RESEARCH PLAN

Deciphering Deep Learning

How does stochastic gradient descent implement algorithms?

1 Summary of the research plan

Deep learning – optimizing deep neural networks (DNNs) with stochastic gradient descent (SGD) – is the technological breakthrough powering both specialized systems like Alphafold for protein folding [28], generally capable large language model (LLM)[1] based assistants [1], as well as many other learning-based systems that have been built within the last two decades [32, 54, 48]. Almost by construction the algorithms encoded in the hundreds of billions of parameters[2] of DNNs are not human interpretable by default. Thus, DNNs are often referred to as "black-boxes".

Mechanistic interpretability (MI), an emerging field dedicated to reverse-engineering DNNs, has begun opening up these "black-boxes". Pioneering work has discovered fundamental circuits[3] for in-context learning (ICL) [43], methods to edit factual knowledge [36], and more [61, 10, 7]. However, the field is young and has research gaps even for well-studied phenomena. For example, it is still unclear how exactly LLM pretraining using next-token[4] prediction leads to ICL capabilities, how it leads to language-agnostic representations [68], and how any of this is implemented within LLMs. The same goes for how instruction tuning on a variety of tasks leads to generalization to unseen tasks [62], or what mechanistically happens when an LLM gets tuned to become a reasoner [11]. Simultaneously, as the field matures, it raises the question of whether MI tools help to extract scientific insights from specialized models like Alphafold.

The goal of my research plan is to leverage and create MI tools [41, 19, 10, 35, 3] to take high-resolution measurements of these phenomena in an effort to gain novel insights into DNNs' internal mechanisms and their learning dynamics. To do so, I will combine controlled studies of toy models with the analysis of large real-world models like in *my previous work* [68, 13, 37, 57, 5]. This will result in novel MI methods (in particular, for reasoners and Alphafold), infrastructure, and implementations supporting such analyses, interactive visualizations, and novel insights packaged into research papers.

Importantly, the current research landscape with an increasing number of powerful DNNs, their training data, code, and intermediate checkpoints becoming publicly available [20] for the first time allows for this kind of research without incurring prohibitive computational and human labor costs. Prof. David Bau's NDIF[5] team also plays a key role in building the infrastructure to support MI research. In other words, the stars are aligned *now* for executing my research plan in Prof. Bau's lab.

Conducting this research will add to the empirical foundation for the deep learning theory of the future. A better mechanistic understanding of DNNs will unlock novel data-efficient ways for us to tailor them to our needs, e.g., to make them multicultural and safe. Further, it will help advance MI methods towards a point at which they allow us to extract scientific insights from models like Alphafold.

[1] LLMs are transformer architecture [60] based DNNs trained on language.
[2] We have learned bigger is better in the case of DNNs [30].
[3] Circuits refer to a subnetwork within a DNN explaining its behavior on a specific task (e.g., on a specific dataset).
[4] Tokens are usually sub-words.
[5] https://ndif.us

2 Research plan

2.1 Current state of research in the field

LLM-based systems like instruction and reasoning models are trained over multiple stages. The first stage is called *self-supervised pretraining* and consists of teaching the LLM to predict the next-token in an internet-scale corpus. Remarkably, LLMs of a sufficient size trained in this way exhibit ICL capabilities [6] that allow them to learn from their input context (e.g., when a few input-output pairs are provided) without updating model parameters. Additionally, they learn multilingual representations [68] facilitating cross-lingual transfer [33] in the subsequent stages. Next, this is followed by *supervised finetuning* in which the LLM is tuned on a curated dataset teaching it the desired behavior, e.g., instruction-following [62] or reasoning [11]. Remarkably, finetuning on a large set of diverse tasks leads to generalization to unseen tasks. Finally, the target behavior is refined using *reinforcement learning* (RL). RL allows to exceed the quality of the finetuning dataset and instilling additional behaviors such as preventing harmful outputs [11, 1].

Mechanistic interpretability. The outlined training procedure raises the question whether LLMs' remarkable capabilities are genuine or an artifact of, e.g., data contamination. How exactly are they implemented within the LLMs' internal computations and how do they form during training? Reverse engineering LLMs' internal algorithms using MI presents a promising approach to address such questions from a new angle grounded in experiments. E.g., useful MI techniques include intermediate decoding [41, 19], sparse autoencoders for finding interpretable features [14, 3, 10], activation patching-based causal analysis [36, 58], and automatic circuit finding [35].

2.1.1 Open questions

In the following, I outline my research questions and the state of the field.

RQ1: How do ICL capabilities form? The fact that next-token prediction leads to ICL capabilities is a puzzling finding. Olsson et al. [43] have identified induction circuits implementing the behavior "AB ... A $\rightarrow$ B", in which A and B are tokens, and fuzzy versions of it as a fundamental mechanism underlying ICL. Prof. Bau's lab has identified function vectors [58] (in parallel with [21]), encoding which task should be executed on the current token, as another component to ICL. Yang et al. [69] have studied task vectors' formation in toy models trained on, e.g., regression tasks, using next-token prediction and provided a training method encouraging their formation. Bigoulaeva et al. [2] have shown that instruction tuning and ICL performance are correlated, which suggests that instruction-following circuits reuse ICL ones.

RQ2: How do multilingual representations form? In my own MI work, I found evidence for LLMs possessing language-agnostic representations [68, 13, 5]. However, little is known about their training dynamics. To the best of my knowledge, the most closely related work is [47], which tracks the evolution of a multilingual n-gram circuit across training.

RQ3: How does instruction tuning lead to broad generalization? Jain et al. [26] have started to investigate this phenomenon in toy models trained on synthetic settings and find that circuits are heavily reused. This finding is consistent with our findings in [37]. Leveraging circuit reuse might be a promising angle for studying **RQ3** in real models. Additionally, a recent variation of sparse autoencoders, called cross-coders [34], allows for the systematic comparison of base and finetuned models.

RQ4: How does a LLM become a reasoner? General purpose reasoners like OpenAI's o1 model [25] are the major breakthrough of last year. By scaling up inference compute [56] they achieve superior performance across a wide range of tasks. Notably, o3 a recent iteration of o1 led to a 30% jump in performance on the ARC-AGI benchmark [9], on which LLM-based solutions barely progressed for

the last three years. Up until the recent release of the Deepseek's R1 models [11] this month (January 2025), there was no such model openly available. Thus, while there are many interesting questions about reasoner's internals to ask, they only recently can be studied. Since reasoners search over multiple chains-of-thoughts (CoTs)[6], their outputs are long and provide unique challenges for MI. Most MI research still focuses on the next-token prediction setting. Cabannes et al. [7] study in toy models and might provide a good starting point for my analysis of the R1 models.

RQ5: Is it possible to extract scientific insights from Alphafold? As the field of MI matures it becomes conceivable that we will be able to turn powerful DNNs trained on scientific data into cheat sheets for the underlying phenomena. Schut et al. [50] show that it is possible to extract super-human chess concepts teachable to grandmasters from AlphaGo's internal representations [54]. For simpler protein language models (PLMs) it has been shown that they can learn interpretable concepts [55, 24].

2.2 Current state of my own research

LLM interpretability. My works on MI for LLMs are the most relevant for this research plan. In [68] I investigated whether multilingual LLMs pretrained on English-dominated text learn to leverage English as internal pivot language. While I did not find that English is used as a pivot in the literal sense, for simple word-level tasks, I found that independent of the prompt language the English solution can be decoded from the middle layers using the logit lens [41]. I concluded that this is likely caused by the LLMs learning language-agnostic concept representations, which I followed up on with a causal analysis using activation patching in [13]. Simultaneously, we leveraged our finding to extend contrastive decoding to multilingual inputs in [70]. In our most recent work [5] we leveraged sparse autoencoders to study grammatical concepts in the multilingual setting. We found that independent of the language composition of the pretraining corpus, highly multilingual grammatical features are learned.

In a different stream of work [37] we investigated how LLMs balance between the information provided in the context and their prior knowledge learned during pretraining in simple question-answering settings. This project was led by Julian Minder, who I advised on a master thesis on LLM circuit dynamics. Leveraging insights from his thesis, namely, that LLMs' circuits don't change too much during finetuning and are heavily reused, we identified an one-dimensional subspace causally controlling context sensitivity. Importantly, we showed that the subspace transfers back to both base and instruct models. A similar approach leveraging similarities of close-by checkpoints should be applicable to **RQ1**–**RQ4**.

In all of our papers, we were able to replicate our main findings across a wide range of tasks and models differing in sizes and model families. The universality of our results so far makes me optimistic about my future research in this area and its potential.

Text-to-image interpretability. In [57] we extended sparse autoencoders to SDXL Turbo, a recent text-to-image model. In this work, we showed that MI techniques developed for LLMs can be adapted to other modalities, which will be important for **RQ5**. In doing so, we found that different parts of SDXL Turbo specialize in distinct roles like image composition, adding local details, or adding style.

Other research. I contributed to works on constrained decoding [17, 18] ensuring LLM outputs to comply with context-free grammars. Further, I did a PhD on machine learning on data indexed by powersets, lattices, and partially ordered sets [64]. In the scope of my PhD research, I extended my PhD advisors' theory of algebraic signal processing to new domains and problems [65, 38, 44, 45, 51, 52], leveraged it to build various learning techniques based on our own notion Fourier-sparsity and used them to build applications [66, 63, 67, 53, 8, 39]. Other than that, I participated and ranked first in the student leaderboard in a machine learning for combinatorial optimization competition [59].

[6] CoT refers to natural language reasoning trails consisting of intermediate results and their derivations.

2.3 Detailed research plan

I will collaborate with Prof. David Bau and his research group renowned in the area of MI. The group has multiple ongoing research directions relevant to my research plan (see below). In addition to their MI expertise, the lab provides a unique computational infrastructure called national deep inference fabric (NDIF). NDIF serves as a back-end to their `nnsight` library [15] that allows access to internal states of DNNs. NDIF and `nnsight` together allow for the analysis of even the largest openly available models like Llama-3.1-405B [12].

In order to approach my research questions (**RQ1** – **RQ5**) outlined in Sec. 2.1, I will break them down into four projects, **P1, P2, P3** and **P4**. In **P1** (Sec. 2.3.1) I will create minimal examples of the target learning dynamics. In **P2** (Sec. 2.3.2) I will investigate them in real models, with a focus on broad generalization. In **P3** (Sec. 2.3.3) I will investigate LLM-based reasoners. Finally, in **P4** (Sec. 2.3.4) I will investigate the feasibility of concept extraction for Alphafold.

2.3.1 Project 1: Minimal examples of target dynamics

MI analyses of toy models have provided fresh angles onto ICL [43], superposition [14], and grokking [40]. The goal of this project is to create minimal examples of the target learning dynamics as well as to understand them mechanistically. In particular, I want to do so for the creation of reasoners (milestone **M1.1**) and for broad generalization akin to what we observe in instruction tuning (milestone **M1.2**). Using board games it is straightforward to train reasoners by following the recipe of, e.g., AlphaGo [54]. This also has been done in the next-token prediction setting we are interested in to study inference compute scaling laws [27]. In this setting many natural MI questions come up: do the models learn to look ahead, to self-verify, to model the reward, etc.? How do mechanisms found in the reasoner relate to its base model?

Next, we can add the construction from [23] that transforms the Orthello board game into a testbed of multilingualism by creating different sequence representations of game trajectories. The resulting setting allows for the study of broad generalization and whether language-agnostic representations are the key mechanism behind it.

Methodology. Highly controlled settings like this allow for the development of automated interpretability techniques. In particular, sub-component probing [4, 16] achieved via manually decomposing a problem into its sub-components is a promising technique. For board games, the following sub-components directly come to mind: (1.) board state, (2.) current player, (3.) legal next moves, (4.) value of the board, etc. If it is possible to train accurate sub-component probes, this hints at the existence of corresponding mechanisms within the model. Additionally, tracking probing performance across checkpoints should provide hints about how these mechanisms form. Importantly, minimal examples allow for short development cycles. Further, experiments will be cheap enough to verify the robustness of our results through repetitions and sensitivity analysis with respect to hyperparameters.

Alternatives. To study the broad generalization question and its connection to ICL also other constructions come to mind. E.g., as mentioned ICL has been studied in the context of simple regression tasks [69]. Additionally, Prof. Bau's lab is currently setting up toy models displaying ICL capabilities by letting them fill out Cayley tables of groups with randomly relabeled elements. Another more realistic testbed for multilingualism is the creation of cloned languages [49]. The recent paper [29] provides nice ideas for the construction of perturbed cloned languages. A more realistic setting for general reasoners potentially can be achieved by teaching language models of modest size how to execute code step by step by training them on intermediate outputs derived from a debugger.

2.3.2 Project 2: Broad generalization in the wild

Intermediate checkpoints of powerful models like OLMO [20, 42] allow us to study **RQ1** and **RQ2** in real LLMs without incurring prohibitive costs. On top of that, their accompanying training code and instruction-following datasets allow for the creation of intermediate checkpoints for the study of **RQ3**.

Pretraining. Tools for the tracking the presence of ICL capabilities (**RQ1**) can derived from Prof. Bau's work [58] and from [43]. Similarly, tools for tracking the presence of language-agnostic representations (**RQ2**) can be derived from my work [68, 13, 5]. Evaluating the intermediate checkpoints using these tools should allow for the identification of phase transitions in learning trajectories indicative of the formation of ICL capabilities and language-agnostic representations. Next, the analysis of these phase transitions will inform the creation of automatic MI pipelines akin to the sub-component tracking described in Sec. 2.3.1. The resulting pipeline should allow for higher resolution measurements of the circuits' formation (milestone **M2.1**).

Finetuning. As a next step, we are going to create our own intermediate checkpoints for multiple instruction tuning trajectories for studying **RQ3**. In order to do so, we control how many tasks we introduce as well as when and from which task-families (math, code, question answering, natural language processing tasks, etc.). Since instruction tuning has not been much studied through the lens of MI, we cannot rely on existing results. Thus, I consider it already a milestone to better understand how instruction-following capabilities work and how they form (**M2.2**). It would be interesting to find out whether instruction-following models mainly reuse existing mechanisms from the base model such as induction circuits and function vectors or whether they do more than that. Also it would be interesting to find whether there is a shared general instruction-following circuit or many task-specific ones.

Additional methods. Additional methods facilitating the analysis of different checkpoints of the same training trajectory include (1.) the systematic comparison of next-token distributions across multiple tokens [46], which allows to find key token positions for which the base and finetuned models disagree; (2.) a more costly method of training cross-coders that are aimed at identifying both shared as well as different features between the compared checkpoints [34]; and (3.) the analysis of gradient updates directly by extending [31]; As the training progresses and the models' internal states become meaningful, so should also their gradients.

2.3.3 Project 3: Reasoners and meta-reasoners

The recent open-source release of Deepseek-R1 and its distillations into smaller models [11] opens up many interesting research directions. Due to the unique requirements of analyzing long multi-token outputs as they are produced by CoT I expect that we will need to create novel MI methods (milestone **M3.1**). In fact, in an ongoing project with researchers from EPFL and ETH Zurich, in which we analyze LLMs on a theory of mind tasks we are currently held back by the lack of availability of methods that would allow us to track the evolution of the true world state along with various belief states of the different actors along a CoT produced by a model. For the creation of these methods comparing base and CoT model next-token distributions akin to [46] could allow to identify interesting token positions and thereby reduce the complexity of the analysis. Additionally, the work by [7] on iteration circuits describing how CoT works in toy models will be a good starting point.

Next, I am going to investigate whether I can find human interpretable elements of reasoning within the internal states and computations of reasoners. E.g., mechanisms for recognizing which algorithm to run and how to run it; mechanisms for deciding the next step, e.g., look ahead mechanisms; mechanisms for self-verification; mechanisms for reward modeling; In order to approach this search, I will work together with Alex Loftus a PhD student in Prof. Bau's lab and leverage the full MI toolkit (in particular, the methods from Sec. 2.3.1 and Sec. 2.3.2).

Finally, I am curious about whether the reasoners' capabilities can be explained via the capabilities of its base model (**RQ4**). E.g., if we find elements of reasoning in Deepseek-R1, can we trace their origins back to its base model or does reasoning tuning lead to something fundamentally new and different? The second milestone of this project is to find elements of reasoning in reasoners and to relate them to the corresponding base model (**M3.2**).

2.3.4 Project 4: Super-human concepts in Alphafold

Despite the remarkable success of LLMs some of the most groundbreaking accomplishments driven by deep learning are achieved on non-verbal scientific domains. Alphafold [28] achieving super-human[7] performance on the protein folding task is the prime example of this. Instead of manually implementing a protein folding algorithm (which up until today is an impossible task), they *discovered* one using SGD – Alphafold, a DNN that maps from an augmented amino-acid sequence of a protein to the 3D structure of its crystallized form. However, whatever super-human insight "SGD might have had" when implementing this protein folding algorithm remains encrypted in the parameters of the Alphafold DNN.

My final project aims to change that and to extract super-human insights from Alphafold (**RQ5**). In order to achieve this ambitious goal, we are going to follow the footsteps of the pioneering work of Schut et al. [50], which achieved the successful extraction of super-human chess concepts from AlphaGo that were also teachable to chess grandmasters. While the Alphafold uses a specialized architecture, e.g., using triangle attention to iteratively refine the representations of the distance matrix encoding spatial relationships between pairs of amino acids, from a MI point of view it still should be amenable to the concept extraction techniques used in [50]. Additionally, these techniques can be updated to leverage the most recent MI breakthroughs such as sparse autoencoders [3, 10]. My experience in porting sparse autoencoders to few-step text-to-image diffusion models [57] will be helpful for that. To set up and test our Alphafold concept discovery pipeline we can check whether Alphafold's representations encode known protein properties (milestone **M4.1**). To create datasets for that, we can follow the construction of [24] who build a dataset encompassing more than 700 protein properties.

Next, some of the concept filtering techniques akin to [50] based on intermediate training checkpoints, which allow to filter for concepts forming at different stages in training and for concepts that are teachable to weaker checkpoints, can be used to find potentially super-human concepts (e.g., formed late in training). To address the additional challenge of this domain, which is that both inputs (amino acid sequences) and outputs (3D protein structures) are not easy to interpret, we will have to create high quality visualizations of the found concepts. For each concept, we will collect all of the amino acid sequences that result in the presence of the concept within Alphafold's representations as well as all of the resulting 3D structures. For each protein, we will also highlight the most relevant subsequences in the sequence data and substructures in the 3D data. Finally, we are going to iteratively analyze and refine these visualizations together with biologists (milestone **M4.2**).

Risks and alternatives. Since I am not a trained biologist, for this project to succeed it will be important to find collaborators with a solid biological background. Prof. Bau is excited about this direction too and willing to hire a PhD student with the relevant background. Since **M4.2** may fail due to unforeseen complications (e.g., difficulties finding collaborators) or may fail to deliver the desired super-human insights, I have a back-up direction. In short, this direction consists of extending our work Surkov et al. [57] to handle diffusion models with many denoising steps and to recent models like FLUX [8] or text-to-video models [22]. Prof. Bau's lab also has a relevant work-stream led by Rohit Gandikota.

[7] Not only humans obviously cannot fold proteins outside of their body, but also they cannot explicitly write an efficient algorithm that does so.

[8] https://huggingface.co/black-forest-labs/FLUX.1-dev

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Irina Bigoulaeva, Harish Tayyar Madabushi, and Iryna Gurevych. The inherent limits of pretrained LLMs: The unexpected convergence of instruction tuning and in-context learning capabilities. *arXiv preprint arXiv:2501.08716*, 2025.

[3] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, and Adam et al. Jermyn. Towards monosemanticity: Decomposing language models dictionary learning. *Transformer Circuits*, October 2023. URL https://transformer-circuits.pub/2023/monosemantic-features.

[4] Jannik Brinkmann, Abhay Sheshadri, Victor Levoso, Paul Swoboda, and Christian Bartelt. A mechanistic analysis of a transformer trained on a symbolic multi-step reasoning task. *arXiv preprint arXiv:2402.11917*, 2024.

[5] Jannik Brinkmann, Chris Wendler, Christian Bartelt, and Aaron Mueller. Large language models share representations of latent grammatical concepts across typologically diverse languages. *To appear in NAACL 2025, arXiv preprint arXiv:2501.06346*, 2025.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[7] Vivien Cabannes, Charles Arnal, Wassim Bouaziz, Alice Yang, Francois Charton, and Julia Kempe. Iteration head: A mechanistic study of chain-of-thought. *arXiv preprint arXiv:2406.02128*, 2024.

[8] Mathieu Chevalley, Jacob Sackett-Sanders, Yusuf Roohani, Pascal Notin, Artemy Bakulin, Dariusz Brzezinski, Kaiwen Deng, Yuanfang Guan, Justin Hong, Michael Ibrahim, et al. The CausalBench challenge: A machine learning contest for gene network inference from single-cell perturbation data. *arXiv preprint arXiv:2308.15395*, 2023.

[9] Francois Chollet, Mike Knoop, Gregory Kamradt, and Bryan Landers. Arc prize 2024: Technical report. *arXiv preprint arXiv:2412.04604*, 2024.

[10] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

[11] Team DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning. https://github.com/deepseek-ai/DeepSeek-R1/blob/main/DeepSeek_R1.pdf, 2025. Accessed: 2025-01-24.

[12] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[13] Clément Dumas, Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. Separating tongue from thought: Activation patching reveals language-agnostic concept representations in transformers. In *ICML Mechanistic Interpretability Workshop*, 2024.

[14] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits*, September 2022. URL https://transformer-circuits.pub/2022/toy_model/index.html.

[15] Jaden Fiotto-Kaufman, Alexander R Loftus, Eric Todd, Jannik Brinkmann, Caden Juang, Koyena Pal, Can Rager, Aaron Mueller, Samuel Marks, Arnab Sen Sharma, et al. Nnsight and NDIF: Democratizing access to foundation model internals. *arXiv preprint arXiv:2407.14561*, 2024.

[16] Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. Finding alignments between interpretable causal variables and distributed neural representations. In *Causal Learning and Reasoning*, pages 160–187. PMLR, 2024.

[17] Saibo Geng, Berkay Döner, Chris Wendler, Martin Josifoski, and Robert West. Sketch-guided constrained decoding for boosting blackbox large language models without logit access. In *Proceedings of the Association for Computational Linguistics*, 2024.

[18] Saibo Geng, Sankalp Gambhir, Chris Wendler, and Robert West. Byte BPE tokenization as an inverse string homomorphism. *arXiv preprint arXiv:2412.03160*, 2024.

[19] Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscope: A unifying framework for inspecting hidden representations of language models. *arXiv preprint arXiv:2401.06102*, 2024.

[20] Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. Olmo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838*, 2024.

[21] Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[22] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. In *The Eleventh International Conference on Learning Representations*, 2022.

[23] Tianze Hua, Tian Yun, and Ellie Pavlick. mOthello: When do cross-lingual representation alignment and cross-lingual transfer emerge in multilingual models? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1585–1598, 2024.

[24] Aya Abdelsalam Ismail, Tuomas Oikarinen, Amy Wang, Julius Adebayo, Samuel Stanton, Taylor Joren, Joseph Kleinhenz, Allen Goodman, Héctor Corrada Bravo, Kyunghyun Cho, et al. Concept bottleneck language models for protein design. *arXiv preprint arXiv:2411.06090*, 2024.

[25] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[26] Samyak Jain, Robert Kirk, Ekdeep Singh Lubana, Robert P Dick, Hidenori Tanaka, Tim Rocktäschel, Edward Grefenstette, and David Krueger. Mechanistically analyzing the effects of fine-tuning on procedurally defined tasks. In *The Twelfth International Conference on Learning Representations*, 2024.

[27] Andy L Jones. Scaling scaling laws with board games. *arXiv preprint arXiv:2104.03113*, 2021.

[28] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

[29] Julie Kallini, Isabel Papadimitriou, Richard Futrell, Kyle Mahowald, and Christopher Potts. Mission: Impossible language models. *arXiv preprint arXiv:2401.06416*, 2024.

[30] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[31] Shahar Katz, Yonatan Belinkov, Mor Geva, and Lior Wolf. Backward lens: Projecting language model gradients into the vocabulary space. *arXiv preprint arXiv:2402.12865*, 2024.

[32] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[33] Xiaochen Li, Zheng Xin Yong, and Stephen Bach. Preference tuning for toxicity mitigation generalizes across languages. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13422–13440, 2024.

[34] Jack Lindsey, Adly Templeton, Jonathan Marcus, Thomas Conerly, Joshua Batson, and Christopher Olah. Sparse crosscoders for cross-layer features and model diffing. *Transformer Circuits*, 2024. URL https://transformer-circuits.pub/2024/crosscoders/index.html.

[35] Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *arXiv preprint arXiv:2403.19647*, 2024.

[36] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022.

[37] Julian Minder, Kevin Du, Niklas Stoehr, Giovanni Monea, Chris Wendler, Robert West, and Ryan Cotterell. Controllable context sensitivity and the knob behind it. *To appear in ICLR 2025, arXiv preprint arXiv:2411.07404*, 2024.

[38] Panagiotis Misiakos, Chris Wendler, and Markus Püschel. Diagonalizable shift and filters for directed graphs based on the Jordan-Chevalley decomposition. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5635–5639. IEEE, 2020.

[39] Panagiotis Misiakos, Chris Wendler, and Markus Püschel. Learning DAGs from data with few root causes. *Advances in Neural Information Processing Systems*, 36, 2024.

[40] Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*, 2023.

[41] Nostalgebraist. Interpreting GPT: The logit lens. LessWrong, 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

[42] Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

[43] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

[44] Markus Püschel and Chris Wendler. Discrete signal processing with set functions. *IEEE Transactions on Signal Processing*, 69:1039–1053, 2020.

[45] Markus Püschel, Bastian Seifert, and Chris Wendler. Discrete signal processing on meet/join lattices. *IEEE Transactions on Signal Processing*, 69:3571–3584, 2021.

[46] Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep. *arXiv preprint arXiv:2406.05946*, 2024.

[47] Lucia Quirke, Lovis Heindrich, Wes Gurnee, and Neel Nanda. Training dynamics of contextual n-grams in language models. *arXiv preprint arXiv:2311.00863*, 2023.

[48] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[49] Anton Schäfer, Shauli Ravfogel, Thomas Hofmann, Tiago Pimentel, and Imanol Schlag. Language imbalance can boost cross-lingual generalisation. *arXiv preprint arXiv:2404.07982*, 2024.

[50] Lisa Schut, Nenad Tomasev, Tom McGrath, Demis Hassabis, Ulrich Paquet, and Been Kim. Bridging the human-ai knowledge gap: Concept discovery and transfer in alphazero. *arXiv preprint arXiv:2310.16410*, 2023.

[51] Bastian Seifert, Chris Wendler, and Markus Püschel. Wiener filter on meet/join lattices. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5355–5359. IEEE, 2021.

[52] Bastian Seifert, Chris Wendler, and Markus Püschel. Causal Fourier analysis on directed acyclic graphs and posets. *IEEE Transactions on Signal Processing*, 2022.

[53] Bastian Seifert, Chris Wendler, and Markus Püschel. Learning Fourier-sparse functions on dags. In *ICLR2022 Workshop on the Elements of Reasoning: Objects, Structure and Causality*, 2022.

[54] 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. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[55] Elana Simon and James Zou. InterPLM: Discovering interpretable features in protein language models via sparse autoencoders. *bioRxiv*, pages 2024–11, 2024.

[56] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

[57] Viacheslav Surkov, Chris Wendler, Mikhail Terekhov, Justin Deschenaux, Robert West, and Caglar Gulcehre. Unpacking SDXL turbo: Interpreting text-to-image models with sparse autoencoders. *arXiv preprint arXiv:2410.22366*, 2024.

[58] Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[59] Romeo Valentin, Claudio Ferrari, Jérémy Scheurer, Andisheh Amrollahi, Chris Wendler, and Max B Paulus. Instance-wise algorithm configuration with graph neural networks. *Rank 3 in NeurIPS 2021 Machine Learning for Combinatorial Optimization Competition*, 2022.

[60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[61] Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. *arXiv preprint arXiv:2211.00593*, 2022.

[62] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

[63] Jakob Weissteiner, Chris Wendler, Sven Seuken, Ben Lubin, and Markus Püschel. Fourier analysis-based iterative combinatorial auctions. In *Thirty-First International Joint Conference on Artificial Intelligence*, 2020.

[64] Chris Wendler. *Machine learning on non-Euclidean domains: Powersets, lattices, posets.* PhD thesis, ETH Zurich, 2023.

[65] Chris Wendler and Markus Püschel. Sampling signals on meet/join lattices. In *Global Conference on Signal and Information Processing (GlobalSIP)*, 2019.

[66] Chris Wendler, Dan Alistarh, and Markus Püschel. Powerset convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 927–938, 2019.

[67] Chris Wendler, Andisheh Amrollahi, Bastian Seifert, Andreas Krause, and Markus Püschel. Learning set functions that are sparse in non-orthogonal Fourier bases. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10283–10292, 2021.

[68] Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. Do Llamas work in English? on the latent language of multilingual transformers. In *Proceedings of the Association for Computational Linguistics*, 2024.

[69] Liu Yang, Ziqian Lin, Kangwook Lee, Dimitris Papailiopoulos, and Robert Nowak. Task vectors in in-context learning: Emergence, formation, and benefit. *arXiv preprint arXiv:2501.09240*, 2025.

[70] Wenhao Zhu, Sizhe Liu, Shujian Huang, Shuaijie She, Chris Wendler, and Jiajun Chen. Multilingual contrastive decoding via language-agnostic layers skipping. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8775–8782, 2024.