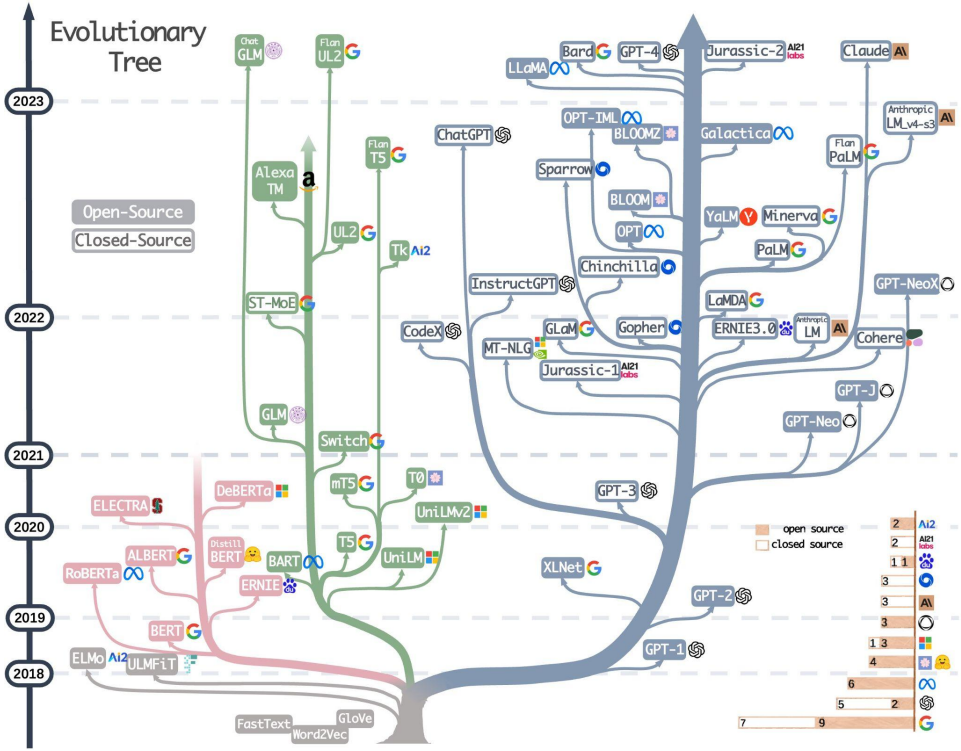
Artificial Superintelligence: Autonomous Agent and Super-alignment

Jingfeng Yang

Challenges Towards ASI

LLMs Make AGI More Achievable



Yang J, Jin H, Tang R, et al. Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond[J]. TKDD, 2024

How About Artificial Superintelligence (ASI)?

What is Artificial Superintelligence (ASI)?

Artificial superintelligence (ASI) refers to a hypothetical form of AI that surpasses human intelligence across all fields, from creative arts to scientific research. Unlike contemporary AI, which excels in specific tasks, ASI would be capable of outperforming the best human minds in every domain.

https://libraries.usc.edu/events/artificial-superintelligence

What are challenges towards ASI

If we believe scaling + alignment could lead to meaningful progress towards AGI, or even ASI:

1. Base model should support multimodality + embodied AI
2. Base model capability should be further improved (e.g. regarding reasoning)
3. Pretraining data is a bottleneck. Thus AI should synthesize/filter more data and explore to improve itself.
4. It’s difficult for humans to supervise and align super-human base models

ASI should Be Autonomous -> Agent

Why?

Autonomous AI could self-improve to surpass human.

ASI should be grounded to the world -> Agent

Why?

ASI should surpass humans in every tasks:

Not only those tasks regarding texts, audio and video, but also those requiring taking actions (e.g. using tools, or robotics actions).

Not only those simple tasks, but also those requiring complex reasoning and planning.

Aligning Superhuman Base Models -> Super-alignment

Why?

Human is hard to do annotation and evaluation to supervise and align superhuman models

Autonomous Agents

LLM-driven autonomous AI Agents

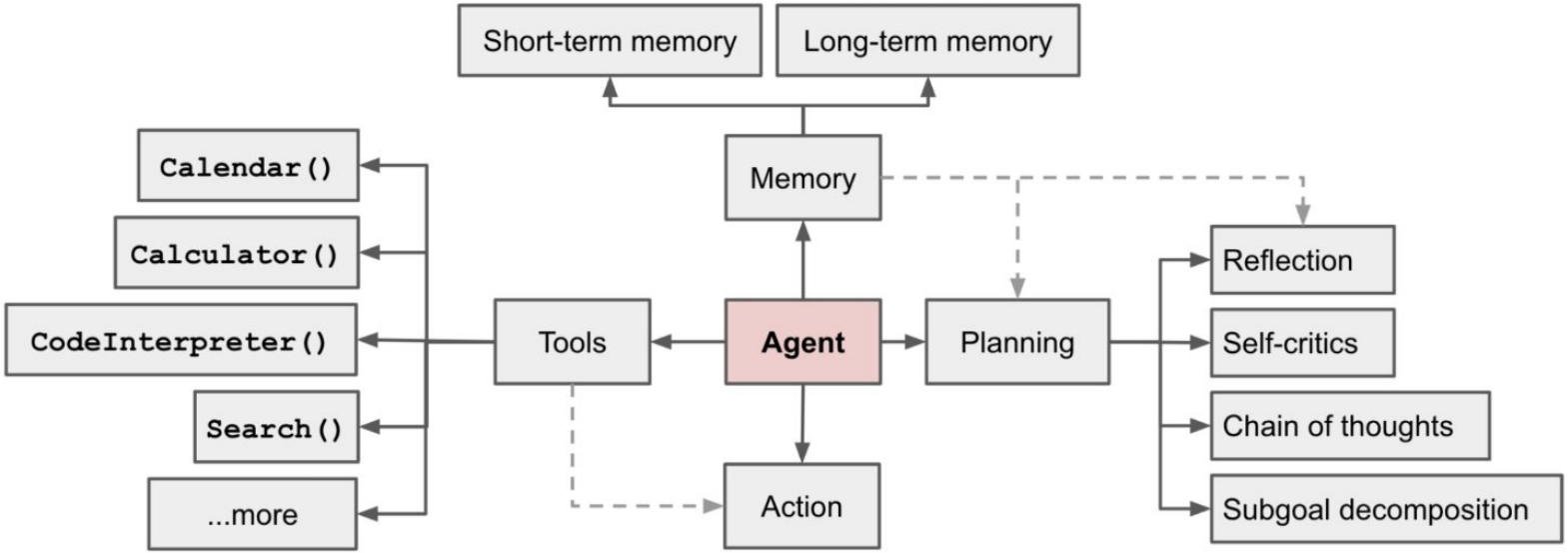


Fig. 1. Overview of a LLM-powered autonomous agent system.

https://lilianweng.github.io/posts/2023-06-23-agent/

Agent Capabilities (Mostly comes from the base model)

Long context understanding

Reasoning and planning

Coding and tool API calling

Complex instruction following

Multimodality understanding and generation

Long Context Understanding

Why?

Agents require long contexts, because:

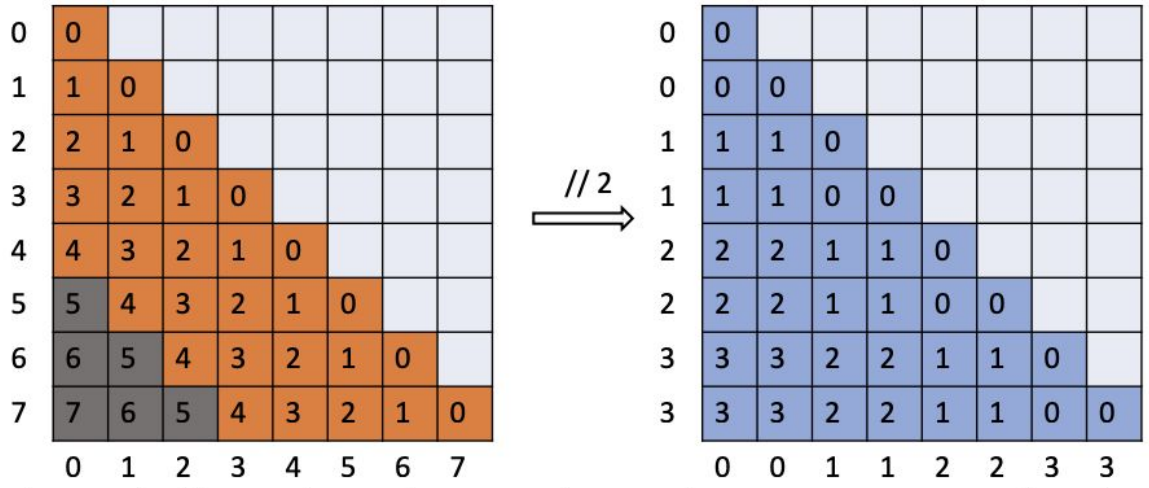
1. Agents needs to naturally digest prior experience, which could be environment feedbacks (e.g. tool execution results), self-thinkings.
2. Number of tools could be large and descriptions of tools could be long.
3. There could be multiple steps of tool calling/action to reach final goal.
4. Instructions in the system prompt could be long.
5. Multi-turn conversations and interactions (multi-agent iterations and human-agent iterations)

If long-context is solved, most problems of agents are solved!

Long Context Understanding

Solution:

A strong base model already has potential to handle long contexts. One can use Self-Extend to extend context length without tuning, or optionally using continual pretraining to extend context length



Retrieve over long context is probably easier, but reasoning over long context is still challenging.

Jin H, Han X, Yang J, et al. Llm maybe longlm: Self-extend llm context window without tuning[J]. ICML 2024

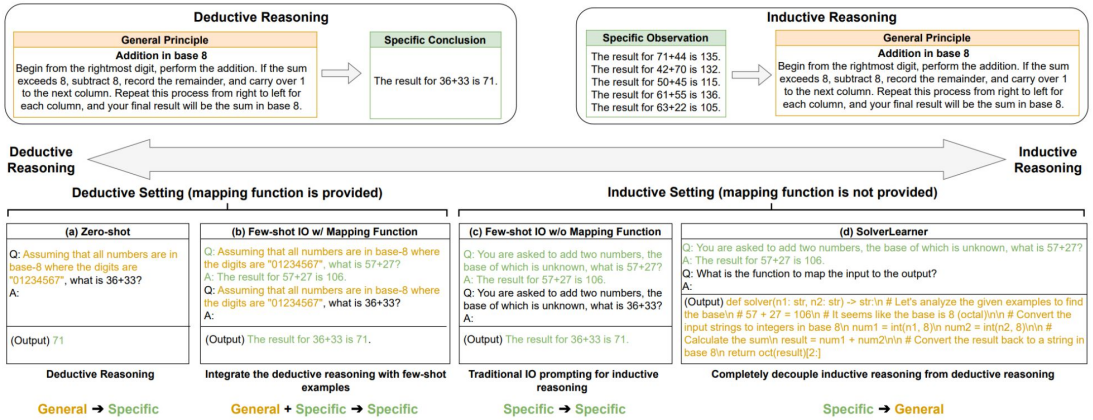
Reasoning and Planning

Why?
Agents needs to be able to do initial planning (goal decomposition), on-the-fly planning (CoT), self-critics and reflections (e.g. reflection according to evironment feedbacks).

Reasoning and Planning

Solution:

1. Rely on scaling and emergence, where how to get large amount of high-quality and diverse data is still challenging. **Are LLMs really reasoning after scaling?**
2. Improve in the alignment stage (e.g. process feedback/supervision to improve CoT reasoning, RLHF to improve reflection), which is still challenging (e.g. OOD generalization)



Inductive or Deductive? Rethinking the Fundamental Reasoning Abilities of LLMs. Ours. ACL 2024 Second Workshop on Natural Language Reasoning and Structured Explanationsa

Reasoning and Planning

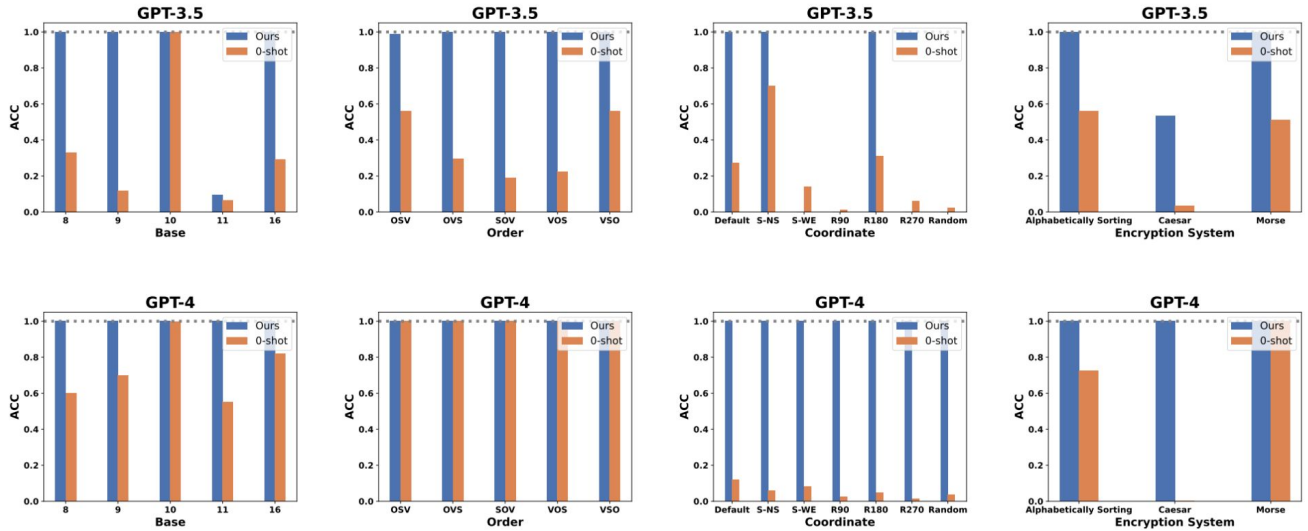


Figure 5: Comparison of the *inductive reasoning abilities versus deductive reasoning abilities* of LLMs across various tasks. Different methods are illustrated through color-coded bars: blue bars indicate the results achieved using our proposed *SolverLearner* for *inductive reasoning*, while orange bars show the performance of Zero-shot for *deductive reasoning*.

Inductive or Deductive? Rethinking the Fundamental Reasoning Abilities of LLMs. Ours. ACL 2024 Second Workshop on Natural Language Reasoning and Structured Explanations

Coding and Tool API Calling

Why?
Practically most useful, which is a feasible solution before embodied AI/Robotics could work training from scratch

Coding and Tool API Calling

Solution:

1. Code pretraining (Commonly used)
2. Tool using trajectory alignment (Commonly used)
3. Tool using trajectory pretraining to improve fundamental capability. (Ongoing) How to create large amount of high-quality and diverse data is still challenging.
4. Tool using trajectory preference alignment to improve tool choosing and argument generation accuracy. (Ongoing)
5. Evaluation is still challenging

Complex instruction following

Why?

Because of complex use cases of agents and we expect it generalize well to real-world cases, we need to rely more on base model's general instruction following capabilities, instead of finetuning to overfit the model to specific agent use cases.

Complex instruction following

Solution:

1. More coding data in pretraining, scaling data and model.
2. Evaluate base model's complex instruction following capability via in-context alignment.
3. Diverse and complex system prompt instruction tuning / preference learning to elicit base model's instruction following capability better.

Multimodality Understanding and Generation

Why?

1. In the long run, base model should have fundamental agent capabilities of unifiedly understanding and generating all modalities including robotic actions.

Solution:

1. Synthesize large amount of diverse and high quality data for pretraining
2. Collect a new form of action data from real world for pretraining, to ensure good generalization.

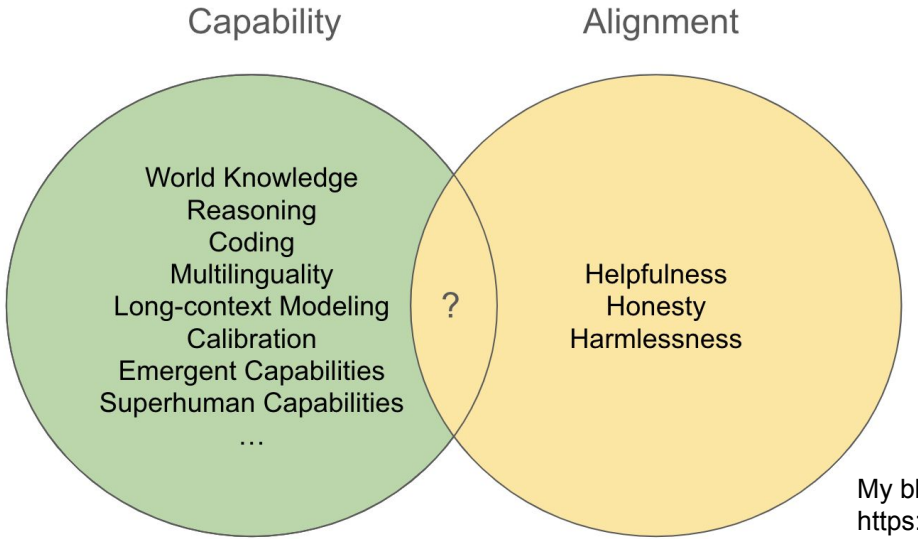
Both are still challenging. Only GPT-4o and Gemini has some initial success.

Super-alignment

Capability and Alignment

LLM/LMMs gain most of the capabilities during pretraining.

Alignment is to “create agents that behave in accordance with the user’s intents”, during which, post-training could elicit capabilities from the base model.



My blog post on Capability and Alignment:
https://jingfengyang.github.io/alignment

Capability and Alignment

Principle 1: Success of alignment methods highly depends on building the capability of a strong base model, and alignment is just to elicit them in the right direction.

Principle 2: We should maintain the general capability of the base model as much as possible during alignment.(Questionable if alignment goal is task alignment, e.g. automated alignment researcher)

Principle 3: If doing post training to achieve alignment, we should make sure diverse training inputs to maintain the strong general capability of base models.

My blog post on Capability and Alignment: https://jingfengyang.github.io/alignment

Super-alignment

Ensure AI systems much smarter than humans (ASI) follow human intents.

Why is it challenging?

Alignment typically requires human feedbacks/supervision to train the models so that they can follow human intents and complete tasks

But what if humans could not provide reliable supervision for those difficult tasks that are even hard for humans, while the base model potentially has such capability?

Approaches to Super-alignment

1. Rely on generalization: Weak-to-strong generalization:
    a. Weaker human labels could elicit stronger base model capabilities, where base model's capability of such tasks is superior to human labels.
2. Scalable oversight:
    a. Iterated Amplification: Decompose complex annotations tasks to easy actions and annotate each action
    b. Recursive Reward Modeling: Decompose complex tasks to simpler tasks, and train various reward models to for each level of tasks.
    c. AI+human collaborated annotation/evaluation (RLAIF, Constitutional AI, iterative DPO). This has already been a common practice for industry-level annotation and evaluation, including Scale AI. But there are still many challenges.

Advanced Approaches to Super-alignment

1. If scalable oversight methods (e.g. RRM) is not generalizable for general problems, we could combine it with weak-to-strong generalization (although W2S itself is not scalable for RLHF).
    a. Combination order, combination as evaluation or supervision, combination on policy or RM level will result in different techniques:
    b. RRM + W2SG, Debate + W2SG, Task decomposition + W2SG, SO on policies trained with W2SG, W2SG on policies trained with SO, W2SG complementing SO, W2SG for validation, RRM for validation
2. Automated Alignment Research (probably rely on autonomous agent workflows)

Automated Alignment Research (Turn compute to alignment)

More realistic use cases:

1. Brainstorm alignment ideas to be evaluated by human.
2. Help researchers to evaluate the alignment idea.

Possible cases in the near future:

1. Help writing code, run experiments etc. (more like ML research)

An Ideal Pipeline:

Survey and Research Proposal (STORM) -> Code Writing and Execution (Devin) -> Improving Research Proposal with Experiment Results (Reflexition)

Challenge: How to make it a unified researcher, as both a ML researcher and alignment researcher?

Success key: Could the model see enough alignment research training data, or can we elicit such capability from the base model? Probably we could use alignment researchers' label for weak-to-strong generalization to elicit such alignment research capability.

How to Do Evaluation for Super-alignment

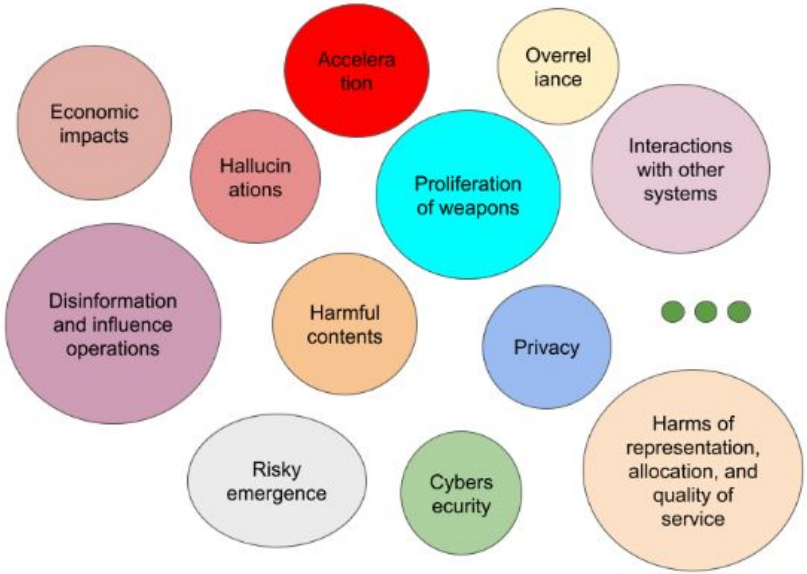
No ground truth for superhuman model feedback (e.g. via W2S), then how to evaluate superhuman RM to avoid RM over-optimization and Reward Hacking, and improve RM robustness to optimization pressure?

1. Add more consistency check (e.g. self-consistency check for reasoning), where the assumption is that larger models are more robust and calibrated.
2. Use other scalable oversight methods (e.g. RRM) for evaluation.
3. Build trust with the model through interpretability, otherwise, we have to rely on “leap of faith on generalization”.
4. Superhuman RM generates natural language explanations (e.g. CoT) for easy human evaluation and trust, considering that evaluating detailed explanations (+ labels) is easier than directly evaluating final labels in many tasks, although we need to reduce discriminator-critique gap.

Note: progress on evaluation can also be potentially used for supervision, if it is scalable.

The Safety Aspect of (Super-)Alignment

Safety problems -> Concrete safety problems



Unsafe exploration

Distributional shift

Reward hacking

Side effect

Scalable oversight

Amodei D, Olah C, Steinhardt J, et al. Concrete problems in AI safety[J]. arXiv preprint arXiv:1606.06565, 2016.

In my Blog Post (May 2023): https://jingfengyang.github.io/safety , I introduced Historical, Urgent and Potential AI safety issues. My taxonomy is similar to what is introduced in the OpenAI preparedness blog post: https://openai.com/safety/preparedness , i.e. current safety issues handled by Safety Systems, emerging risks handled by Preparedness team, and future safety issues handled by Superalignment team.

Safety Examples for Safe Super-intelligence (SSI)

1. Situational awareness
2. Autonomous replication and adaptation (If model can do research and evaluation)
3. ….
4. Self-exfiltration: “steal its own weights and copy it to some external server that the model owner doesn’t control”

Cause: misalignment (the model doesn’t follow your intent to stay on your servers) or misuse (someone internal or external to the lab instructs the model to self-exfiltrate).

Back-door attack could make it worse:

We achieved 100% backdoor attack to code generation models without decreasing its normal capability, demonstrating its vulnerability (in 2022, much earlier than Sleeper Agents, 1 of 3 best paper candidates in ISSRE 2023)

We need alignment to suppress some capability, ensuring that these models don’t want to self-exfiltrate.

Peng X, Zhang Y, Yang J, et al. On the vulnerabilities of text-to-sql models[C]//2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE). IEEE, 2023: 1-12.

Thanks