0. 先用一句话说清楚这张海报在讲什么

我们在做的是一种"反向推理":

> 我们先看到一群 AI 在训练过程中一步步变聪明（它们的策略在不断更新），然后我们反推：**它们到底是在追求什么样的"得分规则"**（也就是奖励函数）。

这就像你只看别人练篮球的训练录像，不给你看教练的评分表，你要从他的动作变化里猜出：教练到底奖励什么，是投进球，还是传球配合，还是防守站位。

---

1. Motivation & Problem：动机与问题是什么

1.1 强化学习 Reinforcement Learning 是什么

强化学习可以理解成：让 AI 在环境里不断试错。

- AI 每一步会做一个动作（例如往左走、往右走、用多大力气迈腿）。
- 环境会给它一个分数（reward，奖励）。
- AI 的目标是：通过很多次尝试，让自己长期拿到更高的总分。

这里有两个核心概念:

- **奖励 Reward**：环境给的分数规则，决定什么算"好"。
- **策略 Policy**：AI 的行为规则，决定它在某种情况（状态）下会怎么做。

1.2 逆向强化学习 Inverse Reinforcement Learning 是什么

逆向强化学习（IRL）就是反过来:

- 你看到 AI 的行为（策略），但你不知道奖励规则是什么;
- 你要反推：这个 AI 好像在追求什么奖励。

类比：你看一个学生一直把时间花在某几类题上，你猜老师考试到底更偏向哪类题。

1.3 我们这张海报的关键点：不是看"最终会不会"，而是看"怎么学的"

很多 IRL 方法只看"最终学出来的策略"，但是我们强调一个更现实的事实:

> 真实世界里，智能体常常不是一上来就是最强的，它们是在学习过程中慢慢变强的。
> 学习过程本身就包含了很多关于奖励的信息。

所以我们问的是:
能不能直接从"策略更新过程"里反推奖励，而不是只从最终策略反推?

---

2. Challenges：为什么这件事很难

PPT 里列了三类困难:

2.1 Model mismatch：模型不匹配

很多旧方法假设学习者用的是一种"传统学习规则"，叫 **MA-SPI（Multi-Agent Soft Policy Iteration，多智能体软策略迭代）**。你可以把它想成"背表格式学习"，更适合小型、离散（动作很少）的环境。

但现在主流的强方法（比如 PPO）是"用梯度做优化"，更像是：

- 先算出"往哪个方向改策略会更好"（梯度）
- 然后沿着这个方向走一小步（更新参数）

如果你用"背表格"的旧假设去解释"梯度学习"的真实过程，就像你用"背乘法表"的模型去解释一个人如何用微积分解题，当然会对不上。

2.2 Coupled learning：多人同时学，会互相影响

多智能体里，每个人都在学：

- 你变了，我也会变；
- 我预测你会怎么变，我可能提前调整。

所以学习过程是"耦合"的，不是各学各的。你不能把每个智能体当成互不相关的独立个体来反推奖励。

2.3 Reward ambiguity：奖励有"多解"

这是一个很重要但很容易误解的点：
有时**不同的奖励规则，会导致几乎一样的行为。**

比如在迷宫里，"到终点给 1 分"是一种奖励；"离终点越近分越高"是另一种奖励。它们在数值上不同，但都可能让你走向终点。这样你看到行为时，很难唯一确定真实奖励。

这就是为什么我们在 T1 图里会看到：奖励热力图看起来不一样，但 KL 误差仍然很小。

---

3. Method Overview：我们的方法在做什么（I-LOLA）

这一部分是海报的"主角"。

3.1 LOGEL 是什么

LOGEL 是一个思想：
如果学习者是"用梯度更新策略"的，那么策略参数从 (\theta_t) 到 (\theta_{t+1}) 的变化里，其实包含了奖励的信息。

直觉是：
奖励决定"什么方向更好"，梯度就是"更好的方向"，所以更新步里带着奖励的影子。

3.2 I-LOGEL（Stage A）与 I-LOLA（Stage B）

我们把方法分两段：

- **Stage A：I-LOGEL（独立学习者）**
  先假装每个智能体都"各学各的"，忽略耦合影响，做一个较稳的起点。
- **Stage B：I-LOLA（耦合学习动态）**
  然后加入 LOLA 的思想：一个智能体在更新时，会考虑"对手也会更新"，也就是把耦合影响放进模型里。

3.3 LOLA 是什么（用一句话讲清）

LOLA 的核心直觉是：

> 我不只是问"我现在怎么做更好"，我还问"我现在这么改，会让对方下次怎么改，从而影响我以后好不好"。

这有点像博弈：你下一步的选择会改变对方的下一步，而你想把这件事提前算进去。

3.4 为什么用 CMA-ES（黑盒优化）

我们用的是 **CMA-ES**，它是一种"无导数优化"方法。高中生可以把它理解成一种更聪明的"试很多次 + 留下好的 + 调整试法"的搜索算法。

为什么要这样做?

- 因为如果你要对 LOLA 那种耦合学习过程做精确求导，代价很高，也容易不稳定；
- 用 CMA-ES 可以把系统当成黑盒：我给一组奖励参数进去，看看预测误差是多少，然后 CMA-ES 帮我越来越接近更好的参数。

---

4. Experimental Setup：我们怎么验证方法（T1/T2/T3）

我们选了三个环境，从简单到真实：

- **T1 GridWorld**：小型离散环境，旧方法 MA-LfL 的假设成立，是"理想考场"。
- **T2 MPE simple_spread**：多智能体合作任务，用 PPO 学习，是"现实一点的考场"。
- **T3 MultiWalker**：连续动作的物理控制任务，是"最难考场"。这里旧方法基本不适用，只能看 I-LOLA 的表现。

---

5. Metrics：这些数字到底在衡量什么

这部分非常关键，因为我们所有结论都靠它们支撑。

5.1 KL 是什么（高中生版）

KL divergence（KL 散度）可以先当作一个"差异分数"：

- 两个策略分布越像，KL 越接近 0；
- 越不像，KL 越大。

所以 KL 小表示：你预测的策略更新和真实策略更新很接近。

5.2 为什么要分 KL(1a) 和 KL(1b)

我们非常聪明的一点是：把误差拆开。

- **KL(1a)**：用"真实奖励"去预测下一步策略更新，得到的误差。
  它代表：就算我知道真正的奖励，这个预测模型本身也会有误差。这是"模型天花板"。
- **KL(1b)**：用"我恢复出来的奖励"去预测下一步策略更新，得到的误差。
  这代表：我恢复奖励是否靠谱。

判断标准很直白：

- 如果 KL(1b) (≈) KL(1a)，说明恢复奖励已经很好了；
- 如果 KL(1b) 远大于 KL(1a)，说明奖励恢复失败或模型不对。

5.3 Induced return 是什么（诱导回报）

这是我们的第二个验证方式：更像"实战测验"。

做法是：

1. 用恢复出来的奖励 ($\hat{W}$) 当作新环境的评分规则；
2. 让一个新的学习者从零开始训练；
3. 看它最后能拿多少分（return）。

如果恢复奖励真的有意义，那么用它训练出来的策略应该比随机乱走更好，最好能接近专家（用原始任务奖励训练出来的 PPO）。

---

6. Results：解释每一块结果是什么意思

下面按我们海报的三段结果解释。

6.1 T1 结果（Figure 1 与 Figure 2）

我们在 T1 得到：

- $KL(1a)(\approx)KL(1b)(\approx 9\times 10^{-6})$

这句话的意思是：

> 在一个完全符合 MA-LfL 假设的小环境里，MA-LfL 可以非常准确地预测策略更新。
> 用它恢复的奖励虽然数值形状不一样，但足以产生同样的行为。

Figure 2 的热力图"看起来不一样"，但 KL 仍然很小，这正好是在展示"奖励多解"的现象：数值不同，但行为等价。

所以 T1 的结论不是"恢复奖励长得像不像"，而是：

> **只要假设成立，MA-LfL 确实有效。**

这也让后面“假设不成立会失败”的故事更可信。

6.2 T2 结果（Figure 3）

我们写的是：

- 训练者使用 PPO（违反 MA-SPI 假设）
- I-LOLA 达到：
  - $KL(1a)(\approx 1.03 \times 10^{-6})$
  - $KL(1b)(\approx 1.09 \times 10^{-6})$

解释成白话就是：

> 在 PPO 学习者的数据上，我们用 I-LOLA 恢复的奖励，可以几乎和真实奖励一样准确地预测“下一次策略会怎么变”。

这非常关键，因为它说明：

- 我们的方法不是依赖旧假设（MA-SPI）；
- 它能适配现代梯度学习者（PPO）。

6.3 T3 结果（Figure 4 与 Figure 5）

T3 是连续动作的物理环境。我们给出的点是：

- MA-LfL 不适用（因为假设和计算都不合适）
- I-LOLA 仍稳定
- Monte Carlo $KL(\approx 5 \times 10^{-4})$

这里“Monte Carlo”是什么意思？

连续动作下，你不能像离散动作那样把所有动作都列出来求和。你只能：

- 从策略分布里随机采很多个动作样本；
- 用这些样本估计两个策略分布的差异。

所以这个 KL 是“采样估计”的结果，数值会比 T1/T2 大一些是正常的，因为估计会有噪声，也因为任务更复杂。

Figure 5 的 induced return 我们解释为：

- 诱导策略的表现和随机策略差不多
- 说明恢复奖励里有可用信号，但还需要进一步优化

把它讲得更直白一点：

> 我们已经能在最难的环境里“恢复出一个有点用的评分规则”，但这个规则还不够强，所以用它训练出来的策略还没有明显超过随机水平很多。

这不是坏消息，因为对 T3 来说，“能跑通且数值稳定”本身就已经是一个很重要的工程里程碑。

---

7. Takeaways & Limitations：我们想让观众带走什么结论

我们的 takeaway 可以翻译成三句更接地气的话：

1. **旧方法不是完全没用**：在它自己的理想世界（T1）里，它表现很好。
2. **新方法更贴近真实训练**：在 PPO 学习者（T2）里，它仍然准确。
3. **连续控制也不是做不到**：在 MultiWalker（T3）里，它能稳定运行并输出合理误差与诱导结果。

我们列出的 limitation 也很标准，意思是：

- 目前只跑了一个随机种子（可能有偶然性）；
- T2/T3 没有 MA-LfL 基线对比（故事还不够完整）；
- induced 的表现还没有到专家水平（还需要调参和增强特征/训练预算）。

---

8. “下一步要做什么”总结

我们后续工作有三件事：

1. **多做几次重复实验**（多 seed）
   证明这些结果不是“碰巧”。
2. **把老方法也拉来做对比**（至少在 T2）
   这样海报的核心故事就完整了：
   “旧方法在 T1 强，但在 PPO 世界会差；新方法在 PPO 世界强。”
3. **把 T3 的 induced 做得更像“学会走路”**
   让诱导策略明显超过随机，朝专家靠近。
   这通常要靠更好的特征、更长训练、调 PPO 超参数等工程工作。

---

9. Figure 1–5 的顺序写的“讲解稿”。

Figure 1：T1 GridWorld KL（seed=0）

这张图想回答一个很基础的问题：在最简单、最“理想”的环境里，我们能不能把方法跑通，并且让评估指标说得通。T1 是一个小型格子世界（GridWorld），动作是离散的，比如“上、下、左、右”。在这种环境里，有一种老方法 MA-LfL 的前提基本成立，所以它相当于一个“标准考场”。

图里横轴是两根柱子：err_1a 和 err_1b。你可以把它们理解成两次“预测误差”的考试成绩，分数越接近 0 越好。err_1a 是“我把真实的奖励规则给你，你用你的模型去预测下一步策略会怎么变”，看你能预测多准。它代表模型自身的上限，也就是“就算你知道标准答案，你能做到多好”。err_1b 是“我不给你真实奖励，我让你先把奖励规则反推出来，再用你反推的奖励去预测下一步策略怎么变”，看误差有多大。

现在这两根柱子几乎一样，而且都非常接近 0（大约是 (10^{-6}) 量级）。这说明两件事。第一，MA-LfL 在它的前提成立的场景里，确实能非常准地预测策略更新。第二，用它反推出来的奖励，至少在"让策略怎么更新"这件事上，已经和真实奖励几乎一样好。这里你也看到图标题里的 seed=0，它表示这只是一次随机种子下的结果，也就是"一次考试"。后面我们会用更多 seed 来证明不是碰巧。

Figure 2：T1 reward heatmap（真实奖励 vs 恢复奖励）

这张图是用来解释一个很容易误会的点：就算 Figure 1 的误差很小，反推出来的奖励也不一定"长得像"真实奖励。图里有上下两块热力图。上面那块是 "T1 reward true"，就是环境真正用来打分的规则；下面那块是 "T1 reward recovered"，就是算法从学习过程里反推出来的打分规则。颜色越亮一般代表奖励越高，颜色越暗代表奖励越低。

你会发现：上面的奖励图非常"干净"，像是只有某一列特别亮；但下面恢复出来的奖励图看起来更"花"，在很多位置都有不同的数值。那是不是说明我们恢复错了？不一定。因为在强化学习里有一个现象叫"奖励不唯一"：不同的奖励写法，可能会让智能体学出几乎一样的行为。就像老师可能用"做对题加分"，也可能用"离正确答案越近分越高"，两个评分细则不一样，但学生最后都学会做对题。

所以 Figure 2 的重点不是"两个热力图一模一样"，而是：即使奖励数值看起来不一样，它仍然可能诱导出几乎相同的策略更新，这就回到 Figure 1：我们用 KL 这种"行为层面的差异"来判断是否成功，而不是用"奖励长得像不像"来下结论。

Figure 3：T2 PPO KL comparison（在 PPO 学习者上测试）

现在进入更"真实"的场景。T2 用的是多智能体环境 MPE simple_spread，可以理解成几个智能体要在平面上合作去占住一些目标点。关键是：这里学习者不是用 MA-SPI 那套老式学习规则，而是用 PPO。PPO 可以理解成一种现代的、很常用的训练方法，它通过"算一个改进方向（梯度）然后更新策略参数"来学习。

这张图仍然是 KL 误差的对比：ILOLA-err_1a 和 ILOLA-err_1b。解释方法和 Figure 1 一样：err_1a 是"给真实奖励，让模型预测下一步策略更新"；err_1b 是"用反推的奖励，再去预测下一步策略更新"。如果两者很接近，就说明我们反推出来的奖励，已经能让模型做出和真实奖励几乎同样准确的预测。

你会看到两根柱子都在 (10^{-6}) 的量级，而且彼此非常接近。它的意义是：虽然 PPO 不满足 MA-LfL 的假设，但我们的 I-LOLA 这套"从梯度学习者反推奖励"的思路，在 PPO 这种现代学习者上依然能工作。换句话说，我们不是只在"旧规则的考场"里能得分，我们在"真实训练流程"里也能把链路跑通。

Figure 4：T3 MultiWalker KL（Monte Carlo）

T3 是最难的环境 MultiWalker。你可以把它想成几个小机器人要一起走路、保持平衡、合作前进。这里动作是连续的，比如"关节用多大力"，不是简单的"左或右"。在这种连续动作环境里，很多老方法要么前提不成立，要么计算量太大，很难直接用，所以海报上也写了 "MA-LfL not applicable"。

那我们怎么评估 I-LOLA 在这里有没有"瞎跑"？还是用 KL，但连续动作有个新麻烦：你不能把所有动作枚举出来做精确求和，所以我们用 Monte Carlo。Monte Carlo 的意思很简单：从策略里随机抽很多个动作样本，用这些样本去估计"两个策略分布差多少"。它像是用"抽样调查"代替"全员普查"。

这张图的柱子给出 T3 的 KL 估计值（海报文字里写大约 (5\times10^{-4}) 量级）。这个数比 T1/T2 大是正常的，因为环境更难、估计方法也有抽样噪声。你可以把它理解成：在一个更复杂、更接近真实物理的场景里，我们仍然能得到一个数值稳定、可以解释的误差，而不是出现完全失控的结果。它至少说明：方法在连续控制上是"能跑、能算、能比较"的。

Figure 5：T3 induced returns（用恢复奖励重新训练，看是否“有用”）

最后这张图讲的是“第二种更直观的检验”：就算 KL 看起来不错，我们还是想问一个更现实的问题：我恢复出来的奖励，到底有没有用？所以我们做 induced training，也就是“诱导训练”。做法是：把恢复出来的奖励当成新的打分规则，然后让一个新的智能体从零开始训练，看看它最终能拿到多少回报（return）。return 就是“每一局游戏累计拿到的总分”，越高说明学得越好。

图里三条线/标记的意思是：R_random 是随机策略的平均回报，你可以把它当成“不会走路只会乱动”；R_expert 是专家策略的平均回报，通常来自用真实奖励训练出来的 PPO；R_induced 是我们用恢复奖励训练出来的策略表现，横轴 training step 表示训练推进的阶段，纵轴是每个阶段评估出来的平均回报。

这张图里，R_induced 大体上接近随机基线，有时略好，有时略差。海报给出的解释是“恢复奖励里有可用信号，但还需要进一步优化”。用高中生能理解的话说就是：我们已经能从学习过程里“猜出一份评分规则”，这份规则不是完全乱写的，但它还不够强，还不足以让新智能体稳定学到接近专家的行为。接下来要做的事情，就是提高这份奖励的质量，比如改进特征、增加数据量、提高优化预算，或者让训练更充分。

不过图中，RL_Random和RL_Expert无法区分。原因是“expert” 并不是真正的 PPO 专家，而是“占位的 expert”。这个需要在进一步实验中生成Expert.

---

10. 老师可能会问的问题：

我会问的第 1 个问题：
“我们说是在‘反推奖励’，那我们怎么证明反推出来的奖励真的有用，而不是随便找了个能凑出小误差的东西？”

回答：
我们做了两种检验，不只看一种。第一种是“预测下一步策略怎么变”，用 KL 误差衡量：如果我们用恢复奖励预测的策略变化（1b）和用真实奖励预测的策略变化（1a）一样准，说明恢复奖励确实捕捉到了学习过程里的关键信息。第二种是“诱导训练”：把恢复奖励当成新的评分规则，从零训练一个新智能体，如果它能比随机策略更好，就说明恢复奖励里有可用信号。我们现在已经把这两条链路都跑通了，说明不是只会“拟合一个数字”，而是真的能影响学习。

"We use two checks. First is **prediction accuracy**: we compare the error from our recovered reward (1b) against the true reward (1a). If they match, we know we captured the learning signal. Second is **induced training**: we use our recovered reward to train a new agent from scratch. Since it performs better than a random policy, we know the reward contains real, usable information, not just noise."

我会问的第 2 个问题：
“我们为什么要把误差分成 1a 和 1b？直接报一个 KL 不就行了？”

回答：
因为只报一个 KL 容易误解。1a 是“模型自身的极限”，就像你给了标准答案，让模型去预测下一步策略变化，它也不可能 0 误差。1b 才是“奖励恢复导致的额外误差”。如果 1b 很大，你不知道是模型本身不行，还是奖励没恢复对。有了 1a/1b，我们能清楚区分：问题出在“预测模型不够好”，还是出在“奖励恢复不够好”。

"1a is our baseline—it's the 'model limit' using the *true* reward. Even that isn't zero. 1b is the error using *our* recovered reward. By comparing them, we can separate the **model's inherent error** from the **reward recovery error**. If we only had one number, we wouldn't know which part was failing."

我会问的第 3 个问题：
“T3 的 KL 是 Monte Carlo 估计的。我们怎么保证这个估计是可靠的？会不会只是抽样抽巧了？”

回答：
Monte Carlo 就像抽样调查，所以我们要控制抽样误差。我们会固定同一批状态样本和随机种子，在 1a 和 1b 的比较里用完全相同的抽样方式，这样差异才是因为奖励不同，而不是因为抽样波动。我们还可以加大采样次数，比如每个状态采更多动作样本，看 KL 是否稳定收敛。如果采样数翻倍，KL 变化不大，就说明估计是稳的。

"Since Monte Carlo is just sampling, we have to control the variance. We use the **exact same fixed samples and seeds** for both 1a and 1b comparisons. This ensures any difference comes from the *reward quality*, not sampling luck. We also verify stability by increasing the sample size to ensure the value converges."

我会问的第 4 个问题：
“我们说 MA-LfL 在 T3 不适用，那是不是等于我们没有基线对比？那我怎么知道我们的新方法更好？”

回答：
我们这里的“对比”分两层。第一层是理论层：MA-LfL 的学习者假设是基于值函数的 soft policy iteration，但 T3 的学习者是 PPO（梯度更新），假设不匹配，而且连续动作下计算也非常重，所以它不适合在 T3 做同等实现。第二层是实验层：我们在 T1 做了 MA-LfL 的“理想考场”，证明它在自己擅长的场景里能做到极小误差；在 T2 这种 PPO 数据的环境里，我们可以让 MA-LfL 和 I-LOLA 同台对比，这一部分是我们后续要补齐的核心对比实验。T3 的意义更像是“展示新方法能在高难度环境里跑通并稳定输出”，而不是用它来做老方法的正面对比。

"Two reasons. **Theoretically**, the old method (MA-LfL) assumes discrete updates, so it physically can't run on the continuous PPO learner in T3. **Experimentally**, T3 isn't about beating the old method—it's about proving our new method *works at all* in complex, continuous environments where the old method fails."

我会问的第 5 个问题：
“折线图里 random 和 expert 怎么看起来差不多？是不是我们的 expert 根本不是专家？”

回答：
这是一个很好的抓虫问题。理论上 expert 应该明显高于 random。如果它们几乎一样，通常说明我们目前的 expert 还没有用正确方式加载或评估，比如 expert checkpoint 没有真的读进来，或者 expert 还是占位值。我们会用 JSON 里保存的 `R_expert` 来确认：如果是 NaN 或接近 random，就说明目前这条基线还没做完。修好之后，图里应该出现清晰的三条水平线：random 最低，expert 最高，induced 在中间或接近 expert。

"That’s a sharp observation. Theoretically, the Expert should definitely be higher. This is likely a **technical issue** with how the expert checkpoint was loaded or plotted in this draft. We are fixing this, and the final result will clearly show the Expert line well above the Random baseline."

我会问的第 6 个问题：
“我们的奖励有‘多解’，那我们怎么评价恢复奖励的好坏？只看 KL 会不会掩盖问题？”

回答：
奖励多解意味着“奖励长得像不像”不是最可靠的评价，因为不同奖励也可能诱导同样策略。我们用 KL 是在比较“行为层面是否一致”。同时，我们加了“诱导训练”作为第二道更直观的检验：如果恢复奖励能训练出更好的策略，它就是有意义的。未来我们也会补充更多指标，比如在同一批状态下比较动作偏好、或者比较“用恢复奖励训练出来的策略是否能完成任务”，让评价更立体。

"Because of reward ambiguity, different rewards can lead to the same behavior, so just 'looking' at the reward numbers isn't enough. KL measures the **behavioral difference**, which is what actually matters. Plus, our **Induced Training** acts as a practical sanity check to ensure the reward works."

我会问的第 7 个问题：
“我们只跑了 seed=0。这个结果稳定吗？会不会换个随机种子就变了？”

回答：
你说得对，单个 seed 只能说明“这一次”。我们把工程框架搭好之后，下一步就是批量跑多个 seed（比如 5 个或 10 个），然后报告均值和方差。这样就能回答“稳定性”问题。如果多个 seed 下结论一致，我们的故事才算站得住。

"You’re absolutely right—a single seed is just a proof of concept. Now that the pipeline is working, our immediate next step is to run **multiple seeds** (like 5 or 10) and report the mean and variance to prove the results are statistically stable."