# Recent history of the field

Mech interp is a young field, and the prevailing wisdom and most popular techniques change rapidly, which can be confusing to newcomers, and often leads to people doing projects on directions which I think are likely dead ends. This will no doubt go out of date fast, but as of June 2025, here's a brief overview of key changes in the field from my perspective

Caveats: This is pretty focused on the AGI Safety/industry side of the community's perspective, and emphasises the works that had the biggest impact on community focus, rather than originality/novelty.

- Until 2021, most mech interp was on weights-based reverse engineering of vision models
- Sparked in part by A Mathematical Framework for Transformer Circuits in Dec 2021, the field's focus largely shifted to LLMs, still thinking about weights-based reverse engineering, aiming for complete understanding. The measure of success was things like how well you could make a simple model to approximate the network's behaviour
- In my grokking work in mid 2022 I (almost) completely reverse engineered a modular addition network, and a lot of subsequent work tried to reverse engineer small models on algorithmic tasks, or studying training dynamics of algorithmic models.
    - I largely do not think this work has been very productive, and I recommend against this research direction apart from for learning
- Sparked by the Indirect Object Identification paper, the ROME paper, and Atticus Geiger's work, causal intervention based circuit finding became popular from late 2022 onwards
    - This was a big shift: rather than reverse-engineering from the weights to understand a model in general, it used causal interventions on a task-specific distribution to understand which model components/connections between them were useful for that task, in that narrow distribution.
        - But the focus remained on finding as correct and accurate an approximation as possible
    - I think the techniques, especially activation patching and comparing two inputs that are close together but differ in some key detail (contrastive pairs) remain highly valuable, and things like understanding which tokens and residuals streams key information lives on is useful
    - But I largely think that "finding important subgraphs of model components" is not that helpful, as the components aren't too interpretable
- Further work on polysemanticity (model components that do multiple unrelated roles) and superposition (models compressing in more concepts than they have directions, and dealing with this with error correction), notably Toy Models of Superposition in late 2022, became popular and suggested we needed deeper approaches to understand MLP layers.
    - An implicit idea here is the older idea of the linear representation hypothesis, from the old word2vec days, that concepts are represented as directions in activation space that combine additively

    - I basically think these ideas have held up well, though the superposition hypothesis is closer to a useful approximation than it literally being true that model activations are exactly a sparse linear combination of concept vectors
- With Towards Monosemanticity and Cunningham et al in late 2023, Sparse Autoencoders became a very popular technique for understanding LLM activations.
    - Many of the more prominent groups (and my team) focused on these in 2024
    - Opinions differ, but by and large I think the community (including myself) got too excited about SAEs, though they're a useful tool
- Opinions on SAEs vary a lot nowadays
    - There was a bunch of work using SAEs on real-world tasks to evaluate how well they captured important concepts, which I consider to have given fairly underwhelming results – this led to my team de-prioritising SAEs, which you can read about here
    - Overall, I think Sparse Autoencoders (SAEs) are a useful tool for unsupervised discovery and identifying unexpected phenomena inside models. However, if you know what you're looking for, simpler supervised methods using labeled data tend to be superior.
        - SAEs are often best used to help understand what's happening initially, while other methods are more useful for actually solving problems.
    - But others are excited about continuing to use them and deepen the approach with e.g. transcoders for analysing model circuitry or alternative approaches like attribution-based parameter decomposition
- As of June 2025 there's a more diverse range of directions being pursued. Examples include (but are not limited to):
    - Aiming for higher level qualitative understanding of what's happening inside a model, on verifiable tasks, e.g. Marks et al creating a model with hidden goals and having teams compete to find it with different techniques
        - Other work tries to study this "model biology" with more ambitious bets on specific techniques, such as Anthropic's cross-layer transcoder approach
    - Basic science of what's happening inside networks, both work in the vein of SAEs and beyond
    - Directly doing useful things on real world tasks, often with simple techniques like probes to e.g. detect harmful behaviour in models cheaply
- For newcomers interested in these various domains, my advice is to understand the broad ideas and techniques in each, but to focus on projects that study specific tasks, e.g. extracting information from model organisms, but not being wedded to one approach to these tasks. It's important to try multiple techniques, including simple and dumb baselines, rather than assuming any one approach is privileged.