Article title: Iterative improvements from feedback for language models
Authors: Yuxi Li[1]
Affiliations: rl4reallife.org[1]
Orcid ids: 0000-0002-4270-2487[1]
Contact e-mail: yuxili@gmail.com

DOI: 10.14293/PR2199.000220.v1
Preprint first posted online: 07 July 2023
Keywords: language models, reinforcement learning

Iterative improvements from feedback for language models

Yuxi Li
yuxili@gmail.com

Abstract

Iterative improvements from feedback is a general approach for many, if not all, successful systems. Ground-truth-in-the-loop is critical. Language models (LMs) like ChatGPT are phenomenal, however, there are still issues like hallucinations and a lack of planning and controllability. We may leverage LMs' competence of language to handle tasks by prompting, fine-tuning, and augmenting with tools and APIs. AI aims for optimality. (Current) LMs are approximations, thus induce an LM-to-real gap. Our aim is to bridge such a gap. Previous study shows that grounding, agency and interaction are the cornerstone for sound and solid LMs. Iterative improvements from feedback is critical for further progress of LMs and reinforcement learning is a promising framework, although pre-training then fine-tuning is a popular approach. Iterative updates are too expensive for monolithic large LMs, thus smaller LMs are desirable. A modular architecture is thus preferred. These help make LMs adapt to humans, but not vice verse. We discuss challenges and opportunities, in particular, data & feedback, methodology, evaluation, interpretability, constraints and intelligence.

1 Introduction

Iterative improvements from feedback appears as a universal principle, e.g., gradient descent in optimization, expectation-maximum, boosting, and temporal difference learning in AI, trial and error in animal learning, policy iteration in dynamic programming, close-loop feedback control, (agile) software development, free market for economy, and evolution of our humankind.

Principle: Most, if not all, successful systems make iterative improvements from feedback.

A successful system should be built on ground truth, although it may start with a learned, approximate model or simulator. For an AI system, this includes trustworthy training data and evaluation feedback, and when planning is involved, a reliable world model. For a system with human users, human data and feedback are paramount, and human-in-the-loop is relevant or may be a must. Prominent AI systems like search engines and large language models are built on valuable data, from the Internet and from user feedback. AlphaGo series and games AI have made remarkable achievements, where a perfect game rule, i.e. a model, is a core factor: it can generate high quality or perfect data including game scores. We should not deploy an AI system trained purely from a simulator, especially for high stake systems like healthcare, robotics and autonomous vehicles. We evaluate a system with ground truth for dependable performance results. A system should not self-evaluate itself, e.g., a student should not self-grade the assignment. Section 4 discusses more about approximation.Section 6 discusses more about data and evaluation.

Principle: Ground-truth-in-the-loop.

Language models (LMs), in particular, ChatGPT (OpenAI, 2022a) and GPT-4 (OpenAI, 2023a), have being taken us by storm. Both opportunities and challenges abound for LMs, with vast potential applications, and issues like hallucinations and a lack of planning and controllability, see e.g., OpenAI (2023a) and OpenAI (2023b). In this article, we discuss if and how the principle of iterative improvements from feedback can be applied to LMs.

Mahowald et al. (2023) study LMs' linguistic ("knowledge of rules and patterns of a given language") vs functional ("a host of cognitive abilities required for language understanding and use in the real world") competence and experimental results show impressive yet imperfect linguistic competence, however, at the same time, failures on tests requiring functional competence .

Premise: Current LMs have strong linguistic but weak functional competence.

Then we can leverage LMs' competence as a

good model of language. Moreover, we can manage to improve the functional competence, e.g., factuality, safety, planning and controllability. Prompting is a natural way to utilize LMs, based on the capacity of in-context learning (Brown et al., 2020). Fine-tuning an LM can further improve its expertise. A parameter efficient approach makes fine-tuning large LMs feasible considering the cost. Integrating LMs with tools and APIs can achieve various functionalities.

There are more and more "small" LMs around 10B parameters or less. They are actually very large and in a relative sense. When resources become cheaper, larger models are more affordable. When models become stronger, smaller models may be good enough. Being environment friendly, smaller models are preferred.

Most LMs are trained without optimizing for downstream tasks. Most works utilizing LMs focus on feasibility and correctness. There is a room for further improvements w.r.t. optimality.

Purpose: AI aims for optimality.

To achieve optimality for an AI system, computational bounded rationality leads to approximations (Gershman et al., 2005). Popular methods for general intelligence recently are learning to learn, like transfer / few-shot / multi-task / meta-learning. This boils down to finding a feasible/optimal solution with multiple objectives and/or multiple constraints. Multi-objective optimization usually will not optimize all objectives. The more and the tighter constraints, the less chance to find a feasible solution. Also, with negative transfer, the previous knowledge may interfere with later learning.

Premise: General intelligence is approximation.

A general purpose AI system approximates the underlying world model, i.e., there is a gap between a learned and the real model. We dub this "LM-to-real gap" or LM2real gap or LM to reality gap, following recent study on simulation to reality gap or sim-to-real gap in robotics and RL communities. LMs can represent both Language Models and Large Models, which also include foundation models (Bommasani et al., 2022). Our goal is to bridge such a gap. See Section 4 and 6.5 about approximation and constraints, respectively. Section 6.6 discusses more about intelligence.

Purpose: Bridge LM-to-real gap.

As in Bisk et al. (2020), a language describes the physical world and facilitates the social interactions, and we can't learn language from a radio (Internet), from a television, or by ourselves. Grounding and agency from interactions with the physical and social world are indispensable for LMs.

Premise: Grounding, agency and interaction are indispensable for sound and solid LMs.

Among LMs, GPT-4 is by far regarded as the most capable. However, it is too large to iteratively improve relatively frequently. On the contrary, small LMs are improving, even surpassing GPT-4 in certain tasks, and are conducive to iterative improvements from feedback. To satisfy certain performance thresholds, we may have to limit the number of tasks, so that they can be solved together to attain a feasible solution. This justifies a modular approach: each module handles certain tasks, and all collaborate together. Moreover, issues like privacy and compliance with regulations may favour modularity. Modularity and small LMs will become competent and versatile.

Premise: Modularity and small LMs facilitate iterative improvements from feedback.

Prompt engineering, a popular approach to using LMs, shows how humans have to adapt to AI by deciphering how to use AI, e.g., Zamfirescu-Pereira et al. (2023) study how non-AI experts try and fail to design prompts. This does not align well with one goal of developing AI: AI should adapt to humans. AI should figure out a human's intention and help achieve the goal. Although humans may have to adapt to tools to some extent, LMs are at their early stage, e.g., as a user interface, there is a large room for LMs to improve.

Purpose: AI adapts to humans, not vice versa.

Iterative improvements from feedback can achieve optimality and improve adaptability of AI to humans. Reinforcement learning (RL) (Sutton and Barto, 2018) is a promising framework to learn from feedback and for adaptive control, and thus to advance language models. AlphaGo (Silver et al., 2016) set a landmark in AI by defeating a world champion in Go, using self-play RL to make iterative improvements. It is desirable to harness the achievements in AlphaGo and games AI.

Premise: Reinforcement learning is promising for iterative improvements from feedback.

See Figure 1 for a brief illustration. In the following, we discuss the above principles, premises and purposes in more detail.

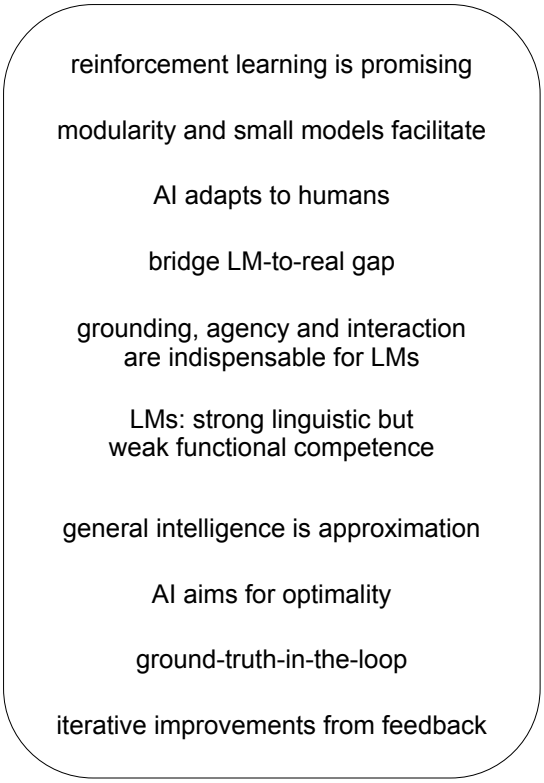

Figure 1: A brief conceptual framework of iterative improvements from feedback for language models: principles, premises and purposes. See text for more detail.

2 Background

2.1 Experience grounds language

As illustrated in Figure 2, Bisk et al. (2020) define five levels of world scopes. Together with texts, perception, embodiment and social interaction contextualize a language with the world.

Grounding is about meaning, understanding, and being appropriate and consistent with the context. Embodiment is about interaction, with action, reward and planning, in a physical world. Social interaction is about communication in human society. Agency, with belief, desire and intention, is about acting to achieve goals. An agent is a learner and decision maker (Sutton and Barto, 2018).

As discussed in Carta et al. (2023), symbol grounding makes actions based on the internal symbol system to be affordable in the environment, direct grounding associates elementary symbols with high-dimensional perceptions, and grounding transfer associates abstract concepts with elementary symbols. Carta et al. (2023) propose functional grounding to manipulate internal symbols to model, predict and control external processes.

In Smith and Gasser (2005), the embodiment also includes social interaction: "The embodiment hypothesis is the idea that intelligence emerges in the interaction of an agent with an environment and as a result of sensorimotor activity." Smith and Gasser (2005) summarizes six lessons from babies for the development of embodied cognition: be multimodal, be incremental, by physical, explore, by social, and use language. Bohg et al. (2017) and Ostrovski et al. (2021) show the importance of active perception, following Held and Hein (1963). Lampinen et al. (2023) show evidence for passive learning of active causal strategies, however, admit that active learning is more beneficial and confounding is challenging for passive learners. See also Roy et al. (2021).

Bisk et al. (2020) prioritize grounding and agency and highlight the importance of physical and social context of language. Computer vision, speech recognition, robotics, simulators and videogames facilitate investigation of language.

2.2 Reinforcement learning

Reinforcement learning is a general framework for sequential decision making with broad applications (Bertsekas, 2019; Littman, 2015; Powell, 2021; Sutton and Barto, 2018; Szepesvári, 2010). An RL agent interacts with the environment over time to learn a policy, by trial and error, that maximizes a long-term, cumulative reward. At each time step, the agent receives an observation, selects an action to be executed in the environment, following a policy, which is the agent's behaviour, i.e., a mapping from an observation to actions. The environment responds with a scalar reward and by transitioning to a new state according to the environment dynamics. Deep RL is at the intersection of deep learning (Bengio et al., 2021; LeCun et al., 2015; Goodfellow et al., 2016; Schmidhuber, 2015) and RL, with deep learning to approximate functions for value, policy, reward, transition, etc.

RL has remarkable achievements like AlphaGo series (Silver et al., 2016, 2017, 2018), ChatGPT, contextual bandits (Li et al., 2010), Decision Service (Agarwal et al., 2016), ReAgent (Gauci et al., 2019), ride-hailing order dispatching (Qin et al., 2020), AlphaStar (Vinyals et al., 2019) for StarCraft II, DeepStack (Moravčík et al., 2017) and Libratus (Brown and Sandholm, 2017) for Texas Hold'em Poker, Cicero (Bakhtin et al., 2022) for Diplomacy, Gran Turismo Sophy (Wurman et al.,

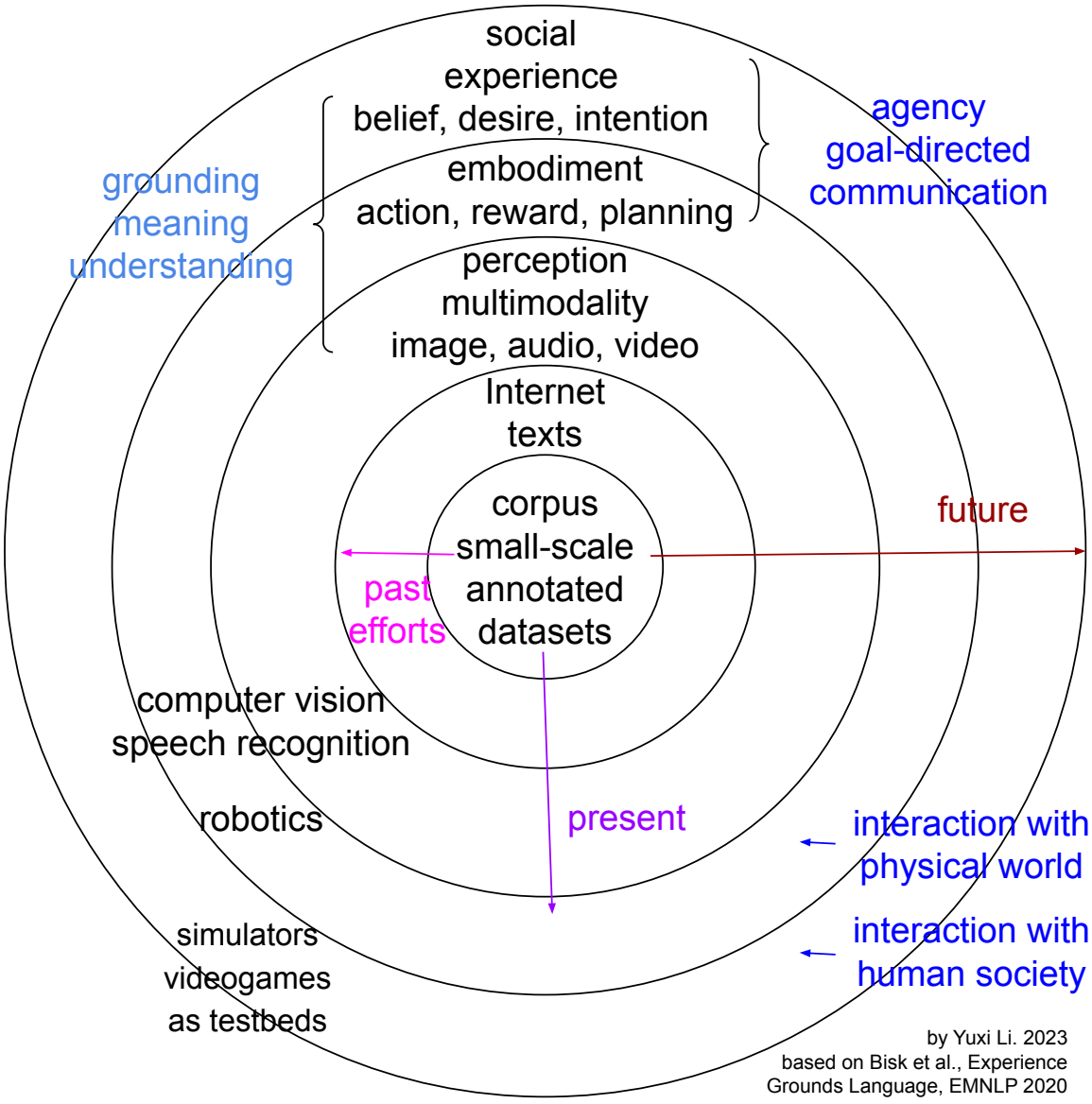

Figure 2: Five levels of World Scopes based on Bisk et al. (2020): small scale Corpus (our past) for corpora and representations, large scale Internet (most of current NLP) for the written world, Perception (multimodal NLP) for the world of sights and sounds, Embodiment and action for interaction with physical world, and Social world for interaction with human society.

2022), Degrave et al. (2022) for magnetic control of tokamak plasmas, Bellemare et al. (2020) for navigating stratospheric balloons, AlphaTensor (Fawzi et al., 2022) for matrix multiplication, and AlphaDev (Mankowitz et al., 2023) for sorting.

See Li (2017) for an overview about deep RL and Li (2022) for discussion about RL in practice.

2.3 Auto-regressive language model

A language model is about the probability distribution of a sequence of tokens. In ChatGPT and many LMs, an auto-regressive language model is the probability distribution of next token, given previous tokens, i.e., the conditional probability distribution:

$$probability(\text{next token}|\text{previous tokens}).$$

We focus on such LM here. It can be regarded as a policy, where the "previous tokens" are the state (observation) and the "next token" is the action.

Many problems in natural language processing (NLP) are sequential decision making problems, thus RL is a natural framework. See e.g., Gao et al. (2019); Li (2017).

2.4 Large LMs

GPT stands for Generative Pre-trained Transformer (Radford et al., 2018, 2019; Brown et al., 2020; OpenAI, 2022a, 2023a). Transformers (Vaswani et al., 2017) are the backbone of LMs, featuring self-attention, conducive to long range dependancy and large scale implementation. GPT series and many LM variants are based on deep learning, in particular, self-attention Transformers, self-supervised learning (Balestriero et al., 2023), and pre-training models.

There are many large LMs, e.g., GPT-3, ChatGPT, GPT-4, BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2023f), T5 (Raffel et al., 2020), LaMDA (Thoppilan et al., 2022), PaLM (Chowdhery et al., 2022), Sparrow (Glaese et al., 2022), Claude (Bai et al., 2022a), Chinchilla (Hoffmann et al., 2022) Megatron-Turing NLG (Smith et al., 2022), Gopher (Rae et al., 2022), BLOOM (BigScience Workshop et al., 2023), LLaMA (Touvron et al., 2023). The parameter sizes are huge, e.g., 175 billion for GPT-3.

2.5 Specialized LMs

There are many specialized models, e.g., AlphaFold (Tunyasuvunakool et al., 2021), Codex (Chen et al., 2023b), AlphaCode (Li et al., 2022), WebGPT (Nakano et al., 2022), Robotics Transformer (RT-1) (Brohan et al., 2022), BiomedGPT (Zhang et al., 2023a), Clinical Camel (Toma et al., 2023), BloombergGPT (Wu et al., 2023b), FinGPT (Yang et al., 2023a), Med-PaLM 2 (Singhal et al., 2023), MusicLM (Agostinelli et al., 2023), AudioGPT (Huang et al., 2023).

2.6 "Small" LMs

Following LLaMA (Touvron et al., 2023), many "small" LMs appear, with around 10B or smaller, e.g., Alpaca (Taori et al., 2023), Dolly (Conover et al., 2023), Koala (Geng et al., 2023), Vicuna/StableVicuna (Chiang et al., 2023), ChatGLM (Du et al., 2020; Zeng et al., 2023), StableLM (Stability AI, 2023), Guanaco (Dettmers et al., 2023), Pythia (Biderman et al., 2023), GPT4All[1], Open-Assistant[2], ColossalChat (You, 2023). "Small" is a relative concept: as software and hardware improve, the current large models may become small. See Kim (2023b) for a list of open sourced fine-tuned LMs.

There are also specilized small LMs, e.g., Gorilla (Patil et al., 2023), a LLaMA-7B-based model, surpassing GPT-4 w.r.t. API calls, TinyStories (Eldan and Li, 2023) for fluent and consistent stories with <10M parameters, and phi-1 (Gunasekar et al., 2023) for good coding performance with 1.3B parameters and 7B tokens.

2.7 Modularity

Mahowald et al. (2023) propose a modular architecture with a language component, a problem solver, a grounded experiencer, a situation modeler, a reasoner, and a goal setter. Laird et al. (2017) discuss the Soar cognitive architecture, including perception, motor, representation, working memory, (procedural, semantic, and episodic) long-term memories, reinforcement learning, semantic learning, episodic learning, and decision procedure.

Modularity may enhance adaptability, compositionality, efficiency, scalability, consistency, robustness, and interpretability and mitigate catastrophic forgetting (Pfeiffer et al., 2023).

Karpas et al. (2022) propose Modular Reasoning, Knowledge and Language (MRKL), a modular, neuro-symbolic architecture to combine LMs, external knowledge sources and discrete reasoning.

[1] https://github.com/nomic-ai/gpt4all
[2] https://github.com/LAION-AI/Open-Assistant

Modularity is related to hierarchical learning and planning (Russell and Norvig, 2020), in particular, hierarchical RL (Sutton et al., 1999).

2.8 Discussions & debates about LMs

There are all sorts of discussions & debates, e.g., the dangers of stochastic parrots (Bender et al., 2021), limitation of neural networks (Delétang et al., 2023), limitation of autoregressive models (Lin et al., 2021), lack of causality (Jin et al., 2023), lack of compositionality (Dziri et al., 2023), lack of recursion (Zhang et al., 2023b), limitations (Deshpande et al., 2023; McKenzie et al., 2023) of scaling laws (Kaplan et al., 2020; Hoffmann et al., 2022), model collapse (Shumailov et al., 2023), artificial general intelligence (AGI) (Allyn-Feuer and Sanders, 2023; Bubeck et al., 2023; Marcus, 2023), evaluation of AI (Burnell et al., 2023), social norms (Browning and LeCun, 2023), distortion of human beliefs (Kidd and Birhane, 2023), risks and benefits (Goldman, 2023), existential risk (Bengio, 2023), court hearing due to hallucination (Novak, 2023), risk of further concentration of wealth (Chiang, 2023), eight things to know (Bowman, 2023). See more discussions about AI alignment with human value, e.g., Russell (2019); Mitchell (2020); Christian (2021). See surveys, e.g. LMs in practice (Yang et al., 2023b)

3 How to improve LMs?

Besides pre-training and fine-tuning, we can classify methods to improve LMs as follows: a) format/content of prompts: a.1) vanilla text, a.2) multimodality, a.3) augmentation with tools, a.4) integration of advanced techniques like search, learning and coding; b) feedback, b.1) open-loop, no feedback during inference, b.2) close-loop, entirely/mainly from LMs, b.3) close-loop, from environment (games, code interpreter, robotics, etc.) including LMs; c) fixed LMs vs iterative improvements of LMs.

See Table 1 for an illustration of the taxonomy with example methods. Admittedly, methods and models in all the tables are not comprehensive.

3.1 Pre-training

A common approach is pre-training then fine-tuning LMs. In the pre-training stage, LMs, and foundation models (Bommasani et al., 2022) in general, are trained on broad data, usually with self-supervised learning (Balestriero et al., 2023) at scale, being widely adaptive to downstream tasks. Table 2 compare several pre-training models.

Radford et al. (2018) introduce generative pre-training for LMs, which could be regarded as "GPT-1". Radford et al. (2019) introduce GPT-2, an unsupervised multitask learning LM. Brown et al. (2020) introduce GPT-3, a few-shot learning LM, popularizing the concept of in-context learning (Dong et al., 2023). OpenAI (2022a) introduces ChatGPT and OpenAI (2023a) introduces GPT-4.

Devlin et al. (2019) introduce Bidirectional Encoder Representations from Transformers (BERT). See a survey about BERT Rogers et al. (2020). Raffel et al. (2020) introduce Text-to-Text Transfer Transformer (T5).

There are foundation models for control/RL. Gato (Reed et al., 2022) is a generalist policy for multi-task, multi-modality, and multi-embodiments. Adaptive Agent (AdA) (Adaptive Agent Team et al., 2023) is an RL foundation model adaptive to a vast and diverse task space at human timescale. Sun et al. (2023b) propose self-supervised multi-task pre-training with control transformer (SMART).

3.2 Fine-tuning

Fine-tuning further improve LMs. Table 3 compare several fine-tuning models.

Large LMs like GPT-3 have a huge number of parameters so that it is prohibitively expensive to make a full refinement. Fortunately, parameter efficient fine-tuning (PEFT) methods, like Low-Rank Adaptation (LoRA) (Hu et al., 2021) and quantized LoRA (QLoRA) (Dettmers et al., 2023) have shown that it is possible to fine-tune a small number of parameters while achieving comparable performance. Liu et al. (2022) show that PEFT outperforms in-context learning. See more studies about PEFT, e.g., Mao et al. (2022), He et al. (2023), Ding et al. (2023), Chen et al. (2023a).

RL from human feedback (RLHF) is adopted by many LMs. Christiano et al. (2017) propose RLHF, i.e., by defining a reward function with preferences between pairs of trajectory segments, to tackle the problems without well-defined goals and without experts' demonstrations, and to help improve the alignment between human value and the objective of RL system. Ouyang et al. (2022) propose to fine-tune GPT-3 with human feedback, in particular, with RL, to follow instructions for better alignment with human value. ChatGPT, after

methods to improve language models		examples
pre-training		BERT, T5, GPT-3, GATO, AdA, SMART
fine-tuning		SFT, RLHF, RLAIF, PEFT (LoRA, QLoRA, etc.), LIMA, PEBBLE, CoH, rewarded soup, DPO
prompting	vanilla text	CoT, prefix tuning, prompt tuning, Least-To-Most, TEMPERA, RLPrompt, ReAct, Cicero, GLAM
	multi-modality	Gato, SayCan, Inner Monologue, Code as Policies, Voyager, RoboCat, Visual ChatGPT, Chameleon
	augmentation with tools	Toolsformer, HuggingGPT, DSP, PAL, LLM+P, Code as Policies, ReProver, Visual ChatGPT, Chameleon
	with advanced techniques: search, learning, coding	ToT, RAP, AdaPlanner, PG-TD, Voyager, ReProver
feedback	open-loop, no feedback	CoT, prefix tuning, prompt tuning, HuggingGPT
	close-loop, feedback from LMs and environment	ReAct, ToT, RAP, DSP, DESP, AdaPlanner, SayCan, Inner Monologue, Code as Policies, Voyager, Cicero, RoboCat, CodeRL, ILQL, GLAM, ReProver
language models	fixed	most methods except those below
	iterative improvements	Cicero, RoboCat, CodeRL, ILQL, GLAM, ReProver

Table 1: Methods to improve language models. See Sections 3 for more details including abbreviations.

model	training method	purpose
BERT	SL & SSL	masked token prediction next sentence prediction
T5	SL & SSL	text-to-text
GPT-3	SSL	next token prediction
GATO	SL	control
SMART	SSL	control
AdA	RL	control

Table 2: Pre-training models. Feedback during training, but not during inference. SL: supervised learning. SSL: self-supervised learning. RL: reinforcement learning.

pre-training, conducts 1) supervised fine-tuning (SFT), 2) reward model learning, 3) reinforcement learning (OpenAI, 2022a), where the last two steps constitute RLHF.

Instruction following or instruction tuning includes supervised fine-tuning and RLHF, both of them are imitation learning; see Section 6.1.

Christiano et al. (2017), Ouyang et al. (2022) and many RLHF papers use Proximal Policy Optimization (PPO) (Schulman et al., 2017) to optimize a policy. Ramamurthy et al. (2023) propose Natural Language Policy Optimization (NLPO). Zhu et al. (2023) propose Advantage-Induced Policy Alignment (APA).

RLHF plays a critical role in human alignment and facilitates learning of the objective function. ChatGPT collects human data. Bai et al. (2022b) propose Constitutional AI with rules or principles and RL from AI Feedback (RLAIF) with supervised learning and RL to reduce the reliance on human involvements in learning an LM. Glaese et al. (2022) also design rules in Sparrow.

Lee et al. (2021) propose unsupervised pre-training and preference-based learning via relabeling experience (PEBBLE) to improve the efficiency of human-in-the-loop feedbacks with binary labels, i.e. preferences, provided by a supervisor. Liu et al. (2023b) propose Chain of Hindsight (CoH) to convert all feedback into sentences. Wu et al. (2023c) propose fine-grained RLHF to learn from and multiple reward models, each of which associates with a specific error category with dense signals at segment level. Human Feedback Gives Better Rewards for Language Model Training Rame et al. (2023) propose rewarded soups to handle the heterogeneity of diverse rewards by interpolation of multiple strategies to achieve Pareto-optimal alignment.

Rafailov et al. (2023) propose Direct Preference Optimization (DPO) without reward modelling or RL. However, DPO applies only to the Bradley-Terry model underling current RLHF to estimate score functions from pairwise preferences. There may be other ways to handle human preference and non-preference ways to handle value alignment, e.g., Knox and Stone (2008) uses ratings to transmit

method	training method
SFT	supervised learning
PEFT	supervised learning
LIMA	supervised learning
RLHF	reward modelling & RL
RLAIF	reward modelling & RL
PEBBLE	reward modelling & RL
CoH	reward modelling & RL
fine-grained RLHF	reward modelling & RL
rewarded soup	reward modelling & RL
DPO	classification

Table 3: Fine-tuning methods. Feedback during training, but not during inference. SFT: supervised fine-tuning. PEFT: parameter efficient fine-tuning, e.g., LoRA.

human knowledge to an RL agent.

LIMA (Zhou et al., 2023a) shows the importance of a high-quality pre-training model and carefully curated instruction data.

Zhang et al. (2021) survey human guidance for sequential decision-making. Wirth et al. (2017) present a survey of preference-based RL methods. Lambert et al. (2022) is a blog about RLHF. RL from human feedback goes back at least to Knox and Stone (2008).

Goldberg (2023) discusses that a "traditional" language model is trained with natural text data alone, while ChatGPT is not traditional any more: it is augmented with instruction tuning, programming language code data, and RLHF.

3.3 Prompting

Prompts serve as the user interface for LMs. In this sense, most methods are about improving prompts, in particular, those with fixed LMs.

Here we discuss open-loop methods, which do not benefit from feedback, e.g., Chain-of-Thought (CoT) (Wei et al., 2022), Least-to-Most prompting (Zhou et al., 2023b), Zero-Shot Planners (Huang et al., 2022a) and Chameleon (Lu et al., 2023a). Human users may improve their prompt engineering skills after seeing outputs of prompts. However, such methods do not improve themselves based on such feedback.

There are works for prompt optimization / automation, e.g., prefix-tuning (Li and Liang, 2021), prompt tuning (Lester et al., 2021), symbol tuning (Wei et al., 2023), RLPrompt (Deng et al., 2022), TEMPERA (Zhang et al., 2023d), PromptPG (Lu et al., 2023b). Such methods improve prompts in an "offline" manner, i.e., not interactively while using prompts. Note, prefix-tuning and prompt tuning are classified as parameter efficient fine-tuning, in e.g., Ding et al. (2023), He et al. (2023), Ruder et al. (2022). See a survey (Liu et al., 2023e).

method	training method
Chain of Thought	no training
Chameleon	no training
Least-to-Most	no training
prefix tuning	supervised
prompt tuning	supervised
TEMPERA	RL
RLPrompt	RL

Table 4: Prompt improvement methods. Note: Prompts are a user interface for LMs, so that many methods improving LMs are about improving prompts. This table lists open-loop methods: 1) no training and 2) feedback during training, but not during inference.

The drawbacks of prompting are inefficiency, poor performance, sensitivity to prompt, and lack of clarity (Ruder et al., 2022). It also lacks adaptability to users, so that uses have to figure out how to use LMs with prompt engineering. Users' heavy reliance on prompt engineering implies that LMs are not good enough; otherwise, an LM can adapt to a user and guide a user how to use the LM.

To extract the capacity of LMs, prompts integrate sophisticated methods like 1) search, e.g., Tree of Thought, 2) coding, e.g., Code as Policies and PAL, and 3) planning with code, e.g., AdaPlanner and Voyager, as in next section.

3.4 Close-loop feedback with self-reference to fixed LMs

Close-loop methods benefit from feedback and make improvements. Feedback may come from external sources like games, code interpreter and robotics. In the following, we discuss recent work with close-loop feedback with fixed LMs. Table 5 presents a brief comparison.

Multimodality and embodiments may integrate with visual Transformers and pre-training for perception, like Contrastive Language-Image Pre-training (CLIP) (Radford et al., 2021), diffusion model (Rombach et al., 2022), or ControlNet (Zhang and Agrawala, 2023). PaLM-E (Driess et al., 2023) is an embodied multimodal LM.

Liang et al. (2023b) propose High-Modality Mul-

method	feedback	evaluator	refinement
ReAct	partial results & external sources	LM	action
ToT / RAP	from a tree built with an LM	LM	whole plan
AdaPlanner	results of sub-goals	LM	whole plan
SELF-REFINE	feedback for prompt	LM	prompt
RCI	output of prompt	LM	prompt
ProgPrompt	output of code (prompt)	LM	prompt
DEPS	output of prompt	LM	whole plan
Reflexion	evaluation function or LM	LM	
Voyager	environment feedback, execution errors, self-verification of tasks	MineCraft & LM	whole plan
Plan4MC	LM helps RL with high level plan	Minecraft	skill, plan
LATM	unit tests	LM	tool (code)
PG-TD	quality of partial solution	MCTS with LM	code gen.
SayCan	skill success	LM & RL	whole plan
Code as Policies	environment	LM	code/prompt
Inner Monologue	success detection, scene description, and human interaction	environment	whole plan

Table 5: Close-loop feedback with self-reference to fixed LMs.

timodal Transformer (HighMMT) to handle 10 modalities: text, image, audio, video, sensors, proprioception, speech, time-series, sets and tables. Here we treat multi-modality basically as multi-media, like image, audio and video.

General methods

ReAct (Yao et al., 2023b) integrates dynamic reasoning with high-level plans for task-specific actions, with feedback from LM and external sources.

Tree of Thoughts (ToT) (Yao et al., 2023a) builds a tree and an evaluation function with an LM to explore multiple different multi-step scenarios, look ahead and backtrack with search algorithms like breadth first search (BFS) and depth first search (DFS). Reasoning via Planning (RAP) (Hao et al., 2023) follow a similar vein and study planning methods like Monto Carlo Tree Search (MCTS).

AdaPlanner (Sun et al., 2023a) is a planning method with an LM for both planning and refining, together with a skill memory.

Wong et al. (2023) propose a probabilistic language of thought (PLoT) for rational meaning construction to integrate neural models of language with probabilistic models for rational inference.

See more work, e.g., Self-Refine (Madaan et al., 2023), RCI (Kim et al., 2023), Reflexion (Shinn et al., 2023), DEPS (Wang et al., 2023d). See also Auto-GPT [3] and BabyAGI [4].

Games

Games to AI is like fruit flies to genetics. Games AI, in particularly with RL, is promising to push LMs and AI further.

Fan et al. (2022) propose MINEDOJO, an open-ended task suite based on Minecraft game with Internet-scale domain knowledge, together with an agent learning algorithm with large pre-trained models. Voyager (Wang et al., 2023a) explores Minecraft continuously with the modules of automatic curriculum, skill library and iterative prompting mechanism. Plan4MC (Yuan et al., 2023) improves skill learning and planning for Minecraft tasks with assistance from LM.

See Cicero (Bakhtin et al., 2022) in next section. See also Generative Agents (Park et al., 2023) and CAMEL (Li et al., 2023a).

Programming language

Programming language is more formal and thus relatively easier than natural language. Moreover, we may utilize a program interpreter to help judge the correctness and quality of generated codes.

Zhang et al. (2023c) propose Planning-Guided Transformer Decoding (PG-TD) to integrate a

[3] https://github.com/Significant-Gravitas/Auto-GPT

[4] https://github.com/yoheinakajima/babyagi

planning algorithm like Monte Carlo tree search (MCTS) and the Transformer of an LM to improve the correctness of generated code.

Chen et al. (2023c) propose Self-Debugging to teach an LM to debug generated code via few-shot prompting by identifying mistakes from explanations in natural language, without feedback for code correctness or error messages.

Cai et al. (2023) propose LATM to create reusable tools (Python utility functions) with LMs for both tool making and tool using.

Yang et al. (2023c) propose InterCode, an RL environment for interactive code generation, where observations are execution feedback, actions are code and rewards are either a binary completion score or more complex criteria defined by users.

See early discussion for AdaPlanner (Sun et al., 2023a) and PLoT (Wong et al., 2023) and later for Code as Policies (Liang et al., 2023a) in this section. See Section 6 for CodeRL (Le et al., 2022) and Haluptzok et al. (2023).

Robotics

Robotics come with multi-modality, embodiments and interaction, with perception and action.

There are a series of efforts for robotics: Robotics Transformer (RT-1) (Brohan et al., 2022) for real-world robotics control at scale, Inner Monologue (Huang et al., 2022b) for chaining together perception models, robotic skills, and human feedback for processing and planning in robotic control, SayCan (Ahn et al., 2022) for grounding LM in robotic affordances, ROSIE (Yu et al., 2023) for scaling robot learning with semantically imagined experience, Code as Policies (Liang et al., 2023a) for leveraging LMs to generate policy code for embodied control, and ProgPrompt (Singh et al., 2023) for generating plans with LMs. See RoboCat (Bousmalis et al., 2023) in next section.

3.5 Close-loop feedback with iterative improvements of LMs

As discussed in the last section, most methods with a close-loop feedback share a common feature of self-reference to fixed LMs, which raises the concern of how to handle mistakes by LMs. Incorporating multimodality, embodiment and interaction information can help improve grounding and mitigate the issue. Ideally, we can make iterative improvements of LMs from feedback.

Most existing methods focus on feasibility and correctness, rather than optimality. There are emerging works to take one step further by refining LMs with iterative improvements from feedback. See Table 6 for a brief comparison.

Cicero (Bakhtin et al., 2022) integrates an LM with planning and RL algorithms in the seven-player game of Diplomacy to infer players' beliefs and intentions from conversations and to generate dialogues for negotiation and tactical coordination.

CodeRL Le et al. (2022) follows an actor-critic RL approach during training, with an actor LM for code generation and a critic network for error prediction of generated code as feedback to the actor. During inference, CodeRL leverages unit tests to further improve code generation.

Haluptzok et al. (2023) propose to synthesize programming puzzles and solutions verified by execution and improve code generation by self-play.

Yang et al. (2023d) propose LeanDojo: an open-source Lean[5] playground with toolkits, data, models, and benchmarks for theorem proving, together with ReProver, a retrieval-augmented prover LM based on a T5-like encoder-decoder Transformer.

Snell et al. (2023) propose implicit language Q-learning (ILQL) to fine-tune an LM to maximize user-specified utility functions.

Carta et al. (2023) propose GLAM to improve functional grounding in interactive environments with RL using an LM as a policy.

RoboCat (Bousmalis et al., 2023) is a visual goal-conditioned foundation model for robotic manipulation, with zero-shot and few-shot generalization, based on Gato (Reed et al., 2022).

Levine (2023b) discusses the purpose of an LM beyond predicting next token and how RL can help fulfill it. Levine (2023a) discusses when data and optimization collaborate, we can solve problems in new ways and in real world outside of simulators.

Yang et al. (2023e) show that foundation models are helpful for all components in decision making: states, actions, rewards, transition dynamics, agents, environments, and applications, with generative modeling or representation learning, thus they will benefit mutually from each other.

RLHF is one application of RL for LMs to handle human value alignment. The discussion above shows that RL may advance LMs in many ways.

3.6 Augmented LMs with tools

A natural way to harnesses the language competence of LMs is by utilizing tools like a search

[5] https://leanprover.github.io

method	feedback	evaluator
Cicero	game play	game engine, agent
CodeRL	error prediction	critic model
ReProver	environment	Lean
ILQL	environment	LM
GLAM	environment	BabyAI_Text
RoboCat	demonstration & generated data	model

Table 6: Close-loop feedback with iterative improvements of (language) models.

method	tool	tool language
Toolformer	general	text
HuggingGPT	general	text
DSP	retrieval model	code
PAL	code interpreter	code
Code as Policies	code interpreter	code
LLM+P	planners	PDDL
ReProver	math library	code

Table 7: Augmented LMs with tools.

engine, a vector database, a code interpreter, or a symbolic AI solver to handle tasks. A common approach is: 1) converts the natural language description of the problem into the language by the tool, 2) the tool solves the problem, and 3) translates the solution back into text. Table 7 shows a brief comparison.

A method needs to answer questions like which tools/APIs to call, when to call, with what arguments, and how to translate the results back into LMs. Toolformer (Schick et al., 2023) and Ge et al. (2023) follows a self-supervised and an RL approach, respectively. HuggingGPT (Shen et al., 2023) relies on an LM.

Demonstrate-Search-Predict (DSP) expresses high-level programs for demonstrations aware of the LM and the retrieval model, relevant passages searches and grounded predictions generation for the LM and the retrieval model to process more reliably (Khattab et al., 2023), and shows task-aware are favourable to task-agnostic strategies.

Program-Aided Language models (PAL) (Gao et al., 2022) converts a relevant piece of text to code and uses a runtime like a Python interpreter to solve the problem.

LLM+P (Liu et al., 2023a) converts a language description of a planning problem into the planning domain definition language (PDDL), solves it with classical planners, and translates the solution back into text.

See also LangChain[6], Visual ChatGPT (Wu et al., 2023a), TaskMatrix.AI (Liang et al., 2023c), RCI (Kim et al., 2023), etc.

Domain expertise is still required, see e.g., ChemCrow (Bran et al., 2023). See Mialon et al. (2023) for a survey about augmented LMs.

[6] https://langchain.com

APIBank (Li et al., 2023c) is a benchmark for augmented LMs with tools.

HuggingGPT follows an open-loop without learning from feedback. Toolformer, LLM+P and PAL have feedback during training, but not during inference. DSP and Code as Policies incorporate feedback for improvements, with fixed LMs. ReProver incorporates feedback to improve the LM.

4 LMs are approximations

A model specifies how an agent interacts with an environment. A model refers to the transition probability and the reward function, mapping states and actions to distributions over next states and expected rewards, respectively. The agency requires both state and reward prediction, so do LMs. Andreas (2022) admits that, besides predicting text, an agent is what we want for human language technologies, with beliefs and goal achieving.

A model may be built with prior knowledge, from a dataset by estimating parameters, and/or by a generative approach. A simulator may be built based on a model explicitly, e.g., from game rules like the Arcade Learning Environment for Atari games (Bellemare et al., 2013; Machado et al., 2018) or computer Go, chess and shogi (Silver et al., 2018), and physics like Mujoco (Todorov et al., 2012), or implicitly, e.g. those with generative models (Ho and Ermon, 2016; Chen et al., 2019). Planning works with a model or a simulator.

Andreas (2022) emphasizes that current LMs are approximations. Moreover, degrees of approximations should vary for different tasks. It is desirable to characterize such approximation errors.

There are many concrete examples. Kocoń et al. (2023) shows that ChatGPT is Jack of all trades, master of none. Valmeekam et al. (2023) shows only 3% success rate of executable plans gener-

ated by GPT-3. Li et al. (2023b) shows that OthelloGPT struggles with generating legal moves for the game Othello. Yao et al. (2023a) propose Tree of Thought, and experiments show that GPT-4 can not fully solve the Game of 24.

Errors occur naturally from an approximate model. Compounding errors are particularly serious for sequential decision makings, like a sequence of tokens. When applying an LM in practice, we need to handle errors. One question for practitioners is if it is always feasible to fix an error from a strong LM.

4.1 AlphaGo vs ChatGPT

Next we discuss if LMs may borrow ideas from AlphaGo series, which set a landmark in AI by tackling a very hard problem pursued by many researchers for decades.

The lessons from AlphaGo series follow. 1) With a game rule, there is a perfect model, which can generate infinite high quality data, esp., reliable feedback. 2) This supports iterative improvements of the policy, with trial and error, using general policy iteration, by self play, to achieve a strong computer program. 3) Imitation learning is not enough: In and before AlphaGo, studies use expert games for training. However, self play RL achieves super-human performance in Go, chess, and shogi from scratch, without human knowledge, and also in many other games.

Moreover, Levine (2023a) illustrates that RL can stitch parts of policies to attain a better policy. Levine (2023b) shows that in a tech support application, RL can learn from several specialists for different aspects to improve the job.

For LMs, there is no perfect rule for most problems, neither perfect feedback. Games and code generation appear as exceptions to some extent, with reliable feedback from a game engine and a code interpreter, respectively. The approach in ChatGPT can be treated as imitation learning.

4.2 Bridge LM-to-real gap

An LM like GPT-4 is used as a simulator of the underlaying model in many cases like SELF-INSTRUCT (Wang et al., 2023b) and React (Yao et al., 2023b). However, a simulator usually can not precisely reflect the reality. Also, from the discussion above, LMs are approximations. Then there is a language model to reality gap, or LM-to-real / LM2real gap for short. How to bridge such a gap is critical and challenging.

In applications with physical systems like robotics and autonomous driving, where it is much easier to train an agent in simulation than in reality, simulation to reality gap, or sim-to-real, or sim2real, or reality gap, attract much attention recently. Some LM applications may tolerate more errors, like a writing aid; however, some may be high-stake and/or involve physical systems, e.g., healthcare like Med-PaLM 2 (Singhal et al., 2023) and robotics like SayCan (Ahn et al., 2022).

LMs may borrow ideas from similar study in robotics to reduce the LM-to-real gap. Here is a brief discussion. Chebotar et al. (2019) study how to adapt simulation randomization with real world experience. James et al. (2019) propose to adapt from randomized to canonical scenes, without real-world data. James et al. (2020) propose RLBench, a robot learning benchmark and environment. Deitke et al. (2020) propose RoboTHOR, an open sim-to-real embodied AI platform. Gondal et al. (2019) propose a distanglement dataset to study the sim-to-real transfer of inductive bias. Hanna et al. (2021) study sim-to-real RL with grounded action transformation. Kadian et al. (2020) develop a library Habitat-PyRobot Bridge (HaPy) to execute identical code in simulation and on real robots seamlessly, and investigate sim2real predictivity with a new performance metric Sim-vs-Real Correlation Coefficient. Zhao et al. (2020) present a brief survey on sim-to-real in deep RL for robotics. Lavin et al. (2021) discuss simulation intelligence.

5 Iterative improvements from feedback

Iterative improvements from feedback provides a general approach to achieving optimality and improving adaptability of LMs. It makes the foundation sound and solid, by improving the world model, improving grounding for better understanding and better consistency with the world, and improving agency for goal achieving.

Figure 3 illustrates the framework of iterative improvements from feedback with modularity in a general sense. Modules 0-N send decisions to and receive feedback from other modules, tools, APIs, and the task. Module 0 serves as a coordinator. Modules can interact and learn from each other. Tools and APIs can be regarded as fixed modules; i.e., they do not learn. One or more of Modules 1-N may be large LMs, and keep fixed, i.e., not learn from feedback. From RL's perspective, a learning module is an agent, and the rest is its environment.

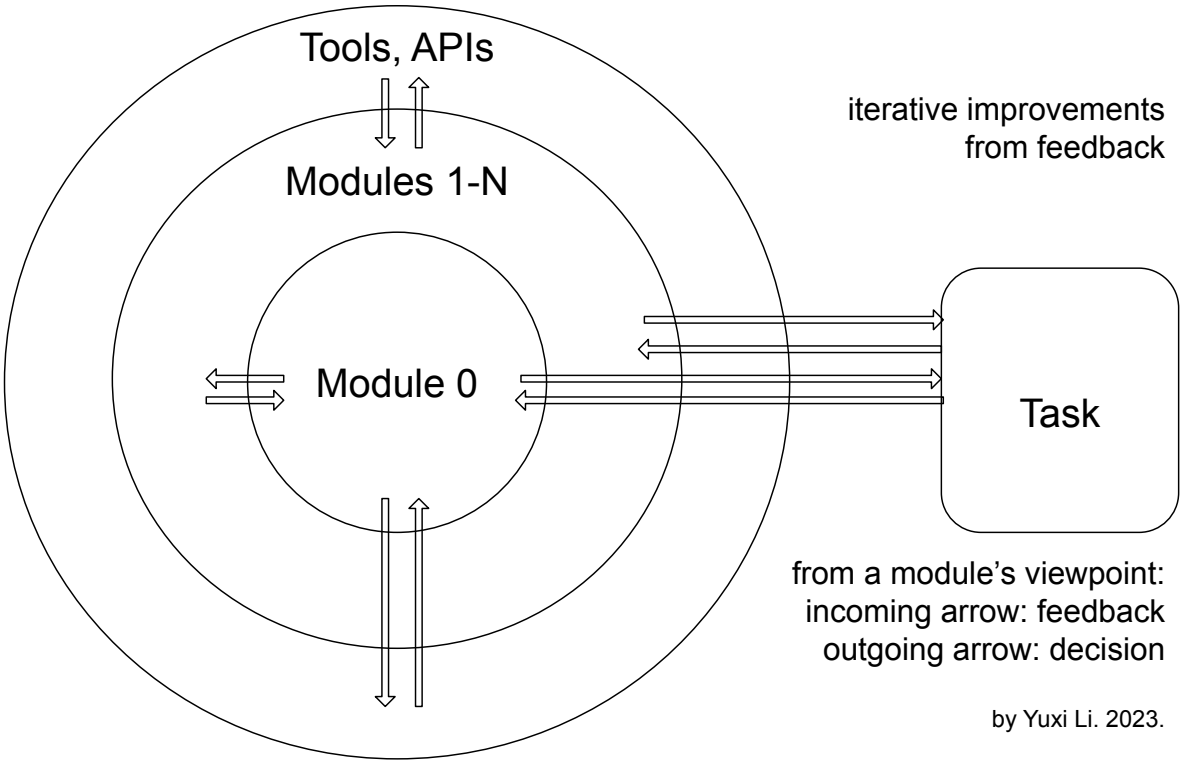

Figure 3: Iterative improvements from feedback

Such a framework is general. We focus on LMs here. The prohibitive cost of updating large LMs will not allow for relatively frequent iterative improvements from feedback. As a result, next generation LM systems would follow a modular architecture, potentially with many small LMs, rather than a monolithic, general-purpose large LM.

There will be many modules with specific expertise, for horizontal functionalities and vertical applications. In particular, one module serves as the interface to users and coordinates all LMs, tools, and APIs. Another module builds the world model from interactions with the world. Moreover, one or more module are dedicated to safety and ethics.

We may deploy one or a couple of large LMs to harness the language competence and functionalities large LMs outperform small LMs significantly. Large LMs are kept fixed, to avoid prohibitive updating costs and/or due to their being proprietary, until when new versions are available. Small LMs can improve iteratively from feedback, from users, other LMs, tools, APIs, and all the rest of the world, using reinforcement learning and AI algorithms. Small LMs are preferred to large LMs, esp. when small LMs are good enough.

Modularity with small models is promising for further progress in AI. Augmenting LMs with tools, APIs and plugins is an evidence for modularity. It also becomes feasible for players without huge resources, thus more amenable for AI academics (Togelius and Yannakakis, 2023).

5.1 Connection with existing methods

Pre-training then fine-tuning is common for LMs. As discussed in Section 3, most approaches improve prompts with fixed LMs, like GPT-4.

ChatGPT follows iterative deployment (OpenAI, 2022a,b), making updates after collecting a big batch of users' feedback. This may be the edge GPT-4 over other large LMs. This is helpful for making improvements. It is reasonable since a huge LM is too costly to update often. However, it is desirable to make more frequent refinements.

It is feasible to fine-tune a small LM frequently, or even to incorporate feedback when building it.

Prompting provides the initial condition of a language model, or the starting "previous tokens" in ChatGPT. It is desirable to leverage mature tools. Prompting and augmentation with tools depend on the capability of an LM, need to handle errors, and are not able to improve an LM.

Consider LM as a policy, supervised fine-tuning,

parameter efficient fine-tuning and reinforcement learning from human feedback follow the approach of imitation learning. As discussed in Section 4.1, imitation learning is not enough. RL is thus promising for LMs.

5.2 Experience and model

Sutton (2022a) talks about the increasing role of sensorimotor experience in AI to be more grounded, learnable and scalable. Sensorimotor experience is the sensations and actions of an agent's ordinary interaction with the world.

Sutton (2022b) proposes the common model of intelligent agent, which is model-based RL. It integrates experience with model, aligning with Pearl (2020). It reconciles nature with nurture, empiricism with rationalism, and connectionism with symbolism.The common model is thus promising for achieving sound and solid grounding and agency. The common mode is conceptually similar to the autonomous machine intelligence architecture (LeCun, 2022), which includes configurator, perception, world model, actor, critic, intrinsic cost, and short term memory. Mialon et al. (2023) discuss a modular implementation, which applies similarly to the common model. See Section 6.6 for more discussion about general intelligence.

Building on trial and error with experience, close-loop optimal control with dynamic programming, and temporal difference learning (Sutton and Barto, 2018), reinforcement learning naturally implements iterative improvements from feedback.

6 Challenges and opportunities

We discuss challenges and opportunities to implement iterative improvements from feedback, in particular, data, feedback, methodology, evaluation, interpretability, constraints and intelligence.

6.1 Data and feedback

The importance of training data are second to none for big data problems, like many with deep learning, specifically, LMs. It is critical to follow the principle pf ground-truth-in-the-loop. Recent success of LMs, among many machine learning applications, provides evidence.

Olausson et al. (2023) show the importance of feedback from tests and human programmers for repairing code generated by LMs. phi-1 (Gunasekar et al., 2023) shows the importance of high-quality textbook-like data. LIMA (Zhou et al., 2023a) shows the importance of high-quality of instruction data.

Gudibande et al. (2023) show the issue of imitating large LMs. Shumailov et al. (2023) discuss the issue of model collapse due to training with generated data from LMs as well as Gaussian Mixture Models (GMMs) and Variational Autoencoders (VAE), and show the importance of genuine human data for LMs. It explicitly indicates issues with methods generating data from LMs like SELF-INSTRUCT (Wang et al., 2023b). It also casts doubts on many self-reference methods, like most of those discussed in Section 3, namely, those rely on feedback from fixed LMs.

Feedback is indispensable for an iterative approach. In RL, rewards provide evaluative feedback for agents to make decisions. We discuss feedback in the context of LMs in Section 3. Here we discuss reward, interaction, as well as their connection with psychology, which will shed light on iterative improvements of LMs with feedback. Feedback is also data. We single it out here from data to highlight the nature of interaction.

Sparse reward

Rewards may be so sparse that it is challenging for learning algorithms, e.g., in text generation with RLHF, a reward may occur at the completion of the text. Lightman et al. (2023) propose process supervision rather than output supervision to have denser feedback. Hindsight Experience Replay (HER) (Andrychowicz et al., 2017) is a way to handle sparse rewards. Unsupervised auxiliary learning (Jaderberg et al., 2017) is an unsupervised way harnessing environmental signals. Intrinsic motivation (Barto, 2013; Singh et al., 2010) is a way to provide intrinsic rewards. Colas et al. (2020) present a short survey for intrinsically motivated goal-conditioned RL.

Reward shaping is to modify reward function to facilitate learning while maintaining optimal policy (Ng et al., 2000). It is usually a manual endeavour. Jaderberg et al. (2018) employ a learning approach in an end-to-end training pipeline.

Imitation learning

Reward functions may not be available for some RL problems. In imitation learning (Osa et al., 2018), an agent learns to perform a task from expert demonstrations, with sample trajectories, without reinforcement signals. Two main approaches are behavioral cloning and inverse RL. Behavioral

cloning, or learning from demonstration, maps state-action pairs from expert trajectories to a policy, maybe as supervised learning, without learning the reward function (Levine, 2021). Inverse RL is determines a reward function given observations of optimal behavior (Ng and Russell, 2000). Probabilistic approaches are developed for inverse RL with maximum entropy (Ziebart et al., 2008) to deal with uncertainty in noisy and imperfect demonstrations. Ross et al. (2010) reduce imitation learning and structured prediction to no-regret online learning, and propose Dataset Aggregation (DAGGER), which requires interaction with the expert. Abbeel and Ng (2004) approach apprenticeship learning via IRL. Syed and Schapire (2007), Syed et al. (2008), and Syed and Schapire (2010) study apprenticeship learning with linear programming, game theory and reduction to classification.

Supervised fine-tuning follows a behavioral cloning approach. RLHF follows an inverse RL approach. Both of them follow imitation learning.

Reward function

A reward function may not represent the intention of the designer. A negative side effect of a misspecified reward refers to potential poor behaviors resulting from missing important aspects. An old example is about the wish of King Midas, that everything he touched, turned into gold. Unfortunately, his intention did not include food, family members, and many more. Russell and Norvig (2020) give an example that a vacuum cleaner collects more dust to receive more rewards by ejecting collected dust. Hadfield-Menell et al. (2016) propose a cooperative inverse RL (CIRL) game for the value alignment problem. Hadfield-Menell et al. (2017) introduce inverse reward design (IRD) to infer the true reward function, based on a designed reward function, an intended decision problem, e.g., an MDP, and a set of possible reward functions. Dragan (2020) talks about optimizing intended reward functions.

Embodiments and social interaction

As discussed in Section 2.1, LMs can interact with physical and human worlds through embodiments and social interaction to improve grounding and agency. Shumailov et al. (2023) discuss the model collapse issue when sampling from a learned model like an LM. Iterative improvements from feedback by interacting with the world, like the Dyna framework (Sutton and Barto, 2018), can mitigate or even eliminate such an issue.

Liu et al. (2023d) propose Sirius for human-in-the-loop learning for robotics. As discussed earlier, Lee et al. (2021) propose PEBBLE leveraging human-in-the-loop feedback. Before the large language model era, Abbeel (2021) discusses that, similar to the pre-training then finetuning in computer vision on ImageNet and in NLP, like GPT-X and BERT, on Internet text, we may be able to pretrain large-scale neural networks for robotics as a general solution, with unsupervised representation learning on Internet video and text, with unsupervised (reward-free) RL pre-training, mostly on simulators and little on the real world data, with human-in-the-loop RL, and with few shot imitation learning on demonstrations.

Reinforcement learning integrates with social learning, e.g., Krishna et al. (2022) show that socially situated AI helps learning from human interaction, and Ndousse et al. (2021) study social learning via multi-agent RL. Wang et al. (2023c) survey interactive NLP, considering interactions with humans, knowledge bases (KBs), models and tools, and environments. In Figure 3, users together with tools and APIs including KBs and models are part of the environment. Bolotta and Dumas (2022) discuss social interaction as the "dark matter" of AI.

Connection with psychology

When humans are involved, psychology and behavioural science may provide insights. From self-motivation theory (Ryan and Deci, 2020), the basic psychological needs of autonomy, competence and relatedness mediate positive user experience outcomes such as engagement, motivation and thriving (Peters et al., 2018). Flow is about the psychology of optimal experience (Csikszentmihalyi, 2008). As such, they constitute specific measurable parameters for which designers can design in order to foster these outcomes within different spheres of experience. Such self-motivation theory and flow, or positive psychology, may help the design of reward and human-computer interaction (HCI), and there are applications in games (Tyack and Mekler, 2020), education (Ryan and Deci, 2020), etc. Cruz and Igarashi (2020) survey design principles for interactive RL. Intrinsic motivation has been applied in RL as discussed earlier. RLHF is an approach dealing with preference and value alignment. However, it appears that self-motivation theory, flow

and positive psychology are under-explored in AI.

6.2 Methodology

AI, in particular, LMs, is enjoy a rapid progress. With ample resources including talents, compute and fundings and the focused attention, there will be more efficient and effective solutions from hardware to software, from theory to practice, including but not limited to processor, system level softwares like compilers and schedulers, neural network architecture, learning algorithms like those for pre-training and fine-tuning, distributed and/or decentralized algorithms, all sorts of applications ranging from enterprise to customer and from cloud to edge devices, and solving issues like hallucination, privacy, safety and human-value alignment.

See Tay et al. (2022) for a survey about efficient Transformers. See Treviso et al. (2023) for a survey about efficient methods for NLP.

For concrete examples, see e.g., Backpack (Hewitt et al., 2023) for a new network architecture, RWKV (Peng et al., 2023) for reinvention of RNN for LMs, Sophia (Liu et al., 2023c) a second-order optimizer for speed-up, AWQ (Lin et al., 2023) for compression and acceleration, and Goat (Liu and Low, 2023) outperforming GPT-4 on arithmetic tasks. We introduce Gorilla (Patil et al., 2023), TinyStories (Eldan and Li, 2023) and phi-1 (Gunasekar et al., 2023) in Section 2.6. See a talk (Choi, 2022) about small vs large LMs.

Ramamurthy et al. (2023) propose an open-source modular library, Reinforcement Learning for Language Models (RL4LMs), General Reinforced-language Understanding Evaluation (GRUE) benchmark, and an RL algorithm Natural Language Policy Optimization (NLPO).

Sutton (2019) states that search and learning are general purpose methods that scale arbitrarily with computation, and also highlights the importance of meta methods. People may tend to scale up the sizes of the neural network and the training dataset to achieve better performance, refering to the Bitter Lesson as a support, which, however, is a misreading. For example, scaling up heuristic search algorithms like A* and IDA* failed to achieve a superhuman Go. AlphaGo resulted from the culmination of achievements in deep learning, reinforcement learning, and Monte-Carlo tree search (MCTS), together with powerful computing. See also Brooks (2019); Kaelbling (2019).

Deshpande et al. (2023) study downscaling effects with a shrunk language, showing benefits of pre-training models of 1.25M parameters and that compute-optimal models break the power law. McKenzie et al. (2023) provide 11 datasets for empirical analysis of inverse scaling laws and discuss the importance of data and objectives for training LMs. Zhang et al. (2023e) propose NeQA, a dataset containing questions with negation and exhibit inverse, U-shaped, or positive scaling. For a "historical" context, Kaplan et al. (2020) study scaling laws that the overall cross-entropy loss of an LM improves with the increased scale of model, dataset and compute for training, and Hoffmann et al. (2022) show that the model and data should be scaled equally for compute-optimal training.

language model	parameter size
BERT	100/340M
T5	60/220/770M, 3/11B
GPT-3	175B
LLaMA	7/13/65B
Alpaca	7B
Dolly	7/13B
Vicuna	7/13B
Guanaco	7/13B
Goat	7B
Gorilla	7B
phi-1	1.3B
TinyStories	10M

Table 8: A glimpse of LM parameter sizes. B: billion. M: million.

Consider the journey from ENIAC in 1945, 27 tons, equivalent to US$6,200,000 in 2021 to iPhone in 2007, 3.5 inch, 135g, 600+ MHz CPU, GPU, 128MB eDRAM, 16GB flash memory, US$499. with faster iterations, we expect smaller, cheaper, yet more capable LMs to appear soon.

Table 8 shows a glimpse of LM parameter sizes.

6.3 Evaluation

Evaluation provides feedback to researchers and developers, as well as to learning algorithms, to make improvements. Evaluation and benchmarks for NLP and language models have been making steady progress. However, there are still lots of challenges, in particular, for interactive applications.

Burnell et al. (2023) presents guidelines for robust evaluation practices with more granular reporting, in particular, in-depth performance break-

downs beyond aggregate metrics and instance-by-instance evaluation results.

Gehrmann et al. (2022) survey obstacles in evaluation of test generation and propose to evaluate a model with multiple datasets via multiple metrics and document human evaluation well. The authors propose the following best best practice & implementation: make informed evaluation choices and document them, measure specific generation effects, analyze and address issues in the used dataset(s), evaluate in a comparable setting, run a well-documented human evaluation, produce robust human evaluation results, document results in model cards, and release model outputs and annotations.

Srivastava et al. (2022) introduce the Beyond the Imitation Game benchmark (BIG-bench) which has more than 200 tasks.

Liang et al. (2022) present HELM, Holistic Evaluation of Language Models, to improve transparency of LMs. The authors present a taxonomy of scenarios and metrics to evaluation LMs with a multi-metric approach: a) evaluate 16 core scenarios each with 7 metrics, namely, accuracy, calibration, robustness, fairness, bias, toxicity, and efficiency; b) conduct 7 targeted evaluations for 26 targeted scenarios for specific aspects like knowledge, reasoning, memorization, copyright and disinformation; and c) evaluate 30 LMs on all 42 scenarios.

Lee et al. (2022) introduce Human-AI Language-based Interaction Evaluation (HALIE) to extend non-interactive evaluation w.r.t three factors: 1) targets, including full process and final output, 2) perspectives, including first-person and third-party, and 3) criteria including preference and quality.

Biderman et al. (2023) propose Pythia to study the process of training LMs with checkpoints for 16 LMs with parameter sizes from 70M to 12B.

Maynez et al. (2023) benchmark LM capacities for 27 generation tasks and provide recommendations on the selection of tasks, methods and metrics, and on practice to monitor generation capacities including benchmarks, automated metrics, and efficient utilization of computational resources.

Mozannar et al. (2023) propose CodeRec User Programming States (CUPS) to model user behaviour and costs in AI-assisted programming with GitHub Copilot and show that 34.3% of total session time spends on double-checking and editing suggestions.

Francis et al. (2023) discuss the principles for social robot navigation: safety, comfort, legibility, politeness, social competency, agent understanding, proactivity and responsiveness to context, and based on which, the guidelines for evaluation w.r.t. metrics, scenarios, benchmarks, datasets and simulators.

6.4 Interpretability

Explainability and interpretability are critical for AI (Barredo Arrieta et al., 2020). We briefly review some work and make connection with many concepts/issues in AI.

First we discuss the definition. As in Rudin et al. (2021), explainable AI (XAI) "attempts to explain a black box using an approximation model, derivatives, variable importance measures, or other statistics", whereas interpretable ML creates "a predictive model that is not a black box". In Murdoch et al. (2019), interpretable ML includes explainable ML, intelligible ML, and transparent ML. Lipton (2018) argues that explanation is post hoc interpretability. Miller (2019) treats explainability and interpretability as the same.

Miller (2019) survey how people define, generate, select, evaluate, and present explanations in philosophy, psychology, and cognitive science and the implication for explainable AI. The major findings are: explanations are contrastive, explanation are selected in a biased manner, probabilities probably don't matter, and explanations are social. The author summarized that "explanations are not just the presentation of associations and causes (causal attribution), they are contextual".

Doshi-Velez and Kim (2017) propose to define interpretability "as the ability to explain or to present in understandable terms to a human". The authors discuss the relationship between interpretability with other desiderata of ML systems. Fairness or unbiasedness concerns with groups being protected from explicit or implicit discrimination. Privacy is about the protection of sensitive information in the data. An algorithm is reliable and robust if it can achieve a certain level of performance with variation in parameters or inputs. The predicted change in output due to a perturbation, according to causality, will occur in the real system. A method is usable if it provides information to help users to accomplish a task. Trust is about a system with confidence of human users. Interpretability qualitatively assists to meet these

properties: fairness, privacy, reliability, robustness, causality, usability and trust.

Lipton (2018) discusses the desiderata and methods for interpretable AI. Desiderata include trust, causality, transferability, informativeness, and fair and ethical decision making. Techniques and model properties for interpretability include transparency and post hoc explanations. The different levels of transparency are: simulatability for the entire model, decomposability for individual components such as parameters, and algorithmic transparency for the training algorithm.

Murdoch et al. (2019) propose to define interpretable machine learning as "the extraction of relevant knowledge from a machine-learning model concerning relationships either contained in data or learned by the model". The authors propose the predictive, descriptive, relevant framework, with desiderata for evaluation: predictive accuracy, descriptive accuracy, and relevancy judged relative to a human audience.

Rudin et al. (2021) propose to define interpretable ML in one sentence: "an interpretable model is constrained, following a domain-specific set of constraints that make reasoning processes understandable". The authors discusses five principles and ten grand challenges of interpretable ML.

Kim (2023a) proposes to build a language to communicate with AI for alignment with our values, by reflecting the nature of the machines and expanding what we know.

Neural networks are notoriously known as blackboxes, especially giant ones like GPT-3. Bills et al. (2023) propose to explain neurons in LMs with LMs, rather than with ground truths. A local explainable method has inherent limitations since distributed representation is critical for neural networks. Moreover, a local method like saliency maps may have issues, e.g., see Adebayo et al. (2018) and Rudin (2019). Rudin (2019) discusses issues with post hoc explainable methods for high stakes decisions and argues to use inherent interpretable approaches instead. Explainability and interpretability for (large) LMs is still nascent and calls for more investigations.

6.5 Constraints

AI for good is a goal. Besides predictive and optimal, it is desirable for an AI system to be safe, robust, adaptive, reliable, stable, transparent, fair, trustworthy, explainable, etc., and not to have behaviours like discrimination w.r.t. race, gender, nationality, etc. Constrains may express them.

A predictive model is built on domain knowledge, real-world data, and high-fidelity simulators; a robust method accounts for worst-case scenarios and takes conservative actions, and an adaptive method learns from online observations and adapts to unknown situations (Brunke et al., 2021).

Thomas et al. (2019) discuss that, to prevent undesirable behaviour of intelligent machines, a user of a standard ML algorithm needs to constrain the algorithm's behaviour in the objective function (with soft constraints or robust and risk-sensitive methods) or in the feasible set (with hard constraints, chance constraints, or robust optimization methods), both of which requires domain knowledge or extra data analysis. The authors propose a framework to shift the burden from the user to the designer of the algorithm, by allowing the user to place probabilistic constraints on the solution directly, for classification, regression, and RL.

Wiens et al. (2019) discuss how to do no harm in the context of healthcare, which may somewhat generalize to AI. AI practitioners, esp. those with AI power and resources and/or those dealing with high-stake applications like healthcare and autonomous driving, may need to take a "Hippocratic oath" or even go under stricter regulation.

Wing (2021) reviews trustworthy AI. Brunke et al. (2021) survey safe learning in robotics, with perspectives from learning-based control to safe RL. Szepesvári (2020) discusses multi-objective and constrained RL. García and Fernández (2015) present a survey on safe RL. It is interesting to explore how to incorporate ideas about constraints to LMs.

6.6 Intelligence

There are long-standing debates about nature versus nurture, empiricism versus rationalism, and connectionism vs symbolism. We discuss earlier the importance of experience (Sutton, 2022a), a common model of intelligent agent (Sutton, 2022b), an autonomous machine intelligence architecture (LeCun, 2022), and a modular architecture Mahowald et al. (2023).

Lake et al. (2017) discuss that we should build machines toward human-like learning and thinking. In particular, we should 1) build causal world models to support understanding and explanation, seeing entities rather than just raw inputs or features,

rather than just pattern recognition, 2) support and enrich the learned knowledge grounding in intuitive physics and intuitive psychology, and 3) represent, acquire, and generalize knowledge, leveraging compositionality and learning to learn, rapidly adapt to new tasks and scenarios, recombining representations, without retraining from scratch.

Jordan (2019) highlights the need of meaning and reasoning for NLP, causality, representations of uncertainty and long-term goals.

Pearl (2020) discusses that learning is guided by data and model, and argues the importance of balancing empiricism with a model for expediency, transparency and explainability.

Bengio et al. (2021) propose a neuro-symbolic approach to combine the merits from both sides: symbolic AI for system 2 abilities like reasoning, composability, and abstraction, and strengths of deep learning including "efficient large-scale learning using differentiable computation and gradient-based adaptation, grounding of high-level concepts in low-level perception and action, handling uncertain data, and using distributed representations". Littman et al. (2021) also highlights a neuro-symbolic approach.

Legg and Hutter (2007) compare tests of intelligence w.r.t. the following properties: valid, informative, wide range, general, dynamic, unbiased, fundamental, formal, objective, fully defined, universal, practical, and test vs. definition. Chollet (2019) presents the Abstraction and Reasoning Corpus (ARC) benchmark.

Learning to learn is a core ingredient to achieve strong AI (Botvinick et al., 2019; Kaelbling, 2020; Lake et al., 2017; Sutton, 2019), and has a long history, e.g., Schmidhuber (1987), Bengio et al. (1991), and Thrun and Pratt (1998).

Learning to learn, a.k.a. meta-learning, is learning about some aspects of learning. It includes concepts as broad as transfer learning, multi-task learning, one/few/zero-shot learning, learning to reinforcement learn, learning to optimize, learning combinatorial optimization, hyper-parameter learning, neural architecture design, automated machine learning (AutoML)/AutoRL/AutoAI, etc. It is closely related to continual learning and life-long learning (Khetarpal et al., 2020).

The aim of few-shot meta-learning is to train a model adaptive to a new task quickly, using only a few data samples and training iterations (Finn et al., 2017). Transfer learning is about transferring knowledge learned from different domains, possibly with different feature spaces and/or different data distributions (Taylor and Stone, 2009; Pan and Yang, 2010). Curriculum learning (Bengio et al., 2009; Narvekar et al., 2020; Baker et al., 2020; Vinyals et al., 2019), model distillation/compression (Hinton et al., 2014; Czarnecki et al., 2019), and sim-to-real are particular types of transfer learning. Multitask learning (Caruana, 1997) learns related tasks with a shared representation in parallel, leveraging information in related tasks as an inductive bias, to improve generalization, and to help improve learning for all tasks. Schölkopf et al. (2021) discuss causal representation learning for transfer learning, multitask learning, continual learning, RL, etc. SeeHutter et al. (2019) for a book on AutoML, Hospedales et al. (2021) for a survey on meta-learning, Singh (2017) for a tutorial about continual learning, (Chen et al., 2021) for a survey and a benchmark on learn to optimize, and (Portelas et al., 2020) for a survey on automatic curriculum learning for deep RL.

The above sheds lights on how to achieve more general and stronger intelligence.

Gershman et al. (2005) discuss that computational rationality leads to approximations when maximizing expected utility for decision making, considering the cost of computation in complicated real-world problems. Although there is exponentially growth in computation, this is (very likely) still a valid principle.

Learning to learn, like transfer / few-shot / multi-task / meta-learning are popular methods for general intelligence recently. General intelligence thus boils down to multi-objective and/or multi-constraint problems, which are about approximation and compromise. See Section 4 for approximation and Section 6.5 for constraints.

When building an AI system, we need to consider the boundary and set a pragmatical goal. Instead of training a system optimizing everything or handling all potential tasks satisfactorily, we may follow how humans have been organizing the society in the long history, i.e., decompose the whole into parts, solve them separately and let them collaborate. In the pre-training then fine-tuning pipeline, we may pre-train with data from selected rather than all sorts of tasks to improve the quality of representation (Schölkopf et al., 2021) and to avoid issues like negative transfer (Taylor and Stone, 2009; Pan and Yang, 2010).

7 Conclusion

Iterative improvements from feedback is a general approach for many, if not all, successful systems. Ground-truth-in-the-loop is critical. Reinforcement learning is a promising framework to achieve sound and solid grounding and agency of language models, by interacting with physical and social world, although pre-training then fine-tuning is a popular approach. Small modules are feasible for frequent updates. This helps bridge the LM-to-real gap and achieve optimality and controllability, besides feasibility and correctness. This facilitates adaptability of language models to humans, but not vice versa, as current prompt engineering may require significant efforts for humans to adapt to language models. This requires valuable and reliable data, feedback and evaluation, with sample-, time-, and space-efficient algorithms, considering ethical and social issues.

Limitations

This is a perspective paper with a brief survey. More considerations are needed for ethical and social aspects of language models and AI.

References

Pieter Abbeel. 2021. Towards a general solution for robotics. https://www.youtube.com/watch?v=l9pwlXXsi7Q. CVPR 2021 Keynote.

Pieter Abbeel and Andrew Y Ng. 2004. Apprenticeship learning via inverse reinforcement learning. In *ICML*.

Adaptive Agent Team, Jakob Bauer, Kate Baumli, Satinder Baveja, Feryal Behbahani, Avishkar Bhoopchand, Nathalie Bradley-Schmieg, Michael Chang, Natalie Clay, Adrian Collister, Vibhavari Dasagi, Lucy Gonzalez, Karol Gregor, Edward Hughes, Sheleem Kashem, Maria Loks-Thompson, Hannah Openshaw, Jack Parker-Holder, Shreya Pathak, Nicolas Perez-Nieves, Nemanja Rakicevic, Tim Rocktäschel, Yannick Schroecker, Jakub Sygnowski, Karl Tuyls, Sarah York, Alexander Zacherl, and Lei Zhang. 2023. Human-timescale adaptation in an open-ended task space. *arXiv*.

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity checks for saliency maps. In *NeurIPS*.

Alekh Agarwal, Sarah Bird, Markus Cozowicz, Luong Hoang, John Langford, Stephen Lee, Jiaji Li, Dan Melamed, Gal Oshri, Oswaldo Ribas, Siddhartha Sen, and Alex Slivkins. 2016. Making contextual decisions with low technical debt. *arXiv*.

Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank. 2023. MusicLM: Generating music from text. *arXiv*.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. 2022. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on Robot Learning (CoRL)*.

Ari Allyn-Feuer and Ted Sanders. 2023. Transformative agi by 2043 is <1% likely. *arXiv*.

Jacob Andreas. 2022. Language models as agent models. In *EMNLP*.

M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba. 2017. Hindsight experience replay. In *NIPS*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and

Jared Kaplan. 2022b. Constitutional AI: Harmlessness from AI feedback. *arXiv*.

Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. 2020. Emergent tool use from multi-agent autocurricula. In *ICLR*.

Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyan Hu, Athul Paul Jacob, Mojtaba Komeili, Karthik Konath, Minae Kwon, Adam Lerer, Mike Lewis, Alexander H. Miller, Sash Mitts, Aditya Renduchintala, Stephen Roller, Dirk Rowe, Weiyan Shi, Joe Spisak, Alexander Wei, David Wu, Hugh Zhang, and Markus Zijlstra. 2022. Human-level play in the game of Diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074.

Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, Avi Schwarzschild, Andrew Gordon Wilson, Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. 2023. A cookbook of self-supervised learning. *arXiv*.

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. 2020. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115.

A.G. Barto. 2013. Intrinsic motivation and reinforcement learning. In G. Baldassarre and M. Mirolli, editors, *Intrinsically Motivated Learning in Natural and Artificial Systems*. Springer, Berlin, Heidelberg.

Marc G. Bellemare, Salvatore Candido, Pablo Samuel Castro, Jun Gong, Marlos C. Machado, Subhodeep Moitra, Sameera S. Ponda, and Ziyu Wang. 2020. Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature*, 588(7836):77–82.

Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. 2013. The arcade learning environment: An evaluation platform for general agents. *JAIR*, 47:253–279.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *ACM conference on fairness, accountability, and transparency*.

Y. Bengio, S. Bengio, and J. Cloutier. 1991. Learning a synaptic learning rule. In *International Joint Conference on Neural Networks (IJCNN)*.

Yoshua Bengio. 2023. How rogue AIs may arise. https://yoshuabengio.org/2023/05/22/how-rogue-ais-may-arise/.

Yoshua Bengio, Yann LeCun, and Geoffrey Hinton. 2021. Deep learning for AI. *Communications of the ACM*, 64(7):58–65.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *ICML*.

Dimitri P. Bertsekas. 2019. *Reinforcement Learning and Optimal Control*. Athena Scientific.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. *arXiv*.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, Dragomir Radev, Eduardo González Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza Benyamina, Hieu Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Frohberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro Von Werra, Leon Weber, Long Phan, Loubna Ben allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto Luis López, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, Shayne Longpre, Somaieh Nikpoor, Stanislav Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev,

Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Davut Emre Taşar, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Al-shaibani, Matteo Manica, Nihal Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Fevry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiangru Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Yallow Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre François Lavallée, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aurélie Névéol, Charles Lovering, Dan Garrette, Deepak Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Jordan Clive, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shachar Mirkin, Shani Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdeněk Kasner, Alice Rueda, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ana Santos, Anthony Hevia, Antigona Unldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh HajiHosseini, Bahareh Behroozi, Benjamin Ajibade, Bharat Saxena, Carlos Muñoz Ferrandis, Danish Contractor, David Lansky, Davis David, Douwe Kiela, Duong A. Nguyen, Edward Tan, Emi Baylor, Ezinwanne Ozoani, Fatima Mirza, Frankline Ononiwu, Habib Rezanejad, Hessie Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nejadgholi, Jesse Passmore, Josh Seltzer, Julio Bonis Sanz, Livia Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, Muhammed Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nour Fahmy, Olanrewaju Samuel, Ran An, Rasmus Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas Wang, Sourav Roy, Sylvain Viguier, Thanh Le, Tobi Oyebade, Trieu Le, Yoyo Yang, Zach Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Singh, Benjamin Beilharz, Bo Wang, Caio Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel León Periñán, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrimann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Imane Bello, Ishani Dash, Jihyun Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthik Rangasai Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, Maria A Castillo, Marianna Nezhurina, Mario Sänger, Matthias Samwald, Michael Cullan, Michael Weinberg, Michiel De Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patrick Haller, Ramya Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Sinee Sang-aroonsiri, Srishti Kumar, Stefan Schweter, Sushil Bharati, Tanmay Laud, Théo Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yash Shailesh Bajaj, Yash Venkatraman, Yifan Xu, Yingxin Xu, Yu Xu, Zhe Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. 2023. BLOOM: A 176B-parameter open-access multilingual language model. *arXiv*.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. 2023. Language models can explain neurons in language models. https://tinyurl.com/3j25tfnb.

Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph Turian. 2020. Experience grounds language. In *EMNLP*.

J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. S. Sukhatme. 2017. Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics*, 33(6):1273–1291.

Samuele Bolotta and Guillaume Dumas. 2022. Social Neuro AI: Social interaction as the "dark matter" of AI. *Frontiers in Computer Science*, (4).

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar

Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2022. On the opportunities and risks of foundation models. *arXiv*.

Matthew Botvinick, Sam Ritter, Jane X. Wang, Zeb Kurth-Nelson, Charles Blundell, and Demis Hassabis. 2019. Reinforcement learning, fast and slow. *Trends in Cognitive Sciences*, 23(5):408–422.

Konstantinos Bousmalis, Giulia Vezzani, Dushyant Rao, Coline Devin, Alex X. Lee, Maria Bauza, Todor Davchev, Yuxiang Zhou, Agrim Gupta, Akhil Raju, Antoine Laurens, Claudio Fantacci, Valentin Dalibard, Martina Zambelli, Murilo Martins, Rugile Pevceviciute, Michiel Blokzijl, Misha Denil, Nathan Batchelor, Thomas Lampe, Emilio Parisotto, Konrad Żołna, Scott Reed, Sergio Gómez Colmenarejo, Jon Scholz, Abbas Abdolmaleki, Oliver Groth, Jean-Baptiste Regli, Oleg Sushkov, Tom Rothörl, José Enrique Chen, Yusuf Aytar, Dave Barker, Joy Ortiz, Martin Riedmiller, Jost Tobias Springenberg, Raia Hadsell, Francesco Nori, and Nicolas Heess. 2023. RoboCat: A self-improving foundation agent for robotic manipulation. *arXiv*.

Samuel R. Bowman. 2023. Eight things to know about large language models. *arXiv*.

Andres M Bran, Sam Cox, Andrew D White, and Philippe Schwaller. 2023. Chemcrow: Augmenting large-language models with chemistry tools. *arXiv*.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. 2022. RT-1: Robotics transformer for real-world control at scale. *arXiv*.

Rodney Brooks. 2019. A better lesson. https://rodneybrooks.com/a-better-lesson/.

Noam Brown and Tuomas Sandholm. 2017. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS*.

Jacob Browning and Yann LeCun. 2023. AI chatbots don't care about your social norms. https://www.noemamag.com/ai-chatbots-dont-care-about-your-social-norms/.

Lukas Brunke, Melissa Greeff, Adam W. Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P. Schoellig. 2021. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv*.

Ryan Burnell, Wout Schellaert, John Burden, Tomer D. Ullman, Fernando Martinez-Plumed, Joshua B. Tenenbaum, Danaja Rutar, Lucy G. Cheke, Jascha Sohl-Dickstein, Melanie Mitchell, Douwe Kiela, Murray Shanahan, Ellen M. Voorhees, Anthony G. Cohn, Joel Z. Leibo, and Jose Hernandez-Orallo. 2023. Rethink reporting of evaluation results in ai. *Science*, 380(6641):136–138.

Tianle Cai, Xuezhi Wang, Tengyu Ma, Xinyun Chen, and Denny Zhou. 2023. Large language models as tool makers. *arXiv*.

Thomas Carta, Clément Romac, Thomas Wolf, Sylvain Lamprier, Olivier Sigaud, and Pierre-Yves Oudeyer. 2023. Grounding large language models in interactive environments with online reinforcement learning. In *ICML*.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28(1):41–75.

Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan Ratliff, and Dieter Fox. 2019. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *ICRA*.

Jiaao Chen, Aston Zhang, Xingjian Shi, Mu Li, Alex Smola, and Diyi Yang. 2023a. Parameter-efficient fine-tuning design spaces. *arXiv*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2023b. Evaluating large language models trained on code. *arXiv*.

Tianlong Chen, Xiaohan Chen, Wuyang Chen, Howard Heaton, Jialin Liu, Zhangyang Wang, and Wotao Yin. 2021. Learning to optimize: A primer and a benchmark. *arXiv*.

Xinshi Chen, Shuang Li, Hui Li, Shaohua Jiang, Yuan Qi, and Le Song. 2019. Generative adversarial user model for reinforcement learning based recommendation system. In *ICML*.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. 2023c. Teaching large language models to self-debug. *arXiv*.

Ted Chiang. 2023. Will A.I. become the new McKinsey? https://www.newyorker.com/science/annals-of-artificial-intelligence/will-ai-become-the-new-mckinsey.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Yejin Choi. 2022. David V.S. Goliath: the art of leaderboarding in the era of extreme-scale neural models. https://www.youtube.com/watch?v=5ey0mMwfVnA.

François Chollet. 2019. On the measure of intelligence. *arXiv*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling language modeling with pathways. *arXiv*.

Brian Christian. 2021. *The Alignment Problem: Machine Learning and Human Values*. WW Norton.

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *NIPS*.

Cédric Colas, Tristan Karch, Olivier Sigaud, and Pierre-Yves Oudeyer. 2020. Intrinsically motivated goal-conditioned reinforcement learning: a short survey. *arXiv*.

Mike Conover, Matt Hayes, Ankit Mathur, Xiangrui Meng, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free Dolly: Introducing the world's first truly open instruction-tuned LLM. https://tinyurl.com/3v9jss39.

Christian Arzate Cruz and Takeo Igarashi. 2020. A survey on interactive reinforcement learning: Design principles and open challenges. In *ACM Designing Interactive Systems Conference*.

Mihaly Csikszentmihalyi. 2008. *Flow: The Psychology of Optimal Experience*. Harper Perennial Modern Classics.

Wojciech Marian Czarnecki, Razvan Pascanu, Simon Osindero, Siddhant M. Jayakumar, Grzegorz Swirszcz, and Max Jaderberg. 2019. Distilling policy distillation. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de las Casas, Craig Donner, Leslie Fritz, Cristian Galperti, Andrea Huber, James Keeling, Maria

Tsimpoukelli, Jackie Kay, Antoine Merle, Jean-Marc Moret, Seb Noury, Federico Pesamosca, David Pfau, Olivier Sauter, Cristian Sommariva, Stefano Coda, Basil Duval, Ambrogio Fasoli, Pushmeet Kohli, Koray Kavukcuoglu, Demis Hassabis, and Martin Riedmiller. 2022. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602:414–419.

Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew Wallingford, Luca Weihs, Mark Yatskar, and Ali Farhadi. 2020. RoboTHOR: An open simulation-to-real embodied AI platform. In *CVPR*.

Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Chris Cundy, Marcus Hutter, Shane Legg, Joel Veness, and Pedro A. Ortega. 2023. Neural networks and the chomsky hierarchy. In *ICLR*.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric P. Xing, and Zhiting Hu. 2022. RLPrompt: Optimizing discrete text prompts with reinforcement learning. In *EMNLP*.

Vijeta Deshpande, Dan Pechi, Shree Thatte, Vladislav Lialin, and Anna Rumshisky. 2023. Honey, i shrunk the language: Language model behavior at reduced scale. In *ACL*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional Transformers for language understanding. *arXiv*.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, Jing Yi, Weilin Zhao, Xiaozhi Wang, Zhiyuan Liu, Hai-Tao Zheng, Jianfei Chen, Yang Liu, Jie Tang, Juanzi Li, and Maosong Sun. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey on in-context learning. *arXiv*.

Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv*.

Anca Dragan. 2020. Optimizing intended reward functions: Extracting all the right information from all the right places. https://www.youtube.com/watch?v=ZNi_Isvlzos.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. 2023. PaLM-E: An embodied multimodal language model. *arXiv*.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2020. GLM: General language model pretraining with autoregressive blank infilling. In *ACL*.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. 2023. Faith and fate: Limits of transformers on compositionality. *arXiv*.

Ronen Eldan and Yuanzhi Li. 2023. TinyStories: How small can language models be and still speak coherent English? *arXiv*.

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. 2022. MineDojo: Building open-ended embodied agents with internet-scale knowledge. In *NeurIPS*.

Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hubert, Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Francisco J. R. Ruiz, Julian Schrittwieser, Grzegorz Swirszcz, David Silver, Demis Hassabis, and Pushmeet Kohli. 2022. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610(7930):47–53.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*.

Anthony Francis, Claudia Perez-D'Arpino, Chengshu Li, Fei Xia, Alexandre Alahi, Rachid Alami, Aniket Bera, Abhijat Biswas, Joydeep Biswas, Rohan Chandra, Hao-Tien Lewis Chiang, Michael Everett, Sehoon Ha, Justin Hart, Jonathan P. How, Haresh Karnan, Tsang-Wei Edward Lee, Luis J. Manso, Reuth Mirksy, Soeren Pirk, Phani Teja Singamaneni, Peter Stone, Ada V. Taylor, Peter Trautman, Nathan Tsoi, Marynel Vazquez, Xuesu Xiao, Peng Xu, Naoki Yokoyama, Alexander Toshev, and Roberto Martin-Martin. 2023. Principles and guidelines for evaluating social robot navigation algorithms. *arXiv*.

Jianfeng Gao, Michel Galley, and Lihong Li. 2019. Neural approaches to Conversational AI. *Foundations and Trends in Information Retrieval*, 13(2-3):127–298.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2022. Pal: Program-aided language models. *arXiv*.

Javier García and Fernando Fernández. 2015. A comprehensive survey on safe reinforcement learning. *JMLR*, 16:1437–1480.

Jason Gauci, Edoardo Conti, Yitao Liang, Kittipat Virochsiri, Yuchen He, Zachary Kaden, Vivek Narayanan, Xiaohui Ye, and Zhengxing Chen. 2019. Horizon: Facebook's open source applied reinforcement learning platform. In *RL4RealLife*.

Yingqiang Ge, Wenyue Hua, Jianchao Ji, Juntao Tan, Shuyuan Xu, and Yongfeng Zhang. 2023. OpenAGI: When LLM meets domain experts. *arXiv*.

Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2022. Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text. *arXiv*.

Xinyang Geng, Arnav Gudibande, Hao Liu, Eric Wallace, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. Koala: A dialogue model for academic research. Blog post.

Samuel J. Gershman, Eric J. Horvitz, and Joshua B. Tenenbaum. 2005. Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science*, 349(6245):273–278.

Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, Lucy Campbell-Gillingham, Jonathan Uesato, Po-Sen Huang, Ramona Comanescu, Fan Yang, Abigail See, Sumanth Dathathri, Rory Greig, Charlie Chen, Doug Fritz, Jaume Sanchez Elias, Richard Green, Soňa Mokrá, Nicholas Fernando, Boxi Wu, Rachel Foley, Susannah Young, Iason Gabriel, William Isaac, John Mellor, Demis Hassabis, Koray Kavukcuoglu, Lisa Anne Hendricks, and Geoffrey Irving. 2022. Improving alignment of dialogue agents via targeted human judgements. *arXiv*.

Yoav Goldberg. 2023. Some remarks on large language models. https://gist.github.com/yoavg/59d174608e92e845c8994ac2e234c8a9.

Sharon Goldman. 2023. Top AI researcher dismisses AI 'extinction' fears, challenges 'hero scientist' narrative. https://tinyurl.com/bdd772p5.

Muhammad Waleed Gondal, Manuel Wüthrich, Djordje Miladinović, Francesco Locatello, Martin Breidt, Valentin Volchkov, Joel Akpo, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. 2019. On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. In *NeurIPS*.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. The false promise of imitating proprietary llms. *arXiv*.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks are all you need. *arXiv*.

Dylan Hadfield-Menell, Anca Dragan, Pieter Abbeel, and Stuart Russell. 2016. Cooperative inverse reinforcement learning. In *NIPS*.

Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart Russell, and Anca Dragan. 2017. Inverse reward design. In *NIPS*.

Patrick Haluptzok, Matthew Bowers, and Adam Tauman Kalai. 2023. Language models can teach themselves to program better. In *ICLR*.

Josiah P. Hanna, Siddharth Desai, Haresh Karnan, Garrett Warnell, and Peter Stone. 2021. Grounded action transformation for sim-to-real reinforcement learning. *Machine Learning*, 110:2469–2499.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. *arXiv*.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2023. Towards a unified view of parameter-efficient transfer learning. In *ICLR*.

Richard Held and Alan Hein. 1963. Movement-produced stimulation in the development of visually guided behaviour. *Journal of comparative and physiological Psychology*, (56):872–876.

John Hewitt, John Thickstun, Christopher D. Manning, and Percy Liang. 2023. Backpack language models. In *ACL*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2014. Distilling the knowledge in a neural network. In *NIPS 2014 Deep Learning Workshop*.

Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. In *NIPS*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models. *arXiv*.

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. 2021. Meta-learning in neural networks: A survey. *TPAMI*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-rank adaptation of large language models. *arXiv*.

Rongjie Huang, Mingze Li, Dongchao Yang, Jiatong Shi, Xuankai Chang, Zhenhui Ye, Yuning Wu, Zhiqing Hong, Jiawei Huang, Jinglin Liu, Yi Ren, Zhou Zhao, and Shinji Watanabe. 2023. AudioGPT: Understanding and generating speech, music, sound, and talking head. *arXiv*.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. 2022a. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. *arXiv*.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Noah Brown, Tomas Jackson, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. 2022b. Inner monologue: Embodied reasoning through planning with language models. *arXiv*.

Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren, editors. 2019. *Automatic Machine Learning: Methods, Systems, Challenges*. Springer.

M. Jaderberg, W. M. Czarnecki, I. Dunning, L. Marris, G. Lever, A. Garcia Castaneda, C. Beattie, N. C. Rabinowitz, A. S. Morcos, A. Ruderman, N. Sonnerat, T. Green, L. Deason, J. Z. Leibo, D. Silver, D. Hassabis, K. Kavukcuoglu, and T. Graepel. 2018. Human-level performance in first-person multiplayer games with population-based deep reinforcement learning. *arXiv*.

Max Jaderberg, Volodymir Mnih, Wojciech Czarnecki, Tom Schaul, Joel Z. Leibo, David Silver, and Koray Kavukcuoglu. 2017. Reinforcement learning with unsupervised auxiliary tasks. In *ICLR*.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. 2020. RLBench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019 – 3026.

Stephen James, Paul Wohlhart, Mrinal Kalakrishnan, Dmitry Kalashnikov, Alex Irpan, Julian Ibarz, Sergey Levine, Raia Hadsell, and Konstantinos Bousmalis. 2019. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In *CVPR*.

Zhijing Jin, Jiarui Liu, Zhiheng Lyu, Spencer Poff, Mrinmaya Sachan, Rada Mihalcea, Mona Diab, and Bernhard Schölkopf. 2023. Can large language models infer causation from correlation? *arXiv*.

Michael I. Jordan. 2019. Artificial Intelligence—the revolution hasn't happened yet. *Harvard Data Science Review*, 1(1).

Abhishek Kadian, Joanne Truong, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra. 2020. Sim2Real predictivity: Does evaluation in simulation predict real-world performance? *IEEE Robotics and Automation Letters*, 5(4).

Leslie Kaelbling. 2019. Engineering AI. https://medium.com/@lpk_61328/engineering-ai-e310b8044d78.

Leslie Pack Kaelbling. 2020. The foundation of efficient robot learning. *Science*, 369(6506):915–916.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv*.

Ehud Karpas, Omri Abend, Yonatan Belinkov, Barak Lenz, Opher Lieber, Nir Ratner, Yoav Shoham, Hofit Bata, Yoav Levine, Kevin Leyton-Brown, Dor Muhlgay, Noam Rozen, Erez Schwartz, Gal Shachaf, Shai Shalev-Shwartz, Amnon Shashua, and Moshe Tenenholtz. 2022. Mrkl systems: A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning. *arXiv*.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2023. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp. *arXiv*.

Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. 2020. Towards continual reinforcement learning: A review and perspectives. *arXiv*.

Celeste Kidd and Abeba Birhane. 2023. How ai can distort human beliefs. *Science*, 380(6651):1222–1223.

Been Kim. 2023a. Beyond interpretability: developing a language to shape our relationships with ai. https://tinyurl.com/ycykjca2.

Geunwoo Kim, Pierre Baldi, and Stephen McAleer. 2023. Language models can solve computer tasks. *arXiv*.

Sung Kim. 2023b. List of open sourced fine-tuned large language models (LLM). https://tinyurl.com/ykf57jd6.

W. Bradley Knox and Peter Stone. 2008. TAMER: Training an agent manually via evaluative reinforcement. In *IEEE 7th International Conference on Development and Learning*.

Jan Kocoń, Igor Cichecki, Oliwier Kaszyca, Mateusz Kochanek, Dominika Szydło, Joanna Baran, Julita Bielaniewicz, Marcin Gruza, Arkadiusz Janz, Kamil Kanclerz, Anna Kocoń, Bartłomiej Koptyra, Wiktoria Mieleszczenko-Kowszewicz, Piotr Miłkowski, Marcin Oleksy, Maciej Piasecki, Łukasz Radliński,

Konrad Wojtasik, Stanisław Woźniak, and Przemysław Kazienko. 2023. ChatGPT: Jack of all trades, master of none. *arXiv*.

Ranjay Krishna, Donsuk Lee, Li Fei-Fei, and Michael S. Bernstein. 2022. Socially situated artificial intelligence enables learning from human interaction. *PNAS*, 119(39).

John E. Laird, Christian Lebiere, and Paul S. Rosenbloom. 2017. A standard model of the mind: Toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics. *AI Magazine*, 38(4):13–26.

Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. 2017. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40.

Nathan Lambert, Louis Castricato, Leandro von Werra, and Alex Havrilla. 2022. Illustrating reinforcement learning from human feedback (RLHF). https://huggingface.co/blog/rlhf.

Andrew Kyle Lampinen, Stephanie C Y Chan, Ishita Dasgupta, Andrew J Nam, and Jane X Wang. 2023. Passive learning of active causal strategies in agents and language models. *arXiv*.

Alexander Lavin, Hector Zenil, Brooks Paige, David Krakauer, Justin Gottschlich, Tim Mattson, Anima Anandkumar, Sanjay Choudry, Kamil Rocki, Atılım Güneş Baydin, Carina Prunkl, Brooks Paige, Olexandr Isayev, Erik Peterson, Peter L. McMahon, Jakob Macke, Kyle Cranmer, Jiaxin Zhang, Haruko Wainwright, Adi Hanuka, Manuela Veloso, Samuel Assefa, Stephan Zheng, and Avi Pfeffer. 2021. Simulation intelligence: Towards a new generation of scientific methods. *arXiv*.

Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven C. H. Hoi. 2022. CodeRL: Mastering code generation through pretrained models and deep reinforcement learning. In *NeurIPS*.

Yann LeCun. 2022. A path towards autonomous machine intelligence. https://openreview.net/pdf?id=BZ5a1r-kVsf.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature*, 521:436–444.

Kimin Lee, Laura Smith, and Pieter Abbeel. 2021. PEBBLE: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In *ICML*.

Mina Lee, Megha Srivastava, Amelia Hardy, John Thickstun, Esin Durmus, Ashwin Paranjape, Ines Gerard-Ursin, Xiang Lisa Li, Faisal Ladhak, Frieda Rong, Rose E. Wang, Minae Kwon, Joon Sung Park, Hancheng Cao, Tony Lee, Rishi Bommasani, Michael Bernstein, and Percy Liang. 2022. Evaluating human-language model interaction. *arXiv*.

Shane Legg and Marcus Hutter. 2007. Universal intelligence: A definition of machine intelligence. *arXiv*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *EMNLP*.

Sergey Levine. 2021. Deep reinforcement learning course. http://rail.eecs.berkeley.edu/deeprlcourse/.

Sergey Levine. 2023a. The bitterest of lessons: The role of data and optimization in emergence. https://www.youtube.com/watch?v=aDzQwewwv00.

Sergey Levine. 2023b. Offline RL and large language models. https://sergeylevine.substack.com/p/offline-rl-and-large-language-models.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023a. CAMEL: Communicative agents for "mind" exploration of large scale language model society. *arXiv*.

Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023b. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *ICLR*.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *WWW*.

Minghao Li, Feifan Song, Bowen Yu, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. 2023c. API-Bank: A benchmark for tool-augmented LLMs. *arXiv*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL*.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de Masson d'Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey Cherepanov, James Molloy, Daniel J. Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de Freitas, Koray Kavukcuoglu, and Oriol Vinyals. 2022. Competition-level code generation with AlphaCode. *Science*, 378(6624):1092–1097.

Yuxi Li. 2017. Deep reinforcement learning: An overview. *arXiv*.

Yuxi Li. 2022. Reinforcement learning in practice: Opportunities and challenges. *arXiv*.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. 2023a. Code as policies: Language model programs for embodied control. *arXiv*.

Paul Pu Liang, Yiwei Lyu, Xiang Fan, Jeffrey Tsaw, Yudong Liu, Shentong Mo, Dani Yogatama, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2023b. High-modality multimodal transformer: Quantifying modality & interaction heterogeneity for high-modality representation learning. *Transactions on Machine Learning Research*.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2022. Holistic evaluation of language models. *arXiv*.

Yaobo Liang, Chenfei Wu, Ting Song, Wenshan Wu, Yan Xia, Yu Liu, Yang Ou, Shuai Lu, Lei Ji, Shaoguang Mao, Yun Wang, Linjun Shou, Ming Gong, and Nan Duan. 2023c. TaskMatrix.AI: Completing tasks by connecting foundation models with millions of APIs. *arXiv*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv*.

Chu-Cheng Lin, Aaron Jaech, Xin Li, Matthew R. Gormley, and Jason Eisner. 2021. Limitations of autoregressive models and their alternatives. In *NAACL*.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, and Song Han. 2023. AWQ: Activation-aware weight quantization for LLM compression and acceleration. *arXiv*.

Zachary C. Lipton. 2018. The mythos of model interpretability. *ACM Queue*, 16(3).

Michael L. Littman. 2015. Reinforcement learning improves behaviour from evaluative feedback. *Nature*, 521:445–451.

Michael L. Littman, Ifeoma Ajunwa, Guy Berger, Craig Boutilier, Morgan Currie, Finale Doshi-Velez, Gillian Hadfield, Michael C. Horowitz, Charles Isbell, Hiroaki Kitano, Karen Levy, Terah Lyons, Melanie Mitchell, Julie Shah, Steven Sloman, Shannon Vallor, and Toby Walsh. 2021. *Gathering Strength, Gathering Storms: The One Hundred Year Study on Artificial Intelligence (AI100) 2021 Study Panel Report*. Stanford University, Stanford, CA, USA.

Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. 2023a. LLM+P: Empowering large language models with optimal planning proficiency. *arXiv*.

Hao Liu, Carmelo Sferrazza, and Pieter Abbeel. 2023b. Chain of hindsight aligns language models with feedback. *arXiv*.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *NeurIPS*.

Hong Liu, Zhiyuan Li, David Hall, Percy Liang, and Tengyu Ma. 2023c. Sophia: A scalable stochastic second-order optimizer for language model pre-training. *arXiv*.

Huihan Liu, Soroush Nasiriany, Lance Zhang, Zhiyao Bao, and Yuke Zhu. 2023d. Robot learning on the job: Human-in-the-loop autonomy and learning during deployment. In *RSS*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023e. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Tiedong Liu and Bryan Kian Hsiang Low. 2023. Goat: Fine-tuned LLaMA outperforms GPT-4 on arithmetic tasks. *arXiv*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2023f. RoBERTa: A robustly optimized bert pretraining approach. *arXiv*.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. 2023a. Chameleon: Plug-and-play compositional reasoning with large language models. *arXiv*.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. 2023b. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. In *ICLR*.

Marlos C. Machado, Marc G. Bellemare, Erik Talvitie, Joel Veness, Matthew J. Hausknecht, and Michael Bowling. 2018. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *Journal of Artificial Intelligence Research*, 61:523–562.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Sean Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, Amir Yazdanbakhsh, and Peter Clark. 2023. SELF-REFINE: Iterative refinement with self-feedback. *arXiv*.

Kyle Mahowald, Anna A. Ivanova, Idan A. Blank, Nancy Kanwisher, Joshua B. Tenenbaum, and Evelina Fedorenko. 2023. Dissociating language and thought in large language models: a cognitive perspective. *arXiv*.

Daniel J. Mankowitz, Andrea Michi, Anton Zhernov, Marco Gelmi, Marco Selvi, Cosmin Paduraru, Edouard Leurent, Shariq Iqbal, Jean-Baptiste Lespiau, Alex Ahern, Thomas Köppe, Kevin Millikin, Stephen Gaffney, Sophie Elster, Jackson Broshear, Chris Gamble, Kieran Milan, Robert Tung, Minjae Hwang, Taylan Cemgil, Mohammadamin Barekatain, Yujia Li, Amol Mandhane, Thomas Hubert, Julian Schrittwieser, Demis Hassabis, Pushmeet Kohli, Martin Riedmiller, Oriol Vinyals, and David Silver. 2023. Faster sorting algorithms discovered using deep reinforcement learning. *Nature*, 618(7964):257–263.

Yuning Mao, Lambert Mathias, Rui Hou, Amjad Almahairi, Hao Ma, Jiawei Han, Wen tau Yih, and Madian Khabsa. 2022. UniPELT: A unified framework for parameter-efficient language model tuning. In *ACL*.

Gary Marcus. 2023. The sparks of AGI? or the end of science? https://cacm.acm.org/blogs/blog-cacm/271354-the-sparks-of-agi-or-the-end-of-science/fulltext.

Joshua Maynez, Priyanka Agrawal, and Sebastian Gehrmann. 2023. Benchmarking large language model capabilities for conditional generation. *arXiv*.

Ian R. McKenzie, Alexander Lyzhov, Michael Pieler, Alicia Parrish, Aaron Mueller, Ameya Prabhu, Euan McLean, Aaron Kirtland, Alexis Ross, Alisa Liu, Andrew Gritsevskiy, Daniel Wurgaft, Derik Kauffman, Gabriel Recchia, Jiacheng Liu, Joe Cavanagh, Max Weiss, Sicong Huang, The Floating Droid, Tom Tseng, Tomasz Korbak, Xudong Shen, Yuhui Zhang, Zhengping Zhou, Najoung Kim, Samuel R. Bowman, and Ethan Perez. 2023. Inverse scaling: When bigger isn't better. *arXiv*.

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. 2023. Augmented language models: a survey. *arXiv*.

Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38.

Melanie Mitchell. 2020. *Artificial Intelligence: A Guide for Thinking Humans*. Picador.

Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. 2017. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356(6337):508–513.

Hussein Mozannar, Gagan Bansal, Adam Fourney, and Eric Horvitz. 2023. Reading between the lines: Modeling user behavior and costs in ai-assisted programming. *arXiv*.

W. James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. 2019. Definitions, methods, and applications in interpretable machine learning. *PNAS*, 116(44):22071–22080.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2022. WebGPT: Browser-assisted question-answering with human feedback. *arXiv*.

Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E. Taylor, and Peter Stone. 2020. Curriculum learning for reinforcement learning domains: A framework and survey. *JMLR*, 21:1–50.

Kamal Ndousse, Douglas Eck, Sergey Levine, and Natasha Jaques. 2021. Emergent social learning via multi-agent reinforcement learning. In *ICML*.

Andrew Ng and Stuart Russell. 2000. Algorithms for inverse reinforcement learning. In *ICML*.

Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. 2000. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*.

Matt Novak. 2023. Lawyer uses chatgpt in federal court and it goes horribly wrong. https://tinyurl.com/5n7uk84m.

Theo X. Olausson, Jeevana Priya Inala, Chenglong Wang, Jianfeng Gao, and Armando Solar-Lezama. 2023. Demystifying gpt self-repair for code generation. *arXiv*.

OpenAI. 2022a. Introducing chatgpt. https://openai.com/blog/chatgpt.

OpenAI. 2022b. Lessons learned on language model safety and misuse. https://openai.com/research/language-model-safety-and-misuse.

OpenAI. 2023a. GPT-4. https://openai.com/research/gpt-4.

OpenAI. 2023b. GPT-4 technical report. *arXiv*.

T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, and J. Peters. 2018. An algorithmic perspective on imitation learning. *Foundations and trends in Robotics*, 7(1-2):1–179.

Georg Ostrovski, Pablo Samuel Castro, and Will Dabney. 2021. The difficulty of passive learning in deep reinforcement learning. In *NeurIPS*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *arXiv*.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345 – 1359.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. *arXiv*.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. 2023. Gorilla: Large language model connected with massive apis. *arXiv*.

Judea Pearl. 2020. Radical empiricism and machine learning research. https://tinyurl.com/3me5fev2.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, Kranthi Kiran GV, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartlomiej Koptyra, Hayden Lau, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Xiangru Tang, Bolun Wang, Johan S. Wind, Stansilaw Wozniak, Ruichong Zhang, Zhenyuan Zhang, Qihang Zhao, Peng Zhou, Jian Zhu, and Rui-Jie Zhu. 2023. RWKV: Reinventing RNNs for the Transformer era. *arXiv*.

Dorian Peters, Rafael A. Calvo, and Richard M. Ryan. 2018. Designing for motivation, engagement and wellbeing in digital experience. *Frontier in Psychology*, 9(797).

Jonas Pfeiffer, Sebastian Ruder, Ivan Vulić, and Edoardo Maria Ponti. 2023. Modular deep learning. *arXiv*.

Remy Portelas, Cedric Colas, Lilian Weng, Katja Hofmann, and Pierre-Yves Oudeyer. 2020. Automatic curriculum learning for deep rl: A short survey. In *IJCAI*.

Warren B. Powell. 2021. *Reinforcement Learning and Stochastic Optimization: A unified framework for sequential decisions (in progress)*. Wiley.

Zhiwei (Tony) Qin, Xiaocheng Tang, Yan Jiao, Fan Zhang, Zhe Xu, Hongtu Zhu, and Jieping Ye. 2020. Ride-hailing order dispatching at DiDi via reinforcement learning. *INFORMS Journal on Applied Analytics*, 50(5):272–286.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *arXiv*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *arXiv*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *arXiv*.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2022. Scaling language models: Methods, analysis & insights from training Gopher. *arXiv*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text Transformer. *JMLR*, 21(140):1–67.

Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2023. Is reinforcement learning (not) for natural language processing: Benchmarks, baselines, and building blocks for natural language policy optimization. In *ICLR*.

Alexandre Rame, Guillaume Couairon, Mustafa Shukor, Corentin Dancette, Jean-Baptiste Gaya, Laure Soulier, and Matthieu Cord. 2023. Rewarded soups: towards Pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. *arXiv*.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. 2022. A generalist agent. *Transactions on Machine Learning Research*.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *CVPR*.

Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. 2010. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Nicholas Roy, Ingmar Posner, Tim Barfoot, Philippe Beaudoin, Yoshua Bengio, Jeannette Bohg, Oliver Brock, Isabelle Depatie, Dieter Fox, Dan Koditschek, Tomas Lozano-Perez, Vikash Mansinghka, Christopher Pal, Blake Richards, Dorsa Sadigh, Stefan Schaal, Gaurav Sukhatme, Denis Therien, Marc Toussaint, and Michiel Van de Panne. 2021. From machine learning to robotics: Challenges and opportunities for embodied intelligence. *arXiv*.

Sebastian Ruder, Jonas Pfeiffer, and Ivan Vulic. 2022. Modular and parameter-efficient fine-tuning for nlp models. In *EMNLP: Tutorial Abstracts*.

Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215.

Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. 2021. Interpretable machine learning: Fundamental principles and 10 grand challenges. *arXiv*.

Stuart Russell. 2019. *Human Compatible: Artificial Intelligence and the Problem of Control*. Viking.

Stuart Russell and Peter Norvig. 2020. *Artificial Intelligence: A Modern Approach (3rd edition)*. Pearson.

Richard M. Ryan and Edward L. Deci. 2020. Intrinsic and extrinsic motivation from a self-determination theory perspective: Definitions, theory, practices, and future directions. *Contemporary Educational Psychology*, 61.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *arXiv*.

J. Schmidhuber. 1987. *Evolutionary principles in self-referential learning*. Diploma thesis, Tech. Univ. Munich.

Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. 2021. Toward causal representation learning. *Proceedings of the IEEE*, 109(5).

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. 2017. Proximal Policy Optimization Algorithms. *arXiv*.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. HuggingGPT: Solving AI tasks with ChatGPT and its friends in HuggingFace. *arXiv*.

Noah Shinn, Federico Cassano, Beck Labash, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *arXiv*.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. 2023. The curse of recursion: Training on generated data makes models forget. *arXiv*.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. 2018. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. 2017. Mastering the game of Go without human knowledge. *Nature*, 550:354–359.

Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. 2023. Progprompt: Generating situated robot task plans using large language models. *arXiv*.

Satinder Singh. 2017. Steps towards continual learning. https://mila.quebec/en/cours/deep-learning-summer-school-2017/. Deep

Learning and Reinforcement Learning Summer School 2017.

Satinder Singh, Richard Lewis, Andrew Barto, and Jonathan Sorg. 2010. Intrinsically motivated reinforcement learning: An evolutionary perspective. *IEEE Transactions on Autonomous Mental Development*, 2(2).

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, Mike Schaekermann, Amy Wang, Mohamed Amin, Sami Lachgar, Philip Mansfield, Sushant Prakash, Bradley Green, Ewa Dominowska, Blaise Aguera y Arcas, Nenad Tomasev, Yun Liu, Renee Wong, Christopher Semturs, S. Sara Mahdavi, Joelle Barral, Dale Webster, Greg S. Corrado, Yossi Matias, Shekoofeh Azizi, Alan Karthikesalingam, and Vivek Natarajan. 2023. Towards expert-level medical question answering with large language models. *arXiv*.

Linda Smith and Michael Gasser. 2005. The development of embodied cognition: Six lessons from babies. *Artificial Life*, 11:13–29.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, Elton Zhang, Rewon Child, Reza Yazdani Aminabadi, Julie Bernauer, Xia Song, Mohammad Shoeybi, Yuxiong He, Michael Houston, Saurabh Tiwary, and Bryan Catanzaro. 2022. Using DeepSpeed and Megatron to train Megatron-Turing NLG 530B, a large-scale generative language model. *arXiv*.

Charlie Snell, Ilya Kostrikov, Yi Su, Mengjiao Yang, and Sergey Levine. 2023. Offline rl for natural language generation with implicit language q learning. In *ICLR*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew Dai, Andrew La, Andrew Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakaş, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartłomiej Bojanowski, Batuhan Özyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Bryan Orinion, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, César Ferri Ramírez, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Chris Waites, Christian Voigt, Christopher D. Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Moseguí González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Dylan Schrader, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ellie Pavlick, Emanuele Rodola, Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A. Chi, Ethan Dyer, Ethan Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Martínez-Plumed, Francesca Happé, Francois Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Wang, Gonzalo Jaimovitch-López, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannah Kim, Hannah Rashkin, Hannaneh Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Shevlin, Hinrich Schütze, Hiromu Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, Jackson Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B. Simon, James Koppel, James Zheng, James Zou, Jan Kocoń, Jana Thompson, Janelle Wingfield, Jared Kaplan, Jarema Radom, Jascha Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Joan Waweru, John Burden, John Miller, John U. Balis, Jonathan Batchelder, Jonathan Berant, Jörg Frohberg, Jos Rozen, Jose Hernandez-Orallo, Joseph Boudeman, Joseph Guerr, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh D. Dhole, Kevin Gimpel, Kevin Omondi, Kory Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras-Ochando, Louis-Philippe Morency, Luca Moschella, Lucas Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros Colón, Luke Metz, Lütfi Kerem Şenel, Maarten Bosma, Maarten Sap, Maartje ter Hoeve, Maheen Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, Maria Jose Ramírez Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L. Leavitt, Matthias Hagen, Mátyás Schubert, Medina Orduna Baitemirova, Melody Arnaud,

Melvin McElrath, Michael A. Yee, Michael Cohen, Michael Gu, Michael Ivanitskiy, Michael Starritt, Michael Strube, Michał Swędrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimee Xu, Mirac Suzgun, Mitch Walker, Mo Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdeh Gheini, Mukund Varma T, Nanyun Peng, Nathan A. Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nicole Martinez, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha S. Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter Chang, Peter Eckersley, Phu Mon Htut, Pinyu Hwang, Piotr Miłkowski, Piyush Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ramon Risco, Raphaël Millière, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan LeBras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Ruslan Salakhutdinov, Ryan Chi, Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Samuel R. Bowman, Samuel S. Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima, Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy, Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Biderman, Stephanie Lin, Stephen Prasad, Steven T. Piantadosi, Stuart M. Shieber, Summer Misherghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq Ali, Tatsu Hashimoto, Te-Lin Wu, Théo Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo Schick, Timofei Kornev, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay Ramasesh, Vinay Uday Prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yufang Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui Wang, and Ziyi Wu. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv*.

Stability AI. 2023. StableLM: Stability AI language models. https://github.com/stability-AI/stableLM/.

Haotian Sun, Yuchen Zhuang, Lingkai Kong, Bo Dai, and Chao Zhang. 2023a. AdaPlanner: Adaptive planning from feedback with language models. *arXiv*.

Yanchao Sun, Shuang Ma, Ratnesh Madaan, Rogerio Bonatti, Furong Huang, and Ashish Kapoor. 2023b. SMART: Self-supervised multi-task pretraining with control transformers. In *ICLR*.

Rich Sutton. 2019. The bitter lesson. http://incompleteideas.net/IncIdeas/BitterLesson.html.

Richard S. Sutton. 2022a. The increasing role of sensorimotor experience in artificial intelligence. https://www.youtube.com/watch?v=r6o05gOOtpg.

Richard S. Sutton. 2022b. The quest for a common model of the intelligent decision maker. *arXiv*.

Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction (2nd Edition)*. MIT Press.

Richard S. Sutton, Doina Precup, and Satinder Singh. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2):181–211.

Umar Syed, Michael Bowling, and Robert E. Schapire. 2008. Apprenticeship learning using linear programming. In *ICML*.

Umar Syed and Robert E. Schapire. 2007. A game-theoretic approach to apprenticeship learning. In *NIPS*.

Umar Syed and Robert E. Schapire. 2010. A reduction from apprenticeship learning to classification. In *NIPS*.

Csaba Szepesvári. 2010. *Algorithms for Reinforcement Learning*. Morgan & Claypool.

Csaba Szepesvári. 2020. Constrained mdps and the reward hypothesis. https://readingsml.blogspot.com/2020/03/constrained-mdps-and-reward-hypothesis.html.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2022. Efficient transformers: A survey. *ACM Computing Surveys*, 55(6).

Matthew E. Taylor and Peter Stone. 2009. Transfer learning for reinforcement learning domains: A survey. *JMLR*, 10:1633–1685.

Philip S. Thomas, Bruno Castro da Silva, Andrew G. Barto, Stephen Giguere, Yuriy Brun, and Emma Brunskill. 2019. Preventing undesirable behavior of intelligent machines. *Science*, 366:999–1004.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Vincent Zhao, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Pranesh Srinivasan, Laichee Man, Kathleen Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. 2022. LaMDA: Language models for dialog applications. *arXiv*.

Sebastian Thrun and Lorien Pratt, editors. 1998. *Learning to Learn*. Springer.

Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. MuJoCo: A physics engine for model-based control. In *IROS*.

Julian Togelius and Georgios N. Yannakakis. 2023. Choose your weapon: Survival strategies for depressed ai academics. *arXiv*.

Augustin Toma, Patrick R. Lawler, Jimmy Ba, Rahul G. Krishnan, Barry B. Rubin, and Bo Wang. 2023. Clinical camel: An open-source expert-level medical language model with dialogue-based knowledge encoding. *arXiv*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and efficient foundation language models. *arXiv*.

Marcos Treviso, Ji-Ung Lee, Tianchu Ji, Betty van Aken, Qingqing Cao, Manuel R. Ciosici, Michael Hassid, Kenneth Heafield, Sara Hooker, Colin Raffel, Pedro H. Martins, André F. T. Martins, Jessica Zosa Forde, Peter Milder, Edwin Simpson, Noam Slonim, Jesse Dodge, Emma Strubell, Niranjan Balasubramanian, Leon Derczynski, Iryna Gurevych, and Roy Schwartz. 2023. Efficient methods for natural language processing: A survey. *TACL*.

Kathryn Tunyasuvunakool, Jonas Adler, Zachary Wu, Tim Green, Michal Zielinski, Augustin Žídek, Alex Bridgland, Andrew Cowie, Clemens Meyer, Agata Laydon, Sameer Velankar, Gerard J. Kleywegt, Alex Bateman, Richard Evans, Alexander Pritzel, Michael Figurnov, Olaf Ronneberger, Russ Bates, Simon A. A. Kohl, Anna Potapenko, Andrew J. Ballard, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Ellen Clancy, David Reiman, Stig Petersen, Andrew W. Senior, Koray Kavukcuoglu, Ewan Birney, Pushmeet Kohli, John Jumper, and Demis Hassabis. 2021. Highly accurate protein structure prediction for the human proteome. *Nature*, 596(7873):590–596.

April Tyack and Elisa D. Mekler. 2020. Self-determination theory in HCI games research: Current uses and open questions. In *CHI*.

Karthik Valmeekam, Sarath Sreedharan, Matthew Marquez, Alberto Olmo, and Subbarao Kambhampati. 2023. On the planning abilities of large language models (a critical investigation with a proposed benchmark). *arXiv*.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. 2017. Attention is all you need. In *NIPS*.

Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michae?l Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Remi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wunsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575:350–354.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023a. Voyager: An open-ended embodied agent with large language models. *arXiv*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. Self-Instruct: Aligning language model with self generated instructions. In *ACL*.

Zekun Wang, Ge Zhang, Kexin Yang, Ning Shi, Wangchunshu Zhou, Shaochun Hao, Guangzheng Xiong, Yizhi Li, Mong Yuan Sim, Xiuying Chen, Qingqing Zhu, Zhenzhu Yang, Adam Nik, Qi Liu, Chenghua Lin, Shi Wang, Ruibo Liu, Wenhu Chen, Ke Xu, Dayiheng Liu, Yike Guo, and Jie Fu. 2023c. Interactive natural language processing. *arXiv*.

Zihao Wang, Shaofei Cai, Anji Liu, Xiaojian Ma, and Yitao Liang. 2023d. Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. *arXiv*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-Thought prompting elicits reasoning in large language models. In *NeurIPS*.

Jerry Wei, Le Hou, Andrew Lampinen, Xiangning Chen, Da Huang, Yi Tay, Xinyun Chen, Yifeng Lu, Denny Zhou, Tengyu Ma, and Quoc V. Le. 2023. Symbol tuning improves in-context learning in language models. *arXiv*.

Jenna Wiens, Suchi Saria, Mark Sendak, Marzyeh Ghassemi, Vincent X. Liu, Finale Doshi-Velez, Kenneth Jung, Katherine Heller, David Kale, Mohammed Saeed, Pilar N. Ossorio, Sonoo Thadaney-Israni, and Anna Goldenberg. 2019. Do no harm: a roadmap for responsible machine learning for health care. *Nature Medicine*, 25:1337–1340.

Jeannette M. Wing. 2021. Trustworthy AI. *Communications of the ACM*, 64(10).

Christian Wirth, Riad Akrour, Gerhard Neumann, and Johannes Fürnkranz. 2017. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18:1–46.

Lionel Wong, Gabriel Grand, Alexander K. Lew, Noah D. Goodman, Vikash K. Mansinghka, Jacob Andreas, and Joshua B. Tenenbaum. 2023. From word models to world models: Translating from natural language to the probabilistic language of thought. *arXiv*.

Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. 2023a. Visual ChatGPT: Talking, drawing and editing with visual foundation models. *arXiv*.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023b. BloombergGPT: A large language model for finance. *arXiv*.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A. Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023c. Fine-grained human feedback gives better rewards for language model training. *arXiv*.

Peter R. Wurman, Samuel Barrett, Kenta Kawamoto, James MacGlashan, Kaushik Subramanian, Thomas J. Walsh, Roberto Capobianco, Alisa Devlic, Franziska Eckert, Florian Fuchs, Leilani Gilpin, Piyush Khandelwal, Varun Kompella, HaoChih Lin, Patrick MacAlpine, Declan Oller, Takuma Seno, Craig Sherstan, Michael D. Thomure, Houmehr Aghabozorgi, Leon Barrett, Rory Douglas, Dion Whitehead, Peter Dürr, Peter Stone, Michael Spranger, and Hiroaki Kitano. 2022. Outracing champion gran turismo drivers with deep reinforcement learning. *Nature*, 602(7896):223–228.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023a. FinGPT: Open-source financial large language models. *arXiv*.

Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Bing Yin, and Xia Hu. 2023b. Harnessing the power of LLMs in practice: A survey on ChatGPT and beyond. *arXiv*.

John Yang, Akshara Prabhakar, Karthik Narasimhan, and Shunyu Yao. 2023c. InterCode: Standardizing and benchmarking interactive coding with execution feedback. *arXiv*.

Kaiyu Yang, Aidan M. Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan Prenger, and Anima Anandkumar. 2023d. LeanDojo: Theorem proving with retrieval-augmented language models. *arXiv*.

Sherry Yang, Ofir Nachum, Yilun Du, Jason Wei, Pieter Abbeel, and Dale Schuurmans. 2023e. Foundation models for decision making: Problems, methods, and opportunities. *arXiv*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. *arXiv*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023b. ReAct: Synergizing reasoning and acting in language models. In *ICLR*.

Yang You. 2023. ColossalChat: An open-source solution for cloning ChatGPT with a complete RLHF pipeline. https://bit.ly/42ZTwW4.

Tianhe Yu, Ted Xiao, Austin Stone, Jonathan Tompson, Anthony Brohan, Su Wang, Jaspiar Singh, Clayton Tan, Dee M, Jodilyn Peralta, Brian Ichter, Karol Hausman, and Fei Xia. 2023. Scaling robot learning with semantically imagined experience. *arXiv*.

Haoqi Yuan, Chi Zhang, Hongcheng Wang, Feiyang Xie, Penglin Cai, Hao Dong, and Zongqing Lu. 2023. Plan4mc: Skill reinforcement learning and planning for open-world minecraft tasks. *arXiv*.

J.D. Zamfirescu-Pereira, Richmond Y. Wong, Bjoern Hartmann, and Qian Yang. 2023. Why Johnny can't prompt: How non-AI experts try (and fail) to design LLM prompts. In *CHI*.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. GLM-130B: An open bilingual pre-trained model. In *ICLR*.

Kai Zhang, Jun Yu, Zhiling Yan, Yixin Liu, Eashan Adhikarla, Sunyang Fu, Xun Chen, Chen Chen, Yuyin Zhou, Xiang Li, Lifang He, Brian D. Davison, Quanzheng Li, Yong Chen, Hongfang Liu, and Lichao Sun. 2023a. BiomedGPT: A unified and generalist biomedical generative pre-trained transformer for vision, language, and multimodal tasks. *arXiv*.

Lvmin Zhang and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. *arXiv*.

Ruohan Zhang, Faraz Torabi, Garrett Warnell, and Peter Stone. 2021. Recent advances in leveraging human guidance for sequential decision-making tasks. *Autonomous Agents and Multi-Agent Systems*, 35(31).

Shizhuo Dylan Zhang, Curt Tigges, Stella Biderman, Maxim Raginsky, and Talia Ringer. 2023b. Can transformers learn to solve problems recursively? *arXiv*.

Shun Zhang, Zhenfang Chen, Yikang Shen, Mingyu Ding, Joshua B. Tenenbaum, and Chuang Gan. 2023c. Planning with large language models for code generation. In *ICLR*.

Tianjun Zhang, Xuezhi Wang, Denny Zhou, Dale Schuurmans, and Joseph E. Gonzalez. 2023d. TEMPERA: Test-time prompting via reinforcement learning. In *ICLR*.

Yuhui Zhang, Michihiro Yasunaga, Zhengping Zhou, Jeff Z. HaoChen, James Zou, Percy Liang, and Serena Yeung. 2023e. Beyond positive scaling: How negation impacts scaling trends of language models. In *ACL*.

Wenshuai Zhao, Jorge Pena Queralta, and Tomi Westerlund. 2020. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *IEEE Symposium Series on Computational Intelligence (SSCI)*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023a. LIMA: Less is more for alignment. *arXiv*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023b. Least-to-most prompting enables complex reasoning in large language models. In *ICLR*.

Banghua Zhu, Hiteshi Sharma, Felipe Vieira Frujeri, Shi Dong, Chenguang Zhu, Michael I. Jordan, and Jiantao Jiao. 2023. Fine-tuning language models with advantage-induced policy alignment. *arXiv*.

Brian D. Ziebart, Andrew Maas, J.Andrew Bagnell, and Anind K. Dey. 2008. Maximum entropy inverse reinforcement learning. In *AAAI*.