

深度强化学习 Sim-to-Real 技术演进综述

231880394 翟笑晨

南京大学/匡亚明学院

January 15, 2026

Abstract

深度强化学习（DRL）在模拟环境中展现了惊人的决策能力，但将其迁移至真实物理世界（Sim-to-Real）时，往往因动力学参数偏差与感知噪声而面临巨大的“现实鸿沟（Reality Gap）”。本文旨在系统梳理 Sim-to-Real 技术的发展脉络，构建了从早期的均匀域随机化（Uniform DR）到数据驱动的自动域随机化（ADR/BayRn），再到基于隐式适应（RMA）与生成式 AI（GenAI）辅助的技术演进图谱。文章深入剖析了现有方法在“参数搜索效率”与“物理语义缺失”方面的局限性，并重点探讨了 DrEureka、GenSim 等最新范式如何利用大语言模型（LLM）的物理常识与代码生成能力，实现了从“数值参数优化”到“语义环境生成”的范式转移。通过综合分析核心文献，本文揭示了 Sim-to-Real 技术正向着零样本（Zero-shot）、多模态（Multimodal）和全自动化（Fully Automated）方向演进的趋势 [1, 2]。

关键词：深度强化学习；Sim-to-Real；域随机化；元学习；大语言模型；具身智能

Contents

1 引言 (Introduction)	3
1.1 基本定义	3
1.2 为什么“轻微动力学误差”会造成大性能坍塌	3
2 核心技术演进 (Core Technology Evolution)	5
3 均匀域随机化 (Uniform Domain Randomization)	5
4 自动域随机化与数据驱动 (Automatic & Active DR)	6
4.1 ADR: 课程学习式的“边界推进”	6
4.2 Active DR: 用“最有信息的参数”主动提问	6
4.3 BayRn: 利用少量真实数据反向拟合仿真分布	6
4.4 DORAEMON: 熵最大化的“找麻烦”随机化	7
5 隐式适应 (Implicit Adaptation)	7
5.1 把动力学视作隐藏变量: POMDP 视角	7
5.2 RMA: 隐变量编码 + Teacher-Student 蒸馏	7
5.3 Meta-RL: 把“适应过程”本身学出来	8
5.4 任务驱动适应与多行为泛化	8
6 语义生成与全流程自动化 (Generative Sim-to-Real)	8
6.1 Eureka: 把 Reward Design 形式化为“程序搜索”	8
6.2 DrEureka: 物理先验引导的参数生成	8
6.3 GenSim & RoboGen: 从参数生成到环境/任务生成	9
7 思考与展望 (Discussion & Outlook)	10
7.1 从数值优化到语义推理 (From Numerical to Semantic)	10
7.2 从被动鲁棒到主动生成 (From Passive to Generative)	11
7.3 零样本迁移的边界 (The Limits of Zero-Shot)	12
7.4 多模态鸿沟 (The Multimodal Gap)	13

1 引言 (Introduction)

随着深度强化学习的发展，机器人在仿真环境中掌握复杂技能已成为常态。然而，直接将仿真训练的策略部署到真实机器人上通常会失败，这种现象被称为“现实鸿沟 (Reality Gap)”。从实践演化看，早期成功案例既包括基于视觉随机化的端到端飞行控制 (CAD2RL) [6]，也包括面向动力学鲁棒控制的随机化训练 [3–5]，以及后续在灵巧手与抓取等任务上的规模化验证 [7, 21, 22]。

1.1 基本定义

强化学习通常被建模为马尔可夫决策过程 (MDP)：

$$\mathcal{M} = (S, A, T, R, \gamma), \quad (1)$$

其中状态 $s \in S$ ，动作 $a \in A$ ，转移概率 $T(s'|s, a)$ ，奖励 $R(s, a)$ ，折扣因子 $\gamma \in (0, 1)$ 。策略 $\pi(a|s)$ 的期望折扣回报为

$$J(\pi; \mathcal{M}) = \mathbb{E}_{\tau \sim p(\tau|\pi, \mathcal{M})} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right]. \quad (2)$$

Sim-to-Real 的核心挑战在于：仿真域与真实域存在分布偏移 (Distribution Shift)，尤其体现在转移函数不一致：

$$T_{sim}(s'|s, a; \xi) \neq T_{real}(s'|s, a), \quad (3)$$

其中 $\xi \in \Xi \subset \mathbb{R}^d$ 表示仿真器动力学参数 (摩擦、质量、阻尼、关节延迟、传感器噪声等)。

1.2 为什么“轻微动力学误差”会造成大性能坍塌

若每一步转移的差异 (例如用总变差距离度量) 上界为

$$\epsilon_T \triangleq \sup_{s, a} \text{TV}(T_{real}(\cdot|s, a), T_{sim}(\cdot|s, a)), \quad (4)$$

且单步奖励有界 $|r_t| \leq R_{\max}$ ，则同一策略在真实/仿真回报差异会随时间增长而放大：

$$|J_{real}(\pi) - J_{sim}(\pi)| \lesssim \frac{2\gamma R_{\max}}{(1-\gamma)^2} \epsilon_T. \quad (5)$$

这解释了为何长时序控制中“看似很小的动力学偏差”会滚动累积并导致策略失效。

为了解决这一问题，Sim-to-Real 技术经历了三个阶段的演进：

- 随机化阶段：通过覆盖大量参数 ξ 来“包围”真实世界（Domain Randomization, DR）[3, 4]；
- 适应阶段：通过系统辨识、在线辨识或元学习，让策略在线适应真实参数（Adaptation/Meta-RL）[11, 13, 15]；
- 生成阶段：利用大语言模型（LLM）的通识能力，自动生成合理参数分布甚至仿真任务与环境（Generative Sim-to-Real）[16–19]。

本文将围绕这三条主线，综述近年的关键技术进展，特别是分析 Generative AI 如何解决传统方法中“参数搜索盲目”和“奖励设计困难”的痛点 [1]。



Figure 1: 本文结构纲要：展示了从均匀域随机化，到自动化与数据驱动，再到隐式适应与生成式 AI 辅助的 Sim-to-Real 技术演进路径 (AI 生成，存在部分文字错误)

2 核心技术演进 (Core Technology Evolution)

为了清晰展示 Sim-to-Real 技术的迭代逻辑, 本文构建了技术演进逻辑图 (如图 2 所示) 并将围绕其展开论述。

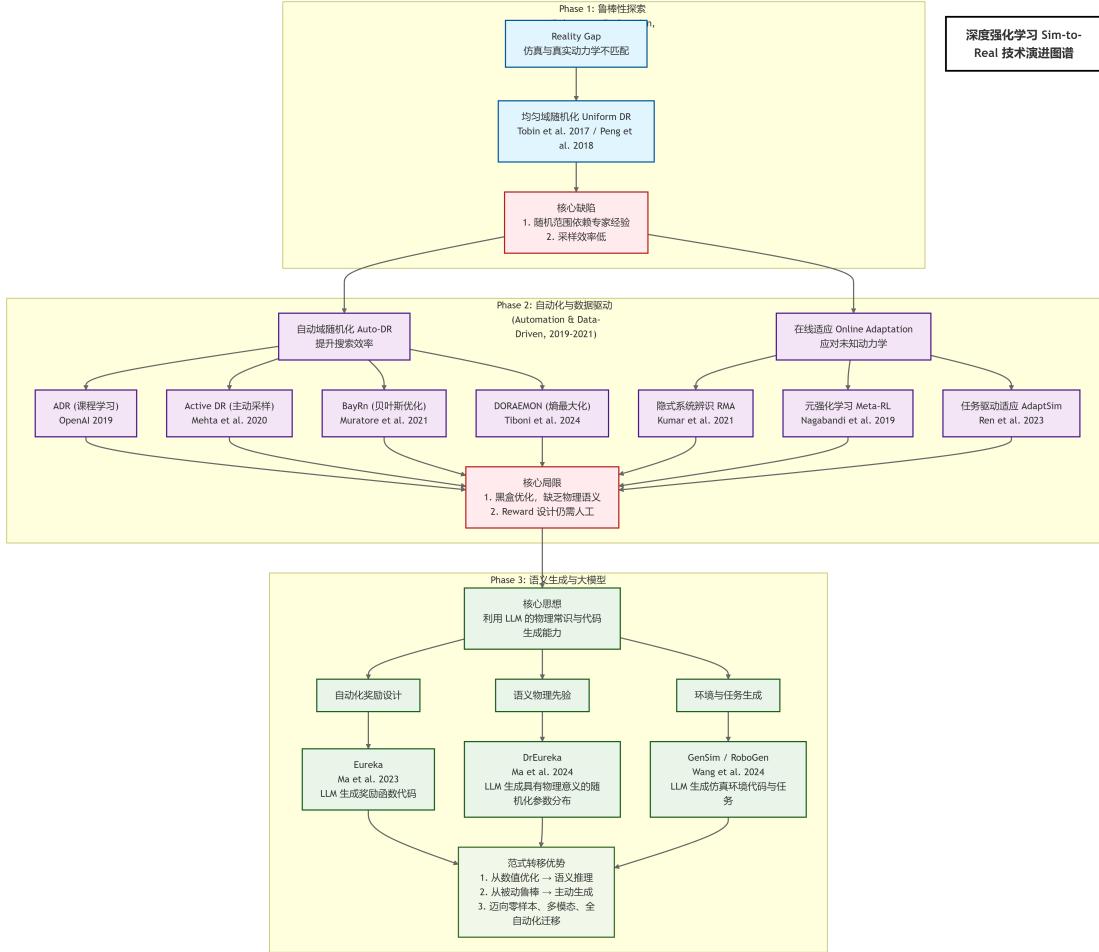


Figure 2: Sim-to-Real 技术演进逻辑图。展示了从基于规则的随机化, 到基于数据的自动化优化, 最终汇聚于基于语义的生成式方法的演进过程。

3 均匀域随机化 (Uniform Domain Randomization)

Tobin 等人 [3] 提出的域随机化 (DR) 通过最大化在参数分布 $P(\xi)$ 下的期望回报来训练策略。该思想不仅用于视觉域 (通过渲染与纹理随机化提升感知鲁棒性), 也可直接用于控制域中的动力学随机化 [4,5], 并在灵巧操作与抓取任务中得到规模化验证 [7,21,22]。

期望鲁棒优化

$$\pi^* = \operatorname{argmax}_{\pi} \mathbb{E}_{\xi \sim P(\xi)} [J(\pi; \mathcal{M}(\xi))], \quad (6)$$

其中 $\mathcal{M}(\xi)$ 表示由参数 ξ 决定的仿真 MDP。动力学随机化常令

$$\xi = [m, \mu, b, k, \Delta t, \sigma_{sensor}, \dots]. \quad (7)$$

保守性来源 更保守的形式是最坏情形鲁棒：

$$\pi^* = \operatorname{argmax}_{\pi} \min_{\xi \in \Xi} J(\pi; \mathcal{M}(\xi)). \quad (8)$$

实践中用“更宽的随机化范围”近似(8)，但会导致过度保守与样本效率低的问题[1,2]。

4 自动域随机化与数据驱动 (Automatic & Active DR)

4.1 ADR: 课程学习式的“边界推进”

OpenAI 在 Rubik's Cube 机器人手任务中展示了通过逐步扩张随机化边界提升真实迁移的路线[7]。可将其抽象为参数分布族 $P_\phi(\xi)$ ：

$$\pi^*(\phi) = \operatorname{argmax}_{\pi} \mathbb{E}_{\xi \sim P_\phi(\xi)} [J(\pi; \mathcal{M}(\xi))]. \quad (9)$$

并依据边界成功率更新分布参数（课程学习/自适应扩张）：

$$\phi \leftarrow \phi + \eta \cdot \nabla_\phi \mathbb{E}_{\xi \sim P_\phi} [\mathbb{I}(\text{success}(\pi, \xi))]. \quad (10)$$

4.2 Active DR: 用“最有信息的参数”主动提问

Mehta 等提出 Active Domain Randomization [8] 强调：与其无差别采样，不如选择能最大化学习信号/暴露失败模式的参数区域。直观上，它可被理解为在 ξ 空间中进行一种“主动实验设计”，提升随机化的样本效率（与 BayRn 的“用真实数据做反向拟合”互补）。

4.3 BayRn: 利用少量真实数据反向拟合仿真分布

Muratore 等[9]的 BayRn 可抽象为：用少量真实轨迹 τ^{real} 反向优化分布参数 ϕ ，使仿真轨迹统计更一致：

$$f(\phi) = D(\tau^{real}, \tau^{sim}(\phi)), \quad \phi^* = \operatorname{argmin}_\phi f(\phi). \quad (11)$$

其关键在于 $f(\phi)$ 常不可导且真实采样昂贵，因此采用高斯过程代理模型：

$$f(\phi) \sim \mathcal{GP}(m(\phi), k(\phi, \phi')), \quad (12)$$

并通过 acquisition function 选取下一个评估点:

$$\phi_{t+1} = \underset{\phi}{\operatorname{argmax}} \alpha(\phi; \mathcal{D}_t). \quad (13)$$

4.4 DORAEMON: 熵最大化的“找麻烦”随机化

Tiboni 等 [10] 的核心思想是: 不依赖真实数据, 直接寻找“挑战性最大且多样”的参数分布。典型形式为最大化分布熵并保持任务可学:

$$\max_{P(\xi)} H(P) \quad \text{s.t.} \quad \mathbb{E}_{\xi \sim P}[J(\pi; \mathcal{M}(\xi))] \geq \beta, \quad (14)$$

或拉格朗日松弛:

$$\max_{P(\xi)} H(P) + \lambda \cdot \mathbb{E}_{\xi \sim P}[J(\pi; \mathcal{M}(\xi))]. \quad (15)$$

尽管 BayRn、Active DR 与熵最大化提高了搜索效率 [8–10], 但它们本质上仍偏“数值层面的黑盒优化”: 优化器缺乏物理语义约束, 可能探索到不合理参数组合; 同时 BayRn 仍需真实采样成本。

5 隐式适应 (Implicit Adaptation)

5.1 把动力学视作隐藏变量: POMDP 视角

若把动力学参数 ξ 视为 episode 级别固定但未知的隐藏状态, 则策略应基于历史形成对 ξ 的信念:

$$b_t(\xi) = p(\xi \mid o_{0:t}, a_{0:t-1}). \quad (16)$$

这一视角与“在线系统辨识 + 通用策略”的经典思路一致: Yu 等 [13] 提出通过在线系统辨识模块估计隐藏动力学, 并驱动通用策略在未知环境中快速适应。

5.2 RMA: 隐变量编码 + Teacher-Student 蒸馏

Kumar 等 [11] 的 RMA 采用 teacher-student: 教师可访问特权信息 (如 ξ 或更完整状态), 学生仅基于本体感觉历史推断隐变量并控制。可写为

$$z_t = f_\theta(h_t), \quad h_t = (o_{t-k:t}, a_{t-k:t-1}), \quad (17)$$

$$a_t^{stu} \sim \pi_\psi(\cdot \mid s_t, z_t), \quad (18)$$

$$a_t^{tea} \sim \pi_{priv}(\cdot \mid s_t, \xi), \quad (19)$$

学生通过蒸馏损失拟合教师动作/分布:

$$\mathcal{L}_{distill} = \mathbb{E}[\|a_t^{stu} - a_t^{tea}\|^2] \quad \text{或} \quad \mathbb{E}[\text{KL}(\pi_{priv}(\cdot \mid s_t, \xi) \parallel \pi_\psi(\cdot \mid s_t, z_t))]. \quad (20)$$

5.3 Meta-RL: 把“适应过程”本身学出来

Nagabandi 等 [15] 代表了另一条适应思路：用元强化学习在训练阶段经历多环境变化，让策略在真实动态环境中快速更新/内化适应机制（可理解为学习一个“能快速变化的控制器”）。这与 RMA/UPOSI 同样有“宁可学适应，也不求精确辨识”的特性。

5.4 任务驱动适应与多行为泛化

Ren 等 [12] 的 AdaptSim 强调任务驱动的仿真适配：以任务性能为反馈，自动调整仿真以更好支持迁移。另一方面，Margolis & Agrawal [14] 强调通过“行为多样性 (multiplicity of behavior)” 提升泛化鲁棒性，可视为在策略空间层面做覆盖而非仅在参数空间覆盖。

6 语义生成与全流程自动化 (Generative Sim-to-Real)

6.1 Eureka: 把 Reward Design 形式化为“程序搜索”

Eureka [17] 把奖励函数视为可搜索的程序 $r_\omega(\cdot)$ （由 LLM 生成代码候选），形成双层优化：

$$\omega^* = \underset{\omega}{\operatorname{argmax}} \mathcal{S}(\pi^*(\omega)), \quad (21)$$

$$\pi^*(\omega) = \underset{\pi}{\operatorname{argmax}} \mathbb{E} \left[\sum_{t \geq 0} \gamma^t r_\omega(s_t, a_t) \right]. \quad (22)$$

同类“语言到奖励”的研究也表明：自然语言约束可被编译为 reward/成本项，从而降低手工 shaping 的负担 [20]。

6.2 DrEureka: 物理先验引导的参数生成

DrEureka [16] 强调让 LLM 输出具有物理意义的随机化先验（语义条件分布）：

$$\xi \sim P_{LLM}(\xi | c), \quad (23)$$

其中 c 是场景/任务描述（草地/冰面/沙地等），先验相当于用语言常识剪枝参数空间；同时继承 Eureka 式 reward 自动化，可把安全/平滑写成 reward shaping：

$$r(s, a) = r_{\text{task}}(s, a) - \lambda_1 \|a_t - a_{t-1}\|^2 - \lambda_2 \|\tau_t\|^2 - \lambda_3 \cdot \mathbb{I}(\text{fall}). \quad (24)$$

6.3 GenSim & RoboGen：从参数生成到环境/任务生成

GenSim [18] / RoboGen [19] 进一步将生成对象从参数 ξ 扩展到环境/任务代码 e (地形、物体、URDF、布局、事件等):

$$e \sim p_\theta(e | \text{prompt}), \quad \pi^* = \underset{\pi}{\operatorname{argmax}} \mathbb{E}_{e \sim p_\theta} [J(\pi; \mathcal{M}_e)]. \quad (25)$$



Figure 3: Sim-to-Real 技术的三阶段演进示意图。展示了从均匀域随机化，到自动化与数据驱动，再到语义生成与全流程自动化的技术路线 (AI 生成，存在部分文字错误)

7 思考与展望 (Discussion & Outlook)

7.1 从数值优化到语义推理 (From Numerical to Semantic)

早期的数据驱动路线（以 BayRn 为代表）本质上把 Sim-to-Real 视为“在真实样本预算有限的条件下，反向校准仿真分布参数”的问题，其优化形式可由前文 BayRn 的目标式 (11) 概括：通过少量真实轨迹统计去约束仿真轨迹统计，使仿真分布逐步靠近真实分布 [9]。更一般地，从“轨迹分布/占用测度匹配”的视角，可以将这一类方法抽象为：在固定策略 π 或迭代策略更新的过程中，匹配真实与仿真的轨迹分布或状态-动作占用测度 (occupancy measure)：

$$\min_{\xi} \text{KL}(p_{real}(\tau) \| p_{sim}(\tau; \xi)) \quad \text{或} \quad \min_{\xi} \mathcal{D}(d_{real}^{\pi}(s, a), d_{sim}^{\pi}(s, a; \xi)), \quad (26)$$

其中 $p(\tau)$ 为轨迹分布， $d^{\pi}(s, a)$ 为策略诱导的占用测度。该抽象强调了两点关键难题：
(i) 只有在策略访问到的子空间内，匹配才“可见”（不可观测区域无法被数据约束）；
(ii) 匹配目标往往非凸、不可导且带噪，导致优化高度依赖启发式与采样预算。

Active DR 与熵最大化路线提升了搜索效率，但更偏向“怎样更快地找到困难参数/困难区域”，并没有从根本上解决“搜索空间物理可行性与语义合理性”问题 [8,10]。相比之下，Phase 3 的 DrEureka/GenSim 通过语义先验（前文式 (23)）与环境生成（前文式 (25)）将物理常识显式注入搜索过程：把大量“物理上不合理”的区域在生成阶段就剪枝。用 Bayesian 视角可写成：

$$p(\xi | \mathcal{D}_{real}) \propto p(\mathcal{D}_{real} | \xi) p_{LLM}(\xi | c), \quad (27)$$

其中 $p_{LLM}(\xi | c)$ 把语言知识转化为结构化先验，使后续的数值优化更集中在“物理可行子空间”内，从而体现出从数值拟合到语义推理的演进趋势 [1]。

Key Takeaways

- **数值拟合视角 (Phase 2 / Auto-DR):** BayRn 可由 (11) 概括为“用少量真实轨迹统计反向校准仿真分布”，其有效性受限于采样预算与可观测子空间 [9]。
- **效率改进但仍黑盒 (Phase 2 / Active & Entropy):** Active DR 与熵最大化解决“搜得更快”，但难以保证“搜得更合理”，物理语义仍缺位 [8,10]。
- **语义剪枝的范式转移 (Phase 3 / GenAI):** DrEureka/GenSim 用语义先验 ((23)、(27)) 与环境生成 ((25)) 把“盲搜”转为“先语义定位、再数值微调” [1]。
- **趋势:** 更可能是“语义先验 + 数值优化”的混合系统：Phase 3 提供结构化先验/约束，Phase 2 负责局部精调与验证。

7.2 从被动鲁棒到主动生成 (From Passive to Generative)

UDR/ADR 更偏被动鲁棒：其目标是扩大训练扰动覆盖，使策略在未知变化下尽量保持性能。它们在形式上分别对应前文的“期望鲁棒目标”(6)与“近似最坏情形鲁棒”(8)，优势是通用、实现直接，但代价往往是训练成本高、策略趋于保守，且对扰动集合设计仍有较强工程依赖 [3, 7]。

RoboGen/GenSim 则体现了主动生成范式：不只训练策略，还把“训练环境/任务的分布”本身当作可优化对象 [18, 19]。其核心可以用双层视角表达为（该式在前文未出现，作为新的抽象保留）：

$$\max_{p(e)} \mathcal{G}(\pi^*(p(e))) \quad (28)$$

$$\text{s.t. } \pi^*(p(e)) = \underset{\pi}{\operatorname{argmax}} \mathbb{E}_{e \sim p(e)} [J(\pi; \mathcal{M}_e)], \quad (29)$$

其中 \mathcal{G} 衡量泛化或部署表现。AdaptSim 可视为 Phase 2 向 Phase 3 过渡的重要桥梁：它用任务反馈塑形仿真，使“仿真适配”成为学习过程的一部分 [12]；而行为多样性路线强调在策略空间构造覆盖，从另一维度提升泛化，与“生成更多样环境”的思路互补 [14]。

Key Takeaways

- **被动鲁棒 (Phase 1/2):** UDR/ADR 在形式上对应 (6) 与 (8)，通过扩大扰动集合来“硬抗变化”[3, 7]。
- **主动生成 (Phase 3):** GenSim/RoboGen 把重点从“调参训练策略”扩展为“设计训练分布/生成环境”(式 (29))，实现从“包围真实”到“生成训练价值”的升级 [18, 19]。
- **过渡形态 (Phase 2→3):** AdaptSim 用任务反馈驱动仿真适配，连接了 Phase 2 的自动化与 Phase 3 的生成式范式 [12]。
- **互补维度 (Phase 2B):** 行为多样性扩策略空间，环境生成扩环境空间；两者可叠加增强鲁棒泛化 [14]。

7.3 零样本迁移的边界 (The Limits of Zero-Shot)

DrEureka 展示了“在缺少真实数据时，依赖语义先验与自动化 reward 仍可能获得可部署策略”的潜力：其核心机制分别对应前文的语义先验 ((23)) 与 Eureka 的程序化奖励搜索 ((21)–(22)) [16]。然而，零样本迁移的根本风险是“先验失配”：当真实世界的动力学/视觉因素落在 LLM 经验分布之外时，生成的随机化范围与奖励偏好可能系统性偏离，从而导致训练分布与部署分布不一致。一个直接刻画这种失配的方式是用先验分布与真实分布之间的散度：

$$\Delta_{\text{prior}} \triangleq \text{KL}(p_{\text{real}}(\xi) \| p_{\text{LLM}}(\xi | c)), \quad (30)$$

当 Δ_{prior} 较大时，即便策略在训练分布上表现良好，部署性能也可能显著下降（尤其在长时序控制任务中，这类误差会被滚动放大）。

与此同时，Phase 2B 的隐式辨识与在线适应路线提供了对抗失配的“兜底机制”：通过历史推断隐变量并条件化控制，使策略在部署时能够吸收未建模误差，其结构在前文已由 RMA 的隐变量编码形式 (17) 体现 [11, 13, 15]。因此，更现实的终局往往不是“纯 Zero-shot”，而是“语义生成（给出良好初始化）+ 在线适应（吸收剩余误差）”的闭环系统：语义模块减少搜索盲目性，适应模块处理长尾与不可预见因素 [1, 16]。

Key Takeaways

- **零样本优势 (Phase 3):** DrEureka 依赖语义先验 (23) 与自动化 reward ((21)–(22)) 提升无真实数据部署可能性 [16]。
- **边界刻画 (新增不重复公式):** 零样本的核心风险是先验失配，可用 Δ_{prior} (式 (30)) 刻画训练分布与真实分布偏离程度。
- **在线适应兜底 (Phase 2B):** UPOSI/RMA/Meta-RL 通过历史推断 (结构见 (17)) 弥补长尾动力学误差 [11, 13, 15]。
- **终局形态:** Phase 3 (语义生成) 与 Phase 2B (在线适应) 更可能组合成闭环系统，而非相互取代 [1]。

7.4 多模态鸿沟 (The Multimodal Gap)

视觉上的 Sim-to-Real 仍然困难：即便动力学被充分随机化，真实世界的光照、材质、透明/反光与传感器成像链路也会带来显著偏移。CAD2RL 强调仅用仿真图像通过视觉随机化实现真实飞行 [6]，其可抽象为对外观参数 ϕ 的期望风险最小化（该式为该小节的关键支撑，且不与前文动力学公式重复）：

$$\min_{\theta} \mathbb{E}_{\phi \sim P(\phi)} [\mathcal{L}(f_{\theta}(I(\phi)), y)], \quad (31)$$

其中 $I(\phi)$ 为渲染器生成图像， f_{θ} 为感知网络/策略网络。Sim-to-Sim 则展示了用“从一个仿真到另一个更贴近真实的仿真”来降低真实数据需求 [22]，其直觉可由“差异可分解”表达为：

$$\mathcal{D}(p_{sim}, p_{real}) \leq \mathcal{D}(p_{sim}, p_{mid}) + \mathcal{D}(p_{mid}, p_{real}), \quad (32)$$

其中 p_{mid} 是更贴近真实统计的中间仿真域。Dextreme 表明在操作任务中，大规模随机化仍能支持从仿真到真实的灵巧操作迁移 [21]，但训练成本与可控性仍是挑战。未来，多模态 Sim-to-Real 很可能依赖“生成式场景 + 物理一致性约束”的 world simulator：在更真实的视觉-物理耦合模拟中，以更少的人为建模假设覆盖更广的真实分布，从而进一步缩小多模态现实鸿沟 [1, 2]。

Key Takeaways

- **视觉随机化 (Parallel branch, 与 Phase 1 同源)：**CAD2RL 用外观随机化实现无需真实图像迁移 (式 (31))，与 Phase 1 的“随机化包围真实”在结构上对偶 [6]。
- **桥接策略 (Parallel branch, 连接 Phase 2 的数据效率诉求)：**Sim-to-Sim 通过中间域分解差异 (式 (32))，减少真实采样压力 [22]。
- **规模化证据 (Parallel branch 与 Phase 1/2 经验一致)：**Dextreme 表明在高复杂任务中“大规模随机化 + 足够算力”仍有效，但训练成本高 [21]。
- **趋势 (与 Phase 3 呼应)：**未来多模态 Sim-to-Real 可能依赖“生成式场景 + 物理一致性约束”的 world simulator，推动从“渲染随机化”走向“语义可控生成” [1, 2]。

References

- [1] Da, L., et al. (2025). A Survey of Sim-to-Real Methods in RL: Progress, Prospects and Challenges with Foundation Models. *arXiv preprint*.
- [2] Zhao, W., et al. (2020). Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: A Survey. *IEEE SSCI*.
- [3] Tobin, J., et al. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. *IROS*.
- [4] Peng, X. B., et al. (2018). Sim-to-real transfer of robotic control with dynamics randomization. *ICRA*.
- [5] Tan, J., et al. (2018). Sim-to-real: Learning agile locomotion for quadruped robots. *RSS*.
- [6] Sadeghi, F., & Levine, S. (2017). CAD2RL: Real single-image flight without a single real image. *RSS*.
- [7] OpenAI, et al. (2019). Solving Rubik’s Cube with a Robot Hand. *arXiv*.
- [8] Mehta, B., et al. (2020). Active domain randomization. *CoRL*.
- [9] Muratore, F., et al. (2021). Data-efficient domain randomization with bayesian optimization. *IEEE RA-L*.
- [10] Tiboni, G., et al. (2024). Domain Randomization via Entropy Maximization. *ICLR*.
- [11] Kumar, A., et al. (2021). RMA: Rapid motor adaptation for legged robots. *RSS*.
- [12] Ren, A. Z., et al. (2023). AdaptSim: Task-Driven Simulation Adaptation for Sim-to-Real Transfer. *CoRL*.
- [13] Yu, W., et al. (2017). Preparing for the unknown: Learning a universal policy with online system identification. *RSS*.
- [14] Margolis, G., & Agrawal, P. (2023). Walk these ways: Tuning robot control for generalization with multiplicity of behavior. *CoRL*.
- [15] Nagabandi, A., et al. (2019). Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. *ICLR*.
- [16] Ma, Y. J., et al. (2024). DrEureka: Language Model Guided Sim-To-Real Transfer. *RSS*.
- [17] Ma, Y. J., et al. (2023). Eureka: Human-level reward design via coding large language models. *ICLR*.

- [18] Wang, L., et al. (2024). GenSim: Generating Robotic Simulation Tasks via Large Language Models. *ICLR*.
- [19] Wang, Y., et al. (2024). RoboGen: Towards Unleashing Infinite Data for Automated Robot Learning. *ICML*.
- [20] Yu, W., et al. (2023). Language to rewards for robotic skill synthesis. *CoRL*.
- [21] Handa, A., et al. (2023). Dextreme: Transfer of agile in-hand manipulation from simulation to reality. *ICRA*.
- [22] James, S., et al. (2019). Sim-to-Real via Sim-to-Sim: Data-efficient Robotic Grasping. *CVPR*.