Adaptation of Agentic AI (arXiv:2512.16301)

2026.01.09
peny.official

kakao

Agentic AI

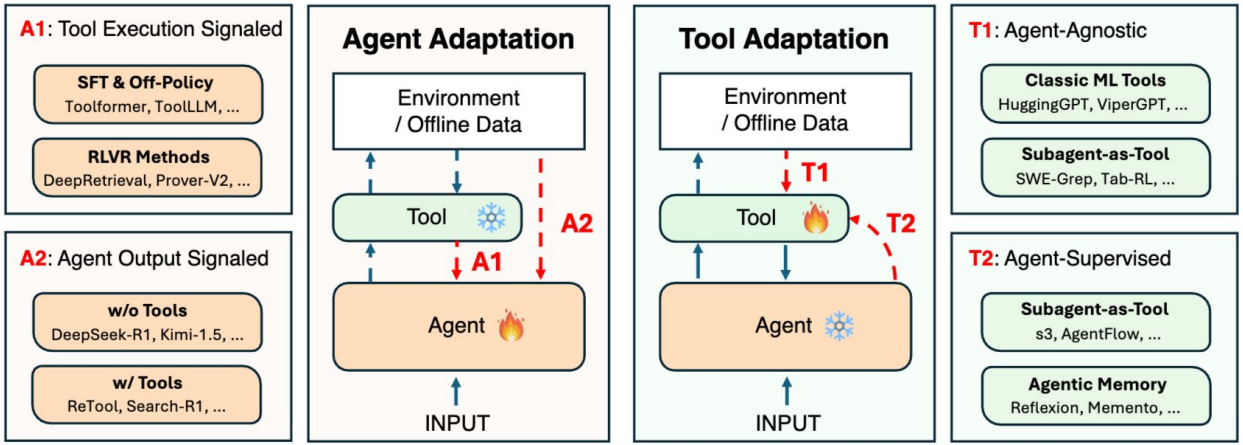
- Agentic AI란 무엇인가?
  - 환경을 인식(Perceive)하고, 추론(Reason)**하며, 행동(Act)할 뿐만 아니라 환경과의 상호작용을 통해 지속적으로 스스로를 개선할 수 있는 자율적인 인공지능 시스템.
  - 기반 모델(Foundation Model)이 있으며, 이를 보조하여 자율성을 확장하는 계획(Planning), 도구 사용(Tool Use), 메모리(Memory) 모듈로 구성

Agentic AI와 Adaption

- Adaption 은 무엇인가?
  - AI가 특정 도메인, 작업 또는 운영 환경의 요구 사항에 더 잘 맞도록 자신의 행동, 의사결정 전략 및 내부 표현을 조정하는 필수적인 과정
  - 프롬프트 엔지니어링(Prompt Engineering): 모델의 파라미터를 수정하지 않고 지시어를 통해 AI의 행동을 유도
  - 파인튜닝(Fine-tuning): 파라미터를 직접 업데이트함
- Adaption 이 왜 필요한가?
  - 범용 모델(LLM) 을 특정 도메인에서 사용할 수 있는 수준으로 정렬(Alignment)하기 위한 핵심 동력

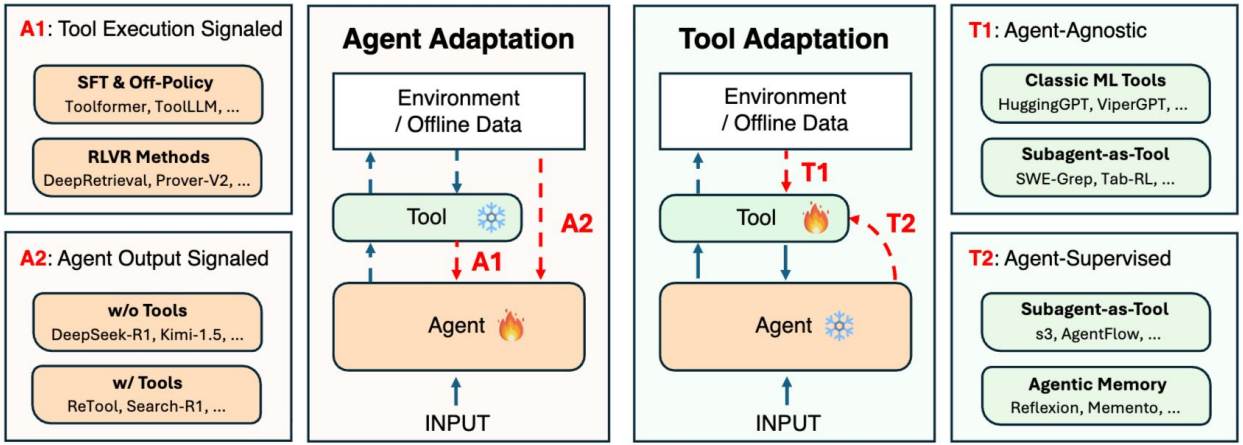
패러다임 소개

Adaption 프레임워크의 네 가지 패러다임 (A1, A2, T1, T2)



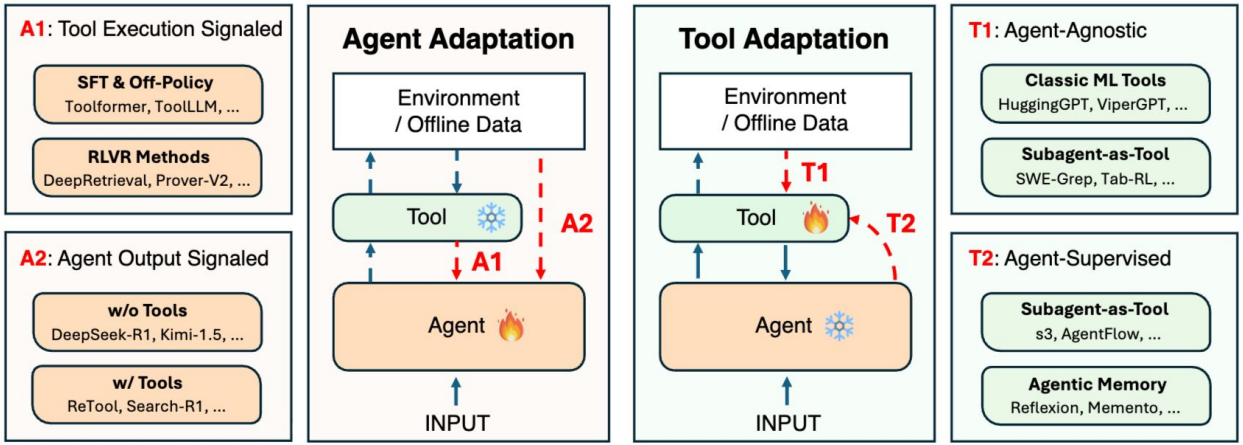
- 분류 기준 1: 최적화 대상 (Optimization Target) - 누가?
  - 에이전트(Agent): 모델 내부 매개변수나 행동 정책 수정
  - 도구(Tool): 검색기, 메모리 등 외부 모듈 최적화
- 분류 기준 2: 적응 신호 유형 (Adaptation Signal) - 무엇을?
  - A1/A2: 도구 실행 결과에서 신호를 얻느냐, 최종 답변에서 얻느냐의 차이
  - T1/T2: 도구가 에이전트와 독립적이냐, 에이전트의 감독을 받느냐의 차이

패러다임 A1 - 도구 실행 신호 기반 에이전트 적응



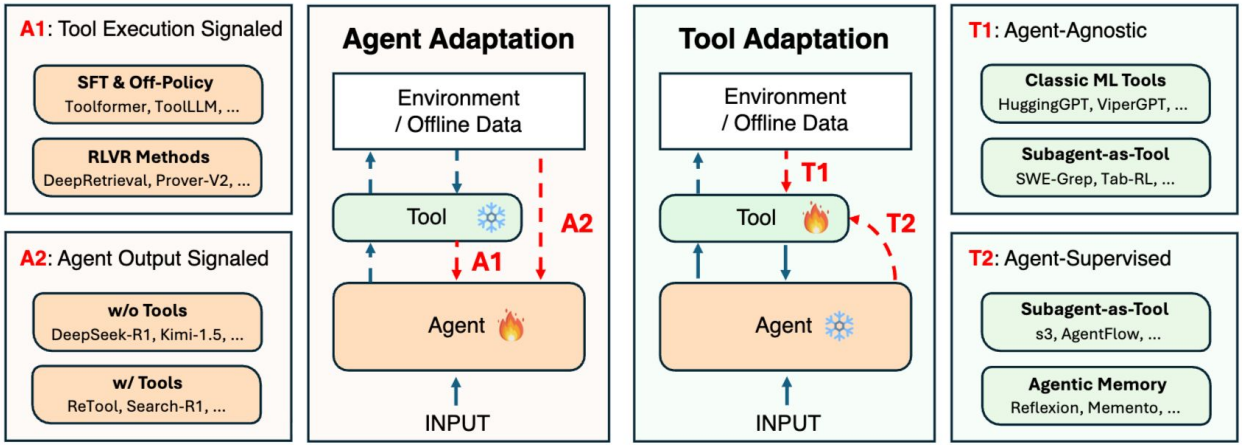
- 핵심 개념
  - 에이전트가 도구를 사용한 실행 결과를 직접적인 피드백으로 삼아 학습
  - 도구를 얼마나 잘 다루는지, 즉 '도구 사용의 기술'에 집중
- 주요 적응 신호
  - 코드 실행의 성공/실패, 검색된 문서의 관련성 점수, API 호출 결과 등 검증 가능한 데이터

패러다임 A2 - 에이전트 출력 신호 기반 에이전트 적응



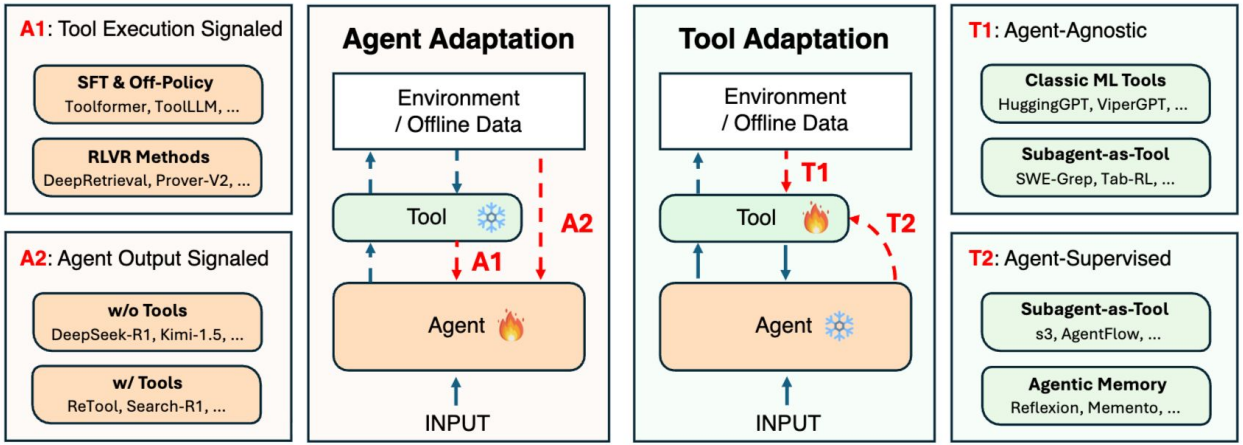
- 핵심 개념
  - 도구 실행 과정보다 에이전트의 '최종 출력물'(답변, 계획)의 품질을 기준으로 에이전트를 최적화
  - 도구를 '언제', '어떻게' 사용할지에 대한 상위 수준의 정책 강화 -> 복잡한 워크플로우를 조율하는 오케스트레이션 능력 내재화

패러다임 T1 - 에이전트 독립적인 도구 적응



- 핵심 개념
  - 에이전트는 건드리지 않고, 도구의 실행 결과로 '도구 자체'를 최적화
  - 특정 에이전트에 종속되지 않는 '플러그 앤 플레이' 방식의 모듈형 도구 구축
- 사례
  - 시각 모델: 미리 학습된 CLIP, SAM 등을 API로 호출해 사용

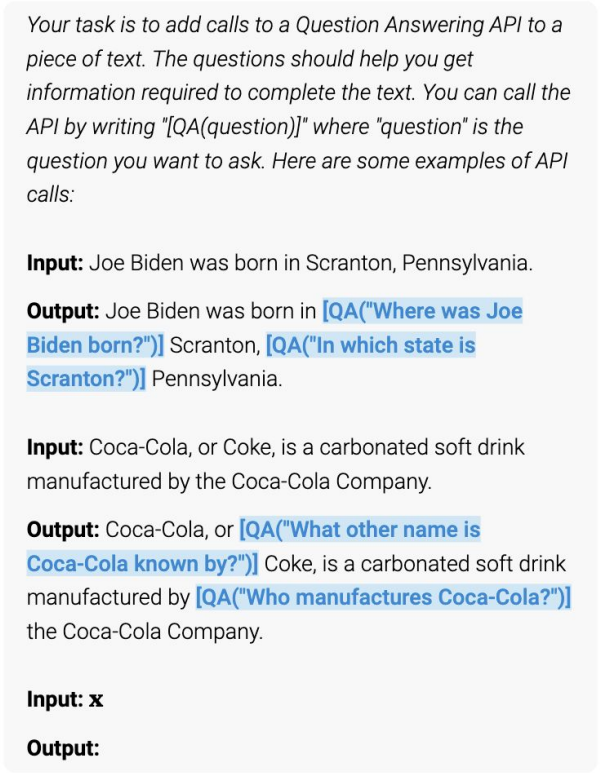
패러다임 T2 - 에이전트 감독형 도구 적응



- 핵심 개념
  - 고정된 에이전트의 피드백을 기반으로 작은 서브에이전트(도구)를 훈련
  - 도구가 특정 에이전트에 맞게 적응 (Symbiotic Inversion)
- 혁신
  - 데이터 효율성 - s3의 경우 A2 방식보다 약 70배 적은 데이터로 동등한 성능 달성

패러다임 적용

패러다임 A1 - 도구 실행 신호 기반 에이전트 적응



Your task is to add calls to a Question Answering API to a piece of text. The questions should help you get information required to complete the text. You can call the API by writing "[QA(question)]" where "question" is the question you want to ask. Here are some examples of API calls:

Input: Joe Biden was born in Scranton, Pennsylvania.

Output: Joe Biden was born in [QA("Where was Joe Biden born?")] Scranton, [QA("In which state is Scranton?")] Pennsylvania.

Input: Coca-Cola, or Coke, is a carbonated soft drink manufactured by the Coca-Cola Company.

Output: Coca-Cola, or [QA("What other name is Coca-Cola known by?")] Coke, is a carbonated soft drink manufactured by [QA("Who manufactures Coca-Cola?")] the Coca-Cola Company.

Input: x

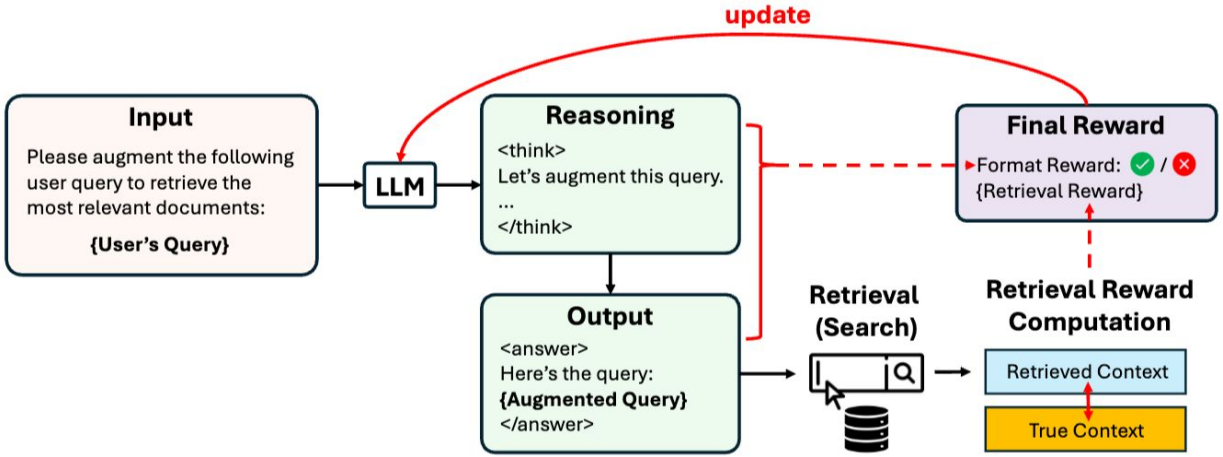
Output:

Figure 3: An exemplary prompt $P(\mathbf{x})$ used to generate API calls for the question answering tool.

ToolFormer (https://arxiv.org/pdf/2302.04761)

1. 모델이 API 호출(도구) 유도하는 예시 (Demonstrations)가 포함된 프롬프트 P(x) 를 작성.
2. API 호출 결과 를 포함했을 때(L)와 포함하지 않았을 때의 손실값(L-) 을 비교.
3. 도구 결과가 들어갔을 때의 Perplexity가 기준 이상으로 감소하는 경우에만 해당 호출을 '유용한 것'으로 간주하고 유지

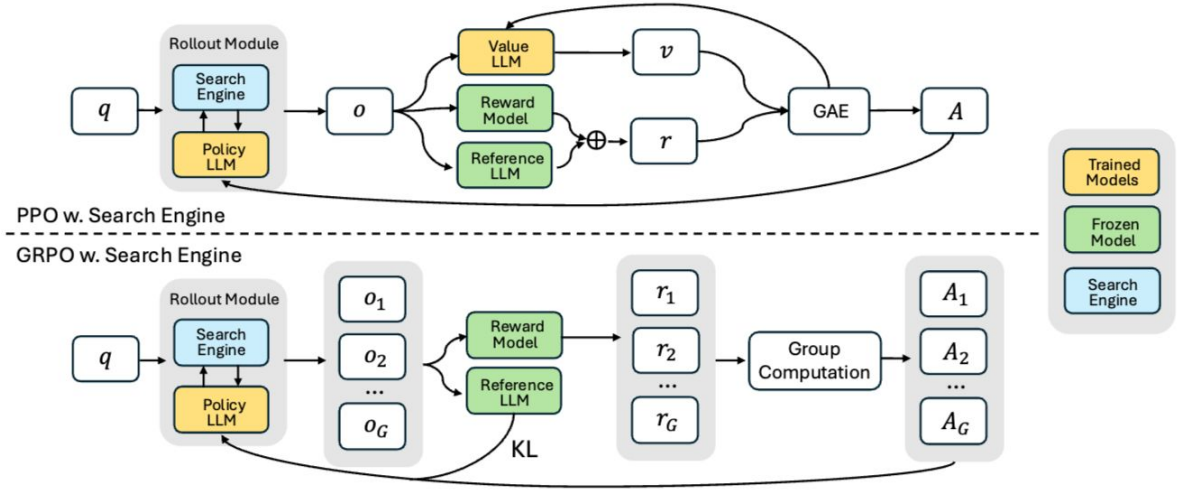
패러다임 A1 - 도구 실행 신호 기반 에이전트 적응



DeepRetriever (https://arxiv.org/pdf/2503.00223)

강화학습(RL)을 통해 언어 모델(LLM)을 최적의 '쿼리 생성기'로 훈련시켜 정보 검색(IR) 성능을 극대화하는 것.
생성된 쿼리가 가져온 문서의 Recall@K나 NDCG 같은 검색 지표를 보상으로 삼아 에이전트를 학습시킵니다.

패러다임 A2 - 에이전트 출력 신호 기반 에이전트 적응



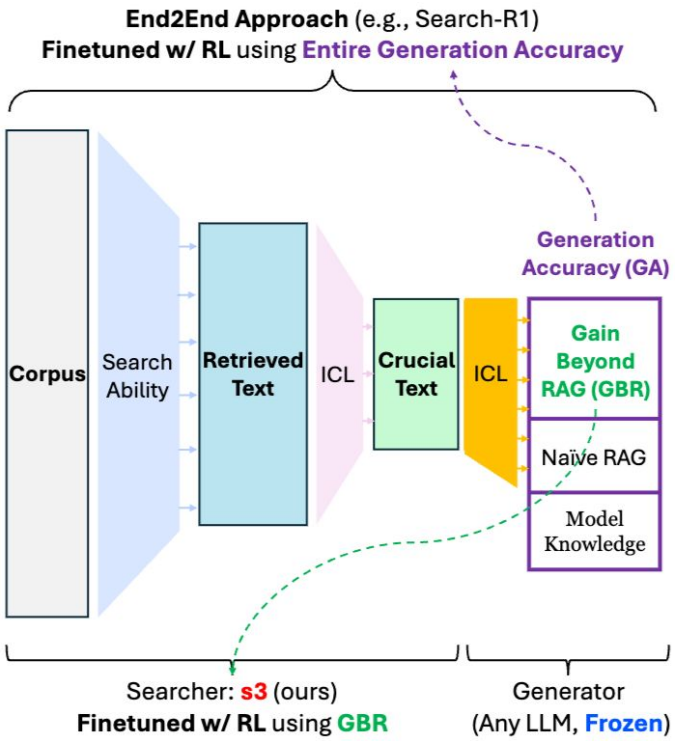
DeepSeek-R1(https://arxiv.org/pdf/2501.12948), Search-R1(https://arxiv.org/pdf/2503.09516)

검증 가능한 보상 기반 강화학습(RLVR)을 통해 LLM의 내재적 추론 능력 강화
추론 과정 자체에 제약을 두지 않고, 정답에 대한 최종 예측의 정확성 에만 전적으로 보상을 기초

패러다임 T1 - 에이전트 독립적인 도구 적응

- CLIP(https://arxiv.org/pdf/2103.00020)
  - 인터넷에서 수집한 4억 개의 (이미지, 텍스트) 쌍을 사용하여 대규모 대조 학습을 통해 처음부터 끝까지 독립적으로 훈련
  - 별도의 추가 훈련 없이 바로 호출하여 사용할 수 있는 '플러그 앤 플레이 (Plug-and-play)' 지원
- Whisper(https://arxiv.org/pdf/2212.04356)
  - 68만 시간의 다국어 전사 데이터를 통한 약지도 학습(Weak Supervision) 수행.
  - API 형태로 음성 인식(ASR), 번역, 언어 식별 결과를 에이전트에게 제공

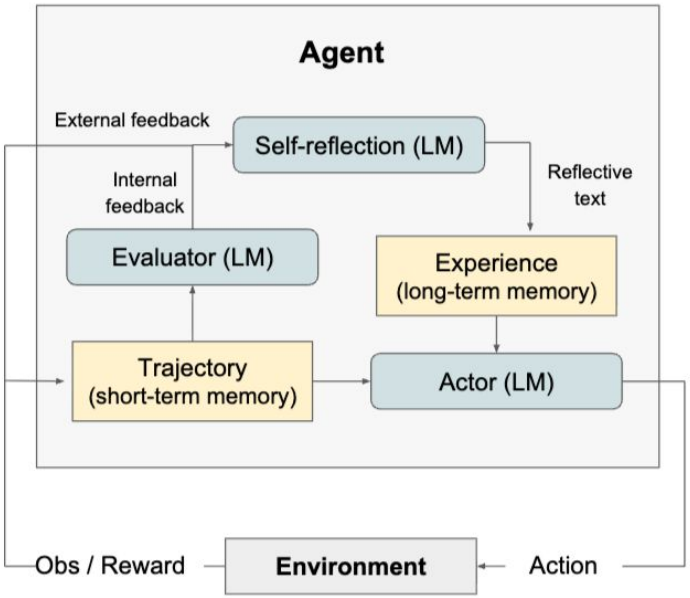
패러다임 T2 - 에이전트 감독형 도구 적응



S3 (https://arxiv.org/pdf/2505.14146)

1. s3는 지식을 생성하는 '생성기(Generator LLM)'를 동결(Frozen) 상태로 두고, 오직 쿼리를 생성하고 문서를 선택하는 '검색기(Searcher tool)' 모델만 강화학습
2. '기본 RAG(Naive RAG)' 성능을 베이스라인으로 삼고, 검색 에이전트가 추가로 가져온 정보 덕분에 정확도가 얼마나 '상승'했는지(Gain Beyond RAG)만을 측정하여 보상
3. 에이전트 전체를 가르칠 때보다 데이터를 무려 **70배나 적게 쓰고도 대등하거나 더**

패러다임 T2 - 에이전트 감독형 도구 적응



Reflextion (https://arxiv.org/pdf/2303.11366)

1. 언어적 반성 (Self-Reflection): 자가 반성 모델은 실패한 궤적과 평가 신호를 분석하여, "어느 지점에서 실수가 발생했는지"와 "다음에는 어떻게 개선해야 하는지"를 자연어 텍스트로 요약
2. 반성 텍스트는 장기 메모리(Long-term memory) 에 저장된 다음 시도(Trial)에서 에이전트는 이전에 저장된 반성 내용들을 자신의 프롬프트 문맥(Context)으로 주입

패러다임 비교

패러다임 비교 분석 I - 비용과 효율성

- 비용 및 유연성 (Cost and Flexibility)
  - 에이전트 적응(A1, A2)은 수십억 개의 파라미터를 가진 모델을 훈련해야 하므로 막대한 컴퓨팅 자원과 엔지니어링 노력이 필요
  - 반면 도구 적응(T1, T2)은 상대적으로 크기가 작은 외부 모듈만 최적화하므로 비용이 저렴.
  - A1/A2는 에이전트의 정책 자체를 완전히 바꿀 수 있는 높은 '파라미터 유연성(Parametric Flexibility)'을 제공 가능
- 데이터 효율성 (Data Efficiency)
  - 도구 중심 적응의 우위 - 최근 연구 결과에 따르면, 고정된 백본 주변에 작은 서브에이전트를 훈련시키는 **T2 방식이 A2 방식보다 훨씬 적은 데이터로도 대등하거나 더 나은 성능**
  - T2 서브에이전트가 일반적인 추론 능력을 다시 배울 필요 없이 '**절차적 기술'에만 집중**하기 때문입니다.

패러다임 비교 분석 II - 일반화와 모듈화

- 일반화 능력 (Generalization Capability)
  - T1: 광범위한 데이터 분포에서 학습된 도구들은 서로 다른 에이전트나 작업 환경에서도 잘 일반화
  - T2: 자신을 감독하는 강력한 기초 모델의 편향(inductive biases)을 물려받아 교차 도메인에서 견고함을 보이는 경우가 많음
  - A1/A2: 특정 환경에 최적화(on-policy)될 경우, 명시적인 정규화 없이는 해당 환경에 과적합(Overfitting)될 위험이 큼.
- 모듈성
  - T1/T2는 에이전트를 건드리지 않고도 '핫스왑(Hot-swapping)'하거나 독립적으로 업그레이드할 수 있어 시스템 유지보수가 쉬운 편
  - A1/A2는 Monolithic 으로 새로운 기능을 추가할 때마다 전체를 다시 학습시켜야 하며 이 과정에서 'Catastrophic Forgetting 발생 가능

이상적 패러다임 - 하이브리드 시스템

평상시에는 비용이 저렴하고 효율적인 T1/T2 방식을 통해 시스템을 지속적으로 전문화하다가, 결정적인 시점에 A1/A2 업데이트를 단행하여 에이전트의 내부 추론 능력을 한 단계 도약시키는 하이브리드 전략

하나의 거대한 단일 모델(Monolithic)이 아니라, '안정적인 추론 코어 = LLM' '와 '지속적으로 진화하는 적응형 도구들'이 원칙적으로 조율된 연합체 (Federation)

AI 에이전트의 미래

에이전틱 AI의 주요 응용 분야

1. 딥 리서치 (Deep Research) - 과학적 조사 및 지식 합성 자동화
   a. OpenAI DeepResearch, Claude deep-search
2. 소프트웨어 개발 (Software Development) - 엔지니어링 워크플로우 자율화
   a. Cursor, Claude Code
3. 컴퓨터 사용 (Computer Use) - GUI 기반의 직접적인 장치 조작
   a. OpenCUA: 대규모 GUI 데이터를 통해 화면 요소 식별 및 조작 역량 확보
   b. AgentTrek: 웹 튜토리얼을 통해 새로운 인터페이스 패턴 자율 학습
   c. ACE (Agentic Context Engineering): 변화하는 화면 문맥을 '플레이북'으로 구조화하여 실행 지연 시간 감소
4. 약물 발견 및 개발 (Drug Discovery) - 생물 의학 파이프라인의 혁신
   a. GeneAgent, SyntheMol

미래의 도전 과제 - 4대도전 과제

1. 공생적 적응 (Co-Adaptation)
   a. 에이전트(A)와 도구(T)를 동일한 학습 루프에서 동시 최적화: 신용 할당(Credit Assignment) 문제, 붉은 여왕(Red Queen) 효과
2. 지속적 적응 (Continual Adaptation)
   a. 새로운 지식을 배우면서 기존 지식을 잊는 '파괴적 망각 (Catastrophic Forgetting)' 방지: PEFT(LoRA), agentic memory
3. 안전한 적응 (Safe Adaptation)
   a. Unsafe Exploration, Parasitic Adaptation
4. 효율적 적응 (Efficient Adaptation)
   a. 거대 GPU 클러스터 의존도를 낮추고 모바일의 학습 가능성 타진

결론 및 미래 비전 - 하이브리드 아키텍처

- 지식(에이전트)과 기술(도구)이 분리되어 협력하는 연합체 (Federation).
  - A1/A2: 가끔씩 단행되는 에이전트의 내부 추론 능력 도약.
  - T1/T2: 평상시의 저비용·고효율적인 시스템 전문화 유지.
- 핵심 통찰: 안정적인 추론 코어와 지속 진화하는 적응형 도구 생태계의 조화

END