## How my research interests have changed

My research interests and views have changed a fair bit over the last 12 to 18 months. This means that many of the topics I've done past work on aren't necessarily the topics I'm most excited about supervising future work on, which is often misunderstood by applicants. To help you choose a problem I'm likely to be interested in, here’s a summary of how my thinking has evolved. I’ve also written a separate doc with a very stylised recent history of the field and its trends, if you want a more zoomed out view

I’ve been finding the following taxonomy of the field useful:

1. **Basic science**: pursuing understanding for understanding's sake, and seeing how far we can push it
2. **Model biology**: trying to understand the qualitative, high-level properties of a model and its behavior
3. **Applied interpretability**: applying interpretability to interesting, real-world tasks, and beating baselines.

The overarching theme is a shift from **ambitious, purist goals** towards **pragmatism** and "**model biology**." I am most excited by projects in applied interpretability and model biology, and have a higher bar for 'basic science' work.

My general hope with detailing these shifts is to give you a better idea of what I'm excited about and what I'm looking for. If you hear all this and are like, that sounds really boring, I am no longer interested, then great - we probably wouldn't have been a good match! It's much better to learn that now than later. There's a bunch of other MATS mentors who'll be opening applications soon, hopefully one of them is more aligned with what you're looking for.

Further thoughts below

## Suggested Research Problems

The below are a bunch of recommendations for things I would be excited about. Strong applications often riff off of these ideas - coming up with their own approach, but along similar themes to the below. You should not feel constrained to the problems on this list, but hopefully it can serve as some guidance for the types of questions I'd be excited to see.

**Warning**: The ideas below have **not** been filtered for “I am confident someone could make progress on this in 20 hours”. Pick something where you have some idea of how to get started (or read around the field a bit and try to generate ideas and a sketch plan before picking a problem)

I broadly categorize my interests into three areas: **Applied Interpretability**, **Model Biology**, and **Basic Science**.

### Model Biology

This is about studying the high-level, qualitative properties of a model's behavior, treating it like a biological organism we're trying to understand. This is especially interesting for weird, emergent, or safety-relevant phenomena.

#### Understanding weird behaviour

- **Deep dives into mysterious behavior:** Take a weird model behavior (e.g., a model seemingly exhibiting self-preservation, blackmailing, or faking alignment) and try to understand what's happening.
  - **Tip**: Start with the smallest model that shows the behavior.
  - Example work from my team at GDM: We analyzed why models seemed to stop you from shutting them down (in a prior demo) and found that the model was actually just confused. We started with the simplest technique of reading the chain of thought to generate hypotheses and testing them causally by varying the prompt, and this just worked. Remember to start simple!
- **Debugging**: Can you take an example of model behaviour where it's behaving in unintended ways, understand what's happening, and ideally fix it? E.g. why it thinks 9.8 > 9.11

#### Reasoning Models

- **Chain of thought faithfulness**: An obvious safety strategy is “let’s just read the chain of thought and see if it thinks anything suspicious”. Can you find evidence for or against this being effective?
  - Can you learn anything by studying examples of unfaithful chain of thought? Eg from Arcuschin et al or Chen et al
  - Can you create situations where the model does use suspicious reasoning? E.g. by training the CoT to be distorted, or telling it that its chain of thought is being monitored - does it do the bad behaviour but without suspicious reasoning? How complex can the behaviour be?

        - I would love to see someone use synthetic document fine-tuning to train a model to believe that it should have unfaithful chain of thought and see if we can interpret what's happening.
    - Can you design good monitors or metrics for whether the chain of thought is telling us what we think?
    - Can we tell when a CoT was causally important for a model giving its answer?
        - This is a subtle concept to pin down – models may have a causally irrelevant CoT that gives the same answer. And editing the CoT throws the model off distribution, which may change its answer for unrelated reasons
    - What factors lead to different forms of "unfaithful" CoT? Can we distinguish them?
        - Deliberately rationalizing a preconceived answer (Arcuschin et al., Lindsey et al.)
        - Models changing their answer because of a hint but not admitting it (Chen et al.)
        - Models taking logical shortcuts in maths problems after getting stuck, to claim they've achieved a valid "proof" (Arcuschin et al.)
        - Models giving a reasonable chain of thought, but at the last minute "flipping" to a different final answer (Arcuschin et al.)
- **Thought anchors**: In Bogdan et al, my scholars present a paradigm for what mech interp could look like for reasoning models, where we study sentences as our main unit of analysis, and use tools like resampling to understand which sentences are important, and do causal interventions to understand the dependence between pairs of sentences. How can you extend and build on these techniques? Can you find anything interesting by using them? Can you find any weaknesses or limitations?
- **Steganography**: Can models encode information in their chain of thought? Ideally, in a way where the chain of thought is still plausible to us, but even in ways where we know it's encoded but we don't know what it means. Can you train a model to have reasoning we don't understand and use interpretability tools to decipher it? Relevant work
    - Note that this needs to allow the model to do tasks it couldn't do without a chain of thought to be interesting.

### Interesting phenomena

- **User models**: Chen et al shows that LLMs form surprisingly accurate and detailed models of the user, eg their gender, age, socioeconomic status, and level of education, and do this from very little information. They can find these with probes, and steer with these to change the model's actions in weird ways.
    - This is wild! What else can we learn here? What else do models represent about the user? How are these inferred? How else do they shape behaviour?
    - Do LLMs form dynamic models of users for attributes that vary across turns, eg emotion, what the user knows, etc.
        - As a stretch goal, do LLMs ever try to intentionally manipulate these? Eg detect when a user is sad and try to make them happy

- **Out Of Context Reasoning:** Sometimes models generalize much further than expected. Most famously, emergent misalignment, where training a model to write insecure code turns it into a Nazi. What's up with this? Some past work from my scholars suggests this is often downstream of learning a single direction, with hints that it's because the general solution is more efficient. But there's a lot we don't understand - is this the whole story? Why are some solutions easier to learn than others? Do these weird effects come up in any real use cases?
    - A notable example is synthetic document fine-tuning, where training on LLM-generated documents from a world where some false fact is true can get LLMs to internalize it and act on the consequences of that false belief. What's going on here? Does this really work? How robust is it? Etc.
- **Concept Representations**: How are specific interesting concepts computed and represented?
    - Can we train a truth probe that generalizes well to real situations?
    - What about a deception probe?
    - How is uncertainty represented?
    - Why on earth is there a misalignment direction?
    - How is the awareness of whether or not it is being evaluated represented? Nemotron 49B seems like a good model to study here.
- **Conflicting information**: How do models deal with conflicts between instructions or goals, or their prior knowledge and the context?
- **Model Diffing:** What changes during fine-tuning? Comparing a model before and after a change (e.g., chat-tuning, instruction-tuning, or fine-tuning on fake facts) can be a powerful way to isolate what was learned - see my past scholar's work on diffing chat finetuning for more pointers.
    - Seeing what happens during reasoning fine-tuning could be particularly interesting. Venhoff et al used simple techniques like per token KL divergence to study the high level differences, while Ward et al zoomed in on backtracking, a specific behaviour.

## Circuit analysis

- **Attribution graphs**: Are attribution graphs a useful technique for understanding model biology? Can you find anything interesting with the graphs on Neuronpedia? Can you find ways to overcome some of their limitations? Can you find things with them that cannot be found with simpler techniques like guessing and checking?
    - How important is precision? One notable consequence of the attribution graph approach vs, e.g. prompting, is that it can find much more nuanced and detailed hypotheses, like the addition analysis in Lindsey et al. Are there tasks where this precision is important?
- **Baselines**: There's a bunch of simple methods that fundamentally boil down to guessing hypotheses and checking them. Far more effort has gone into fancy techniques like attribution graphs than these. How far can we push them?
    - Linear probes can be highly effective at identifying concepts the model is representing – can we automate and scale the process of testing many linear probes, at all appropriate layers / token positions, for a given task?

    - Scaling the process of reading a model's chain of thought. How can we best analyze and aggregate them to look for unexpected properties, across many prompts? Docent is one interesting approach in this direction.
    - Simply observing model behavior in response to an appropriate mix of prompts can be highly effective to infer mechanistic hypotheses, but there's an art to doing it well. What do best practices here look like? Can they be automated?
- **Automation**: Can we automate the full hypothesis generation + validation loop with LLM agents?
    - Automated hypothesis generation
        - Can LLMs simply guess the high-level casual graph of a task? Can an agent make more headway if we let it iteratively choose diverse prompts and read the output
        - How good are LLMs at interpreting an attribution graph and how good can we make them with the right prompt and scaffold?
    - Automated validation
        - Can we automate the design of probes to test for the presence of predicted features?
        - Can we automate intervention experiments, and synthetic / out-of-distribution inputs, used for hypothesis validation?

### Objectively Measuring Interpretability

- **Eliciting Latent Knowledge:** Can we use interpretability to elicit secret knowledge from a model? What techniques work best?
    - In Cywiński et al my scholars taught a model a secret word by training it on descriptions of that word, and then retrieved it with both black and white box techniques. Can you do better? (their models)
    - Can you create more interesting model organisms? Marks et al did a much more ambitious version, involving teaching a model fake facts with synthetic documents, what else can you do with that technique?
- **Understanding-based downstream tasks**: In addition to the above, what other objective tasks are there that test our success at understanding? Movva et al. is another nice example.

## Applied Interpretability

I'm excited about a work that finds practical, real-world applications of interpretability, especially for safety. This isn't just using downstream tasks for grounding. The point is to choose a problem that actually matters and show that interpretability helps. I find this an exciting line of work because if we want interpretability to eventually be useful for making AGI safe, figuring out how to do things now seems like important practice.

- **Monitoring:** An extremely important problem in safety is that of monitoring: as a model runs, seeing whether a certain concept is present. The classic technique of probing is extremely cheap and is SOTA for cheap monitoring on frontier models for detecting misuse. What else can we do with probes?

    - How can probes be improved? Can we address cases where traditional probes work less well, like when information is spread across tokens or when there is a long context with lots of room for false positives? Attention head probes in Kantamneni et al are a good starting point.
- **Analyzing Chain of Thought (CoT):** Can you use or analyze a reasoning model's CoT to understand its behaviour, or otherwise achieve some practical uses? Can you steer the behaviour by resampling or editing the CoT?
    - The simplest way is to read or have an LLM read the CoT
    - Bogdan et al may be helpful if you need more powerful techniques
- **Other techniques:** Some other techniques that I think may have promising practical applications.
    - Conditional steering: applying a steering vector only if a probe fires. This lowers the side effects of steering a lot.
    - Training data attribution: A family of methods, including influence functions, to study which data points would have influenced a model to take a particular behavior more. The mathematical claims here are basically bullshit, but I think that being able to associate model behaviors with data points opens interesting use cases like debugging or removing noisy data points or filtering for the best data to finetune on
        - Warning: If you haven't played with TDA before, this may not be practical to work with in 20 hours
    - Abliteration: In refusal is mediated by a single direction my scholars cheaply jailbroke models by removing the refusal direction from the weights. How else can the idea of "abliteration" be applied?

## Basic Science

I am generally excited about work that moves forward our understanding of key problems in interpretability. This is less of a focus of mine than it used to be, but I am still excited to supervise such work. Note that I am not particularly interested in work on toy models, algorithmic tasks, or interpretability during training unless there's a great pitch.

- **Understanding Reasoning Models:** What is actually happening inside reasoning models that produce long chains of thought? Can we intervene on their reasoning process?
    - It's surprisingly difficult to edit a model's chain of thoughts, since if you regenerate from that point onwards they will often immediately correct any errors introduced. What's up with this? Can we stop it? If you token force the next sentence, is that enough? Etc.
    - How do models trained with RL compare to those that are distilled from an RL-trained model? E.g., comparing QwQ to an R1 distill.
- **Steering Fine-tuning**: In Casademunt et al my scholars showed that you can control how a model generalises after fine-tuning, with zero change to the data or loss, by ablating concepts we don't want it to use. They used this to mostly fix emergent misalignment. This is really cool! Where else can we apply it?

- **Circuit finding:** What are tools like transcoders or attribution graphs actually telling us about circuits? Are they doing what we think they're doing? What are they missing?
    - How big a deal are the cross-layer connections in cross-layer transcoders? What are they really doing?
- **Basic science of SAEs**:
    - Why are some concepts learned, and not others? How is this affected by the data, SAE size, etc.
        - Scaling Monosemanticity had some awesome preliminary results here, but I've seen no follow-ups
    - How big an improvement *are* Matryoshka SAEs? Should we just switch to using them all the time, or do they have some flaws? What are they really doing?
- **Sanity checking superposition:**
    - Can we find the "true" direction corresponding to a concept? How could we tell if we've succeeded?
    - Can we find a compelling case study of concepts represented in superposition, that couldn't just be made up of a smaller set of orthogonal concepts? How confident can we be that superposition is really a thing?
    - Can we find examples of non-linear representations? (Note: it's insufficient to just find concepts that live in subspaces of greater than one dimension)

### Novelty

- **New ideas**: For anyone feeling ambitious, I'm extremely impressed with any application showing ideas and applications of interpretability that are new to me or that I didn't expect to work
    - One of my favorite recent examples was in Casademunt et al, where my scholars showed it was possible to steer finetuning without changing the data.

## Other research philosophy updates

I used to see the North Star of the field as ambitious reverse-engineering: the idea that there was some deep, human-interpretable truth of how systems worked, and with enough effort we could largely understand it. I now believe this is likely doomed, and that models are just not that nice. Models seem less like engineered programs that we could hope to reverse-engineer, and more like biological systems: they have beautiful emergent structure, but also a long tail of messiness, noise, and inscrutable heuristics, which I don't see going away.

I still think there's a bunch of exciting and important things to do! The field has had a bunch of wins, like beating baselines on tasks that matter, building better tools, and successfully eliciting hidden goals - this is a big upgrade over my previous position of "well, interpretability is basically useless right now, but it could be a really big deal one day". Ultimately, my main goal is to help ensure AGI is safe, and most theories of impact here (e.g. detecting deception) only have ambitious interpretability as a means to an end, not a requirement. But I am trying to be more pragmatic and grounded.

One way these updates are actionable is that I'm skeptical of approaches to interpretability which assume precision, e.g. that we will be extremely confident we missed nothing, that we found the exact correct concept directions, etc. But I remain optimistic about approaches to interpretability that are robust to imprecision, like model biology and applied interpretability.

Some other things I currently believe:

- **Diversify**: I'm more skeptical of our ability to identify a priori the most promising directions. I think we should be pursuing a range of work, across those three categories.
    - Historically, the field has been over-invested into basic science IMO. I think it should still be a meaningful fraction of the field, but I think it's more impactful for me to focus on model biology and applied interpretability. I also think these are more robust to my skepticism that we will find precise insights.
- **Understanding > control**: A machine learning technique can either try to change a model's behavior (control) or merely to understand it.
    - Interpretability could, in theory, be applied to both. But almost all of machine learning is about control and almost none of it is about understanding.
    - So I view interpretability's comparative advantage as understanding (with some exceptions)
- **Grounding with downstream tasks**: I don't know how to interpret numbers like "this sparse autoencoder has 95% loss recovered". I don't think the error is going to go to zero, I don't think it *needs* to go to zero, but what's the relevant threshold?
    - I found it more useful to ground out interpretability questions by measuring their effectiveness on some downstream tasks: objectively measurable tasks that non-interpretability researchers would agree are real.
    - Now we can compare to baselines to get a threshold with *some* meaning
    - This gives hard-to-fake grounding that *something* real has been understood
    - E.g. my understanding of sparse autoencoders has been substantially clarified by testing them in downstream tasks like extracting hidden goals and detecting harmful intent in user prompts
- **On Sparse Autoencoders (SAEs)**: I was initially very excited that SAEs might be a transformative tool for reverse-engineering. My view is now much more moderate.
    - They are an excellent tool for discovery—finding concepts when you don't know what you're looking for. They're a valuable tool for model biology, e.g. debugging weird phenomena in models.
    - However, for known concepts, they tend to underperform supervised tasks.
    - They are a useful tool, but I think were too big a focus of the field.