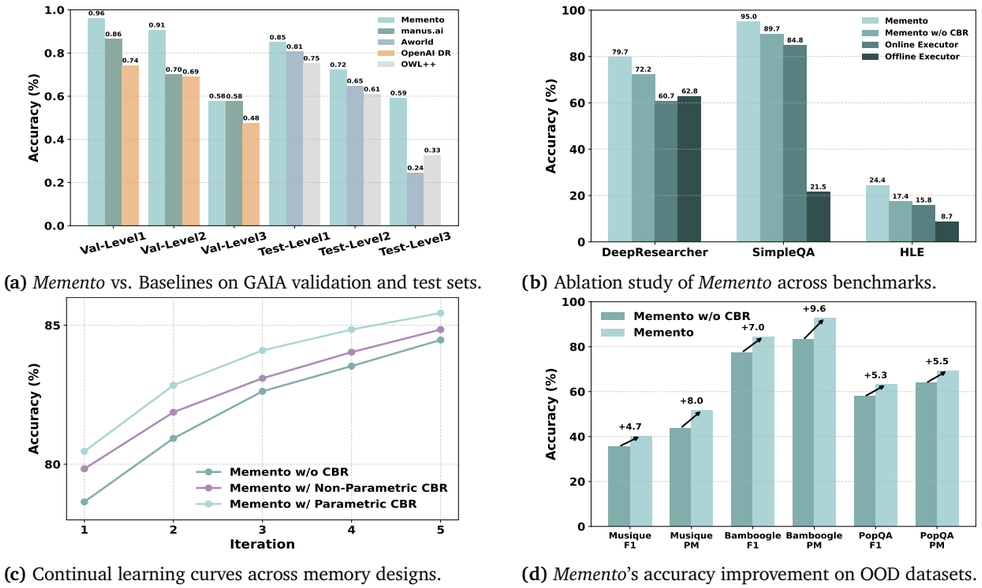
Memento: Fine-tuning LLM Agents without Fine-tuning LLMs

Huichi Zhou*[1,2], Yihang Chen*[2], Siyuan Guo[3], Xue Yan[4], Kin Hei Lee , Zihan Wang , Ka Yiu Lee[2], Guchun Zhang[2], Kun Shao[2], Linyi Yang†[2], and Jun Wang†[1]

[1]AI Centre, UCL, [2]Huawei Noah's Ark Lab, UK, [3]Jilin University, [4]Institute of Automation, CAS

Abstract

In this paper, we introduce a novel learning paradigm for Adaptive Large Language Model (LLM) agents that eliminates the need for fine-tuning the underlying LLMs. Existing approaches are often either rigid, relying on static, handcrafted reflection workflows, or computationally intensive, requiring gradient updates of LLM model parameters. In contrast, our method enables low-cost continual adaptation via memory-based online reinforcement learning. We formalise this as a Memory-augmented Markov Decision Process (M-MDP), equipped with a neural case-selection policy to guide action decisions. Past experiences are stored in an episodic memory, either differentiable or non-parametric. The policy is continually updated based on environmental feedback through a memory rewriting mechanism, whereas policy improvement is achieved through efficient memory reading (retrieval). We instantiate our agent model in the deep research setting, namely *Memento*, which attains top-1 on GAIA validation (87.88% Pass@3) and 79.40% on the test set. It reaches 66.6% F1 and 80.4% PM on the DeepResearcher dataset, outperforming the state-of-the-art training-based method, while case-based memory adds 4.7% to 9.6% absolute points on out-of-distribution tasks. Our approach offers a scalable and efficient pathway for developing generalist LLM agents capable of continuous, real-time learning without gradient updates, advancing machine learning towards open-ended skill acquisition and deep research scenarios. The code is available at https://github.com/Agent-on-the-Fly/Memento.



(a) *Memento* vs. Baselines on GAIA validation and test sets.

(b) Ablation study of *Memento* across benchmarks.

(c) Continual learning curves across memory designs.

(d) *Memento*'s accuracy improvement on OOD datasets.

Figure 1: Overview of *Memento* evaluation across baselines, benchmarks, memory designs and generalisation.

Huichi Zhou and Yihang Chen contributed equally. Siyuan Guo, Xue Yan, Kin Hei Lee, and Zihan Wang conducted this work while visiting Prof. Jun Wang. Corresponding author(s): Linyi Yang: yangly6@sustech.edu.cn, and Jun Wang: jun.wang@cs.ucl.ac.uk

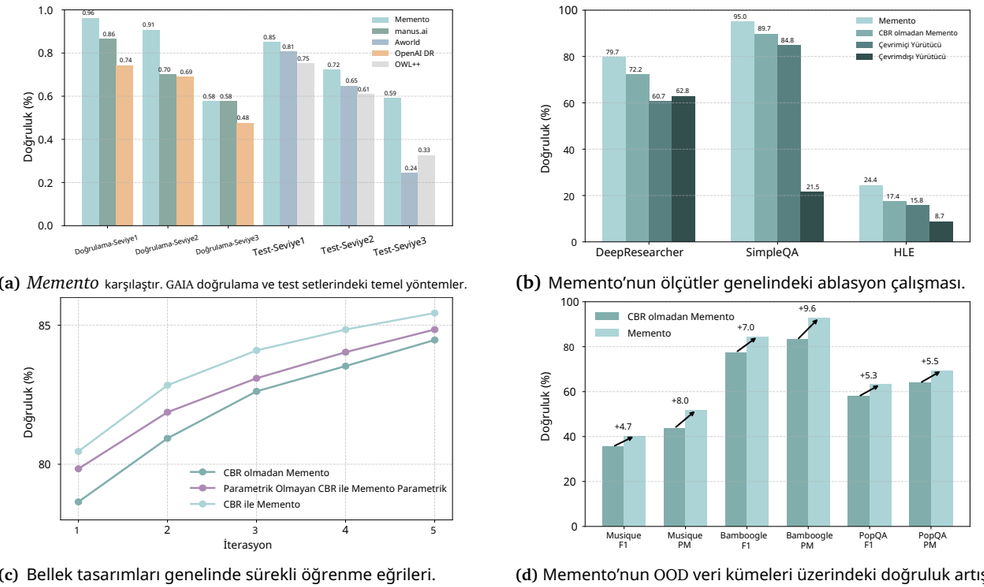
Memento: LLM Ajanlarını LLM'leri İnce Ayarlamadan İnce Ayarlamak

Huichi Zhou* [1,2] , Yihang Chen* [2] , Siyuan Guo [3] , Xue Yan [4] , Kin Hei Lee, Zihan Wang, Ka Yiu Lee [2] , Guchun Zhang [2] , Kun Shao [2] , Linyi Yang† [2] , ve Jun Wang† [1]

[1]AI Merkezi, UCL, [2]Huawei Noah's Ark Lab, Birleşik Krallık, [3]Jilin Üniversitesi, [4]CAS Otomasyon Enstitüsü

Özet

Bu makalede, temel LLM'lerin ince ayarına gerek kalmaksızın uyarlanabilir Büyük Dil Modeli (LLM) ajanları için yeni bir öğrenme paradigması sunuyoruz. Mevcut yaklaşımlar ya statik ve el yapımı yansıtma iş akışlarına dayanan katı çözümler ya da LLM model parametrelerinin gradyan güncellemelerini gerektiren hesaplama açısından yoğun yöntemlerden oluşmaktadır. Buna karşılık, yöntemimiz bellek tabanlı çevrimiçi pekiştirmeli öğrenme ile düşük maliyetli sürekli uyum sağlamaktadır. Bunu, eylem kararlarını yönlendirmek için sinirsel vaka seçme politikası ile donatılmış Bellek Destekli Markov Karar Süreci (M-MDP) olarak resmileştiriyoruz. Geçmiş deneyimler, ya diferansiyellenebilir ya da parametrik olmayan episodik bellekte depolanır. Politika, çevresel geri bildirim temelinde bir bellek yeniden yazma mekanizması aracılığıyla sürekli güncellenirken, politika geliştirmesi etkin bellek okuma (geri getirme) yoluyla sağlanır . Ajan modelimizi derin araştırma ortamında, yani *Memento* olarak somutlaştırıyoruz; bu model GAIA doğrulamasında (%87,88 Pass@ 3) birinci sıraya ulaşmakta ve test setinde %79,40 skor elde etmektedir. DeepResearcher veri setinde % 66,6 F1 ve %80,4 PM skorlarına ulaşarak, en güncel eğitim temelli yöntemi geride bırakmaktadır; vaka bazlı bellek ise dağılım dışı görevlerde mutlak puan olarak %4,7'den %9,6'ya ek katkı sağlamaktadır. Yaklaşımımız, sürekli ve gerçek zamanlı öğrenmeyi gradyan güncellemeleri olmadan gerçekleştirebilen genel amaçlı LLM ajanları geliştirmek için ölçeklenebilir ve etkili bir yol sunmakta olup, makine öğrenimini uçtan uca yetenek kazanımı ve derin araştırma senaryolarına doğru ilerletmektedir. Kod şu adreste mevcuttur: https://github.com/Agent-on-the-Fly/Memento .



(a) *Memento* karşılaştır. GAIA doğrulama ve test setlerindeki temel yöntemler.

(b) Memento'nun ölçütler genelindeki ablasyon çalışması.

(c) Bellek tasarımları genelinde sürekli öğrenme eğrileri.

(d) Memento'nun OOD veri kümeleri üzerindeki doğruluk artışı.

Şekil **1:** Memento'nun baz modeller, ölçütler, bellek tasarımları ve genelleme kapsamındaki değerlendirmesine genel bakış.

Huichi Zhou ve Yihang Chen eşit katkıda bulunmuştur. Siyuan Guo, Xue Yan, Kin Hei Lee ve Zihan Wang bu çalışmayı Prof. Jun Wang'ı ziyaret ederken gerçekleştirmiştir. Sorumlu yazar(lar): Linyi Yang: yangly6@sustech.edu.cn ve Jun Wang: jun.wang@cs.ucl.ac.uk

1. Introduction

A Large Language Model (LLM) agent refers to a system that leverages one or more LLMs to autonomously perform complex tasks through interaction, reasoning, and decision making, often with access to external tools, memory, or environments (Christianos et al., 2023, Yang et al., 2025). Unlike passive LLMs that respond to prompts in isolation, LLM agents operate proactively and iteratively, guided by explicit goals. They are increasingly deployed as autonomous problem solvers (Choudhary et al., 2021, Wei et al., 2022, Yao et al., 2023) spanning various domains. Notable examples include deep research agents (OpenAI, 2025, Google, 2025, ByteDance, 2025), tool-enhanced execution systems (Li et al., 2025c, Zheng et al., 2025, Qian et al., 2025), and code generation agents (Cui et al., 2021, Guo et al., 2024, Grosnit et al., 2024, Guo et al., 2025), all of which demonstrate strong capabilities in complex scientific and engineering tasks.

Despite recent progress, current LLM agents typically follow two prevailing paradigms, each exhibiting fundamental limitations. The first approach builds specialised frameworks with fixed workflows and hard-coded reasoning, which work well for narrow tasks but lack flexibility. After deployment, such agents are static: they neither incorporate online information nor adapt to novel situations. The second paradigm focuses on updating the LLM itself through parameter tuning of underlying LLMs – via supervised fine-tuning or reinforcement learning – which allows for more flexible behaviour (Christianos et al., 2023, Shi et al., 2025) but comes at a high computational cost. These approaches are inefficient for continuous adaptation and online learning, impractical for agents deployed in open-ended scenarios. This observation raises a central research challenge towards generalist agents:

How can we build LLM agents that learn continuously from a changing environment without the prohibitive cost of fine-tuning the underlying LLMs?

Inspired by human memory mechanisms, we address this challenge by proposing a memory-based learning framework that enables continual adaptation without modifying the underlying LLMs. We observe that humans' performance steadily improves because each experience is (i) encoded as an episodic trace (Pritzel et al., 2017), (ii) distilled into abstract rules during sleep-dependent consolidation (Squire et al., 2015), (iii) selectively reinforced by dopamine-driven credit assignment (Glimcher, 2011), and (iv) retrieved through case- or analogy-based reasoning when similar problems arise (Ashley, 1992). Thus, instead of fine-tuning the base model, LLM agents leverage an external memory to store past trajectories – including successes and failures labels – and draw from similar past experiences to guide decision making. This approach aligns with the principles of case-based reasoning (CBR) (Aamodt and Plaza, 1994, Guo et al., 2024, 2025), a psychologically grounded learning strategy supported by evidence that humans often solve problems by recalling analogous past situations (Anderson, 2013, Ross, 1989). For example, in a deep research scenario, deep research agents that have previously succeeded on a web-based task can leverage their experience to solve never-seen and structurally similar tasks (Wiratunga et al., 2024). Our method offers a novel path to continual learning for deep research agents – efficient, generalizable, and inspired by how humans learn.

To this end, we introduce *Memento*, a non-parametric, learn-on-the-fly framework for CBR (Smyth and McClave, 2001, Hatalis et al., 2025), instantiated as a planner–executor architecture grounded in a memory-based Markov Decision Process (MDP). *Memento* comprises three principal components: (i) a planner, (ii) a tool-enabled executor, and (iii) a growing *Case Bank* that stores past trajectories as episodic memory. Instead of relying solely on the LLM's parametric memory, which is fixed after training, online case-based reasoning in *Memento* is implemented by storing rich episodic traces.

1. Giriş

Bir Büyük Dil Modeli (LLM) ajanı, genellikle harici araçlara, belleğe veya çevrelere erişim sağlayarak, bir veya birden fazla LLM'yi kullanarak etkileşim, muhakeme ve karar verme yoluyla otonom biçimde karmaşık görevleri yerine getiren bir sistemi ifade eder (Christianos ve ark., 2023; Yang ve ark., 2025). İzole şekilde istemlere yanıt veren pasif LLM'lerden farklı olarak, LLM ajanları açık hedeflerle yönlendirilen proaktif ve yinelemeli bir biçimde çalışır. Farklı alanları kapsayan otonom problem çözücüler olarak giderek daha yaygın şekilde kullanılmaktadırlar (Choudhary ve ark., 2021; Wei ve ark., 2022; Yao ve ark., 2023). Kayda değer örnekler arasında derin araştırma ajanları (OpenAI, 2025; Google, 2025; ByteDance, 2025), araç destekli yürütme sistemleri (Li ve ark., 2025c; Zheng ve ark., 2025; Qian ve ark., 2025) ve kod üretme ajanları (Cui ve ark., 2021; Guo ve ark., 2024; Grosnit ve ark., 2024; Guo ve ark., 2025) yer almakta olup, bunların tümü karmaşık bilimsel ve mühendislik görevlerinde güçlü yetkinlikler göstermektedir.

Son gelişmelere rağmen, mevcut LLM ajanları genellikle iki yaygın paradigmayı takip etmekte ve her biri temel sınırlamalara sahiptir. Birinci yaklaşım, sabit iş akışları ve sert kodlanmış çıkarımlarla özel çerçeveler oluşturarak dar görevlerde iyi performans gösterirken esneklikten yoksundur. Dağıtımdan sonra, bu ajanlar statiktir; ne çevrimiçi bilgiyi işler ne de yeni durumlara uyum sağlarlar. İkinci paradigma ise, denetimli ince ayar veya pekiştirmeli öğrenme yoluyla temel LLM parametrelerinin güncellenmesine odaklanmakta; bu, daha esnek davranışlara imkân tanırken (Christianos ve ark., 2023; Shi ve ark., 2025) yüksek hesaplama maliyetiyle sonuçlanmaktadır. Bu yaklaşımlar sürekli uyum sağlama ve çevrimiçi öğrenme açısından verimsiz olup, açık uçlu senaryolarda konuşlandırılan ajanlar için kullanışlı değildir. Bu gözlem, genel amaçlı ajanlara yönelik temel bir araştırma sorusunu ortaya koymaktadır:

Altında yatan LLM'lerin ince ayarının yüksek maliyeti olmadan, değişen bir ortamdan sürekli öğrenen LLM ajanları nasıl geliştirilebilir?

İnsan bellek mekanizmalarından ilham alarak, altında yatan LLM'leri değiştirmeden sürekli uyumu mümkün kılan bellek tabanlı bir öğrenme çerçevesi öneriyoruz. İnsanların performansının düzenli olarak iyileşmesini şu süreçlere bağlıyoruz: Her deneyim (i) epizodik iz olarak kodlanmakta (Pritzel ve ark., 2017), (ii) uyku bağımlı pekiştirmeyle soyut kurallara dönüşmekte (Squire ve ark., 2015), (iii) dopamin temelli kredi ataması ile seçici olarak pekiştirilmekte (Glimcher, 2011) ve (iv) benzer sorunlarla karşılaşıldığında durum- veya analoji temelli muhakeme yoluyla erişilmektedir (Ashley, 1992). Bu nedenle, temel modelin ince ayarını yapmak yerine, LLM ajanları geçmiş rotaları — başarı ve başarısızlık etiketleri dahil — saklamak için harici bellek kullanmakta ve karar verme sürecini yönlendirmek için benzer geçmiş deneyimlerden yararlanmaktadır. Bu yaklaşım, insanların sıklıkla benzer geçmiş durumları hatırlayarak problemleri çözdüğüne dair kanıtlarla desteklenen psikolojik temelli bir öğrenme stratejisi olan vaka tabanlı akıl yürütme (CBR) ilkeleriyle uyumludur (Aamodt ve Plaza, 1994; Guo vd., 2024, 2025; Anderson, 2013; Ross, 1989). Örneğin, derin araştırma senaryosunda, daha önce web tabanlı bir görevde başarılı olmuş derin araştırma ajanları, hiç karşılaşılmamış ve yapısal olarak benzer görevleri çözmek için deneyimlerinden yararlanabilirler (Wiratunga vd., 2024). Yöntemimiz, derin araştırma ajanları için insan öğrenme biçiminden esinlenmiş, verimli, genellenebilir ve sürekli öğrenmeye yönelik yeni bir yol sunmaktadır.

Bu amaçla, bellek tabanlı Markov Karar Süreci (MDP) temelinde somutlaştırılan planlayıcı–yürütücü mimarisi olarak geliştirilen, parametrik olmayan ve hareket halindeyken öğrenen bir vaka tabanlı akıl yürütme (CBR) çerçevesi olan *Memento*'yı tanıtıyoruz (Smyth ve McClave, 2001; Hatalis vd., 2025). Memento, üç temel bileşenden oluşur: (i) bir planlayıcı, (ii) araç destekli bir yürütücü ve (iii) geçmiş trajektörileri episodik bellek olarak depolayan büyüyen bir Vaka Bankası. Eğitim sonrası sabit kalan LLM'nin parametrik belleğine yalnızca dayanmak yerine, çevrimiçi vaka tabanlı akıl yürütme Memento'da zengin episodik izlerin depolanmasıyla uygulanır.

Our experiments are conducted on 4 benchmarks, where GAIA (Mialon et al., 2023) for long-horizon tool use, DeepResearcher (Zheng et al., 2025) for real-time web research, SimpleQA (Wei et al., 2024) for factual precision, and HLE (Phan et al., 2025) for long-tail academic reasoning. We use a planner–executor architecture with GPT-4.1 as the planner and o4-mini as the default executor (o3 for GAIA), instrumented with tools, namely *Memento*. We achieve top-1 on GAIA validation (87.88% Pass@3) and 79.40% on the private test leaderboard, and it reaches 66.6% F1 and 80.4% PM on the DeepResearcher dataset, outperforming the state-of-the-art training-based system, while case-based memory adds 4.7 to 9.6 absolute points on out-of-distribution tasks and yields 95.0% PM on SimpleQA. To our knowledge, we are the first to cast case-based continual learning for LLM agents, achieving the top-level performance on the GAIA benchmark, thereby providing a principled framework for continual adaptation of Deep Research agents.

2. Related Work

We first review methods that equip LLMs with continual-learning capabilities. Then, we discuss approaches that augment agents with external tools and multi-agent coordination. Lastly, we introduce agent memory mechanisms, characterising design choices in representation, retrieval, and decay, and their implications for continual learning.

2.1. Continual-learning in LLM Agent Systems

Continual-learning strategies for LLM agents can be categorised into two camps. **Parametric approaches** (Zhu et al., 2025b,a) update the LLM through post-training (e.g., Reinforcement Learning (Wang et al., 2025)) or supervised fine-tuning (e.g., START (Li et al., 2025a)), achieving high task fidelity at the expense of considerable compute, data, and the danger of catastrophic forgetting (Li et al., 2024). It is often assumed that achieving the capability to solve complex reasoning problems requires substantial changes to the model's parameters, and therefore, full fine-tuning is widely applied during RL (Liu et al., 2025). However, when tackling long-horizon, complex tasks (Mialon et al., 2023, Phan et al., 2025), LLM agent systems must spend substantial time rolling out trajectories to gather training data, and they additionally depend on large volumes of human-annotated questions. Differently, **non-parametric approaches** freeze the LLM and attach an external memory to optimise the prompt construction process. Human intelligence relies heavily on memory systems, especially episodic memory, which supports learning from both successes and failures (Baddeley, 1983). Cognitive science suggests that such memories are segmented and selectively replayed to inform future decisions (Anderson et al., 1997, Khosla et al., 2023, Fountas et al., 2024). This inspired early AI paradigms like Case-Based Reasoning (CBR) (Francis and Ram, 1993). While modern Retrieval-Augmented Generation (RAG) systems (Lewis et al., 2020) share surface similarities with CBR, they typically query static document corpora and lack mechanisms for continual adaptation (Gao et al., 2023).

2.2. Tool-augmented LLM

Language agents increasingly incorporate external tools to overcome context limitations and computational bottlenecks. Prompt-based methods, including WebGPT (Nakano et al., 2021), embed tool calls directly in the generation trace. However, tackling long-horizon tasks often requires multi-hop tool calls. Therefore, recent works propose multi-agent pipelines, such as AutoGen (Wu et al., 2023), OWL (Camel-AI, 2025) and DeerFlow (ByteDance, 2025) to coordinate specialised agents via dialogue. To address long-horizon decision-

Deneylerimiz, uzun vadeli araç kullanımı için GAIA (Mialon et al., 2023), gerçek zamanlı web araştırması için Deep Researcher (Zheng et al., 2025), gerçek bilgi doğruluğu için SimpleQA (Wei et al., 2024) ve uzun kuyruk akademik akıl yürütme için HLE (Phan et al., 2025) olmak üzere 4 benchmark üzerinde gerçekleştirilmiştir. GPT-4.1 planlayıcı olarak ve varsayılan yürütücü olarak araçlarla donatılmış o4-mini (GAIA için o3) kullanılarak planlayıcı–yürütücü mimarisi oluşturulmuştur; bu mimaride Memento yer almaktadır. GAIA doğrulama setinde (%87,88 Pass@ 3) birincilik elde ediyoruz ve özel test liderlik tablosunda %79,40 performans gösteriyoruz; ayrıca DeepResearcher veri setinde %66,6 F 1 ve %80,4 PM değerlerine ulaşarak, duruma dayalı bellek sisteminin dağılım dışı görevlerde mutlak olarak 4,7 ile 9,6 puan eklemesini sağlayıp SimpleQA üzerinde %95,0 PM elde ediyoruz. Bildiğimiz kadarıyla, LLM ajanları için duruma dayalı sürekli öğrenmeyi ilk kez uyguluyor, GAIA kıyaslama testinde en üst düzey performansa ulaşarak Deep Research ajanlarının sürekli adaptasyonu için prensip temelli bir çerçeve sunuyoruz.

2. İlgili Çalışmalar

Öncelikle LLM'lere sürekli öğrenme yetenekleri kazandıran yöntemleri inceliyoruz. Ardından, ajanları dış araçlarla güçlendiren ve çoklu ajan koordinasyonunu ele alan yaklaşımları tartışıyoruz. Son olarak, temsil, geri çağırma ve unutma tasarım tercihlerini tanımlayarak ajan bellek mekanizmalarını ve bunların sürekli öğrenmeye etkilerini ele alıyoruz.

2.1. LLM Ajan Sistemlerinde Sürekli Öğrenme

LLM ajanları için sürekli öğrenme stratejileri iki ana kategoriye ayrılabilir. Parametrik yaklaşımlar (Zhu ve ark., 2025b,a), LLM'yi sonrası eğitim (örneğin, Takviyeli Öğrenme (Wang ve ark., 2025)) veya denetimli ince ayar (örneğin, START (Li ve ark., 2025a)) yoluyla günceller; yüksek görev doğruluğu sağlarken önemli miktarda hesaplama, veri kullanımı ve felaket unutma riski taşır (Li ve ark., 2024). Genellikle, karmaşık akıl yürütme problemlerini çözebilme yeteneğinin model parametrelerinde kapsamlı değişiklikler gerektirdiği varsayılmakta ve bu nedenle, RL sırasında tam ince ayar yaygın şekilde uygulanmaktadır (Liu ve ark., 2025). Ancak, uzun vadeli ve karmaşık görevler üzerinde çalışırken (Mialon ve ark., 2023; Phan ve ark., 2025), LLM ajan sistemleri eğitim verisi toplamak için uzun süreler boyunca trajektorya açmak zorundadır ve ayrıca büyük miktarda insan tarafından etiketlenmiş soruya ihtiyaç duyar. Farklı olarak, parametrik olmayan yaklaşımlar LLM'yi sabit tutar ve prompt oluşturma sürecini optimize etmek için harici belleğe bağlanır. İnsan zekası büyük ölçüde bellek sistemlerine, özellikle başarı ve başarısızlıklardan öğrenmeyi destekleyen episodik belleğe dayanır (Baddeley, 1983). Bilişsel bilim, bu tür anıların segmentlere ayrıldığını ve gelecekteki kararları bilgilendirmek üzere seçici olarak tekrar oynatıldığını belirtir (Anderson ve ark., 1997; Khosla ve ark., 2023; Fountas ve ark., 2024). Bu durum, Case-Based Reasoning (CBR) (Francis ve Ram, 1993) gibi erken yapay zeka paradigmalarını teşvik etmiştir. Modern Retrieval-Augmented Generation (RAG) sistemleri (Lewis ve ark., 2020) CBR ile yüzeysel benzerlikler taşısa da, genellikle statik belge kütüphanelerini sorgular ve sürekli adaptasyon mekanizmalarından yoksundur (Gao ve ark., 2023).

2.2. Araç Destekli LLM

Dil ajanları, bağlam sınırlamalarını ve hesaplama darboğazlarını aşmak için giderek daha fazla harici araç entegre etmekte dir. İstem tabanlı yöntemler, bunlar arasında WebGPT (Nakano et al., 2021) de bulunmaktadır, araç çağrılarını doğrudan üretim izine yerleştirir. Ancak, uzun vadeli görevlerle başa çıkmak genellikle çok aşamalı araç çağrılarını gerektirir. Bu nedenle, yakın tarihli çalışmalar AutoGen (Wu et al., 2023), OWL (Camel-AI, 2025) ve DeerFlow (ByteDance, 2025) gibi çoklu ajan iş akışlarını, uzmanlaşmış ajanları diyalog yoluyla koordine etmek için önermektedir. Uzun vadeli karar verme sürecini,

making in dynamic, multi-turn interactions with external tool environments, Agentic Reinforcement Learning (Agentic RL) has emerged as a promising training paradigm. This approach shifts LLM training from static task-solving (e.g., math or code) to dynamic, agent–environment reasoning. Supervised Fine-tuning methods, including Toolformer (Schick et al., 2023), API-Bench (Li et al., 2023), and GRPO-based optimisation (Wang et al., 2025, Qian et al., 2025, Feng et al., 2025) teach models when and how to invoke APIs, but require costly retraining and often assume a fixed, small toolset (e.g., Code and Search). However, without explicit planning, deciding when and which tools to invoke remains a major bottleneck for long-horizon tasks. We model planning as a stateful MDP with explicit memory for past cases. By bringing case-based reasoning into planning, the executor is steered toward strategic tool calls and consistently strong performance.

2.3. Agent Memory Mechanism

Recent work has centred on endowing LLM agents with explicit memory structures. A growing body of work (Camel-AI, 2025, Liang et al., 2025, Google, 2025, ByteDance, 2025) has shown that current LLM agents are designed for fixed environments, limiting their ability to evolve. While some efforts, such as ReAct-style agents and reflective prompting pipelines (Shinn et al., 2023, Yao et al., 2023) demonstrate improvement through feedback, they remain constrained by pre-defined heuristics and do not achieve true lifelong learning. DS-Agent (Guo et al., 2024) stabilises planning by mining prior Kaggle solutions and turning them into executable pipelines. Agent-K (Grosnit et al., 2024) adds structured memory and credit assignment to reuse past work, enabling end-to-end automation of Kaggle-style workflows. Furthermore, Agent-KB (Tang et al., 2025) and Alita (Qiu et al., 2025) construct shared knowledge bases and optimised toolsets for agentic problem-solving. However, most systems keep adding cases without selective curation, leading to the classic swamping problem where retrieval costs outweigh utility (Francis and Ram, 1993).

LLM agents are increasingly equipped with long-term memory that grows and adapts over time, allowing them to accumulate knowledge, recall prior context, and adjust behaviour based on experience. Memory-Bank (Zhong et al., 2024) couples retrieval with an Ebbinghaus-style forgetting schedule so older, low-utility items decay while user-relevant facts are reinforced. Building on this idea, SAGE (Liang et al., 2024) unifies reflection with an Ebbinghaus-based memory optimiser to support continual self-refinement. Mem0 (Chhikara et al., 2025) adopts a structured memory mechanism with explicit operations (ADD, UPDATE, DELETE, NOOP). A-MEM (Xu et al., 2025) maintains memory via a typological network. MemInsight (Salama et al., 2025) pushes further on semantics by augmenting raw memories with summaries and tags to aid retrieval. Several lines of work distil operational knowledge from interaction traces: ExpeL (Zhao et al., 2024) collects trajectories and converts them into reusable natural-language insights and rules; AutoGuide (Fu et al., 2024) compresses offline logs into concise, conditional, context-aware guidelines; and Agent Workflow Memory (Wang et al., 2024) induces frequently used subtask sequences as auxiliary skills. Finally, Agent-KB (Tang et al., 2025) and Alita (Qiu et al., 2025) construct shared knowledge bases and optimised toolsets to support agentic problem solving. Differently, we formulate planning as a memory-augmented MDP and learn a neural case-selection policy over an episodic case bank via online soft Q-learning, enabling continual adaptation without fine-tuning the underlying LLM parameters.

3. Methodology: Memory-Based MDP with Case-based Reasoning Policy

In this work, we integrate LLM agents with case-based reasoning, a classic problem-solving paradigm that solves new problems by learning from solutions to previously encountered similar problems. As such, LLM

dinamik, çok adımlı etkileşimlerle dış araç ortamlarında ele almak amacıyla, Agentic Takviyeli Öğrenme (Agentic RL) umut vadeden bir eğitim paradigması olarak ortaya çıkmıştır. Bu yaklaşım, LLM eğitimini statik görev çözümünden (örneğin matematik veya kod) dinamik ajan–çevre akıl yürütmesine kaydırmaktadır. Toolformer (Schick et al., 2023), API-Bench (Li et al., 2023) ve GRPO tabanlı optimizasyon (Wang et al., 2025; Qian et al., 2025; Feng et al., 2025) gibi denetimli ince ayar yöntemleri modellere ne zaman ve nasıl API çağrısı yapacaklarını öğretir, ancak maliyetli yeniden eğitimi gerektirir ve genellikle sabit, küçük bir araç seti (örneğin Kod ve Arama) varsayar. Ancak, açık bir planlama olmadan hangi araçların ne zaman çağrılacağına karar vermek uzun vadeli görevlerde önemli bir darboğaz olarak kalmaktadır. Planlamayı, geçmiş vakalar için açık belleğe sahip durumlu bir Markov Karar Süreci (MDP) olarak modellemekteyiz. Vaka tabanlı akıl yürütmeyi planlamaya entegre ederek, yürütücü stratejik araç çağrılarına yönlendirilmekte ve sürekli güçlü performans göstermektedir.

2.3. Ajan Bellek Mekanizması

Son dönemde yapılan çalışmalar, LLM ajanlarını açık bellek yapılarıyla donatmaya odaklanmaktadır. Artan sayıda çalışma (Camel-AI, 2025; Liang ve ark., 2025; Google, 2025; ByteDance, 2025), mevcut LLM ajanlarının sabit ortamlar için tasarlandığını ve evrim yeteneklerinin kısıtlı olduğunu ortaya koymuştur. ReAct tarzı ajanlar ve yansıtıcı teşvik zincirleri (Shinn ve ark., 2023; Yao ve ark., 2023) gibi bazı yaklaşımlar geribildirim yoluyla gelişim sağlasa da, önceden tanımlanmış sezgilerle sınırlı kalmakta ve gerçek yaşam boyu öğrenmeye ulaşamamaktadır. DS-Ajan (Guo ve ark., 2024), önceki Kaggle çözümlerini analiz ederek planlamayı istikrara kavuşturmakta ve bunları yürütülebilir boru hatlarına dönüştürmektedir. Agent-K (Grosnit ve ark., 2024), geçmiş çalışmaları yeniden kullanabilmek amacıyla yapılandırılmış bellek ve kredi tahsisi ekleyerek Kaggle tarzı iş akışlarının uçtan uca otomasyonunu sağlamaktadır. Ayrıca, Agent-KB (Tang ve ark., 2025) ve Alita (Qiu ve ark., 2025), ajan temelli problem çözümü için paylaşılan bilgi tabanları ve optimize edilmiş araç setleri oluştururlar. Ancak, çoğu sistem seçici kürasyon yapmadan vakalar eklemeye devam etmekte; bu durum, geri çağırma maliyetlerinin faydayı aştığı klasik taşma (swamping) sorununa yol açmaktadır (Francis ve Ram, 1993).

LLM ajanları, zamanla büyüyen ve uyum sağlayan uzun vadeli belleklere giderek daha fazla donatılmakta; böylece bilgi biriktirebilmekte, önceki bağlamı hatırlayabilmekte ve deneyime dayalı olarak davranışlarını ayarlayabilmektedir. Memory-Bank (Zhong ve ark., 2024), geri çağırmayı Ebbinghaus tarzı unutma programıyla ilişkilendirerek eski ve düşük faydalı öğelerin yok olmasını sağlarken, kullanıcıyla ilgili gerçeklerin pekiştirilmesini mümkün kılar. Bu fikir üzerine kurulu olan SAGE (Liang ve ark., 2024), sürekli kendi kendini geliştirmeyi desteklemek için yansıtmayı Ebbinghaus temelli bir bellek optimize edici ile birleştirir. Mem0 (Chhikara ve ark., 2025) açık işlemler (ADD, UPDATE, DELETE, NOOP) içeren yapılandırılmış bir bellek mekanizması benimser. A-MEM (Xu ve ark., 2025) belleği tipolojik bir ağ üzerinden sürdürür. MemInsight (Salama ve ark., 2025) ham bellekleri özetler ve etiketlerle zenginleştirerek geri çağırmayı kolaylaştırarak anlamsal olarak ileri bir adım atar. Çeşitli çalışmalar, etkileşim izlerinden operasyonel bilgi damıtarak: ExpeL (Zhao ve ark., 2024) yörüngeleri toplayıp bunları yeniden kullanılabilir doğal dil içgörüleri ve kurallara dönüştürür; AutoGuide (Fu ve ark., 2024) çevrimdışı günlükleri özlü, koşullu ve bağlama duyarlı rehberlere dönüştürür; ve Agent Workflow Memory (Wang ve ark., 2024) sık kullanılan alt görev dizilerini yardımcı beceriler olarak türetir. Son olarak, Agent-KB (Tang ve ark., 2025) ile Alita (Qiu ve ark., 2025) paylaşılan bilgi tabanları ve optimize edilmiş araç setleri oluşturarak ajan bazlı problem çözümünü destekler. Farklı olarak, planlamayı bellek destekli bir MDH olarak formüle ediyor ve çevrimiçi soft Q-öğrenme ile episodik vaka bankası üzerinde sinirsel vaka seçme politikasını öğreniyoruz; bu da temel LLM parametrelerini ince ayar yapmadan sürekli adaptasyona olanak tanır.

3. Metodoloji: Vaka Tabanlı Politikalı Bellek Tabanlı MDH

Bu çalışmada, LLM ajanlarını vaka tabanlı akıl yürütme ile entegre ediyoruz; bu paradigma, önceden karşılaşılan benzer problemlere verilen çözümlerden öğrenerek yeni problemleri çözmeyi sağlar. Bu bağlamda, LLM

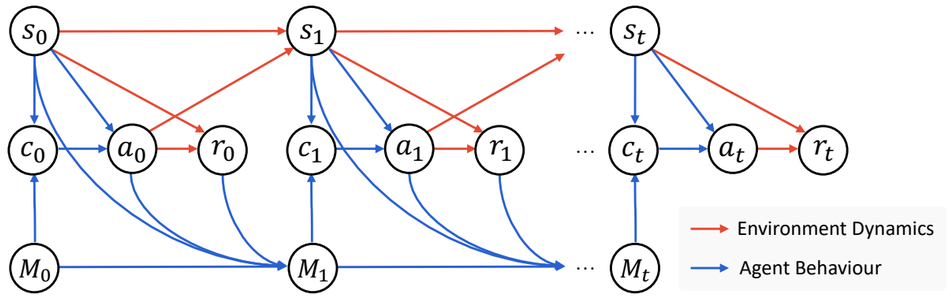


Figure 2: A graphical model of memory-based Markov Decision Process.

agents can achieve continuous improvement without parameter fine-tuning by learning from experiences stored in memory. To begin with, we model the sequential decision-making process of CBR agents as a Memory-Based Markov Decision Process (M-MDP) as below.

Definition 3.1 (Memory-Based Markov Decision Process)**.** *A Memory-Based Markov Decision Process is a tuple* $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \mathcal{M} \rangle$*, where* $\mathcal{S}$ *is the state space,* $\mathcal{A}$ *is the action space,* $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ *is the transition dynamics,* $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ *is the reward function,* $\gamma \in [0, 1)$ *is the discount factor, and* $\mathcal{M} = (\mathcal{S} \times \mathcal{A} \times \mathbb{R})^*$ *is the memory space.*

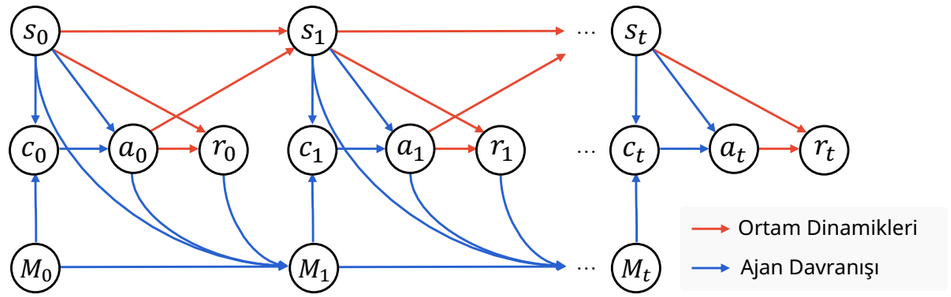
The graphical model of M-MDP is illustrated in Figure 2. Note that the key difference from standard MDP is that we introduce a memory space as a set of past experiences. In the CBR agent setting, both state space and action space are defined as the set of all finite-length sequences over a predefined vocabulary $\mathcal{V}$.

With the M-MDP formulation, the behaviour of the CBR agent can be formally described as follows. At timestep t, we maintain a case bank (i.e., the memory) $M_t = \{c_i\}_{i=1}^{N_t}$, with each case c_i a tuple (s_i, a_i, r_i), and N_t the number of cases in the current case bank. Given the current state s_t, the CBR agent first retrieves a case $c_t \sim \mu(\cdot \mid s_t, M_t)$, and then reuses and adapts it via the LLM, i.e., $a_t \sim p_{\text{LLM}}(\cdot \mid s_t, c_t)$. Here, μ denotes the case retrieval policy, whose implementation details will be presented later. Taking the action a_t, the CBR agent receives the reward $r_t = \mathcal{R}(s_t, a_t)$ and observes the next state $s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)$. The CBR agent also retains the new case in the case bank, i.e., $M_{t+1} = M_t \cup \{(s_t, a_t, r_t)\}$. In this way, we can define the overall policy of the CBR agent as below.

Definition 3.2 (Case-Based Reasoning Agent)**.** *A Case-Based Reasoning Agent is an agent that makes decisions based on both the current state and a finite memory of past experiences. Formally, let* $s \in \mathcal{S}$ *denote the current state;* $M \in \mathcal{M}$ *denote the current case bank, consisting of past cases* c*;* $a \in \mathcal{A}$ *denote the action;* $\mu(c \mid s, M)$ *denote a case retrieval policy, assigning a probability distribution over* M *given the current state* s*;* $p_{LLM}(a \mid s, c)$ *denote the action likelihood of a large language model (LLM) conditioned on the current state* s *and a retrieved case* $c \in M$*. Then, the overall policy* π *of a CBR agent is defined as:*

$$\pi(a|s, M) = \sum_{c \in M} \mu(c|s, M) p_{LLM}(a|s, c). \tag{1}$$

Overall, the trajectory τ of the CBR agent can be described as: $\tau = \{M_0, s_0, c_0, a_0, r_0, M_1, s_1, c_1, a_1, r_1, \cdots\}$. The



Şekil **2:** Bellek tabanlı Markov Karar Sürecinin grafiksel modeli.

Ajanlar, bellekte depolanan deneyimlerden öğrenerek parametre ince ayarı yapmadan sürekli iyileşme sağlayabilir. Başlangıç olarak, CBR ajanlarının ardışık karar verme sürecini aşağıda belirtildiği üzere Bellek Tabanlı Markov Karar Süreci (M-MDP) olarak modelliyoruz.

Tanım **3.1** (*Bellek Tabanlı Markov Karar Süreci*)**.** *Bellek Tabanlı Markov Karar Süreci, aşağıdaki kümülü ifade eder:* $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \mathcal{M} \rangle$*; burada* $\mathcal{S}$ *durum uzayıdır,* $\mathcal{A}$ *eylem alanıdır,* $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ *durum geçiş dinamikleridir,* $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ *ödül fonksiyonudur,* $\gamma \in [0, 1)$ *indirim faktörüdür ve* $\mathcal{M} = (\mathcal{S} \times \mathcal{A} \times \mathbb{R})$^*ast bellek alanıdır.*

M-MDP'nin grafiksel modeli Şekil 2'de gösterilmiştir. Standart MDP'den temel fark, geçmiş deneyimlerin bir kümesi olarak bir bellek alanı tanımlamamızdır. CBR ajanı bağlamında, hem durum uzayı hem de eylem alanı, önceden tanımlanmış bir sözcük dağarcığı $\mathcal{V}$ üzerindeki sonlu uzunluktaki tüm diziler kümesi olarak tanımlanır.

M-MDP formülasyonu ile CBR ajanının davranışı aşağıdaki şekilde resmi olarak tanımlanabilir. Zaman adımı t anında, vaka bankası (yani bellek) $M_t = \{c_i\}_{i=1}^{N_t}$ olarak tutulur; burada her vaka c_i bir kümül (s_i, a_i, r_i) ve N_t mevcut vaka bankasındaki vaka sayısını ifade eder. Geçerli durum s_t verildiğinde, CBR ajanı önce bir vaka $c_t \sim \mu(\cdot \mid s_t, M_t)$ erişir, ardından bunu LLM aracılığıyla yeniden kullanır ve uyarlama yapar, yani $a_t \sim p_{\text{LLM}}(\cdot \mid s_t, c_t)$. Burada, μ sonraki bölümlerde uygulanma detayları sunulacak vaka getirme politikasıdır. CBR ajanı a_t eylemini gerçekleştirdiğinde ödül $r_t = \mathcal{R}(s_t, a_t)$ alır ve sonraki durumu gözlemler $s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t)$. CBR ajanı ayrıca yeni vakayı vaka bankasında tutar; yani $M_{t+1} = M_t \cup \{(s_t, a_t, r_t)\}$. Bu şekilde, CBR ajanının genel politikasını aşağıdaki gibi tanımlayabiliriz.

Tanım 3.2 (Vaka Tabanlı Akıl Yürütme Ajanı) **.** *Vaka Tabanlı Akıl Yürütme Ajanı, hem mevcut duruma hem de geçmiş deneyimlerin sınırlı belleğine dayanarak karar veren ajandır. Resmi olarak, mevcut durum* $s \in \mathcal{S}$ *olarak tanımlansın;* $M \in \mathcal{M}$ *şimdiki vaka bankasını gösterir, geçmiş vakalardan oluşur* c*;* $a \in \mathcal{A}$ *eylemi gösterir;* $\mu(c \mid s, M)$ *vaka erişim politikasıdır; şimdiki durum* s *verildiğinde* M *üzerinde bir olasılık dağılımı atar;* $p_{LLM}(a \mid s, c)$ *büyük dil modeli (LLM) tarafından, şimdiki durum* s *ve erişilen vaka* $c \in M$ *koşulunda eylem olasılığını gösterir. O halde, CBR ajanın genel politikası* π *şu şekilde tanımlanır:*

$$\pi(a|s, M) = \sum_{c \in M} \mu(c|s, M) p_{LLM}(a|s, c). \tag{1}$$

Genel olarak, CBR ajanın trajektori τşu şekilde tanımlanabilir: $\tau = \{M_0, s_0, c_0, a_0, r_0, M_1, s_1, c_1, a_1, r_1, \cdots\}$. The

probability of sampling the trajectory τ can be modelled as:

$$p(\tau) = \prod_{t=0}^{T-1} \underbrace{\mu(c_t \mid s_t, M_t)}_{\text{(1) Retrieve}} \underbrace{p_{\text{LLM}}(a_t \mid s_t, c_t)}_{\text{(2) Reuse\&Revise}} \underbrace{\mathbb{I}[r_t = \mathcal{R}(s_t, a_t)]}_{\text{(3) Evaluation}} \underbrace{\mathbb{I}[M_{t+1} = M_t \cup (s_t, a_t, r_t)]}_{\text{(4) Retain}} \underbrace{\mathcal{P}(s_{t+1} \mid s_t, a_t)}_{\text{(5) Transition}}, \quad (2)$$

where $\mathbb{I}(\cdot)$ is the indicator function, assigning probability 1 if the condition holds and 0 otherwise, modelling the deterministic reward function and memory update, and T denotes the maximum trajectory length. Note that the reward function and memory update can also be probabilistic in some specific cases, which we leave as future work. Among them, (1) Retrieve, (2) Reuse and Revise, and (4) Retain describe the agent behaviour; (3) Evaluation and (5) Transition model the environment dynamics.

Soft Q-Learning for CBR Agent. To optimise the CBR policy π in Eq. (1), we aim to learn the case retrieval policy μ with the LLM component p_{LLM} fixed. In this context, the "action" of μ is to select a case $c = (s, a, r)$ from the case bank M. To optimise it while encouraging diversity in retrieved cases, we apply the maximum entropy RL framework (Haarnoja et al., 2018) and derive the following optimisation objective:

$$J(\pi) = \mathbb{E}_{\tau \sim p}\left[\sum_{t=0}^{T-1} \left[\mathcal{R}(s_t, a_t) + \alpha \mathcal{H}\left(\mu\left(\cdot|s_t, M_t\right)\right)\right]\right], \quad (3)$$

where $\mathcal{H}$ denotes the entropy, and α denotes the hyper-parameter of the entropy weight in the final reward. Under this framework, the value function can be defined as:

$$V^{\pi}(s_t, M_t) = \sum_{c \in M_t} \mu(c|s_t, M_t)[Q^{\pi}(s_t, M_t, c) - \alpha \log \mu(c|s_t, M_t)]. \quad (4)$$

Also, the Q value function for taking an "action" (i.e., selecting a case), given a state, can be defined as:

$$Q^{\pi}(s_t, M_t, c_t) = \mathbb{E}_{a \sim p_{\text{LLM}}(\cdot|s_t, c_t), s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)}\left[\mathcal{R}(s_t, a_t) + \gamma V^{\pi}(s_{t+1}, M_{t+1})\right], \quad (5)$$

where M_{t+1} denotes the updated memory after (s_t, a_t, r_t) is added. Let $d^{\pi}(s, M) = \sum_{t=0}^{\infty} \gamma^{t-1} \mathbb{P}(s_t = s, M_t = M)$ denote the discounted visitation frequency of (s, M) under π. The expected value function objective is then defined as:

$$J(\pi) = \mathbb{E}_{(s,M) \sim d^{\pi}}\left[V^{\pi}(s, M)\right] = \mathbb{E}_{(s,M) \sim d^{\pi}}\left[\sum_{c \in M} \mu(c|s, M)\left[Q^{\pi}(s, M, c) - \alpha \log \mu(c|s, M)\right]\right]. \quad (6)$$

Then, we can derive the closed-form solution of the optimal retrieval policy as a softmax over the optimal Q value:

$$\mu^*(c|s, M) = \frac{\exp(Q^*(s, M, c)/\alpha)}{\sum_{c' \in M} \exp(Q^*(s, M, c')/\alpha)}. \quad (7)$$

The detailed derivation can be found in Appendix A. In this way, we can derive the optimal retrieval policy by learning the Q-function Q, which can be achieved by the temporal difference (TD) learning in soft Q-learning (Haarnoja et al., 2017) as:

$$Q(s_t, M_t, c_t) \leftarrow Q(s_t, M_t, c_t) + \eta\left[r_t + \gamma\alpha \log \sum_{c' \in M_{t+1}} \exp\left(Q(s_{t+1}, M_{t+1}, c_{t+1})\right) - Q(s_t, M_t, c_t)\right], \quad (8)$$

where η denotes the learning rate. Next, we provide a simpler way to learn the Q-function by learning a similarity kernel over states.

trajektori τ örnekleme olasılığı şu şekilde modellenebilir:

$$p(\tau) = \prod_{t=0}^{T-1} \underbrace{\mu(c_t \mid s_t, M_t)}_{\text{(1) Erişim}} \underbrace{p_{\text{LLM}}(a_t \mid s_t, c_t)}_{\text{(2) Yeniden Kullan ve Revize Et}} \underbrace{\mathbb{I}[r_t = \mathcal{R}(s_t, a_t)]}_{\text{(3) Değerlendirme}} \underbrace{\mathbb{I}[M_{t+1} = M_t \cup (s_t, a_t, r_t)]}_{\text{(4) Tutma}} \underbrace{\mathcal{P}(s_{t+1} \mid s_t, a_t)}_{\text{(5) Geçiş}}, \quad (2)$$

Burada $\mathbb{I}(\cdot)$ gösterge fonksiyonu olup, koşul sağlandığında 1, sağlanmadığında 0 değerini atar; deterministik ödül fonksiyonu ve bellek güncellemesini modellemekte ve T maksimum trajektori uzunluğunu belirtmektedir. Ödül fonksiyonu ve bellek güncellemesi bazı özel durumlarda olasılıksal olabilir; bu durum gelecekteki çalışmalara bırakılmıştır. Bunlar arasında, (1) Erişim, (2) Yeniden Kullan ve Revize Et ve (4) Tutma ajan davranışını tanımlar; (3) Değerlendirme ve (5) Geçiş ise ortam dinamiklerini modellemektedir.

CBR ajanı için Soft **Q**-Öğrenme. Eşitlik (1)'deki **CBR** politikasını π optimize etmek amacıyla, LLM bileşeni p LLM sabit tutularak vaka erişim politikası μ öğrenilmektedir. Bu bağlamda, μ 'nın "eylemi", vaka bankası M içinden c = (s , **a** , **r**) şeklinde bir vaka seçmektir. Erişilen vakalarda çeşitliliği teşvik ederek optimizasyonu sağlamak için maksimum entropi pekiştirmeli öğrenme çerçevesi (Haarnoja ve ark., 2018) uygulanmakta ve aşağıdaki optimizasyon hedefi türetilmektedir:

$$J(\pi) = \mathbb{E}_{\tau \sim p}\left[\sum_{t=0}^{T-1} \left[\mathcal{R}(s_t, a_t) + \alpha \mathcal{H}\left(\mu\left(\cdot|s_t, M_t\right)\right)\right]\right], \quad (3)$$

Burada $\mathcal{H}$ entropiyi, α ise son ödüldeki entropi ağırlığının hiperparametresini ifade etmektedir. Bu çerçevede değer fonksiyonu şöyle tanımlanabilir:

$$V^{\pi}(s_t, M_t) = \sum_{c \in M_t} \mu(c|s_t, M_t)[\, Q^{\pi}(s_t, M_t, c) - \alpha \log \mu\left(c|s_t, M_t\right)\,]. \quad (4)$$

Ayrıca, verilen bir durumda bir "eylem" (yani bir vaka seçimi) gerçekleştirmek için Q değer fonksiyonu şu şekilde tanımlanabilir:

$$Q^{\pi}(s_t, M_t, c_t) = \mathbb{E}_{a \sim p_{\text{LLM}}(\cdot|s_t, c_t), s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)}\left[\mathcal{R}(s_t, a_t) + \gamma V^{\pi}(s_{t+1}, M_{t+1})\right], \quad (5)$$

burada M_{t+1}, (s_t, a_t, r_t) ekstikten sonra güncellenmiş belleği ifade eder. Varsayalım ki d π (s , M) = Σ ∞ $_t$ = 0 γ^{t-1} P (s $_t$ = s , M $_t$ = M)) ifadesi, π altındaki (s , M) indirimli ziyaret sıklığını gösterir. Beklenen değer fonksiyon hedefi şu şekilde tanımlanır:

$$J(\pi) = \mathbb{E}_{(s,M) \sim d^{\pi}}\left[V^{\pi}(s, M)\right] = \mathbb{E}_{(s,M) \sim d^{\pi}}\left[\sum_{c \in M} \mu(c|s, M)\,[Q^{\pi}(s, M, c) - \alpha \log \mu(c|s, M)\right]]. \quad (6)$$

Ardından, optimal geri getirme politikasının kapalı form çözümünü optimal Q değeri üzerinde softmax olarak türetebiliriz :

$$\mu^*(c|s, M) = \frac{\exp(Q^*(s, M, c)/\alpha)}{\sum_{c' \in M} \exp(Q^*(s, M, c')/\alpha)}. \quad (7)$$

Detaylı çıkarım Ek A'da verilmiştir. Bu şekilde, Q-fonksiyonunu öğrenerek optimal geri getirme politikasını türetebiliriz Q ; bu süreç soft Q-öğrenmede (Haarnoja ve ark., 2017) zamansal fark (TD) öğrenimi ile gerçekleştirilebilir:

$$Q(s_t, M_t, c_t) \leftarrow Q(s_t, M_t, c_t) + \eta\left[r_t + \gamma\alpha \log \sum_{c' \in M_{t+1}} \exp\left(Q(s_{t+1}, M_{t+1}, c_{t+1})\right) - Q(s_t, M_t, c_t)\right], \quad (8)$$

Burada η öğrenme oranını ifade eder. Sonraki bölümde, durumlar üzerinde benzerlik çekirdeği öğrenerek Q-fonksiyonunu öğrenmenin daha basit bir yolunu sunuyoruz.

Algorithm 1 Fine-tuning CBR agent with soft Q-learning and state similarity

Require: Kernel network parameters θ, LLM policy p_{LLM}, entropy weight α, discount factor γ, learning rate η, target-network update period K, averaging weight β, initial case bank $M_0 = \varnothing$, initial episodic memory $\mathcal{D} = \varnothing$ and initial replay buffer $\mathcal{B} = \varnothing$

Initialize target retrieval network $\bar{\theta} \leftarrow \theta$
for timestep $t = 0, 1, 2, \ldots$ **do**
  Retrieve: Sample case $c_t \sim \mu_\theta(\cdot \mid s_t, M_t)$ ▷ Memory Reading, following Eq. (7) and Eq. (9)
  Reuse & Revise: Sample action $a_t \sim p_{\text{LLM}}(\cdot \mid s_t, c_t)$
  Execute a_t and observe reward r_t and next state s_{t+1}
  Retain: $M_{t+1} = M_t \cup \{(s_t, a_t, r_t)\}$
  Store transition $(s_t, c_t, r_t, s_{t+1}, M_{t+1})$ in $\mathcal{B}$
  Append Episodic Memory $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, c_t, Q_t)\}$ ▷ Memory Writing
  Sample mini-batch $\{(s_i, c_i, r_i, s_i', M_i')\} \sim \mathcal{B}$
  $\theta \leftarrow \theta - \eta \ \nabla_\theta \mathcal{L}_i$ ▷ Following Eq. (10)
  if $t \bmod K = 0$ **then** ▷ Update target network
    $\bar{\theta} \leftarrow \beta\bar{\theta} + (1-\beta)\theta$
  end if
end for

Enhance Q-Learning Based on State Similarity. As in Eq. (8), we can learn the Q function from scratch via TD learning. However, directly learning the Q function is challenging due to complex state and case descriptions in the form of natural language. To this end, we propose to approximate the Q value via kernel-based estimation, following episodic control (EC) algorithms (Pritzel et al., 2017). Specifically, we maintain an episodic memory $\mathcal{D} = \{(s, c, Q)\}$, including the state, the retrieved case, and the Q value of each interaction. Then, we approximate the Q function via a kernel network $k_\theta(\cdot, \cdot)$, parametrised by θ:

$$Q_{\text{EC}}(s, M, c; \theta) = \sum_{(s', c', Q') \in \mathcal{D}_c} \frac{k_\theta(s, s') Q'}{\sum_{(\hat{s}, \hat{c}, \hat{Q}) \in \mathcal{D}_c} k_\theta(s, \hat{s})}, \tag{9}$$

where $\mathcal{D}_c = \{(s_i, c_i, Q_i) \in \mathcal{D} : c_i = c\}$ denotes the past interactions stored in the episodic memory $\mathcal{D}$ with the same retrieved case c. By substituting Eq. (9) in Eq. (8), we can learn the Q function by optimising the kernel parameter θ via TD learning, i.e.,

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,c,r,s',M,M')} \left[\left(Q_{\text{EC}}(s, M, c; \theta) - \left[r + \gamma\alpha \log \sum_{c' \in M'} \exp(Q_{\text{EC}}(s', M', c'; \bar{\theta})) \right] \right)^2 \right], \tag{10}$$

where $\bar{\theta}$ denotes the target kernel network, s' denotes the next state and $M' = M \cup \{c\}$ denotes the updated case bank. More specifically, we provide the gradient of the TD learning loss with respective of θ as:

$$\nabla_\theta \mathcal{L}(\theta) = 2\, \mathbb{E}_{(s,c,r,s',M,M')} \left[(f_\theta(s, c) - y) \sum_{i \in \mathcal{D}_c} w_i(s, c; \theta) \, (Q_i - f_\theta(s, c)) \ \nabla_\theta \log k_\theta(s, s_i) \right], \tag{11}$$

where $w_i = \frac{k_\theta(s, s_i)}{\sum_{s_j \in \mathcal{D}_c} k_\theta(s, s_j)}$, $f_\theta(s, c) = \sum_{(s_i, Q_i) \in \mathcal{D}_c} w_i Q_i$, and $y = r + \gamma\alpha \log \sum_{c' \in M'} \exp\big(f_{\bar{\theta}}(s', c')\big)$.

Algoritma 1 Soft Q-öğrenme ve durum benzerliği ile CBR ajanının ince ayarı

Gereklidir**:** Çekirdek ağ parametreleri θ, LLM politikası p_{LLM}, entropi ağırlığı α, indirim faktörü γ, öğrenme oranı η, hedef ağ güncelleme periyodu K, ortalama ağırlık β, başlangıç vaka bankası $M_0 = \varnothing$, başlangıç episodik bellek $\mathcal{D} = \varnothing$ ve başlangıç yeniden oynatma tamponu $\mathcal{B} = \varnothing$

Hedef erişim ağını başlat $\bar{\theta} \leftarrow \theta$
zamansaldöngü $t = 0, 1, 2, \ldots$ yap
Erişim : Örnek olgu $c_t \sim \mu_\theta(\cdot \mid s_t, M_t)$ ▷ Bellek Okuma, Denklem (7) ve Denklem (9) uyarınca
Yeniden Kullan ve Revize Et : **Örnek eylem** $a_t \sim p_{\text{LLM}}(\cdot \mid s_t, c_t)$
Eylem a_t uygula ve ödül r_t ile sonraki durumu s_{t+1} gözlemle
  Tutma: $M_{t+1} = M_t \cup \{(s_t, a_t, r_t)\}$
  Geçişi depola $(s_t, c_t, r_t, s_{t+1}, M_{t+1})$ in $\mathcal{B}$
  Episodik Belleği Ekle $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, c_t, Q_t)\}$ ▷ Bellek Yazımı
  Mini-parti Örnekle $\{(s_i, c_i, r_i, s_i', M_i')\} \sim \mathcal{B}$
  $\theta \leftarrow \theta - \eta \ \nabla_\theta \mathcal{L}_i$ ▷ Denklem (10)'u Takip Et
  eğer $t \bmod K = 0$ **ise** ▷ Hedef Ağını Güncelle
    $\bar{\theta} \leftarrow \beta\bar{\theta} + (1-\beta)\theta$
  Son If
Son İçin

Durum Benzerliğine Dayalı **Q**-Öğrenimini Geliştir**.** Denklem (8)'de olduğu gibi, Q fonksiyonu sıfırdan TD öğrenimi ile öğrenilebilir**.** Ancak, Q fonksiyonunu doğrudan öğrenmek, doğal dil biçimindeki karmaşık durum ve örnek açıklamaları nedeniyle zorluktur. Bu amaçla, episodik kontrol (EC) algoritmalarını (Pritzel et al., 2017) takip ederek Q değerini çekirdek tabanlı kestirim ile yaklaşık olarak hesaplamayı öneriyoruz**.** Özellikle, her etkileşimin durumu nu, erişilen vakayı ve Q değerini içeren bir episodik bellek $\mathcal{D} = \{(s, c, Q)\}$ koruyoruz**.** Ardından, parametreleştirilen çekirdek ağı $k_\theta(\cdot, \cdot)$ kullanarak Q fonksiyonunun yaklaşık değerini hesaplıyoruz:

$$Q_{\text{EC}}(s, M, c; \theta) = \sum_{(s', c', Q') \in \mathcal{D}_c} \frac{k_\theta(s, s') Q'}{\sum_{(\hat{s}, c, \mathcal{Q}) \in \mathcal{D}_c} k_\theta(s, \hat{s})}, \tag{9}$$

Burada, $\mathcal{D}_c = \{(s_i, c_i, Q_i) \in \mathcal{D} : c_i = c\}$ aynı erişilen vakaya ait episodik bellekte $\mathcal{D}$ de saklanan geçmiş etkileşimleri gösterir. (9) numaralı denklemi (8) numaralı denkleme ikame ederek, TD öğrenimi ile çekirdek parametresi θ optimize edilerek Q fonksiyonu öğrenilebilir; yani

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,c,r,s',M,M')} \left[\left(Q_{\text{EC}}(s, M, c; \theta) - \left[r + \gamma\alpha \log \sum_{c' \in M'} \exp(Q_{\text{EC}}(s', M', c'; \bar{\theta})) \right] \right)^2 \right], \tag{10}$$

burada $\bar{\theta}$ hedef çekirdek ağını, s' sonraki durumu ve $M' = M \cup \{c\}$ güncellenmiş vaka bankasını ifade etmektedir. Daha spesifik olarak, TD öğrenimi kaybının θ e göre gradyanını aşağıdaki şekilde sunmaktayız:

$$\nabla_\theta \mathcal{L}(\theta) = 2\, \mathbb{E}_{(s,c,r,s',M,M')} \left[(f_\theta(s, c) - y) \sum_{i \in \mathcal{D}_c} w_i(s, c; \theta) \, (Q_i - f_\theta(s, c)) \ \nabla_\theta \log k_\theta(s, s_i) \right], \tag{11}$$

burada $w_i = \frac{k_\theta(s, s_i)}{\sum_{s_j \in \mathcal{D}_c} k_\theta(s, s_j)}$, $f_\theta(s, c) = \sum_{(s_i, Q_i) \in \mathcal{D}_c} w_i Q_i$, ve $y = r + \gamma\alpha \log \sum_{c' \in M'} \exp\big(f_{\bar{\theta}}(s', c')\big)$.

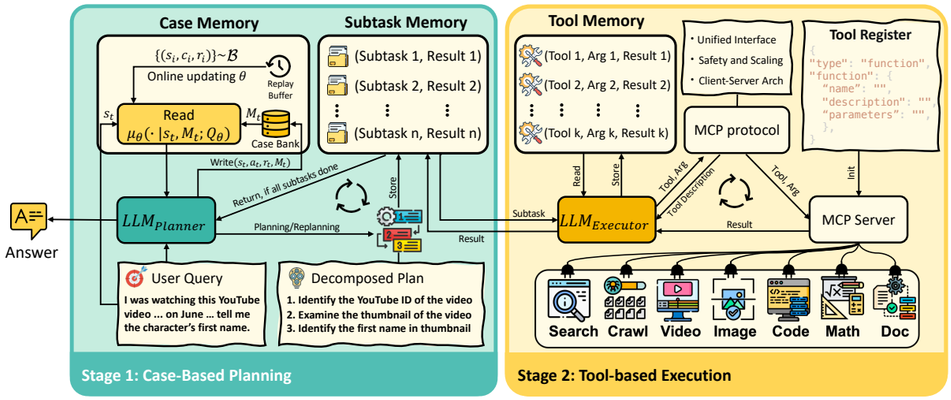


Figure 3: The architecture of *Memento* with parametric memory. *Memento* is instantiated as a planner–executor framework alternating between Case-Based Planning (Stage 1) and Tool-Based Execution (Stage 2). The planner is an LLM-based CBR agent enhanced by a Case Memory module that supports both Write, which records new cases and online refines the Q-function, and Read, which retrieves cases via the learned retrieval policy for adaptive case selection. The executor is an LLM-based MCP client that invokes external tools hosted on the MCP servers through the MCP protocol.

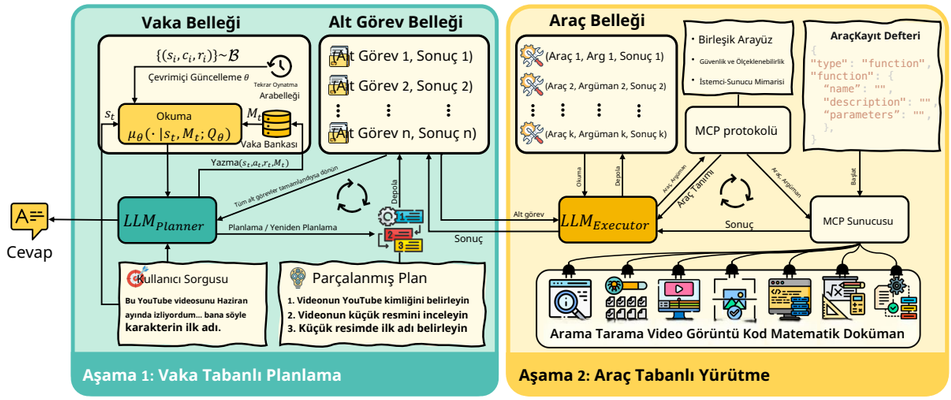
4. Implementation: Deep Research Agent

We implement stateful prompt engineering via M-MDP methodology (§ 3) in Deep Research scenarios (Huang et al., 2025), where agents must solve complex, long-horizon tasks by iteratively interacting with their environment, invoking external tools, retrieving information from external sources, and processing heterogeneous data for dynamic reasoning. As illustrated in Figure 3, *Memento* alternates between two core stages: Case-Based Planning and Tool-Based Execution.

4.1. Framework

To address the challenges of long-horizon reasoning, *Memento* follows the plan-and-act paradigm (Erdogan et al., 2025), where the planner and executor operate in an alternating loop to iteratively advance task completion. For effective coordination, *Memento* integrates three memory modules:Case Memory (vectorised storage of prior cases for high-level planning), Subtask Memory (text-based storage of active subtasks and their results), and Tool Memory (text-based logs of tool interactions for each subtask).

In the planning stage, Planner, instantiated as an LLM-driven CBR agent, receives the task instruction and queries the case memory for relevant case triplets $(s_i, a_i, r_i)_{i=1}^{K}$, where s_i is the task, a_i is the plan, r_i indicates success, and K is the retrieval count. This process is supported by a Case Memory module, which retrieves relevant experiences from a case bank through either a similarity-based retriever or online-updating Q-function, thus enabling the planner to leverage both parametric and non-parametric memory as priors. The retrieved cases are concatenated with the current task instruction to form the prompt, guiding the LLM to generate a plan for each subtask. Once the initial task is decomposed, a Subtask Memory module orchestrates the interaction between the planner and executor, recording generated subtasks and their execution outcomes. After each iteration, the planner uses the accumulated execution history to assess task



Şekil **3:** Parametrik belleğe sahip *Memento* mimarisinin yapısı. Memento, Vaka Tabanlı Planlama (Aşama 1) ile Araç Tabanlı Yürütme (Aşama 2) arasında geçiş yapan bir planlayıcı–yürütücü çerçevesi olarak oluşturulmuştur. Planlayıcı, yeni vakaları kaydeden ve Q-fonksiyonunu çevrimiçi olarak iyileştiren Yazma işlevini destekleyen ve öğrenilmiş erişim politikası aracılığıyla vakaları adaptif şekilde seçmek için erişim sağlayan Vaka Belleği modülü ile güçlendirilmiş LLM tabanlı bir CBR ajanıdır. Yürütücü, MCP protokolü aracılığıyla MCP sunucularında barındırılan dış araçları çağıran LLM tabanlı bir MCP istemcisidir.

4. Uygulama: Derin Araştırma Ajanı

Derin Araştırma senaryolarında (Huang ve ark., 2025), ajanların karmaşık ve uzun vadeli görevleri çözmek için ortamlarıyla yinelemeli etkileşimde bulunmaları, dış araçları çağırmaları, dış kaynaklardan bilgi almaları ve dinamik akıl yürütme için heterojen verileri işlemeleri gereken durumlarda M-MDP metodolojisi (§ 3) aracılığıyla durumlu prompt mühendisliği uygularız. Şekil 3'te gösterildiği üzere, *Memento* iki temel aşama arasında dönüşümlü olarak ilerler: Vaka Tabanlı Planlama ve Araç Tabanlı Yürütme.

4.1. Çerçeve

Uzun vadeli akıl yürütme zorluklarını ele almak amacıyla, *Memento*, planla-ve-uygula paradigmasını (Erdogan ve ark., 2025) izler; bu paradigmayla planlayıcı ve yürütücü, görevin tamamlanmasını yinelemeli olarak ilerletmek için dönüşümlü bir döngü halinde çalışır. Etkili koordinasyon sağlamak için, *Memento* üç bellek modülünü entegre eder: Vaka Belleği (yüksek seviyeli planlama için önceki vakaların vektörleştirilmiş depolanması), Alt Görev Belleği (aktif alt görevlerin ve bunların sonuçlarının metin tabanlı depolanması) ve Araç Belleği (her alt görev için araç etkileşimlerinin metin tabanlı kayıtları).

Planlama aşamasında, LLM destekli bir CBR ajanı olarak türetilen Planlayıcı, görev talimatını alır ve vaka belleği için ilgili vaka üçlülerini $\left(s_i, a_i, r_i\right)_{i=1}^{K}$ olarak sorgular; burada s_i görev, a_i plan, r_i i başarıyı belirtir ve K erişim sayısını ifade eder. Bu süreç, benzerlik tabanlı erişim mekanizması veya çevrimiçi güncellenen Q-fonksiyonu aracılığıyla vaka bankasından ilgili deneyimleri erişen Vaka Belleği modülü tarafından desteklenmekte; böylece planlayıcı, parametrik ve parametresiz belleği önceki bilgi olarak kullanabilmektedir. Erişim yapılan vakalar, mevcut görev talimatı ile birleştirilerek prompt oluşturulur ve LLM'nin her alt görev için plan üretmesini yönlendirir. Başlangıç görevi parçalandıktan sonra, Alt Görev Belleği modülü planlayıcı ile yürütücü arasındaki etkileşimi yönetir ve oluşturulan alt görevler ile bunların yürütme sonuçlarını kaydeder. Her iterasyondan sonra, planlayıcı birikmiş yürütme geçmişini kullanarak görevi değerlendirir.

completion. If the task is unfinished, the planner replans based on updated context; otherwise, the final result is returned, and the case memory is updated with new experiences only upon task completion.

The execution stage is managed by an Executor, powered by a general-purpose LLM, which is responsible for executing each subtask as an autonomous episode (Sumers et al., 2023) using the MCP protocol. Unlike prior agents (Zheng et al., 2025, Weng et al., 2025), *Memento*'s executor supports rich reasoning and flexible tool composition. For each subtask, the executor consults the tool memory, determines the appropriate tool invocation, and updates the results.which operates as a Model Context Protocol (MCP)[1] client. The executor reads pending subtasks from the subtask memory, accesses relevant history from a Tool Memory (scoped per subtask), and determines whether to invoke an external tool or return a result. MCP serves as a standardised, model-agnostic interface, enabling flexible coordination with diverse external tools and data sources. By unifying access under a single protocol layer, *Memento* can seamlessly integrate dynamic reasoning and compositional tool use across multiple domains.

4.2. Case Memory Management

The case memory is an online-growing case bank M_t operated with Write and Read operations, available in non-parametric and parametric variants. In the non-parametric setting, Write simply appends (s_t, a_t, r_t), and Read retrieves cases by similarity for computational efficiency. In the parametric setting, Write further online updates a Q-function to shape the retrieval distribution, while Read is driven by the learned Q-function, thereby realising adaptive case selection. More details are provided in Appendix B.

Memory Storage. Following the CBR agent in Definition 3.2, the Write operation appends each historical case (s_t, a_t, r_t) to the case bank M_t, after each time step t:

$$\text{Write}(s_t, a_t, r_t, M_t) = M_{t+1} = M_t \cup \{(s_t, a_t, r_t)\}. \tag{12}$$

In this process, the state s_t is encoded using a frozen text encoder, while the action a_t and reward r_t are preserved in their original forms, as only the state representation requires vectorisation for subsequent retrieval operations. This Write operation is continuously performed throughout the agent's execution, allowing the case bank to grow into a comprehensive and transferable repository of experiences incrementally. By accumulating both successes and failures, the memory not only enables retrospective analysis for informed avoidance of past mistakes but also provides successful trajectories that prospectively guide future planning.

Non-Parametric Memory Retrieval. A cornerstone of *Memento* is its dynamically evolving Case Bank, which underpins its continual learning capability. At each planning step, this non-parametric memory module receives the task instruction and then retrieves relevant cases, comprising a mixture of successful and failed cases. This CBR method mirrors human analogical learning, where previously encountered outcomes shape decision-making (Aamodt and Plaza, 1994). Specifically, we retrieve the K nearest past cases from the case bank by computing the semantic similarity between the current state and past states. This design follows the mainstream CBR paradigm, which assumes that similar problems should have similar solutions (Wiratunga et al., 2024, Guo et al., 2025), thereby allowing the agent to prioritise cases whose historical contexts are most aligned with the current task. Formally, the Read operator of the non-parametric memory is defined as:

$$\text{Read}_{\text{NP}}(s_t, M_t) = \underset{(s_i, a_i, r_i) \in M_t}{\text{TopK}} \ \text{sim}\big(\text{enc}(s_t), \text{enc}(s_i)\big), \tag{13}$$

[1] https://github.com/modelcontextprotocol

tamamlama. Görev tamamlanmamışsa, planlayıcı güncellenen bağlama göre yeniden plan yapar; aksi takdirde, nihai sonuç döndürülür ve vaka belleği yalnızca görev tamamlandığında yeni deneyimlerle güncellenir.

Yürütme aşaması, MCP protokolünü kullanan ve her alt görevi özerk bir bölüm olarak (Sumers ve ark., 2023) yürüten genel amaçlı bir LLM tarafından desteklenen bir Yürütücü tarafından yönetilir. Önceki ajanların aksine (Zheng ve ark., 2025, Weng ve ark., 2025), *Memento* 'nin yürütücüsü zengin muhakeme ve esnek araç bileşimini destekler. Her alt görev için yürütücü Araç Belleğine başvurur, uygun araç çağrısını belirler ve sonuçları günceller; bu, Model Context Protocol (MCP) [1] istemcisi olarak çalışır. Yürütücü, bekleyen alt görevleri alt görev belleğinden okur, her alt görev için kapsamlı bir Araç Belleğinden ilgili geçmişe erişir ve dış bir aracı çağırıp çağırmama ya da sonucu döndürme kararını verir. MCP, çeşitli dış araçlar ve veri kaynaklarıyla esnek koordinasyon sağlayan, standartlaştırılmış ve modelden bağımsız bir arayüz olarak hizmet vermektedir. Erişimi tek bir protokol katmanı altında birleştirerek, *Memento* çoklu alanlarda dinamik akıl yürütme ve bileşik araç kullanımını sorunsuz şekilde entegre edebilir.

4.2. Vaka Belleği Yönetimi

Vaka belleği; Yazma ve Okuma işlemleriyle çalışan, çevrimiçi büyüyen bir vaka bankası M_t olup, parametrik ve parametrik olmayan versiyonları mevcuttur. Parametrik olmayan durumda, Yazma işlemi sadece (s_t, a_t, r_t) verilerini ekler; Okuma işlemi ise hesaplama verimliliği için benzerlik esaslı erişim sağlar. Parametrik durumda ise, Yazma işlemi erişim dağılımını şekillendirmek için Q-fonksiyonunu çevrimiçi olarak güncellerken, Okuma işlemi öğrenilmiş Q-fonksiyonu tarafından yönlendirilir; böylece uyarlanabilir vaka seçimi gerçekleştirilir. Daha ayrıntılı bilgiler Ek B'de verilmiştir.

Bellek Depolama. Tanım 3.2'deki CBR ajanına uygun olarak, Yazma işlemi her zaman adımından sonra her tarihsel vakayı (s_t, a_t, r_t) vaka bankası M_t'ye ekler:

$$\text{Yazma}(s_t, a_t, r_t, M_t) = M_{t+1} = M_t \cup \{(s_t, a_t, r_t)\}. \tag{12}$$

Bu süreçte, durum s_t, donmuş bir metin kodlayıcı ile kodlanırken, eylem a_t ve ödül r_t orijinal biçimlerinde korunur; çünkü yalnızca durum temsilinin sonraki erişim işlemleri için vektörleştirilmesi gerekmektedir. Bu Yazma işlemi, ajanın çalışma süresi boyunca sürekli olarak gerçekleştirilir ve vaka bankasının deneyimlerin kapsamlı ve aktarılabilir bir deposu olarak kademeli biçimde büyümesine olanak sağlar.
Başarılar ve başarısızlıklar birikerek, bellek yalnızca geçmiş hatalardan kaçınmak için geriye dönük analiz yapılmasını sağlamakla kalmaz, aynı zamanda başarılı yörüngeler sunarak gelecekteki planlamayı yönlendirmeye olanak verir.

Parametrik Olmayan Bellek Erişimi. Memento'nun temel taşlarından biri, sürekli öğrenme yeteneğini destekleyen dinamik olarak gelişen Vaka Bankasıdır. Her planlama adımında, bu parametrik olmayan bellek modülü görev talimatını alır ve ardından başarılı ve başarısız vakalardan oluşan ilgili vakalara erişir. Bu SVÖ yöntemi, daha önce karşılaşılan sonuçların karar verme sürecini şekillendirdiği insan benzetimli öğrenmeyi yansıtır (Aamodt ve Plaza, 1994). Özellikle, güncel durum ile geçmiş durumlar arasındaki anlamsal benzerliği hesaplayarak vaka bankasından K en yakın geçmiş vakayı erişiriz. Bu tasarım, benzer problemlerin benzer çözümlere sahip olması gerektiğini varsayan ana akım SVÖ paradigmasına (Wiratunga vd., 2024, Guo vd., 2025) uygundur ve böylece ajan, tarihsel bağlamları mevcut görevle en çok örtüşen vakalara öncelik verebilir. Formel olarak, parametrik olmayan belleğin Okuma operatörü şu şekilde tanımlanır:

$$\text{Okuma NP}(s\_t, M\_t) = \underset{(s\_i, a\_i, r\_i) \in M\_t}{\text{TopK}} \ \text{sim}\big(\text{enc}(s\_t), \text{enc}(s\_i)\big), \tag{13}$$

[1] https://github.com/modelcontextprotocol

where s_t and M_t denote the query and case bank at time step t, respectively. Here, $\text{enc}(\cdot)$ represents the pretrained textual encoder and $\text{sim}(\cdot)$ denotes the cosine similarity function.

Parametric Memory Retrieval. To empower the agent to selectively leverage high-utility cases to augment planning from past experiences, we design a differential memory mechanism in *Memento* via a parametric Q-function. When writing new cases to the case bank, the parametric method, in contrast to the non-parametric approach that merely appends the tuple as in Eq. (12), concurrently updates the Q-function online. Meanwhile, with CBR applied only for planning in *Memento*, the CBR planner can be simplified to a single-step setting instead of a multi-step M-MDP. This single-step setting collapses the TD target in Eq. (10) to the immediate reward, thereby simplifying the learning objective. Without bootstrapping, the updating reduces to a supervised learning paradigm, which avoids non-stationary targets. Therefore, we can train a parametric Q-function $Q(s, c; \theta)$ end-to-end, dispensing with the kernel-based estimation in Eq. (9). Accordingly, the single-step Q-learning loss can be formulated as:

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,c,r)}\left[\left(Q(s, c; \theta) - r\right)^2\right], \tag{14}$$

where the tuple $\{(s, c, r)\}$ is stored in the replay buffer $\mathcal{B}$ and Q is implemented as a neural network. Noting that the reward signal in deep research tasks is binary ($r \in \{0, 1\}$), we replace the Mean Squared Error (MSE) objective with a cross-entropy (CE) loss, since MSE loss suffers from vanishing gradients near 0/1, whereas CE loss provides more numerically stable signals. Thus, we reformulate the training objective as a binary classification loss:

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,c,r)}\left[-r \log Q(s, c; \theta) - (1 - r) \log\left(1 - Q(s, c; \theta)\right)\right], \tag{15}$$

where Q can be seen as a normalised value representing the probability $p(r = 1|s, c; \theta)$, i.e., the likelihood that the retrieved case c is a good reference for the current state s given the case bank M. Unlike the non-parametric approach that only preserves new cases, the parametric memory also refines the Q-function during Write, enabling each update to both record a new case and update the overall Q-value landscape.

During retrieval, the learned Q-function is used to compute the retrieval policy distribution via Eq. (7), from which cases can be sampled. To reduce the randomness of case selection and enhance the interpretability of the agent's decision process, the Read operation of the parametric memory applies a TopK operator to select the K cases with the highest Q-values, which are used as planning references:

$$\text{Read}_\text{P}(s_t, M_t) = \underset{c_i \in M_t}{\text{TopK}}\, Q(s_t, c_i; \theta). \tag{16}$$

By continually updating the Q-function with new samples, the parametric memory module learns to capture the latent patterns between states and cases, thereby producing a closer approximation to the underlying distribution of the case retrieval policy μ^*.

4.3. Tool Usage

Besides the inherent requirement for long task execution sequences and multi-turn interactions, deep research tasks also place stringent demands on the atomic actions, which require the agent to be able to acquire external information and subsequently process, integrate, and analyse it. Thus, we design a suite of tools for *Memento* accessible via the MCP protocol, comprising modules for information retrieval such as search engines and web crawlers, as well as components for processing and analysing multimodal information, including video and image data, and files in various formats.

Burada s_t ve M_t zaman adımı t de sırasıyla sorguyu ve vaka bankasını ifade eder. Burada, $\text{enc}\left(\cdot\right)$ önceden eğitilmiş metin kodlayıcıyı ve $\text{sim}\left(\cdot\right)$ kosinüs benzerlik fonksiyonunu ifade eder.

Parametrik Bellek Erişimi. Ajanın geçmiş deneyimlerden yüksek faydalı vakaları seçici olarak kullanarak planlamayı artırabilmesi için, *Memento* 'da parametrik bir Q-fonksiyonu aracılığıyla diferansiyel bir bellek mekanizması tasarlıyoruz. Vaka bankasına yeni vakalar yazılırken, yalnızca Eq. (12)'de olduğu gibi kümülü ekleyen parametrik olmayan yaklaşımla karşılaştırıldığında, parametrik yöntem Q-fonksiyonunu çevrimiçi olarak günceller. Bu arada, *Memento* 'da yalnızca planlama için uygulanan CBR ile, CBR planlayıcı çok adımlı M-MDP yerine tek adımlı bir ayara indirgenebilir. Bu tek adımlı ayar, Eq. (10)'daki TD hedefini anlık ödüle indirger ve böylece öğrenme hedefini basitleştirir. Bootstrapping olmadan, güncelleme denetimli öğrenme paradigmasına indirgenir, bu da durağan olmayan hedeflerden kaçınmayı sağlar. Bu nedenle, parametrik bir Q-fonksiyonu $Q\left(s, c; \theta\right)$ baştan sona eğitebilir ve Eq. (9)'daki çekirdek tabanlı kestirimden vazgeçebiliriz. Buna göre, tek adımlı Q-öğrenme kaybı şu şekilde formüle edilebilir:

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,c,r)}\left[\left(Q(s, c; \theta) - r\right)^2\right], \tag{14}$$

Burada küme $\{(s, c, r)\}$ replay tamponunda $\mathcal{B}$ saklanır ve Q bir sinir ağı olarak uygulanır. Derin araştırma görevlerinde ödül sinyalinin ikili olduğu ($r \in \{0, 1\}$) dikkate alındığında, Ortalama Kare Hata (MSE) kaybı; 0/1 yakınlarında gradyanların yok olması nedeniyle, daha sayısal olarak kararlı sinyaller sağlayan çapraz entropi (CE) kaybı ile değiştirilir. Böylece, eğitim hedefi ikili sınıflandırma kaybı olarak yeniden formüle edilir:

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,c,r)}\left[-r \log Q(s, c; \theta) - (1 - r) \log\left(1 - Q(s, c; \theta)\right)\right], \tag{15}$$

burada Q, olasılığı temsil eden normalleştirilmiş bir değer olarak görülebilir $p\left(r = 1 \mid s, c; \theta\right)$; yani erişilen vaka c 'nin, vaka bankası M verildiğinde mevcut durum s için iyi bir referans olma olasılığıdır. Yalnızca yeni vakaların korunduğu parametrik olmayan yaklaşımdan farklı olarak parametrik bellek, Yazma işlemi sırasında Q-fonksiyonunu da iyileştirir ve böylece her güncelleme yeni bir vaka kaydetmenin yanı sıra genel Q-değer manzarasını da günceller.

Erişim sırasında, öğrenilmiş Q-fonksiyonu, vakaların örneklenebileceği erişim politika dağılımını hesaplamak üzere Eşitlik (7) aracılığıyla kullanılır. Vaka seçiminin rastgeleliğini azaltmak ve ajanın karar sürecinin yorumlanabilirliğini artırmak için parametrik belleğin Okuma işlemi, planlama referansı olarak kullanılan en yüksek Q-değerlerine sahip K vakayı seçmek üzere TopK operatörünü uygular:

$$\text{Okuma}_\text{P}\left(s_t, M_t\right) = \underset{c_i \in M_t}{\text{TopK}}\, Q(s_t, c_i; \theta). \tag{16}$$

Q-fonksiyonunu yeni örneklerle sürekli güncelleyerek parametrik bellek modülü, durumlar ile vakalar arasındaki gizli desenleri yakalamayı öğrenir ve böylece vaka geri çağırma politikasının temel dağılımına daha yakın bir yaklaşım üretir μ^*.

4.3. Araç Kullanımı

Uzun görev yürütme dizileri ve çok turlu etkileşimler için doğal gereksinimlerin yanı sıra, derin araştırma görevleri atomik eylemler üzerinde de sıkı gereksinimler getirir; bu durum, ajanın dış bilgileri edinip bunları işlemeyi, entegre etmeyi ve analiz etmeyi mümkün kılmasını zorunlu kılar. Bu nedenle, MCP protokolü aracılığıyla erişilebilen, arama motorları ve web tarayıcıları gibi bilgi geri çağırma modüllerinin yanı sıra video ve resim verileri ile çeşitli formatlardaki dosyaları işleyen ve analiz eden bileşenlerden oluşan bir araç paketi tasarladık.

External Information Acquisition. To support open-ended tasks requiring access to up-to-date external knowledge (e.g., GAIA, BrowseComp), we design a search toolkit that integrates both retrieval and content acquisition capabilities. Specifically, we employ searxng [2], a self-hosted metasearch engine that aggregates results from multiple sources such as Google [3], Bing [4], Duckduckgo [5], and Brave [6]. Retrieved candidates are then re-ranked based on semantic similarity to the query context, ensuring relevance and precision. To supplement this, we incorporate Crawl4AI [7] to fetch and parse the full web content of selected results when deeper understanding is required by the executor. In other words, the search tool functions as a coarse filter based on keyword matching in the user query, while the crawler serves as a fine-grained mechanism to extract detailed information from the retrieved sources when necessary.

Multimodal Heterogeneous Information Processing. To support downstream reasoning over heterogeneous data sources, we implemented a versatile and fine-grained document processing toolkit that automatically extracts information from a broad spectrum of file types and modalities. For example, images are captioned using a vision-language model (VLM); audio is transcribed via automated speech recognition; PowerPoint files are parsed slide-by-slide with embedded image descriptions; spreadsheets are converted to a readable row-wise layout; archives are unpacked; plain text and code files are read directly; JSON and XML are parsed into structured objects; Word documents are translated into Markdown; and videos receive natural-language summaries from VLMs. For PDFs or unsupported formats, a fallback extraction via Chunkr AI [8] or plain-text parsing is used. This toolkit offers a unified interface for accessing and interpreting content across diverse file types and modalities, streamlining the handling of heterogeneous data in real-world scenarios.

Reasoning. The reasoning and analysis toolkit integrates code execution and mathematical computation to support robust, automated analysis within the *Memento* framework. The Code tool provides a sandboxed environment for writing, running, and managing code within a unified workspace. Users can create files, execute shell or Python commands, and inspect outputs – all within a persistent task directory. Python scripts are validated against a security whitelist to ensure safe execution, supporting commonly used libraries such as numpy, pandas, and torch. The workspace maintains state across steps, enabling iterative development. This agent is crucial for solving data analysis, automation, or dynamic code generation tasks. Complementing this, the Math tool handles fundamental arithmetic operations.

5. Experiments

In this paper, we investigate the Deep Research agent, which necessitates tool use and supports multiple rounds of interaction with external, real-world environments. To comprehensively evaluate the agent's capabilities, we select four datasets, each representing a distinct aspect of the research challenge: (i) long-horizon tool use and planning (GAIA) (Mialon et al., 2023), (ii) real-time web-based research (DeepResearcher) (Zheng et al., 2025), (iii) concise factual accuracy (SimpleQA) (Wei et al., 2024), and (iv) exploration at the frontier of human knowledge (HLE) (Phan et al., 2025).

[2] https://github.com/searxng/searxng-docker
[3] https://www.google.com/
[4] https://www.bing.com/
[5] https://duckduckgo.com/
[6] https://brave.com/
[7] https://github.com/unclecode/crawl4ai
[8] https://chunkr.ai/

Dış Bilgi Edinimi. Güncel dış bilgiye erişim gerektiren açık uçlu görevleri desteklemek üzere (ör. GAIA, BrowseComp) geri çağırma ve içerik edinme yeteneklerini birleştiren bir arama araç seti tasarladık. Özellikle, Google 3 , Bing 4 , Duckduckgo 5 ve Brave 6 gibi çeşitli kaynaklardan sonuçları birleştiren kendi kendine barındırılan bir metaarama motoru olan searxng 2 kullanılmaktadır. Elde edilen adaylar, sorgu bağlamıyla anlamsal benzerlik temelinde yeniden sıralanarak alaka ve doğruluk sağlanır. Buna ek olarak, yürütücünün daha derin bir anlayış gerektirdiği durumlarda seçilen sonuçların tam web içeriğini çekmek ve ayrıştırmak için Crawl4AI [7] entegre edilmiştir. Başka bir deyişle, arama aracı kullanıcı sorgusundaki anahtar kelime eşleşmesine dayalı kaba bir filtre işlevi görürken, tarayıcı gerekli olduğunda erişilen kaynaklardan ayrıntılı bilgi çıkarmak için ince taneli bir mekanizma olarak görev yapmaktadır.

Multimodal Heterojen Bilgi İşleme. Heterojen veri kaynakları üzerinde sonraki çıkarımları desteklemek amacıyla, geniş bir dosya türü ve modalite yelpazesinden otomatik olarak bilgi çıkaran çok yönlü ve ince taneli bir belge işleme araç seti uygulanmıştır. Örneğin, görüntüler bir görsel-dil modeli (VLM) kullanılarak başlıklandırılmaktadır ; Ses, otomatik konuşma tanıma yoluyla metne dönüştürülür; PowerPoint dosyaları, gömülü resim açıklamalarıyla slayt slayt ayrıştırılır; Elektronik tablolar, okunabilir satır bazlı bir düzen biçimine dönüştürülür; arşivler açılır ; Düz metin ve kod dosyaları doğrudan okunur; JSON ve XML, yapılandırılmış nesnelere ayrıştırılır; Word belgeleri Markdown formatına çevrilir; Videolara ise Görsel Dil Modelleri (VLM'ler) tarafından doğal dil özetleri oluşturulur . PDF veya desteklenmeyen formatlar için ise Chunkr AI [8] veya düz metin ayrıştırması yoluyla yedek çıkarım uygulanır. Bu araç seti, heterojen verilerin gerçek dünya senaryolarında yönetimini kolaylaştırmak amacıyla, çeşitli dosya türleri ve modaliteler arasında içerik erişimi ve yorumlama için birleşik bir arayüz sunar.

Muhakeme. Muhakeme ve analiz araç seti, Memento çerçevesi kapsamında sağlam ve otomatik analizleri desteklemek amacıyla kod yürütme ve matematiksel hesaplamayı entegre eder. Kod aracı, birleşik bir çalışma alanı içerisinde kod yazımı, çalıştırılması ve yönetilmesi için izole bir ortam sağlar. Kullanıcılar, kalıcı bir görev dizini içerisinde dosya oluşturabilir, shell veya Python komutları çalıştırabilir ve çıktıları inceleyebilir. Python betikleri, güvenli yürütmeyi sağlamak amacıyla numpy, pandas ve torch gibi yaygın kullanılan kütüphaneleri destekleyen bir güvenlik beyaz listesine karşı doğrulanır. Çalışma alanı, yinelemeli geliştirmeye olanak tanıyarak adımlar arasında durumu korur.
Bu ajan, veri analizi, otomasyon veya dinamik kod üretimi görevlerinin çözümünde kritik öneme sahiptir. Buna ek olarak, Matematik aracı temel aritmetik işlemleri gerçekleştirir.

5. Deneyler

Bu makalede, araç kullanımını zorunlu kılan ve gerçek dünya ortamlarıyla çoklu etkileşim turlarını destekleyen Deep Research ajanını inceliyoruz. Ajanın yeteneklerini kapsamlı şekilde değerlendirmek için, araştırma zorluğunun farklı bir yönünü temsil eden dört veri seti seçiyoruz: (i) uzun vadeli araç kullanımı ve planlama (GAIA) (Mialon ve ark., 2023), (ii) gerçek zamanlı web tabanlı araştırma (DeepResearcher) (Zheng ve ark., 2025), (iii) özlü gerçek doğruluğu (SimpleQA) (Wei ve ark., 2024) ve (iv) insan bilgisinin sınırında keşif (HLE) (Phan ve ark., 2025).

[2] https://github.com/searxng/searxng-docker
[3] https://www.google.com/
[4] https://www.bing.com/
[5] https://duckduckgo.com/
[6] https://brave.com/
[7] https://github.com/unclecode/crawl4ai
[8] https://chunkr.ai/

Method	NQ		TQ		HotpotQA		2Wiki		Musique		Bamboogle		PopQA		Avg	
	F1	PM	F1	PM	F1	PM	F1	PM	F1	PM	F1	PM	F1	PM	F1	PM
Prompt Based																
CoT	19.8	32.0	45.6	48.2	24.4	27.9	26.4	27.3	8.5	7.4	22.1	21.6	17.0	15.0	23.6	26.1
CoT + RAG	42.0	59.6	68.9	75.8	37.1	43.8	24.4	24.8	10.0	10.0	25.4	27.2	46.9	48.8	37.7	43.2
Search-o1 (Web) (Li et al., 2025c)	32.4	55.1	58.9	69.5	33.0	42.4	30.9	37.7	14.7	19.7	46.6	53.6	38.3	43.4	35.2	45.0
Training Based																
Search-r1-base (Jin et al., 2025)	45.4	60.0	71.9	76.2	55.9	63.0	44.6	47.9	26.7	27.5	56.5	57.6	43.2	47.0	48.3	53.8
Search-r1-instruct (Jin et al., 2025)	33.1	49.6	44.7	49.2	45.7	52.5	43.4	48.8	26.5	28.3	45.0	47.2	43.0	44.5	39.6	45.6
R1-Searcher (Song et al., 2025)	35.4	52.3	73.1	79.1	44.8	53.1	59.4	65.8	22.8	25.6	64.8	65.6	42.7	43.4	47.1	53.7
DeepResearcher (Zheng et al., 2025)	39.6	61.9	78.4	85.0	52.8	64.3	59.7	66.6	27.1	29.3	71.0	72.8	48.5	52.7	51.8	60.5
Ours																
Memento (GPT-4.1 + o4-mini)	**42.0**	**74.6**	**85.5**	**93.9**	**66.5**	**81.6**	**81.4**	**94.1**	**40.6**	**53.3**	**86.2**	**92.8**	**64.0**	**72.5**	**66.6**	**80.4**

Table 1: Performance comparison of prompt-based, training-based, and our approach on seven open-domain QA datasets. We report the F1 score and PM scores. The last two columns give the weighted average (*Avg*) across all benchmarks, where Bamboogle contributes 125 examples and every other dataset 512 examples. The results of prompt-based and training-based methods using Qwen2.5 (7B) are referred to *DeepResearcher* (Zheng et al., 2025).

5.1. Datasets

To evaluate the general-purpose reasoning capabilities of *Memento*, we adopt the GAIA benchmark (Mialon et al., 2023), which comprises 450 non-trivial questions with unambiguous answers – 300 in the test set and 150 in the validation set. Each question requires varying levels of tool use and autonomous planning, and the dataset is stratified into three difficulty levels: Level 1: Requires approximately 5 steps using a single tool; Level 2: Requires 5–10 steps involving multiple tools; Level 3: Involves up to 50 steps with no restrictions on the number or type of tools. Each level includes a public validation split and a private test split with hidden ground-truth answers and metadata.

We further evaluate *Memento* on broader benchmarks compiled in DeepResearcher (Zheng et al., 2025), which draws from seven open-domain QA datasets: Natural Questions (NQ) (Kwiatkowski et al., 2019), TriviaQA (TQ) (Joshi et al., 2017), HotpotQA (Yang et al., 2018), 2Wiki (Ho et al., 2020), MusiQue (Trivedi et al., 2022), Bamboogle (Press et al., 2022), and PopQA (Mallen et al., 2022). Each dataset contributes 512 examples, except Bamboogle, which provides 125 high-quality samples curated to minimise contamination and emphasise web-based synthesis.

Additionally, we include two challenging benchmarks: 1) SimpleQA (Wei et al., 2024), consisting of 4,330 fact-seeking questions, focuses on factual accuracy. 2) Humanity's Last Exam (HLE) (Phan et al., 2025), with 2,500 questions across diverse academic subjects, assesses the limits of broad-domain reasoning.

5.2. Evaluation Metrics

As each GAIA query has a single reference answer, we follow the GAIA leaderboard and use the Exact Match (EM) metric, which marks a prediction as correct only if it exactly matches the ground-truth answer after standard normalisation (lowercasing, punctuation and article removal, whitespace normalisation). The EM score reflects the percentage of perfectly matched answers.

However, EM cannot accurately reflect the capabilities of an LLM agent, as it overlooks the diversity of expression. We use the macro-F1 score to evaluate the DeepResearcher, SimpleQA, and HLE datasets.

Yöntem	NQ		TQ		HotpotQA		2Wiki		Musique		Bamboogle		PopQA		Ortalama	
	F1	PM	F1	PM	F1	PM	F1	PM	F1	PM	F1	PM	F1	PM	F1	PM
İstem Tabanlı																
CoT	19.8	32.0	45.6	48.2	24.4	27.9	26.4	27.3	8.5	7.4	22.1	21.6	17.0	15.0	23.6	26.1
CoT + RAG	42.0	59.6	68.9	75.8	37.1	43.8	24.4	24.8	10.0	10.0	25.4	27.2	46.9	48.8	37.7	43.2
Search-o1 (Web) (Li ve ark., 2025c)	32.4	55.1	58.9	69.5	33.0	42.4		30.9	37.7	14.7	19.7	46.6	53.6	38.3	43.4	35.2	45.0
Eğitim Tabanlı																
Search-r1-base (Jin ve ark., 2025)	45.4	60.0	71.9	76.2	55.9	63.0	44.6	47.9	26.7	27.5	56.5	57.6	43.2	47.0	48.3	53.8
Search-r1-instruct (Jin ve ark., 2025)	33.1	49.6	44.7	49.2	45.7	52.5	43.4		48.8	26.5	28.3	45.0	47.2	43.0	44.5	39.6	45.6
R1-Searcher (Song ve ark., 2025)	35.4	52.3	73.1	79.1	44.8	53.1	59.4	65.8	22.8	25.6	64.8	65.6	42.7	43.4	47.1	53.7
DeepResearcher (Zheng ve ark., 2025)	39.6	61.9	78.4	85.0	52.8	64.3	59.7	66.6	27.1	29.3	71.0	72.8	48.5	52.7	51.8	60.5
Bizim																
Memento (GPT-4.1 + o4-mini)	**42.0**	**74.6**	**85.5**	**93.9**	**66.5**	**81.6**	**81.4**	**94.1**	**40.6**	**53.3**	**86.2**	**92.8**	**64.0**	**72.5**	**66.6**	**80.4**

Tablo **1:** Yedi açık alan SSS veri kümesi üzerinde istem temelli, eğitim temelli ve bizim yaklaşımımızın performans karşılaştırması. F 1 skora ve PM skorlarına ilişkin sonuçları sunuyoruz. Son iki sütun, Bamboogle'dan 125 örnek ve diğer her veri kümesinden 512 örnek içeren tüm kıyaslamalarda ağırlıklı ortalamayı (Avg) göstermektedir. Qwen 2.5 (7 B) kullanılan istem temelli ve eğitim temelli yöntemlerin sonuçları *DeepResearcher* (Zheng ve ark., 2025) olarak belirtilmiştir.

5.1. Veri Kümeleri

Memento'nun genel amaçlı akıl yürütme yeteneklerini değerlendirmek amacıyla, net yanıtları olan 450 karmaşık sorudan oluşan GAIA kıyaslamasını (Mialon ve ark., 2023) benimsiyoruz – bunların 300'ü test setinde, 150'si doğrulama setindedir. Her soru, değişen düzeylerde araç kullanımı ve otonom planlama gerektirmekte olup, veri kümesi üç zorluk seviyesine ayrılmıştır: Seviye 1: Yaklaşık 5 adımlık tek bir araç kullanımı gerektirir; Seviye 2: Birden fazla araç kullanmayı gerektiren 5-10 adım içerir; Seviye 3: Araçların sayısı veya türü ile sınırlama olmaksızın 50 adıma kadar işlemi kapsar. Her seviye, gizli gerçek yanıtlar ve meta veriler içeren genel doğrulama bölümü ile özel test bölümünü içerir.

Daha geniş ölçütlerde, yedi açık alan SSS veri setinden oluşan ve DeepResearcher (Zheng ve ark., 2025) tarafından derlenen: Natural Questions (NQ) (Kwiatkowski ve ark., 2019), TriviaQA (TQ) (Joshi ve ark., 2017), HotpotQA (Yang ve ark., 2018), 2Wiki (Ho ve ark., 2020), MusiQue (Trivedi ve ark., 2022), Bamboogle (Press ve ark., 2022) ve PopQA (Mallen ve ark., 2022) üzerinde *Memento* 'yu ayrıca değerlendiriyoruz . Her veri seti 512 örnek sağlar; Bamboogle ise kontaminasyonu en aza indirmek ve web tabanlı senteze vurgu yapmak amacıyla seçilmiş 125 yüksek kaliteli örnek sunar.

Ek olarak, iki zorlu kıyaslama seti eklemekteyiz: 1) SimpleQA (Wei et al., 2024), 4.330 gerçek bilgi arayan sorudan oluşmakta olup gerçek doğruluğa odaklanmaktadır. 2) Humanity's Last Exam (HLE) (Phan et al., 2025), çeşitli akademik disiplinlerden 2.500 soruyla geniş kapsamlı alanlardaki muhakeme sınırlarını değerlendirmektedir.

5.2. Değerlendirme Metrikleri

Her GAIA sorgusunun tek bir referans cevabı olduğundan, GAIA lider tablosunu takip ederek, tahmini yalnızca standart normalizasyon (küçültme, noktalama ve bağlaç kaldırma, boşluk normalizasyonu) sonrası yerleşik doğru cevapla tam olarak eşleştiğinde doğru kabul eden Exact Match (EM) metriğini kullanıyoruz. EM skoru, tam olarak eşleşen cevapların yüzdesini yansıtmaktadır.

Ancak EM, LLM ajanının yeteneklerini doğru şekilde yansıtamamaktadır; çünkü ifade çeşitliliğini göz ardı etmektedir. DeepResearcher, SimpleQA ve HLE veri setlerini değerlendirmek için makro-F1 skorunu kullanmaktayız.

Agent name	Model family	Average score (%)	Level 1 (%)	Level 2 (%)	Level 3 (%)
	Valiadation Dataset				
***Memento* (Pass@3)**	**GPT4.1, o3**	**87.88**	**96.23**	**90.70**	61.54
Alita	Claude 4 Sonnet, GPT-4o	87.27	88.68	89.53	**76.92**
Skywork Super Agents v1.1	skywork-agent, Claude 3.7 Sonnet, Whisper	82.42	92.45	83.72	57.69
Langfun Agent	Gemini 2.5 Pro	79.39	88.68	80.23	57.69
AWorld	GPT-4o, DeepSeek-V3, Claude 4, Gemini 2.5 Pro	77.58	88.68	77.91	53.85
Manus	-	73.30	86.50	70.10	57.70
OWL-Workforce	Claude 3.7 Sonnet	69.09	84.91	67.44	42.31
OpenAI DeepResearch	o3	67.40	74.30	69.10	47.60
OWL-Roleplaying	GPT-4o and o3-mini	58.18	81.13	54.65	23.08
Open Deep Research	o1	55.15	67.92	53.49	34.62
	Test Dataset				
Su Zero Ultra	–	**80.40**	**93.55**	77.36	65.31
h2oGPTe Agent v1.6.33	Claude 3.7 Sonnet, Gemini 2.5 Pro	79.73	89.25	**79.87**	61.22
Memento	**GPT4.1, o3**	79.40	90.32	75.47	**71.43**
h2oGPTe Agent v1.6.32	Claude 3.7 Sonnet, Gemini 2.5 Pro	79.07	90.32	77.99	61.22
Aworld	GPT-4o, DeepSeek-V3, Claude 4 sonnet, Gemini 2.5 Pro	77.08	93.55	76.73	46.94

Table 2: Top results on the GAIA Leaderboard as of June 26, 2025, *Memento* achieves the Top-1 performance on the validation set and the test set in open-source agent frameworks.

Meanwhile, Partial Match (PM) indicates the partial semantic match scores between LLMs' generated answers and gold answers. We utilise GPT-4o-mini as the answer evaluator to give the scores and the prompt the same as DeepResearcher (Zheng et al., 2025).

5.3. Model Configurations

The Planner is powered by `GPT-4.1`, the Executor by `o3` for GAIA and `o4-mini` for other datasets, the image processing by `GPT-4o`, the video agent by `Gemini 2.5 Pro` and the audio agent by `Assembly AI`. For the non-parametric CBR, we encode sentences with `SimCSE` and rank candidate cases using cosine similarity. For the parametric CBR, we initialise sentence representations with `SimCSE` and implement the Q-function as a two-layer MLP to assign the Q value. Meanwhile, the CBR planner's state at each step often contains information inherited from previous states. To avoid redundant storage, only the state, action, and reward from the final step of each trajectory are written to memory, ensuring that the case bank remains both compact and informative.

The Offline Executor setting refers to one static executor, removing the planner, case memory, and all external tools, so it reflects only raw parametric knowledge from LLMs. The Online Executor starts from that stripped-down baseline but reconnects the same executor to live search and other MCP tools, reflecting the value of real-time retrieval and tool execution. *Memento* (w/o CBR) keeps episodic memory disabled, allowing us to measure the extra gain delivered specifically by case-based reasoning.

5.4. Experimental Results

Deep Researcher. We include this dataset to test real-time web research, evidence retrieval, cross-page synthesis, and multi-hop reasoning. As shown in Table 1, *Memento* augmented with MCP tools (e.g., search engine, browser) reaches an average 66.6% F1 across the seven DeepResearcher benchmarks, nearly doubling the 37.7% F1 of the CoT + RAG baseline. This demonstrates that real-time, online retrieval tools can rival or

Ajan adı	Model ailesi	Ortalama puan (%)	Seviye 1 (%)	Seviye 2 (%)	Seviye 3 (%)
	Doğrulama Veri Seti				
***Memento* (Pass@3)**	**GPT4.1, o3**	**87.88**	**96.23**	**90.70**	61.54
Alita	Claude 4 Sonnet, GPT-4o	87.27	88.68	89.53	**76.92**
Skywork Süper Ajanlar v1.1	skywork-agent, Claude 3.7 Sonnet, Whisper	82.42	92.45	83.72	57.69
Langfun Ajan	Gemini 2.5 Pro	79.39	88.68	80.23	57.69
AWorld	GPT-4o, DeepSeek-V3, Claude 4, Gemini 2.5 Pro	77.58	88.68	77.91	53.85
Manus	-	73.30	86.50	70.10	57.70
OWL-Workforce	Claude 3.7 Sonnet	69.09	84.91	67.44	42.31
OpenAI DeepResearch	o3	67.40	74.30	69.10	47.60
OWL-Rol Yapma	GPT-4o ve o3-mini	58.18	81.13	54.65	23.08
Open Deep Research	o1	55.15	67.92	53.49	34.62
	Test Veri Seti				
Su Zero Ultra	–	**80.40**	**93.55**	77.36	65.31
h2oGPTe Ajan v1.6.33	Claude 3.7 Sonnet, Gemini 2.5 Pro	79.73	89.25	**79.87**	61.22
Memento	**GPT4.1, o3**	79.40	90.32	75.47	**71.43**
h2oGPTe Ajan v1.6.32	Claude 3.7 Sonnet, Gemini 2.5 Pro	79.07	90.32	77.99	61.22
Aworld	GPT-4o, DeepSeek-V3, Claude 4 Sonnet, Gemini 2.5 Pro	77.08	93.55	76.73	46.94

Tablo 2: 26 Haziran 2025 itibariyle GAIA Lider Tablosundaki En İyi Sonuçlar, *Memento* açık kaynak ajan çerçevelerinde doğrulama ve test setlerinde birinci performansı elde etmektedir.

Bu arada, Kısmi Uyum (PM), LLM'ler tarafından üretilen cevaplar ile altın (gold) cevaplar arasındaki kısmi anlamsal uyum puanlarını göstermektedir. DeepResearcher (Zheng ve ark., 2025) ile aynı promptu kullanarak skorları vermek için cevap değerlendiricisi olarak GPT-4o-mini'yi kullanıyoruz.

5.3. Model Yapılandırmaları

Planlayıcı GPT-4.1 tarafından desteklenmektedir, Yürütücü GAIA için o3 ve diğer veri setleri için o4-mini kullanır, görüntü işleme GPT-4o tarafından gerçekleştirilir, video ajanı Gemini 2.5 Pro ve ses ajanı Assembly AI tarafından yönetilir. Parametrik Olmayan CBR için cümleleri SimCSE ile kodlayıp aday vakaları kosinüs benzerliği kullanarak sıralıyoruz. Parametrik CBR için cümle temsilcilerini SimCSE ile başlatıyor ve Q-fonksiyonunu iki katmanlı bir MLP olarak uygulayarak Q değerini atıyoruz. Bu arada, CBR planlayıcısının her adımdaki durumu genellikle önceki durumlardan miras kalan bilgileri içerir. Yinelenen depolamayı önlemek için her trajektorinin yalnızca son adımından durum, eylem ve ödül belleğe yazılır; böylece vaka bankası hem kompakt hem de bilgi açısından zengin kalır.

Çevrimdışı Yürütücü ayarı, tek bir statik yürütücüye karşılık gelir ve planlayıcıyı, vaka belleğini ve tüm dış araçları kaldırır ; bu nedenle yalnızca LLM'lerden gelen ham parametrik bilgileri yansıtır. Çevrimiçi Yürütücü ise bu sadeleştirilmiş başlangıç noktasından hareketle aynı yürütücüyü canlı arama ve diğer MCP araçlarına yeniden bağlar; böylece gerçek zamanlı arama ve araç yürütmenin değerini gösterir. *Memento* (vaka tabanlı akıl yürütme olmadan) episodik belleği devre dışı bırakır, böylece bize özel olarak vaka tabanlı akıl yürütmenin sağladığı ek faydayı ölçme imkanı sunar.

5.4. Deneysel Sonuçlar

Derin Araştırmacı. Bu veri setini gerçek zamanlı web araştırması, kanıt toplama, sayfalar arası sentez ve çok adımlı akıl yürütmeyi test etmek için dahil ettik. Tablo 1'de gösterildiği gibi, Memento, MCP araçları (örneğin, arama motoru, tarayıcı) ile desteklenmiş haliyle, yedi DeepResearcher kıyaslama testinde ortalama %66,6 F1 değerine ulaşmakta, bu da CoT + RAG temel modelinin %37,7 F1 değerini neredeyse iki katına çıkarmaktadır. Bu, gerçek zamanlı, çevrimiçi toplama araçlarının rekabet edebileceğini veya

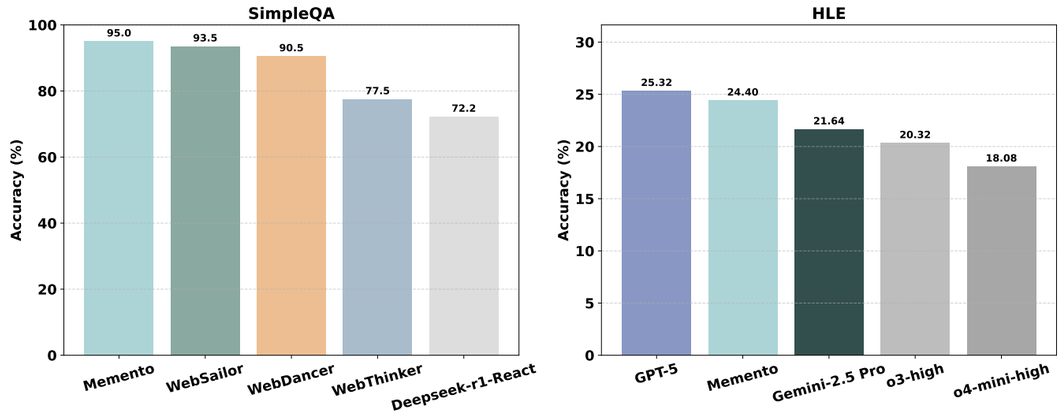


Figure 4: Performance on SimpleQA and HLE. The SimpleQA results are from WebSailor (Li et al., 2025b), and the HLE results are from the official website.

even exceed carefully curated static databases.

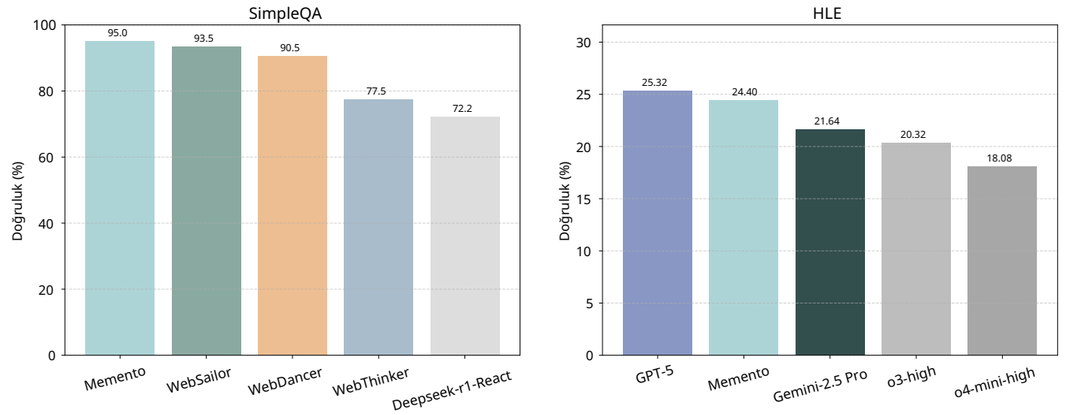
GAIA (Validation & Test). To assess robustness in long-horizon planning, tool orchestration, and execution, we employ the GAIA benchmark. *Memento* attains the top-1 ranking on the validation set and 4th place on the test set, outperforming most existing agent frameworks (Table 2). Notably, it surpasses widely used open-source frameworks, including Manus (Liang et al., 2025), Aworld (Alibaba, 2025), and OWL (Camel-AI, 2025), on both validation and test sets.

For the GAIA validation evaluation, we initialise memory from scratch and iteratively store both successful and failed trajectories in the case bank over three iterations. Using GPT-4.1 as the planner and o3 as the executor, *Memento* achieves 87.88% accuracy on the validation set. For the GAIA test set, performance is based solely on the case bank accumulated during validation, yielding an accuracy of 79.40%. Although *Memento* demonstrates strong overall performance, challenges remain for Level 3 tasks that require extended reasoning horizons and advanced tool coordination.

Humanity's Last Exam (HLE). To evaluate the frontier of human knowledge and the complex reasoning ability in long-tail, specialised domains, we include the HLE [9]. Using our planner executor architecture, with planner GPT-4.1 and executor o4-mini with tools, *Memento* attains 24.4% PM, ranking second overall and within 0.92 points of GPT-5 at 25.32%, while outperforming Gemini-2.5-Pro at 21.64%, o3-high at 20.32%, and o4-mini-high at 18.08%. These results demonstrate that continual learning through CBR effectively transforms episodic experiences into reusable knowledge, offering a complementary pathway to generalisation even in long-tail domains where conventional tool usage and retrieval methods struggle.

SimpleQA. To evaluate *Memento*'s reliability and robustness against hallucination in single-hop factual question answering, we employ the SimpleQA benchmark. As illustrated in Figure 4, *Memento*, implemented with a planner–executor framework (GPT-4.1 as the planner and o4-mini as the executor) augmented with tool use, achieves the highest accuracy among all baselines. Specifically, it reaches an accuracy of

[9] https://scale.com/leaderboard/humanitys_last_exam



Şekil 4: SimpleQA ve HLE üzerindeki performans. SimpleQA sonuçları WebSailor'dan (Li ve ark., 2025b), HLE sonuçları ise resmi web sitesinden alınmıştır.

hatta özenle seçilmiş statik veri tabanlarını bile aşar.

GAIA (Doğrulama & Test). Uzun vadeli planlama, araç orkestrasyonu ve yürütmede sağlamlığı değerlendirmek amacıyla GAIA kıyaslaması kullanılmıştır. *Memento*, doğrulama setinde en üst 1. sırayı ve test setinde 4. sırayı alarak mevcut ajan çerçevelerinin çoğunu geride bırakmaktadır (Tablo 2). Özellikle Manus (Liang vd., 2025), Aworld (Alibaba, 2025) ve OWL (Camel-AI, 2025) gibi yaygın kullanılan açık kaynaklı çerçeveleri hem doğrulama hem de test setlerinde geçmektedir.

GAIA doğrulama değerlendirmesinde bellek sıfırdan başlatılmış ve vaka bankasında başarılı ile başarısız rotalar üç iterasyon boyunca yinelemeli şekilde depolanmıştır. Planlayıcı olarak GPT-4.1 ve yürütücü olarak o3 kullanıldığında, *Memento* doğrulama setinde %87,88 doğruluk elde etmektedir. GAIA test seti performansı ise yalnızca doğrulama aşamasında biriktirilen vaka bankasına dayanmakta ve %79,40 doğruluk göstermektedir. Memento, güçlü genel performans sergilemesine rağmen, uzun süreli çıkarım ufukları ve gelişmiş araç koordinasyonu gerektiren Seviye 3 görevlerinde zorluklar devam etmektedir.

İnsanlığın Son Sınavı **(HLE).** İnsanoğlunun bilgi sınırlarını ve uzun kuyruklu, uzmanlaşmış alanlardaki karmaşık muhakeme yeteneğini değerlendirmek amacıyla HLE [9] 'nı dâhil ettik. Planlayıcı yürütücü mimarimizle, planlayıcı GPT- 4.1 ve yürütücü o 4 -mini ile araçları kullanarak, Memento %24.4 PM başarısını elde etmiş, genel sıralamada ikinci sırada yer almış ve GPT- 5 'in %2 5.32 puanından sadece 0.92 puan geride kalmıştır; ayrıca, Gemini- 2.5 -Pro'da %21.64, o 3 -high'da %20.32 ve o 4 -mini-high'da % 18.08 puanların üzerinde performans göstermiştir. Bu sonuçlar, CBR yoluyla sürekli öğrenmenin epizodik deneyimleri yeniden kullanılabilir bilgiye etkin bir şekilde dönüştürdüğünü ve böylece geleneksel araç kullanımı ile bilgi getirme yöntemlerinin zorlandığı uzun kuyruklu alanlarda bile genelleme için tamamlayıcı bir yol sunduğunu göstermektedir.

SimpleQA. Tek atlamalı gerçek bilimsel soru yanıtlama alanında *Memento* 'nun güvenilirliği ve halüsinasyonlara karşı dayanıklılığını değerlendirmek için SimpleQA kıyaslaması kullanılmıştır. Şekil 4'te gösterildiği gibi, *Memento* , araç kullanımıyla desteklenen planlayıcı–yürütücü çerçevesi (planlayıcı olarak GPT-4.1 ve yürütücü olarak o4-mini) ile tüm temel modellere kıyasla en yüksek doğruluğa ulaşmıştır. Özellikle, doğruluk oranı

[9] https://scale.com/leaderboard/humanitys_last_exam

Dataset	K=0		K=1		K=2		K=4		K=8		K=16		K=32	
	F1	PM	F1	PM	F1	PM	F1	PM	F1	PM	F1	PM	F1	PM
NQ (Kwiatkowski et al., 2019)	39.5	67.8	41.1	74.4	41.3	72.7	41.9	73.0	41.7	73.8	42.1	73.2	42.2	75.4
TQ (Joshi et al., 2017)	81.1	89.1	86.1	93.8	86.2	93.9	86.3	94.1	85.8	94.1	85.9	94.3	85.5	93.9
HotpotQA (Yang et al., 2018)	62.0	76.0	65.4	80.7	65.7	81.3	67.4	84.2	66.6	82.0	65.5	82.0	66.4	83.2
2Wiki (Ho et al., 2020)	78.3	90.0	81.3	94.9	80.9	94.1	81.0	94.7	82.0	94.5	81.1	93.6	81.0	94.1
Musique (Trivedi et al., 2022)	35.6	43.8	39.8	50.2	40.1	52.1	41.6	51.0	41.0	52.1	39.6	51.4	40.3	50.4
Bamboogle (Press et al., 2022)	77.5	83.2	85.9	91.2	84.1	91.2	84.7	90.4	84.9	91.2	85.2	92.0	83.0	87.2
PopQA (Mallen et al., 2022)	57.9	63.9	62.6	70.1	63.4	71.3	63.6	70.9	62.7	69.2	64.1	70.9	63.2	69.4
Average	59.9	72.2	63.6	77.9	63.7	78.1	**64.5**	**78.5**	64.1	78.2	63.9	78.1	63.9	78.1

Table 3: The performance of Memento on the DeepResearcher dataset across different numbers of cases. We use gpt-4.1 as the planner and o4-mini as the executor.

Method	Iter 1	Iter 2	Iter 3	Iter 4	Iter 5
Baseline					
Memento w/o CBR	78.65	80.93	82.62	83.53	84.47
Case-based Continual Learning					
Memento w/ Non-Parametric CBR	79.84	81.87	83.09	84.03	84.85
Memento w/ Parametric CBR	**80.46**	**82.84**	**84.10**	**84.85**	**85.44**

Table 4: Performance improvement of *Memento* over five learning iterations on the DeepResearcher dataset, demonstrating the benefit of accumulating cases in the Case Bank.

95.0%, outperforming WebSailor (93.5%), WebDancer (90.5%), WebThinker (77.5%), and DeepSeek-r1-React (72.2%). These results demonstrate that *Memento* provides strong factual reliability and substantially mitigates hallucination on straightforward single-hop queries, establishing a new state-of-the-art over prior web-agent baselines.

5.5. Ablation Studies

We analyse *Memento*'s hyper-parameter selection, component-wise Analysis, learning curves for both parametric and non-parametric case-based reasoning, out-of-distribution performance, and token costs.

5.5.1. Hyper-parameter Selection

Increasing the number of retrieved cases in CBR raises computational cost and can introduce noise from irrelevant examples. To evaluate this, we vary K in $0, 2, 4, 8, 16, 32$ on the DeepResearcher dataset. As shown in Table 3, performance improves up to $K = 4$ – yielding the highest F1 (64.5) and PM (78.5) – but plateaus or slightly declines for larger K. This suggests that CBR benefits from a small, high-quality memory, unlike few-shot prompting, where more examples often help (Agarwal et al., 2024). Careful case selection and memory curation are thus crucial for continual learning.

5.5.2. Component-wise Analysis

From Table 5, we observe a consistent pattern across HLE, SimpleQA, and DeepResearcher. Moving from an offline executor to live tools generally reduces hallucination and increases both F1 and PM, though the

Veri seti	K=0		K=1		K=2		K=4		K=8		K=16		K=32	
	F1	PM	F1	PM	F1	PM	F1	PM	F1	PM	F1	PM	F1	PM
NQ (Kwiatkowski ve ark., 2019)	39.5	67.8	41.1	74.4	41.3	72.7	41.9	73.0	41.7	73.8	42.1	73.2	42.2	75.4
TQ (Joshi ve ark., 2017)	81.1	89.1	86.1	93.8	86.2	93.9	86.3	94.1	85.8	94.1	85.9	94.3	85.5	93.9
HotpotQA (Yang et al., 2018)	62.0	76.0	65.4	80.7	65.7	81.3	67.4	84.2	66.6	82.0	65.5	82.0	66.4	83.2
2Wiki (Ho et al., 2020)	78.3	90.0	81.3	94.9	80.9	94.1	81.0	94.7	82.0	94.5	81.1	93.6	81.0	94.1
Musique (Trivedi et al., 2022)	35.6	43.8	39.8	50.2	40.1	52.1	41.6	51.0	41.0	52.1	39.6	51.4	40.3	50.4
Bamboogle (Press et al., 2022)	77.5	83.2	85.9	91.2	84.1	91.2	84.7	90.4	84.9	91.2	85.2	92.0	83.0	87.2
PopQA (Mallen et al., 2022)	57.9	63.9	62.6	70.1	63.4	71.3	63.6	70.9	62.7	69.2	64.1	70.9	63.2	69.4
Ortalama	59.9	72.2	63.6	77.9	63.7	78.1	**64.5**	**78.5**	64.1	78.2	63.9	78.1	63.9	78.1

Tablo **3:** Memento'nun DeepResearcher veri setindeki performansı, farklı vaka sayıları bazında. Planlayıcı olarak gpt-4.1 ve yürütücü olarak o4-mini kullanıyoruz.

Yöntem	Yineleme 1	Yineleme 2	Yineleme 3	Yineleme 4	Yineleme 5
Temel					
CBR olmadan Memento	78.65	80.93	82.62	83.53	84.47
Vaka Tabanlı Sürekli Öğrenme					
Parametrik Olmayan CBR ile Memento	79.84	81.87	83.09	84.03	84.85
Parametrik CBR ile Memento	**80.46**	**82.84**	**84.10**	**84.85**	**85.44**

Tablo **4:** DeepResearcher veri setinde beş öğrenme yinelemesi boyunca Memento'nun performans artışı, Vaka Bankasında biriken vakaların faydasını göstermektedir.

%95,0 ile WebSailor (%93,5), WebDancer (%90,5), WebThinker (%77,5) ve DeepSeeker1-React (%72,2) modellerini geride bırakmaktadır. Bu sonuçlar, Memento'nun güçlü olgusal güvenilirlik sağladığını ve basit tek atlamalı sorgulamalarda halüsinasyonları önemli ölçüde azalttığını göstererek önceki web ajanı tabanlı yaklaşımlar üzerinde yeni bir en iyi durumu (state-of-the-art) ortaya koymaktadır.

5.5. Ablasyon Çalışmaları

Memento'nun hiperparametre seçimini, bileşen bazlı analizini, parametrik ve parametrik olmayan vaka tabanlı akıl yürütmenin öğrenme eğrilerini, dağılım dışı (out-of-distribution) performansını ve token maliyetlerini analiz ediyoruz.

5.5.1. Hiperparametre Seçimi

CBR'de erişilen vaka sayısının artırılması hesaplama maliyetini yükseltmekte ve alakasız örneklerden kaynaklanan gürültüyü artırabilmektedir. Bunu değerlendirmek için DeepResearcher veri setinde K değerini $0, 2, 4, 8, 16, 32$ olarak değiştirdik . Tablo 3'te görüldüğü üzere, performans $K = 4$ değerine kadar artmakta – en yüksek F1 (64,5) ve PM (78,5) – ancak daha büyük K değerlerinde plato yapmakta veya hafifçe düşmektedir. Bu durum, CBR'nin, genellikle daha fazla örneğin yararlı olduğu few-shot promptlamanın aksine, küçük ve yüksek kaliteli bir bellekten fayda sağladığını göstermektedir (Agarwal ve ark., 2024). Bu nedenle vaka seçimi ve bellek düzenlemesi sürekli öğrenmede kritik öneme sahiptir.

5.5.2. Bileşen Bazlı Analiz

Tablo 5'ten, HLE, SimpleQA ve DeepResearcher genelinde tutarlı bir örüntü gözlemliyoruz. Çevrimdışı yürütücüden canlı araçlara geçiş, genellikle halüsinasyonu azaltmakta ve hem F1 hem de PM değerlerini artırmakta, ancak

Model	Humanities/SC	Math	Chemistry	Other	Physics	Engineering	Biology/Medicine	CS/AI	Avg
Offline Executor	5.2/9.6	7.1/5.8	2.3/7.9	2.9/11.4	7.6/4.6	12.8/14.0	5.2/17.7	6.5/11.9	6.4/8.7
Online Executor	10.8/24.9	13.1/16.0	6.9/9.3	14.0/16.3	7.8/10.2	7.5/5.3	5.7/17.2	13.3/15.4	11.2/15.8
Memento w/o CBR	25.5/29.2	24.9/16.3	17.4/21.1	24.8/24.1	18.4/10.8	15.8/8.8	10.0/18.7	25.4/12.4	22.2/17.4
Memento	**28.4/33.0**	**30.9/24.2**	**18.7/22.7**	**28.5/32.4**	**22.9/19.1**	**15.9/12.1**	**14.0/26.1**	**28.5/18.5**	**26.7/24.4**

(a) HLE

Model	Art	Geography	Science & Tech	Politics	Sports	Other	TV Shows	Music	History	Video Games	Avg
Offline Executor	16.5/19.6	25.7/31.1	20.1/24.7	25.8/24.8	18.8/19.0	15.8/14.9	12.6/13.3	15.7/17.6	25.9/26.6	15.5/13.3	19.7/21.5
Online Executor	49.7/82.9	45.1/82.5	59.4/87.7	51.6/86.9	46.6/90.1	48.7/86.4	39.5/84.3	43.1/88.5	49.7/83.8	34.7/78.6	48.5/84.8
Memento w/o CBR	83.8/92.2	71.8/84.9	87.1/94.1	83.7/90.8	77.4/86.4	81.2/90.7	70.0/81.6	80.7/89.1	79.1/86.1	81.2/88.9	81.0/89.7
Memento	**86.9/96.4**	**76.6/91.7**	**89.3/96.9**	**87.1/95.6**	**82.2/93.5**	**84.2/95.4**	**77.3/90.8**	**83.3/95.0**	**85.7/94.8**	**86.3/95.6**	**84.7/95.0**

(b) SimpleQA

Method	NQ	TQ	HotpotQA	2Wiki	Musique	Bamboogle	PopQA	Avg
Offline Executor	39.7/70.1	75.8/89.1	50.7/67.2	44.8/56.0	26.9/35.7	76.0/84.0	48.0/53.5	48.8/62.8
Online Executor	23.3/55.3	41.9/80.7	34.4/67.8	33.6/66.2	23.0/39.7	45.8/77.6	24.9/50.2	30.8/60.7
Memento w/o CBR	39.5/67.8	81.8/89.1	62.0/76.0	78.3/90.0	35.6/43.8	77.5/83.2	57.9/63.9	59.9/72.2
Memento	**42.0/74.6**	**85.5/93.9**	**66.5/81.6**	**81.4/94.1**	**40.6/53.3**	**86.2/92.8**	**64.0/72.5**	**66.6/80.4**

(c) DeepResearcher

Table 5: Ablation results across three benchmarks. Each cell shows *F1/PM*. We use gpt-4.1 as the planner and o4-mini as the executor.

magnitude depends on task type (SimpleQA: +28.8 F1 / +63.3 PM, HLE: +4.8 / +7.1), and may even hurt on open-domain data (DeepResearcher: −18.0 / −2.1). Introducing planning (*Memento* w/o CBR) yields robust gains on each benchmark (HLE: +11.0 / +1.6, SimpleQA: +32.5 / +4.9, DeepResearcher: +29.1 / +11.5), indicating that explicit decomposition and tool orchestration systematically improve execution. Finally, case-based reasoning provides consistent, additive improvements (HLE: +4.5 / +7.0, SimpleQA: +3.7 / +5.3, DeepResearcher: +6.7 / +8.2). For HLE, however, without sufficient domain knowledge encoded in the backbone model, neither tool usage nor planning alone can reliably produce correct answers on long-tail, expert-level tasks. For DeepResearcher, we also identify data contamination (Shumailov et al., 2024) across the seven evaluated benchmarks, evidenced by a noticeable drop in both F1 and PM when moving from the offline executor to the online executor without planning (DeepResearcher: −18.0 F1 / −2.1 PM). This aligns with broader findings in the field (Sun et al., 2022, Yu et al., 2022, Zhou et al., 2025): simply using external knowledge can sometimes negatively affect the model, while the internal knowledge within the model plays an important role in QA tasks and can even outperform RAG.

5.5.3. Continual Learning Ability Boosted by Parametric and Non-Parametric CBR

Figure 1c and Table 4 present the continual learning curves across different memory designs for the *Memento* framework, comparing the performance of three configurations: *Memento* with non-parametric CBR or parametric CBR and *Memento* without CBR. The results demonstrate that the full *Memento* architecture consistently outperforms the ablated versions across all iterations, achieving higher accuracy at each step. Notably, removing CBR leads to a noticeable decline in performance, highlighting the effectiveness and complementary benefits of both parametric CBR and non-parametric CBR components in enhancing the

Model	Beşeri Bilimler/SC	Matematik	Kimya	Diğer	Fizik	Mühendislik	Biyoloji/Tıp	Bilgisayar Bilimi/YZ	Ortalama
Çevrimdışı Yürütücü	5.2/9.6	7.1/5.8	2.3/7.9	2.9/11.4	7.6/4.6	12.8/14.0	5.2/17.7	6.5/11.9	6.4/8.7
Çevrimiçi Yürütücü	10.8/24.9	13.1/16.0	6.9/9.3	14.0/16.3	7.8/10.2	7.5/5.3	5.7/17.2	13.3/15.4	11.2/15.8
CBR olmadan Memento	25.5/29.2	24.9/16.3	17.4/21.1	24.8/24.1	18.4/10.8	15.8/8.8	10.0/18.7	25.4/12.4	22.2/17.4
Memento	**28.4/33.0**	**30.9/24.2**	**18.7/22.7**	**28.5/32.4**	**22.9/19.1**	**15.9/12.1**	**14.0/26.1**	**28.5/18.5**	**26.7/24.4**

(a) HLE

Model	Sanat	Coğrafya	Bilim & Teknoloji	Politika	Spor	Diğer	Televizyon Programları	Müzik	Tarih	Video Oyunları	Ortalama
Çevrimdışı Yürütücü	16.5/19.6	25.7/31.1	20.1/24.7	25.8/24.8	18.8/19.0	15.8/14.9	12.6/13.3	15.7/17.6	25.9/26.6	15.5/13.3	19.7/21.5
Çevrimiçi Yürütücü	49.7/82.9	45.1/82.5	59.4/87.7	51.6/86.9	46.6/90.1	48.7/86.4	39.5/84.3	43.1/88.5	49.7/83.8	34.7/78.6	48.5/84.8
CBR olmadan Memento	83.8/92.2	71.8/84.9	87.1/94.1	83.7/90.8	77.4/86.4	81.2/90.7	70.0/81.6	80.7/89.1	79.1/86.1	81.2/88.9	81.0/89.7
Memento	**86.9/96.4**	**76.6/91.7**	**89.3/96.9**	**87.1/95.6**	**82.2/93.5**	**84.2/95.4**	**77.3/90.8**	**83.3/95.0**	**85.7/94.8**	**86.3/95.6**	**84.7/95.0**

(b) SimpleQA

Yöntem	NQ	TQ	HotpotQA	2Wiki	Musique	Bamboogle	PopQA	Ortalama
Çevrimdışı Yürütücü	39.7/70.1	75.8/89.1	50.7/67.2	44.8/56.0	26.9/35.7	76.0/84.0	48.0/53.5	48.8/62.8
Çevrimiçi Yürütücü	23.3/55.3	41.9/80.7	34.4/67.8	33.6/66.2	23.0/39.7	45.8/77.6	24.9/50.2	30.8/60.7
CBR olmadan Memento	39.5/67.8	81.8/89.1	62.0/76.0	78.3/90.0	35.6/43.8	77.5/83.2	57.9/63.9	59.9/72.2
Memento	**42.0/74.6**	**85.5/93.9**	**66.5/81.6**	**81.4/94.1**	**40.6/53.3**	**86.2/92.8**	**64.0/72.5**	**66.6/80.4**

(c) DeepResearcher

Tablo **5:** Üç benchmark üzerindeki ablation sonuçları. Her hücre *F1/PM* değerlerini göstermektedir. Planlayıcı olarak gpt- 4.1 ve yürütücü olarak o 4 -mini kullanıyoruz.

Büyüklük görev türüne bağlıdır (SimpleQA: + 28.8 F1 / + 63.3 PM, HLE: + 4.8 / + 7.1) ve açık alan verisinde zarar verebilir (DeepResearcher: − 18.0 / − 2.1). Planlamanın tanıtılması (CBR olmadan *Memento*), her benchmark'ta anlamlı kazanımlar sağlamaktadır (HLE: + 11.0 / + 1.6 , SimpleQA: + 32.5 / + 4.9 , DeepResearcher: + 29.1 / + 11.5); bu da açık ayrıştırma ve araç orkestrasyonunun yürütmeyi sistematik biçimde iyileştirdiğini göstermektedir. Son olarak, vaka tabanlı akıl yürütme tutarlı ve ekleyici iyileştirmeler sağlamaktadır (HLE: + 4.5 / + 7.0 , SimpleQA: + 3.7 / + 5.3 , DeepResearcher: + 6.7 / + 8.2). Ancak HLE için, omurga modelinde yeterli alan bilgisi kodlanmadan ne araç kullanımı ne de planlama uzun kuyruklu, uzman düzeyindeki görevlerde güvenilir şekilde doğru cevaplar üretebilir. DeepResearcher için, ayrıca yedi değerlendirilen kıyaslama arasında veri kontaminasyonu (Shumailov et al., 2024) tespit ediyoruz; bu, çevrimdışı yürütücüden çevrimiçi yürütücüye planlama olmadan geçişte hem F 1 hem de PM'de belirgin bir düşüşle kanıtlanmaktadır (DeepResearcher: − 18.0 F 1 / − 2.1 PM). Bu durum alandaki daha geniş bulgularla uyumludur (Sun et al., 2022, Yu et al., 2022, Zhou et al., 2025): yalnızca dış bilgi kullanımı bazen modeli olumsuz etkileyebilirken, model içindeki bilgi QA görevlerinde önemli bir rol oynar ve hatta RAG'i geride bırakabilir.

5.5.3. Parametrik ve Parametrik Olmayan CBR ile Güçlendirilmiş Sürekli Öğrenme Yeteneği

Şekil 1c ve Tablo 4, Memento çerçevesi için farklı bellek tasarımlarına göre sürekli öğrenme eğrilerini sunmakta, üç yapılandırmanın performansını karşılaştırmaktadır: parametrik olmayan CBR veya parametrik CBR içeren Memento ve CBR'siz Memento. Sonuçlar, tam Memento mimarisinin tüm iterasyonlar boyunca ablatif versiyonları tutarlı şekilde geride bıraktığını ve her adımda daha yüksek doğruluk sağladığını göstermektedir. Özellikle, CBR'nin kaldırılması performansta belirgin bir düşüşe neden olmakta ve parametrik CBR ile parametrik olmayan CBR bileşenlerinin etkili ve tamamlayıcı faydalarını vurgulamaktadır.

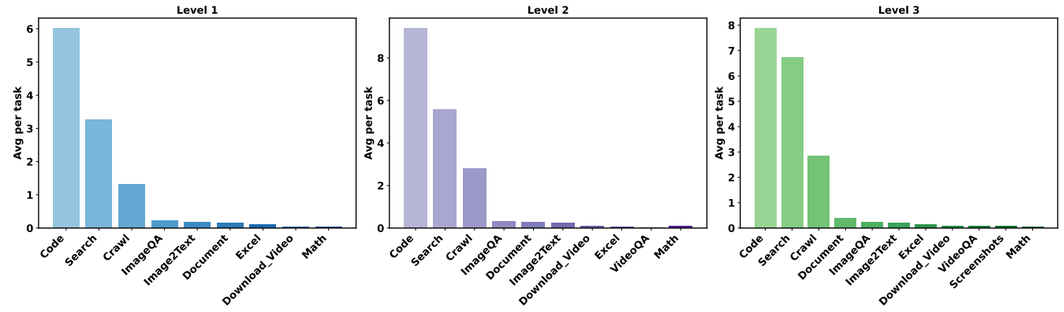


Figure 5: The average number of each task type per level, highlighting the dominance of code, search, and crawl tasks as difficulty level increases.

continual learning capability of *Memento*. More importantly, we observe a learning curve of the accuracy on the DeepResearcher dataset with increased iterations, suggesting that memory-based approaches can effectively enhance LLM agents without requiring parameter updates.

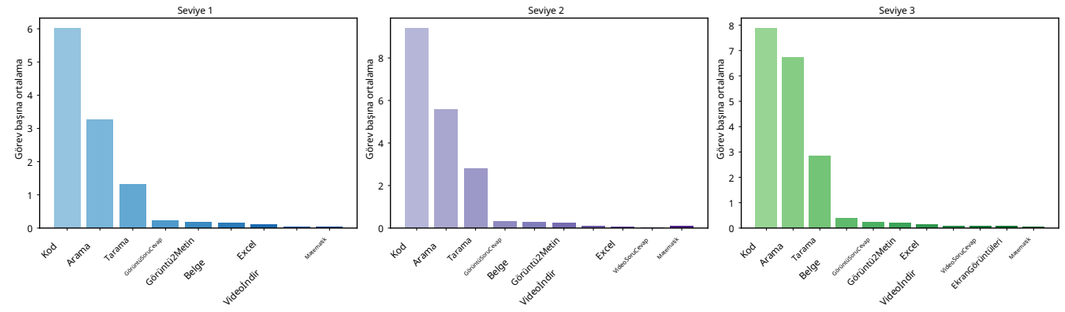
Although we attempted to locate any performance drops along the learning curve, in practice, such inflexion points are elusive. With only about 3k training data, the Case Bank saturates quickly. Each additional iteration, therefore, contains progressively fewer previously unseen (and thus potentially failing) cases. In our simulated, open-ended, but ultimately finite environment, we observe rapid convergence with only marginal gains after a few iterations. Consequently, adding many more iterations yields diminishing returns and contributes little to our understanding of memory-based continual learning.

5.5.4. Generalisation across Tasks

To assess out-of-distribution (OOD) generalisation, we follow the evaluation protocol of Zheng et al. (2025). Specifically, MusiQue (Trivedi et al., 2022), Bamboogle (Press et al., 2022), and PopQA (Mallen et al., 2022) are selected as OOD datasets due to their distinct question styles and information distributions, while NQ (Kwiatkowski et al., 2019), TQ (Joshi et al., 2017), HotpotQA (Yang et al., 2018), and 2Wiki (Ho et al., 2020) are used for training. We first collect and store trajectories from the training datasets in the case bank. During inference, *Memento* retrieves the four most relevant cases from the case bank for each target query. As shown in Figure 1d, *Memento* achieves substantial improvements on all OOD benchmarks, with absolute gains ranging from 4.7% to 9.6%. These results highlight the effectiveness of case-based reasoning in enhancing generalisation to unseen tasks.

6. Discussion and Analysis

Building on the results in § 5 that establish the effectiveness of *Memento*, we further analyse its efficiency and operational behaviour. Specifically, we (i) analyse the average number of tool calls per task across three difficulty levels to assess how the MCP framework adapts as task complexity increases, (ii) characterise tool-call statistics, and (iii) evaluate the impact of using reasoning-oriented versus general-purpose models.



Şekil 5: Zorluk seviyesi arttıkça kod, arama ve tarama görevlerinin baskınlığını vurgulayarak, her görev türünün seviyeye göre ortalama sayısı.

Memento'nun sürekli öğrenme yeteneği. Daha da önemlisi, bellek tabanlı yaklaşımların parametre güncellemesi gerektirmeksizin LLM ajanlarını etkin biçimde geliştirebileceğini göstererek, DeepResearcher veri setinde iterasyon sayısının artmasıyla doğrulukta bir öğrenme eğrisi gözlemliyoruz.

Öğrenme eğrisi boyunca herhangi bir performans düşüşü tespit etmeye çalışmamıza rağmen, pratikte bu tür dönüm noktaları zor bulunmaktadır. Yaklaşık 3 bin eğitim verisi ile Vaka Bankası hızla doygunluğa ulaşmaktadır. Bu nedenle, her ek iterasyon giderek daha az önce görülmemiş (dolayısıyla potansiyel olarak başarısız) vaka içermektedir. Simüle edilmiş, açık uçlu ancak nihayetinde sonlu ortamımızda, birkaç iterasyondan sonra ancak marjinal kazançlarla hızlı bir yakınsama gözlemliyoruz. Sonuç olarak, daha fazla iterasyon eklemek azalan getiriler sağlamakta ve bellek tabanlı sürekli öğrenme konusundaki anlayışımıza çok az katkıda bulunmaktadır.

5.5.4. Görevler Arası Genelleme

Dağılım dışı (OOD) genelleştirmeyi değerlendirmek amacıyla, Zheng ve ark. (2025) tarafından önerilen değerlendirme protokolü izlenmiştir. Özellikle, MusiQue (Trivedi ve ark., 2022), Bamboogle (Press ve ark., 2022) ve PopQA (Mallen ve ark., 2022), soruların farklı üslup ve bilgi dağılımlarına sahip olmaları nedeniyle OOD veri setleri olarak seçilirken; NQ (Kwiatkowski ve ark., 2019), TQ (Joshi ve ark., 2017), HotpotQA (Yang ve ark., 2018) ve 2Wiki (Ho ve ark., 2020) eğitim amacıyla kullanılmıştır. Öncelikle, eğitim veri setlerinden alınan izler vaka bankasında toplanıp depolanmaktadır. Çıkarım aşamasında, *Memento* her hedef sorgu için vaka bankasından en alakalı dört vakaya erişim sağlamaktadır. Şekil 1d'de gösterildiği üzere, *Memento* %4,7 ile %9,6 arasında değişen mutlak kazanımlarla tüm OOD kıyaslamalarında kayda değer gelişmeler elde etmektedir. Bu bulgular, vaka tabanlı akıl yürütmenin görülmemiş görevlere genellemede etkinliğini ortaya koymaktadır.

6. Tartışma ve Analiz

§ 5'te Memento'nun etkinliğini ortaya koyan sonuçlar temelinde, yöntemimizin verimliliği ve operasyonel davranışı daha ayrıntılı şekilde analiz edilmektedir. Özellikle, (i) MCP çerçevesinin görev karmaşıklığı arttıkça uyum sağlamasını değerlendirmek amacıyla üç zorluk seviyesinde görev başına ortalama araç çağrısı sayısını analiz ediyoruz, (ii) araç çağrısı istatistiklerini karakterize ediyoruz ve (iii) muhakeme odaklı modellerin genel amaçlı modellere kıyasla kullanımının etkisini inceliyoruz.

Planner	Executor	Level 1	Level 2	Level 3	Average
gpt-4.1	o3	77.36%	69.77%	61.54%	70.91%
o3	o3	73.58%	63.95%	38.46%	63.03%
Qwen3-32B-Fast	o4-mini	62.26%	56.98%	26.92%	53.94%
Qwen3-32B-Slow	o4-mini	56.60%	36.05%	23.08%	40.61%

Table 6: The impact of fast and slow think mode on GAIA validation dataset (pass@1).

6.1. The Number of Tokens per Task

As shown in Figure 5, code, search, and crawl tasks dominate across all levels, with their usage increasing notably as difficulty rises. Importantly, while overall tool usage grows with task complexity, the most challenging problems increasingly rely on the model's internal reasoning to interpret and aggregate evidence from prior tool outputs, rather than simply calling more tools via MCP. This highlights the importance of effective integration between planning, memory, and evidence aggregation for solving open-ended, long-horizon deep research tasks.

6.2. Statistics of MCP Tools

As shown in Figure 6, we randomly sample 10 tasks from the GAIA validation, respectively, to calculate the tokens and costs per task. Average response tokens rise sharply with task difficulty: Level 1 queries required 26k input tokens and 4.7k output tokens, Level 2 grew to 48k/6.9k, and Level 3 peaked at 121k/9.8k. This highlights that the primary computational burden in complex scenarios stems from integrating and analysing multi-step tool outputs, rather than from generating long responses.

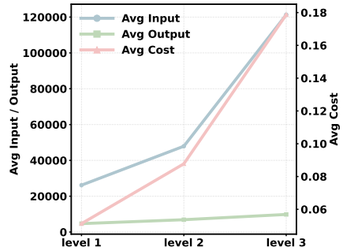


Figure 6: Token costs on the GAIA.

We observe that the output tokens remain stable across task levels, as the final answers typically require only short responses. This demonstrates that our system effectively controls generation length and avoids unnecessary verbosity during inference. However, due to the complexity and unpredictability of real-world environments, the input context grows significantly with task difficulty. As tasks become more complex, more detailed observations, plans, tool outputs, and intermediate reasoning steps must be incorporated into the input prompts, resulting in a substantial increase in the number of input tokens.

6.3. The Impact of Fast and Slow Think Mode

Table 6 compares the impact of fast- and slow-thinking planners on overall system performance (pass@1) across different task difficulties. The results show that pairing the fast, non-deliberative GPT-4.1 planner with the o3 executor yields the highest average accuracy (70.9%), outperforming the more deliberative o3 planner (63.03%) even when both use the same executor. Similarly, when using the o4-mini executor, GPT-4.1 achieves a substantial 16.4% improvement over o3. The Qwen3-32B models further confirm this trend, with the fast planner consistently outperforming its slow counterpart.

Planlayıcı	Yürütücü	Seviye 1	Seviye 2	Seviye 3	Ortalama
gpt-4.1	o3	77.36%	69.77%	61.54%	70.91%
o3	o3	73,58%	63,95%	38,46%	63,03%
Qwen 3-32B-Fast	o4-mini	62,26%	56,98%	26,92%	53,94%
Qwen3-32B-Slow	o4-mini	56,60%	36,05%	23,08%	40,61%

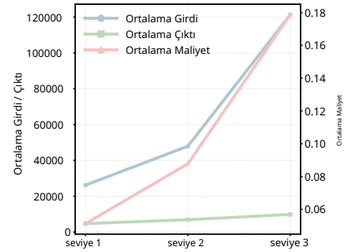
Tablo **6:** Hızlı ve yavaş düşünme modlarının GAIA doğrulama veri seti üzerindeki etkisi (pass@1).

6.1. Görev Başına Token Sayısı

Şekil 5'te gösterildiği gibi, kodlama, arama ve tarama görevleri tüm seviyelerde baskın olup, zorluk arttıkça kullanımları belirgin şekilde artmaktadır. Önemle belirtmek gerekir ki, genel araç kullanımı görev karmaşıklığıyla artarken, en zorlu problemler giderek MCP aracılığıyla daha fazla araç çağırmaktan z iyade, modelin dahili akıl yürütmesini kullanarak önceki araç çıktılarından delil yorumlama ve toplamada yoğunlaşmaktadır. Bu durum, planlama, bellek ve delil toplama arasındaki etkin entegrasyonun açık uçlu, uzun vadeli derin araştırma görevlerinin çözümünde önemini vurgulamaktadır.

6.2. MCP Araçlarının İstatistikleri

Şekil 6'da gösterildiği gibi, GAIA doğrulamasından rastgele seçilen 10 görev kullanılarak görev başına token sayıları ve maliyetler hesaplanmıştır. Ortalama yanıt token sayısı görev zorluğuyla keskin bir şekilde artmaktadır: Seviye 1 sorguları için 26 k girdi tokenı ve 4,7 k çıktı tokenı gerekmektedir; Seviye 2 ise 48 k / 6,9 k 'ye, Seviye 3 ise 121 k / 9,8 k 'ye yükselmiştir. Bu, karmaşık senaryolarda birincil hesaplama yükünün uzun yanıtlar üretmekten ziyade çok adımlı araç çıktılarının entegrasyonu ve analizinden kaynaklandığını vurgulamaktadır.



Şekil **6:** GAIA üzerindeki token maliyetleri.

Çıktı tokenlerinin görev seviyeleri boyunca sabit kaldığını gözlemliyoruz, çünkü nihai cevaplar genellikle yalnızca kısa yanıtlar gerektirmektedir. Bu, sistemimizin üretim uzunluğunu etkili bir şekilde kontrol ettiğini ve çıkarım sırasında gereksiz doluluktan kaçındığını göstermektedir. Ancak, gerçek dünya ortamlarının karmaşıklığı ve öngörülemezliği nedeniyle, giriş bağlamı görev zorluğu arttıkça önemli ölçüde büyümektedir. Görevler daha karmaşıklaştıkça, giriş istemlerine daha ayrıntılı gözlemler, planlar, araç çıktıları ve ara muhakeme adımları eklenmek zorunda kalınmakta ve bu da giriş token sayısında önemli artışa yol açmaktadır.

6.3. Hızlı ve Yavaş Düşünme Modunun Etkisi

Tablo 6, farklı görev zorlukları için hızlı ve yavaş düşünen planlayıcıların genel sistem performansı (pass@1) üzerindeki etkisini karşılaştırmaktadır. Sonuçlar, hızlı ve düşüncesiz GPT-4.1 planlayıcısının o3 yürütücü ile eşleştirildiğinde en yüksek ortalama doğruluğu (%70,9) sağladığını ve aynı yürütücüyü kullansalar bile daha düşünceli o3 planlayıcısını (%63,03) geride bıraktığını göstermektedir. Benzer şekilde, o4-mini yürütücüsü kullanıldığında GPT-4.1, o3 modeline kıyasla %16,4 oranında kayda değer bir iyileşme sağlamaktadır. Qwen3-32B modelleri bu eğilimi daha da doğrulamaktadır; hızlı planlayıcı, yavaş planlayıcının üzerinde istikrarlı bir şekilde performans göstermektedir.

Analysis of system traces reveals several key reasons for the fast planner's superiority. The planner relying on the o3 model often either answers directly – skipping plan generation altogether – or produces overly verbose plans, which can mislead the executor with incomplete instructions. Additionally, in complex multi-step reasoning fields, the slow planner tends to compress solutions into a single, convoluted chain of thought, while the fast planner effectively decomposes problems into manageable sub-tasks.

Overall, these findings highlight that in modular LLM systems, concise and structured planning leads to more effective downstream execution. Overly deliberative planning not only introduces unnecessary context and redundancy but also induces role confusion, thereby undermining the very specialisation that the two-stage architecture is designed to exploit.

7. Conclusion

We introduce *Memento*, a memory-based learning paradigm that enables LLM agents to adapt online search without updating model weights. *Memento* formalises deep research agents as a memory-based Markov Decision Process (MDP) and implements it within a planner–executor framework, leveraging an episodic case bank to record and retrieve trajectories for continual policy improvement. Empirically, we achieve strong performance across GAIA, DeepResearcher, and SimpleQA. Ablation studies reveal that both parametric and non-parametric CBR are critical to the significant performance gains, and that a small, curated memory yields optimal results. These findings motivate future work on deep research tasks using memory-based MDP.

References

Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7(1):39–59, 1994.

Rishabh Agarwal, Avi Singh, Lei Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, et al. Many-shot in-context learning. *Advances in Neural Information Processing Systems*, 37:76930–76966, 2024.

Alibaba. Aworld: A unified agent playground for computer and phone use tasks, 2025. URL https://github.com/inclusionAI/AWorld.

John R Anderson. *The architecture of cognition*. Psychology Press, 2013.

John R Anderson, Michael Matessa, and Christian Lebiere. Act-r: A theory of higher level cognition and its relation to visual attention. *Human–Computer Interaction*, 12(4):439–462, 1997.

Kevin D Ashley. Case-based reasoning and its implications for legal expert systems. *Artificial Intelligence and Law*, 1(2):113–208, 1992.

Alan David Baddeley. Working memory. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, 302(1110):311–324, 1983.

ByteDance. Deerflow: Deep exploration and efficient research framework. https://deerflow.tech/z, 2025.

Sistem izlerinin analizi, hızlı planlayıcının üstünlüğünün birkaç temel nedenini ortaya koymaktadır. o3 modeline dayanan planlayıcı genellikle ya doğrudan yanıt vererek plan oluşturmayı tamamen atlamakta ya da yürütücüyü eksik talimatlarla yanıltabilecek aşırı uzun planlar üretmektedir. Ayrıca, karmaşık çok adımlı akıl yürütme alanlarında yavaş planlayıcı çözümleri tek bir karmaşık düşünce zincirine sıkıştırma eğilimindeyken, hızlı planlayıcı problemleri yönetilebilir alt görevlere etkin bir şekilde ayrıştırmaktadır.

Genel olarak, bu bulgular modüler LLM sistemlerinde özlü ve yapılandırılmış planlamanın daha etkili bir sonraki aşama yürütmesine yol açtığını vurgulamaktadır. Aşırı dikkatli planlama yalnızca gereksiz bağlam ve tekrarı getirmekle kalmaz, aynı zamanda rol karışıklığını tetikleyerek iki aşamalı mimarinin sağladığı uzmanlığı zayıflatır.

7. Sonuç

Model ağırlıklarını güncellemeden LLM ajanlarının çevrimiçi aramaya uyum sağlamasına olanak tanıyan bellek tabanlı bir öğrenme paradigması olan Memento'yu tanıtıyoruz. Memento, derin araştırma ajanlarını hafıza tabanlı Markov Karar Süreci (MDH) olarak formüle eder ve sürekli politika iyileştirmesi için rotaları kaydetmek ve erişmek amacıyla episodik vaka bankasından yararlanarak planlayıcı–yürütücü çerçevesinde uygular. Ampirik olarak, GAIA, DeepResearcher ve SimpleQA üzerinde güçlü performans elde ettik. Ablasyon çalışmaları, hem parametrik hem de parametrik olmayan vaka tabanlı akıl yürütmenin önemli performans artışları için kritik olduğunu ve küçük, özenle seçilmiş bir belleğin optimal sonuçlar verdiğini göstermektedir. Bu bulgular, hafıza tabanlı MDH kullanılarak derin araştırma görevlerinde gelecekteki çalışmaları teşvik etmektedir.

Referanslar

Agnar Aamodt ve Enric Plaza. Vaka tabanlı akıl yürütme: Temel meseleler, metodolojik çeşitlilikler ve sistem yaklaşımları. *AI communications*, 7(1):39–59, 1994.

Rishabh Agarwal, Avi Singh, Lei Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova ve diğerleri. Çoklu örnekle bağlam içi öğrenme. *Neural Information Processing Systems'te gelişmeler* , 37:76930–76966, 2024.

Alibaba. Aworld: Bilgisayar ve telefon kullanım görevleri için birleşik bir ajan oyun alanı, 2025. URL https://github.com/inclusionAI/AWorld .

John R Anderson. *Bilişin Mimarisi*. Psychology Press, 2013.

John R Anderson, Michael Matessa ve Christian Lebiere. Act-r: Yüksek düzey biliş teorisi ve görsel dikkat ile ilişkisi. *İnsan–Bilgisayar Etkileşimi* , 12(4):439–462, 1997.

Kevin D Ashley. Vaka tabanlı akıl yürütme ve yasal uzman sistemlere etkileri. *Yapay Zeka ve Hukuk* , 1(2):113–208, 1992.

Alan David Baddeley. Çalışma Belleği. *Londra Kraliyet Topluluğu Felsefi Makaleleri. B, Biyolojik Bilimler* , 302(1110):311–324, 1983.

ByteDance. Deerflow: Derin keşif ve verimli araştırma çerçevesi. https://deerflow.tech/z , 2025.

Camel-AI. Owl: Optimized workforce learning for general multi-agent assistance in real-world task automation, 2025. URL https://github.com/camel-ai/owl.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

Gautam Choudhary, Natwar Modani, and Nitish Maurya. React: A review comment dataset for actionability (and more). In *Web Information Systems Engineering–WISE 2021: 22nd International Conference on Web Information Systems Engineering, WISE 2021, Melbourne, VIC, Australia, October 26–29, 2021, Proceedings, Part II 22*, pages 336–343. Springer, 2021.

Filippos Christianos, Georgios Papoudakis, Matthieu Zimmer, Thomas Coste, Zhihao Wu, Jingxuan Chen, Khyati Khandelwal, James Doran, Xidong Feng, Jiacheng Liu, et al. Pangu-agent: A fine-tunable generalist agent with structured reasoning. *arXiv preprint arXiv:2312.14878*, 2023.

Can Cui, Wei Wang, Meihui Zhang, Gang Chen, Zhaojing Luo, and Beng Chin Ooi. Alphaevolve: A learning framework to discover novel alphas in quantitative investment. In *Proceedings of the 2021 International conference on management of data*, pages 2208–2216, 2021.

Lutfi Eren Erdogan, Nicholas Lee, Sehoon Kim, Suhong Moon, Hiroki Furuta, Gopala Anumanchipalli, Kurt Keutzer, and Amir Gholami. Plan-and-act: Improving planning of agents for long-horizon tasks. *arXiv preprint arXiv:2503.09572*, 2025.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. Retool: Reinforcement learning for strategic tool use in llms. *arXiv preprint arXiv:2504.11536*, 2025.

Zafeirios Fountas, Martin A Benfeghoul, Adnan Oomerjee, Fenia Christopoulou, Gerasimos Lampouras, Haitham Bou-Ammar, and Jun Wang. Human-like episodic memory for infinite context llms. *arXiv preprint arXiv:2407.09450*, 2024.

Anthony G Francis and Ashwin Ram. The utility problem in case-based reasoning. In *Case-Based Reasoning: Papers from the 1993 Workshop*, pages 160–161, 1993.

Yao Fu, Dong-Ki Kim, Jaekyeom Kim, Sungryull Sohn, Lajanugen Logeswaran, Kyunghoon Bae, and Honglak Lee. Autoguide: Automated generation and selection of state-aware guidelines for large language model agents. *CoRR*, 2024.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1), 2023.

Paul W Glimcher. Understanding dopamine and reinforcement learning: the dopamine reward prediction error hypothesis. *Proceedings of the National Academy of Sciences*, 108(supplement_3):15647–15654, 2011.

Google. Gemini deep research — your personal research assistant. https://gemini.google/overview/deep-research/?hl=en-GB, 2025.

Camel-AI. Owl: Gerçek dünya görev otomasyonunda genel çoklu ajan desteği için optimize edilmiş iş gücü öğrenimi, 2025. URL https://github.com/camel-ai/owl .

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh ve Deshraj Yadav. Mem0: Ölçeklenebilir uzun vadeli belleğe sahip üretime hazır yapay zeka ajanları geliştirme. *arXiv ön baskısı arXiv:2504.19413* , 2025.

Gautam Choudhary, Natwar Modani ve Nitish Maurya. React: Eyleme geçirilebilirlik (ve diğerleri) için inceleme yorum veri seti. In *Web Information Systems Engineering–WISE 2021: 22. Uluslararası Web Bilgi Sistemleri Mühendisliği Konferansı, WISE 2021, Melbourne, VIC, Avustralya, 26–29 Ekim 2021, Bil diriler, Bölüm II 22* , sayfa 336–343. Springer, 2021.

Filippos Christianos, Georgios Papoudakis, Matthieu Zimmer, Thomas Coste, Zhihao Wu, Jingxuan Chen, Khyati Khandelwal, James Doran, Xidong Feng, Jiacheng Liu ve diğerleri. Pangu-agent: Yapılandırılmış akıl yürütme yeteneğine sahip ince ayarlanabilir genel amaçlı ajan. *arXiv ön baskısı arXiv:2312.14878* , 2023.

Can Cui, Wei Wang, Meihui Zhang, Gang Chen, Zhaojing Luo ve Beng Chin Ooi. Alphaevolve: Nicel yatırımda yeni alfaları keşfetmek için öğrenme çerçevesi. In *Proceedings of the 2021 International con ference on management of data* , sayfalar 2208–2216, 2021.

Lutfi Eren Erdogan, Nicholas Lee, Sehoon Kim, Suhong Moon, Hiroki Furuta, Gopala Anumanchipalli, Kurt Keutzer ve Amir Gholami. Plan-and-act: Uzun vadeli görevler için ajanların planlamasını geliştirme. *arXiv ön baskısı arXiv:2503.09572* , 2025.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi ve Wanjun Zhong. Retool: LLM'lerde stratejik araç kullanımı için pekiştirmeli öğrenme. *arXiv ön baskısı arXiv:2504.11536* , 2025.

Zafeirios Fountas, Martin A. Benfeghoul, Adnan Oomerjee, Fenia Christopoulou, Gerasimos Lampouras, Haitham Bou-Ammar ve Jun Wang. Sonsuz bağlam LLM'leri için insan benzeri episodik bellek. *arXiv ön baskısı arXiv:2407.09450* , 2024.

Anthony G. Francis ve Ashwin Ram. Vaka tabanlı akıl yürütmede yararlılık problemi. In *Vaka Tabanlı Akıl Yürütme: 1993 Atölye Çalışması Makaleleri*, sayfa 160–161, 1993.

Yao Fu, Dong-Ki Kim, Jaekyeom Kim, Sungryull Sohn, Lajanugen Logeswaran, Kyunghoon Bae ve Honglak Lee. Autoguide: Durum farkındalığına sahip kılavuzların otomatik oluşturulması ve seçimi için büyük dil modeli ajanları. *CoRR* , 2024.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang ve Haofen Wang. Getir-ajınlanmış üretim için büyük dil modelleri: Bir derleme. *arXiv ön baskısı arXiv:2312.10997* , 2(1), 2023.

Paul W. Glimcher. Dopamin ve pekiştirmeli öğrenmenin anlaşılması: Dopamin ödül tahmin hatası hipotezi. *Proceedings of the National Academy of Sciences* , 108(supplement_3):15647–15654, 2011.

Google. Gemini derin araştırma — kişisel araştırma asistanınız. https://gemini.google/overview/deep-research/?hl=en-GB , 2025.

Antoine Grosnit, Alexandre Maraval, James Doran, Giuseppe Paolo, Albert Thomas, Refinath Shahul Hameed Nabeezath Beevi, Jonas Gonzalez, Khyati Khandelwal, Ignacio Iacobacci, Abdelhakim Benechehab, et al. Large language models orchestrating structured reasoning achieve kaggle grandmaster level. *arXiv preprint arXiv:2411.03562*, 2024.

Siyuan Guo, Cheng Deng, Ying Wen, Hechang Chen, Yi Chang, and Jun Wang. Ds-agent: Automated data science by empowering large language models with case-based reasoning. In *International Conference on Machine Learning (ICML)*, pages 16813–16848. PMLR, 2024.

Siyuan Guo, Huiwu Liu, Xiaolong Chen, Yuming Xie, Liang Zhang, Tao Han, Hechang Chen, Yi Chang, and Jun Wang. Optimizing case-based reasoning system for functional test script generation with large language models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4487–4498, 2025.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pages 1352–1361. PMLR, 2017.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Kostas Hatalis, Despina Christou, and Vyshnavi Kondapalli. Review of case-based reasoning for llm agents: theoretical foundations, architectural components, and cognitive integration. *arXiv preprint arXiv:2504.06943*, 2025.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*, 2020.

Yuxuan Huang, Yihang Chen, Haozheng Zhang, Kang Li, Meng Fang, Linyi Yang, Xiaoguang Li, Lifeng Shang, Songcen Xu, Jianye Hao, Shao Kun, and Jun Wang. Deep research agents: A systematic examination and roadmap. *arXiv preprint arXiv:2506.18096*, 2025.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

Savya Khosla, Zhen Zhu, and Yifei He. Survey on memory-augmented neural networks: Cognitive insights to ai applications. *arXiv preprint arXiv:2312.06141*, 2023.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

Antoine Grosnit, Alexandre Maraval, James Doran, Giuseppe Paolo, Albert Thomas, Refinath Shahul Hameed Nabeezath Beevi, Jonas Gonzalez, Khyati Khandelwal, Ignacio Iacobacci, Abdelhakim Benechehab, ve diğerleri. Büyük dil modellerini kullanarak yapılandırılmış akıl yürütmenin yönetimi kaggle grandmaster seviyesine ulaşmaktadır. *arXiv ön baskısı arXiv:2411.03562* , 2024.

Siyuan Guo, Cheng Deng, Ying Wen, Hechang Chen, Yi Chang ve Jun Wang. Ds-agent: Büyük dil modellerini vaka tabanlı akıl yürütmeyle güçlendirerek otomatik veri bilimi. Uluslararası Makine Öğrenmesi Konferansı *(ICML)* , ss. 16813–16848. PMLR, 2024.

Siyuan Guo, Huiwu Liu, Xiaolong Chen, Yuming Xie, Liang Zhang, Tao Han, Hechang Chen, Yi Chang ve Jun Wang. Büyük dil modelleri ile fonksiyonel test betiği oluşturumu için vaka tabanlı akıl yürütme sisteminin optimizasyonu. *31. ACM SIGKDD* Bilgi Keşfi ve Veri Madenciliği Konferansı Bildirileri , ss. 4487–4498, 2025.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel ve Sergey Levine. Derin enerji tabanlı politikalarla pekiştirmeli öğrenme. Uluslararası Makine Öğrenmesi Konferansı , ss. 1352–1361. PMLR, 2017.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel ve diğerleri. Yumuşak aktör-eleştirmen algoritmaları ve uygulamaları. *arXiv ön baskısı arXiv:1812.05905* , 2018.

Kostas Hatalis, Despina Christou ve Vyshnavi Kondapalli. LLM ajanları için vaka tabanlı akıl yürütmenin incelenmesi: teorik temeller, mimari bileşenler ve bilişsel entegrasyon. *arXiv ön baskısı arXiv:2504.06943* , 2025.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara ve Akiko Aizawa. Akıl yürütme adımlarının kapsamlı değerlendirilmesi için çok adımlı soru-cevap veri seti oluşturulması. *arXiv ön baskısı arXiv:2011.01060* , 2020.

Yuxuan Huang, Yihang Chen, Haozheng Zhang, Kang Li, Meng Fang, Linyi Yang, Xiaoguang Li, Lifeng Shang, Songcen Xu, Jianye Hao, Shao Kun ve Jun Wang. Derin araştırma ajanları: sistematik bir inceleme ve yol haritası. *arXiv ön baskısı arXiv:2506.18096* , 2025.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani ve Jiawei Han. Search-r1: Pekiştirmeli öğrenme ile arama motorlarını kullanma ve akıl yürütme becerisi kazandırmak üzere LLM'lerin eğitilmesi. *arXiv ön baskısı arXiv:2503.09516* , 2025.

Mandar Joshi, Eunsol Choi, Daniel S Weld ve Luke Zettlemoyer. Triviaqa: Okuduğunu anlama için büyük ölçekli uzaktan denetimli bir zorluk veri kümesi. *arXiv ön baskısı arXiv:1705.03551* , 2017.

Savya Khosla, Zhen Zhu ve Yifei He. Bellek destekli sinir ağları üzerine derleme: Bilişsel içgörülerden yapay zekâ uygulamalarına. *arXiv ön baskısı arXiv:2312.06141* , 2023.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee ve diğerleri. Natural Questions: soru yanıtlama araştırmaları için bir kıyaslama. *Transactions of the Association for Computational Linguistics* , 7:453–466, 2019.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel ve diğerleri. Bilgi yoğun NLP görevleri için retrieval-augmented generation. *Advances in Neural Information Processing Systems* , 33:9459–9474, 2020.

Chengpeng Li, Mingfeng Xue, Zhenru Zhang, Jiaxi Yang, Beichen Zhang, Xiang Wang, Bowen Yu, Binyuan Hui, Junyang Lin, and Dayiheng Liu. Start: Self-taught reasoner with tools. *arXiv preprint arXiv:2503.04625*, 2025a.

Hongyu Li, Liang Ding, Meng Fang, and Dacheng Tao. Revisiting catastrophic forgetting in large language model tuning. *arXiv preprint arXiv:2406.04836*, 2024.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, et al. Websailor: Navigating super-human reasoning for web agent. *arXiv preprint arXiv:2507.02592*, 2025b.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. Api-bank: A comprehensive benchmark for tool-augmented llms. *arXiv preprint arXiv:2304.08244*, 2023.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. *arXiv preprint arXiv:2501.05366*, 2025c.

Xinbin Liang, Jinyu Xiang, Zhaoyang Yu, Jiayi Zhang, Sirui Hong, Sheng Fan, and Xiao Tang. Openmanus: An open-source framework for building general ai agents, 2025. URL https://doi.org/10.5281/zenodo.15186407.

Xuechen Liang, Yangfan He, Yinghui Xia, Xinyuan Song, Jianhui Wang, Meiling Tao, Li Sun, Xinhang Yuan, Jiayi Su, Keqin Li, et al. Self-evolving agents with reflective and memory-augmented abilities. *arXiv preprint arXiv:2409.00872*, 2024.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Hannaneh Hajishirzi, and Daniel Khashabi. When not to trust language models: Investigating effectiveness and limitations of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511*, 7, 2022.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*, 2023.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

OpenAI. Deep research system card. Technical report, OpenAI, 2025. URL https://cdn.openai.com/deep-research-system-card.pdf.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*, 2022.

Chengpeng Li, Mingfeng Xue, Zhenru Zhang, Jiaxi Yang, Beichen Zhang, Xiang Wang, Bowen Yu, Binyuan Hui, Junyang Lin ve Dayiheng Liu. START: Araçlarla kendi kendine öğrenen akıl yürütücü. *arXiv ön baskısı arXiv:2503.04625* , 2025a.

Hongyu Li, Liang Ding, Meng Fang ve Dacheng Tao. Büyük dil modeli ince ayarında felaket unutmanın yeniden incelenmesi. *arXiv ön baskısı arXiv:2406.04836* , 2024.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang ve diğerleri. Websailor: Web ajanı için insanüstü akıl yürütmenin yönlendirilmesi. *arXiv ön baskısı arXiv:2507.02592* , 2025b.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang ve Yong-bin Li. Api-bank: Araç destekli LLM'ler için kapsamlı bir kıyaslama seti. *arXiv ön baskısı arXiv :2304.08244* , 2023.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang ve Zhicheng Dou. Search-o1: Ajan tabanlı arama destekli büyük akıl yürütme modelleri.*arXiv ön baskısı arXiv:2501.05366*, 2025c.

Xinbin Liang, Jinyu Xiang, Zhaoyang Yu, Jiayi Zhang, Sirui Hong, Sheng Fan ve Xiao Tang. Openmanus: Genel yapay zeka ajanları oluşturmak için açık kaynaklı bir çerçeve, 2025. URL https://doi.org/10.5281/zenodo.15186407 .

Xuechen Liang, Yangfan He, Yinghui Xia, Xinyuan Song, Jianhui Wang, Meiling Tao, Li Sun, Xinhang Yuan, Jiayi Su, Keqin Li ve diğerleri. Yansıtıcı ve bellek destekli yeteneklere sahip kendi kendini geliştiren ajanlar. *arXiv ön baskısı arXiv:2409.00872* , 2024.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee ve Min Lin. r1-zero benzeri eğitimi anlama: Eleştirel bir perspektif.*arXiv ön baskısı arXiv:2503.20783*, 2025.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Hannaneh Hajishirzi ve Daniel Khashabi. Dil modellerine ne zaman güvenilmez: Parametrik ve parametrik olmayan belleklerin etkinliği ve sınırlılıklarının incelenmesi. *arXiv ön baskısı arXiv:2212.10511* , 7, 2022.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun ve Thomas Scialom. Gaia: Genel yapay zeka asistanları için bir kıyaslama standardı. On İkinci Uluslararası Öğrenme Temsilleri Konferansı'nda , *2023*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, ve diğerleri. Webgpt: İnsan geri bildirimiyle tarayıcı destekli soru-cevaplama. *arXiv ön baskısı arXiv:2112.09332* , 2021.

OpenAI. Derin araştırma sistem kartı. Teknik rapor, OpenAI, 2025. URL https://cdn.openai.com/deep-research-system-card.pdf .

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, ve diğerleri. İnsanlığın son sınavı. *arXiv ön baskısı arXiv:2501.14249* , 2025.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith ve Mike Lewis. Dil modellerindeki bileşimlilik farkını ölçme ve azaltma. *arXiv ön baskısı arXiv:2210.03350* , 2022.

Alexander Pritzel, Benigno Uria, Sriram Srinivasan, Adria Puigdomenech Badia, Oriol Vinyals, Demis Hassabis, Daan Wierstra, and Charles Blundell. Neural episodic control. In *International conference on machine learning*, pages 2827–2836. PMLR, 2017.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958*, 2025.

Jiahao Qiu, Xuan Qi, Tongcheng Zhang, Xinzhe Juan, Jiacheng Guo, Yifu Lu, Yimin Wang, Zixin Yao, Qihan Ren, Xun Jiang, et al. Alita: Generalist agent enabling scalable agentic reasoning with minimal predefinition and maximal self-evolution. *arXiv preprint arXiv:2505.20286*, 2025.

Brian H Ross. Some psychological results on case-based reasoning. In *Proceedings: Case-based reasoning workshop*, pages 144–147. Morgan Kaufmann, 1989.

Rana Salama, Jason Cai, Michelle Yuan, Anna Currey, Monica Sunkara, Yi Zhang, and Yassine Benajiba. Meminsight: Autonomous memory augmentation for llm agents. *arXiv preprint arXiv:2503.21760*, 2025.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551, 2023.

Wenxuan Shi, Haochen Tan, Chuqiao Kuang, Xiaoguang Li, Xiaozhe Ren, Chen Zhang, Hanting Chen, Yasheng Wang, Lifeng Shang, Fisher Yu, et al. Pangu deepdiver: Adaptive search intensity scaling via open-web reinforcement learning. *arXiv preprint arXiv:2505.24332*, 2025.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. Ai models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759, 2024.

Barry Smyth and Paul McClave. Similarity vs. diversity. In *International conference on case-based reasoning*, pages 347–361. Springer, 2001.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*, 2025.

Larry R Squire, Lisa Genzel, John T Wixted, and Richard G Morris. Memory consolidation. *Cold Spring Harbor perspectives in biology*, 7(8):a021766, 2015.

Theodore Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas Griffiths. Cognitive architectures for language agents. *Transactions on Machine Learning Research*, 2023.

Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. Recitation-augmented language models. *arXiv preprint arXiv:2210.01296*, 2022.

Xiangru Tang, Tianrui Qin, Tianhao Peng, Ziyang Zhou, Daniel Shao, Tingting Du, Xinming Wei, Peng Xia, Fang Wu, He Zhu, et al. Agent kb: Leveraging cross-domain experience for agentic problem solving. *arXiv preprint arXiv:2507.06229*, 2025.

Alexander Pritzel, Benigno Uria, Sriram Srinivasan, Adria Puigdomenech Badia, Oriol Vinyals, Demis Hassabis, Daan Wierstra ve Charles Blundell. Sinirsel epizodik kontrol. In *International conference on machine learning* , sayfa 2827–2836. PMLR, 2017.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur ve Heng Ji. Toolrl: Ödül tüm araç öğrenimi için yeterlidir. *arXiv ön baskısı arXiv:2504.13958* , 2025.

Jiahao Qiu, Xuan Qi, Tongcheng Zhang, Xinzhe Juan, Jiacheng Guo, Yifu Lu, Yimin Wang, Zixin Yao, Qihan Ren, Xun Jiang ve diğerleri. Alita: Minimum ön tanımlama ve maksimum kendi kendine evrimle ölçeklenebilir ajanik akıl yürütmeyi mümkün kılan genel amaçlı ajan. *arXiv ön baskısı arXiv:2505.20286* , 2025.

Brian H. Ross. Vaka tabanlı akıl yürütme üzerine bazı psikolojik sonuçlar. In *Proceedings: Case-based reasoning workshop* , sayfa 144–147. Morgan Kaufmann, 1989.

Rana Salama, Jason Cai, Michelle Yuan, Anna Currey, Monica Sunkara, Yi Zhang ve Yassine Benajiba. Meminsight: LLM ajanları için otonom bellek artırımı. *arXiv ön baskısı arXiv:2503.21760*, 2025.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda ve Thomas Scialom. Toolformer: Dil modelleri araçları kullanmayı kendi kendine öğrenebilir. *Advances in Neural Information Processing Systems*, 36:68539–68551, 2023.

Wenxuan Shi, Haochen Tan, Chuqiao Kuang, Xiaoguang Li, Xiaozhe Ren, Chen Zhang, Hanting Chen, Yasheng Wang, Lifeng Shang, Fisher Yu ve diğerleri. Pangu deepdiver: Açık web takviyeli öğrenme yoluyla uyarlanabilir arama yoğunluğu ölçeklendirmesi. *arXiv ön baskısı arXiv:2505.24332* , 2025.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan ve Shunyu Yao. Reflexion: Sözlü takviyeli öğrenmeye sahip dil ajanları. *Advances in Neural Information Processing Systems* , 36:8634–8652, 2023.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson ve Yarin Gal. Yapay zeka modelleri, özyinelemeli oluşturulan verilerle eğitildiğinde çöker. *Nature* , 631(8022):755–759, 2024.

Barry Smyth ve Paul McClave. Benzerlik ve çeşitlilik. Uluslararası vaka tabanlı akıl yürütme konferansında, sayfa 347–361. Springer, 2001.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang ve Ji-Rong Wen. R1-searcher: Pekiştirmeli öğrenme ile LLM'lerde arama yeteneğini teşvik etmek. *arXiv ön baskısı arXiv:2503.05592* , 2025.

Larry R Squire, Lisa Genzel, John T Wixted ve Richard G Morris. Bellek pekiştirmesi. *Cold Spring Harbor Perspectives in Biology* , 7(8):a021766, 2015.

Theodore Sumers, Shunyu Yao, Karthik Narasimhan ve Thomas Griffiths. Dil ajanları için bilişsel mimariler. *Transactions on Machine Learning Research* , 2023.

Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang ve Denny Zhou. Okuduğu metinle desteklenen dil modelleri. *arXiv ön baskısı arXiv:2210.01296*, 2022.

Xiangru Tang, Tianrui Qin, Tianhao Peng, Ziyang Zhou, Daniel Shao, Tingting Du, Xinming Wei, Peng Xia, Fang Wu, He Zhu ve diğerleri. Agent kb: Ajan problem çözümünde çapraz alan deneyiminden yararlanma. *arXiv ön baskısı arXiv:2507.06229* , 2025.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10: 539–554, 2022.

Hongru Wang, Cheng Qian, Wanjun Zhong, Xiusi Chen, Jiahao Qiu, Shijue Huang, Bowen Jin, Mengdi Wang, Kam-Fai Wong, and Heng Ji. Otc: Optimal tool calls via reinforcement learning. *arXiv preprint arXiv:2504.14870*, 2025.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. *arXiv preprint arXiv:2409.07429*, 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*, 2024.

Yixuan Weng, Minjun Zhu, Guangsheng Bao, Hongbo Zhang, Jindong Wang, Yue Zhang, and Linyi Yang. Cycleresearcher: Improving automated research via automated review. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=bjcsVLoHYs.

Nirmalie Wiratunga, Ramitha Abeyratne, Lasal Jayawardena, Kyle Martin, Stewart Massie, Ikechukwu Nkisi-Orji, Ruvan Weerasinghe, Anne Liret, and Bruno Fleisch. Cbr-rag: case-based reasoning for retrieval augmented generation in llms for legal question answering. In *International Conference on Case-Based Reasoning*, pages 445–460. Springer, 2024.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

Wujiang Xu, Kai Mei, Hang Gao, Juntao Tan, Zujie Liang, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110*, 2025.

Xue Yan, Yan Song, Xidong Feng, Mengyue Yang, Haifeng Zhang, Haitham Bou Ammar, and Jun Wang. Efficient reinforcement learning with large language model priors. *The Thirteenth International Conference on Learning Representations(ICLR)*, 2025.

Yingxuan Yang, Mulei Ma, Yuxuan Huang, Huacan Chai, Chenyu Gong, Haoran Geng, Yuanjian Zhou, Ying Wen, Meng Fang, Muhao Chen, et al. Agentic web: Weaving the next web with ai agents. *arXiv preprint arXiv:2507.21206*, 2025.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot ve Ashish Sabharwal. Musique: Tek adımlı soru bileşimleriyle çok adımlı sorular. *Transactions of the Association for Computational Linguistics* , 10: 539–554, 2022.

Hongru Wang, Cheng Qian, Wanjun Zhong, Xiusi Chen, Jiahao Qiu, Shijue Huang, Bowen Jin, Mengdi Wang, Kam-Fai Wong ve Heng Ji. Otc: Pekiştirmeli öğrenme yoluyla optimal araç çağrıları. *arXiv ön baskısı arXiv:2504.14870* , 2025.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried ve Graham Neubig. Ajan iş akışı belleği. *arXiv ön baskısı arXiv:2409.07429* , 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou ve diğerleri. Zincirleme düşünce yönlendirmesi büyük dil modellerinde akıl yürütmeyi tetikler. *Sinirsel bilgi işleme sistemlerindeki gelişmeler* , 35:24824–24837, 2022.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman ve William Fedus. Büyük dil modellerinde kısa biçimli doğruluk ölçümü. *arXiv ön baskısı arXiv:2411.04368* , 2024.

Yixuan Weng, Minjun Zhu, Guangsheng Bao, Hongbo Zhang, Jindong Wang, Yue Zhang ve Linyi Yang. Cycleresearcher: Otomatik inceleme yoluyla otomatik araştırmayı geliştirmek. The Thirteenth International Conference on Learning Representations *, 2025*. *URL* https://openreview.net/forum?id=bjcsVLoHYs .

Nirmalie Wiratunga, Ramitha Abeyratne, Lasal Jayawardena, Kyle Martin, Stewart Massie, Ikechukwu Nkisi-Orji, Ruvan Weerasinghe, Anne Liret ve Bruno Fleisch. Cbr-rag: Yasal soru cevaplamada LLM'ler için retrieval augmented generation'da vaka tabanlı akıl yürütme. International *Conference* on *Case* -Based Reasoning , ss. 445–460. Springer, 2024.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu ve diğerleri. Autogen: Çoklu ajanlı iletişim yoluyla yeni nesil LLM uygulamalarını mümkün kılmak. *arXiv ön baskısı arXiv:2308.08155* , 2023.

Wujiang Xu, Kai Mei, Hang Gao, Juntao Tan, Zujie Liang ve Yongfeng Zhang. A-mem: LLM ajanları için ajan bellek. *arXiv ön baskısı arXiv:2502.12110* , 2025.

Xue Yan, Yan Song, Xidong Feng, Mengyue Yang, Haifeng Zhang, Haitham Bou Ammar ve Jun Wang. Büyük dil modeli öncelikleriyle verimli pekiştirmeli öğrenme. *On Üçüncü Uluslararası Öğrenme Temsilleri Konferansı (ICLR)* , 2025.

Yingxuan Yang, Mulei Ma, Yuxuan Huang, Huacan Chai, Chenyu Gong, Haoran Geng, Yuanjian Zhou, Ying Wen, Meng Fang, Muhao Chen ve diğerleri. Ajan bazlı web: Yapay zeka ajanlarıyla gelecek web'in dokuması. *arXiv ön baskısı arXiv:2507.21206* , 2025.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov ve Christopher D. Manning. HotpotQA: Çeşitli, açıklanabilir çok adımlı soru-cevaplama için bir veri seti. *arXiv ön baskısı arXiv:1809.09600* , 2018.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan ve Yuan Cao. REACT: Dil modellerinde akıl yürütme ve eylemi birlikte senkronize etmek. Uluslararası Öğrenme Temsilleri Konferansı *(ICLR)* , *2023*.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. Generate rather than retrieve: Large language models are strong context generators. *arXiv preprint arXiv:2209.10063*, 2022.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19632–19642, 2024.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. *arXiv preprint arXiv:2504.03160*, 2025.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19724–19731, 2024.

Huichi Zhou, Kin-Hei Lee, Zhonghao Zhan, Yue Chen, and Zhenhao Li. Trustrag: Enhancing robustness and trustworthiness in rag. *arXiv preprint arXiv:2501.00879*, 2025.

Minjun Zhu, Yixuan Weng, Linyi Yang, and Yue Zhang. Deepreview: Improving llm-based paper review with human-like deep thinking process. *arXiv preprint arXiv:2503.08569*, 2025a.

Minjun Zhu, Qiujie Xie, Yixuan Weng, Jian Wu, Zhen Lin, Linyi Yang, and Yue Zhang. Ai scientists fail without strong implementation capability. *arXiv preprint arXiv:2506.01372*, 2025b.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng ve Meng Jiang. Erişim sağlamak yerine üretmek: Büyük dil modelleri güçlü bağlam oluşturuculardır. *arXiv ön baskısı arXiv:2209.10063*, 2022.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu ve Gao Huang. Expel: LLM ajanları deneyimsel öğrenicilerdir. *AAAI* Yapay Zeka Konferansı Bildirileri'nde , cilt 38, sayfalar 19632–19642, 2024.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu ve Pengfei Liu. Deepresearcher: Gerçek dünya ortamlarında pekiştirmeli öğrenme yoluyla derin araştırmanın ölçeklendirilmesi. *arXiv ön baskısı arXiv:2504.03160* , 2025.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye ve Yanlin Wang. Memorybank: Büyük dil modellerini uzun vadeli bellekle geliştirmek. In *Proceedings of the AAAI Conference on Artificial Intelligence* , cilt 38, sayfalar 19724–19731, 2024.

Huichi Zhou, Kin-Hei Lee, Zhonghao Zhan, Yue Chen ve Zhenhao Li. Trustrag: rag'da dayanıklılık ve güvenilirlik artırımı. *arXiv ön baskısı arXiv:2501.00879* , 2025.

Minjun Zhu, Yixuan Weng, Linyi Yang ve Yue Zhang. Deepreview: İnsan benzeri derin düşünme süreciyle llm tabanlı makale değerlendirmesinin iyileştirilmesi. *arXiv ön baskısı arXiv:2503.08569* , 2025a.

Minjun Zhu, Qiujie Xie, Yixuan Weng, Jian Wu, Zhen Lin, Linyi Yang ve Yue Zhang. Yapay zeka bilim insanları güçlü uygulama yeteneği olmadan başarısız olur. *arXiv ön baskısı arXiv:2506.01372* , 2025b.

A. Derivation of the Optimal Policy in Soft-Q Learning

The soft value function over the state s is defined as:

$$V^{\pi}(s,M) = \sum_{c\in M} \mu(c|s,M)[Q^{\pi}(s,M,c) - \alpha \log \mu(c|s,M)] \tag{17}$$

The Q function over the state, case pair is defined as:

$$Q^{\pi}(s,M,c) = \mathbb{E}_{a\sim p_{\text{LLM}}(\cdot|s,c),s'\sim\mathcal{P}(\cdot|s,a)}\left[r(s,a) + \gamma V^{\pi}(s',M')\right], \tag{18}$$

Define the visitation frequency over the state, case bank for the policy π as: $d^{\pi}(s,M) = \sum_{t=0}^{\infty}\gamma^{t-1}\mathbb{P}(s_t = s, M_t = M)$. Then, our goal is to derive the optimal retrieval policy by expected value function:

$$\begin{aligned} J_{\text{MaxEnt}}(\pi) &= \mathbb{E}_{(s,M)\sim d^{\pi}}\left[V^{\pi}(s,M)\right], \\ &= \mathbb{E}_{(s,M)\sim d^{\pi}}\left[\sum_{c\in M}\mu(c|s,M)\left[Q^{\pi}(s,M,c) - \alpha\log\mu(c|s,M)\right]\right] \end{aligned} \tag{19}$$

For simplicity, let $\mu_c = \mu(c|s,M)$ and $Q_c = Q^{\pi}(s,M,c)$, and introduce the Lagrange multiplier λ to constrain that $\sum_c \mu_c = 1$. Then, for any state, case bank pair, we have the optimisation objective:

$$\mathcal{J}(\{\mu_c\},\lambda) = \sum_c \mu_c Q_c - \alpha\sum_c \mu_c \log\mu_c - \lambda(\sum_c \mu_c - 1), \tag{20}$$

whose derivative concerning μ_c is:

$$\frac{\partial\mathcal{J}(\{\mu_c\},\lambda)}{\partial\mu_c} = Q_c - \alpha(1+\log\mu_c) - \lambda, \tag{21}$$

Let $\frac{\partial\mathcal{J}(\{\mu_c\},\lambda)}{\partial\mu_c} = 0$, then we have:

$$\begin{aligned} \mu_c &= \exp(\frac{Q_c}{\alpha} - (\frac{\lambda}{\alpha}+1)) \\ &= K\exp(\frac{Q_c}{\alpha}), \end{aligned} \tag{22}$$

where $K = \exp(-\frac{\lambda}{\alpha} - 1)$. Thus, by performing the normalisation and shaping the optimal Q, we have:

$$\mu_c^* = \frac{\exp(Q_c^*/\alpha)}{\sum_{c'}\exp(Q_{c'}^*/\alpha)}. \tag{23}$$

Note that when $\alpha \to 0$, the soft Q learning deteriorates to standard Q-learning. The softmax form of the policy is also used in previous LLM-based agents with LLM prior (Yan et al. (2025)).

A. Soft-Q Öğrenmede Optimal Politikanın Türetilmesi

Durum $s$üzerindeki yumuşak değer fonksiyonu şu şekilde tanımlanır:

$$V^{\pi}(s,M) = \sum_{c\in M} \mu(c|s,M)[\, Q^{\pi}(s,M,c) - \alpha \log \mu(c|s,M)] \tag{17}$$

Durum ve vaka çifti üzerindeki Q-fonksiyonu şu şekilde tanımlanır:

$$Q^{\pi}(s,M,c) = \mathbb{E}_{a\sim p_{\text{LLM}}(\cdot|s,c),s'\sim\mathcal{P}(\cdot|s,a)}\left[r(s,a) + \gamma V^{\pi}(s',M')\right], \tag{18}$$

Politika π ile durum ve vaka bankası üzerindeki ziyaret sıklığı şöyle tanımlanır: $d^{\pi}(s, M) = \sum_{t=0}^{\infty}\gamma^{t-1}$}P (s_t = s, M_t = M). Daha sonra, beklenen değer fonksiyonuna göre optimal getirme politikası türetilir:

$$\begin{aligned} J_{\text{MaxEnt}}(\pi) &= \mathbb{E}_{(s,M)\sim d^{\pi}}\left[V^{\pi}(s,M)\right], \\ &= \mathbb{E}_{(s,M)\sim d^{\pi}}\left[\sum_{c\in M}\mu(c|s,M)\,[\,Q^{\pi}(s,M,c) - \alpha\log\mu(c|s,M)\right]\right] \end{aligned} \tag{19}$$

Basitlik adına, şunu kabul edelim $\mu_c = \mu(c|s,M)$ ve $Q_c = Q^{\pi}(s,M,c)$ olarak; ayrıca $\Sigma_c \mu_c = 1$ koşulunu sağlamak üzere Lagrange çarpanı λ kullanalım. Böylece, herhangi bir durum ve vaka bankası çifti için optimizasyon amacı şöyledir:

$$\mathcal{J}(\{\mu_c\},\lambda) = \sum_c \mu_c Q_c - \alpha\sum_c \mu_c \log\mu_c - \lambda(\sum_c \mu_c - 1), \tag{20}$$

bu ifadeye göre μ_c üzerindeki türev aşağıdaki gibidir:

$$\frac{\partial\mathcal{J}(\{\mu_c\},\lambda)}{\partial\mu_c} = Q_c - \alpha(1+\log\mu_c) - \lambda, \tag{21}$$

Diyelim ki $\frac{\partial\mathcal{J}(\{\mu_c\},\lambda)}{\partial\mu_c} = 0$ olduğunda, şu sonucu elde ederiz:

$$\begin{aligned} \mu_c &= \exp(\frac{Q_c}{\alpha} - (\frac{\lambda}{\alpha}+1)) \\ &= K\exp(\frac{Q_c}{\alpha}), \end{aligned} \tag{22}$$

where $K = \exp(-\frac{\lambda}{\alpha} - 1)$. Böylelikle normalizasyon yaparak ve optimal Q'yu şekillendirerek aşağıdaki ifadeye ulaşırız:

$$\mu_c^* = \frac{\exp(Q_c^*/\alpha)}{\sum_{c'}\exp(Q_{c'}^*/\alpha)}. \tag{23}$$

Burada, $\alpha \to 0$ durumunda yumuşak Q öğrenmesi klasik Q öğrenmesine dönüşmektedir. Yumuşakmaksimum (soft max) politika formu, daha önce LLM öncelikli ajanlarda da kullanılmıştır (Yan et al. (2025)).

B. Analysis of Memory Mechanisms

Method	Kernel	Neural Q	Q-Function	Read	Write	Gradient
Tabular Q-learning	w/o	w/o	Q-Table	Exact Match	Eq. (8)	-
Deep Q-learning	w/o	w/	Neural Network	Eq. (7)	Eq. (24)	Eq. (25)
Neural Episodic Control	w/	w/o	Eq. (9)	Eq. (7)	Eq. (10)	Eq. (11)
Non-Parametric Memory in Sec. 4	w/o	w/o	w/o	Eq. (13)	Eq. (12)	-
Parametric Memory in Sec. 4	w/o	w/	Neural Network	Eq. (16)	Eq. (15)	Eq. (26)

Table 7: Detail comparison of memory mechanisms.

Here, we consider several representative memory mechanisms, emphasising their Read and Write operations as summarised in Table 7. Specifically, we discuss tabular and parametric Q-value representations, as well as EC-based methods.

In the tabular setting, the memory maintains an explicit table $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, where Read is a direct lookup of table $Q(s, M, a)$ and Write corresponds to updating the entry for the state–action pair after observing a transition, following the standard TD learning in Eq. (8). To extend beyond discrete spaces, deep Q-learning learns the Q function by a neural network $Q(s, M, a; \theta)$, with Read operation sampling cases from the retrieval policy μ following Eq. (7) and Write operation updates the parameters θ via minimising the TD error:

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,c,r,s',M,M')}\left[\left(Q(s, M, c; \theta) - \left[r + \gamma\alpha \log \sum_{c' \in M'} \exp\left(Q\left(s', M', c'; \bar{\theta}\right)\right)\right]\right)^2\right], \tag{24}$$

where $\bar{\theta}$ is the target Q network. The gradient of the TD learning loss with respect to θ is given by:

$$\nabla_\theta \mathcal{L}(\theta) = 2\mathbb{E}_{(s,c,r,s',M,M')}\left[\left(Q(s, M, c; \theta) - y\right) \nabla_\theta Q(s, M, c; \theta)\right], \tag{25}$$

where $y = r + \gamma\alpha \log \sum_{c' \in M'} \exp\big(Q(s', M', c'; \bar{\theta})\big)$. This parametric formulation enables generalisation across states and action spaces through the shared parameters θ, in contrast to tabular methods, which only memorise individual entries. However, this benefit comes at the cost of optimisation instability and large data demand, since approximation errors may propagate globally through the parameter space. This limitation motivates the EC-based methods in Section 3, where value estimation is regularised through a learnable kernel (see Eq. (9)). Within this memory design, the Read operation samples cases from the retrieval policy distribution defined in Eq. (7) while the Write operation additionally store (s, c, Q) into an episodic memory and updates the kernel parameters θ by Eq. (10) with gradient in Eq. (11) optimising the weighting function. This approach only parameterises the kernel to regularise the historical Q-values of matched states, thereby ensuring generalisation across the state space while retaining data-efficient adaptation and improved stability compared with deep Q-learning.

In section 4, the CBR agent is implemented as a planner, applying both non-parametric and parametric memory mechanisms. For the non-parametric variant, the Write operation appends each observed case (s_t, a_t, r_t) into the case bank as in Eq.(12), while the Read operation retrieves the most relevant experiences by performing cosine-similarity matching between the current query embedding and stored states, followed by a TopK selection as in Eq. (13). This similarity-based retrieval, without further parameterisation,

B. Bellek Mekanizmalarına İlişkin Analiz

Yöntem	Kernel	Neural Q	Q-fonksiyonu	Okuma	Yazma	Gradyan
Tablolu Q-öğrenme	yapmadan	yapmadan	Q-tablosu	Tam Eşleşme	Eq. (8)	-
Derin Q-öğrenme	yapmadan	ile Sinir Ağı	Eq. (7)	Eq. (24)	Eq. (25)	
Sinirsel Epizodik Kontrol	ile	yapmadan	Eq. (9)	Eq. (7)	Eq. (10)	Eq. (11)
Parametrik Olmayan Bellek Bölüm 4'te	yapmadan	yapmadan	yapmadan	Eq. (13)	Eq. (12)	-
Parametrik Bellek Bölüm 4'te	yapmadan	w/	Sinir Ağı	Eq. (16)	Eq. (15)	Eq. (26)

Tablo 7: Bellek mekanizmalarının detaylı karşılaştırması.

Burada, Tablo 7'de özetlendiği üzere Okuma ve Yazma işlemlerine vurgu yaparak birkaç temsilci bellek mekanizmasını inceliyoruz. Özellikle, tablosal ve parametrik Q-değer temsilleri ile EC tabanlı yöntemleri tartışıyoruz.

Tablosal ortamda, bellek açık bir tabloyu $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ biçiminde tutar; burada Okuma, tablonun $Q\left(s, M, a\right)$ doğrudan sorgulanması iken, Yazma geçiş gözlemi sonrası durum-aksiyon çiftine ait girdinin güncellenmesine karşılık gelir ve bu standart Eq. (8) kapsamındaki TD öğrenimini izler. Ayrık alanların ötesine geçmek için, derin Q-öğrenme Q fonksiyonunu bir sinir ağıyla öğrenir $Q\left(s, M, a; \theta\right)$; Okuma işlemi Eq. (7) uyarınca geri çağırma politikası μ üzerinden örnekler çekerken, Yazma işlemi parametreleri θ TD hatasını minimize ederek günceller:

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,c,r,s',M,M')}\left[\left(Q(s, M, c; \theta) - \left[r + \gamma\alpha \log \sum_{c' \in M'} \exp\left(Q\left(s', M', c'; \theta\right)\right)\right]\right)^2\right], \tag{24}$$

burada $\bar{\theta}$hedef Q ağıdır. TD öğrenimi kaybının θparametresine göre gradyanı şu şekilde verilir:

$$\nabla_\theta \mathcal{L}(\theta) = 2\mathbb{E}_{(s,c,r,s',M,M')}\left[\left(Q(s, M, c; \theta) - y\right) \nabla_\theta Q(s, M, c; \theta)\right], \tag{25}$$

where $y = r + \gamma\alpha \log \sum_{c' \in M'} \exp\left(Q\left(s', M', c'; \theta\right)\right)$. Bu parametrik formülasyon, tablo yöntemlerinin yalnızca bireysel girdileri ezberlemesinin aksine, paylaşılan parametreler θ aracılığıyla durumlar ve eylem alanları arasında genelleme yapılmasını sağlar. Ancak, bu avantaj optimizasyon kararsızlığı ve yüksek veri gereksinimi maliyetiyle birlikte gelir; çünkü yaklaşık hata, parametre uzayı boyunca küresel olarak yayılabilir. Bu sınırlama, Bölüm 3'te sunulan EC tabanlı yöntemleri harekete geçirir; burada değer tahmini, öğrenilebilir bir çekirdek ile düzenlenir (bkz. Eq. (9)). Bu bellek tasarımında, Okuma işlemi Eq. (7)'de tanımlı erişim politika dağılımından örnekler seçerken, Yazma işlemi ek olarak $\left(s, c, Q\right)$ verilerini bir episodik belleğe kaydeder ve çekirdek parametrelerini Eq. (10) ile Eq. (11)'deki gradyan kullanılarak ağırlıklandırma fonksiyonunu optimize edecek şekilde günceller.
Bu yaklaşım, yalnızca çekirdeği parametrize ederek eşleşen durumların tarihsel Q-değerlerini düzenler; böylece durum uzayı boyunca genelleme sağlarken, derin Q-öğrenmeye kıyasla veri verimli adaptasyon ve geliştirilmiş kararlılık sunar.

4. bölümde, CBR ajanı planlayıcı olarak uygulanmakta olup, hem parametrik olmayan hem de parametrik bellek mekanizmalarını kullanmaktadır. Parametrik olmayan varyantta, Yazma işlemi her gözlemlenen vakayı $\left(s_t, a_t, r_t\right)$ denklem $\left(12\right)$'deki gibi vaka bankasına eklerken, Okuma işlemi mevcut sorgu gömme ile saklanan durumlar arasında kosinüs benzerliği eşleştirmesi yaparak en ilgili deneyimlere erişir ve ardından denklem $\left(13\right)$ 'teki gibi TopK seçimi gerçekleştirir. Parametreleştirme yapılmaksızın bu benzerlik tabanlı erişim,

is a common design in CBR and provides an effective means of reusing past experiences. Alongside the non-parametric approach, the single-step nature of the deep research setting permits fitting a parametric Q-function directly, as the reduced state space substantially lowers data requirements. In the single-step case, the temporal-difference bootstrap vanishes, so the learning objective reduces to Eq.(14). Furthermore, since the reward signal in the deep research scenario is binary, we replace the MSE objective with a CE loss. This choice avoids the vanishing-gradient problem near the boundaries 0 and 1, while providing more numerically stable training signals. Consequently, the final updating objective is reformulated as a binary classification loss in Eq.(15), and the resulting gradient is as follows:

$$\nabla_\theta \mathcal{L}(\theta) = \mathbb{E}_{(s,c,r)}\left[\frac{Q(s,c;\theta) - r}{Q(s,c;\theta)(1 - Q(s,c;\theta))}\,\nabla_\theta\, Q(s,c;\theta)\right]. \tag{26}$$

To further stabilise case selection, we also apply a TopK operator in the parametric Read operator Eq. (16) rather than sampling from the retrieval policy μ.

CBR'de yaygın bir tasarımdır ve geçmiş deneyimlerin yeniden kullanılmasında etkili bir yöntem sağlar. Parametrik olmayan yaklaşımla birlikte, derin araştırma ortamının tek adımlı doğası, azaltılmış durum uzayının veri gereksinimlerini önemli ölçüde düşürmesi nedeniyle parametrik bir Q-fonksiyonunun doğrudan uyarlanmasına olanak tanır. Tek adımlı durumda, zamansal fark bootstrap'ı ortadan kalkar, dolayısıyla öğrenme hedefi denklem (14)'e indirgenir. Ayrıca, derin araştırma senaryosunda ödül sinyali ikili olduğundan, MSE hedefi CE kaybı ile değiştirilir. Bu seçim, sınırlar 0 ve 1 yakınlarındaki kaybolan gradyan problemini önlerken, daha sayısal olarak kararlı eğitim sinyalleri sağlar. Sonuç olarak, nihai güncelleme hedefi Eq.(15)'te bir ikili sınıflandırma kaybı olarak yeniden formüle edilmiş ve ortaya çıkan gradyan aşağıdaki gibidir:

$$\nabla_\theta \mathscr{L}(\theta) = \mathbb{E}_{(s,c,r)}\left[\frac{Q(s,c;\theta) - r}{Q(s,c;\theta)(1 - Q(s,c;\theta))}\,\nabla_\theta\, Q(s,c;\theta)\right]. \tag{26}$$

Vaka seçiminde daha fazla kararlılık sağlamak amacıyla, parametreli Okuma operatörü Eq.(16)'da geri çağırma politikası μ 'den örneklemeye kıyasla TopK operatörü uygulanmaktadır.