用于机器人学习的自适应改进循环

Calvin Luo[*,1] , Zilai Zeng[*,1], Mingxi Jia[1], Yilun Du[2], Chen Sun[1]
[1]Brown University, [2]Harvard University

Abstract: 基于专家示范训练的视频生成模型已被用作高性能文本条件的视觉规划器，用于解决机器人任务。然而，未见任务的泛化仍然是一个挑战。虽然通过利用来自额外预收集离线数据源（如网络规模的视频数据集）的学习先验知识可以促进改进泛化，但在经验时代，我们旨在设计能够通过自我收集的行为以在线方式持续改进的代理。在这项工作中，我们提出了自适应改进循环（SAIL），其中域内视频模型通过适应互联网规模的预训练视频模型收集的自我生成轨迹进行迭代更新，并稳定提高指定的任务性能。我们将 SAIL 应用于多样化的 MetaWorld 任务套件，以及在真实机器人手臂上进行的两个操作任务，发现性能改进在多个迭代中不断涌现于最初在域内视频模型训练时未见的新任务。此外，我们发现 SAIL 对于自我收集经验是否以及如何过滤，以及初始域内示范的质量，表现出惊人的鲁棒性。通过总结互联网规模数据的适应和在线经验学习，我们展示了一种通过自我改进迭代引导高性能视频模型的方法，以解决新颖的机器人任务。可视化和代码可以在 diffusion-supervision.github.io/sail/ 处找到。

Keywords: Planning, Adaptation, Self-Improvement, Robots, Learning

1 引言

视频生成建模能力的进步直接导致它们在机器人应用中作为视觉规划者的使用增加 [1, 2, 3, 4] 。通过文本条件生成的视频帧形式的合成视觉计划，可以通过逆动力学模型（IDMs）转化为可执行的动作。虽然 IDMs 通常在任务中具有稳健性，但视频生成模型所训练的数据会严重影响下游机器人性能和泛化能力。当在领域内的专家行为示例上进行显式优化时，这些视觉规划者能够以稳健的方式合成成功的计划来解决演示的任务。然而，对于任意的机器人设置而言，大规模的专家质量数据集可能并不容易获得，收集起来可能成本非常高昂。数据规模的匮乏可能会限制仅在领域内视频上训练的视频模型在新的任务中展示广义规划能力。

整合来自互联网收集的大规模文本和视频数据的知识，即使在缺乏充足的特定领域视频情况下，也促进了改进的泛化。近期的工作 Adapt2Act [5] ，通过将一个在网络规模视频数据上预训练的大规模模型与一个通过得分组合在少量特定领域演示中训练的视频模型结合，创造了一个强大、具有泛化能力、受文本调控的视觉规划器。在宏观层面上，该被改进的视频模型借助于从网络预训练视频模型中获得的大规模动作先验和强大的零样本文本调控能力来促进泛化。同时，它可以利用特定领域视频模型来更好地生成符合机器人环境特定视觉特征和动态的视觉计划。结果就是，一个改进后的视频模型能够生成由自然语言调控的新颖、未见过的任务的特定领域外观计划。

尽管将视觉规划所用的数据量扩展到了互联网层次，模型仍然只能访问纯粹离线的数据，这在下游性能上仍可能是有限的。相反，在经验时代，我们旨在设计能够通过自我收集的行为和反馈来不断提升的智能体。以这种方式，智能体可以突破所提供的数据的限制，自行学习以改进对特定任务的性能。因此，我们提出了自适应改进循环（SAIL），通过在线经验迭代地自我提升视频模型，即使是对于最初环境演示数据集中未曾出现的行为。如图 1 所示，我们通过让机器人智能体按照视觉计划收集的数据迭代更新视频生成模型来构建一个循环，其中计划的质量通过适应冻结的、经过互联网预训练的视频模型得到改进。

*: Equal contribution. Correspondence to: calvin_luo@brown.edu and zilai_zeng@brown.edu.

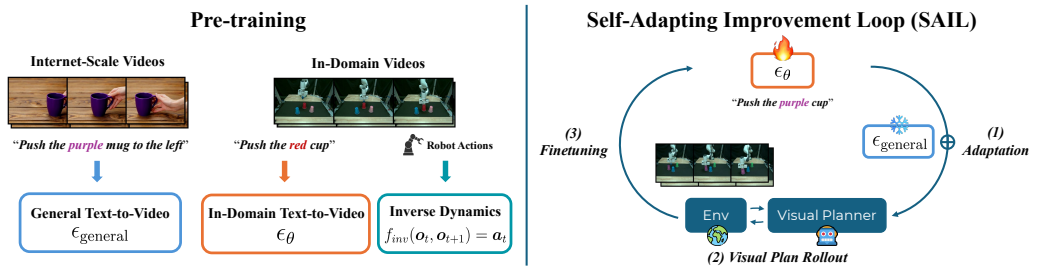


Figure 1: SAIL 框架。SAIL 利用两个预训练的视频生成模型（左）：一个是在互联网规模数据上进行的一般预训练模型，另一个是在一组通用的领域内演示上进行预训练的模型。将这两个组件组合起来可以形成一个具有强先验知识的视觉规划器，当用于与环境交互时，即使是初次未见的任务也能够生成成功率提高的轨迹。在自适应改进循环（SAIL）中，这些轨迹会被迭代地反馈以微调该领域内模型（右），从而通过自我收集的在线经验改善整体上适应后的视觉规划器质量。

我们在 MetaWorld 任务套件上对 SAIL 进行了广泛评估，重点关注在域内模型初始训练期间未见过的新任务。我们发现，通过适应合成的视觉计划的成功率确实在迭代中得到了提高。至关重要的是，我们强调利用大规模预训练的文本条件视频模型进行适应对于促进自我改进至关重要，因为它提供了文本条件的泛化能力和运动先验。此外，通过对设计决策的消融实验，我们发现 SAIL 对自我收集体验的过滤策略的存在以及域内模型初始训练所用的示范数据的质量具有相对的鲁棒性。我们还将 SAIL 应用于实际的机器人手臂，用于两种不同的操作任务：选择并推动一个有颜色的物体，以及选择并打开一个有颜色的抽屉。我们展示了在初始离线训练期间未见过的颜色组合的性能通过 SAIL 在多次迭代中有所改善。

2 相关工作

用于决策制定的视频生成。最近在视频模型方面的进展在视频合成中实现了前所未有的视觉质量和物理逼真性 [6, 7, 8, 9, 10] 。这表明通过视频来总结世界动态的潜力 [11, 12] ，并激发了将视频模型应用于解决决策问题的灵感 [13, 14, 2, 15, 4] 。先前的工作已将视频生成模型用作奖励函数 [16, 13, 17] 、动态模型 [2, 12, 18] 和基于像素的规划器 [3, 19, 1, 20] 。如同在 UniPi [1] 中，我们利用视频模型预测文本条件的视觉计划，这些计划描绘了未来的结果，随后通过逆向动态转化为行动。虽然这种视觉规划器的性能经常受到其离线预训练数据的限制，但我们的方法允许通过从在线环境交互中学习来进行迭代改进。

调整预训练的视频模型。在将一般预训练的视频模型应用于专业任务时，通常需要进行调整以实现定制化生成。对于基于图像模型的视频模型 [6, 21, 22, 23] ，可以使用图像定制技术，如文本反演 [24] 和 DreamBooth [25] ，来注入特定主题信息进行视频生成。为了获得细粒度的可控性，DreamVideo [26] 分别学习了两个特定的适配器，用于捕捉主体外观和运动控制。

此外，Video Adapter [27] 提出了概率自适应 (PA)，这是一种在采样阶段通过分数组合进行自适应的技术，无需微调大型预训练模型的权重。Adapt2Act [5] 将概率自适应扩展为其逆向 (IPA)，并利用视频自适应技术创建一个性能优良的视觉规划器，以解决基于自然语言的全新决策任务。在本文中，IPA 作为一种方法，用于提高新任务的视觉规划能力，其中编排后的结果会被收集为经验，并用于迭代微调域内视频模型，以提高其规划能力。

自我改进生成模型。通过从自生产的累积经验中学习不断改进是智能体的一项基本能力。之前的研究已经证明，通过自生成输出改进大型语言模型的有效性 [28, 29, 30] ，其中大型语言模型可以作为其自身的奖励函数 [31] 进行偏好优化，或作为数据合成器 [32] 用于监督微调。然而，用于视频生成模型的类似自我改进方法仍然未得到充分探索。与我们的工作最相关的是，VideoAgent [33] 通过自我条件一致性和来自视觉语言模型的反馈来优化视频生成，并收集成功的计划展开用于微调。我们则基于自适应改进循环，在初始域内训练时未看到的任务中，利用网络规模的视频先验合成改进的视觉计划。此外，我们的方法即使在初始模型训练于次优数据且对微调数据的筛选要求明显放宽的情况下，仍能实现自我改进。

Algorithm 1 自适应改进循环 (SAIL)

Input: Initial in-domain video model ϵ_θ , Inverse dynamics model f , Frozen internet-pretrained video model $\epsilon_{\text{general}}$, Number of iterations K , Number of rollouts per iteration N , Environment `env` , Task prompt g , In-domain initial training data $\mathcal{D}_{\text{ini}}$

Output: Self-improved in-domain model $\hat{\epsilon}_\theta$

$\hat{\epsilon}_\theta \leftarrow \epsilon_\theta$

$\mathcal{D} \leftarrow \mathcal{D}_{\text{ini}}$ or ϕ ▷ Initialize finetuning data with $\mathcal{D}_{\text{ini}}$ or an empty set

for $i = 1, ..., K$ do

  $\mathcal{D}_{\text{self}} \leftarrow \phi$

  $\tilde{\epsilon}_{\text{inv}} \leftarrow \texttt{IPA}(\hat{\epsilon}_\theta, \epsilon_{\text{general}}, g)$

  for $j = 1, ..., N$ do

    `env.reset`(g)

    $\mathcal{D}_{\text{self}} \leftarrow \mathcal{D}_{\text{self}} \cup$ `Visual_Planning_Rollout`(`env`, $\tilde{\epsilon}_{\text{inv}}$, f) ▷ Optional data filtering

  end for

  $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_{\text{self}}$

  Finetune in-domain model $\hat{\epsilon}_\theta$ with accumulated data $\mathcal{D}$ ▷ f can be optionally finetuned

end for

return $\hat{\epsilon}_\theta$

3 方法

我们引入了自适应改进循环 (SAIL)，该方法中视频生成模型最初在通用域内示例集上训练，然后以自适应的方式迭代提高其在特定感兴趣任务上的视觉规划性能。在第 3.2 节中，我们描述了如何将一个小型域内视频模型与一个经过通用预训练的文本到视频模型整合，以生成一个强大的、可推广的域内视觉规划器。最后，在第 3.3 节中，我们展示了 SAIL 如何通过在自收集经验上进行迭代微调，将一个域内视频模型启动为一个高性能的视觉规划器，以解决新颖的机器人控制任务。

3.1 视频模型作为视觉规划器

在想象中合成一个视觉方案，然后通过将其转换为动作来执行它，是利用视频生成模型进行决策的一种直观而有效的方法。以前的工作已经成功地将文本引导的视频生成应用于任务规划 [14, 1, 19] ，跨越各种机器人配置和环境设置。

具体来说，我们基于 UniPi 框架 [14] 实现，其中使用文本到视频模型来合成文本条件下的未来帧序列作为任务计划。为了实际实现该计划，我们使用一个单独训练的逆动力学模型 (IDM) 将连续的视觉帧对转换为可执行的机器人动作，然后直接在与环境的交互中执行这些动作。视觉规划为实践者提供了灵活的计算折中；在高层次上，频繁重新规划通常会产生高计算成本，但通常会提高计划的准确性，而不频繁重新规划则成本低，但可能会受到累计误差的影响。由于 IDM 是任务无关的，并且可以从一般领域内交互数据中进行训练，因此视觉规划框架下的任务泛化和性能很大程度上是视频生成模型质量的产物。在这项工作中，我们关注的是这样的视频生成模型如何通过在线自我收集的经验泛化和自适应到一个新的感兴趣的任务。

3.2 逆概率适应

之前的研究 [5] 已经调查了如何将领域内演示数据最佳集成到大规模预训练的视频模型中，以实现可推广的视觉规划；在这项工作中，我们利用类似的见解成功地将实时经验整合到视觉规划者中以进行迭代自我改进。反向概率适应 [5, 27] (IPA) 是一种无需训练的方法，它适应一般预训练的文本到视频模型的特定领域视频生成。为了执行适应，采样过程中领域内视频模型 ϵ_θ （训练于小样本演示数据集）所预测的得分与网络规模预训练模型 $\epsilon_{\text{general}}$

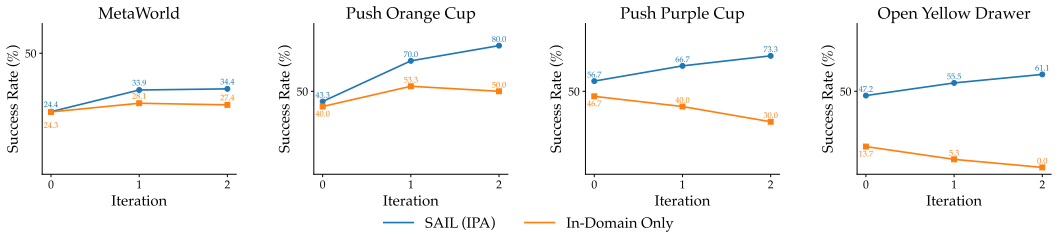


Figure 2: SAIL 在 MetaWorld 和 Panda Arm 上的结果。我们报告了 MetaWorld 上 6 个任务的平均性能，以及用于 Panda Arm 实验的两个新推送和一个新抽屉打开任务。与仅在域内的方法相比，SAIL 表现出更稳健的改进行为而没有性能下降，并在真实机器人任务上实现了持续改进。

的得分预测组合，如下图函数所示：

$$\tilde{\epsilon}_{\text{inv}} = \epsilon_{\text{general}}(\tau_t, t) + \alpha\Big(\epsilon_{\text{general}}(\tau_t, t \mid \text{text}) + \gamma\epsilon_\theta(\tau_t, t \mid \text{text}) - \epsilon_{\text{general}}(\tau_t, t)\Big) \tag{1}$$

其中 γ 是先验强度，而 α 是文本条件引导比例。从直觉上看，小规模的领域内文本到视频模型作为一种概率知识先验，在采样过程中引导小规模领域内模型的生成过程。先前的工作 [5] 发现，通过 IPA 构建的视觉规划器展示了强大的泛化能力和领域内理解能力；它能够合成表现优异的视觉计划，即使是视频模型训练中未见的新任务，这可能源于 IPA 利用大规模预训练模型作为主要降噪器，其固有拥有更强的文本条件泛化能力。

3.3 自适应改进循环

尽管通过增加使用的数据量至互联网规模，可能对新任务实现更强的文本条件泛化，但任务性能仍然是所用视频模型的一个固定函数，进而是所观察到的数据的一个固定函数。因此，在本文中，我们希望设计能够不仅将离线数据作为有助于泛化的先验来利用的智能体，还能在此基础上，通过自身收集的在线经验数据迭代改进的智能体。

因此，我们提出了自适应改进循环框架，这个框架结合了离线数据和在线体验，以创建一个视觉规划器，针对特定感兴趣的任务进行迭代改进。SAIL 框架以一个领域内视频模型 ϵ_θ 开始，该模型经过在环境中一组任务演示的预训练。在每个迭代中，领域内视频模型通过 IPA 与一个大规模预训练的视频模型 $\epsilon_{\text{general}}$ 进行整合。经过适应的视频模型然后作为一个视觉规划器与环境互动，解决初始训练阶段未必观察到的任务；在 SAIL 框架中，通过这种互动收集的轨迹被用于进一步微调领域内的视频模型（如在算法 1 中所示）。随着领域内模型对在一个新的任务中部署时自己收集的经验进行适应，它随着时间推移提高解决特定任务的能力。通过这种方式，SAIL 通过自适应改进循环将一个领域内视频模型迭代引导成为一个强大的视觉规划器，以解决特定感兴趣的任务。

我们证明，正是使用网络规模的数据与自我收集的经验相结合，才促进了一个良性循环；在我们的实验中，我们展示了单独进行训练未能显示出强烈的迭代改进。我们进一步通过初始化数据质量以及过滤策略的消融实验对我们的框架进行了压力测试。我们发现，SAIL 是一种稳健的方法，可以通过有效利用离线数据和在线经验来逐步适应任务。

4 实验

我们研究了 SAIL 如何能够改进初始在有限的示范和任务集上训练的领域内视频模型，以通过自收集的经验进一步解决新的机器人控制任务。我们重点关注两种主要的机器人设置来评价 SAIL：MetaWorld-v2 [34] 模拟环境，以及一个实际的 Franka Emika Panda 机械臂机器人。我们描述了每个环境的实验设置，以及考虑的不同设计决策。

4.1 实验设置与评估

合成环境：MetaWorld 包含了大量的任务选择，使我们能够通过 SAIL 对许多不同的未公开新颖任务彻底评估视觉规划性能趋势。此外，MetaWorld 提供了真实的成功评估，能够

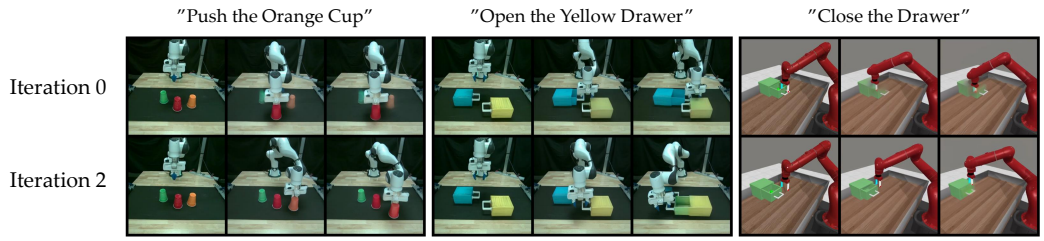


Figure 3: 视觉计划优化的定性结果。我们展示了在不同任务和设置中视觉计划的样例，分别在迭代 0（上图）和迭代 2（下图），初始时对象位置是随机的。虽然迭代 0 的视觉计划呈现模糊的对象，未能完成指定任务，但经过两个 SAIL 迭代后，我们的方法可以合成出正确的视觉计划（伴随轻微的色彩漂移）。

在任务表现和改进方面进行严格的定量比较。在 MetaWorld 实验中，我们首先从 7 个不同的任务中收集 25 个演示（在表 一个 1 中用星号表示）用于初始领域内视频模型和逆动力学模型的训练。随后，我们使用通过 IPA 与大规模预训练文本到视频模型适配的领域内视频模型作为 6 个任务的视觉规划工具，其中 5 个是新任务（在表 一个 1 中未用星号表示）。我们利用 SAIL 通过自我收集的经验迭代改进领域内模型；在每次迭代中，我们在视觉规划期间从环境中收集 30 条轨迹以进行领域内微调。

真实世界环境：在真实世界中将 SAIL 部署在机器人手臂上展示了该方法的实用性，并测试了其对真实世界因素的鲁棒性，例如光照条件。在一个实验中，我们使用 Franka Emika Panda 机器人手臂执行用户提供文本提示指定的推杯任务。与每个任务都有自己独特视觉设置的 MetaWorld 设置相比，我们将杯子实验构建为一个由三个不同颜色杯子组成的一致场景设置（图 1 ）。成功的衡量标准是机器人手臂能否准确定位指定颜色的杯子并向前推。在自然语言条件下，为测试泛化，我们评估在未见过的杯子颜色上的成功规划和执行性能。在实践中，我们使用四种颜色（红色、绿色、蓝色、粉色）进行域内训练，并使用两种新颜色（橙色、紫色）进行测试泛化。这转化为由看过的颜色组合而成的 12 个可能的独特任务，并且我们对每个任务进行 10 次人类远程操作演示训练，总计 120 个训练视频。然后，泛化评估是对每个可能的看过颜色与新颜色的组合进行 5 次展开并取平均值，总计 30 个视频。对于两种新颜色，我们使用相同的预训练域内视频模型初始化 SAIL。在每次 SAIL 迭代中，我们将之前自我收集的数据与初始演示结合进行域内微调。

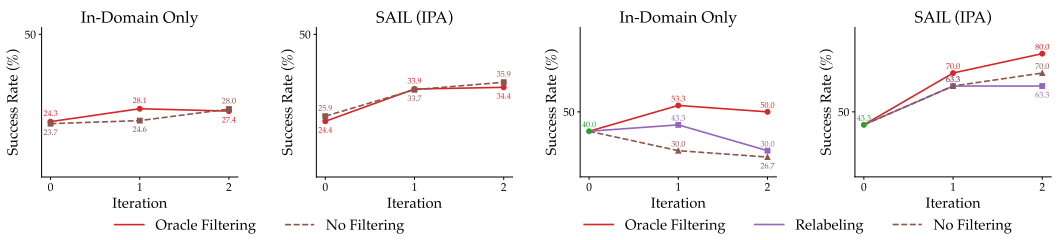
在第二个真实机器人实验中，我们利用 Panda 机械臂选择并打开一个由用户提供的文本提示指定的抽屉。场景被设置为两个颜色截然不同的关闭抽屉，机器人会收到一个特别颜色的提示，并期望打开相应的抽屉。我们使用一组三种颜色（红色、绿色、蓝色）进行域内训练，并使用一种新颜色（黄色）来测试泛化能力。在每对已见颜色的 24 种可能的抽屉放置组合中，总共有 6 对，因此总共使用了 144 段人工远程操控示范训练视频。与推杯实验一致，我们用一半的可能组合进行评估；因此，性能计算为每次新颜色与已见颜色配对的 12 次实验结果的平均值，总共每次迭代收集 36 个自行收集的轨迹。

在两个真实机器人实验中，成功与否由人为来判断和评估。相同的成功信号也用于对运行结果进行可选的数据过滤。我们在第 4.3 节研究数据过滤的影响，并且在第 4.4 节的实验中不使用过滤。

实现细节：我们基于 AVDC [3] 实现我们域内视频模型，在去噪 U-Net 的每一层添加了一个交叉注意力层，以进一步提高文本条件能力。我们训练域内视频模型以预测未来 8 帧，这些帧以当前观测和任务提示为条件，在 MetaWorld 中帧间隔为 1，在真实机器人实验中帧间隔为 16。对于大型预训练文本到视频模型，我们使用 AnimateDiff [6]（∼ 2B 参数），该模型在 WebVid-10M [35] 上进行了预训练。SAIL 的每次迭代对域内视频模型在 MetaWorld 和 Panda Arm 抽屉打开任务上进行 10,000 次迭代精调，学习率为 1e-5，在 Panda Arm 推动任务上进行 8,000 次迭代精调，学习率为 2e-5。

4.2 使用 SAIL 进行视觉规划

我们报告了通过 3 次 SAIL 迭代针对 MetaWorld 和 Panda 机械臂的增量视觉规划结果。在每次迭代中，我们会过滤掉不成功的轨迹，仅对成功的轨迹进行微调。在图 2 中左侧，我



(a) 元世界　　(b) 熊猫机械臂推送

Figure 4: 数据过滤的消融实验。我们评估了使用理想成功信号过滤自收集数据如何影响 SAIL 在 MetaWorld（4a）和 Panda 机械臂（4b）设置上的性能。我们还提供了在真实机器人实验中使用重新标记策略的额外结果。我们观察到，SAIL 在两种基准测试中均能提高任务性能，而无需对收集的数据进行过滤，这再次确认了我们的方法在缺乏理想过滤信号时的鲁棒性。

们展示了 6 个 MetaWorld 任务（其中 5 个是新颖任务）中的平均成功率，对比了仅在域内的情况和 IPA（每个任务的性能详见表 一个 5 ）。我们发现，通过适应，初始成功率在各个任务中更高，凸显了使用大规模离线数据作为新颖任务泛化的强大先验的好处。此外，我们发现 SAIL 在利用自收集经验以进一步提升性能方面是有效的，因为性能随着每次迭代而提高。值得注意的是，单独使用域内模型确实在初始阶段有一些提升，但这种提升在多次迭代中并不稳定，且总体性能不如通过 SAIL 达到的那么高。

在图 2 的中间两个图中，我们展示了 SAIL 在熊猫机械臂上的应用，用于推动橙色和紫色杯子，这些都是最初未见过的颜色。在 30 次试验中，跨越新颜色与之前见过的颜色的不同组合，我们发现 SAIL 在每次迭代中持续提高性能。与 MetaWorld 的结果一样，出现了类似的趋势，即仅使用域内模型并不能带来显著的改进；相反，在推动紫色杯子的情况下，即使域内模型同样在过滤后的自收集经验上进行了微调，性能还是单调下降。在图 2 的最右图中，我们展示了 SAIL 迭代中的结果，该结果用于打开一个新颜色的抽屉（在图 一个 8 和图 一个 9 中可视化）。平均每次迭代进行 36 次实验后，我们再一次展示了如何通过 SAIL 在过滤后的经验上微调，提高了稳步改进，而仅使用域内模型则导致性能稳步下降。总体而言，这些结果表明 SAIL 通过利用大规模离线数据结合在线经验数据，在新任务的模拟环境和现实环境中均实现了自我改进的性能。

在图 3 中，我们定性地展示了真实机器人操作和 MetaWorld 任务在迭代 0（上）和迭代 2（下）的视觉计划。在迭代 0 未观察到指定任务的任何演示时，IPA 常常合成一个带有模糊物体的视觉计划，使得机器人手臂错误地执行任务。另一方面，经过两次 SAIL 迭代，不仅提高了视觉计划的清晰度，而且还展示了在相同初始布局下成功完成任务的行为。通过遵循一种逆动力学模型的计划，机器人手臂能够在实际环境交互中成功执行任务（如附录 ?? 所示）。

4.3 **无经验过滤的** SAIL

虽然利用自采数据是一种具有前景的规模化自我改进方法，但过滤所收集的经验往往需要一定程度的人为干预，无论是通过人工确定成功轨迹还是设计质量控制的启发式方法。因此，我们研究了不同的过滤技术如何影响 SAIL 的性能，或者 SAIL 是否对这种设计决策具有鲁棒性。在 MetaWorld 和 Panda Arm 设置中，我们比较了使用真实值或人工评估的成功概念来过滤域内模型进行微调的轨迹与完全不使用任何过滤并利用所有已实现的轨迹而不考虑结果之间的区别。

在图 4a 中，我们观察到对于域内和 SAIL，忽略过滤实际上略微优于进行过滤。这是一个令人惊讶的结果，因为这表明即使是失败的演示也可能作为有意义行为的来源，并进一步促进整体任务的改进。另一方面，在图 4b 中，对于 Panda 机械臂，我们观察到即使不进行过滤，SAIL 仍然在每次迭代中持续促进改进。这是一个令人鼓舞的发现，因为这表明即使在手动挑选经验代价高昂的环境中，自我改进仍然可以发生。

我们还研究了一种新颖的滤波方案是否对 Panda 机械臂的推送任务有用，这种方案被称为重标记。在这种设置中，所有轨迹再次用于微调域内文本到视频模型，但不成功的轨迹前面

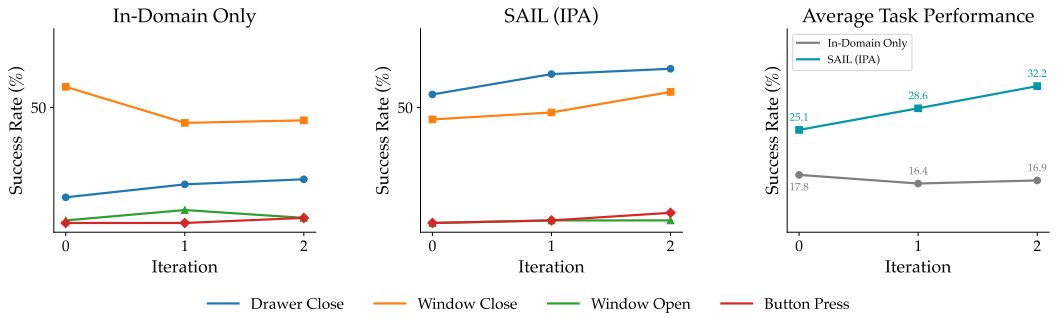


Figure 5: SAIL 在域内数据不足的情况下的结果。我们报告了 SAIL 在四个新 MetaWorld 任务上的单独表现，以及其在 SAIL 迭代过程中平均表现。即使在域内数据不足的情况下，SAIL 的持续改进行为仍然保持强劲，超过了仅使用域内数据的基线。

会加上“not”字样的文本提示以表示失败。我们发现，对于域内模型，重标记确实比不使用任何过滤更可取，但在利用大规模文本到视频的先验知识时，并不能实质性地帮助提高性能。

4.4 使用次优数据的 SAIL

视觉规划器通常直接在领域内专家的演示上进行训练，这不仅在优化过程中向生成模型传递环境特定的视觉特征、物理性质和交互动态，还传递了成功的概念和最优行为。然而，对于任意环境而言，这种高质量的领域内数据可能在大规模收集和策划过程中成本高昂。另一方面，次优演示数据，例如在收集过程中利用随机动作，通常可能更便宜；然而，使用大量低质量数据进行训练可能不会产生一个绩效良好的视觉规划模型，无法生成值得遵循的计划。一个自然的问题是，SAIL 对初始化数据的鲁棒性如何，或者当只有次优演示可用时，是否仍然可以创建一个性能卓越的视频规划器。

在我们的设定中，我们构建次优数据为模拟轨迹，其中 70 % 的时间选择的是随机动作，30 % 使用的是专家动作。由于这种交互过程，生成的轨迹无法成功解决复杂任务。我们也继续保持之前不使用任何过滤策略的设定。尽管这种设置，SAIL 继续通过 IPA 有效结合大规模离线数据和自主收集的经验来实现性能优化。在 MetaWorld 中，我们发现对于四个突出的任务（全部为未见过任务），SAIL 表现出持续改善的行为，如图 5 中的中间和最右图所示。SAIL 对初始域内数据质量的鲁棒性可能归因于 IPA 克服次优性差距的能力 [5] 。此外，如图 5 所示，仅使用域内模型并未显示出平均上的显著改善。没有大规模文本到视频模型的适应，仅在次优数据上训练的域内模型可能难以收集足够成功的在线经验，并随后通过未经滤过的微调强化其次优行为。因此，这强调了 SAIL 的鲁棒性——尽管没有使用任何过滤策略，并且仅从次优数据演示开始初始化，它仍然能够通过自主收集的经验引导一个有力的视觉规划器来应对新的任务。

5 结论

在这项工作中，我们提出了 SAIL，一种通过视觉规划来解决新型机器人任务的自适应改进循环。从一组小型示范预训练的域内视频模型初始化，SAIL 利用 IPA 与一个预训练在互联网上的大规模视频模型作为高性能的可泛化视觉规划器，迭代地收集经验轨迹来提升域内视频模型。这种方式使得 SAIL 能够结合大规模的离线数据与在线自获取的经验，为所需任务启动一个高性能的文本条件视觉规划器。

通过我们的实验，我们证明了 SAIL 是一个稳健的框架，不仅在没有过滤技术的情况下，而且在初始示范集的质量方面。我们展示了 SAIL 不仅能够在合成环境中成功作为一个自我改进的视觉规划器，而且还可以在真实世界的机器人手臂上部署。

6

局限性

SAIL 隐含地假设，初始的域内模型通过与互联网上预训练的视频模型的适应，能够实现合理的成功率，以便于收集在线经验并自我改进模型。当新任务过于具有挑战性时，这一假设可能不成立。此外，选择互联网预训练的视频模型时，在计算成本上对视频质量（因此运动先验的强度等）提出了权衡。在这项工作中，我们选择 AnimateDiff [6] 作为一种具有合理生成质量和良好计算效率的大规模预训练文本到视频模型，更多近期的视频生成模型可以被探索以获得更好的视觉质量和对下游机器人性能的潜在改进。

Acknowledgments

This work is partially supported by Samsung and NASA. Our research was conducted using computational resources at the Center for Computation and Visualization at Brown University. We would like to thank Professors George Konidaris and Stefanie Tellex for their generous support for our real-robot experiments, and Skye Thompson for helpful initial discussions. Calvin thanks Kayan Shih and family for their kindness and support during the paper writing process.

References

[1] Y. Du, S. Yang, P. Florence, F. Xia, A. Wahid, brian ichter, P. Sermanet, T. Yu, P. Abbeel, J. B. Tenenbaum, L. P. Kaelbling, A. Zeng, and J. Tompson. Video language planning. In International Conference on Learning Representations (ICLR) , 2024.

[2] M. Yang, Y. Du, K. Ghasemipour, J. Tompson, D. Schuurmans, and P. Abbeel. Learning interactive real-world simulators. arXiv preprint arXiv:2310.06114 , 2023.

[3] P.-C. Ko, J. Mao, Y. Du, S.-H. Sun, and J. B. Tenenbaum. Learning to act from actionless videos through dense correspondences. In International Conference on Learning Representations (ICLR) , 2024.

[4] J. Liang, R. Liu, E. Ozguroglu, S. Sudhakar, A. Dave, P. Tokmakov, S. Song, and C. Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation. arXiv preprint arXiv:2406.16862 , 2024.

[5] C. Luo, Z. Zeng, Y. Du, and C. Sun. Solving new tasks by adapting internet video knowledge. In The Thirteenth International Conference on Learning Representations , 2025. URL https://openreview.net/forum?id=p01BR4njlY.

[6] Y. Guo, C. Yang, A. Rao, Y. Wang, Y. Qiao, D. Lin, and B. Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. arXiv preprint arXiv:2307.04725 , 2023.

[7] Z. Yang, J. Teng, W. Zheng, M. Ding, S. Huang, J. Xu, Y. Yang, W. Hong, X. Zhang, G. Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. arXiv preprint arXiv:2408.06072 , 2024.

[8] T. Brooks, B. Peebles, C. Holmes, W. DePue, Y. Guo, L. Jing, D. Schnurr, J. Taylor, T. Luhman, E. Luhman, C. Ng, R. Wang, and A. Ramesh. Video generation models as world simulators. OpenAI Blog , 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

[9] Veo-Team, :, A. Gupta, A. Razavi, A. Toor, A. Gupta, D. Erhan, E. Shaw, E. Lau, F. Belletti, G. Barth-Maron, G. Shaw, H. Erdogan, H. Sidahmed, H. Nandwani, H. Moraldo, H. Kim, I. Blok, J. Donahue, J. Lezama, K. Mathewson, K. David, M. K. Lorrain, M. van Zee, M. Narasimhan, M. Wang, M. Babaeizadeh, N. Papalampidi, N. Pezzotti, N. Jha, P. Barnes, P.-J. Kindermans, R. Hornung, R. Villegas, R. Poplin, S. Zaiem, S. Dieleman, S. Ebrahimi, S. Wisdom, S. Zhang, S. Fruchter, S. Nørly, W. Hua, X. Yan, Y. Du, and Y. Chen. Veo 2. 2024. URL https://deepmind.google/technologies/veo/veo-2/.

[10] A. Wang, B. Ai, B. Wen, C. Mao, C.-W. Xie, D. Chen, F. Yu, H. Zhao, J. Yang, J. Zeng, J. Wang, J. Zhang, J. Zhou, J. Wang, J. Chen, K. Zhu, K. Zhao, K. Yan, L. Huang, M. Feng, N. Zhang, P. Li, P. Wu, R. Chu, R. Feng, S. Zhang, S. Sun, T. Fang, T. Wang, T. Gui, T. Weng, T. Shen, W. Lin, W. Wang, W. Wang, W. Zhou, W. Wang, W. Shen, W. Yu, X. Shi, X. Huang, X. Xu, Y. Kou, Y. Lv, Y. Li, Y. Liu, Y. Wang, Y. Zhang, Y. Huang, Y. Li, Y. Wu, Y. Liu, Y. Pan, Y. Zheng, Y. Hong, Y. Shi, Y. Feng, Z. Jiang, Z. Han, Z.-F. Wu, and Z. Liu. Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 , 2025.

[11] S. Yang, J. Walker, J. Parker-Holder, Y. Du, J. Bruce, A. Barreto, P. Abbeel, and D. Schuurmans. Video as the new language for real-world decision making. arXiv preprint arXiv:2402.17139 , 2024.

[12] J. Bruce, M. Dennis, A. Edwards, J. Parker-Holder, Y. Shi, E. Hughes, M. Lai, A. Mavalankar, R. Steigerwald, C. Apps, Y. Aytar, S. Bechtle, F. M. P. Behbahani,

S. Chan, N. M. O. Heess, L. Gonzalez, S. Osindero, S. Ozair, S. Reed, J. Zhang, K. Zolna, J. Clune, N. de Freitas, S. Singh, and T. Rocktaschel. Genie: Generative interactive environments. arXiv preprint arXiv:2402.15391 , 2024.

[13] A. Escontrela, A. Adeniji, W. Yan, A. Jain, X. B. Peng, K. Goldberg, Y. Lee, D. Hafner, and P. Abbeel. Video prediction models as rewards for reinforcement learning. In Conference on Neural Information Processing Systems (NeurIPS) , 2023.

[14] Y. Du, S. Yang, B. Dai, H. Dai, O. Nachum, J. Tenenbaum, D. Schuurmans, and P. Abbeel. Learning universal policies via text-guided video generation. Advances in Neural Information Processing Systems , 36, 2024.

[15] R. McCarthy, D. C. Tan, D. Schmidt, F. Acero, N. Herr, Y. Du, T. G. Thuruthel, and Z. Li. Towards generalist robot learning from internet video: A survey. arXiv preprint arXiv:2404.19664 , 2024.

[16] C. Luo, M. He, Z. Zeng, and C. Sun. Text-aware diffusion for policy learning. In Advances in Neural Information Processing Systems , volume 37, 2024.

[17] T. Huang, G. Jiang, Y. Ze, and H. Xu. Diffusion reward: Learning rewards via conditional video diffusion. arXiv preprint arXiv:2312.14134 , 2023.

[18] D. Valevski, Y. Leviathan, M. Arar, and S. Fruchter. Diffusion models are real-time game engines. arXiv preprint arXiv:2408.14837 , 2024.

[19] A. Ajay, S. Han, Y. Du, S. Li, A. Gupta, T. Jaakkola, J. Tenenbaum, L. Kaelbling, A. Srivastava, and P. Agrawal. Compositional foundation models for hierarchical planning. In Conference on Neural Information Processing Systems (NeurIPS) , 2023.

[20] S. Zhou, Y. Du, J. Chen, Y. Li, D. Y. Yeung, and C. Gan. Robodreamer: Learning compositional world models for robot imagination. arXiv preprint arXiv:2404.12377 , 2024.

[21] J. Z. Wu, Y. Ge, X. Wang, S. W. Lei, Y. Gu, Y. Shi, W. Hsu, Y. Shan, X. Qie, and M. Z. Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In Proceedings of the IEEE/CVF International Conference on Computer Vision , pages 7623–7633, 2023.

[22] U. Singer, A. Polyak, T. Hayes, X. Yin, J. An, S. Zhang, Q. Hu, H. Yang, O. Ashual, O. Gafni, D. Parikh, S. Gupta, and Y. Taigman. Make-a-video: Text-to-video generation without text-video data. In The Eleventh International Conference on Learning Representations , 2023. URL https://openreview.net/forum?id=nJfylDvgzlq.

[23] A. Blattmann, T. Dockhorn, S. Kulal, D. Mendelevitch, M. Kilian, D. Lorenz, Y. Levi, Z. English, V. Voleti, A. Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 , 2023.

[24] R. Gal, Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, and D. Cohen-or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In International Conference on Learning Representations (ICLR) , 2023.

[25] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition , 2023.

[26] Y. Wei, S. Zhang, Z. Qing, H. Yuan, Z. Liu, Y. Liu, Y. Zhang, J. Zhou, and H. Shan. Dreamvideo: Composing your dream videos with customized subject and motion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition , pages 6537–6549, 2024.

[27] M. Yang, Y. Du, B. Dai, D. Schuurmans, J. B. Tenenbaum, and P. Abbeel. Probabilistic adaptation of text-to-video models. arXiv preprint arXiv:2306.01872 , 2023.

[28] X. Yu, B. Peng, M. Galley, J. Gao, and Z. Yu. Teaching language models to self-improve through interactive demonstrations. In K. Duh, H. Gomez, and S. Bethard, editors, Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers) , pages 5127–5149, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi:10.18653/v1/2024.naacl-long.287. URL https://aclanthology.org/2024.naacl-long.287/.

[29] Y. Tian, B. Peng, L. Song, L. Jin, D. Yu, L. Han, H. Mi, and D. Yu. Toward self-improvement of LLMs via imagination, searching, and criticizing. In Conference on Neural Information Processing Systems (NeurIPS) , 2024.

[30] J. Huang, S. Gu, L. Hou, Y. Wu, X. Wang, H. Yu, and J. Han. Large language models can self-improve. In Conference on Empirical Methods in Natural Language Processing , 2022.

[31] W. Yuan, R. Y. Pang, K. Cho, X. Li, S. Sukhbaatar, J. Xu, and J. E. Weston. Self-rewarding language models. In International Conference on Machine Learning (ICML) , 2024.

[32] A. Patel, M. Hofmarcher, C. Leoveanu-Condrei, M.-C. Dinu, C. Callison-Burch, and S. Hochreiter. Large language models can self-improve at web agent tasks. arXiv , 2405.20309, 2024.

[33] A. Soni, S. Venkataraman, A. Chandra, S. Fischmeister, P. Liang, B. Dai, and S. Yang. Videoagent: Self-improving video generation. arXiv preprint arXiv:2410.10076 , 2024.

[34] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In Conference on robot learning , pages 1094–1100. PMLR, 2020.

[35] M. Bain, A. Nagrani, G. Varol, and A. Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In IEEE International Conference on Computer Vision (ICCV) , 2021.

[36] A. Majumdar, K. Yadav, S. Arnaud, J. Ma, C. Chen, S. Silwal, A. Jain, V.-P. Berges, T. Wu, J. Vakil, P. Abbeel, J. Malik, D. Batra, Y. Lin, O. Maksymets, A. Rajeswaran, and F. Meier. Where are we in the search for an artificial visual cortex for embodied intelligence? In Conference on Neural Information Processing Systems (NeurIPS) , 2023.

[37] Y. Guo, C. Yang, A. Rao, M. Agrawala, D. Lin, and B. Dai. SparseCtrl: adding sparse controls to text-to-video diffusion models. In European Conference on Computer Vision (ECCV) , 2024.

[38] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In International Conference on Learning Representations (ICLR) , 2021.

下面我们列出了用于评估 SAIL 的任务及相关文本提示。带有星号的任务是在域模型训练期间看到的演示。

Task	In-Domain Model Prompts	Internet-Domain Model Prompts
Assembly *	assembly	a robot arm placing a ring over a peg
Dial Turn *	dial turn	a robot arm turning a dial
Reach *	reach	a robot arm reaching a red sphere
Peg Unplug Side *	peg unplug side	a robot arm unplugging a gray peg
Lever Pull *	lever pull	a robot arm pulling a lever
Coffee Push *	coffee push	a robot arm pushing a white cup towards a coffee machine
Door Close *	door close	a robot arm closing a door
Window Close	window close	a robot arm closing a window
Window Open	window open	a robot arm opening a window
Drawer Close	drawer close	a robot arm closing a drawer
Drawer Open	drawer open	a robot arm open a drawer
Button Press	button press	a robot arm pushing a button
Push Red Cup *	red	a robot arm pushing the red cup
Push Blue Cup *	blue	a robot arm pushing the blue cup
Push Green Cup *	green	a robot arm pushing the green cup
Push Pink Cup *	pink	a robot arm pushing the pink cup
Push Orange Cup	orange	a robot arm pushing the orange cup
Push Purple Cup	purple	a robot arm pushing the purple cup
Open Red Drawer *	red	a robot arm opening the red drawer
Open Green Drawer *	green	a robot arm opening the green drawer
Open Blue Drawer *	blue	a robot arm opening the blue drawer
Open Yellow Drawer	yellow	a robot arm opening the yellow drawer

Table 一个 1：任务-提示组合。我们包括一份全面的任务列表及其文本提示，用于领域内的训练和评估。“ ∗ ”表示在领域内模型初始训练期间看到的任务。我们还提供了用于在 IPA 适应期间与互联网预训练的文本到视频模型接口的提示。

A 实现细节

我们在下文中提供了 SAIL 中使用的模型的详细架构配置及其相关的超参数设置。

逆动力学：根据之前的工作 [5]，我们设计了一个逆动力学模型，该模型是一个小型 MLP 网络，构建在预训练的基于像素的表示网络之上。IDM 以两个视频帧的嵌入作为输入，这些嵌入是使用 VC-1 [36] 提取的，输出的是在提供的帧之间实现过渡的动作预测。

在 Panda 机械臂实验中，IDM 的任务是预测所提供的最后一帧的末端执行器位置。然后，这一位置通过逆运动学在物理环境中执行。此外，两个视频帧之间有 16 帧的跳帧率；摄像机查询轨迹的频率非常高，以至于观察两帧时间上连续的帧与仅观察最后一帧相比，并没有显著区别。对于 MetaWorld 实验，这两个视频帧是连续的，因此跳帧率为 1。

用于实验的 IDM 的总参数数量为 85.81M。其中，85.80M 参数是从 VC-1 继承的，而我们的 IDM 设计由于额外的 MLP，仅额外贡献了 10759 个参数。

为了公平起见，我们在相同环境内的所有任务中重复使用相同的 IDM，并且在 SAIL 迭代中使用随后自收集的数据时不进行任何微调。以这种方式，IDM 是在一组已见任务上训练的，但即使对于那些具有新视觉设置的任务（如 MetaWorld），也能应用于潜在的新任务，而无需进一步修改。因此，在这些新任务上的后续成功不仅强调了已学习 IDM 的鲁棒性，还强调了合成视觉计划的视觉质量。IDM 训练的详细超参数在表 一个 2 中提供。

域内模型：我们重用了一个小规模扩散模型的实现，该模型依赖自然语言和来自 [3] 的初始像素帧。为了提高模型对文本的条件能力，我们在 U-Net 的每一层添加了一个额外的交叉注意力层，该层注重于 CLIP 编码的文本提示。具体来说，我们在 MetaWorld 设置中实例化了具有 3 个 ResNet 块的 UNet，在 Panda 机械臂任务中使用 2 个 ResNet 块。我们在表 一个 3 中报告了模型参数的详细列表。总共，域内模型在 MetaWorld 实验中包含 179.91M 个参数，在现实世界实验中包含 156.58M 个参数。我们在 MetaWorld 上进行了 70K 训练步和在 Panda 上进行了 88K 步的初始域内训练，批量大小为 8，学习率为 2e-5。在每个 SAIL 迭代中，我们以批量大小 4 和学习率 1e-5 在 MetaWorld 上对域内视频模型进行 10K 步的微调。在 Panda 机械臂上，我们在杯子推动任务中以批量大小 8 和学习率

Hyperparameter	Value
Input Dimension	1536
Output Dimension (MetaWorld)	4
Output Dimension (Panda)	7
Training Epochs	20
Learning Rate	1e-5
Optimizer	AdamW

Table 一个 2：逆动力学模型训练的超参数。我们列出了训练逆动力学模型的相关超参数。

2e-5 微调 8,000 步，在抽屉打开任务中以批量大小 8 和学习率 1e-5 微调 10,000 步。所有实验均在单个 NVIDIA A6000 或 RTX3090 GPU 上进行。

Component	# Parameters (Millions)
U-Net (MetaWorld)	116.71
U-Net (Panda Arm)	93.38
Text Encoder (`openai/clip-vit-base-patch32`)	63.2

Table 一个 3：领域内模型组件。SAIL 依赖于一个小型的领域内文本到视频模型，我们的实现基于之前的工作 [3] 。我们列出了所使用的模型架构组件的大小。

互联网-域模型：继 Adapt2Act [5] 之后，我们采用 AnimateDiff [6] 作为用于逆概率自适应的冻结互联网预训练视频模型。此外，我们使用 SparseCtrl [37] 以实现图像条件视频生成。模型组件及其参数数量列在表 一个 4 中。总的来说，AnimateDiff 包含 2.005B 个参数。

Component	# Parameters (Millions)
VAE (Encoder)	34.16
VAE (Decoder)	49.49
U-Net	1302.16
Text Encoder	123.06
ControlNet	496.73

Table 一个 4：AnimateDiff 组件。SAIL 依赖于一个互联网规模的文本到视频模型；在这项工作中，我们使用了 AnimateDiff。因此，我们列出了用于 AnimateDiff 检查点的组件大小。该检查点仅用于推理，不会以任何方式被修改或更新。注意，在我们的框架中未使用 VAE 解码器。

视觉规划超参数：在视觉规划中，我们根据当前观测和任务提示预测未来 8 帧。我们遵循 [5] 执行 DDIM [38] 采样进行 25 步来合成视觉计划，其中在 MetaWorld 实验中，文字条件引导尺度设置为 2.5，在 Panda Arm Pushing 中设置为 7.0。对于逆概率适配，我们使用 0.5 作为先验强度。

控制环的选择：视觉规划为用户提供了在执行质量和速度之间进行控制的能力。在我们的实验中，每个视觉计划由 9 帧组成，包括一个当前观察帧和八个未来帧，并可以被转换为 8 个动作。通过执行开环控制，我们从一个视觉计划中顺序执行所有 8 个动作，而无需进行任何重新规划。虽然合成一个视觉计划通常涉及多个采样步骤，因此可能耗时，但开环控制极大地提高了交互效率。然而，由于开环控制并不根据环境的反馈调整控制动作，计划中的后续动作可能对最新状态来说变得次优并导致误差累积。为缓解此问题，闭环控制会在每个交互步骤调整动作。具体而言，我们只执行计划中的第一个动作，并基于从环境接收到的新观察进行重新规划。尽管这种控制方式使我们能够最可靠地进行交互，但由于频繁的重新规划会带来较大的计算开销。为了平衡执行质量和效率，我们可以灵活地选择一个介于开环和闭环两种极端之间的控制环。例如，我们在重新规划之前执行计划的一半（例如 4 个动作），我们称之为半开环控制。

为了实现最佳的执行速度，我们在 Panda Arm Pushing 和 Drawer Opening 任务中采用开环控制。在这些任务中，我们发现视觉计划可以很好地执行，在实际执行中偏差极小。对于所有 MetaWorld 实验，我们使用半开环控制来平衡性能和效率。

B 图 2 的 MetaWorld 任务性能分解

	In-Domain Only			SAIL (IPA)		
	Iter. 0	Iter. 1	Iter. 2	Iter. 0	Iter. 1	Iter. 2
Door-Close*	71.1 ± 15.8	87.8 ± 5.1	90.0 ± 6.7	64.4 ± 3.8	90.0 ± 3.3	92.2 ± 1.9
Drawer-Close	6.7 ± 3.3	11.1 ± 5.1	13.3 ± 3.3	27.8 ± 7.7	43.3 ± 14.5	37.8 ± 9.6
Drawer-Open	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
Window-Close	64.4 ± 6.9	68.9 ± 5.1	58.9 ± 1.9	52.2 ± 13.9	67.8 ± 6.9	73.3 ± 5.8
Window-Open	3.3 ± 3.3	1.1 ± 1.9	1.1 ± 1.9	1.1 ± 1.9	2.2 ± 1.9	3.3 ± 0.0
Button-Press	0.0 ± 0.0	0.0 ± 0.0	1.1 ± 1.9	1.1 ± 1.9	0.0 ± 0.0	0.0 ± 0.0
Average	24.3	28.1	27.4	24.4	33.9	34.4

Table 一个 5：MetaWorld 任务表现。我们提供了图 2 中最左边图的任务表现的详细列表。我们报告了在 6 个任务中的平均成功率，每个任务聚合了 3 个种子。在设置中，以改善行为为特征的部分用阴影背景强调。与仅在域内的基线相比，SAIL (IPA) 使得平均任务表现在每次迭代中持续改善，并在第 2 次迭代中取得了最佳的总体成功率。

C 完整 MetaWorld 次优结果

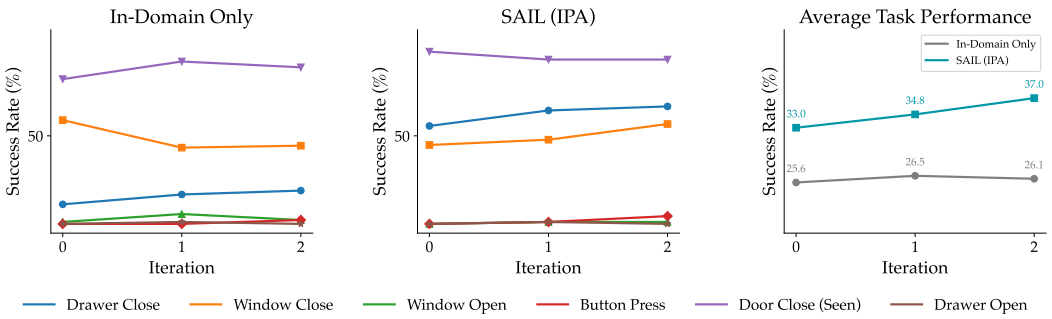


Figure 一个 1：SAIL 的结果与次优的域内数据，没有经验过滤（6 个任务）。

我们在 6 个 MetaWorld 任务上评估 SAIL，其中 5 个是未见过的任务。在第 4.4 节中，报告了 4 个新颖的 MetaWorld 任务的有意义结果，并在图 5 中进行了突出显示。我们在图 一个 1 中展示了完整的结果，此前由于改进效果不明显而忽略了门关闭和窗户打开任务的结果。此外，我们在表 一个 6 中报告了仅在域内和 SAIL 的详细任务表现，结果是汇集了 3 个种子的结果。

虽然在没有使用适应时，迭代的改善趋势不如预期一致（如图 一个 1 和表 一个 6 最左边的图示所示），但 SAIL (IPA) 在四个未见过的任务上表现出持续改善的行为，并在第 2 次迭代时达到最高的平均任务表现，这突显了自我适应的有效性。

关键的是，与抽屉打开的情况类似，我们发现当只能收集到少量成功轨迹时，性能的提升是困难的。在这种情况下，由于没有应用过滤，模型将继续在大多数次优轨迹上增强自身，就像仅在域内的情况一样，因此几乎无法观察到有意义的性能提升。这与窗口打开的发现类似，尽管有所增加，但增幅不大，很可能是由于缺乏成功的收集示范来在每次迭代中利用。

然而，我们发现，平均而言，如图 一个 1 最右侧的图所示，与仅在域内的情况相比，即使初始数据不理想，SAIL 在不同任务上的表现随着迭代次数的增加显著提高。

	In-Domain Only			SAIL (IPA)			SAIL (PA)		
	Iter. 0	Iter. 1	Iter. 2	Iter. 0	Iter. 1	Iter. 2	Iter. 0	Iter. 1	Iter. 2
Door-Close*	82.2 ± 10.2	92.2 ± 3.8	88.9 ± 1.9	97.8 ± 3.8	93.3 ± 0.0	93.3 ± 3.3	85.6 ± 5.1	90.0 ± 3.3	96.7 ± 5.8
Drawer-Close	11.1 ± 3.8	16.7 ± 3.3	18.9 ± 10.2	55.6 ± 6.9	64.4 ± 6.9	66.7 ± 10.0	32.2 ± 1.9	46.7 ± 8.8	53.3 ± 3.3
Drawer-Open	0.0 ± 0.0	1.1 ± 1.9	0.0 ± 0.0	0.0 ± 0.0	1.1 ± 1.9	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	1.1 ± 1.9
Window-Close	58.9 ± 11.7	43.3 ± 8.8	44.4 ± 9.6	44.4 ± 6.9	47.8 ± 10.2	56.7 ± 11.5	76.7 ± 11.5	70.0 ± 5.8	61.1 ± 5.1
Window-Open	1.1 ± 1.9	5.6 ± 1.9	2.2 ± 3.8	0.0 ± 0.0	1.1 ± 1.9	1.1 ± 1.9	1.1 ± 1.9	1.1 ± 1.9	0.0 ± 0.0
Button-Press	0.0 ± 0.0	0.0 ± 0.0	2.2 ± 1.9	0.0 ± 0.0	1.1 ± 1.9	4.4 ± 1.9	0.0 ± 0.0	0.0 ± 0.0	1.1 ± 1.9
Average	25.6	26.5	26.1	33.0	34.8	37.0	32.6	34.6	35.6

Table 一个 6：具有次优初始数据的详细任务表现。我们比较了仅在域内和 SAIL (IPA) 以及附加 SAIL (PA) 设置中的视觉规划表现。我们报告了 6 个任务中的平均成功率，每个任务聚合了 3 个种子。在背景中用阴影突出显示了行为改善的设置。

C.1 概率适应

根据之前的研究 [5] 提出的 IPA，是建立在概率适应（PA） [27] 之上的。与公式 1 相比，PA 采用以下采样形式：

$$\tilde{\epsilon} = \epsilon_\theta(\tau_t, t) + \alpha\Big(\epsilon_\theta(\tau_t, t \mid \text{text}) + \gamma\epsilon_{\text{general}}(\tau_t, t \mid \text{text}) - \epsilon_\theta(\tau_t, t)\Big) \tag{2}$$

其中通用文本到视频模型作为概率知识先验，在采样过程中引导小领域模型的生成过程。一个自然的问题是，与其他基于得分组合的适应方法相比，IPA 是否是促进自我改进行为的最佳适应技术。因此，我们评估 SAIL，使用 PA 作为替代适应策略，并将其与不使用适应（仅限领域内）和 IPA 进行比较。

如表 一个 6 的最右列所示，概率适应在多个任务和平均任务表现上的改进行为相似。具体而言，在未见过的 6 个任务中，有 3 个通过 SAIL（PA）不断改进，而其逆向使得在迭代中 4 个未见过的任务得到改进。此外，SAIL（IPA）在平均任务表现上更高，并在最后一次迭代中达到了最佳成功率。总体而言，我们认为 IPA 是一种更为稳健的适应技术，尤其是在域内初始化不佳的情况下，可以通过视觉规划收集到更高性能的轨迹，并随后通过 SAIL 促进行业内视频模型的改进。

我们展示了 SAIL 在多个环境和任务中的额外视觉计划，以及它们的执行结果。

下图展示了具有经验过滤的 SAIL 的

C.2 带有经验过滤的 SAIL

视觉计划及其执行过程。

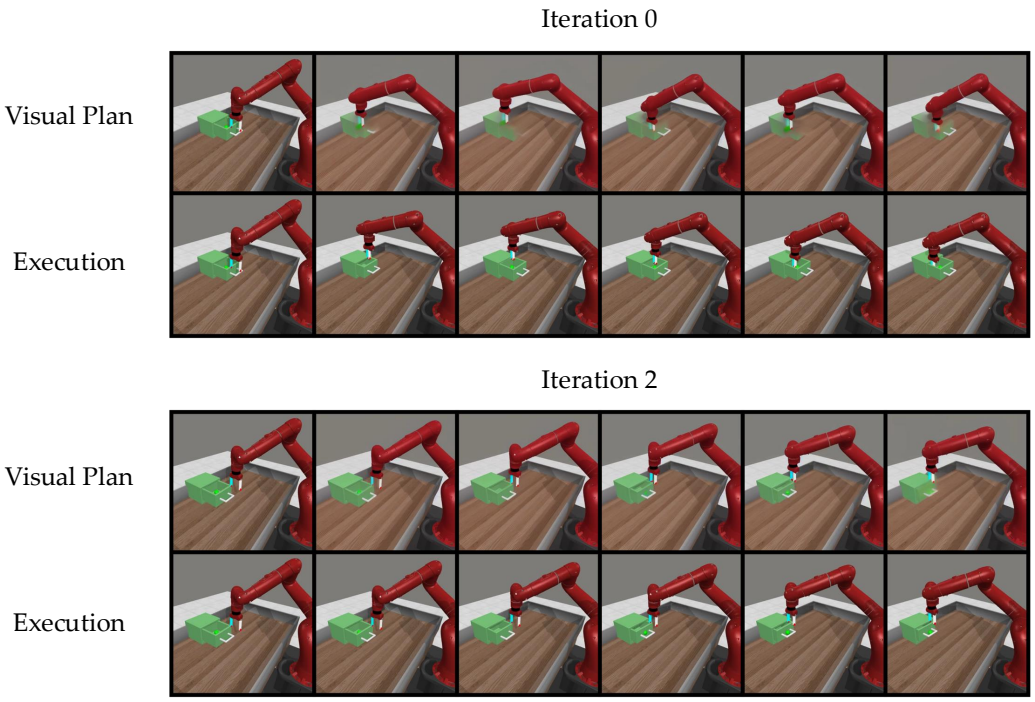


Figure 一个 2：将 SAIL 应用于抽屉关闭，同时进行经验筛选。

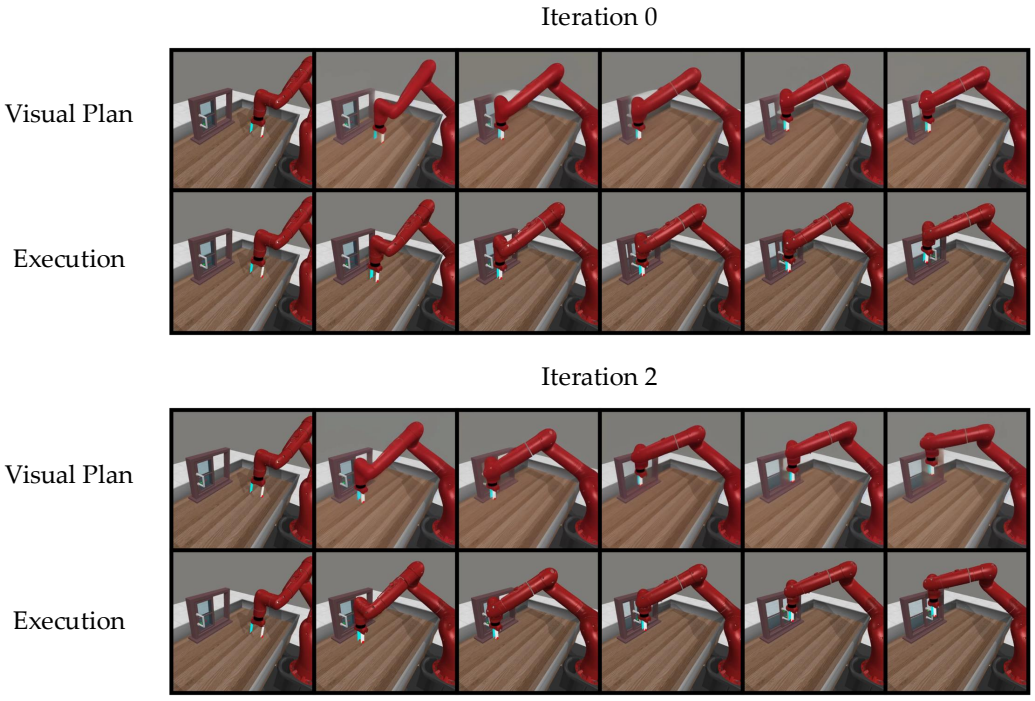


Figure 一个 3：SAIL 在窗口关闭时进行经验过滤。

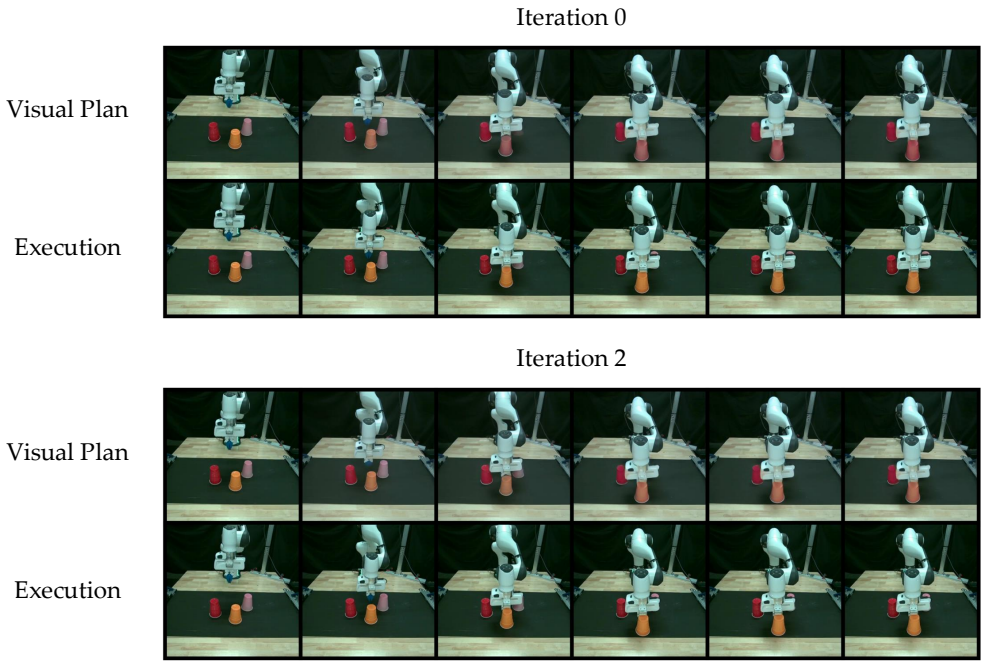


Figure 一个 4：SAIL 在橙色杯推动（红/粉/橙）与经验过滤。

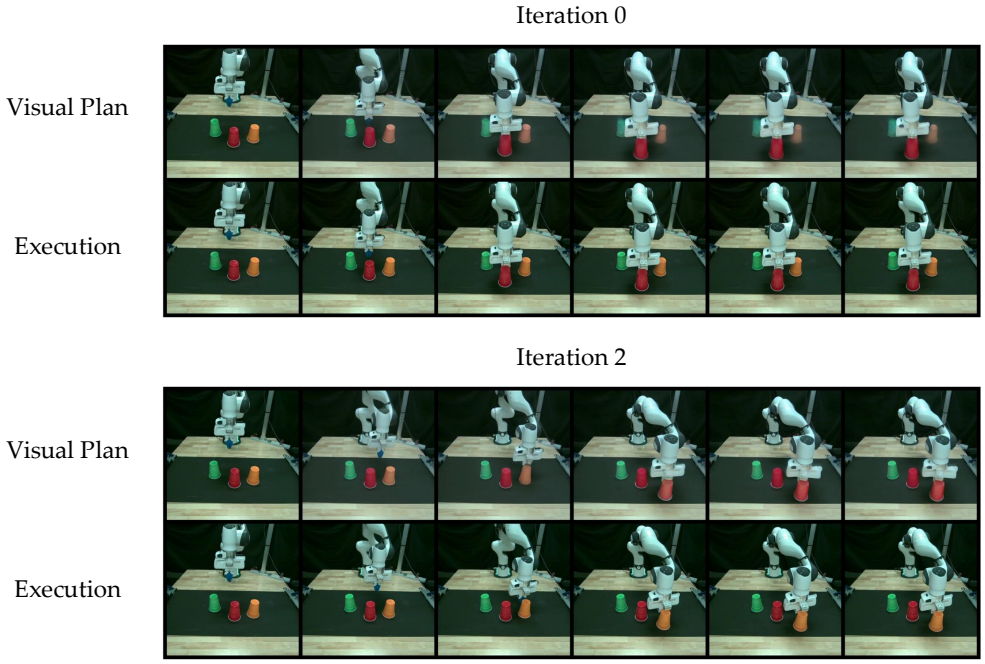


Figure 一个 5：SAIL 在桔色杯子推动（红色/绿色/桔色）中的经验过滤。

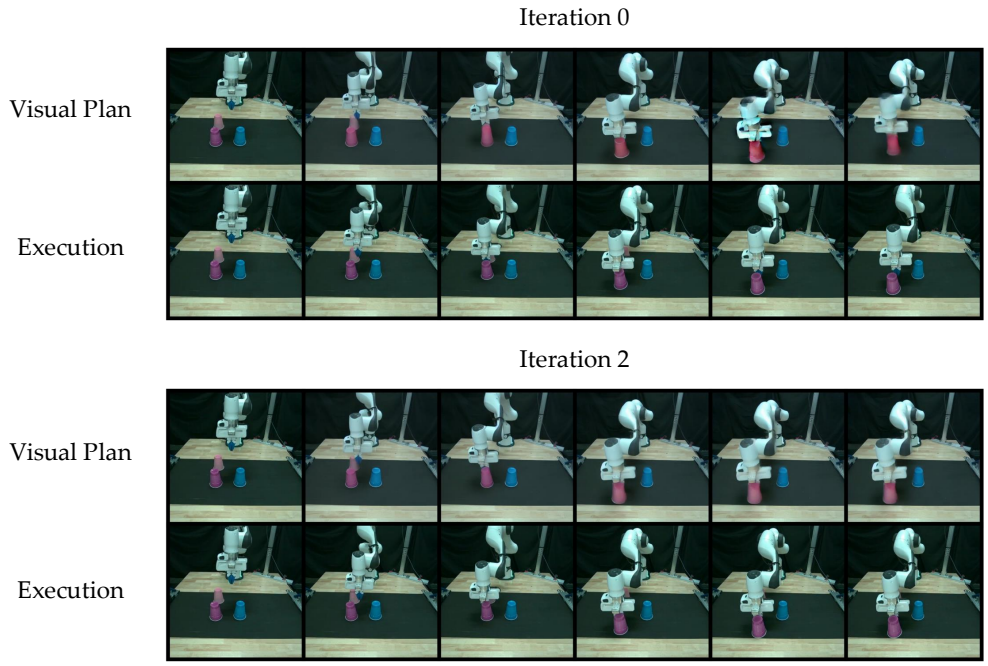


Figure 一个 6：SAIL 在紫杯推送（蓝/粉/紫）中带有经验过滤。

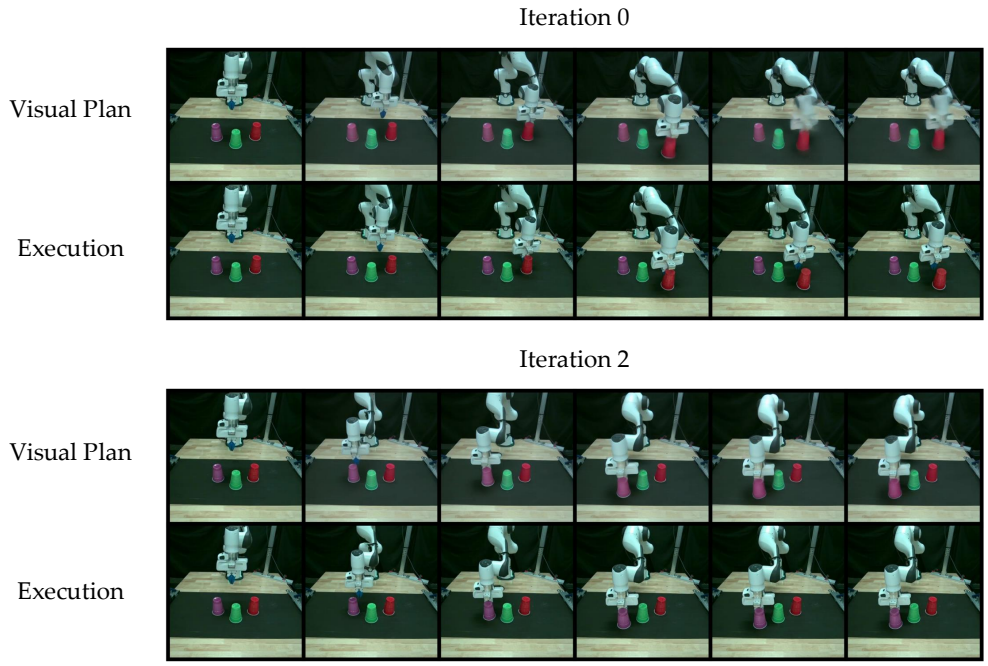


Figure 一个 7：在紫色杯子的推动任务（红色/绿色/紫色）中应用经验过滤的 SAIL。

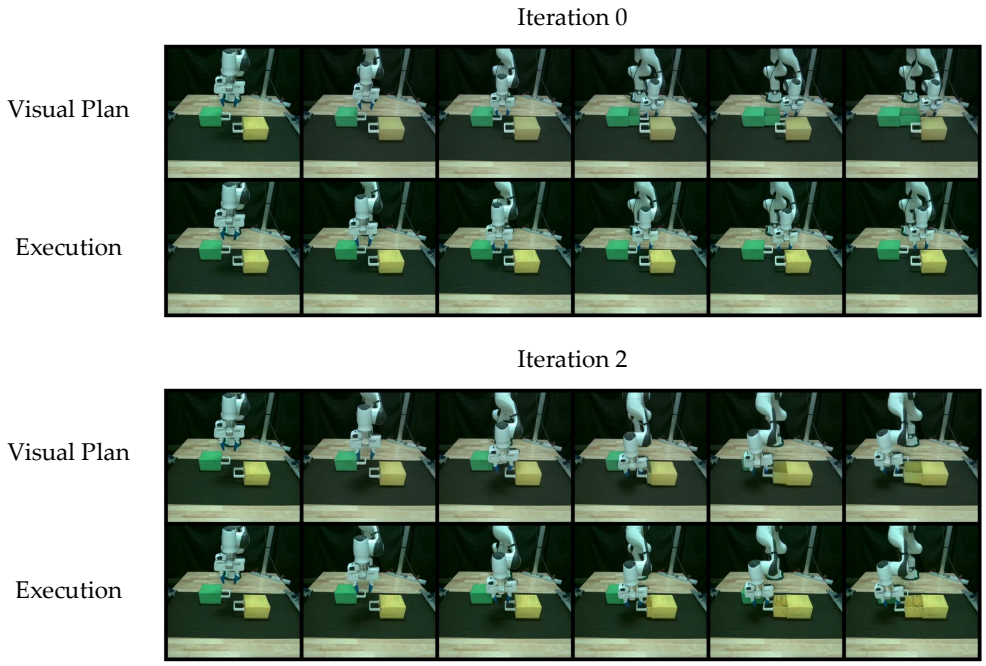


Figure 一个 8：在黄色抽屉打开上航行（黄色/绿色），具有经验过滤。

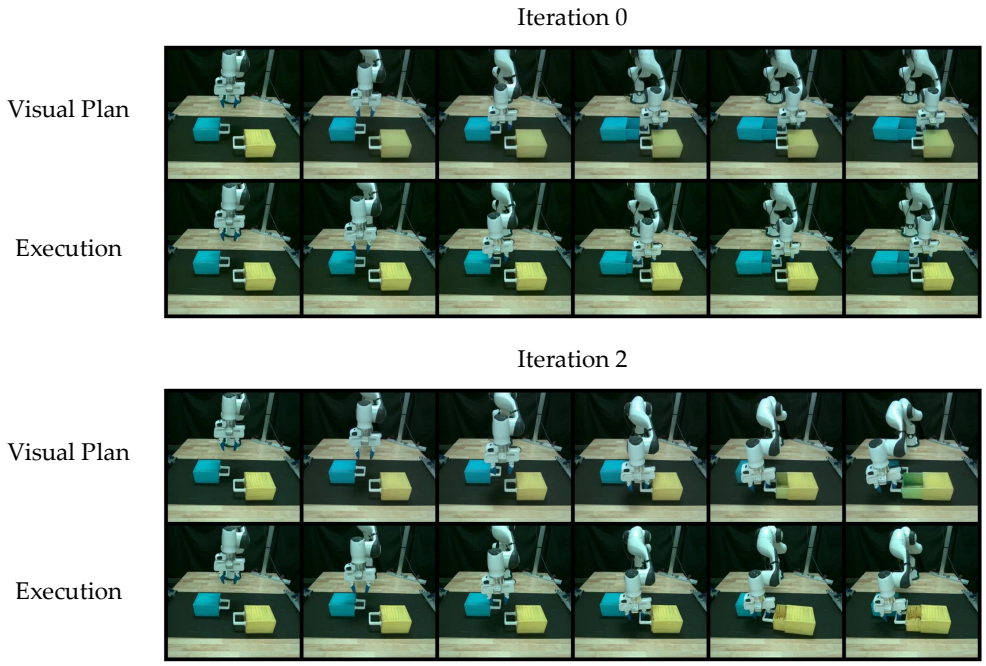


Figure 一个 9：在有经验筛选的情况下，黄色抽屉开启上的 SAIL（黄色/蓝色）。

未经过经验过滤的 SAIL 的视觉计划及其执行如下所示。

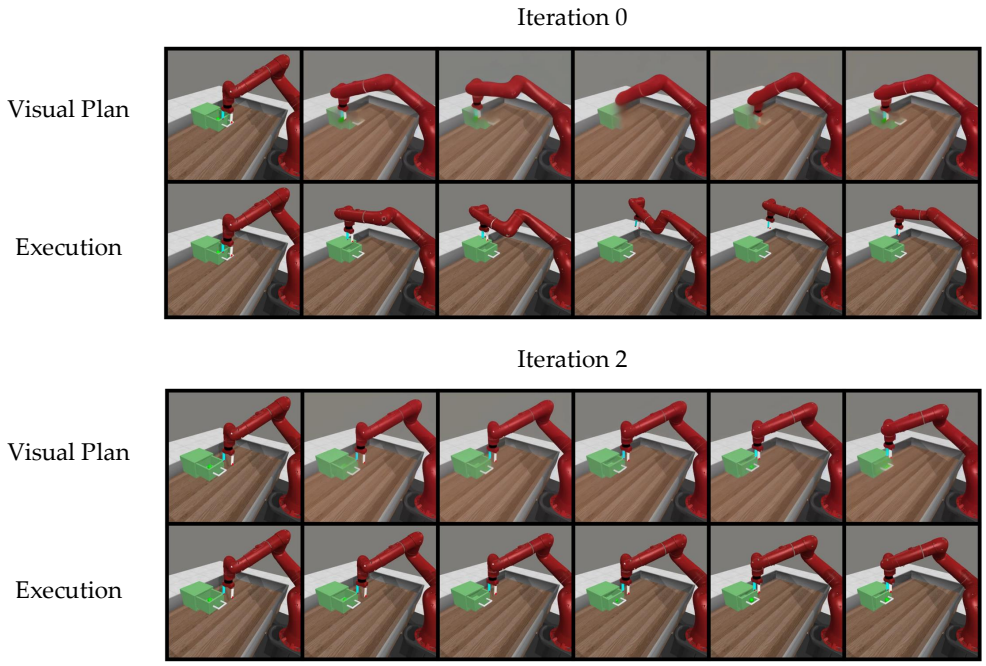


Figure 一个 10：关闭抽屉时的 SAIL，无需经验过滤。

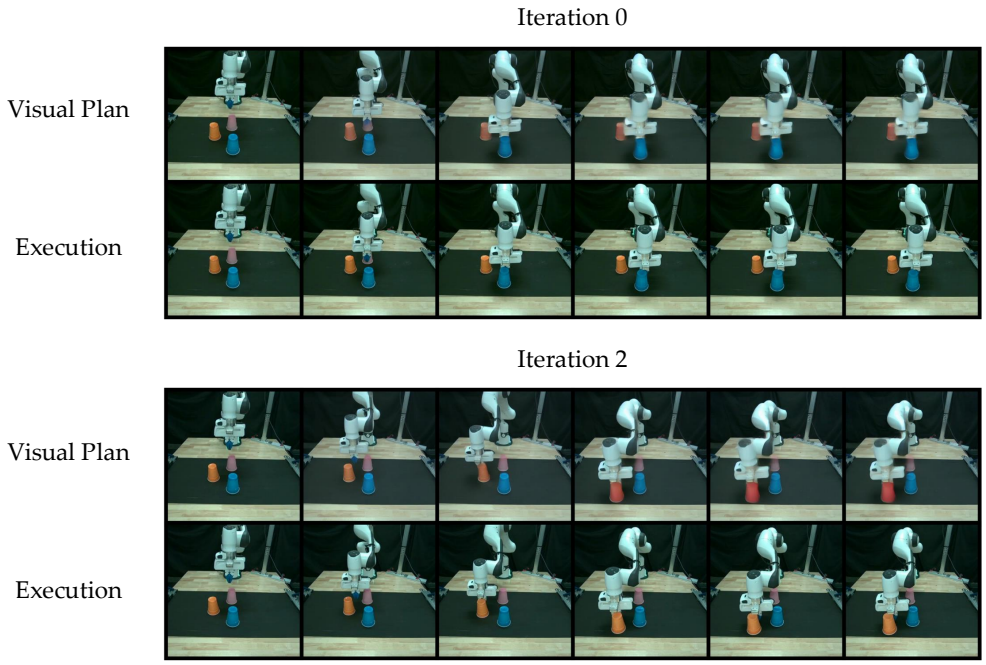


Figure 一个 11：SAIL 在橙杯推动（蓝/粉/橙）中不进行经验过滤。

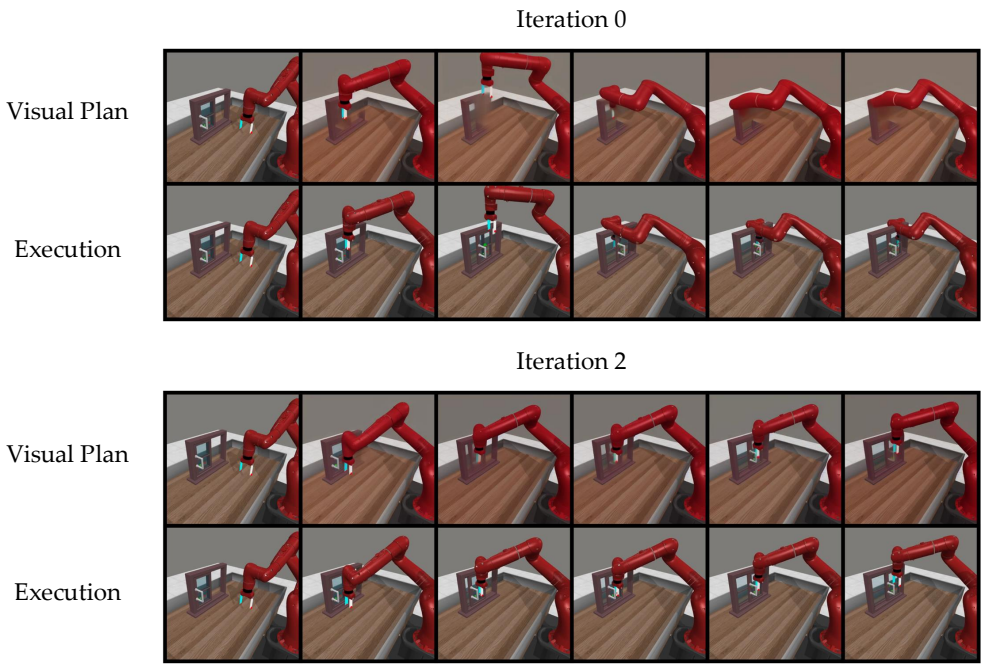


Figure 一个 12：在不进行经验过滤的情况下关闭窗口时的 SAIL（带有次优数据）。