

2024 ve 2025 Yıllarında Zincirleme Düşünce (Chain-of-Thought) Yöntemleri ve Çok Adımlı Mantık Yürütme Üzerine Araştırmalar

Aşağıda, 2024 ve 2025 yıllarında yayımlanmış önde gelen çalışmalar arasından, büyük dil modellerinde (LLM) **Zincirleme Düşünce (Chain-of-Thought, CoT)** yöntemlerinin çok adımlı problem çözme ve mantık yürütme üzerindeki etkisini inceleyen araştırmalar özetlenmiştir. Her çalışma, başlık, yazarlar, yayın bilgileri, katkılar, kullanılan yöntemler, elde edilen sonuçlar ve varsa belirlenen sınırlılıklar başlıkları altında sunulmuştur.

Active Prompting with Chain-of-Thought for Large Language Models

Yazarlar ve Yayın: Shizhe Diao, Pengcheng Wang, Yong Lin, Rui Pan, Xiang Liu, Tong Zhang – ACL 2024 (62. Yıllık ACL Toplantısı, Uzun Makale) ¹ ² .

Katkılar: Bu çalışma, *chain-of-thought prompting* tekniğini **aktif öğrenme** ile birleştirerek, LLM'lerin farklı görevlerde daha etkili akıl yürütme yapabilmesi için otomatik örnek seçim stratejisi sunar ³ . Yazarlar, *Active-Prompt* adını verdikleri yöntemle, her görev için sabit bir insan yazımı örnek setine bağlı kalmak yerine, belirsizlik temelli bir ölçüt kullanarak hangi soruların açıklamalı (CoT'lu) örnekler olarak seçileceğini belirlemektedir ⁴ . Bu sayede farklı görevler için en faydalı düşünce zinciri örneklerinin seçilmesi hedeflenir.

Yöntem: *Active-Prompt* yönteminde, öncelikle hedef görevle ilgili bir soru havuzu oluşturulur. Ardından, bir LLM'nin bu sorular üzerindeki cevaptan emin olamama düzeyini ölçen çeşitli **belirsizlik metrikleri** (aktif öğrenmeden uyarlanmış) tanımlanır ⁵ . En belirsiz (modelin en çok tereddüt ettiği) sorular, insan tarafından zincirleme düşünce açıklamaları ile anotasyonlanmak üzere seçilir. Bu seçilmiş CoT'lu örnekler, ilgili göreve özel yeni bir *few-shot* CoT isteminde (prompt'unda) kullanılarak modeli uyarlamaya hizmet eder. Böylece her görev için, modelin zorlandığı alanlara odaklanan, dinamik olarak belirlenmiş örneklerle zincirleme düşünce yönlendirmesi yapılır.

Sonuçlar: Önerilen *Active-Prompt* stratejisi, sekiz farklı karmaşık akıl yürütme görevinde, sabit örnekler kullanan geleneksel CoT istemlerine kıyasla belirgin performans artışı sağlamıştır ⁶ . Bu görevler arasında aritmetik problem çözme ve sağduyu çıkarım gibi alanlar bulunmaktadır. Deneysel sonuçlar, aktif seçimin her bir görev için daha uygun örnekler sağladığını ve bunun LLM cevaplarının doğruluğunu yükselttiğini göstermiştir ⁶ . Çalışmada ayrıca farklı belirsizlik metriklerinin etkinliği, havuz boyutunun etkisi, sıfır atış (zero-shot) senaryolar ve model cevabı kesinliği ile doğruluk arasındaki ilişki gibi konular analiz edilmiş, önerilen yöntemin bu açılardan da tutarlı ve başarılı olduğu rapor edilmiştir ⁶ .

Sınırlılıklar: *Active-Prompt* her ne kadar insan emeğini verimli kullansa da, yöntem tamamen insan anotasyonundan bağımsız değildir – seçilen belirsiz soruların yine uzmanlar tarafından CoT açıklamalarıyla etiketlenmesi gerekmektedir. Bu durum, özellikle çok sayıda görev veya soru havuzu için uygulandığında belli bir maliyet oluşturabilir. Ayrıca, hangi belirsizlik ölçütünün her durumda en iyi

seçimi yaptığı konusu göreve göre değişebilir; yazarlar farklı metrikler denemiş olsa da, bazı görevlerde optimum örnek seçimi için ek ayarlamalar gerekebilir.

The Expressive Power of Transformers with Chain of Thought

Yazarlar ve Yayın: William Merrill, Ashish Sabharwal – ICLR 2024 (Poster Sunumu) ⁷.

Katkılar: Bu çalışma, zincirleme düşünce (veya *scratchpad*) kullanımının bir transformer modelinin kuramsal hesaplama gücünü nasıl genişlettiğini analiz eden **teorik bir katkı** sunmaktadır. Standart bir dönüştürücü modelin, girdi okur okumaz yanıt ürettiği senaryoda, bazı basit karar problemlerini (örneğin bir grafin bağlantılı olup olmadığının denetimi veya sonlu durum makinelerinin simülasyonu) çözemeyeceği önceki kuramsal işlerde gösterilmişti ⁸. Bu çalışma ise, **transformer’ların ara adım üretebilmesine izin verilmesinin** (yani modelin cevap vermeden önce belli sayıda ara düşünce adımı üretip bunları girdi olarak geri beslemesinin) modelin hesaplama yeteneklerini nasıl değiştirdiğini inceliyor.

Yöntem: Yazarlar, *decoder* tipi bir transformer’ın kaç adet ara düşünce adımı üretebildiğini (girdi uzunluğunun bir fonksiyonu olarak) dikkate alan bir karmaşıklık analizi yapmışlardır. **Logaritmik sayıda** ara adım (girdi boyutuna göre) izin verilmesinin, modeli klasik transformerlara göre çok az ek güç kattığını bulmuşlardır ⁹. Buna karşılık, **girdi uzunluğuna orantılı (lineer) sayıda** ara adım üretebilen bir transformer mimarisinin (özellikle “projected pre-norm” adı verilen hafif genelleştirilmiş bir mimari altında), modelin **tüm düzenli dilleri (regular languages)** tanıyabilme gücüne erişebildiğini ispatlamışlardır ¹⁰. Daha da önemlisi, polinomik sayıda ara adım üretebilen bir modelin ise, uygun mimari koşullarla birlikte, tam olarak polinom zamanda çözülebilen problem sınıfını (yani P sınıfını) çözebildiği gösterilmiştir ¹¹. Bu sonuç, belirli tipte bir transformer için hesaplama kuramsal bir eşiğin belirlenmesidir ve ilk defa bir transformer çeşidi, standart karmaşıklık sınıflarıyla tam olarak ilişkilendirilmiştir ¹¹.

Sonuçlar: Elde edilen sonuçlar, bir transformer’ın “zincirleme düşünce veya yazboz tahtası (scratchpad)” uzunluğunun, modelin çözebileceği problem türlerini ve zorluk düzeyini dramatik biçimde etkilediğini göstermektedir. Özetle, yeterince uzun bir zincirleme düşünce dizisi kullanabilen bir dil modeli teorik olarak çok daha karmaşık görevleri çözebilir hale gelmektedir. Örneğin, belli bir oranda ara adım kullanımı modelin **bağlam-duyarlı dillere** kadar çıkabileceğini, polinomik sayıda ara adımın ise **polinom-zamanlı tüm problemleri** kapsayabildiğini ortaya koymaktadır ¹¹. Bu çalışma, pratik dil modellerindeki gözlemlerle de uyumlu bir çerçeve sunar: Model girdiyi okur okumaz tek adımda cevap vermek zorunda kalmazsa, arada düşünme adımları ekleyerek daha önce çözemedikleri mantık görevlerini çözebilir hale gelir. Bu bulgular, zincirleme düşüncenin *LLM*’lerin gücünü nasıl temelden artırdığını, ancak bunun için ne kadar “düşünme süresi” gerektiğini niceliksel olarak ortaya koymaktadır.

Sınırlılıklar: Çalışma teorik bir analiz sunmaktadır ve pratikte belirtilen oranlarda ara adımların uygulanması bazı mühendislik kısıtlarına takılabilir. Örneğin, “polinomik sayıda” ara adım üretmek, gerçek modellerde bellek ve zaman açısından uygulanması zor bir durum olabilir. Ayrıca elde edilen sonuçlar, varsayımsal sonsuz model kapasitesi ve belirli mimari varsayımlar (projected pre-norm gibi) altında geçerlidir. Gerçek *LLM*’lerde bu tür sonuçlara yaklaşmak, eğitim ve mimari olarak ek geliştirmeler gerektirebilir. Yine de çalışma, olası sınırları netleştirerek gelecekteki model geliştirmelerine yön vermektedir.

MuSR: Testing the Limits of Chain-of-Thought with Multistep Soft Reasoning

Yazarlar ve Yayın: Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, Greg Durrett – ICLR 2024 (Spotlight Poster) ¹².

Katkılar: Bu çalışma, mevcut büyük dil modellerinin zincirleme düşünce teknikleriyle bile **zorlu çok adımlı akıl yürütme gerektiren senaryolarda yetersiz kalabildiğine** dikkat çekerek, bu sınırları ölçmek amacıyla yeni bir değerlendirme veri kümesi sunar. **MuSR** (Multistep Soft Reasoning) adı verilen bu veri kümesi, gerçek dünya mantık yürütme problemlerine benzeyen **uzun metinli akıl yürütme senaryoları** içerir ¹³. Özellikle dedüksiyon ve çıkarım içeren, örneğin ~1000 kelimelik “cinayet gizemi” tarzı metinler oluşturulmuştur ve bu metinler GPT-4 gibi güçlü modelleri dahi zorlayacak karmaşıklığa sahiptir ¹⁴. MuSR veri kümesi, nörosimgesel (neurosymbolic) bir yöntemle *sunı-özden doğal dile* (synthetic-to-natural) çevrim yapılarak oluşturulmuştur; böylece son derece kompleks ama insan tarafından da anlaşılabilir ve çözümü doğrulanabilir problemler elde edilmiştir ¹⁴.

Yöntem: Veri kümesi oluşturulurken önce sentetik bir mantık yapısı veya senaryo kurgulanmış, daha sonra bu yapı, büyük bir dil modeli yardımıyla doğal dilde bir hikâye/anlatı haline getirilmiştir. Bu sayede her bir problem, *görünürde gerçekçi bir metin* formunda sunulur (örneğin bir dedektif hikâyesi gibi) ancak arka planda bilinçli olarak yerleştirilmiş çıkarım yapma gerektiren ipuçları ve mantık adımları barındırır ¹⁵. MuSR’in iki önemli özelliği vardır: (1) Görev zorluğu istenildiği gibi artırılabilir – yazarlar, mevcut LLM’leri zorlayan senaryolar oluşturduklarını ve ileride daha güçlü modeller çıktıkça veri kümesinin daha da ölçeklenebilir olduğunu belirtmektedir ¹⁴. (2) Problemler tamamen serbest metin (free-form narrative) biçimindedir ve gerçek dünyadaki akıl yürütme durumlarını andırır; bu da onları daha önceki saf sentetik mantık bulmacalarından daha zor ama insanlar için takip edilebilir kılar ¹⁶.

Sonuçlar: Yazarlar, MuSR veri kümesi üzerinde çeşitli büyük dil modellerini (GPT-4 dahil) farklı zincirleme düşünce yönlendirme teknikleri ile değerlendirmişlerdir. Sonuçlar, mevcut CoT yaklaşımlarıyla bile LLM’lerin bu zorlu çok adımlı akıl yürütme senaryolarında önemli hatalar yaptığını ortaya koymuştur. Zincirleme düşünce kullanımı performansı artırsa da, **halen bu tür karmaşık görevlerde güvenilir bir mantık yürütme yapılamadığı, CoT tekniklerinin sınırlarının daha aşılmadığı** gösterilmiştir ¹⁷. Çalışma, kullanılan her bir model ve yönlendirme tekniği için başarısız olunan noktaları da nitel olarak analiz ederek, **zincirleme düşüncenin hangi tür mantık ilişkilerinde yetersiz kaldığını** karakterize etmektedir ¹⁷. Bu sayede gelecekteki araştırmalara, LLM’lerin hangi açılardan geliştirilmesi gerektiğine dair içgörüler sunulmuştur.

Sınırlılıklar: MuSR her ne kadar gerçekçi anlatılar sunsa da, sonuçta *sentetik olarak üretilmiş* bir veri kümesidir. Gerçek hayatta karşılaşılan problemleri ne derece temsil ettiği tartışılabilir. Ayrıca veri kümesinin zorluğu GPT-4 seviyesine göre ayarlandığından, ileride daha güçlü modeller için yeniden ölçekleme gerekebilir. Yine de bu çalışma, LLM’lerin halihazırdaki üst sınırlarını görmek ve onları zorlamak adına önemli bir adım olup, CoT yöntemlerinin gerçekçi senaryolardaki etkinliğini ölçmek için standart bir benchmark ihtiyacını karşılamaktadır.

Chain of Preference Optimization: Improving Chain-of-Thought Reasoning in LLMs

Yazarlar ve Yayın: Xuan Zhang, Chao Du, Tianyu Pang, Qian Liu, Wei Gao, Min Lin – NeurIPS 2024 (Ana Konferans Poster) ¹⁸.

Katkılar: Bu çalışma, *Tree-of-Thought (ToT)* yöntemi ile iyileştirilen mantık yürütme performansını, aynı seviyede tutarken **çalışma zamanındaki verimsizliği ortadan kaldırmayı** amaçlayan yeni bir yaklaşımdır. *Tree-of-Thought* (Düşünce Ağacı) tekniği, bir problemi çözerken dallanıp budaklanan birçok olası düşünce yolunu arama yoluyla keşfederek, saf zincirleme düşüncenin atlayabileceği daha iyi çözüm yolları bulabilmektedir ¹⁹. Ancak bu yaklaşım, defalarca model çalıştırma ve dalları değerlendirme gerektirdiğinden, çıkarım zamanında çok yüksek bir hesaplama maliyeti getirir ¹⁹. Yazarlar, önerdikleri **CPO (Chain-of-Preference Optimization)** yöntemiyle, ToT'nin kapsamlı arama yeteneğini modelin kendisine öğretmeyi, böylece **modeli sonradan çalıştırırken bu ağır arama yüküne ihtiyaç kalmadan benzer performansı elde etmeyi** başarmışlardır ²⁰.

Yöntem: CPO yönteminde, bir LLM öncelikle *Tree-of-Thought* yaklaşımıyla çeşitli problem çözme ağaçlarını oluşturmak üzere kullanılır. Bu ağaçlarda dallar arasındaki **tercih bilgisi** (preference) CPO'nun temelini oluşturur: Model, aynı problemi çözen farklı düşünce zincirlerinden hangisinin doğru cevaba ulaştığına dair bir tercihi içselleştirmelidir ²¹. Bunun için LLM, ToT sürecinde elde edilen arama ağacındaki her adımı referans alacak şekilde **ince ayar ile yeniden eğitilir** ²¹. Başka bir deyişle, modelin ürettiği bir adımın, ToT'nin bulduğu en iyi adımla uyumlu olması hedeflenir. Böylece model, *düşünce adımlarını sıralama ve tercih etme becerisini* öğrenir. Sonuç olarak, CPO ile eğitilmiş model, **yalnızca zincirleme düşünceyle** (ek bir ağaç araması yapmaksızın) neredeyse ToT seviyesinde mantıklı ve isabetli ara adımlar üretebilir hale gelir ²⁰.

Sonuçlar: Deneysel değerlendirmelere göre, CPO ile ince ayar yapılmış LLM, çeşitli karmaşık problem türlerinde belirgin performans artışı göstermiştir ²². Özellikle zorlayıcı akıl yürütme gerektiren görevlerde (örneğin matematik problemleri, bilgi doğrulama ve çok adımlı mantık soruları gibi) CPO uygulaması, modelin doğruluk oranını anlamlı derecede yükseltmiştir ²². Önemli bir bulgu, CPO'nun bu kazanımları sağlarken, çalışma zamanında herhangi bir ek arama prosedürü gerektirmediğidir – model, tek bir zincirleme düşünce akışıyla sonuca varırken önceki haline kıyasla daha doğru sonuçlar üretmiştir ²⁰. Bu da, ToT'nin avantajlarının büyük ölçüde modele kazandırılabilirliğini ve çıkarım sırasında neredeyse **ToT seviyesinde düşünme** yapabildiğini göstermektedir.

Sınırlılıklar: CPO yöntemi, ToT'nin sağladığı iyileşmeleri *eğitim zamanında* ek maliyet ödeyerek kazanır. Yani, bu yaklaşımın başarısı, yeterli kalitede ToT arama ağaçları üretebilmeye ve bunlarla modeli eğitebilmeye bağlıdır. Büyük modeller için bu eğitimin maliyeti yüksek olabilir ve her yeni görev veya model tipi için yeniden bir ağaç tabanlı ince ayar süreci gerekebilir. Ayrıca CPO, ToT'nin bulduğu çözümlere dayanarak modeli yönlendirdiğinden, eğer arama süreci yanlış veya hatalı ise bu durumda modelin öğrenmesi de hatalı olabilir. Yine de, sonuçlar CPO'nun geniş bir yelpazedeki görevlerde etkili olduğunu gösterdiğinden, yöntem zincirleme düşünce performansını artırmak için pratik ve güçlü bir araç sunmaktadır.

Chain-of-Thought Reasoning Without Prompting

Yazarlar ve Yayın: Xuezhi Wang, Denny Zhou – NeurIPS 2024 (Ana Konferans Poster) ²³.

Katkılar: Google araştırmacıları tarafından yapılan bu çalışma, **hiçbir özel CoT yönlendirmesi verilmeden** dil modellerinin içsel olarak çok adımlı akıl yürütebilme kabiliyetlerini ortaya çıkarmaya odaklanır. Bilindiği gibi, *zero-shot* veya *few-shot CoT prompting* yöntemlerinde modele “Haydi adım adım düşünelim” gibi ipuçları verilir ya da örnekli açıklamalar sunulur. Bu çalışma ise şu soruyu sorar: *Bir LLM, herhangi bir CoT talimatı olmadan da adım adım akıl yürütebilir mi?* Yazarların bulgusu, **evet** – doğru çıkarım stratejisiyle, önceden eğitilmiş dil modellerinin içindeki zincirleme düşünce yollarını ortaya çıkarmak mümkündür ²⁴.

Yöntem: Çalışmada önerilen teknik, modelin çıkarım sürecinde izlediği olası alternatif yolları incelemeye dayanır. Normalde bir LLM, en olası sonraki kelimeyi seçerek (greedy decoding) cevap üretir. Bu yöntem genelde modelin **tek bir düşünce yoluna saplanmasına** neden olur. Wang ve Zhou, bunun yerine **top-k olasılıklı seçimleri** değerlendirerek, modelin aslında içsel olarak düşünmüş olabileceği farklı yolları araştırır ²⁵. Yani, her adımdaki en yüksek olasılıklı kelime yerine ikinci, üçüncü olasılıklı devamlar gibi alternatifler de keşfedilir. Bu sayede, modelin orijinal cevabında açıkça görünmeyen ancak **modelin olasılık dağılımında gizli kalmış zincirleme düşünce yolları** sıklıkla ortaya çıkarılabilmektedir ²⁵. Önemli olarak, bu yaklaşım herhangi bir özel istem mühendisliği (prompt engineering) gerektirmez; tamamen **çıkartım zamanı** stratejisi ile ilgilidir. Bu sayede yazarlar, modelin *içsel* mantık yürütme kapasitesini müdahale etmeden ölçebildiklerini belirtirler ²⁶.

Sonuçlar: Araştırma bulgularına göre, herhangi bir CoT talimatı verilmediği halde, modelin olası çıktı uzayında mantıksal ara adımların belirttiği gözlemlenmiştir ²⁵. Dahası, bir çıktı dizisinde eğer zincirleme düşünce adımları mevcutsa, modelin son cevabına olan güveninin belirgin şekilde daha yüksek olduğu rapor edilmiştir ²⁷. Yazarlar, modelin ürettiği cevabın olasılıksal güven skoru ile zincirleme düşünce içerip içermediğinin güçlü bir şekilde ilişkili olduğunu göstermişler; bu sayede, **modelin sadece çıktısına bakarak, aslında adım adım düşünüp düşünmediğini anlamak mümkün olmuştur** ²⁷. Çeşitli mantık yürütme benchmark'ları üzerinde yapılan deneyler, önerilen *CoT-decoding* yönteminin, standart greedy çıktı durumunda gizli kalan akıl yürütme yetilerini açığa çıkardığını ortaya koymuştur ²⁸. Bu yaklaşım, karmaşık görevlerde dahi modelin performansını, herhangi bir ek ince ayar veya talimat vermeksizin iyileştirebilmiştir ²⁸.

Sınırlılıklar: Bu yöntemin pratikteki bir sınırlaması, top-k arama ile alternatif düşünce yollarını çıkarmanın **hesaplama yükünü** artırabilmesidir. Çok adımlı bir soruda her adım için birden fazla alternatif genişletmek, özellikle büyük modellerde, çıkarımı yavaşlatabilir. Ayrıca her alternatif yol her zaman anlamlı veya tutarlı olmayabilir; yazarlar güven ölçütüyle bunları ayıklamış olsalar da, modelin "içsel mantığının" her zaman doğruyu yansıtmayabileceğini unutmamak gerekir. Yine de, bu çalışma önemli bir kavrayış sunar: LLM'ler *kendi kendilerine de* zincirleme düşünce üretebilme potansiyeline sahiptir ve bunu açığa çıkarmak için ille de karmaşık istem hilelerine gerek yoktur.

Chain-of-Thought Unfaithfulness as Disguised Accuracy

Yazarlar ve Yayın: Ana Marasović, Nathan Stringham, Oliver Bentham – NeurIPS 2024 (Ana Konferans Poster) ²⁹.

Katkılar: Bu çalışma, zincirleme düşünce yaklaşımlarının güvenilirliği ve *doğruluk-sadakat* ilişkisini irdeleyen **analitik bir araştırmadır**. Özellikle, bir LLM'in ürettiği zincirleme düşüncenin, modelin içsel hesaplamalarını ne derece yansıttığını (faithfulness) sorgular. Lanham ve arkadaşlarının 2023'te önerdiği bir metrik, modelin son cevabı üretirken kendi CoT sürecine ne kadar "bağımlı" olduğunu ölçmeye çalışmış ve bu metriği CoT'un *sadakat*i için bir vekil olarak önermişti ³⁰. İlginç bir şekilde, Lanham et al. bir dizi kapalı model ailesinde (farklı boyutlarda) bu metriğin ölçeklendikçe önce artıp sonra azalan (scaling-then-inverse-scaling) bir ilişki sergilediğini ve **13 milyar parametrel**i bir modelin, 810M ile 175B arasındaki diğer modellerden daha yüksek *sadakat* değeri gösterdiğini rapor etmişti ³¹. Marasović ve ekibi, bu gözlemlerin genel bir fenomen olup olmadığını, farklı model ailelerinde de geçerli olup olmadığını araştırmıştır.

Yöntem: Çalışma, Lanham ve ark.'nın deney düzeneğini üç farklı LLM ailesi üzerinde tekrarlayarak başlıyor ³². Farklı boyutlardaki modellerin, orijinal metrik altındaki CoT sadakat skorları inceleniyor ve kısmen orijinal çalışmanın bulguları yeniden elde ediliyor (yani bazı modeller için benzer ölçekte "orta boy model en yüksek sadakatte" durumu gözlemleniyor) ³³. Sonrasında yazarlar, metrikte önemli bir

ayarlar yapıyor: Modelin belirli cevaba eğilimini (bias) normalize etmek. Özellikle, modelin bazı yanıt seçeneklerine (örneğin “evet” veya belirli bir sayıya) doğal bir yatkınlığı varsa, bu durum ham metrikte CoT’un etkisini yanlış yüksek gösteriyor olabilir. Normalize edilmiş metriğe geçildiğinde çok çarpıcı bir sonuç ortaya çıkıyor: **Küçük ve daha az yetenekli modellerin görünürdeki “sadakatsizliği” büyük oranda düşüyor** ³⁴. Dahası, normalize edilmiş CoT sadakat metriği ile model doğruluğu arasında çok güçlü bir doğrusal ilişki bulunuyor ($R^2 = 0.74$) ³⁴. Bu da demek oluyor ki, orijinal metrik aslında büyük ölçüde modelin doğruluğunu ölçüyormuş; yani *CoT sadakatinden ziyade CoT kullanan modelin doğru cevap verip vermediğini* yansıtmış.

Sonuçlar: Elde edilen bulgular ışığında yazarlar, Lanham ve ark.’nın CoT sadakat metriğinin **geçerliliğini sorgulamaktadır** ³⁴. Zira bu metriğin yüksek çıkması, modelin gerçekten mantık yürütmesini yansıtmak yerine sadece doğru cevaba ulaşmış olmasıyla ilgili olabilir. Çalışma, zincirleme düşünce çıktılarının model içi duruma sadık olup olmadığını değerlendirmek için daha dikkatli ölçütlere ihtiyaç olduğunu vurgular. Aksi halde, bir model yalnızca tesadüfen doğru cevabı verdiğinde CoT açıklaması da “sadık” görünebilir, ancak bu modelin gerçekten o çözüm yolunu izlediği anlamına gelmez. Bu sonuçlar, **zincirleme düşüncelere güvenme konusunda temkinli olunması** gerektiğine işaret eder. Bir LLM’in açıkladığı ara adımlar her zaman modelin gerçek karar sürecini birebir temsil etmeyebilir; dolayısıyla CoT çıktıları bir çeşit “hesap verme” aracı olarak kullanırken bu bulgu akılda tutulmalıdır.

Sınırlılıklar: Bu çalışma, özellikle Lanham ve ark.’nın metriğine odaklandığından, *faithfulness (sadakat)* ölçümünün zorluğunu bir örnek üzerinden göstermiştir. Elbette ki model karar süreçlerinin şeffaflığı hala açık bir problemdir ve burada ele alınan metrik tek çözüm önerisiydi. Sınırlılık olarak denebilir ki, farklı yaklaşımlar (örneğin karşılaştırmalı nöral aktivite analizleri veya başka metrikler) incelenmemiştir. Ancak çalışma genel anlamda bir uyarı niteliğindedir: Zincirleme düşünce yöntemlerini değerlendirirken, “model bu adımları gerçekten aklından geçirdi mi, yoksa sonradan mı uydurdu?” sorusuna dair güvenilir metrikler geliştirmek gerekecektir.

Graph Chain-of-Thought: Augmenting Large Language Models by Reasoning on Graphs

Yazarlar ve Yayın: Bowen Jin, Chulin Xie, Jiawei Zhang, Kashob Kumar Roy, Yu Zhang, Zheng Li, Ruirui Li, Xianfeng Tang, Suhang Wang, Yu Meng, Jiawei Han – ACL 2024 ³⁵ ³⁶.

Katkılar: Bu çalışma, büyük dil modellerinin zincirleme düşünce esnasında harici **graf tabanlı bilgi kaynaklarını** kullanmasını sağlayarak, bilgi yoğun ve çok adımlı akıl yürütme gerektiren görevlerdeki performanslarını artırmayı hedefler. Yazarlar, pek çok gerçek dünya bilgisinin düz metinler yerine **graf yapıları** (örneğin atıf grafikleri, sosyal ağlar, bilgi grafikleri) şeklinde organize edildiğini vurgular. Klasik CoT yöntemleri genellikle modelin dış bilgiyi metin parçaları şeklinde alıp işlemesine dayanırken, bu çalışma LLM’leri doğrudan grafiklerle etkileşime sokan yeni bir çerçeve sunar: **Graph-CoT**. Ayrıca, bu amaçla **GRBench** adında yeni bir değerlendirme veri kümesi oluşturulmuştur; 10 farklı alan bilgisini temsil eden grafikler kullanılarak 1.740 adet soru hazırlanmıştır ³⁷. Bu soruların yanıtları, ilgili grafikteki düğümler ve bağlantılar aracılığıyla bulunabilir, yani modelin sadece tek tek metinleri değil, bunların grafiksel bağlamını da anlayıp akıl yürütmesi gerekmektedir ³⁸ ³⁷.

Yöntem: *Graph-CoT* çerçevesi, her bir akıl yürütme adımını üç alt adıma ayırarak işler ³⁹: (1) **LLM reasoning:** Dil modeli mevcut soruya ve önceki adımdaki bilgiye dayanarak bir ara adım (düşünce) üretir. (2) **LLM-graph interaction:** Modelin ürettiği bu ara adım, grafikte bir sorgu veya gezinme adımına dönüştürülür. Örneğin model “X kişinin Y ile ilişkisini bul” gibi bir düşünce üretirse, sistem grafikte X ile Y arasında bir yol olup olmadığına bakar. (3) **Graph execution:** Grafik üzerinde gerekli arama/sorgu gerçekleştirilir ve elde edilen sonuç (örneğin ilgili düğüm veya kenar bilgisi) dil modeline

geri beslenir. Model bu yeni bilgiyle bir sonraki reasoning adımına geçer. Bu döngü, model soruyu yanıtlayana dek devam eder; böylece dil modeli, **grafik üzerinde adım adım akıl yürüterek** sonuca ulaşır ³⁹. Yazarlar bu yaklaşımı farklı boyutlardaki üç LLM üzerinde denemişlerdir.

Sonuçlar: *Graph-CoT* yöntemi, GRBench veri kümesinde, her üç model için de geleneksel yöntemlere kıyasla tutarlı bir üstünlük sağlamıştır ⁴⁰. Özellikle, yalnızca metin tabanlı bilgi çağırma ile akıl yürütme yapan modellere göre, grafik etkileşimli CoT'nin daha doğru ve tutarlı cevaplar ürettiği gösterilmiştir. Bu durum, modelin **bağlamsal ve bağlantısal bilgiyi** daha verimli kullanabildiğini, graf yapısındaki ilişkileri adım adım çözerek sonuca varabildiğini göstermektedir. Sonuçlara göre *Graph-CoT*, bilgi grafikleri gibi bağlantılı bilgi kaynaklarının önemli olduğu alanlarda LLM'lerin hata yapma oranını düşürmüş ve muhakeme kabiliyetlerini artırmıştır ⁴⁰.

Sınırlılıklar: *Graph-CoT*'nin başarısı, ilgili problemin grafik biçiminde temsil edilebilir olmasına dayanır. Yani, her görev veya veri kümesi böyle yapısal bir bilgiye sahip olmayabilir. GRBench özelinde sorular, önceden tanımlanmış alan grafiklerine göre hazırlandığından, gerçek dünya için bu yaklaşımın genelleştirilebilirliği veri mevcudiyetine bağlı olacaktır. Ayrıca model ile grafik arasındaki etkileşim adımları, her ne kadar insanlar için anlaşılır olsa da, LLM açısından ekstra karmaşıklık demektir; modelin hem doğal dil akıl yürütme hem de graf sorgulama diline hakim olması gerekir. Yine de bu çalışma, **araç destekli CoT** alanında önemli bir açılım sunar: LLM'lerin düşünme zincirine grafiksel bilgi entegrasyonu, gelecekte karmaşık bilgi tabanlı sorunlarda performansı iyileştirmenin etkili bir yolu olabilir.

Enhancing LLM Reasoning via Vision-Augmented Prompting

Yazarlar ve Yayın: Ziyang Xiao, Dongxiang Zhang, Xiongwei Han, Xiaojin Fu, Wing Yin Yu, Tao Zhong, Sai Wu, Yuan Wang, Jianwei Yin, Gang Chen – NeurIPS 2024 (Ana Konferans) ⁴¹.

Katkılar: Bu çalışma, zincirleme düşünce yöntemlerini metin dışı modalitelere genişleterek, özellikle **görsel ve mekânsal akıl yürütme** içeren problemler üzerindeki LLM performansını artırmayı hedefler. İnsanlar karmaşık bir mantık probleminde genellikle hem sözel hem görsel düşünür; örneğin bir geometri probleminde şekil çizer, bir rota probleminde harita üzerinde yol çizeriz. Güncel dil modeli tabanlı CoT yaklaşımları ise neredeyse tamamen sözel/sayısal düşünceye odaklıdır, bu da görsel ipuçları içeren problemlerde kısıtlı kalmalarına yol açar ⁴². Yazarlar bu açığı kapatmak için **Vision-Augmented Prompting (VAP)** adını verdikleri bir çerçeve önerirler. VAP, modelin sadece metin üreterek değil, aynı zamanda görüntüler oluşturarak ve bunları değiştirerek mantık yürütmesini sağlar ⁴³.

Yöntem: VAP, bir problem tanımı verildiğinde otomatik olarak o probleme dair bir **görsel temsil** oluşturmakla başlar ⁴⁴. Örneğin problem bir geometri sorusuysa, sistem problemde anlatılan şekli çizebilecek bir *dış çizim aracını* kullanarak bir başlangıç görüntüsü üretir ⁴⁵. Ardından, LLM hem metinsel açıklamalar hem de görsel üzerindeki değişiklikler şeklinde bir zincirleme düşünce sürecine girer: Metin ortamında bir ara adım düşünebilir veya gerekli görürse görüntü üzerinde bir düzenleme yapması için araca talimat verebilir ⁴⁵. Bu şekilde model, *çift modlu bir düşünce zinciri* oluşturmuş olur – metin ve görüntü birbirini destekleyerek adım adım çözüm aranır. Örneğin Sudoku çözümü için model tablo görüntüsünü doldururken aynı anda mantık adımlarını listeler; geometri sorusunda şekil üzerinde açılırlar/gölgeleri işaretlerken sözel çıkarım yapar. Süreç boyunca model gerektiğinde görüntüyü yeniden sentezleyip iyileştirir (iteratif refine) ⁴⁵. En sonunda, model kendi içinde hem görsel hem metinsel tutarlılığı sağlayarak, **öz-düzenleme (self-alignment)** adını verdikleri bir mekanizma ile nihai cevabı üretir ⁴⁶.

Sonuçlar: VAP çerçevesi, dört farklı görev türünde test edilmiştir: *Geometri problemleri çözme*, *Sudoku bulmacaları*, *zaman serisi (seri) tahmini* ve *gezgin satıcı (travelling salesman) probleminin çözümü* ⁴⁷. Bu

görevler, görsel-uzamsal mantık yürütmeyi gerektiren veya görsel bir dış temsilden fayda gören çeşitli senaryoları temsil etmektedir. DeneySEL sonuçlar, VAP'ın mevcut LLM tabanlı mantık yürütme yöntemlerine kıyasla belirgin üstünlük sağladığını göstermiştir ⁴⁷. Örneğin, geometri problemlerinde şeklin çizilmesi modelin doğru teoremleri uygulamasını kolaylaştırmış; Sudoku'da görsel tablonun doldurulması hataları azaltmıştır. Model, *yalnızca metinle düşünmeye zorlanan* haline göre çok daha yüksek başarı oranlarına ulaşmıştır ⁴⁷. Bu bulgular, görsel ipuçlarının entegre edilmesiyle bir dil modelinin adım adım problem çözme yetisinin güçlendirilebileceğini kanıtlamaktadır.

Sınırlılıklar: VAP yaklaşımı, ek bir görüntü sentezleme ve işleme bileşeni gerektirir ki bu, her LLM için kolayca uygulanabilir bir şey değildir. Dış araç entegrasyonu, özellikle karmaşık sahnelerde (örneğin gerçekçi görseller) hataya açık olabilir. Çizilen resmin doğruluğu veya yeterliliği, modelin doğru çıkarım yapması için kritiktir; sentezlenen görüntü hatalıysa model de hataya sürüklenebilir. Ayrıca, VAP'ın etkinliği gösterilen dört senaryoyla sınırlıdır ve diğer modalite kombinasyonlarında (örneğin sesli mantık yürütme) benzer yaklaşımların nasıl genellenebileceği açık bir araştırma konusudur. Yine de bu çalışma, LLM'lerde *tool-augmented reasoning* (araç destekli mantık yürütme) alanına önemli bir katkı sağlayarak, daha insan-benzeri çok modlu düşünme yeteneklerinin kapısını aralamaktadır.

COCONUT: Chain of Continuous Thought (Latent Space Reasoning)

Yazarlar ve Yayın: Shibo Hao, Sainbayar Sukhbaatar, Di Jia Su, Xian Li, Zhiting Hu, Jason E. Weston, Yuandong Tian – (ICLR 2025 Konferansı için sunulmuş, 2024 sonu itibarıyla değerlendirilmiş araştırma)

⁴⁸

Katkılar: Bu çalışma, zincirleme düşünce kavramını kökten farklı bir bağlama oturtarak, **dil modellerinin düşünme süreçlerini insan dilinin kısıtlarından kurtarmayı** amaçlar. Güncel LLM'ler, mantık yürütme süreçlerini dilsel ara adımlar olarak dışa vurmaktadır; yani düşündüklerini yine birer cümle veya token dizisi şeklinde ifade ederler. *COCONUT* (*Chain of Continuous Thought*) yaklaşımı ise, modelin düşünme adımlarını **gizli vektör temsilleri** (latent states) düzeyinde yapmasını öngörür ⁴⁹. Böylece model, doğal dilin getirdiği gereksiz kelimelerden, gramer yükünden veya belirli bir kelime dizisi oluşturmaktan bağımsız, daha saf bir düşünme sürecine sahip olabilir.

Yöntem: Yazarlar, bir LLM'in her adımda ürettiği son gizli durum vektörünü, o andaki "düşünce durumu" olarak ele alıyorlar ⁵⁰. Normalde, model bu vektörden bir sonraki kelimeyi (token'i) çıkarır ve çıktıyı tekrar kendine girdi yaparak devam eder. *COCONUT*'ta ise bu son gizli durum doğrudan bir sonraki adımın girdi vektörü olarak modele geri besleniyor, **hiç kelimeye dönüştürülmeden** ⁵¹. Bu şekilde model, arka planda kendi iç dilinde düşünüyor gibi bir durum oluşuyor. Ara adımlar dil olarak dışa vurulmadığından, model insanlar için anlaşılır bir scratchpad üretmiyor; onun yerine model içindeki *sürekli (continuous) düşünce alanında* hesaplama sürüyor. Bu yaklaşımın eğitim ve kullanımı için, yazarlar modeli önce bu tarz sürekli düşünecek şekilde eğitiyorlar (muhtemelen özel bir kayıp fonksiyonu veya mimari ayar ile). Elde edilen model, hem salt CoT (dışa yazdırarak düşünme) modunda hem de *COCONUT* modunda çeşitli görevlere sokulup karşılaştırılıyor.

Sonuçlar: Deneyler, *COCONUT* paradigmasının bir dil modelini çeşitli akıl yürütme görevlerinde belirgin biçimde geliştirebildiğini göstermiştir ⁵². Hatta bazı zorlayıcı mantık problemlerinde (özellikle ciddi planlama gerektiren bulmacalar gibi), *COCONUT* yaklaşımıyla düşünen model, klasik dil ortamında zincirleme düşünceyle düşünen halini geride bırakmıştır ⁵². İlginç bir bulgu da, gizli alanda gerçekleşen sürekli düşüncenin, modelin birden fazla olası gelecek adımı *aynı anda* temsil etmesine imkân tanımasıdır ⁵³. Yazarlar, latent düşünce durumunun adeta birden fazla olasılığı süperpoze edebildiğini ve bunun bir çeşit **genişlik-öncelikli arama (breadth-first search)** etkisi yarattığını

belirtmişlerdir ⁵³ . Klasik CoT'da model bir yola karar verip onun boyunca derinleşirken, *COCONUT* modelinin tek bir seferde birçok yönde düşünebildiği, böylece tek bir yolak hatalıysa diğerini değerlendirebildiği gözlemlenmiştir. Bu durum, zincirleme düşüncenin tek bir deterministik yola erken saplanma sorununu hafifletebileceğini gösterir. Genel olarak, çalışma *latent reasoning* (örtük alan akıl yürütmesi) konusunda ufuk açıcı sonuçlar sunmakta ve dil modellerinin düşünme biçimleri üzerine yeni bir perspektif getirmektedir ⁵⁴ .

Sınırlılıklar: Bu yenilikçi yaklaşımın en bariz sınırlılığı, modelin düşünce sürecini insanlar tarafından doğrudan izlenemez hale getirmesidir. Zincirleme düşüncenin bir avantajı, modelin ara adımlarını inceleyerek hataları düzeltme veya modelin nasıl düşündüğünü anlama imkânı sunmasıydı. *COCONUT* ise bu adımları gizli hale getiriyor. Bu durum, **yorumlanabilirlik** açısından bir geri adım olarak görülebilir. Ayrıca, sürekli uzayda düşünmek, modelin dil geri dönüşü (feedback) almasını zorlaştırabilir; örneğin klasik CoT'da model yanlış bir ara sonuç yazdığında bunu tespit etmek mümkünken latent alanda yanlış yolda gidip gitmediğini anlamak güç olabilir. Son olarak, bu yaklaşım özel bir eğitim süreci gerektirdiğinden, mevcut dil modellerine doğrudan uygulanması pratikte zordur; yeni baştan eğitilmiş modeller gerekebilir. Yine de *COCONUT*, dil modellerinin mantık yürütme sınırlarını zorlayan radikal bir fikir olarak, ileride daha verimli ve güçlü akıl yürüten yapay zeka sistemleri tasarlamak için değerli bir bakış açısı sunmaktadır.

Q*: Improving Multi-step Reasoning for LLMs with Deliberative Planning

Yazarlar ve Yayın: Chaojie Wang, Yanchen Deng, Zhiyi Lü, Shuicheng Yan, Bo An – (2024 yılında çevrimiçi yayınlanmış, muhtemelen konferans değerlendirme aşamasından geçen çalışma) ⁵⁵ .

Katkılar: *Q* adını verdikleri bu çalışma, LLM'lerin çok adımlı mantık yürütme problemine bir planlama ve arama perspektifinden yaklaşır. Yazarlar, bir LLM'in adım adım çözüm üretme sürecini bir Markov Karar Süreci (Markov Decision Process, MDP) olarak çerçevelendiriyorlar ⁵⁶ . Bu sayede, dil modeli için her ara düşünce adımı bir eylem olarak, o ana kadarki çözüm durumu bir durum olarak ve nihai doğru cevaba ulaşma başarısı da bir ödül olarak tanımlanabiliyor ⁵⁶ . Bu yaklaşımla, klasik planlama algoritmalarını (A, BFS gibi arama tekniklerini) LLM'in düşünce zincirine uygulamak mümkün hale geliyor. Özetle, *Q yöntemi*, düşünce zincirlerini arama uzayında en iyi yol olarak bulma* problemine indirgemekte ve LLM çıktısını bu aramayla yönlendirmektedir.

Yöntem: Yazarlar öncelikle bir **Q-değeri** fonksiyonu öğreniyorlar; bu fonksiyon verilen bir ara durum (o ana dek üretilen düşünce zinciri) ve bir olası sonraki adım için, nihai çözüme ulaşmada ne kadar umut vadettiğini skorlayacak şekilde eğitiliyor ⁵⁷ ⁵⁸ . Bu Q-fonksiyonunu öğrenmek için çeşitli yöntemler kullanılmış; simülasyonlar (rollout) ile sonuç değerlendirme, daha güçlü bir modelle tamamlanmış çözümlerden öğrenme gibi teknikler makalede bahsedilmektedir ⁵⁹ ⁶⁰ . Eğitilen Q-fonksiyonuna “tak-çıkart (plug-and-play) biçimde” erişilebildiği için, asıl dil modelini fine-tune etmeye gerek kalmadan, çıkarım esnasında bu Q değerlerini bir *heuristic* (yordamsal yol gösterici) olarak kullanmak mümkün olur ⁵⁷ . Özellikle, arama algoritması olarak **A*** (A-star) yönteminden esinlenilmiştir: Model bir ara adım ürettikten sonra, Q-fonksiyonu o adımın ne kadar işe yarayacağını değerlendirir; model, en yüksek Q değerine sahip adımları tercih ederek **best-first search** tarzı bir ilerleme kaydeder ⁶¹ . Böylece, zincirleme düşünce süreci körlemesine derinlemesine gitmek yerine, her adımda “bu adım beni çözüm yolunda ilerletir mi?” sorusuna yanıt arayarak dallanır. Bu yaklaşımın önemli avantajı, her yeni görev için modele özel fine-tuning yapmaya gerek kalmamasıdır – Q* çerçevesi, bir kez eğitilen Q-fonksiyonuyla farklı görevlerde de uygulanabilir ⁵⁷ ⁵⁸ .

Sonuçlar: Q yöntemi, özellikle matematik problemleri (GSM8K, MATH veri setleri) ve programlama kodu üretimi (MBPP gibi) görevlerinde kapsamlı deneyimlerle değerlendirilmiştir. Sonuçlar, Q çerçevesinin bu görevlerde LLM'lerin çok adımlı akıl yürütme kabiliyetini belirgin biçimde geliştirdiğini ortaya koymaktadır ⁶². Örneğin, matematik sorularında ara adımların tutarlılığı ve doğruluğu artmış, kod üretiminde gerektiğinde ileri-geri planlama yapılarak sentaks ve mantık hataları azalmıştır. Yazarlar, Q uygulamasının hiçbir model parametresini değiştirmeden* (yani sadece çıkarım zamanı rehberliğiyle) başarıya ulaştığının altını çizmektedirler ⁵⁷. Bu, derin öğrenme modellerinde nadir görülen bir avantajdır; genelde performans artışı için modelin tekrar eğitilmesi beklenir. Ayrıca yöntem, Monte Carlo Tree Search gibi alternatif planlama tekniklerine göre de daha verimli bulunmuştur – çünkü her adımda komple bir simülasyon yapmak yerine tek adımlık bir kestirim ile yönlendirme yapmaktadır ⁶³ ⁵⁸.

Sınırlılıklar: Q yönteminin de bazı potansiyel sınırlılıkları mevcuttur. Öncelikle, Q-değeri tahmin modelinin doğru çalışması kritik; eğer bu model hatalı yönlendirme yaparsa, dil modeli yanlış bir yola sapabilir. Q-fonksiyonunun eğitimi için kullanılan yöntemler (rollout, güçlü model yardımı vb.), her problem türü için farklı ayar gerektirebilir ve her senaryoda yeterince iyi bir kestirim sağlanamayabilir. Ayrıca Q çıkarım sırasında modelin her adımını değerlendirdiği için, standart zincirleme düşünceye nazaran biraz daha hesaplama yapar; ancak yazarlar bunun tam ağaç aramasından çok daha ucuz olduğunu belirtmiştir ⁶³. Son olarak, Q yaklaşımı LLM çıktısını kısmen arama algoritmasının insafına bırakmaktadır; bu, dilsel yaratıcılık veya esneklik gerektiren görevlerde istenmeyen kısıtlamalar getirebilir. Yine de bu çalışma, planlama ve arama tekniklerinin LLM'lere entegre edilmesi* konusunda önemli bir adım olup, çok adımlı mantık yürütme problemlerinde standart CoT yöntemlerinin ötesine geçmek için güçlü bir çerçeve sunmaktadır.

Kaynaklar: Çalışmalarda sunulan tüm teknik detaylar ve sonuçlar, ilgili makalelerin özetlerinden derlenmiştir. Her bir araştırmamanın bulguları, bağlı kaynaklarda belirtilen satır aralıkları ile doğrulanabilir:

- Active Prompting with CoT ³ ⁶
- Expressive Power of Transformers with CoT ¹⁰ ¹¹
- MuSR: Multistep Soft Reasoning ¹⁴ ¹⁷
- Chain of Preference Optimization (CPO) ²⁰ ²²
- CoT Reasoning Without Prompting ²⁵ ²⁸
- CoT Unfaithfulness vs Accuracy ³¹ ³⁴
- Graph Chain-of-Thought (Graph-CoT) ³⁷ ³⁹
- Vision-Augmented Prompting (VAP) ⁴⁵ ⁴⁷
- COCONUT: Latent CoT ⁵¹ ⁵³
- Q*: Deliberative Planning for LLMs ⁵⁷ ⁶²

- 1 2 **Active Prompting with Chain-of-Thought for Large Language Models - ACL Anthology**
<https://aclanthology.org/2024.acl-long.73/>
- 3 4 5 6 **Active Prompting with Chain-of-Thought for Large Language Models | OpenReview**
<https://openreview.net/forum?id=wabp68RoSP>
- 7 8 9 10 11 **ICLR Poster The Expressive Power of Transformers with Chain of Thought**
<https://iclr.cc/virtual/2024/poster/18776>
- 12 13 14 15 16 17 **ICLR Poster MuSR: Testing the Limits of Chain-of-thought with Multistep Soft Reasoning**
<https://iclr.cc/virtual/2024/poster/18015>
- 18 19 20 21 22 **NeurIPS Poster Chain of Preference Optimization: Improving Chain-of-Thought Reasoning in LLMs**
<https://neurips.cc/virtual/2024/poster/96804>
- 23 24 25 26 27 28 **NeurIPS Poster Chain-of-Thought Reasoning Without Prompting**
<https://neurips.cc/virtual/2024/poster/96654>
- 29 30 31 32 33 34 **NeurIPS Poster Chain-of-Thought Unfaithfulness as Disguised Accuracy**
<https://neurips.cc/virtual/2024/poster/99347>
- 35 36 37 38 39 40 **[2404.07103] Graph Chain-of-Thought: Augmenting Large Language Models by Reasoning on Graphs**
<https://arxiv.org/abs/2404.07103>
- 41 42 43 44 45 46 47 **Enhancing LLM Reasoning via Vision-Augmented Prompting**
https://proceedings.neurips.cc/paper_files/paper/2024/hash/328c922d068dd4ccb23cec5c64e6c7fc-Abstract-Conference.html
- 48 49 50 51 52 53 54 **Training Large Language Model to Reason in a Continuous Latent Space | OpenReview**
<https://openreview.net/forum?id=tG4SgayTtk>
- 55 56 57 58 59 60 61 62 63 **Q*: Improving Multi-step Reasoning for LLMs with Deliberative Planning**
<https://arxiv.org/html/2406.14283v1>