跨领域智能体工作流生成的组合学习

Jialiang Wang[1], Shengxiang Xu[3], Hanmo Liu[12], Jiachuan Wang[4], Yuyu Luo[2], Shimin Di[3], Min-Ling Zhang[3], Lei Chen[12]

jwangic@connect.ust.hk,xushx@seu.edu.cn,hliubm@connect.ust.hk,wangjc@slis.tsukuba.ac.jp,yuyuluo@hkust-gz.edu.cn,shimin.di@seu.edu.cn,zhangml@seu.edu.cn,leichen@cse.ust.hk[1]香港科技大学，香港特别行政区，中国[2]香港科技大学（广州），广州，中国[3]东南大学，南京，中国[4]筑波大学，筑波，日本

arXiv:2602.11114v1 [cs.MA] 11 Feb 2026

摘要

自动生成智能体工作流——即用于编排推理、验证和修复的可执行算子图或代码——已成为解决复杂任务的一种实用方法，能够应对单次大语言模型生成难以可靠处理的任务。然而，何为优质工作流，很大程度上取决于任务分布和可用算子。在领域偏移情况下，当前系统通常依赖迭代式工作流优化，从庞大的工作流空间中寻找可行工作流，这会带来高昂的迭代成本，并导致不稳定且特定于领域的行为。为此，我们将分解-重组-决策 机制内化到开源大语言模型中，以实现跨域工作流生成。为实现分解，我们学习一组紧凑的可复用工作流能力，覆盖多个不同领域。为实现重组，我们将每个输入任务映射为这些基组的稀疏组合，从而通过单次生成得到任务特定工作流。为实现决策，我们将工作流生成的成功或失败归因于所学能力的反事实贡献，从而通过边际效应识别实际推动成功的能力。在严格的多领域、跨领域和未见领域评估中，我们的单次生成器超越了需要消耗20次迭代的最先进优化基线，同时大幅降低了生成延迟和成本。

关键词

智能体工作流、大语言模型、能力学习、组合泛化

1 引言

大语言模型（LLM）在开放域问答和代码生成方面展现出强大的零样本能力 [5, 7, 28]。然而，这些能力最初是通过单次生成实现的：模型一次性生成最终答案或程序。对于复杂任务，单次生成往往会遇到结构性瓶颈 [9, 17]。解决方案不仅要正确，还必须满足约束、能够调用外部工具、支持纠错，并在严格的延迟和成本预算下保持可靠。

为了突破这一上限，智能体工作流 [23, 45, 52]在思维链 [41]之后已成为一种切实可行的方法。通过使求解过程更加显式，智能体工作流将任务分解为由算子构成的有序可执行组合，并在控制结构下执行它们。具体而言，复杂任务的工作流生成需要决定调用哪些算子，以及如何将它们组合成一种拓扑结构（例如序列、分支图），该拓扑结构决定中间状态如何产生与消费 [46]。例如，开放域推理受益于检索证据并比较多个方面的算子 [38]；数学问题需要验证算子与反例检查算子 [10]；代码任务则要求测试与修复循环的拓扑结构 [24]。

近年来，基于多智能体工作流构建智能体系统取得了快速进展。开创性工作 [9, 17] 采用手动设计的算子流水线或协作结构——这些方法在特定设置下有效，但通常不会随输入任务而变化，因此限制了适应性和通用性。鉴于手动设计成本高且脆弱，近期研究转向自动化工作流生成，旨在减少人工工程投入，并根据任务定制工作流。例如，AFlow [47] 等系统及相关自动化框架 [18, 39, 43]致力于以最少的手动指定来生成或改进工作流，从而提升其在不同领域中的可扩展性。

这些自动化系统背后的常见策略，是将工作流生成置于推理循环中（如图1(a)所示）。给定一项任务，模型首先生成一个或多个候选工作流，然后通过搜索或迭代优化对其进行采样和改进——采用最佳N项采样 [20], 自我反思重写 [39], 启发式结构编辑 [47], 或进化过程 [27]，反复选择、变异并重新评估工作流。总体而言，这种工作流优化范式将工作流生成视为在庞大的工作流空间中进行的试错式推理，以较高的迭代成本换取有效性和通用性。

然而，工作流能够提升性能上限，并不意味着大语言模型能够自然而然地为不同领域生成有效的任务特定工作流。事实上，过去的范式更像是在推理时附加一个外部优化器，而非赋予模型可迁移的工作流生成能力。图1(a)揭示了应对领域偏移时的两项局限。首先，工作流生成通常由仅依赖大语言模型的启发式方法 [18] 或随机优化 [47]驱动。这种有限的稳定性和可控性，无法保证任务分布发生偏移时工作流的质量。其次，难以在不同领域间统一良好的工作流标准和策略。在一个领域奏效的工作流标准和启发式方法，在另一个领域可能失效，甚至适得其反 [36, 48], , 导致明显的泛化方差。因此，工作流生成往往是在推理时针对单个任务反复试错，以找到可行的工作流，而不是

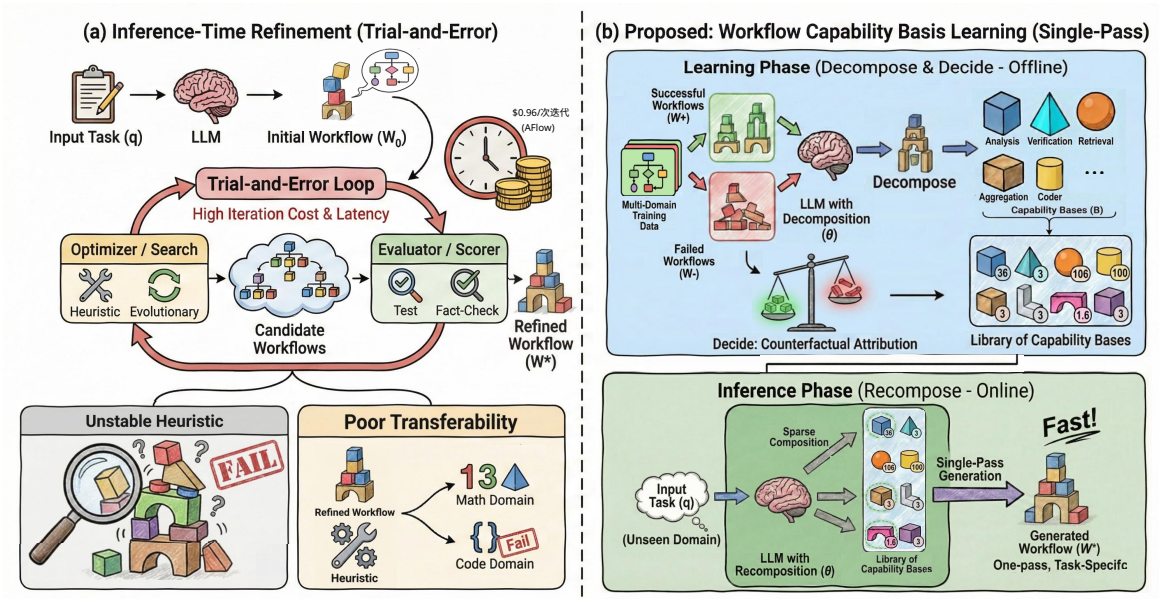


图 1：两种工作流生成范式。左：推理时优化依赖于在庞大的工作流空间中进行试错。右：CapFlow 将"分解-重组-决策"内化到大语言模型中，从而实现跨领域的单次生成。

学习从任务语义到工作流结构决策的可迁移映射。我们进一步通过两个相互关联的差距来刻画这一机制缺失：（1）能力分解差距：大语言模型通常在内容层面表示任务，却缺少一种能够直接揭示所需工作流相关能力的表示；（2）能力重组差距：即使已知有用的工作流模式，模型仍缺乏一种可控的方式，为新任务选择并组合恰当的能力。

值得注意的是，这些差距并不意味着工作流生成完全是特定领域的。我们在第 4.1 节中的研究表明，尽管不同领域的任务表面形式有所差异，许多成功的工作流反复运用了相似的底层能力因子（例如多方面分析、验证/修复和聚合）。如图 1(b) 所示，如果模型能够以参数化、可复用的形式表征这些能力因子，并根据新任务的需求进行组合，跨域工作流生成便可从试错式推理转向单次结构决策。

为弥合分解与重新组合之间的差距，我们将"分解-重组-决策"机制内化到开源大语言模型中，使工作流生成不再依赖启发式的试错推理。在分解阶段，我们从多个领域中学习一组紧凑的可复用能力基底，以捕捉能在潜在空间中泛化的重复性工作流因素。在重新组合阶段，我们将输入任务映射为这些基组上的稀疏组合，从而以可控的方式复用能力来处理新任务。在决策阶段，我们将工作流生成的成功或失败归因于所学能力的反事实贡献，通过这些能力在领域迁移下的边际效应，识别出真正有助于工作流生成的能力基。最终模型通过单次生成，即可生成可执行的特定任务智能体工作流，从而避免推理过程中成本高昂的优化。我们的贡献如下：

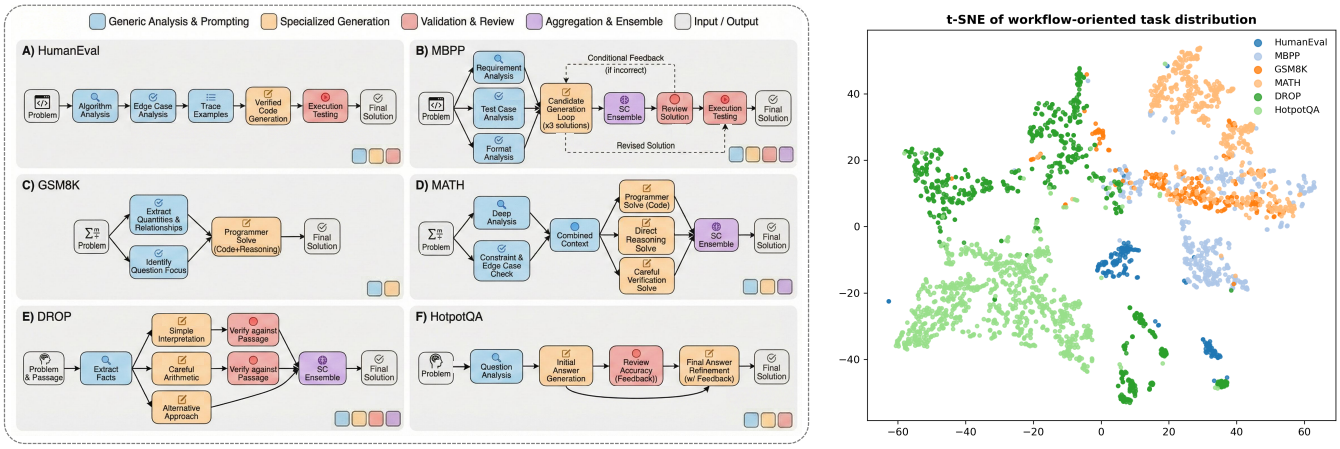
- 我们提出了工作流能力基学习，将跨领域工作流生成从试错式推理重新定义为一种可学习的能力分解与重组问题，从而实现单次处理的任务专用工作流生成。
- 我们提出了工作流泛化框架 CapFlow，该框架以共享能力基和任务条件组合为核心，在不替换底层基座模型的前提下，实现结构化决策与组合式迁移。
- 我们整理多领域的成功与失败工作流数据，以构建反事实贡献归因，并将其与偏好驱动监督相结合，使基学习与推动成功的因素保持一致，从而提升泛化能力。
- 在多领域、跨领域和未见领域设置下的实证结果表明，我们的单次生成方法能够超越需要进行 20 次迭代的推理时优化基线，大幅降低工作流生成成本。

2 相关工作

2.1 自动化智能体工作流

智能体工作流通过明确并使求解流程可执行，突破了大语言模型的单次答案生成模式 [9, 17] q 。典型的工作流将用户查询/任务映射为由算子（例如多方面分析、验证、聚合）构成的可执行组合，并采用特定的拓扑结构（例如序列、分支图或修复循环），通常还会保留中间状态并进行工具调用。实践中，工作流会以计划 [30], 图 [48], 神经网络 [22], 或程序 [18]的形式实例化。然而，早期方法通常依赖于以下两者之一：（i）手动整理的工作流模板库 [9, 17]，或（ii）在不同任务间复用的通用规划策略 [24, 38, 41]。随着算子选择、拓扑结构和提示策略的组合空间呈指数级增长，这些方法很快便面临可扩展性限制，因此催生了自动化工作流生成的两条研究路线。

自动化工作流优化。一种常见范式是在工作流空间 W 上构建外部优化器，并将



(a) 我们在编程、数学和推理领域中最成功的 6 个智能体工作流。(b) 面向工作流的任务分布（t-SNE）。图 2：跨领域智能体工作流分析：（左：结构分析）各领域成功率最高的工作流；（右：潜在分析）根据学习到的工作流能力嵌入任务后的 t-SNE 可视化。

工作流生成视为推理时优化问题：

$$W^* = \arg\max_{W \in \mathcal{W}} G(W; q) \tag{1}$$

在这一范式中，模型会采样候选工作流W，然后通过搜索或迭代优化来改进这些工作流，以最大化给定任务 q的评估函数 G。例如，AFlow [47] 会对以代码表示的工作流进行搜索，并结合大语言模型参与的优化和蒙特卡洛树搜索；GPTSwarm [52] 则通过编辑节点/边来优化图结构多智能体拓扑。ADAS [18] 和 MASS [51] 也主张将智能体系统构建视为对模块化构建块及其组合进行搜索。尽管性能出色，这种试错式推理成本高昂且依赖特定领域：**(i)** 每次迭代都会产生高昂且难以预估的工作流执行成本，且收益递减 [47]；**(ii)** 工作流生成往往由仅依赖大语言模型的启发式方法或随机编辑驱动，导致可控性有限、性能表现不稳定 [43]；**(iii)** 有效的工作流标准和编辑启发式方法难以迁移到不同领域，导致在领域偏移下出现显著的泛化方差 [36, 48]。

学习生成工作流。除了启发式优化之外，还有一些研究通过优化工作流生成器 $p_\theta(W \mid q)$，从数据 D 中学习智能体系统的决策策略，并将工作流视为轨迹或代码，其质量由任务成功与否来评判：

$$\theta^* = \arg\max_{\theta} \mathbb{E}_{q\sim\mathcal{D}}\, \mathbb{E}_{W\sim p_\theta(\cdot|q)}[\, G(W; q)\,] \tag{2}$$

这种基于学习的范式涵盖监督微调或行为克隆，以模仿高质量工作流 [34], 基于偏好的学习，利用候选工作流之间的相对判断 [32], 以及在有反馈时采用的强化式优化。例如，ScoreFlow [39] 利用直接偏好优化来利用定量反馈。FlowReasoner [15] 通过强化学习训练查询级元智能体，以生成个性化的多智能体系统。MAS-GPT [46]通过面向一致性的数据构建流程构造查询-工作流对，以训练工作流生成。这种基于学习的视角为手工设计的启发式方法提供了一种有原则的替代方案

通过将工作流生成建立在数据之上。然而，跨领域迁移仍存在两个明显的局限：（i）在任务分布偏移下，简单的模仿可能很脆弱 [13], , 因为它无法分离出哪些工作流因素在因果层面推动了成功；（ii）学习到的行为往往纠缠在整体式参数中，缺乏支持组合式迁移的可控机制 [29]，而新任务可能要求以新颖方式重新组合熟悉的因素。我们的工作与这种基于学习的范式一致，同时明确致力于在领域偏移下实现可迁移工作流生成的可控重组。

2.2 参数高效与模块化适配

与以工作流为中心的系统并行，参数高效的大语言模型适配也取得了显著进展，其中包括低秩参数化 [34]，可在无需全面微调的情况下实现专门化。与此相关，模块化和条件计算方法会将输入路由至一部分参数，通常将其表述为专家混合 [6] 或适配器混合 [42, 50], , 并与低维预测子空间 [1, 3]相关联。然而，至关重要的是，先前关于路由和模块化适配的大多数研究都以高效计算、容量扩展或多任务专门化为目标，许多多领域方法则使用显式领域标识符来驱动离散专门化 [11, 49]。与之相反，我们的问题要求学习一种结构化决策机制，使其在领域偏移下与可执行工作流的成功相契合。我们并不将模块化视为分配计算资源的工程手段，而是将其视为实现能力重组的途径：经路由的组件对应可复用的工作流能力因子，而路由决策经过训练，旨在超越单次工作流生成。这一区别是我们工作的核心：我们学习一种“分解-重组-决策”策略，以捕捉可迁移的结构化决策规则，而非在专家之间进行预测融合。

3 预备知识

动机研究。为验证我们关于可复用工作流能力的设想，我们对跨领域的成功工作流进行了结构和潜在分析。我们将最

数据集中各领域最成功的工作流可视化呈现在图2a中，并分析了它们的算子和拓扑结构。我们观察到，不同领域（B、D、E）中反复出现一些工作流模式，例如多方面分析（蓝色节点）和集成聚合（紫色节点）。相比之下，领域特定行为主要源于专用算子，例如面向编码任务的代码测试（A、B中的红色节点）以及面向问答任务的多视角任务求解（E中的橙色节点）。

为进一步考察将任务语义与工作流行为对齐的重要性，我们利用学习到的工作流能力对每个任务进行嵌入，并通过图2b中的 t-SNE 将所得的面向工作流的任务分布可视化。在学习过程中不使用任何领域标识的情况下，来自同一领域的任务自然聚集在一起，这表明：（1）同一领域的任务通常需要相似的工作流能力。与此同时，来自不同领域的任务可能在工作流空间中发生重叠，表明：（2）尽管表层任务存在差异，成功工作流通常共享潜在因素。这些观察结果促使我们对可复用的工作流能力进行显式建模，并将其重新组合以应对新任务。

问题设定与数据。每个开放域任务都包含一个问题文本 q 和一个格式化提示（包括指令和算子描述）。目标输出是一个可执行工作流程序W，其中同时包含：（i）工作流拓扑结构（控制流）；以及（ii）定制化算子提示。由于现有基准测试很少同时提供可执行工作流程序以及成功和失败结果，我们整理了一个多领域数据集，其中包含 180+ 个独特工作流，并记录了来自3个不同领域（编程、数学和推理）的6个数据集上的性能：

$$\mathcal{D} = \{(q_i, \{(W_j, s_{i,j})\}_{j=1}^{N_w})\}_{i=1}^{N_q}, \quad s_{i,j} \in [0,1] \tag{3}$$

其中$s_{i,\ j}$表示该工作流是否能够求解任务。我们将工作流生成表述为条件代码生成：

$$W \sim p_{\Theta(q)}(W \mid x(q)) \tag{4}$$

其中， $x(q)$ 是格式化提示指令， $\Theta(q)$ 表示由我们的能力组合机制诱导出的任务条件化模型参数。

4 方法

现有的试错范式无法产生可迁移的决策规则——因此带来了第 2.1 节所讨论的成本激增、不稳定性和跨领域差异。相反，我们提出一个不同的问题：工作流生成能否转变为一种学习得到的决策机制，在单次生成中将任务语义映射为定制化的工作流结构？其背后有两个关键挑战：(i) 能力分解差距——任务并不会自然地表示在一个与有效工作流因素对齐的空间中；以及 (ii) 能力重组差距 —— 即使存在可复用的模式，模型也缺乏一种可控的方式，在领域偏移下对这些模式进行重新组合。

为弥合这些差距，我们通过以下方式将“分解-重组-决策”机制内化到模型中：(1) 引入一小组基于潜在工作流因素的可复用能力基底；(2) 训练一个任务条件化能力组合器，为每个新任务选择这些基底的稀疏组合。CapFlow 是一种单次生成器，能够生成可执行且针对特定任务的智能体工作流程序，无需在推理时进行迭代优化。

4.1 工作流能力基

我们注意到，许多优秀工作流尽管表面上的任务存在差异，却共享潜在能力因子——我们通过将这些因子参数化为一组紧凑且可复用的能力基来实现这一点。具体而言，我们从冻结权重的大语言模型 f_{θ_0} 出发，将一小组 K 轻量级适配基引入线性变换的选定子集（例如注意力和 MLP 投影）中。每个基都充当一个可复用的潜在工作流因子，能够使模型倾向于选择某些算子、控制流模式和提示词编写行为，而这些行为会反复出现在成功工作流中。

形式化地，考虑一个权重矩阵为 $M \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ 的线性变换。我们通过以下 K 秩-r 基更新对其进行扩充：

$$\Delta B_k \;=\; c_k\, U_k V_k^{\top}, \quad U_k \in \mathbb{R}^{d_{\text{out}} \times r},\ V_k \in \mathbb{R}^{d_{\text{in}} \times r},\ c_k \in \mathbb{R}_+, \tag{5}$$

其中， U_k和 V_k用于参数化能力基，而 c_k是一个可学习的逐基缩放因子。这些参数是模块专属的，但在所有任务之间共享。直观而言，任务特定工作流的生成源于对这些基的选择性组合。

4.2 任务条件化能力组合器

能力基提供了与工作流相契合的表示空间，但我们仍需要一种机制，能够将任务语义映射到该空间，并在领域偏移下选择如何重新组合各个因子。我们将其建模为一个任务条件化能力组合器：

$$\alpha(q) \;=\; g_\psi\big(z(q)\big), \tag{6}$$

其中， $z(q)$ 是任务嵌入，而g_ψ 则输出各基的混合权重。这种设计使组合器依赖于纯粹的任务，而非大语言模型启发式方法中偶然产生的伪影；实验证明，这能稳定训练并减少幻觉。

自适应组合。组合中的一项核心难题是在专门化与复用之间取得平衡。如果组合过早坍缩，少数几个基会占据主导地位，机制也就不再具有组合性；如果组合过于分散，模型的表现就会像一个较弱的集成，并丧失可控性 [21]。为了解决这一问题，我们设计了一种组合器，通过耦合的多层感知机 $h_\psi(\cdot)$ 和 $\Delta t_\psi(\cdot)$ 来预测 (a) 组合逻辑值，以及 (b) 每个任务的温度调整：

$$u(q) = h_\psi(z(q)) \in \mathbb{R}^K, \qquad T(q) = \exp\big(\log T_0 + \Delta t_\psi(z(q))\big), \tag{7}$$

并通过以下方式生成概率

$$\alpha(q) \;=\; \mathrm{Softmax}\!\left(\frac{u(q)}{T(q)}\right) \in \Delta^{K-1}, \tag{8}$$

直观而言， $T(q)$ 控制组合器的决断程度：允许多种工作流能力的任务可以保留较高的熵，而模式明确的任务则可以产生更明确的组合。

4.3 工作流能力分解

结合能力基（第 4.1 节）与任务条件化组合器（第 4.2 节），我们为智能体工作流生成器推导出任务特定参数化。给定任务 q，每个适配线性变换的权重组合如下：

$$M(q) \;=\; M + \sum_{k=1}^{K} \alpha_k(q)\, \Delta B_k. \tag{9}$$

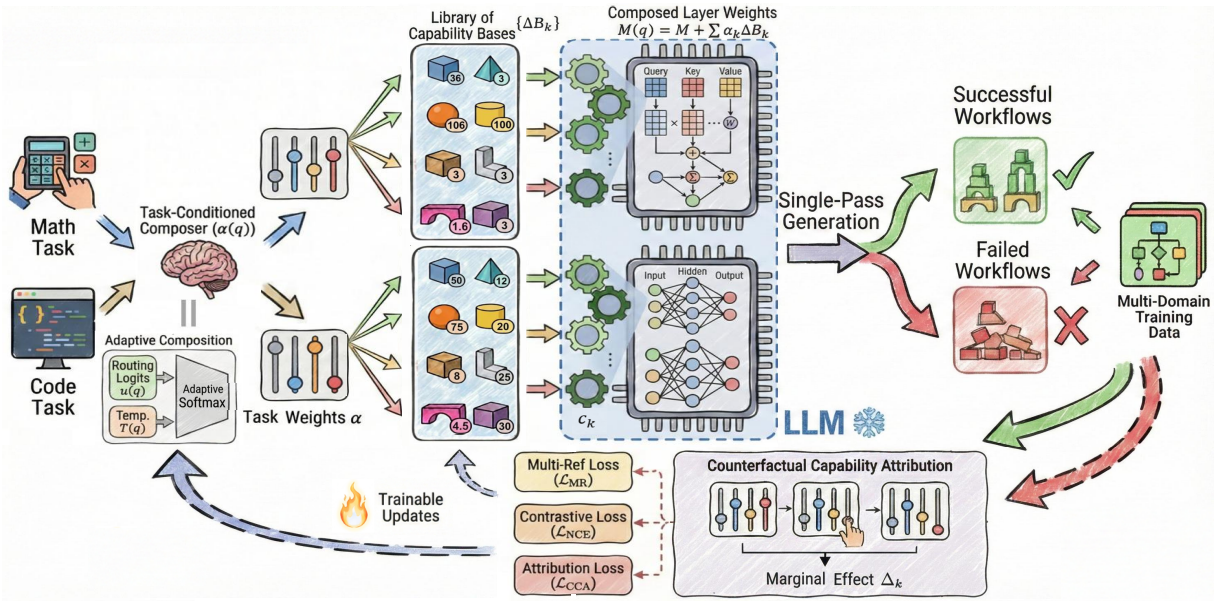


图 3：工作流能力组合（CapFlow）概览：任务条件化组合器将每个查询分解为可复用能力基底的稀疏混合，引导大语言模型生成成功工作流，避免生成失败工作流。

其中，$\alpha(q) \in \Delta^{K-1}$是由任务条件化组合器生成的概率单纯形向量。关键在于，同一个 $\alpha(q)$ 会广播至所有适配层，因此，全局任务级能力决策能够连贯地塑造整个工作流生成器（算子和拓扑结构），而不是产生逐层且不协调的门控。为鼓励重新组合行为并保持推理效率，我们对式（9）采用稀疏实例化方式：每个任务仅激活 top-m个基组，并对其权重重新归一化，从而提供一种受控的归纳偏置，使工作流生成倾向于使用少数能力因子（与图2a中的经验模式一致）。总体而言，这一机制弥补了能力分解差距：我们不再要求模型针对每个任务都从头开始、高成本地细化整个工作流结构，而是将任务表示在一个空间中，在该空间里，相关能力因子能够被明确寻址并组合。

4.4 学习基本工作流能力

除第 2.1 节所述之外，纯粹通过模仿进行训练并不适合工作流生成，因为（i）每项任务通常有多个正确的工作流；（ii）重要的不是与参考答案在风格上相似，而是工作流能否成功。我们的数据集通过为每项任务提供成功和失败工作流，弥补了这一监督信号的缺失。我们利用这些数据来学习（a）能够提高成功工作流代码似然的能力基，以及（b）其任务分解在因果上 而非相关性上与成功相一致的组合器（第 4.5 节）。

令 W表示工作流程序（代码），$x(q)$表示格式化后的输入。我们采用目标词元 w_t 上的长度归一化因果语言模型对数似然，以避免偏向较短的参考答案：

$$\ell(W, x(q)) = \frac{1}{|W|} \sum_t \log p_{\Theta(q)}(w_t \mid x, w_{<t}), \tag{10}$$

多参考成功似然。 对于任务 q_i，令 P_i 表示成功工作流的集合，N_i 表示失败工作流的集合。由于可能有多个工作流成功，我们不 强迫模型模仿单一参考。相反，我们通过多参考目标，最大化分配给成功集合的总概率质量：

$$\mathcal{L}_{\mathrm{MR}} = -\sum_i \log \sum_{W \in \mathcal{P}_i} \exp\left(\ell(W, x(q_i))\right). \tag{11}$$

该目标鼓励生成器覆盖多样化的成功模式，这对于学习可迁移能力因子至关重要，而不是过拟合于狭窄的模板。

任务内对比式分离。 为明确抑制相同 任务的已知失败模式，我们加入组级对比损失：

$$\mathcal{L}_{\mathrm{NCE}} = -\sum_i \frac{1}{|\mathcal{P}_i|} \sum_{W^+ \in \mathcal{P}_i} \log \frac{\exp(\ell(W^+, x(q_i))/\tau)}{\sum_{W \in \mathcal{P}_i \cup \mathcal{N}_i} \exp(\ell(W, x(q_i))/\tau)}, \tag{12}$$

其中 τ 表示温度。不同于作用于孤立的（提示词、选定项、拒绝项）比较的成对偏好目标，公式（12）执行任务局部的、列表式判别。在相同语义和约束下，它将多个成功工作流与任务匹配的失败工作流进行对比，从而产生稳定的列表式训练信号。

4.5 反事实能力归因与可控重组

上述目标训练生成器来建模成功的工作流，但仅凭这些目标并不能保证组合器学会结构化 且 可迁移的 决策规则。具体而言，组-合过程可能变成一个黑箱 [12]：它可能与 q中的表层特征相关，而没有选择实际促成成功的能力因子。为了解决这一问题，我们引入一种反事实归因机制，根据边际效应将哪些能力基起作用转化为可训练的信用分配信号。**反事实能力归因。** 对于样本 $(q,\ W,\ s)$，令 $\boldsymbol{\alpha} = \alpha(q)$ 为组合器输出，并令 ℓ^{main} 表示在 $\boldsymbol{\alpha}$下经长度归一化的对数似然。对于每个基 ΔB_k，我们通过屏蔽并重新归一化，构造一个

表1：多领域、跨领域和未见领域设置下各数据集的工作流求解率与可执行性（%）对比。

			推理		编程		Math		科学（未见）		总体	
Type	设置	方法	HotpotQA	DROP	HumanEval	MBPP	GSM8K	MATH	SciBench	GPQA	求解	执行
手动	–	GPT-4o-mini	81.54	81.37	90.07	70.67	90.14	51.64	28.40	34.16	66.00	—
		CoT	81.75	83.25	91.60	70.96	89.95	53.29	24.83	35.73	66.42	—
		CoT-SC	82.07	83.28	92.36	72.72	90.23	53.70	27.27	35.95	67.20	—
		自我优化	81.18	82.50	90.83	69.50	88.53	50.20	27.03	30.25	65.00	—
		SPP	81.62	81.62	92.36	72.72	91.18	52.88	28.22	33.72	66.79	—
优化	单领域	ADAS（20 次迭代）	82.84	82.23	90.83	70.08	92.03	52.67	30.36	36.12	67.14	79.87
		AFlow（20 次迭代）	85.87	85.75	93.89	82.99	93.93	57.40	41.40	42.18	72.93	88.13
学习	单领域	ScoreFlow（5 次迭代）	86.00	86.14	95.41	82.69	94.21	59.25	34.20	38.69	72.07	97.36
	多领域	ScoreFlow（5 次迭代）	85.37	85.16	93.89	81.23	93.45	57.81	34.20	38.69	71.22	95.89
		CapFlow（单次通行）	**88.12**	**87.12**	**96.18**	**83.28**	**94.97**	**59.87**	**41.60**	**42.41**	**74.19**	**98.03**
			Code,数学->推理		**推理 ,数学->代码**		**推理 ,代码->数学**		**全部->科学**			
	跨领域	**CapFlow（1 次遍历）**	86.37	86.25	93.89	82.11	92.51	54.32	**41.60**	**42.41**	72.43	96.87

反事实组合向量：

$$\boldsymbol{\alpha}^{(-k)} \propto \boldsymbol{\alpha} \odot \big(\mathbf{1} - \mathbf{e}_k\big), \tag{13}$$

并计算在 $\boldsymbol{\alpha}$（$-k$）条件下的反事实得分 ℓ（$-k$）。我们将基 ΔB_k 的边际贡献定义为

$$\Delta_k = \ell^{\text{main}} - \ell^{(-k)}. \tag{14}$$

较大的正值 Δ_k 表明，移除基 ΔB_k 会降低生成该工作流的概率，这说明 ΔB_k 编码了与工作流成功相关的能力因子。

用于组合的偏好加权策略梯度。我们使用工作流结果 $s \in [0,1]$ 来分配带符号的监督信号 $\text{sgn}(s) = 2s - 1 \in [-1,1]$。随后，我们通过鼓励在成功工作流中组合边际贡献为正的基，并抑制在失败工作流中的此类组合，来更新组合器。具体而言，对于一组基 K(q)，我们优化：

$$\mathcal{L}_{\text{CCA}} = -\mathbb{E}\Big[\sum_{\Delta B_k \in \mathcal{K}(q)} \text{sgn}(s) \cdot \Delta_k \cdot \log \alpha_k(q)\Big] + \lambda_{\text{dead}}\, \mathcal{L}_{\text{dead}}, \tag{15}$$

其中，L_{dead} 会惩罚将概率质量分配给贡献仍接近于零的基（以防止出现无效或冗余基）。公式（15）实现了一种结构化的反事实信用分配：组合器不再依赖间接的似然梯度，而是经过训练，将概率质量分配给那些移除后会反事实地降低成功率的能力因子。

完整目标。综上，在训练时，我们优化：

$$\min_{\Phi,\psi} \quad \lambda_{\text{MR}}\mathcal{L}_{\text{MR}} + \lambda_{\text{NCE}}\mathcal{L}_{\text{NCE}} + \lambda_{\text{CCA}}\mathcal{L}_{\text{CCA}} + \mathcal{L}_{\text{reg}} \tag{16}$$

其中，Φ 表示所有能力基参数，ψ 表示组合器参数。正则化项 L^{reg} 包含正交性、熵和温度偏差（详见附录 A.1.2 ）。推理时，CapFlow 不执行迭代优化。如图 3 所示，给定一个新任务 q，我们仅计算一次 $\alpha(q)$，激活适配层中的 top-m 个能力基，并按照公式 (4) 在单个生成轮次中解码工作流程序 W。由此得到的可执行工作流程序同时具备控制流拓扑结构和定制化算子提示，直接解决工作流优化中成本激增、不稳定性和跨领域方差的问题。

5 实验

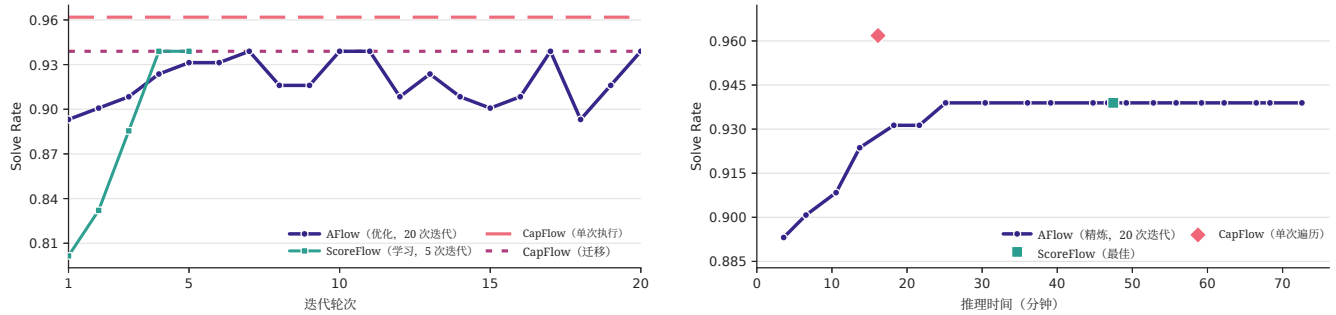
5.1 实验设置

任务与数据集。我们在涵盖 4 个领域的 8 个基准测试上进行评估：推理 [14, 44],编程 [4, 8],数学 [10, 16], 和科学 [33, 37]，并采用成熟的数据划分实践 [46, 47]。科学是用于未见领域 研究的留出领域。对于 3 个训练领域，单领域 设置针对每个数据集分别训练一个模型，然后对其进行测试；多领域设置则使用所有推理、编程和数学数据集训练一个通用模型；而跨领域 设置通过留一数据集法（LODO）评估迁移：对于每个留出数据集，我们在其余数据集上训练，并仅在该留出数据集上测试。

基线方法。我们比较了三类方法。*(1)* 手动提示方法 由人类专家设计，包括直接调用、思维链 [41], CoT 与自一致性 [38], 自我优化 [24], 以及 SPP [40]。*(2)* 工作流优化方法 在推理时迭代生成并改进工作流，最多进行 20 次迭代，包括 ADAS [18] 和 AFlow [47]。一次迭代 是指在验证集上执行/评估候选工作流一次，并应用优化规则生成下一个候选工作流。*(3)* 基于学习的工作流生成器 使用工作流数据进行训练，并直接生成最终工作流。对于 ScoreFlow [39]，我们同时考虑单领域 和多领域 模型设置。

实现细节。我们采用 Llama-3 2 3 [26, . - B-Instruct 作为微35] 3 1 8 [25] 调模型，并使用 Llama- . - B-Instruct 和 Qwen2.5-Coder-3B-Instruct [19, 31] 进行比较。主要结果是在 8 个基组和 Top-m=3设置下报告的，使用 1 块 A100 GPU 训练 35 个 epoch，学习率分别为 2e-4（基组）和 3e-4（合成器）。对于 ADAS 和 AFlow，我们使用 Claude-3.5-sonnet [2] 作为优化器。

评估指标。为确保公平比较，所有生成的工作流均由温度设为 0 的 GPT-4o-mini 执行。所有结果均取平均值



（a）HumanEval 上的迭代轮次与求解率。 (b) HumanEval 上推理时间与求解率的关系。

图4：HumanEval 上的工作流生成权衡。左：强基线方法依赖随机优化，评估成本显著更高，却只能获得相近的提升。右：优化基线方法随着轮数增加而有所改进，但收益递减；相比之下，CapFlow 在单个生成轮次中即可达到较高的求解率。

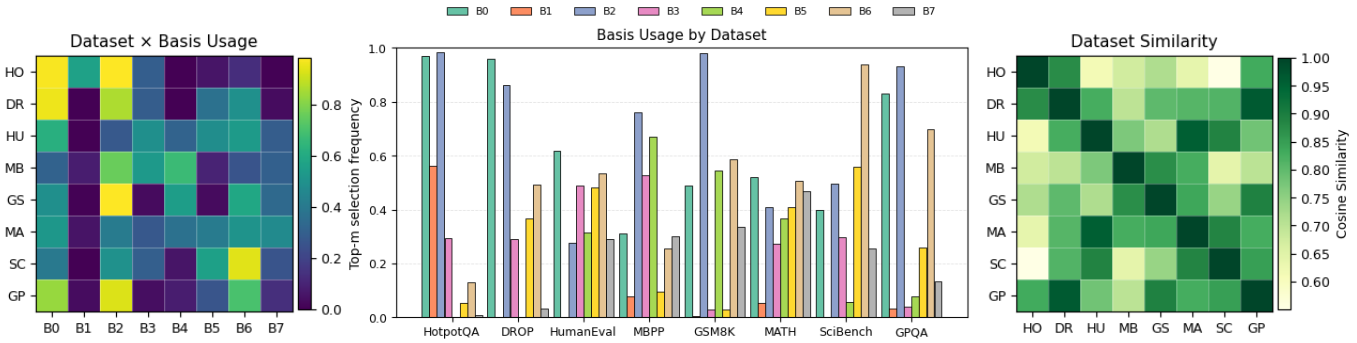


图5：跨领域的能力基使用情况。CapFlow 保持了未坍缩的基选择，并呈现出有意义的跨数据集重叠，支持其预期的“可复用基底”行为。

自 5 次运行。我们报告每个测试数据集上的求解率 (%)（用于求解任务）和可执行率 (%)——工作流执行期间未出现运行时错误——以量化有效性和稳定性。为评估效率，我们还报告求解率随迭代次数和推理时长变化的结果，以刻画其中的权衡关系。

5.2 主要结果

有效性。 表1汇总了多领域、跨领域和未见领域 设置下的测试集结果。总体而言，CapFlow 在各数据集上均取得最高求解率，同时保持较强的可执行性，从而可靠地生成工作流。即使优化基线方法在每个数据集上耗费了 20 次迭代预算，我们的单轮工作流生成仍始终优于这些方法。值得注意的是，在跨领域设置下，CapFlow 在留出数据集上的性能仍可媲美、甚至超过在这些数据集上训练或优化的基线方法。这支持了我们的主张：学习一种可迁移的分解–重组–决策 机制，相较于推理时启发式的试错式优化或训练时纯粹的模仿学习，能够更有效地生成任务特定工作流。

预算–质量权衡。 工作流优化通常是一个对迭代预算敏感的随机过程。图4a绘制了求解率随优化迭代次数变化的曲线。我们观察到，在 AFlow 上，超过适度预算后收益递减：额外的迭代次数仅带来边际性能提升，同时使评估成本至少线性增加。相比之下，CapFlow 无需额外的评估成本即可实现具有竞争力的性能。

推理时成本与延迟。 尽管优化基线以迭代次数作为预算单位，但由于需要反复执行工作流并重新生成内容，迭代会带来显著的推理时开销。大语言模型 API 提供商通常会对这些操作按不同级别收费。除价格外，图4b和表2还报告了成本代理指标，显示与其他数据集相比，CapFlow 显著降低了工作流生成的推理时延迟，同时在不同领域实现了更高的求解率。

5.3 能力使用与归因

我们考察 CapFlow 是否能够学习可复用的能力基和结构化的重新组合能力，以应对领域偏移。具体而言，我们统计 Top-m 组合器在每个数据集上选择各基的频率，并在图5中以条形图和热力图展示结果。

不发生坍缩的可复用基底。 图5展示了清晰的可迁移使用模式：. 组合器会激活多个基，而非坍缩为单一基；同时，有多个基（例如 B0、B2）会在不同数据集间被反复使用。这表明 CapFlow 捕获了可迁移的工作流因素，可在领域偏移下复用，而不是只学习数据集专用适配器。

有意义的跨数据集重叠。 为了解释重叠，我们根据各数据集的基底使用频率计算它们之间的余弦相似度（图5）。相似度矩阵并非均匀分布：

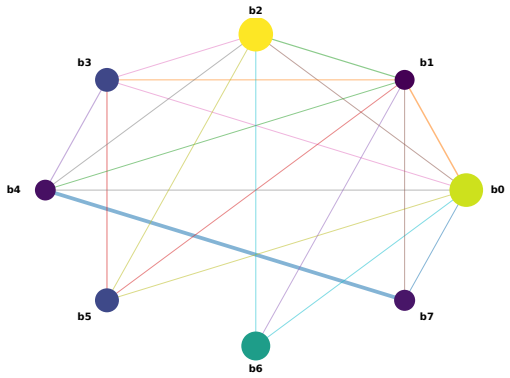


图 6：基组共激活网络。节点表示基组。边表示前 3 个基组选择事件之间的正PMI；概率根据 3706 个任务的经验共激活计数估算，并展示前20对。

表 2：跨领域的消融分解与成本代理指标。Para.：额外可训练参数存储（MB）。训练/轮次和推理（在 HumanEval 上）均为运行时间，单位为分钟。 K 和 Top-m 是 CapFlow 的不同配置。

变体	Para.（MB）	训练/轮次	推理	推理	代码	数学	科学
AFlow	0.00	0.00	73.23	85.81	88.44	75.67	41.79
ScoreFlow	13.02	188.40	47.45	86.07	89.05	76.73	36.45
K=1, LoRA	15.28	19.95	14.17	87.08	87.95	76.41	37.47
K=8, LoRAs	96.44	20.13	17.25	86.74	87.95	76.21	36.65
K=4，顶部-m=3	50.07	28.95	14.93	**88.22**	89.67	75.03	38.15
K=8，顶部-m=3	96.45	34.03	16.13	87.62	**89.73**	**77.42**	**42.00**
K=16，顶部-m=6	189.20	45.75	20.60	87.72	87.72	75.54	37.66
K=32，Top-m=9	374.73	76.13	20.92	87.16	87.91	76.03	37.88

同一领域内的数据集通常更相似（对角线颜色更深），但仍存在显著的跨领域重叠，这与预期的“可复用基底”特性相符，而非彼此隔离的逐领域路由。这一结论与我们在第3节（图2）中的动机研究一致。

结构化重组与归因。为了超越边际使用情况，我们通过图6中的正PMI共激活网络分析协同基底交互。PMI 明确控制每个基的边际选择率，从而突出那些共同激活频率高于独立性假设下预期的基对。由此形成的网络稀疏且不均匀，只有少数边始终具有较高的 PMI，表明 CapFlow 的组合器以协调的方式重新组合能力基，而非将其独立开关。重要的是，这些由 PMI 识别出的配对提供了一种可解释的视角，揭示模型依赖哪些能力组合来构建有效的工作流，支持了我们的主张：CapFlow 学会的是结构化重组，而非启发式试错路由。

5.4 消融实验：分解-重组-决策

我们围绕分解-重组-决策 机制进行消融实验，以确定哪些组件能够提升性能。

分解：能力基的数量。表2改变能力基的数量 K，以检验是否需要一个紧凑的可复用能力因子库。我们发现，能力基过少会导致对多样化工作流因素的拟合不足，并降低迁移效果；而过度扩增 K 则会在领域偏移下带来收益递减（并可能增加归因难度）。我们采用 $K = 8$ 和 Top-m = 3。

表3：重组消融。稀疏组合提升了多领域和未见领域设置下的可控性。

变体	m	自适应	推理	Code	Math	Sci.
单一	1	✓	86.61	84.7	74.31	39.68
稀疏	2	✓	87.12	88.08	75.82	39.13
稀疏	3	✓	**87.62**	**89.73**	77.42	**42.00**
稀疏	3		86.47	88.49	**77.78**	36.27
稠密	8	✓	86.71	87.28	77.58	33.48

表4：决策消融。反事实归因提升了迁移能力，并在领域偏移下稳定了重新组合。

变体	决策信号	推理	Code	Math	Sci.
CapFlow（完整）	CCA + 偏好	**87.62**	**89.73**	**77.42**	**42.00**
不含 NCE	CCA + 模仿	86.54	86.94	75.90	40.44
不含 CCA	偏好	85.71	86.34	74.56	36.44
SFT	模仿	86.97	85.52	76.63	35.65

表 5：骨干大语言模型消融。

骨干网络	推理	Code	Math	Sci.
Llama-3.2-3B-Instruct	87.62	**89.73**	**77.42**	**42.00**
Llama-3.1-8B-Instruct	86.65	88.70	76.13	40.55
Qwen2.5-Coder-3B-Instruct	**87.90**	87.29	76.84	41.70

重新组合：稀疏组合与稠密组合。 表 3 比较了稀疏 top-m 组合与稠密混合，并测试了对 m的敏感性。稀疏组合带来了更好的可控性和迁移能力，而稠密混合往往会模糊能力因子并增大偏移方差。此外，第 4.2 节中采用的自适应温度和逐任务调整优于标准温度调度，因为它能够捕捉不同领域中的任务特定方差。

决策：反事实贡献归因。 表 4 评估了移除反事实能力归因（CCA）或组级对比损失（NCE）后决策 组件的影响。CCA 显著提高了所有数据集上的求解率，表明结构化信用分配对于学习可迁移的能力重组至关重要。

骨干大语言模型。 作为合理性检查，我们在评估套件上比较了几种小规模指令微调骨干模型（表 5）。Llama-3.1-8B-Instruct 和 Qwen2.5-Coder-3B-Instruct 也是跨领域大语言模型微调的有竞争力选择。出于效率考虑，我们选择了 Llama-3.2-3B-Instruct。

6 结论

本研究针对领域偏移下智能体工作流生成在效率和迁移能力方面的核心瓶颈。我们提出工作流能力基学习，将分解-重组-决策 机制内化到开源大语言模型中，从而单次生成可执行的任务特定工作流。我们的 CapFlow 将工作流模式分解 为一组精简的可复用能力基，随后通过任务条件化的

稀疏组合器为新任务重组 这些能力基，并借助反事实归因进行决策，使基选择与真正推动工作流成功的能力相匹配，同时利用成功和失败的工作流提供偏好监督，超越单纯模仿。在多领域、跨领域和未见领域评估中，我们的单次生成器超越了强大的20轮迭代优化基线，同时大幅降低推理时间成本并保持高可靠性。未来工作包括扩展至不断演进的算子库和工具可用性，结合执行反馈实现持续适应，并开发更强的诊断方法和理论，以将所学能力因子泛化到时域更长、拓扑结构更复杂的工作流。

参考文献

[1] Rie Kubota Ando、Tong Zhang 和 Peter Bartlett。2005。基于多任务和无标签数据学习预测结构的框架。《机器学习研究期刊》6, 11（2005）。[2] Anthropic。2024。《Claude 3.5 Sonnet 模型卡补充说明》。https://www-cdn.anthropic.com/fed9cc193a14b84131812372d8d5857f8f304c52/Model_Card_Claude_3_Addendum.pdf 模型卡补充说明，访问于2026年2月9日。[3] Andreas Argyriou、Theodoros Evgeniou 和 Massimiliano Pontil。2006。多任务特征学习。《神经信息处理系统进展》19（2006）。[4]Jacob Austin、Augustus Odena、Maxwell Nye、Maarten Bosma、Henryk Michalewski、David Dohan、Ellen Jiang、Carrie Cai、Michael Terry、Quoc Le 等。2021。使用大语言模型进行程序合成。arXiv预印本 arXiv:2108.07732（2021）。[5] Tom Brown、Benjamin Mann、Nick Ryder、Melanie Subbiah、Jared D Kaplan、Prafulla Dhariwal、Arvind Neelakantan、Pranav Shyam、Girish Sastry、Amanda Askell 等。2020。语言模型是少样本学习者。《神经信息处理系统进展》33（2020），1877–1901。[6] Weilin Cai、Juyong Jiang、Fan Wang、Jing Tang、Sunghun Kim 和 Jiayi Huang。2025。大语言模型中专家混合的综述。《IEEE知识与数据工程汇刊》（2025）。[7] Mark Chen。2021。评估基于代码训练的大语言模型。arXiv预印本 arXiv:2107.03374（2021）。[8] Mark Chen、Jerry Tworek、Heewoo Jun、Qiming Yuan、Henrique Ponde de Oliveira Pinto、Jared Kaplan、Harri Edwards、Yuri Burda、Nicholas Joseph、Greg Brockman、Alex Ray、Raul Puri、Gretchen Krueger、Michael Petrov、Heidy Khlaaf、Girish Sastry、Pamela Mishkin、Brooke Chan、Scott Gray、Nick Ryder、Mikhail Pavlov、Alethea Power、Lukasz Kaiser、Mohammad Bavarian、Clemens Winter、Philippe Tillet、Felipe Petroski Such、Dave Cummings、Matthias Plappert、Fotios Chantzis、Elizabeth Barnes、Ariel Herbert-Voss、William Hebgen Guss、Alex Nichol、Alex Paino、Nikolas Tezak、Jie Tang、Igor Babuschkin、Suchir Balaji、Shan- tanu Jain、William Saunders、Christopher Hesse、Andrew N. Carr、Jan Leike、Josh Achiam、Vedant Misra、Evan Morikawa、Alec Radford、Matthew Knight、Miles Brundage、Mira Murati、Katie Mayer、Peter Welinder、Bob McGrew、Dario Amodei、Sam McCandlish、Ilya Sutskever 和 Wojciech Zaremba。2021。评估基于代码训练的大语言模型。(2021 2107 03374 [ ])。arXiv: . cs.LG [9] Weize Chen、Yusheng Su、Jingwei Zuo、Cheng Yang、Chenfei Yuan、Chi-Min Chan、Heyang Yu、Yaxi Lu、Yi-Hsin Hung、Chen Qian 等。2024。AgentVerse：促进多智能体协作并探索涌现行为……发表于 ICLR。[10] Karl Cobbe、Vineet Kosaraju、Mohammad Bavarian、Mark Chen、Heewoo Jun、Lukasz Kaiser、Matthias Plappert、Jerry Tworek、Jacob Hilton、Reiichiro Nakano 等。2021。训练验证器以求解数学应用题。arXiv预印本 arXiv:2110.14168（2021）。[11] Yongxing Dai、Xiaotong Li、Jun Liu、Zekun Tong 和 Ling-Yu Duan。2021。利用相关性感知专家混合实现可泛化的行人重识别。发表于《IEEE/CVF计算机视觉与模式识别会议论文集》。16145–16154。[12] Alexander D'Amour、Katherine Heller、Dan Moldovan、Ben Adlam、Babak Alipanahi、Alex Beutel、Christina Chen、Jonathan Deaton、Jacob Eisenstein、Matthew D Hoffman 等。2022。规格不足给现代机器学习的可信度带来挑战。《机器学习研究期刊》23, 226（2022），1–61。[13] Pim De Haan、Dinesh Jayaraman 和 Sergey Levine。2019。模仿学习中的因果混淆。《神经信息处理系统进展》32（2019）。[14] Dheeru Dua、Yizhong Wang、Pradeep Dasigi、Gabriel Stanovsky、Sameer Singh 和 Matt Gardner。2019。DROP：需要对段落进行离散推理的阅读理解基准。发表于《2019年计算语言学协会北美分会会议：人类语言技术》，第1卷（长篇和短篇论文）。2368–2378。[15] Hongcheng Gao、Yue Liu、Yufei He、Longxu Dou、Chao Du、Zhijie Deng、Bryan Hooi、Min Lin 和 Tianyu Pang。2025。FlowReasoner：强化查询级元智能体。arXiv预印本 arXiv:2504.15257（2025）。[16] Dan Hendrycks、Collin Burns、Saurav Kadavath、Akul Arora、Steven Basart、Eric Tang、Dawn Song 和 Jacob Steinhardt。[n. d.]。使用 MATH 数据集衡量数学问题求解能力。发表于第三十五届神经信息处理系统会议数据集与基准测试专题（第二轮）。[17] Sirui Hong、Mingchen Zhuge、Jonathan Chen、Xiawu Zheng、Yuheng Cheng、Jinlin Wang、Ceyao Zhang、Zili Wang、Steven Ka Shing Yau、Zijuan Lin 等。2023。MetaGPT：面向多智能体协作框架的元编程。发表于第十二届国际学习表征会议。[18] Shengran Hu、Cong Lu 和 Jeff Clune。 [n. d.]。智能体系统的自动化设计。发表于第十三届国际学习表征会议。[19] Binyuan Hui、Jian Yang、Zeyu Cui、Jiaxi Yang、Dayiheng Liu、Lei Zhang、Tianyu Liu、Jiajun Zhang、Bowen Yu、Kai Dang、An Yang、Rui Men、Fei Huang、Xingzhang Ren、Xuancheng Ren、Jingren Zhou 和 Junyang Lin。2024。Qwen2.5-Coder 技术报告。CoRR abs/2409.12186（2024）。arXiv:2409.12186 doi:10.48550/ARXIV.2409.12186[20] Omar Khattab、Arnav Singhvi、Paridhi Maheshwari、Zhiyuan Zhang、Keshav Santhanam、Saiful Haq、Ashutosh Sharma、Thomas T Joshi、Hanna Moazam、Heather Miller 等。2024年。Dspy：将声明式语言模型调用编译为最先进的流水线。载于第十二届国际学习表征会议。[21] Dmitry Lepikhin、HyoukJoong Lee、Yuanzhong Xu、Dehao Chen、Orhan Firat、Yanping Huang、Maxim Krikun、Noam Shazeer 和 Zhifeng Chen。 [n. d.]。GShard：借助条件计算和自动分片扩展巨型模型。载于国际学习表征会议。[22] Zijun Liu、Yanzhe Zhang、Peng Li、Yang Liu 和 Diyi Yang。2023年。动态大语言模型智能体网络：一种通过智能体团队优化实现协作的大语言模型智能体框架。arXiv 预印本 arXiv:2310.02170（2023年）。[23] Zijun Liu、Yanzhe Zhang、Peng Li、Yang Liu 和 Diyi Yang。2024年。一种面向任务的智能体协作动态大语言模型驱动智能体网络。载于首届语言建模会议。[24] Aman Madaan、Niket Tandon、Prakhar Gupta、Skyler Hallinan、Luyu Gao、Sarah Wiegreffe、Uri Alon、Nouha Dziri、Shrimai Prabhumoye、Yiming Yang 等。2023年。自我优化：借助自我反馈进行迭代优化。神经信息处理系统进展 36（2023年），46534–46594。[25] Meta。2024年。Llama-3.1-8B-Instruct。https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct Hugging Face 模型卡，访问于2026年2月9日。[26] Meta。2024年。Llama-3.2-3B-Instruct。https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct Hugging Face 模型卡，访问于2026年2月9日。[27] Alexander Novikov、Ngân V˜u、Marvin Eisenberger、Emilien Dupont、Po-Sen Huang、Adam Zsolt Wagner、Sergey Shirobokov、Borislav Kozlovskii、Francisco JR Ruiz、Abbas Mehrabian 等。2025年。AlphaEvolve：用于科学与算法发现的编程智能体。arXiv 预印本 arXiv:2506.13131（2025年）。[28] Long Ouyang、Jeffrey Wu、Xu Jiang、Diogo Almeida、Carroll Wainwright、Pamela Mishkin、Chong Zhang、Sandhini Agarwal、Katarina Slama、Alex Ray 等。2022年。训练语言模型，使其遵循带有人类反馈的指令。神经信息处理系统进展 35（2022年），27730–27744。[29] Jonas Pfeiffer、Aishwarya Kamath、Andreas Rücklé、Kyunghyun Cho 和 Iryna Gurevych。2021年。AdapterFusion：用于迁移学习的非破坏性任务组合。载于欧洲计算语言学协会第16届会议论文集：主会场。487–503。[30] Shuofei Qiao、Runnan Fang、Zhisong Qiu、Xiaobin Wang、Ningyu Zhang、Yong Jiang、Pengjun Xie、Fei Huang 和 Huajun Chen。 [n. d.]。智能体工作流生成基准测试。载于第十三届国际学习表征会议。[31] Qwen团队。2024年。Qwen2.5-Coder-3B-Instruct。https://huggingface.co/Qwen/Qwen2.5-Coder-3B-Instruct Hugging Face 模型卡，访问于2026年2月9日。[32] Rafael Rafailov、Archit Sharma、Eric Mitchell、Christopher D Manning、Stefano Ermon 和 Chelsea Finn。2023年。直接偏好优化：你的语言模型其实暗中充当奖励模型。神经信息处理系统进展 36（2023年），53728–53741。[33] David Rein、Betty Li Hou、Asa Cooper Stickland、Jackson Petty、Richard Yuanzhe Pang、Julien Dirani、Julian Michael 和 Samuel R Bowman。 [n. d.]。GPQA：研究生水平、谷歌无法搜索的问答基准。载于首届语言建模会议。[34] Yelong Shen、Phillip Wallis、Zeyuan Allen-Zhu、Yuanzhi Li、Shean Wang 等。[n. d.]。LoRA：大语言模型的低秩适配。（[n. d.]）。[35] Llama团队。2024年。Llama 3模型家族。CoRR abs/2407.21783（2024年）。arXiv:2407.21783 doi:10.48550/ARXIV.2407.21783[36] Patara Trirat、Wonyong Jeong 和 Sung Ju Hwang。2025年。智能体预测器：通过多视图编码预测智能体工作流性能。arXiv 预印本 arXiv:2505.19764（2025年）。[37] Xiaoxuan Wang、Ziniu Hu、Pan Lu、Yanqiao Zhu、Jieyu Zhang、Satyen Subra- maniam、Arjun R Loomba、Shichang Zhang、Yizhou Sun 和 Wei Wang。2024年。SciBench：评估大语言模型的大学水平科学问题解决能力。载于国际机器学习会议。PMLR，50622–50649。[38] Xuezhi Wang、Jason Wei、Dale Schuurmans、Quoc V Le、Ed H Chi、Sharan Narang、Aakanksha Chowdhery 和 Denny Zhou。 [n. d.]。自洽性提升语言模型中的思维链推理。载于第十一届国际学习表征会议。[39] Yinjie Wang、Ling Yang、Guohao Li、Mengdi Wang 和 Bryon Aragam。2025年。ScoreFlow：通过基于评分的偏好优化掌握大语言模型智能体工作流。arXiv 预印本 arXiv:2502.04306（2025年）。[40] Zhenhailong Wang、Shaoguang Mao、Wenshan Wu、Tao Ge、Furu Wei 和 Heng Ji。2024年。在大语言模型中释放涌现认知协同：通过多角色自我协作实现任务求解的智能体。载于2024年计算语言学协会北美分会年会：人类语言技术（长篇论文第1卷）论文集。257–279。[41] Jason Wei、Xuezhi Wang、Dale Schuurmans、Maarten Bosma、Fei Xia、Ed Chi、Quoc V Le、Denny Zhou 等。2022年。思维链提示可激发大语言模型的推理能力。神经信息处理系统进展 35（2022年），24824–24837。

[42] Xun Wu、Shaohan Huang 和 Furu Wei。2023年。Mole：LoRA专家混合。载于第十二届国际学习表征会议。[43] Shengxiang Xu、Jiayi Zhang、Shimin Di、Yuyu Luo、Liang Yao、Hanmo Liu、Jia Zhu、Fan Liu 和 Min-Ling Zhang。2025年。RobustFlow：迈向稳健的智能体工作流生成。*arXiv* 预印本 *arXiv:2509.21834*（2025年）。[44] Zhilin Yang、Peng Qi、Saizheng Zhang、Yoshua Bengio、William Cohen、Ruslan Salakhutdinov 和 Christopher D Manning。2018年。HotpotQA：用于多样化、可解释多跳问答的数据集。载于*2018*年自然语言处理中的经验方法会议论文集。2369–2380。[45] Shunyu Yao、Jeffrey Zhao、Dian Yu、Nan Du、Izhak Shafran、Karthik R Narasimhan 和 Yuan Cao。2022年。ReAct：促进语言模型中的推理与行动协同。载于第十一届国际学习表征会议。[46] Rui Ye、Shuo Tang、Rui Ge、Yaxin Du、Zhenfei Yin、Siheng Chen 和 Jing Shao。[n. d.]。MAS-GPT：训练大语言模型以构建基于大语言模型的多智能体系统。载于第四十二届国际机器学习会议。[47] Jiayi Zhang、Jinyu Xiang、Zhaoyang Yu、Fengwei Teng、Xiong-Hui Chen、Jiaqi Chen、Mingchen Zhuge、Xin Cheng、Sirui Hong、Jinlin Wang 等。[n. d.]。AFlow：自动化智能体工作流生成。载于第十三届国际学习表征会议。[48] Yuanshuo Zhang、Yuchen Hou、Bohan Tang、Shuo Chen、Muhan Zhang、Xi- aowen Dong 和 Siheng Chen。2025年。图神经网络作为智能体工作流性能预测器。*arXiv* 预印本 *arXiv:2503.11301*（2025年）。[49] Zijian Zhang、Shuchang Liu、Jiaao Yu、Qingpeng Cai、Xiangyu Zhao、Chunxu Zhang、Ziru Liu、Qidong Liu、Hongwei Zhao、Lantao Hu 等。2024年。M3oE：多领域多任务专家混合推荐框架。载于第*47*届国际*ACM SIGIR*信息检索研究与发展会议论文集。893–902。[50] Ming Zhong、Yelong Shen、Shuohang Wang、Yadong Lu、Yizhu Jiao、Siru Ouyang、Donghan Yu、Jiawei Han 和 Weizhu Chen。2024年。用于图像生成的多LoRA组合。机器学习研究汇刊 2024（2024年）。[51] Han Zhou、Xingchen Wan、Ruoxi Sun、Hamid Palangi、Shariq Iqbal、Ivan Vulić、Anna Korhonen 和 Sercan Ö Arık。2025年。多智能体设计：通过更优的提示和拓扑结构优化智能体。*arXiv* 预印本 *arXiv:2502.02533*（2025年）。[52] Mingchen Zhuge、Wenyi Wang、Louis Kirsch、Francesco Faccio、Dmitrii Khizbullin 和 Jürgen Schmidhuber。2024年。GPTSwarm：作为可优化图的语言智能体。载于第四十一届国际机器学习会议。

A 附录

A.1 训练与正则化细节

本附录提供了 L^{reg} 中正则化项以及总体训练过程的实现细节。

A.1.1 通过分离路由进行双时间尺度优化。 一个实用的稳定性细节是，将生成器（能力基）和组合器（路由）接收梯度的方式解耦。在工作流生成期间，我们将路由权重的 分离 快照注入所有适配层，即 $\tilde{\alpha}(q) = \text{stopgrad}(\alpha(q))$。这样可以防止组合器通过每个适配模块接收高方差的词元级梯度。相反，组合器主要通过反事实归因（公式（15））以及轻量级路由正则化项进行优化，而能力基则通过 L_{MR} 和 L_{NCE} 进行优化。

A.1.2 正则化。 我们加入轻量级正则项，以防止坍塌并促进组合能力库的形成。与主论文一致，L^{reg} 包括：(i) 基正交性、(ii) 路由熵，以及 (iii) 基丢弃。

(i) 基正交性（跨基的多样性）。 对于每个适配层 ℓ，我们通过惩罚基特定参数向量之间的相关性，促使不同的基编码不同的方向。令$\mathbf{B}_U^{(\ell)} \in \mathbb{R}^{K\times D_U}$ 和$\mathbf{B}_V^{(\ell)} \in \mathbb{R}^{K\times D_V}$ 分别表示跨基堆叠并展平后的 LoRA 因子，其中第 k行是层 ℓ中基 k 的展平参数向量。我们施加：

$$\mathcal{L}_{\text{ortho}} = \sum_{\ell\in\mathcal{L}} \left(\left\| \mathbf{B}_U^{(\ell)} \mathbf{B}_U^{(\ell)\top} - I \right\|_F^2 + \left\| \mathbf{B}_V^{(\ell)} \mathbf{B}_V^{(\ell)\top} - I \right\|_F^2 \right), \tag{17}$$

其中， I是 $K \times K$ 单位矩阵。

(ii) 路由熵（避免过早坍塌）。 我们对路由熵进行正则化，以促进训练早期的探索：

$$\mathcal{L}_{\text{ent}} = -\lambda_{\text{ent}}\, \mathbb{E}_q[H(\alpha(q))], \qquad H(\alpha) = -\sum_{k=1}^{K} \alpha_k \log \alpha_k. \tag{18}$$

最小化 L_{ent} 会增加熵，并防止组合器过早坍塌为单一基。在实践中， λ_{ent} 会被安排为从探索逐步过渡到专门化。

（iii）基丢弃（组合鲁棒性）。在训练期间，我们对 $\alpha(q)$ 应用基丢弃，随机屏蔽路由权重，同时始终保留 argmax 基处于激活状态，随后进行重新归一化：

$$\begin{aligned} &\tilde{\alpha}(q) \propto \alpha(q) \odot m(q), \\ &m_k(q) \sim \text{Bernoulli}(1 - p_{\text{drop}}), \\ &m_{\arg\max_k \alpha_k(q)}(q) = 1, \end{aligned} \tag{19}$$

并将 $\tilde{\alpha}(q)$ 重新归一化，使其总和为 1。这使生成器在缺失能力因子的情况下仍能保持功能，并减少单基主导。

可选稳定器。我们的实现还包含两个稳定器，可在早期探索阶段启用，而不改变核心方法。首先，我们通过詹森-香农惩罚项，使批次平均路由分布与均匀分布匹配，以鼓励全局利用均衡：

$$\mathcal{L}_{\text{bal}} = \lambda_{\text{bal}}\, \text{JS}(\mathbb{E}_{\text{batch}}[\alpha(q)] \,\|\, \mathbf{u}), \tag{20}$$

算法 1 工作流能力基学习（训练） 要求：数据集 $\mathbf{D} = \{(q_i, \mathbf{P}_i, \mathbf{N}_i)\}$；冻结的基础大语言模型 f_{θ_0}；能力基 Φ；组合器 g_ψ；稀疏度 top-m；归因间隔 E。确保：训练后的 Φ, ψ。1：初始化 Φ, ψ；冻结 θ_0。2：对于每个训练步：3：从一个任务 ID 中采样一个任务局部小批次，其中包含成功样本 $\mathbf{P}_i$ 和失败样本 $\mathbf{N}_i$。4：计算任务嵌入$z\ q$ $\alpha\ q\ =$ $g_\psi\ z\ q$ ( ) 和路由权重 ( ) (( ))。5：应用基丢弃（式 (19)）；为适配器注入设置 $\tilde{\alpha}(q)$ =stopgrad($\alpha(q)$)。6：从 $\tilde{\alpha}(q)$ 中选取 top-m 个基，并广播到所有适配层。7：更新能力基 Φ：计算 $\mathbf{L}_{\text{MR}}$ 和 $\mathbf{L}_{\text{NCE}}$，加入 $\mathbf{L}_{\text{ortho}}$ （式 (17)），并对 Φ执行一个梯度步。8：如果 step mod $E = 0$，则：9：反事实归因：对于基 $k \in \mathbf{K}$ (q)，使用反事实路由 $\alpha(-k)$，根据式 (14) 计算 Δ_k。10：更新组合器 ψ：最小化 $\mathbf{L}_{\text{CCA}}$ （式 (15)）以及路由正则项（式 (18) 和式 (20) 的温度正则化）。11：结束 if。12：结束 for。

其中，$\mathbf{u}$ 是 K 个基上的均匀分布。其次，使用自适应温度（第 4.2 节）时，我们用它来正则化极端的温度变化。

A.1.3 反事实归因：实现细节。我们使用反事实路由消融来实例化公式（15）。对于每个基 k，我们通过将 α_k 设为 0 并重新归一化来构造 $\alpha^{(-k)}$，然后重新计算对数似然

$$\ell \qquad \Delta_k = (-k) \text{和}$$

$\ell^{\text{main}} - \ell^{(-k)}$ 14 边际效应（公式（ ））。为降低方差，在将 $\{\Delta_k\}$ 用作信用信号之前，我们对其进行逐样本中心化和裁剪。**失活基底惩罚项**。为使公式（15）中的 L_{dead} 可操作化，我们惩罚那些将概率质量分配给边际效应始终接近零的基的行为：

$$\mathcal{L}_{\text{dead}} = \mathbb{E}\left[\sum_{k\in\mathcal{K}(q)} \alpha_k(q) \cdot \frac{\max(0, \gamma - |\Delta_k|)}{\gamma} \right], \tag{21}$$

其中 γ 为边际阈值。直观而言，如果移除某个基不会改变似然（$|\Delta_k| \approx 0$），那么就会抑制向其分配路由权重。

计算参数。由于反事实归因需要额外的前向传播，我们每隔 E 步计算一次，并可选择在一个批次的子采样集合上进行。在我们的实现中，可以对所有 K 个基组进行归因，同时每次反事实前向传播仍只激活排名前m 的基组，以控制开销。

A.2 训练过程

算法 1 总结了总体训练过程。

收到日期：2026年2月12日；修订日期：2026年2月12日；接受日期：2026年2月12日