大模型范式模块

对于 VLM 来说，这就是一道看图答题。只不过"答案"不再是一句话，而是一串代表动作的数字。

3.3 关键训练技巧：共同微调 (Co-Fine-Tuning)

这是论文中**非常重要的工程细节**。

训练策略	做法	效果
从头训练 (from scratch)	不用预训练权重，直接用机器人数据训	❌ 非常差（5B 模型几乎不work）
纯微调 (fine-tuning)	只用机器人数据微调预训练模型	⚠️ 还行，但会遗忘互联网知识
共同微调 (co-fine-tuning) ✅	同时用互联网数据 + 机器人数据微调	✅ 最好！既保留常识又学会动手

> 💡 **比喻：** 这就像一个英语老师去学日语。如果他完全放下英语只学日语（纯微调），时间久了英语会退步。但如果他每天一半时间教英语、一半时间学日语（共同微调），两门语言都能保持好。这在深度学习里叫**灾难性遗忘 (Catastrophic Forgetting)**，共同微调就是对抗遗忘的策略。

灾难性遗忘这个概念已经见了很多次了

三个芯片特点

维度	CPU (通用处理器)	GPU (图形处理器)	TPU (张量处理器)
打比方	能够处理各种杂事的“管家”	擅长大量简单计算的“工人队”	专门为矩阵运算设计的“超级工厂”
设计初衷	运行操作系统、各类软件	渲染图形、并行计算	**极速处理深度学习任务**
强项	复杂的逻辑判断	视频渲染、通用的算力加速	**大规模模型训练与高吞吐推理**
开发者	Intel / AMD	NVIDIA (英伟达)	**Google (谷歌)**

3. 数值转换：从 Token 到物理量

如果你看的是 RT-2 这种大模型，它有一个独特的“翻译”步骤：

- **Token 化预测**：RT-2 不直接计算 0.5 这样的数字，而是预测 7 个连续的 **Token**。
- **映射逻辑**：它将机器人的动作范围（比如 -1.0 到 1.0）映射到 0-255 的整数空间。
  - 例如：预测出的 Token 是 `128`，翻译回来就是 `0.0`（中间位置）。
- **为什么是 7 个？** 因为对于一个标准的 6 自由度机械臂，我们需要：
  1. **位置**：x, y, z（空间坐标）。
  2. **姿态**：$roll, pitch, yaw$（旋转角度）。
  3. **末端执行器**：$gripper$（夹具开合）。

将这个token转换为这个bins表示的时候，最终这个bins可以代表这个具体的动作序列

十二、RT 系列发展脉络 🧬

RT-1 (2022)
35M 参数，专门为机器人设计的 Transformer
能力：跟着指令做简单抓放
局限：不理解常识，泛化差
▼
RT-2 (2023) 👈 本文
5B~55B 参数，基于互联网 VLM
突破：动作即语言，常识迁移，涌现能力
局限：闭源，计算贵，不能学新动作
▼
RT-X / Open X-Embodiment (2023)
跨机器人的统一数据集 + 统一模型
▼
OpenVLA (2024)
开源的 VLA，你可以实际跑的模型

看了这么多文章，相当于是将这个发展大致看完了

Palm-E模型

总结

- **PaLM-E** 解决了机器人“**看懂并思考**”的问题。
- **RT-2** 解决了机器人“**思考并直接动手**”的问题。

你目前在读的 PaLM-E 是这一切的基础。理解了 PaLM-E 是如何把图像和文本放在一个“空间”里思考的，你就理解了为什么 RT-2 能这么直接地控制机器人。

既然你在研究 10 天学习计划，需要我帮你把 RT-2 的核心技术点也加入到你的笔记中吗？

这是两个模型的特点

PaLM-E 最核心的 3 个贡献：

1. **多模态句子**——把图像/状态/文字全部统一成 token 塞进 LLM，设计极其优雅
2. **正迁移效应**——混合训练（机器人 + VQA + 网络图文）反而让每个单独任务都变好，机器人数据只占 8.9% 就够了
3. **规模抵抗遗忘**——562B 模型学了多模态后只丢了 3.9% 的语言能力

对你课题最直接的启发：

- **状态编码进 LLM**：工业机械臂的传感器数据（力矩、夹爪角度等）可以用 State MLP 编码成 token，辅助消歧判断
- **Entity Referrals → 消歧占位符**：用特殊 token 指代候选实体，注入视觉特征后让模型做消歧推理
- 但对于你的课题，**模块化方案比端到端更实际**（需要可解释性 + 数据量有限）

有任何部分想深入讨论的，随时问我！

Palm e 阅读笔记
Document · MD
Download

这个对我们启发最大的还带是这一点，将这个状态也，传感器状态也转换为这个token序列，那么万事万物是不是都可以进行这个转换，然后对所有的数据进行处理，得出结果，大一统

- 移动操作：混合训练显著超过 SayCan（用了 oracle affordance）

> **哲学启示：** 这印证了"博学多才"的智慧——学习看似无关的知识（VQA、图像描述）反而能帮助解决机器人任务。这类似于人类的"通识教育"：学文学的工程师可能比只学工程的更有创造力，因为不同领域的知识会发生正向迁移。

对你课题的启发： 你在做工业机械臂消歧时，不要只用机器人数据训练。混入通用的 VQA 数据、指代表达理解（RefCOCO）数据，可能会让消歧能力更强。

真是有意思哈哈哈

💡 启发 2：端到端 vs 模块化的权衡

方面	PaLM-E（端到端）	SayCan（模块化）
Grounding 方式	隐式（训练中学到）	显式（affordance 模型）
灵活性	高（一个模型搞定）	高（模块可替换）
可解释性	低（黑箱）	高（知道为什么选这个）
对你课题的适用性	需要大量数据	**更适合**——消歧需要可解释性

确实端到端的，通过大量数据训练出来的结果，这个可解释性质会降低

💡 启发 3：Entity Referrals → 消歧占位符

PaLM-E 的 Entity Referrals 机制给了一个关键启发：

原始 PaLM-E 用法：
"<obj1> is to the left of <obj2>"
↑ 每个标记被替换成该物体的视觉编码

你的消歧系统可以这样用：
"把 <candidate1|螺丝A|置信度0.8> 拧到 <candidate2|孔位B|置信度0.6>"
↑ 每个候选实体附带视觉特征和置信度
→ 当置信度接近时触发澄清："你指的是左边还是右边的螺丝？"

这个启发可以呀

---

5) 这背后的“哲学设计思想”（顺便点一下）

PaLM-E 这类工作背后信奉的是：

> **与其为每个机器人/每个任务从零造一个专用模型，不如训练一个“通用底座”，再用少量数据适配。**
> 也就是“从专用系统 → 通用模型 + 轻量适配”的路线。

通用底座，少量配置，真是非常精彩的思想，我们写论文一般，除非天才，否则都是遵循这个思想，在现有的基础下作少量的突破

你问得很关键：**horizon（地平线/时域）**几乎是具身智能里“任务难不难”的核心词之一。

---

1) Horizon 在具身智能里到底是什么意思？

先把它翻成最“顾名思义”的中文：

> **horizon = 你需要“往未来看多远”才能把事做完的时间跨度（时域）**
> 也可以理解为：**一个任务要跨多少步（多少个动作/决策）才能完成。**

两个最常见搭配

- **short-horizon（短时域/短程任务）**：
  1～3 个动作就能完成，比如“把积木推一下”“抓起杯子放到旁边”。
- **long-horizon（长时域/长程任务）**：
  要很多步，而且中间会遇到变化/失败，需要不断修正，比如“去抽屉拿薯片给我”。

你可以把它类比成打游戏：

- 短时域：一枪一个小怪
- 长时域：做一个主线任务（要跑地图、找钥匙、开门、拿道具、回来交任务，中间还可能被打断）

将这个horizon翻译成这个时域就变得非常好理解了，和我们学习的这个diffusion policy相对应

7. Put it down（放下）
   → 最后 success

它要强调的点：

> 长时域任务里环境会变、会被打断；PaLM-E 能“看见变化 → 调整下一步”，而不是死按原计划走。

这就是 long-horizon 的难点：**不是步数多而已，而是“要能扛变化”。**

长时域的完成是要能抗变化，这个思想非常精彩