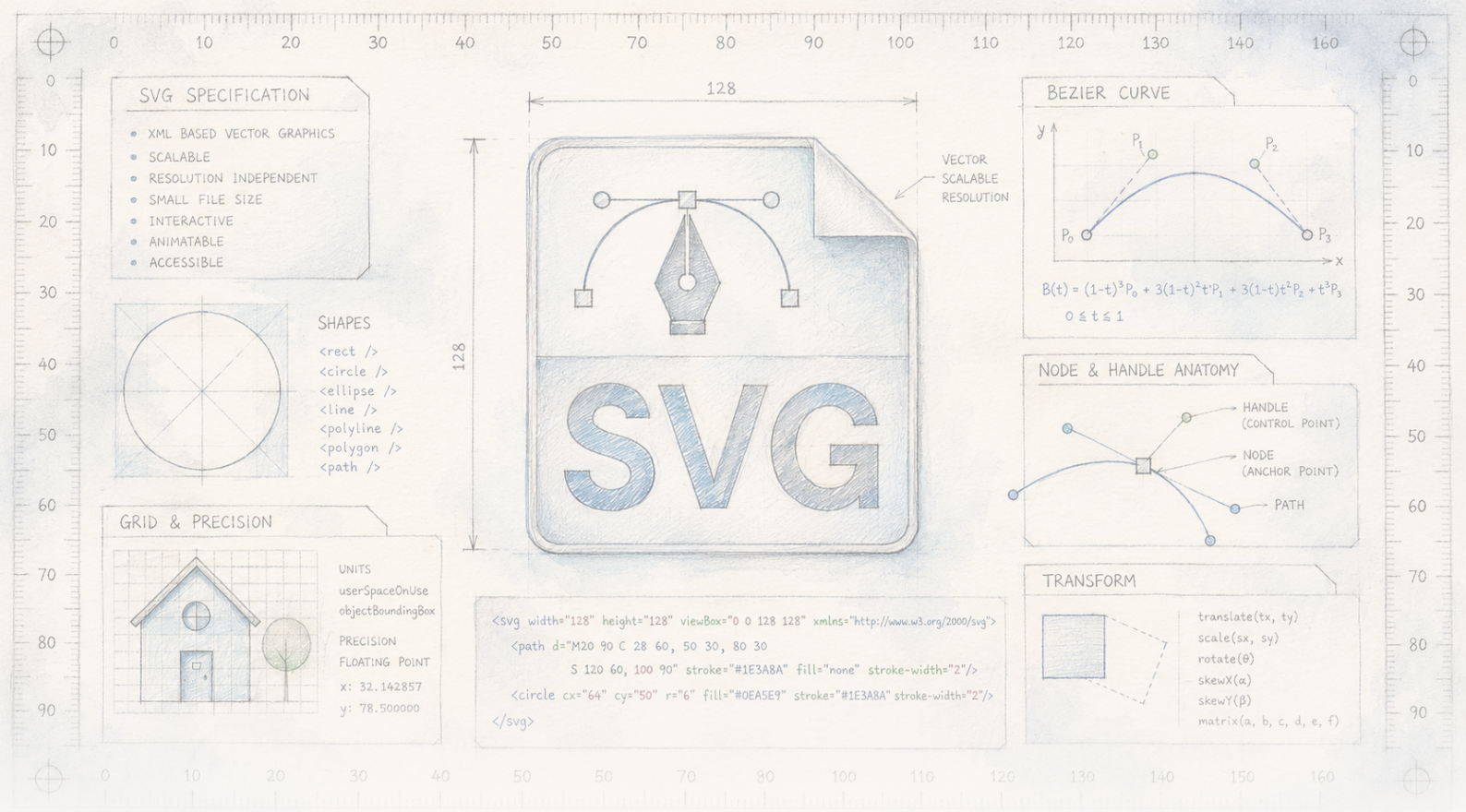


The Clean Paper

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

Om 'n teken-KI te leer om na die bladsy te kyk

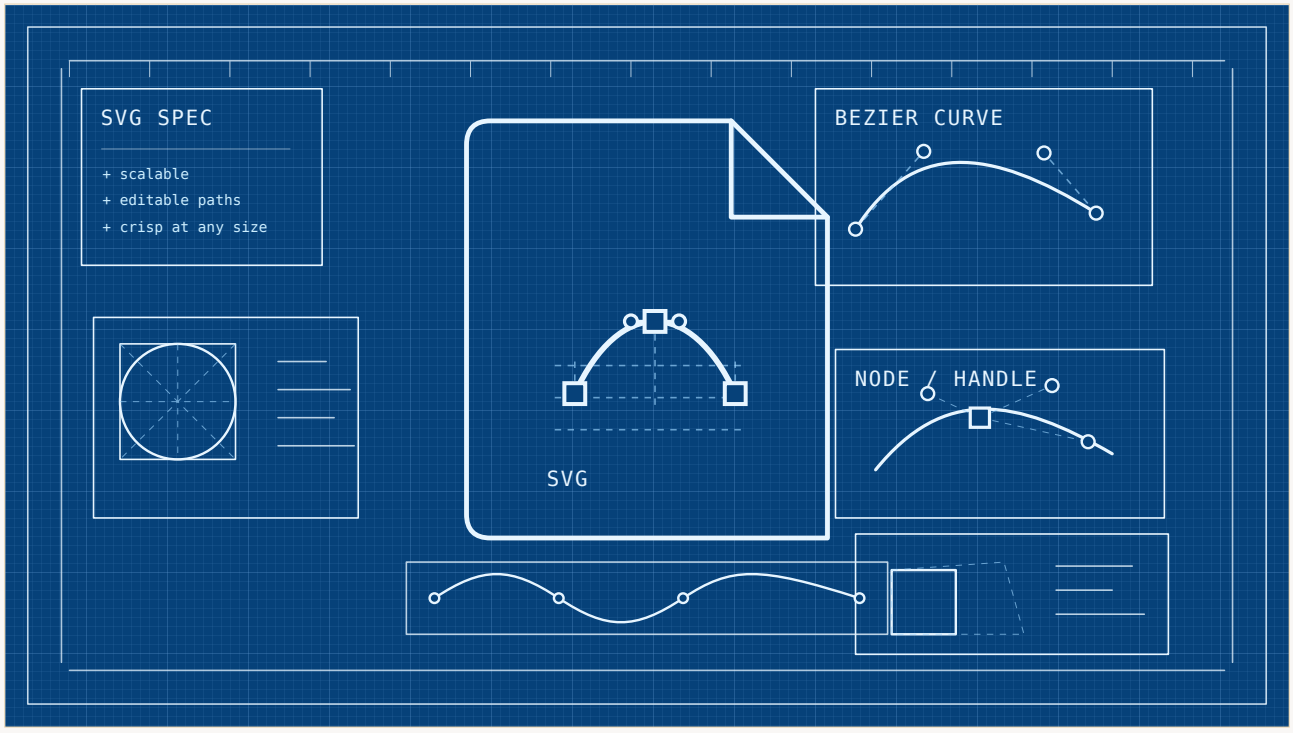
Taalmodelle wat vektorgrafika genereer, doen dit blind — hulle skryf die tekenopdragte uit sonder om die resultaat ooit te sien. 'n Nuwe metode laat die model toekyk hoe sy eie doek invul, haal vir haal, en die eerlike kinkel is dat om dit bloot oë te gee sake vererger: dit moet herafgerig word om hulle te gebruik. Die beloning is 'n model wat mededingers wat op tot twintig keer meer data afgerig is, ewenaar of net-net verbysteek — 'n egte, versigtige resultaat op een maatstaf, nie 'n revolusie nie.

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)

Teken met jou oë toe

Vra een van vandag se KI-modelle om vir jou ’n prentjie te teken, en onder die enjinkap gebeur een van twee baie verskillende dinge. Die bekende beeldgenereerders — dié wat ’n foto uit ’n sin tower — verf pixels direk, en hulle kan die doek sien terwyl hulle werk. Maar daar is ’n tweede, stiller soort teken, waar die model glad nie verf nie. Dit skryf instruksies: *teken hier ’n sirkel, daar ’n lyn, vul hierdie vorm blou in*. Daardie instruksies is kode — dieselfde Scalable Vector Graphics, oftewel SVG, wat agter die meeste ikone en logo’s op die web sit — en die aantrekkingskrag is eg: die resultaat is nie ’n vaste rooster van pixels nie, maar ’n stel vorms wat jy vir altyd kan herskaal, herkleur en redigeer sonder dat dit dof word.



Oorspronklike SVG-bloudrukkunswerk: die beeld is self ‘n redigeerbare vektorlêer, opgebou uit paaie, roosterlyne, nodusse en handvatsels eerder as vaste pixels.

Laura Nesso / The Clean Paper Permission on file

Die vangplek is dat ’n model wat hierdie tekenkode skryf, dit tot onlangs blind gedoen het. Dit het die hele reeks instruksies in een deurloop uitgeskryf — sirkel, lyn, vul — sonder om hulle ooit te render om te sien wat dit gemaak het. Stel jou voor jy skets ’n gesig met jou oë toe: jy kry die oë dalk min of meer waar oë hoort, maar jy sou geen manier hê om agter te kom dat die tweede een op die wang beland het nie, of dat die vorm wat jy vir die hare geteken het nou oor die neus sit nie. Dit is min of meer hoe hierdie modelle gewerk het, en daarom het hulle so dikwels kode gelewer wat volkome geldig en tog visueel ’n gemors was.

’n Span onder leiding van Guotao Liang het nou die byna komies voor die hand liggende ding gedoen: hulle laat die model sy oë oopmaak. Hul metode, *Render-in-the-Loop*, render die halfklaar tekening ná elke stap en gee daardie prent aan die model terug voordat dit die volgende haal trek. Die werklik

interessante deel is nie dat dit help nie — dit is wat hulle moes doen om dit hoegenaamd te laat help.

> **Hoe gee jy 'n tekening punte?**
>
> Wanneer 'n artikel sê sy model "klop" 'n ander een, is dit billik om te vra: klop dit waarin, en hoe gemeet? Om 'n gegenereerde prent te beoordeel is werklik moeilik — daar is geen enkele regte antwoord op "teken 'n skootrekenaar" nie — dus leun die veld op 'n handjievol outomatiese tellings, elkeen 'n onvolmaakte plaasvervanger vir 'n mens wat na die resultaat kyk.
> 'n Paar verskyn in hierdie artikel. **FID** vergelyk die algehele statistiese geur van 'n hele bondel gegenereerde beelde met egte beelde; laer is beter, maar dit beskryf die bondel, nie enige enkele prent nie. **CLIP score** vra 'n aparte KI of 'n beeld by die woorde van die opdrag pas — nuttig, maar net so skerp soos daardie beoordelaar. **DINO**, **SSIM** en **LPIPS** vergelyk 'n gerekonstrueerde beeld met 'n teiken, op vlakke wat wissel van rou pixels tot aangeleerde kenmerke.
> Nie een hiervan is die waarheid nie. Hulle is benaderings, en hulle beweeg gewoonlik in klein trappies. Wanneer jy lees dat een model 127.6 behaal waar 'n ander 128.8 behaal, is dit 'n werklike verskil in die bedoelde rigting — maar dit is 'n stootjie, nie 'n aardverskuiwing nie, en boonop in 'n getal wat net losweg volg wat jou oog sou sê. Die moeite werd om te onthou wanneer die opskrif lui: "klop 'n model wat op twintig keer soveel data afgerig is."

Wat die outeurs gedoen het

Die probleem wat die outeurs wou regstel, is die blinde teken self. Bestaande modelle hanteer die skryf van SVG as 'n suiwer tekstaak: voorspel die volgende stuk kode uit die kode tot dusver, en render dit nooit nie. Dit laat die kragtige "oë" — die visie-enkodeerder — wat moderne multimodale modelle reeds saamdra, ongebruik lê. Render-in-the-Loop herstruktureer die taak as 'n stap-vir-stap visuele proses. Ná elke fragment tekenkode word die gedeeltelike SVG na 'n beeld gerender en as 'n prent aan die model teruggevoer, sodat die volgende fragment gekies word terwyl die model werklik na die doek tot dusver kyk.

Hul eerste bevinding is 'n waarskuwing, en dit strek hulle tot eer dat hulle dit reguit rapporteer: om hierdie lus bloot aan 'n bestaande model van die rak vas te bout, werk nie. Toe hulle tussentydse renders aan sterk algemene modelle gevoer het sonder enige spesiale afrigting, het die kwaliteit nie verbeter nie — dit het oor die hele linie *versleg*. 'n Model wat nooit geleer is om sy oë hiervoor te gebruik nie, weet nie skielik hoe nie.

Die meeste van die werk lê dus in die leerproses. Hulle herbou die afrigtingsdata sodat elke tekening in baie klein, visueel betekenisvolle stappe opgebreek word — komplekse vorms word in eenvoudiger stukke verdeel sodat daar by elke stadium iets nuuts te sien is — en verfyn dan 'n oop model met agt miljard parameters (gebou op Qwen3-VL) op hierdie stap-vir-stap reekse. Hulle noem dit Visual Self-Feedback. Hulle voeg 'n tweede meganisme by tydens die teken, Render-and-Verify: voordat elke nuwe haal aanvaar word, render die model dit en kontroleer of dit die prent werklik verander het, of bloot die vorige een herhaal het. Hale wat niks byvoeg nie, word weggegooi, en wanneer niks meer

help nie, word die model aangesê om op te hou. Opvallend genoeg loop dit alles op ’n taamlik klein datastel — sowat 850 000 voorbeelde, minder as die helfte van wat een mededinger gebruik het en ’n klein breukdeel van ’n ander s’n.

Wat hulle gevind het

- So afgerig, teken die model beter as sy blinde eweknie — die sigbaarste by die mislukkingsgevalle, waar ’n blinde model ’n oog op ’n wang plaas, of ’n gevraagde staafgrafiek oorslaan en eerder ’n generiese monitor uitlewer.
- Op die standaardmaatstaf (MMSVGBench) kom die metode mededingend uit teen sterk mededingers — en op verskeie maatstawwe effens voor hulle — insluitend OmniSVG, afgerig op meer as twee keer soveel data, en InternSVG, afgerig op ongeveer twintig keer soveel.
- Die marges is klein. Op die ikoonstel, byvoorbeeld, staan sy hoof-beeldkwaliteittelling op 127.6 teenoor die beste mededinger se 128.8, en sy opdrag-passingstelling op 0.293 teenoor 0.291 — eg en konsekwent, maar skraal.
- Albei bygevoegde onderdele verdien hul plek: skakel die spesiale afrigting af, of die verifikasie tydens teken, en die syfers daal meetbaar. Veral die verifikasiestap keer dat die model vashaak en dieselfde ding oor en oor teken.
- Die resultaat wat die outeurs self beklemtoon, is doeltreffendheid — om so ver te kom met veel minder afrigtingsdata as wat die voorlopers gebruik het.

Wat dit nie bewys nie

- Dit wys **nie** dat om ’n model te laat “sien” ’n gratis wins is nie. Inteendeel: die artikel se eie eksperiment wys dat visuele terugvoer *sonder* herafrigting sake vererger. Die wins kom van die herafrigting, nie van die oë alleen nie.
- Dit vestig **nie** ’n groot of deurslaggewende voorsprong nie. Op die meeste tellings is die metode kop aan kop met sy mededingers; “klop ’n model wat op 20× die data afgerig is” is waar, maar met stootjies op benaderde maatstawwe, op een maatstaf.
- Dit demonstreer **nie** algemene artistieke vermoë nie. Dit is ’n navorsingsmodel met agt miljard parameters wat ikone en eenvoudige illustrasies teken teen ’n klein vaste resolusie (224×224 pixels), nie ’n algemene-doel-ontwerper nie.
- Die vergelyking met ’n groot algemene model (GPT-5) is **nie** appels met appels nie: daardie model is nie vir hierdie eng kodetekentaak gebou of ingestel nie, dus sê om dit hier te klop min oor enige van die twee modelle in die algemeen.
- Dit kom **nie** verniet nie. Om die doek by elke stap te render en weer te lees maak generering stadiger as om die kode in een blinde deurloop uit te skryf — ’n koste wat die outeurs erken.

Hoe sterk is die bewyse

- **Solied** waar dit ’n gekontroleerde vergelyking op sy eie terme is. Die ablasies is skoon: verwyder die afrigting, of verwyder die verifikasie, en die syfers val — die twee bestanddele doen dus werklik die werk wat die outeurs beweer.
- **Eerlik oor sy eie verrassing.** Die bevinding dat naïewe visuele terugvoer *skade doen*, word gerapporteer, nie begrawe nie, en dit is die interessantste ding in die artikel — ’n nuttige regstelling van die intuïsie dat meer insette altyd beter is.
- **Dun waar dit ’n ranglys-aanspraak is.** Die oorwinnings oor mededingers met meer data is klein en leef op ’n enkele maatstaf wat deur die outeurs van een van daardie mededingers gebou is. Klein marges op benaderde maatstawwe op een toetsstel is suggestief, nie beklink nie.
- **Onbeproef op skaal en in die praktyk.** Alles hier is ikone en eenvoudige illustrasies teen lae resolusie. Of dieselfde idee geld vir komplekse, hoë-resolusie- of werklike ontwerpwerk, word as toekomstige werk gelaat — die outeurs sê self so.

Waarom dit saak maak

Die idee in die kern van hierdie artikel is byna verleentheidwekkend eenvoudig, en dit strek ver verby teken. As ’n program iets gaan genereer deur kode te skryf — ’n webblad, ’n grafiek, ’n diagram, ’n 3D-toneel — kan dit óf die hele ding blind skryf en hoop, óf render soos dit vorder en koers regstel. Om daardie lus te sluit is vir ’n mens tweede natuur; ons loer gedurig na die bladsy. Vir hierdie modelle is dit ’n verrassend onlangse skuif.

Wat die artikel die lees werd maak, is die sterretjie wat dit aanheg. Om ’n model ’n manier te gee om te sien is nie dieselfde as om dit te leer kyk nie. Die oë moet afgerig word, en eers dan betaal die lus af — en selfs dan is die opbrengs eg maar gematig: ’n stewiger tekenaar, nie ’n ander soort kunstenaar nie. Vir enigiemand wat gereedskap bou wat ’n beskrywing in ’n redigeerbare grafika omskep — die soort ding wat uiteindelik agter ’n toep se ikone en illustrasies sit — is die stil, praktiese les die nuttige een. Die wins hier het gekom van goedkoper, slimmer afrigting, nie van meer data nie. Dit is ’n beter ding om te leer as nog ’n punt op ’n ranglys.

Skoon opsomming

Taalmodelle wat vektorgrafika genereer — die redigeerbare, herskaalbare kode agter die meeste web-ikone en logo’s — het dit tradisioneel “blind” gedoen: hulle skryf al die tekenopdragte uit sonder om hulle ooit te render om die resultaat te sien. ’n Span onder leiding van Guotao Liang stel Render-in-the-Loop voor: render die halfklaar tekening ná elke stap en voer dit terug, sodat die model die volgende haal trek terwyl dit na die doek kyk. Hul sentrale, eerlike bevinding is dat om dit bloot op ’n bestaande model toe te pas dit *slegter* maak; die wins verskyn eers nadat die model herafgerig is om die visuele terugvoer te gebruik, gehelp deur ’n kontrole tydens teken wat hale weggooi wat niks verander nie. Die herafgerigte model met agt miljard parameters ewenaar of klop effens mededingers wat op tot

twintig keer meer data afgerig is, op ’n standaardmaatstaf — ’n egte resultaat, maar met klein marges op benaderde tellings, vir ikone en eenvoudige illustrasies teen lae resolusie. Die les wat die outeurs beklemtoon, en die een wat die hou werd is, gaan oor doeltreffendheid: om jou werk goed te sien kan blote dataskaal vervang.

No-BS-toets

Wat die artikel wys: Om ’n vektorgrafikamodel her af te rig om sy eie halfklaar tekening stap vir stap te render en te bekyk, lewer beter en vollediger resultate as blinde teken — mededingend met, en effens voor, mededingers met meer data op een standaardmaatstaf, uit minder afrigtingsdata.

Wat aanneemlik maar nie bewys is nie: Dat “jou werk sien” in die breë ’n beter resep is as om data op te skaal; dat dieselfde idee sal help met komplekse, hoë-resolusie- of werklike grafika; dat die klein maatstafmarges ’n verskil weerspieël wat ’n mens werklik sou opmerk.

Wat dit nie wys nie: Dat visuele terugvoer op sy eie help (sonder herafrigting doen dit skade); dat die voorsprong oor mededingers groot of deurslaggewend is; algemene tekenvermoë; ’n billike direkte kragmeting met algemene modelle soos GPT-5, wat nie vir hierdie taak gebou is nie.

Belangrikste beperkings: Een maatstaf, deels gebou deur ’n mededinger se outeurs; klein marges op benaderde maatstawwe; ’n 8B-model, ikone en eenvoudige illustrasies teen 224×224; stadiger generering weens die render-by-elke-stap-lus.

Hoeveel vertroue behoort ’n algemene leser te hê? Hoog dat om die lus te sluit — render en herlees terwyl jy teken — werklik help, en dat dit ingeoefen moet word, nie net aangeskakel nie. Laag tot matig dat hierdie spesifieke model sy mededingers deurslaggewend klop; hanteer die reël “klop 20× die data” as ’n egte maar beskeie resultaat op ’n enkele maatstaf. Hoog oor die een idee wat die onthou werd is: vir kode wat teken, klop kyk terwyl jy teken blinde teken.

Sources

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/