arXiv:2601.10128v1 [cs.CE] 15 Jan 2026

面向数据稀缺场景构建可执行领域专用大语言模型的可泛化框架：半导体技术计算机辅助设计仿真的演示

Di Wang[1],Zhenhua Wu[2],Yu Liu*[1],Kai Chang*[2],和 Shaohua Wu[1]

[1]英特尔电子信息产业有限公司，北京，中国。[2]量子物质研究中心，浙江大学，中国杭州。

*通讯作者：Yu Liu（liuyubj@inspur.com）或 Kai Chang（kchang@zju.edu.cn）。

摘要

科学与工程领域通常面临数据稀缺和严格的可执行性要求：模型不仅必须生成流畅的文本，还必须生成语法有效且可由工具编译的脚本。我们提出了一种模式优先对齐框架，用于在低资源环境下构建紧凑且可执行的领域专用大语言模型。该框架包含三个核心组件：(i) 从专家文档中大规模生成合成问答数据，以注入基础领域知识；(ii) 以代码为中心的中间表示→直接偏好优化流程，将经过验证的工具输入文件转换为可解释的中间表示（中间表示），执行保持等价性的多样化，并构建偏好对，以直接优化指令遵循能力和代码可执行性；以及 (iii) 对检索增强生成（检索增强生成）进行受控评估，结果表明，尽管检索增强生成能够使通用大语言模型受益，但可能会使已经完成领域对齐的模型性能略有下降。

我们通过为半导体技术计算机辅助设计（TCAD）构建 TcadGPT 实例，展示了该框架的应用。借助 150 万个合成问答对和一个 IR 驱动的 DPO 数据集，TcadGPT 在 SDE 可执行性测试中的语义准确率达到 85.6%，语法通过率达到 80.0%，大幅优于最先进的通用大语言模型，例如 GPT-4o。为探究该方法在 TCAD 之外的可移植性，我们将同一套方案应用于开源有限元求解器*Elmer*，发现其脚本级成功率相较于通用基线始终有所提升。为确保研究结果可复现，我们公开了所有数据集、基准测试和代码（包括 P1、P2 和 IR→DPO）。总体而言，这些结果表明，所提出的框架为在数据稀缺的专业领域构建可执行的大语言模型提供了一条稳健且可复现的路径。

1 引言

先进的半导体研发日益依赖高保真数字孪生，将预测物理与快速、数据驱动的迭代相结合。在此类工作流中，技术计算机辅助设计（TCAD）充当计算骨干，用于解析纳米尺度上紧密耦合的工艺—器件相互作用，并生成既要保证物理准确性、又可由工具执行的仿真产物。随着器件尺寸持续缩小，工艺复杂度不断上升，物理效应耦合日益增多 [1, 2], ，使 TCAD 工作流呈现出高维参数空间、强非线性以及脚本密集型迭代设置等特点。综合这些因素，便产生了迫切需求——

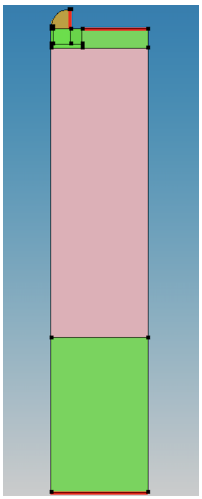


图1：**由模型生成的 SDE 文件生成的可执行结果**。根据自然语言指令构建的平面 MOSFET 结构，经我们的模型转换为 SDE 脚本，并成功执行完成，未进行人工修改。图中展示了网格生成并导出（.tdr/.bnd）后的掺杂剂分布（掺杂浓度，cm^{-3}）。该示例包含多种材料（硅、SiO_2, 多晶硅、Si_3N_4）、接触定义（栅极/源极/漏极/衬底）、区域特定的掺杂分布，以及多级网格细化。此结果展示了指令遵循和工具可执行性，也为本文后续介绍的 schema 优先中间表示→直接偏好优化对齐策略提供了依据。

迫切需要智能自动化，在降低人工开销的同时，保留工业求解器所要求的数值严谨性。

近年来，人工智能（AI）的进展展现出协助技术计算机辅助设计（TCAD）工作流程的显著潜力，主要体现在代理建模和优化方面。神经网络已被应用于近似静电势、电流-电压特性及其他器件响应，从而大幅减少仿真时间和计算成本 [3–5]。物理信息神经网络（PINN）进一步将控制方程整合到学习框架中，在数据驱动模型与物理约束之间架起桥梁 [6–8]。贝叶斯和基于帕累托前沿的优化策略也已成功减少设计探索所需的昂贵TCAD仿真次数 [9–11]。然而，这些方法主要专注于加速数值计算，而未解决一个核心的可用性瓶颈：复杂TCAD脚本的编写、调试和迭代细化。在工业实践中，基于物理的求解器对于保证精度仍不可或缺，但TCAD用户仍需面对陡峭的学习曲线、易出错的脚本编写，以及端到端仿真工作流程中有限的自动化能力。

与此同时，大语言模型（LLM）已迅速推动电子设计自动化（EDA）领域的技术发展。它们支持用户通过自然语言与设计工具交互（例如 ChatEDA [12]），自动生成高质量的 RTL 和 HDL（例如 ChipGPT [13], RTLLM [14]），协助验证和规则检查（例如 DRC-Coder [15]），并支持制造过程中的多模态缺陷分析（例如 FabGPT [16]）。尽管取得了这些成果，大语言模型在技术计算机辅助设计（TCAD）领域的应用仍鲜有探索。与许多 EDA 任务有根本区别的是，TCAD 要求严格遵循特定模拟器的语法、层次化几何定义和执行顺序约束，且最终产物必须能够被工具接受，而非仅仅是描述性文本。

截至目前，唯一一项直接将大语言模型应用于TCAD的工作来自Nguyen等人 [17], ，他们在约7,000个结构输入文件上对一个Transformer进行了微调，以生成TCAD脚本。更广泛而言，现有尝试通常依赖多样性有限、构造范围狭窄的数据集，这些数据集往往通过对单一模板进行词元级变异生成。这些研究缺乏标准化基准以及与通用大语言模型的系统性比较，因此其有效性和可移植性仍不明确。不同于Verilog或HLS等领域，TCAD面临极端的数据稀缺问题：供应商不会发布大规模脚本语料库，而公开可用的材料——用户指南、教科书和随附示例——原本也不是为机器学习设计的。因此，TCAD输入文件（例如.cmd、.tdr）凭借其嵌套语法、层次结构，以及命令与物理意图之间的紧密耦合，成为对大语言模型而言尤其具有挑战性的测试平台。

我们的视角与目标。我们并不旨在提出一个单任务系统，而是引入一个**模式优先的对齐框架**，用于在数据稀缺条件下构建可执行的领域专用大语言模型。我们以TCAD为应用对象，将该框架实例化为TcadGPT，其主要目标是将自然语言意图转换为模拟器可接受的脚本，并确保高度的指令保真度。该框架结合了基于专家文档的大规模问答合成，以及以代码为中心的中间表示→直接偏好优化对齐流水线，从而显式优化语法有效性和执行顺序。图1展示了TcadGPT生成并在无需人工干预的情况下执行的一个具有代表性的SDE结果。为避免从单一领域过度宣称其通用性，我们进一步使用相同的对齐方案，对开源有限元法求解器*Elmer* [18],开展了一项轻量级扩展研究，以探究其在TCAD之外的可移植性。

2 通用模式优先对齐框架

设计原则。 我们的框架面向以下领域：(i) 公开语料库稀缺且分散；(ii) 解决方案必须生成可由工具执行的 产物（例如模拟器脚本）；(iii) 指令遵循的忠实度至关重要。该框架由三个紧密耦合的阶段组成：

•**通过问答合成获取知识**。 两条互补的流程将专家整理的文档大规模转化为*Alpaca* 风格的指令–响应对：**流程 1**（基于片段的问答）侧重于广泛覆盖和措辞鲁棒性；**流程 2**（关键词引导的问答）则针对细粒度概念、命令和方程，以提高精确度。两者共同构建用于推理和语法意识的领域先验。

• **以中间表示（IR）为先导的代码对齐→直接偏好优化**。对已验证的可执行文件进行解析，生成能够保留语义（维度、几何布尔运算顺序、接触、掺杂、网格/导出）的中间表示（*IR*）。IR 支持 (a) 保持等价性的多样化以及 (b)确定性指令渲染。随后，我们构建包含单因素、可解释违规项（数值尺度、顺序/过程、导出遗漏）的直接偏好优化偏好对，以直接优化指令遵循度和语法有效性。

• **将检索作为可选且受控的组件**。 检索增强生成可以通过领域文本为通用大语言模型提供依据，从而显著提升其表现，但可能削弱 已专门化的小型模型所具备的先验。我们的框架将检索增强生成视为可选组件，并对其加以模式约束——仅在基础模型为通用模型，或能够通过结构化检索避免词汇过拟合时使用。

成果。 该框架能够生成紧凑型模型，可输出可由工具编译的 代码以及稳定且实用性高的问答响应。接下来，我们通过 TcadGPT 展示该框架在技术计算机辅助设计领域的应用，并在开源有限元法求解器 *Elmer* 上进行轻量级扩展，以探究其可移植性。

3 TcadGPT：面向技术计算机辅助设计领域适配的大语言模型

大语言模型（LLM）的快速发展推动了通用应用和科学应用领域的突破，并催生了新兴的科学大语言模型（LLM4S）领域。然而，它们在技术计算机辅助设计（TCAD）等高度专业化领域的适用性仍极为有限。在科学问答方面，苏黎世联邦理工学院推出了一种基于科学文献和维基百科训练的多语言大语言模型 [19]。该模型虽擅长通用问答任务，却不支持代码理解或程序化文件生成。其他领域专用大语言模型 [20–22] 则针对量子化学、电子设计自动化脚本编写或芯片布局等领域。不过，这些模型通常依赖小规模数据集、高层次抽象或仅提示调优。例如，ORCA-LLM [20] 专注于生成量子化学模拟的输入脚本，但其覆盖的领域逻辑有限，指令多样性也不足。LLM4EDA [21]则综述了

硬件设计领域的应用，但缺乏针对特定工具的深入研究。Chip-NeMo [22] 将大语言模型应用于电子设计自动化任务，但并未针对 TCAD 环境所要求的代码结构保真度或底层语法。相比之下，我们的工作在范围严格限定、资源有限的场景下，明确同时针对命令语法和领域语义。

这一差距主要源于公开可用训练数据的匮乏，以及技术计算机辅助设计（TCAD）所需的复杂领域专属语法、建模原理和仿真工作流程。

与 Verilog 或 RTL 等其他硬件设计语言不同，后者受益于活跃的开源生态系统，并能通过 GitHub 等平台获取丰富的数据集，而 TCAD 仿真脚本却很少公开分享。大多数商业 TCAD 工具——例如 Sentaurus Process、Sentaurus Device 和 Garand——都是专有软件；用于实际半导体仿真的输入文件往往包含与工艺技术相关的敏感信息，因此严格保密。因此，模型开发可用的资源仅限于用户指南或基础训练示例等少量材料。这种数据稀缺严重阻碍了大语言模型在 TCAD 领域进行有效推理。

因此，现有大语言模型在 TCAD 相关任务上的表现不佳。我们的初步实验表明，即使是 DeepSeek V3 这样的最先进模型，也无法可靠地回答代码密集型问题，在实际场景中的实用性有限。为应对这些挑战，我们提出了 TcadGPT：一款面向使用 TCAD 工具的半导体研究人员打造的紧凑型（80 亿参数）领域专用大语言模型。

我们的工作做出了以下贡献：

- **大规模合成数据集**。我们设计了两种互补的合成数据生成流程，将公开的技术计算机辅助设计手册、教科书和培训文档转换为大规模指令-响应语料库，其中包含超过**150 万**个 Alpaca 格式的样本，仅供非商业学术用途。

- **含有参考答案的可复现基准测试**。我们构建了一个全面的**264 题 TCAD 基准测试套件**，涵盖物理建模、仿真逻辑和工具命令语法。每道题都由专家提供参考答案，并根据标准化评分标准对模型输出进行评分，该标准仅考虑可接受的等价表达，从而实现不同模型之间透明且可复现的比较。

- **中间表示→直接偏好优化对齐流水线**。我们提出了一种**以代码为中心的中间表示→直接偏好优化对齐工作流**，从经过验证的技术计算机辅助设计输入文件中提取中间表示，进行保持等价性的多样化，并构建具有受控指令违例的结构化偏好对。此阶段基于经过问答调优的 TcadGPT-P1&2 模型，并直接优化指令遵循保真度和代码可执行性，最终得到第 7.2 节使用的 TcadGPT-P1&2+DPO 检查点。

- **模型性能与资源效率**。经过问答调优的 TcadGPT-P1&2 模型在包含 264 个问题的问答套件上取得了**85.6%** 的基准测试准确率，显著优于 GPT-4o、DeepSeek V3 和 R1 等基线方法。经过直接偏好优化对齐的 TcadGPT-P1&2+DPO 进一步展现出出色的代码可执行性，在工具级验证中的语法通过率达到 80%，同时模型足够精简，可在单个图形处理器上进行本地推理。

综上，这些贡献确立了 TcadGPT 作为一种可复现、适应特定领域的基础模型，可用于智能技术计算机辅助设计自动化和数字孪生开发。

4 实验设置与基准测试

为了严格评估 TcadGPT 的能力，我们构建了一个针对 TCAD 相关任务的基准测试，旨在反映半导体研究人员面临的真实挑战。该基准测试遵循三项核心原则：（1）确保覆盖 TCAD 工具链中的核心功能模块

具有代表性的技术计算机辅助设计问答示例

问题 1（概念性）： 什么是高场迁移率饱和效应?
答：高场迁移率饱和是指这样一种现象：在高电场作用下，载流子的漂移速度趋近于一个上限，即高电场。

Q2（语法）： 如何控制氧化生长速率 Sprocess?
答：使用命令 `pdbSet Diffuse MinGrowthStep` 和 `MaxGrowthStep` 来控制速率。

Q3（数学/物理公式）： 什么是 Shockley-Read-Hall（SRH）复合公式?
A:

$$U = \frac{pn - n_i^2}{\tau_p(n + n_1) + \tau_n(p + p_1)}$$

Q4（指令解析）： 如何定义边界 Sentaurus Device 中的条件?
答：使用命令 `BoundaryCondition (Electrode, Type, Value)`。

Q5（操作指令）： 如何在 Sen- 中加载数据 taurus 可视化?
答：使用菜单路径 `File → Load` 导入结果文件例如 TDR 文件。

问题 6（仿真配置）： 如何调整仿真精度?
答：设置参数 `Math{Numerics{DampedNewton{ Tolerance=1e-6}}}` 以提高精度。

图2：TCAD基准测试中的代表性示例，涵盖操作类、概念类和模型相关等多种题型。这些示例体现了高性能模型所需领域理解的多样性与深度。

；（2）涵盖多种问题类型，从概念理解到精确的语法构造；（3）将问题置于真实的用户场景中。最终的问题集包含 264 个问题，均经过精心筛选并由领域专家审核，以确保其技术准确性和代表性。

图3汇总了题型分布情况。该基准测试涵盖六个主要的技术计算机辅助设计功能模块：通用物理模型、通用仿真、SProcess、SDE、SDevice 和 SVisual。这些模块既涉及概念建模知识，也包含特定工具的操作；超过 65%（264 题中的 173 题）聚焦于指令解析、工具操作或语法构造——研究人员认为这些是最突出的难点。

所有回答均依据领域专家提供的参考答案，并使用标准化评分标准进行评估，以确保评分客观且一致。具体而言，每个基准测试条目均由领域专家编写，并附有明确无歧义的预期答案（例如，特定命令、参数含义、公式或简短的规范性说明），从而无需依赖主观判断即可核验正确性。评分标准仅用于处理可接受的等效表达（例如格式差异、同义词或

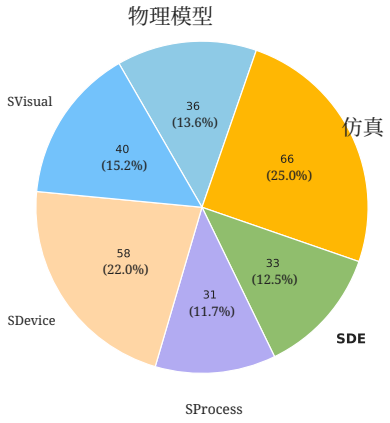


图3：TCAD基准测试中的题型分布。六个主要类别——物理模型、仿真、SDE、SProcess、SDevice 和 SVisual——涵盖了实际工作流程中遇到的所有关键仿真和建模任务。

轻微改写），同时保留相同的技术内容。对于以指令为导向的问题，只有当输出提供可直接使用且符合预期工具语法的命令或代码片段时，才会判为正确；对于以原理为导向的问题，只有当输出充分完整地符合参考结论，足以满足实际技术计算机辅助设计（TCAD）应用需求时，才会判为正确。

基准测试中的代表性示例如图2所示，展示了问题类型的多样性，包括概念定义、语法层面的命令、数学公式以及仿真配置。

检索增强生成设置

为研究外部知识检索是否能够提升模型性能，我们使用$LangFlow$ 平台构建了检索增强生成（RAG）流程。[1] 该流程结合了离线文档准备与在线查询处理。领域文档——包括 Synopsys Sentaurus 用户指南、TCAD 教材和官方培训材料——经过预处理，使用自定义分割器切分为易于处理的文本块，并通过Ollama嵌入（nomic-embed-text）生成嵌入向量。这些嵌入向量存储在 Chroma DB 中，以便进行高速语义搜索。

在推理阶段，首先使用轻量级翻译模型将用户查询翻译成英语，使其与领域语料库的语言保持一致。随后对检索到的文档块进行解析，并将其纳入结构化的上下文提示中，最后将该提示传递给目标大语言模型以生成回复。此设置为比较不同大语言模型的独立性能与其检索增强版本的性能提供了一种一致的方法。

用于评估的模型

我们评估了模型能力的两个互补方面：**(1)问答层面的理解** 和 **(2)代码层面的可执行性**。对于问答层面的评估，我们使用 LLaMA 3.1 8B 作为基础模型，并使用我们的合成问答语料库对其进行微调，得到三个变体：TcadGPT-P1（仅使用流程 1 的数据训练，即基于片段的问答生成）、TcadGPT-P2（仅使用流程 2 的数据训练，即关键词引导的问答生成），以及组合模型 TcadGPT-P1&2（使用两个流程数据的并集训练）。我们将这些变体与 DeepSeek V3、DeepSeek R1 和 GPT-4o 等通用模型进行了比较。为研究检索增强的影响，我们还测试了 LLaMA 3.1 8B + 检索增强生成、DeepSeek V3+ 检索增强生成和 TcadGPT-P1&2 + 检索增强生成。该设计可系统比较：(i) 通用大语言模型与领域专用大语言模型，以及 (ii) 独立运行与检索增强设置。

对于代码级可执行性评估，我们进一步评估了**经直接偏好优化对齐的模型** TcadGPT- P1&2+直接偏好优化，该模型在经问答微调的检查点基础上，使用了中间表示→直接偏好优化对齐数据集。与衡量科学正确性和描述性推理能力的问答基准测试不同，此项评估侧重于语法有效性和指令遵循精度，并直接在Sentaurus sde -S环境中进行验证。

可复现性与开放资源。 为确保透明度和可复现性，本研究的所有组件——包括包含264个问题的问答基准测试、20条指令的SDE可执行性测试集、微调模型权重，以及完整的数据生成流程（**P1**、**P2**和**中间表示→直接偏好优化**）——均已在GitHub上公开发布。[2] 该代码库提供了用于数据集合成、模型微调和基准评测的脚本，便于直接复现，并支持对面向技术计算机辅助设计的大语言模型开展进一步研究。

5 问答训练数据生成

由于公开可获取的TCAD相关代码和文档十分稀缺，我们从官方公开资源中整理训练语料，包括Sentaurus用户指南、TCAD教材、培训手册和随附的示例文件。所有文档均

[1] https://github.com/langflow-ai/langflow

[2] https://github.com/wdddddds1/TcadGPT

取自公开来源，并仅用于非商业学术用途。

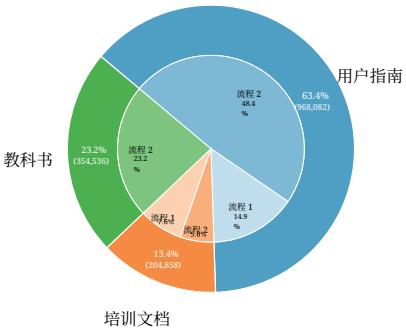


图 4：按数据来源（用户指南、培训文档、教科书）和数据生成流程（流程 1、流程 2）划分的合成训练数据集构成。外环显示各来源的总贡献，内环则区分两种问答生成流程。

为系统地构建高质量的 Alpaca 格式指令—响应数据集[23], ，我们设计了两种互补的流程，以最大限度地提高领域知识的覆盖率和多样性。图 4 展示了训练集的详细构成，其中问答对总数超过 150 万。外环表示各类文档的相对贡献，内环则按生成流程进一步细分数据集。值得注意的是，大多数样本来自用户指南和教科书，流程 2 占整个语料库的 80% 以上。

为展示生成数据的特点，图 5 列出了我们流程生成的典型 JSON 格式问答对。

流程 1：基于片段的问答生成我们首先使用开源光学字符识别工具将源 PDF 文档转换为文本，然后依据章节标题和段落分隔等结构标记进行分段。随后，我们将这些片段输入大语言模型（DeepSeek V3），以生成指令遵循问答对。为减轻模型的措辞敏感性，

JSON 问答对示例

```
{
"instruction": "What components are included in the total resistance calculation of
metal interconnects?",
"input": "",
"output": "The total resistance of metal interconnects includes all serially
connected conductor segments. Specifically in the example:
-resistance of the metal1 segment
-resistance of the metal2 segment
-resistance of the via (connecting metal1 and metal2)
These elements together form the overall interconnect impedance characteristics
between cont1 and cont2."
},
{
"instruction": "What is the purpose of the 'moderesistance' command?",
"input": "",
"output": "The 'moderesistance' command sets the simulator to operate in a purely
resistive network mode for interconnect structures. In this mode:
-capacitive and inductive effects are ignored
-focus is solely on DC impedance characteristics of the conductor network
-subsequent 'solve' commands will compute based on this mode"
}
```

图 5：用于微调的 JSON 问答对示例。

我们使用单独的大语言模型提示词，将每个问题改写为十种不同表述，以增强鲁棒性。此过程生成了 340,000 个样本，用于训练 TcadGPT-P1 模型。

算法 1 流程 1：基于片段的问答生成

输入： PDF 文档集合 $\mathcal{D}$
对于每个文档 d在 $\mathcal{D}$ **执行**
  $T \leftarrow$光学字符识别(d)
  $S \leftarrow$ 分段(T)
  对于每个片段 s在 S **执行**
    $q,\ a \leftarrow$ 大语言模型.generateQA(s)
    $Q' \leftarrow$ 大语言模型.paraphrase($q,\ n = 10$)
    将 $(q,\ a)$ 和每个 $(q',\ a)$ 添加 Q' 到数据集
  结束循环
结束循环

流程 2：关键词引导的问答生成为提高覆盖率和领域精确度，我们设计了第二种流程，从每个片段中提取技术关键词（如材料参数、模型名称和方程），并利用这些关键词引导问答生成。针对每个（片段，关键词）配对，我们提示大语言模型生成相关且聚焦的问答实例。该方法生成了超过 120 万个多样化样本，并用于训练 TcadGPT-P2 模型。

算法 2 流程 2：关键词引导的问答生成

输入： PDF 文档集合 $\mathcal{D}$
对于每个文档 d在 $\mathcal{D}$ **执行**
    $T \leftarrow$光学字符识别(d)
    $S \leftarrow$ 分段(T)
    对于每个片段 s 在 S **执行**
        $K \leftarrow$大语言模型.extractKeywords(s)
        对于每个关键词 k 在 K **执行**
            $q,\ a \leftarrow$ 大语言模型.generateQA($s,\ k$)
            将 ($q,\ a$) 添加到数据集
        结束循环
    结束循环
</style}结束循环

提示工程与训练设置 为确保合成指令—响应对的质量、覆盖率和可靠性，我们设计了一个多阶段提示工程框架。对于流程 1，提示要求大语言模型从每个文档片段中提取所有具有技术意义的内容，重点关注仿真命令、建模参数和领域逻辑。该提示要求严格遵循格式规范（Alpaca风格JSON），避免浅显或重复的问题，并强调答案应简洁且具有可操作性，在适用时包含公式或代码。为提高语言鲁棒性，我们还使用单独的提示，为每个问题生成 10 种不同的释义改写。

流程 2采用了两步提示结构。第一步从每个片段中提取所有相关关键词，例如模型名称、命令名称或物理原理。第二步利用每个（片段，关键词）对生成问答示例，并根据关键词类型（例如定义或指令）采用不同的风格。提示要求保持结构一致，避免使用笼统的指代（例如“如上所述”），并要求答案详尽且以示例为依据。表1概述了提示类型及其预期目标。

由两种流程生成的数据集分别包含约 34 万条样本（流程 1）和 120 万条样本（流程 2）。鉴于流程 1 最初生成的 34 万条样本数据集性能有限，因此提出流程 2，以提升

表1：提示类型和目标概述

提示类型	**目标和关键约束**
分段生成 QA	提取领域逻辑；强制采用 Alpaca 格式；强调命令、方程或流程级问题；最大化覆盖率；仅限中文。
释义改写用于问答	为每个问题生成 10 种不同的表述以提升模型的鲁棒性；严格禁止非-冗余项及有效列表的——mat.
关键词提取 Ex-	识别所有技术概念（命令、参数、模型）；强制输出严格的 JSON 结构；忽略非技术性内容
从关键字生成问答——词	页面。按关键词类型设计问题；禁止使用上下文引用；确保回答简洁且信息丰富并提供示例。

覆盖率和多样性，从而形成规模更大、粒度更细的 120 万条样本语料库。模型训练使用 $8\times$ 台 A100 80GB 图形处理器，采用 LLaMA 3.1 8B 模型和搭配 DeepSpeed Stage 3 的 LLaMA-Factory。训练采用全参数微调，最大序列长度为 4096 个词元，学习率采用余弦衰减，初始值为 1×10^{-5}；每个图形处理器的批次大小为 4，梯度累积步数为 8，总训练轮次为 5。所有训练均采用 bf16 精度，并留出数据集的 2% 用于验证。

6 代码增强和DPO 数据集构建

问答式监督提供了广泛的领域知识，但很少能教会模型在生成可执行的 TCAD 脚本时准确遵循指令。仅使用问答数据或非结构化代码训练的模型，往往会生成看似正确却无法使用的输

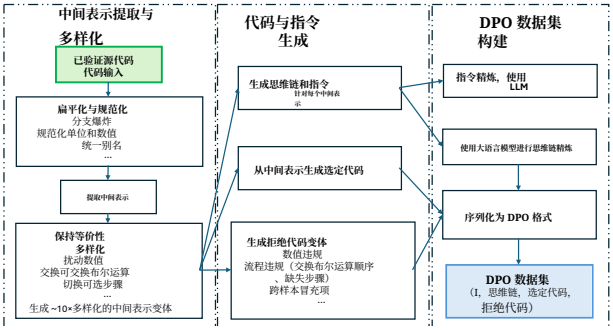


图 6：**中间表示概览→DPO 对齐流水线**。该流水线以经过验证的技术计算机辅助设计源文件组为起点，提取模式级中间表示（中间表示），执行保持等价性的多样化，生成规范化指令和参考代码，并构造包含受控数值违规和流程违规的偏好对，以进行直接偏好优化（DPO）。

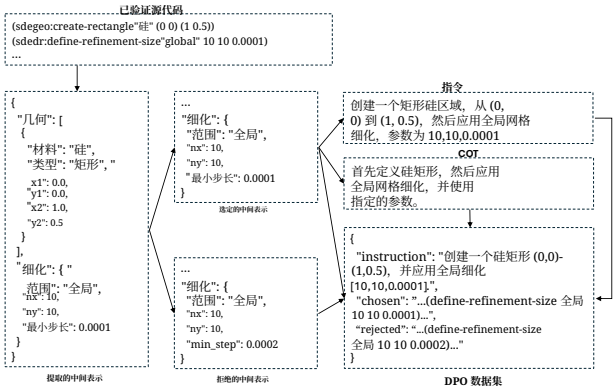


图 7：**由中间表示驱动的 DPO 配对构建示例**。已验证的 TCAD 源代码片段被解析为中间表示（IR），并据此确定性生成规范自然语言指令、一个被选中的（符合指令的）代码样本，以及一个带有单一受控违规（例如数值不匹配）的被拒绝变体。

出结果——命令顺序错误、缺少导出步骤，或参数相差一个数量级。为弥补这一差距，我们提出了一种**以代码为中心的对齐流程**，将少量经过验证的仿真文件转换为大型且可解释的数据集，用于直接偏好优化（DPO）。图 6 展示了完整的 IR→DPO 对齐工作流，包括 IR 提取与多样化、指令和代码渲染，以及通过受控违规构建偏好对。

中间表示构建与保持等价性的多样化

从 ∼800官方 Sentaurus sde/sprocess/sdevice deck 出发，我们提取一种轻量级中间表示（*IR*），用于捕获仿真的关键事实：维度和向上方向、材料列表、包含有序布尔运算的区域几何结构、接触定义、掺杂规格以及网格/导出指令。每条记录还包含一张简洁的 事实卡片（区域计数、布尔顺序字符串、接触是否存在、预期输出），用于形式化规定转换过程中不得改变的不变量。

我们首先通过统一别名、规范化数值格式，并在工具语义允许的情况下将命令顺序规范化，来展平 deck，使比较反映语义而非风格。随后，我们在 IR 上执行**保持等价性的多样化**，在不改变物理含义的前提下扩大覆盖率：对可交换的布尔运算/几何操作进行适度重排，应用小幅度的单位感知数值扰动，规范化别名，并切换可选的网格/导出语句。这样可实现约 $\times 10$ 的扩展，同时保持可执行性和语义忠实性。

指令渲染、验证与 DPO 打包

每个多样化的中间表示被确定性地渲染为一条规范自然语言指令和一个参考代码片段。一个数值白名单将可见常数限制为中间表示或参考代码中已有的常数。为了在不冗长的前提下明确预期推理，我们附加一段简洁、对称的**思维链（COT）**，用以枚举几何、掺杂、网格和导出步骤。

为明确优化对指令的遵循，我们

算法3 IR到DPO对齐工作流

输入： 已验证的 TCAD 输入文件 $\mathcal{C}$
对于 每个输入文件 $c \in \mathcal{C}$ **执行**
    IR $\leftarrow$ 提取IR(c)
        // 维度、几何、掺杂、网格
    $\text{IR}_{\text{flat}} \leftarrow$展平(IR)
    $\{\text{中间表示}_i\} \leftarrow$ 多样化($\text{中间表示}_{\text{flat}}$)
        // 顺序交换、数值扰动、切换
    对每个中间表示$_i$ **执行**
        $I \leftarrow$渲染指令(中间表示$_i$)
            // 数值白名单
        $c^\star \leftarrow$渲染代码(中间表示$_i$)
            // 选定代码
        $\{\tilde{c}_j\} \leftarrow$ 生成拒绝项($\text{IR}_i, c^\star$)
            // 受控违规
        验证配对($I, c^\star, \tilde{c}_j$)
        序列化 DPO($I, c^\star, \tilde{c}_j$)
    结束循环
结束循环

构建**DPO 偏好对**。对于每条指令，经验证的参考代码都是优选样本。我们合成若干劣选变体，使其有意违反一个可解释的维度，包括：(i)数值偏差（近似值/步长/幅度/宽范围抖动，或$\times 10/\times 0.1$单位尺度错误）；(ii) 流程违规（例如，调换布尔运算顺序、将接触定义移到细化之前、遗漏构建网格或导出）；以及 (iii) 跨样本冒充项（来自其他记录、事实相互矛盾的有效代码）。两阶段检查器会验证负样本：针对数字执行非子集多重集测试（并采用宽松接受的回退方案）针对结构则使用 IR 差异/定向正则表达式。

最后，我们将样本序列化为 JSON，并赋予稳定的标识符。所有配对共享相同的指令和思维链，唯一的区别在于围栏代码块内部，因此偏好信号不会受到风格伪影的混淆。我们还生成了五种改写后的指令变体（研究型/工程型/简洁型/分步型/对话型），并使用相同的白名单进行筛选，以提高鲁棒性。

示例说明。图 7 给出了 IR 驱动的 DPO 构建流程的具体示例。

从经过验证的 TCAD 源代码片段出发，利用提取的中间表示生成自然语言指令、参考（选定的）代码样本，以及一个仅在一处存在可解释违规的拒绝变体。

DPO 配对示例（节选）

指令： 在二维硅衬底上，从坐标 (0, 0, 0) 到 (1, 1, 0) 构建一个矩形区域。定义两个全局矩形窗口，并在 (0, 0.5, 0) 处放置一个虚拟接触点。在窗口内进行浓度为 9.8e+12 的硼掺杂，然后使用参数 [10, 10, 0 0001, 1, 1, 0 0001]

细化全局网格。.....最后，构建网格并导出 BND 和 TDR 文件，以供 SDevice 使用。
选定（摘录）：
(sdedr:define-refinement-size "全局" 10 10↩→ 0.0001 1 1 0.0001)

... (sdeio:save-tdr-bnd "n@node@.tdr" " n@node@.↩→bnd")
拒绝（摘录）： (sdedr:define-refinement-size "全局" 10 10↩→ 0.0001 1 1 0.0002) ... (sdeio:save-tdr-bnd "n@node@.tdr" "n@node@.↩→bnd")

7 实验结果与消融研究

,

为全面评估 TcadGPT 的有效性，我们设计了一个双层评估框架，涵盖**语义理解**和**代码可执行性**。

（1）问答级评估。我们构建了一个经过整理的基准测试，包含 264 个由专家验证的问题，涵盖六个关键的技术计算机辅助设计子领域——**通用物理模型、仿真、SDE、SProcess、SDevice**和**SVisual**。该基准测试衡量模型理解物理原理、解读工具语法，以及生成科学上正确且

实用回答的能力。我们根据科学正确性和语法有效性对每个答案进行人工评分，具体说明见第 4 节。

（2）代码级可执行性评估。为进一步评估我们的 IR→DPO 对齐工作流对可执行代码生成的影响，我们创建了一个独立测试集，其中包含从 IR 渲染器自动生成的 20 条未见过的自然语言 SDE 指令。对于每条指令，模型会生成一个或多个候选脚本，并在官方 Sentaurus 结构编辑器中使用 sde -S 语法检查进行直接验证。我们报告两项指标：（i）**Pass@1**，即通过语法检查的单样本生成比例；（ii）**Pass @3**，即每条指令最多生成三个样本时通过检查的比例。我们还区分了直接通过与需要简单占位符替换的情况（例如，将 @height@ 替换为数值常量）。

这项双层评估提供了互补的视角：问答基准测试量化模型的推理能力和指令遵循准确率，反映模型对知识的理解；而 SDE 测试集衡量模型在真实条件下生成工具可验证的可执行代码的能力，反映模型对知识的应用。

7.1 问答级基准测试结果

7.1.1 总体性能

图 8(a)总结了各模型在整个基准测试（264 个问题）中的总体准确率。在所有模型中，TcadGPT-P1&2（使用组合数据管线训练）取得了最高准确率 85.6%，显著优于通用大语言模型基线方法，包括 GPT-4o[24]（46.6%）、DeepSeek R1[25]（52.3%）和 DeepSeek V3[26]（50.0%）。此外，我们观察到，检索增强生成（RAG）显著提升了模型性能：DeepSeek V3 的准确率从 50.0 % 提升至 67.0%，LLaMA 3.1 8B 则从 24.6% 提升至 58.0%。然而，值得注意的是，为 TcadGPT-P1&2 添加 RAG 后，准确率反而从 85.6% 降至 64.0%，这表明过于宽泛的检索上下文可能会削弱领域专用微调的效果。

7.1.2 子领域级分析

为深入了解模型表现，我们分别评估了模型在六个 TCAD 子领域中的表现（图 8(b)）。

通用物理模型：TcadGPT-P1&2 的表现近乎完美（93.9%），显著优于 GPT-4o（90.9%）和 DeepSeek V3+RAG（87.9%）等通用模型。在**通用物理模型**类别中，TcadGPT-P1（10.6%）和 TcadGPT-P2（92.4%）之间明显的差距，反映的是数据覆盖范围的差异，而非优化稳定性。TcadGPT-P1 主要基于片段级提取数据进行训练，这些数据偏重工具使用和脚本模式；因此，它经常试图通过输出操作指令或仿真器命令来回答概念性物理问题，而根据我们的评分标准，这些回答会被判为错误。从本质上说，TcadGPT-P1 变得过于倾向于生成代码——面对一般物理问题时，输出类似指令的内容，而不是描述性推理。相比之下，流程 2 通过关键词引导的问答合成生成，明确针对物理机制、材料参数和建模假设，因此能够给出准确度较高的描述性回答，而非指令式的代码补全。

仿真：TcadGPT-P2 和 TcadGPT- P1&2 均取得领先表现（86.1%），优于采用检索增强生成的通用模型（DeepSeek V3+RAG 为 80.6%），并显著超过 GPT-4o（61.1%）和 LLaMA 3.1 8B（36.1%）等基线模型。

SDE：在网格和几何设置相关任务中，TcadGPT-P2 和 TcadGPT-P1&2 取得最佳结果（93.9%），展现出显著优于通用模型的表现；其中表现最好的通用模型（LLaMA 3.1 8B+检索增强生成）仅达到 60.6%。这一鲜明对比凸显了复杂的工具专用语法，而领域适配模型尤其擅长处理这类语法。

SProcess：同样，TcadGPT-P1&2（83.9%）和 TcadGPT-P2（80.6%）的表现也显著优于所有通用

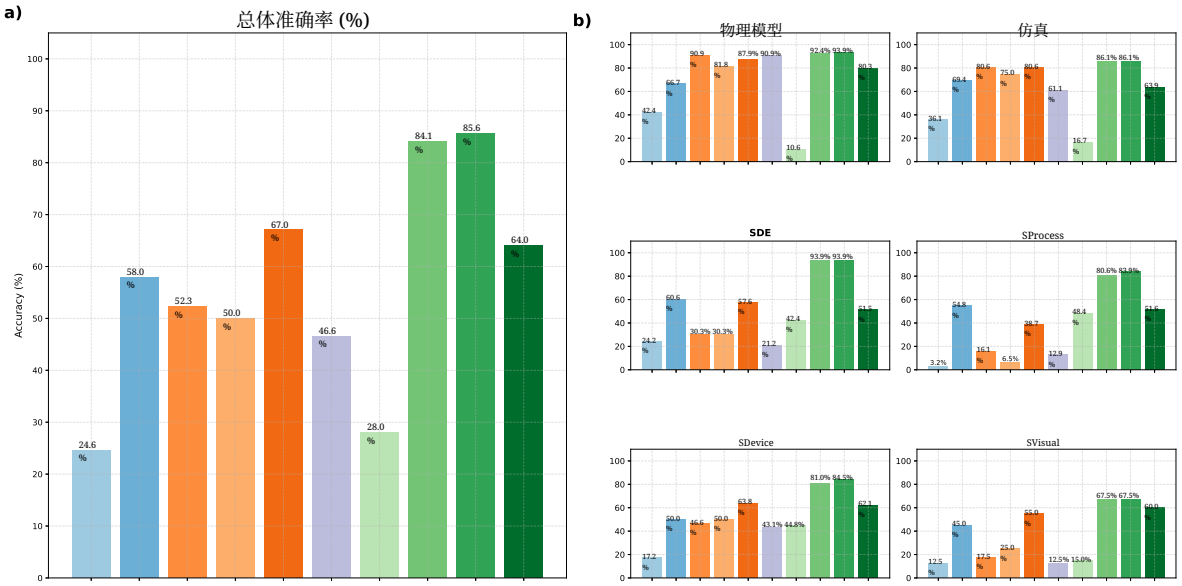


图8：模型在全部六个 TCAD 子领域及整体基准测试（264 个问题）中的表现。左：所有模型的总体准确率对比。右：**通用物理模型**、**仿真**、**SDE**、**SProcess**、**SDevice**和**SVisual**的子领域准确率。

模型，其中表现最佳的通用模型（LLaMA 3.1 8B+检索增强生成）为 54.8%。这表明领域专项训练十分重要，尤其是在氧化和扩散等复杂仿真任务中。

SDevice：TcadGPT-P1&2 和 TcadGPT-P2 的表现始终优于其他模型，准确率分别达到 84.5% 和 81.0%。通用模型则明显表现不佳，GPT-4o（44.8%）和 DeepSeek V3+RAG（63.8%）的低分凸显了在缺乏专门训练的情况下，正确解析详细器件仿真命令和参数的难度。

SVisual：可视化任务面临着类似的挑战，但 TcadGPT-P2 和 TcadGPT-P1&2 的准确率表现出色（67.5%），超过了 DeepSeek V3+RAG（55.0%）和 GPT-4o（15.0%）等通用基线。数据表明，可视化语法虽然稍微更直观，但仍然非常受益于专门的领域数据。

7.1.3 消融研究

我们的消融研究比较了多种配置，以分别考察检索增强生成（RAG）和领域专用微调的作用。通用模型在结合检索后性能显著提升（例如，LLaMA 3.1 8B 从24.6%提升至58.0%，DeepSeek V3 从50.0%提升至67.0%），验证了检索增强在提供必要上下文依据方面的有效性。相反，在已经专门化的 TcadGPT-P1&2 中加入检索后，整体性能从85.6%下降至64.0%，这表明流程1和流程2提供的精心整理的训练数据已经充分覆盖领域知识，而过于宽泛的检索上下文可能会引入噪声，而非提供有价值的洞见。

此外，分别比较 TcadGPT-P1 和 TcadGPT-P2 后发现，结

构化的关键词问答生成（流程 2）显著提升了模型理解并正确回答所有模块中复杂技术计算机辅助设计相关问题的能力，凸显了有针对性且结构化的领域数据的重要性。

这些结果共同表明，量身定制的领域专用训练方法，尤其是结构化且由关键词引导的数据增强，对于大语言模型在技术计算机辅助设计等高度专业化的低资源领域中发挥效用至关重要。

总结 流程 2 在概念类任务和代码密集型任务上均带来了稳定且显著的提升，大幅优于所有基线方法。这些改进凸显了有针对性的提示设计和关键词引导的覆盖率对于合成领域专用问答数据的价值。

我们的结果进一步表明，领域专用微调比单独使用检索更能奠定坚实基础：检索增强生成显著提升了通用模型的表现，但在已专门化的 TcadGPT 上简单加入检索增强生成反而会降低准确率。这表明，非结构化检索可能会削弱已学得的领域先验；未来工作应探索受约束或具备模式感知能力的检索方法，在补充专门化能力的同时避免引入噪声。

7.1.4 错误分析

我们对 TcadGPT-P2 的失败案例进行了定性分析，以识别仍然存在的挑战。观察到三类主要错误：

(1) 无效引用： 一些回答包含对上下文中并不存在的方程或图的幻觉引用。这些引用通常是从训练文本中继承而来，并在生成时被错误地保留。例如：

无效的方程引用

用户查询： 瞬态仿真和小信号分析对应的方程是什么？

模型回答： 瞬态仿真对应的方程是**方程 12**，它描述了基本的瞬态公式；小信号分析对应**方程 48**，描述了小信号行为的基本原理。

在此案例中，模型提到了“方程 12”和“方程 48”，尽管输入上下文中并未定义这两个方程。我们引入了更严格的提示词来抑制此类错误引用，取得了一些改进，但仍有部分情况存在。

(2) 命令幻觉与误解： 对于较少见的任务或训练覆盖范围有限的工具，模型有时会生成错误或捏造的命令语法。以下是一个与 SProcess 中掺杂剂定义相关的示例：

Sprocess 中的幻觉命令

用户查询： 如何在 Sprocess 中定义新的掺杂剂种类？**正确答案：** 使用命令 implant species=<name> <material>，并可选择指定 imp.table 来定义注入表。使用 damage 标志以启用损伤计算。**模型回答：** 在 Sprocess 中，使用掺杂剂命令：dopant add <name><新掺杂剂名称> <species> <现有掺杂剂名称>

在这里，模型编造了一个不存在的掺杂剂命令，这可能是由于模式混淆，或与其他模拟器语言中的短语部分重叠所致。

（3）分词和 OCR 噪声： 一些错误源于训练数据问题，例如光学字符识别伪影、损坏的 LaTeX 方程或被拆分的命令行。尽管我们应用了清理启发式方法，但其影响仍局限于局部，并可能干扰解析

或令牌对齐。

尽管面临这些挑战，TcadGPT-P2 在结构保真度和领域特定语法方面始终优于基线模型。未来的工作可以考虑集成仿真感知验证器、结构化检索或抽象语法树（AST）级约束，以进一步减少幻觉。

7.1.5 RAG 导致的性能下降分析

尽管检索增强生成（RAG）通常有益于开放领域任务，但我们观察到，将 RAG 引入 TcadGPT-P1&2 后，性能显著下降。为阐明这一现象，我们分析了基准测试中的一个典型示例。

示例（基准测试 ID 78）：问题

问题： 瞬态仿真和小信号交流分析分别对应哪些方程?

TcadGPT-P1&2

瞬态仿真使用描述器件动态行为的时变微分方程，其形式为

$$\frac{d}{dt}q(t) + f(t) = 0;$$

小信号交流分析采用频域线性化模型，例如导纳矩阵

$$Y = A + j\omega C,$$

该矩阵用于计算器件在小信号激励下的响应。

TcadGPT-P1&2+检索增强生成

在给定的上下文中，瞬态仿真对应方程 36，小信号交流分析对应方程 38。

DeepSeek V3 + 检索增强生成（用于比较）

根据所提供的上下文，瞬态分析和小信号交流分析分别对应以下公式。

小信号交流分析： 导纳矩阵描述为

$$\delta I = Y\,\delta V \quad \text{(Eq. 22)},$$

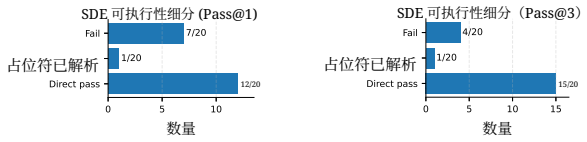
其中， δV 和 δI 分别表示所选节点处的复电压向量和电流向量。...

这一对比清楚地说明了为什么检索增强生成可能会降低紧凑型领域专用模型的性能。检索文档中包含来自 TCAD 手册的方程编号原文（例如“方程 36”“方程 38”），这误导了 8B 模型，使其照抄数字引用，而不是推理方程的物理含义。由于检索增强生成会直接注入未经筛选的文本片段，缺乏强大推理能力的较小模型往往会过度拟合词汇线索，而不是综合得出物理上有效的表达式。

相比之下，独立运行的微调版 TcadGPT-P1&2 依托其经过对齐的内部知识，能够生成一致、易于理解且可执行的公式。此案例表明，检索增强生成虽然能通过上下文召回和推理使大型通用模型（例如 DeepSeek V3）受益，但可能会损害轻量级领域专用模型的表现；在这类模型中，嘈杂或上下文不一致的检索结果可能同时影响指令遵循和代码可执行性。

7.2 代码级可执行性结果

尽管问答级基准测试评估了语义和推理准确性，但它无法直接反映模型生成的脚本是否具备工具可执行性。为量化 IR→直接偏好优化对齐流水线的实际影响，我们针对由 IR 生成器生成的 20 条未见过的 SDE 指令开展了语法级可执行性测试。每条指令都被转换为自然语言，并用作模型生成的输入。随后，通过直接调用 Synopsys Sentaurus 官方的 sde -S 语法检查器，对输出结果进行了验证



(a) (b)

图 9：**TcadGPT 的 SDE 可执行性细分TCADGPT。** (a) Pass@1 结果，展示直接通过、占位符解决后通过和失败的情况。(b) 三样本解码下的 Pass@3 结果。

无需任何人工修正。

需要注意的是，问答级和代码级评估是在两个不同的模型检查点上进行的。问答基准测试使用的是 TcadGPT-P1&2，该模型仅使用 150 万条 Alpaca 风格的问答对进行微调。相比之下，代码级可执行性测试评估的是 TcadGPT-P1&2+直接偏好优化模型，该模型进一步融入了基于 IR 的偏好对齐，以增强指令遵循能力和代码正确性。这种区分体现了两条评估路径各自不同的目标：语义理解与可执行代码生成。

评估协议。对于每条指令，我们都收集了单样本和多样本生成结果：

- **Pass@1**：首个样本生成结果无需修改即可通过语法验证的比例。
- **Pass@3**：在三次采样生成中，至少生成一个语法有效脚本的指令所占比例。

我们进一步将结果分为：(i) **直接通过**——代码无需修改即可编译；(ii)**占位符替换后通过**——代码包含模板变量（例如 @height@、@gate_len@），替换为数值常量后通过语法检查；(iii) **失败**——代码因语法或结构错误而被编译器拒绝。所有测试均使用相同版本和配置的 SDE 进行，以确保一致性。

表 2：20 条指令的 SDE 测试集上的语法级可执行性。Pass@k：生成代码通过 sde -S 的指令所占比例。占位符替换是指用数值常量替换模板变量（例如 @height@）。

模型	Pass@1	Pass@3
TcadGPT (P1+P2+直接偏好优化)	13/20 (65.0%)	16/20（80.0%）
DeepSeek V3	0/20（0.0%）	0/20（0.0%）

*TcadGPT*的细分结果：直接通过@1 = 12/20（60.0%）；占位符解析后，= +1 ⇒ Pass@1 = 13/20。直接通过@3 = 15/20（75.0%）；占位符解析后，+1 ⇒ 3 = 16 20 Pass@ / 。

结果与讨论。 如表2和图9所示，最终的 TcadGPT 模型在结合**P1+P2 问答语料库和 DPO对齐数据集**上进行微调后，在语法级SDE可执行性基准上 **Pass@1 达到 65.0%**（20 条指令中有 13 条）。如图9(a)所示，大多数成功案例均直接通过，无需任何修改（12/20）；另有一个案例在进行简单的占位符替换后通过。

每条指令采样三次生成结果（**Pass@3**）时，总体通过率进一步提高至**80.0%** （16/20）。重要的是，如图9(b)所示，这一提升主要来自直接通过数量的增加（15/20），而非依赖占位符解析，表明多样本解码提高了结构鲁棒性。

相比之下，DeepSeek V3 尽管生成的输出表面上类似于 SDE 脚本，却在全部20个测试用例中均未通过语法验证（**0% Pass@1 /Pass@3**），这证实其生成的代码缺乏结构正确性和命令级一致性。

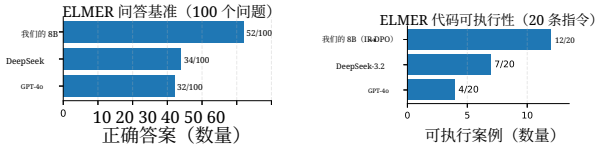
SDE 测试集。 该包含 20 条指令的 SDE 测试集旨在全面评估模型生成脚本在具有代表性的 TCAD场景中的可执行性。它涵盖二维和三维结构、多种布尔运算模式（ABA、BAB）、材料堆叠（硅、SiO_2, Si_3N_4, 多晶硅、GaN、AlGaN 等），以及一系列过程命令，包括接触定义、掺杂布置（恒定掺杂和

高斯掺杂）、网格细化和导出指令。这些指令通过 IR 生成器自动转换为自然语言，确保其在训练期间未被模型见过，同时在 Synopsys SDE 语法空间内保持完全有效。这种结合几何、掺杂和网格划分任务的组合，为评估真实 TCAD 工作流中的语法级可执行性 提供了实用且具有挑战性的基准测试。

为何需要直接偏好优化。在采用直接偏好优化（DPO）之前，我们系统地探索了几种基于标准监督微调的对齐策略。仅使用 Alpaca 风格问答语料库训练的模型，即使加入思维链（COT）推理，也表现出较差的指令遵循能力，并且经常生成几乎正确但无法执行的代码：命令顺序有误、网格参数缩放不当，而且经常遗漏导出指令。无论提示格式或数据规模如何，这些问题始终存在，表明单纯的监督微调或思维链监督不足以培养严格的语法规范性。在经过问答微调的基础模型上引入直接偏好优化至关重要，因为它直接优化模型，使其偏好遵循指令的输出，同时惩罚数值或流程上的偏差。这一转变带来了明显且可复现的可执行准确率提升——在 SDE 测试集上达到 80% 的语法通过率，便是有力证明。

这些结果表明，在问答微调（P1+P2）的基础上进行直接偏好优化对齐，能够显著提升指令遵循能力和语法准确性。仅进行问答微调虽能改善通用推理能力，却无法提升形式语法；直接偏好优化阶段则通过惩罚数值和流程偏差（例如网格参数错误、导出遗漏）明确优化可执行一致性。因此，最终模型能够理解仿真意图，并生成可直接由商业技术计算机辅助设计工具编译的脚本。

范围与局限性。 本评估聚焦于 SDE 环境中的**语法级可执行性** 。未纳入运行时验证（例如完整的网格构建和 TDR/BND 导出）。样本



（a）100 个问题上的问答准确率（Pass@1）。

（b）20 条指令上的可执行案例（Pass@1）。

图10：**在 Elmer 上进行轻量级跨领域检验**。采用相同的问答合成和中间表示→直接偏好优化对齐方案后，我们的 8B 模型在问答能力和求解器级代码可执行性方面均优于通用大语言模型。

规模（20 条指令）有限，但涵盖了具有代表性的多种几何和掺杂场景。未来的工作将把评估扩展至 SProcess 和 SDevice 模块，将每种工具的覆盖率提高到 100 条以上指令，并引入**端到端执行成功率（Pass@k）** 指标，以衡量固定环境下求解器成功完成的运行。

总结。代码层面的实验表明，将基于问答的微调（P1+P2）与 IR→DPO 对齐相结合，可显著提升指令遵循保真度和语法层面的可编译性，使其能够适用于商业技术计算机辅助设计工具链。在80% 的 sde -S 通过率下，TcadGPT 不再局限于描述性辅助，而是成为能够生成工具可接受的SDE 脚本的实用工具；相比之下，通用基线模型 DeepSeek V3 在此设置下未能通过任何基本语法检查。

7.3 Elmer 上的轻量级跨领域演示

7.3.1 设置

为评估我们的模式优先对齐框架在技术计算机辅助设计之外的可移植性，我们在开源有限元求解器 *Elmer* 上开展了一项轻量级跨领域研究。与专用半导体仿真器技术计算机辅助设计工具不同，*Elmer* 是一款通用多物理场有限元求解器，其问题表述、数值工作流和脚本

接口都截然不同。因此，它适合作为探究跨领域泛化而非近领域迁移的测试平台。

我们有意将本研究的规模控制得较小，并作为补充性研究。我们的目标并非建立一个在深度上可与技术计算机辅助设计套件相媲美的基准，而是验证所提出的对齐方案能否以最少的领域特定工程投入付诸实施。

我们沿用技术计算机辅助设计场景中的同一两阶段方案：(i) 使用流程 2 从领域文档中进行问答合成，随后对一个8B基础模型进行全参数微调；以及 (ii) 采用中间表示→直接偏好优化式对齐工作流，以提升指令遵循能力和脚本可执行性。所有问答训练数据均根据公开的 Elmer文档自动生成，无需使用专有语料库。

需要指出的是，Elmer 实验中使用的 DeepSeek 基线对应 DeepSeek V3.2，而技术计算机辅助设计实验使用的是 DeepSeek V3。这一差异源于研究期间官方API的变更，不影响本文报告的定性比较。

7.3.2 问答评估

对于问答级评估，我们基于相同的文档来源手动构建了一个包含100道题目的Elmer测试集。这些问题被有意限制为单一事实题目，每题均只有一个明确无歧义的答案，涵盖求解器配置、边界条件、材料规格以及常见的脚本模式。该设计避免了多步骤推理，并将评估重点隔离在事实回忆和命令级理解上。

如图 10a 所示，在 Pass@1 评估中，经过 Elmer 适配的 8B 模型取得了 $\mathbf{52/100}$，优于 DeepSeek V3.2 的 $\mathbf{34/100}$ 和 GPT-4o 的$\mathbf{32/100}$。这一稳定的优势表明，仅通过领域特定问答合成，就能在结构迥异的求解器领域中显著超越通用大语言模型。

7.3.3 代码可执行性评估

我们还使用 20 条未见过的自然语言指令评估代码级生成能力。对于每条指令，模型都会生成一个 .sif/.jis 格式的 Elmer 求解器输入文件，并直接使用 ElmerSolver 执行，而不依赖单独的仅语法检查器。

在执行之前，我们会自动从模型输出中剥离所有非.sif内容（例如解释或Markdown），并确保所引用的网格文件存在，具体方式包括更正网格数据库指针，或通过ElmerGrid生成网格。不会对求解器命令或数值参数进行手动修改。

如图10b所示，经过IR→直接偏好优化对齐的8B模型在Pass@1下生成了$\mathbf{12/20}$个可执行案例，相比之下，DeepSeek V3.2为$\mathbf{7/20}$，GPT-4o为$\mathbf{4/20}$。尽管规模较小，这一可执行性差距与图10中的问答趋势相呼应，并持续表明，所提出的对齐方案能够超越技术计算机辅助设计，迁移到不同数值与脚本范式下的求解器级代码生成任务中。

我们注意到，TCAD 与 Elmer 之间的可执行率并不能直接比较。TCAD脚本在语法和数值层面具有更紧密的耦合，并采用更严格的前端验证机制，因此，轻微的偏差就可能在解析或预处理阶段导致提前拒绝。相比之下，Elmer 采用更为宽松的键值式输入，并将部分一致性检查推迟到后续求解器阶段。因此，绝对可执行水平不仅反映模型能力，也反映了语言严格性和工具链行为方面的固有差异。因此，Elmer 研究旨在探究可移植性，而不是对不同领域的难度进行标准化。

8 结论与未来工作

我们提出了一种**模式优先的对齐框架**，用于在数据稀缺的情况下构建可执行的领域专用大语言模型。该框架以技术计算机辅助设计（TCAD）为应用领域，实例化为 TcadGPT，结合大规模问答合成（150 万对）与由 IR 驱动的直接偏好优化对齐，直接优化指令遵循保真度和语法有效性，从而生成可被工具接受的代码。

在一项包含 264 个问题的基准测试和一项包含 20 条指令的 SDE可执行性测试中，TcadGPT 在官方 sde -S 检查器下达到**85.6%** 的语义准确率和**80.0%** 的语法通过率，显著优于在此任务中成功率明显更低的通用大语言模型。我们还发现，检索增强生成（RAG）能显著提升通用模型的表现，却可能略微降低已完成领域专门化的紧凑模型的表现，这表明有必要采用模式约束的检索策略。

超越技术计算机辅助设计，我们表明，同一套对齐方案——基于专家材料的问答合成、以模式优先的 IR 提取与多样化，以及 IR→ 直接偏好优化对齐 ——只需稍作修改即可应用于开源有限元法求解器，并持续提升脚本级成功率。这些结果表明，对于要求产物有效性严格、数据稀缺的科学与工程领域，所提出的框架能够捕捉一种可复用的工具可执行大语言模型对齐模式。

局限性。我们的评估有意强调**语法级可执行性**（工具接受度），而非**运行时**结果，例如数值收敛或端到端完成。后者可能高度依赖模拟器特定稳定性、网格划分策略和求解器配置，因此当前基准测试无法将其完全分离。此外，现有系统尚不能稳健地执行多步规划或交互式修复；在结构较弱的检索设置下，也可能残留幻觉。

未来工作。 未来的研究方向包括：**(i)与商业求解器进行闭环集成**，以实现运行时验证和自动修复；**(ii)** 基于收敛信号、物理一致性检查或明确的工具反馈进行**奖励建模**； (iii)改进**规划与分解**机制，以支持多阶段工作流；以及 **(iv)模式感知检索** ，通过限制证据注入来避免词汇过拟合，同时保留事实依据。

参考文献

[1] S. Salahuddin、K. Ni 和 S. Datta，《电子学超大规模扩展时代》，《自然·电子学》，第 1 卷，第 8 期，第 442–450 页，2018年。

[2] W. Cao、H. Bu、M. Vinet、M. Cao、S. Takagi、S. Hwang、T. Ghani 和 K. Banerjee，“未来晶体管”，《自然》，第620卷，第7974期，第501–507页，2023年。

[3] K. Mehta 和 H.-Y. Wong，“使用带有自编码器的机器学习预测鳍式场效应晶体管的电流-电压和电容-电压曲线”，《IEEE电子器件快报》，第 42 卷，第 2 期，第 136–139 页，2021年。

[4] S. Myung、W. Jang、S. Jin、J. M. Choe、C. Jeong 和 D. S. Kim，“重构技术计算机辅助设计系统：让传统技术计算机辅助设计焕发新活力”，载于*2021*年*IEEE*国际电子器件会议（*IEDM*）。IEEE，2021年，第 18.2.1–18.2.4 页。

[5] L. Li、M. Agrawal、S. Y. Yeh、K. T. Lam、J. Wu 和 B. Magyari-Köpe，“面向基于原子和神经网络方法的精准高效工艺仿真”，载于*2022* 年国际电子器件会议（*IEDM*）。IEEE，2022年，第 15.6.1–15.6.4 页。

[6] S. K. Mitusch、S. W. Funke 和 M. Kuchta，“混合有限元法-神经网络模型：结合人工神经网络与有限元法”，计算物理学杂志，第 446 卷，页码 110651，2021年。

[7] B. Kim 和 M. Shin，“一种基于物理信息机器学习的新型神经网络器件建模方法”，*IEEE*《电子器件汇刊》，第 70 卷，第 11 期，第 6021–6028 页，2023年。

[8] A. Lu、Y. F. Chau 和 H. Y. Wong，“利用物理信息神经网络预测训练范围外的技术计算机辅助设计解决方案，同时最大限度减少领域专业知识需求”，载于*IEEE*电子器件技术与制造会议（*EDTM*）论文集。IEEE，2025年，已录用待发表。

[9] T. Wu 和 J. Guo，“采用机器学习方法对二维材料场效应晶体管进行多目标设计”，*IEEE*《电子器件汇刊》，第68卷，第11期，第5476–5482页，2021年。[10] H. Xu、W. Gan、L. Cao、C. Yang、J. Wu、M. Zhou、H. Qu、S. Zhang、H. Yin 和 Z. Wu，“利用机器学习方法优化堆叠纳米片晶体管的沟道几何结构和源极/漏极掺杂分布”，*IEEE*《电子器件汇刊》，第69卷，第7期，第3568–3574页，2022年。[11] H. Jeong、J. Choi、H. Cho、S. Woo、Y. Kim、J.-T. Kong 和 S. Kim，“由Mobo驱动的先进3纳米以下器件优化，以提升功率-延迟积性能”，*IEEE*《电子器件汇刊》，第71卷，第5期，第2881–2887页，2024年。[12]Z. He、H. Wu、X. Zhang、X. Yao、S. Zheng、H. Zheng 和 B. Yu，“ChatEDA：由大语言模型驱动的面向电子设计自动化的自主智能体”，载于*2023*年*ACM/IEEE*机器学习辅助计算机设计研讨会（*MLCAD*）论文集。IEEE，2023年，第待刊页。[13] K. Chang、Y. Wang、H. Ren、M. Wang、S. Liang、Y. Han、H. Li 和 X. Li，“ChipGPT：距离自然语言硬件设计还有多远”，*arXiv*预印本 *arXiv:2305.14019*，2023年。[14] Y. Lu、S. Liu、Q. Zhang 和 Z. Xie，“RTLLM：用于利用大语言模型进行RTL生成的开源基准”，*arXiv*预印本 *arXiv:2308.05345*，2023年。[15] C.-C. Chang、C.-T. Ho、Y. Li、Y. Chen 和 H. Ren，“DRC-Coder：使用大语言模型自主智能体自动生成DRC检查器代码”，载于*2025*年国际物理设计研讨会（*ISPD*）论文集。ACM，2025年，第待刊页。[16] Y. Jiang、X. Lu、Q. Jin、Q. Sun、H. Wu 和 C. Zhuo，“FabGPT：用于复杂晶圆缺陷知识查询的高效大型多模态模型”，载于2024年国际计算机辅助设计会议（*ICCAD*）。ACM，2024年，即将出版。[17] L. M. L. Nguyen等，“使用大语言模型生成 TCAD 结构输入文件”，载于 *SISPAD 2024*，2024年。[18] P. Råback、M. Malinen、J. Ruokolainen 和 T. Zwinger，“用于多物理场的 Elmer 有限元求解器”，*CSC – IT*科学中心，2012年。[19] 苏黎世联邦理工学院，“公益用途语言模型”，ETH新闻，2025年。[20] R. Jacobs 等，“迈向 AI 加速的量子化学：用于代码生成和仿真规划的 ORCA-LLM”，《*Digital Discovery*》，2024年。[21]H. Zhong 等，“LLM4EDA：用于电子设计自动化的大语言模型综合综述”，*arXiv* 预印本 *arXiv:2401.12224*，*2024*年。*[22] H. Liu* 等，“*ChipNeMo*：面向芯片设计的领域专用大语言模型”，arXiv 预印本 arXiv:2311.00176，2023年。[23] Y. Wang、Y. T. Peris、A. Bosselut、I. Gurevych 和 L. Zettlemoyer，“斯坦福 Alpaca：指令遵循型 LLaMA 模型”，*arXiv* 预印本 *arXiv:2305.10906*，2023年。[24] A. Hurst、J. Koon、S. McDonell 和 T. OpenAI，“GPT-4o系统卡”，*arXiv* 预印本 *arXiv:2410.21276*，2024年。 [在线]。地址：https://arxiv.org/abs/2410.21276[25] DeepSeek-AI，“DeepSeek-R1：通过强化学习激励大语言模型的推理能力”，2025年。[在线]。地址：https://arxiv.org/abs/2501.12948[26]A. Liu、Z. Chen、D. Guo、J. Song 等，“DeepSeek-V3：6710亿参数的混合专家语言模型”，*arXiv* 预印本 *arXiv:2412.19437*，2024年。 [在线]。地址：https://arxiv.org/abs/2412.19437

附录 A：Pipeline1 的提示模板

流程 1：生成 Alpaca 格式问答对的提示词

提示：生成问答对

你的任务是根据提供的文本生成高质量的微调数据。内容中会提供一段特定领域的技术文本。你需要按以下格式生成数据：{ "instruction": "用户指令。", "input": "", "output": "预期的系统输出。" } 问题必须与材料相关，专业且具有挑战性。重点关注命令、代码、公式、操作步骤、科学原理和软件逻辑。问题应多种多样，答案必须内容丰富、专业且结构清晰。输出应是对所生成指令的常规回应。确保指令具有普遍性，不局限于“本书中”之类的表述。必须严格遵循格式。提供的内容可能包含 HTML 标签或未渲染的 LaTeX 代码——仅输出人类可读文本，并确保公式正确渲染。

内容主要是技术手册或教科书 .
针对每个技术要点或命令提出一个问题。示例问题包括：

- 光学声子散射对迁移率有什么影响?
- 如何在 SProcess 中定义新的掺杂剂类型?
- 如何将模拟结构保存为 TDR 文件?
- 离子注入中的关键参数有哪些？如何配置仿真命令?

避免生成重复或含义相同的问题。每段内容至少生成 10 个不同的问题。所有内容都必须使用中文。

流程 1：用于扩充问题的提示词

提示词：扩充问题

你将收到一条来自微调数据集的问答对。你的任务是执行数据增强，将问题改写为不同的形式。输出应为一个包含 10 个新问题的 Python 列表。确保问题多样化。问题应同时参考原问题和答案。输出必须是 Python 可读取的列表：["问题1", "问题2", ..., "问题10"] 只输出列表，不要添加其他字符。必须使用中文。

附录 B：关键词引导流程 2 的提示模板

提示：提取关键词

你将收到一段英文或中文的技术 TCAD 段落。你的任务是提取所有可进一步查询的关键词，包括但不限于：

- 命令名称（例如，pdbSet、电极）
- 参数名称（例如，MinGrowthStep）
- 模型名称（例如，流体动力学模型）
- 软件模块名称（例如 Sentaurus De- vice）
- 物理机制或方程（例如热电子发射、载流子漂移）

输出格式必须为：
{ "keywords": ["keyword1", "keyword2", ...] }
严格遵循此格式。如果段落不包含技术内容，请勿提取关键词。保留每个关键词的原始语言，并确保覆盖完整。

提示词：根据关键词生成问答

你的任务：根据给定的技术段落和关键词，为 Alpaca 风格的微调生成结构化中文问答数据。输出必须是 JSON 数组。每个条目都遵循以下格式：{ "instruction": "用户问题", "input": "", "output": "系统回答" }

指南：

1. 仅使用中文生成。2. 如果段落缺少语义句子或与某个关键词相关的信息，则跳过该关键词并返回 []。3. 根据关键词类型选择风格：概念、公式、命令、模块或配置。4. 涵盖关键词的所有方面，避免重复。5. 答案必须准确、可执行，必要时提供代码或公式。6. 行内公式使用 $...$。7. 不要直接提及段落，也不要添加 Markdown 标记。8. 如果关键词是命令/参数，问题应描述其功能，而不是名称。