Initial Steps in Integrating Large Reasoning and Action Models for Service Composition

Ilche Georgievski and Marco Aiello, IAAS-SC, University of Stuttgart, DE

SummerSOC 2025

AI is eating the world

- OpenAI
- Google
- DeepSeek
- Inflection AI
- Databricks
- AWS
- Meta AI
- Amazon
- Cohere
- Microsoft
- Meta
- Nvidia
- GPT-4
- Claude
- Why Labs
- Claude 3 Haiku
- Mistral AI
- Anthropic
- 01.AI
- Contextual AI, Inc.
- Mosaic ML, Inc.
- IBM
- Alibaba
- Falcon

AI is eating the world

OpenAI

estimated at 300B

Rank	Name	Market Cap	Price	Today	Price (30 days)	Country
▲1 1	Microsoft MSFT	$3.476 T	$467.68	▲0.82%		USA
▼1 2	NVIDIA NVDA	$3.414 T	$139.99	▼1.36%		USA
3	Apple AAPL	$2.996 T	$200.63	▼1.08%		USA
4	Alphabet (Google) GOOG	$2.049 T	$169.81	▲0.25%		USA
5	Meta Platforms (Facebook) META	$1.721 T	$684.62	▼0.48%		USA
6	Tesla TSLA	$917.00 B	$284.70	▼14.26%		USA
7	Oracle ORCL	$479.91 B	$171.14	▲1.81%		USA
8	Palantir PLTR	$282.97 B	$119.91	▼7.77%		USA
9	IBM IBM	$248.01 B	$266.86	▲0.50%		USA
10	Adobe ADBE	$176.95 B	$415.20	▲0.31%		USA
11	CoreWeave CRWV	$64.82 B	$135.05	▼17.20%		USA
12	Cambricon Technologies 688256.SS	$35.43 B	$84.88	▼1.59%		China
13	Dynatrace DT	$16.46 B	$54.94	▲1.55%		USA
14	Mobileye MBLY	$13.39 B	$16.50	▼1.70%		Israel
15	Tempus AI TEM	$10.15 B	$58.66	▼6.56%		USA
16	Aurora Innovation AUR	$9.99 B	$5.65	▼2.75%		USA
17	UiPath PATH	$7.09 B	$13.26	▲1.61%		USA

"Company with no business plan buys company with no product"

For 6.5 billion dollars

Builder.ai

AI for software engineering or 700 SE humans

promised to make software creation "as easy as ordering pizza"

raised $450 million and achieved a valuation of $1.5 billion

reportedly owes $85 million to Amazon and $30 million to Microsoft in unpaid cloud services

on the other hand

much more sustainable than real AI…

To estimate the **daily electricity consumption** for 700 people:

- **Annual per capita consumption**: 1,395 kWh
- **Daily per capita consumption**:
  1,395 kWh ÷ 365 days ≈ **3.82 kWh/day**
- **Total for 700 people**:
  3.82 kWh/day × 700 people ≈ **2,674 kWh/day**

based on arxiv publications


5’6”
5’0”
4’6”
4’0”
3’6”
3’0”
POLICE DEPT
GUILTY
00.0
Marco Aiello
01729204812


The AI Revolution and the Future of Work: Threats and Opportunities
A Challenge for the Next 50 Years of Automated Service Composition
Service composition in the ChatGPT era
A Paradigm Shift in Service Research: The Case of Service Composition
Compositio Prompto: An Architecture to Employ Large Language Models in Automated Service Computing
Offenlegungsschrift

Service Composition

The Case of Service Composition

Definition

Service Composition is the process of integrating independent looslely coupled services starting from a user request based on the ones available in the execution context. The services communicate over a network and are modular, allowing for flexible and dynamic composition. The *orchestrator* is responsible for coordinating the service composition.

Service Composition as AI Planning

- **Artificial Intelligence Planning and Scheduling** is a branch of Artificial Intelligence devoted to the study of algorithms and systems to empower intelligent agents with the ability to pursue their goals.

- *Goal:* a description of the state of the world to realise
  user request

- *Plan:* an algorithm that describes how to reach a goal state
  a composition to orchestrate

- *Environment:* a system the state of which can be sensed and changed by the planning actor
  APIs, service states, domain knowledge

Large Reasoning Models

- A neural network-based model optimized for multi-step logical and symbolic reasoning
- Trained on heterogeneous datasets: natural language, formal logic, math, code, and multimodal inputs
- Excels at structured problem-solving via in-context learning, chain-of-thought prompting, and tool augmentation
- Designed to perform algorithmic reasoning, planning, and hypothetical simulation
- May incorporate external memory, RAG, or tool use (e.g., calculators, search APIs, WolframAlpha)
- Good for: automated theorem proving, scientific discovery, decision support, etc.

The Illusion of Thinking: Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity

Parshin Shojaee*† Iman Mirzadeh* Keivan Alizadeh
Maxwell Horton Samy Bengio Mehrdad Farajtabar

Apple

Abstract

Recent generations of frontier language models have introduced Large Reasoning Models (LRMs) that generate detailed thinking processes before providing answers. While these models demonstrate improved performance on reasoning benchmarks, their fundamental capabilities, scaling properties, and limitations remain insufficiently understood. Current evaluations primarily focus on established mathematical and coding benchmarks, emphasizing final answer accuracy. However, this evaluation paradigm often suffers from data contamination and does not provide insights into the reasoning traces' structure and quality. In this work, we systematically investigate these gaps with the help of controllable puzzle environments that allow precise manipulation of compositional complexity while maintaining consistent logical structures. This setup enables the analysis of not only final answers but also the internal reasoning traces, offering insights into how LRMs "think". Through extensive experimentation across diverse puzzles, we show that frontier LRMs face a complete accuracy collapse beyond certain complexities. Moreover, they exhibit a counter-intuitive scaling limit: their reasoning effort increases with problem complexity up to a point, then declines despite having an adequate token budget. By comparing LRMs with their standard LLM counterparts under equivalent inference compute, we identify three performance regimes: (1) low-complexity tasks where standard models surprisingly outperform LRMs, (2) medium-complexity tasks where additional thinking in LRMs demonstrates advantage, and (3) high-complexity tasks where both models experience complete collapse. We found that LRMs have limitations in exact computation: they fail to use explicit algorithms and reason inconsistently across puzzles. We also investigate the reasoning traces in more depth, studying the patterns of explored solutions and analyzing the models' computational behavior, shedding light on their strengths, limitations, and ultimately raising crucial questions about their true reasoning capabilities.

Stop Anthropomorphizing Intermediate Tokens as Reasoning/Thinking Traces!

Subbarao Kambhampati **Kaya Stechly** **Karthik Valmeekam** **Lucas Saldyt** **Siddhant Bhambri**

Vardhan Palod **Atharva Gundawar** **Soumya Rani Samineni** **Durgesh Kalwar** **Upasana Biswas**

School of Computing & AI
Arizona State University

Abstract

Intermediate token generation (ITG), where a model produces output before the solution, has been proposed as a method to improve the performance of language models on reasoning tasks. These intermediate tokens have been called "reasoning traces" or even "thoughts" – implicitly anthropomorphizing the model, implying these tokens resemble steps a human might take when solving a challenging problem. In this paper, we present evidence that this anthropomorphization isn't a harmless metaphor, and instead is quite dangerous – it confuses the nature of these models and how to use them effectively, and leads to questionable research.

Large Action Models

- A parameter-rich neural policy model trained to map from high-dimensional observations and goals to action distributions
- Leverages transformer-based architectures for sequence modeling of action trajectories
- Operates over state-action-return triples (or variations) for temporal credit assignment and long-horizon planning
- Trained via offline reinforcement learning, behavior cloning, or trajectory-level supervision from expert demonstrations or synthetic data
- Can ingest multimodal inputs and output low-level control signals or symbolic action commands
- Supports zero-shot generalization across tasks via goal-conditioning, prompting, or language grounding
- Frequently deployed in embodied agents, robotic manipulation, navigation, game environments, and tool use contexts

Language Models (LMs)

Large Language Models (LLMs)

- Large-scale knowledge of language patterns
- Understands natural language
- Excels at generating coherent text
- Efficient for translation Q&A, etc.

- No deliberate, iterative reasoning
- No direct interaction with environment
- Tends to hallucinate

GPT-3.5, GPT-4, Claude, LLaMa, BERT, Qwen, Grok, Gemini 1

Large Reasoning Models (LRMs)

- Understands natural language
- Deep, explicit reasoning
- Strong at planning and problem-solving
- Often uses explicit structure of reasoning

- Typically does not act on the environment
- May require extensive computational resources
- Can be slower

OpenAI o1/o3, DeepSeek R1, Gemini 2.0, QwQ

Large Action Models (LAMs)

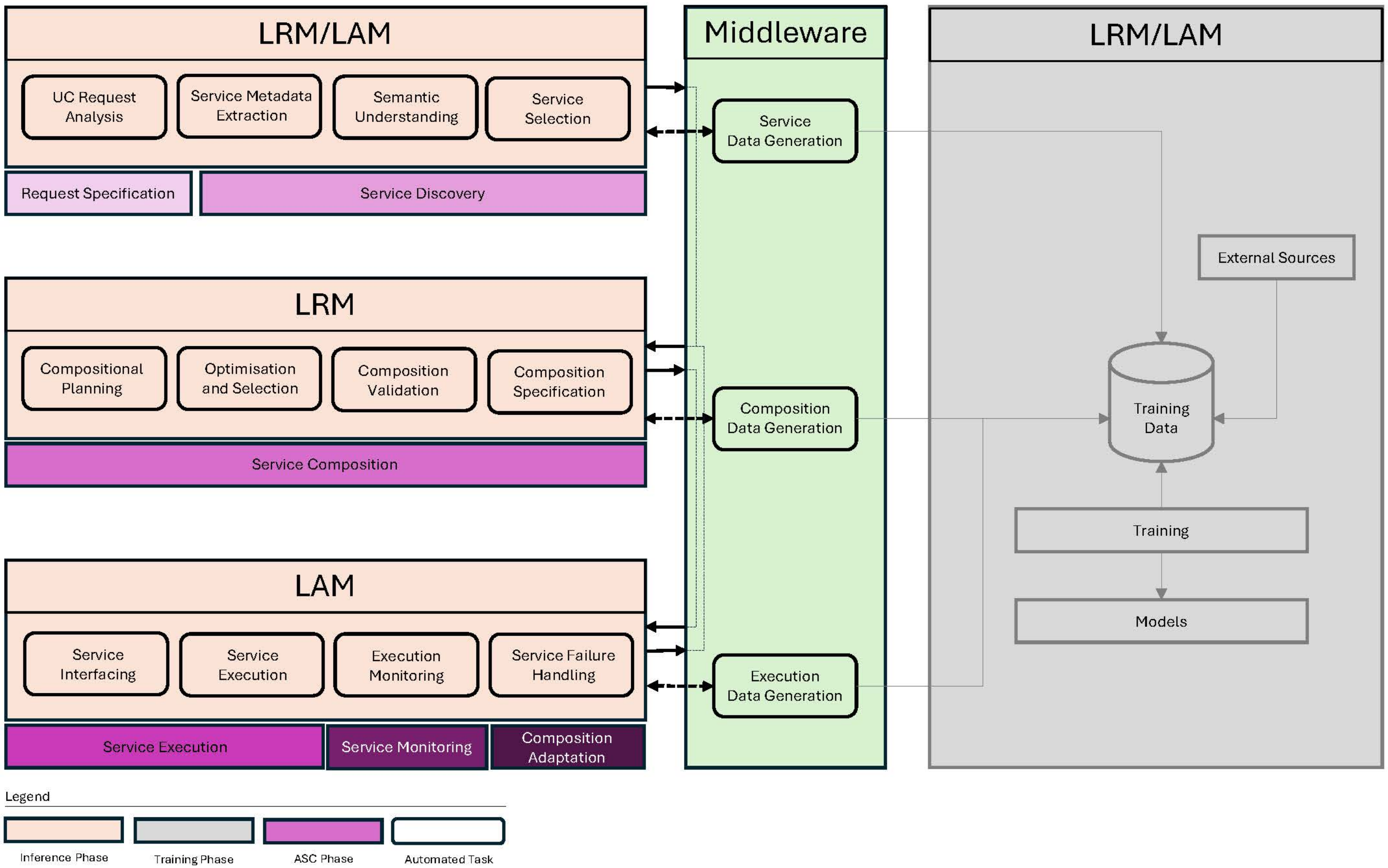
- Understands multimodal inputs
- Directly executes actions in environments
- Integrates sensing with action outputs
- Can handle tasks in real time

- Often weak at (high-level) reasoning
- Requires specialised data (e.g., action logs)
- Can be difficult to train for safety and reliability

Google RT-1/RT-2, DeepMind Gato, Rabbit R1, CogAgent, ScreenAI, xLAM

Legend

Strengths	Limitations	Examples


LRM/LAM
UC Request Analysis
Service Metadata Extraction
Semantic Understanding
Service Selection
Request Specification
Service Discovery
LRM
Compositional Planning
Optimisation and Selection
Composition Validation
Composition Specification
Service Composition
LAM
Service Interfacing
Service Execution
Execution Monitoring
Service Failure Handling
Service Execution
Service Monitoring
Composition Adaptation
Middleware
Service Data Generation
Composition Data Generation
Execution Data Generation
LRM/LAM
External Sources
Training Data
Training
Models
Legend
Inference Phase
Training Phase
ASC Phase
Automated Task

Concluding remarks

Reflection on Initial Steps

- LLM, LRM, LAM can cover various aspects of Service Composition
- Promising technologies with some known and yet unknown limitations
- See you at SummerSOC 2026 for more

Universität Stuttgart
Institute of Architecture of Application Systems

Thank you! Vielen Dank! Grazie! Merci! Bedankt!

多謝!

Marco Aiello

E-Mail marco.aiello@iaas.uni-stuttgart.de
Telefon +49 (0) 711 685-88471
www.iaas.uni-stuttgart.de/en/department-service-computing/ or aiellom.it

Universität Stuttgart
Service Computing Dep — IAAS
Universitätsstr. 38, 70569 Stuttgart