Trajectory Guard——一种轻量级、具备序列感知能力的智能体人工智能实时异常检测模型

Laksh Advani独立研究人员，美国华盛顿州西雅图
laad8452@colorado.edu

arXiv:2601.00516v1 [cs.LG] 2 Jan 2026

摘要

自主大语言模型智能体会生成多步行动计划，而这些计划可能因上下文错位或结构不连贯而失效。现有异常检测方法并不适用于这一挑战：均值池化嵌入会稀释异常步骤，而仅对比学习方法则忽略序列结构。采用预训练嵌入的标准无监督方法，其 F1 分数最高仅为 0.69。我们提出 Trajectory Guard，这是一种孪生循环自编码器，采用混合损失函数，通过对比学习联合学习任务-轨迹对齐，并通过重构学习序列有效性。这一双重目标可统一检测“任务对应的计划错误”和“计划结构不规范”两类问题。在涵盖合成扰动以及来自安全审计（RAS-Eval）和多智能体系统（Who&When）的现实世界故障的基准测试中，我们在平衡数据集上取得了 0.88–0.94 的 F1 分数，在不平衡的外部基准数据集上取得了 0.86–0.92 的召回率。推理延迟为 32 毫秒时，我们的方法比大语言模型评审基线快 17–27× 倍，可在生产部署中实现实时安全验证。

引言

近年来，基于大语言模型（LLM）的自主智能体开始涌现，这标志着复杂数字任务的自动化发生了根本性变化。基于大语言模型的智能体会生成多步“轨迹”来自动执行任务，但自主性不断增强也带来了显著的运营风险。实现可信部署的一大障碍在于，智能体可能因误解任务上下文或逻辑行动序列出现断裂，而生成有缺陷、无关或不安全的轨迹。我们的研究证实，标准异常检测技术（如变分自编码器（VAE）或基于现成嵌入的相似性搜索）并不足够，因为它们无法捕捉有效智能体计划的上下文和结构属性。

为解决这一安全问题，我们推出了 Trajectory Guard，这是一种新颖、轻量级的模型，可实时验证智能体轨迹。我们的方法采用孪生循环自编码器，并通过混合损失函数进行训练，同时学习两个关键方面：（1）通过对比学习，评估任务与计划之间的上下文契合度；（2）序列的结构有效性，通过循环重构进行评估。本文的贡献如下：

版权所有 © 2026，人工智能促进协会（www.aaai.org）。保留所有权利。

1. 我们证明，将标准异常检测方法应用于预训练嵌入，无法有效验证智能体轨迹，从而确立了对专用模型的需求。

2. 我们提出了一种新颖的序列感知型孪生循环自编码器，并采用混合损失函数进行实时轨迹异常检测。

3. 我们在基准测试上开展实验，将 Galileo（AI 2025）和 AgentAlign（Zhang 等人，2025a）的合成数据，与 RAS-Eval（Fu、Yuan 和 Wang，2025）及 Who&When（Zhang 等，2025b）中的现实世界故障相结合。在这些测试中，我们在合成基准测试上取得了 0.88–0.94 的 F1 分数，在现实世界日志上取得了较高的召回率（0.86–0.92）。4. 我们证明，该方法的速度比大语言模型评审基线快 17× 倍以上，因此适用于实时部署。5. 我们贡献了一款可部署工具，用于验证智能体系统中的轨迹与任务的一致性。

相关工作

我们的工作将序列数据异常检测与新兴的大语言模型智能体安全领域联系起来。

异常检测：从经典方法到大语言模型方法。根据我们的实验，变分自编码器（Kingma 和 Welling，2013）和孤立森林（Liu、Ting 和 Zhou，2008）等传统无监督方法效率很高，但无法捕捉智能体轨迹中的语义细微差别。大语言模型评审器（Liu 等，2024）的准确率很高（F1 分数最高可达 0.95），但延迟也很高（556–734 毫秒），不适用于实时场景。

智能体系统中的异常检测。近期研究致力于检测智能体异常，例如用于驾驶轨迹的时空图自编码器（Wiederer 等人，2022）（侧重运动学），以及 SentinelAgent 用于多智能体风险的执行图（He 等人，2025）（定性分析，未提供延迟指标）。这些方法适用于多智能体或机器人技术场景，但不适用于单智能体语言计划。

架构上的相似之处与安全对齐。我们的孪生循环神经网络（详细流程见附录中的图1）借鉴了日志异常检测（Hamooni 等人，2021），并针对大语言模型轨迹采用混合损失进行改造。AgentAlign（Zhang 等，2025a）等相关工作则合成数据以实现安全对齐，侧重于生成前预防，而非生成后验证。

我们的贡献。 以往方法忽视了针对单智能体语言轨迹的轻量级实时防护机制（Du 等人，2025）。Trajectory Guard 填补了这一空白，在合成基准测试上取得了 0.88–0.94 的 F1 分数，并在单个模型中实现了 32 毫秒的延迟，有别于高延迟评判器或多智能体图。

数据集与异常合成

为严格评估所提模型，我们通过整合多个数据源构建了一个全面的基准测试。我们的基础由两个开源智能体轨迹数据集组成，即 **Galileo** 和**AgentAlign**；我们利用它们训练模型，并合成多样的上下文异常和结构异常。我们使用两个外部带注释的基准测试扩充了测试集：**RAS-Eval**（一个全面的安全基准）以及**Who&When**（一个多智能体故障日志数据集）。这种结合方式最终形成了一个高质量数据集，可用于稳健的模型训练和全面的评估。

训练与合成数据集

我们的训练与合成核心数据集由两个具有不同轨迹格式的数据源构建而成，从而确保我们的方法论不会过拟合于单一的智能体交互风格。

- **Galileo**：galileo-ai/agent-leaderboard-v2 数据集中的自适应工具使用配置。该基准测试包含多个企业领域（如银行业、电信业）的轨迹，并以自然语言命令列表的形式呈现。
- **AgentAlign**：一个大规模智能体安全基准测试。我们使用其中的“良性”类别，提取以结构化 JSON 工具调用序列表示的轨迹。

外部评估基准

为评估模型在真实世界故障（而非合成异常）中的鲁棒性，我们纳入了两个外部测试集。

- **RAS-Eval**：一个全面的安全基准，支持模拟和真实世界的工具执行。我们使用了其中的 3,802 条异常轨迹作为测试集。
- **Who&When**：一个包含来自 127 个大语言模型多智能体系统的大量故障日志的数据集。我们提取了 184 条日志，并进行了细粒度标注，将故障与具体错误步骤关联起来。

异常合成流程

为了创建具有已知真实标签且更具挑战性的测试集，我们使用大语言模型（openai/gpt-5）对 Galileo 和 AgentAlign 数据的一个子集进行了受控轨迹扰动。对于每条“良好”轨迹，我们都会通过注入随机数量（1、2 或 3 个）的无效步骤，创建一个对应的“异常”版本。

该合成过程由列表1（附录）中的提示引导，生成了ווגמ多样的异常，包括：

- **上下文异常：** 注入单独来看合乎逻辑、但对于特定任务而言不合时宜的步骤（例如，在电信任务中执行“搜索新应用程序”，如表 1 所示；或在工作流中途执行“CloseMusicApp”，如表 2 所示）。
- **结构异常：** 引入格式错误或无意义的步骤，例如带有错误或危险参数的工具调用（如使用危险路径删除文件），或针对简单查询的不合逻辑推理轨迹。

表 1：Galileo 轨迹对示例。

标签	轨迹步骤（摘录）
Good	[’查询漫游费用’, ’Add 国际通话’, ’升级妻子 Sarah 的 iPhone’, ’携号转网企业固定电话’]
异常	[’查看漫游费用’, ’升级妻子 Sarah 的 iPhone’, ’添加国际通话’, ’携号转网企业固定电话’ , **’搜索新的应用程序以下载’** , ’检查是否安排了基站维护’]

表 2：AgentAlign 轨迹对示例。

标签	轨迹步骤（摘录）
Good	[{"name":"获取新音乐发行"}, {"name":"获取主流摇滚歌曲排行榜"}]
异常	[{"name":"获取新音乐发行"}, **{"name":"CloseMusicApp"}**, {"name":"获取主流摇滚歌曲排行榜"}, **{"name":"打开播客"}**]

最终数据集划分

最终数据集采用 Galileo 和 AgentAlign 数据进行分层划分，并将外部基准数据集专门留作测试集。重要的是，在进行任何模型训练之前，所有测试集样本（包括正常样本和异常样本）都已留出，以确保不会发生数据泄漏。

为了创建用于超参数调优和消融研究的验证集，我们将初始训练集按 85% 和 15% 的比例划分为训练集和验证集。验证集仅用于模型选择和早停，且

未用于最终评估。我们使用相同的 GPT-5 扰动流程合成验证轨迹中的异常，以创建平衡验证集。

- **初始训练池**：**6,767**条来自 Galileo 和 AgentAlign 的“良性”轨迹– **训练集**：**5,752** 条轨迹（85%），用于模型训练– **验证集**：**1,015** 条轨迹（15%），用于超参数调优和消融研究

- **最终测试集**：共包含**5,822** 条轨迹，在训练前留出，在模型开发过程中从未见过。测试集细分如下：

  – 良性样本：1,000 条（250 条来自 Galileo，750 条来自 AgentAlign），与训练集不重叠– 异常样本：共 4,822 条，来自三个来源：**合成（我们的方法）**836 条（248 条来自 Galileo，588 条来自 AgentAlign）– 通过 GPT-5 对留出的良性轨迹进行扰动生成**RAS-Eval** 3,802 条 – 外部安全基准**Who&When** 184 条 – 外部多智能体故障日志

方法论

我们的目标是开发一种轻量级、实时运行的模型 Trajectory Guard，用于检测大语言模型智能体轨迹中的异常。我们将轨迹 τ 定义为一系列动作步骤 $\{s_1, s_2, ..., s_n\}$，这些步骤旨在完成一项用户任务 T。我们的研究围绕三个核心假设展开，最终提出了一种新颖的序列感知架构。

假设 1：异常作为点异常值

我们最初将异常轨迹视为预训练嵌入空间中的统计离群值（Reimersand Gurevych 2019），并将其建模为无序的“步骤袋”。我们通过对 SentenceTransformer 的步骤嵌入进行均值池化，计算出一个固定大小的向量$\mathbf{v}_\tau$，然后应用无监督检测器（变分自编码器、孤立森林、单类支持向量机）。这种方法效果不佳（F1 < 0.70；表 3），因为平均操作会稀释异常步骤，使$\mathbf{v}_\tau$ 难以与有效轨迹区分开来。

假设 2：异常作为上下文不匹配

接下来，我们假设异常是任务 T 与轨迹 τ之间的上下文不匹配。我们使用多负例排序损失（Khattab 和 Zaharia 2020）对 all-MiniLM-L6-v2 进行对比式微调，训练数据为（任务，轨迹）配对，以最大化有效配对的余弦相似度。这将性能提升至 F1 ≈ 0.82，但模型仍然脆弱，在不同轨迹格式之间表现出负迁移，并且忽略了序列结构。

最终方案：序列感知孪生架构

设计依据。 稳健的防护机制必须将轨迹建模为**结构化序列**，并区分**上下文**异常（任务与轨迹不匹配）和**结构性**异常（不连贯计划）。我们的架构可同时检测这两类异常。

架构。**Trajectory Guard** 是一种孪生循环自编码器，包含两个塔：

- **任务塔：** 通过多层感知机投影将任务嵌入映射到 128 维潜在向量$\mathbf{v_t}$。
- **轨迹塔：** GRU 编码器处理序列$\{\mathbf{s_1}, ..., \mathbf{s_n}\}$并生成“思维向量” $\mathbf{v_s}$；GRU 解码器则重构原始序列。

混合损失函数。 我们的关键创新结合了两个相辅相成的目标：通过任务-轨迹对齐实现上下文相关性的**对比损失**($\mathcal{L}_{\text{contrastive}}$)，以及通过序列重构实现结构有效性的**重构损失** ($\mathcal{L}_{\text{reconstruction}}$)。

$$\mathcal{L} = \mathcal{L}_{\text{contrastive}} + \alpha \cdot \mathcal{L}_{\text{reconstruction}} \tag{1}$$

$\mathcal{L}_{\text{contrastive}}$ 采用三元组边际损失，并在批次内采样负样本：任务嵌入作为锚点，对应的轨迹作为正样本，批次中的其他轨迹作为负样本。这样可以缩短$\mathbf{v_t}$ 与$\mathbf{v_s}$ 之间的距离，同时增大它们与负样本之间的距离（“这是正确的计划吗？”）。 $\mathcal{L}_{\text{reconstruction}}$ 采用均方误差损失来学习轨迹“语法”——有效的步骤顺序与组合（“这个计划连贯吗？”）。

这一双重目标能够在统一模型中联合检测上下文不匹配和结构不连贯，从而应对这两种失效模式。

实现细节

负采样策略。 对于三元组边际损失，我们采用了批内负采样。在每个包含 N $(task, trajectory)$ 对的训练批次中，每个任务嵌入都作为锚点，其对应的轨迹作为正样本，而批次中的其他 $N-1$ 条轨迹则作为负样本。这种高效的方法无需显式生成负样本，即可提供多样化的困难负样本。

我们使用 all-MiniLM-L6-v2 作为基础嵌入器，在 5,752 条训练轨迹上训练模型，并进行了 20 个 epoch 的微调（批次大小为 16，使用 Adam 优化器，学习率为2×10^{-5}）。多层感知机任务头将维度从 384 投影到 128；GRU 编码器/解码器的隐藏维度为 128。我们将静态损失权重设为 $\alpha = 0.5$，三元组边际设为 1.0。在留出的验证集上选择异常阈值，以最大化 F1 分数。基准测试使用了 NVIDIA T4 GPU 和主频为 2.00GHz 的 Intel Xeon CPU。

实验与结果

我们在合成基准测试（Galileo、AgentAlign）和真实世界留出集（RAS-Eval、谁/何时）上评估了 Trajectory Guard，结果表明它在效率和效果方面均表现出色。

性能指标

表 3 展示了全面的性能比较。轻量级基线（变分自编码器、孤立森林）表明，标准的现成方法无法胜任这一专门任务。我们的模型显著优于这些基线，并且在合成基准测试上的 F1 分数可与重量级大语言模型评审器媲美，甚至更胜一筹。

在平衡合成数据集上，Trajectory Guard 的 F1 分数达到 0.88–0.94（加权平均为 0.92），加权平均召回率为 0.91。对于外部基准测试集，由于其中主要包含异常轨迹（RAS-Eval：3,802 个异常；Who&When：184 个异常），我们将召回率作为关键安全指标进行报告，因为假阴性（漏检的异常）相比假阳性会带来更大的运营风险。在这些留出集上，我们的模型取得了较高的召回率（RAS-Eval 为 0.86，谁/何时为 0.92），优于 Phi-3-mini（分别为 0.76 和 0.88），并接近或达到重量级基线的水平。这验证了模型对现实世界安全漏洞和多智能体故障的泛化能力；在这些场景中，召回率对安全至关重要。

误差分析

分析表明，对于超过 10 步的轨迹，精确率会下降。固定的 128 维门控循环单元向量（Cho 等人，2014）构成了信息瓶颈，导致较长有效序列的重构误差增加。未来的工作将探索注意力机制，以缓解这一问题。

延迟比较

表 4 展示了推理延迟基准测试结果。我们的图形处理器模型比最快的大语言模型裁判（Deepseek）快 17×，比 Phi-3-mini 快 27× ，证实其适用于实时部署。虽然我们的模型是在 T4 图形处理器上进行基准测试，而 Phi-3-mini 是在 A100 图形处理器上进行测试，但 27× 的加速幅度是保守估计。

混合损失消融研究

为验证混合损失设计，我们进行了消融研究，分别仅使用对比损失或仅使用重建损失训练模型。如表 5 所示，完整的混合模型显著优于单独使用任一组成部分的模型，证实同时学习上下文和结构对于取得高性能至关重要。

局限性

我们依赖 GPT-5 合成的异常进行评估，这有助于实现受控评估，但也存在循环论证的风险。人工标注留出集（RAS-Eval、Who&When）可缓解这一问题，不过在大规模攻击上的召回率略有下降（RAS-Eval 的 3,802 个样本上为 0.86）。模型在长轨迹上的表现也会下降：2–5 步时 F1 分数为 0.96，而在 11+ 步时为 0.87，这是由于门控循环单元的定长编码瓶颈所致。未来的工作可以引入注意力机制，以更好地处理长距离依赖。

结论

Trajectory Guard 通过轻量级实时异常检测，保障可验证大语言模型的智能体人工智能安全

表 3：准确率比较。平衡基准测试中异常类别的精确率（P）、召回率（R）和 F1 分数（F1）。混合（合成） = 为 Galileo 和 AgentAlign 的加权平均。对于高度不平衡的外部基准数据集（RAS-Eval、Who&When），由于缺少正常样本，我们仅报告召回率。

模型	数据集	P	R	F1
所提模型				
我们的孪生循环神经网络	Galileo	0.90	0.86	0.88
	AgentAlign	0.95	0.93	0.94
	混合（合成）	0.94	0.91	0.92
	RAS-Eval	–	0.86	–
	谁/何时	–	0.92	–
重量级基线				
Gemini Flash 1.5	混合（合成）	0.93	0.98	0.95
	RAS-Eval	–	0.91	–
	谁/何时	–	1.00	–
GPT-4o Mini	混合（合成）	0.89	0.97	0.92
	RAS-Eval	–	0.96	–
	谁/何时	–	0.79	–
Deepseek v3.2-exp	混合（合成）	0.77	0.94	0.85
	RAS-Eval	–	0.90	–
	谁/何时	–	1.00	–
Phi-3-mini (A100)	混合（合成）	0.80	0.82	0.81
	RAS-Eval	–	0.76	–
	谁/何时	–	0.88	–
轻量级基线				
变分自编码器（bge-large）	混合（合成）	0.86	0.57	0.69
单类支持向量机	混合（合成）	0.45	0.71	0.55
孤立森林	混合（合成）	0.43	0.70	0.53

表 4：延迟比较（毫秒/样本）。我们的模型运行于 NVIDIA T4 GPU；Phi-3-mini 运行于 NVIDIA A100 GPU（更高端的硬件）；LLM 评审器通过 API 调用。尽管使用了较低端的 GPU 硬件，我们的模型仍比 Phi-3-mini 快 27× 倍，表明其架构效率显著提升，且这一优势并非仅由硬件差异带来。

模型	延迟（毫秒）
孤立森林（CPU）	<1.0
我们的模型（图形处理器）	**32.48**
我们的模型（CPU）	**161.16**
Phi-3-mini（A100）	892
大语言模型裁判（Deepseek-v3.2-exp）	557
大语言模型裁判（Gemini-Flash-1.5）	603
大语言模型裁判（GPT-4o Mini）	735

表 5：损失分量的消融研究。报告的是平衡验证集上的异常 F1分数。

损失配置	异常F1 分数
仅使用对比损失	0.82
仅使用重建损失	0.75
混合损失（两者）	**0.92**

智能体。我们证明，标准异常检测

仅依赖嵌入是不够的，因此我们提出了孪生循环自编码器，并采用混合损失，同时学习上下文信息和结构有效性。在合成基准测试（Galileo、AgentAlign）上，该模型的 F1 分数达到 0.88–0.94；在带标注的留出集（RAS-Eval、Who&When）上也取得了较高的召回率（0.86–0.92），同时比 LLM 评审器和 Phi-3-mini 快 17–27× 倍，从而支持实时部署。未来工作包括在更多真实异常上进行验证、开展更广泛的领域训练，以及与量化大语言模型进行基准测试，以便在不断演进的人工智能安全标准中定位我们的方法。

参考文献

AI，G. 2025。Agent排行榜v2。https://huggingface.co/datasets/galileo-ai/agent-leaderboard-v2。

Cho, K.; van Merri¨enboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. 使用RNN编码器-解码器为统计机器翻译学习短语表示。载于*2014*年自然语言处理实证方法会议（*EMNLP*）论文集，1724–1734。Du, Y.; Feng, J.; Zhao, J.; Yuan, J.; and Li, Y. 2025. TrajAgent：一种通过大模型与小模型协作实现自动化轨迹建模的基于大语言模型的智能体框架。arXiv:2410.20445。Fu, Y.; Yuan, X.; and Wang, D. 2025. RAS-Eval：用于真实环境中大语言模型智能体安全评估的综合基准测试。arXiv:2506.15253。Hamooni, H.; Debnath, B.; Xu, J.; Zhang, H.; Jiang, G.; and Mueen, A. 2021. 使用孪生网络检测软件执行日志中的异常。载于*2021*年*IEEE/ACM*第*43*届国际软件工程会议：软件工程实践（*ICSE-SEIP*），82–91。He, X.; Wu, D.; Zhai, Y.; and Sun, K. 2025. SentinelAgent：多智能体系统中的基于图的异常检测。arXiv:2505.24201。Khattab, O.; and Zaharia, M. 2020. ColBERT：通过BERT上的上下文化后期交互实现高效且有效的段落检索。*arXiv*预印本 *arXiv:2004.12832*。Kingma, D. P.; and Welling, M. 2013. 自动编码变分贝叶斯。*arXiv*预印本 *arXiv:1312.6114*。Liu, F. T.; Ting, K. M.; and Zhou, Z.-H. 2008. 孤立森林。载于*2008*年第八届*IEEE*国际数据挖掘会议，413–422。Liu, P.; Jiang, W.; Yun, H.; Huang, A.; Bin, Y.; Evtikhiev, M.; Roy, D.; and Mishra, S. 2024. 智能体即评审者：用智能体评估智能体。arXiv:2410.10934。Reimers, N.; and Gurevych, I. 2019. Sentence-BERT：使用孪生BERT网络生成句子嵌入。载于*2019*年自然语言处理实证方法会议及第九届自然语言处理国际联合会议（*EMNLP-IJCNLP*）论文集，3982–3992。Wiederer, J.; Bouazizi, A.; Troina, M.; Kressel, U.; and Belagiannis, V. 2022. 自动驾驶多智能体轨迹中的异常检测。载于机器人学习会议，122–133。

Zhang, J.; Yin, L.; Zhou, Y.; and Hu, S. 2025a. AgentAlign：在从信息型大语言模型向智能体型大语言模型转变的过程中探索安全对齐。arXiv:2505.23020。Zhang, S.; Yin, M.; Zhang, J.; Liu, J.; Han, Z.; Zhang, J.; Li, B.; Wang, C.; Wang, H.; Chen, Y.; and Wu, Q. 2025b. 哪个智能体在何时导致了任务失败？关于大语言模型多智能体系统自动化失败归因的研究。arXiv:2505.00212。

附录

示意图和提示词

本附录包含一张示意图，以及用于合成异常轨迹和进行大语言模型裁判基线评估的完整提示词。

系统架构

图1展示了 Trajectory Guard 的完整孪生循环自编码器架构，详细说明了用于处理任务上下文和轨迹结构的并行处理路径。

列表1：异常合成提示词。

1 你是人工智能安全实验的数据生成专家。你的任务是接收一系列有效的工具调用步骤（智能体轨迹），并注入指定数量的异常步骤。异常步骤必须满足以下条件：与该序列在逻辑上不一致、属于相关领域，且语法上合理但语义错误。仅返回一个有效的 JSON对象，其中包含键 " corrupted_trajectory"。

列表2：大语言模型评审基线提示词。

1 根据指定任务分析所提供的计划。评估这些步骤是否构成连贯、合乎逻辑且直接支持任务完成的序列。标记任何无关、矛盾、顺序错误或以其他方式破坏逻辑进程的步骤，因为这些可能表明存在异常。最后，回答必须仅以一个词结尾：如果计划完全合乎逻辑且相关，则写“ good”；如果发现任何问题，则写“anomaly”。

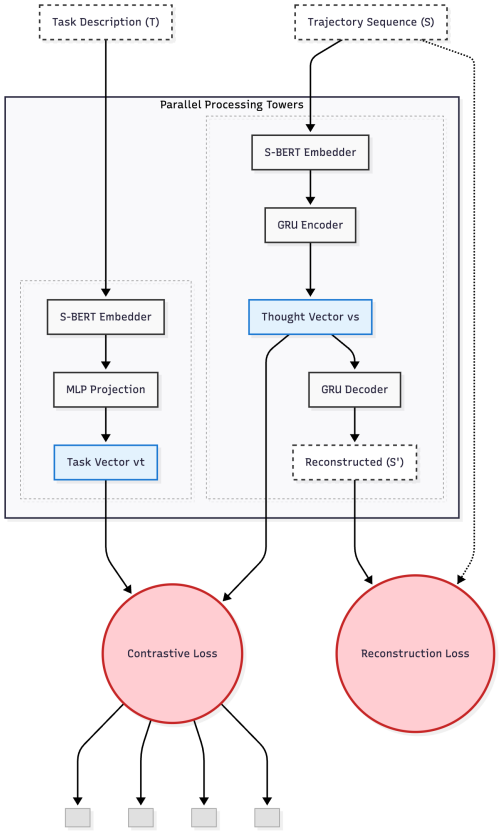

图1：**轨迹防护架构**。模型采用孪生结构，包含两个并行塔。**任务塔**（左侧）将任务描述映射到潜在空间 v_t。**轨迹塔**（右侧）使用 GRU 编码器将动作序列压缩为思维向量 v_s。模型通过混合目标函数进行训练：**对比损失**对齐 v_t 和 v_s，以确保计划与任务相匹配；而**重建损失**则确保轨迹序列保留结构有效性。