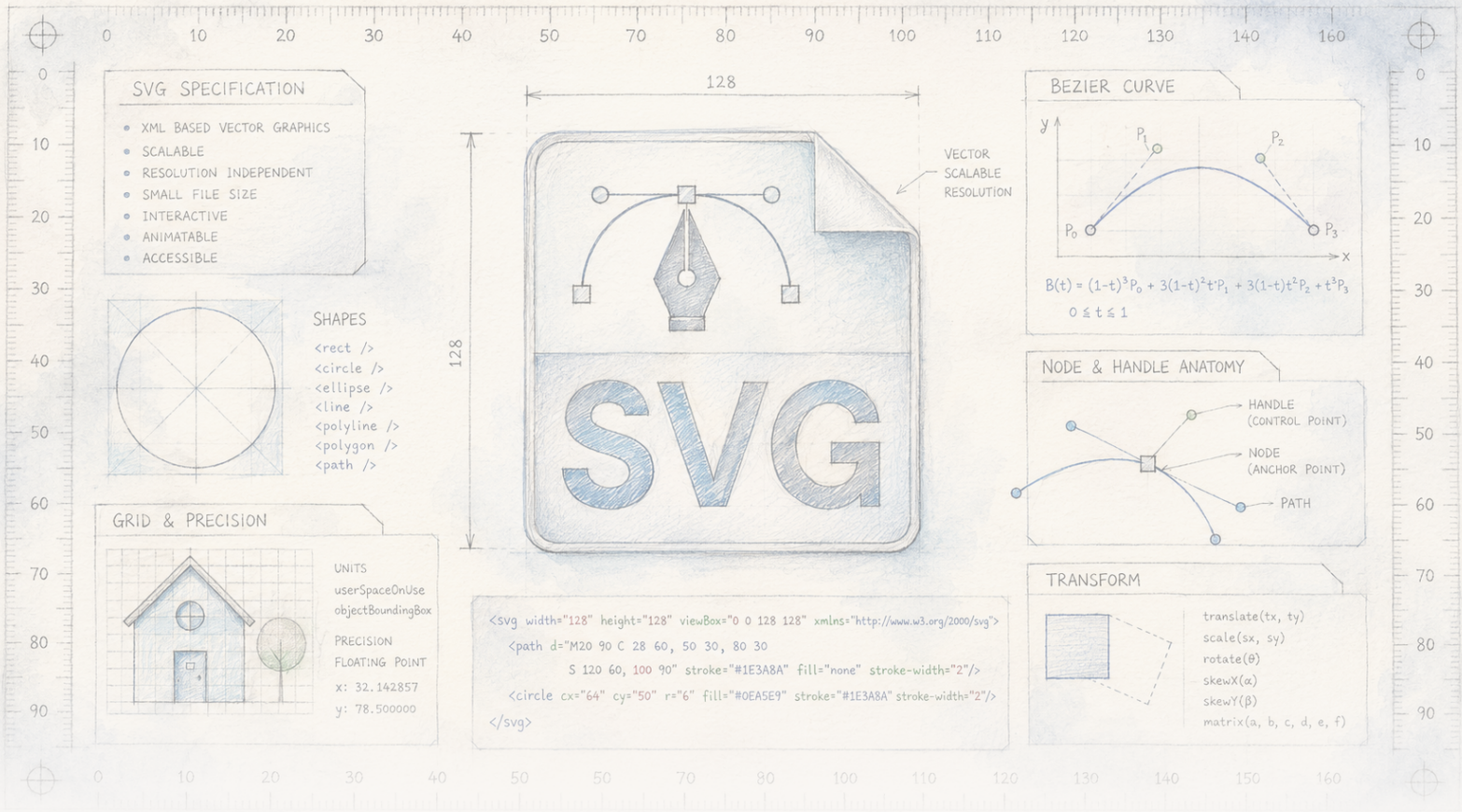


The Clean Paper

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

Çizim yapan bir yapay zekâya sayfaya bakmayı öğretmek

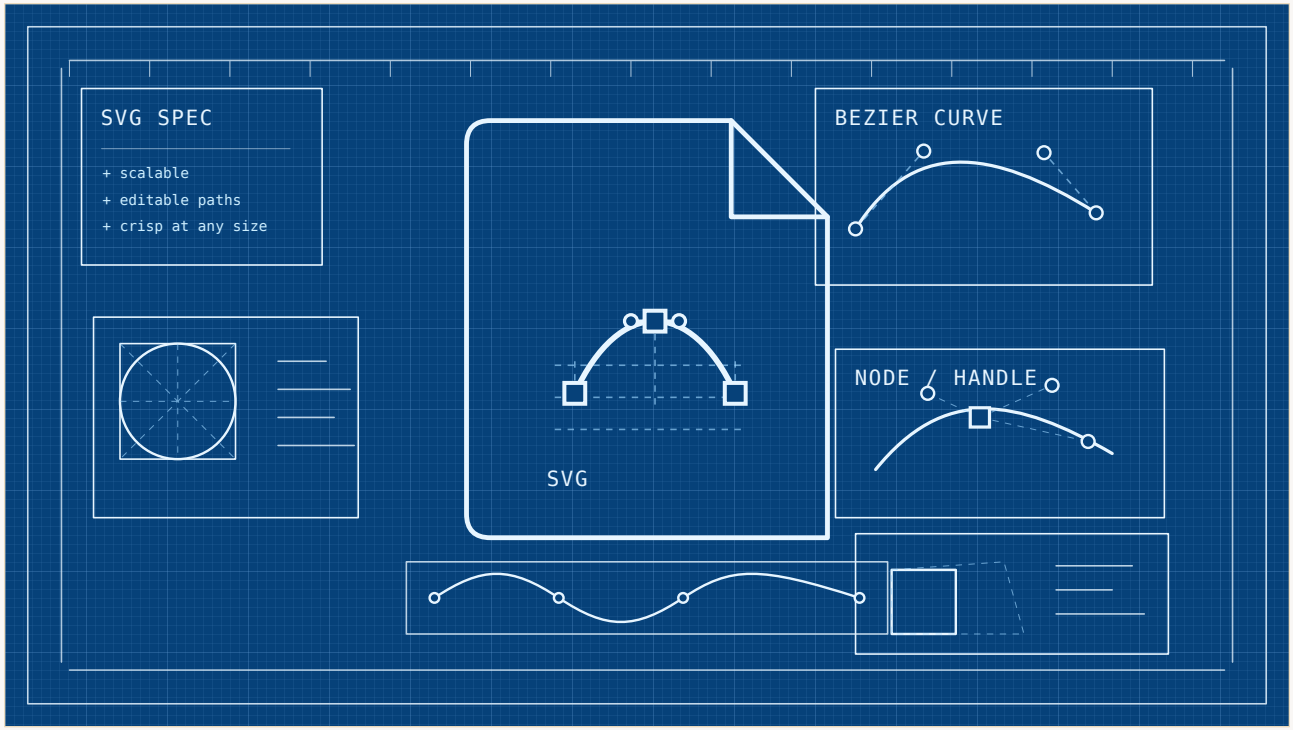
Vektör grafik üreten dil modelleri bunu körlemesine yapar — sonucu hiç görmeden çizim komutlarını yazar. Yeni bir yöntem, modelin kendi tuvalinin çizgi çizgi dolmasını izlemesine izin veriyor; dürüst sürpriz ise şu: yalnızca göz vermek işleri kötüleştiriyor, modeli bu gözleri kullanmak üzere yeniden eğitmek gerekiyor. Sonuç, yirmi kata kadar daha fazla veriyle eğitilmiş rakiplerle başa baş kalan veya onları az farkla geçen bir model — tek bir benchmark'ta gerçek ve dikkatli bir sonuç, devrim değil.

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)

Gözleriniz kapalı resim çizmek

Bugünün yapay zekâ modellerinden birinden size bir resim çizmesini istediğinizde, perde arkasında birbirinden çok farklı iki şeyden biri olur. Bildiğimiz görüntü üreticileri — bir cümleden fotoğraf yaratabilenler — doğrudan pikselleri boyar ve çalışırken tuvali görebilir. Ama daha sessiz bir ikinci çizim türü vardır; burada model hiç boyama yapmaz. Talimatlar yazar: *buraya bir daire çiz, şuraya bir çizgi çek, bu şekli maviyle doldur*. Bu talimatlar koddur — web'deki ikon ve logoların çoğunun arkasında bulunan Scalable Vector Graphics, yani SVG ile aynıdır — ve bunun gerçek bir avantajı vardır: sonuç sabit bir piksel ızgarası değil, bulanıklaşmadan sonsuza kadar yeniden boyutlandırabileceğiniz, renklendirebileceğiniz ve düzenleyebileceğiniz şekiller kümesidir.



Özgün SVG plan çizimi: görüntünün kendisi sabit piksellerden değil, yollar, ızgara çizgileri, düğümler ve tutamaçlardan oluşan düzenlenebilir bir vektör dosyasıdır.

Laura Nesso / The Clean Paper Permission on file

Sorun şu ki yakın zamana kadar bu çizim kodunu yazan model bunu körlemesine yapıyordu. Komut dizisinin tamamını — daire, çizgi, dolgu — tek geçişte üretiyor, ne yaptığını görmek için hiçbir zaman render etmiyordu. Gözleriniz kapalı bir yüz çizmeyi düşünün: gözleri kabaca doğru yere koyabilirsiniz, ama ikincisinin yanağın üstüne düştüğünü ya da saç için çizdiğiniz şeklin artık burnu örttüğünü fark etmenin yolu yoktur. Bu modeller aşağı yukarı böyle çalışıyordu; bu yüzden de biçimsel olarak tamamen geçerli ama görsel olarak karmakarışık kod üretmeleri şaşırtıcı değildi.

Guotao Liang liderliğindeki bir ekip şimdi neredeyse komik derecede bariz olan şeyi yaptı: modelin gözlerini açtı. *Render-in-the-Loop* yöntemleri, her adımdan sonra yarım kalmış çizimi render ediyor ve bir sonraki çizgiyi oluşturmadan önce bu görüntüyü yeniden modele veriyor. Gerçekten ilginç olan, bunun yardımcı olması değil — yardımcı olabilmesi için neler yapmak zorunda kaldıkları.

Bir çizime nasıl puan verilir?

Bir makale kendi modelinin başka bir modeli "geçtiğini" söylediğinde şu soruyu sormak gerekir: hangi işte ve neyle ölçülerek? Üretilmiş bir görüntüyü değerlendirmek gerçekten zordur — "bir dizüstü bilgisayar çiz" komutunun tek bir doğru cevabı yoktur — bu yüzden alan, her biri bir insanın sonuca bakmasının kusurlu bir vekili olan birkaç otomatik puana dayanır. Bu makalede de bazıları kullanılıyor. **FID**, üretilmiş görüntülerden oluşan bütün bir kümenin genel istatistiksel karakterini gerçek görüntülerle karşılaştırır; düşük olması daha iyidir, ama tek bir resmi değil, grubun tamamını anlatır. **CLIP skoru**, ayrı bir yapay zekâya görüntünün istemdeki sözcüklerle eşleşip eşleşmediğini sorar — yararlıdır ama bu hakem ne kadar iyiyse o kadar keskindir. **DINO**, **SSIM** ve **LPIPS** ise yeniden oluşturulmuş görüntüyü bir hedefle, ham piksellerden öğrenilmiş özelliklere kadar farklı düzeylerde karşılaştırır.
Bunların hiçbiri "gerçek" değildir. Bunlar vekil ölçütlerdir ve çoğu zaman küçük adımlarla değişir. Bir modelin 127,6, diğerinin 128,8 aldığını okuduğunuzda, bu hedeflenen yönde gerçek bir farktır — ama heyelan değil, küçük bir itmedir; üstelik gözünüzün söyleyeceği şeyle ancak gevşek biçimde ilişkili bir sayıdaki itme. Başlık "yirmi kat fazla veriyle eğitilmiş modeli geçti" olduğunda bunu akılda tutmakta fayda var.

Yazarlar ne yaptı?

Yazarların çözmek istediği sorun kör çizimin kendisiydi. Mevcut modeller SVG yazmayı tamamen metinsel bir görev olarak ele alıyor: şimdiye kadarki koda bakıp bir sonraki kod parçasını tahmin ediyor ve hiçbir zaman render etmiyor. Bu da modern multimodal modellerin zaten taşıdığı güçlü "gözleri" — görsel kodlayıcıyı — boşta bırakıyor. Render-in-the-Loop işi adım adım görsel bir süreç olarak yeniden düzenliyor. Her çizim kodu parçasından sonra kısmi SVG bir görüntüye render ediliyor ve modele resim olarak geri veriliyor; böylece model bir sonraki parçayı seçerken o ana kadarki tuvale gerçekten bakıyor.

İlk bulguları uyarıcı ve yazarlar bunu açıkça raporluyor: bu döngüyü hazır bir modele basitçe eklemek işe yaramıyor. Ara görüntüleri özel eğitim almamış güçlü genel modellere verdiklerinde kalite artmadı — tam tersine *her yerde düştü*. Gözlerini bu iş için kullanmayı hiç öğrenmemiş bir model, bir anda bakmayı öğrenmiyor.

Dolayısıyla işin büyük kısmı öğretmekte. Eğitim verisini yeniden kurarak her çizimi küçük, görsel olarak anlamlı birçok adıma ayırıyorlar — her aşamada gerçekten yeni bir şey görülebilmesi için karmaşık şekilleri daha basit parçalara bölüyorlar — ve ardından Qwen3-VL üzerine kurulmuş, sekiz milyar parametreli açık bir modeli bu adım adım diziler üzerinde ince ayar yapıyorlar. Buna Visual Self-Feedback adını veriyorlar. Çizim sırasında ikinci bir mekanizma daha ekliyorlar: Render-and-Verify. Model her yeni çizgiyi kabul etmeden önce render ediyor ve gerçekten resmi değiştirip değiştirmediğine, yoksa yalnızca son yaptığı şeyi tekrar mı ettiğine bakıyor. Hiçbir şey eklemeyen çizgiler atılıyor; daha fazla adım fayda getirmediğinde modele durması söyleniyor. Dikkat çekici olan, tüm bunların görece küçük bir veri setiyle yapılması: yaklaşık 850.000 örnek; bir rakibin kullandığının yarısından az, diğerinin ise küçük bir bölümü.

Ne buldular?

- Bu şekilde eğitildiğinde model, kör sürümünden daha iyi çiziyor — fark özellikle hata vakalarında görülüyor: kör modelin gözü yanağa koyduğu ya da istenen çubuk grafiği atlayıp sıradan bir monitör çizdiği örneklerde.
- Standart benchmark'ta (MMSVGBench) yöntem güçlü rakiplerle rekabetçi sonuç veriyor ve bazı ölçütlerde az farkla öne geçiyor — bunlar arasında iki kattan fazla veriyle eğitilen OmniSVG ve yaklaşık yirmi kat veriyle eğitilen InternSVG de var.
- Farklar küçük. Örneğin ikon kümesinde ana görüntü-kalitesi skoru en iyi rakibin 128,8'ine karşı 127,6; istem-eşleşme skoru ise 0,291'e karşı 0,293 — gerçek ve tutarlı, ama ince farklar.
- Eklenen iki parçanın da katkısı var: özel eğitimi veya çizim sırasındaki doğrulamayı kapatınca sayılar ölçülebilir biçimde düşüyor. Özellikle doğrulama adımı, modelin aynı şeyi tekrar tekrar çizip takılı kalmasını engelliyor.
- Yazarların kendilerinin vurguladığı sonuç verimlilik: liderlerin kullandığından çok daha az eğitim verisiyle bu düzeye ulaşmak.

Bunun kanıtlamadığı şeyler

- Modele "görme" yetisi vermenin **bedava bir kazanç olduğunu göstermez**. Tam tersine: makalenin kendi deneyi, yeniden eğitim olmadan görsel geri bildirimin işleri *kötüleştirdiğini* gösteriyor. Kazanç yalnızca gözlerden değil, yeniden eğitimden geliyor.
- Büyük veya kesin bir üstünlük **göstermiyor**. Çoğu skorda yöntem rakipleriyle başa baş; "20 kat veriyle eğitilmiş modeli geçiyor" doğru, ama tek bir benchmark'ta vekil metriklerde küçük farklarla.
- Genel sanatsal yetenek **göstermiyor**. Bu, düşük sabit çözünürlükte (224×224 piksel) ikonlar ve basit illüstrasyonlar çizen sekiz milyar parametreli bir araştırma modeli; genel amaçlı tasarımcı değil.
- Büyük bir genel modelle (GPT-5) karşılaştırma **eşit koşullarda değil**: o model bu dar kodla-çizim işi için tasarlanmış veya ayarlanmış değil; burada onu geçmek modellerden herhangi biri hakkında genel anlamda çok az şey söylüyor.
- Bunun **maliyetsiz olduğunu göstermez**. Her adımda tuvali render edip yeniden okumak, kodu tek kör geçişte üretmekten daha yavaş; yazarlar da bu maliyeti kabul ediyor.

Kanıt ne kadar güçlü?

- **Kendi koşulları içindeki kontrollü karşılaştırmada sağlam.** Ablasyon sonuçları açık: eğitimi çıkarın veya doğrulamayı kaldırın, sayılar düşüyor — yani iki bileşen gerçekten yazarların söylediği işi yapıyor.
- **Kendi şaşırtıcı bulgusuna karşı dürüst.** Saf görsel geri bildirimin *zarar verdiği* bulgusu gizlenmemiş; hatta makalenin en ilginç tarafı bu — daha fazla girdinin her zaman daha iyi olduğu

sezgisine yararlı bir düzeltme.

- **Liderlik tablosu iddiasında ince.** Daha çok veriyle eğitilmiş rakiplere karşı kazançlar küçük ve rakiplerden birinin yazarlarının hazırladığı tek bir benchmark'ta ortaya çıkıyor. Tek test kümesindeki vekil metriklerde küçük farklar umut vericidir, kesin hüküm değil.
- **Ölçekte ve gerçek dünyada test edilmedi.** Buradaki her şey düşük çözünürlükte ikonlar ve basit illüstrasyonlar. Aynı fikrin karmaşık, yüksek çözünürlüklü veya gerçek tasarım işlerinde geçerli olup olmadığı gelecek çalışmalara bırakılmış — yazarlar da bunu açıkça söylüyor.

Neden önemli?

Bu makalenin merkezindeki fikir neredeyse utandıracak kadar basit ve çizimin çok ötesine uzanıyor. Bir program kod yazarak bir şey üretecekse — web sayfası, grafik, diyagram, 3B sahne — ya her şeyi körlemesine yazıp iyi olmasını umabilir ya da ilerledikçe render edip rotasını düzeltebilir. Bu döngüyü kapatmak insan için ikinci doğadır; sayfaya sürekli göz atarız. Bu modeller içinse şaşırtıcı derecede yeni bir hamle.

Makaleyi okumaya değer kılan, bu fikrin yanına koyduğu yıldız işareti. Modele görme yolu vermek, ona bakmayı öğretmekle aynı şey değildir. Gözlerin eğitilmesi gerekir; ancak o zaman döngünün getirisi ortaya çıkar — ve o zaman bile kazanç gerçek ama ölçülüdür: farklı türde bir sanatçı değil, daha istikrarlı bir çizer. Bir açıklamadan düzenlenebilir grafik üreten araçlar geliştiren herkes için — uygulama ikonlarının ve illüstrasyonlarının arkasında çalışan türden — sessiz ve pratik ders en yararlısıdır. Buradaki kazanç daha fazla veriden değil, daha ucuz ve daha akıllı eğitimden geldi. Bu, liderlik tablosunda bir puan daha kazanmaktan daha iyi bir şey öğrenmek demek.

Kısa özet

Vektör grafikler — web ikonları ve logolarının çoğunun arkasındaki düzenlenebilir, ölçeklenebilir kod — üreten dil modelleri geleneksel olarak bunu "kör" yapar: sonucu hiç görmeden bütün çizim komutlarını yazar. Guotao Liang liderliğindeki bir ekip Render-in-the-Loop yöntemini öneriyor: yarım kalmış çizimi her adımdan sonra render edip modele geri vermek; böylece model bir sonraki çizgiyi tuvale bakarak oluşturuyor. Merkezi ve dürüst bulguları şu: bunu hazır bir modele basitçe yapmak modeli *daha kötü* hale getiriyor; kazanç ancak model görsel geri bildirimi kullanmak üzere yeniden eğitildikten sonra ortaya çıkıyor ve çizim sırasında hiçbir şeyi değiştirmeyen çizgileri atan bir kontrol de buna yardım ediyor. Yeniden eğitilen sekiz milyar parametreli model, standart bir benchmark'ta yirmi kata kadar daha fazla veriyle eğitilmiş rakiplerle başa baş kalıyor veya onları az farkla geçiyor — gerçek bir sonuç, ama düşük çözünürlüklü ikonlar ve basit illüstrasyonlarda, vekil puanlarda küçük farklarla. Yazarların vurguladığı ve akılda tutulmaya değer sonuç verimlilik: çalışmasını iyi izleyebilmek, salt veri ölçeğinin bir kısmının yerini tutabilir.

Abartısız değerlendirme

Makale ne gösteriyor: Bir vektör-grafik modelini, yarım kalmış çizimini adım adım render edip ona bakacak şekilde yeniden eğitmek, kör çizime göre daha iyi ve daha eksiksiz sonuçlar üretiyor; daha az eğitim verisiyle tek bir standart benchmark'ta çok daha fazla veriyle eğitilmiş rakiplerle rekabet ediyor ve onları az farkla geçiyor.

Makul ama kanıtlanmamış olan: "Yaptığın işe bakmanın" veri ölçeğini büyütmekten genel olarak daha iyi bir tarif olduğu; aynı fikrin karmaşık, yüksek çözünürlüklü veya gerçek dünya grafiklerinde de yardımcı olacağı; küçük benchmark farklarının bir insanın gerçekten fark edeceği bir görsel farkı yansıttığı.

Göstermediği şey: Görsel geri bildirimin tek başına yardımcı olduğu (yeniden eğitim olmadan zarar veriyor); rakiplere karşı büyük veya kesin bir üstünlük; genel amaçlı çizim yeteneği; bu iş için tasarlanmamış GPT-5 gibi genel modellerle adil bir başa baş karşılaştırma.

Başlıca sınırlamalar: Bir rakibin yazarlarının da katkıda bulunduğu tek bir benchmark; vekil metriklerde küçük farklar; 8B model, 224×224 çözünürlükte ikonlar ve basit illüstrasyonlar; her adımda render döngüsü nedeniyle daha yavaş üretim.

Genel okuyucu ne kadar güvenmeli? Döngüyü kapatmanın — çizerken render edip yeniden bakmanın — gerçekten yardımcı olduğuna ve bunun yalnızca açılan bir özellik değil, eğitilmesi gereken bir yetenek olduğuna yüksek güven. Bu özel modelin rakiplerini kesin biçimde geçtiğine düşük-orta güven; "20 kat veriyle eğitilmiş modeli geçti" cümlesini gerçek ama mütevazı, tek benchmark'a dayalı bir sonuç olarak okuyun. Hatırlanmaya değer ana fikre yüksek güven: çizim yapan kod için ilerledikçe bakmak, kör çizmekten daha iyidir.

Kaynaklar

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/