arXiv:2601.09428v1 [cs.LG] 14 Jan 2026

像欧几里得那样作图：教授Transformer模型使用尺规作图步骤生成CAD轮廓

Siyi Li、Joseph G. Lambourne、Longfei Zhang、Pradeep Kumar Jayaraman、Karl. D.D. Willis Autodesk Research

摘要

我们提出一种通过一系列简单的几何构造来生成计算机辅助设计（CAD）轮廓的新方法，其中包括曲线偏移、旋转和求交。这些构造序列以设计师提供的几何图形为起点，逐步构建最终轮廓中的点和曲线。我们证明，在设计师的输入几何与最终轮廓之间加入构造步骤，可以提升生成质量，其效果类似于在语言模型中引入思维链。与参数化 CAD 模型中的约束类似，构造序列将建模形状的自由度缩减至少量参数值；设计师可以调整这些参数，并对构造出的几何进行参数化编辑，同时确保其计算结果达到浮点精度。此外，我们还表明，对构造序列应用强化学习，能够在多种指标上进一步提升表现，其中包括一些未被明确优化的指标。

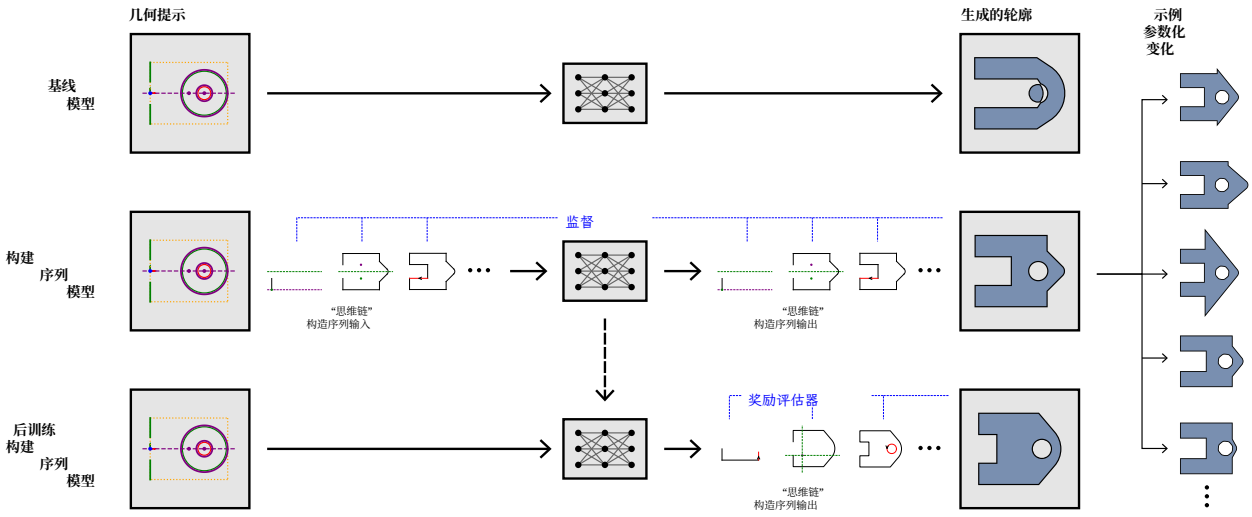


图1：我们的模型（中间）从设计师最初的几何“提示”（左）出发，通过一系列直尺、圆规和量角器构造，逐步生成计算机辅助设计轮廓，最终得到最终轮廓。与基线模型（上）相比，我们的方法生成的轮廓与设计师输入的匹配度更高，且自相交更少；基线模型省略了构造步骤，直接从几何提示映射到轮廓。我们还在后训练阶段进一步优化作图序列模型（下），利用奖励引导模型生成有效的构造序列。由于这些序列编码了参数关系，因此可以根据单个作图轨迹生成一系列相关轮廓（右）。

1 引言

计算机辅助设计（CAD）工具在几乎所有制造物品的设计中都发挥着关键作用。CAD 模型通过定义由直线、圆弧和圆组成的集合来创建，这些图形围成称为轮廓的二维区域。随后可以对这些轮廓进行拉伸或旋转，以定义实体体积；这些体积还可以通过布尔运算组合，从而构建复杂形状。可以为轮廓添加几何约束，以规定曲线之间的关系。例如，可以约束直线彼此平行、圆彼此同心，也可以强制曲线相切。还可以使用参数控制特定曲线之间的距离和角度，并通过调整这些参数来修改形状，同时保持设计中的对称性、规则图案和恒定厚度等关键特征。

近年来，用于生成二维计算机辅助设计几何的机器学习模型取得了长足进展，但这些方法存在两个主要局限。首先，目前所有方法的精度都有限。将几何表示为离散令牌序列（这些令牌表示量化后的点或坐标）来创建几何的方法，其精度受量化网格间距的限制 [1, 2, 3, 4, 5] ；而扩散方法的精度则受扩散过程的收敛情况和潜向量解码的限制 [6, 7, 8] 。其次，虽然有些方法能够预测约束和尺寸，但这是在几何生成之后作为第二步完成的。随后需要借助外部约束求解器对曲线几何进行后处理，以生成最终形状。研究表明，在后处理中应用生成的约束往往会使几何发生移动 [4, 3] ；如果这些变化显著改变了设计，这种情况就不理想 [9]。

在这项工作中，我们探讨一种生成二维轮廓的替代策略：使用统一序列同时定义几何和形状属性，而这些属性通常需要通过约束来强制实现。受近期语言模型研究的启发——研究表明，中间思维链（CoT）能够显著提高最终输出的精度 [10, 11], ——我们思考，生成某些中间几何是否有助于计算机辅助设计生成任务。我们注意到，计算机辅助设计设计师构建形状时，往往会先绘制一些中间的“构造几何”，以确定设计的重要要素，例如对称线或构造圆。随后，他们会在构造几何与轮廓曲线之间添加约束，以确保重合和对称性等属性。在几何约束求解器内部，由几何与约束构成的图会经过递归处理，生成一系列细粒度的简单几何构造，逐步构建出最终形状 [12, 13]。这些构造序列的作用类似于算法执行轨迹；后者已在各种搜索问题中发挥了显著作用 [14, 15, 16]。

本文通过实验，训练 Transformer模型，使其处理类似几何约束求解器所用方法的直尺、圆规和量角器作图步骤的中间构造序列。由于形状生成这一高层任务被拆解为具有闭式解的小型原子步骤，因此可以分别衡量网络解决这些子任务并将其组合起来构建一致的“CAD程序”的性能。通过在序列开头提供“提示几何”并将其作为后续作图步骤的输入，可以控制形状。生成后，这些序列可按浮点精度重放，使推理时已知的精确数值能够沿作图步骤传递，从而构建最终轮廓。生成的几何由少量数值参数化；重放序列时可以改变这些数值，从而对形状进行参数化编辑，如图1右侧所示。

我们通过定量实验表明，引入构造序列能够提升生成模型的性能，减少自相交，并证明其更能遵循设计要求。此外，我们还展示了，对整个构造序列应用强化学习可以进一步改善结果，如语言建模案例 [17, 18] 所示。

我们的贡献可概括如下

- 我们提出了一种领域专用语言，可将二维轮廓几何构建为由直尺、圆规和量角器构造步骤组成的序列。这些构造步骤可以以浮点精度重放，从而能够对形状进行参数化编辑。

- 我们表明，利用包含这些中间构造步骤的序列生成的几何图形，自相交更少；在自动补全部分设计时精度更高，也更符合对称线等其他设计要求。

- 我们展示了，采用会惩罚几何图形自相交的奖励函数进行强化学习，能够在多项指标上取得提升，其中许多指标并未被明确优化。

2 相关工作

2.1 算法轨迹

许多研究探索了训练神经网络模仿启发式算法执行的逻辑步骤。Vinyals等人 [19]展示了循环神经网络能够复现一些基础几何算法，例如求凸包和构建德劳内三角剖分。Yang 等人 [14] 尝试将蒙特卡洛树搜索轨迹用于迷宫导航、机器人操作和 Atari 游戏。Lehnert 等人 [15] 利用来自 A^* 搜索的轨迹学习解决迷宫问题，而 Gandhi 等人 [16] 则展示了如何利用搜索和回溯能力来玩 Count-down 游戏。在本研究中，我们探讨了这些技术应用于计算机辅助设计的可行性，并使用了与几何约束求解器所用轨迹类似的算法轨迹。

2.2 草图生成

大型约束草图数据集 [1, 3] 的出现，使 CAD 草图生成和草图自动约束成为社区关注的任务。Seff 等 [1] 提出了一种基于消息传递网络的自回归模型，通过迭代预测约束图的边和节点属性来生成参数化草图。随后需要借助外部几何约束求解器来构造最终几何，而生成草图的复杂度也受到限制。Willis 等 [2]展示了无条件生成二维草图的可行性：先生成点列表，再使用 PointerNetworks [19] 将这些点分组以定义曲线。此前的若干研究提出了这样的模型：首先生成二维曲线，然后在第二步中使用 PointerNetworks，根据这些几何信息预测约束和尺寸 [3, 4, 20]。由于这些架构只有在几何生成后才预测设计意图，因此无法利用约束来引导曲线的放置。相反，必须在后处理中应用外部约束求解器。这可能会改变草图几何 [3, 4, 9],的位置，从而表明初始曲线放置并未体现预期的设计意图。相比之下，本文提出的构造序列表示能够在构造几何之前预测设计意图，使网络在放置点和曲线时能够直接将其纳入考虑。

2.3 CAD 中的思维链

Text2CAD [21] 使用 DeepCAD 数据集 [22] 中的参数化配方，来定义对所用 CAD 建模特征的自然语言描述，以创建实体。CAD-Coder [23]使用 Deepseek-V3 将这些描述转换为自然语言思维链，随后生成 CadQuery 代码。研究人员使用这些数据对 Qwen2.5-7B-Instruct 模型进行了微调，并采用基于倒角距离的奖励函数和 GRPO 强化学习算法进一步提升结果。Seek-CAD [24] 使用 DeepSeek-R1 根据文本描述生成自然语言思维链和 CAD 命令。随后，将思维链与生成的 CAD 模型图像一同输入 Gemini-2.0，以便在迭代细化循环中提供视觉反馈。这些模型使用的自然语言思维链，是可执行代码的辅助表示。虽然其中包含与设计意图相关的陈述，但这些陈述并未以可由 CAD 内核直接执行的形式语言定义。

2.4 用于计算机辅助设计的强化学习

近期有多篇论文展示了将强化学习应用于各种计算机辅助设计相关任务的良好前景。Casey 等人 [9] 使用多种强化学习算法，对一个用于计算机辅助设计的自动约束模型进行了微调。结果表明，这提高了由约束完全定义的实体比例，同时降低了应用约束时几何发生移动的草图比例。RL-CAD [25] 研究了如何从边界表示模型中恢复参数化特征流程。一个演员-评论家网络会选择目标边界表示中的面进行拉伸或旋转，并根据恢复形状与目标形状之间的相似度给予奖励。CADCrafter [26] 使用直接偏好优化来提升图像到CAD命令序列网络的性能；Cadrille [27]则研究了利用强化学习提升从点云进行CAD重建的方法。

3 数据

3.1 数据集创建

尽管已有大规模 CAD 草图几何数据集 [1],，但这些数据集中的曲线通常不会构成闭合环，而这正是构建拉伸体或旋转体等下游建模操作所要求的。我们的训练数据由闭合轮廓组成，这些轮廓是从 ABC数据集 [28]中的 CAD 模型派生而来。我们使用 Open Cascade [29] 建模内核创建轮廓：将每个边界表示实体沿三个坐标系轴的法线方向，分别用 5 个等间距的剖切平面进行切片

。将不相交区域分开，使每个提取出的轮廓都包含一个外环和零个或多个内环。由此得到的闭合轮廓环由线段、圆弧和圆组成。数据去重过程将每个轮廓转换为一个图，其中节点为顶点，边为曲线。边根据曲线类型标注，节点则根据顶点坐标标注，并将坐标量化为 8×8 个分箱。随后计算 Weisfeiler-Lehman 图哈希 [30]，并移除哈希重复的轮廓。之后，数据按 95% 训练集、3% 验证集和 2% 测试集进行划分。

从原始轮廓几何中提取构造序列时，会用到一组简单的启发式算法，下面将对其进行简要介绍。每个算法阶段的详细说明见附录 B。为模拟设计师的输入，我们在序列开头加入用于形状控制的信息，并将其称为“几何提示”。不同于 [20]，中根据随机选取的部分草图几何自动补全草图，我们从凸包中提取线段，并提取内部圆环的位置。此外，还包括轮廓的面积和边界框，以及任何对称线。接下来，在“分析阶段”中，我们识别曲线之间的几何关系，例如平行线、同心圆和圆角弧。随后，这些关系会被转换为构造步骤，例如曲线偏移和圆角操作。然后，我们构建了一个二分数据流图，其中节点表示几何图形和构造步骤，有向边表示几何图形如何流入和流出各项操作。该图最初会包含循环和冗余分支，并在“图简化阶段”中将其移除。随后，通过字典序拓扑排序获得构造序列，并利用最终轮廓中曲线的顺序来消解拓扑排序顺序中的歧义。数据提取过程共得到 318,208 个轮廓及其对应的构造序列。

3.2 学习得到的序列

我们的实验使用由以下三个部分组成的序列。序列以“几何提示”开头，用于控制形状。接下来是构造步骤，它们类似于思维链：从提示几何开始，构造定义最终轮廓所需的点和曲线。最后是轮廓几何，类似于推理大语言模型给出的最终答案。

构造步骤表示一些简单的几何操作，例如曲线偏移、曲线交点、曲线反转、将点关于对称线镜像，以及构造圆角弧。表 1 提供了几种受支持的构造步骤示例，其余步骤列于附录 A.1 的表 7 中。每个构造步骤都包含一种操作类型、一组输入几何和一组输出几何。某一步骤的输出几何可以用作后续步骤的输入，从而逐步构建数据流图的描述。构造步骤的排列顺序使曲线按照其在轮廓中出现的顺序创建；其中，第一条曲线选取为终点最接近左下角的曲线。附录 A 介绍了用于编码这些序列的领域专用语言（DSL）和标记化方法。

4 方法

4.1 监督学习

我们使用交叉熵损失，在第 3.2 节所述的序列上训练一个仅解码器自回归 Transformer。实验中，我们使用 8 个注意力头、8 层深度、1024 的嵌入维度，以及 128 的注意力头维度。我们使用 Adam 优化器 [31]，学习率为 $3e-4$，随机失活率为 0.1。训练在 4 块 RTX 6000 GPU 上进行。

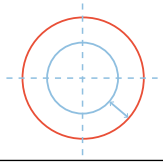
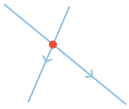
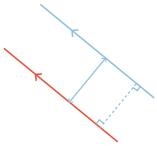
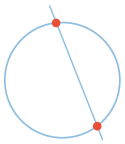
我们训练了该模型的两个变体。基线模型仅包含几何提示中的信息以及最终轮廓的几何信息；作图序列模型则额外包含中间构造步骤。第 5.3 节对这些模型的性能进行了比较。

4.2 强化学习（RL）

轮廓生成模型的微调可表述如下：对于几何提示 x，轮廓生成模型 $\pi_\theta(\tau|x)$ 会生成轮廓序列 τ。给定一组几何提示 $D=\{x_i\}_i^N$以及奖励函数 r，该函数会提供标量值 $r(x,\ \tau)$，用于评估轮廓序列 τ 的质量，以及其满足 x 所规定几何约束的程度。强化学习微调通过最大化期望奖励来进行

$$\max_\theta \mathbb{E}_{x\sim\rho}\mathbb{E}_{\tau\sim\pi_\theta(\cdot|x)}[r(x,\tau)], \tag{1}$$

表 1：尺规作图步骤。构造步骤的输入几何图形以蓝色显示，输出以红色显示。

描述	说明	示例
CircleOffsetCircle **输入：** 圆$_1$，　　偏移 **输出：** 圆$_2$	给定一个定向圆和一个正的偏移距离　　，求出并返回偏移圆　　.	
LineXLine **输入：** 直线$_1$，　直线$_2$ **输出：** 点	给定两条直线　，求出并返回它们的交点。	
直线偏移直线 **输入：** 直线$_1$，　偏移 **输出：** 直线$_2$	给定一条有向直线和一个偏移设置距离　，查找并返回直线从此直线向左侧偏移偏移距离。	
LineXCicle **输入：** 直线，　圆 **输出：** 点	给定一条直线和一个圆，找出并返回交点。	

其中，　ρ 表示几何提示的分布，　$\theta \in \mathbb{R}^d$ 表示轮廓生成模型的训练参数。

4.2.1 奖励设计

虽然基于人类反馈的强化学习（RLHF）往往涉及模糊且主观的人类偏好，但计算机辅助设计轮廓生成任务可以通过可测量的有效性指标，直接、客观地评估生成的轮廓序列，因此无需学习得到的奖励模型。强化学习所用的奖励定义如下：

- 语法有效的轮廓序列的奖励 τ:$r_{\text{无自相交}}$：不含自相交曲线的轮廓所占比例，$r_{\text{无短边}}$：不含短于预定义最小长度的边的轮廓所占比例，

- 语法无效的轮廓序列的奖励：
  $r_{\text{invalid profile}}$：对生成了语法无效的领域特定语言代码且无法反标记化的序列施加的惩罚。

4.2.2 策略梯度方法

我们重点关注三种序列级策略梯度方法：ReMax [32], GRPO [17], 和 RLOO [33]。这些方法使用生成序列的总对数概率来计算策略梯度，并采用带有学习得到的基线的 REINFORCE 风格估计量 [34]。与 PPO [35], 不同，后者在每个解码步骤使用单独的对数概率进行令牌级策略更新；这些方法在序列级运行，因此无需单独训练价值网络。我们使用这些方法对轮廓生成策略进行微调。

ReMax [32] 将策略网络贪心解码生成的序列的奖励用作基线，以对随机采样的序列奖励进行归一化。

从策略 $\pi_\theta(\tau|x)$中采样得到序列 τ，ReMax 基线 $b_{\theta,\text{ ReMax}}$为 argmax($\pi_\theta(\tau|x)$)，ReMax 的策略梯度目标为：

$$\mathbb{E}_{\tau\sim\pi_\theta(\cdot|x)}\Big[\big(r(x,\tau)-b_{\theta,\text{ReMax}}\big)\cdot\nabla_\theta\log\pi_\theta(\tau\mid x)\Big] \tag{2}$$

组相对策略优化（GRPO） [17] 针对每个几何提示 x采样一组包含 G 个轮廓序列 $\{\tau\}_{g=1}^{G}$ 的样本。第 g个轮廓序列的优势通过对组级奖励 $\{r(x,\ \tau_g)\}_{g=1}^{G}$进行归一化计算得出：

$$A_g = \frac{r(x,\tau_g) - \text{mean}(\{r(x,\tau_g)\}_{g=1}^{G})}{\text{std}(\{r(x,\tau_g)\}_{g=1}^{G})}. \tag{3}$$

GRPO策略梯度目标为：

$$\mathbb{E}_{\{\tau_g\}_{g=1}^{G}\sim\pi_\theta(\cdot|x)}\Big[\min(\psi_g A_g, \text{clip}\big(\psi_g, 1-\epsilon, 1+\epsilon\big)A_g) - \beta D_{\text{KL}}(\pi_\theta\|\pi_{\text{ref}})\Big], \tag{4}$$

其中：

$$\psi_g = \frac{\nabla_\theta \pi_\theta(\tau_g \mid x)}{\pi_{\text{ref}}(\tau_g \mid x)}, \qquad D_{\text{KL}}\big(\pi_\theta \,\|\, \pi_{\text{ref}}\big) = \frac{1}{\psi_g} + \log\psi_g - 1 \tag{5}$$

REINFORCE留一法（RLOO） [33] 会针对每个几何提示 x采样一组 G 独立的轮廓序列$\{\tau\}_{g=1}^{G}$。组内每个样本的奖励 r_g 都作为其他所有样本的基线，因此策略梯度目标如下：

$$\mathbb{E}_{\{\tau_g\}_{g=1}^{G}\sim\pi_\theta(\cdot|x)}\Big[\frac{1}{G}\sum_{g=1}^{G}[\big(r_g - \frac{1}{G-1}\sum_{i\neq g} r_i\big)\cdot\nabla_\theta \log \pi_\theta(\tau_g \mid x)]\Big]. \tag{6}$$

5 结果

5.1 通过几何提示进行形状控制

几何提示包含设计师可用于控制生成轮廓形状的信息。这里展示一些形状控制的定性示例。本节每幅图中，几何提示的一个值会发生变化，其余值则固定为表 2 所示的值。绿色标出了轮廓应与之匹配的线段位置，紫色虚线标出了对称线，紫色圆点则表示重心。

表 2：本节几何提示中使用的默认属性。

属性	值
Area	0.4
对称性	垂直对称线
切线连续顶点比例	0.5
边数	16
重心	0.0, 0.0
顶部宽度	0.1

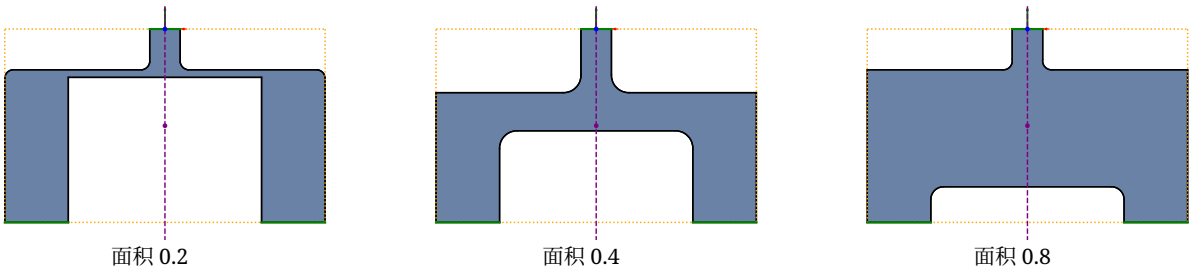


图 2：指定三种不同轮廓面积时生成的轮廓。指定面积为 0.2 个模型单位时，生成的轮廓包含狭窄区域。增大指定面积后，生成的形状便不会包含此类狭窄区域。

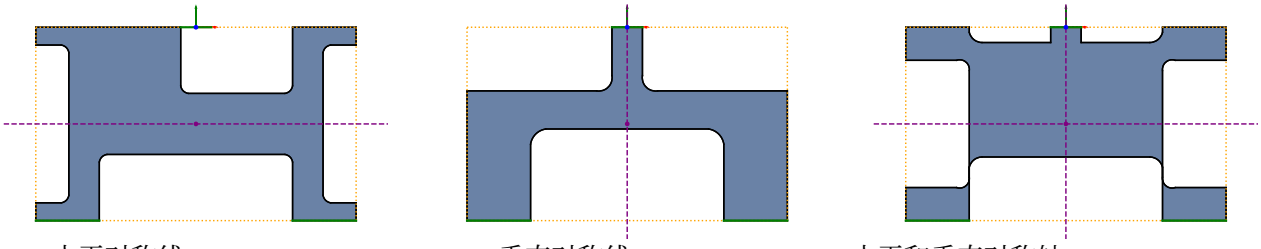


图 3：在几何提示中请求不同的对称线。

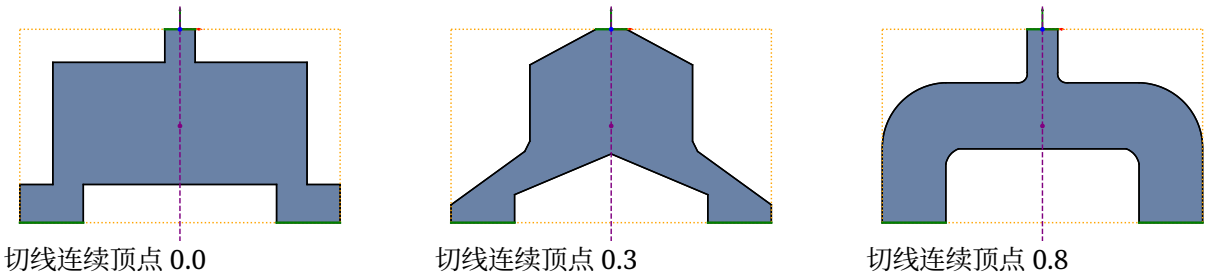


图 4：通过改变所请求的切线连续顶点比例获得的轮廓。

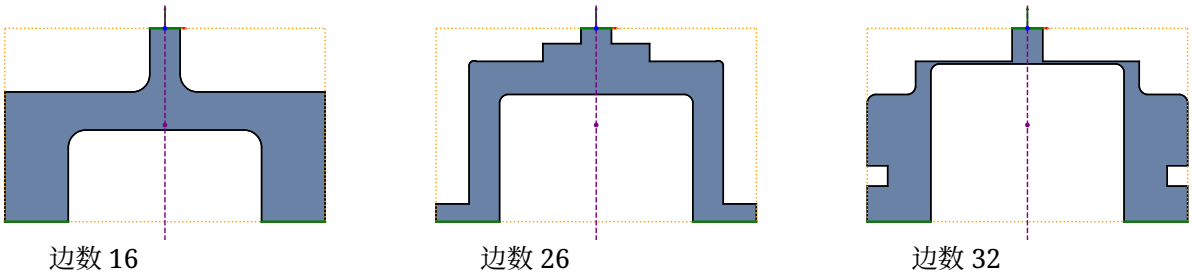


图 5：通过在几何提示中请求不同的边数来控制形状复杂度。

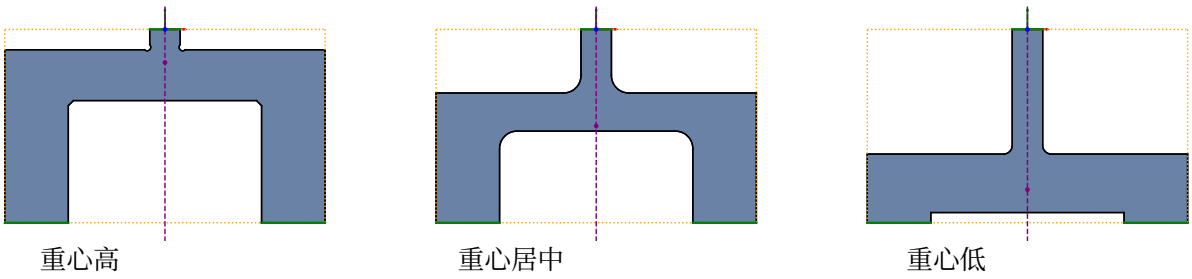


图 6：提高和降低指定重心的效果。重心以紫色点标示。

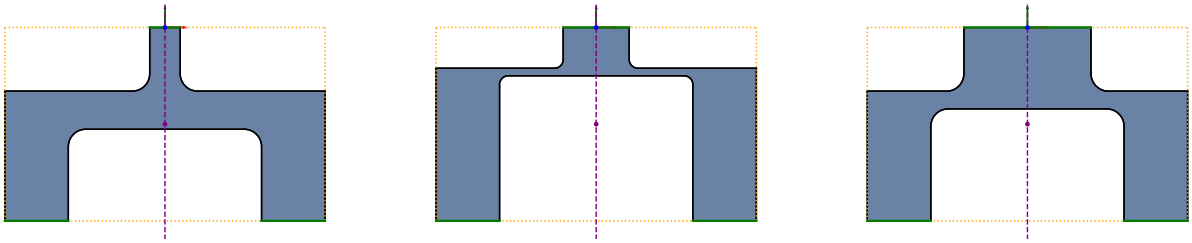


图 7：改变图形顶部有界线段（绿色）的长度所产生的影响。随着该长度变化，轮廓的顶部会自然地变宽。

设计师通常希望能够控制所建模部件的机械强度和重量。我们发现，改变生成轮廓的指定面积可以在一定程度上控制厚度，因此能够以增加质量为代价提升机械强度。图 2 展示了指定面积各不相同的三种生成形状。

对于设计师而言，能够设计出对称的形状也很重要。许多情况下，设计师会先绘制对称线，然后使用约束使各种轮廓曲线相对于这些对称线保持对称。图 3 展示了在几何提示中加入水平和垂直对称轴对生成结果的影响。请注意，要求轮廓匹配的有界线段（绿色直线）仅呈现出与垂直对称线相符的对称性。当指定水平对称线时，网络无法同时匹配有界线段和所要求的对称性。附录 D 的图 11 展示了同时加入水平和垂直对称线时的详细构造序列。从该序列中可以看出，最终形状中的对称性是通过将点关于给定的对称线镜像得到的。

一种有效控制形状整体风格的方法，是要求形状具有尖角或切线连续顶点。在图 4 中可以看到，提高要求的切线连续顶点比例，可以使形状从直线形态（左）逐渐变为右侧所示的平滑曲线形状。

通过在几何提示中指定边数，可以调整生成轮廓的复杂度。图 5 展示了边数增加时，形状也会变得更加复杂。然而，我们发现，要求较高复杂度时，新增细节往往定义得不够清晰。实际应用中，降低要求的复杂度有助于消除这些多余或含糊的特征。

当目标形状不明确时，还可以通过指定轮廓的目标重心来进一步控制。在图 6 中，我们展示了沿对称线升高或降低重心如何改变最终形状。

最后，图 7 展示了在表 2 定义的其他属性值均保持不变的情况下，改变提示中提供的有界线段，也会带来合理的形状变化。

5.2 评估指标

二维参数化轮廓生成的评估指标大致可分为两类：有效性指标，用于评估生成的轮廓序列在句法上是否正确、在几何上是否合理；以及提示满足度指标，用于评估生成的轮廓形状对输入几何提示中指定的约束和属性的符合程度。

5.2.1 有效性指标

有效性指标用于衡量生成的轮廓是否同时符合领域专用语言的句法要求，以及格式良好的轮廓形状所隐含的几何预期。这些指标包括：

句法有效性：生成的序列是否能够按照领域专用语言的严格句法规则成功进行反标记化。

无自相交：生成的轮廓是否不存在自相交。

无短边：所有边是否都超过由量化箱尺寸定义的最小长度。

这些是典型的均报告为布尔指标，并汇总为有效轮廓生成的总体比例 s.

5.2.2 提示满足度指标

随后即可定量评估生成的轮廓对几何提示的遵循程度，从而直接、客观地评估提示满足度。部分提示满足度指标包括：

面积：通过几何提示中规定的面积与生成的轮廓面积之差进行测量。

线段：对于提示中指定的每条线段，根据轮廓中存在的与其共线、重叠且长度相等的线段计算指标。同时记录所请求线段与生成线段端点之间的距离。

重心：通过几何提示中定义的重心与生成的轮廓重心之间的距离进行测量。

孔：通过几何提示中定义的孔中心与生成的轮廓中孔中心之间的距离进行测量。

对称线：通过轮廓与其关于对称线的镜像之间的交并比（IoU）进行测量，并对提示中的所有对称线取平均值。

外接边界框：通过几何提示中定义的外接边界框与生成的轮廓的交并比来衡量。

切线连续顶点比例：通过比较几何提示规定的切线连续顶点比例与生成的轮廓中的该比例来衡量。

5.3 定量结果

表 3 对五种模型的评估指标进行了比较分析：一种是在未使用构造序列的情况下训练的基线模型；一种是使用构造序列训练、且超参数与基线模型相同的构造步骤模型；以及该构造步骤模型的三种强化学习微调变体。这三种配置一致的变体 ReMax、GRPO 和 RLOO，使用第 4.2.1 节定义的奖励函数进行优化。所有模型均采用 k 等于 1 的贪心采样进行评估。
引入构造序列后，所有有效性指标和大多数提示满足度指标均较基线模型有显著提升。句法有效性从 88.1 % 提高到 94.0%，无自相交轮廓的比例从 81.9% 提高到 84%，而符合最小边长要求的比例则从 88.2% 提高到 94.3%。在提示满足度指标中，线段符合度和镜像对称性的提升最为显著。这些结果证实，仅整合构造序列就能显著提升结构有效性以及与几何约束的契合度。

通过基于强化学习的对齐可以进一步提升性能。与未对齐的构造步骤模型相比，对齐变体在句法和几何有效性指标上始终表现更佳，包括避免自相交以及满足最小边长要求。例如，RLOO 和 GRPO 在生成自相交几何图形方面均实现了超过 6% 的降幅。值得注意的是，尽管奖励函数明确旨在优化几何有效性，我们仍观察到一系列几何提示指标均获得了稳定且往往十分显著的提升，包括面积准确度、边界框对齐、对称性和孔的位置。这表明，对齐所带来的结构改进不仅满足了低层约束，还增强了高层几何属性，即使这些属性在优化过程中并未受到直接激励。

6 定性结果

图 8 展示了不同模型生成轮廓的定性比较。第一列展示用于驱动生成过程的几何提示，其余各列则展示生成的轮廓几何。这些结果表明，当基线模型无法遵循提示中编码的复杂结构约束时，构造步骤模型及其对齐变体能够成功生成几何上有效且符合设计规范的结果。更多定性结果见附录5.1和D。这些结果展示了如何利用几何提示控制生成轮廓的形状，并包含完整构造序列的示例，以及通过改变驱动参数可由这些序列得到的一系列形状。

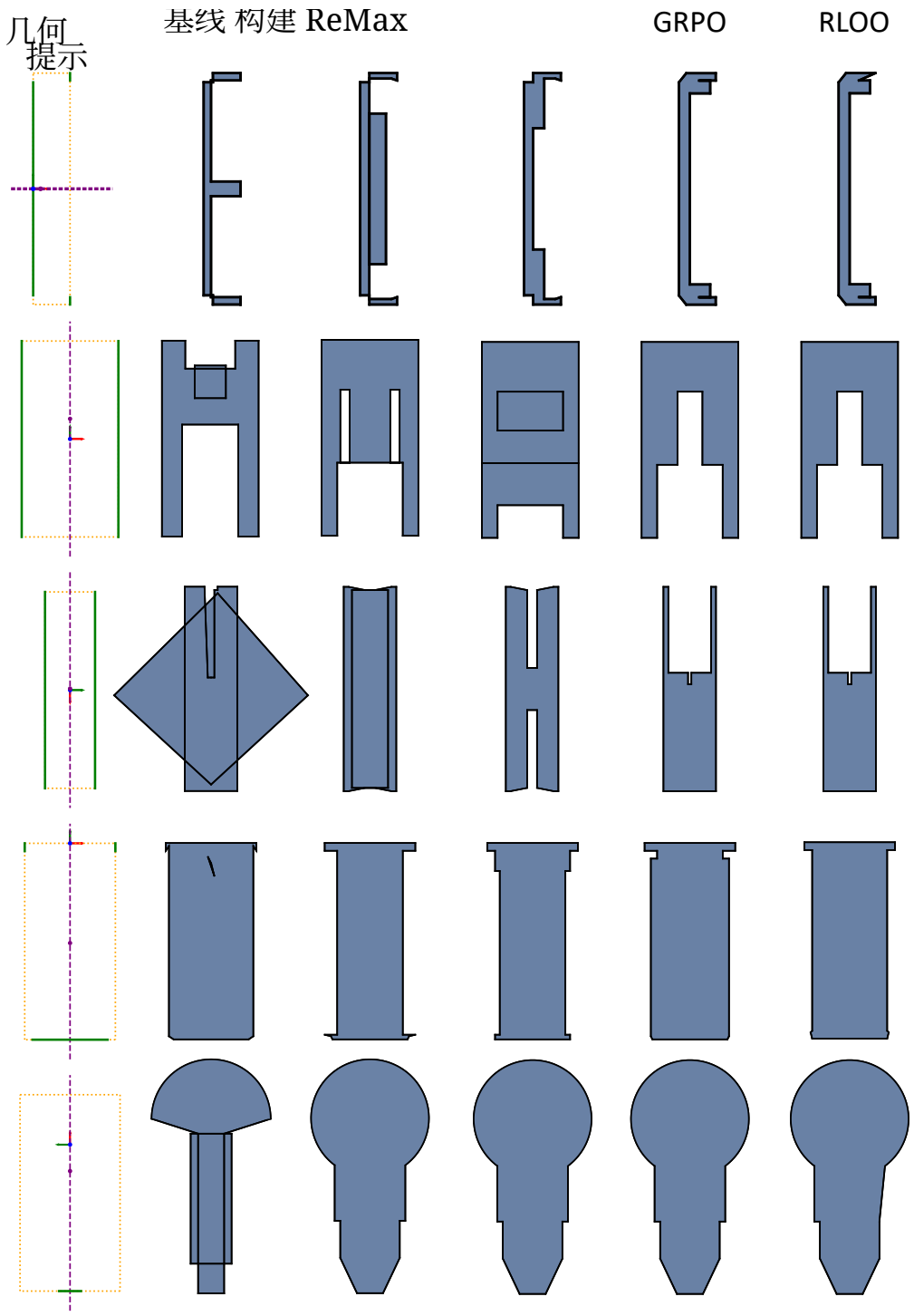


图 8：基础模型（不含构造序列）、构造序列模型和对齐模型生成轮廓的视觉比较。

7 结论

在这项工作中，我们为计算机辅助设计生成引入了一种新的序列表示，该表示通过一系列简单的几何构造步骤来构造轮廓。在设计师的输入与最终形状之间加入这些构造步骤，能够提升生成质量，并促进生成结果遵循设计要求。此外，我们

表 3：不同模型关键指标比较

指标	基线模型	构建步骤模型	构建步骤模型（ReMax）	构建步骤模型（GRPO）	构建步骤模型（RLOO）
句法有效性（↑）	0.881	0.940	0.945	0.975	**0.976**
无自相交（↑）	0.819	0.840	0.853	0.903	**0.905**
无短边（↑）	0.882	0.943	0.948	0.976	**0.978**
面积差（↓）	0.253	0.238	0.210	0.170	**0.162**
线段距离（↓）	0.00313	0.00152	**0.00043**	0.00082	0.00111
线段比率（↑）	0.963	**0.983**	0.981	0.978	0.976
重心距离（↓）	**0.0247**	0.0267	0.0285	0.0254	0.0265
孔中心距离（↓）	0.0255	0.0318	**0.0153**	0.0232	0.0402
镜像交并比 (↑)	0.818	0.859	0.865	**0.886**	**0.886**
外部边界框边界框交并比 (↑)	**0.990**	0.984	0.981	0.985	0.983
切线连续顶点差异 (↓)	0.0907	**0.0786**	0.0799	0.0814	0.0816

结果表明，在惩罚自相交的奖励函数引导下，强化学习在多种指标上均取得了稳定改进，包括那些未被明确设为优化目标的指标。由于生成的序列可以按浮点精度重放，它们克服了以往方法的精度局限。此外，通过将自由度缩减为一小组参数，最终形状可以像参数化 CAD 系统中的编辑操作一样进行参数化调整。

参考文献

[1] Ari Seff、Yaniv Ovadia、Wenda Zhou 和 Ryan P. Adams。《SketchGraphs：用于建模计算机辅助设计中关系几何的大规模数据集》。发表于*ICML 2020* 面向对象学习研讨会，2020年。[2] Karl D. D. Willis、Pradeep Kumar Jayaraman、Joseph G. Lambourne、Hang Chu 和 Yewen Pu。《面向计算机辅助设计的工程草图生成》。发表于*2021 IEEE/CVF*计算机视觉与模式识别会议研讨会 *(CVPRW)*，第2105–2114 页，2021年。[3] Yaroslav Ganin、Sergey Bartunov、Yujia Li、Ethan Keller 和 Stefano Saliceti。《将计算机辅助设计视为语言》。发表于第 *35* 届神经信息处理系统国际会议论文集，NIPS ’21，美国纽约州雷德胡克，2021年。Curran Associates Inc.[4] Wamiq Reyaz Para、Shariq Farooq Bhat、Paul Guerrero、Tom Kelly、Niloy Mitra、Leonidas Guibas 和 Peter Wonka。《Sketchgen：生成受约束的 CAD 草图》。发表于第 *35* 届神经信息处理系统国际会议论文集，NIPS ’21，美国纽约州雷德胡克，2021年。Curran Associates Inc.[5] Ronghuan Wu、Wanchao Su、Kede Ma 和 Jing Liao。《Iconshop：借助自回归 Transformer 实现文本引导的矢量图标合成》。*ACM* 图形学汇刊，42(6)，2023年12月。[6] Xiang Xu、Joseph Lambourne、Pradeep Jayaraman、Zhengqing Wang、Karl Willis 和 Yasutaka Furukawa。《Brepgen：一种具有结构化潜在几何的边界表示生成式扩散模型》。*ACM*图形学汇刊 *(TOG)*，43(4):1–14，2024。[7] Jiajie Fan、Babak Gholami、Thomas B¨ack 和 Hao Wang。《Neuronurbs：为三维实体学习高效的曲面表示》。*arXiv* 预印本 *arXiv:2411.10848*，2024。

[8] Mingi Lee、Dongsu Zhang、Cl´ement Jambon 和 Young Min Kim。Bre暴露diff：单阶段边界表示扩散模型。载于计算机图形学与交互技术特别兴趣小组会议论文集，SIGGRAPH Conference Papers ’25，美国纽约州纽约，2025年。计算机协会。

[9] Evan Casey、Tianyu Zhang、Shu Ishida、John Roger Thompson、Amir Khasahmadi、Joseph George Lambourne、Pradeep Kumar Jayaraman 和 Karl D. D. Willis。使约束生成与参数化CAD中的设计意图保持一致，2025年。[10] Jason Wei、Xuezhi Wang、Dale Schuurmans、Maarten Bosma、Brian Ichter、Fei Xia、Ed H. Chi、Quoc V. Le 和 Denny Zhou。思维链提示能够激发大语言模型的推理能力。载于第*36*届神经信息处理系统国际会议论文集，NIPS '22，美国纽约州雷德胡克，2022。Curran Associates Inc.[11] Takeshi Kojima、Shixiang Shane Gu、Machel Reid、Yutaka Matsuo 和 Yusuke Iwasawa。大语言模型是零样本推理器。载于第*36*届神经信息处理系统国际会议论文集，NIPS '22，美国纽约州雷德胡克，2022。Curran Associates Inc.[12] J. C. Owen。利用尺寸约束对几何进行代数解法。载于首届*ACM*实体建模基础与*CAD/CAM*应用研讨会论文集，SMA '91，第397–407页，美国纽约州纽约，1991年。计算机协会。[13] William Bouma、Ioannis Fudos、Christoph Hoffmann、Jiazhen Cai 和 Robert Paige。几何约束求解器。计算机辅助设计，27(6)：487–501页，1995年。[14] Mengjiao Yang、Dale Schuurmans、P. Abbeel 和 Ofir Nachum。通过过程克隆进行思维链模仿。*arXiv*，abs/2205.10816，2022。[15] Lucas Lehnert、Sainbayar Sukhbaatar、Paul Mcvay、Michael Rabbat 和 Yuandong Tian。超越A*：通过搜索动态自举，利用Transformer实现更优规划。*arXiv*，abs/2402.14083，2024。[16] Kanishk Gandhi、Denise Lee、Gabriel Grand、Muxin Liu、Winson Cheng、Archit Sharma 和 Noah D. Goodman。搜索流（SOS）：学习在语言中进行搜索。*arXiv*，abs/2404.03683，2024。[17] Zhihong Shao、Peiyi Wang、Qihao Zhu、Runxin Xu、Junxiao Song、Xiao Bi、Haowei Zhang、Mingchuan Zhang、Y. K. Li、Y. Wu 和 Daya Guo。DeepSeekMath：突破开放式语言模型数学推理的极限，2024。[18] DeepSeek-AI、Daya Guo、Dejian Yang、Haowei Zhang、Junxiao Song、Ruoyu Zhang、Runxin Xu、Qihao Zhu、Shirong Ma、Peiyi Wang、Xiao Bi、Xiaokang Zhang、Xingkai Yu、Yu Wu、Z. F. Wu、Zhibin Gou、Zhihong Shao、Zhuoshu Li、Ziyi Gao、Aixin Liu、Bing Xue、Bingxuan Wang、Bochao Wu、Bei Feng、Chengda Lu、Chenggang Zhao、Chengqi Deng、Chenyu Zhang、Chong Ruan、Damai Dai、Deli Chen、Dongjie Ji、Erhang Li、Fangyun Lin、Fucong Dai、Fuli Luo、Guangbo Hao、Guanting Chen、Guowei Li、H. Zhang、Han Bao、Hanwei Xu、Haocheng Wang、Honghui Ding、Huajian Xin、Huazuo Gao、Hui Qu、Hui Li、Jianzhong Guo、Jiashi Li、Jiawei Wang、Jingchang Chen、Jingyang Yuan、Junjie Qiu、Junlong Li、J. L. Cai、Jiaqi Ni、Jian Liang、Jin Chen、Kai Dong、Kai Hu、Kaige Gao、Kang Guan、Kexin Huang、Kuai Yu、Lean Wang、Lecong Zhang、Liang Zhao、Litong Wang、Liyue Zhang、Lei Xu、Leyi Xia、Mingchuan Zhang、Minghua Zhang、Minghui Tang、Meng Li、Miaojun Wang、Mingming Li、Ning Tian、Panpan Huang、Peng Zhang、Qiancheng Wang、Qinyu Chen、Qiushi Du、Ruiqi Ge、Ruisong Zhang、Ruizhe Pan、Runji Wang、R. J. Chen、R. L. Jin、Ruyi Chen、Shanghao Lu、Shangyan Zhou、Shan-huang Chen、Shengfeng Ye、Shiyu Wang、Shuiping Yu、Shunfeng Zhou、Shuting Pan 和 S. S. Li。DeepSeek-R1：通过强化学习激发大语言模型的推理能力。*CoRR*，abs/2501.12948，2025年1月。[19] Oriol Vinyals、Meire Fortunato 和 Navdeep Jaitly。指针网络。*NIPS*，2015。[20] Ari Seff、Wenda Zhou、Nick Richardson 和 Ryan P. Adams。Vitruvion：一种参数化CAD草图生成模型。*arXiv*，abs/2109.14124，2021年。[21] Mohammad Sadil Khan、Sankalp Sinha、Sheikh Talha Uddin、Didier Stricker、Sk Aziz Ali 和 Muhammad Ze-shan Afzal。Text2CAD：根据从入门到专家级的文本提示生成序列化CAD设计。载于神经信息处理系统进展，第37卷，第7552–7579页。Curran Associates, Inc.，2024年。[22] Rundi Wu、Chang Xiao 和 Changxi Zheng。Deepcad：一种用于计算机辅助设计模型的深度生成网络。载于*IEEE/CVF*国际计算机视觉大会（*ICCV*）论文集，2021年10月。[23] Yandong Guan、Xilin Wang、Xingxi Ming、Jing Zhang、Dong Xu 和 Qian Yu。CAD-Coder：结合思维链与几何奖励的文本转CAD生成，2025年。[24] Xueyang Li、Jiahao Li、Yu Song、Yunzhong Lou 和 Xiangdong Zhou。Seek-CAD：通过DeepSeek的局部推理实现三维参数化CAD的自我优化生成建模，2025年。[25] Xiaolong Yin、Xingyu Lu、Jiahang Shen、Jingzhe Ni、Hailong Li、Ruofeng Tong、Min Tang 和 Peng Du。Rlcad：用于涉及旋转的CAD命令序列生成的强化学习训练环境，2025年。[26] Cheng Chen、Jiacheng Wei、Tianrun Chen、Chi Zhang、Xiaofeng Yang、Shangzhan Zhang、Bingchen Yang、Chuan-Sheng Foo、Guosheng Lin、Qixing Huang 和 Fayao Liu。CADCrafter：从无约束图像生成计算机辅助设计模型，2025年。

[27]Maksim Kolodiazhnyi、Denis Tarasov、Dmitrii Zhemchuzhnikov、Alexander Nikulin、Ilya Zisman、Anna Vorontsova、Anton Konushin、Vladislav Kurenkov 和 Danila Rukhovich。Cadrille：利用在线强化学习进行多模态计算机辅助设计重建。*arXiv* 预印本 *arXiv:2505.22914*，2025。[28] Sebastian Koch、Albert Matveev、Zhongshi Jiang、Francis Williams、Alexey Artemov、Evgeny Burnaev、Marc Alexa、Denis Zorin 和 Daniele Panozzo。ABC：面向几何深度学习的大型计算机辅助设计模型数据集。载于*IEEE*计算机视觉与模式识别会议论文集，第 9593–9603 页，2019。[29]Open Cascade。Open Cascade Technology（OCC）。https://www.opencascade.com/open-cascade-technology/，2025。访问日期：2025-05-21。[30] Nino Shervashidze、Pascal Schweitzer、Erik Jan van Leeuwen、Kurt Mehlhorn 和 Karsten M. Borgwardt。Weisfeiler-Lehman图核。机器学习研究期刊，12（null）：2539–2561，2011年11月。[31] Diederik P. Kingma 和 Jimmy Ba。Adam：一种随机优化方法。*CoRR*，abs/1412.6980，2014。[32] Ziniu Li、Tian Xu、Yushun Zhang、Zhihang Lin、Yang Yu、Ruoyu Sun 和 Zhi-Quan Luo。ReMax：一种用于对齐大语言模型的简单、有效且高效的强化学习方法。载于第*41*届国际机器学习会议论文集，ICML’24。JMLR.org，2024。.[33] Arash Ahmadian、Chris Cremer、Matthias Gall´e、Marzieh Fadaee、Julia Kreutzer、Olivier Pietquin、Ahmet Ust¨un 和 Sara Hooker。回归基础：重新审视用于从大语言模型的人类反馈中学习的 REINFORCE 式优化，2024。[34] Lex Weaver 和 Nigel Tao。基于梯度的强化学习的最优奖励基线。载于第十七届人工智能不确定性会议论文集，UAI’01，第 538–545 页，美国加利福尼亚州旧金山，2001。Morgan Kaufmann出版社。[35] John Schulman、Filip Wolski、Prafulla Dhariwal、Alec Radford 和 Oleg Klimov。近端策略优化算法。*CoRR*，abs/1707.06347，2017。[36] Xiang Xu、Karl DD Willis、Joseph G Lambourne、Chin-Yi Cheng、Pradeep Kumar Jayaraman 和 Yasutaka Furukawa。SkexGen：使用解耦码本自回归生成计算机辅助设计构造序列。载于国际机器学习会议，第 24698–24724 页。PMLR，2022。[37] Chuan Li、Michael Wand、Xiaokun Wu 和 Hans-Peter Seidel。利用共现分析进行近似三维部分对称性检测。载于*2015*年国际三维视觉会议，第 425–433 页，2015。

A 构造序列的领域专用语言与标记化

A.1 几何

用于表示几何提示、构造序列和最终轮廓的领域专用语言，共享一套用于几何实体的词汇。标量值、点和无限直线均以单个词元表示，如表 4 所示。随后，通过组合表示实体类型的特殊标记与表示几何形状的原始标记，构造更复杂的实体，如表 5 所示。

表 4：由单个词元表示的几何和标量值

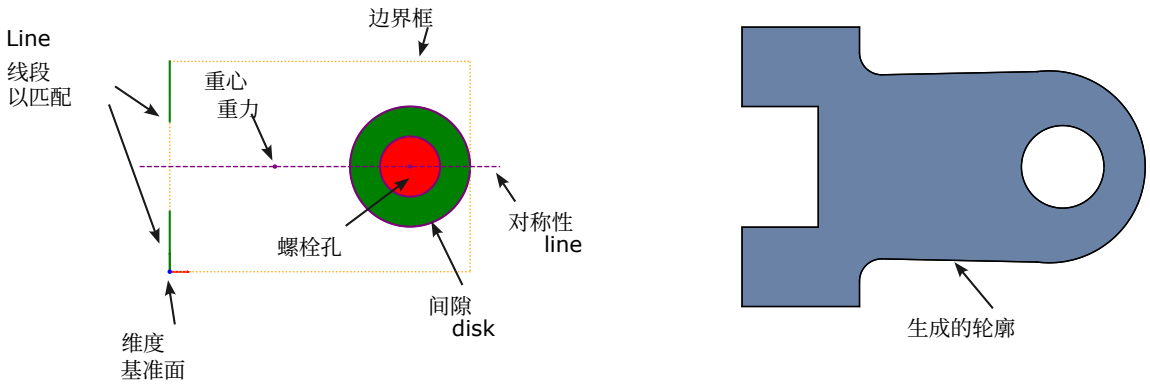
令牌	**说明**
长度	范围为 $[-1, 1]$ 的有符号长度被量化为 127 个分箱。特意选择奇数个分箱，是因为这样在将量化值转换回浮点数时，可以精确还原 -1.0、0.0 和 1.0。量化值转换回浮点数后即可得到这些数值。
角度	范围为 $[0, 2\pi)$ 的角度被量化为 121 个分箱。我们特别确保角度 0.0 和 2π 映射到第一个分箱的中心。由于 120 的质因数包括 2、3 以及 5，这种量化策略可以精确重建常见角度，例如 0.0, π, $\pi/2$ 和 $\pi/3$ 精确重建。
点	点被量化到一个 127×127 网格上。每个方向上的网格单元数量均选择为奇数以确保反量化量化表示时，可以精确重建定义域中的中心点量化表示经过反量化。
Infline	有向无限直线采用 Hesse 法式表示。直线与和 x轴之间的角度编码为 121 个分箱，有符号距离原点的距离编码为为 127 个分箱。随后，无限直线被表示为单个词元，共有 121×127 种可能的取值。
`Area`	几何提示中使用面积值来控制轮廓厚度。在进行归一化后，轮廓的面积将处于 $[0, 1]$ 范围内，并可量化为 127 个分箱。
复杂度	轮廓中的曲线数量用于衡量复杂度。复杂度令牌有 127 种可能的取值，最后一个令牌值用于包含超过 127 条曲线的轮廓。
循环数	轮廓中的环数由一个单个词元表示，该词元有 127 种可能的取值。
平滑顶点	切线连续的顶点比例被量化为 127 个分箱。

表 5：由多个词元组合定义的几何实体。每个多词元实体都以一个特殊标记开头，该标记表示实体类型，后面跟着一系列用于定义所需几何的词元。

令牌		说明
BoundLine		有界线段以 BoundedLine 令牌开头，后接两个
点	// 起点	表示线段起点和终点的点令牌。
点	// 终点	
Arc		圆弧由一个圆弧类型标记及其后跟的三个点令牌表示
点	// 起点	分别表示圆弧段的起点、中点和终点。
点	// 中点	
点	// 终点	
圆		有向圆用一个点令牌表示圆心点，并用一个
点	// 中心点	长度令牌表示圆的半径。另一个标记表示该
长度	// 半径	圆是顺时针还是逆时针定向。
是否逆时针	// 逆时针	
包围盒		边界框由最小和最大角点处的两个点表示。
点	// 最小点	
点	// 最大点	
尺寸基准		尺寸基准定义了一个固定的参考点，其他几-
点	// 基准点	何图形可由此构建。它由一个基准点和两条无限直线组成
Infline	// *x* 轴	穿过该点，并与 x、y 坐标系轴对齐。
Infline	// *y* 轴	
螺栓孔		螺栓孔在机械零件中很常见。它们用于几何
点	// 中心点	提示中，用来定义孔的位置以及其周围所需的间隙。
长度	// 半径	标记化包含中心点、表示孔径大小的半径，以及
长度	// 间隙圆盘	表示应当移除的材料的间隙圆盘。
		应保留在孔周围，以防止螺栓发生撕裂破坏。

A.2 几何提示

生成过程由几何提示引导，这些提示规定了生成的轮廓应符合的几何特征，以及轮廓面积和重心等其他重要属性。附录 B.2 更详细地说明了几何提示中各组成部分的设置依据，附录 5.1 则展示了一些如何利用提示控制形状的示例。提示的标记化方式见表 6，其中包含表 4 和表 5 中更低层级的实体。图 9 展示了几何提示及其生成的轮廓。



Fi图 9：几何提示中使用的几何元素及其生成的轮廓 .

A.3 构造步骤

第 3.2 节介绍了构造步骤，并给出少量示例。其余构造步骤列于表 7。每个构造步骤都包含一个描述步骤类型的特殊标记，后面跟着一系列输入和输出几何，其编码方式如表 4 和表 5 所述。

表 6：几何提示实体

令牌	说明
提示起始	表示几何提示开始的特殊标记。
尺寸基准	尺寸基准。构造步骤将逐步构建相对于此固定基准点的轮廓几何。
包围盒	所生成轮廓所需的边界框。
Area	所生成轮廓所需的面积。
复杂度	我们希望生成的轮廓包含的曲线的大致数量的。
循环数	生成的轮廓所需的环数。
平滑顶点	我们希望成为平滑顶点的顶点比例切线连续。
重心点	轮廓的目标重心。
对称线	定义我们希望的镜像对称性的直线
无穷线列表	生成的轮廓所应具备的。
BoundLines	生成的轮廓应包含的线段列表
边界线列表	应包含 .
螺栓孔	应包含在生成的
螺栓孔列表	生成的轮廓。
提示结束标记	表示几何提示结束的特殊标记。

构造步骤列表以特殊标记 StartOfConstruction 开始，并以标记 EndOfConstruction 结束。在使用参数值的构造步骤之前，会按照附录 A.4 所述，在序列中提供一个 UseParameterN 参数索引标记。一旦创建了足够的几何以生成最终轮廓中的一条曲线，就会添加曲线创建标记 CreatedCurve，后面跟着表 5 所述的 BoundLine、圆弧或圆标记。我们发现，引入这些标记对网络的性能至关重要。若没有这些标记，序列末尾预测出的轮廓往往会出现自相交。若干代的完整构造序列见附录 D。

A.4 参数

构造序列将生成的轮廓中的自由度减少到少量参数，这些参数可用于交互式形状编辑。这些参数的值由模型预测，但设计师可以调整这些值，并以浮点精度重新执行序列。在构造步骤中使用任何参数值之前，都会包含一个特殊的 UseParameterN 标签标记，用于指示要使用的参数索引。共有 32 个独立标记：UseParameter0、UseParameter1、.....，最多可定义 32 个独立参数。该标记之后给出参数值。参数值由 Length 或 Angle 标记表示。

A.5 轮廓几何

序列以轮廓几何结尾，其编码采用一种与 [22, 36] 类似的领域专用语言。在由多个圆弧和线段组成的环中，每条曲线的起点都是前一条曲线的终点。直线可通过其端点定义，圆弧则由中点和端点表示。一个特殊标记用于指示新环的起点。由于圆是闭合曲线，因此始终位于独立的环中，并由中心点和半径表示。轮廓几何的标记化方式见表 8。

表 7：构造步骤示例（续）

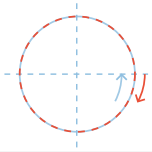
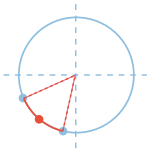
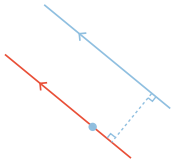
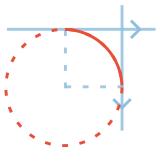
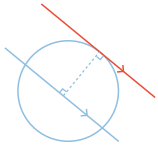
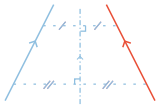
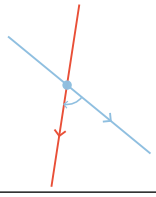
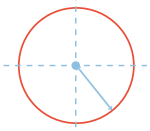
描述	说明	示例
CircleReverseCircle **输入：** 圆$_1$ **输出：** 圆$_2$	给定一个定向圆 ，返回相同的圆，但方向相反向 ·	
圆-点-点-圆弧 **输入：** 圆， 点$_{\text{start}}$，点$_{\text{end}}$ **输出：** 点$_{\text{arc}}$	给定一个有向圆和一个起点以及圆上的终点。求出圆弧的中点，这将将圆修剪至该范围，该范围从起点开始，并沿着圆的方向延伸至终点。	
直线-基准-平行线 **输入：** 直线$_1$， 点$_{\text{datum}}$， **输出：** 直线$_2$	给定一条直线和一个基准点，求出并返回一条与给定直线平行的直线且经过该基准点。	
直线-直线圆角 **输入：** 直线$_1$， 直线$_2$，半径 **输出：** 圆弧	给定两条有向直线和一个圆角——dius，求出并返回圆角弧该圆角弧可与其 按顺序排列的起点和终点一起构造。	
直线-圆平行线 **输入：** 直线$_1$， 圆$_1$ **输出：** 直线$_2$	给定一条有向直线和一个圆，计算与原直线平行且与该圆相切的直线。这样的直线恰好有两条，圆的两侧各一个 ，the输入直线的方向决定哪一个会被选中。	
对称轴直线直线 **输入：** 直线$_1$， 直线$_{\text{sym}}$ **输出：** 直线$_2$	给定一条直线和一条对称线，求出并返回这条直线的像即它关于对称线的反射像。	
点-直线对称点步骤 **输入：** 点$_1$， 直线$_{\text{sym}}$ **输出：** 点$_2$	给定一个点和一条对称线，求出并返回该点关于对称线的对称点 line.	
LineReverseLine **输入：** 直线$_1$ **输出：** 直线$_2$	给定一条有向直线，返回其反向方向。	
LineAxisRotatedLine **输入：** 直线$_1$， 角度 **输出：** 直线$_2$	给定一条直线、一个点和一个角度，构造并返回一条旋转后的直线，旋转角度为顺时针方向的角度，或或逆时针方向。	
点半径圆 **输入：** 点， 半径 **输出：** 圆	给定一个点和半径，创建并返回一个圆 ·	

表 8：最终轮廓的标记化。由于轮廓仅包含闭合环，因此前一条曲线的终点可作为当前曲线的起点

令牌	**说明**
轮廓起始标记	用于指示轮廓起始位置的特殊标记
多段线圆弧环	表示包含多条直线和圆弧的环的起始位置
单圆环 点 // 中心点 长度 // 半径	表示由单个圆组成的环
轮廓线段 点 // 终点	由其终点定义的轮廓线段。上一条曲线的终点 用作此线段的起点。
轮廓圆弧 点 // 中点 点 // 终点	由其中点和终点定义的轮廓圆弧。该圆弧的终点 前一条曲线用作此圆弧的起点。
轮廓结束	表示轮廓数据结束的特殊标记

B 从轮廓几何中提取构造序列

B.1 预处理

在开始提取构造序列之前，会对轮廓进行归一化，使其适合位于以原点为中心的单位正方形内。这样可以在下游处理中使用一组一致的公差。长度公差定义为 1/127模型单位，与标记化所使用的量化区间大小相匹配。如果两点之间的距离小于该公差，则认为它们重合。当向量的偏差小于 1 度时，则认为它们平行。

随后执行以下预处理。

- 移除短于长度公差的曲线。移动相邻曲线的端点以闭合间隙。
- 合并共线线段
- 每个环中的曲线按照直线段和圆弧段的终点进行字典序排序。
- 环按照外环在前、内环按字典序排序的方式排列。对于多曲线环，使用第一条曲线的终点进行排序。对于圆，则使用底部的点。

B.2 提取几何提示

我们没有像 [20], 中那样，训练模型根据曲线的随机子集自动补全轮廓，而是根据常见的设计要求选择纳入几何提示的几何元素。下面将介绍所提取的几何信息以及提取这些信息的推理依据。

线段：当拉伸体组合起来构建三维模型时，它们通常会在平面处彼此对齐。因此，在定义轮廓几何时，能够指定将成为生成的轮廓凸包一部分的线段非常有用。为构建几何提示，我们首先识别凸包上的所有线段。然后从中选取一个子集，并尽量保留成组的对称线段。对于那些在移除后不会缩小线段集合凸包的组，则随机舍弃。
螺栓孔：螺栓通常用于连接机械部件。因此，机械部件往往需要在特定位置开设螺栓孔。我们将轮廓中的螺栓孔识别为圆形内环。每个螺栓孔由孔中心、表示孔尺寸的半径以及间隙圆盘来指定；间隙圆盘定义了螺栓周围所需的零件强度。如果螺栓周围的材料不足以承受横向力，螺栓可能会从螺栓孔中撕脱，因此，螺栓周围间隙的大小是设计师希望控制的重要因素。

轮廓面积：这使得生成的轮廓厚度可控，从而能够在零件的物理强度、重量和材料利用率之间取得平衡。

切线连续顶点的比例：允许设计师控制切线连续顶点的比例，使他们能够决定生成的形状具有尖角还是圆角。通过比较相邻曲线端点处的切线，可以轻松检测出切线连续顶点。

对称线：许多机械零件都具有对称性，设计师也需要能够强制实现对称性。我们通过 [37] 的二维实现来识别对称性，并将其作为无限直线纳入几何提示中。

边界框：零件将在单位正方形中生成，导入 CAD 应用程序时再进行缩放和定位。设计师指定生成的轮廓的尺寸和纵横比，一种直观的方法是在 CAD 软件的坐标系中使用定向边界框。该边界框可映射为一个紧贴单位正方形的二维边界框。随后，可以将生成的轮廓变换并缩放回 CAD 系统的坐标系中。生成模型还可以利用边界框的纵横比作为额外的形状控制。
重心：如果所需轮廓仅占边界框面积的一小部分，且提示中的其他几何信息对其定义不足，则可以使用重心来引导形状。当形状需要避开可能导致装配环境中发生干涉的区域时，这种控制尤其有用。

轮廓复杂度：通常需要能够控制轮廓的复杂度。复杂度过高时，生成算法可能会生成不需要的额外细节；复杂度过低时，则可能生成非常简单的形状。几何提示中使用轮廓的边数作为复杂度的衡量指标。

内部环的数量：设计师通常希望创作兼具强度与轻量化特点的设计。通过控制轮廓中内部环的数量，并结合轮廓面积，我们可以促使模型生成更多内部环，从而减轻形状的重量。

尺寸标注基准：设计师通常希望根据某些固定几何形状对轮廓进行参数化。在参数化 CAD 中，这通常是坐标系的固定原点，或由先前的建模特征创建的参考点。为了让设计师能够控制这些固定几何形状，我们在几何提示中加入一个基准点。基准点从以下位置中随机选取：

- 轮廓边界框的一个角
- 边界框某条边的中点
- 边界框的中心
- 轮廓中最大圆的圆心

附录 5.1 展示了一些几何提示如何控制形状的示例。

B.3 作图步骤序列提取

第 3.1 节概述了作图步骤序列提取算法。下面将详细介绍该算法的各个阶段。

B.3.1 分析阶段

轮廓分析阶段类似于启发式自动约束器的第一步。系统会分析轮廓曲线的几何特性，并检测出以下几组几何实体

- 共线点组
- 平行直线组
- 圆角弧及其相邻曲线
- 对称线，以及关于这些对称线对称放置的点和曲线
- 同心圆或圆弧组

B.3.2 常用距离的识别

记录以下曲线对之间的距离

- 所有平行线对之间的距离
- 所有同心圆之间的半径差
- 所有圆和圆弧的半径

随后统计这些值出现的频率。优先为出现频率较高的长度值创建长度参数。

B.3.3 源线的识别

构造序列旨在从提示中定义的几何开始，逐步构建出构成轮廓所需的几何。为实现这一点，我们识别出以下“源线”，后续构造直线将通过一系列偏移由这些源线派生。

- 对称线。
- 由提示中提供的所有线段推导出的无限直线。
- 一条水平线和一条垂直线穿过尺寸基准。
- 对于迄今为止识别出的源直线，如果某组平行线既不与其中任何一条平行，也不与其反平行，则必须对该组进行处理。此时会创建一个 LineAxisRotatedLine 步骤，围绕尺寸基准旋转最近的坐标系轴线，从而生成一条与该组直线平行的新源直线。

B.3.4 平行线之间偏移的识别

给定一组有向平行线，只要带符号偏移距离适当，就可以从其中任意一条构造出另一条。我们会逐组处理平行线，以找出能够最佳构造该组直线的一系列偏移操作。

识别出可通过对称线的对称偏移构建的线对，并优先采用这些对称构造。

随后，算法尝试用一系列从“源线”开始的偏移来描述剩余组中的每条直线。以直线为节点、直线间的距离为边，构建一个无向完全图。然后从该图中删除边，首先删除对应于上述识别出的出现频率最低的距离的边。如果删除某条边会使图变得不连通，则不能删除该边。最后，确定直线的创建顺序。我们从源线开始遍历图，并按照 Dijkstra最短路径算法进行遍历。此遍历确定哪些直线是 LineOffsetLine步骤的输入和输出。

B.3.5 圆之间偏移的识别

与上文所述的平行线算法类似，根据整个轮廓中出现频率最高的长度值，创建同心圆/圆弧之间的偏移操作。每组同心圆都至少包含一个操作，用于从中心点和半径创建初始圆。

B.3.6 创建构造步骤

接下来，将创建一份详尽的构造步骤列表，涵盖已识别出的几何实体之间的所有关系。构造步骤的创建标准如下:

LineXLine：当轮廓的某个顶点与两条直线的交点重合时，会创建一个 LineXLine 步骤。输入直线的方向始终与轮廓中相应线段的方向一致。

直线-直线圆角步骤：当两条线段之间存在圆角弧时，会创建一个直线-直线圆角步 p

PointLineSymPoint：在轮廓中某个顶点可以通过将另一个顶点沿对称线镜像来定义的位置，会创建一对 PointLineSymPoint 步骤。哪个点作为原始点、哪个点作为镜像点，将在之后的图简化阶段确定。在此阶段，会分别以这两个不同的点作为输入，创建两个操作。

LineOffsetLine：对于按照附录 B.3.4 所述流程找到的每条直线及其偏移，都可以创建直线偏移步骤。

LineDatumParallelLineStep：当需要一条与“源直线”平行、且经过提示中给出的某条线段端点之一的直线时，会创建 LineDatumParallelLineStep。由于这些步骤不会引入新的参数值，因此会优先于 LineOffsetLine 步骤使用。

LineCircleParallelLine：构建一个圆的列表，其中包含所有提示孔的间隙圆盘，以及轮廓中的任何圆或圆弧。随后找出轮廓中与这些圆相切的直线。创建相应步骤，将源直线偏移到与圆相切的位置。与圆相切的直线有两种解，使用初始直线的方向来选择其中一种。与上述情况一样，这些步骤不会引入新的参数值，因此会被优先选用。

圆-点-点-圆弧：对于轮廓中每一段非由圆角定义的圆弧，都会创建一个步骤，根据底层圆以及该圆弧的起点和终点来确定其中点。方向标志用于指示选择顺时针还是逆时针圆弧。引入这些步骤后，只要已知端点，就可以从圆推导出圆弧。

LineXCicle：对于轮廓中每个处于直线与圆弧非相切交点处的顶点，都会创建一个 LineXCicle 步骤。

B.3.7 图补全

随后，利用这些构造步骤构建一个二分数据流图，其中几何和构造步骤作为节点，边表示几何流入和流出各项操作的过程。接着，在有向图中搜索没有入边的几何节点。提示中引入的几何预计不会有入边，

而其他没有入边的几何则存在问题。需要向图中添加新的构造步骤，以便从提示构造出这些几何。在许多情况下，可以通过引入 LineReverseLine 或 CircleReverseCircle 步骤，从现有几何创建直线或圆。只要中心点已构造出来，也可以使用 PointRadiusCircle 步骤创建圆。图剪枝阶段完成后，仍包含没有入边的几何的图将被丢弃。

B.3.8 参数

LineOffsetLine、CircleOffsetCircle、PointRadiusCircle 和 LineLineFilletStep 步骤需要输入长度参数，而 LineAxisRotatedLine 步骤需要输入角度参数。所需参数会被识别出来，并创建特殊的几何节点来表示其值。单个参数可以作为多个构造步骤的参数，例如，多个圆角使用相同半径是很常见的情况。

B.3.9 环路消除

初始数据流图中会包含循环，在继续处理之前必须将其消除。图中的循环通过深度优先搜索算法检测，并逐个处理。首先找出构成每个循环的边，然后根据以下标准选择要移除的边：

必要边：从定义最终轮廓所需的几何体开始，我们检查其入边。如果只有一条入边，则该几何体由单个构造步骤的输出派生而来。随后必须将该边加入“必要边”集合。接着，我们转到产生这些必要边的构造步骤，并重复此过程。整个图都会被遍历，从而找到完整的必要边集合。当到达具有多条入边的几何节点时，遍历停止。我们无法从图中移除任何必要边。

距提示的跳数：我们希望为每个几何对象找到尽可能短的构造序列。从提示几何出发，我们使用Dijkstra算法，找出图中到达每个节点所需的最少跳数。由于图中包含环，因此必须注意避免重复访问节点。

构造步骤优先级：构造步骤的优先级如下。

- 将圆弧创建为圆角的优先级最高
- 其次是创建与圆相切的直线
- 接着是通过直线与圆的交点创建点
- 然后是对直线或点执行镜像操作
- 最后，所有其他操作

随后通过从每个循环中移除一条边来打破循环。必要边绝不会被移除。优先移除非必要边中距提示几何跳数最多的节点所对应的边。然后，以每条边末端操作的优先级作为平局判定依据。

B.3.10 图剪枝

在传统约束求解器中，具有多条入边的几何节点等同于过约束但一致的几何。图剪枝算法会为每个几何选择一条入边，并移除通向其他解的图分支。选择依据是对图中每条边计算的代价函数。

- 提示中提供的几何代价为零
- 构造步骤会使所有输出边的代价增加 0.1
- 参数的代价为 1.0。
- 任何未由提示定义且没有前驱节点的几何，其代价均为无穷大

计算出各边的成本后，我们遍历几何节点。对于入边多于一条的几何节点，保留成本最低的边，并移除其他所有边。随后移除图中所有未以轮廓几何为终点的分支。如果无法利用提示中的几何构造出创建最终形状所需的任何几何，便会将相应轮廓从数据集中剔除。

B.3.11 序列排序

现在，数据流图包含了创建构造序列所需的全部信息。首先，使用字典序拓扑排序对图中的节点进行排序。此排序遵循图所定义的几何与构造步骤之间的依赖关系，同时允许通过次级标准消除排序歧义。主要标准是曲线在最终轮廓中出现的顺序；若出现并列，则使用中间几何的字典序排列来打破平局。无限直线则根据其与轮廓边界框相交处中字典序最小的点进行排序。

随后，根据首次使用这些参数的构造步骤，对定义形状所需的参数进行排序。每个参数都会根据该顺序分配一个索引，该索引用于附录 A.4 中所述的 UseParameterN 标记。

C 其他结果

C.1 强化学习算法

用于强化学习模型对齐的超参数详见表 9。训练在 3 个 NVIDIA A10 GPU 上进行，使用 AdamW 优化器，学习率为 1×10^{-6}，共执行一千步。图 10 展示了所使用的不同强化学习算法在各训练步数上的性能。事实证明，应用 KL 惩罚至关重要。若不使用该惩罚，模型会通过持续生成无效序列来利用奖励函数。由于无效序列的奖励会被屏蔽，只有针对极其简单提示的生成结果才能获得非零奖励，尽管没有进行有意义的学习，却导致平均奖励轻易达到很高水平。

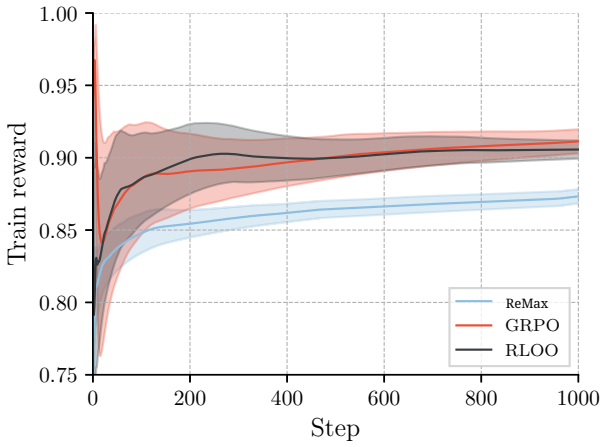


图 10：强化学习训练奖励

表 9：不同强化学习算法的超参数比较。

超参数	ReMax	GRPO	RLOO
有效批量大小	768	144	144
组样本大小	–	16	16
学习率	1e-6	1e-6	1e-6
Top-p采样	0.3	0.3	0.3
词元级KL惩罚	0.1	–	0.1
序列级KL惩罚	–	0.01	–
策略裁剪比率	–	0.2	–

C.2 构造步骤精度

由于每个构造步骤都表示一个具有明确定义输出几何的原子操作，因此我们可以衡量 Transformer 学习执行构造步骤的准确度。表10展示了所有构造步骤模型的这些数值。距离是在构建轮廓的单位正方形中测量的。角度以弧度为单位进行测量，布尔标志之间的比较结果为：当标志一致时为 1.0，不一致时为 0.0。所有数值均是在测试集中的 6427 个提示上取平均得到的。

我们观察到，Transformer 能够以完美的准确度学会精确复制令牌。例如，反转圆时，只有 IsCCW 标志会发生变化，而中心点和半径均完全吻合。对于以单个词元作为输入和输出的构造步骤，其精度也很高，因此反转直线时，到原点的距离和角度也能很好地吻合。

涉及多条曲线作为参数的复杂几何操作，误差较大。例如，直线-直线和直线-圆相交（LineXLine 和 $\text{LineXCircle}_{\text{dist}}$），以及创建圆角弧的中点（$\text{LineLineFillet}_{\text{precision}}$）。

将强化学习算法应用于这些序列，效果甚微。这表明，网络主要是在监督预训练期间学会如何执行几何操作。强化学习奖励随后帮助网络将各个操作组合成 CAD程序，从而生成语法和几何上均有效的轮廓，而非提升其执行底层几何计算的能力。

表10：不同模型之间构造步骤指标的比较

指标	**构建步骤模型**	**构建步骤模型 (ReMax)**	**构建步骤模型 (GRPO)**	**构建步骤模型（RLOO）**
$\text{CircleOffsetCircle}_{\text{center}}$ (↓)	**0.0000**	**0.0000**	**0.0000**	**0.0000**
$\text{CircleOffsetCircle}_{\text{radius}}$ (↓)	0.0109	**0.0107**	0.0124	0.0133
圆-点-点-圆弧$_{\text{dist}}$ (↓)	0.1413	0.1438	0.1413	**0.1102**
$\text{CircleReverseCircle}_{\text{ccw}}$ (↑)	**1.0000**	**1.0000**	**1.0000**	**1.0000**
$\text{CircleReverseCircle}_{\text{center}}$ (↓)	**0.0000**	**0.0000**	**0.0000**	**0.0000**
$\text{CircleReverseCircle}_{\text{radius}}$ (↓)	**0.0000**	**0.0000**	**0.0000**	**0.0000**
解存在性 (↑)	0.9981	0.9983	0.9987	**0.9988**
$\text{LineAxisRotatedLine}_{\text{angle}}$ (↓)	**0.0204**	0.0242	0.0243	0.0226
$\text{LineAxisRotatedLine}_{\text{dist}}$ (↓)	**0.0130**	0.0148	0.0132	0.0139
直线-圆平行线$_{\text{angle}}$ (↓)	0.0017	0.0059	0.0020	**0.0009**
直线-圆平行线$_{\text{dist}}$ (↓)	0.0239	0.0248	0.0236	**0.0231**
直线-基准-平行线$_{\text{angle}}$ (↓)	0.0020	0.0019	0.0022	**0.0017**
直线-基准-平行线$_{\text{dist}}$ (↓)	**0.0061**	0.0063	0.0066	0.0066
直线-直线圆角$_{\text{precision}}$ (↓)	**0.0347**	0.0367	0.0362	0.0365
直线-直线圆角$_{\text{valid circle}}$ (↑)	0.9320	0.9372	**0.9522**	0.9504
直线偏移直线$_{\text{angle}}$ (↓)	0.0040	0.0040	**0.0039**	**0.0039**
直线偏移直线$_{\text{dist}}$ (↓)	0.0095	0.0097	0.0095	**0.0094**
$\text{LineReverseLine}_{\text{angle}}$ (↓)	0.0015	0.0007	**0.0006**	0.0007
$\text{LineReverseLine}_{\text{dist}}$ (↓)	**0.0001**	**0.0001**	**0.0001**	**0.0001**
直线-圆$_{\text{dist}}$ (↓)	0.0866	0.0870	**0.0745**	0.0787
LineXLine (↓)	0.0245	0.0203	0.0194	**0.0184**
点-直线对称点 (↓)	0.0060	0.0066	0.0057	**0.0049**
点半径圆$_{\text{center}}$ (↓)	**0.0000**	**0.0000**	**0.0000**	**0.0000**
点半径圆$_{\text{radius}}$ (↓)	**0.0000**	**0.0000**	**0.0000**	**0.0000**
对称直线偏移直线直线$_{\text{angle}}$ (↓)	**0.0003**	0.0004	**0.0003**	0.0005
对称直线偏移直线直线$_{\text{dist}}$ (↓)	**0.0055**	**0.0055**	0.0056	0.0056

D 生成构造序列示例

本节展示模型生成的一些额外构造序列，并将其与驱动参数变化时形状的参数行为联系起来。

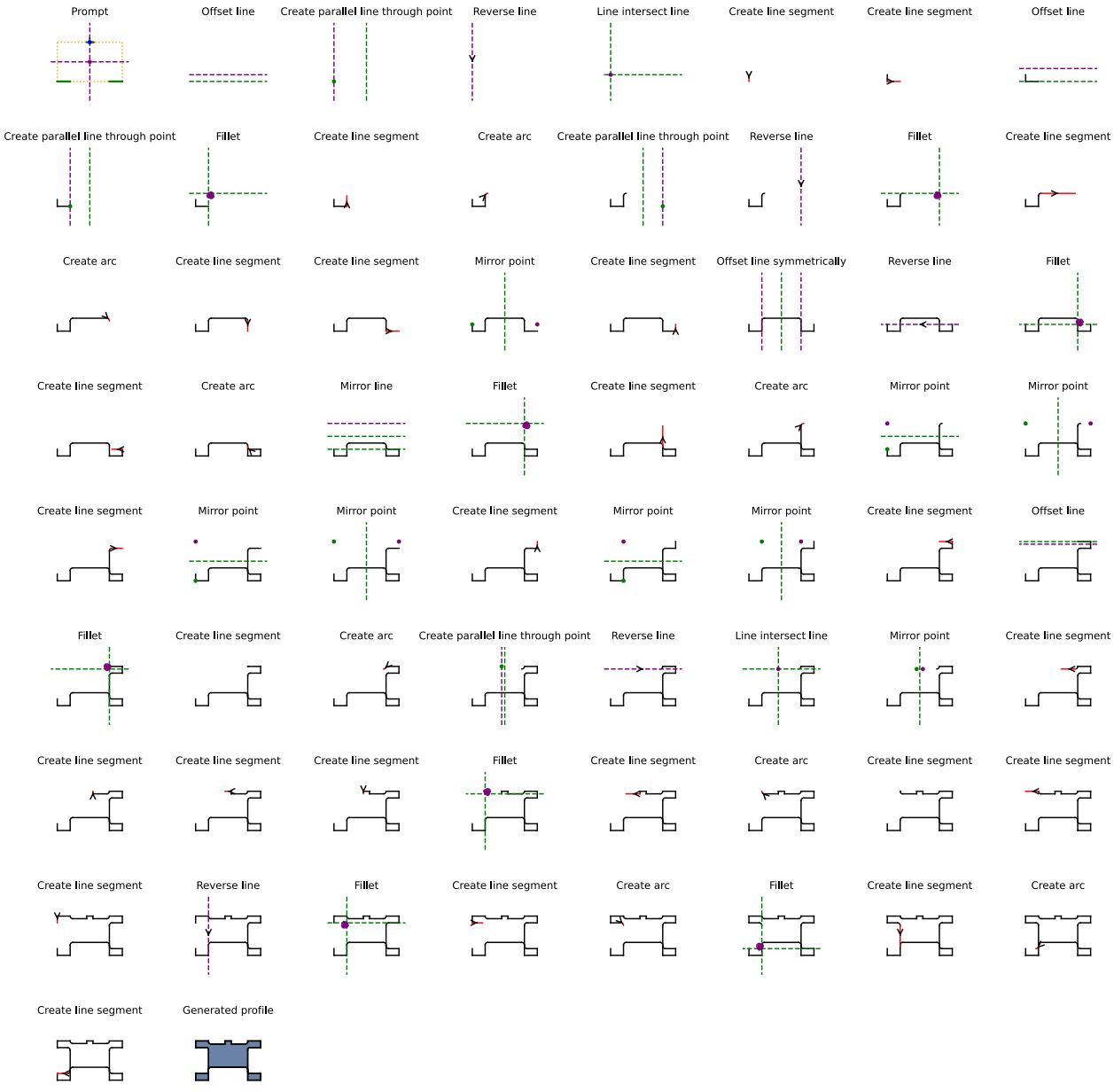


图 11：同时包含水平线和垂直线的示例的构造序列。提示中的点关于两条对称线进行镜像，并用于构建最终形状。

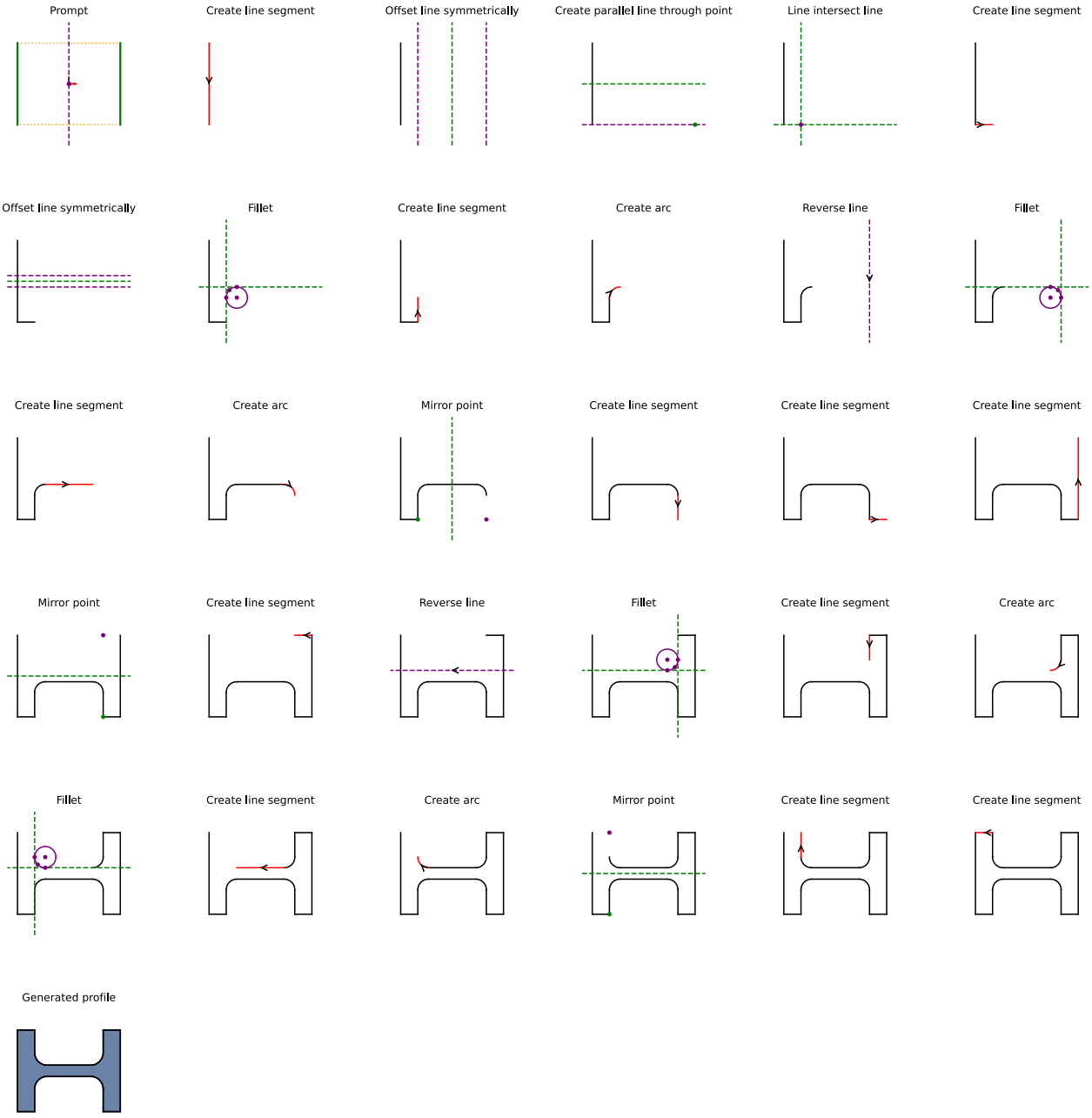


图 12：工字梁形状的构造序列。请注意，垂直对称线已定义，但尺寸基准位于中心位置，因此隐含了一条水平对称线。

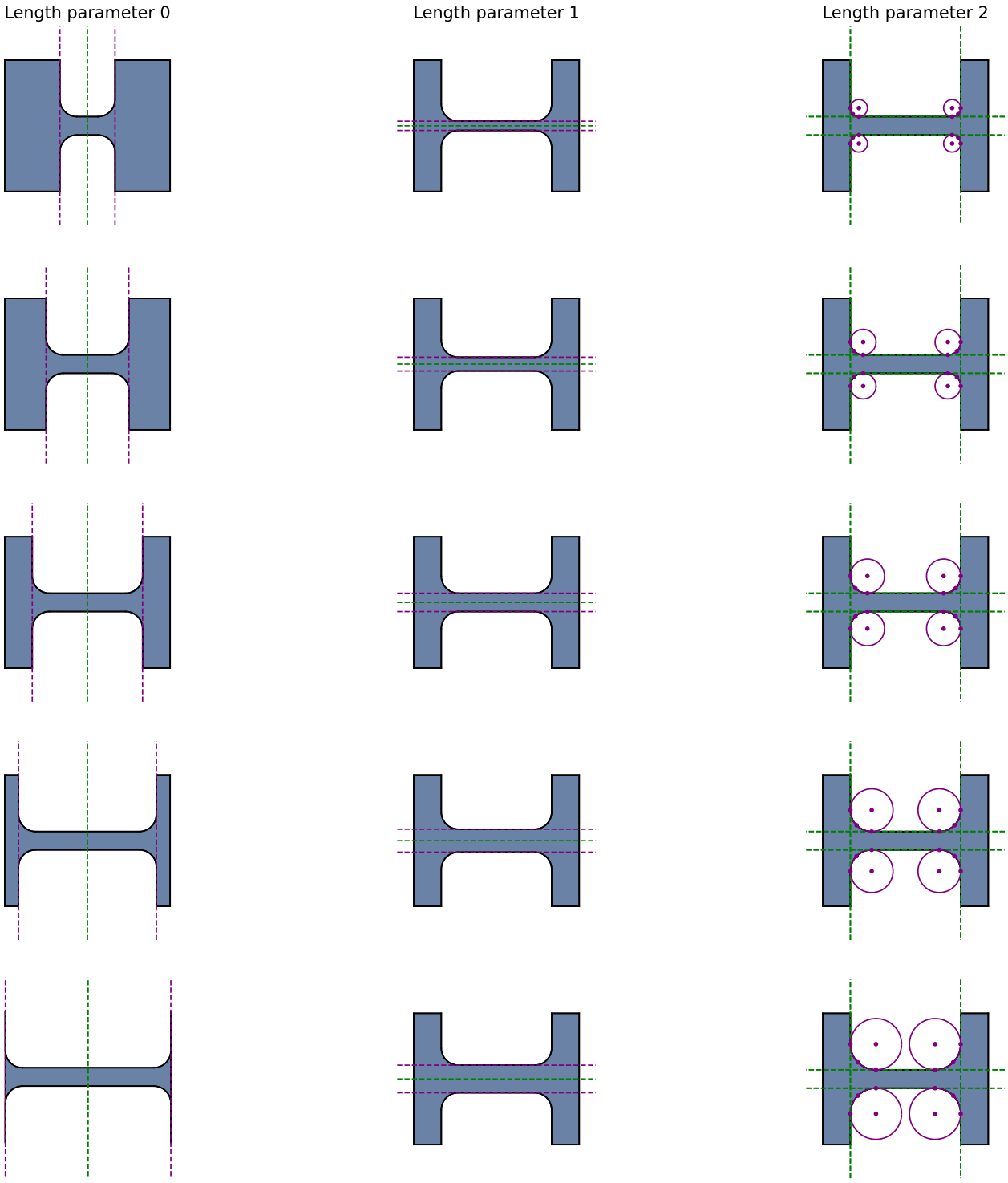


图13：工字梁的参数变化。该轮廓由三个值进行参数化，分别用于控制两侧侧翼的宽度、中央梁的宽度以及四个圆角弧的半径。

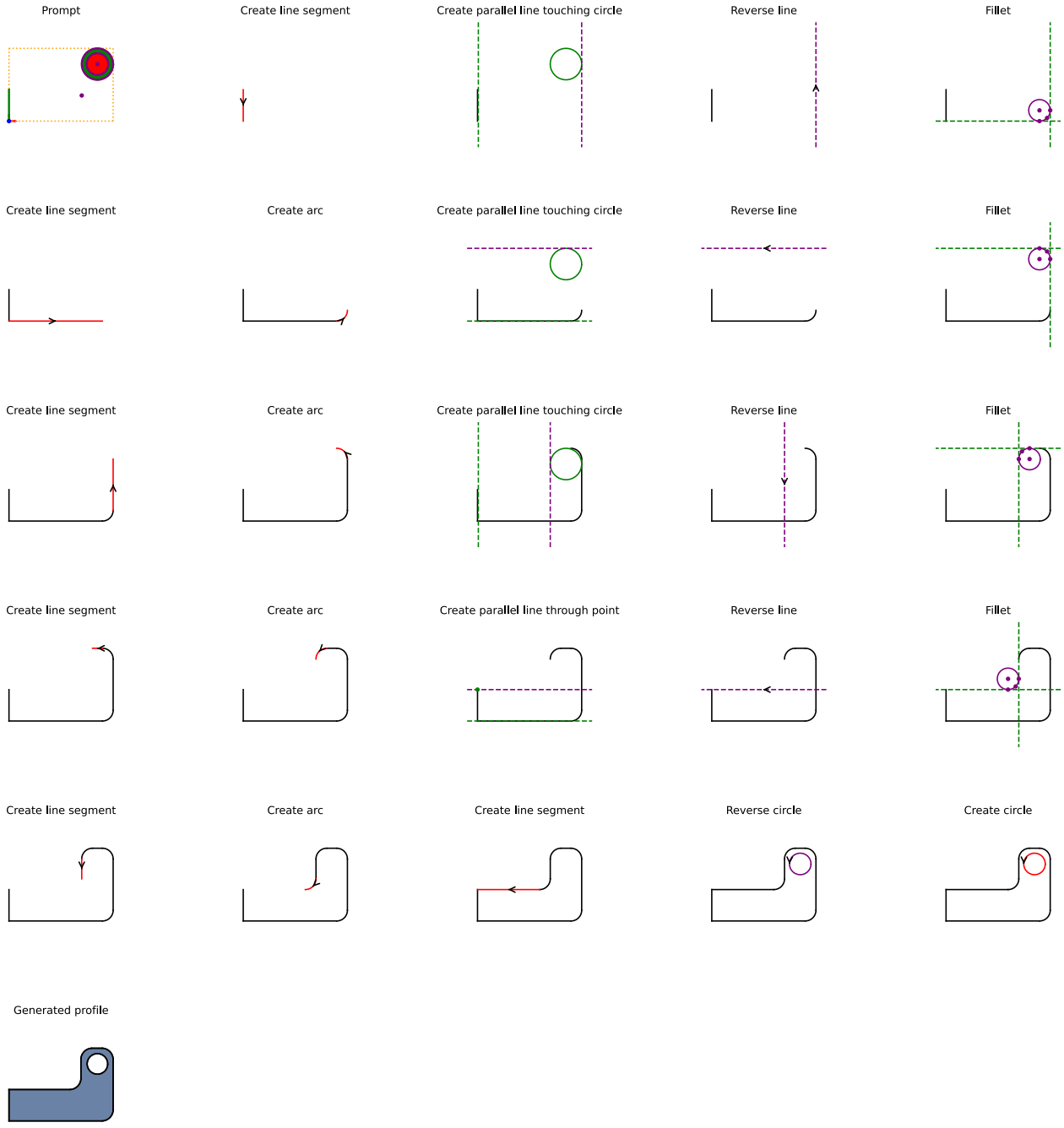


图14：一个没有自由参数的构造序列示例。请注意，垂直线是通过偏移轴线，直到它们与间隙圆盘相切而创建的。水平线则根据提示中给出的线段端点和尺寸基准确定。

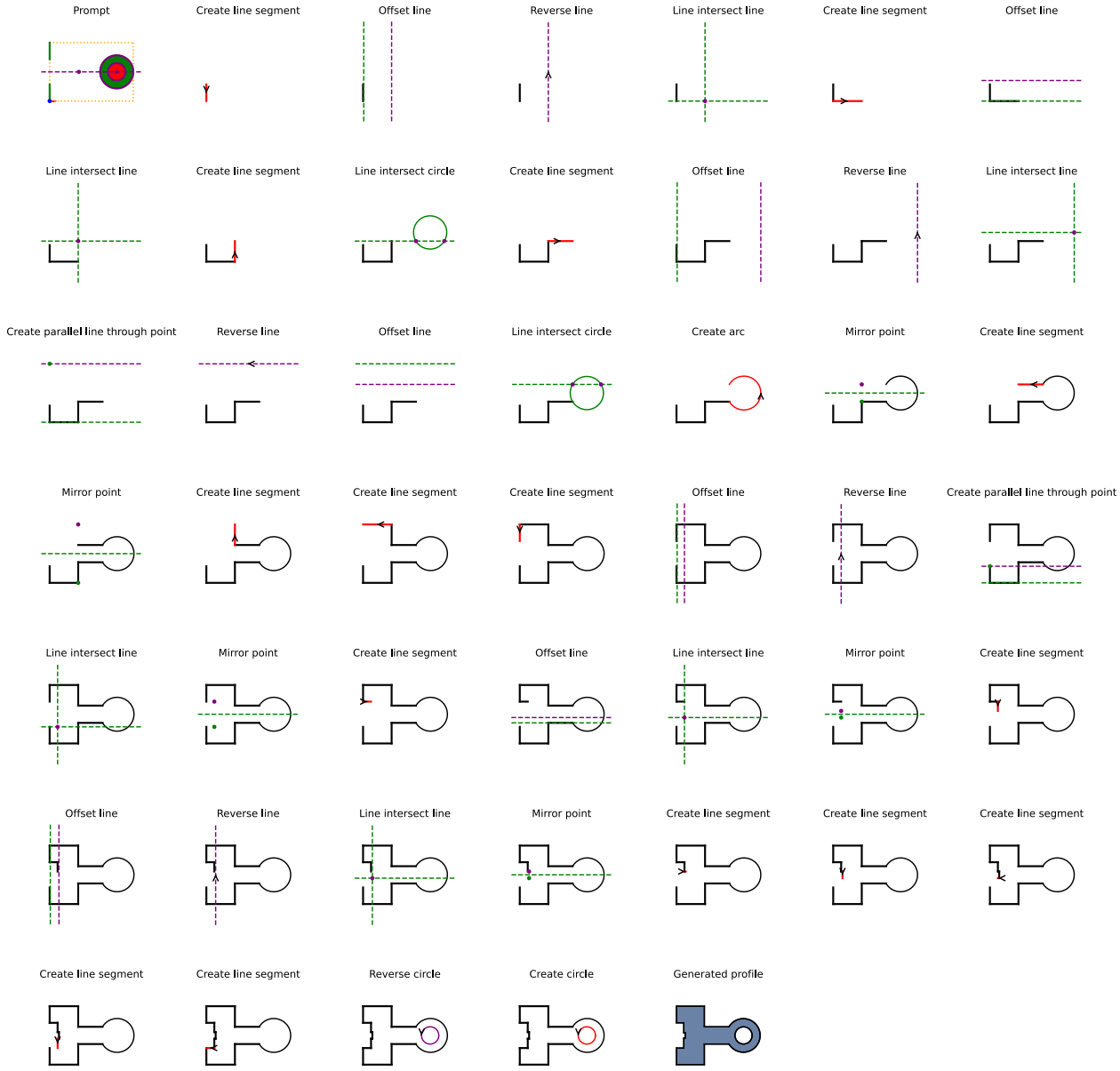


图15：更复杂形状的构造序列。

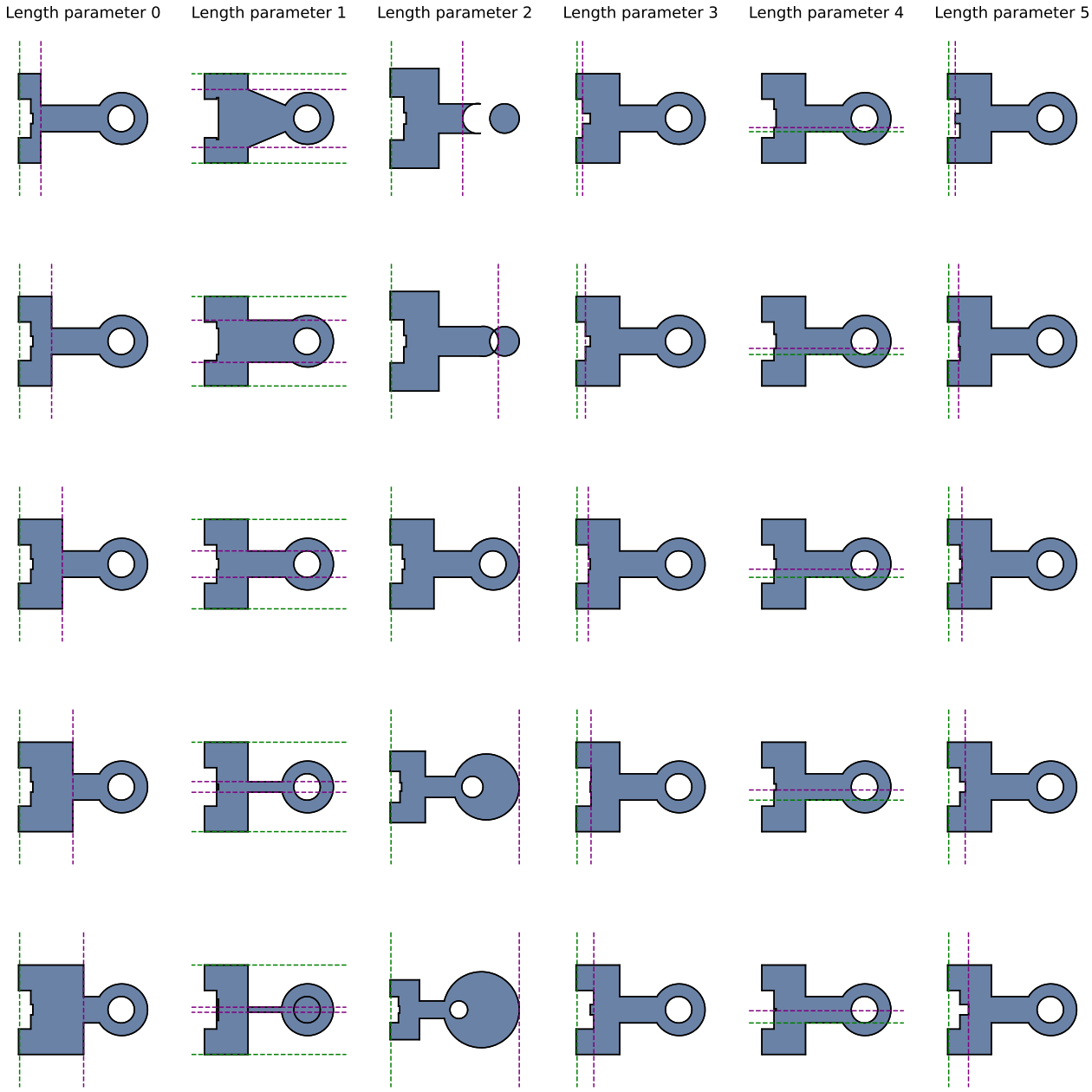


图16：复杂形状的参数变化。我们看到，大多数情况下参数化是合理的，不过这个示例确实展示了两种失效模式。当参数1使线段移动到无法与圆相交的位置时，此操作便会失败。点不会更新，从而导致出现斜线。参数2定义了一个直线偏移，预期位于形状的右侧。此直线与中心线相交，以确定外圆弧的中点。移动此直线后，外圆弧与内圆不再同心，暴露出参数化存在的问题。如何改进所得轮廓的参数化行为，有待后续工作研究。