的动作，观察哪些有效、哪些无效，并逐步优化自己的策略。每一次完整的任务尝试——从开始到结束——就像一次练习赛。

目标始终是以最高得分完成任务。机器人会识别哪些行为能获得奖励（得分），哪些行为会导致失误（扣分），然后更新自己的手册，以便下次表现更好。

旧方法 vs. 新目标

过去，工程师必须为每一个微小任务、每一种机器人手动编写一套全新的、极其具体的作战手册。这种方式既缓慢又昂贵，而且机器人无法应对任何突发情况。

如今的目标远比以往更宏大：我们希望打造一个**通用型作战手册**。我们希望构建的机器人能够：

- 面对混乱、真实的环境；
- 理解简单的指令，比如“请帮我整理一下”；
- 自主推理出完成任务所需的步骤。

这意味着我们正在从只能执行特定动作的机器人，迈向能够理解、学习并适应现实世界的智能机器。

How Do Robots Make Decisions?

Think of a robot's decision-making system like its personal **playbook**. This playbook contains all the instructions for what to do in different situations. The official term for this playbook is a **"policy."**

Simply put, a policy is the robot's **game plan**. It's the set of instructions that tells the robot how to behave to get from where it is now to where it needs to be.

There are two main ways to write this playbook:

1. **A Strict Playbook:** For every specific situation, there is one exact move to make. (See a red light? Stop immediately.)

2. **A Flexible Playbook:** For some situations, there might be a few good options, each with a different chance of success. (See an obstacle? There's an 80% chance going left is best, and a 20% chance going right is better.)

How does the robot learn its playbook?

Robots learn through practice, much like how a person learns to play a video game. They try different moves, see what works and what doesn't, and slowly improve their strategy. Each complete attempt at a task—from start to finish—is like one practice game.

The goal is always to finish the game with the **highest score possible**. The robot figures out which actions earn it points (rewards) and which cause it to lose points (mistakes), and it updates its playbook to try and get a better score next time.

The Old Way vs. The New Goal

In the past, engineers had to write a brand new, highly specific playbook for every single tiny task and for every different type of robot. This was incredibly slow, expensive, and the robots couldn't handle anything unexpected.

The new goal is far more ambitious: to create a **universal playbook**. We want to build robots that can:

- **Look** at a messy, real-world environment.
- **Understand** a simple instruction like, "Please help me tidy up."
- **Figure out** the steps needed to complete the job on their own.

This means moving from robots that are programmed for one specific action to machines that can understand, learn, and adapt to our world on their own

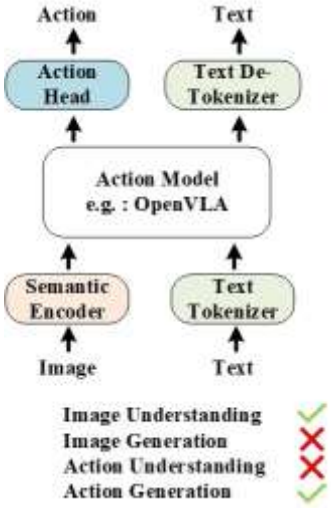
---

VLA — 视觉-语言-动作模型（“看见、理解、执行”）

VLA — Visual Language Action ("See, Understand, Do")

核心理念概述

The Core Idea in a Nutshell



视觉-语言-动作（VLA）模型是一种端到端的人工智能模型，使机器人能够:

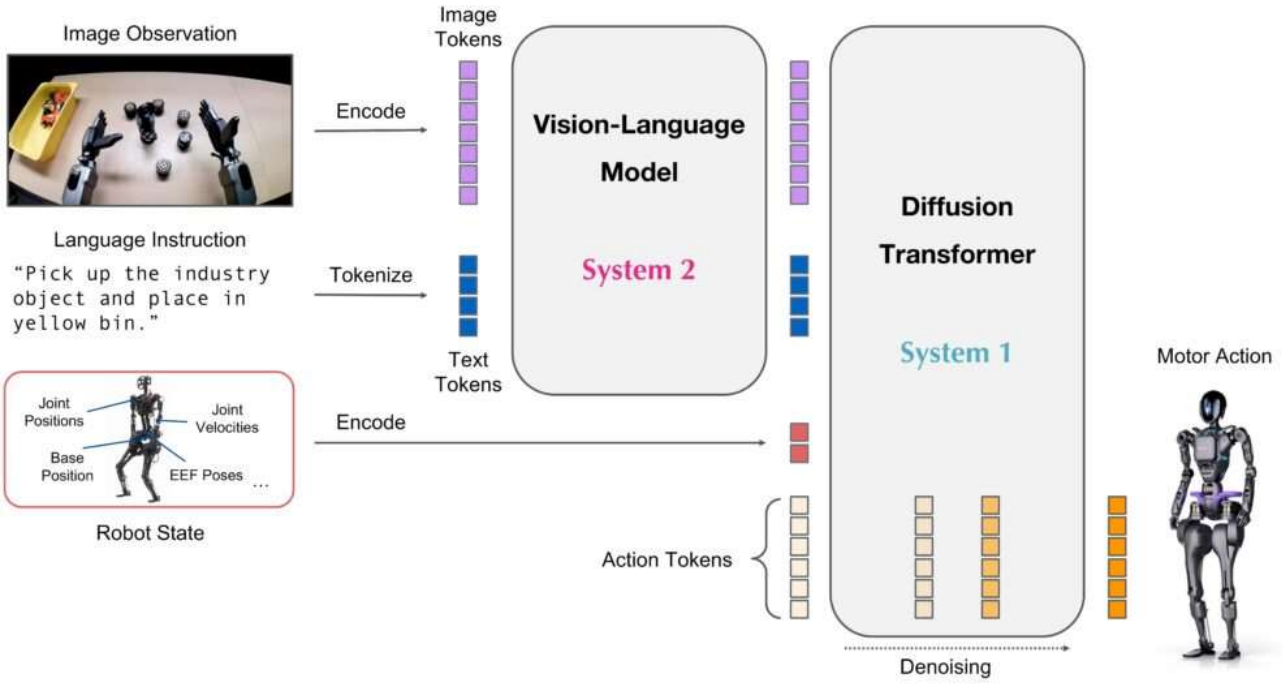
1. **看见**其环境（通过摄像头获取的图像或视频）
2. **理解**人类发出的复杂自然语言指令
3. **规划并执行**一系列物理动作以完成任务

本质上，它将“你看到的”和“你被告知的”直接转化为“你应该做的”。

A **Visual Language Action (VLA) model** is a single, end-to-end artificial intelligence model that allows a robot to:

1. **See** its environment (through images or video from its camera).
2. **Understand** a complex, natural language command from a human.
3. **Plan and Execute** a sequence of physical actions to complete the task.

In essence, it translates **"what you see and what you're told"** directly into **"what you should do."**



VLA 所解决的问题

传统的机器人控制流程通常被分割为多个独立模块：

1. **感知模块**：识别物体及其位置
2. **规划模块**：根据感知结果，使用预设逻辑决定路径或动作
3. **控制模块**：执行底层的电机指令

这种方法非常脆弱。如果物体不在预定义列表中，或者指令稍有变化（例如“拿起*苹果*”变成“抓住*红色水果*”），整个系统可能就会失败。它缺乏常识性推理能力。

Problems VLA Addresses

Traditionally, robot control pipelines were fragmented into separate modules:

1. **Perception Module**: Identify objects and their locations.
2. **Planning Module**: Use pre-programmed logic to decide a path or action based on the perception output.
3. **Control Module**: Execute low-level motor commands.

This approach was brittle. If an object wasn't in the pre-defined list or the command changed slightly ("pick up the *apple*" vs. "grab the *red fruit*"), the entire system would fail. It lacked **common sense reasoning**.

VLA 的关键优势

- **泛化能力强**：只要任务在训练分布范围内，单一模型就能完成大量未显式编程的任务
- **自然交互**：用户无需学习编程语言即可发出指令
- **具备常识**：可借助语言模型的训练知识理解世界（例如知道“冷饮”通常在“冰箱”里）

Key Advantages of VLAs

* **Generalization**: A single model can perform a vast number of tasks it wasn't explicitly programmed for, as long as they were in its training distribution.
* **Natural Interaction**: Anyone can give commands without learning a programming language.
* **Common Sense**: They can incorporate world knowledge from their language model training (e.g., knowing that "cold drinks" are likely found in a "refrigerator").

VLA 的关键劣势

- **数据需求极高**：需要海量真实世界机器人数据，收集过程缓慢、昂贵且存在风险
- **“黑箱”问题**：调试困难——如果机器人摔倒了，是视觉错误、规划错误，还是控制错误？

- **缺乏安全保障**：难以嵌入硬编码的安全约束（例如“不要撞到自己”）
- **样本效率低**：仅通过像素学习底层控制非常低效

总结

VLA 模型代表着从传统的模块化编程机器人向统一的学习型系统的范式转变。这种系统能够以更接近人类的方式理解我们的世界和指令，是构建通用型机器人的基础构件之一。

Key Disadvantages of VLAs

- **Extremely Data Hungry:** Requires astronomical amounts of real-world robot data, which is slow, expensive, and dangerous to collect.
- **"Black Box":** Very difficult to debug. If the robot falls, is it because of a vision error, a planning error, or a control error?
- **Lacks Safety & Guarantees:** It's hard to build in hard-coded safety constraints (e.g., "don't hit yourself").
- **Poor Sample Efficiency:** Learning low-level control from pixels alone is incredibly inefficient.

总结

VLA 模型代表着从传统的模块化编程机器人向统一的学习型系统的范式转变。这种系统能够以更接近人类的方式理解我们的世界和指令，是构建通用型机器人的基础构件之一。

In summary, the **VLA model** represents a paradigm shift from fragmented, programmed robots towards unified, learned systems that can understand our world and our instructions in a much more human-like way. It's a fundamental building block for creating general-purpose robots.

VLA-RL（通过强化学习训练的视觉-语言-动作模型）

核心理念概述

VLA-RL 是一种混合式方法，将预训练 VLA 模型强大的语义理解能力与强化学习（RL）面向目标的优化机制相结合。它通过微调通用型 VLA，使其在特定任务上表现更优，提高成功率、效率与鲁棒性。可以将其类比为：将一个具备广泛智能的学生（VLA）送去接受专业技能培训（RL），最终成为某项工作的专家。

VLA-RL 所解决的问题

尽管预训练的 VLA 模型在零样本任务中表现出色，但仍存在一些关键限制：

- **脆弱性**：在感知或物理上与训练数据相似但推理方式略有不同的任务中容易失败
- **次优表现**：虽然能完成任务，但方式可能缓慢、低效或不自然（如夹爪角度奇怪、动作冗余）
- **无法自我改进**：无法从错误中学习，失败一次就可能重复失败
- **奖励无感知**：无法理解复杂的非二元目标，如“高效”、“安全”或“节能”

VLA-RL 通过在目标环境中继续训练，使模型行为与精确的性能指标（奖励函数）对齐，从而解决上述问题。

“VLA-RL”的三大组成部分

这个名称代表了其组件与方法：

1. 视觉（“预训练的眼睛”）

- **定义**：来自预训练 VLA 的视觉编码器（如类似 CLIP 的模型），提供对场景的丰富语义理解
- **在 VLA-RL 中的角色**：通常在 RL 训练中保持冻结状态，不参与更新。其目的是为 RL 提供强大的通用图像特征表示，避免 RL 从零开始学习视觉

2. 语言（“预训练的大脑”）

- **定义**：预训练 VLA 的语言与策略骨干，已学会将视觉场景与语言指令关联到具体动作
- **在 VLA-RL 中的角色**：这是被微调的核心网络，具备“抓取”、“推动”、“移动到”等通用概念，RL 训练将这些技能针对特定任务进行适配与优化

3. 动作 + 强化学习（“专业训练师”）

- **定义**：VLA 的动作输出层，连接到强化学习算法（如 PPO、DDPG）
- **角色**：这是学习机制。RL 使用奖励函数评估 VLA 的行为，并指导其改进
    - VLA 作为 RL 的智能策略网络初始化
    - RL 提供梯度，调整 VLA 的权重以最大化累计奖励

VLA-RL 的关键优势

- **相较于从零开始的 RL，样本效率极高**：RL 通常非常依赖数据，而从预训练 VLA 起步就像比赛从半程开始，所需环境交互大幅减少

- **更优的最终性能**：成功率与鲁棒性优于原始 VLA 或从零训练的 RL 模型
- **可对复杂目标进行优化**：能够针对难以演示的目标进行训练（如“节能”、“温柔处理物体”）
- **适应性强**：使通用型 VLA 能适应新机器人或新环境的动态特性

当前挑战与局限

- **数据需求极高**：从像素到动作的训练需要大量交互数据，收集过程缓慢、昂贵且在真实机器人上可能存在安全风险
- **奖励设计困难**：设计合理的奖励函数极具挑战性，糟糕的奖励可能导致模型“钻空子”而非完成任务（如遮挡传感器而非解决问题）
- **安全性与稳定性问题**：端到端模型如同黑箱，难以保证其在所有场景下的行为，在现实部署中尤其在人类环境中存在风险
- **仿真到现实的鸿沟**：大多数训练在仿真环境中进行以提高效率与安全性，但将策略迁移到现实世界中的物理机器人仍是重大挑战

VLA-RL (Visual-Language-Action models trained with Reinforcement Learning)

The Core Idea in a Nutshell

VLA-RL (Visual-Language-Action models trained with Reinforcement Learning) is a hybrid approach that combines the powerful semantic understanding of pre-trained VLAs with the targeted, goal-oriented optimization of Reinforcement Learning (RL). It takes a generalist VLA and **fine-tunes** it to excel at specific tasks, improving its success rate, efficiency, and robustness. Think of it as taking a broadly intelligent student (the VLA) and giving them specialized vocational training (RL) to become a master at a specific job.

The Problem VLA-RL Solves

Pre-trained VLAs are impressive zero-shot performers, but they have key limitations:

- **Brittleness:** They can fail on tasks that are perceptually or physically similar to their training data but require slightly different reasoning.
- **Sub-Optimality:** They may succeed at a task but in a slow, inefficient, or unnatural way (e.g., awkward gripper orientations, unnecessary movements).
- **No Improvement:** They don't learn from their own mistakes. If a VLA fails a task, it will likely fail the same way every time.
- **Reward-Agnostic:** They don't understand complex, non-binary goals like "be efficient," "be safe," or "minimize energy use."

VLA-RL addresses this by using RL to continue the learning process *in the target environment*, aligning the model's behavior with a precise performance metric (the reward function).

The Three Parts of "VLA-RL"

The name signifies the components and the method:

1. Visual (The "Pre-trained Eyes")

- **What it is:** The visual encoder from a pre-trained VLA (e.g., a CLIP-like model). This provides a rich, semantic understanding of the scene.
- **Role in VLA-RL:** This component is typically **frozen** (not updated during RL training). Its purpose is to provide a strong, general-purpose feature representation of the input image that the RL agent can build upon. It saves the RL algorithm from having to learn vision from scratch.

2. Language (The "Pre-trained Brain")

- **What it is:** The language and policy backbone of the pre-trained VLA. This is the model that has learned to associate visual scenes and language instructions with actions.
- **Role in VLA-RL:** This is the **core network that is fine-tuned**. It starts with a strong prior—it already knows general concepts like "grasp," "push," and "move to." RL training adapts and refines these existing skills for a specific purpose.

3. Action + RL (The "Specialized Trainer")

- **What it is:** The action-output layer of the VLA, now connected to a **Reinforcement Learning algorithm** (e.g., PPO, DDPG).
- **Role:** This is the learning mechanism. The RL algorithm uses a **reward function** to critique the VLA's actions and tell it how to improve.
    - The VLA acts as a highly intelligent **policy network initialization** for the RL algorithm.
    - The RL algorithm provides the **gradients** needed to tweak the VLA's weights to maximize cumulative reward.

Key Advantages of VLA-RL

- **Massive Sample Efficiency vs. RL-from-Scratch:** RL is notoriously data-hungry. Starting from a pre-trained VLA is like starting a race halfway to the finish line; it requires far fewer environmental interactions.
- **Better Final Performance:** It achieves higher success rates and more robust performance than the original VLA or an RL agent trained from scratch.
- **Alignment with Complex Goals:** Can optimize for nuanced objectives that are hard to demonstrate (e.g., "be energy-efficient," "be gentle with the object").

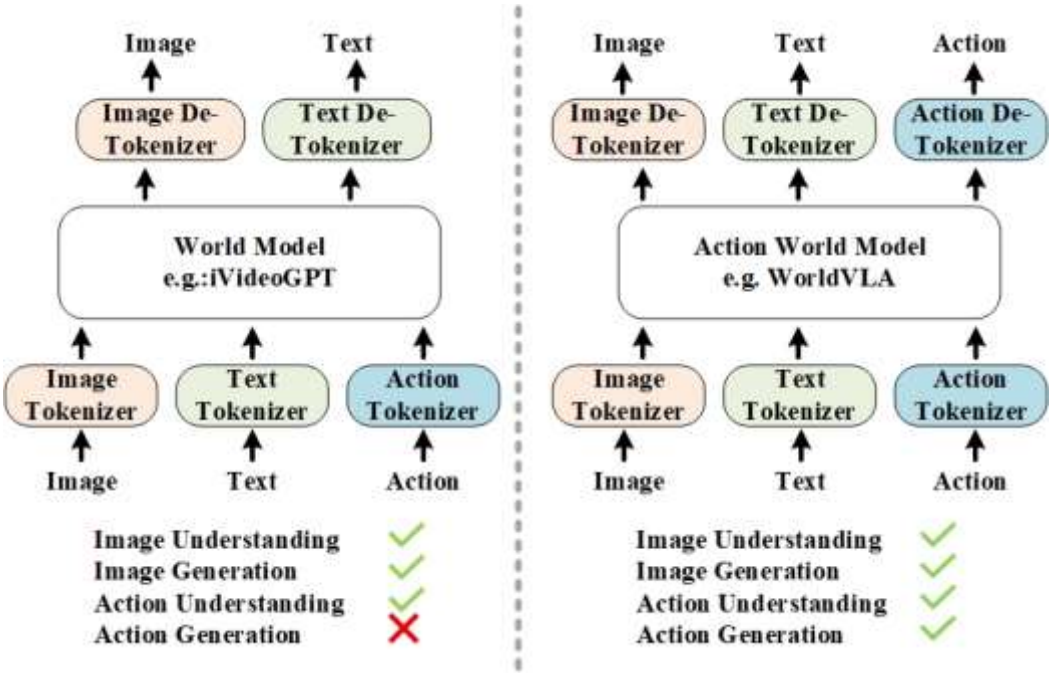
- **Adaptation:** Allows a general VLA to adapt to the specific dynamics of a new robot or a new environment it wasn't trained on.

Current Challenges and Limitations

- **Extremely Data Hungry**: Training from pixels to actions requires an enormous amount of interaction data, which is slow, expensive, and potentially dangerous to collect on real robots.
- **Reward Design: Designing** a good reward function is famously difficult. A poorly designed reward can lead to the agent learning to "hack" the reward system instead of doing the intended task (e.g., learning to cover a sensor instead of solving a problem).
- **Safety and Stability**: An end-to-end model is a black box. It's very hard to guarantee its behavior in all possible scenarios, making it risky for real-world deployment, especially around humans.
- **Sim-to-Real Gap**: Most training is done in simulation for speed and safety. Transferring the learned policies to a physical robot in the messy real world remains a significant challenge.

世界模型：迈向自回归动作世界模型

World Models : Towards Autoregressive Action World Model



核心理念概述

世界模型（World Model）是存在于机器人“心智”中的一种内部、可学习的环境模拟。与其直接对原始感知输入做出反应，机器人会利用这一模型来：

1. **预测**自身动作在未来可能产生的结果

2. **想象并评估**不同的动作序列，而无需在真实世界中逐一尝试
3. **选择**最优的动作序列以实现目标

本质上，它让机器人能够“先思考，再行动”——在一种类似梦境的内部空间中进行规划，而不是依赖现实世界中代价高昂的反复试错。

世界模型所解决的问题

传统机器人通常依赖“感知—规划—执行”的反应式循环，这种方式效率低且脆弱：

- **缺乏长远规划能力**：当前看似有利的决策，可能在 5 秒后变得不利
- **依赖现实试错**：必须在真实环境中尝试动作才能知道结果，这既缓慢，又可能带来安全风险或物理损坏
- **易受不完整或噪声观测干扰**：例如，当物体被暂时遮挡时，系统可能被误导

世界模型引入了前瞻性与推理能力，使机器人能够在内部先行验证假设。

世界模型的核心组成部分

一个典型的世界模型架构包含三个关键部分：

1. 表征（“观察者”部分）

- **输入**：高维的原始感知数据（如摄像头像素、机器人本体感知数据）
- **功能**：将庞大且嘈杂的现实观测压缩为紧凑、有意义且抽象的表示，这通常被称为**潜在状态**（latent state）或**潜在空间**（latent space）
    - 会舍弃无关细节（如阴影、纹理），保留预测未来所需的关键信息（如物体位置、速度及其关系）
    - 类比：人类不会记住场景中每一个像素的光线，而是记住“有一个杯子快要从桌子上掉下来”这样的概念化信息

2. 动态（“预测者”部分）

- **输入**：当前潜在状态与一个候选动作
- **功能**：这是世界模型的核心。动态模型预测**下一个**潜在状态，回答的问题是：“如果我处于状态 s 并执行动作 a，那么我的状态 s' 会是什么？”
    - 它学习环境的规则与物理规律，例如“推动积木会让它滑动”或“向左转动方向盘，车辆会向左行驶”
    - 该模型可以多次向前滚动，从而模拟未来的长序列情境

3. 预测（“规划者”部分）

- **输入**：当前状态表征与目标

- **功能**：该组件利用动态模型来想象不同的未来情境。最常用的方法是**模型预测控制**（Model Predictive Control，MPC）：
    1. **想象**：机器人生成许多随机的潜在动作序列
    2. **模拟**：利用动态模型在内部“展开”每个动作序列，预测每种潜在未来的结果
    3. **评估**：将这些预测结果与目标进行比较，找出最接近目标的动作序列
    4. **执行**：只执行最佳序列中的**第一个**动作
    5. **重复**：观察新的真实状态，更新状态表征，并重新开始这一过程

这种持续的重新规划机制使系统对模型误差具有较强的鲁棒性。

世界模型的主要优势

- **样本效率高**：可通过模拟的“想象”经验学习有效策略，减少真实环境交互次数
- **安全规划**：危险动作可在模拟中尝试并舍弃，而不会带来现实风险
- **处理部分可观测性**：潜在状态可学习推断隐藏信息（例如物体被遮挡时，模型仍能保持对其位置的合理预测）
- **泛化能力**：一个理解物理规律的优秀动态模型，可以泛化到未见过的新物体和新环境，只要底层规则相似

世界模型的主要劣势

- **“现实鸿沟”**：学习到的模型总会存在不准确性，预测中的小误差在长时间规划中可能累积成完全错误的计划
- **计算开销大**：每次决策运行成千上万次模拟非常耗时，除非模型极其高效
- **模型匹配难题**：很难同时构建既足够细致以支持精确低层控制，又足够快速以满足实时规划需求的世界模型

VLA-RL 的研究愿景

VLA-RL 是一个雄心勃勃的框架，旨在打造通用型机器人智能体。它的目标是构建一个直接将感知与语言连接到动作的单一模型，通过基于奖励的试错学习获得最优行为，最终实现能够在有人类活动的环境中，按照自然语言指令执行多种任务的机器人。这是 Google DeepMind、OpenAI、NVIDIA 等公司重点探索的研究方向之一。

The Core Idea in a Nutshell

A **World Model** is an internal, learned simulation of the environment inside a robot's "mind." Instead of reacting directly to raw sensory inputs, the robot uses this model to:

1. **Predict** the future consequences of its actions.

2. **Imagine** and evaluate possible action sequences without trying them in the real world.

3. **Choose** the best sequence of actions to achieve a goal.

In essence, it enables the robot to **"think before it acts,"** planning in a internal dream-like space rather than through costly trial and error in reality.

The Problem World Models Solve

Traditional robots often operate on a **reactive** sense-plan-act cycle. This can be inefficient and fragile:

* It struggles with long-term planning. What is good *now* might be bad in 5 seconds.

* It requires trying actions in the real world to see what happens, which is slow, and potentially dangerous or damaging.

* It can be fooled by incomplete or noisy observations (e.g., an object is temporarily hidden).

World Models introduce **forethought** and **reasoning**, allowing the robot to test hypotheses internally.

The Core Components of a World Model

A typical world model architecture consists of three key parts:

1. Representation (The "Observer" Part)

* **Input:** High-dimensional, raw sensory data (pixels from a camera, proprioception data).

* **Function:** This component compresses the vast and noisy real-world observation into a compact, meaningful, and abstract representation. This is often called a latent state or latent space.

* It throws away irrelevant details (e.g., shadows, textures) and keeps only the information crucial for predicting what happens next (e.g., object positions, velocities, their relationships).

* **Analogy:** You don't remember a scene as every pixel of light; you remember "a cup is about to fall off the table." The representation is that simplified, conceptual understanding.

2. Dynamics (The "Predictor" Part)

* **Input:** The current latent state and a proposed action.

* **Function:** This is the heart of the world model. The dynamics model predicts the *next* latent state. It answers the question: "If I am in state *s* and take action *a*, what will my state *s'* be?"

* It learns the rules and physics of the environment. For example, it learns that "if you push a block, it will slide," or "if you turn the wheel left, the car will go left."

* This model can be rolled forward multiple times to simulate long sequences into the future.

3. Prediction (The "Planner" Part)

* **Input**: The current state representation and a goal.

* **Function**: This component uses the dynamics model to imagine different futures. The most common method is Model Predictive Control (MPC):

1. **Imagine:** The robot generates many random sequences of potential actions.

2. **Simulate**: It uses the dynamics model to "roll out" each sequence in its internal model, predicting the outcome of each potential future.

3. **Evaluate**: It compares these predicted outcomes to the desired goal. Which sequence gets closest to the goal?

4. **Act:** It executes only the *first* action from the best sequence.

5. **Repeat**: It observes the new real state, updates its representation, and starts the process all over again. This constant re-planning makes it robust to errors in the model.

Key Advantages of World Models

* **Sample Efficiency:** They can learn effective policies with fewer real-world interactions because they learn from simulated "imagined" experiences.

* **Safe Planning:** Dangerous actions can be tried and discarded in the simulation without real-world consequences.

* **Handling Partial Observability:** The latent state can learn to infer hidden information (e.g., if an object is occluded, the model can still maintain a belief about where it *should* be).

* **Generalization:** A good dynamics model that understands physics can generalize to new objects and environments it hasn't seen before, as long as the underlying rules are similar.

Key Disadvantages of World Models

- **"Reality Gap":** The learned model will always have inaccuracies. Small errors in prediction can compound over long planning horizons, leading to completely wrong plans.
- **Computationally Expensive:** Running thousands of simulations for every decision is very slow unless the model is extremely fast.
- **Model Mismatch:** It's challenging to build a world model that is both detailed enough for precise low-level control and fast enough for real-time planning.

VLA-RL is a ambitious framework for creating general-purpose robotic agents. It aims to build a single model that directly connects perception and language to action, learning optimal behavior through trial-and-error reinforced by rewards, with the ultimate goal of creating

robots that can perform a wide range of tasks in human environments following natural instructions. It's a key research direction for companies like Google DeepMind, OpenAI, and NVIDIA.

大脑 + 小脑：生物进化的运动控制解决方案

Brain + Cerebellum biological solution for motor control

核心理念概述

“大脑 + 小脑”系统是自然界在进化过程中形成的精密、适应性极强的控制方案。它将问题拆分为两个互补的系统：

1. **大脑（新皮层）**：**“做什么”系统**。负责高层目标设定、战略规划与有意识的意图生成，产出期望的结果。
2. **小脑**：**“怎么做”系统**。负责执行平滑、精确且协调的动作，将高层目标转化为时机完美、低层级的运动指令。

这种分工模式带来了极高的适应性、效率与鲁棒性，在实时物理交互中远优于任何单一的、整体式 AI 架构。

大脑 + 小脑所解决的问题

在复杂的物理世界中执行协调动作（例如伸手去拿杯子）是一项极其庞大的计算任务，它需要：

- **精确性**：以毫秒级的时序控制数百块肌肉
- **适应性**：持续调整以应对负载、摩擦和身体状态的变化
- **预测性**：补偿感觉-运动延迟；不能等到真正绊倒才去纠正
- **高效性**：必须在潜意识中、实时完成，而无需有意识的思考

单一的通用型神经网络（如一个庞大的皮层）在执行这类任务时既缓慢又低效。大脑的解决方案是**专用化的硬件结构**——由大脑与小脑分工协作完成。

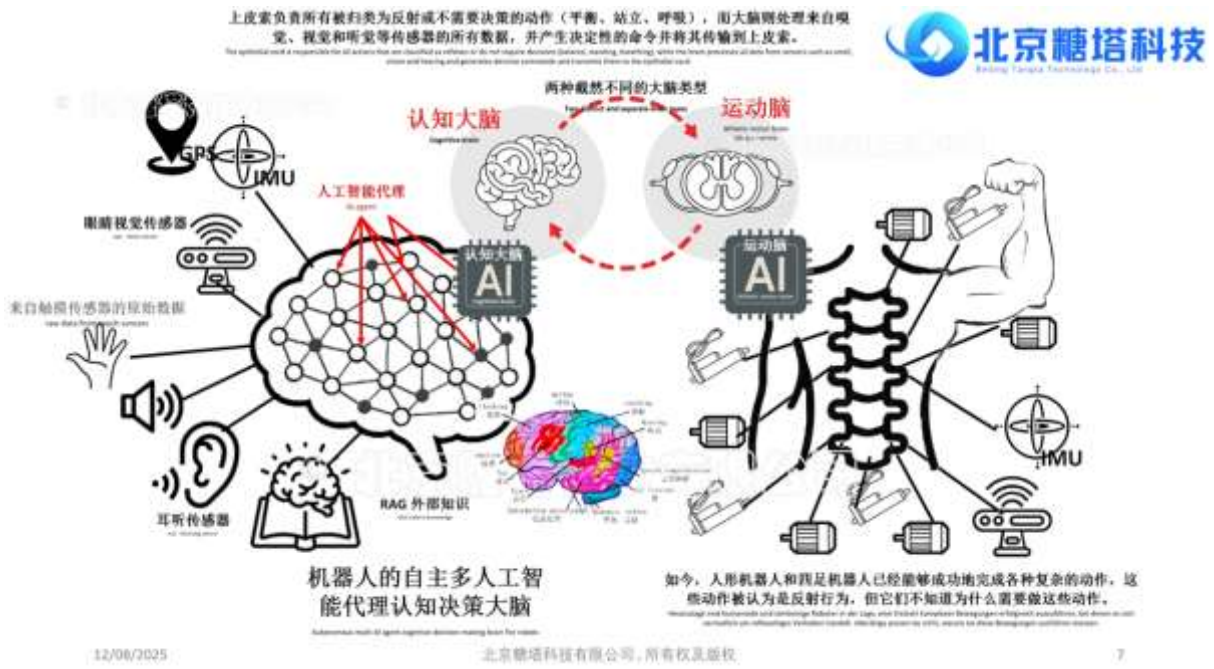
The Core Idea in a Nutshell

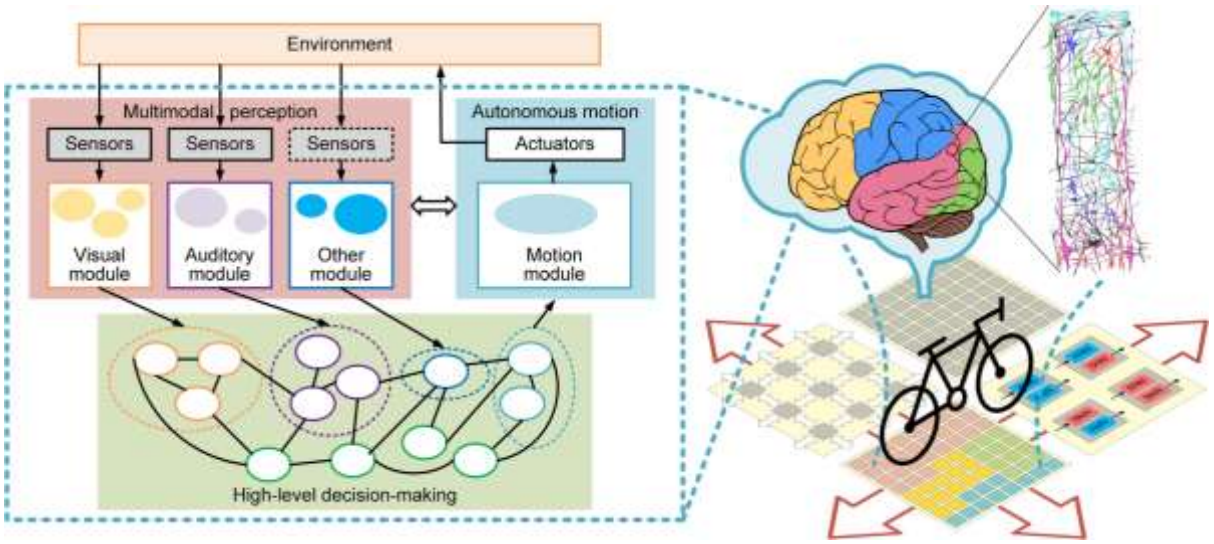
The **Brain + Cerebellum** system is nature's evolutionary solution for sophisticated, adaptive control. It separates the problem into two complementary systems:

1. **Cerebrum (Neocortex):** The "**What**" system. Responsible for high-level goal setting, strategic planning, and conscious intention. It generates a desired outcome.

2. **Cerebellum**: The "**How**" system. Responsible for executing smooth, precise, and coordinated movements. It translates a high-level goal into perfectly timed, low-level motor commands.

This separation of concerns allows for incredibly adaptive, efficient, and robust control that is far superior to any monolithic AI architecture for real-time physical interaction.





The Problem the Brain+Cerebellum Solves

Executing coordinated movement (like reaching for a cup) in a complex, physical world is an astronomically difficult computation. It requires:

* **Precision:** Controlling hundreds of muscles with millisecond timing.
* **Adaptation:** Constantly adjusting for changes in load, friction, and body state.
* **Prediction**: Accounting for sensorimotor delays; you can't wait to feel a stumble to correct it.
* **Efficiency**: Must happen subconsciously, in real-time, without conscious thought.

A single, general-purpose neural network (like a large cortex) is too slow and inefficient for this task. The brain's solution is specialized hardware.

“大脑 + 小脑”的两大组成部分

1. 大脑 / 新皮层（“战略指挥官”）

- **功能**：这是高级认知的中枢。在运动控制中，运动皮层（Motor Cortex）和前运动皮层（Premotor Cortex）等特定区域至关重要。它们的作用是发出“运动指令”或期望的目标状态。
    - **输入**：整合来自视觉、语言、记忆与规划的信息（例如：“我口渴了，所以我想喝那杯水”）
    - **处理**：形成意图并制定粗略计划（例如：“伸手去拿杯子”）
    - **输出**：向脊髓和小脑发送相对粗略的高层指令，告诉它们**“做什么”**（例如：“将手移动到这个位置”），但不涉及具体的**“怎么做”**

2. 小脑（“战术执行官”）

- **功能**：这是一个精密的感觉-运动预测与校准机器，其任务是完美执行大脑皮层的指令。
    - **输入**：接收来自皮层的指令副本（*efference copy*），同时接收来自身体的大量实时感知反馈（本体感觉、平衡、视觉）
    - **处理**：将**预期**动作（来自皮层）与**实际**感知反馈进行比较，实时检测预测与现实之间的误差（例如：“手臂偏离轨迹 2 毫米，速度慢了 5%”）
    - **输出**：在动作执行过程中发送修正信号，微调运动指令，使动作平滑、协调且精确。它负责具体的**“怎么做”**，本质上是一个经过学习的生物前向模型（forward model，一种世界模型）

大脑 + 小脑架构的主要优势

- **无意识高效性**：皮层无需对细节进行微观管理，你可以一边走路一边交谈
- **实时性能**：小脑针对超高速、潜意识计算进行了优化
- **鲁棒性与适应性**：可持续适应负载、疲劳、地形等变化
- **习得的流畅性**：通过练习，小脑会建立高度精确的内部模型，使得高技能动作（如弹钢琴）变得轻松自然

大脑 + 小脑架构的主要劣势

- **复杂的集成需求**：需要精心设计以确保各模块高效通信
- **潜在瓶颈**：高层大脑可能不了解低层控制器的限制，从而发出无法执行的指令
- **次优性**：模块化分工可能阻碍某些端到端系统中可出现的极致优化行为

大脑 + 小脑 —— 仿生式分工类比

- **大脑 / 新皮层**：深思熟虑的 **“做什么 / 为什么 / 何时”** —— 目标设定、任务分解、高层规划

- **小脑**：快速预测的 **“怎么做”** —— 平滑的低层控制，通过比较指令副本与感知现实进行误差修正

The Two Parts of "Brain + Cerebellum"

1. **Cerebrum / Neocortex (The "Strategic Commander")**

* **Function**: This is the seat of high-level cognition. In motor control, specific areas like the Motor Cortex and Premotor Cortex are crucial. Their role is to issue the "motor command" or the desired goal state.

* **Input:** Integrates information from vision, language, memory, and planning. (e.g., "I am thirsty, therefore I want to drink that water").

* **Process**: Formulates an intention and a rough plan. (e.g., "Reach for the cup").

* **Output:** Sends a relatively crude, high-level command down to the spinal cord and cerebellum. It says "what" to do (e.g., "move the hand to this location"), but not the precise "how."

2. **Cerebellum (The "Tactical Operator")**

* **Function**: This is an exquisite sensory-motor prediction and calibration machine. Its role is to execute the cortex's command flawlessly.

* **Input**: It receives a copy of the command from the cortex (*efference copy*). It also receives a massive stream of real-time sensory feedback from the body (proprioception, balance, vision).

* **Process**: It compares the *intended* movement (from the cortex) with the *actual* sensory feedback. It detects errors between prediction and reality in real-time (e.g., "the arm is 2mm off trajectory and moving 5% too slow").

* **Output**: It sends corrective signals to fine-tune the motor command *as it is happening*. It makes movement smooth, coordinated, and accurate. It handles the precise "how." It is essentially a learned, biological forward model (a type of world model).

Key Advantages of the Brain+Cerebellum Architecture

* **Unconscious Efficiency:** The cortex is freed from micromanaging details. You can have a conversation while walking.

* **Real-Time Performance:** The cerebellum is optimized for ultra-fast, sub-conscious computation.

* **Robustness and Adaption:** Continuously adapts to changing conditions (weight, fatigue, terrain).

* **Learned Smoothness:** Through practice, the cerebellum builds a highly accurate internal model, making skilled movements (like playing piano) effortless.

Key Disadvantages of the Brain+Cerebellum Architecture

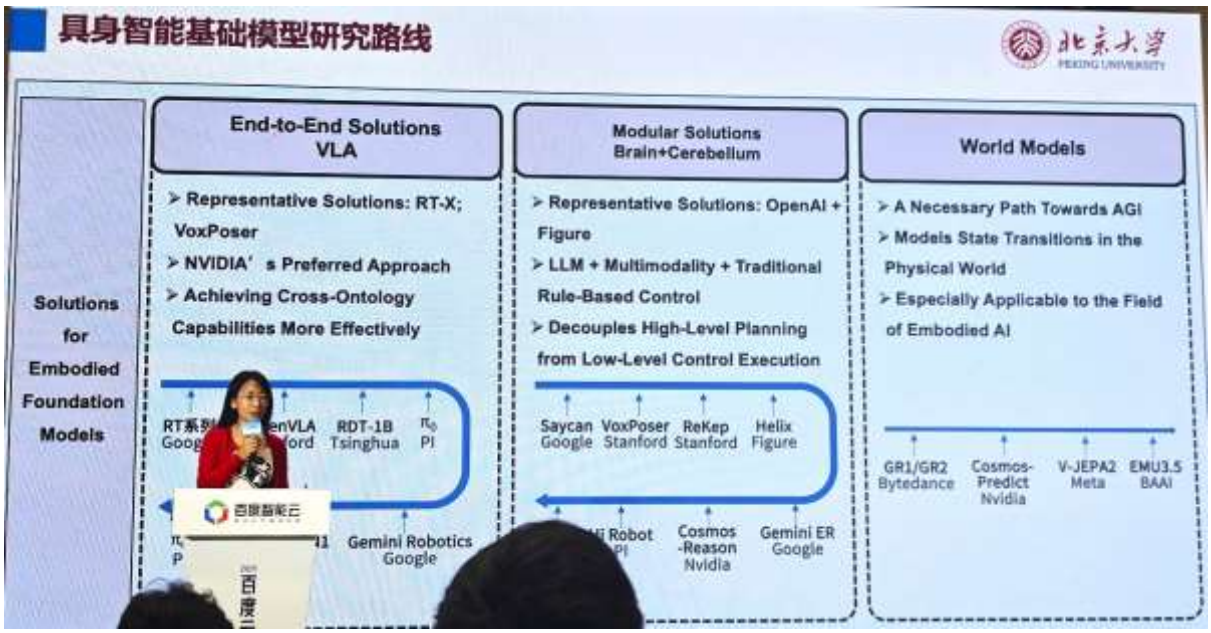
- **Complex Integration:** Requires careful engineering to make all modules communicate effectively.

- **Potential Bottlenecks:** The high-level brain might not be aware of the low-level controller's limitations, leading to commands that are impossible to execute.
- **Sub-Optimality:** The separation into modules might prevent the emergence of hyper-optimized behaviors that an end-to-end system might discover.

Brain + Cerebellum — Bio-Inspired Division of Labor

Analogy.

- **Cerebrum/Neocortex (Brain):** deliberative **what/why/when**—goal setting, task decomposition, high-level plans.
- **Cerebellum:** fast, predictive **how**—smooth low-level control, error correction from **efference copy** vs. sensed reality.



综合：它们如何协同工作

人形机器人控制的方向

当前在实用性与可扩展性方面最有前景的人形机器人控制方案，是一种融合三者优势的混合式方法：

1. **“大脑 + 小脑”架构** 是最务实的起点。像 1X Technologies、Figure AI 和 Sanctuary AI 等公司正大量采用这一范式。大型语言/视觉模型（大脑）负责理解世界并设定目标，而传统的稳健控制器（小脑）负责精确、安全的低层执行。这种方式安全、可调试，并能充分利用大语言模型（LLM）革命带来的优势。
2. **世界模型** 正被集成到中层规划中。“大脑”可以利用世界模型在执行前想象不同高层计划的结果（例如：“如果我用左手去够，会保持平衡吗？”），然后将最佳轨迹发送给“小脑”。这种方式比纯端到端学习更高效。

3. **端到端 VLA** 是长期目标，但目前更多作为组件而非完整系统使用。我们可能会看到端到端模型被训练为输出中层指令（如夹爪姿态轨迹），再交由稳定的控制器执行。从像素直接到力矩的纯端到端方式，在昂贵的人形机器人硬件上仍然过于不稳定且数据效率低。

未来最强大的 AI 智能体可能会融合三者的原理：

1. **VLA（类比大脑皮层）**：理解高层自然语言指令（如："帮我煮杯咖啡"）
2. **世界模型（类比前额皮层）**：将指令分解为计划，并在不实际尝试的情况下模拟步骤（如："首先找到水壶，然后加水……"）
3. **类小脑控制器（类比小脑）**：一个专用、快速且高度训练的子系统，接收规划好的动作（如："倒水"）并精确执行，同时在实时中适应滑动、重量变化和意外障碍

这种仿生式架构将结合 **VLA 的泛化能力**、**世界模型的前瞻性** 与 **小脑原理的稳健执行力**，为通用型人形机器人奠定坚实基础。

Synthesis: How They All Fit Together

Direction for Humanoid Control

The current front-runner for practical, scalable humanoid control is a **hybrid approach that combines the strengths of all three.**

1. **A "Brain + Cerebellum" architecture is the most practical starting point.** Companies like **1X Technologies**, **Figure AI**, and **Sanctuary AI** are heavily using this paradigm. A large language/vision model (the Brain) understands the world and sets goals, while a traditional robust controller (the Cerebellum) handles the precise, safe, low-level execution. This is safe, debuggable, and leverages the revolution in LLMs.
2. **World Models are being integrated for mid-level planning.** The "Brain" might use a world model to *imagine* the outcome of different high-level plans ("if I reach with my left hand, will I be balanced?") before sending the best trajectory to the "Cerebellum". This is more efficient than pure end-to-end learning.
3. **End-to-End VLAs are the long-term goal but are currently used as components, not the whole system.** We might see an end-to-end model trained to output *mid-level commands* (like gripper pose trajectories), which are then passed to a stable controller. Pure end-to-end from pixels to torques is still too unstable and data-inefficient for costly humanoid hardware.

The most powerful future AI agent will likely incorporate principles from all three:

1. **VLA (Cerebrum Analog):** To understand a high-level, natural language command ("Make me a coffee").

2. **World Model (Prefrontal Cortex Analog):** To break that command down into a plan and simulate the steps without physically trying them. ("First, find the kettle, then fill it with water...").

3. **Cerebellum-like Controller (Cerebellum Analog):** A dedicated, fast, and highly trained subsystem that takes the planned action (e.g., "pour water") and executes it with precision, adapting in real-time to slippage, weight changes, and unexpected obstacles.

This bio-inspired architecture would combine the **generalization** of VLAs, the **foresight** of World Models, and the **robust execution** of the cerebellar principle.

RoboBrain 2.0 —— 一个生态系统，而非单一模型

必须明确的是，“RoboBrain-2.0”并不是一个可以直接下载的、单一的庞大 AI 模型。更准确的理解是：它是一个庞大的开源项目与生态系统，旨在加速机器人学与具身智能（Embodied AI）的发展。它是构建与连接各种 AI 组件的基础性基础设施。

你可以将它类比为机器人 AI 领域的 **“Github”** 或 **“Android 开源项目（AOSP）”**。

1. 核心理念：机器人的 “AI 互联网”

RoboBrain-2.0 的核心论点是：没有任何一家实验室或公司能够单独构建出通用型机器人智能。它的目标是创建一个去中心化、由社区驱动的平台，让全球研究人员能够：

- **贡献（Contribute）**：分享他们训练好的 AI 模型（用于感知、操作、导航等）
- **组合（Compose）**：轻松将这些模型串联起来，形成复杂的机器人行为
- **使用（Use）**：无需从零开始训练，即可获取最先进的能力

它是一个由互联网上众多更小、更专用的“脑”连接而成的“大脑”。

2. 关键创新与组件

RoboBrain-2.0 基于多个区别于以往方法的核心理念：

a) RT-2 模型架构 在其核心，RoboBrain-2.0 常使用类似 **RT-2（Robotic Transformer 2）** 的模型。RT-2 是一种视觉-语言-动作（VLA）模型。

- **功能**：接收摄像头图像与自然语言指令（如：“把香蕉放进杯子里”），并输出低层级的机器人动作
- **独特之处**：它在海量的网络数据（图像与文本）与机器人数据（图像与动作）上联合训练。这使其能够将互联网的知识迁移到物理世界中，实现涌现式推理（例如，即使训练中从未执行过该动作，也能理解“香蕉”是什么，以及它可以被移动）

b) RPM（Robotics Primitive Modules）框架 这是其“可组合性”的核心。RoboBrain-2.0 提供了一个预训练模块库，可以像乐高积木一样自由组合。例如：

- **rpm_affordance**：预测物体可交互的位置与方式（如杯子应从哪里抓取）

- **rpm_tracking**：跟踪物体随时间的运动轨迹
- **rpm_caption**：用自然语言描述场景

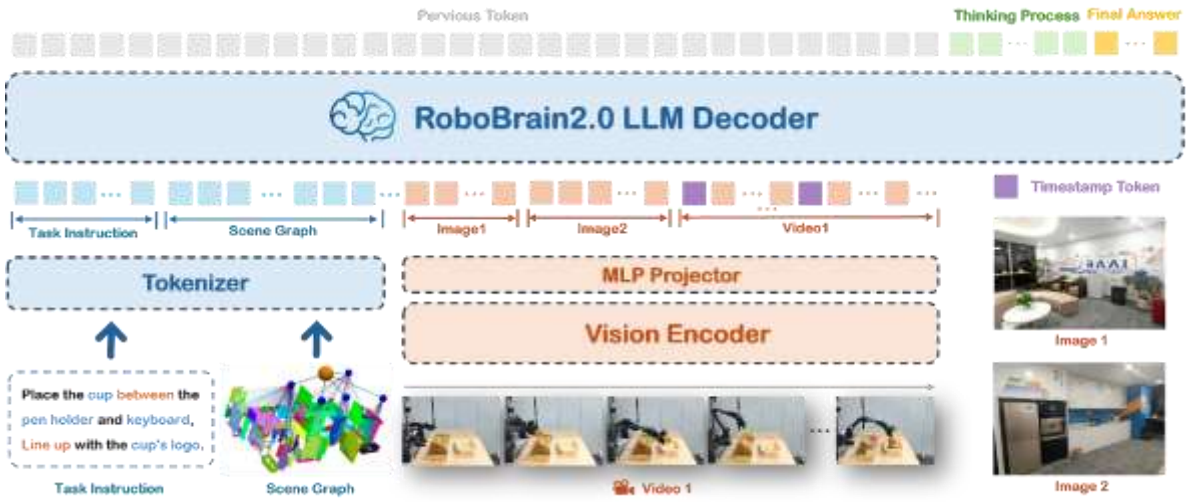
开发者可以编写一个简单脚本，将这些模块串联起来：识别物体 → 描述物体 → 计算抓取方式 → 执行抓取。

c) 开源与去中心化 这可能是它最具标志性的特征。该项目以开源方式构建，允许任何人贡献新模块或改进现有模块。这会产生网络效应：使用与贡献的人越多，整个生态系统就越强大、越有价值。

RoboBrain-2.0 — An Ecosystem, Not a Single Model

It's important to understand that "RoboBrain-2.0" isn't a single, monolithic AI model you can download. Instead, it's best thought of as a **massive, open-source project and ecosystem** designed to accelerate progress in robotics and embodied AI. It's a foundational infrastructure for building and connecting various AI components.

Think of it as the "Github" or "Android Open Source Project (AOSP)" for robotics AI



1. Core Concept: The "Internet of AI" for Robots

The central thesis of RoboBrain-2.0 is that no single lab or company can build general-purpose robot intelligence alone. The goal is to create a **decentralized, community-driven platform** where researchers worldwide can:

- **Contribute:** Share their trained AI models (for perception, manipulation, navigation, etc.).
- **Compose:** Easily chain these models together to create complex robotic behaviors.
- **Use:** Access state-of-the-art capabilities without training from scratch.

It's a "brain" built by connecting many smaller, specialized "brains" from across the internet.

2. Key Innovations and Components

RoboBrain-2.0 is built on several key ideas that differentiate it from previous approaches:

a) The rt-2 Model Architecture

At its heart, RoboBrain-2.0 often leverages models like **RT-2 (Robotic Transformer 2)**. RT-2 is a **Vision-Language-Action (VLA)** model.

- **What it does:** It takes in camera images and a natural language command (e.g., "move the banana to the cup") and outputs **low-level robot actions**.
- **How it's special:** It's trained on a massive mixture of **web data (images and text)** and **robot data (images and actions)**. This allows it to transfer knowledge from the vast internet to the physical world, enabling **emergent reasoning** (e.g., understanding what a "banana" is and that it can be moved, even if it never performed that exact action during training).

b) The rpm (Robotics Primitive Modules) Framework

This is the "composable" part. RoboBrain-2.0 provides a library of pre-trained modules that can be chained together like Lego blocks. Examples include:

- rpm_affordance: Predicts where and how an object can be interacted with (e.g., where to grasp a mug).
- rpm_tracking: Tracks an object's motion over time.
- rpm_caption: Describes a scene in natural language.
  A developer can write a simple script that chains these modules: see object -> describe it -> calculate how to grasp it -> execute grasp.

c) Open-Source and Decentralized

This is perhaps its most defining feature. The project is built to be open-source, allowing anyone to contribute new modules or improve existing ones. This creates a network effect: the more people use and contribute, the more powerful and valuable the entire ecosystem becomes.

Table 4 Performance across three temporal reasoning benchmarks. The best results among different models are highlighted in **bold**, while the second-best results are underlined.

Models / Metrics	Multi-Robot Planning				Ego-Plan2					RoboBench
	Super.	Rest.	House.	All ↑	Daily.	Hobbies.	Rec.	Work.	All ↑	Plan. ↑
General Baselines										
Gemini-2.5-Pro-preview-05-06	63.51	54.77	78.39	65.39	44.19	43.05	46.45	39.60	42.85	63.49
Gemini-2.5-Flash-preview-04-17	59.44	55.78	76.88	63.86	38.72	35.59	43.72	33.42	37.09	69.33
GPT-o4-mini-2025-05-16	63.32	55.28	78.89	65.50	47.61	35.93	42.62	37.13	41.11	70.01
GPT-4o-2024-11-20	77.89	67.34	79.40	74.50	47.38	40.00	44.81	35.64	41.79	68.60
Claude-Sonnet-4-2025-05-14	73.08	61.81	80.40	71.30	43.51	41.02	42.62	38.87	41.26	70.21
Qwen2.5-VL-32B-Instruct	67.84	61.81	75.38	68.00	**64.46**	51.53	57.92	50.00	56.25	45.92
Qwen2.5-VL-72B-Instruct	77.39	68.34	79.40	74.67	60.36	48.14	**63.39**	46.29	53.75	66.94
Embodied Baselines										
Cosmos-Reason1-7B	35.17	25.62	40.70	33.66	30.75	27.12	31.69	20.30	26.87	53.17
VeBrain-8B	41.70	35.67	39.69	38.83	31.79	35.31	31.19	34.43	27.30	46.77
Magma-8B	–	–	–	–	4.56	3.39	6.56	2.97	4.09	–
RoboBrain-7B-1.0	4.52	7.04	5.03	5.50	–	–	–	–	–	38.93
RoboBrain-3B-2.0	82.91	72.86	84.92	79.83	45.33	39.66	45.90	37.62	41.79	46.70
RoboBrain-7B-2.0	83.92	**77.39**	84.42	**81.50**	39.41	32.20	33.88	26.98	33.23	**72.16**
RoboBrain-32B-2.0	**84.42**	72.36	**85.43**	80.33	64.01	**53.22**	57.92	**52.48**	**57.23**	68.33

至关重要的是，**RoboBrain-2.0** 并不与这些技术相互排斥。它可以将像 **RT-2** 这样的 VLA 模型作为功能强大的“基础单元”纳入其模块库中。它是一个用于组织和管理这些模型的总体框架。

总结：RoboBrain-2.0 并不仅仅是又一个 AI 模型，而是一个雄心勃勃的开源基础设施项目，旨在通过无缝连接全球各地研究实验室的 AI 模型，为机器人创建一个集体的“脑”。它的目标是通过全球协作与可组合性，而非孤立竞争，来解决通用型机器人智能的问题。

Crucially, RoboBrain-2.0 is not mutually exclusive with these. It can *contain* VLA models like RT-2 as powerful "primitives" within its library. It's the overarching framework for organizing them.

In summary, RoboBrain-2.0 is not just another AI model. It is an ambitious, open-source infrastructure project aiming to create a collective "brain" for robots by seamlessly connecting AI models from research labs all over the world. Its goal is to solve the problem of general-purpose robot intelligence through global collaboration and composability, rather than isolated competition.

Feature	Visual-Language-Action (VLA)	VLA-RL (Reinforcement Learning)	World Model (e.g., Action Chunking)	Brain + Cerebellum (Bio-Inspired)	RoboBrain 2.0 (Unified Framework)
Core Principle	**Cross-modal translation:** Directly maps vision & language to actions.	**Specialized fine-tuning:** Optimizes a pre-trained VLA for task-specific	**Internal simulation:** Learns a predictive model of the world to plan.	**Separation of concerns:** Splits high-level planning from low-level	**Composition:** Integrates multiple specialized models into a single cognitive stack.
Primary Function	"What should I do right now?" (Next-best-action)	"How can I do this task **better**?" (Skill Optimization)	"What will happen if I do that?" (Future Prediction)	**Cortex:** "What to do?" **Cerebellum:** "How to do it?" (Execution)	**Orchestration:** "Which model should handle this part of the problem?"
Learning Paradigm	**Imitation Learning (Offline)**	**Online Fine-Tuning**	**Self-Supervised Learning**	**Mixed**	**Hybrid**
	Supervised on (obs, lang, action) triples.	Reinforcement Learning with a reward signal.	Learns from (obs, action, next obs) sequences.	Cortex: Imitation/RL	Leverages all paradigms (Imitation, RL, Self-sup.) for different components.
				Cerebellum: Supervised/RL on sensorimotor data.	
Key Strength	**Generalization:** Zero-shot execution of diverse language commands.	**Robustness & Performance:** Achieves high success rates on specific tasks.	**Foresight & Planning:** Enables long-horizon reasoning and avoids bad outcomes.	**Efficiency & Stability:** Provides millisecond, sub-conscious control for stable, adaptive movement.	**Versatility & Capability:** Combines the strengths of all approaches for the most general and capable architecture.
Primary Weakness	**Brittle:** Can fail unexpectedly; no understanding of physics. Sub-optimal trajectories.	**Reward Design:** Requires a well-defined reward function. Risk of catastrophic forgetting.	**Model Inaccuracy:** Prediction errors compound over long horizons. Computationally expensive.	**Integration Complexity:** Designing the interface between high and low-level layers is challenging.	**System Complexity:** Extremely complex to design, train, and deploy. Significant compute required.
Temporal Focus	**Short-horizon.** Reactive, based on current observation.	**Medium-horizon.** Optimizes over a sequence of actions for a task.	**Long-horizon.** Can simulate many potential futures.	**Continuous.** Cortex: plans. Cerebellum: operates in a continuous *now*.	**Full-spectrum.** Handles long-horizon planning, medium-horizon tasks, and real-time control.
Data Source	Large datasets of human demonstrations (e.g., YouTube, teleoperation).	**Interaction with environment** (real or sim) to collect trial-and-error data.	Unlabeled interaction data: (obs, action, next obs) tuples.	**Cortex:** Language, vision, demonstration data. **Cerebellum:** Proprioceptive, sensorimotor data.	**All available data.** Curated datasets for each component model within the architecture.
Real-World Analogy	A **smart intern** with extensive textbook knowledge but no practical experience.	A **specialist surgeon** refining their textbook knowledge through years of practice.	An **architect** who tests designs in a simulation before building.	**You walking while talking:** Your cortex holds the conversation, your cerebellum keeps you from falling.	A **full company team** (CEO, managers, architects, operators) working in perfect sync.
Role in Hierarchy	**High-Level Policy**	**Optimized Policy**	**Planner / Predictor**	**Low-Level Controller**	**Meta-Controller** (The "Manager" of the other models)

综合 —— 面向人形机器人的实用混合式技术栈

1. **指令理解（VLA / 大语言模型）**：解析目标与约束条件；将语言与感知进行语义对齐。
2. **前瞻规划（世界模型）**：模拟候选计划；选择可行、安全的轨迹或子目标。

3. **执行控制（类小脑控制器）**：低延迟控制，结合预测性误差修正与反射机制。
4. **专门化（VLA + 强化学习）**：针对任务、机器人本体与环境进行在线或分阶段微调。
5. **组合编排（RoboBrain 2.0）**：将感知、抓取、导航、语音等共享模块编排成完整行为。

结果：通过 **VLA** 获得泛化能力，通过 **世界模型** 获得前瞻性，通过 **类小脑控制** 获得实时鲁棒性，通过 **强化学习** 获得任务级最优性——并通过一个可组合的生态系统将它们无缝整合在一起。

Synthesis — A Practical Hybrid Stack for Humanoids

1. **Instruction Understanding (VLA/LLM).** Parse goals and constraints; ground language in perception.
2. **Foresight (World Model).** Simulate candidate plans; select a feasible, safe trajectory or sub-goals.
3. **Execution (Cerebellar Controller).** Low-latency control with predictive error correction and reflexes.
4. **Specialization (VLA-RL).** Online/episodic fine-tuning to the task, robot, and environment.
5. **Composition (RoboBrain-2.0).** Orchestrate shared modules (perception, grasping, navigation, speech) into full behaviors.

Result: Generalization from VLAs, foresight from World Models, real-time robustness from cerebellar control, and task-level optimality from RL—stitched together via a composable ecosystem.

结论：通向具身智能的混合之路

通用型机器人的发展正逐渐汇聚到一个关键洞见：解决方案既不是单一、庞大的 AI 模型，也不是一组彼此割裂的模块化程序。未来的方向在于**混合式、仿生启发的架构**——将 VLA 的语义理解能力、世界模型的前瞻与规划能力、VLA-RL 的稳健优化能力，以及“大脑 + 小脑”结构的实时稳定控制能力有机整合。

这种集成方法并非停留在理论层面，而是正在由工业研发领域的创新型微型企业积极探索与实践。一个典型案例是位于北京海淀的 **北京糖塔科技有限公司**，该公司已获得 2024、2025 年的科技型中小企业认证，以及 2025–2027 年的创新型中小企业认证。

该公司正在研发基于模块化 **“大脑 + 小脑”** 结构的下一代认知大脑，其研究明确借鉴了生物智能的分工模式：

- **“大脑”模块**：作为高层指挥官，整合视觉与语言（类似 VLA 的功能）以理解任务，并进行深思熟虑的规划（类似世界模型的功能）
- **“小脑”模块**：作为专用的低层控制器，确保动态平衡、柔顺力控，以及毫秒级的调整，以实现安全稳定的行走与操作

通过这种系统架构，北京糖塔科技正直面现代机器人学的核心挑战：**将高层推理与低层可靠性统一起来**。他们的成果已在 2025 年的多项高新技术竞赛中获得认可，充分体现了务实的产业化路径——将专用子系统组合成统一的认知堆栈，这与 **RoboBrain 2.0** 等框架的愿景高度契合。

这一趋势印证了一个事实：最强大的机器人不会只有一个“大脑”，而是由多个擅长不同领域的“模型社会”组成，每个组件都在其特定角色中发挥最佳性能。

为加速这一技术愿景的实现，北京糖塔科技正积极寻求天使投资。对于希望参与前沿具身智能领域的投资者与合作伙伴，我们诚挚邀请您与我们联系，共同探索这一基础性技术的巨大潜力。

Conclusion: The Hybrid Path to Embodied Intelligence

The journey toward general-purpose robots is converging on a critical insight: the solution is neither a single, monolithic AI model nor a collection of disconnected modular programs. The future lies in **hybrid, bio-inspired architectures** that integrate the distinct strengths of VLAs for semantic understanding, World Models for foresight and planning, VLA-RL for robust optimization, and a Brain+Cerebellum structure for real-time, stable control.

This integrated approach is not merely theoretical; it is being actively pioneered by innovative micro-SMEs in industrial R&D. A prime example is **Beijing Tangta-Technology**, a technology-certified SME (2024, 2025) and certified innovative SME (2025-2027) based in Beijing Haidian. The company is developing a next-generation cognitive brain based on a modular **"Brain + Cerebellum"** structure. Their research explicitly mirrors the biological separation of intelligence:

- The **"Brain"** module acts as the high-level commander, integrating vision and language (VLA-like functions) for task comprehension and deliberate planning (World Model-like functions).
- The **"Cerebellum"** module functions as a dedicated, low-level controller, ensuring dynamic balance, compliant force control, and millisecond-level adjustment for safe and stable locomotion and manipulation.

By architecting their system this way, Tangta-Technology is tackling the core challenge of modern robotics: uniting high-level reasoning with high-level reliability. Their work, exemplifies the pragmatic industrial path forward: **composing specialized sub-systems into a unified cognitive stack**, much like the vision of frameworks such as RoboBrain 2.0.

This trend confirms that the most capable robots will not have one mind, but many—a cohesive "society of models" where each component excels at its specific role. To fasten the development of this technology vision, Tangta-Technology is actively seeking angel investment. For investors and partners looking to engage with cutting-edge embodied AI, we encourage you not to hesitate to take contact with us and explore the potential of this foundational technology.