How to pursue a career in technical AI alignment

This guide is written for people who are considering direct work on technical AI alignment. I expect it to be most useful for people who are not yet working on alignment, and for people who are already familiar with the arguments for working on AI alignment. If you aren't familiar with the arguments for the importance of AI alignment, you can get an overview of them by reading Why AI alignment could be hard with modern deep learning (Cotra, 2021) and one of The Most Important Century Series (Karnofsky, 2021) and AGI Safety from First Principles (Ngo, 2019).

It might not be best for you to work on technical AI alignment. You can have a large impact on reducing existential risk from AI by working on AI strategy, governance, policy, security, forecasting, support roles, field-building, grant-making, and governance of hardware. That's not counting other areas, such as bio-risk. It is probably better to do great work in one of those areas than mediocre technical alignment work, because impact is heavy-tailed. One good exercise is to go through Holden Karnofsky's aptitudes podcast/post, and think about which of the aptitudes you might be able to become great at. Then ask yourself or others how you could use those aptitudes to solve the problems you care about. I also recommend applying to speak with 80,000 Hours.

I'll probably be wrong but I might be helpful. Feedback was broadly positive, but I wouldn't be surprised if some people think that this guide is net-negative. For example, because it pushes people toward/away from theoretical research, or empirical research, or ML engineering, or getting a PhD. I have tried to communicate my all-things-considered view here, after integrating feedback. But I can only suggest that you try to <form your own view> on what's best for you to do, and take this guide as one input to that process.

I had lots of help. Neel Nanda helped me start this project. I straight-up copied stuff from Rohin Shah, Adam Gleave, Neel Nanda, Dan Hendryks, Catherine Olson, Buck Shlegeris, and Oliver Zhang. I got great feedback from Adam Gleave, Arden Koehler, Rohin Shah, Dan Hendrycks, Neel Nanda, Noa Nabeshima, Alex Lawson, Jamie Bernadi, Richard Ngo, Mark Xu, Oliver Zhang, Andy Jones, and Emma Abele. I wrote most of this at Wytham Abbey, courtesy of Elizabeth Garrett.

Types of alignment work (from Rohin Shah) **2**

What type of alignment work should you do? **4**

High-level heuristics for choosing which work to do 4

Some things to keep in mind when exploring different paths 5

How to pursue alignment work **6**

How to pursue empirical alignment work 6

Activities that are useful for both empirical research leads and contributors 6

Whether and how to do a PhD 8

How to pursue research contributor (ML engineering) roles 10

How to pursue theoretical alignment work 12

Learning **15**

Basic deep learning 15

Machine learning 16

AI alignment 18

One path for learning about alignment 18

Forming your own views on alignment is important when you have control over the direction of your work 19

Funding **22**

Broadly useful career advice **23**

Look for ways to demonstrate your competence 23

Focus on becoming excellent early in your career 23

Engaging with the AI alignment community will help you a lot 24

Take care of yourself 24

Types of alignment work (adapted from Rohin Shah)

For direct technical alignment research aimed at solving the problem (i.e. ignoring meta work, field building, AI governance, etc), these are the rough paths:

1. **Research Lead (theoretical):** These roles come in a variety of types (industry, nonprofit, academic, or even independent). You are expected to propose and lead research projects; typically ones that can be answered with a lot of thinking and writing in Google Docs/LaTeX, and maybe a little bit of programming. Theoretical alignment work can be more conceptual or more mathematical—the output of math work tends to be a proof of a theorem or a new mathematic framework, whereas in conceptual work math is used as one (very good) tool to tell if a problem has been solved. Conceptual work is more philosophical. A PhD is not required but is helpful. Relevant skills: *extremely* strong epistemics and research taste[1], strong knowledge of AI alignment; this is particularly important due to the lack of feedback loops from reality.

2. **Research Contributor (theoretical):** These roles are pretty rare; as far as I know they are only available at ARC [as of May 2022]. You should probably just read their hiring post.

3. **Research Lead (empirical):** Besides academia, these roles are usually available in industry orgs and similar nonprofits, such as DeepMind, OpenAI, Anthropic, and Redwood Research. You are expected to propose and lead research projects; typically ones that involve achieving or understanding something new with current ML systems. A PhD is not strictly required but in practice most Research Leads have one. Relevant skills: strong research taste, strong knowledge of AI alignment and ML, moderate skill at programming and ML engineering.

4. **Research Contributor (empirical):** These roles are usually available at industry orgs or similar nonprofits, such as DeepMind, OpenAI, Anthropic, and Redwood Research. You are expected to work on a team to execute on research projects proposed by others. A PhD is *not* required. Relevant skills: strong skill at programming, moderate research taste, moderate knowledge of AI alignment, jobs vary in how much they require skill at ML engineering (but most require strong skill).

5. **Professor:** This is a specific route for either of the "Research Lead" career paths, but with additional requirements: as an academic, you are not only expected to propose and lead a research agenda, but also to take on and mentor grad students in pursuit of that research agenda, to teach classes, etc. A PhD is required; that's the clear first step on this career path. Relevant skills: strong research taste, strong AI knowledge, moderate technical communication. Programming ability and ML ability is typically *not* tested or required, though they are usually needed to be successful during the PhD.

6. **Software Engineer:** Many organizations can also benefit from strong software engineers — for example,

[1] My best definition so far is that research taste is the ability to judge whether a research question is worth pursuing. See Rohin's Career FAQ (Ctrl+F: "What skills will I learn from a PhD?").

by creating frameworks for working with large neural nets that don't fit on a GPU, or by reorganizing codebases to make them cleaner and more modular to enable faster experimentation. However, I expect you should only aim for this if you already have these skills (or can gain them quickly), or if for some reason you think you could become a world-class expert in these areas but not in any of the other paths.

The main difference between research leads and research contributors is that the research leads are expected to add value primarily by choosing and leading good research projects, while the research contributors are expected to add value primarily by executing projects quickly. However, it isn't feasible to fully separate these two activities, and so [research] leads still need to have some skill in executing projects, and contributors still need to have some skill in choosing how to move forward on a project. Some orgs like DeepMind make the difference explicit ("Research Scientist" and "Research Engineer" titles), while others like OpenAI [Anthropic] do not ("Member of Technical Staff" title).

The main reason I carve up roles as "lead" vs "contributor" is that as far as I can tell, "lead" roles tend to be filled by people with PhDs. DeepMind explicitly requires PhDs for the Research Scientist role, but *not* for the Research Engineer role. (Both roles are allowed to lead projects, if they can convince their manager and collaborators that it is worth pursuing, but it's only an explicit expectation for Research Scientists.) Other orgs don't have a PhD as an explicit requirement, but nonetheless it seems like most people who end up choosing and leading research projects have PhDs anyway. I think this is because PhDs are teaching research skills that are hard to learn by other routes.

I don't want to emphasize this too much — it is still possible to lead projects without a PhD. In April 2022, I could name 10 people without PhDs whose work was best categorized as "Research Lead", who seemed clearly worth funding. (Note that "clearly worth funding without a PhD" doesn't necessarily mean the PhD is a bad choice: for several of these people, it's plausible to me that they would do much better work in 5 years time if they got a PhD instead of doing the things they are currently doing.)

What type of alignment work should you do?

I don't have a strong view on what type of alignment work is most valuable, so I'll mostly focus on personal fit. There is widespread disagreement in the community about the relative value of different work. However, the main decision you'll have to make early on is whether, if at all, to pursue empirical or theoretical alignment work. And I think most people believe there's good work to be done in both camps. If that's true, it means you can probably just focus on becoming excellent at either theoretical or empirical work based on your personal fit, while you <form your own views> about what specific theoretical/empirical alignment work is worth doing.

However, I think most people agree that if you can become a research lead who can set good, novel research agendas, then you should do that. You'll need to have strong research taste and <end-to-end thinking on AI alignment form your own views >, which is a high bar. Paul Christiano and Chris Olah are examples of people who did this.

High-level heuristics for choosing which work to do

If you're already a strong software engineer, consider applying to non-ML roles immediately, or <retraining as an ML engineer HTPMLE**>.** Some engineering work on alignment teams doesn't require ML knowledge. For example, creating frameworks for working with large neural nets that don't fit on a GPU, or reorganizing codebases to make them cleaner and more modular to enable faster experimentation. Some ML engineering roles might not even require experience with ML if you're a sufficiently strong software engineer. That is at least the case at Anthropic: "Lots of history writing code and learning from writing code is the hard part. ML is the easy bit, we can teach that." I suggest reading AI Safety Needs Great Engineers, DeepMind is hiring for the scalable alignment and alignment teams, and 80,000 Hours' Software Engineering career review.

To the extent that you think you might enjoy machine learning and coding, consider looking into <How to pursue empirical work>. You can test whether you like ML and coding by learning <Basic deep learning>. The early steps for research leads and research contributors are similar, so you can pursue those steps while figuring out which is better for you.

To the extent that you love theory, have or could get a very strong math/theoretical CS background, and think you might enjoy building end-to-end models of AI alignment, consider looking into <How to pursue theoretical alignment work>.

Some things to keep in mind when exploring different paths

Pay attention to whether you're enjoying yourself and growing and flourishing and kicking ass. But don't give up immediately if you're not. Enjoying yourself is really important, especially for research. But often people enjoy things more as they gain more mastery, or think they should already be good and suffer until they get there. Often people have bad luck. If you're enjoying yourself and kicking ass then that's a great sign. If you're not enjoying yourself and kicking ass after a while then consider switching to something else.

Sometimes very capable people are insecure about how good they are, and miscalibrated about how good they could become. Here are some more objective indications you can use to assess your fit:

- Empirical research leads: To have a decent shot at getting into a top-20 ML PhD programme, you'll need (very roughly) a first author workshop paper and a ~3rd author conference paper at NeurIPS, ICML, or similar.
- Empirical research contributor: "As a rough test for the Research Engineer role, if you can reproduce a typical ML paper in a few hundred hours and your interests align with ours, we're probably interested in interviewing you (DeepMind)". Grades matter less than people think.
- Theoretical research: If you think you could get into a top PhD programme in math or CS theory if you (had) optimized your undergrad for that purpose, that's a good sign.

Talk to people and ask them to honestly evaluate whether you're on track to do good technical work. This is a good way to address the point above. Make it easy for them to tell you that you're not on track in worlds where you're not—for example, by emphasising to them how helpful it would be for you to switch to something you're better at sooner. You could do this at Effective Altruism Global, or by talking to 80,000 Hours.

Recommended resources:

- 80,000 Hours article on personal fit.
- Holden Karnofsky's aptitudes podcast and post.

How to pursue alignment work

This is a high-level section that gives context and high-level heuristics for pursuing different types of alignment work, with pointers to other places in the doc that go into more depth.

How to pursue empirical alignment work

The early steps for research leads and research contributors are similar, so you can pursue those steps while figuring out which is better for you. Whether you want to pursue research lead or research contributor roles will mostly depend on how much you like and are good at research, end-to-end thinking on alignment, and machine learning, relative to how much you like and are good at ML engineering. Also whether you want and are able to get into a top PhD programme. If you're uncertain, I recommend learning <basic Deep learning>, doing some ML implementation, and trying to get some research experience (see the next section). Then assessing personal fit from there, which might include talking to people about your fit.

Activities that are useful for both empirical research leads and contributors

Everyone should learn Basic deep learning: You'll need to learn basic Python coding, basic math (linear algebra, calculus, and probability), and get a basic understanding of deep learning (DL) models and how to implement them. DL is by far the dominant paradigm within machine learning, which in turn is the dominant paradigm within AI safety. I've included the best resources I know of in Basic deep learning.

You'll need to become a decent ML engineer, even if you want to become a research lead. To become good at ML engineering, you'll need to get experience implementing DL models.

- It helps if you're already a good software engineer (SWE), and a SWE internship in your first year of undergrad might be a good investment, especially if there's a good chance you'll want to do ML engineering.
- A good way to get experience implementing DL models, after learning Basic deep learning, is to replicate a few foundational papers in a sub-field you might want to work in. See <How to pursue research contributor (ML engineering) roles> for details on how to do that. Paper replications are essential for

contributor roles, and useful for lead roles. <Look for ways to demonstrate your competence>, by open-sourcing your code and maybe writing a blog post on your work. You can apply for <funding> to do paper replications.

- You can also get better at ML engineering by doing practical homeworks as part of a course, or through a research internship. But how much you learn will depend strongly on the mentorship and the project: academia is a generally poor place to learn ML engineering: your peers likely won't be excellent engineers, and your advisor probably won't invest much in helping you.
- I also recommend applying to Redwood Research's (competetive) Machine Learning for Alignment Bootcamp (MLAB). The deadline for the application has passed, but there might be future cohorts. Practising leetcode problems is probably useful for getting accepted.

Research experience is essential for research leads, and useful for research contributors.

- ML PhDs are very competitive nowadays, and publications and reference letters are the main determinants of whether you get in. To have a decent shot at getting into a top-20 PhD programme, you'll need (very roughly) a first author workshop paper and a ~3rd author conference paper at NeurIPS, ICML, or similar. *Publications will also look good for ML engineering roles.* If you want to get a PhD, you should try to get as much research experience as you can before applying to PhD programmes, after learning Basic deep learning. For example, in the summer of your second and third years for four-year degrees, because publications from after you submit your application won't count.
- Adam Gleave writes about how to get research experience here. When reaching out to potential advisors for research internships, make sure to mention the stuff you've done (relevant courses you've taken, definitely any paper replications you've done, your GitHub if it shows you can code), your grades if they're good. Aim to clearly distinguish yourself from people who spam professors with requests to do research with them. One way to do this is to read some of their papers and the main papers in their field, and mention that too. If your university can't fund you to do research experience with them, you can apply for <funding>.
- Note that it is somewhat difficult to get two publications before your 4th year, and it will involve luck! If you don't get into a strong programme right away, don't get down on yourself. It might be worth taking a year or more after university to do research assistant work in order to get publications. You might be able to get <funding> to do that.
- For research projects, find someone who supervises well and who has time for you (this tends not to be the most famous/cool professor). ML is a very new field, so some professors who ostensibly work on ML don't do stuff that's relevant to DL. Make sure they're regularly publishing at top conferences. And work on a topic that your supervisor finds interesting to get lots of feedback: getting good mentorship is key, and pursuing your own ideas at this point is risky and usually means that you won't get much mentorship. Don't worry about working on something that helps with alignment. Take project graduate courses where you can—they will probably be graded leniently.
- I recommend Research as a stochastic decision process (Steinhardt, 2019) for getting better at research execution.

<Learning Machine learning>: how, and how much? It's easiest to learn by being immersed in a research environment, so it's sensible to focus on learning enough ML to get to that point. That means having enough breadth to talk about the main areas of DL sensibly and know about the recent advances, and having depth in the area you want to go into. *You don't need to learn all of ML to become part of a research environment.* Though research leads should probably eventually know a bunch of ML. You can get breadth by taking courses in the most important subfields of ML (see <Machine learning>), and using resources that curate and summarise/explain recent advances (see <Machine Learning>). You can get depth by reading a bunch of a sub-field's main papers (~10+, or until you get diminishing returns) and doing your own research, or practical homeworks, or < paper replications How to pursue research contributor (ML engineering) roles>. You can see what areas people are interested in by looking at blogs of the labs you're interested in working at, or by checking the Alignment Newsletter. If you can take ML courses for credits, that is probably a great idea. See <Machine learning> for more details.

<Learning AI alignment>: how, and how much? I recommend AGI Safety from First Principles (Ngo, 2020) and My Overview of the AI Alignment Landscape (Nanda, 2022) to get started, then the AGI safety fundamentals seminar programme or similar alignment reading sometime after learning Basic deep learning. Learning AI alignment is a lot more important for research leads than research contributors—doing the stuff above is not sufficient for research leads and is not necessary for some research contributor roles, but it will likely be pretty useful for both. There's much more detailed advice in <AI alignment>.

Whether and how to do a PhD

If you want to be a research lead, the default path is to get a PhD. However, it is also possible to start working as a research engineer and gradually transition toward a research lead role, though as a research engineer you'll have less time for research activities than you would in a PhD programme. It is also possible to become a research lead without a PhD, if you do a residency program. It's worth noting that the research-engineer boundary is dissolving at places like Anthropic and OpenAI. This is partially because they care less about the signalling of PhDs, and partially because their research leans relatively heavier on engineering (scaling) than on coming up with novel research directions. *The most important thing for becoming a good research lead is getting mentorship from a great researcher and being able to practice research in a good environment.* That's most often achieved in a PhD but is sometimes possible in industry.

There is pretty widespread disagreement about how good PhDs are. My impression is that the bulk of the disagreement comes down to how effectively PhDs train research taste and skills that are useful for alignment research, and secondarily, how quickly people expect AGI will be developed—if 5 years then PhDs don't look good—because they likely won't do any useful work—if 15 years then it's less of an issue. My understanding of the main benefit of a PhD is that it develops your research taste and skills so that when you graduate, ideally, you're able to set and execute your own (good) alignment research agenda in industry (at an existing or new org) or in academia. Failing that, the idea is that you'd come away from a PhD with great research skills that help with alignment research. A PhD also opens some doors that ML engineering wouldn't be able to, for example, research scientist roles at DeepMind or Google Brain.

Here are some simplifying questions you can ask yourself to make the decision easier:

- Do you want to mainly do research rather than ML engineering or software engineering? (If no, then don't do a PhD.)

- Do you think you could publish a first author workshop paper and a ~3rd author conference paper at NeurIPS, ICML, or similar, while in undergrad, or shortly after undergrad (or after 6-12 months of full-time ML research)? (This is the bar for having a decent chance at getting into a top-20 programme; it's higher for top-10 programmes) (If no, then maybe don't shoot for a PhD, but I suggest actually trying out research first.)
- Would you feel a lot better making a lot more money? As an engineer you would probably make upwards of $100,000. You can <apply funding> to get a top-up on a PhD salary, but I think it would be somewhat unusual to get a top-up to $100,000.
- Comparing specific options:
    - With some work, do you think you could get an ML engineering/research role where you could grow/flourish/get better mentorship than you expect to in a PhD? (If yes, then probably don't do a PhD.) (You can ask people to help compare your options.) (You can speak to grad students of the professor and attend open days.)
    - Are you able to do work in the PhD that is directly or instrumental useful for alignment work; that has a path to impact? (If yes, then a PhD looks much better. If not, it looks worse.)
    - Are you going to be tied down to a PhD topic that you're not confident in (few other desirable topics you could switch to at the university)? (If yes, then a PhD looks worse, because it's quite common to want to switch.)
- PhDs can sometimes make people miserable. People enter into them because they're the default path for people who did well in undergrad, they don't really know what they're getting into, and then they burn out. *It's useful to think in advance about whether that is more or less likely to happen to you.*
    - Do you expect to thrive in an unstructured environment with weak incentives (it's hard to get fired), and potentially poor feedback loops and support, but lots of freedom? (If no, then probably don't do a PhD.) (Conscientiousness is probably a factor here.)
    - Here are some other factors that interact poorly with PhDs: high neuroticism, depression, anxiety, and ADHD. You can do some quick diagnostic tests (neuroticism, depression, anxiety, burnout) to see whether any of those things might be a concern for you. Probably don't take the tests too seriously, but I know many smart people who took a year to realise they had depression, so it might be worth investigating and following up (e.g. with therapy) if any of those are concerning. Having these factors doesn't rule out a PhD, but research will be much harder, and you'll want to be very careful to choose a supportive advisor, which limits options.

If you're uncertain about which path to pursue, it might be worth optimising for doing research in the short term while you get a better sense of whether a PhD makes sense for you (or whether you get offers from a top programme), and decide later, or apply to both PhDs and ML research engineering roles and compare options. Doing research will look pretty good for engineering roles as long as you stay away from theory-heavy research topics and eventually do enough ML engineering. And it's a good test of fit. But optimising for ML

engineering won't help as much for PhDs, because publications and reference letters are key. You can however apply for a PhD after doing ML research engineering in industry.

How to do a PhD: If you are considering doing a PhD, I strongly recommend reading Careers in Beneficial AI Research (Gleave, 2020), Rohin Shah's Career FAQ, Andrej Karpathy's survival guide for PhDs, and Machine Learning PhD Applications — Everything You Need to Know.

How to pursue research contributor (ML engineering) roles

Read Activities that are useful for both empirical research leads and contributors. That section talks about how to learn <Basic deep learning>, ML, and AI alignment, and how to get research experience. If you're sure you want to shoot for research contributor/ML engineering work, getting research experience is less important than for research lead roles, but might still be a useful source of mentorship and skill-building. Strong knowledge of AI alignment is also less important for getting research contributor roles, but how much you want to invest will depend on how much you want to eventually direct your own research, and investing where possible seems valuable. See <AI alignment> for more details.

Being a good software engineer will make you a better ML engineer. If you can get a software engineering (SWE) internship at a top company early on, that will likely prove valuable. More broadly, getting mentored by someone much better than you at SWE will likely be valuable, as will reading and writing lots of code. In addition to internships and jobs and your own projects, you might be able to get mentorship by contributing to open-source projects and asking some senior person on that project whether they might mentor you. Perhaps check out 80,000 Hours' Software Engineering career review.

Do some paper replications. To become good at ML engineering, you'll need to get experience implementing ML models. A good way to do that is to replicate a few foundational papers in a sub-field you might want to work in. This is similar to the task of implementing novel algorithms, but with training wheels: you know that the algorithm works and what good performance looks like. It will also give you a great understanding of the methods you implement. <Look for ways to demonstrate your competence>, by open-sourcing your code and maybe writing a blog post on your work. You can apply for <funding> to do paper replications. You can apply for <funding> to do paper replications. See "Paper replication resources" below for more advice.

Below are some paper replication ideas. These are pretty off-the-cuff. If you're serious about spending a couple of hundred hours on paper replications, it might be a good idea to reach out to a lab you want to work at with a specific plan so that they can give feedback on it. Ideally, see if you can get someone to mentor you. It will be useful to have an open-source codebase on hand, so try to find one before you set out. Check out <Machine learning> for the relevant background.

- Language models:

    - Easier: Train a small GPT-2 model from scratch using existing implementations of the architecture (e.g. from Hugging Face). Maybe try grokking.
    - Harder: Implement the transformer yourself. You could also fine-tune with human preferences, as in Ziegler et al. (2019) or Stiennon et al. (2020).
- Reinforcement learning (I probably wouldn't start with these):
    - Easier: Try to replicate one of the common RL algorithms, like DQN/PPO/A3C. There are papers that talk about ways these methods don't replicate. See Spinning up in deep RL as a way to ease into these exercises.
    - Harder: Deep RL from human preferences (see this also) or GAIL.
- Computer vision:
    - Very easy: train an MLP on MNIST.
    - Easy: train a ResNet or another close-to-state-of-the-art model on ImageNet.
    - Medium: do some basic adversarial attacks and defences. You might want to play with this first. You could try out some attacks and defences from this list of papers.

Apply to MLAB: Redwood Research is running another fully funded (competitive) coding bootcamp in summer 2022. The deadline for the application has passed, but there might be future cohorts. Practising leetcode problems is probably useful for getting accepted.

What does it take to get a job?

- DeepMind: "As a rough test for the Research Engineer role, if you can reproduce a typical ML paper in a few hundred hours and your interests align with ours, we're probably interested in interviewing you." You can read about their available roles here.

- Andy Jones (Anthropic): "It's hard to judge sight-unseen whether a specific person would suit AI safety engineering, but here's a good litmus test: With a few weeks' work, could you - hypothetically! - write a new feature or fix a serious bug in a major ML library?" (Important caveat: this is who Anthropic would hire immediately and expect to be contributing in week one. It is *not* a test for whether you could become such a person.) Anthropic talks about their hiring process here and what they're looking for here.
- From a Redwood Research staff member: "If you can implement a transformer in pytorch, you should probably already be speaking to Redwood". You can read about their roles here.
- For other places, like Google Brain/Facebook, you'll also need to be able to leetcode and have a high GPA. Having ML research experience helps, as does a master's degree.

Where should you work? Adam Gleave: "The best way to learn research engineering is to work somewhere there is both high-quality engineering and cutting-edge research. Apply to [very competitive] residency programs at industrial labs. The top-4 labs are DeepMind, OpenAI, Google Brain and Facebook AI Research (FAIR); there are also smaller (but good) safety-focused labs like Anthropic and Redwood Research. There are also many smaller players like Amazon AI, NVidia, Vicarious, etc. These are generally less desirable, but still good options." Since Adam wrote that, some new organisations focused on language models have formed that could be good places to

build skills. Those are conjecture (safety-focused), cohere.ai (some near-term safety and lots of EAs working there; I wouldn't bet on it being good to end up there though), and Hugging Face (no existential safety).

For the first couple of years, it might be worth going where you'll grow the most. After that, you'll want to go wherever you can do the best alignment research. However, I am personally worried about people skill-building for a couple of years and then not switching to doing the most valuable alignment work they can, because it can be easy to justify that your work is helping when it isn't. This can happen even at labs that claim to have a safety focus! Working at any of Anthropic, DeepMind, Redwood Research, or OpenAI seems like a safe bet though. If you can't work at one of those places, whether skill-building outside of safety teams (e.g. at Google Brain or FAIR) is good will depend pretty strongly on whether you expect to be able to later shift to more impactful work (requires continuing to <form your own views on alignment> and agency), whether you'll be motivated doing work that doesn't help with alignment, and how useful it is to be surrounded by people who work on alignment relative to people who are great ML engineers—the former is more important the more you want to direct your own research, the latter is more important the more you expect ML engineering to be your main contribution.

Paper replication resources:

- Advice on paper replication (Ngo, 2022).
- ML engineering for AI safety and robustness (Olsson, 2018)
- Lessons Learned Reproducing a Deep Reinforcement Learning Paper (Rahtz, 2018) as an example, and as evidence that implementing papers can be very educational, but hard, especially in deep RL.
- A recipe for training neural networks (Karpathy, 2019).
- <Look for ways to demonstrate your competence>.

Career resources: If you intend to pursue ML engineering, I recommend reading these articles:

- ML engineering for AI safety and robustness (Olsson, 2018)
- AI safety needs great engineers (Jones, 2021); what skills are sought-after by large engineering companies working on safety?
- DeepMind is hiring for the scalable alignment and alignment teams.

How to pursue theoretical alignment work

I don't know that much about theoretical work, sorry. If you are a theoretical researcher and have thoughts on how to improve this section, please let me know! The paths to doing theoretical work are also a lot less well-scoped than the path to empirical work, so it's not *all* my fault. Anyway, here's what I've got:

Theoretical alignment work can be more conceptual or more mathematical.

What does conceptual work look like? Conceptual alignment work often involves reasoning about hypothetical behaviour. For example, Mark Xu (of the Alignment Research Center) describes most of his work as "coming up with good properties for algorithms to have, checking if algorithms have those properties, and trying to find algorithms that have those properties." This is pretty similar to a skill-set you'd expect a theoretical computer scientist to have. The work tends to involve a lot of mathematical and philosophical reasoning. Conceptual researchers also need strong research taste, and strong knowledge of AI alignment. This is so that they don't get lost in theoretical research that doesn't help with alignment, which is easy to do since theory work has poor feedback loops. Examples of conceptual research include Paul Christiano's Eliciting Latent Knowledge (ELK), Evan Hubinger's Risks from Learned Optimization, John Wentworth's Natural Abstractions, and MIRI's agent foundations work.

What does mathematical work look like? I think the main difference is that in math work, the output is a proof of a theorem or a counterexample or a new mathematic framework, whereas in conceptual work math is used as one (very good) tool to tell if a problem has been solved. Conceptual work is more philosophical: the arguments are rarely watertight, and a lot more judgement is required. Examples of mathematical work include Michael Cohen's Pessimism About Unknown Unknowns Inspires Conservatism, Vanessa Kosoy's Infrabayesianism, Scott Garabrant's work on Logical induction, Cartesian frames, and Finite factored sets, Cooperative Inverse Reinforcement Learning, and Tom Everett's work (thesis, current work). You can see more topics here. This is in contrast to semi-formal, conceptual work, of which Evan Hubinger's Risks from Learned Optimization is a central example.

Where does this work happen? The space is pretty weird. There aren't established orgs doing shovel-ready work. It's more like a mixed bag of people in academia (mostly math stuff, e.g. CIRL and Michael Cohen's stuff), independent people on grants (such as John Wentworth), the Machine Intelligence Research Institute (MIRI) (houses Evan Hubinger, Scott Garabrand, and Vanessa Kosoy among others), the Alignment Research Center (ARC) (which Paul Christiano directs), a few people at DeepMind (e.g. Ramana Kumar, and now some stuff at conjecture.dev too.

I don't have a great sense of whether math or conceptual research is better to work on. Fortunately, the skill-sets are pretty similar, so you can probably just try each a bit while you develop your own views about work work is most valuable, and then decide based on where you think you'll do the best work.

How to test fit for conceptual research: (I don't really know, sorry.)

- You will probably have a sense of how much you love and are good at theory, which is an important part of it. If you think you could get into a top PhD programme in math or CS theory if you (had) optimized your undergrad for that purpose, that's a good sign.
- The other component is being able to form your own views on AI alignment and having interesting things to say there.
    - The first step for testing that is to learn about <AI alignment>, perhaps to around the level of the AGI safety fundamentals seminar programme. It might be useful to learn <Basic deep learning> before doing that.

    - Then, if you enjoyed digging into the alignment reading, you could try to absorb a conceptual researcher's worldview, perhaps by reading and distilling (summarising) some of their research. Here is some research you could try distilling. The ML Alignment Theory Scholars programme (fully funded) is a structured way to absorb someone's worldview—you'll be paired with a mentor whose worldview you can absorb. The applications are closed (as of June 2022) but there will likely be future cohorts. Another way to absorb a worldview is to read through the Reading List for Evan Hubinger's AI Safety Worldview, and write up and maybe email him your notes, summaries, comments, disagreements, takes, opinions, etc..
    - A different option is to spend ~50 hours reading up on a specific topic in AI alignment that interests you, then spend ~50 hours trying to say something new and interesting about that topic. (For example, try producing a proposal for ELK.) Don't update too much if you don't have much to say; the point is more to see whether or not you enjoy the work and whether it feels productive.[2]
    - You can apply for <funding> to do any of these exercises.

How to test fit for mathematical research: (I don't really know, sorry).

- You'll probably have a sense of how much you love and are good at theory, which is the main thing. If you think you could get into a top PhD programme in math or CS theory if you (had) optimized your undergrad for that purpose, that's a good sign.
- One test is to go through Michael Cohen's lecture and have a go at the assignment, then can ask Michael to look at what you wrote (his email is firstname.lastname@eng.ox.ac.uk, and yes, he's happy to look at assignments).
- You could also spend ~50 hours reading up on a specific topic in mathematical AI alignment that interests you (you can see some of the literature here), then spend ~50 hours trying to say something new and interesting about that topic (or distilling, if saying something new is too hard).
- Finally, you could do a research internship with a mathematical researcher in an area related to mathematical alignment (or do research directly on alignment, but only if your supervisor already works on that). Ctrl+F for "Research experience is essential for research leads, and useful for research contributors" for advice on getting research experience. You might be able to reach out to mathematical alignment researchers for an internship. Perhaps after writing something, as above.

It's worth bearing in mind that pursuing theoretical alignment work is much riskier than ML-focused work, because you'll build fewer transferable skills than ML work, you'll have less credibility outside the alignment community, and the infrastructure for this work is just starting to be built. That said, if you think you could have a good fit, it might be worth testing it out!

How to pursue conceptual alignment research: Again, I don't really know. For that reason, getting mentorship seems pretty important. If you can produce something, perhaps from one of the exercises above, I think Mark Xu or Evan Hubinger would consider chatting with you and giving you career advice. Here are some

[2] HT Rohin Shah.

short-to-medium-term options: work independently on a grant (or at an existing organisation, though you'd probably need a PhD for that), work at ARC or MIRI (not sure whether MIRI is hiring as of June 2022), apprentice under a conceptual researcher, or do a PhD (in math/CS theory, with a smart and open professor who's regularly publishing in COLT or FOCS or similar. You probably won't be able to publish conceptual alignment work during a PhD, but you might build useful skills). My guess is that mentorship should be the main consideration early on in your career: if you can work with and get mentored by a strong conceptual alignment researcher, that is probably better than a PhD (unless you have the opportunity to work closely with a really strong or value-aligned advisor), and a good PhD probably looks better than independent work. If you want to try to apprentice under a conceptual researcher, or work at ARC/MIRI, some of the exercises in the previous section will be useful: reading and distilling and absorbing someone's worldview, posting on the AI Alignment Forum, and trying to get more mentorship from there. More broadly, I recommend spending time learning about <AI alignment> and <forming your own views form your own views>. It's worth noting that conceptual research is particularly mentorship constrained at the moment, so it might be hard to work closely with a strong conceptual researcher. It's probably still worth trying though, and in particular everyone should probably apply to ARC.

How to pursue mathematical alignment research: (I don't really know, sorry.) Probably read a bunch of the mathematical alignment literature (you can see some of the literature here). More broadly, I recommend spending time learning about <AI alignment> and <form your own views>. If you can get a theory PhD at the Center for Human-compatible AI (CHAI), that seems like a great bet. If you can do a theory PhD on something related to alignment, that is probably good too. It should be doable even if the professor doesn't work on alignment, as long as they're really smart and you can convince them that the topic is publishable. You could also work on something that's useful skill-building for alignment, such as probability theory as applied to AI, or some part of theoretical CS (look for profs who publish in COLT of FOCS or similar). You might get better supervision that way. Ctrl+F "How to do a PhD" for resources on how to get an ML PhD; a lot of it should transfer to theory PhDs. Please try to speak to someone more knowledgeable than me before jumping into a PhD though!

Learning

Basic deep learning

This is just the basics: I've included stuff that's sufficient to get you a *basic* understanding of deep learning models and how to implement them. This isn't all you need to become a great empirical research lead or contributor. In particular, investing in coding and math beyond what is indicated here will prove worthwhile. Please skip my suggestions if you already have the knowledge/skill.

When to do what: The coding and math can be done in parallel. The deep learning (DL) courses require basic coding and math. Strictly speaking, you can understand DL with a very basic understanding of linear algebra and calculus. But sooner or later your lack of foundation will cause problems. That said, you can probably comfortably start studying DL after a semester of math classes, alongside building stronger mathematical foundations.

Coding: You'll need to know how to read and write code in python. www.learnpython.org/ is good for that. There's also the skill of being able to do stuff in the python ecosystem, which people often end up picking up slowly because it's not taught. For that, I recommend The Hitchhiker's Guide to Python, and The Great Research Code Handbook. You might be able to get <funding> for a tutor. Here are some extra resources you might find helpful: Things I Wish Someone Had Told Me When I Was Learning How to Code, learntocodewith.me/resources/coding-tools/.

Math: Here are the areas of math required to learn basic DL. Other areas of math—like statistics—can be directly useful, and mathematical maturity beyond what is written here is certainly useful.

- **Linear algebra:** This 3Blue1Brown video series is good for intuition, as a supplement. For a stronger foundation, you'll want to take your university's intro class (or MIT's—I haven't taken it but it's probably good) and probably a more theoretical class. If you'd like a textbook for after an intro class, I recommend Linear Algebra Done Right.
- **Calculus:** This 3Blue1Brown video series covers basic calculus. Take your university's multivariable calculus course for a stronger foundation (or MIT's—I haven't taken it but it's probably good).
- **Probability:** One course is Introduction to Probability (MITx), but your university might have a course that covers similar content. Taking courses in statistics where possible will probably help too.

Deep learning: (DL) is by far the dominant paradigm within machine learning, which in turn is the dominant paradigm within AI. *Getting a good understanding of DL is essential for all empirical alignment work.* I recommend that you get practical experience by doing something like (1), and do one of (2) and (3). Participating in the ML Safety Scholars Programme (fully funded, applications close May 31st 2022) over the summer seems like a great, structured way to learn DL.

1. fast.ai is a practical course in deep learning (DL) that approaches DL from a coding (not math/statistics) perspective. If you already have some knowledge of how DL works, it is probably better to learn from the PyTorch tutorials. Or learn from those tutorials after doing fast.ai. PyTorch is a good framework to start with, but if you're already good with TensorFlow or JAX you probably don't need to pick PyTorch up until a project/job requires it.

2. Deep Learning Specialization (Ng), your standard DL class (CS 230 at Stanford).
3. Deep Learning by NYU (LeCun).

Machine learning

Summary: It's easiest to learn by being immersed in a research environment, so it's sensible to focus on doing enough to get to that point. That means having enough breadth to talk about the main areas of DL sensibly and know about the recent advances, and having depth in the area you want to go into. *You don't need to learn all of ML to become part of a research environment.* Though ML researchers should eventually know a lot of ML, and taking university courses in ML where you can is probably a good idea. You can get breadth by taking courses in the most important subfields of DL (see Learning about DL sub-fields), and using resources that curate and

summarise/explain recent advances (see Resources). You can get depth by reading a bunch of a sub-field's main papers (~10+, or until you get diminishing returns) and doing your own research, or practical homeworks, or <paper replications> (though this takes a while, and might not be worth it for researchers). You can see what areas people are interested in by looking at blogs of the labs you're interested in working at, or by checking the Alignment Newsletter (see Resources).

Learning about DL sub-fields: Once you finish Basic deep learning, you should have the background to go into any of these areas. I wouldn't worry too much about nailing all of these areas straight away, especially if it trades off against research or engineering.

- Natural language processing:
    - Modern transformers: lecture, OG transformer paper, GPT-2 or GPT-3, fine-tuning with human preferences.
    - Old school NLP techniques: Stanford CS 224n: (syllabus with link to the notes, youtube lectures)
        - (Unclear how useful this is, but it might be worth it if you want to do NLP research.)
- Reinforcement Learning:
    - Deep Reinforcement Learning: Pong from Pixels (short blog-post)
    - OpenAI Spinning Up in Deep RL: (link) (summary of Deep RL theory, and lots of implementation exercises)
    - Berkeley Deep RL (youtube lectures). Requires some probability theory as a background, especially for the later lectures on inference
    - DeepMind RL Lectures (youtube lectures).
- Computer vision:
    - Module 2 from Stanford's CS 231n: (course notes, youtube lectures) (pretty short, ~20 pages worth.)

Resources: (You don't have to keep up-to-date with all of these things! See which sources you like and benefit from.)

- The Alignment Newsletter highlights and summarizes important papers across many ML sub-areas relevant to safety. You can check those papers and areas out here. Keeping up with the AN is good for breadth, and the database is good for going deep into a sub-field.

- Looking at NeurIPS and ICML paper presentations from a sub-field is a decent way to get an understanding of that sub-field: what they're working on and what they care about. Can do in an afternoon.
- Jack Clark's Import AI newsletter.
- Dan Hendrycks' ML Safety Newsletter; r/mlsafety.
- Two Minute Papers.
- Yannic Kilcher: explains a wide variety of machine learning papers in 30-60 minute videos.

- ML Street Talk: podcast discussing a whole variety of ML related topics. Very good for seeing what leading experts in the field are thinking about.
- arXiv sanity preserver; connected papers.
- Follow the blogs or the Twitter accounts of the big AI players/researchers (I mostly get my information from Twitter).
- AGI Safety Core Twitter list.
- Lilian Weng's blog.

How to read papers: At some point you'll need to be able to read papers well. Here are some resources for learning how to do that. Most of the time, you'll want to be in "skim mode" or "understand deeply" mode, not somewhere in between.

- Andrew Ng's Guide Summarized (Original Link)

- Yannic's Guide
- Ctrl+F How should I read things efficiently? (Rohin)
- Anki's spaced repetition software is really useful for some people. Skip to "Using Anki to thoroughly read a research paper in an unfamiliar field".

AI alignment

Compared to other research fields—like math or theoretical physics—the EA-focused alignment space doesn't have that much content. It still takes months of full-time study to get fully up to date, but you can 80/20 much faster than that, and not everyone has to be an expert.

> Buck: "I think it's quite normal for undergraduates to have a pretty good understanding of whatever areas of [alignment] they've looked into."
>
> Buck: "Try to spend a couple of hours a week reading whatever AI safety content and EA content interests you. Your goal should be something like "over the years I'm in college, I should eventually think about most of these things pretty carefully" rather than "I need to understand all of these things right now"."

One path for learning about alignment

Getting started: I recommend AGI Safety from First Principles (Ngo, 2019) and My Overview of the AI Alignment Landscape (Nanda, 2022). If you would like to learn more about the motivation for AI risk, I recommend Why AI alignment could be hard with modern deep learning (Cotra, 2021) and The Most Important Century Series (Karnofsky, 2021), which are also available in podcast format.

AGI safety fundamentals seminar programme: I recommend applying to participate in the alignment track. If you have time, the governance track might also be valuable. Each track takes around 5h per week, for 8 weeks. To get the most out of the programme I would do it after Basic deep learning.

The Alignment Newsletter is really good. It summarises recent work in AI alignment and ML. One exercise (among many) that will help orient you on what is happening is reading the highlight sections from the 20-50 most recent Alignment Newsletters (takes around 10h). The AN requires some background in machine learning, so you might need to get that before reading, or alongside. Some tips:

- When you're in the wild and considering reading something, check the AN database first to see if there's a summary. The database houses papers and their summaries, and you can filter by area (e.g. interpretability) and by importance. It's particularly useful if you want to get clued up on an area fast. You might consider clicking on this right now to see how cool the database is :)

- Follow your sense of excitement, curiosity, and confusion. Dig into papers in depth if it feels exciting.
- If you don't understand something, you might need to jump back to an older linked newsletter.
- Consider motivating your reading with a question or uncertainty, such as "why do people think interpretability is important?", "what is going on with scaling laws and why are they important?", or whatever your most important uncertainties are.

Keep up to date: with the Alignment Newsletter, LessWrong, the EA Forum, the AI Alignment Forum (AF), the ML Safety Newsletter; reading posts that excite you. Blogs/Twitter from the alignment labs. There is also the 80,000 Hours podcast, the AXRP podcast (Richard and Paul's episodes are great starting points; Beth's and Evan's are great too), and the FLI podcast. And Rob Miles' Youtube channel. There is a bunch of content so you'll need to filter! One way to filter is by looking through the Alignment Newsletter. If you want to read old stuff, on the AF you can sort by upvotes.

Some people think that reading a lot is good, especially for conceptual work. The advice is "read everything". This won't be possible or good for most people! But if you can find a way to enjoyably sink 500h into active reading of alignment content, that will probably be really good for forming your own views. You might want to try out several resources, because some will be way more fun for you to read. The Alignment Newsletter is one source. Others include Paul Christiano's blog (difficult to read but you might love it), the MIRI dialogues (also hard to read but juicy), and Rationality: From AI to Zombies (some people love this and others are put off). Reading lots is less good advice if you're trying to do very competitive stuff, such as an ML PhD, because you'll need to spend a lot of time getting research experience.

Forming your own views on alignment is important when you have control over the direction of your work

I recommend reading Rohin Shah's Career FAQ (ctrl+F for "How can I do good AI alignment research?"), How I Formed My Own Views About AI Safety (Nanda, 2022), and Want to be an expert? Build deep models (Bye, 2021). I'll copy from these and add my own spin, but I think it's probably worth reading them directly.

> Rohin Shah: "We want to think, figure out some things to do, and then, if we do those things, the world will be better. An important part of that, obviously, is making sure that the things you think about, matter for the outcomes you want to cause to happen.
>
> In practice, it seems to me that what happens is people get into an area, look around, look at what other people are doing. They spend a few minutes, possibly hours thinking about, "Okay, why would they be doing this?" This seems great as a way to get started in a field. It's what I did.
>
> But then they just continue and stay on this path, basically, for years as far as I can tell, and they don't really update their models of "Okay, and this is how the work that I'm doing actually leads to the outcome." They don't try to look for flaws in that argument or see whether they're missing something else.
>
> Most of the time when I look at what a person is doing, I don't really see that. I just expect this is going to make a lot of their work orders of magnitude less useful than it could be."

What does it mean to "form your own views"? I mean something like forming a detailed model, starting from some basic and reasonable beliefs about the world, that gets you to a conclusion like 'working on AI alignment is important', or 'this research direction seems like it might shift the needle on AI-induced x-risk', or 'Power-seeking AI poses a decent chance of extinction', without having to defer to other people. Ideally that model has depth, so that if you double-click on any part of the argument chain, there's likely to be substance there. Good examples of this kind of reasoning include Buck Shlegeris' My Personal Cruxes for Working on AI Safety, Richard Ngo's AGI Safety from First Principles, and Joseph Carlsmith's report on Existent ial Risk from Power-Seeking AI."

Why form your own views?

- **You'll do much better research.** When work is open-ended, you need your own conceptual framework to work within, and you need to consult that framework to decide what to do. Even within areas like robustness or interpretability, there are directions that are orders of magnitude more valuable than others. We mostly don't have shovel-ready projects at the moment: If we were sure that solving the Alignment Theorem™ or implementing the Alignment Technique™ were sufficient to prevent AI-induced extinction, there would be no need to form your own views. Unfortunately, that isn't the world we live in.
- **You might be able to improve the frontier of knowledge:** There is widespread disagreement about what work might help—most researchers think that most alignment research that isn't theirs is ~useless-to-harmful. That is really a weird place to be. Very few people have thought through the whole problem in detail. And a lot of the best work from the past few years has come from (junior) people thinking big-picture.
- **It's motivating to do work you believe in, especially if you're doing research.**

You don't need your own views straight away, and maybe not at all:

- The more control you have over the direction of your work, the more you'll want to invest in forming your own views; research leads need this a lot more than research contributors: DeepMind is happy to talk to research contributors who have very basic alignment knowledge (e.g. "can explain instrumental convergent subgoals"), which means you could usefully contribute without ever forming detailed views about alignment. Though in this case, you'd have to trust DeepMind to be doing valuable research, and you wouldn't be able to lead research.
- You can form your own views gradually. Certainly while learning ML/engineering/research you won't need strong views on alignment. But you'll want to have thought about alignment pretty carefully by the time you're choosing what to study at PhD level, for example.
- Being told that forming from your own views is important can be scary and paralyzing. If you are scared or paralyzed, I suggest reading How I Formed My Own Views About AI Safety.

How do you form your own views? Here are some ideas:

- **Model other people:** talk to people and try to understand their views (actively reading someone's work also works, maybe in advance of talking to them). Build a model of what they believe and why, and then try to integrate that into your own model and your models of other people. You can ask them, for example, "Do you have a theory of change for how your research reduces x-risk, and could you try explaining it to me?". Then paraphrase until you'd be able to describe their view to someone else—you won't understand by default; paraphrasing makes it easy for them to correct you—and follow up with any confusions you might have.
- **Conduct a minimal trust investigation** (Karnofsky, 2021).
- **Go through some Alignment research exercises** (Ngo, 2022). These exercises are quite specific, and might not be aimed at the most important stuff. But they are structured, which is nice.
- **Think from first principles:** open up a blank google doc, set a one hour timer, and start writing about a question, maybe one from below, or about whether AI alignment is the most important problem for you to work on.
- **Refine an Impact Plan:** choose something that might have an impact—like your career or a programme or a research agenda—and write down your current, implicit, end-to-end story for how that thing produces value, noticing and any holes or leaps you're making. Make sure to end the story with the value being created, e.g. "somehow, we don't all die". Then make that plan as detailed as possible, for example by telling stories about specific people/research/organisations/actions, all the while paying attention to places you're confused. By the end, you'll have a better sense of how and whether the plan is likely to achieve value, of where the weaknesses of the plan are, and of where you're confused.

Forecasting questions:

- "How likely is extinction from AI" is the main one. Then there are some sub-questions that feed into that:
    - "When will the first AGI be developed?"
    - "What will the world look like in the five years before and one year after the first superintelligence"

    - "Will bad things mostly happen from alignment failures or coordination failures?". More broadly, "what do you expect market forces to cover and what do you expect to be neglected?"
    - "If we get alignment right on the first try, how likely is it that a misaligned AGI is later deployed?

Technical questions:

- "What alignment strategies actually align the AI?" (This is where a lot of disagreement is in practice.)
- "What alignment strategies are scalable and competitive?"
- "How can we make sure that the theoretically possible good strategies are practical enough to be implemented?"

Resources:

- Rohin Shah's Career FAQ (ctrl+F for "How can I do good AI alignment research?").
- How I Formed My Own Views About AI Safety (Nanda, 2022).
- Want to be an expert? Build deep models (Bye, 2021).
- Some thoughts on deference and inside-view models (Shlegeris, 2020).
- Buck's talk on orienting to safety research.

Funding

People don't apply for funding enough. Here are some rebuttals to common objections to applying for funding: You don't need to be doing valuable AI alignment research right now in order to get funded; students are prime targets for funding, because money is likely to be particularly useful to them; getting rejected probably won't negatively affect you down the line, as long as you're honest and well-intentioned; often people are miscalibrated about whether their proposal is worth the money; grant-makers really want to fund good projects.

What can you apply for funding for? Here are some things that you could apply to the Long Term Future Fund (LTFF) for:

- **Learning time:** to self-study ML or AI alignment, to do research assistant work at a university (totally fine if this isn't alignment research), to visit AI alignment hubs to speak or work with the people there.
- **Independent work:** direct independent alignment work (see this), or to help build the AI alignment community at your university or elsewhere.
- **Personal/productivity stuff:** Paying for a therapist (link to depression test—I know many smart people who took a year to realise they had depression), a productivity coach, a nice laptop or desk or chair, a workspace, your own productivity budget, buying you out of any non-useful work you have to do to live—including teaching buy-outs.

- **Tutoring:** for ML, coding, university classes, econ (?), or funding for an ML or coding bootcamp.
- **Formal degrees:** scholarships are available for undergraduate and postgraduate degrees; grants are available for conference fees and PhD application fees.

It is often easy to apply for funding – e.g. the application for the Long-Term Future Fund takes 1-2 hours.

How to apply: Aim to have an application that is honest and straightforward. If the point is to help directly with alignment, give your best guess as to whether and how your project helps alignment. If the point is to advance your career, write about how you expect it to advance your career relative to the counterfactual. If you don't have trustworthy signals of your competence and alignment, it helps to have a reference who knows you and is respected by the funding body. If you have either of those, consider applying immediately. If not, still consider applying immediately. But if you want a better shot, you might do an alignment project first and post it to LessWrong, for example as part of the AGI safety fundamentals seminar programme, or the ML Safety Scholars Programme (fully funded, applications close May 31st 2022), or as part of <building your own views on alignment from your own views>.

Funding sources:

- Long Term Future Fund. You can apply at any time, and they have a short turnaround. *I'd default to applying here.*

- Your university might fund you to do research with them.
- Open Philanthropy Undergraduate Scholarship.
- Open Philanthropy Early-career funding for individuals interested in improving the long-term future.
- The Open Phil AI Fellowship. For PhD students in technical AI safety (fairly competitive).
- Survival and Flourishing Projects (closely related to the Survival and Flourishing Fund).
- FTX Future Fund.

Broadly useful career advice

Look for ways to demonstrate your competence

I have mostly talked about how to become competent. This is the most important thing and it should be your main focus early on; it is also much easier to appear competent when you actually are. But when you start to do competitive stuff like job or PhD applications, it's useful to be able to demonstrate your competence in order to distinguish yourself from others.

Once you know which competencies to shoot for, find hard-to-fake signals that you are competent and work them into projects that build your competence. Search for ways to cache in on your

competencies/cool shit you do. You can also ask people in the community/employers what signals they'd consider hard to fake. For PhDs, doing research < ArXiv paper < published paper < published paper + reference letter from someone who has seen lots of students and has a good track record of predicting research success. Similarly, ML paper replication < open-source paper replication < open-source replication plus blog post about what you learned. Failed research < blog post about failed research… You'll probably soon have lots of knowledge/skills/cool stuff that you've done, that people won't know about. Sometimes, it's easy to transform those into a competency signal by making your knowledge/skill/cool stuff legible and visible.

Focus on becoming excellent early in your career

Most of your impact comes from later in your career. Early in your career (for the first few years out of undergrad, at least), your focus should be on doing things where you can grow and become excellent. You can ask yourself (and others) where you're likely to grow the most, and then go there. That might be alignment organisations, and it might not. Growth is largely a function of your environment and the mentorship available to you. The vast majority of good mentorship can be found outside of alignment, and alignment is heavily mentorship-constrained. If you become an excellent ML engineer/researcher or theoretical researcher, it will probably be easy to later specialise in empirical or theoretical alignment work. It is certainly fine (and maybe preferable, because of publications) to do non-alignment research as an undergraduate.

That said, it might not be good to become excellent if it means advancing AI capabilities. Though there is nuance in 'capabilities': working on improving Bayesian inference approximation (useless-to-maybe-helpful for alignment) is very different from scaling up large language models (probably pretty bad). However, Anthropic believe that staying at the frontier of capabilities is necessary for doing good alignment work, so I don't know how coherent the capabilities-safety dichotomy is (this is an active area of debate).

One way that working on stuff that doesn't help with alignment could go badly, is that you get stuck doing research that sounds like it helps but doesn't actually have a path to impact, like random robustness or interpretability research. *This can happen even if you join a safety team*. To avoid this, I recommend continuing to <build your own views on alignment>, speaking with more knowledgeable alignment people about your career decisions, and holding the intention to actually consider where you can do the best alignment research once you've built some skills.

Engaging with the AI alignment community will help you a lot

Why? I'm finding it a little hard to explain this. When I see people start to hang around in alignment communities, they seem to start doing much better stuff. That might be because they're supported or mentored, they pick up implicit knowledge, they're more motivated, or because they become aware of opportunities. Here are some ways to engage:

- 80,000 Hours advising. I encourage everyone to apply. It's a good way to get career advice and connect to alignment researchers. The application is quick.

- Being in places with AI people, like Berkeley, or to a lesser extent Oxford/New York/London. Talking regularly with AI alignment people is the main thing, and that can be done anywhere but is easier in some places than others.
- Effective Altruism Global. You can talk to people and make connections to get mentorship. Aim to have one-on-one's with people more senior than you. Tell them your plan and ask how you can improve it.
- AI safety support + their newsletter. They offer chats too.
- The AGI Fundamentals Fellowship has a community slack.
- Local (EA) groups, though they might not have a strong alignment community.
- Posting on LessWrong/the AI Alignment Forum.

Take care of yourself

I don't really know what to write here. I do know that taking care of yourself is extremely important. I burned out while trying to work on AI alignment, and can attest that burnout can be really bad. I don't feel super qualified to give advice here, but I do have some things that seem useful to say: If your work becomes a slog/grind that daunts you when you wake up, as opposed to a source of strong internal desire, I think that's worth paying attention to. You can take diagnostic tests right now or regularly for depression, anxiety, and burnout (takes less than 30 minutes in total). And maybe see a therapist if any of those are concerning, or preventatively, which you can get <funding> for. Having good mentors, managers, and buddies will help a lot.

Trying to work on AI alignment might be particularly bad for some people's mental health. Here are some reasons for that: Believing that we might all die might be really scary and totalising; there aren't that many jobs in alignment at the moment, and ML opportunities in general are pretty competitive; you might not be able to help with technical alignment work, and that might be crushing; some of the actions I suggest are hard and unstructured—such as forming your own views on alignment, or doing paper replications—and a lot of people don't thrive in unstructured environments; "technical AI alignment" is not a well-scoped career path or set of paths—and it's often hard to know what's best to do.

I don't want you to feel bad about yourself if you're struggling, or can't help in a specific way. If you're struggling, consider talking to your friends, people who have been through similar experiences, talking with AI safety support, taking time off, getting therapy, or trying a different type of work or environment.