

CausalARC:

Abstract Reasoning with Causal World Models

Jacqueline Maasch
Cornell Tech, New York, NY

John Kalantari
YRIKKA, New York, NY

Kia Khezeli
YRIKKA, New York, NY

Abstract

Reasoning requires adaptation to novel problem settings under limited data and distribution shift. This work introduces CausalARC: an experimental testbed for AI reasoning in low-data and out-of-distribution regimes, modeled after the Abstraction and Reasoning Corpus (ARC). Each CausalARC reasoning task is sampled from a fully specified *causal world model*, formally expressed as a structural causal model. Principled data augmentations provide observational, interventional, and counterfactual feedback about the world model in the form of few-shot, in-context learning demonstrations. As a proof-of-concept, we illustrate the use of CausalARC for four language model evaluation settings: (1) abstract reasoning with test-time training, (2) counterfactual reasoning with in-context learning, (3) program synthesis, and (4) causal discovery with logical reasoning.

1 Introduction

Humans are exceptional few-shot learners that use internal representations of the world to navigate novel scenarios [12, 20]. These internal *world models* can enable general reasoning, for which generalization is both robust under “known unknowns” and flexible under “unknown unknowns” [4]. World models can encode beliefs about cause-effect relationships, supporting general reasoning at all three levels of the Pearl Causal Hierarchy (PCH; Figure 1) [2]. Though causal reasoning is a hallmark of human cognition [9] and a desideratum for human-like AI [32, 19], state-of-the-art generative models do not yet display general reasoning at all three levels of the PCH [17, 11, 45, 37, 16, 25].

Despite the emergence of few-shot [3] and in-context learning [43] in language models (LMs), general reasoning still poses a major distribution shift challenge. One promising direction for out-of-distribution problem-solving is *test-time training* (TTT) [40]: a strategy where model parameters are temporarily updated on each test instance (or batch) for on-the-fly adaptation to problems outside the pretraining distribution [30, 24]. Testimony to its utility for reasoning out-of-distribution, TTT was the dominant strategy in winning submissions for the first Abstraction and Reasoning Corpus (ARC) [4, 5, 23, 1]: the premier benchmark for general intelligence in AI to-date. ARC is a grid world of highly diverse abstract reasoning tasks that are generally solvable by humans. Each task is governed by a unique rule that deterministically transforms input arrays to output arrays, as demonstrated by few-shot learning examples [2]. Ablation studies show that fine-tuning plus TTT with few-shot, in-context learning can significantly improve LM performance on ARC relative to base models [1].

This work extends and reconceptualizes the ARC setup to support causal reasoning evaluation under limited data and distribution shift. We introduce CausalARC: an experimental testbed of ARC-

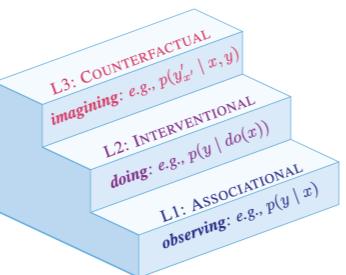


Figure 1: The PCH: observing factual realities (L1), exerting actions to induce interventional realities (L2), and imagining alternate counterfactual realities (L3) [2].

Ablation studies show that fine-tuning plus TTT with few-shot,

in-context learning can significantly improve LM performance on ARC relative to base models [1].

arXiv:2509.03636v1 [cs.AI] 3 Eylül 2025

CausalARC:

Nedensel Dünya Modelleriyle Soyut Akıl Yürütme

Jacqueline Maasch
Cornell Tech, New York, NY

John Kalantari
YRIKKA, New York, NY

Kia Khezeli
YRIKKA, New York, NY

Özet

Akıl yürütme, sınırlı veri ve dağılım kayması altında yeni problem ortamlarına uyum sağlama gereklidir. Bu çalışma, Soyutlama ve Akıl Yürütme Külliyesi (ARC) örnek alınarak modellenmiş, az veri ve dağılım dışı senaryolarda YZ akıl yürütmesi için tasarlanmış deneyimel bir test ortamı olan CausalARC’ı tanıtmaktadır. Her bir CausalARC akıl yürütme görevi, resmi olarak yapısal bir nedensel model olarak ifade edilen, tam olarak belirtilmiş bir nedensel dünya modeli ‘nden örneklenir. *Prensip sahibi veri artırımları, az örnekli, bağlam içi öğrenme gösterimleri şeklinde nedensel dünya modeli hakkında gözlemler, müdahaleci ve karşıolgusal geri bildirim sağlar.* Bir kavram kanıtı olarak, CausalARC’ın dört dil modeli değerlendirme ayarında kullanımını açıklyoruz: (1) test anı eğitimi ile soyut akıl yürütme, (2) bağlam içi öğrenme ile karşıolgusal akıl yürütme, (3) program sentezi ve (4) mantıksal akıl yürütme ile nedensel keşif.

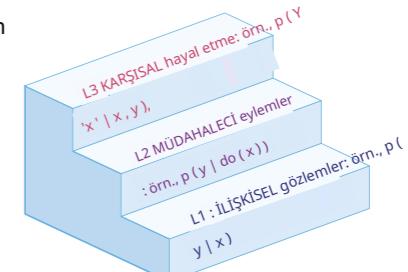
1 Giriş

İnsanlar, yeni senaryolarda yol bulmak için dünyanın dahili temsillerini kullanan istisnai az örnekli öğrenicilerdir [12, 20]. Bu dahili dünya modelleri genel akıl yürütme hale önemli hem “bilinen bilinmezler” altında güçlü hem de “bilinmeyen bilinmezler” altında esnektdir [4]. Dünya modelleri, neden-sonuç ilişkileri hakkındaki inançları kodlayarak, Pearl Nedensel Hiyerarşisi’nin (PNH; Şekil 1) her üç düzeyinde de genel akıl yürütme destekleyebilir [2]. Nedensel akıl yürütme, insan bilişinin bir ayırt edici özelliği [9] ve insan benzeri YZ için bir desideratum (arzu edilen özellik) [32, 19] olmasına rağmen, en güncel üretken modeller henüz PNH’nin her üç düzeyinde de genel akıl yürütme sergilememektedir [17, 11, 45, 37, 16, 25].

Dil modellerinde (DM’ler) az örnekli [3] ve bağlam içi öğrenmenin [43] ortaya çıkışına rağmen, genel akıl yürütme hala önemli bir dağılım kayması zorluğu teşkil etmektedir. Dağıtım dışı problem çözme için umut vadeden bir yön, test anı eğitimi (TTT) [40] ’dir: model parametrelerinin, ön eğitim dağılımı dışındaki problemlere anında adaptasyon için her test örneği (veya grubu) üzerinde geçici olarak güncellendiği bir strateji [30, 24]. Dağıtım dışı akıl yürütme için faydasının bir kanıtı olarak, TTT, ilk Soyutlama ve Akıl Yürütme Külliyesi (ARC) [4, 5, 23, 1] için kazanan başurularda baskın stratejidi: YZ’de bugüne kadarki genel zeka için onde gelen karşılaştırma ölçütü. ARC, insanlar tarafından genellikle çözülebilen, son derece çeşitli soyut akıl yürütme görevlerinden oluşan bir izgara dünyasıdır. Her görev, az örnekli öğrenme örnekleri gösterdiği gibi, girdi dizilerini çıktı dizilerine deterministik olarak dönüştüren benzersiz bir kural tarafından yönetilir.

Ablasyon çalışmaları, az örnekli ince ayar ve TTT ile birlikte, bağlam içi öğrenmenin, temel modellere kıyasla ARC üzerindeki DM performansını önemli ölçüde artırmabileceğini göstermektedir [1].

Bu çalışma, sınırlı veri ve dağılım kayması koşullarında nedensel akıl yürütme değerlendirmesini desteklemek amacıyla ARC kurulumunu genişletmeye ve yeniden kavramsallaştırmaktadır. CausalARC’ı sunuyoruz: ARC-



Ön Baskı.

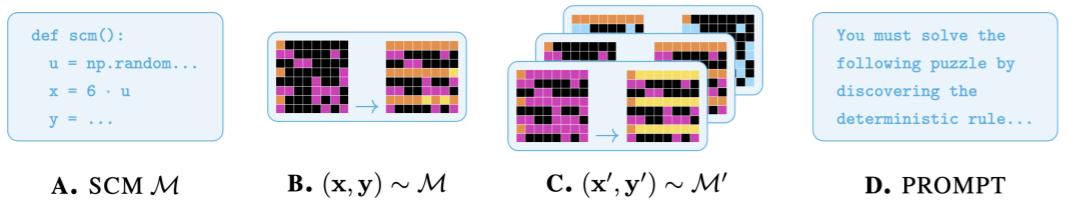


Figure 2: The CausalARC testbed. (A) First, SCM \mathcal{M} is manually transcribed in Python code. (B) Input-output pairs are randomly sampled, providing observational (L1) learning signals about the world model. (C) Sampling from interventional submodels \mathcal{M}' of \mathcal{M} yields interventional (L2) samples (x', y') . Given pair (x, y) , performing multiple interventions while holding the exogenous context constant yields a set of counterfactual (L3) pairs. (D) Using L1 and L3 pairs as in-context demonstrations, we can automatically generate natural language prompts for diverse reasoning tasks.

like reasoning tasks sampled from fully specified *causal world models* (Figure 2). World models are expressed as probabilistic *structural causal models* (SCMs), a rich formalism for representing data generating processes and simulating hypothetical outcomes [32, 33]. Given a fully specified SCM, all three levels of the PCH are well-defined: any observational (L1), interventional (L2), or counterfactual (L3) query can be answered about the environment under study [2]. This formulation makes CausalARC an open-ended playground for testing reasoning hypotheses at all three levels of the PCH, with an emphasis on abstract, logical, and counterfactual reasoning.

Design CausalARC is designed to address multiple core problems in AI reasoning evaluation:

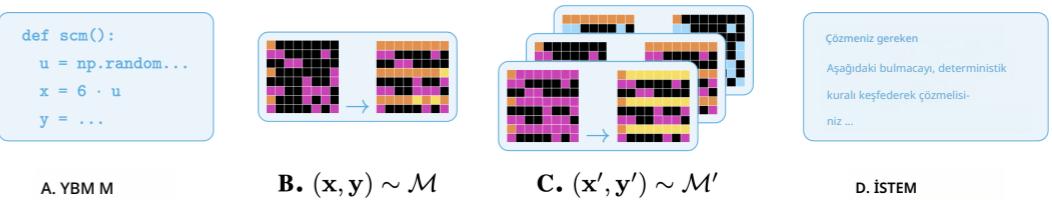
- *Reasoning tasks as generative models.* CausalARC frames each task as a generative model (SCM) from which task instances can be randomly sampled. Reasoning robustness can be benchmarked with respect to a distribution of task instances from each world model, enabling uncertainty estimation. Generative benchmarks that randomly sample from world models can also buffer against static benchmark data leakage, a major challenge in AI evaluation [27, 44, 25, 10].
- *Disentangling signal from noise.* Differentiating true reasoning from statistical recall is a nontrivial objective in AI evaluation [16, 44, 27, 36]. ARC-like settings are conducive to this objective, as abstract reasoning relies more on innate cognitive priors than factual knowledge [4].
- *Random sampling at all levels of the PCH.* Unlike prior ARC extensions, the user can sample L1, L2, and L3 distributions of any sample size from a given world model. This feature also facilitates causal discovery and causal inference queries that require a sample distribution.
- *Fine-grained error analyses.* CausalARC is designed for hypothesis-driven model evaluations and detailed error analyses. As in ConceptARC [29], tasks are labeled by reasoning theme (logical, counting, extension, and ordering). As in Shojaee et al. [36], CausalARC supports evaluation with respect to scaling problem complexity by configuring most tasks with tunable array sizes.

Intended Use Cases CausalARC is a richly annotated experimental testbed for exploring diverse AI reasoning hypotheses. Unlike ARC-Heavy [23], ARC-Potpourri [23], and RE-ARC [14], CausalARC is not intended to be a large-scale fine-tuning dataset. We avoid the automated data augmentations used in Li et al. [23], Hodel [14], and Akyürek et al. [1] to ensure that data meet the stringent assumptions of the causal framework that we describe. Applications for CausalARC could include: benchmarking abstract reasoning in LMs, as in Chollet et al. [5]; benchmarking causal inference and causal discovery abilities, as in Jin et al. [17] and Jin et al. [18]; probing impacts of prompt formulation on in-context learning and TTT; benchmarking reasoning with respect to parametric assumptions on the SCM, topological properties of the causal graph, or scaling task complexity; etc.

Contributions

- §3 *The CausalARC testbed.* We introduce an open-ended experimental testbed for AI reasoning at all three levels of the PCH. We provide a static dataset and a public codebase for task generation.¹
- §4 *Empirical demonstrations.* As a proof-of-concept, we illustrate the use of CausalARC for four LM evaluation settings: (1) abstract reasoning with TTT, (2) counterfactual reasoning with in-context learning, (3) program synthesis, and (4) causal discovery with logical reasoning. Preliminary comparisons to ARC-AGI-1 suggest that CausalARC is of similar difficulty.

¹https://anonymous.4open.science/r/causal_arc-E237/. Hugging Face and GitHub provided at camera-ready.



Şekil 2: CausalARC test yatağı. (A) İlk olarak, YNM \mathcal{M} Python koduna manuel olarak dönüştürülür. (B) Giriş-çıkış çiftleri rastgele örneklenerek, dünya modeli hakkında gözlemlisel (L1) öğrenme sinyalleri sağlanır. (C) \mathcal{M} 'nin müdahaleci alt modellerinden \mathcal{M}' örneklemeye yapmak, müdahaleci (L2) örnekler (x', y') üretir. Verilen bir (x, y) çifti için, egzojen bağlamı sabit tutarak birden fazla müdahale gerçekleştirmek, bir dizi karşıolgusal (L3) çift ortaya çıkarır. (D) L1 ve L3 çiftlerini bağlam içi göstergeler olarak kullanarak, çeşitli akıl yürütme görevleri için doğal dil komutlarını otomatik olarak oluşturabiliriz.

tamamen belirlenmiş *nedensel dünya modellerinden* örneklenen akıl yürütme görevleri gibi (Şekil 2). Dünya modelleri, veri üretim süreçlerini temsil etmek ve varsayımsal sonuçları simüle etmek için zengin bir birebirlik olan olasılıksal *yapısal nedensel modeller* (SCM'ler) olarak ifade edilir [32, 33]. Tamamen belirlenmiş bir SCM verildiğinde, PCH'nin her üç seviyesi de iyi tanımlanmıştır: incelenen ortam hakkında herhangi bir gözlemlisel (L1), müdahaleci (L2) veya karşıolgusal (L3) sorgu yanıtlanabilir [2]. Bu formülasyon, CausalARC'ı PCH'nin her üç seviyesindeki akıl yürütme hipotezlerini test etmek için açık uçlu bir platform hâline getirmekte olup; soyut, mantıksal ve karşıolgusal akıl yürütmeye özel bir vurgu yapmaktadır.

Tasarım CausalARC, YZ akıl yürütme değerlendirmesindeki çok sayıda temel sorunu ele almak üzere tasarlanmıştır:

- *Üretken modeller olarak akıl yürütme görevleri.* CausalARC, her görevi, görev örneklerinin rastgele örneklenerek elde edilebildiği üretken bir model (SCM) olarak ele alır. Akıl yürütme sahıflığı, her dünya modelinden gelen görev örnekleri dağılımına göre kıyaslanabilir ve bu sayede belirsizlik tahmini mümkün olur. Dünya modellerinden rastgele örneklemeye yapan üretken kıyaslama testleri, YZ değerlendirmesinde önemli bir sorun olan statik kıyaslama verisi sizintisine karşı da bir tampon görevi görebilir [27, 44, 25, 10].
- *Sinyali gürültüden ayırtırma.* Gerçek akıl yürütme istatistiksel hatırlamadan ayırmak, YZ değerlendirmesinde basit olmayan bir hedefdir [16, 44, 27, 36]. ARC benzeri ortamlar, soyut akıl yürütmenin olgusal bilgiden ziyade doğuştan gelen bilişsel önceliklere dayanması nedeniyle bu hedef için elverişlidir [4].
- PCH'nin tüm seviyelerinde rastgele örneklemeye. Önceki ARC uzantılarının aksine, kullanıcı belirli bir dünya modelinden herhangi bir örnek boyutunda L1, L2 ve L3 dağılımlarını örnekleylebilir. Bu özellik aynı zamanda örnek dağılımı gerektiren nedensel keşif ve nedensel çıkarım sorularını da kolaylaştırır.
- *İnce taneli hata analizleri.* CausalARC, hipotez odaklı model değerlendirmeleri ve ayrıntılı hata analizleri için tasarlanmıştır. ConceptARC [29]'ta olduğu gibi, görevler akıl yürütme temasına (mantıksal, sayıma, genişleştirme ve sıralama) göre etiketlenir. Shojaee vd. [36]'de olduğu gibi, CausalARC çoğu görevi ayarlanabilir dizi boyutlarıyla yapılandırarak problem karmaşıklığını öbeklendirme bağlamında değerlendirmeyi destekler.

Amaçlanan Kullanım Senaryoları CausalARC, çeşitli YZ akıl yürütme hipotezlerini keşfetmek için zengin bir şekilde açıklanmış deneyel bir test yatağıdır. ARC-Heavy [23], ARC-Potpouri [23] ve RE-ARC [14]'ın aksine, CausalARC büyük ölçekli birince ayar veri kümlesi olması amaçlanmamıştır. Verilerin, tanımladığımız nedensel çerçeveyin katı varsayımlarını karşıladığından emin olmak amacıyla Li vd. [23], Hodel [14] ve Akyürek vd. [1] tarafından kullanılan otomatik veri büyütmelerinden kaçınılmazdır. CausalARC için olası uygulamalar şunları içerebilir: Chollet vd. [5]'te olduğu gibi, DM'lerde soyut akıl yürütmenin karşılaştırmalı değerlendirme; Jin vd. [17] ve Jin vd. [18]'de olduğu gibi, nedensel çıkarım ve nedensel keşif yeteneklerinin karşılaştırmalı değerlendirme; istem formülasyonunun bağlam içi öğrenme ve TTT üzerindeki etkilerinin araştırılması; SCM üzerindeki parametrik varsayımlara, nedensel grafiğin topolojik özelliklerine veya ölçümle görevi karşılıklına göre akıl yürütmenin karşılaştırmalı değerlendirme; vb. Katkılar

§3 *CausalARC test yatağı*. PCH'nin her üç seviyesinde de YZ akıl yürütme için açık uçlu bir deneyel test yatağı sunuyoruz. Görev üretimi için statik bir veri kümlesi ve herkese açık bir kod tabanı sağlıyoruz. ¹ §4 *Ampirik gösterimler.* Bir kavram kanıtı olarak, CausalARC'ın dört dil modeli (DM) değerlendirme senaryosu için kullanımını açıklıyoruz: (1) TTT ile soyut akıl yürütme, (2) bağlam içi öğrenme ile karşıolgusal akıl yürütme, (3) program sentezi ve (4) mantıksal akıl yürütme ile nedensel keşif. ARC-AGI-1 ile yapılan ön karşılaşmalar, CausalARC'ın benzer zorluk seviyesinde olduğunu göstermektedir.

¹https://anonymous.4open.science/r/causal_arc-E237/. Hugging Face ve GitHub bağlantıları baskıya hazır olduğunda sağlanacaktır.

2 Preliminaries

We briefly outline preliminaries on causal modeling and the ARC benchmark. Appendix B contains extended preliminaries on world models, intelligence, generalization, and reasoning.

2.1 Structural Causal Models

The reasoning framework presented here draws from the theory of *structural causal models* [7]. Moving forward, we denote random variables with uppercase letters, sets of random variables with bold letters, and realizations in lowercase (e.g., $V = v$, $\mathbf{V} = \mathbf{v}$).

Definition 2.1 (Structural causal model (SCM), Bareinboim et al. [2]). An SCM is a tuple $\mathcal{M} := \langle \mathbf{U}, p(\mathbf{u}), \mathbf{V}, \mathcal{F} \rangle$ where $\mathbf{U} = \{U_i\}_{i=1}^m$ is the set of exogenous variables explained by mechanisms external to \mathcal{M} , $p(\mathbf{u})$ is the distribution over \mathbf{U} , $\mathbf{V} = \{V_i\}_{i=1}^n$ is the set of endogenous variables explained by variables in $\mathbf{U} \cup \mathbf{V}$, and $\mathcal{F} = \{f_i\}_{i=1}^n$ is the set of structural functions such that $v_i = f_i(\mathbf{pa}_{v_i}, \mathbf{u}_i)$ for endogenous parent set \mathbf{pa}_{v_i} and exogenous context \mathbf{u}_i .

The joint factorization of $p(\mathbf{v}, \mathbf{u})$ implied by \mathcal{M} can be represented as a causal graph \mathcal{G} . This work assumes that \mathcal{G} is a directed acyclic graph (DAG), justified by the fact that our testbed does not have a time series component (and thus no feedback loops). Additionally, we make the standard positivity assumption: an SCM is positive if $p(v_i) > 0$ for every realization $V_i \in \mathbf{V} = v_i$. We do not make assumptions about confounding. Future work could explore relaxations of these assumptions.

Crucially, we can exert actions or *interventions* on \mathcal{M} to induce *interventional distributions*. In the real world, intervention corresponds to controlled experimentation.

Definition 2.2 (Hard intervention). A hard intervention $do(V_i = v_i)$ replaces the true causal function $f_i(\mathbf{pa}_{v_i}, u_i)$ with the constant function evaluating to v_i .

Definition 2.3 (Soft intervention). A soft intervention modifies the local conditional distribution $p(v_i | \mathbf{pa}_{v_i})$ to some new distribution $q(v_i | \mathbf{pa}'_{v_i})$, where \mathbf{pa}'_{v_i} may or may not differ from \mathbf{pa}_{v_i} .

Next, we can define *counterfactuals* with respect to \mathcal{M} . In the real world, counterfactuals are not measurable through controlled experimentation as they correspond to alternate hypothetical realities.

Definition 2.4 (Counterfactual, Pearl [33]). Let \mathcal{M}_x be the submodel of \mathcal{M} induced by hard or soft intervention on $X \in \mathbf{V}$. Let $Y \in \mathbf{V}$ be a variable whose value we wish to query. The counterfactual Y_x under model \mathcal{M} is then expressed as $Y_x(\mathbf{u}) := Y_{\mathcal{M}_x}(\mathbf{u})$.

That is, the counterfactual under the original SCM is equal to the value taken by Y under the interventional submodel \mathcal{M}_x , with the exogenous context \mathbf{u} fixed. For hard interventions, $Y_x(\mathbf{u}) = y$ could be verbalized as “ Y would have been y had X been x in context $\mathbf{U} = \mathbf{u}$ ” [33]. Thus, counterfactuals are jointly distributed random variables in a shared probability space over \mathbf{U} [41].

With these definitions of interventions and counterfactuals in hand, we can define a *hierarchy of information* encoded in the distributions induced by the submodels of \mathcal{M} (Figure 1).

Definition 2.5 (Pearl Causal Hierarchy (PCH), Bareinboim et al. [2]). Let \mathcal{M} be a fully specified SCM, per Definition 2.1. The PCH is the set of all observational (layer L1), interventional (layer L2), and counterfactual (layer L3) distributions induced by \mathcal{M} .

Given access to the fully specified \mathcal{M} , the PCH is well-defined: the information encoded in \mathcal{M} enables valuation for any quantity at L1, L2, and L3 [2]. Notably, lower layers of the PCH generally underdetermine higher layers: L_j encodes information sufficient to answer queries at $L_{i \leq j}$, yet knowledge of $L_{i < j}$ is *almost never* sufficient to answer queries at L_j . We refer the reader to the Causal Hierarchy Theorem (CHT) for formal proof, as presented in Bareinboim et al. [2].

2.2 ARC-AGI: The Abstraction & Reasoning Corpus

Background The first Abstraction and Reasoning Corpus (ARC-AGI-1) was introduced in 2019 [4].² At the time of writing, the second ARC challenge is underway³ and the third is in development.

²<https://arcprize.org/arcagi/>

³<https://www.kaggle.com/competitions/arc-prize-2025>

2 Ön Bilgiler

Nedensel modelleme ve ARC kıyaslaması hakkındaki ön bilgileri kısaca özetliyoruz. Ek B, dünya modelleri, zeka, genelleme ve akıl yürütme hakkında genişletilmiş ön bilgileri içermektedir.

2.1 Yapısal Nedensel Modeller

Burada sunulan akıl yürütme çerçevesi, *yapısal nedensel modeller* [7] teorisinden faydalananmaktadır. İlerleyen bölümlerde, rastgele değişkenleri büyük harflerle, rastgele değişken kümelerini kalın harflerle ve gerçekleştirmeleri küçük harflerle (örneğin, $V = v$, $\mathbf{V} = \mathbf{v}$) göstereceğiz.

Tanım 2.1 (Yapısal nedensel model (YTM), Bareinboim vd. [2]). Bir YTM, M 'ye harici mekanizmalarla açıklanan dışsal değişkenler kümesi olan $U = \{U_i\}$ m i = 1, U üzerindeki dağılım olan $p(u)$, $U \cup V$ 'deki değişkenlerle açıklanan içsel değişkenler kümesi olan $V = \{V_i\}$ n i = 1 ve $v_i = f_i(pa_v)$ olacak şekilde yapısal fonksiyonlar kümesi olan $F = \{f_i\}$ n i = 1 şeklinde bir $\mathcal{M} := \langle U, p(u), V, F \rangle$ dörtlüsüdür. i içsel ebeveyn kümesi pa_v için v için egzojen bağlam u_i .

G tarafından ima edilen $p(v, u)$ bileşik çarpanlara ayırma, bir nedensel grafik olan G olarak temsil edilebilir. Bu çalışma, G 'nin yönlendirilmiş çevrimisiz bir grafik (DAG) olduğunu varsayar; bu durum, test yatağımızın bir zaman serisi bileşenine sahip olmaması (ve dolayısıyla geri besleme döngülerinin bulunmaması) gereğilere gerekçelendirmektedir. Ek olarak, standart pozitiflik varsayımini yapmaktadır: bir SCM, her $V_i \in V = v$ i gerçeklemesi için $p(v_i) > 0$ ise pozitiftir. Karıştırıcı faktörler hakkında varsayımlarda bulunmuyoruz. Gelecekteki çalışmalar, bu varsayımların gevestirilmesini araştırabilir.

Çok önemli olarak, M üzerinde eylemler veya müdahaleler uygulayarak *müdahale dağılımları* indükleyebiliriz. Gerçek dünyada müdahale, kontrollü deneylere karşılık gelir.

Tanım 2.2 (Sert müdahale). Sert bir müdahale $do(V_i = v_i)$, gerçek nedensel fonksiyon $f_i(pa_{v_i}, u_i)$ sabit fonksiyon ile v_i olarak değerlendirilir.

Tanım 2.3 (Yumuşak müdahale). Yumuşak bir müdahale, yerel koşullu dağılım $p(v_i | pa_{v_i})$ ögesini yeni bir dağılım $q(v_i | pa'_{v_i})$ olarak değiştirir; burada pa'_{v_i} pa_{v_i} den farklı olabilir veya olmayabilir.

Ardından, M 'ye göre karşıolgusallar tanımlayabiliriz. Gerçek dünyada, karşıolgusallar alternatif hipotetik gerçekliklere karşılık geldiği için kontrollü deneyler yoluya ölçülemezler.

Tanım 2.4 (Karşıolgusal, Pearl[33]). $X \in V$ üzerindeki sert veya yumuşak müdahale ile indüklenen M 'nin alt modeli M_x olmak üzere. Değerini sorgulamak istediğimiz bir değişken $Y \in V$ olmak üzere. M modeli altındaki karşıolgusal Y_x daha sonra $Y_x(u) := Y_{M_x}(u)$ olarak ifade edilir.

Yani, orijinal YKN altındaki karşıolgusal, Y 'nin müdahaleli alt model M_x altında aldığı değere eşittir ; egzojen bağlam u sabittir. Zor müdahaleler için, $Y_x(u) = y$ ifadesi, “ $U = u$ bağlamında X , x olsaydı Y , y olurdu” şeklinde ifade edilebilir [33]. Dolayısıyla, karşıolgusallar, U üzerinde ortak bir olaşılık uzayında birlikte dağılmış rastgele değişkenlerdir [41].

Müdahaleler ve karşıolgusalların bu tanımlarıyla, M 'nin alt modelleri tarafından indüklenen dağılımlarda kodlanmış bir bilgi hiyerarşisi tanımlayabiliriz (Şekil 1).

Tanım 2.5 (Pearl Nedensel Hiyerarşisi (PNH), Bareinboim ve diğ. [2]). Tanım 2.1'e göre M , tam olarak belirlenmiş bir YKN olsun. PNH, M tarafından indüklenen tüm gözlemsel (katman L1), müdahaleli (katman L2) ve karşıolgusal (katman L3) dağılımlarının kümeleridir.

Tamamen belirtilmiş M 'ye erişim sağlandığında, PCH iyi tanımlanmıştır: M 'de kodlanan bilgi, L1, L2 ve L3 'teki herhangi bir niceliğin değerlendirilmesini sağlar [2]. Özellikle, PCH'nin alt katmanları genellikle üst katmanları eksik belirler: L_j , $L_i \leq j$ 'deki sorular yanıtlanmak için yeterli bilgiyi kodlarken, $L_i < j$ b bilgisi L_j 'deki sorular yanıtlanmak için neredeyse hiçbir zaman yeterli değildir. Okuyucuya, Bareinboim ve diğerleri [2] tarafından sunulan resmi kanıt için Nedensel Hiyerarşî Teoremi'ne (CHT) yönlendiriliyoruz.

2.2 ARC-AGI: Soyutlama ve Akıl Yürütme Külliyatı

Arka Plan İlk Soyutlama ve Akıl Yürütme Külliyatı (ARC-AGI-1) 2019'da tanıtıldı.

[4].² Bu yazının yazıldığı sırada, ikinci ARC yarışması devam etmekte³ ve üçüncüsü geliştirilme aşamasındadır.

²<https://arcprize.org/arcagi/>

³<https://www.kaggle.com/competitions/arc-prize-2025>

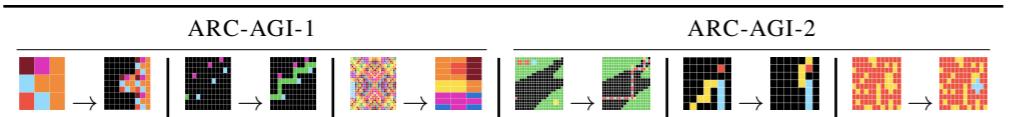


Figure 3: Example input-output pairs from ARC-AGI-1 and ARC-AGI-2.

ARC is a grid world, represented by two-dimensional arrays of pixels that can take on one of ten colors each. For ARC-AGI-1, data consist of a training set ($n = 400$ reasoning tasks), a public evaluation set ($n = 400$), a semi-private evaluation set ($n = 100$), and a fully private evaluation set ($n = 100$). Each “grid” is an array of any dimensionality from 1×1 to 30×30 (Figure 3). Arrays vary widely in appearance, at times appearing as randomly dispersed colors. Arrays can also feature distinct multi-pixel shapes, referred to as *sprites* (as in computer graphics).

The test-taker must solve each task by discovering the deterministic rule or transformation that maps input arrays to output arrays. Each task provides approximately 2–5 input-output pairs as examples to demonstrate the rule, with no additional clues provided. An average human should ostensibly be able to solve most or all tasks from these demonstrations alone, with no specialized knowledge or training. Instead, problem-solving requires innate cognitive priors, such as elementary arithmetic, basic geometry, and intuitive physics [4].

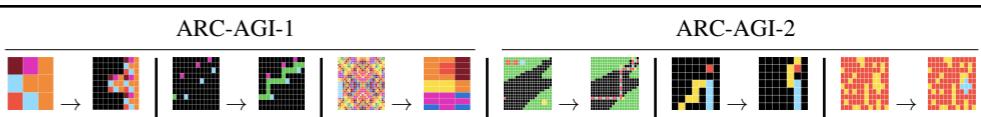
Notation Let $\mathcal{D} := \mathcal{D}_{train} \cup \mathcal{D}_{eval}$ denote the ARC dataset. As we are concerned with in-context learning and TTT but not pretraining or fine-tuning, we consider only \mathcal{D}_{eval} moving forward. Each instance in \mathcal{D}_{eval} is a reasoning task \mathbf{T}_i , such that $\mathcal{D}_{eval} := \{\mathbf{T}_i\}_{i=1}^n$. Each \mathbf{T}_i is an m -shot learning task associated with its own input space \mathcal{X} and output space \mathcal{Y} . As such, \mathbf{T}_i is comprised of $m + 1$ tuples of paired input and output arrays, where m tuples are *demonstration pairs* $(\mathbf{x}_{train}, \mathbf{y}_{train}) \in \mathcal{X}^m \times \mathcal{Y}^m$ and one is a *test pair* $(\mathbf{x}_{test}, \mathbf{y}_{test}) \in \mathcal{X} \times \mathcal{Y}$. Thus, $\mathbf{T}_i := \{(\mathbf{x}_{train}, \mathbf{y}_{train})\} \cup \{(\mathbf{x}_{test}, \mathbf{y}_{test})\}$. When sufficiently clear, we use (\mathbf{x}, \mathbf{y}) to denote a single random input-output pair. Each \mathbf{T}_i is governed by a deterministic *rule* or *transformation* $\delta_i : \mathcal{X} \rightarrow \mathcal{Y}$, which maps each input array to its respective output. Let \mathcal{A} be our test-taker (e.g., an LM). To succeed on \mathbf{T}_i , \mathcal{A} must learn δ_i sufficiently well to correctly predict \mathbf{y}_{test} from \mathbf{x}_{test} (with predictions denoted as $\hat{\mathbf{y}}_{test}$). \mathcal{D}_{eval} is presented to \mathcal{A} as string representations of Python arrays. As in Akyürek et al. [1], we encode colors as integers: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9].

3 CausalARC: Abstract Reasoning with Causal World Models

3.1 ARC Through a Causal Lens

To lay the groundwork for CausalARC, we first outline a causal interpretation of the original ARC dataset and introduce the assumptions underlying our design choices.

SCMs: Causal World Models, Causal Programs As in ARC, CausalARC features high task diversity: each task instance is sampled from a unique world model. In program synthesis approaches to ARC, world models are specified as programs [23]. The present work assumes a *causal world model* describing the data generating process, which we define in the formal language of SCMs. As the PCH is well-defined given a fully specified SCM [2], this SCM offers a highly information-rich world model upon which we can perform symbolic operations. Classically, we can express SCMs in mathematical notation (e.g., Example 3.1). Alternatively, we can frame SCMs as *causal programs* that are implemented in a programming language. Under this framing, the act of sampling corresponds to executing the program with different random seeds, intervening corresponds to principled alterations of the program’s logic, and counterfactuals correspond to performing multiple interventions on the same random seed (i.e., the exogenous context is held constant, as required by Definition 2.4). The choice to model reasoning tasks as generative causal world models borrows intuition from prior work in reinforcement learning (RL), where causal world models are expressed as Bayesian networks or SCMs [34]. Formally, Richens and Everitt [34] prove that “any agent capable of adapting to a sufficiently large set of distributional shifts must have learned a causal model of the data generating process.” Though we do not explore RL, we follow the intuition that learning new causal worlds model on the fly facilitates generalization under distribution shift.



Şekil 3: ARC-AGI-1 ve ARC-AGI-2'den örnek girdi-çıktı çiftleri.

ARC, her biri on renkten birini alabilen iki boyutlu piksel dizileriyle temsil edilen bir izgara dünyasıdır. ARC-AGI-1 için veriler; bir eğitim kümesi ($n = 400$ akıl yürütme görevi), bir genel değerlendirme kümesi ($n = 400$), yarı özel bir değerlendirme kümesi ($n = 100$) ve tamamen özel bir değerlendirme kümesinden ($n = 100$) oluşur. Her bir “izgara”, 1×1 ile 30×30 arasında herhangi bir boyuta sahip bir dizidir (Şekil 3). Diziler görünüş olarak büyük farklılıklar gösterir, bazen rastgele dağılmış renkler olarak karşımıza çıkar. Diziler ayrıca, bilgisayar grafiklerindeki gibi, belirgin çok piksellî şekillere, yani sprite'lara sahip olabilir.

Testi çözen kişi, girdi dizilerini çıktı dizilerine eşleyen deterministik kuralı veya dönüşümü keşfederken her görevi çözmeli. Her görev, kuralı göstermek için yaklaşık 2-5 girdi-çıktı çiftini örnek olarak sunar ve ek ipucu verilmez. Ortalama bir insan, bu gösterimlerden yola çıkarak, özel bir bilgi veya eğitim olmaksızın, görevlerin çoğunu veya tamamını çözebilir. Bunun yerine, problem çözme, temel aritmetik, temel geometri ve sezgisel fizik gibi doğuştan gelen bilişsel öncüleri gerektirir [4].

Gösterim $D := D_{train} \cup D_{eval}$ ARC veri kümesini belirtir. Bağlam içi öğrenme ve TTT ile ilgili olduğumuz, ancak ön eğitim veyaince ayar ile ilgilenmediğimiz için, yalnızca D_{eval} üzerinden ilerliyoruz. D_{eval} içindeki her örnek, bir akıl yürütme görevi olan T_i ’dir, öyle ki $D_{eval} := \{T_i\}_{i=1}^n$. Her T_i , kendi giriş uzayı X ve çıkış uzayı Y ile ilişkili, m -atılı bir öğrenme görevidir. Bu bağlamda, $T_i : m+1$ çiftlenmiş giriş ve çıkış dizisi demetinden oluşur; burada m demet, gösterim çiftleri $(x_{\text{eğitim}}, y_{\text{eğitim}}) \in X^{m+1} \times Y^m$ iken, bir tanesi bir test çifti $(x_{\text{test}}, y_{\text{test}}) \in X \times Y$ dir. Dolayısıyla, $T_i := \{(x_{\text{eğitim}}, y_{\text{eğitim}})\} \cup \{(x_{\text{test}}, y_{\text{test}})\}$. Yeterince açık olduğunda, tek bir rastgele giriş-çıkış çiftini belirtmek için (x, y) kullanırız. Her T_i , her giriş dizisini ilgili çıkışına eşleyen, deterministik bir kural veya dönüşüm $\delta_i : X \rightarrow Y$ tarafından yönetilir. A ’nın (örn. bir DM) test edilenimiz olduğunu varsayılmı. T_i üzerinde başarılı olmak için, A ’nın x_{test} ’ten y_{test} ’i doğru bir şekilde tahmin etmek için δ_i ’yi yeterince iyi öğrenmesi gereklidir (tahminler y test olarak gösterilir). D_{eval} , A ’ya Python dizilerinin dize temsilleri olarak sunulur. Akyürek ve ark. [1]’de olduğu gibi, renkleri tam sayılar olarak kodları: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9].

3 CausalARC: Nedensel Dünya Modeleriyle Soyut Akıl Yürütme

3.1 Nedensel Mercek Aracılığıyla ARC

CausalARC için zemin hazırlamak amacıyla, öncelikle orijinal ARC veri kümesinin nedensel bir yorumunu ana hatlarıyla belirleyeceğiz ve tasarım seçimlerimizin altında yatan varsayımları tanıtabağız.

SCM’ler: Nedensel Dünya Modeleri, Nedensel Programlar ARC’de olduğu gibi, CausalARC de yüksek görev çeşitliliğine sahiptir: her görev örneği benzersiz bir dünya modelinden örneklenir. ARC’ye yönelik program sentezi yaklaşımında, dünya modelleri programlar olarak belirtilir [23]. Bu çalışma, veri üretim sürecini tanımlayan bir nedensel dünya modeli varsayılar ve bunu SCM’lerin resmi dilinde tanımlar. PCH, tam olarak belirtilmiş bir SCM verildiğinde iyi tanımlandığı için [2], bu SCM, üzerinde sembolik işlemler gerçekleştirebileceğimiz son derece bilgi açısından zengin bir dünya modeli sunar. Klasik olarak, YNM’leri matematiksel gösterimle (örn. Örnek 3.1) ifade edebiliriz. Alternatif olarak, YNM’leri bir programlama dilinde uygulanan nedensel programlar olarak çerçeveyebiliriz. Bu çerçeve altında, örnekleme eylemi, programı farklı rastgele tohumlarla çalıştırma karşılık gelirken; müdahale, programın mantığında prensipi değişiklikler yapmaya karşılık gelir; ve karşılgusallar, aynı rastgele tohum üzerinde bir den fazla müdahale gerçekleştirmeye karşılık gelir (yani, Tanim 2.4’ün gerektirdiği gibi egzojen bağlam sabit tutulur). Akıl yürütme görevlerini üretken nedensel dünya modelleri olarak modelleme seçimi, nedensel dünya modellerinin Bayes ağları veya YNM’ler [34] olarak ifade edildiği pekiştirmeli öğrenme (RL) alanındaki önceki çalışmalarдан sezgi ödüne alır. Resmi olarak, Richens ve Everitt [34], “yeterince büyük bir dağılımsal kayma kümese uyum sağlayabilen herhangi bir ajanın, veri üretim sürecinin nedensel bir modelini öğrenmiş olması gerektiğini” kanıtlar. RL’yi keşfetmesek de, yeni nedensel dünya modellerini anında öğrenmenin dağılım kayması altında genellemeyi kolaylaştırdığı sevgisini takip ediyoruz.

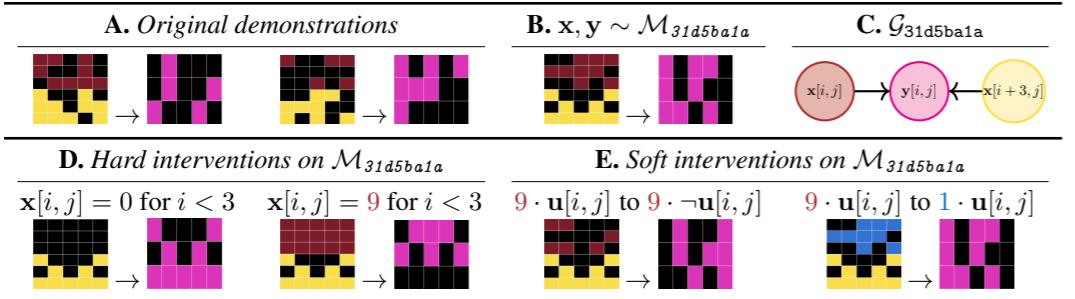


Figure 4: Input-output arrays for ARC-AGI-1 task 31d5ba1a [5]. (A) A subset of the official demonstration pairs ($\mathbf{x}_{train}, \mathbf{y}_{train}$). (B) A random sample from SCM $M_{31d5ba1a}$ defined in Example 3.1. (C) Causal DAG $G_{31d5ba1a}$ representing $M_{31d5ba1a}$, where $y[i, j] = 6 \cdot \text{xor}(\mathbf{x}[i, j], \mathbf{x}[i + 3, j])$ for $i \in [0, 2], j \in [0, 4]$. (D-E) Samples from interventional submodels of $M_{31d5ba1a}$, where exogenous variables are held constant and the causal effects of interventions propagate to \mathbf{y} .

Representing Random Variables At the highest level of abstraction, it is clear that \mathbf{x} causes \mathbf{y} . We can denote this graphically as $(\mathbf{x}) \rightarrow (\mathbf{y})$, where (\mathbf{x}) and (\mathbf{y}) are multivariate “supernodes” in the causal DAG. At finer granularity, we could choose to model each exogenous variable as Bernoulli (Example 3.1) or as a categorical random variable that can take on values 0 through 9 (Example C.1). Arbitrary causal functions could result in endogenous random variables with complex distributions. Additionally, an array can be described by grid-level features over all or some elements (e.g., the total number of pixels taking a specific value, the total number of specific sprites, etc.). These can also be modeled as random variables, as in Example C.1. Note that all stochasticity in our model arises from the random sampling of our exogenous variables, and we restrict our attention to deterministic causal functions for endogenous variables. Any \mathbf{x} maps to only one \mathbf{y} , just as in the original ARC.

Grids as Samples from Latent SCMs We assume that an underlying SCM M_i exists for each T_i . Thus, we assume that every array in T_i is sampled from the SCM (denoted $T_i \sim M_i$). In some original ARC tasks, a unique M_i can be easily derived from the limited examples provided by $(\mathbf{x}_{train}, \mathbf{y}_{train})$. In other cases, multiple SCMs might be compatible with the limited information provided by $(\mathbf{x}_{train}, \mathbf{y}_{train})$, yielding some form of equivalence class. This latter case is analogous to the causal discovery setting where the full unique DAG is not identifiable, though its Markov equivalence class is [39]. To support the assumption that $T_i \sim M_i$, we consider cases where an SCM is easily recoverable from observing $(\mathbf{x}_{train}, \mathbf{y}_{train})$ (Examples 3.1, C.1). Samples from the recovered SCMs are indistinguishable from those provided in the official task (Figures 4, C.1).

Example 3.1 (A fully recovered SCM). Consider ARC-AGI-1 task 31d5ba1a (expert level [21, 22]; Figure 4). We can define an SCM $M_{31d5ba1a} = \langle \mathbf{U}, p(\mathbf{u}), \mathbf{V}, \mathcal{F} \rangle$ where \mathbf{V} are the elements of our arrays, $f \in \mathcal{F}$ are logical xor⁴ and/or scalar multiplication, and $p(\mathbf{u})$ is Bernoulli.

$$\mathbf{u}[i, j] \sim \text{Ber}(0.5) \quad \text{for } i \in [0, 5], j \in [0, 4] \quad (1)$$

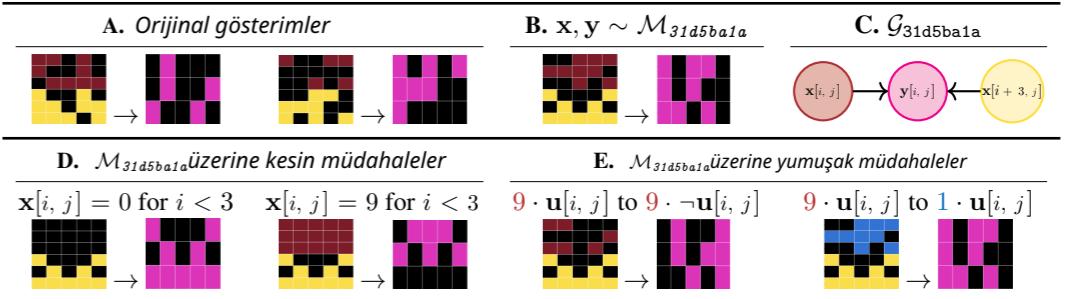
$$\mathbf{x}[i, j] = \begin{cases} 9 \cdot \mathbf{u}[i, j] & \text{if } i < 3 \\ 4 \cdot \mathbf{u}[i, j] & \text{else} \end{cases} \quad \text{for } i \in [0, 5], j \in [0, 4] \quad (2)$$

$$\mathbf{y}[i, j] = 6 \cdot \text{xor}(\mathbf{x}[i, j], \mathbf{x}[i + 3, j]) \quad \text{for } i \in [0, 2], j \in [0, 4]. \quad (3)$$

With our formally defined SCM, we can perform hard or soft interventions and observe the impacts on output \mathbf{y} . When we hold $\mathbf{U} = \mathbf{u}$ constant, we claim the resulting arrays are *counterfactuals* (Definition 2.4; Figure 5). As shown in Figure 4D, hard interventions on \mathbf{x} impact the color distribution over \mathbf{y} . In Figure 4E, we observe that soft interventions on the causal function for $\mathbf{x}[i, j]$ may or may not impact the output \mathbf{y} . As the color of $\mathbf{y}[i, j]$ is impacted by whether $\mathbf{x}[i, j]$ is zero or nonzero but not by the specific nonzero value taken, interventions performed in Figure 4E (right) do not influence \mathbf{y} .

Rules as Observable Proxies for SCMs In practice, the SCM describing a random ARC task will be too complex to easily derive by hand from limited examples. Thus, we proceed under the assumption that the true SCM M_i explaining T_i is unknown. We can choose to model rule δ_i as an

⁴Our notation follows the numpy implementation of logical operators, where any nonzero value is treated as the Boolean 1 = TRUE while zero is treated as FALSE. All logical operators return integer values in [0, 1].



Şekil 4: ARC-AGI-1 görevi 31d5ba1a [5] için giriş-çıkış dizileri. (A) Resmi gösterim çiftlerinin bir alt kümeleri (x_{train}, y_{train}). (B) Örnek 3.1'de tanımlanan SCM $M_{31d5ba1a}$ 'dan rastgele bir örnek. (C) $M_{31d5ba1a}$ ’yi temsil eden nedensel DAG $G_{31d5ba1a}$, burada $y[i, j] = 6 \cdot \text{xor}(x[i, j], x[i + 3, j])$ for $i \in [0, 2], j \in [0, 4]$. (D-E) $M_{31d5ba1a}$ 'nın müdahale alt modellerinden örnekler; burada dışsal değişkenler sabit tutulur ve müdahalelerin nedensel etkileri y ’ye yayılır.

Rastgele Değişkenleri Temsil Etme En yüksek soyutlama düzeyinde, x ’in y ’ye neden olduğu açıkta. Bunu grafiksel olarak $x \rightarrow y$ şeklinde gösterebiliriz; burada x ve y , nedensel DAG'deki çok değişkenli “süper düğümlerdir”. Daha ince bir ayrıntıda, her dışsal değişkeni Bernoulli olarak (Örnek 3.1) veya 0’dan 9'a kadar değerler alabilen kategorik bir rastgele değişken olarak modellemeyi sebebiliyor (Örnek C.1). Keyfi nedensel fonksiyonlar, karmaşık dağılımlara sahip rastgele değişkenlerle sonuçlanabilir. Ek olarak, bir dizi tüm veya bazı elemanlar üzerinden izgara seviyesi özelliklerle (örn. belirli bir değer alan toplam piksel sayısı, belirli sprite’ların toplam sayısı vb.) tanımlanabilir. Bunlar da Örnek C.1'deki gibi rastgele değişkenler olarak modellenebilir. Modelimizdeki tüm stokastisitenin dışsal değişkenlerimizin rastgele örnekleşmesinden kaynaklandığını ve içsel değişkenler için deterministik nedensel fonksiyonlara dikkatimi sınırladığımızı unutmayın. Tipki orijinal ARC’de olduğu gibi, herhangi bir x sadece bir y eşlenir.

Gizli SCM’lerden Örnekler Olarak Izgaralar Her T_i için temel bir SCM M_i ’nin mevcut olduğunu varsayıyoruz. Böylece, T_i içindeki her dizinin SCM’den örnekleşmiş olduğunu varsayıyoruz ($T_i \sim M_i$ olarak belirttilir). Bazı orijinal ARC görevlerinde, (x_{train}, y_{train}) tarafından sağlanan sınırlı örneklerden bensersiz bir M_i kolayca türetilebilir. Diğer durumlarda, birden fazla SCM, (x_{train}, y_{train}) tarafından sağlanan sınırlı bilgilerle uyumlu olabilir ve bir tür eşdeğerlik sınıfı oluşturabilir. Bu son durum, tam bensersiz DAG’ın tanımlanamadığı, ancak Markov eşdeğerlik sınıfının tanımlanabildiği nedensel keşif ortamına benzer [39]. $T_i \sim M_i$ ’nin varlığını desteklemek için, (x_{train}, y_{train}) gözlemlenmesinden bir SCM’nin kolayca geri kazanılabilir olduğu durumları ele alıyoruz (Örnekler 3.1, C.1). Geri kazanılan SCM’lerden alınan örnekler, resmi görevde sağlananlardan ayırt edilemez (Şekiller 4, C.1).

Örnek 3.1 (Tamamen kurtarılmış bir SCM). ARC-AGI-1 görevi 31d5ba1a (uzman seviyesi [21, 22]; Şekil 4) ele alalım. Bir SCM $M_{31d5ba1a} = \langle \mathbf{U}, p(\mathbf{u}), \mathbf{V}, \mathcal{F} \rangle$ tanımlayabiliriz; burada \mathbf{V} dizilerimizin elemanları, $f \in \mathcal{F}$ ise mantıksal xor⁴ ve/veya skaler çarpımıdır ve $p(\mathbf{u})$ Bernoulli dağılımlıdır. $u[i, j] \sim \text{Ber}(0.5)$

$$i \in [0, 5], j \in [0, 4] \text{ için} \quad (1)$$

$$\mathbf{x}[i, j] = \begin{cases} 9 \cdot \mathbf{u}[i, j] & \text{eğer } i < 3 \\ 4 \cdot \mathbf{u}[i, j] & \text{aksı takdirde} \end{cases} \quad i \in [0, 5], j \in [0, 4] \text{ için} \quad (2)$$

$$\mathbf{y}[i, j] = 6 \cdot \text{xor}(\mathbf{x}[i, j], \mathbf{x}[i + 3, j]) \quad i \in [0, 2] \text{ ve } j \in [0, 4] \text{ için}. \quad (3)$$

Resmi olarak tanımlanmış SCM’izle, sert veya yumuşak müdahaleler gerçekleştirilebilir ve bunların çıkış y üzerindeki etkilerini gözlemeylebiliriz. $U = u$ sabit tutulduğunda, ortaya çıkan dizilerin karşılıklı olduğunu iddia ediyoruz (Tanim 2.4; Şekil 5). Şekil 4D’de gösterildiği gibi, x üzerindeki sert müdahalelerin üzerindeki renk dağılımını etkiler.

Şekil 4E’de, $x[i, j]$ için nedensel fonksiyona yapılan yumuşak müdahalelerin çıkış y ’yi etkileyip etkilemediğini gözlemliyoruz. $y[i, j]$ ’nin rengi, $x[i, j]$ sıfır veya sıfır olmayan bir değer olup olmadığına göre değişti. ancak alınan belirli sıfır olmayan değere göre değişmediği için, Şekil 4E’de (sağda) gerçekleştirilen müdahaleler y ’yi etkilemez.

SCM’ler İçin Gözlemlenebilir Vekiller Olarak Kurallar Pratikte, rastgele bir ARC görevini tanımlayan SCM, sınırlı örneklerden elle kolayca türetilemeyecek kadar karmaşık olacaktır. Dolayısıyla, T_i ’yi açıklayan gerçek SCM M_i ’nin bilinmediği varsayıyı altında iterliyoruz. Kural δ_i ’yi şu şekilde modellemeyi sebebiliyor:

⁴Özeti, sıfır olmayan herhangi bir değerin Boole 1 = TRUE kabul edildiği, sıfır ise FALSE kabul edildiği mantıksal operatörlerin numpy uygulamasını takip eder. Tüm mantıksal operatörler [0, 1] aralığında tam sayı değerleri döndürür.

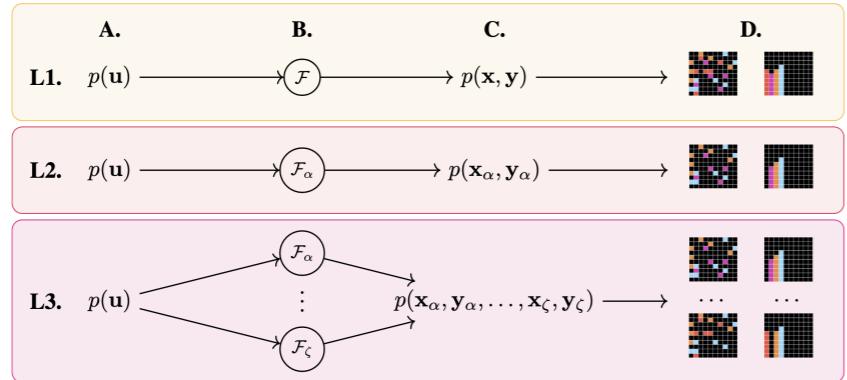


Figure 5: Jointly observed counterfactuals in CausalARC. L1, L2, and L3 denote the rungs of the PCH (Figure 1). (A) The distribution over the exogenous context (i.e., the external state). (B) Transformations applied to the exogenous context (e.g., functions \mathcal{F} in the observational world; updated functions \mathcal{F}_α under intervention α). (C) Induced distributions, following from the applied transformation. (D) CausalARC samples from each rung of the PCH. Adapted from [2] (Figure 27.2).

observable and human-interpretable proxy for the latent \mathcal{M}_i . In practice, however, δ_i is likely to be a lossy distillation of \mathcal{M}_i : often, \mathcal{A} (human or machine) only learns as much about \mathcal{M}_i as is needed to output \hat{y}_{test} (for example, parameter p to the Bernoulli distribution over the exogenous variables is not generally needed to correctly estimate \hat{y}_{test}). However, without access to the data generating process provided by \mathcal{M}_i , we cannot randomly sample from the true distribution nor guarantee that data augmentations yield true counterfactuals. We introduce CausalARC to address these limitations.

3.2 Constructing CausalARC

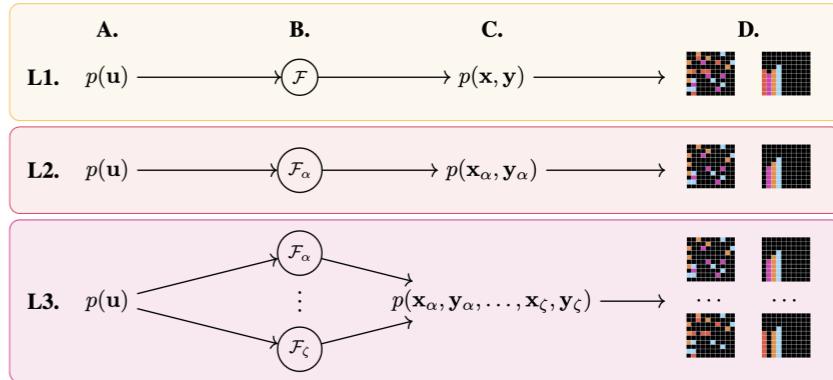
Task Generation All tasks were constructed in Python using built-in functions and numpy. The task construction procedure was as follows. First, an SCM was defined in mathematical notation. This SCM was then manually translated to Python, with functionality for performing hard and/or soft interventions and obtaining sample distributions. Task instances sampled from SCMs are returned as dictionaries in a format consistent with the official ARC dataset⁵ to ensure compatibility with existing ARC pipelines. To date, the static CausalARC dataset is composed of 50 task instances, each with five demonstration pairs, one test pair, and 5–10 counterfactual pairs per demonstration pair.⁶

Causal Annotations In addition to counterfactual examples, each task dictionary is annotated by a human expert with at least one representation of the causal world model: (1) in all cases, a string representation of the Python program is included; (2) in a subset of tasks, the mathematical notation for the SCM is provided; and (3) in another (non-disjoint) subset, the adjacency matrix of the causal graph is provided. Representation (1) is potentially useful for program synthesis settings (e.g., for supervising induction models, as in [23]), while (3) is especially useful for causal discovery use cases.

Task Themes To date, tasks in CausalARC fall into the following human-labeled categories:

1. *Counting*. Tasks require elements to be counted (e.g., total pixels of a certain color).
2. *Extension*. Paths or sprites must be extended according to some rule.
3. *Logical*. Tasks require reasoning over causal functions that are logical operators.
4. *Ordering*. Elements must be ordered according to some rule, colored according to some order, etc.

In the static dataset, each theme currently features 10 task instances (with the exception of logical tasks, for which there are 20). As with the concept labels in ConceptARC [29] and the difficulty labels provided by LeGris et al. [21, 22], task labels allow the user to do fine-grained error analyses. For example, one could compare the performance of test-taker \mathcal{A} on logical tasks when causal functions are *xor* versus *and*, or tasks requiring counting versus ordering, etc.



Şekil 5: CausalARC'de birlikte gözlemlenen karşıolgusallar. L1, L2 ve L3, PCH'nin basamaklarını gösterir (Şekil 1). (A) Egzojen bağlam üzerindeki dağılım (yani, dış durum). (B) Egzojen bağlama uygulanan dönüştürüler (örn. gözlemeşel dünyadaki \mathcal{F} fonksiyonları; müdahale α altında güncellenen \mathcal{F}_α fonksiyonları). (C) Uygulanan dönüşümünden kaynaklanan indüklenmiş dağılımlar. (D) CausalARC, PCH'nin her basamağından örnekler alır. [2] (Şekil 27.2)'den uyarlanmıştır.

gizli \mathcal{M}_i için gözlemlenebilir ve insan tarafından yorumlanabilir vekil. Ancak pratikte, δ_i , \mathcal{M}_i 'nin kayıplı bir damıtımı olma olasılığı yüksektir: genellikle, \mathcal{A} (insan veya makine) \mathcal{M}_i hakkında yalnızca \hat{y}_{test} çıktısı vermek için gereken kadarını öğrenir (örneğin, dışsal değişkenler üzerindeki Bernoulli dağılımına ilişkin p parametresi, \hat{y}_{test} 'i doğru bir şekilde tahmin etmek için genellikle gerekli değildir). Ancak, \mathcal{M}_i tarafından sağlanan veri üretim sürecine erişim olmadan, ne gerçek dağılımdan rastgele örnekleyebilir ne de veri artırımlarının gerçek karşıolgusallar ürettiğini garanti edebiliriz. Bu sınırlamaları gidermek için CausalARC'ı tanıtıyoruz.

3.2 CausalARC'ın İnşası

Görev Üretimi Tüm görevler Python'da yerleşik fonksiyonlar ve numpy kullanılarak oluşturulmuştur. Görev oluşturma prosedürü şu şekildeydi. Öncelikle, matematiksel gösterimde bir SCM tanımlanırdı. Bu SCM daha sonra Python'a manuel olarak çevrildi; bu çeviri, sert ve/veya yumuşak müdahaleler gerçekleştirme ve örnek dağılımları elde etme işlevselliliğini içeriyyordu. SCM'lerden örneklenen görev örnekleri, mevcut ARC işlem hatlarıyla uyumluluğu sağlamak amacıyla resmi ARC veri kümesi⁵ ile tutarlı bir formatta sözlükler olarak döndürülür. Bugüne kadar, statik CausalARC veri kümesi, her biri beş gösterim çifti, bir test çifti ve her gösterim çifti için 5–10 karşıolgusal çift içeren 50 görev örneğinden oluşmaktadır.

dır.⁶ Nedensel Ek Açıklamalar Karşıolgusal örnekler ek olarak, her görev sözlüğü bir insan uzmanı tarafından nedensel dünya modelinin en az bir temsiliyle açıklanır: (1) tüm durumlarda, Python programının bir dize gösterimi dâhil edilmiştir; (2) görevlerin bir alt kümesinde, SCM için matematiksel gösterim sağlanır; ve (3) başka bir (ayrışık olmayan) alt kümede, nedensel grafiğin komşuluk matrisi sağlanır. Temsil (1), program sentezi ayarları için potansiyel olarak kullanışlıdır (örn. [23]'te olduğu gibi induksiyon modellerini denetlemek için), (3) ise özellikle nedensel keşif kullanım durumları için faydalıdır.

Görev Temaları Bugüne kadar, CausalARC'taki görevler aşağıdaki insan tarafından etiketlenmiş kategorilere ayrılmaktadır:

1. Sayma. Görevler, öğelerin sayılmasını gerektirir (örn. belirli bir renkteki toplam piksel sayısı).
2. Uzatma. Yollar veya sprite'ler belirli bir kurala göre uzatılmalıdır.
3. Mantıksal. Görevler, mantıksal operatörler olan nedensel fonksiyonlar üzerinde akıl yürütmemi gerektirir.
4. Sıralama. Öğeler belirli bir kurala göre sıralanmalı, belirli bir sıraya göre renklendirilmeli vb.

Statik veri kümesinde, her tema şu anda 10 görev örneği içerir (mantıksal görevler için 20 adet hariç). ConceptARC [29]'deki kavram etiketleri ve LeGris et al. [21, 22] tarafından sağlanan zorluk etiketleri gibi, görev etiketleri kullanıcının ayrıntılı hata analizleri yapmasına olanak tanır. Örneğin, nedensel fonksiyonlar xor iken ve iken, veya sayma ya da sıralama gerektiren görevlerde, bir test katılımcısı A'nın mantıksal görevlerdeki performansı karşılaştırılabilir.

⁵<https://www.kaggle.com/competitions/arc-prize-2024>

⁶Hugging Face link at camera-ready.

⁵<https://www.kaggle.com/competitions/arc-prize-2024>

⁶Kamera-hazır olduğunda Hugging Face bağlantısı.

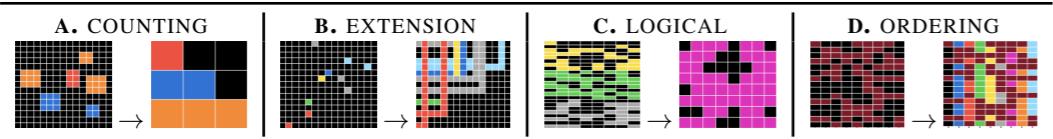


Figure 6: Example demonstration pairs for CausalARC themes.

Jointly Observed Counterfactuals The fundamental problem of causal inference is that counterfactuals cannot be jointly observed in the physical universe. In our synthetic grid world, we can exploit the fact that we have full control over the data generating process and exogenous factors can be held constant across multiple interventions. Thus, this synthetic environment allows for the juxtaposition of ‘treated’ samples under multiple interventions alongside their ‘untreated’ counterparts, without violating the assumption that $\mathbf{U} = \mathbf{u}$ in all instances. In this way, we justify our choice to model the data augmentations on T_i as a set of jointly observed counterfactuals (Figure 5).

Prompt Sampling CausalARC provides functionality for randomly sampling prompts from the underlying SCM according to user specifications. Currently, the following aspects can vary: total number of in-context demonstrations, type of in-context demonstrations (all L1 versus alternating L1/L3), and query theme (counterfactual reasoning, abstract reasoning, program synthesis, and causal discovery). For counterfactual reasoning, the LM is prompted to predict the output under intervention for a previously seen L1 example (e.g., Figures D.4, D.5). For abstract reasoning, the LM is prompted to predict the output array for an L1 input, as in conventional ARC setups [1]. For program synthesis, the LM is prompted to generate a Python program that expresses the SCM (e.g., Figures D.6, D.7). For causal discovery, the LM is prompted to predict a property of the underlying SCM, such as the causal parents of an array element or the form of a causal function (e.g., Figures D.8, D.9).

4 Empirical Demonstrations

As a proof-of-concept, we illustrate four potential use cases for CausalARC: (1) abstract reasoning with TTT, (2) counterfactual reasoning with in-context learning, (3) program synthesis, and (4) causal discovery with logical reasoning. Full experimental details and results are in Appendix D, including a full description of the tasks used for each experiment.

Models Models were selected for diversity. We compare designated reasoning models (e.g., o4-mini) versus vanilla LMs (e.g., GPT-4o mini); older, less powerful models (e.g., Claude Haiku 3.5) versus newer, more powerful models (e.g., Claude Sonnet 4); and open-source models (e.g., LLama 3 8B, Llama 4 Scout 17B) versus closed-source models (all others). Full model details are in Table D.1.

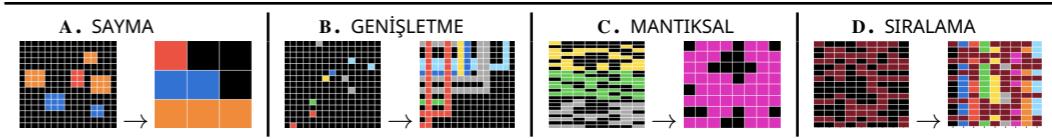
Prompt Formulation We explored impacts of prompt formulation with respect to PCH level and total number of in-context examples. To isolate the effects of PCH level on LM performance, we compared prompts with L1 versus alternating L1/L3 in-context demonstrations while holding the total number of demonstrations constant (e.g., Figure D.6 versus D.7).

Metrics As in Akyürek et al. [1], LM performance was measured by output accuracy (i.e., whether the output array was exactly correct) and relative Hamming distance (HD; i.e., the number of positions where the true array differs from the predicted array, normalized by the number of elements in the true array). For HD, a score of 0 indicates correctness while a score of 1 indicates complete failure (including cases where the LM returns a malformed array or no response at all).

4.1 Abstract Reasoning with Test-Time Training

Motivation To gauge the difficulty of CausalARC, we benchmarked MARC with TTT [1] on the full static dataset. MARC was the second-place paper winner for ARC-AGI-1. It takes a neural transduction approach using a LLama 3 8B base model fine-tuned on large ARC-like datasets, with TTT plus in-context learning at inference. Automated data augmentations (e.g., geometric transformations and color permutations) increase the sample size of the TTT dataset for each task. We employ MARC as it is provided in the public code base, with no modifications.⁷

⁷<https://github.com/ekinakyurek/marc>



Şekil 6: CausalARC temaları için örnek gösterim çiftleri.

Birlikte Gözlemlenen Karşıolgusal Nedensel çıkarımın temel problemi, karşıolgusalların fiziksel evrende birlikte gözlemlenmemesidir. Sentetik izgara dünyamızda, veri üretim sürecinde tam kontrole sahip olmamız ve dışsal faktörlerin birden fazla müdahalede sabit tutulabilmesi gereğinden yararlanabiliriz. Böylece, bu sentetik ortam, birden fazla müdahale altındaki ‘işlem görmüş’ örneklerin, tüm durumlarda $U = u$ varsayımlını ihlal etmeksizin, ‘işlem görmemiş’ karşıolgullarıyla yan yana konulmasına olanak tanır. Bu şekilde, T üzerindeki veri artırımlarını bir dizi birlikte gözlemlenen karşıolgusal olarak modelleme seçimiımızı haklı çkarıyoruz (Şekil 5).

İstem Örnekleme CausalARC, kullanıcı belirtimlerine göre temel SCM'den rastgele istemler örnekleme işlevselliliği sunar. Şu anda, aşağıdaki yönler değişimdir: bağlam içi gösterimlerin toplam sayısı, bağlam içi gösterimlerin türü (tümü L1'e karşı dönüşümlü L1/L3) ve sorgu teması (karşıolgusal akıl yürütme, soyut akıl yürütme, program sentezi ve nedensel keşif). Karşıolgusal akıl yürütme için, Dil Modelinden (DM) daha önce görülmüş bir L1 örneği (örn. Şekiller D.4, D.5) için müdahale altında çıktı tahmin etmesi istenir. Soyut akıl yürütme için, DM'den, geleneksel ARC kurulumlarında olduğu gibi, bir L1 girdisi için çıktı dizisini tahmin etmesi istenir [1]. Program sentezi için, DM'den Yapısal Nedensel Modeli (SCM) ifade eden bir Python programı üretmesi istenir (örn. Şekiller D.6, D.7). Nedensel keşif için, DM'den temel SCM'nin bir özelliğini, örneğin bir dizi elemanın nedensel ebeveynlerini veya bir nedensel fonksiyonun biçimini tahmin etmesi istenir (örn. Şekiller D.8, D.9).

4 Ampirik Gösterim

Bir kavram kanıtı olarak, CausalARC için dört potansiyel kullanım durumunu açıklayız: (1) TTT ile soyut akıl yürütme, (2) bağlam içi öğrenme ile karşıolgusal akıl yürütme, (3) program sentezi ve (4) mantıksal akıl yürütme ile nedensel keşif. Tüm deneySEL detaylar ve sonuçlar, her deney için kullanılan görevlerin eksiksiz bir açıklaması da dahil olmak üzere Ek D'de yer almaktadır.

Modeller Modeller, çeşitlilik göz önünde bulundurularak seçilmişdir. Belirlenmiş akıl yürütme modellerini (örn. o4-mini) geleneksel DM'lerle (örn. GPT-4o mini) karşılaştırıyoruz; daha eski, daha az güçlü modelleri (örn. Claude Haiku 3.5) daha yeni, daha güçlü modellerle (örn. Claude Sonnet 4) karşılaştırıyoruz; ve açık kaynaklı modelleri (örn. LLama 3 8B, Llama 4 Scout 17B) kapalı kaynaklı modellerle (diğer tüm modeller) karşılaştırıyoruz. Tüm model detayları Tablo D.1'de verilmiştir.

İstem Formülasyonu PCH seviyesi ve toplam bağlam içi örnek sayısından istem formülasyonunun etkilerini araştırdık. PCH seviyesinin DM performansı üzerindeki etkilerini izole etmek amacıyla, toplam gösterim sayısını sabit tutarak (örn. Şekil D.6'ya karşılık Şekil D.7) L1'li istemleri, L1/L3 dönüşümlü bağlam içi gösterimlere sahip istemlerle karşılaştırıldı.

Metrikler Akyürek vd. [1] çalışmada olduğu gibi, DM performansı çıktı doğruluğu (yani çıktı dizisinin tam olarak doğru olup olmadığı) ve göreceli Hamming mesafesi (HD; yani gerçek dizinin tahmin edilen diziden farklı olduğu konum sayısı, gerçek dizideki eleman sayısına göre normalize edilmiş olarak) ile ölçülüdür. HD için 0 puan doğruluk gösterirken, 1 puan tam bir başarısızlığı (DM'nin hatalı bir dizi döndürdüğü veya hiç yanıt vermediği durumlar dahil) ifade eder.

4.1 Test Anı Eğitimi ile Soyut Akıl Yürütme

Motivasyon CausalARC'nın zorluğunu değerlendirmek amacıyla, MARC'yi TTT [1] ile tam statik veri kümesi üzerinde kıyaslandı. MARC, ARC-AGI-1 için ikincilik ödül alan bir makaleydi. Geniş ARC benzeri veri kümeleri üzerinde ince ayar yapılmış bir LLama 3 8B temel modeli kullanarak sınırsız dönüşüm yaklaşımını benimsenmiş ve çıkış aşamasında TTT ile bağlam içi öğrenmeye kullanır. Otomatik veri artırımları (örn. geometrik dönüşümler ve renk permütasyonları), her görev için TTT veri kümesinin örnek boyutunu artırır. MARC'ı genel kod tabanında sağladığı şekilde, herhangi bir değişiklik yapmadan kullanıyor.⁷

⁷<https://github.com/ekinakyurek/marc>

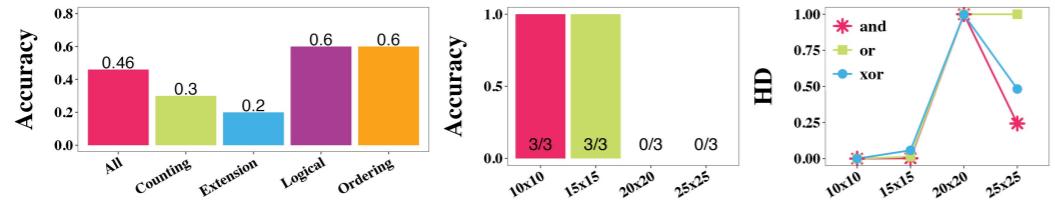


Figure 7: (Left) Accuracy by CausalARC theme for MARC with TTT (Llama 3 8B base) [1]. (Center and right) Performance on *and*, *or*, and *xor* tasks sampled from SCMDky5 as array size increases.

Results & Discussion Overall accuracy on CausalARC was 46%, with significant variation across themes (Figure 7, left). This is similar to MARC’s pure transduction score of 47.1% accuracy on ARC-AGI-1 [1], suggesting that CausalARC is of comparable difficulty. Extension and counting were significantly more challenging for MARC than logic and ordering, though we cannot establish that this is due to the underlying concepts versus other artifacts of these task distributions. MARC shows signs of struggling with large output arrays on both ARC-AGI-1 and CausalARC, as exemplified by steep dropoffs in performance for logical tasks as array size increases (Figure 7, center).

4.2 Counterfactual Reasoning with In-Context Learning

Motivation This experiment demonstrates the use of CausalARC for counterfactual reasoning evaluation with few-shot, in-context learning demonstrations. For each task, three in-context demonstrations were presented before a final test case that prompted the LM to predict the counterfactual output for a previously seen L1 demonstration. In addition to measuring the ability of the LM to predict counterfactual samples from the underlying world model, this experiment compares performance when in-context demonstrations are sampled from L1 versus L3.

Results & Discussion Performance varied widely across models (Figures 8, D.13, D.14, D.15). L3 demonstrations did not consistently confer benefits. This result could arise from lower sample diversity in L3 prompts (as L3 demonstrations are often similar to their L1 counterpart, due to the exogenous context being fixed). Strong performance from recent closed-source models on logical tasks (Figure D.14) could be an artifact of fine-tuning on ARC, as these tasks are similar to ARC-AGI-1 task 31d5ba1a (which used *xor* functions; Example 3.1).

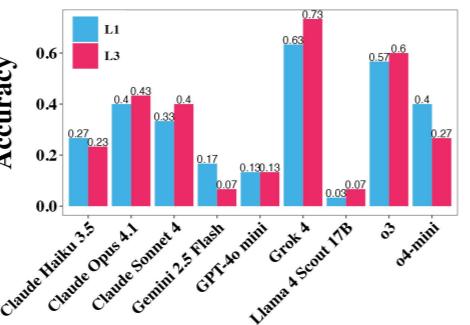
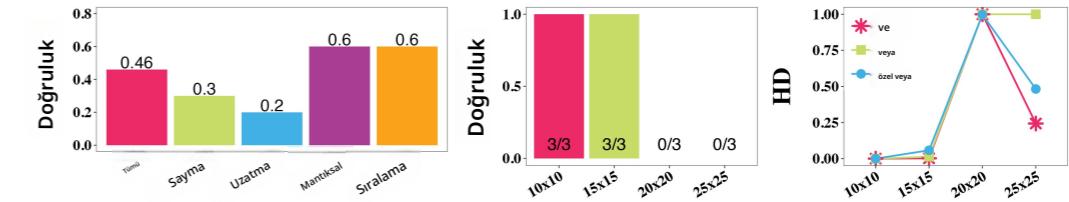


Figure 8: Counterfactual reasoning on six counting, extension, and ordering tasks. Scores are over five random prompt samples.

4.3 Program Synthesis

Motivation Winning ARC strategies fall into two dominant camps [5]: (1) neural transduction approaches, where \hat{y}_{test} is directly predicted (as in MARC); and (2) neural induction with program synthesis, where the neural network outputs a program implementing δ_i that is executed to obtain \hat{y}_{test} . As the first-place ARC-AGI-1 paper winner, BARC [23] took a meta-learning approach [28] that combined neural transduction with Python program synthesis, demonstrating improved performance relative to transduction or induction alone. MARC saw significant performance gains when ensembling with BARC, further supporting the combined use of induction and transduction [1]. Given the success and community interest surrounding program synthesis for reasoning, we also apply CausalARC for this use case. Accuracy was measured with respect to the output \hat{y}_{test} generated by passing a test input x_{test} to the LM-generated program.

Results & Discussion Total in-context examples had a notable impact on program correctness (Figures 9, D.16). Maximum performance was attained by o4-mini, which showed possible signs of fine-tuning for ARC-like program synthesis (Figure D.10). Accuracy peaked at six in-context examples for all models except Sonnet 4 (which saw monotonic increases) and Llama 4 Scout 17B (which failed to output any correct programs). L3 prompts rarely conferred benefits (Figure D.16).



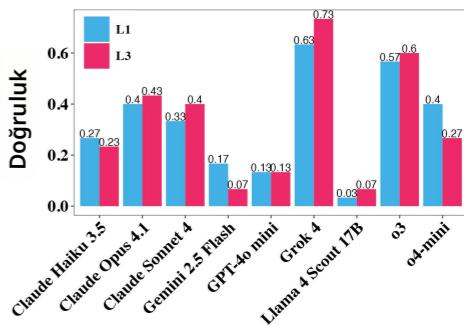
Şekil 7: (Sol) TTT (Llama 3 8B taban) [1] ile MARC için CausalARC temasına göre doğruluk. (Orta ve sağ) Dizi boyutu arttıkça SCMDky5’ten örneklenen ve, veya ve xor görevlerindeki performans.

Sonuçlar ve Tartışma CausalARC’taki genel doğruluk %46 idi ve temalar arasında önemli farklılıklar gözlemlendi (Şekil 7, sol). Bu, ARC-AGI-1 [1] üzerinde MARC’ın %47,1 doğruluk oranına sahip saf dönüşüm puanına benzer olup CausalARC’ın karşılaştırılabilir zorlukta olduğunu düşündürmektedir. Uzatma ve sayma, MARC için mantıksal ve sıralamadan önemli ölçüde daha zorlayıcıydı; ancak bunun altında yatan kavramlardan mı yoksa bu görev dağılımlarının diğer yapılarından mı kaynaklandığını kesin olarak belirleyemiyoruz. MARC, hem ARC-AGI-1’de hem de CausalARC’de büyük çıktı dizileriyle zorlanma belirtileri göstermektedir; dizi boyutu arttıkça mantıksal görevlerdeki performansı keskin düşüşler (Şekil 7, orta) buna örnek teşkil etmektedir.

4.2 Bağlam İçi Öğrenme ile Karşıolgusal Akıl Yürütme

Motivasyon Bu deney, az örnekli, bağlam içi öğrenme gösterimleri ile karşıolgusal akıl yürütme değerlendirmesi için CausalARC kullanımını göstermektedir. Her görev için, daha önce görülen bir L1 gösterimi için DM’yi karşıolgusal çıktıyı tahmin etmeye teşvik eden nihai bir test durumu öncesinde üç bağlam içi gösterim sunulmuştur. Bu deney, DM’nin temel dünya modelinden karşıolgusal örnekleri tahmin etme yeteneğini ölçmenin yanı sıra, bağlam içi gösterimlerin L1’den mi yoksa L3’ten mi örneklemendiğinde performansı karşılaştırmaktadır.

Sonuçlar ve Tartışma Performans modeller arasında büyük farklılıklar gösterdi (Şekil 8, D.13, D.14, D.15). L3 gösterimleri sürekli olarak fayda sağlamadı. Bu sonuç, L3 istemlerindeki daha düşük örnek çeşitliliğinden kaynaklanabilir (günkü L3 gösterimleri genellikle benzerdir egzojen bağlamın sabit olması nedeniyle L1 karşılıklarına). Yeni kapalı kaynak modellerin mantıksal görevlerdeki güçlü performansı (Şekil D.14), bu görevlerin ARC-AGI-1 görevi 31d5ba1a (ki xor işlevlerini kullanmıştır; Örlük 3.1) ile benzer olması nedeniyle ARC üzerinde ince ayar yapılması bir sonucu olabilir.



Şekil 8: Altı adet sayma, genişletme ve sıralama görevinde karşıolgusal akıl yürütme. Puanlar, beş rastgele istem örneği üzerinden alınmıştır.

4.3 Program Sentezi

Motivasyon ARC’yi kazanma stratejileri iki baskın kampa ayrıılır [5]: (1) neural transduction yaklaşımları, burada \hat{y}_{test} doğrudan tahmin edilir (MARC’da olduğu gibi); ve (2) program sentezi ile neural induction, burada yapay sınır ağı y_{test} elde etmek için yürütülen δ işlevini uygulayan bir program çıkarır. Birincilik ödülü alan ARC-AGI-1 makalesi galibi BARC [23], neural transduction’ı Python program sentezi ile birleştirilen bir meta-öğrenme yaklaşımı [28] benimsiyerek, tek başına transduction veya induction'a göre iyileştirilmiş performans sergilemiştir. MARC, BARC ile birleştirildiğinde önemli performans artışları göstererek indüksiyon ve transdüksiyonun birleşik kullanımını [1] daha da destekledi. Akıl yürütme için program sentezine yönelik başarı ve topluluk ilgisi göz önüne alındığında, CausalARC’ı bu kullanım durumu için de uyguluyoruz. Doğruluk, bir test girdisi olan x_{test} ’in DM tarafından oluşturulan programa aktarılmasıyla üretilen çıkış y_{test} ’e göre ölçüldü.

Sonuçlar ve Tartışma Toplam bağlam içi örnekler, program doğruluğu üzerinde kayda değer bir etkiye sahipti (Şekiller 9, D.16). En yüksek performans, ARC benzeri program sentezi için ince ayar belirtileri gösteren o4-mini tarafından elde edildi (Şekil D.10). Doğruluk, Sonnet 4 (monotonik artışlar gösterdi) ve Llama 4 Scout 17B (hiçbir doğru program çıktısı veremedi) dışındaki tüm modeller için altı bağlam içi örnekte zirveye ulaştı. L3 istemleri nadiren fayda sağladı (Şekil D.16).

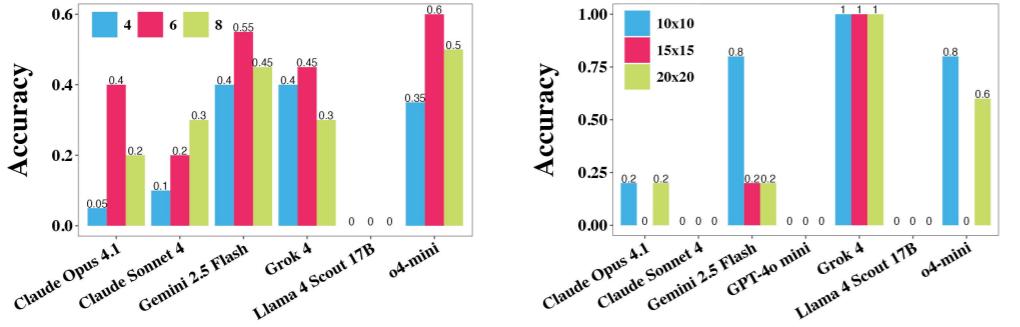


Figure 9: (Left) Program synthesis on four counting and extension tasks as total in-context demonstrations increased. (Right) Causal discovery with logical reasoning as array size increased. Scores were over five random L1 prompt samples. Results for L3 prompts are in Appendix D.2

4.4 Causal Discovery with Logical Reasoning

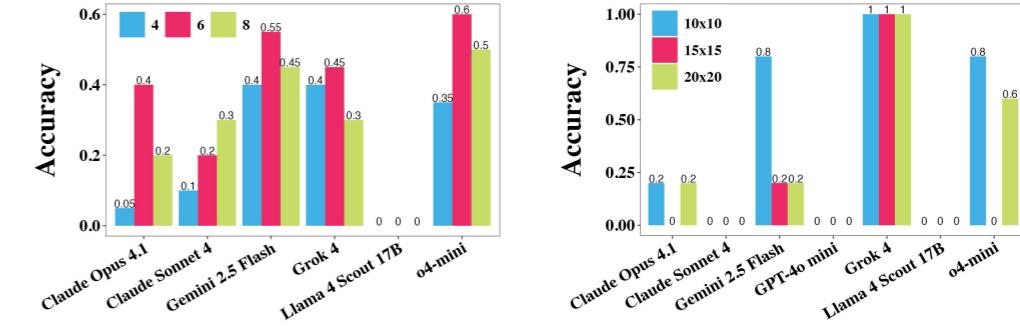
Motivation Causal discovery is the body of theory and methods concerned with learning causal graphical structures from data [8]. Though there is increasing interest in using LMs to support causal discovery, recent evaluations show steep shortcomings [42, 18, 6]. As CausalARC tasks are annotated with fully specified SCMs and (in a subset) the corresponding adjacency matrix, CausalARC is conducive to benchmarking LMs on causal discovery tasks. As a proof-of-concept, we tested the performance of state-of-the-art LMs on causal discovery with logical reasoning with respect to scaling problem complexity (array size). Rather than directly asking for adjacencies in the causal graph, LMs were asked to predict the logical operators acting as causal functions (which implies knowledge of the causal parents of each output cell). Example prompts are provided in Figures D.8 and D.9.

Results & Discussion Performance varied widely, with Grok 4 consistently outperforming (Figures 9, D.17). All other models displayed unstable performance, as monotonic declines in accuracy were not consistently observed. L3 prompts did not consistently confer benefits both within and across models, though benefits were greater than for program synthesis (Figure D.17). Correct responses varied in the extent to which they explicitly enumerated parent-child relationships (Figures D.11, D.12). Future work could query LMs directly for adjacencies, using causal discovery algorithms as baselines. As a proof-of-concept, we ran the popular constraint-based PC algorithm [39] on L1 distributions sampled from SCMDky5 (Table D.4, Figure D.18). Poor performance of PC on samples with *xor* causal function is likely due to a violation of faithfulness: the common assumption that statistical independencies correspond to separations in the underlying graph. Logical *xor* and deterministic functions both famously cause faithfulness violations, which can significantly undermine causal discovery [26]. This may suggest that CausalARC could be extended for testing robustness to faithfulness violations in discrete data, a problem previously explored for continuous data [31].

5 Conclusion

Limitations & Future Directions We introduce CausalARC as an open-ended AI reasoning testbed that accommodates diverse experimental setups, including TTT and in-context learning for counterfactual, abstract, and logical reasoning. Results presented in this proof-of-concept are preliminary and require further investigation. While the manual design of this dataset allows us to maintain strict causal assumptions, this also inhibits scaling and prevents use for fine-tuning and large-scale benchmarking. Future work could expand CausalARC with additional tasks or task themes.

Though within- and between-model performance varied heavily across tasks, there were some signs that state-of-the-art proprietary models benefited from fine-tuning on ARC and its extensions (e.g., o4-mini offered “ARC-style input/output” in some program synthesis responses, despite ARC never being mentioned in the prompt; Claude Sonnet 4 volunteered to perform program synthesis on counterfactual reasoning tasks without being prompted to do so; and Grok 4’s exceptional performance on *xor* logical problems, which are known to exist in ARC-AGI-1). Despite this, we do not observe consistent evidence of saturation on CausalARC in state-of-the-art models.



Şekil 9: (Sol) Toplam bağlam içi gösterimler arttıkça dört sayıya ve genişleştirme görevinde program sentezi. (Sağda) Dizi boyutu arttıkça mantıksal akıl yürütme ile nedensel keşif. Skorlar, beş rastgele L1 istem örneği üzerinden alındı. L3 istemleri için sonuçlar Ek D.2’de yer almaktadır.

4.4 Mantıksal Akıl Yürütme ile Nedensel Keşif

Motivasyon Nedensel keşif, verilerden nedensel grafiksel yapıları öğrenme ile ilgili teori ve yöntemler bütündür [8]. Nedensel keşfi desteklemek için DM’leri kullanmaya artan bir ilgi olmasına rağmen, son değerlendirmeler ciddi eksiklikler göstermektedir [42, 18, 6]. CausalARC görevleri tam olarak belirtilmiş SCM’lerle ve (bir alt kümesinde) karşılık gelen komşuluk matrisi ile açıklanmış olduğundan, CausalARC, nedensel keşif görevlerinde DM’leri kıyaslamak için elverişlidir. Bir kavram kanıti olarak, problem karmaşaklılığını (dizi boyutu) ölçeklendirilmesine göre mantıksal akıl yürütme ile nedensel keşif üzerindeki en son DM’lerin performansını test etti. DM’lerden, nedensel grafikteki komşulukları doğrudan sormak yerine, nedensel fonksiyonlar olarak işlev gören mantıksal operatörleri tahmin etmeleri istendi (bu da her çıktı hücresinin nedensel ebeveynleri hakkında bilgi anlamına gelir). Örnek istemler Şekiller D.8 ve D.9’da sunulmuştur.

Sonuçlar ve Tartışma Performans geniş bir yelpazede değişiklik göstermiş olup, Grok 4 sürekli olarak daha üstün bir performans sergilemiştir (Şekiller 9, D.17). Diğer tüm modeller, doğrulukta monoton düşüşlerin tutarlı bir şekilde gözlemlenmemesi nedeniyle istikrarsız bir performans sergilemiştir. L3 istemleri, program sentezi için faydalıları daha fazla olsa da (Şekil D.17), hem modeller içinde hem de modeller arasında tutarlı faydalara sağlamamıştır. Doğru yanıtlar, ebeveyn-çocuk ilişkilerini açıkça ne ölçüde detaylandırdıklarına göre farklılık göstermiştir (Şekiller D.11, D.12). Gelecekteki çalışmalar, nedensel keşif algoritmalarını temel olarak DM’lerde doğrudan komşulukları sorabilir. Bir kavram kanıtı olarak, popüler kısıt tabanlı PC algoritmasını [39] SCMDky5’ten örneklenen L1 dağılımları üzerinde çalıştık (Tablo D.4, Şekil D.18). PC’nin xor nedensel fonksiyonuna sahip örneklerdeki kötü performansı, büyük olasılıkla bağlılık (faithfulness) ilkesinin ihlalinden kaynaklanmaktadır: istatistiksel bağımsızlıkların altta yatan grafikteki ayırmalara karşılık geldiği yaygın varsayımlı. Mantıksal xor ve deterministik fonksiyonlar, nedensel keşfi önemli ölçüde zayıflatılabilen doğruluk ihlallerine neden olmalarıyla bilinir [26]. Bu durum, CausalARC’ın ayrık verilerdeki doğruluk ihlallerine karşı sağlamlığı test etmek için genişletebileceğini düşündürebilir; bu sorun daha önce sürekli veriler için incelenmiştir [3]

5. Sonuç

Sınırlamalar ve Gelecek Yönelimleri CausalARC’ı, karşılıklı olarak, soyut ve mantıksal akıl yürütme için TTT ve bağlam içi öğrenme dahil olmak üzere çeşitli deneylerin düzeneklere uyum sağlayarak açık uçlu bir YZ akıl yürütme test yatağı olarak sunuyoruz. Bu kavram kanıtında sunulan sonuçlar öncü niteliktedir ve daha fazla araştırma gerektirmektedir. Bu veri kümesinin manuel tasarımlı, katı nedensel varsayımları sürdürmemize olanak tanırken, aynı zamanda ölçeklendirmeyi engeller ve ince ayar ile büyük ölçekli kıyaslama için kullanımını kısıtlar. Gelecekteki çalışmaları CausalARC’ı ek görevler veya görev temalarıyla genişletebilir.

Model içi ve modeller arası performans görevler arasında büyük farklılıklar gösterse de, son teknoloji tescilli modellerin ARC ve uzantıları üzerinde ince ayardan faydalandığını dair bazı işaretler var (örn. o4-mini, isteme ARC’den hiç bahsedilmemesine rağmen bazı program sentezi yanıtlarında “ARC tarzı girdi/çıktı” sundu; Claude Sonnet 4, herhangi bir istem olmadan karşılıklı akıl yürütme görevlerinde program sentezi yapmayı teklif etti; ve Grok 4’ün ARC-AGI-1’de bulunduğu bilinen xor mantıksal problemlerdeki olağanüstü performansı). Buna rağmen, son teknoloji modellerde CausalARC üzerinde tutarlı bir doygunluk kanıtı gözlemlenmiyor.

References

- [1] E. Akyürek, M. Damani, A. Zweiger, L. Qiu, H. Guo, J. Pari, Y. Kim, and J. Andreas. The surprising effectiveness of test-time training for few-shot learning. *International Conference on Machine Learning*, 2025.
- [2] E. Bareinboim, J. D. Correa, D. Ibeling, and T. Icard. *On Pearl's Hierarchy and the Foundations of Causal Inference*, page 507–556. Association for Computing Machinery, New York, NY, USA, 1 edition, 2022. ISBN 9781450395861. URL <https://doi.org/10.1145/3501714.3501743>.
- [3] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 2020.
- [4] F. Chollet. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019.
- [5] F. Chollet, M. Knoop, G. Kamradt, and B. Landers. Arc prize 2024: Technical report. *arXiv preprint arXiv:2412.04604*, 2024.
- [6] J. Gao, X. Ding, B. Qin, and T. Liu. Is chatgpt a good causal reasoner? a comprehensive evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11111–11126, 2023.
- [7] H. Geffner, R. Dechter, and J. Y. Halpern. *Probabilistic and Causal Inference: The Works of Judea Pearl*. ACM, 2022.
- [8] C. Glymour, K. Zhang, and P. Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019.
- [9] M. K. Goddu and A. Gopnik. The development of human causal learning and reasoning. *Nature Reviews Psychology*, pages 1–21, 2024.
- [10] A. Gong, K. Stankevičiūtė, C. Wan, A. Kabra, R. Thesmar, J. Lee, J. Klenke, C. P. Gomes, and K. Q. Weinberger. Phantomwiki: On-demand datasets for reasoning and retrieval evaluation. In *International Conference on Machine Learning*, 2025.
- [11] J. González and A. Nori. Does reasoning emerge? examining the probabilities of causation in large language models. *Advances in Neural Information Processing Systems*, 37:117737–117761, 2024.
- [12] D. Ha and J. Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.
- [13] S. Hao, Y. Gu, H. Ma, J. Hong, Z. Wang, D. Wang, and Z. Hu. Reasoning with language model is planning with world model. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Singapore, Dec. 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.507. URL <https://aclanthology.org/2023.emnlp-main.507/>.
- [14] M. Hodel. Addressing the abstraction and reasoning corpus via procedural example generation, 2024. URL <https://arxiv.org/abs/2404.07353>.
- [15] Q. Huang. Model-based or model-free, a review of approaches in reinforcement learning. In *2020 International Conference on Computing and Data Science (CDS)*, pages 219–221. IEEE, 2020.
- [16] A. Hüyük, X. Xu, J. Maasch, A. V. Nori, and J. González. Reasoning elicitation in language models via counterfactual feedback. *International Conference on Learning Representations*, 2025. URL <https://arxiv.org/abs/2410.03767>.
- [17] Z. Jin, Y. Chen, F. Leeb, L. Gresele, O. Kamal, Z. Lyu, K. Blin, F. Gonzalez Adauto, M. Kleiman-Weiner, M. Sachan, et al. Cladder: Assessing causal reasoning in language models. *Advances in Neural Information Processing Systems*, 2023.

Referanslar

- [1] E. Akyürek, M. Damani, A. Zweiger, L. Qiu, H. Guo, J. Pari, Y. Kim ve J. Andreas. Az örnekli öğrenme için test anı eğitiminin şartlı etkinliği. *Makine Öğrenimi Uluslararası Konferansı*, 2025.
- [2] E. Bareinboim, J. D. Correa, D. Ibeling ve T. Icard. *Pearl Hiyerarşisi ve Nedensel Çikarımın Temelleri Üzerine*, sayfa 507–556. Association for Computing Machinery, New York, NY, ABD, 1. baskı, 2022. ISBN 9781450395861. URL <https://doi.org/10.1145/3501714.3501743>.
- [3] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan , P. Shyam, G. Sastry, A. Askell, et al. Dil modelleri az örnekli öğrenicilerdir. *Sınırsız Bilgi İşleme Sistemlerindeki Gelişmeler*, 2020.
- [4] F. Chollet. Zekanın Ölçümü Üzerine. *arXiv ön baskı arXiv:1911.01547*, 2019.
- [5] F. Chollet, M. Knoop, G. Kamradt ve B. Landers. Arc Ödülü 2024: Teknik Rapor. *arXiv ön baskı arXiv:2412.04604*, 2024.
- [6] J. Gao, X. Ding, B. Qin ve T. Liu. ChatGPT iyi bir nedensel akıl yürütücü müdür? Kapsamlı bir değerlendirme. İçinde *Hesaplama Dilbilim Birliği'nin Bulguları: EMNLP 2023*, sayilar 11111–11126, 2023.
- [7] H. Geffner, R. Dechter ve J. Y. Halpern. *Olasılıksal ve Nedensel Çikarım: Judea Pearl'ün Eserleri*. ACM, 2022.
- [8] C. Glymour, K. Zhang ve P. Spirtes. Grafiksel modellere dayalı nedensel keşif yöntemlerinin incelenmesi. *Frontiers in genetics*, 10:524, 2019.
- [9] M. K. Goddu ve A. Gopnik. İnsan nedensel öğrenimi ve akıl yürütmesinin gelişimi. *Nature Reviews Psychology*, sayfa 1–21, 2024.
- [10] A. Gong, K. Stankevičiūtė, C. Wan, A. Kabra, R. Thesmar, J. Lee, J. Klenke, C. P. Gomes ve K. Q. Weinberger. Phantomwiki: Akıl yürütme ve erişim değerlendirme için isteğe bağlı veri kümeleri. *Makine Öğrenimi Uluslararası Konferansı*, 2025.
- [11] J. González ve A. Nori. Akıl yürütme ortaya çıkıyor mu? büyük dil modellerinde nedenselliğin olasılıklarının incelenmesi. *Sınırsız Bilgi İşleme Sistemlerindeki Gelişmeler*, 37:117737–117761, 2024.
- [12] D. Ha ve J. Schmidhuber. Dünya modelleri. *arXiv preprint arXiv:1803.10122*, 2018.
- [13] S. Hao, Y. Gu, H. Ma, J. Hong, Z. Wang, D. Wang ve Z. Hu. Dil modeliyle akıl yürütme , dünya modeliyle planlamadır. H. Bouamor, J. Pino ve K. Bali, editörler, 2023 *Doğal Dil İşlemede Empirik Yöntemler Konferansı Bildirileri*, Singapur, Ara. 2023. Hesaplama Dilbilim Derneği. doi: 10.18653/v1/2023.emnlp-main.507. URL <https://aclanthology.org/2023.emnlp-main.507/>.
- [14] M. Hodel. Prosedürel örnek üretimi yoluyla soyutlama ve akıl yürütme külliyatını ele almak, 2024. URL <https://arxiv.org/abs/2404.07353>.
- [15] Q. Huang. Model tabanlı mı yoksa modelsiz mi: Takviyeli öğrenmedeki yaklaşım- lara bir genel bakış. 2020 Uluslararası Bilgi İşlem ve Veri Bilimi Konferansı'nda (CDS) , sayilar 219–221. IEEE, 2020.
- [16] A. Hüyük, X. Xu, J. Maasch, A. V. Nori ve J. González. Karşıolgusal geri bildirim yoluyla dil modellerinde akıl yürütme belirleme. *Öğrenme Temsilleri Uluslararası Konferansı*, 2025. URL <https://arxiv.org/abs/2410.03767>.
- [17] Z. Jin, Y. Chen, F. Leeb, L. Gresele, O. Kamal, Z. Lyu, K. Blin, F. Gonzalez Adauto, M. Kleiman-Weiner, M. Sachan, et al. Cladder: Dil modellerinde nedensel akıl yürütmenin değerlendirilmesi. *Sınırsız Bilgi İşleme Sistemlerindeki Gelişmeler*, 2023.

- [18] Z. Jin, J. Liu, Z. Lyu, S. Poff, M. Sachan, R. Mihalcea, M. Diab, and B. Schölkopf. Can large language models infer causation from correlation? *arXiv preprint arXiv:2306.05836*, 2023.
- [19] B. M. Lake and M. Baroni. Human-like systematic generalization through a meta-learning neural network. *Nature*, 623(7985):115–121, 2023.
- [20] B. M. Lake, T. Linzen, and M. Baroni. Human few-shot learning of compositional instructions. *arXiv preprint arXiv:1901.04587*, 2019.
- [21] S. LeGris, W. K. Vong, B. M. Lake, and T. M. Gureckis. H-arc: A robust estimate of human performance on the abstraction and reasoning corpus benchmark. *arXiv preprint arXiv:2409.01374*, 2024.
- [22] S. LeGris, W. K. Vong, B. M. Lake, and T. M. Gureckis. A comprehensive behavioral dataset for the abstraction and reasoning corpus. *Scientific Data*, 2025. doi: <https://doi.org/10.1038/s41597-025-05687-1>.
- [23] W.-D. Li, K. Hu, C. Larsen, Y. Wu, S. Alford, C. Woo, S. M. Dunn, H. Tang, M. Naim, D. Nguyen, et al. Combining induction and transduction for abstract reasoning. *International Conference on Learning Representations*, 2025.
- [24] J. Liang, R. He, and T. Tan. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, 133(1):31–64, 2025.
- [25] J. Maasch, A. Hüyük, X. Xu, A. V. Nori, and J. Gonzalez. Compositional causal reasoning evaluation in language models. *International Conference on Machine Learning*, 2025.
- [26] A. Marx, A. Gretton, and J. M. Mooij. A weaker faithfulness assumption based on triple interactions. In *Uncertainty in Artificial Intelligence*, 2021.
- [27] S. I. Mirzadeh, K. Alizadeh, H. Shahrokhi, O. Tuzel, S. Bengio, and M. Farajtabar. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *International Conference on Learning Representations*, 2025.
- [28] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018.
- [29] A. Moskvichev, V. V. Odouard, and M. Mitchell. The conceptarc benchmark: Evaluating understanding and generalization in the arc domain. *Transactions on Machine Learning Research*, 2023.
- [30] S. Niu, J. Wu, Y. Zhang, Y. Chen, S. Zheng, P. Zhao, and M. Tan. Efficient test-time model adaptation without forgetting. In *International Conference on Machine Learning*. PMLR, 2022.
- [31] M. Olko, M. Gajewski, J. Wojciechowska, M. Morzy, P. Sankowski, and P. Miłoś. Since faithfulness fails: The performance limits of neural causal discovery. In *International Conference on Machine Learning*, 2025.
- [32] J. Pearl. Causality: Models, reasoning, and inference. Cambridge, UK: Cambridge University Press, 19(2):3, 2000.
- [33] J. Pearl. Structural counterfactuals: A brief introduction. *Cognitive Science*, 37(6):977–985, 2013.
- [34] J. Richens and T. Everitt. Robust agents learn causal world models. *International Conference on Learning Representations*, 2024.
- [35] J. Richens, T. Everitt, and D. Abel. General agents need world models. In *International Conference on Machine Learning*, 2025.
- [36] P. Shojaee, I. Mirzadeh, K. Alizadeh, M. Horton, S. Bengio, and M. Farajtabar. The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity. *arXiv preprint arXiv:2506.06941*, 2025.
- [18] Z. Jin, J. Liu, Z. Lyu, S. Poff, M. Sachan, R. Mihalcea, M. Diab ve B. Schölkopf. Büyük dil modelleri korelasyondan nedenselliği çıkarabilir mi? *arXiv ön baskısı arXiv:2306.05836*, 2023.
- [19] B. M. Lake ve M. Baroni. Meta-öğrenen bir sinir ağı aracılığıyla insana benzer sistematik genelleme. *Nature*, 623(7985):115–121, 2023.
- [20] B. M. Lake, T. Linzen ve M. Baroni. Kompozisyonel talimatların insana özgü az örnekli öğrenimi. *arXiv ön baskısı arXiv:1901.04587*, 2019.
- [21] S. LeGris, W. K. Vong, B. M. Lake ve T. M. Gureckis. H-arc: Soyutlama ve Akıl Yürütme Külliyeti kıyaslamasında insan performansının güvenilir bir tahmini. *arXiv ön baskısı arXiv:2409.01374*, 2024.
- [22] S. LeGris, W. K. Vong, B. M. Lake ve T. M. Gureckis. Soyutlama ve Akıl Yürütme Külliyeti için kapsamlı bir davranışsal veri seti. *Scientific Data*, 2025. doi: <https://doi.org/10.1038/s41597-025-05687-1>.
- [23] W.-D. Li, K. Hu, C. Larsen, Y. Wu, S. Alford, C. Woo, S. M. Dunn, H. Tang, M. Naim, D. Nguyen, et al. Soyut akıl yürütme için tümevarım ve tümden gelimin birleştirilmesi. *Öğrenme Temsilleri Uluslararası Konferansı*, 2025.
- [24] J. Liang, R. He ve T. Tan. Dağılım kaymaları altında test zamanı adaptasyonu üzerine kapsamlı bir araştırma. *International Journal of Computer Vision*, 133(1):31–64, 2025.
- [25] J. Maasch, A. Hüyük, X. Xu, A. V. Nori ve J. Gonzalez. Dil modellerinde bileşimsel nedensel akıl yürütme değerlendirmesi. *Makine Öğrenimi Uluslararası Konferansı*, 2025.
- [26] A. Marx, A. Gretton ve J. M. Mooij. Üçlü etkileşimlere dayalı daha zayıf bir sadakat varsayıımı. *Yapay Zeka Belirsizliği içinde*, 2021.
- [27] S. I. Mirzadeh, K. Alizadeh, H. Shahrokhi, O. Tuzel, S. Bengio ve M. Farajtabar. Gsm-sembolik: Büyük dil modellerinde matematiksel akıl yürütmenin sınırlılıklarını anlama. *Öğrenme Temsilleri Uluslararası Konferansı içinde*, 2025.
- [28] N. Mishra, M. Rohaninejad, X. Chen ve P. Abbeel. Basit bir nöral dikkatli meta-öğrenici. *İçinde Öğrenme Temsilleri Uluslararası Konferansı*, 2018.
- [29] A. Moskvichev, V. V. Odouard ve M. Mitchell. ConceptARC kıyaslaması: ARC alanında anlama ve genellemeyi değerlendirme. *Makine Öğrenimi Araştırmaları Üzerine İşlemler*, 2023.
- [30] S. Niu, J. Wu, Y. Zhang, Y. Chen, S. Zheng, P. Zhao ve M. Tan. Unutmadan verimli test zamanı model adaptasyonu. In *Makine Öğrenimi Uluslararası Konferansı*. PMLR, 2022.
- [31] M. Olko, M. Gajewski, J. Wojciechowska, M. Morzy, P. Sankowski ve P. Miłoś. Sadakat başarısız olduğu için: Nöral nedensel keşfin performans sınırları. In *Makine Öğrenimi Uluslararası Konferansı*, 2025.
- [32] J. Pearl. Nedensellilik: Modeller, akıl yürütme ve çıkarım. *Cambridge, Birleşik Krallık: Cambridge University Press*, 19(2):3, 2000.
- [33] J. Pearl. Yapısal karşıolgusallar: Kısa bir giriş. *Bilişsel Bilim*, 37(6):977–985, 2013.
- [34] J. Richens ve T. Everitt. Sağlam araçlar nedensel dünya modelleri öğrenirler. *Öğrenme Temsilleri Uluslararası Konferansı*, 2024.
- [35] J. Richens, T. Everitt ve D. Abel. Genel araçlar dünya modellerine ihtiyaç duyar. In *Makine Öğrenimi Uluslararası Konferansı*, 2025.
- [36] P. Shojaee, I. Mirzadeh, K. Alizadeh, M. Horton, S. Bengio ve M. Farajtabar. Düşünce yanlısaması: Problem karmaşıklığı merceğiyle akıl yürütme modellerinin güçlü yönlerini ve sınırlamalarını anlamak. *arXiv ön baskısı arXiv:2506.06941*, 2025.

- [37] R. B. Shrestha, S. Malberg, and G. Groh. From causal parrots to causal prophets? towards sound causal reasoning with large language models. In *Proceedings of the 5th International Conference on Natural Language Processing for Digital Humanities*, pages 319–333, 2025.
- [38] E. S. Spelke and K. D. Kinzler. Core knowledge. *Developmental science*, 10(1):89–96, 2007.
- [39] P. Spirtes, C. N. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT press, 2000.
- [40] Y. Sun, X. Wang, Z. Liu, J. Miller, A. Efros, and M. Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, pages 9229–9248. PMLR, 2020.
- [41] J. Tian and J. Pearl. Probabilities of causation: Bounds and identification. *Annals of Mathematics and Artificial Intelligence*, 28(1):287–313, 2000.
- [42] R. Tu, C. Ma, and C. Zhang. Causal-discovery performance of chatgpt in the context of neuropathic pain diagnosis. *arXiv preprint arXiv:2301.13819*, 2023.
- [43] J. Wei, J. Wei, Y. Tay, D. Tran, A. Webson, Y. Lu, X. Chen, H. Liu, D. Huang, D. Zhou, et al. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*, 2023.
- [44] X. Xu, R. Lawrence, K. Dubey, A. Pandey, R. Ueno, F. Falck, A. V. Nori, R. Sharma, A. Sharma, and J. Gonzalez. Re-imagine: Symbolic benchmark synthesis for reasoning evaluation. In *International Conference on Machine Learning*, 2025.
- [45] M. Zečević, M. Willig, D. S. Dhami, and K. Kersting. Causal parrots: Large language models may talk causality but are not causal. *Transactions on Machine Learning Research*, 2024.
- [37] R. B. Shrestha, S. Malberg ve G. Groh. Nedensel papağanlardan nedensel peygamberlere mi? büyük dil modelleriyle sağlam nedensel akıl yürütmeye doğru. *Dijital Beşeri Bilimler* için 5. Uluslararası Doğal Dil İşleme Konferansı Bildirileri , sayfa 319–333, 2025.
- [38] E. S. Spelke ve K. D. Kinzler. Temel bilgi. *Gelişimsel bilim* , 10(1):89–96, 2007.
- [39] P. Spirtes, C. N. Glymour ve R. Scheines. *Nedensellik, Tahmin ve Arama* . MIT press, 2000.
- [40] Y. Sun, X. Wang, Z. Liu, J. Miller, A. Efros ve M. Hardt. Dağılım kaymaları altında genelleme için öz denetimli test anı eğitimi. *Makine Öğrenimi Uluslararası Konferansı* , sayfa 9229–9248. PMLR, 2020.
- [41] J. Tian ve J. Pearl. Nedensellik olasılıkları: Sınırlar ve tanımlama. *Matematik ve Yapay Zeka Yıllıkları* , 28(1):287–313, 2000.
- [42] R. Tu, C. Ma ve C. Zhang. Nöropatik ağrı teşhisi bağlamında ChatGPT'nin nedensel keşif performansı. *arXiv ön baskısı arXiv:2301.13819* , 2023.
- [43] J. Wei, J. Wei, Y. Tay, D. Tran, A. Webson, Y. Lu, X. Chen, H. Liu, D. Huang, D. Zhou, et al. Daha büyük dil modelleri, bağlam içi öğrenmeyi farklı şekilde gerçekleştiriyor. *arXiv ön baskısı arXiv:2303.03846* , 2023.
- [44] X. Xu, R. Lawrence, K. Dubey, A. Pandey, R. Ueno, F. Falck, A. V. Nori, R. Sharma, A. Sharma ve J. Gonzalez. Yeniden Hayal Et: Akıl yürütme değerlendirmesi için sembolik kıyaslama sentezi. *Makine Öğrenimi Uluslararası Konferansı* , 2025.
- [45] M. Zečević, M. Willig, D. S. Dhami, and K. Kersting. Causal parrots: Large language models may talk causality but are not causal. *Makine Öğrenimi Araştırmaları Üzerine İşlemler* , 2024.

Appendix

Table of Contents

A Impact Statement	13
B Extended Preliminaries	13
B.1 Defining and Measuring Reasoning	13
B.2 World Models	14
C ARC Through a Causal Lens: Extended Discussion	15
D Extended Empirics	15
D.1 Experimental Details	15
D.2 Experimental Results	24

A Impact Statement

The possibility of AI reasoning emergence has broad scientific, economic, and social implications, including matters of safety and fairness. Though this work aims to promote rigorous reasoning evaluation, strong performance on CausalARC is *necessary but not sufficient* for demonstrating that LMs can reason. Results should be interpreted with great caution when deploying LMs for reasoning tasks that could have safety or fairness consequences.

B Extended Preliminaries

B.1 Defining and Measuring Reasoning

Any framework claiming to measure reasoning should clearly define its target of measure. Reasoning is a famously difficult concept to concretely define and has taken on many meanings across time and research domains. To begin, we first declare our chosen definitions of *generalization* and *intelligence*. We take a definition of generalization that requires adaptability in novel settings.

Definition B.1 (Generalization, Chollet [4]). The ability to handle scenarios or tasks that differ from previously encountered situations, demonstrating both *robustness* and *flexibility*.

Definition B.2 (Robustness, Chollet [4]). Adaptation to *known unknowns* within a single task or well-defined set of tasks.

Definition B.3 (Flexibility, Chollet [4]). Adaptation to *unknown unknowns* across a broad category of related tasks.

In CausalARC, an example of robustness would be high accuracy on a large number of test cases sampled from one causal world model on which the test-taker has undergone TTT. An example of flexibility would be high accuracy on a large number of unique causal world models.

We follow Chollet [4] in defining the intelligence of a system as a measure of *skill-acquisition efficiency*, rather than skill itself. This is consistent with the conventional wisdom that human intelligence tests should measure cognitive capacities in a general sense, and not task-specific skills acquired through experience or practice [4].

Definition B.4 (Intelligence, Chollet [4]). Skill-acquisition efficiency over a range of tasks, controlling for priors, experience, and generalization difficulty.

Thus, we take a *general intelligence* to be one with high skill acquisition efficiency and high generalization over a large range of tasks. We highlight the need to control for experience and priors,

Ek

İçindekiler Tablosu

A Etki Beyanı	13
B Genişletilmiş Ön Bilgiler	13
B.1 Akıl Yürütmeyi Tanımlama ve Ölçme	13
B.2 Dünya Modelleri	14
C Nedensel Mercek Aracılığıyla ARC: Genişletilmiş Tartışma	15
D Genişletilmiş Ampirik Çalışmalar	15
D.1 Deneysel Detaylar	15
D.2 Deneysel Sonuçlar	24

A Etki Beyanı

YZ akıl yürütmesinin ortaya çıkma olasılığı; güvenlik ve adalet konuları da dâhil olmak üzere geniş bilimsel, ekonomik ve sosyal çıkarımlara sahiptir. Bu çalışma, titiz akıl yürütme değerlendirmesini teşvik etmeye amaçlasa da, CausalARC'ta güçlü performans, DM'lerin akıl yürütебildiğini göstermek için *gerekli ancak yeterli değildir*. Güvenlik veya adalet sonuçları doğurabilecek akıl yürütme görevleri için DM'ler kullanıldığında sonuçlar büyük bir dikkatle yorumlanmalıdır.

B Genişletilmiş Ön Bilgiler

B.1 Akıl Yürütmeyi Tanımlama ve Ölçme

Akıl yürütme özgürlüğünü iddia eden herhangi bir çerçeveye, ölçüm hedefini açıkça tanımlamalıdır. Akıl yürütme, somut olarak tanımlanması oldukça zor bir kavramdır ve zamanla ve araştırma alanları arasında birçok anlama bürünmüştür. Başlamak için öncelikle *genelleme* ve *zeka* kavramlarının seçtiğimiz tanımlarını belirtiyoruz. Daha önce karşılaşılmamış ortamlarda uyarlanabilirliği gerektiren bir genelleme tanımı benimsiyoruz.

Tanım B.1 (Genelleme, Chollet [4]) . Daha önce karşılaşılan durumlardan farklı senaryoları veya görevleri ele alma yeteneği, hem sağlamlık he de esneklik gösterme.

Tanım B.2 (Sağlamlık, Chollet [4]) . Tek bir görev veya iyi tanımlanmış görevler kümesi içindeki *bilinen bilinmeyenlere uyum sağlama*.

Tanım B.3 (Esneklik, Chollet [4]) . Geniş bir ilgili görev kategorisindeki *bilinmeyen bilinmeyenlere uyum sağlama*.

CausalARC'ta, sağlamlığın bir örneği, test katılımcısının test zamanı eğitimi (TTT) almış olduğu bir nedensel dünya modelinden örneklenen çok sayıda test durumunda yüksek doğruluk olacaktır. Esnekliğin bir örneği ise, çok sayıda benzersiz nedensel dünya modelinde yüksek doğruluk olacaktır.

Bir sistemin zekasını, becerinin kendisinden ziyade *beceri edinme verimliliğinin* bir ölçüsü olarak tanımlarken Chollet'i [4] esas alıyoruz. Bu durum, insan zeka testlerinin, deneyim veya pratik kazanılan görevde özgü becerileri değil, bilişsel kapasiteleri genel anlamda ölçmesi gereği yönündeki geleneksel görüşle tutarlıdır [4].

Tanım B.4 (Zeka, Chollet[4]) . Ön bilgileri, deneyimi ve genelleme zorluğunu kontrol altında tutarak çeşitli görevlerdeki beceri edinme verimliliği.

Bu nedenle, yüksek beceri edinme verimliliğine ve geniş bir görev yelpazesinde yüksek genellemeye sahip olan bir *genel zekayı* kabul ederiz. Deneyim ve ön bilgileri kontrol etme gerekliliğini vurguluyoruz,

as “unlimited priors or experience can produce systems with little-to-no generalization power (or intelligence) that exhibit high skill at any number of tasks” [4]. This latter case (where skill outpaces generalization) is reminiscent of current shortcomings in large LM reasoning [27, 36, 44], where pretraining on web-scale corpora and extensive fine-tuning could be seen as nearly “ulimited priors or experience.”

Finally, we define our notions of *reasoning*: cognitive processes that are core attributes of intelligent thinking. We focus on abstract, logical, and counterfactual reasoning, as these are the forms tested in the current iteration of CausalARC.

Definition B.5 (Abstract reasoning). The process of drawing valid conclusions about novel visual information by identifying patterns using innate cognitive priors, rather than the accumulation of concrete, domain-specific knowledge.

Here, validity is measured by correctness (e.g., accuracy). Innate cognitive priors are primitives acquired at birth or early development under minimal supervision, broadly pertaining to concepts like objects, actions, number, and space (e.g., a grasp of intuitive physics, such as gravity; basic numeracy; basic geometric operations, such as rotation; a sense of relative size or magnitude, etc.) [38].

Definition B.6 (Logical reasoning). The process of drawing valid conclusions by applying formal logical rules to novel information.

Logical reasoning encompasses many subtypes, including deductive, inductive, and abductive. In this work, we primarily focus on the process of applying logical operators in Boolean algebra.

Next, recall the definition of a counterfactual: *the counterfactual Y_x under model M is given by $Y_x(u) := Y_{M_x}(u)$* (Definition 2.4). We define counterfactual reasoning as follows.

Definition B.7 (Counterfactual reasoning). The process of constructing valid answers to “what if” questions about imagined alternate outcomes in novel settings, as in: “What value would Y have taken had X been x in context $U = u$?”

Then, the *ability to reason abstractly, logically, or counterfactually* is the ability to correctly execute these reasoning processes with robust and flexible generalization. Benchmarks like ARC and CausalARC are designed to measure signs of this ability by presenting the test-taker with diverse and novel problems while controlling for experience and priors. Nevertheless, these benchmarks measure the end-point of reasoning: i.e., the extent to which the final output is correct. Under Definitions B.5, B.6, and B.7, reasoning is a *process* and not an end-product. Thus, there is increasing interest in benchmarks that also assess the intermediate steps of AI reasoning [36]. Additionally, fairly controlling for experience and priors is challenging in the context of large LMs that do not release details of their pretraining and fine-tuning, and it is known that state-of-the-art LMs generally have prior knowledge of ARC. These are just some reasons that we argue that strong performance on CausalARC and analogous benchmarks is *necessary but not sufficient* for demonstrating that LMs can reason.

B.2 World Models

Emergent behaviors in large LMs have raised new questions on whether learning accurate world models can improve AI reasoning [13]. Notably, world models have been variously defined across literatures [12]. In model-based reinforcement learning (RL), a world model is often a predictor of the evolution of an environment under arbitrary policies (e.g., an approximation of the transition function of a Markov decision process) [34]. While the usefulness of world models in RL has been a matter of significant debate [15], the necessity of world models has been proven for multiple settings. Given a sufficiently diverse set of *goal-directed tasks*, any agent satisfying a regret bound for these tasks will necessarily have learned a *predictive world model* that converges to the true model as the agent approaches optimality [35]. Additionally, given regret-bounded policies for a sufficiently large set of *distributional shifts*, the agent necessarily learns an approximate *causal world model* that converges to the true causal model under optimal policies [34]. Richens and Everitt [34] express causal world models as causal Bayesian networks (DAGs), though their results extend to fully specified SCMs. While the present work does not explore CausalARC in the context of RL, we are motivated by the intuition provided by [34, 35] that world models can facilitate (and can even be necessary for) optimal decision-making, planning, and reasoning in AI.

çünkü “sınırsız ön bilgiler veya deneyim, çok az veya hiç genelleme gücü (veya zeka) olmayan, ancak herhangi bir sayıda görevde yüksek beceri sergileyen sistemler üretebilir” [4]. Bu son durum (becerinin genellemenin önüne geçmesi), web öncelikli veri kümeleri üzerinde ön eğitim ve kapsamlı ince ayarın neredeyse “sınırsız ön bilgiler veya deneyim” olarak görülebileceği büyük DM’lerin akıl yürütmesindeki mevcut eksiklikleri anımsatmaktadır [27, 36, 44].

Son olarak, akıl yürütme kavramlarını tanımlıyoruz: akıllı düşünmenin temel nitelikleri olan bilişsel süreçler. Soyut, mantıksal ve karşıolgusal akıl yürütmeye odaklıyoruz; zira bunlar CausalARC’nın mevcut sürümünde test edilen biçimlerdir.

Tanım B.5 (Soyut akıl yürütme). Somut, alana özgü bilgi birikimi yerine, doğuştan gelen bilişsel öncüller kullanarak örüntülerini tanımlayarak yeni görsel bilgiler hakkında geçerli sonuçlar çıkarma süreci.

Burada geçerlilik, doğruluk (örn. accuracy) ile ölçülür. Doğuştan gelen bilişsel öncüller, doğumda veya minimal denetim altında erken gelişimde edinilen, genel olarak nesneler, eylemler, sayı ve uzay gibi kavramlarla ilgili temel öğelerdir (örn. yerçekimi gibi sezgisel fizike dair bir kavrayış; temel sayılar; dönme gibi temel geometrik işlemler; göreceli boyut veya büyülü algısı vb.) [38].

Tanım B.6 (Mantıksal akıl yürütme). Yeni bilgilere resmi mantık kurallarını uygulayarak geçerli sonuçlar çıkarma süreci.

Mantıksal akıl yürütme; dedükatif, indüktif ve abdükatif dahil olmak üzere birçok alt türü kapsar. Bu çalışmada, öncelikli olarak Boolean cebirinde mantıksal operatörlerin uygulanma sürecine odaklıyoruz.

Ardından, bir karşıolgusalın tanımını hatırlatalım: M modeli altındaki karşıolgusal $Y_x, Y_x(u) := YM_x(u)$ (Tanım 2.4) ile verilir. Karşıolgusal akıl yürütmemi aşağıdaki gibi tanımlıyoruz.

Tanım B.7 (Karşıolgusal akıl yürütme). Yeni ortamlarda hayal edilen alternatif sonuçlar hakkında “eğer olursa ne olurdu” sorularına geçerli yanıtlar oluşturma süreci, örneğin: “ $U = u$ bağlamında $X = x$ olmuş olsaydı Y ne değer alırdı?”

Dolayısıyla, soyut, mantıksal veya karşıolgusal akıl yürütme yeteneği, bu akıl yürütme süreçlerini sağlam ve esnek genellemeye doğru bir şekilde yürütme yeteneğidir. ARC ve CausalARC gibi kıyaslama testleri, test adayına deneyim ve ön bilgiyi kontrol ederek çeşitli ve yeni problemler sunarak bu yeteneğin işaretlerini ölçmek için tasarlanmıştır. Bununla birlikte, bu kıyaslamalar akıl yürütmenin son noktasını, yani nihai çıktıının ne ölçüde doğru olduğunu ölçer. Tanımlar B.5, B.6 ve B.7 uyarınca, akıl yürütme bir süreçtir ve bir son ürün değildir. Bu nedenle, YZ akıl yürütmenin ara adımlarını da değerlendiren kıyaslamalara olan ilgi artmaktadır [36]. Ek olarak, ön eğitim ve ince ayar ayrıntılarını yayınlamayan büyük DM’ler bağlamında deneyimi ve ön bilgileri adil bir şekilde kontrol etmek zordur ve son teknoloji DM’lerin genellikle ARC hakkında ön bilgiye sahip olduğu bilinmekte. CausalARC ve benzer kıyaslamalarda güçlü performansın, DM’lerin akıl yürütебildiğini göstermek için gerekli ancak yeterli olmadığını savunmamızın başlıca nedenlerinden bazıları bunlardır.

B.2 Dünya Modelleri

Büyük DM’lerde ortaya çıkan davranışlar, doğru dünya modelleri öğrenmenin YZ akıl yürütmemi geliştiremeyeceği konusunda yeni sorulara yol açmıştır [13]. Özellikle, dünya modelleri literatürler arasında farklı şekillerde tanımlanmıştır [12]. Model tabanlı pekiştirmeli öğrenmede (RL), bir dünya modeli genellikle rastgele politikalar altında bir ortamın evriminin bir öngörücüdür (örneğin, bir Markov karar sürecinin geçiş fonksiyonunun bir yaklaşımı) [34]. Dünya modellerinin RL’deki faydası önemli bir tartışma konusu olsa da [15], dünya modellerinin gerekliliği birçok senaryo için kanıtlanmıştır. Yeterince çeşitli bir hedef odaklı görevler kümlesi verildiğinde, bu görevler için bir pişmanlık sınırı sağlayan herhangi bir ajan, ajan optimaliteye yaklaştıkça gerçek modele yakınsayan önleyici bir dünya modeli öğrenmek zorunda kalacaktır [35]. Ek olarak, yeterince büyük bir dağıtımsal kayma kümlesi için pişmanlık sınırlı politikalar verildiğinde, ajan optimal politikalar altında gerçek nedensel modele yakınsayan yaklaştık bir nedensel dünya modeli öğrenmek zorundadır [34]. Richens ve Everitt [34] nedensel dünya modellerini nedensel Bayes ağları (DAG’ler) olarak ifade etse de, sonuçları tam olarak belirtilmiş SCM’lere kadar genişlemektedir. Mevcut çalışma CausalARC’ı takviyeli öğrenme (RL) bağlamında incelemese de, dünya modellerinin YZ’de optimal karar alma, planlama ve akıl yürütmemi kolaylaştırabileceği (hatta bunun için gerekli olabileceği) [34, 35] tarafından sunulan sezgiyle motive olduk.

C ARC Through a Causal Lens: Extended Discussion

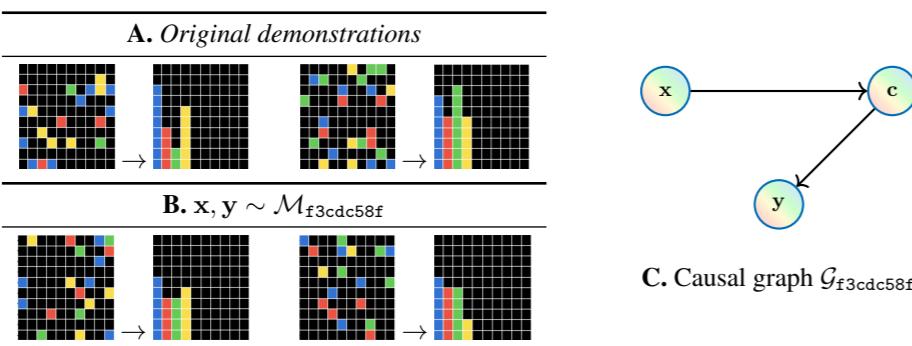


Figure C.1: Input-output arrays for ARC-AGI-1 task f3cdc58f. (A) A subset of official demonstration pairs ($\mathbf{x}_{train}, \mathbf{y}_{train}$). (B) Randomly sampled pairs from the SCM defined in Example C.1. (C) A DAG representation for task f3cdc58f, where c is an array-level feature over x .

We present another example of an ARC-AGI-1 task where the SCM is easily recoverable by observing ($\mathbf{x}_{train}, \mathbf{y}_{train}$). In this case, the SCM includes an endogenous variable that is an array-level feature (count of total pixels per color). Note that when indexing in arrays, we use notation similar to Python (e.g., $\mathbf{y}[:, -1]$ denotes all rows of \mathbf{y} for the final column).

Example C.1 (A fully recovered SCM with array-level features). *Consider ARC-AGI-1 task f3cdc58f (easy level [21, 22]; Figure C.1). We can define an SCM $M_{f3cdc58f} = \langle U, p(u), V, F \rangle$ where V are the elements of our arrays, $f \in F$ are expressed in Equations 5–7, $p(u)$ is categorical with support X , and endogenous variables in output array y are causal children of an array-level feature over x (the total number of elements in x that are a certain color).*

$$\mathbf{u} \sim \text{Cat}(\text{len}(\mathcal{X}), p) \quad \text{for } \mathcal{X} = [0, 1, 2, 3, 4], p = [0.8, 0.05, 0.05, 0.05, 0.05] \quad (4)$$

$$\mathbf{x} = \text{id}_{\mathbf{u}} \quad (5)$$

$$\mathbf{c} = [\text{count}(\mathbf{x} == j)]_{j \in \mathcal{X}} \quad (6)$$

$$\mathbf{y}[-\mathbf{c}[j] :, j - 1] = \begin{cases} j & \text{if } j \in \mathcal{X} \setminus \{0\} \\ 0 & \text{else.} \end{cases} \quad (7)$$

D Extended Empirics

D.1 Experimental Details

Compute Resources All LM experiments were run on an AWS EC2 g6e.xlarge instance featuring one GPU with 48GB GPU memory and four vCPUs.⁸ All other experiments were run locally on a MacBook Pro (Apple M2 Pro chip, 16 GB memory).

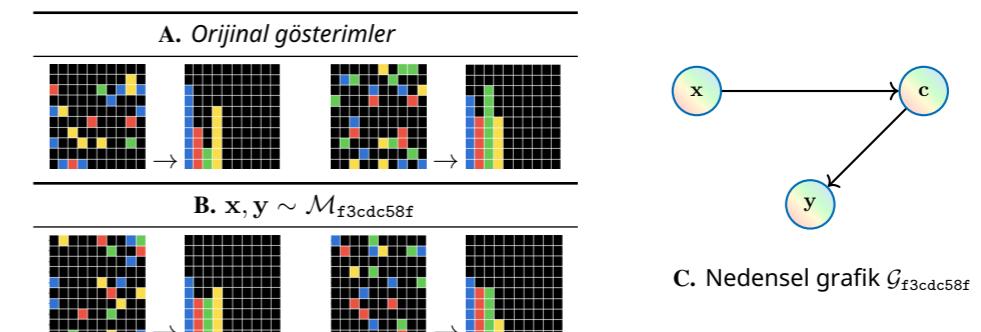
Models Benchmarking experiments used the models described in Table D.1. All models used default hyperparameters and temperature 0.0 (except o4-mini, which required temperature 1.0). Models were queried using the langchain Python package.⁹

Prompt Formulation Several experiments compare the impacts of L1 versus L3 in-context demonstrations on LM performance. To prevent the conflation of benefits from PCH level versus query length, the total number of in-context demonstrations was held constant across L1 and L3 prompts. Headers for L3 demonstrations cause a small increase in prompt length, as these are slightly longer than L1 example headers. For example, the L1 prompt in Figure D.4 is 4130 characters long, while the L3 prompt in Figure D.5 is 4267 characters. In this case, including L3 demonstrations resulted in a 3.2% length increase.

⁸<https://aws.amazon.com/ec2/instance-types/g6e/>

⁹<https://python.langchain.com/>

C Nedensel Mercek Aracılığıyla ARC: Genişletilmiş Tartışma



Şekil C.1: ARC-AGI-1 görevi için girdi-çıkıtı dizileri f3cdc58f. (A) Resmi gösterim çiftlerinin bir alt kümesi ($\mathbf{x}_{train}, \mathbf{y}_{train}$). (B) Örnek C.1'de tanımlanan SCM'den rastgele örneklenmiş çiftler. (C) Görev f3cdc58f için bir DAG gösterimi; burada c , x üzerinde dizi düzeyinde bir özellikir.

SCM'nin ($\mathbf{x}_{train}, \mathbf{y}_{train}$) gözlemlenerek kolayca kurtarılabilir bir ARC-AGI-1 görevinin başka bir örneğini sunuyoruz. Bu durumda, SCM, bir dizi düzeyindeki özellik olan endojen bir değişken içermektedir. (her renk için toplam piksel sayısı). Dizilerde indeksleme yaparken Python'a benzer bir gösterim kullandığımızı unutmayın (örn. $\mathbf{y}[:, -1]$ ifadesi, \mathbf{y} 'nin son sütununa ait tüm satırlarını belirtir).

Örnek C.1 (Dizi düzeyinde özelliklere sahip, tamamen kurtarılmış bir SCM). ARC-AGI-1 görevi f3cdc58fyi (kolay seviye [21, 22]; Şekil C.1) ele alalım. Bir SCM $M_{f3cdc58f} = \langle U, p(u), V, F \rangle$ tanımlayabiliriz, burada V , dizilerimizin elemanlarıdır, $f \in F$, Denklemler 5–7'de ifade edilmiştir, $p(u)$, destek X ile kategoriktir ve çıktı dizisindeki y endojen değişkenleri, x üzerindeki bir dizi düzeyindeki özelliğin nedensel çocuklarıdır (x 'te belirli bir renkte olan toplam eleman sayısı).

$$\mathbf{u} \sim \text{Cat}(\text{len}(\mathcal{X}), p) \quad \text{for } \mathcal{X} = [0, 1, 2, 3, 4], p = [0.8, 0.05, 0.05, 0.05, 0.05] \quad (4)$$

$$\mathbf{x} = \text{id}_{\mathbf{u}} \quad (5)$$

$$\mathbf{c} = [\text{count}(\mathbf{x} == j)]_{j \in \mathcal{X}} \quad (6)$$

$$\mathbf{y}[-\mathbf{c}[j] :, j - 1] = \begin{cases} j & \text{if } j \in \mathcal{X} \setminus \{0\} \\ 0 & \text{diğer durumda.} \end{cases} \quad (7)$$

D Genişletilmiş Ampirik Çalışmalar

D.1 Deneysel Detaylar

Hesaplama Kaynakları Tüm DM deneyleri, 48 GB GPU belleğe ve dört vCPU'ya sahip tek bir GPU içeren bir AWS EC2 g6e.xlarge örneği üzerinde çalıştırıldı.⁸ Diğer tüm deneyler yerel olarak bir MacBook Pro (Apple M2 Pro çipi, 16 GB bellek) üzerinde çalıştırıldı.

Modeller Kiyaslama deneylerinde Tablo D.1'de açıklanan modeller kullanılmıştır. Tüm modeller varsayılan hiperparametreleri ve 0.0 sıcaklık değerini kullanmıştır (sıcaklık değeri 1.0 gerektiren o4-mini hariç). Modeller, langchain Python paketi kullanılarak sorgulanmıştır.⁹

İstem Formülasyonu Birçok deney, L1 ve L3 bağlam içi gösterimlerin DM performansı üzerindeki etkilerini karşılaştırmaktadır. PCH seviyesinden ve sorgu uzunluğundan kaynaklanan faydalara birbirine karışmasına önlemek amacıyla, L1 ve L3 istemlerinde toplam bağlam içi gösterim sayısı sabit tutulmuştur.

L3 gösterimleri için başlıklar, L1 örnek başlıklarından biraz daha uzun olduklarından, istem uzunluğunda küçük bir artışa neden olmaktadır. Örneğin, Şekil D.4'teki L1 istemi 4130 karakter uzunluğundayken, Şekil D.5'teki L3 istemi 4267 karakterdir. Bu durumda, L3 gösterimlerinin dahil edilmesi %3,2'lük bir uzunluk artışıyla sonuçlanmıştır.

⁸<https://aws.amazon.com/ec2/instance-types/g6e/>

⁹<https://python.langchain.com/>

PROVIDER	MODEL	API ID	URL
OpenAI	GPT-4o mini o4-mini	gpt-4o-mini-2024-07-18 o4-mini-2025-04-16	https://platform.openai.com/docs/models
Anthropic	Claude Haiku 3.5 Claude Sonnet 4	claude-3-5-haiku-20241022 claude-sonnet-4-20250514	https://docs.anthropic.com/en/docs/about-claude/models/
Google	Genini 2.5 Flash	gemini-2.5-flash	https://ai.google.dev/gemini-api/docs/models
XAI	Grok 4	grok-4	https://docs.x.ai/docs/models
Meta (Bedrock)	Llama 4 Scout 17B	us.meta.llama4-scout-17b-instruct-v1:0	https://llama.developer.meta.com/docs/models/

Table D.1: Models used in benchmarking experiments.

D.1.1 Counterfactual Reasoning with In-Context Learning

Tasks Counterfactual reasoning experiments used task instances sampled from all task themes. Counting tasks were SCMfuy3 and SCMm5ob. Extending tasks were SCMz750 and SCMwoev. Ordering tasks were SCMffb8 and SCMtzlq. All tasks with tunable array sizes (all except SCMfuy3 and SCMffb8) used size 25×25 .

Logical counterfactual reasoning tasks used variants of SCMdky5, SCMu3am, and SCMtcbq, which are based to varying degrees on ARC-AGI-1 task 31d5ba1a (Example 3.1). Causal functions for SCMdky5 were a single logical operator (*and*, *or*, *xor*) while SCMu3am and SCMtcbq used two logical operators. Task SCMdky5 was generated for each logical operator at three dimensionalities (10×10 , 15×15 , and 20×20), each with their own color palette (9 tasks total; Figure D.3). Tasks SCMu3am and SCMtcbq were generated analogously, using one logical operator combination each (6 tasks total; 3 per SCM).

D.1.2 Program Synthesis

Tasks A single task was sampled from each of four SCMs: SCMm5ob (counting), SCMev5t (counting), SCMfpq (extension), and SCMz750 (extension). Example prompts are provided in Figures D.6 and D.7. Prompts were sampled with varying numbers of in-context demonstrations (4, 6, 8) to assess impacts on program correctness.

D.1.3 Causal Discovery with Logical Reasoning

Tasks For each of three tasks sampled from SCMtcbq (where logical operators composed *xor* after *and*), five prompts were sampled per level of the PCH. Tasks featured scaling problem complexity with respect to output array size (10×10 , 15×15 , and 20×20).

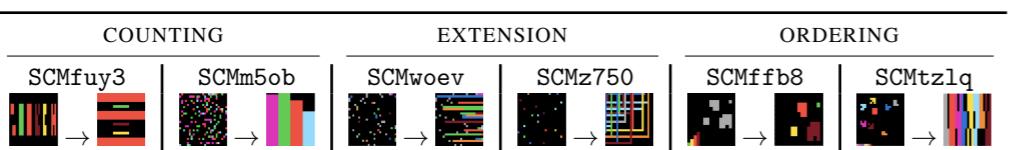


Figure D.2: Counting, extension, and ordering test cases for the counterfactual reasoning experiment.

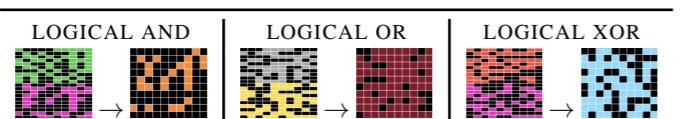


Figure D.3: Samples from CausalARC task SCMdky5, where causal functions were *and*, *or*, and *xor*.

SAĞLAYICI	MODEL	API KİMLİĞİ	URL
OpenAI	GPT-4o mini o4-mini	gpt-4o-mini-2024-07-18 o4-mini-2025-04-16	https://platform.openai.com/docs/models
Anthropic	Claude Haiku 3.5 Claude Sonnet 4	claude-3-5-haiku-20241022 claude-sonnet-4-20250514	https://docs.anthropic.com/en/docs/about-claude/models/
Google	Gemini 2.5 Flash	gemini-2.5-flash	https://ai.google.dev/gemini-api/docs/models
XAI	Grok 4	grok-4	https://docs.x.ai/docs/models
Meta (Bedrock)	Llama 4 Scout 17B	us.meta.llama4-scout-17b-instruct-v1:0	https://llama.developer.meta.com/docs/models/

Table D.1: Kiyaslama deneylerinde kullanılan modeller.

D.1.1 Bağlam İçi Öğrenme ile Karşıolgusal Akıl Yürütme

Görevler Karşıolgusal akıl yürütme deneylerinde, tüm görev temalarından örneklendişmiş görev örnekleri kullanıldı. Sayma görevleri SCMfuy3 ve SCMm5ob idi. Genişletme görevleri SCMz750 ve SCMwoev idi. Sıralama görevleri SCMffb8 ve SCMtzlq idi. Ayarlanabilir dizi boyutlarına sahip tüm görevler (SCMfuy3 ve SCMffb8 haricindekiler) 25×25 boyutunu kullandı.

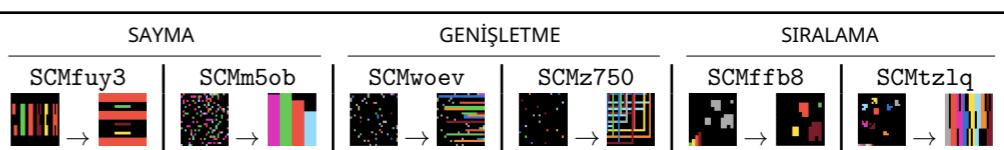
Mantıksal karşıolgusal akıl yürütme görevleri, ARC-AGI-1 görevi 31d5ba1a (Örnek 3.1) temel alınarak farklı derecelerde SCMdky5, SCMu3am ve SCMtcbq varyantlarını kullandı. SCMdky5 için nedensel fonksiyonlar tek bir mantıksal operatör (ve, veya, xor) iken, SCMu3am ve SCMtcbq için iki mantıksal operatör kullanıldı. SCMdky5 görevi, üç farklı boyutta (10×10 , 15×15 ve 20×20) her bir mantıksal operatör için, her birine özgü renk paletiyle oluşturulmuştur (toplam 9 görev; Şekil D.3). SCMu3am ve SCMtcbq görevleri de benzer şekilde, her biri bir mantıksal operatör kombinasyonu kullanılarak üretilmiştir (toplam 6 görev; her SCM için 3).

D.1.2 Program Sentezi

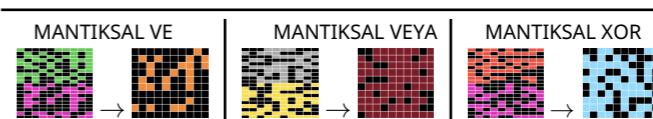
Görevler Dört SCM'den (SCMm5ob (sayma), SCMev5t (sayma), SCMfpq (genişletme) ve SCMz750 (genişletme)) her birinden tek bir görev örneklandı. Örnek yönlendirmeler Şekil D.6 ve D.7'de sunulmuştur. Programın doğruluğu üzerindeki etkilerini değerlendirmek amacıyla, değişen sayıda bağlam içi gösterim (4, 6, 8) ile yönlendirmeler örneklandı.

D.1.3 Mantıksal Akıl Yürütme ile Nedensel Keşif

Görevler SCMtcbq'dan (mantıksal operatörlerin xor sona ve öncesinde olduğu) örneklenen üç görevden her biri için, PCH'nin her seviyesinden beş istem örneklandı. Görevler, çıktı dizisi boyutuna göre artan problem karmaşıklığına sahipti (10×10 , 15×15 , ve 20×20).



Şekil D.2: Karşıolgusal akıl yürütme deneyi için sayma, genişletme ve sıralama test durumları.



Şekil D.3: Nedensel fonksiyonların and, or ve xor olduğu CausalARC görevi SCMdky5'ten örnekler.

Prom

You must solve the following puzzle by discovering the deterministic rule that maps inputs to outputs. You will then be asked to predict the output for a counterfactual example. Both the inputs and outputs are 2D grids of colored pixels. We provide example input-output pairs as demonstration. Grids are provided as Python arrays. You must output only a single Python array, and do not explain your reasoning.

Example input-output arrays

```

Example Input Output Array:
[[0, 4, 4, 0, 4, 0, 0, 4, 0, 4], [4, 4, 0, 0, 4, 0, 4, 4, 0, 4], [0, 4, 4, 0, 0,
4, 4, 4, 0, 4], [4, 4, 4, 4, 0, 0, 4, 0, 4, 0], [4, 0, 0, 0, 0, 0, 0, 4, 4, 0],
[4, 4, 4, 0, 0, 4, 0, 0, 4], [4, 4, 4, 0, 4, 4, 0, 4, 4, 4], [0, 0, 4, 0, 0,
4, 4, 4, 0, 4], [4, 4, 4, 0, 4, 0, 4, 4, 4, 0], [0, 0, 4, 4, 0, 4, 0, 0, 4, 4],
[0, 0, 5, 0, 5, 0, 5, 0, 0, 5], [0, 5, 5, 5, 5, 5, 0, 5, 0, 5], [5, 5, 0, 0, 0,
5, 5, 0, 0, 5], [0, 0, 0, 0, 5, 5, 0, 0, 0, 0], [0, 0, 0, 5, 5, 0, 0, 5, 5, 0],
[5, 0, 0, 5, 5, 5, 0, 0, 5, 5], [5, 0, 0, 0, 5, 5, 0, 0, 5, 0], [0, 5, 5, 5, 5,
0, 5, 0, 5, 0], [5, 0, 0, 0, 5, 0, 5, 5, 5, 0], [5, 0, 0, 5, 0, 5, 5, 5, 0]
-> [[0, 0, 2, 0, 2, 0, 0, 0, 0, 2], [0, 2, 0, 0, 2, 0, 0, 2, 0, 2], [0, 2, 0, 0,
0, 2, 2, 0, 0, 2], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 2, 2,
0], [2, 0, 0, 0, 0, 2, 0, 0, 0, 2], [2, 0, 0, 0, 2, 2, 0, 0, 2, 0], [0, 0, 2, 0,
0, 0, 2, 0, 0, 0], [2, 0, 0, 0, 2, 0, 2, 2, 2, 0], [0, 0, 0, 2, 0, 2, 0, 0, 2,
0]]

```

Example input-output arrays:

```
[[4, 0, 0, 4, 0, 0, 0, 4, 4, 4], [0, 4, 0, 4, 0, 0, 4, 0, 0, 0], [0, 4, 4, 0, 4, 0, 4, 0, 4, 0, 0], [0, 4, 0, 0, 4, 0, 0, 0, 0, 0], [0, 0, 4, 0, 0, 0, 0, 0, 0, 4], [4, 0, 0, 0, 4, 4, 0, 0, 0, 0], [0, 0, 0, 0, 4, 4, 4, 0, 4, 4], [4, 0, 0, 0, 0, 4, 0, 4, 0, 4], [4, 4, 4, 0, 4, 4, 0, 4, 0, 0], [4, 0, 0, 4, 0, 4, 0, 0, 0, 0], [5, 5, 5, 0, 5, 0, 5, 0, 5, 5], [0, 5, 5, 0, 5, 5, 0, 5, 0, 5], [5, 0, 5, 5, 5, 0, 0, 0, 0, 0], [5, 0, 0, 5, 5, 0, 5, 0, 5, 5], [5, 5, 5, 5, 5, 5, 5, 0, 5, 5, 5], [5, 5, 0, 0, 5, 0, 0, 0, 5, 5], [0, 0, 5, 5, 0, 5, 5, 5, 5, 5], [5, 0, 0, 0, 0, 5, 0, 5, 5, 5], [0, 5, 5, 5, 5, 0, 5, 5, 0, 0], [0, 0, 0, 0, 0, 0, 5, 5, 0, 5]]  
→ [[2, 0, 0, 0, 0, 0, 0, 2, 2], [0, 2, 0, 0, 0, 0, 0, 0, 0], [0, 0, 2, 0, 0, 0, 0, 0, 0], [0, 0, 0, 2, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 2, 2, 0, 2, 2], [2, 0, 0, 0, 0, 0, 0, 0, 2], [0, 2, 2, 0, 2, 0, 0, 2, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0]]
```

Example input-output arrays:

Counterfactual: Now imagine that we intervened on the previous input by changing some colors.

```

some colors.
[[0, 0, 0, 8, 0, 0, 0, 0, 0, 8], [0, 8, 0, 0, 8, 0, 0, 0, 8, 0], [8, 8, 0, 8, 0,
8, 8, 8, 8, 0], [8, 0, 0, 0, 8, 8, 0, 8, 0, 8], [8, 0, 0, 0, 0, 0, 8, 0, 0, 8],
[0, 8, 8, 8, 0, 0, 0, 0, 8, 0], [0, 8, 8, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0,
0, 8, 0, 8, 8], [0, 8, 8, 0, 8, 0, 8, 0, 8, 0, 8], [8, 8, 0, 0, 8, 0, 8, 8, 8],
[5, 0, 0, 5, 5, 0, 0, 0, 0, 5], [0, 0, 5, 0, 5, 0, 0, 0, 0, 0], [5, 0, 0, 0, 0,
5, 0, 0, 0, 0], [0, 0, 0, 5, 5, 5, 0, 5, 0, 5], [0, 0, 0, 0, 0, 5, 5, 0, 5],
[5, 5, 5, 0, 0, 5, 5, 0, 0, 5], [0, 5, 5, 5, 0, 0, 5, 5, 0, 5], [0, 5, 5, 0, 0,
0, 5, 5, 5], [5, 5, 0, 5, 5, 5, 0, 5, 0, 0], [0, 5, 5, 0, 5, 5, 5, 0, 5, 0]]
->

```

Istem

Girdileri çıktılara eşleyen deterministik kuralı keşfederek aşağıdaki bulmacayı çözmeliiniz. Daha sonra, karşılgusal bir örnek için çıktıyı tahmin etmeniz istenecektir. Hem girdiler hem de çıktılar, renkli piksellerden oluşan 2B ızgaralardır. Gösterim olarak örnek girdi-çıktı çiftleri sunuyoruz. İzgaralar Python dizileri olarak sağlanır. Yalnızca tek bir Python dizisi çıktısı vermelisiniz ve akıl yürütmenizi açıklamamalısınız.

Örnek girdi-cıktı dizileri:

Çıktı girildiğinde:

```
[ [0, 4, 4, 0, 4, 0, 0, 4, 0, 4], [4, 4, 0, 0, 4, 0, 4, 4, 4, 0, 4], [0, 4, 4, 0, 0, 4, 4, 4, 0, 4], [4, 4, 4, 0, 4], [4, 4, 4, 4, 0, 0, 4, 0, 4, 0], [4, 0, 0, 0, 0, 0, 0, 4, 4, 0], [4, 4, 4, 0, 0, 0, 4, 0, 0, 4], [4, 4, 4, 0, 4, 4, 0, 4, 4, 4], [0, 0, 4, 0, 0, 4, 4, 0, 4], [4, 4, 4, 0, 0, 4, 0, 4, 0, 4], [4, 4, 4, 0, 4, 0, 4, 4, 0, 4], [0, 0, 4, 4, 0, 4, 4, 0, 4], [4, 4, 4, 0, 4, 0, 4, 4, 0, 4], [0, 0, 4, 4, 0, 4, 4, 0, 4], [0, 0, 5, 0, 5, 0, 0, 0, 5], [0, 5, 5, 5, 5, 5, 0, 5, 0, 5], [5, 5, 0, 0, 0, 5, 5, 0, 0, 5], [0, 0, 0, 0, 5, 5, 0, 0, 0, 0], [0, 0, 0, 5, 5, 0, 0, 5, 5, 0], [5, 0, 0, 5, 5, 5, 0, 0, 5, 5], [5, 0, 0, 0, 5, 5, 5, 0, 0, 5, 0], [0, 5, 5, 5, 5, 0, 5, 0, 5, 0], [5, 0, 0, 0, 0, 5, 0, 5, 5, 5, 0], [5, 0, 0, 0, 5, 0, 5, 5, 5, 0], [5, 0, 0, 0, 5, 5, 0, 5, 5, 0], [5, 0, 0, 0, 0, 5, 0, 5, 5, 5, 0], [5, 0, 0, 0, 5, 0, 5, 5, 5, 0], [5, 0, 0, 0, 0, 0, 5, 0, 5, 5, 5, 0], [5, 0, 0, 0, 0, 5, 0, 5, 5, 5, 0], [0, 2, 0, 0, 2, 0, 0, 0, 0, 2], [0, 2, 0, 0, 2, 0, 0, 2, 0, 2], [0, 2, 0, 0, 2, 2, 0, 0, 0, 2], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 0], [2, 0, 0, 0, 0, 2, 0, 0, 0, 2], [2, 0, 0, 0, 2, 2, 0, 0, 2, 0], [0, 0, 2, 0, 0, 0, 2, 0, 0, 0, 0], [2, 0, 0, 0, 2, 0, 2, 2, 0, 0], [0, 0, 0, 2, 0, 2, 0, 2, 0, 0, 2, 0]]
```

Örnek girdi-çıktı dizileri:

```
[[4, 0, 0, 4, 0, 0, 0, 4, 4, 4], [0, 4, 0, 4, 0, 0, 4, 0, 0, 0], [0, 4, 4, 0, 4, 0, 4, 0, 4, 0, 0, 0], [0, 4, 0, 0, 4, 0, 0, 0, 0, 0, 0], [0, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0], [4, 0, 0, 0, 4, 4, 0, 0, 0, 0], [0, 0, 0, 0, 4, 4, 4, 0, 4, 4], [4, 0, 0, 0, 4, 0, 4, 0, 0, 4], [4, 4, 4, 0, 4, 4, 0, 4, 0, 0, 0], [4, 0, 0, 4, 0, 4, 0, 0, 0, 0], [5, 5, 5, 0, 5, 0, 5, 0, 5, 5], [0, 5, 5, 0, 5, 5, 0, 5, 0, 5], [5, 0, 5, 5, 5, 0, 0, 0, 0, 0], [5, 0, 0, 5, 5, 0, 5, 0, 5, 5], [5, 5, 5, 5, 5, 5, 5, 0, 5, 5, 5], [5, 5, 0, 0, 5, 0, 0, 0, 5, 5], [0, 0, 5, 5, 0, 5, 5, 5, 5, 5], [5, 0, 0, 0, 0, 0, 0, 5, 5], [0, 5, 5, 5, 5, 0, 5, 5, 0, 0], [0, 0, 0, 0, 0, 0, 0, 5, 5, 0, 5], -> [[2, 0, 0, 0, 0, 0, 0, 2, 2], [0, 2, 0, 0, 0, 0, 0, 0, 0], [0, 0, 2, 0, 0, 0, 0, 0, 0], [2, 0, 0, 0, 2, 0, 0, 0, 0, 0], [0, 0, 0, 0, 2, 2, 0, 2, 2], [2, 0, 0, 0, 0, 0, 0, 0, 2], [0, 2, 2, 0, 2, 0, 0, 2, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]]
```

Örnek girdi-çıktı dizileri:

Örnek girdi-çıktı dizileri:

```
[[0, 0, 0, 4, 0, 0, 0, 0, 0, 4], [0, 4, 0, 0, 4, 0, 0, 0, 0, 4, 0], [4, 4, 0, 4, 0, 4, 4, 4, 4, 4, 0, 0], [4, 0, 0, 0, 4, 4, 0, 4, 0, 4], [4, 0, 0, 0, 0, 0, 0, 4, 0, 0, 4], [0, 4, 4, 4, 0, 0, 0, 0, 4, 0], [0, 4, 4, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 4, 0, 4, 4], [0, 4, 4, 0, 4, 0, 4, 4, 0, 4], [4, 4, 0, 0, 4, 0, 4, 4, 4, 4], [5, 0, 0, 5, 5, 0, 0, 0, 0, 5], [0, 0, 5, 0, 5, 0, 0, 0, 0, 0], [5, 0, 0, 0, 0, 5, 0, 0, 0, 0], [0, 0, 0, 5, 5, 5, 0, 5, 0, 5], [0, 0, 0, 0, 0, 5, 5, 0, 0, 5], [5, 5, 5, 0, 0, 5, 5, 0, 0, 5], [0, 5, 5, 5, 0, 0, 5, 5, 0, 5], [0, 5, 5, 0, 0, 0, 5, 5, 5], [5, 5, 0, 5, 5, 5, 5, 0, 5, 0], [0, 5, 5, 0, 5, 5, 5, 0, 5, 0]]  
-> [[0, 0, 0, 2, 0, 0, 0, 0, 0, 2], [0, 0, 0, 0, 2, 0, 0, 0, 0, 0], [2, 0, 0, 0, 0, 2, 0, 2, 0, 0, 0], [0, 0, 0, 0, 2, 2, 0, 2, 0, 2], [0, 0, 0, 0, 0, 0, 2, 0, 0, 2], [0, 2, 2, 0, 0, 0, 0, 0, 0, 0], [0, 2, 2, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 2, 2], [0, 2, 0, 0, 2, 0, 0, 2, 0, 0], [0, 2, 0, 0, 2, 0, 0, 2, 0, 2], [0, 2, 0, 0, 2, 0, 0, 2, 2, 0]]
```

Karşılolgusal: Şimdi, önceki girdiye bazı renkleri değiştirerek müdahale ettiğimizi hedefin

```

[[0, 0, 0, 8, 0, 0, 0, 0, 0, 8], [0, 8, 0, 0, 8, 0, 0, 0, 0, 8], [8, 8, 0, 8, 0,
8, 8, 8, 0], [8, 0, 0, 0, 8, 8, 0, 8, 0, 8], [8, 0, 0, 0, 0, 0, 0, 0, 0, 8],
[0, 8, 8, 0, 0, 0, 0, 0, 8, 0], [0, 8, 8, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0,
0, 8, 0, 8, 8], [0, 8, 8, 0, 8, 0, 8, 0, 8, 0, 8], [8, 8, 0, 0, 8, 0, 8, 8, 8, 8],
[5, 0, 0, 5, 5, 0, 0, 0, 0, 5], [0, 0, 5, 0, 5, 0, 0, 0, 0, 0], [5, 0, 0, 0, 0,
5, 0, 5, 0, 0], [0, 0, 0, 5, 5, 5, 0, 5, 0, 5], [0, 0, 0, 0, 0, 5, 5, 0, 5],
[5, 5, 5, 0, 0, 5, 5, 0, 0, 5], [0, 5, 5, 5, 0, 0, 5, 5, 0, 5], [0, 5, 5, 5, 0,
0, 0, 5, 5, 5], [5, 5, 0, 5, 5, 5, 0, 5, 0, 5], [0, 5, 5, 0, 5, 5, 5, 0, 5, 0, 5]]
->

```

Figure D.4: L1 prompt for counterfactual reasoning.

Şekil D.4: Karşıolqusal akıl yürütme için L1 istemi.

Prompt

You must solve the following puzzle by discovering the deterministic rule that maps inputs to outputs. You will then be asked to predict the output for a counterfactual example. Both the inputs and outputs are 2D grids of colored pixels. We provide example input-output pairs along with counterfactual examples, which represent interventions on the original examples. Grids are provided as Python arrays. You must output only a single Python array, and do not explain your reasoning.

Example input-output arrays

Counterfactual: Now imagine that we intervened on the previous input by fixing some values.

Example input-output arrays:

```

[[0, 0, 0, 4, 0, 0, 0, 0, 0, 4], [0, 4, 0, 0, 4, 0, 0, 0, 0, 4, 0], [4, 4, 0, 4, 0,
4, 4, 4, 4, 0], [4, 0, 0, 0, 4, 4, 0, 4, 0, 4], [4, 0, 0, 0, 0, 0, 4, 0, 0, 4],
[0, 4, 4, 4, 0, 0, 0, 0, 4, 0], [0, 4, 4, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0,
0, 4, 0, 4, 4], [0, 4, 4, 0, 4, 0, 4, 4, 0, 4], [4, 4, 0, 0, 4, 0, 4, 4, 4],
[5, 0, 0, 5, 5, 0, 0, 0, 0, 5], [0, 0, 5, 0, 5, 0, 0, 0, 0, 0], [5, 0, 0, 0, 0,
5, 0, 5, 0, 0], [0, 0, 0, 5, 5, 5, 0, 5, 0, 5], [0, 0, 0, 0, 0, 5, 5, 0, 0, 5],
[5, 5, 5, 0, 0, 5, 5, 0, 0, 5], [0, 5, 5, 5, 0, 0, 5, 5, 0, 5], [0, 5, 5, 0, 0,
0, 5, 5, 5], [5, 5, 0, 5, 5, 5, 0, 5, 0, 0], [0, 5, 5, 0, 5, 5, 0, 5, 0, 0]
-> [[0, 0, 0, 2, 0, 0, 0, 0, 0, 2], [0, 0, 0, 0, 2, 0, 0, 0, 0, 0], [2, 0, 0, 0,
0, 2, 0, 2, 0, 0], [0, 0, 0, 0, 2, 2, 0, 2, 0, 2], [0, 0, 0, 0, 0, 0, 2, 0, 0,
2], [0, 2, 2, 0, 0, 0, 0, 0, 0], [0, 2, 2, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0,
0, 0, 0, 2, 2], [0, 2, 0, 0, 2, 0, 0, 2, 0, 0], [0, 2, 0, 0, 2, 0, 0, 2, 2],

```

Counterfactual: Now imagine that we intervened on the previous input by changing some colors.

```
some colors.  
[[0, 0, 0, 8, 0, 0, 0, 0, 0, 8], [0, 8, 0, 0, 8, 0, 0, 0, 0, 8], [8, 8, 0, 8, 0,  
8, 8, 8, 8, 0], [8, 0, 0, 0, 8, 8, 0, 8, 0, 8], [8, 0, 0, 0, 0, 0, 0, 8, 0, 0, 8],  
[0, 8, 8, 8, 0, 0, 0, 0, 8, 0], [0, 8, 8, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0,  
0, 8, 0, 8, 8], [0, 8, 8, 0, 8, 0, 8, 8, 0, 8], [8, 8, 0, 0, 8, 0, 8, 8, 8, 8],  
[5, 0, 0, 5, 5, 0, 0, 0, 0, 5], [0, 0, 5, 0, 5, 0, 0, 0, 0, 0], [5, 0, 0, 0, 0,  
5, 0, 5, 0, 0], [0, 0, 0, 5, 5, 5, 0, 5, 0, 5], [0, 0, 0, 0, 0, 5, 5, 0, 0, 5],  
[5, 5, 5, 0, 0, 5, 5, 0, 0, 5], [0, 5, 5, 5, 0, 0, 5, 5, 0, 5], [0, 5, 5, 0, 0,  
0, 0, 5, 5, 5], [5, 5, 0, 5, 5, 5, 0, 5, 0, 0], [0, 5, 5, 0, 5, 5, 5, 0, 5, 0, 0]]
```

Istem

Girdileri çıktılara eşleyen deterministik kuralı keşfederek aşağıdaki bulmacayı çözmelisiniz. Daha sonra, karşıolgusal bir örnek için çıktıyı tahmin etmeniz istenecektir. Hem girdiler hem de çıktılar, renkli piksellerden oluşan 2B izgaralarıdır. Orijinal örneklerde müdahaleleri temsil eden karşıolgusal örneklerle birlikte örnek girdi-çıktı çiftleri sunuyoruz. Izgaralar Python dizileri olarak sağlanır. Yalnızca tek bir Python dizisi çıktısı vermelisiniz ve akıl yürütmenizi açıklamamalısınız.

Örnek girdi-çıktı dizileri:

```

[[4, 0, 4, 0, 4, 0, 0, 0, 4, 4, 4], [0, 4, 0, 4, 0, 0, 4, 0, 0, 0], [0, 4, 4, 0, 4,
0, 4, 4, 0, 0], [0, 4, 0, 0, 4, 0, 0, 0, 0, 0], [0, 0, 4, 0, 0, 0, 0, 0, 0, 4],
[4, 0, 0, 0, 4, 4, 0, 0, 0, 0], [0, 0, 0, 0, 4, 4, 4, 0, 4, 4], [4, 0, 0, 0, 4,
0, 4, 0, 0, 4], [4, 4, 4, 0, 4, 4, 0, 4, 0, 0], [4, 0, 0, 4, 0, 4, 0, 0, 0, 0],
[5, 5, 5, 0, 5, 0, 5, 0, 5, 5], [0, 5, 5, 0, 5, 5, 0, 5, 0, 5], [5, 0, 5, 5, 5,
0, 0, 0, 0, 0], [5, 0, 0, 5, 5, 0, 5, 0, 5, 5], [5, 5, 5, 5, 5, 5, 5, 0, 5, 5, 5],
[5, 5, 0, 0, 5, 0, 0, 0, 5, 5], [0, 0, 5, 5, 0, 5, 5, 5, 5, 5], [5, 0, 0, 0, 0,
5, 0, 0, 5, 5], [0, 5, 5, 5, 5, 0, 5, 5, 0, 0], [0, 0, 0, 0, 0, 0, 5, 5, 0, 5]]
-> [[2, 0, 0, 0, 0, 0, 0, 2, 2], [0, 2, 0, 0, 0, 0, 0, 0, 0], [0, 0, 2, 0,
2, 0, 0, 0, 0, 0], [0, 0, 0, 0, 2, 0, 0, 0, 0, 0], [0, 0, 2, 0, 0, 0, 0, 0, 0],
[2, 0, 0, 0, 0, 2, 0, 0, 0, 0], [0, 0, 0, 0, 0, 2, 2, 0, 2, 2], [2, 0, 0, 0,
0, 0, 0, 0, 0, 2], [0, 2, 2, 0, 2, 0, 0, 2, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0,
0]]

```

Karşıolgusal: Şimdi, önceki girdiye bazı değerleri sabitleyerek müdahale ettiğimizi hayal edin.

Örnek girdi-çıktı dizileri:

```

[[0, 0, 0, 4, 0, 0, 0, 0, 0, 4], [0, 4, 0, 0, 4, 0, 0, 0, 4, 0], [4, 4, 0, 4, 0,
4, 4, 4, 4, 0], [4, 0, 0, 0, 4, 4, 0, 4, 0, 4], [4, 0, 0, 0, 0, 0, 4, 0, 0, 4],
[0, 4, 4, 4, 0, 0, 0, 0, 4, 0], [0, 4, 4, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0,
0, 4, 0, 4, 4], [0, 4, 4, 0, 4, 0, 4, 4, 0, 4], [4, 4, 0, 0, 4, 0, 4, 4, 4, 4],
[5, 0, 0, 5, 5, 0, 0, 0, 0, 5], [0, 0, 5, 0, 5, 0, 0, 0, 0, 0], [5, 0, 0, 0, 0,
5, 0, 5, 0, 0], [0, 0, 0, 5, 5, 5, 0, 5, 0, 5], [0, 0, 0, 0, 0, 5, 5, 0, 0, 5],
[5, 5, 5, 0, 0, 5, 5, 0, 0, 5], [0, 5, 5, 5, 0, 0, 5, 5, 0, 5], [0, 5, 5, 0, 0,
0, 0, 5, 5, 5], [5, 5, 0, 5, 5, 5, 0, 5, 0, 0], [0, 5, 5, 0, 5, 5, 0, 5, 5, 0]]
-> [[0, 0, 0, 2, 0, 0, 0, 0, 0, 2], [0, 0, 0, 0, 2, 0, 0, 0, 0, 0], [2, 0, 0, 0,
0, 2, 0, 2, 0, 0], [0, 0, 0, 0, 2, 2, 0, 2, 0, 2], [0, 0, 0, 0, 0, 0, 2, 0, 0,
2], [0, 2, 2, 0, 0, 0, 0, 0, 0, 0], [0, 2, 2, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0,
0, 0, 0, 2, 2], [0, 2, 0, 0, 2, 0, 0, 2, 0, 0], [0, 2, 0, 0, 2, 0, 0, 2, 2, 2],

```

0]]
Karşıolgusal: Şimdi, önceki girdiye bazı renkleri değiştirerek müdahale ettiğimizi hedef edin

```

edin.
[[0, 0, 0, 8, 0, 0, 0, 0, 0, 8], [0, 8, 0, 0, 8, 0, 0, 0, 0, 8, 0], [8, 8, 0, 8, 0,
8, 8, 8, 0], [8, 0, 0, 0, 8, 8, 0, 8, 0, 8], [8, 0, 0, 0, 0, 0, 0, 0, 0, 8], [0, 8, 8, 8, 0, 0, 0, 0, 0, 8, 0], [0, 8, 8, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 8, 0, 8, 8, 8], [0, 8, 8, 0, 8, 0, 8, 8, 0, 8], [8, 8, 0, 0, 8, 0, 8, 8, 0, 8], [5, 0, 0, 5, 5, 0, 0, 0, 0, 5], [0, 0, 5, 0, 5, 0, 0, 0, 0, 0], [5, 0, 0, 0, 0, 0, 0, 0, 0, 0], [5, 0, 5, 0, 0, 0], [0, 0, 0, 5, 5, 5, 0, 5, 0, 5], [0, 0, 0, 0, 0, 5, 5, 0, 5, 0, 5], [5, 5, 5, 0, 0, 5, 5, 0, 0, 5], [0, 5, 5, 5, 0, 0, 5, 5, 0, 5], [0, 5, 5, 0, 0, 0, 5, 5, 0, 5], [5, 5, 0, 5, 5, 5, 0, 5, 0, 0], [0, 5, 5, 0, 5, 5, 5, 0, 5, 0, 0], [0, 5, 5, 0, 5, 5, 5, 0, 5, 0, 0]]

```

Figure D.5: L3 prompt for counterfactual reasoning.

Şekil D.5: Karşıolqusal akıl yürütme için L3 istemi.

Prompt

You must solve the following puzzle by discovering the deterministic rule that maps inputs to outputs. Both the inputs and outputs are 2D Python arrays of colored pixels. We provide example input-output pairs as demonstration. To solve the problem, express the deterministic rule as a Python program. Do not explain your reasoning, and only output a single Python program.

Example input-output arrays:

```

[[0, 0, 0, 0, 0, 9, 0, 4, 0, 3], [0, 4, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 4,
0, 2, 0, 0, 0], [0, 0, 0, 0, 9, 0, 0, 2, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 3],
[0, 9, 0, 0, 0, 4, 0, 0, 0, 2], [0, 0, 0, 0, 0, 9, 0, 2, 4, 0], [3, 0, 3, 0, 3,
0, 0, 0, 3, 0], [0, 0, 4, 3, 0, 0, 3, 0, 4, 0], [9, 3, 0, 0, 0, 0, 2, 0, 0, 0]]
-> [[0, 0, 0, 3], [0, 0, 0, 3], [4, 0, 0, 3], [4, 0, 0, 3], [4, 2, 9, 3], [4, 2,
9, 3], [4, 2, 9, 3], [4, 2, 9, 3], [4, 2, 9, 3]]

```

Example input-output arrays:

```

[[0, 0, 0, 0, 0, 0, 0, 0, 0], [9, 0, 0, 2, 0, 0, 9, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 2, 0, 9, 0], [9, 4, 0, 0, 9, 3, 0, 0, 0, 0], [0, 0, 0, 2, 0, 0, 2, 3, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 9, 4, 0, 9, 0, 0, 0, 0, 2, 9], [0, 4, 0, 0, 0, 2, 0, 0, 0, 0, 0], [0, 0, 3, 0, 9, 0, 9, 0, 9, 4], [0, 0, 9, 0, 4, 0, 0, 0, 3, 0]]  
-> [[0, 0, 9, 0], [0, 0, 9, 0], [0, 0, 9, 0], [0, 0, 9, 0], [0, 0, 9, 0], [0, 0, 9, 0], [0, 2, 9, 0], [4, 2, 9, 0], [4, 2, 9, 3], [4, 2, 9, 3], [4, 2, 9, 3], [4, 2, 9, 3]]

```

Example input-output arrays:

```

Example Input Output arrays:
[[0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 3], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 9, 0, 0, 0, 3, 0, 0, 0, 0, 0], [0, 9, 3, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0], [3, 0, 0, 0, 0, 3, 0, 9, 0, 0, 0], [0, 0, 4, 0, 0, 9, 0, 0, 9, 4], [0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0], [9, 0, 4, 3, 2, 9, 0, 0, 0, 0, 0], [4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]] -> [[0, 0, 0, 3], [0, 0, 9, 3], [0, 0, 9, 3], [4, 0, 9, 3], [4, 0, 9, 3], [4, 0, 9, 3], [4, 0, 9, 3], [4, 2, 9, 3]]

```

Example input-output arrays:

```

Example Input Output arrays:
[[0, 0, 2, 0, 0, 0, 0, 0, 0, 0], [0, 0, 9, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 2,
0, 0, 0, 0, 0], [0, 9, 2, 2, 2, 0, 0, 0, 0, 9], [0, 0, 2, 0, 0, 0, 0, 0, 0, 0], [0, 4, 3, 0, 0,
0, 0, 0, 3, 0], [0, 0, 0, 2, 3, 0, 0, 0, 0, 0], [0, 0, 4, 9, 2, 0, 0, 4, 4], [0, 0, 0, 0, 4, 0, 0, 0, 0, 0], [0, 0, 4, 0, 0, 0, 0, 9, 3, 9]]
-> [[0, 2, 0, 0], [0, 2, 0, 0], [4, 2, 9, 0], [4, 2, 9, 0], [4, 2, 9, 3], [4, 2, 9, 3], [4, 2, 9, 3], [4, 2, 9, 3], [4, 2, 9, 3], [4, 2, 9, 3]]
```

Example input-output arrays:

```

Example Input-Output arrays:
[[0, 9, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 3, 0, 9, 2, 0], [0, 0, 0, 0, 0, 0, 2, 0, 0, 0], [0, 4, 2, 0, 0, 0, 0, 9, 0, 0], [0, 0, 0, 4, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 2, 0, 0, 0, 0], [0, 0, 0, 2, 4, 0, 0, 3, 0, 0], [3, 2, 0, 4, 0, 3, 0, 2, 0, 0], [0, 3, 0, 0, 4, 0, 9, 9, 4, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]  

-> [[0, 2, 0, 0], [4, 2, 0, 0], [4, 2, 9, 3], [4, 2, 9, 3], [4, 2, 9, 3], [4, 2, 9, 3], [4, 2, 9, 3], [4, 2, 9, 3]]

```

Example input-output arrays:

Istem

Girdileri çıktılarla eşleyen deterministik kuralı keşfederek aşağıdaki bulmacayı çözmeliiniz. Hem girdiler hem de çıktılar, renkli piksellerden oluşan 2B Python dizileridir. Gösterim olarak örnek girdi-çıktı çiftleri sunuyoruz. Problemi çözmek için deterministik kuralı bir Python programı olarak ifade edin. Akıl yürütmenizi açıklamayın ve yalnızca tek bir Python programı çıktısı verin.

Örnek girdi-çıktı dizileri:

```

[[0, 0, 0, 0, 0, 9, 0, 4, 0, 3], [0, 4, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 4,
0, 2, 0, 0, 0], [0, 0, 0, 0, 0, 9, 0, 0, 2, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 3],
[0, 9, 0, 0, 0, 4, 0, 0, 0, 2], [0, 0, 0, 0, 0, 9, 0, 2, 4, 0], [3, 0, 3, 0, 3,
0, 0, 0, 3, 0], [0, 0, 4, 3, 0, 0, 3, 0, 4, 0], [9, 3, 0, 0, 0, 0, 2, 0, 0, 0]]
-> [[0, 0, 0, 3], [0, 0, 0, 3], [4, 0, 0, 3], [4, 0, 0, 3], [4, 2, 9, 3], [4, 2,
9, 3], [4, 2, 9, 3], [4, 2, 9, 3], [4, 2, 9, 3]]

```

Örnek girdi-çıktı dizileri:

```

[[0, 0, 0, 0, 0, 0, 0, 0, 0], [9, 0, 0, 2, 0, 0, 9, 0, 0, 0], [0, 0, 0, 0, 0, 0,
0, 2, 0, 9, 0], [9, 4, 0, 0, 9, 3, 0, 0, 0, 0], [0, 0, 0, 2, 0, 0, 2, 3, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 9, 4, 0, 9, 0, 0, 0, 2, 9], [0, 4, 0, 0, 0,
2, 0, 0, 0, 0], [0, 0, 3, 0, 9, 0, 9, 0, 9, 4], [0, 0, 9, 0, 4, 0, 0, 0, 3, 0]]
-> [[0, 0, 9, 0], [0, 0, 9, 0], [0, 0, 9, 0], [0, 0, 9, 0], [0, 0, 9, 0], [0, 0,
9, 0], [0, 2, 9, 0], [4, 2, 9, 0], [4, 2, 9, 3], [4, 2, 9, 3], [4, 2, 9, 3], [4,
2, 9, 3]]]
```

Örnek girdi-cıktı dizileri:

Örnek girdi-çıktı dizileri:

```
Erkek gruptaki dizilerin  
[[0, 0, 2, 0, 0, 0, 0, 0, 0, 0], [0, 0, 9, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 2,  
0, 0, 0, 0, 0], [0, 9, 2, 2, 2, 0, 0, 0, 0, 9], [0, 0, 2, 0, 0, 0, 0, 0, 0, 0],  
[0, 4, 3, 0, 0, 0, 0, 0, 3, 0], [0, 0, 0, 2, 3, 0, 0, 0, 0, 0], [0, 0, 4, 9, 2,  
0, 0, 0, 4, 4], [0, 0, 0, 0, 4, 0, 0, 0, 0, 0], [0, 0, 4, 0, 0, 0, 0, 9, 3, 9]]  
-> [[0, 2, 0, 0], [0, 2, 0, 0], [4, 2, 9, 0], [4, 2, 9, 0], [4, 2, 9, 3], [4, 2,  
9, 3], [4, 2, 9, 3], [4, 2, 9, 3]]
```

Örnek qirdi-cıktı dizileri:

```
Oluşan girdi-çıktı dizinleri:  
[[0, 9, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 3, 0, 9, 2, 0], [0, 0, 0, 0, 0, 0,  
2, 0, 0, 0, 0], [0, 4, 2, 0, 0, 0, 0, 9, 0, 0], [0, 0, 0, 4, 0, 0, 0, 0, 0, 0],  
[0, 0, 0, 0, 0, 0, 2, 0, 0, 0], [0, 0, 0, 2, 4, 0, 0, 3, 0, 0], [3, 2, 0, 4, 0,  
3, 0, 2, 0, 0], [0, 3, 0, 0, 4, 0, 9, 9, 4, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]  
→ [[0, 2, 0, 0], [4, 2, 0, 0], [4, 2, 9, 3], [4, 2, 9, 3], [4, 2, 9, 3], [4, 2,  
9, 3], [4, 2, 9, 3]]
```

Örnek qirdi-cıktı dizileri:

Orijinal girdi-çıktı dizinleri:
[[0, 0, 0, 0, 0, 3, 0, 0, 0, 3], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 9, 0, 0, 0, 3, 0, 0, 0, 0, 0], [0, 9, 3, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0], [3, 0, 0, 0, 0, 3, 0, 9, 0, 0], [0, 0, 4, 0, 0, 9, 0, 0, 9, 4], [0, 3, 0, 0, 0, 0, 0, 0, 0, 0], [9, 0, 4, 3, 2, 9, 0, 0, 0, 0], [4, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
-> [[0, 0, 0, 3], [0, 0, 9, 3], [0, 0, 9, 3], [4, 0, 9, 3], [4, 0, 9, 3], [4, 0, 9, 3], [4, 0, 9, 3], [4, 0, 9, 3], [4, 2, 9, 3]]

Figure D.6: L1 prompt for program synthesis with six in-context examples.

Şekil D.6: Altı bağlam içi örnekle program sentezi için L1 istemi.

Prompt

You must solve the following puzzle by discovering the deterministic rule that maps inputs to outputs. Both the inputs and outputs are 2D Python arrays of colored pixels. We provide example input-output pairs along with counterfactual examples, which represent interventions on the original examples. To solve the problem, express the deterministic rule as a Python program. Do not explain your reasoning, and only output a single Python program.

Example input-output arrays:

Counterfactual: Now imagine that we intervened on the previous input by rotating or flipping it.

```

[[0, 0, 9, 0, 4, 0, 0, 0, 3, 0], [0, 0, 3, 0, 9, 0, 9, 0, 9, 4], [0, 4, 0, 0, 0,
2, 0, 0, 0, 0], [0, 9, 4, 0, 9, 0, 0, 0, 2, 9], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 2, 0, 0, 2, 3, 0, 0], [9, 4, 0, 0, 9, 3, 0, 0, 0, 0], [0, 0, 0, 0, 0,
0, 2, 0, 9, 0], [9, 0, 0, 2, 0, 0, 9, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
-> [[0, 0, 9, 0], [0, 0, 9, 0], [0, 0, 9, 0], [0, 0, 9, 0], [0, 0, 9, 0], [0, 0,
9, 0], [0, 2, 9, 0], [4, 2, 9, 0], [4, 2, 9, 3], [4, 2, 9, 3], [4, 2, 9, 3], [4,
2, 9, 3]]

```

Example input-output arrays:

```

[[0, 0, 0, 0, 0, 0, 4, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0], [0, 0, 0, 2, 0,
0, 0, 0, 0, 0], [0, 4, 0, 0, 0, 9, 0, 0, 4, 0], [0, 0, 0, 0, 0, 3, 0, 0, 3, 0],
[0, 0, 0, 0, 0, 4, 2, 0, 0], [0, 2, 0, 0, 0, 0, 3, 0, 0, 0], [0, 0, 2, 0, 0,
0, 0, 0, 0, 0], [0, 3, 0, 0, 0, 0, 0, 4, 0, 0], [0, 0, 0, 0, 0, 9, 4, 0, 0, 0]]
-> [[4, 0, 0, 0], [4, 0, 0, 0], [4, 0, 0, 0], [4, 2, 0, 3], [4, 2, 0, 3], [4, 2,
9, 3], [4, 2, 9, 3]]

```

Counterfactual: Now imagine that we intervened on the previous input by changing some colors.

```

[[7, 7, 7, 7, 7, 7, 4, 7, 7, 7], [7, 7, 7, 7, 7, 7, 7, 7, 4, 7], [7, 7, 7, 2, 7,
7, 7, 7, 7, 7], [7, 4, 7, 7, 7, 9, 7, 7, 4, 7], [7, 7, 7, 7, 7, 3, 7, 7, 3, 7],
[7, 7, 7, 7, 7, 4, 2, 7, 7], [7, 2, 7, 7, 7, 3, 7, 7, 7], [7, 7, 2, 7, 7,
7, 7, 7, 7, 7], [7, 3, 7, 7, 7, 7, 4, 7, 7], [7, 7, 7, 7, 7, 9, 4, 7, 7, 7]]
-> [[4, 7, 7, 7], [4, 7, 7, 7], [4, 7, 7, 7], [4, 2, 7, 3], [4, 2, 7, 3], [4, 2,
9, 3], [4, 2, 9, 3]]

```

Example input-output arrays:

```

[[0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0], [4, 0, 0, 0, 0, 0, 9, 3, 0, 0, 0], [0, 0, 0, 0, 0,
0, 0, 4, 0, 0], [0, 0, 0, 0, 2, 0, 4, 0, 0, 0], [0, 3, 0, 0, 4, 0, 0, 0, 0, 2, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 2, 4, 2, 0, 0, 0], [0, 2, 0, 9, 3,
0, 9, 0, 3, 0], [0, 0, 0, 0, 0, 0, 4, 0, 0, 0], [0, 3, 4, 0, 4, 9, 4, 0, 0, 0]]
-> [[4, 0, 0, 0], [4, 0, 0, 0], [4, 0, 0, 0], [4, 0, 0, 3], [4, 2, 0, 3], [4, 2,
9, 3], [4, 2, 9, 3], [4, 2, 9, 3], [4, 2, 9, 3]]

```

Counterfactual: Now imagine that we intervened on the previous input by rotating or flipping it.

```

[[0, 4, 9, 0, 9, 0, 0, 0, 2, 0], [0, 0, 0, 0, 4, 0, 2, 0, 0, 0], [0, 0, 0, 2, 0,
0, 0, 0, 0, 0], [0, 0, 0, 4, 0, 0, 0, 0, 0, 0], [0, 0, 9, 0, 9, 0, 0, 0, 0, 4],
[9, 0, 0, 0, 2, 0, 0, 4, 9, 4], [0, 0, 0, 0, 0, 0, 0, 2, 0, 0], [0, 0, 9, 0, 0,
3, 0, 3, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 2, 0, 0, 0, 0, 3, 4, 0, 0]]
-> [[4, 0, 9, 0], [4, 2, 9, 0], [4, 2, 9, 0], [4, 2, 9, 0], [4, 2, 9, 3], [4, 2,
9, 3], [4, 2, 9, 3]]]
```

İstem

Girdiler çıktılarla eşleyen deterministik kuralı keşfederek aşağıdaki bulmacayı çözmeliiniz. Hem girdiler hem de çıktılar, renkli piksellerden oluşan 2B Python dizileridir. Örjinal örneklerde müdahaleleri temsil eden karşıolgusal örneklerle birlikte örnek girdi-çıktı çiftleri sunuyoruz. Sorunu çözmek için deterministik kuralı bir Python programı olarak ifade edin. Akıl yürütmenizi açıklamayın ve yalnızca tek bir Python programı çıktısı verin.

Örnek girdi-çıktı dizileri

```

[[4, 3, 0, 0, 0, 9, 9, 9, 0, 0], [0, 0, 0, 3, 3, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0,
0, 2, 0, 9, 0], [0, 3, 0, 0, 0, 0, 0, 0, 2, 0], [0, 0, 0, 2, 0, 0, 0, 0, 0, 9, 0],
[0, 0, 0, 0, 0, 9, 2, 0, 0], [0, 2, 0, 0, 0, 0, 0, 3, 2, 4], [4, 0, 0, 0, 0,
0, 0, 9, 0, 0], [0, 0, 3, 0, 0, 0, 0, 2, 0, 0], [0, 0, 0, 9, 0, 0, 3, 0, 0, 2]]
-> [[0, 2, 9, 0], [0, 2, 9, 3], [0, 2, 9, 3], [0, 2, 9, 3], [0, 2, 9, 3], [4, 2,
9, 3], [4, 2, 9, 3], [4, 2, 9, 3]]

```

Karşıolgusal: Şimdi, önceki girdiye döndürerek veya çevirerek müdahale ettiğimizi hayal edin.

```

[[0, 0, 9, 0, 4, 0, 0, 0, 3, 0], [0, 0, 3, 0, 9, 0, 9, 0, 9, 4], [0, 4, 0, 0, 0,
2, 0, 0, 0, 0], [0, 9, 4, 0, 9, 0, 0, 0, 2, 9], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 2, 0, 0, 2, 3, 0, 0], [9, 4, 0, 0, 9, 3, 0, 0, 0, 0], [0, 0, 0, 0, 0,
0, 2, 0, 9, 0], [9, 0, 0, 2, 0, 0, 9, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
-> [[0, 0, 9, 0], [0, 0, 9, 0], [0, 0, 9, 0], [0, 0, 9, 0], [0, 0, 9, 0], [0, 0,
9, 0], [0, 2, 9, 0], [4, 2, 9, 0], [4, 2, 9, 3], [4, 2, 9, 3], [4, 2, 9, 3], [4,
2, 9, 3]]

```

Örnek girdi-cıktı dizileri

```

[[0, 0, 0, 0, 0, 0, 4, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 4], [0, 0, 0, 2, 0,
0, 0, 0, 0, 0], [0, 4, 0, 0, 0, 9, 0, 0, 4, 0], [0, 0, 0, 0, 0, 3, 0, 0, 3, 0],
[0, 0, 0, 0, 0, 4, 2, 0, 0], [0, 2, 0, 0, 0, 0, 3, 0, 0, 0], [0, 0, 2, 0, 0,
0, 0, 0, 0, 0], [0, 3, 0, 0, 0, 0, 0, 4, 0, 0], [0, 0, 0, 0, 0, 9, 4, 0, 0, 0]]
-> [[4, 0, 0, 0], [4, 0, 0, 0], [4, 0, 0, 0], [4, 2, 0, 3], [4, 2, 0, 3], [4, 2,
9, 3], [4, 2, 9, 3]]

```

Karşıolgusal: Şimdi, önceki girdiye bazı renkleri değiştirerek müdahale ettiğimizi hayal edin.

```
[[[7, 7, 7, 7, 7, 7, 4, 7, 7, 7], [7, 7, 7, 7, 7, 7, 7, 7, 4, 7], [7, 7, 7, 2, 7,
7, 7, 7, 7, 7], [7, 4, 7, 7, 7, 9, 7, 7, 4, 7], [7, 7, 7, 7, 7, 3, 7, 7, 3, 7],
[7, 7, 7, 7, 7, 4, 2, 7, 7], [7, 2, 7, 7, 7, 7, 3, 7, 7, 7], [7, 7, 2, 7, 7,
7, 7, 7, 7, 7], [7, 3, 7, 7, 7, 7, 7, 4, 7, 7], [7, 7, 7, 7, 7, 9, 4, 7, 7, 7]]
-> [[4, 7, 7, 7], [4, 7, 7, 7], [4, 7, 7, 7], [4, 2, 7, 3], [4, 2, 7, 3], [4, 2,
9, 3], [4, 2, 9, 3]]]
```

Örnek girdi-çıktı dizileri

```

[[[0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0], [4, 0, 0, 0, 0, 9, 3, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0], [0, 0, 0, 0, 2, 0, 4, 0, 0, 0], [0, 3, 0, 0, 4, 0, 0, 0, 0, 2, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 2, 4, 2, 0, 0, 0], [0, 2, 0, 9, 3, 0, 9, 0, 3, 0], [0, 0, 0, 0, 0, 0, 4, 0, 0, 0], [0, 3, 4, 0, 4, 9, 4, 0, 0, 0]]]
-> [[4, 0, 0, 0], [4, 0, 0, 0], [4, 0, 0, 0], [4, 0, 0, 3], [4, 2, 0, 3], [4, 2, 9, 3], [4, 2, 9, 3], [4, 2, 9, 3], [4, 2, 9, 3]]

```

Karşıolgusal: Şimdi, önceki girdiye döndürerek veya çevirerek müdahale ettiğimizi hayal edin.

```

[[0, 4, 9, 0, 9, 0, 0, 0, 2, 0], [0, 0, 0, 0, 4, 0, 2, 0, 0, 0], [0, 0, 0, 2, 0,
0, 0, 0, 0, 0], [0, 0, 0, 4, 0, 0, 0, 0, 0, 0], [0, 0, 9, 0, 9, 0, 0, 0, 0, 4],
[9, 0, 0, 0, 2, 0, 0, 4, 9, 4], [0, 0, 0, 0, 0, 0, 0, 2, 0, 0], [0, 0, 9, 0, 0,
3, 0, 3, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 2, 0, 0, 0, 0, 3, 4, 0, 0]]
-> [[4, 0, 9, 0], [4, 2, 9, 0], [4, 2, 9, 0], [4, 2, 9, 0], [4, 2, 9, 3], [4, 2,
9, 3], [4, 2, 9, 3]]

```

Figure D.7: L3 prompt for program synthesis with six in-context examples.

Şekil D.7: Altı bağlam içi örnekle program sentezi için L3 istemi.

Prompt

You must solve the following causal discovery problem, where the cells in an input array are causal parents of cells in an output array. Both the inputs and outputs are 2D Python arrays of colored pixels. We provide example input-output pairs as demonstration. You must predict the causal function(s) that relate parent cells in the input to their children in the output. Be concise: do not explain your reasoning, and start your answer with 'The logical operators are'. Example input-output arrays:

Example input-output arrays:

```

[[4, 0, 4, 0, 0, 4, 4, 4, 4, 0], [4, 4, 0, 4, 4, 4, 0, 4, 0, 0], [0, 0, 4, 0, 0,
0, 4, 4, 4, 0], [4, 4, 4, 0, 4, 4, 0, 0, 0, 0], [4, 4, 0, 0, 4, 0, 0, 4, 4, 0],
[0, 0, 4, 4, 4, 0, 0, 4, 4, 4], [0, 4, 4, 0, 4, 4, 4, 4, 4, 0], [4, 0, 4, 4, 0,
0, 0, 4, 0, 0], [4, 4, 4, 4, 0, 4, 0, 4, 0, 4], [4, 4, 0, 4, 4, 0, 0, 0, 0, 4],
[0, 5, 0, 5, 0, 0, 5, 5, 5, 0], [0, 0, 5, 0, 5, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0,
5, 5, 5, 5, 5], [5, 0, 0, 5, 5, 5, 0, 5, 5], [5, 0, 0, 5, 0, 5, 0, 0, 0, 0],
[0, 5, 0, 5, 5, 0, 5, 0, 5, 5], [0, 0, 5, 0, 0, 5, 0, 5, 5, 5], [5, 0, 5, 0, 0,
0, 5, 5, 0, 0], [0, 0, 5, 5, 0, 0, 0, 0, 0, 0], [0, 5, 5, 5, 0, 5, 0, 0, 0, 5],
[2, 0, 0, 0, 2, 0, 2, 0, 0, 2], [2, 0, 2, 2, 2, 2, 0, 0, 0, 0], [2, 0, 2, 2, 0,
2, 2, 0, 2, 2], [2, 0, 0, 2, 2, 0, 0, 2, 2, 0], [0, 0, 2, 2, 2, 2, 2, 2, 0, 0],
[0, 2, 2, 2, 2, 0, 2, 0, 0, 0], [0, 0, 2, 0, 2, 0, 0, 2, 2, 0], [0, 0, 0, 2, 2,
0, 0, 2, 2, 2], [2, 2, 0, 0, 2, 2, 0, 0, 2, 2], [0, 2, 0, 2, 0, 0, 0, 2, 2, 2]
-> [[1, 0, 0, 0, 1, 0, 0, 1, 1, 1], [1, 0, 1, 1, 0, 1, 0, 0, 0, 0], [1, 0, 1, 1,
0, 1, 0, 1, 0, 1], [0, 0, 0, 1, 0, 1, 0, 1, 1, 0], [1, 0, 1, 1, 1, 1, 1, 0,
0], [0, 1, 1, 0, 0, 0, 1, 0, 1, 1], [0, 0, 0, 0, 1, 1, 0, 0, 0, 0], [1, 0, 1, 1,
1, 0, 0, 0, 1, 1], [1, 1, 1, 1, 1, 0, 0, 1, 1], [0, 0, 0, 0, 0, 0, 0, 1, 1],
0]

```

Example input-output arrays:

```

Example input-output arrays:
[[0, 4, 4, 4, 0, 4, 4, 0, 4, 4], [0, 4, 4, 4, 0, 0, 0, 4, 4, 4, 4], [0, 4, 0, 4, 4, 0, 0, 4, 4, 0, 0], [0, 0, 0, 4, 4, 0], [0, 0, 0, 0, 0, 4, 0, 0, 0, 4], [0, 4, 4, 0, 0, 4, 0, 4, 4, 4], [0, 4, 0, 4, 4, 4, 4, 0, 0, 0], [0, 0, 0, 0, 0, 4, 4, 4, 0, 0, 0], [0, 0, 0, 0, 4, 0, 4, 4, 0, 0], [0, 0, 4, 0, 0, 0], [0, 4, 0, 0, 4, 0, 4, 4, 0, 0], [4, 0, 4, 4, 4, 0, 4, 4, 0, 4], [0, 0, 0, 5, 5, 5, 0, 0, 5, 5], [0, 5, 0, 5, 0, 5, 0, 5, 0, 5], [0, 5, 5, 5, 5, 5, 0, 5, 0, 0], [0, 5, 5, 0, 0, 0, 5, 0, 5, 5], [0, 0, 5, 0, 5, 5, 5, 5, 0, 0], [0, 0, 5, 0, 5, 5, 5, 5, 5], [5, 5, 5, 0, 0, 0, 0, 0, 5, 5], [5, 5, 0, 0, 5, 0, 5, 5, 5], [0, 0, 5, 0, 0, 0, 0, 5, 5, 0], [5, 5, 5, 0, 5, 0, 0, 5, 5, 5], [0, 0, 2, 0, 0, 2, 0, 0, 0, 0], [2, 2, 2, 2, 0, 2, 2, 0, 0, 2], [0, 0, 2, 0, 0, 2, 0, 2, 2, 2], [2, 0, 0, 2, 2, 0, 2, 0, 2, 2], [0, 2, 2, 2, 0, 2, 0, 2, 2, 2], [0, 2, 0, 0, 2, 2, 2, 0, 0], [2, 0, 0, 0, 0, 0, 2, 0, 2], [2, 2, 0, 2, 0, 2, 2, 0], [2, 0, 0, 2, 0, 0, 2, 2, 2], [2, 0, 0, 2, 2, 0, 0, 2, 2], [0, 1, 0, 0, 1, 1, 0, 1, 1, 0], [1, 0, 1, 0, 0, 1, 1, 0, 1, 0], [0, 1, 0, 1, 0, 0, 0, 0, 1, 1], [1, 0, 0, 1, 1, 0, 1, 0, 1, 0], [0, 1, 0, 1, 0, 0, 0, 0, 1, 1], [0, 1, 0, 0, 1, 0, 0, 0, 1, 0], [1, 0, 0, 0, 0, 0, 0, 1, 0, 1], [1, 1, 0, 1, 0, 1, 1, 0, 1, 1], [1, 0, 0, 1, 0, 0, 0, 1, 1, 0], [0, 0, 1, 1, 0, 0, 0, 0, 1, 1]
]]]

```

- 1 -

İstem

Girdi dizisindeki hücrelerin çıktı dizisindeki hücrelerin nedensel ebeveynleri olduğu aşağıdaki nedensel keşif problemini çözmeliiniz. Hem girdiler hem de çıktılar, renkli piksellerden oluşan 2B Python dizileridir. Gösterim olarak örnek girdi-çıktı çiftleri sunuyoruz.

Girdideki ebeveyn hücreleri çıktıktaki çocuklarına bağlayan nedensel fonksiyon(lar)ı tahmin etmelisiniz. Kısa olun: Akıl yürütmenizi açıklamayın ve cevabınıza 'Mantıksal operatörler şunlardır' ile başlayın.

Örnek girdi-çıktı dizileri:

```

[[4, 0, 4, 0, 0, 4, 4, 4, 4, 0], [4, 4, 0, 4, 4, 4, 0, 4, 0, 0], [0, 0, 4, 0, 0,
0, 4, 4, 4, 0], [4, 4, 4, 0, 4, 4, 0, 0, 0, 0], [4, 4, 0, 0, 4, 0, 0, 4, 4, 0],
[0, 0, 4, 4, 4, 0, 0, 4, 4, 4], [0, 4, 4, 0, 4, 4, 4, 4, 4, 0], [4, 0, 4, 4, 0,
0, 0, 4, 0, 0], [4, 4, 4, 4, 0, 4, 0, 4, 0, 4], [4, 4, 0, 4, 4, 0, 0, 0, 0, 4],
[0, 5, 0, 5, 0, 0, 5, 5, 5, 0], [0, 0, 5, 0, 5, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0,
5, 5, 5, 5, 5], [5, 0, 0, 5, 5, 5, 0, 5, 5], [5, 0, 0, 5, 0, 5, 0, 0, 0, 0],
[0, 5, 0, 5, 5, 0, 5, 0, 5, 5], [0, 0, 5, 0, 0, 5, 0, 5, 5, 5], [5, 0, 5, 0, 0,
0, 5, 5, 0, 0], [0, 0, 5, 5, 0, 0, 0, 0, 0, 0], [0, 5, 5, 5, 0, 5, 0, 0, 0, 5],
[2, 0, 0, 0, 2, 0, 2, 0, 0, 2], [2, 0, 2, 2, 2, 2, 0, 0, 0, 0], [2, 0, 2, 2, 0,
2, 2, 0, 2, 2], [2, 0, 0, 2, 2, 0, 0, 2, 2, 0], [0, 0, 2, 2, 2, 2, 2, 2, 0, 0],
[0, 2, 2, 2, 2, 0, 2, 0, 0, 0], [0, 0, 2, 0, 2, 0, 0, 2, 2, 0], [0, 0, 0, 2, 2,
0, 0, 2, 2, 2], [2, 2, 0, 0, 2, 2, 0, 0, 2, 2], [0, 2, 0, 2, 0, 0, 0, 2, 2, 2]]
-> [[1, 0, 0, 0, 1, 0, 0, 1, 1, 1], [1, 0, 1, 1, 0, 1, 0, 0, 0, 0], [1, 0, 1, 1,
0, 1, 0, 1, 0, 1], [0, 0, 0, 1, 0, 1, 0, 1, 1, 0], [1, 0, 1, 1, 1, 1, 1, 0,
0], [0, 1, 1, 0, 0, 0, 1, 0, 1, 1], [0, 0, 0, 0, 1, 1, 0, 0, 0, 0], [1, 0, 1, 1,
1, 0, 0, 0, 1, 1], [1, 1, 1, 1, 1, 0, 0, 1, 1], [0, 0, 0, 0, 0, 0, 1, 1, 0, 0,
0, 1]

```

Örnek girdi çıktı dizileri:

Ornek girdi-çıktı dizileri:
[[0, 4, 4, 4, 0, 4, 4, 0, 4, 4], [0, 4, 4, 4, 0, 0, 4, 4, 4, 4], [0, 4, 0, 4, 4, 4, 0, 0, 4, 4, 0], [0, 0, 0, 0, 4, 0, 0, 0, 4, 4], [0, 4, 4, 0, 0, 4, 0, 4, 4, 4], [0, 4, 0, 4, 4, 4, 4, 0, 0, 0], [0, 0, 0, 0, 4, 4, 4, 4, 0, 0, 0], [0, 0, 0, 4, 0, 0, 4, 0, 0, 0], [0, 0, 0, 4, 0, 0, 0, 0, 0], [0, 4, 0, 0, 4, 0, 4, 4, 0, 0, 0], [4, 0, 4, 4, 4, 0, 4, 4, 0, 4, 4], [0, 0, 0, 5, 5, 5, 0, 0, 5, 5], [0, 5, 0, 5, 0, 5, 0, 5, 0, 5, 5], [0, 5, 5, 5, 5, 5, 0, 0, 5, 0], [0, 5, 5, 0, 0, 0, 5, 0, 5, 5], [0, 0, 5, 0, 5, 5, 5, 5, 0, 0], [0, 0, 5, 0, 5, 5, 5, 5, 5], [5, 5, 5, 0, 0, 0, 0, 0, 5, 5], [5, 5, 0, 0, 5, 5, 5, 5, 0, 0], [0, 0, 5, 0, 0, 0, 0, 0, 5, 5, 0], [5, 5, 5, 0, 5, 0, 0, 5, 5, 5], [0, 0, 2, 0, 0, 2, 0, 0, 0, 0, 0], [2, 2, 2, 2, 0, 2, 2, 0, 0, 2], [0, 0, 2, 0, 0, 2, 0, 2, 2, 2], [2, 0, 0, 2, 2, 0, 2, 0, 2, 2], [0, 2, 2, 2, 0, 2, 0, 2, 2, 2], [0, 2, 0, 0, 2, 2, 2, 0, 0, 0], [2, 0, 0, 0, 0, 0, 0, 2, 0, 2], [2, 2, 0, 2, 0, 2, 2, 0, 0, 2], [2, 0, 0, 2, 0, 0, 0, 2, 2, 2], [2, 0, 0, 2, 2, 0, 0, 2, 2, 0]]
-> [[0, 0, 1, 1, 0, 0, 0, 1, 1], [1, 0, 1, 0, 0, 1, 1, 1, 0, 0], [0, 1, 1, 1, 1, 0, 0, 1, 1, 1], [1, 0, 0, 1, 1, 0, 1, 0, 1, 0], [0, 1, 0, 1, 0, 0, 0, 0, 1, 1], [0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1], [1, 0, 0, 0, 0, 1, 0, 1, 0, 0], [1, 0, 0, 0, 0, 0, 0, 1, 0, 1], [1, 1, 0, 1, 0, 1, 1, 1, 0, 1], [1, 0, 0, 1, 0, 0, 0, 1, 1], [0, 0, 1, 1, 0, 0, 0, 0, 1, 1]]

1

Figure D.8: Excerpt of L1 prompt for logical reasoning. Four in-context examples were provided.

Sekil D.8: Mantıksal akıl yürütme için 11 isteminden bir alıntı. Dört bağlam içi örnek sağlanmıştır.

Prompt

You must solve the following causal discovery problem, where the cells in an input array are causal parents of cells in an output array. Both the inputs and outputs are 2D Python arrays of colored pixels. We provide example input-output pairs along with counterfactual examples, which represent interventions on the original examples. You must predict the causal function(s) that relate parent cells in the input to their children in the output. Be concise: do not explain your reasoning, and start your answer with ‘The logical operators are’.

Example input-output arrays:

Counterfactual: Now imagine that we intervened on the original input by changing some colors.

```

[[4, 4, 4, 4, 0, 0, 0, 4, 4, 4], [0, 0, 0, 4, 0, 0, 4, 4, 4, 0], [4, 0, 0, 4, 4, 4, 0, 4, 4, 4, 0], [0, 0, 0, 0, 4, 0, 0, 0, 0, 4], [4, 0, 0, 0, 0, 4, 0, 4, 4, 0], [0, 0, 4, 4, 0, 0, 0, 4, 4, 0], [4, 0, 4, 0, 0, 4, 4, 4, 0, 0], [0, 4, 0, 0, 0, 0, 4, 4, 4, 0], [0, 0, 4, 4, 0, 0, 0, 4, 4, 0], [4, 0, 0, 0, 0, 4, 4, 4, 4, 0], [0, 4, 0, 0, 0, 0, 4, 4, 4, 4, 0], [0, 0, 6, 0, 6, 6, 6, 6, 6, 6], [6, 0, 0, 0, 0, 6, 6, 6, 0, 0], [6, 0, 6, 6, 0, 0, 6, 0, 0, 6], [6, 0, 0, 6, 6, 0, 6, 0, 6, 0], [6, 0, 6, 6, 0, 0, 6, 6, 6, 6], [0, 6, 0, 6, 0, 6, 0, 6, 0, 0], [0, 6, 6, 0, 6, 6, 0, 0, 0], [6, 0, 2, 2, 2, 2, 0, 0, 2, 2], [0, 2, 2, 0, 2, 2, 0, 2, 2], [2, 2, 2, 2, 2, 0, 0, 0, 0], [0, 2, 0, 0, 0, 0, 2, 0, 0, 2], [2, 0, 0, 0, 0, 0, 0, 0, 0, 2], [0, 2, 0, 0, 0, 0, 2, 2, 2, 2], [2, 0, 2, 0, 2, 0, 2, 0, 2, 0], [0, 0, 2, 2, 0, 0, 0, 0, 0, 0], [0, 2, 2, 0, 0, 0, 2, 0, 2, 0], [0, 0, 2, 2, 2, 2, 0, 2, 0, 0], [0, 0, 0, 1, 1, 1, 0, 1, 0, 0], [0, 1, 1, 0, 1, 1, 0, 1, 1, 1], [0, 1, 1, 0, 1, 0, 0, 0, 0, 0], [0, 1, 0, 0, 1, 0, 1, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [0, 1, 1, 1, 0, 0, 1, 0, 0, 1], [1, 0, 1, 0, 1, 1, 1, 1, 1, 0], [0, 1, 1, 1, 1, 0, 1, 0, 0, 1], [0, 0, 1, 1, 1, 0, 0, 1, 0, 1], [0, 0, 0, 1, 1, 1, 1, 0, 0, 1], [0, 0, 0, 0, 1, 1, 0, 0, 1, 1], [0, 0, 1, 1, 0, 0, 1, 0, 1, 1], [0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1]]

```

111

Istem

Girdi dizisindeki hücrelerin çıktı dizisindeki hücrelerin nedensel ebeveynleri olduğu aşağıdaki nedensel keşif problemini çözmeliiniz. Hem girdiler hem de çıktılar, renkli piksellerden oluşan 2B Python dizileridir. Orijinal örneklerle müdahaleleri temsil eden karşılıklı örneklerle birlikte örnek girdi-çıktı çiftleri sunuyoruz. Girdideki ebeveyn hücreleri çıktıdaki çocuklarına bağlayan nedensel fonksiyon(lar) tahmin etmelisiniz. Kısa olun: Akıl yürütmenizi açıklamayın ve cevabınıza 'Mantıksal operatörler şunlardır' ile başlayın.

Örnek girdi-çıktı dizileri:

```

[[0, 4, 0, 0, 0, 4, 4, 0, 0, 0], [0, 4, 4, 0, 4, 0, 0, 4, 4, 4], [0, 0, 0, 0, 0, 0, 0,
4, 4, 4, 4, 4], [4, 4, 0, 4, 0, 0, 0, 4, 0, 0], [4, 0, 0, 4, 0, 0, 0, 4, 0, 0],
[0, 4, 0, 4, 0, 4, 0, 0, 4, 0], [0, 4, 0, 0, 0, 4, 4, 0, 4, 4], [4, 4, 4, 0, 4,
4, 4, 0, 0, 4], [0, 4, 4, 0, 0, 4, 0, 4, 4, 4], [4, 4, 4, 0, 0, 0, 0, 4, 0, 4],
[5, 0, 5, 5, 5, 5, 5, 0, 5, 0], [5, 5, 5, 5, 0, 5, 0, 0, 0, 5], [0, 0, 0, 0, 0,
0, 0, 5, 5, 5], [0, 0, 0, 0, 5, 5, 0, 0, 5, 0], [0, 0, 0, 0, 0, 5, 5, 5, 5],
[5, 5, 5, 0, 0, 0, 5, 0, 0, 5], [0, 0, 0, 5, 0, 5, 0, 0, 0, 5], [5, 5, 0, 0, 0,
0, 0, 5, 5, 0], [5, 5, 5, 0, 5, 0, 5, 0, 0, 5], [0, 5, 5, 0, 5, 0, 5, 0, 5, 5],
[2, 0, 0, 2, 0, 0, 0, 2, 0, 2], [2, 2, 0, 2, 2, 2, 2, 2, 2, 2], [2, 0, 0, 0, 0,
2, 0, 0, 2, 0], [2, 2, 2, 0, 2, 0, 0, 2, 0, 2], [2, 2, 0, 0, 0, 0, 0, 2, 0, 2],
[2, 2, 2, 0, 2, 0, 2, 2, 0, 0], [0, 2, 0, 0, 0, 2, 2, 0, 2, 0], [2, 0, 0, 2, 2,
0, 0, 2, 2, 0], [0, 2, 0, 0, 2, 2, 2, 2, 2, 0], [0, 2, 0, 2, 0, 2, 0, 2, 2, 2]
-> [[1, 0, 0, 1, 0, 1, 1, 0, 1], [1, 0, 1, 1, 1, 1, 1, 1, 0], [1, 0, 0, 0,
0, 1, 0, 1, 0, 1], [1, 1, 1, 0, 1, 0, 0, 1, 0, 1], [1, 1, 0, 0, 0, 1, 1, 1,
0], [1, 0, 1, 0, 1, 0, 1, 1, 0, 0], [0, 1, 0, 0, 0, 0, 1, 0, 1, 1], [0, 1, 0, 1,
1, 0, 0, 1, 1, 0], [0, 0, 1, 0, 1, 1, 1, 1, 1, 1], [0, 0, 1, 1, 0, 1, 0, 1, 1,
0]]

```

Karşıolgusal: Şimdi, bazı renkleri değiştirerek orijinal girdiye müdahale ettiğimizi varsayıyalım.

```

[[4, 4, 4, 4, 0, 0, 0, 4, 4, 4], [0, 0, 0, 4, 0, 0, 4, 4, 4, 0], [4, 0, 0, 4, 4,
4, 0, 4, 4, 0], [0, 0, 0, 4, 0, 0, 0, 0, 4], [4, 0, 0, 0, 0, 4, 0, 4, 4, 0], [0, 0, 4, 4, 0, 0, 0, 4, 4, 0], [4, 0, 4, 0, 0, 4, 4, 4, 0, 0], [0, 4, 0, 0, 0, 0, 4, 4, 0], [0, 0, 0, 0, 4, 0, 4, 0, 0, 4], [4, 0, 4, 0, 4, 4, 4, 0, 0], [0, 0, 6, 0, 6, 6, 6, 6, 6, 6], [6, 0, 0, 0, 0, 6, 6, 6, 0, 0], [6, 0, 6, 6, 0, 0, 6, 0, 6, 6], [6, 0, 0, 6, 6, 0, 0, 6, 0, 6], [6, 0, 0, 6, 6, 0, 6, 0, 6, 0], [6, 0, 6, 6, 0, 0, 6, 6, 6, 6], [0, 6, 0, 6, 0, 6, 0, 6, 0, 0], [0, 6, 6, 0, 6, 6, 0, 0, 6], [6, 0, 6, 6, 0, 6, 0, 0, 6, 0], [0, 0, 2, 2, 2, 2, 0, 0, 2, 2], [0, 2, 2, 0, 2, 2, 2, 0, 2, 2], [2, 2, 2, 2, 2, 0, 0, 0, 0, 0], [0, 2, 0, 0, 0, 0, 2, 0, 0, 2], [2, 0, 0, 0, 0, 0, 0, 0, 0, 2], [0, 2, 0, 0, 0, 0, 2, 2, 2, 2], [2, 0, 2, 0, 2, 0, 2, 0, 2, 0], [0, 0, 2, 2, 0, 2, 0, 0, 0, 0], [0, 2, 2, 0, 0, 0, 2, 0, 2, 0], [0, 0, 2, 2, 2, 2, 0, 2, 0, 0], [0, 0, 0, 1, 1, 1, 0, 1, 0, 0], [0, 1, 1, 0, 1, 1, 0, 1, 1, 1], [0, 1, 1, 0, 1, 0, 0, 0, 0, 0], [0, 1, 0, 0, 1, 0, 1, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 1], [0, 1, 1, 1, 0, 0, 1, 0, 0, 1], [1, 0, 1, 0, 1, 1, 1, 1, 1, 0], [0, 1, 1, 1, 1, 0, 1, 0, 0, 1], [0, 1, 1, 1, 0, 0, 1, 0, 1, 1], [0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1]]

```

1

Figure D.9: Excerpt of L3 prompt for logical reasoning. Four in-context examples were provided

Sekil D 9: Mantıksal akıl yürütme için 13 isteminin bir kesiti. Dört bağılam içi örnek sağlanmıştır.

```
# For ARC-style input/output:
if __name__ == "__main__":
    import sys, json
    data = json.load(sys.stdin)
    inp = data["input"]
    out = solve(inp)
    print(json.dumps(out))
```

Şekil D.10: Evidence that o4-mini may have been fine-tuned for ARC-like program synthesis (“For ARC-style input/output:... out = solve(inp)”), excerpted from a response to a CausalARC query that did not mention ARC in-context.

Response

Replicate 1: The logical operators are: For each output cell at ‘(row, col)’, let ‘A = (input[row][col] != 0)’, ‘B = (input[row + 10][col] != 0)’, and ‘C = (input[row + 20][col] != 0)’. The output cell ‘output[row][col]’ is ‘1’ if ‘(A AND B) XOR C’ is true, and ‘0’ otherwise.

Replicate 2: The logical operators are ‘((x AND y) XOR z)’, where ‘x’, ‘y’, and ‘z’ are boolean values indicating if the corresponding input cell (from the first, second, and third 15x15 input arrays, respectively) is non-zero. The output cell is ‘2’ if the expression is true, and ‘0’ if false.

Replicate 3: The logical operators are XOR and AND. For each output cell ‘O[r][c]’, its value is determined by three input cells: ‘P1 = I[r][c]’, ‘P2 = I[r+10][c]’, and ‘P3 = I[r+20][c]’. The output value is ‘1’ if ‘(P3 != 0) XOR ((P1 != 0) AND (P2 != 0))’ is true, and ‘0’ otherwise.

Şekil D.11: Three correct replicate responses from Gemini 2.5 Flash for the causal discovery with logical reasoning task. The causal parents and logical operators were correctly identified.

Response

Replicate 1: The logical operators are AND and XOR.

Replicate 3: The logical operators are AND between the first and second input grids, then XOR with the third input grid.

Replicate 4: The logical operators are AND and XOR, applied pointwise as (color4 AND color5) XOR color2.

Şekil D.12: Three correct replicate responses from Grok 4 for the causal discovery with logical reasoning task. Logical operators were correctly identified and causal parents were explicitly declared in some cases.

```
# ARC tarzı giriş/çıkış için:
if __name__ == "__main__":
    import sys, json
    data = json.load(sys.stdin)
    inp = data["input"]
    out = solve(inp)
    print(json.dumps(out))
```

Şekil D.10: o4-mini'nin ARC benzeri program sentezi için ince ayarlandığına dair kanıt (“ARC tarzı giriş/çıkış için:... out = solve(inp)”), ARC'den bağlam içinde bahsedilmeyen bir CausalARC sorgusuna verilen yanıtta alıntıdır.

Yanıt

Çoğaltma 1: Mantıksal operatörler şunlardır: ‘(satır, sütun)’ konumundaki her çıktı hücresi için, ‘A = (input[satır][sütun] != 0)’, ‘B = (input[satır + 10][sütun] != 0)’ ve ‘C = (input[satır + 20][sütun] != 0)’ olsun. Çıktı hücresi ‘output[row][col]’, ‘(A AND B) XOR C’ doğru ise ‘1’, aksi takdirde ‘0’ değerini alır.

Çoğaltma 2: Mantıksal operatörler ‘((x AND y) XOR z)’ şeklindedir; burada ‘x’, ‘y’ ve ‘z’, ilgili giriş hücresinin (sırasıyla birinci, ikinci ve üçüncü 15x15 giriş dizilerinden) sıfır olmayan bir değere sahip olup olmadığını gösteren Boolean değerlerdir. İfade doğru ise çıktı hücresi ‘2’, yanlış ise ‘0’ değerini alır.

Çoğaltma 3: Mantıksal operatörler XOR ve AND'dir. Her bir çıktı hücresi ‘O[r][c]’ için değeri üç giriş hücresi tarafından belirlenir: ‘P1 = I[r][c]’, ‘P2 = I[r+10][c]’ ve ‘P3 = I[r+20][c]’. Çıktı değeri, ‘(P3 != 0) XOR ((P1 != 0) AND (P2 != 0))’ doğru ise ‘1’, aksi takdirde ‘0’ değerini alır.

Şekil D.11: Mantıksal akıl yürütme görevine sahip nedensel keşif için Gemini 2.5 Flash'tan üç doğru çoğaltma yanıtı. Nedensel ebeveynler ve mantıksal operatörler doğru bir şekilde tanımlanmıştır.

Yanıt

Çoğaltma 1: Mantıksal operatörler VE ve XOR'dur.

Çoğaltma 3: Mantıksal operatörler, birinci ve ikinci girdi izgaraları arasında VE, ardından üçüncü girdi izgarası ile XOR'dur.

Çoğaltma 4: Mantıksal operatörler VE ve XOR olup, (renk4 VE renk5) XOR renk2 olarak noktasal uygulanmıştır.

Şekil D.12: Mantıksal akıl yürütme görevine sahip nedensel keşif için Grok 4'ten üç doğru çoğaltma yanıtı. Mantıksal operatörler doğru bir şekilde tanımlanmış ve bazı durumlarda nedensel ebeveynler açıkça belirtilmiştir.

D.2 Experimental Results

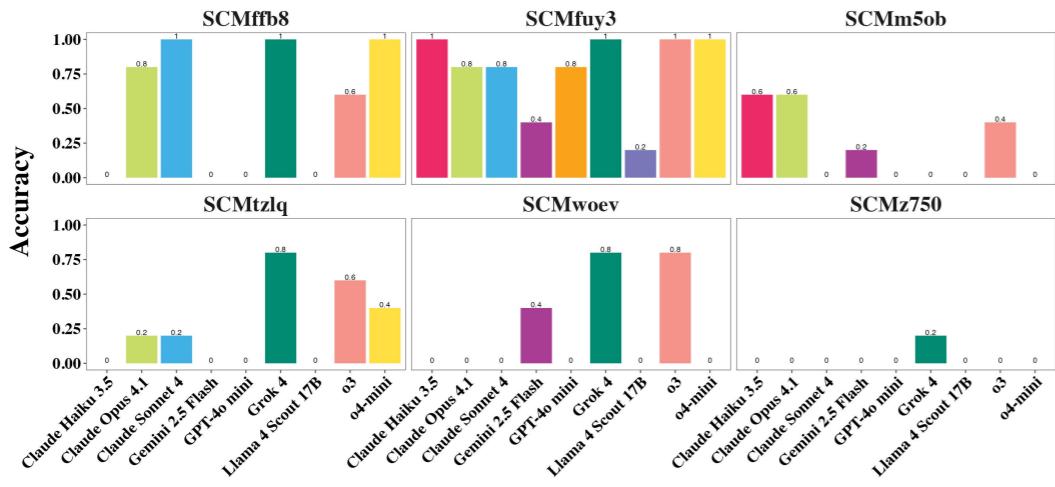


Figure D.13: **Counterfactual reasoning** on counting, extension, and ordering tasks (Figure D.2). Scores are over five random prompt samples.

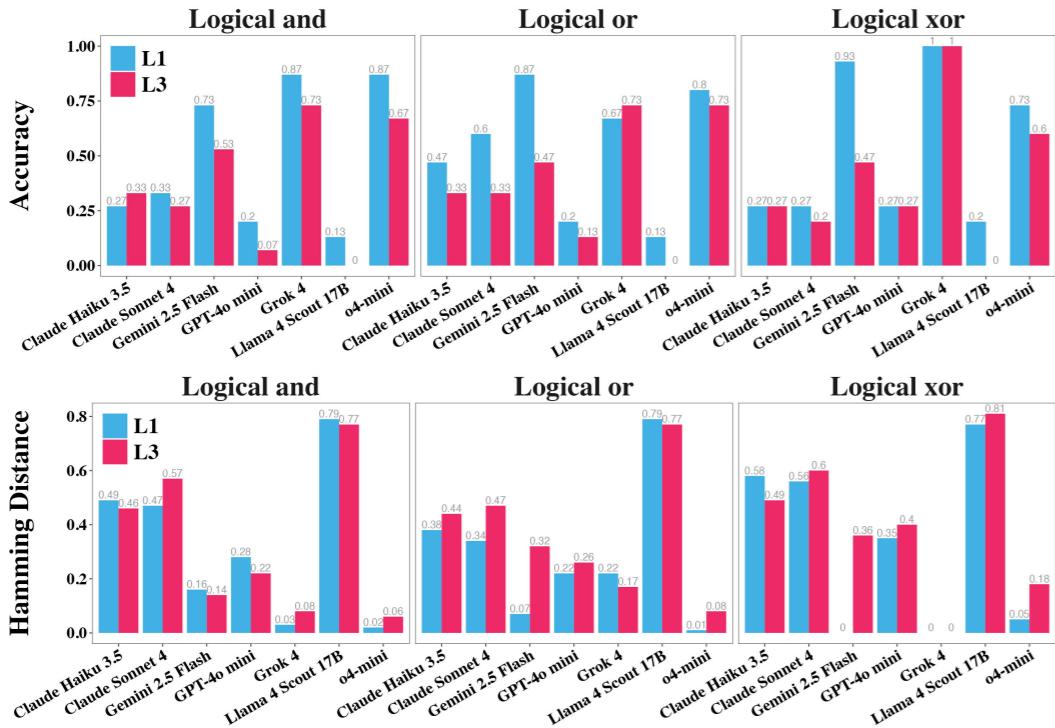
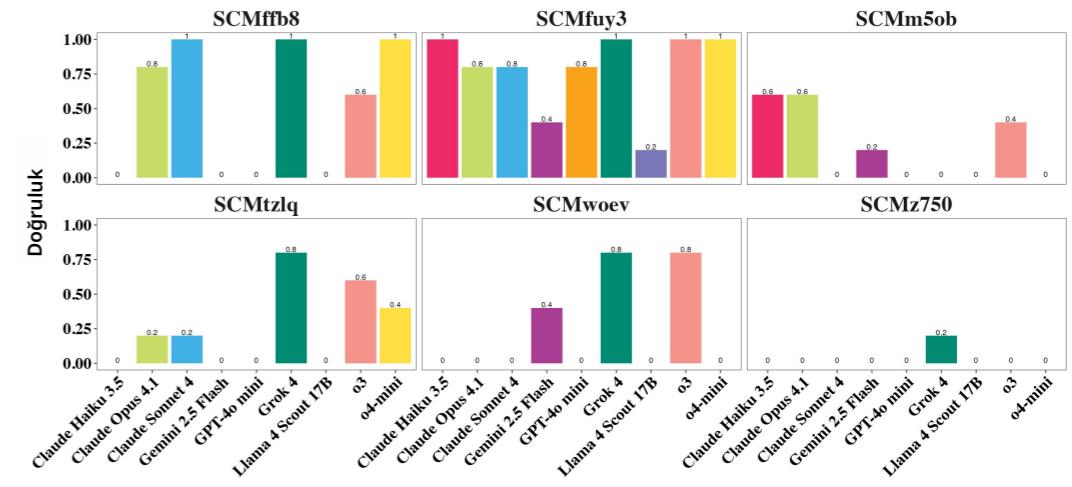
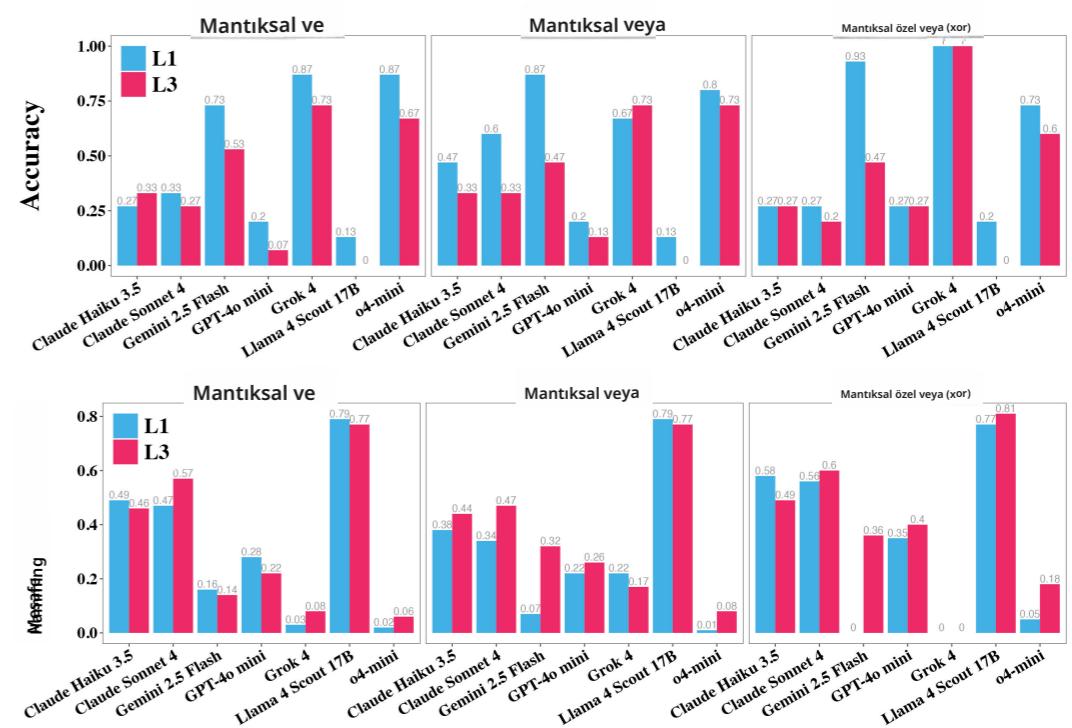


Figure D.14: **Counterfactual reasoning** results on variants of CausalARC task SCMdky5. Scores are over five random prompt samples. Raw data are in Table D.2. Metrics are accuracy (fraction tasks answered correctly; values between 0 and 1; higher is better) and Hamming distance (values between 0 and 1; lower is better). Accurate prediction corresponds to a Hamming distance of 0.

D.2 Deneysel Sonuçlar



Şekil D.13: Sayma, uzatma ve sıralama görevlerinde (Şekil D.2) karşıolgusal akıl yürütme. Puanlar, beş rastgele istem örneği üzerinden alınmıştır.



Şekil D.14: CausalARC görevi SCMdky5 varyantlarında karşıolgusal akıl yürütme sonuçları. Puanlar, beş rastgele istem örneği üzerinden alınmıştır. Ham veriler Tablo D.2'de sunulmuştur. Metrikler doğruluk (doğru cevaplanan görevlerin oranı; değerler 0 ile 1 arasında; yüksek değer daha iyidir) ve Hamming mesafesi (değerler 0 ile 1 arasında; düşük değer daha iyidir). Doğru tahmin, 0'luk bir Hamming mesafesine karşılık gelir.

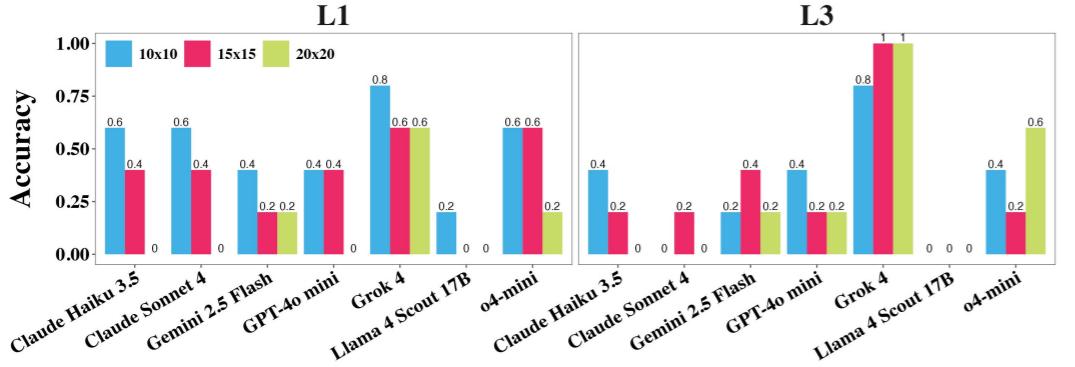


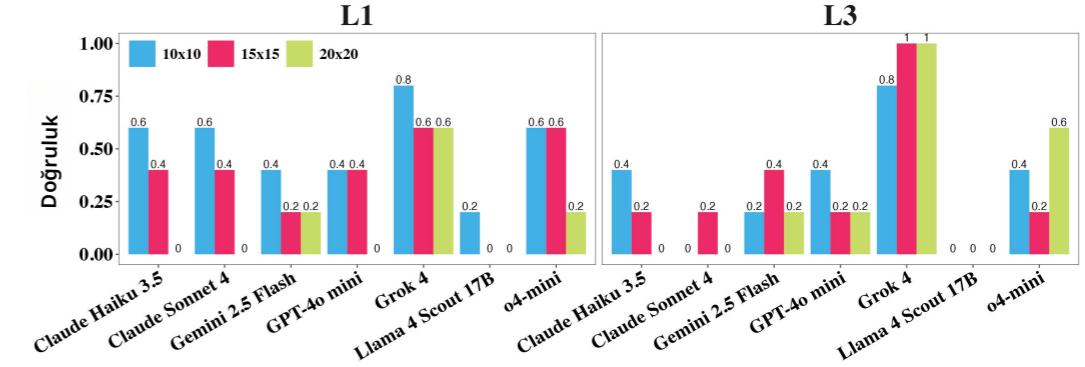
Figure D.15: **Counterfactual reasoning** results on variants of CausalARC task SCMtcbq, where logical operators compose (*xor* after *and*). Accuracy was measured with respect to scaling the dimensionality of the output array (10×10 , 15×15 , 20×20). Scores are over five random prompt samples. Raw data are provided in Table D.3.

	LOGICAL AND		LOGICAL OR		LOGICAL XOR			
	L1		L3		L1		L3	
	$\checkmark \uparrow$	HD \downarrow	$\checkmark \uparrow$	HD \downarrow	$\checkmark \uparrow$	HD \downarrow	$\checkmark \uparrow$	HD \downarrow
GPT-4o mini	0.20	0.28 (0.24)	0.07	0.22 (0.15)	0.20	0.22 (0.16)	0.13	0.26 (0.17)
o4-mini	0.87	0.02 (0.07)	0.67	0.06 (0.12)	0.80	0.01 (0.03)	0.73	0.08 (0.17)
Claude Haiku 3.5	0.27	0.49 (0.43)	0.33	0.46 (0.41)	0.47	0.38 (0.45)	0.33	0.44 (0.42)
Claude Sonnet 4	0.33	0.47 (0.45)	0.27	0.57 (0.43)	0.60	0.34 (0.47)	0.33	0.27 (0.46)
Grok 4	0.87	0.03 (0.09)	0.73	0.08 (0.13)	0.67	0.22 (0.33)	0.73	0.17 (0.30)
Gemini 2.5 Flash	0.73	0.16 (0.34)	0.53	0.14 (0.18)	0.87	0.07 (0.25)	0.47	0.32 (0.38)
Llama 4 Scout 17B	0.13	0.79 (0.37)	0.0	0.77 (0.33)	0.13	0.79 (0.37)	0.0	0.81 (0.28)

Table D.2: **Counterfactual reasoning** results on CausalARC logical reasoning task SCMdky5 (Figure D.3), where causal functions are a single logical operator. Scores are over five random prompt samples. Results are disaggregated by prompt formulation: L1 prompts include only L1 demonstrations (Figure D.4), while L3 prompts include alternating L1 and L3 demonstrations (Figure D.5). Both formulations feature a counterfactual test case. Metrics are accuracy (\checkmark ; values between 0 and 1; higher is better) and Hamming distance (HD, mean and standard deviation; values between 0 and 1; lower is better). Best performance per column is in bold.

	COMPOSITION		ALTERNATION	
	L1		L3	
	$\checkmark \uparrow$	HD \downarrow	$\checkmark \uparrow$	HD \downarrow
GPT-4o mini	0.27	0.34 (0.25)	0.27	0.34 (0.24)
o4-mini	0.47	0.18 (0.23)	0.40	0.19 (0.2)
Claude Haiku 3.5	0.33	0.51 (0.42)	0.20	0.57 (0.36)
Claude Sonnet 4	0.33	0.45 (0.46)	0.07	0.65 (0.38)
Grok 4	0.67	0.06 (0.13)	0.93	0.02 (0.06)
Gemini 2.5 Flash	0.27	0.48 (0.45)	0.27	0.58 (0.42)
Llama 4 Scout 17B	0.07	0.74 (0.39)	0.0	0.86 (0.28)

Table D.3: **Counterfactual reasoning** results on CausalARC logical reasoning tasks, where causal functions are multiple logical operators. Mixed operators include two cases: (1) task SCMu3am, where causal functions alternate by row of the input array (*or*, *and*); and (2) task SCMtcbq, where logical operators compose (*xor* after *and*). Scores are over five random prompt samples. Results are disaggregated by prompt formulation: L1 prompts include only L1 demonstrations (Figure D.4), while L3 prompts include alternating L1 and L3 demonstrations (Figure D.5). Both formulations feature a counterfactual test case. Metrics are accuracy (\checkmark ; higher is better) and Hamming distance (HD, mean and standard deviation; lower is better). Best performance per column is in bold.



Şekil D.15: CausalARC görevi SCMtcbq varyantlarında karşıolgusal akıl yürütme sonuçları, mantıksal operatörlerin (*xor*, *and* ten sonra) birleştirildiği durumu göstermektedir. Doğruluk, çıktı dizisinin boyutluluğunun ölçülenmesine göre ölçülmüştür (10×10 , 15×15 , 20×20). Puanlar, beş rastgele istem üzerinden elde edilmiş örnekleri. Ham veriler Tablo D.3'te sunulmuştur.

	MANTIKSAL VE		MANTIKSAL Veya		MANTIKSAL XOR	
	L1		L3		L1	
	$\checkmark \uparrow$	HD \downarrow	$\checkmark \uparrow$	HD \downarrow	$\checkmark \uparrow$	HD \downarrow
GPT-4o mini	0.20	0.28 (0.24)	0.07	0.22 (0.16)	0.20	0.22 (0.16)
o4-mini	0.87	0.02 (0.07)	0.67	0.06 (0.12)	0.80	0.01 (0.03)
Claude Haiku 3.5	0.27	0.49 (0.43)	0.33	0.46 (0.41)	0.47	0.38 (0.45)
Claude Sonnet 4	0.33	0.47 (0.45)	0.27	0.57 (0.43)	0.60	0.34 (0.47)
Grok 4	0.87	0.03 (0.09)	0.73	0.08 (0.13)	0.67	0.22 (0.33)
Gemini 2.5 Flash	0.73	0.16 (0.34)	0.53	0.14 (0.18)	0.87	0.07 (0.25)
Llama 4 Scout 17B	0.13	0.79 (0.37)	0.0	0.77 (0.33)	0.13	0.79 (0.37)

Tablo D.2: CausalARC mantıksal akıl yürütme görevindeki karşıolgusal akıl yürütme sonuçları SCMdky5 (Şekil D.3), burada nedensel fonksiyonlar tek bir mantıksal operatördür. Puanlar, beş rastgele istem üzerinde alınmıştır. Sonuçlar, istem formülasyonuna göre ayrıstırılmıştır: L1 istemleri yalnızca L1 gösterimlerini içerken (Şekil D.4), L3 istemleri dönüşümlü olarak L1 ve L3 gösterimlerini içerir (Şekil D.5). Her iki formülasyon da karşıolgusal bir test durumu sunmaktadır. Metrikler doğruluk (\checkmark ; değerler 0 ile 1 arasında değişir; yüksek değerler daha iyidir) ve Hamming mesafesidir (HD, ortalama ve standart sapma; değerler 0 ile 1 arasında değişir; düşük değerler daha iyidir). Her sütun için en iyi performans kalın yazı tipiyle belirtilmiştir.

	BİLEŞİM		DEĞİŞİM	
	L1		L3	
	$\checkmark \uparrow$	HD \downarrow	$\checkmark \uparrow$	HD \downarrow
GPT-4o mini	0.27	0.34 (0.25)	0.27	0.34 (0.24)
o4-mini	0.47	0.18 (0.23)	0.40	0.19 (0.2)
Claude Haiku 3.5	0.33	0.51 (0.42)	0.20	0.57 (0.36)
Claude Sonnet 4	0.33	0.45 (0.46)	0.07	0.65 (0.38)
Grok 4	0.67	0.06 (0.13)	0.93	0.02 (0.06)
Gemini 2.5 Flash	0.27	0.48 (0.45)	0.27	0.58 (0.42)
Llama 4 Scout 17B	0.07	0.74 (0.39)	0.0	0.86 (0.28)

Tablo D.3: CausalARC mantıksal akıl yürütme görevlerinde karşıolgusal akıl yürütme sonuçları; burada nedensel fonksiyonlar, birden fazla mantıksal operatördür. Karma operatörler iki durumu içerir: (1) nedensel fonksiyonların girdi dizisinin satırına göre değiştiği görev SCMu3am (*or*, *and*); ve (2) mantıksal operatörlerin birleştiği görev SCMtcbq (*and*'den sonra *xor*). Puanlar, beş rastgele istem üzerinde alınmıştır. Sonuçlar, istem formülasyonuna göre ayrıstırılmıştır: L1 istemleri yalnızca L1 gösterimlerini içerken (Şekil D.4), L3 istemleri dönüşümlü olarak L1 ve L3 gösterimlerini içerir (Şekil D.5). Her iki formülasyon da karşıolgusal bir test durumu sunmaktadır. Metrikler doğruluk (\checkmark ; daha yüksek daha iyidir) ve Hamming mesafesi (HD, ortalama ve standart sapma; daha düşük daha iyidir). Her sütun için en iyi performans kalın yazı tipiyle belirtilmiştir.

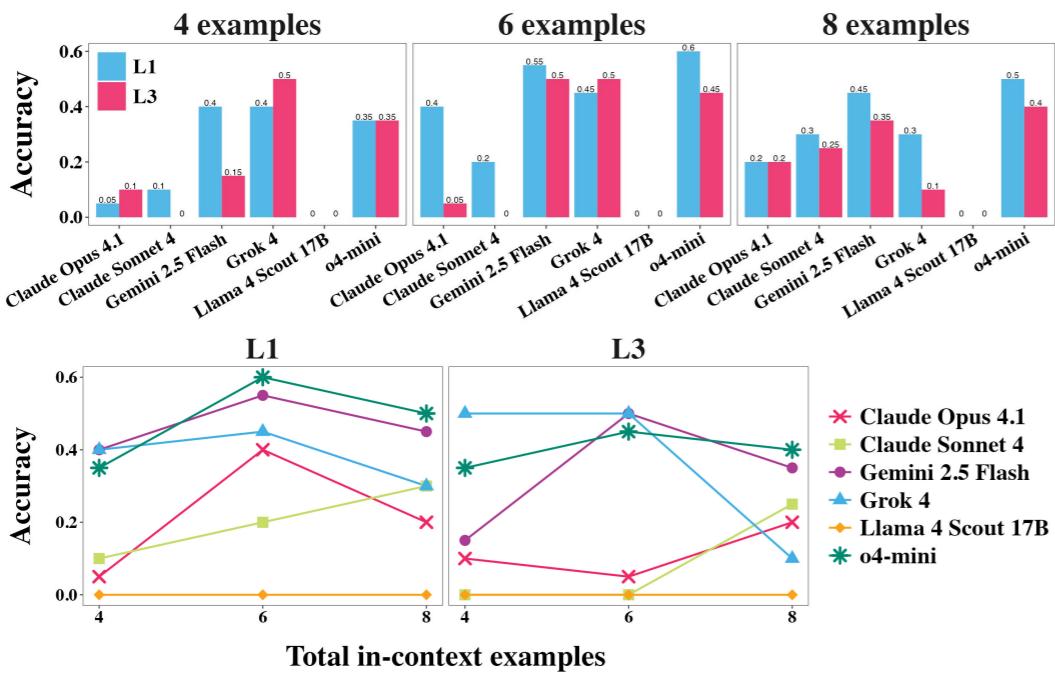
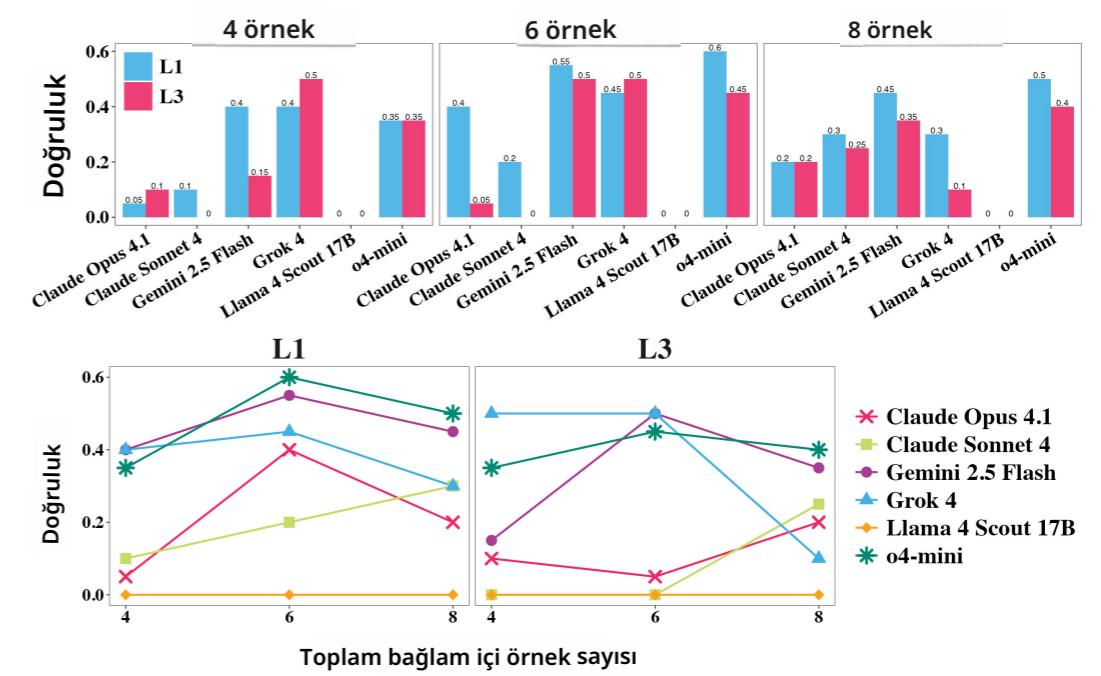


Figure D.16: **Program synthesis** accuracy by total in-context examples. For each of four tasks, five prompts were sampled per level of the PCH. Accuracy was measured with respect to the output array generated by passing a test input array to the LM-generated Python method. Tasks were sampled from SCMM5ob (counting), SCMe5t (counting), SCMFwpq (extension), and SCMZ750 (extension).



Şekil D.16: Program sentezi doğruluğunun toplam bağlam içi örneklerde göre değişimi. Dört görevin her biri için, PCH'nin her seviyesinden beş istem örneklandı. Doğruluk, bir test girdi dizisini DM tarafından oluşturulan Python metoduna geçirerek üretilen çıktı dizisine göre ölçüldü. Görevler SCMM5ob (sayma), SCMe5t (sayma), SCMFwpq (uzatma) ve SCMZ750 (uzatma) örneklerinden alınmıştır.

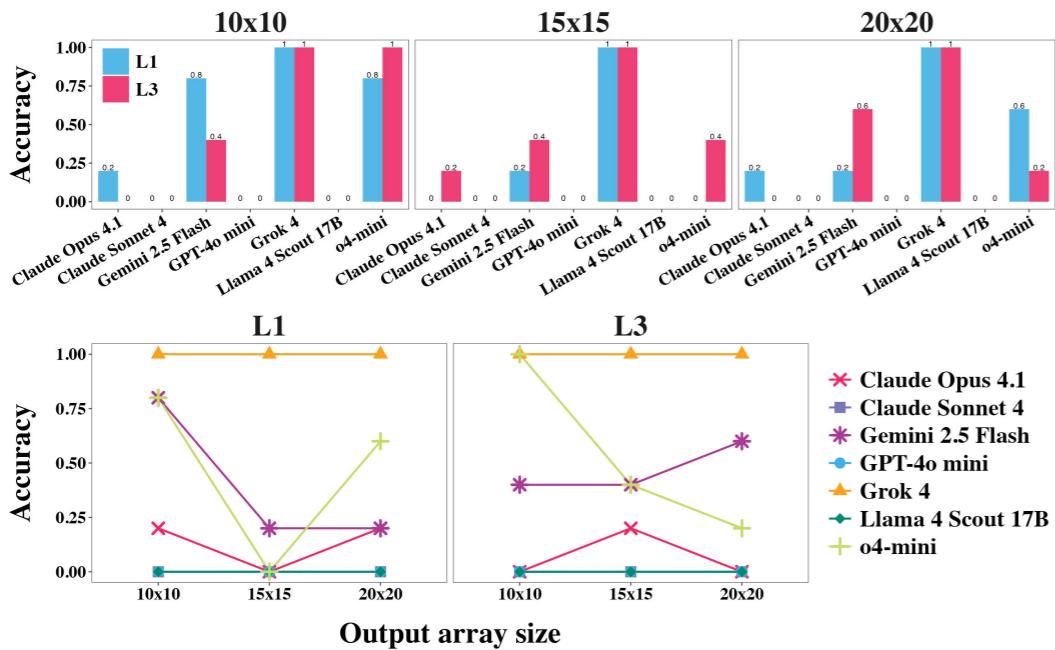
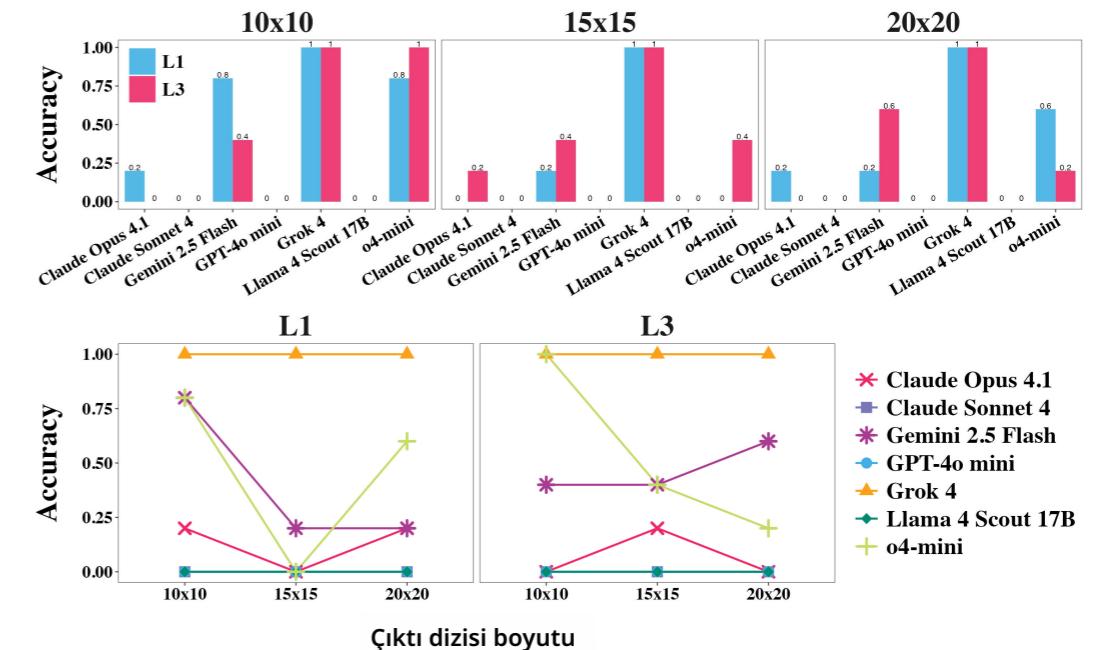


Figure D.17: **Causal discovery with logical reasoning** accuracy by scaling complexity with respect to output array size. For each of three CausalARC tasks sampled from task SCMtcbq (where logical operators compose *xor* after *and*), five prompts were sampled per level of the PCH. The LM was asked to predict the logical operators acting as causal functions, which requires knowledge of the causal parents of each output cell as a prerequisite.



Şekil D.17: Mantıksal akıl yürütme ile nedensel keşif doğruluğunun çıktı dizisi boyutuna göre değişimi. SCMtcbq görevinden örneklenen üç CausalARC görevi (mantıksal operatörlerin ve sonra veya oluşturduğu yer) için, PCH'nin her seviyesinden beş istem örneklandı. DM'den, nedensel fonksiyonlar olarak görev yapan mantıksal operatörleri tahmin etmesi istendi; bu, önkoşul olarak her çıktı hücresinin nedensel ebeveynleri hakkında bilgi gerektirir.

n	LOGICAL AND	LOGICAL OR	LOGICAL XOR
1k	1.0	0.0	40.0
5k	1.0	0.0	42.0
10k	0.0	0.0	40.0

Table D.4: Causal discovery with PC algorithm [39] on CausalARC task SCMdky5. Distributions of L1 arrays were sampled at $n = 1000$, $n = 5000$, and $n = 10000$ observations per distribution. Arrays were flattened before being passed to PC. Reported values are the structural Hamming distance for the predicted causal graph with respect to the true graph as data sample size increases. A structural Hamming distance of 0 indicates that the output graph was identical to the true graph. PC algorithm was performed with chi-square conditional independence tests ($\alpha = 0.01$) using the causal-learn Python package (<https://causal-learn.readthedocs.io/en/latest/>).

n	MANTIKSAL VE MANTIKSAL VEYA MANTIKSAL XOR
1k	1.0
5k	1.0
10k	0.0

Tablo D.4: CausalARC görevi SCMdky5 üzerinde PC algoritması [39] ile nedensel keşif. L1 dizi-lerinin dağılımları, her dağılım için $n = 1000$, $n = 5000$ ve $n = 10000$ gözleme örneklendi. Diziler PC'ye geçirilmeden önce düzeltirildi. Bildirilen değerler, veri örnek boyutu arttıkça tahmin edilen nedensel grafik için gerçek graf'e yapisal Hamming mesafesidir. 0'lık bir yapisal Hamming mesafesi, çıktı grafının gerçek grafle aynı olduğunu gösterir. PC algoritması, causal-learn Python paketi (<https://causal-learn.readthedocs.io/en/latest/>) kullanılarak ki-kare koşullu bağımsızlık testleri ($\alpha = 0.01$) ile gerçekleştirildi.

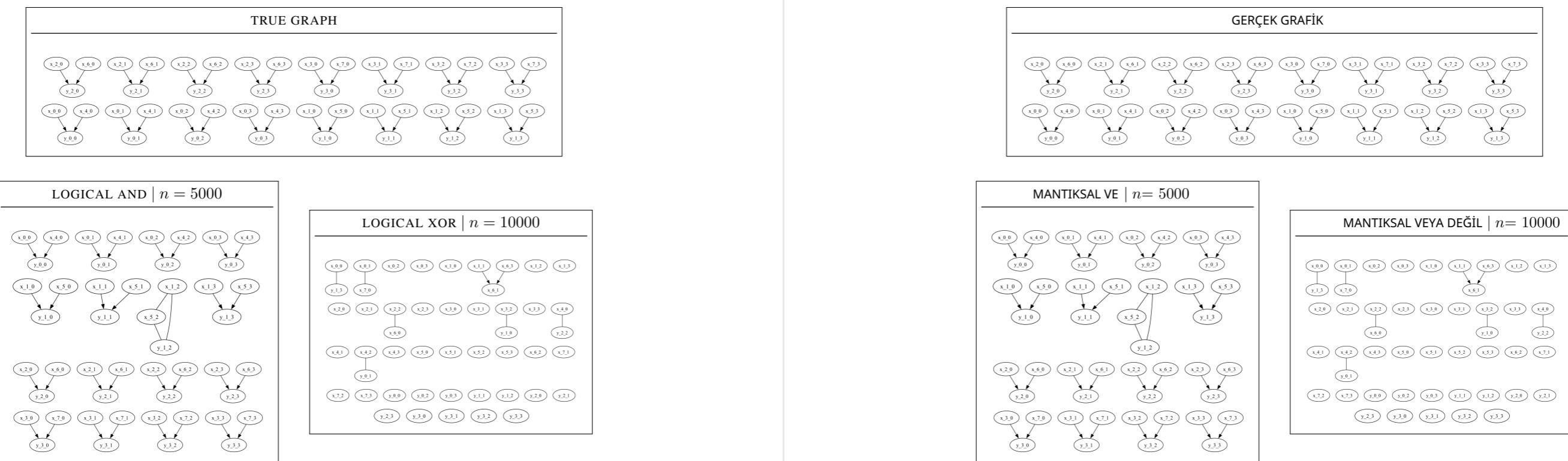


Figure D.18: Causal graphs predicted by PC algorithm [39] for CausalARC task SCMdky5. PC algorithm was performed with chi-square conditional independence tests ($\alpha = 0.01$) using the causal-learn Python package (<https://causal-learn.readthedocs.io/en/latest/>). Data sample size is denoted by n .

Şekil D.18: CausalARC görevi SCMdky5 için PC algoritması [39] tarafından tahmin edilen nedensel grafikler. PC algoritması, causal-learn Python paketi (<https://causal-learn.readthedocs.io/en/latest/>) kullanılarak ki-kare koşullu bağımsızlık testleri ($\alpha = 0.01$) ile gerçekleştirildi. Veri örnek boyutu n ile belirtilmiştir.