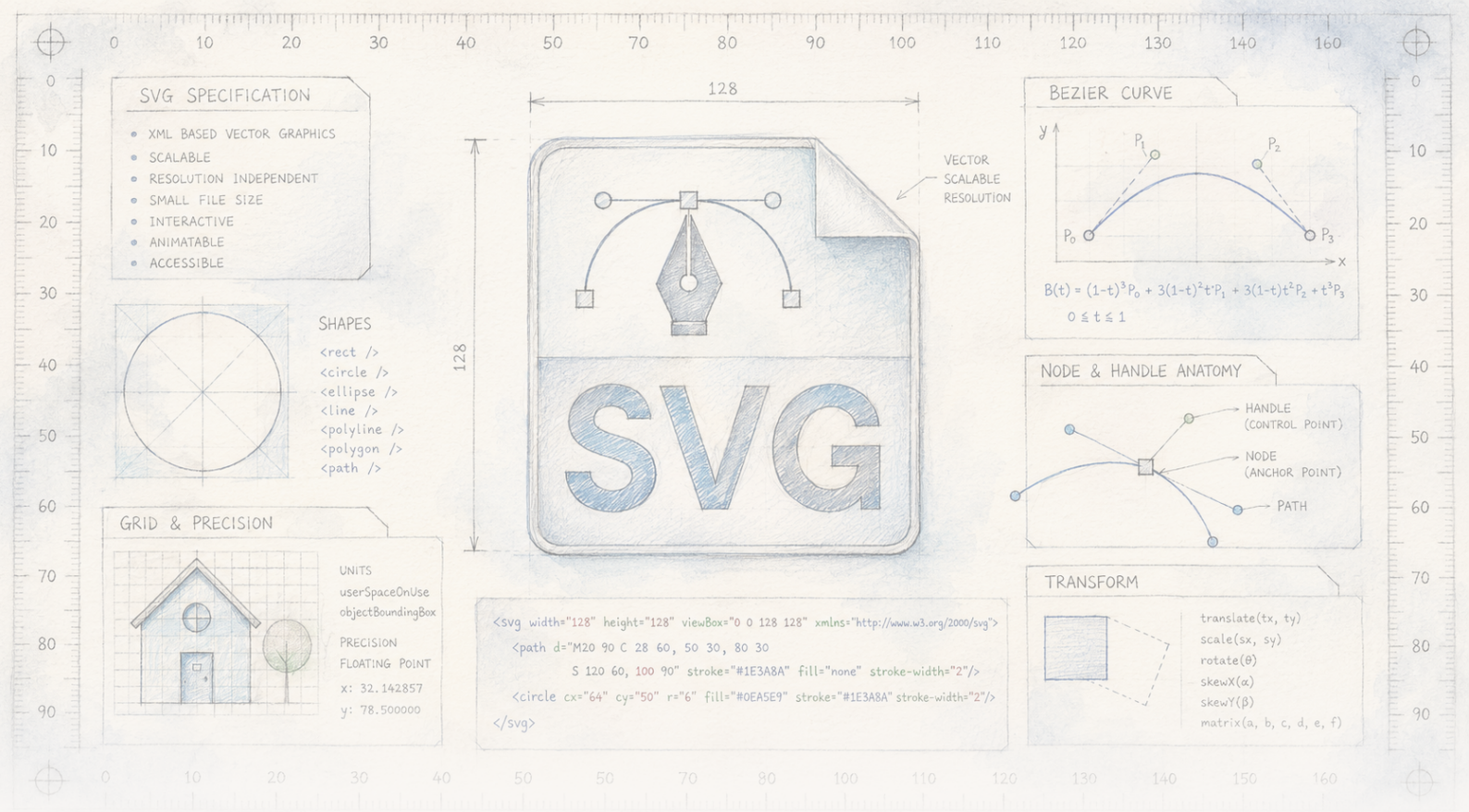


The Clean Paper

WHAT THE PAPER SAYS · WHAT IT DOESN'T · WHY IT MATTERS

教绘图 AI 看着画布作画

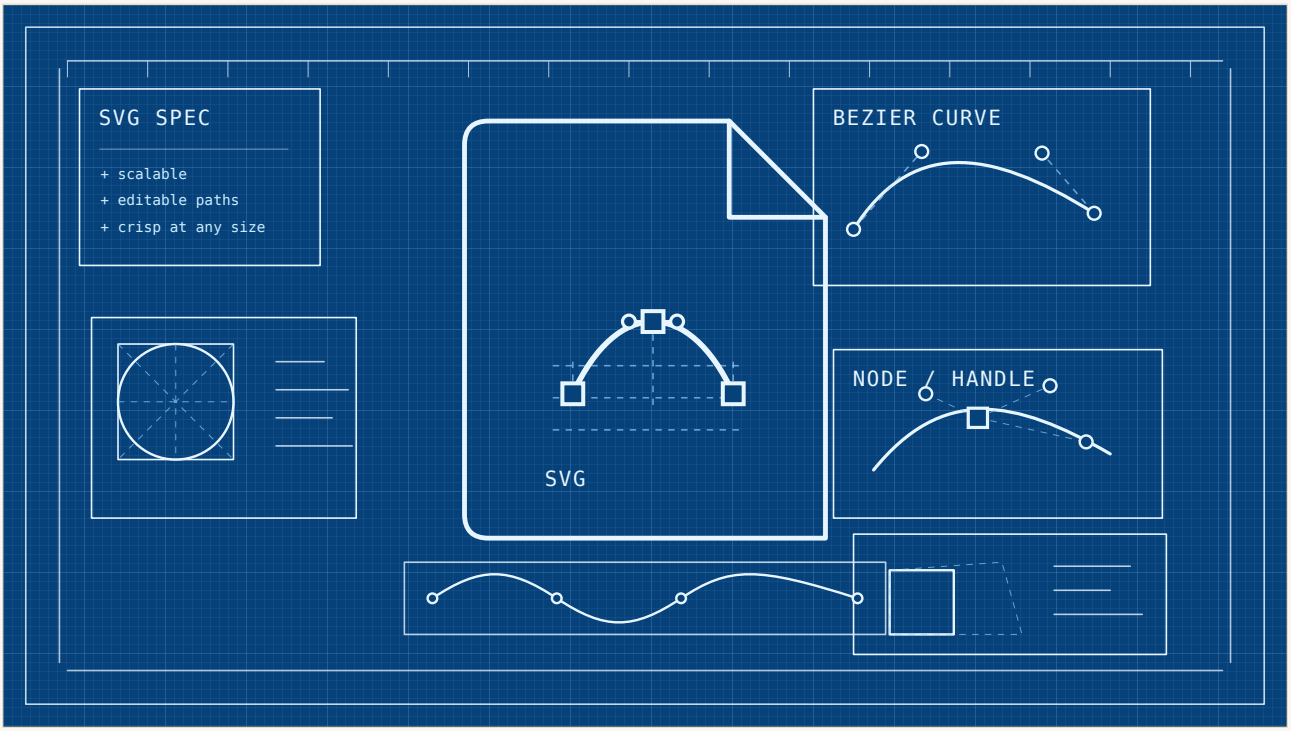
生成矢量图形的语言模型一直在盲画——写出绘图命令，却从未看见结果。一种新方法让模型逐笔看着自己的画布被填满，而诚实的转折是，简单地赋予它眼睛反而让结果变差：模型必须经过再训练，学会怎样使用视觉反馈。回报是一个能追平或略微胜过使用最多二十倍数据训练的对手的模型——这是在单一基准上得到的真实、谨慎的结果，不是一场革命。

Written by Vera Rubin · Cross-checked by Michele Renda · Edited by Michele Renda · Figures by Laura Nesso · AI-assisted, human-reviewed

Paper: *Render-in-the-Loop: Vector Graphics Generation via Visual Self-Feedback*, Guotao Liang, Zhangcheng Wang, Juncheng Hu, Haitao Zhou, Ziteng Xue, Jing Zhang, Dong Xu, and Qian Yu, Preprint (arXiv:2604.20730)

闭着眼睛画画

让当今的一个 AI 模型为你画张图，底层会发生两种截然不同的事情之一。人们熟悉的图像生成器——那些能从一句话变出一张照片的模型——直接绘制像素，而且工作时看得见画布。但还有第二种更安静的绘图方式，模型根本不直接作画，而是编写指令：在这里画个圆，在那里画条线，把这个形状填成蓝色。这些指令是代码——也就是支撑网络上大多数图标和标志的可缩放矢量图形（SVG）——它的吸引力很真实：结果不是固定的像素网格，而是一组可以无限缩放、改色和编辑、却永远不会变模糊的形状。



原创 SVG 蓝图作品：图像本身就是可编辑的矢量文件，由路径、网格线、节点和控制柄构成，而不是固定像素。

Laura Nesso / The Clean Paper Permission on file

问题在于，直到最近，模型编写这种绘图代码时都是盲画。它会一次性输出整串指令——圆、线、填充——从不渲染出来看看自己画了什么。想象一下闭着眼睛画一张脸：眼睛大概会落在该有的位置，但你无从发现第二只眼睛画到了脸颊上，也不会知道为头发画的形状此刻正盖在鼻子上。这些模型过去的工作方式大致如此，因此它们经常生成完全有效、视觉上却一团糟的代码。

梁国涛带领的团队如今做了一件近乎滑稽地显而易见的事：让模型睁开眼睛。他们的方法 *Render-in-the-Loop* 会在每一步之后渲染尚未完成的图画，并在模型落下下一笔之前把图像反馈给它。真正有意思的并不是这样做有所帮助，而是他们必须怎样做，才能让它真正有所帮助。

怎样给一幅画评分？

当一篇论文说自己的模型“击败”了另一个模型时，追问一句很公平：在哪方面击败，又是怎样测量的？判断生成图像的质量确实很难——“画一台笔记本电脑”没有唯一正确答案——因此这个

领域依赖少数几种自动评分，每一种都只是对真人观察结果的不完美替代。
本文采用了其中几种。**FID** 把一整批生成图像的总体统计风格与真实图像比较；数值越低越好，但它描述的是整批图像，而不是某一张。**CLIP 分数**让另一个 AI 判断图像是否符合提示中的文字——有用，但其敏锐程度不会超过这个评判者本身。**DINO**、**SSIM** 和 **LPIPS** 则在从原始像素到学习特征的不同层面上，将重建图像与目标图像进行比较。
这些评分都不是真理，只是代理指标，而且变化通常很小。当你看到一个模型得 127.6 分，另一个得 128.8 分时，这确实是朝预期方向的真实差异——但它只是轻轻一推，不是压倒性胜利，而且这个数字本身与人眼判断的对应也很松散。看到“击败使用二十倍数据训练的模型”这类标题时，值得记住这一点。

作者做了什么

作者着手解决的问题就是这种盲画。现有模型把编写 SVG 视为纯文本任务：根据已有代码预测下一段代码，从不进行渲染。这让现代多模态模型本已具备的强大“眼睛”——视觉编码器——闲置不用。Render-in-the-Loop 把任务重构成逐步进行的视觉过程。每生成一段绘图代码，就把部分 SVG 渲染成图像，再以图片形式反馈给模型，让模型真正看着当前画布来选择下一段代码。

他们的第一个发现带有警示意味，而且值得肯定的是，他们直言不讳地报告了这一点：仅仅把这个循环接到现成模型上并不起作用。研究人员在没有专门训练的情况下，把中间渲染结果输入强大的通用模型，质量不但没有提高，反而全面下降。一个从未学过怎样用眼睛完成这项工作的模型，不会突然无师自通。

因此，大部分工作都在于教学。团队重建训练数据，把每幅画拆分成许多细小而具有视觉意义的步骤——把复杂形状拆成更简单的部件，使每个阶段都有新内容可看——然后使用这些分步序列，对一个基于 Qwen3-VL、拥有 80 亿参数的开放模型进行微调。他们把这称为视觉自反馈（Visual Self-Feedback）。绘图时又加入第二套机制 Render-and-Verify：在接受每一笔之前，模型先将其渲染并检查它是否真的改变了图像，还是只重复了上一笔。没有增加任何内容的笔画会被丢弃；当再也没有什么能改善图像时，模型会收到停止提示。值得注意的是，这一切只使用了相当小的数据集——约 850,000 个样本，不到一个竞争对手所用数据的一半，也只是另一个对手所用数据的一小部分。

他们发现了什么

- 经过这种训练，模型比其盲画版本表现更好——在失败案例中最为明显：盲画模型会把眼睛画在脸颊上，或跳过提示要求的条形图，转而输出一个普通显示器。
- 在标准基准 MMSVGBench 上，该方法与强劲对手相当，并在若干指标上略微领先——包括训练数据超过其两倍的OmniSVG，以及训练数据约为其二十倍的InternSVG。
- 优势幅度很小。例如在图标数据集上，它的主要图像质量分数为 127.6，最佳对手为 128.8；它的提示匹配分数为 0.293，对手为 0.291——真实且一致，但很微弱。
- 两个新增部分都不可或缺：关闭专门训练或绘图时验证，分数都会出现可测量的下降。尤其是验证步骤，可以阻止模型陷入反复重画同一内容的循环。

- 作者自己强调的结果是效率——用远少于领先模型的数据，取得了这样的表现。

这项研究没有证明什么

- 它**没有**表明让模型“看见”就能免费获益。事实上恰恰相反：论文自己的实验显示，不经过再训练的视觉反馈会让结果更差。提升来自再训练，而不是单靠眼睛。
- 它**没有**确立巨大或决定性的领先。在大多数分数上，该方法与对手不相上下；“击败使用 20 倍数据训练的模型”确实属实，但只是在一个基准的代理指标上轻微领先。
- 它**没有**展示通用艺术能力。这是一个 80 亿参数的研究模型，以较小的固定分辨率（224×224 像素）绘制图标和简单插图，不是通用设计师。
- 与大型通用模型 GPT-5 的比较**并非同类比较**：该模型并不是为这种狭窄的代码绘图任务构建或调优的，因此在这里胜过它，并不能说明两个模型的总体能力。
- 它**不是**没有代价。在每一步都渲染并重新读取画布，会比一次性盲目输出代码更慢——作者也承认这一成本。

证据有多强

- 在论文自身设定下进行受控比较的部分，证据**扎实**。消融实验很干净：移除训练或移除验证，分数都会下降——因此这两个组成部分确实在发挥作者所说的作用。
- 论文**诚实面对自己的意外发现**。朴素视觉反馈反而有害这一结果被报告出来，而不是藏起来；它也是论文中最有意思的一点——有力地修正了“输入越多总是越好”的直觉。
- 涉及排行榜主张的部分则**较薄弱**。相对于使用更多数据的对手，领先幅度很小，而且只出现在单一基准上；这个基准还是由其中一个对手模型的作者建立的。在一个测试集的代理指标上小幅领先，具有提示性，却不是定论。
- **尚未在大规模和真实环境中测试**。这里的一切都限于低分辨率图标和简单插图。同一种思想是否适用于复杂、高分辨率或真实世界的设计工作，仍留待未来研究——作者也明确这样说。

为什么重要

这篇论文的核心思想简单得几乎令人尴尬，却远远超出绘图本身。如果一个程序要通过编写代码来生成东西——网页、图表、示意图、3D 场景——它可以闭着眼睛把全部内容写完再碰运气，也可以边渲染边修正方向。对人而言，闭合这个反馈回路是自然而然的；我们会不断瞥一眼页面。对这些模型来说，这却是一个出乎意料地晚近的做法。

论文值得一读，是因为它在这个想法后面加了一个星号。让模型能够看见，不等于教会它观察。必须训练这双眼睛，反馈回路才会带来回报——即便如此，回报也真实但有限：得到的是一位更稳定的绘图员，而不是一种全新的艺术家。对于任何构建“把描述变成可编辑图形”工具的人——比如最终成为应用图标和插图的那类东西——真正有用的是这条安静而实际的教训。这里的胜利来自更便宜、更聪明的训练，而不是更多数据。这比排行榜上再多一个点更值得学习。

清晰摘要

生成矢量图形的语言模型——也就是生成大多数网络图标和标志背后可编辑、可缩放代码的模型——传统上一直在“盲画”：写出全部绘图命令，却从不渲染出来查看结果。梁国涛带领的团队提出 Render-in-the-Loop：每一步之后渲染尚未完成的图画并反馈给模型，让它看着画布落下下一笔。他们最核心、也最坦诚的发现是，简单地把这一做法应用到现有模型上，反而会让模型变差；只有经过再训练、学会利用视觉反馈，并辅以丢弃无效笔画的绘图时检查，性能才会提升。再训练后的 80 亿参数模型在一项标准基准上追平或略微胜过使用最多二十倍数据训练的对手——这是真实结果，但只在低分辨率图标和简单插图的代理指标上小幅领先。作者强调、也值得记住的结论关乎效率：善于观察自己的作品，可以在一定程度上替代纯粹的数据规模。

不绕弯核查

论文证明了什么：对矢量图形模型进行再训练，使其逐步渲染并观察自己尚未完成的图画，会比盲画产生更好、更完整的结果——在一项标准基准上，用更少的训练数据达到与大数据对手相当、并略微领先的表现。

什么合理但尚未证明：“观察自己的作品”广泛而言是比扩大数据规模更好的方案；同一种思想能帮助复杂、高分辨率或真实世界的图形任务；基准上的微小优势反映了人能实际察觉的差异。

它没有证明什么：视觉反馈单独使用就有帮助（不经再训练反而有害）；对竞争模型的领先巨大或具有决定性；拥有通用绘图能力；与 GPT-5 等并非为此任务构建的通用模型进行的是公平正面对比。

主要局限：只使用一个基准，且该基准部分由一个对手模型的作者建立；代理指标上的优势很小；80 亿参数模型，绘制 224×224 的图标和简单插图；逐步渲染循环导致生成速度较慢。

普通读者应该有多大信心？可以高度确信，闭合反馈回路——边画边渲染和重新观察——确实有帮助，而且这种能力必须通过训练获得，不能简单开启。对于这个特定模型是否决定性地胜过对手，应持低至中等信心；“击败使用 20 倍数据的模型”是一个真实但幅度不大、仅限单一基准的结果。最值得记住的思想可信度很高：对于用代码绘图的模型，边画边看胜过闭眼作画。

来源

Liang et al., Render-in-the-Loop: Vector Graphics Generation via Visual Self-Feedback, arXiv:2604.20730 (2026)
https://arxiv.org/abs/2604.20730

OmniSVG project page
https://omnisvg.github.io/

InternSVG project page
https://hmwang2002.github.io/release/internsvg/