Open X-Embodiment & RT-X 论文阅读笔记

> **论文**: Open X-Embodiment: Robotic Learning Datasets and RT-X Models (2023)
> **阅读目的**: Day 6 - 理解多机器人数据如何统一格式，为构建工业数据集做准备
> **与课题关系**: ⭐⭐⭐⭐⭐ 直接相关（数据组织范式）

一句话总结

22种机器人 + 21个机构 + 100万+轨迹 = 统一格式的大规模数据集，证明了"混合训练"比"单独训练"效果更好。

核心贡献（你需要记住的）

贡献	内容	对你的价值
Open X-Embodiment Dataset	1M+ 轨迹，22种机器人，60个数据集	数据格式参考模板
RT-1-X	35M参数，多机器人联合训练	小数据场景的baseline
RT-2-X	55B参数，VLM+机器人动作	理解大模型如何做控制

🔑 核心技术点

1. 数据格式统一（Data Format Consolidation）

问题：不同机器人的观测和动作空间差异巨大

解决方案：粗对齐（Coarse Alignment）

```
python

# 统一的动作格式（7维）
action = [
    x, y, z,         # 末端执行器位置/位置变化
    roll, pitch, yaw,  # 末端执行器旋转/旋转变化
    gripper          # 夹爪开合
]
# 注意：可以是绝对值，也可以是相对值（delta）
```

关键设计：

- 选择一个**主相机视角**作为输入（不是所有相机）

- 动作**先归一化**再离散化（256个bin）
- **不强制对齐坐标系**——让模型自己学习不同机器人的差异

> 💡 **对你的启发**：你的工业数据也不需要完美对齐，粗对齐+归一化就够了

---

2. 动作离散化（Action Tokenization）

```python
# 每个维度离散化为256个bin
action_dim = 7 + 1  # 7维动作 + 1维终止信号
bins_per_dim = 256

# 示例：连续动作 → 离散token
continuous_action = [0.02, -0.01, 0.0, 0, 0, 0.1, 1.0]
discrete_tokens = [132, 125, 128, 128, 128, 140, 255]  # 每个值映射到0-255
```

> 💡 **对你的启发**：动作离散化让VLM可以像生成文本一样生成动作

---

3. 跨机器人迁移（Cross-Robot Transfer）

核心发现：

场景	RT-1-X vs 原方法	结论
小数据集（<1万条）	+50% 成功率	✅ 显著提升
大数据集（>10万条）	持平或略低	需要更大模型

Emergent Skills（涌现技能）：

- RT-2-X 在 Google Robot 上展示了它**从未见过**的技能
- 这些技能来自 Bridge 数据集（WidowX 机器人）
- 证明：**技能可以跨机器人迁移**

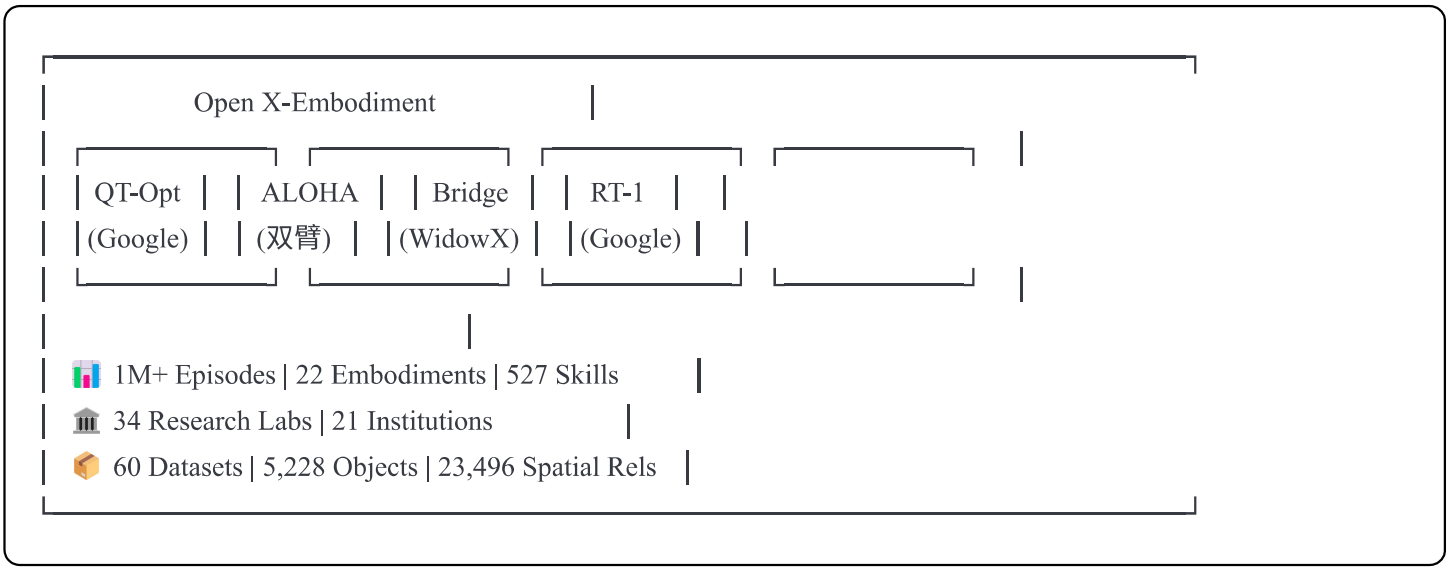
> 💡 **对你的启发**：即使你的工业机械臂数据很少，混入其他数据也能提升性能

📊 必读图表（按重要性排序）

> ⚠️ **阅读建议**：以下4张图是论文精华，务必看懂！

🥇 Figure 1: 数据集全景图（第1页）

这张图告诉你"数据从哪来"



你需要注意的：

- 数据来源多样：从单臂（Franka）到双臂（ALOHA）到移动机器人（Jackal）
- 技能类型：pour（倒）、stack（堆）、route（走线）、pick（抓取）...
- **工业相关数据集**：Cable Routing（走线）与你的工业场景最相关

🥈 Figure 2: 数据集统计分析（第3页）

这张图告诉你"数据长什么样"

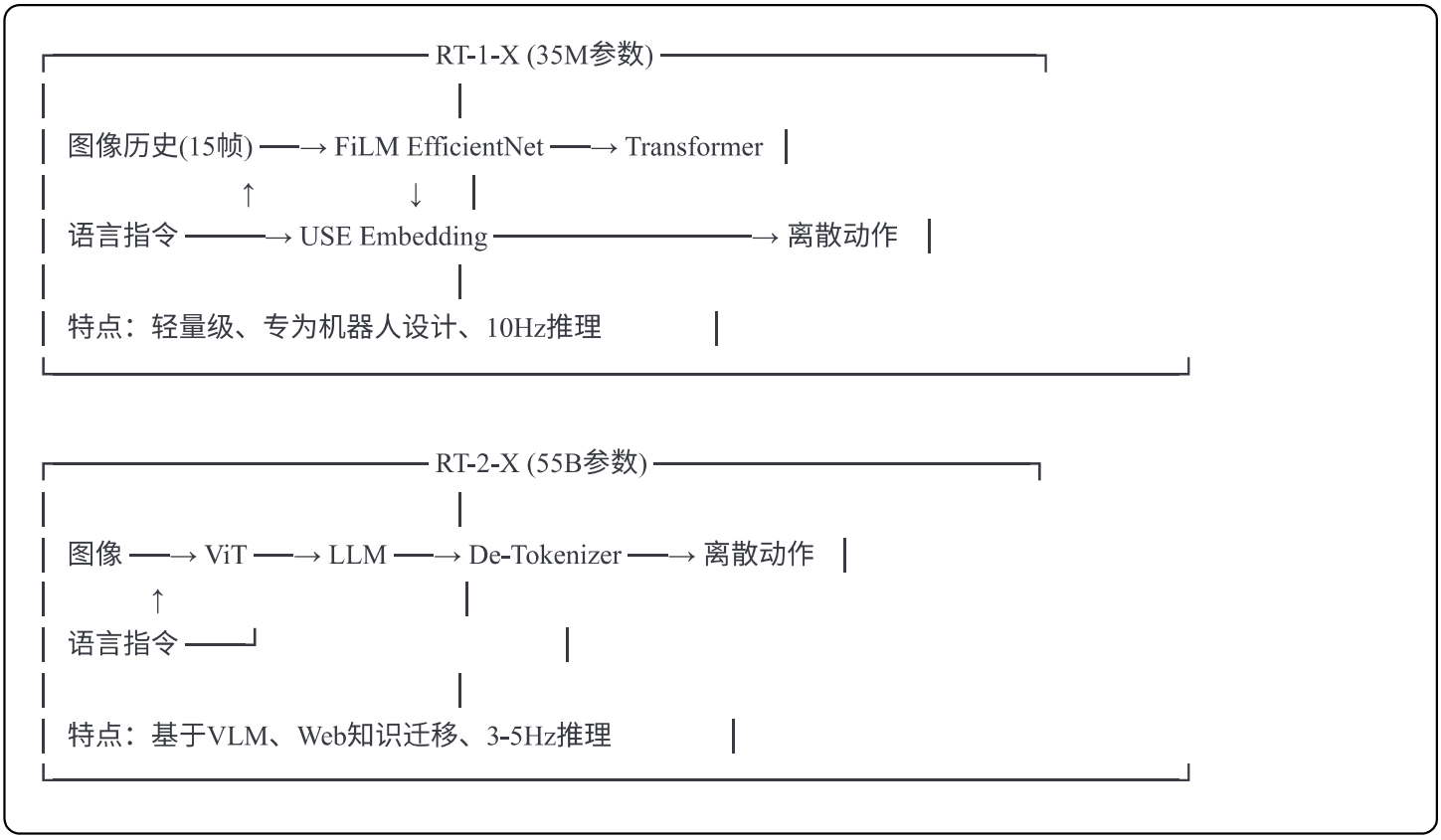
子图	内容	关键数据	对你的启发
(a) Datasets per Embodiment	各机器人的数据集数量	Franka=25, Google=6, xArm=5	Franka生态最丰富
(b) Scenes per Embodiment	各机器人的场景多样性	Franka场景最多样	场景多样性很重要
(c) Trajectories per Embodiment	各机器人的轨迹数量	xArm≈150K, Google≈130K	数据量差异巨大
(d) Common Skills	技能分布	picking >> moving > pushing	抓取是主流技能
(e) Common Objects	物体分布	Shapes, Containers, Food...	物体类别丰富

关键洞察：

- 数据**极度不平衡**：少数大数据集占主导
- 技能分布**长尾**：pick/place占多数，wiping/assembling很少
- **你的机会**：工业装配任务（assembling）数据稀缺，有创新空间

🥉 Figure 3: RT-1-X 与 RT-2-X 架构对比（第4页）

这张图告诉你"模型怎么工作"



动作输出格式（两者相同）：

```python
action_token = [x, y, z, roll, pitch, yaw, gripper, terminate]
# 每维离散化为256个bin
# 示例："1 128 91 241 5 101 127 0" (RT-2-X输出为文本)
```

你需要理解的：

- RT-1-X：**小而快**，适合实时控制（10Hz）
- RT-2-X：**大而强**，适合复杂推理但慢（3Hz）
- 两者都把动作当成"语言"来生成

🥇 Figure 4 & Table I: 核心实验结果（第5页）

这张图告诉你"混合训练到底有多大提升"

小数据集结果（Figure 4）

数据集	原方法	RT-1	RT-1-X	提升
Kitchen Manipulation	43%	48%	**63%**	+47%
Cable Routing	24%	18%	**56%**	+133% 🔥
NYU Door Opening	53%	65%	**80%**	+51%
Autolab UR5	45%	53%	**53%**	+18%
Task-Agnostic Play	33%	41%	**68%**	+106% 🔥
平均	41%	44%	**63%**	**+50%**

> 💡 **核心结论**：小数据场景下，混合训练带来**平均50%的提升**！

大数据集结果（Table I）

评估场景	RT-1	RT-1-X	RT-2-X (55B)
Bridge (Stanford)	40%	27% ❌	**50%** ✅
Bridge (UCB)	30%	27% ❌	**30%**
RT-1 6 skills	92%	73% ❌	**91%** ✅

> 💡 **核心结论**：大数据场景下，需要**更大的模型**（55B）才能受益

🌟 Figure 5 & Table II: 涌现技能实验（第6页）

这张图告诉你"技能真的能跨机器人迁移"

测试场景：在 Google Robot 上测试 Bridge 数据集的技能

```
Bridge数据集（WidowX机器人）的技能：
├── (a) Absolute Motion: "move chip bag to top/bottom right"
├── (b) Object-Relative Motion: "move apple between coke and cup"
└── (c) Preposition Alters Behavior: "put apple on/near cloth"

这些技能 Google Robot 从未见过，但 RT-2-X 能做到！
```

关键消融实验（Table II）

模型	涌现技能成功率	泛化评估
RT-2 (只用Google数据)	27.3%	62%
RT-2-X (混合数据)	**75.8%** 🔥	61%
RT-2-X (去掉Bridge)	42.8%	54%

> 💡 **核心结论**：Bridge数据确实帮助Google Robot学会了新技能，**涌现技能提升3倍**！

📌 图表速查表

图表	页码	核心信息	必看程度
Figure 1	p.1	数据集全景、来源多样性	⭐⭐⭐⭐⭐
Figure 2	p.3	数据统计、分布分析	⭐⭐⭐⭐⭐
Figure 3	p.4	RT-1-X vs RT-2-X 架构	⭐⭐⭐⭐
Figure 4	p.5	小数据集实验结果	⭐⭐⭐⭐⭐
Table I	p.5	大数据集实验结果	⭐⭐⭐⭐
Figure 5	p.6	涌现技能可视化	⭐⭐⭐⭐
Table II	p.6	消融实验、设计决策影响	⭐⭐⭐⭐⭐

🎯 与你课题的直接联系

1. 数据格式设计参考

```python
```

```
# RT-X 的数据格式（你可以参考）
episode = {
    "observation": {
        "image": array,          # 主相机图像
        "state": array,          # 机器人状态（可选）
    },
    "action": array,             # 7维归一化动作
    "language_instruction": str,  # 语言指令
    "is_terminal": bool,         # 是否结束
}
```

2. 你的工业消歧数据需要额外加的字段

python

```
# 在RT-X基础上，为消歧任务添加
episode_disambiguation = {
    # === RT-X 基础字段 ===
    "observation": {...},
    "action": [...],
    "language_instruction": "把那个螺丝拧到那里",

    # === 消歧特有字段 ===
    "is_ambiguous": True,
    "ambiguity_type": ["referent", "spatial"],
    "candidate_objects": [
        {"name": "螺丝A", "bbox": [...], "confidence": 0.85},
        {"name": "螺丝B", "bbox": [...], "confidence": 0.82},
    ],
    "ground_truth_object": "螺丝B",
    "clarifying_question": "你指的是左边还是右边的螺丝？",
    "user_response": "右边那颗",
}
```

3. 关键启示

RT-X 的做法	对你的启发
粗对齐即可	不需要完美标注，先跑起来
混合训练有益	可以混入公开数据集增强泛化
语言指令是关键	指令的多样性很重要
小数据受益最大	你的工业场景正好是小数据

✅ Day 6 检验清单

- [ ] 能说出 RT-X 的统一动作格式（7维是哪7个）
- [ ] 理解"粗对齐"的含义（不强制坐标系对齐）
- [ ] 知道小数据集场景下混合训练的优势（+50%）
- [ ] 能画出你自己数据集的格式草图

📎 资源链接

- 项目主页: https://robotics-transformer-x.github.io/
- 论文: https://arxiv.org/abs/2310.08864
- 数据格式: RLDS format (tfrecord)

术语速查

术语	中文	一句话解释
X-Embodiment	跨具身	多种机器人混合
Positive Transfer	正迁移	A的数据帮助B变强
Action Tokenization	动作令牌化	把连续动作变成离散token
Coarse Alignment	粗对齐	只对齐维度，不对齐坐标系
Emergent Skills	涌现技能	训练时没见过但能做的技能

笔记生成时间: Day 6 学习