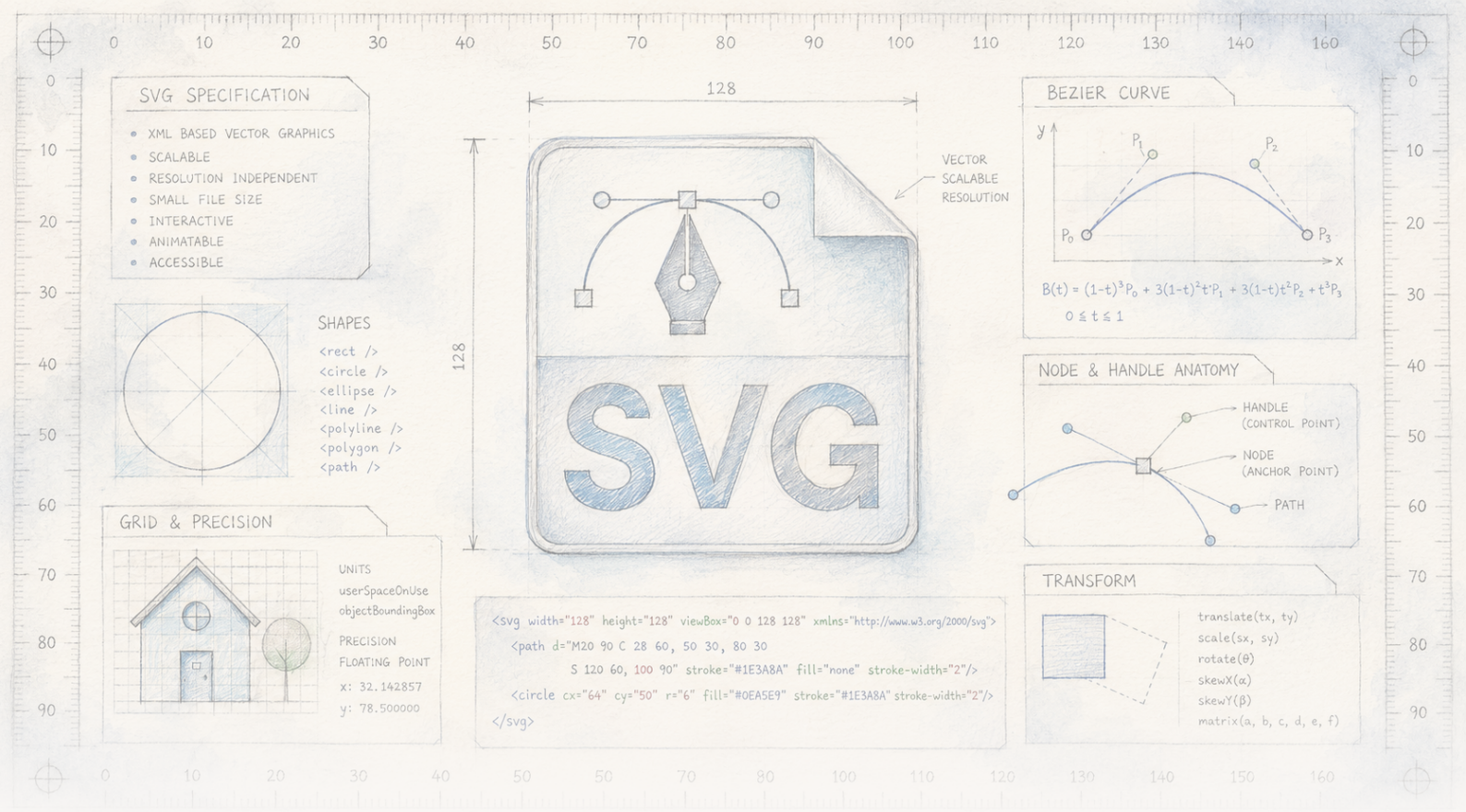


The Clean Paper

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

به AI یاد بده هنگام رسم به کار خودش نگاه کند — اما فقط «چشم بازکردن» کافی نیست

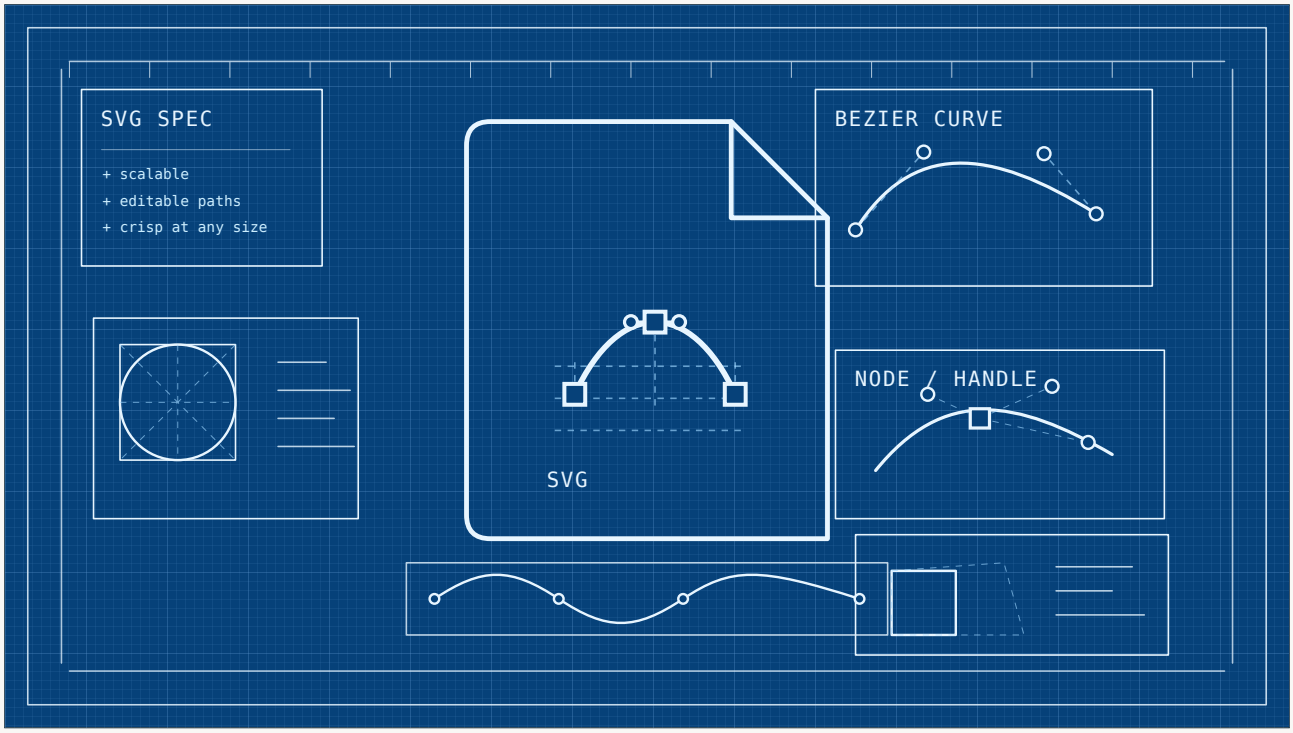
مدل‌های تولید *SVG* معمولاً تمام کد رسم را یک‌باره و بدون دیدن نتیجه می‌نویسند. *Render-in-the-Loop* پس از هر گام تصویر نیمه‌تمام را رندر می کند و به مدل بازمی گرداند. نکتهٔ جالب این است که بازخورد بصری ساده روی مدل آماده کیفیت را بدتر کرد؛ سود فقط پس از بازآموزی مرحله‌به‌مرحله و افزودن *Render-and-Verfiy* ظاهر شد. مدل *8B* حاصل، با حدود ۸۵۰ هزار نمونه، روی *MMSVGBench* با رقبایی که تا ۲۰ برابر داده دیده‌اند برابری یا اندکی برتری دارد — آن هم با حاشیه‌های کوچک روی سنجه‌های جانشین. درس اصلی: دیدن کار خود قابلیتی نیست که خودبه‌خود مفید شود؛ باید برای استفاده از آن آموزش دید.

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

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

نقاشی کردن با چشم‌های بسته

اگر از یکی از مدل‌های هوش مصنوعی امروز بخواهید تصویری بکشد، در پشت صحنه معمولاً یکی از دو اتفاق بسیار متفاوت می‌افتد. مولدهای تصویر آشنا — همان‌هایی که از یک جمله عکس می‌سازند — مستقیماً پیکسل‌ها را نقاشی می‌کنند و هنگام کار می‌توانند بوم را «ببینند». اما نوع دوم و آرام‌تری از رسم وجود دارد که در آن مدل اصلاً پیکسل نقاشی نمی‌کند. دستور می‌نویسد: اینجا دایره بکش، آن‌جا خط بکش، این شکل را آبی پر کن. این دستورها کد هستند — همان Graphics Vector Scalable یا SVG که پشت بیشتر آیکون‌ها و لوگوهای وب قرار دارد — و جذابیتش واقعی است: نتیجه شبکه‌ای ثابت از پیکسل‌ها نیست، بلکه مجموعه‌ای از شکل‌هاست که می‌توان بی‌نهایت بزرگ و کوچک، recolor و ویرایش کرد بدون آن که تار شود.



اثر blueprint اصلی در قالب SVG: خود تصویر یک فایل برداری قابل‌ویرایش است که از pathها، خطوط شبکه، nodeها و handleها ساخته شده، نه پیکسل‌های ثابت.

Laura Nesso / The Clean Paper Permission on file

مشکل این بود که تا همین اواخر، مدلی که این کد رسم را می‌نوشت کور بود. کل دنبالهٔ دستورها را یک‌باره تولید می‌کرد — دایره، خط، fill — بی‌آن که آن‌ها را render کند و ببیند چه ساخته. تصور کنید صورت کسی را با چشم بسته طراحی کنید: شاید چشم اول تقریباً جای درست باشد، اما راهی ندارید بفهمید چشم دوم روی گونه افتاده یا شکلی که برای مو کشیدید حالا روی بینی نشسته. تقریباً همین‌طور این مدل‌ها کار می‌کردند؛ به همین دلیل اغلب کدی کاملاً معتبر تولید می‌کردند که از نظر بصری به‌هم‌ریخته بود.

گروهی به سرپرستی Liang Guotao حالا کار تقریباً خنده‌داری از شدت بدیهی‌بودن انجام داده‌اند: اجازه داده‌اند مدل چشمش را باز کند. روششان، *Render-in-the-Loop*، پس از هر گام نقاشی نیمه‌تمام را render می‌کند و پیش از stroke بعدی همان تصویر را به مدل برمی‌گرداند. بخش واقعاً جالب این نیست که این کمک می‌کند — این است که برای کمک کردن اصلاً چه چیزی لازم بود.

یک نقاشی را چطور امتیاز می‌دهید؟

قضاوت سنجه‌ای. چه با و چیزی چه در بپرسیم است عادلانه داده»، «شکست را دیگری مدل مدلش می‌گوید مقاله وقتی score چند روی حوزه بنابراین — ندارد وجود یگانه جواب یک بکش» لپ‌تاپ «یک برای — است سخت واقعاً تولیدشده تصویر انسان‌اند. نگاه برای ناقصی جانشین کدام هر که می‌کند تکیه خودکار

چند مورد در این مقاله ظاهر می‌شوند. **FID** طعم آماری کلی یک batch تصویر تولیدشده را با تصاویر واقعی مقایسه می‌کند؛ کمتر بهتر است، اما batch را توصیف می‌کند نه یک تصویر منفرد. **score CLIP** از یک AI جداگانه می‌پرسد تصویر با متن prompt چقدر جور است — مفید است، اما به همان اندازه‌ای که داور دقیق باشد. **DINO**، **SSIM** و **LPIPS** تصویر بازسازی‌شده را با هدف مقایسه می‌کنند، از پیکسل خام تا هایfeature یادگرفته‌شده.

هیچ کدام حقیقت نیستند. proxy هستند و معمولاً در گام‌های کوچک حرکت می‌کنند. وقتی می‌خوانید یک مدل ۱۲۷٫۶ و دیگری ۱۲۸٫۸ گرفته، تفاوت واقعی و در جهت مطلوب است — اما تکان کوچکی است، نه رانش زمین، و آن هم در عددی که فقط شل و تقریبی با قضاوت چشم انسان همبستگی دارد. این را وقتی تیتر می‌گوید «مدلی را که با بیست برابر داده آموزش دیده شکست داد» نگه دارید.

نویسندگان چه کردند

مسئله‌ای که می‌خواستند حل کنند همان نقاشی کور بود. مدل‌های موجود تولید SVG را یک کار صرفاً متنی می‌دانند: chunk بعدی کد را از روی کد قبلی پیش‌بینی کن و هرگز render نکن. در نتیجه «چشم‌های» قدرتمند — encoder vision — که مدل‌های multimodal امروز از قبل دارند بیکار می‌ماند. Render-in-the-Loop کار را به فرایندی تصویری و مرحله‌به‌مرحله بازساخت می‌کند. پس از هر تکهٔ کد رسم، SVG جزئی به تصویر render می‌شود و به مدل برمی‌گردد تا بخش بعد را در حالی انتخاب کند که بوم فعلی را واقعاً می‌بیند.

اولین یافته هشداردهنده است و به نفع نویسندگان است که واضح گزارشش کرده‌اند: صرفاً چسباندن این loop به یک مدل آمادهٔ موجود جواب نمی‌دهد. وقتی هایrender میانی را بدون آموزش ویژه به مدل‌های عمومی قوی دادند، کیفیت بهتر نشد — در همه‌جا **بدتر** شد. مدلی که هرگز یاد نگرفته از چشمش برای این کار استفاده کند، ناگهان بلد نمی‌شود.

پس بیشتر کار در آموزش است. دادهٔ آموزش را بازسازی می‌کنند تا هر نقاشی به گام‌های کوچک و از نظر بصری معنی‌دار شکسته شود — شکل‌های پیچیده به اجزای ساده‌تر تقسیم می‌شوند تا در هر مرحله چیز تازه‌ای برای دیدن وجود داشته باشد — و سپس یک مدل باز هشت‌میلیاردپارامتری بر پایهٔ Qwen3-VL روی این هایsequence مرحله‌به‌مرحله fine-tune می‌شود. این را Sefl-Feedback Visual می‌نامند. سازوکار دوم هنگام رسم Render-and-Verfiy است: پیش از پذیرفتن هر stroke تازه، مدل آن را render می‌کند و می‌بیند واقعاً تصویر را تغییر داده یا فقط قبلی را تکرار کرده. هایstroke بی‌اثر حذف می‌شوند و وقتی چیز تازه‌ای کمک نمی‌کند مدل به توقف هدایت می‌شود. نکتهٔ مهم این که همهٔ این‌ها با dataset نسبتاً کوچک — حدود ۸۵۰٬۰۰۰ نمونه، کمتر از نصف دادهٔ یک رقیب و بخش کوچکی از دیگری — انجام شده است.

چه پیدا کردند

- با این آموزش، مدل از نسخهٔ کور خودش بهتر رسم می‌کند — بیش از همه در failure :هاcase جایی که مدل کور چشم را روی گونه می‌گذارد یا chart bar خواسته‌شده را حذف و مانیتور generic تولید می‌کند.
- روی benchmark استاندارد، MMSVGBench روش با رقبای قوی رقابتی و در چند metric کمی جلوتر است — از جمله

OmniSVG که با بیش از دو برابر داده و InternSVG که تقریباً با بیست برابر داده آموزش دیده‌اند.

- حاشیه‌ها کوچک‌اند. مثلاً در مجموعهٔ آیکون، score اصلی کیفیت تصویر ۱۲۷٫۶ در برابر ۱۲۸٫۸ بهترین رقیب است و score تطابق prompt برابر ۰٫۲۹۳ در برابر ۰٫۲۹۱ — واقعی و سازگار، اما باریک.
- هر دو جزء افزوده‌شده کار واقعی انجام می‌دهند: آموزش ویژه را خاموش کنید یا verfiication هنگام رسم را بردارید و اعداد اندازه‌گیری‌شده پایین می‌آیند. verfiication به‌ویژه مانع گیرکردن مدل در بازکشیدن بی‌پایان یک چیز می‌شود.
- نتیجه‌ای که خود نویسندگان برجسته می‌کنند **کارایی** است — رسیدن به این سطح با دادهٔ آموزشی بسیار کمتر از رهبران حوزه.

این مقاله چه چیزی را اثبات نمی‌کند

- نشان نمی‌دهد «دیدن» برای مدل یک برد رایگان است. برعکس: آزمایش خود مقاله نشان می‌دهد feedback visual **بدون بازآموزی کیفیت را بدتر می‌کند.** سود از آموزش می‌آید، نه صرف وجود چشم.
- برتری بزرگ یا قاطع تثبیت نمی‌کند. روی بیشتر هاscore روش شانه‌به‌شانهٔ رقباست؛ «مدلی با 20× داده را شکست می‌دهد» درست است، اما با تکان‌های کوچک روی proxy هاmetric در یک benchmark.
- توانایی هنری عمومی نشان نمی‌دهد. این مدل پژوهشی ۸B، آیکون و illustration ساده با resolution ثابت کوچک (224×224) می‌کشد، نه یک طراح همه‌منظوره.
- مقایسه با مدل عمومی بزرگ GPT-5 **سیب با سیب نیست**: آن مدل برای این task محدودِ کدنویسی رسم ساخته یا tune نشده، پس شکست‌دادنش این‌جا دربارهٔ توان عمومی هیچ‌کدام چیز زیادی نمی‌گوید.
- رایگان نیست. render و دوباره‌خواندن canvas در هر گام تولید را از یک pass کور کندتر می‌کند؛ هزینه‌ای که نویسندگان قبول دارند.

قدرت شواهد چقدر است

- **محکم** در مقایسهٔ کنترل‌شده روی شرایط خودش. هاablation تمیزند: آموزش یا verfiication را بردارید و عددها پایین می‌آیند، پس دو جزء واقعاً همان کاری را می‌کنند که نویسندگان می‌گویند.
- **دربارهٔ شگفتی خودش صادق است.** یافتهٔ این‌که feedback visual ساده ضرر می‌زند پنهان نشده و شاید جالب‌ترین بخش مقاله باشد — اصلاح مفیدی برای شهود «input بیشتر همیشه بهتر است».
- **نازک‌تر در ادعای leaderboard.** بردها بر رقبای با دادهٔ بیشتر کوچک‌اند و روی یک benchmark قرار دارند که یکی از همان رقبا در ساختش نقش داشته. حاشیهٔ کم روی proxy هاmetric در یک test، set پیشنهادکننده است نه نهایی.
- **در مقیاس و دنیای واقعی آزموده نشده.** همه‌چیز آیکون و illustration ساده در resolution پایین است. این‌که ایده برای طراحی پیچیده، high-resolution یا کار واقعی نگه می‌دارد، کار آینده است — خود نویسندگان هم می‌گویند.

چرا مهم است

ایدهٔ مرکزی مقاله تقریباً شرم‌آور ساده است و خیلی فراتر از نقاشی می‌رود. اگر برنامه‌ای قرار است با نوشتن کد چیزی تولید کند — صفحهٔ وب، chart، diagram، صحنهٔ 3D — یا می‌تواند کل چیز را کور بنویسد و امیدوار باشد، یا در طول کار render کند و مسیر را اصلاح کند. برای انسان بستن این loop طبیعی است؛ دائم به صفحه نگاه می‌کنیم. برای این مدل‌ها حرکت نسبتاً تازه‌ای است.

چیزی که مقاله را ارزشمند می‌کند ستاره‌ای است که کنار این ایده می‌گذارد. راه دیدن‌دادن به مدل با آموزش نگاه‌کردن یکی نیست. چشم باید آموزش ببیند؛ فقط آن‌وقت loop سود می‌دهد — و حتی آن‌وقت هم سود واقعی اما اندازه‌گیری‌شده است: draftsman باثبات‌تر، نه نوعی هنرمند متفاوت. برای کسانی که ابزارهای تبدیل توضیح به گرافیک قابل‌ویرایش می‌سازند — همان چیزهایی که پشت آیکون و illustration اپ قرار می‌گیرد — درس آرام و عملی ارزشمند است. برد این‌جا از آموزش ارزان‌تر و هوشمندتر آمد، نه صرفاً دادهٔ بیشتر. این یادگیری بهتر از یک امتیاز دیگر روی leaderboard است.

خلاصهٔ تمیز

مدل‌های زبانی تولیدکنندهٔ graphics vector — کد قابل‌ویرایش و پذیرscale پشت بیشتر آیکون‌ها و لوگوهای وب — سنتاً «کور» کار می‌کردند و همهٔ فرمان‌های رسم را بدون کردنrender نتیجه می‌نوشتند. گروهی به سرپرستی Liang Guotao روش Render-in-the-Loop را پیشنهاد می‌کند: پس از هر گام، نقاشی نیمه‌تمام render و دوباره به مدل داده می‌شود تا stroke بعدی را در حالی بکشد که canvas را می‌بیند. یافتهٔ مرکزی و صادقانه این است که اگر این کار را صرفاً روی مدل موجود روشن کنید، مدل **بدتر** می‌شود؛ سود فقط وقتی می‌آید که مدل برای استفاده از feedback تصویری بازآموزی شود، همراه با چک هنگام رسم که هایstroke بی‌اثر را دور می‌اندازد. مدل بازآموزی‌شدهٔ ۸ میلیاردپارامتری روی benchmark استاندارد با رقبایی که تا بیست برابر داده دیده‌اند برابری یا کمی برتری دارد — نتیجه‌ای واقعی، اما با حاشیه‌های کوچک روی proxy ،هاscore برای آیکون و illustration ساده در resolution پایین. نکته‌ای که نویسندگان بر آن تأکید دارند کارایی است: خوب دیدن کار خود می‌تواند بخشی از نیاز به scale خام داده را جبران کند.

بررسی بی‌پرده

مقاله چه نشان می‌دهد: بازآموزی مدل vector-graphics برای render و نگاه‌کردن به نقاشی نیمه‌تمام خودش در هر گام، نتیجهٔ بهتر و کامل‌تری از رسم کور ایجاد می‌کند — روی یک benchmark استاندارد با دادهٔ آموزشی کمتر با رقبا رقابتی و اندکی جلوتر.

چه چیزی محتمل است اما اثبات نشده: «دیدن کار خود» به‌طور کلی نسخهٔ بهتری از صرفاً کردنscale داده است؛ همان ایده روی graphics پیچیده، high-resolution یا واقعی کمک می‌کند؛ یا حاشیه‌های کوچک benchmark تفاوتی است که انسان واقعاً می‌بیند.

چه چیزی را نشان نمی‌دهد: feedback visual به‌تنهایی کمک می‌کند (بدون بازآموزی ضرر می‌زند)؛ برتری بر رقبا بزرگ یا قاطع است؛ توان طراحی همه‌منظوره؛ یا مقایسهٔ منصفانه با مدل عمومی مانند GPT-5 که برای این task ساخته نشده.

محدودیت‌های اصلی: یک benchmark که بخشی از آن توسط نویسندگان رقیب ساخته شده؛ حاشیه‌های کوچک روی proxy ها؛metric مدل 8B و آیکون/illustration ساده در 224×224؛ تولید کندتر به‌خاطر render loop در هر گام.

یک خوانندهٔ عمومی چقدر باید مطمئن باشد؟ اطمینان بالا که بستن loop — کردنrender و دوباره‌دیدن هنگام رسم — واقعاً کمک می‌کند و باید آموزش داده شود، نه فقط روشن شود. اطمینان پایین تا متوسط که این مدل مشخص قاطعانه رقبا را شکست داده؛ «شکست‌دادن 20× داده» را نتیجه‌ای واقعی اما متوسط و تک-benchmark بدانید. اطمینان بالا به ایده‌ای که ارزش به‌خاطر سپردن دارد: برای کدی که رسم می‌کند، نگاه‌کردن در حین کار از نقاشی کور بهتر است.

منابع

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

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

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