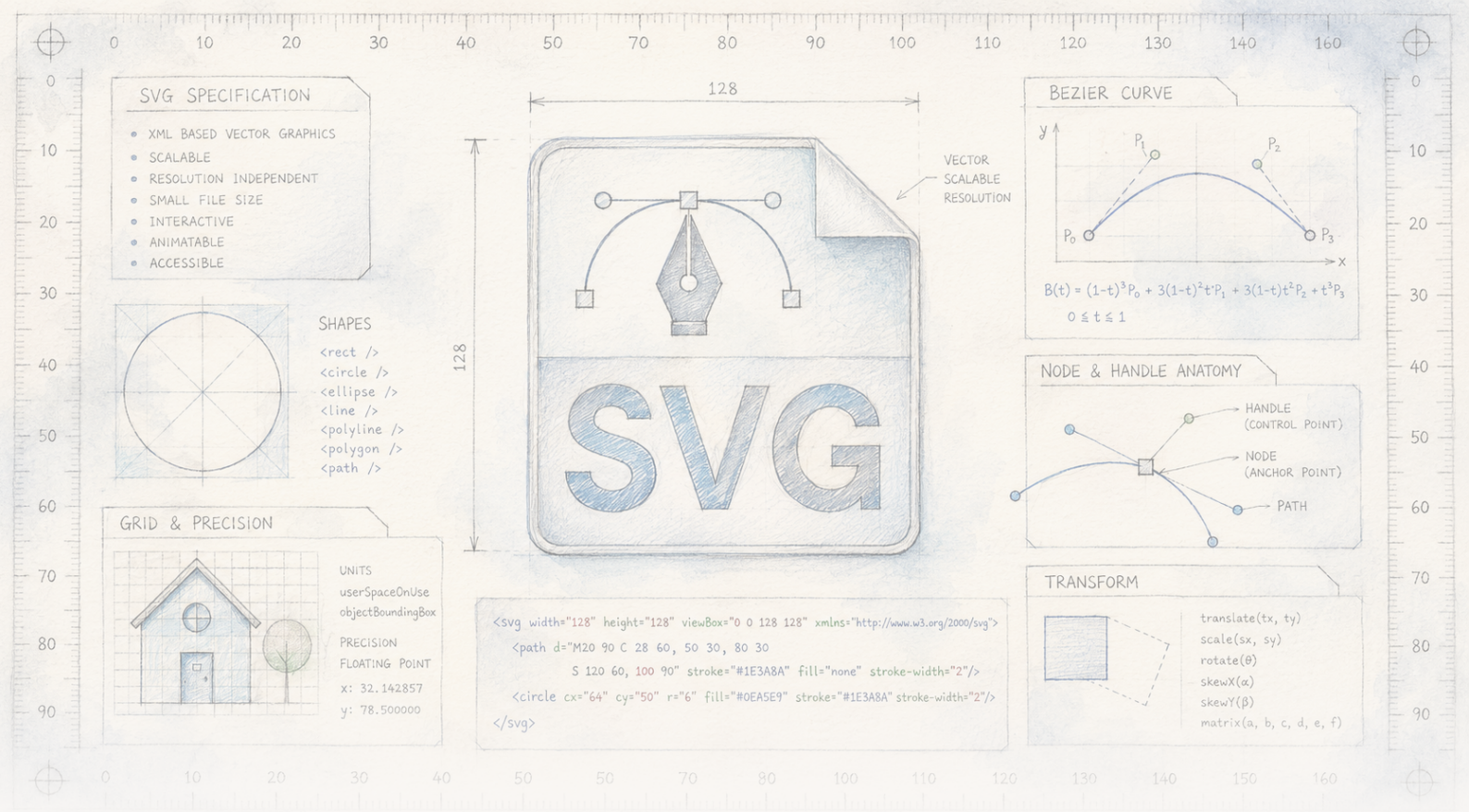


The Clean Paper

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

როგორ ასწავლეს ხატვის AI-ს საკუთარ ტილოზე ყურება

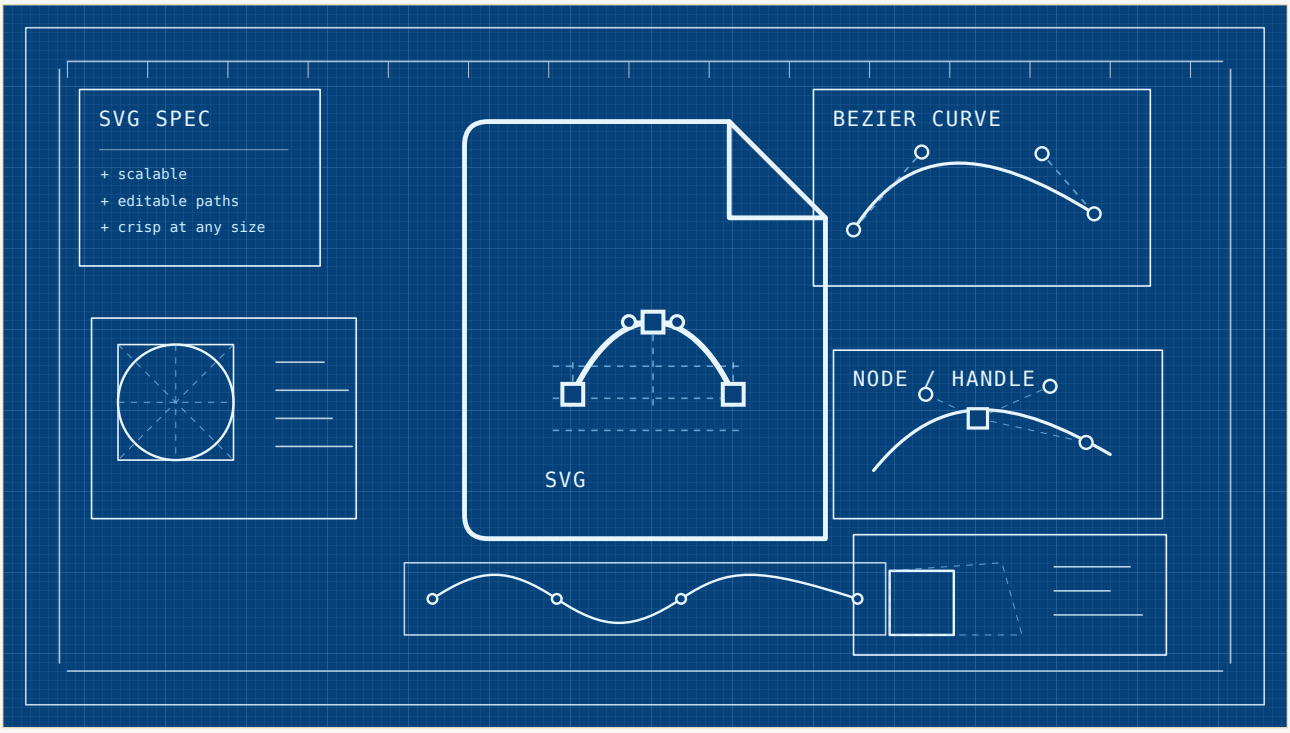
ვექტორული გრაფიკის გენერირებული ენობრივი მოდელები აქამდე ბრმად წერდნენ ხატვის კოდს და შედეგს საერთოდ არ ხედავდნენ. ახალი მეთოდი მოდელს საშუალებას აძლევს თითოეული შტრიხის შემდეგ საკუთარ ტილოს შეხედოს. საინტერესო შემობრუნება ისაა, რომ მხოლოდ „თვალების" მიცემა შედეგს აუარესებს — მოდელს მათი გამოყენებაც უნდა ასწავლო. გადამზადების შემდეგ ის ერთ ბენჩმარკზე უტოლდება ან ოდნავ უსწრებს კონკურენტებს, რომლებიც ოცჯერ მეტ მონაცემზე იყვნენ გაწვრთნილი: რეალური, ფრთხილი შედეგია და არა რევოლუცია.

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)

ხატვა დახუჭული თვალებით

თუ დღევანდელ AI მოდელს სთხოვთ რაიმე დაგიხატოთ, შიგნით ორი სრულიად განსხვავებული პროცესი შეიძლება დაიწყოს. ნაცნობი გამოსახულების გენერატორები — ისინი, რომლებიც ერთი წინადადებიდან ფოტოს ქმნიან — პიქსელებს პირდაპირ „ხატავენ“ და მუშაობისას ტილოს ხედავენ. მაგრამ არსებობს მეორე, უფრო მშვიდი ტიპის ხატვა, სადაც მოდელი საერთოდ არ ხატავს. ის ინსტრუქციებს წერს: *აქ წრე დახატე, იქ ხაზი გაავლე, ეს ფორმა ლურჯად შეავსე*. ეს ინსტრუქციები კოდია — იგივე Scalable Vector Graphics, ანუ SVG, რომელიც ვებგვერდების უმეტეს ხატულებსა და ლოგოებს უდევს საფუძვლად. უპირატესობა აშკარაა: შედეგი პიქსელების ფიქსირებული ბადე კი არა, ფორმების ნაკრებია, რომლის მასშტაბირება, ფერის შეცვლა და რედაქტირება უსასრულოდ შეიძლება, დაბინდვის გარეშე.



ორიგინალური SVG-„ბლუპრინტის“ ილუსტრაცია: თავად გამოსახულება რედაქტირებადი ვექტორული ფაილია, რომელიც ფიქსირებული პიქსელების ნაცვლად ბილიკებისგან, ბადის ხაზებისგან, კვანძებისა და მართვის სახელურებისგან შედგება.

Laura Nesso / The Clean Paper Permission on file

პრობლემა ის იყო, რომ ბოლო დრომდე ასეთი ხატვის კოდის დამწერი მოდელი პრაქტიკულად ბრმად მუშაობდა. ის ინსტრუქციების მთელ მიმდევრობას ერთ გასვლაში ქმნიდა — წრე, ხაზი, შევსება — და არასოდეს არენდერებდა შუალედურ შედეგს, რომ დაენახა რას აკეთებდა. წარმოიდგინეთ, რომ სახეს დახუჭული თვალებით ხატავთ: შეიძლება თვალები დაახლოებით სწორ ადგილას მოხვდეს, მაგრამ ვერ შეამჩნევთ, რომ მეორე თვალი ლოყაზე აღმოჩნდა ან თმისთვის დახატული ფორმა ცხვირს ფარავს. დაახლოებით ასე მუშაობდა ეს მოდელები, ამიტომ ხშირად ქმნიდნენ სრულიად ვალიდურ კოდს, რომელიც ვიზუალურად არეული იყო.

Guotao Liang-ის ხელმძღვანელობით გუნდმა თითქმის კომიკურად აშკარა ნაბიჯი გადადგა: მოდელს „თვალები გაახელინა“. მათი მეთოდი, *Render-in-the-მარყუჟი*, ყოველი ნაბიჯის შემდეგ ნახევრად დასრულებულ ნახატს არენდერებს და მიღებულ გამოსახულებას მოდელს უკან აწვდის, ვიდრე ის შემდეგ შტრიხს შექმნის. საინტერესო არა იმდენად ისაა, რომ ეს შეიძლება სასარგებლო იყოს, არამედ ის, თუ რა გახდა საჭირო იმისთვის, რომ საერთოდ ემუშავა.

როგორ აფასებენ ნახატს?

როცა ნაშრომი ამბობს, რომ მისი მოდელი სხვას „ჯობნის“, სამართლიანი კითხვაა: რაში და როგორ არის ეს გაზომილი? გენერირებული გამოსახულების შეფასება რთულია — კითხვას „დახატე ლეპტოპი“ ერთი სწორი პასუხი არ აქვს — ამიტომ სფერო რამდენიმე ავტომატურ მეტრიკას ეყრდნობა. თითოეული მათგანი მხოლოდ არასრულყოფილი შემცვლელია ადამიანის ვიზუალური შეფასებისა.

ამ ნაშრომშიც რამდენიმე ასეთი მეტრიკაა. **FID** გენერირებული გამოსახულებების მთელი ჯგუფის სტატისტიკურ თვისებებს რეალურ გამოსახულებებს ადარებს; დაბალი მნიშვნელობა უკეთესია, თუმცა მეტრიკა ჯგუფს აღწერს და არა ცალკეულ სურათს. **CLIP score** სხვა AI-ს ეკითხება, რამდენად შეესაბამება გამოსახულება ტექსტურ მოთხოვნას — სასარგებლოა, მაგრამ მხოლოდ იმდენად კარგი, რამდენადაც თავად შემფასებელი. **DINO**, **SSIM** და **LPIPS** აღდგენილ გამოსახულებას მიზნობრივ გამოსახულებას ადარებს, ნედლი პიქსელებიდან ნასწავლ ვიზუალურ ნიშნებამდე სხვადასხვა დონეზე.

არცერთი მათგანი „სიმართლე“ არ არის. ეს proxy-მეტრიკებია და ცვლილებები ხშირად მცირეა. როცა კითხულობთ, რომ ერთმა მოდელმა მიიღო 127.6, მეორემ კი 128.8, ეს სასურველი მიმართულებით რეალური სხვაობაა, მაგრამ მცირე ბიძგია და არა დიდი ნახტომი — თან ისეთ რიცხვში, რომელიც ადამიანის თვალის შეფასებას მხოლოდ ნაწილობრივ შეესაბამება. ეს განსაკუთრებით მნიშვნელოვანია სათაურის წაკითხვისას: „აჯობა მოდელს, რომელიც ოცჯერ მეტ მონაცემზე იყო გაწვრთნილი“.

რა გააკეთეს ავტორებმა

ავტორების ამოცანა სწორედ „ბრმა ხატვის“ გამოსწორება იყო. არსებული მოდელები SVG-ის წერას წმინდა ტექსტურ ამოცანად განიხილავენ: წინა კოდის მიხედვით იწინასწარმეტყველე შემდეგი ფრაგმენტი და არაფერი დაარენდერო. შედეგად თანამედროვე მულტიმოდალურ მოდელებში უკვე არსებული ძლიერი „თვალები“ — vision encoder — გამოუყენებელი რჩება. Render-in-the-მარყუჟი ამოცანას ეტაპობრივ ვიზუალურ პროცესად გარდაქმნის. ყოველი კოდის ფრაგმენტის შემდეგ ნაწილობრივი SVG გამოსახულებად რენდერდება და მოდელს სურათის სახით მიეწოდება; შემდეგ ფრაგმენტს ის უკვე არსებული ტილოს „დანახვის“ შემდეგ ირჩევს.

პირველი შედეგი გამაფრთხილებელია და ავტორები ამას პირდაპირ ამბობენ: ამ ციკლის უბრალოდ დამატება მზა მოდელზე არ მუშაობს. როდესაც შუალედურ რენდერებს ძლიერ ზოგად მოდელებს სპეციალური გაწვრთნის გარეშე აწვდიდნენ, ხარისხი არ გაუმჯობესებულა — **ყველა მიმართულებით გაუარესდა**. მოდელი, რომელსაც არ უსწავლია ამ ამოცანაში საკუთარი ვიზუალური უკუკავშირის გამოყენება, მხოლოდ სურათის დამატებით ამას თავისით ვერ სწავლობს.

ამიტომ სამუშაოს ძირითადი ნაწილი სწავლებაშია. ავტორებმა სასწავლო მონაცემები ისე გადააწყვეს, რომ თითოეული ნახატი ბევრ პატარა, ვიზუალურად აზრიან ნაბიჯად დაიყოს; რთული ფორმები უფრო მარტივ ნაწილებად იშლება, რათა ყოველ ეტაპზე ახალი ვიზუალური ინფორმაცია გაჩნდეს. შემდეგ Qwen3-VL-ზე დაფუძნებული ღია, რვა მილიარდი პარამეტრის მოდელი ამ ნაბიჯ-ნაბიჯ თანმიმდევრობებზე fine-tune-ით გაწვრთნეს. ამას Visual Self-Feedback უწოდეს. ხატვის დროს დაამატეს მეორე მექანიზმიც, Render-and-Verify: ყოველი ახალი შტრიხის მიღებამდე მოდელი მას არენდერებს და ამოწმებს, რეალურად შეცვალა თუ არა სურათი, თუ უბრალოდ წინა მოქმედება გაიმეორა. არაფრის დამმატებელი შტრიხები იშლება; როცა ახალი ნაბიჯი აღარ ეხმარება, მოდელს გაჩერებისკენ უბიძგებენ. აღსანიშნავია, რომ ყველაფერი ეს შედარებით მცირე მონაცემთა ნაკრებზე მუშაობს — დაახლოებით 850,000 მაგალითზე, ერთ კონკურენტზე ნაკლების ნახევარზე და მეორის მონაცემთა მხოლოდ მცირე ნაწილზე.

რა აღმოაჩინეს

- ასე გაწვრთნილი მოდელი თავის „ბრმა“ ვერსიაზე უკეთ ხატავს — განსაკუთრებით აშკარა შეცდომებში, როცა ბრმა მოდელი თვალს ლოყაზე ათავსებს ან მოთხოვნილ სვეტოვან დიაგრამას ტოვებს და მის ნაცვლად ზოგად მონიტორს ხატავს.
- სტანდარტულ MMSVGBench ბენჩმარკზე მეთოდი ძლიერ კონკურენტებთან კონკურენტულია და რამდენიმე მეტრიკით მცირედ უსწრებს მათ — მათ შორის OmniSVG-ს, რომელიც ორჯერ მეტზე მეტ მონაცემზე იყო გაწვრთნილი, და InternSVG-ს, რომელსაც დაახლოებით ოცჯერ მეტი მონაცემი ჰქონდა.
- უპირატესობა მცირეა. მაგალითად, ხატულების ნაკრებზე მისი ძირითადი გამოსახულების ხარისხის ქულაა 127.6, საუკეთესო კონკურენტის — 128.8; prompt-matching ქულა კი 0.293-ია 0.291-ის წინააღმდეგ. მიმართულება რეალური და თანმიმდევრულია, მაგრამ სხვაობა მცირეა.
- ორივე დამატებულ კომპონენტს საკუთარი წვლილი აქვს: სპეციალური სწავლების ან ხატვისას შემოწმების გამორთვა შედეგებს ამცირებს. განსაკუთრებით verification ნაბიჯი უშლის ხელს მოდელს ერთსა და იმავეს უსასრულოდ ხატვაში.
- ავტორები თავად ყველაზე მეტად ეფექტიანობას უსვამენ ხაზს — მსგავს დონემდე მისვლას ლიდერებზე გაცილებით ნაკლები სასწავლო მონაცემით.

რას არ ამტკიცებს ეს

- ნაშრომი **არ აჩვენებს**, რომ მოდელისთვის „ხედვის“ მიცემა უფასო მოგებაა. პირიქით: მათი ექსპერიმენტის მიხედვით ვიზუალური უკუკავშირი **გადამზადების გარეშე შედეგს აუარესებს**. გაუმჯობესება მოდის სწავლიდან და არა მხოლოდ „თვალებიდან“.
- ის **არ ადგენს** დიდ ან გადამწყვეტ უპირატესობას. მეტრიკების უმეტესობაზე მეთოდი კონკურენტებთან ძალიან ახლოსაა; ფრაზა „ოცჯერ მეტ მონაცემზე გაწვრთნილ მოდელს აჯობა“ ფაქტობრივად სწორია, მაგრამ ერთი ბენჩმარკის proxy-მეტრიკებში მცირე სხვაობას ნიშნავს.
- ის **არ ადასტურებს** ზოგად მხატვრულ უნარს. ეს რვა მილიარდი პარამეტრის კვლევითი მოდელია, რომელიც მცირე ფიქსირებულ გარჩევადობაზე (224×224 პიქსელი) ხატულებსა და მარტივ ილუსტრაციებს ქმნის და არა უნივერსალური დიზაინერია.
- დიდი ზოგადი მოდელის (GPT-5) შედარება **არ არის** სრულად თანაბარი: ის ამ ვიწრო კოდით-ხატვის ამოცანისთვის არ არის შექმნილი ან სპეციალურად მორგებული, ამიტომ აქ მისი დამარცხება არც ერთი მოდელის ზოგად შესაძლებლობებზე ბევრს არ ამბობს.
- მეთოდი **უფასო გამოთვლითი ხარჯით არ მოდის**. ყოველი ნაბიჯის რენდერი და ხელახლა წაკითხვა გენერაციას ანელებს ერთჯერად „ბრმა“ კოდირებასთან შედარებით — და ავტორებიც ამას აღიარებენ.

რამდენად ძლიერია მტკიცებულება

- **ძლიერია**, როცა საქმე კონტროლირებულ შედარებას ეხება. ablation-ტესტები მკაფიოა: თუ სპეციალურ სწავლებას ან verification-ს ამოიღებ, შედეგები ეცემა, ამიტომ ორივე კომპონენტი ნამდვილად ასრულებს ავტორების მიერ აღწერილ როლს.
- **საკუთარ მოულოდნელ შედეგთან გულწრფელია.** ის, რომ naive ვიზუალური უკუკავშირი ხარისხს *აუარესებს*, დამალული არ არის. ეს ნაშრომის ერთ-ერთი ყველაზე საინტერესო შედეგია და კარგად ასწორებს ინტუიციას, თითქოს მეტი input ყოველთვის უკეთესია.
- **leaderboard-ის დასკვნები უფრო სუსტია.** უფრო დიდ მონაცემებზე გაწვრთნილ კონკურენტებთან უპირატესობა მცირეა და ერთ ბენჩმარკზეა მიღებული, რომელიც თავად ერთ-ერთი კონკურენტი მოდელის ავტორებმა შექმნეს. მცირე სხვაობები proxy-მეტრიკებში ერთ ტესტზე საინტერესოა, მაგრამ საბოლოო პასუხი არაა.
- **მასშტაბზე და რეალურ გარემოში გამოცდა არ ჩატარებულა.** აქ ყველაფერი დაბალი გარჩევადობის ხატულებსა და მარტივ ილუსტრაციებს ეხება. იმუშავებს თუ არა იგივე მიდგომა რთულ, მაღალი გარჩევადობის ან რეალურ დიზაინზე, მომავალ სამუშაოდ რჩება — ამას ავტორებიც ამბობენ.

რატომ არის მნიშვნელოვანი

ნაშრომის ცენტრალური იდეა თითქმის უხერხულად მარტივია და ხატვას გაცილებით სცდება. თუ პროგრამამ კოდით უნდა შექმნას რაღაც — ვებგვერდი, დიაგრამა, სქემა, 3D სცენა — მას შეუძლია ყველაფერი ბრმად დაწეროს და იმედი ჰქონდეს, რომ სწორად გამოვა, ან გზადაგზა დაარენდეროს და მიმართულება შეასწოროს. ადამიანისთვის ასეთი დახურული ციკლი ბუნებრივია: გვერდს მუდმივად ვუყურებთ. ამ მოდელებისთვის კი ეს გასაკვირად ახალი მიდგომაა.

ნაშრომს ღირებულს სწორედ დამატებული ვარსკვლავი ხდის: მოდელისთვის დანახვის საშუალების მიცემა არ ნიშნავს, რომ მას ყურებაც ასწავლე. „თვალები“ უნდა გაწვრთნა, მხოლოდ ამის შემდეგ იწყებს ციკლი სარგებლის მოტანას — და მაშინაც შედეგი გაზომილია და არა ჯადოსნური: უფრო სტაბილური მხაზველი, და არა სრულიად ახალი ტიპის მხატვარი. მათთვის, ვინც აღწერიდან რედაქტირებად გრაფიკას ქმნის — მაგალითად აპის ხატულებსა და ილუსტრაციებს — ყველაზე პრაქტიკული გაკვეთილი ისაა, რომ მოგება უფრო იაფი და ჭკვიანური სწავლებიდან მოვიდა და არა უბრალოდ მეტი მონაცემიდან.

მოკლე შეჯამება

ვექტორული გრაფიკის გენერირებული ენობრივი მოდელები — ანუ იმ რედაქტირებადი და მასშტაბირებადი კოდის, რომელიც ვებ-ხატულებისა და ლოგოების უკან დგას — ტრადიციულად „ბრმად“ მუშაობდნენ: ყველა ხატვის ბრძანებას წერდნენ ისე, რომ შედეგს საერთოდ არ არენდერებდნენ. Guotao Liang-ის გუნდი გვთავაზობს Render-in-the-Loop-ს: ყოველი ნაბიჯის შემდეგ ნაწილობრივი ნახატი რენდერდება და მოდელს უკან ეძლევა, რათა შემდეგი შტრიხი უკვე ტილოს ნახვის შემდეგ დახატოს. მთავარი და გულწრფელი შედეგი ისაა, რომ ამის უბრალოდ დამატება არსებულ მოდელს **აუარესებს**; გაუმჯობესება მხოლოდ მას შემდეგ ჩნდება, რაც მოდელი ვიზუალური უკუკავშირის გამოყენებაზე გადაიწვრთნება და ხატვისას დამატებითი შემოწმება არაფრის შემცვლელ შტრიხებს აგდებს. გადამზადებული 8-მილიარდ-პარამეტრიანი მოდელი სტანდარტულ ბენჩმარკზე უტოლდება ან ოდნავ უსწრებს კონკურენტებს, რომლებიც ოცჯერ მეტ მონაცემზეც კი იყვნენ გაწვრთნილი. შედეგი რეალურია, მაგრამ სხვაობა მცირეა, მეტრიკები proxy-ებია, ხოლო ამოცანები — დაბალი გარჩევადობის ხატულები და მარტივი ილუსტრაციები. მთავარი დასკვნა ეფექტიანობას ეხება: საკუთარი სამუშაოს სწორად დანახვა ზოგჯერ მონაცემთა უზარმაზარ მასშტაბს ნაწილობრივ ცვლის.

ზედმეტი ხმაურის გარეშე

რას აჩვენებს ნაშრომი: ვექტორული გრაფიკის მოდელის გადამზადება ისე, რომ ის ეტაპობრივად არენდერებდეს და უყურებდეს საკუთარ ნახევრად დასრულებულ ნახატს, „ბრმა“ ხატვაზე უკეთ და სრულ შედეგებს იძლევა; ერთ სტანდარტულ ბენჩმარკზე ის ნაკლები სასწავლო მონაცემით კონკურენტულია და რამდენიმე მეტრიკით ოდნავ უსწრებს უფრო დიდ მონაცემებზე გაწვრთნილ მოდელებს.

რა არის სარწმუნო, მაგრამ დაუმტკიცებელი: რომ „საკუთარი სამუშაოს დანახვა“ ზოგადად უკეთესი სტრატეგიაა, ვიდრე მონაცემთა მასშტაბირება; რომ იგივე იდეა რთულ, მაღალი გარჩევადობის ან რეალურ გრაფიკაზეც იმუშავებს; ან რომ მცირე ბენჩმარკული სხვაობები ადამიანის თვალისთვის შესამჩნევ განსხვავებას ნიშნავს.

რას არ აჩვენებს: რომ ვიზუალური უკუკავშირი თავისთავად ეხმარება (გადამზადების გარეშე ის აზიანებს); რომ კონკურენტებზე უპირატესობა დიდი ან გადამწყვეტია; ზოგად ხატვის უნარს; ან სამართლიან პირდაპირ შედარებას GPT-5-ის მსგავს ზოგად მოდელებთან, რომლებიც ამ ამოცანისთვის არ შექმნილა.

მთავარი შეზღუდვები: ერთი ბენჩმარკი, რომელიც ნაწილობრივ კონკურენტი მოდელის ავტორებმა შექმნეს; მცირე სხვაობები proxy-მეტრიკებში; 8B მოდელი, ხატულები და მარტივი 224×224 ილუსტრაციები; და უფრო ნელი გენერაცია, რადგან ყოველი ნაბიჯი უნდა დაარენდეროს.

რამდენად უნდა ენდოს ამას ზოგადი მკითხველი? მაღალი ნდობა იმისა, რომ ციკლის დახურვა — ხატვისას რენდერი და ხელახლა დანახვა — ნამდვილად ეხმარება და ეს უნარი უნდა ისწავლოს, უბრალოდ „ჩართვა“ არ კმარა. დაბალი-საშუალო ნდობა იმისა, რომ კონკრეტული მოდელი კონკურენტებს გადამწყვეტად სჯობნის; „ოცჯერ მეტ მონაცემს აჯობა“ უნდა წავიკითხოთ როგორც რეალური, მაგრამ მცირე და ერთბენჩმარკიანი შედეგი. მაღალი ნდობა მთავარ იდეაში: თუ კოდი ხატავს, გზადაგზა ყურება ბრმად ხატვას სჯობნის.

წყაროები

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/