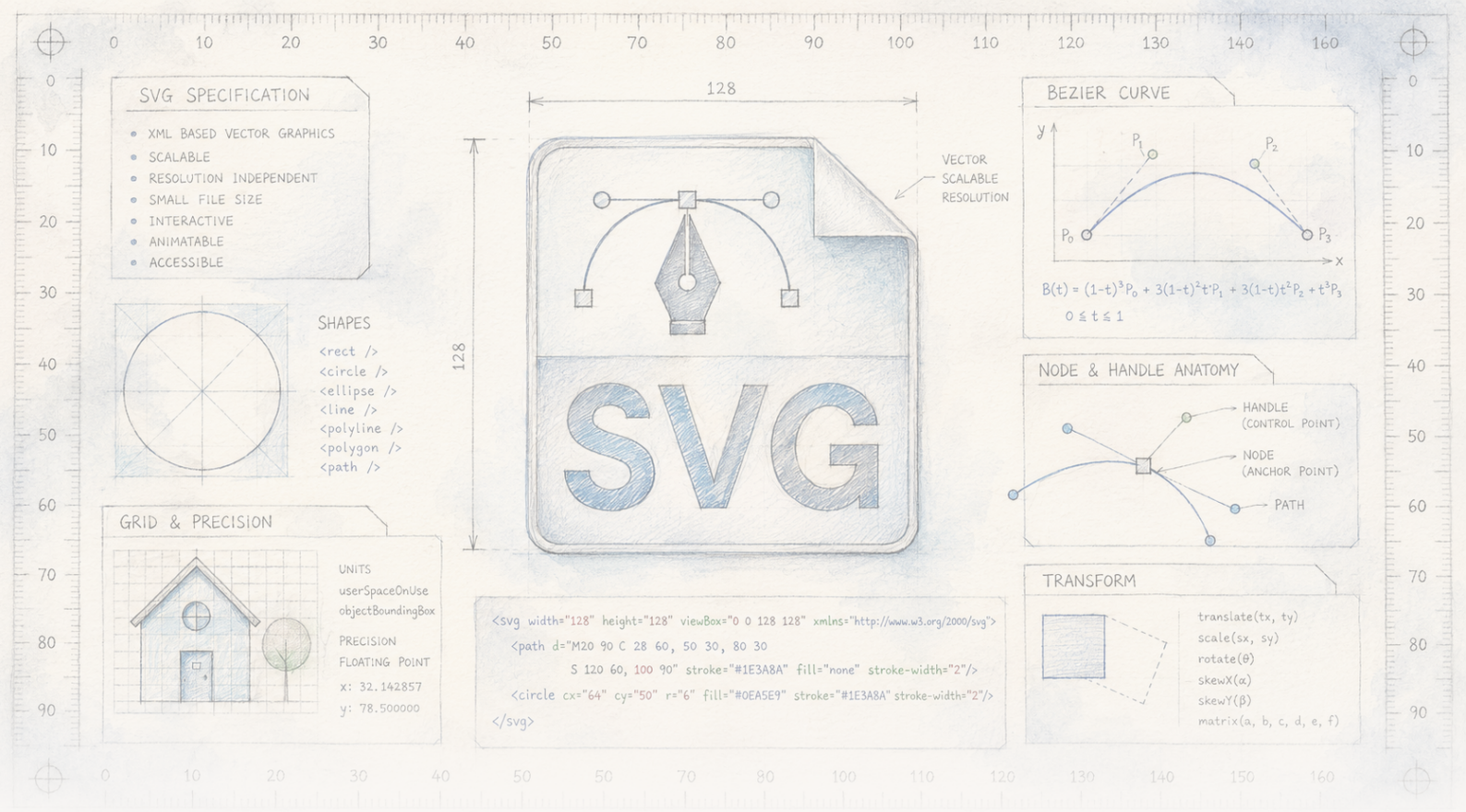


The Clean Paper

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

Да научиш рисуващ AI да гледа листа

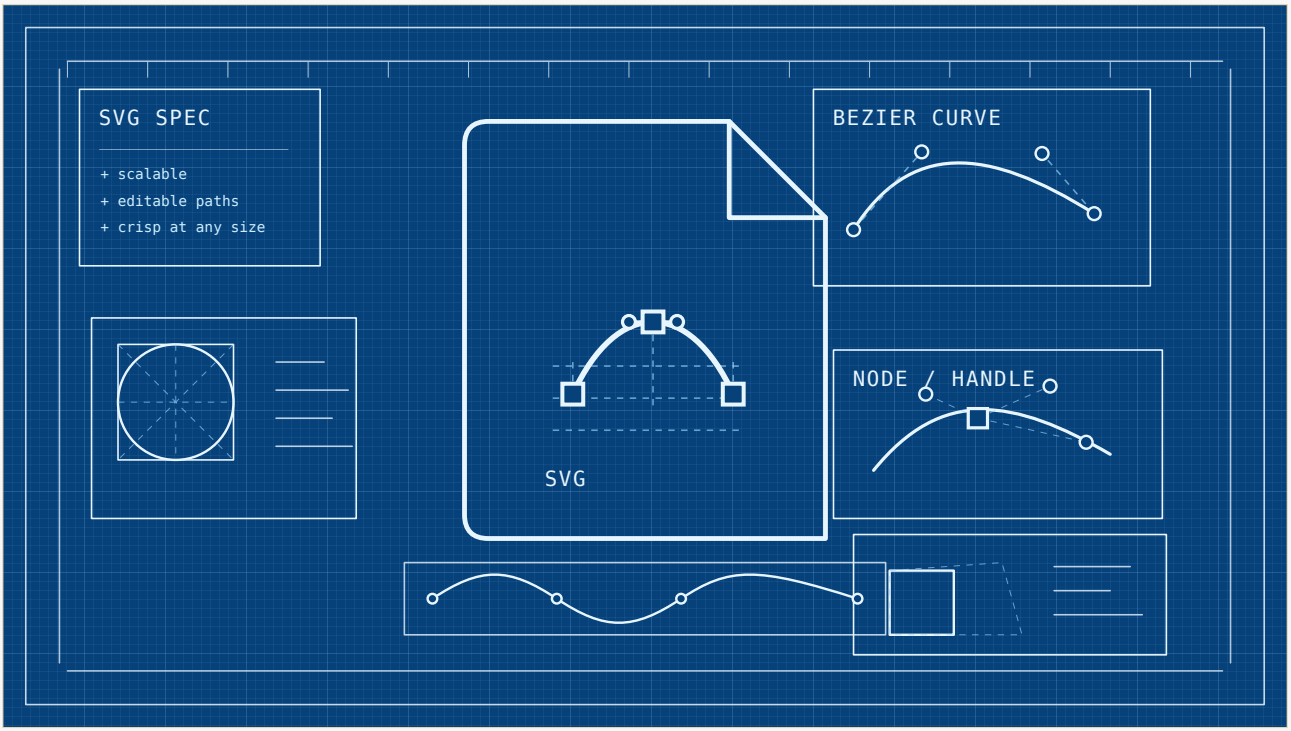
Езиковите модели, които генерират векторна графика, го правят на сляпо — изписват командите за рисуване, без изобщо да виждат резултата. Нов метод позволява на модела да наблюдава как платното се запълва щрих по щрих, а честният обрат е, че самото даване на „очи“ влошава резултатите: моделът трябва да бъде дообучен да ги използва. Наградата е модел, който изравнява или леко превъзхожда съперници, обучени с до двадесет пъти повече данни — реален и внимателен резултат на един benchmark, не революция.

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 Self-Feedback*, Guotao Liang, Zhangcheng Wang, Juncheng Hu, Haitao Zhou, Ziteng Xue, Jing Zhang, Dong Xu, and Qian Yu, Preprint (arXiv:2604.20730)

Рисуване със затворени очи

Помолете днешен AI модел да ви нарисува картина и под капака може да се случи едно от две много различни неща. Познатите генератори на изображения — онези, които създават снимка от изречение — рисуват директно пиксели и могат да виждат платното, докато работят. Но има и втори, по-тих вид рисуване, при който моделът изобщо не рисува. Той пише инструкции: *начертай кръг тук, линия там, запълни тази форма в синьо*. Тези инструкции са код — същият Scalable Vector Graphics, или SVG, който стои зад повечето икони и лога в интернет — и предимството е реално: резултатът не е фиксирана решетка от пиксели, а набор от форми, които могат да се мащабират, преоцветяват и редактират безкрайно, без да се размазват.



Оригинално SVG изображение тип технически чертеж: самото изображение е редактируем векторен файл, изграден от пътища, мрежа, възли и манипулатори, а не от фиксирани пиксели.

Laura Nesso / The Clean Paper Permission on file

Уловката е, че доскоро моделът, който пише такъв код за рисунка, го правеше на сляпо. Той излъчва цялата поредица инструкции наведнъж — кръг, линия, запълване — без да ги рендерира, за да види какво е направил. Представете си да скицирате лице със затворени очи: може приблизително да поставите едното око там, където трябва, но няма как да забележите, че второто е попаднало на бузата или че формата за косата вече е върху носа. Горе-долу така работят тези модели и затова често произвеждат напълно валиден код, който визуално е бъркотия.

Екип, ръководен от Guotao Liang, прави почти комично очевидното: позволява на модела да отвори очи. Методът им, *Render-in-the-Loop*, рендерира недовършената

рисунка след всяка стъпка и връща изображението на модела, преди той да нарисува следващия щрих. Истински интересната част не е, че това помага — а какво е било необходимо, за да започне въобще да помага.

Как се оценява една рисунка?

Когато статия твърди, че модел „побеждава“ друг, справедливо е да попитаме: по какво и измерено как? Оценяването на генерирана картина е трудно — няма един-единствен правилен отговор на „нарисувай лаптоп“ — затова областта разчита на няколко автоматични оценки, всяка несъвършен заместител на човек, който гледа резултата.

Няколко се появяват в тази статия. **FID** сравнява общия статистически характер на цяла група генерирани изображения с реални; по-ниско е по-добре, но оценява групата, не отделна картинка. **CLIP оценката** пита отделен AI дали изображението съответства на думите от инструкцията — полезно, но само толкова добро, колкото е съдията. **DINO**, **SSIM** и **LPIPS** сравняват реконструирано изображение с целево на нива от сурови пиксели до научени признаци.

Нито една от тези метрики не е истина. Те са заместители и обикновено се променят с малки стъпки. Когато четете, че един модел получава 127,6, а друг 128,8, това е реална разлика в желаната посока — но е леко побутване, не свлачище, и то в число, което само приблизително следва това, което би казало окото ви. Полезно е да го помним при заглавие като „побеждава модел, обучен с двадесет пъти повече данни“.

Какво направиха авторите

Проблемът, който авторите искат да поправят, е самото рисуване на сляпо. Съществуващите модели третират писането на SVG като чисто текстова задача: предсказват следващия фрагмент код от кода дотук и никога не го рендерират. Така мощните „очи“ — визуалният енкодер — които съвременните мултимодални модели вече имат, стоят неизползвани. Render-in-the-Loop преструктурира задачата като поетапен визуален процес. След всеки фрагмент код частичното SVG се рендерира като изображение и се подава обратно на модела, така че следващият фрагмент се избира, докато моделът действително гледа платното дотук.

Първият им резултат е предупреждение и, за тяхна чест, го съобщават ясно: просто да добавите този цикъл към готов съществуващ модел не работи. Когато подават междинни рендери на силни общи модели без специално обучение, качеството не се подобрява — то *се влошава* навсякъде. Модел, който никога не е бил учен да използва очите си за тази задача, не започва изведнъж да знае как.

Затова по-голямата част от работата е обучението. Те преструктурират тренировъчните данни така, че всяка рисунка да се раздели на много малки, визуално смислени стъпки

— сложните форми се разбиват на по-прости части, за да има нещо ново за гледане на всеки етап — и след това дообучават отворен модел с осем милиарда параметъра (изграден върху Qwen3-VL) върху тези поетапни последователности. Наричат това Visual Self-Feedback. Добавят и втори механизъм при самото рисуване, Render-and-Verify: преди всеки нов щрих да бъде приет, моделът го рендерира и проверява дали реално е променил картината, или просто е повторил предишното. Щрихове, които не добавят нищо, се отхвърлят, а когато нищо повече не помага, моделът получава указание да спре. Забележително е, че всичко това работи с сравнително малък набор — около 850 000 примера, по-малко от половината на използваното от един конкурент и малка част от данните на друг.

Какво откриха

- Обучен по този начин, моделът рисува по-добре от слепия си вариант — най-видимо в провалите, при които сляп модел поставя око върху бузата или пропуска поисканата стълбовидна диаграма и вместо това рисува общ монитор.
- На стандартния тест MMSVGBench методът е конкурентен и по няколко показатели леко пред силни съперници — включително OmniSVG, обучен с повече от два пъти данните, и InternSVG, обучен с приблизително двадесет пъти повече.
- Разликите са малки. Например при иконите основният показател за качество е 127,6 срещу 128,8 за най-добрия конкурент, а оценката за съответствие с инструкцията е 0,293 срещу 0,291 — реални и последователни, но тънки разлики.
- И двете добавки имат принос: ако се изключи специалното обучение или проверката по време на рисуване, числата измеримо падат. Стъпката за проверка по-специално спира модела да се заклещи в прерисуване на едно и също отново и отново.
- Резултатът, който самите автори подчертават, е ефективността — достигане дотук с много по-малко тренировъчни данни от водещите системи.

Какво не доказва това

- То **не** показва, че да позволиш на модел да „вижда“ е безплатна победа. Всъщност обратното: собственото изследване показва, че визуалната обратна връзка *без* дообучаване влошава резултатите. Печалбата идва от обучението, не само от очите.
- То **не** установява голяма или решаваща преднина. По повечето показатели методът е почти равен с конкурентите; „побеждава модел, обучен с 20× повече данни“ е вярно, но с малки разлики в косвени метрики на един тест за оценяване.
- То **не** демонстрира обща художествена способност. Това е изследователски модел с осем милиарда параметъра, който рисува икони и прости илюстрации при малка фиксирана резолюция (224×224 пиксела), не универсален дизайнер.
- Сравнението с голям общ модел (GPT-5) **не** е apples-to-apples: той не е създаден или настройван за тази тясна задача с кодирано рисуване, така че победата тук казва

малко за двата модела като цяло.
- Това **не** идва без цена. Рендирането и повторното „прочитане“ на платното на всяка стъпка правят генерирането по-бавно от еднократно излъчване на кода на сляпо — цена, която авторите признават.

Колко силни са доказателствата

- **Солидни**, където има контролирано сравнение по собствените правила на работата. Ablation тестовете са чисти: премахнете обучението или проверката и числата падат — значи двете съставки действително вършат работата, която авторите им приписват.
- **Честни към собствената изненада.** Находката, че наивната визуална обратна връзка *вреди*, е съобщена, не скрита, и е най-интересното нещо в статията — полезна корекция на интуицията, че повече вход винаги е по-добре.
- **Слаби като твърдение за класацията.** Победите над конкуренти с повече тренировъчни данни са малки и са върху един тест за оценяване, създаден от авторите на един от тези конкуренти. Малки разлики по косвени метрики на един тестов набор са показателни, не окончателни.
- **Нетествани в мащаб и в реалността.** Всичко тук е икони и прости илюстрации при ниска резолюция. Дали същата идея работи при сложна, високорезолюционна или реална дизайнерска работа остава за бъдещето — авторите го казват директно.

Защо това има значение

Идеята в центъра на статията е почти неудобно проста и стига далеч отвъд рисуването. Ако програма ще генерира нещо чрез писане на код — уеб страница, диаграма, схема, 3D сцена — може или да напише всичко на сляпо и да се надява, или да рендерира по пътя и да коригира курса. Затварянето на този цикъл е естествено за човек; ние непрекъснато поглеждаме листа. За тези модели това е изненадващо нов ход.

Струва си да се чете заради звездичката към идеята. Да дадеш на модел начин да вижда не е същото като да го научиш да гледа. Очите трябва да бъдат обучени и едва тогава цикълът се отплаща — а дори тогава печалбата е реална, но измерена: по-стабилен чертожник, не различен вид художник. За хората, които правят инструменти, превръщащи описание в редактируема графика — точно онези неща зад иконите и илюстрациите в приложения — тихият практически урок е полезният. Печалбата тук идва от по-евтино и по-умно обучение, не от повече данни. Това е по-добър урок от още една точка в класацията.

Кратко и ясно

Езиковите модели, които генерират векторна графика — редактируемия и мащабируем код зад повечето уеб икони и лога — традиционно работят „на сляпо“, изписвайки всички команди за рисуване, без да ги рендерират и виждат резултата. Екип, ръководен от Guotao Liang, предлага Render-in-the-Loop: недовършената рисунка се рендерира след всяка стъпка и се връща на модела, така че следващият щрих се прави с поглед към платното. Централният и честен резултат е, че простото добавяне на това към съществуващ модел го прави *по-лош*; ползата се появява само след дообучаване да използва визуалната обратна връзка, подпомогнато от проверка при рисуване, която отхвърля щрихове без промяна. Дообученият модел с осем милиарда параметъра изравнява или леко превъзхожда съперници, обучени с до двадесет пъти повече данни на стандартен тест за оценяване — истински резултат, но с малки разлики по косвени метрики, за икони и прости илюстрации при ниска резолюция. Изводът, който авторите подчертават и който си струва да се запази, е за ефективността: ако моделът се научи добре да гледа собствената си работа, това може да компенсира част от чистия мащаб на данните.

Проверка без излишен шум

Какво показва статията: Дообучаването на модел за векторна графика да рендерира и гледа собствената си недовършена рисунка стъпка по стъпка дава по-добри и по-пълни резултати от рисуването на сляпо — конкурентни и леко по-добри от съперници с повече данни на един стандартен тест за оценяване, при по-малко тренировъчни данни.

Какво е правдоподобно, но не е доказано: Че „гледай собствената си работа“ е широко по-добра рецепта от увеличаването на данните; че същата идея ще помогне при сложни, високорезолюционни или реални графики; че малките разлики в теста за оценяване отразяват разлика, която човек действително би забелязал.

Какво не показва: Че визуалната обратна връзка помага сама по себе си (без дообучаване вреди); че преднината над конкурентите е голяма или решаваща; обща способност за рисуване; честно директно сравнение с общи модели като GPT-5, които не са създадени за тази задача.

Основни ограничения: Един тест за оценяване, създаден отчасти от автори на конкурент; малки разлики по косвени метрики; 8B модел, икони и прости илюстрации при 224×224; по-бавно генериране заради цикъла с рендериране на всяка стъпка.

Колко уверен трябва да бъде общият читател? Висока увереност, че затварянето на цикъла — рендериране и повторно гледане по време на рисуването — действително помага и че трябва да бъде обучено, не просто включено. Ниска до умерена увереност, че конкретният модел решаващо превъзхожда конкурентите си; „побеждава с 20×

по-малко данни“ е реална, но умерена находка от един тест за оценяване. Висока увереност в една идея, която си струва да се помни: при код, който рисува, да гледаш докато работиш е по-добре от да рисуваш на сляпо.

Източници

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/