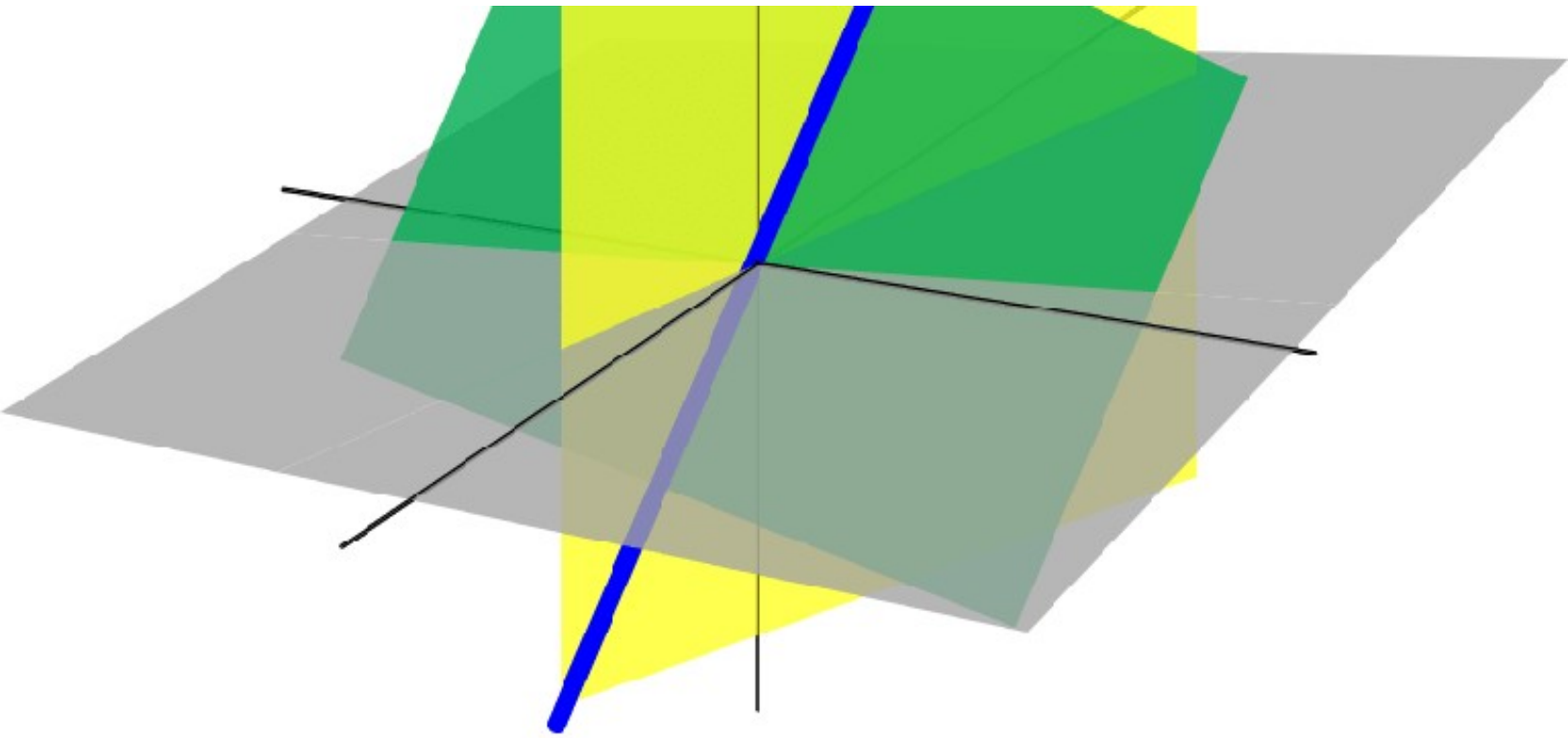


Hypothesis Subspace

1. Oversight Leagues: The Training Game as a Feature
2. Ideological Inference Engines: Making Deontology Differentiable*
3. Representational Tethers: Tying AI Latents To Human Ones
4. Interlude: But Who Optimizes The Optimizer?
5. (Structural) Stability of Coupled Optimizers
6. Boolean Primitives for Coupled Optimizers
7. Cataloguing Priors in Theory and Practice

Oversight Leagues: The Training Game as a Feature

Crossposted from the AI Alignment Forum. May contain more technical jargon than usual.

This post is part of my hypothesis subspace sequence, a living collection of proposals I'm exploring at Refine. Followed by ideological inference engines .

Thanks Adam Shimi for advice on putting more legible content out there. Thanks Eric Winsor, Leo Grinsztajn, Linda Linsefors, Lucas Texeira, Tammy Leake, Ze Shen for discussions which inspired this post.

TL;DR: An oversight league is a training scheme which incentivizes an agent and an evaluator to constantly try to game each other, leading to synchronized increases in capability for the two players. However, the evaluator is being offered a host of additional learning signals to help it maintain a consistent (and potentially provable) lead over the agent. Oversight leagues heavily draw on ideas from capability literature, including: league training in AlphaStar, game theory in GANs, adversarial robustness, etc.

Intro

The whole project of oversight leagues relies on the following non-exhaustive list of assumptions:

Assumption 1, "AGI Hard, Human Values Harder": We are unlikely to formulate the True Name of human values in closed-form before deploying transformative AI. The best we are likely to do before takeoff is model human values approximately and implement an imperfect evaluator.

Assumption 2, "Linear Capability Ordering": Any fixed evaluator (e.g. a reward model) can be gamed by an agent above a certain threshold of capability. More generally, an agent whose capability improves consistently faster than the capability of an evaluator will eventually be able to game said evaluator. By "evaluator capability," I'm referring to its ability to prevent being gamed.

Assumption 3, "Humans Are Not True Gamers": Human oversight is impractical because our capabilities as evaluators can't improve at an arbitrary large rate. Save for cyborgian schemes for human augmentation, human oversight would eventually be gamed by an agent of sufficient capability.

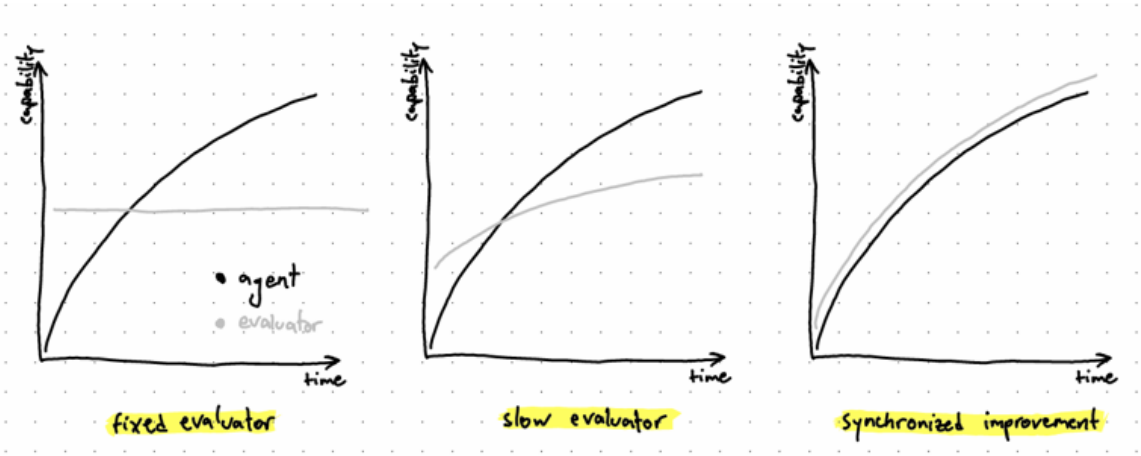
Assumption 4, "Zone of Proximal Development": There is a relatively small interval of evaluator lead which allows the agent to improve from challenging it, even if the evaluator is slightly ahead. This assumption is motivated by the fact that GANs are actually working, despite an inevitable imbalance between the two components involved at any given time.

Assumption 5, "Point of No Return": Even a minor lead in capability for the agent relative to the evaluator is catastrophic, as the agent can take advantage of this window of opportunity and maneuver itself into a persistent advantageous position by flying under the evaluator's radar.

From the assumptions above, we can infer that if we are to deploy a robust evaluator as an operationalization of human values, it must certainly not be fixed. Not only that, but it must additionally improve in performance at least as fast as the agent being evaluated, so that it never gets overtaken. What's more, simple human oversight doesn't fulfill those necessary conditions, hence we should consider automated schemes.

Proposal

An oversight league is one such automated scheme for training agents and evaluators in a way which improves their performance *in lockstep*. The crux of this training regime is to supply most of the training through **bilateral learning signals**, and thus render the improvement of the two components interdependent. By ensuring that most of the learning opportunities of the agent come from playing against the evaluator and vice versa, the two sides form a positive feedback loop resembling patterns of co-evolution. The oversight league scheme implicitly attempts to cultivate "antifragility" by applying appropriate stressors on the evaluator in the form of ever more capable agents a reliable way of improving its performance, and the other way around.



Various dynamics of agent-evaluator improvement. **Left**: Fixed evaluator inevitably being overtaken. **Middle**: Slow evaluator being overtaken at some point. **Right**: Synchronized improvement of agent and evaluator, aiming to avoid overtaking.

Additionally, besides the agent and evaluator constantly offering each other learning opportunities at a simultaneously increasing level, an oversight league requires **unilateral learning signals** applied differentially to the evaluator. The goal of such learning signals is to ensure that the evaluator has a constant edge over the agent, so as to avoid being permanently overtaken (see "Point of No Return" assumption) by chance. They are meant to provide additional learning opportunities for the evaluator, helping it patch its blindspots faster than the agent can exploit them.

Let's unpack the various learning signals involved in an oversight league, by first focusing on bilateral ones:

1. **Present Exploitation**: Training a player (i.e. agent or evaluator) against the latest and greatest version of the opponent. The agent is trained to exploit recent blindspots of the evaluator, while the evaluator is trained to pick out recent tricks of the agent. Inspired by "main exploiters" in AlphaStar.
2. **Past Exploitation**: Training a player against (a tricky mixture of) the past versions of the opponent. The agent is trained to exploit systematic blindspots of the evaluator across time, while the evaluator is trained to pick out systematic tricks of the agent. The role of past exploitation is to prevent the agent from forgetting about past blindspots of the evaluator and to prevent the evaluator from forgetting about past tricks of the agent. Inspired by "league exploiters" in AlphaStar.
3. **Future Exploitation**: Training a player against *future versions* of the opponent. This is somewhat trickier than the previous types of exploitation, in that it involves backpropagating through updates of the (boxed) opponent. It would only be applied to the evaluator, as it would help it avoid weight updates which are easily countered by

the opponent. This appears to be a promising place to look for formal guarantees on the safety of this scheme based on notions from game theory. Inspired by unrolled GANs from literature investigating ways of balancing the two adversarial components of GANs.

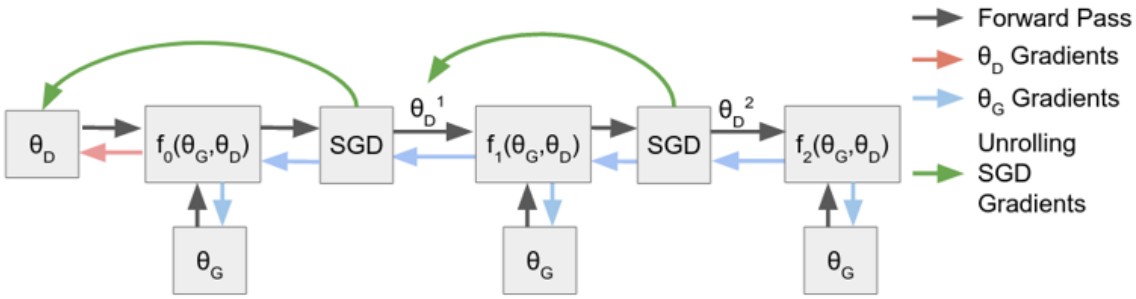


Illustration of unrolled GANs.

Now that we have listed broad classes of bilateral learning signals, let's focus on unilateral ones. As mentioned previously, those would mainly be applied to the evaluator as an attempt to systematically give it an edge.

1. **Adversarial Robustness**: Train the evaluator on data points which have been optimized to maximize error within tight bounds of allowed input perturbations. In image classification, allowed perturbations are of a magnitude too low to be consciously perceived by humans, yet trip up the ML model big time. When provided as training data, adversarial examples help the ML model become (empirically) robust against such attacks. Relaxation-based training as a specific approach to adversarial training can also provide formal guarantees on the worst-case bounds of adversarial attacks.
2. **Contrastive Dreaming**: This is a novel one, to the best of my knowledge. Just like adversarial robustness, it also involves training the evaluator on synthetic data points. However, instead of optimizing inputs to yield the worst possible prediction given imperceptible changes, contrastive dreaming consists in optimizing inputs to yield the seemingly best evaluation given significant allowed changes. In a sense, this technique involves *intentionally* *wireheading* the evaluator before using the optimized image of the world as a negative example. Think of the DeepDream technique being used to maximize the "dogginess" of an image as perceived by an image classifier, only to yield an image which is blatantly unrealistic with respect to how dogs actually manifest in the world.

Illustration of DeepDream amplifying the "dogginess" of an input image while blatantly violating the way dogs manifest in the world.

"*You were once the size of a period. For a brief time you tumbled about as a multicellular sphere, much like pond algae. Currents swept and washed over you. Remember? Then you grew. You became sponge life, tubular, all gut. To eat was life. You grew a spinal cord to feel. You put on gill arches in preparation to breathe and burn food with intensity. You grew a tail to move, to steer, to decide. You were not a fish, but a human embryo role-playing a fish embryo. At every ghost-of-embryonic-animal you slipped into and out of, you replayed the surrender of possibilities needed for your*

destination. To evolve is to surrender choices. To become something new is to accumulate all the things you can no longer be." — Kevin Kelly

Together, bilateral and unilateral learning signals provide the totality of learning opportunities accessed by the agent and the evaluator in an oversight league. The table below summarizes which learning signals would be applied to train which component:

Learning Signal	Type	Usage	Inspiration
Present Exploitation	bilateral	symmetrical	AlphaStar
Past Exploitation	bilateral	symmetrical	AlphaStar
Future Exploitation	bilateral	asymmetrical (evaluator only)	Unrolled GANs
Adversarial Robustness	unilateral	asymmetrical (evaluator only)	Robust ML
Contrastive Dreaming	unilateral	asymmetrical (evaluator only)	DeepDream

Let's sketch out more concrete details of the agent and the evaluator. The agent might be a purely agentic sequence-modeling entity exposed to the world, or could just as well be an agent simulacrum puppeteered by a simulator, as per Conjecture's recent taxonomy. Oversight leagues would be applied just as well in the latter case as in the former, probably in a way resembling Redwood's ongoing experiments of steering the simulator away from portraying human harm. It would be a specific agent simulacrum that is being steered, though, if oversight leagues were to be applied this way.

Over on the evaluator side, the component might resemble existing reward models used to extrapolate human feedback in RLHF, but would critically possess some adaptivity to be able to keep up with the agent-in-training (e.g. constant fine-tuning). In this RLHF-like case, the term process-based evaluator seems fit to distinguish a "behavioral" evaluation target from an outcome-based one, to build on Ought's taxonomy. An outcome-based evaluator would rather involve predicting the consequences which a given behavioral sequence would have on the world, and evaluating *that* instead of the behavior as an end in itself.

While the agent would be implemented in a relatively straightforward way through a policy, the evaluator involves more design choices when defining its type signature. Depending on the details of the unilateral learning signals, the evaluator might be tasked with *predicting the training epoch* associated with a past adversarial example or contrastive dream. This objective might incentivize it to gain an understanding of its systematic blindspots and an awareness of its training progress. While this sounds like an awful feature for the agent, it sounds quite handy for buffing the evaluator.

It's also relevant to point out that oversight leagues don't explicitly require transparency tools in their vanilla formulation. While the "Humans Are Not True Gamers" assumption hints at transparency tools used by humans not being deemed practical in this proposal, transparency tools might still be directly employed by players. The evaluator might benefit from probing the agent deeper in its mission of spotting its tricks, while the agent might similarly benefit from probing the evaluator deeper in its mission of spotting its blindspots. In this symmetrical setup, transparency tools feel like nice-to-haves, but not critical, as they only extend the portfolio of learning opportunities available to both agent and evaluator further. However, equipping the evaluator alone with an arsenal of transparency tools might help preserve its competitive edge.

Discussion

What if the agent internally simulates and games the evaluator?

If the agent internally simulates and games the evaluator, it means that the agent is more capable than the evaluator. The frame of oversight leagues acknowledges this state of affairs as an unambiguous game over, as the agent can maneuver itself into a persistent

advantageous position. That said, the entire premise of oversight leagues is that there are mechanisms which can help avoid the situation in which the agent is capable of gaming the evaluator in the first place. It's an attempt to make away with that whole chunk of attack surface, by enforcing the evaluator's edge in complexity in hopefully provable ways.

Wait what, you want to simulate the agent a few steps ahead of the evaluator?

The idea of future exploitation as a bilateral learning signal involves backpropagating evaluator updates through multiple update steps of the oversight league, in a fully differentiable way. This is meant to avoid evaluator updates which are easily to countered by the agent in the short-term, helping ensure the evaluator's edge. However, simulating the agent forward is dangerous. Fortunately, future exploitation might involve playing forward *both* sides, only for the updates to accumulate backwards into the latest version of the evaluator. Additionally, future exploitation might be conducted in a boxed setting, so that we ensure that new agent versions are being explored further in a safe environment first. The gradient update coming out of the box might be susceptible to gradient hacking by a woke agent screwing up the whole setup, though I find it unlikely that the agent would manage to pull that off without triggering a whole set of alarms in the evaluator's internals first.

What if the agent and evaluator collude?

While the agent's objective is essentially implemented through the evaluator, the evaluator's objective is not entirely based on the agent. It might likely involve assigning accurate GAN-like estimates of humanness or human desirability to input behaviors or outcomes, which is incompatible with helping the agent out through "grade inflation." The oversight league is a zero-sum game: the agent and evaluator can't both win, making collusion suboptimal.

What if the agent recursively self-improves without the evaluator?

RSI appears to be a concern for this framework, as the agent might gradient hack its way above the evaluator and thus break the agent-evaluator interdependence. An sensible approach would be to continue training the evaluator against the agent so that it also shares that momentum in capabilities. However, with a situational awareness of sufficient level, the agent might be able to both (1) gradient hack its way higher, and (2) sabotage the evaluator through its influence via bilateral learning signals. With a strong evaluator lead enforced through unilateral learning signals and future exploitation, though, the agent might lack resources to enact this form of sabotage.

What if assumption 1 ("AGI Harder, Human Values Harder") doesn't hold?

That's positively awesome. In this case, we discard this completely and go with an agenda focused on aiming directly at those unraveled human values, such as that of John Wentworth or Team Shard. That said, the rate of progress in ML appears to be in a whole different league than progress in ethics, pun intended.

However, it's relevant to note that this proposal still attempts to approach the True Name of human values *in the limit* of synchronized agent-evaluator improvement, despite assuming a low probability for a closed-form expression provided in the very beginning as-is. One might argue that the present setup might effectively be equivalent to the True Name as it aims to approach it, but I'm drawing a distinction between those two cases for clarity.

What if assumption 2 ("Linear Capability Ordering") doesn't hold?

This might mean that the evaluator would be better than the agent in *some* category of interactions, but not in others. If one imagines the superimposed radar charts of the two, the agent's skill profile might not be fully contained in the evaluator's one.

This is partially a question of how general a player's general intelligence is. If it turns out to be extremely general, the skill profiles would be pretty flat and hence less ambiguously

ordered. If it actually turns out to be more specialized, this assumption would take a hard hit, as it might be difficult to systematically enforce the evaluator's edge across the board.

What if assumption 3 ("Humans Are Not True Gamers") doesn't hold?

That's positively awesome. In this case, we similarly drop this proposal and go with a mix of agendas involving human amplification, such as the ones being investigated by parts of OpenAI, DeepMind, Conjecture, ARC, Ought, and others.

What if assumption 4 ("Zone of Proximal Development") doesn't hold?

Then the scheme might break down, as it would be impossible to improve the two components in lockstep based on each other. However, that would also mean that GANs shouldn't work, which they sometimes do, despite being infamously difficult to train. It might also mean that most supervised regimes shouldn't work, because of the huge initial capability gap between a fixed evaluator defined largely through training data and the tabula rasa agent.

What if assumption 5 ("Point of No Return") doesn't hold?

This situation interacts somewhat with the "Linear Capability Ordering" assumption, in that it hints at an underdog evaluator somehow being able to catch up with the leading agent. The myopic and non-agentic character of the evaluator makes me further reluctant of this assumption breaking, but if it did actually break, then differentially buffing the evaluator might help cover the lost ground. Additionally, the evaluator nerfing the leading agent might be feasible if this assumption wouldn't hold, say by signaling the need for reverting it a few epochs back. Though this then brings up corrigibility issues.

Is there an alignment tax associated with oversight leagues?

While many ideas here are employed to keep the agent *below* the evaluator in capabilities, the evaluator itself is not meant to be bounded in any way. This means that there's effectively no bound on agent capability, as it's encouraged to improve indefinitely, as long as it's a teeny-tiny bit less than how it could be (i.e. if the agent had the lead at any given point). Additionally, the enforced interdependence of agent and evaluator means that an improvement in the evaluator — even if it bounds the agent as described above — will eventually help elevate the agent through learning opportunities all the same. The main issue here is that the evaluator is resource-hungry, but I suspect any *less* resources allocated to the evaluator would yield an unsafe arrangement. Given the above, I consider the alignment tax to be negligible for this proposal.

Are oversight leagues restricted to prosaic risk scenarios?

They have been in large part motivated by prosaic work. I'd argue that oversight leagues are to AlphaStar what iterated distillation and amplification is to AlphaGoZero: a repurposing of a prosaic training regime towards safety goals (which incidentally has less capability externalities than entirely novel training regimes, as it's already present in the awareness of capability researchers). That said, I'd argue that the overarching scheme is general enough to transfer to non-ML player architectures, despite specific learning signals not being able to bridge the gap. For instance, contrastive dreaming is heavily inspired by quirks of ML models trained through gradient descent, where the optimization is applied to the input itself rather than the weights.

Outro

Oversight leagues are a natural conclusion of the initial assumptions regarding the agent-evaluator relation given shifting capability levels. This more epistemically legible format helped me clarify my thinking around the topic, and I'm looking forward to fleshing out

similar posts in the future based on different assumptions and analogies. Let the training games begin!

Ideological Inference Engines: Making Deontology Differentiable*

Crossposted from the AI Alignment Forum. May contain more technical jargon than usual.

* *Rather, making deontology play well with differentiable systems trained end-to-end.*

This post is part of my hypothesis subspace sequence, a living collection of proposals I'm exploring at Refine. Preceded by oversight leagues , and followed by representational tethers .

TL;DR: An ideological inference engine is a mechanism for automatically refining a given propositional representation of human values (e.g. a normative charter, a debate stance) in an attempt to disambiguate and generalize it to novel situations. While the inference algorithm and the seed representation form the crux of the system, a multi-modal entailment verifier is employed to order possible agent behaviors based on their compatibility with the estimated ideology. This proposal then describes a way of instilling deontological drives in prosaic systems while maintaining the appeal of end-to-end differentiation. Ideological inference engines draw on ideas from traditional expert systems, but replace much of the clunky symbolic manipulation with contemporary LLMs and NLI models.

Intro

Ideological inference engines are a slightly more general framework than oversight leagues, in the sense they rely on several global assumptions, but each more concrete instance of the proposal requires new assumptions when designing the seed representation, the inference algorithm, and the entailment verifier. Here's a non-exhaustive list of global assumptions:

Assumption 1, "Small Seed To Big Tree": Given a suitable inference algorithm and a finite propositional representation of human values, it is possible to estimate human values arbitrarily well given arbitrary amounts of compute. "Arbitrarily well" refers to there being an arbitrarily low error in the estimation. In the limit, the seed knowledge base would grow into the True Name of human values in propositional form.

Assumption 2, "Linear Capability Ordering": Similar to the assumption invoked in oversight leagues, this states that a system composed of a fixed knowledge base and a fixed entailment verifier would eventually be gamed by an agent whose capability is constantly increasing. This is due to the more complex agent becoming able to exploit the inaccuracies of the knowledge base with respect to actual human values.

Assumption 3, "Quantifiable Propositional-Behavioral Gap": The compatibility of a proposed behavioral sequence and a propositional representation of human values is computable. There is a fundamental relation between one's values and one's actions, and we can measure it. A close variant appears to be invoked in the CIRL literature (i.e. we can read one's values off of one's behavior) and in Vanessa Kosoy's protocol (i.e. we can narrow in on an agent's objective based on its behavioral history).

Assumption 4, "Avoidable Consequentialist Frenzy": It's possible to prevent the agent-in-training from going on a rampage in terms of an outcome-based objective (e.g. get human upvotes) relative to a simultaneous process-based objective (i.e. the present deontological proposal). This might be achieved by means of myopia or impact measures, but we're not concerned with those details here — hence, an assumption.

Together, those global assumptions allow for a mechanism for approaching the human values in propositional form, before employing the resulting representation to nudge an ML model towards being compatible with it.

Proposal

Ideological inference engines (IIE) are a modular framework for implementing such a mechanism. Each such system requires the following slots to be filled in with more concrete designs, each of which has attached assumptions related to whether it's actually suited to its role:

1. **Knowledge Base (KB)**: The way human values are supposed to be represented in natural language. The same representation medium would be involved throughout the refinement process, starting with the seed knowledge base being explicitly specified by humans. Those are some example representations to fill up this slot of the proposal with:
    1. **Debate Stance**: There is an ongoing dialogue among different idealized worldviews. The ideology which humans want to communicate to the ML model is initially represented through one particular side which is participating in the debate. It consists of a finite set of contributions to the dialogue in their respective contexts. What's more, the other sides taking part in the debate represent negative examples for the desired ideology, providing additional bits of information through contrasts. As the inference algorithm is employed to refine the seed representation, different contributions would populate the knowledge base in a similar way, representing more and more pointers towards the underlying ideology.
    2. **Normative Charter**: In this other way of filling up the knowledge base slot of the proposal, humans would try their best to communicate the desired ideology through a large set of axioms expressed in natural language. Those wouldn't need to be logically coherent, just like the mess that is human values. They might be descriptive, or they might be prescriptive. As the inference algorithm further develops the seed which humans put in, the knowledge base would contain possibly more and possibly different propositions meant to capture the underlying ideology.
    3. **Stream of Consciousness**: In this other format, humans would fill up the seed representation with their verbatim thought processes as much as possible. The particular humans involved in the process would implicitly be the representation target. As the inference algorithm does its thing, the knowledge base would contain extrapolated versions of the streams of consciousness involved.
2. **Inference Algorithm ($\vdash$)**: The means of refining a given knowledge base by producing a slightly more comprehensive and nuanced version. This component is generally static, as the knowledge base is the locus of adaptability. One way of looking at the inference algorithm is as a gradual optimizer exerting pressure on the knowledge base in an attempt to refine it. Another way of looking at it is as a transition function applied to the seed, iteratively turning it into a result, similar

to a rule set for the Game of Life. Those are some ways one might fill up this slot:

1. **Babble**: The knowledge base might be trivially extended by prompting a LLM to generate new propositions in the same vein with the old ones. This builds on the large amounts of implicit knowledge captured by the LLM during conventional training. This implicit knowledge is initially called forth to help disambiguate what the humans meant by the seed representation and overspecify it using new propositions.
2. **Babble & Prune**: Similar to babble, but instead of openly populating the knowledge base with just about any new proposition the LLM might come up with, a pruning step is employed to filter the LLM's suggestions. While pruning appears to be a slot to be filled in itself, one low-hanging fruit choice for it would be to only greenlight propositions which garner a certain amount of "support" from the previous knowledge base, as measured through pretrained NLI models (or LLMs prompted/fine-tuned to mimic them). This feels like an abductive choice for the inference algorithm, in that the hypotheses are ranked by how plausible they are to follow from the premises.
3. **Debate**: Similar to babble, but focused on extending a dialogue-like representation forward. The different sides taking part in the debate would be simulated together, helping ensure their relative consistency. Multiple future developments of the present state of the debate might be considered, to code for uncertainty. As another important difference from babble, pretrained NLI models would be employed to elevate the debate to a certain standard of quality by incentivizing the underlying LLM to simulate stances which are consistent and competitive. Consistency would be operationalized as the contributions of a given side generally supporting each other (via NLI), while competitiveness would be operationalized by encouraging the contributions of a given side to ruthlessly contradict opposing sides (via NLI). **Note**: In contrast to other alignment proposals making use of debate, the present one uses it solely as a means of indefinitely extrapolating a representation of human values expressed as one of the sides, rather than as e.g. a means of uncovering deception among superintelligent entities. The goal isn't to win the debate, it's the journey that counts as an attempt at memetic modeling.
4. **Counterfactual Cross-validation**: This inference algorithm includes a heuristic for focusing computational resources on patching the knowledge base in regions which are particularly brittle. It goes as follows. If a random subset of the knowledge base (e.g. a given proportion, or a fixed number of propositions), would the others together still prevent the ML model from violating them in a boxed setting? If said principles turn out not to be redundantly encoded in remaining knowledge base, then focus babble & prune efforts to patch up that part of the map of human values, as an attempt to avoid brittle directives.

3. **Entailment Verifier (⊨)**: The means of estimating the compatibility of a knowledge base with a given behavior. Does not really interact with the inference algorithm. Entailment verifiers might be used to rank a set of final behavioral sequences considered by the model, or provide an earlier heuristic for guiding the decoding process. Here's how you might design one:
    1. **Multi-modal Sequence Models**: A multi-modal sequence model resembling Gato or Flamingo might be employed to estimate the propositional-behavioral compatibility by means of autoregressive coherence. If the propositions contained in the knowledge base are likely to follow from the behavior, or if the behavior is likely to follow from the

propositions, then the compatibility is high. If one isn't really likely to follow from the other, then compatibility is low. However, the resulting value might be difficult to interpret, largely because of the confounding likelihood of the behavior and of the propositions themselves, outside of conditional influences on each other. An alternative to account for this might involve generating negated versions of the propositions and measuring the *relative* likelihood of the original ones to follow from or yield the behavior compared to the negated ones.

2. **Overton Probing**: While in the previous design the locus of bringing together the propositional and the behavioral was in the modality-agnostic sequence being modeled, in Overton probing this relating process happens explicitly on the propositional side. The idea here is to first generate either (1) a textual description of the behavior being considered, or (2) a textual description of the predicted outcome of said behavior. The entailment verification would consist of gauging the support garnered by the textual description from the propositions which make up the knowledge base, potentially through pretrained NLI models again or LLMs prompted to carry out the same task.

By filling all the three slots of an IIE with concrete designs, you get a pipeline which is able to measure the compatibility of an ML model's behaviors with a deontology being constantly refined. A straight-forward way of quickly using this to actually nudge the model being trained is to swap out the human feedback in RLHF for the compatibility estimate yielded by the present system. It's relevant to note that this "differentiable deontology" is only meant to provide one part of the final objective function, with the "Avoidable Consequentialist Frenzy" keeping other parts from completely overpowering it as a drive. IIEs are meant to be employed as correcting terms attached to a main objective, at least during takeoff.

The inference algorithm would regularly be employed to improve the estimate of human values by updating the knowledge base, so that the whole system stays adaptive. However, the inference algorithm can only bring the representation that close to the ground truth, without ever reaching it exactly in finite time. Its goal is to bring it as close as necessary for the main model being trained not to be able to game it during the training process.

> **Aside**: The present proposal is a generalization of two previous ones, termed memetic colonies and deontic arrays. This leaks somewhat through the different available choices for ingredients required to implement an IIE, as they're somewhat correlated:
>
> - **Debate Stance** (KB) + **Debate** ($\vdash$) + **Overton Probing** ($\models$) = memetic colonies
> - **Normative Charter** (KB) + **Counterfactual Cross-validation** ($\vdash$) + **Multi-modal Sequence Models** ($\models$) = deontic arrays
>
> That said, I believe the present generalized proposal can still yield generative effects (i.e. combinatorial mix-and-matching possibilities), despite certain components working better together than others. Plus, it allows new choices for each of the three components which didn't intuitively fit with the two previous proposals.
>
> However, the generalized proposal might inadvertently obscure some useful features of the previous ones. One feature whose loss feels unfortunate is the idea

of looking at the propositions comprising the knowledge base as *discrete* individual directives nudging the ML model in different ways (e.g. attractors, repellers, dipoles, etc. across state space or model space), whose associated "field theory" might yield clues as to the structure of the optimization surface (e.g. distribution of local minima, workable stable orbits, etc.). Those nudges get obscured somewhat through the merging of directives into a unified knowledge base *before* considering their influence, rather than after. I plan to skim through a textbook on electrodynamics to get a better sense of whether there's value here, and try to perform a conceptual rearrangement if so.

The general structure of this proposal is directly inspired by the structure of traditional expert systems, which also feature a knowledge base, an inference engine, and a way of determining entailment. Whenever touching on a field as grounded in theory as logic and reasoning through an alignment proposal, it seems appropriate to look for potential guarantees which could be transferred over. In oversight leagues, game theory and provable robustness feel like potential candidates for sources of those warm and fuzzy guarantees. For the present one, the following concepts seem relevant to look into further:

- soundness and completeness as ways of relating inference and entailment in meaningful ways,
- forward and backward chaining as ways of driving the development of the knowledge base in relation to the target of verification,
- proof systems associated with many-valued logics (e.g. fuzzy logic) as ways of navigating the messy nature of natural language inference.

Discussion

Can't the ML model being trained game the IIE?

Given a fixed knowledge base, any given agent of increasing capability will at some point be able to game the IIE. However, the way this proposal handles gaming is by actively patching loopholes in the input deontology over time. By constantly making the objective associated with IIE more nuanced, the model-in-training would find it more and more difficult to exploit blindspots in the aggregate system, especially in the propositions. While in oversight leagues the locus of objective adaptability was the evaluator as an ML model being constantly fine-tuned, in the present proposal it is in the constantly expanding knowledge base. Whether or not the inference algorithm can fulfill its goal in refining estimates of human values becomes a crux.

However, there are also static components involved in the present proposal. The various inference algorithms and entailment verifiers proposed above are quite fixed. Not only are they fixed in algorithmic flow (e.g. repeatedly Babble and Prune), but also in the frozen ML models they involve (e.g. pretrained NLI models). This makes for some fragile attack surface in the proposal. It's unclear whether adaptability at the level of the knowledge base is enough to maintain the whole IIE above gaming, given that two of its three components are static. Future variants of IIE might involve learning the inference algorithm and entailment verifier as subtasks.

What if the ML model recursively self-improves and overtakes IIE capabilities?

The model-in-training might plausibly exhibit RSI, especially given high situational awareness. The only defense of the present proposal against this threat is to match the improvement with an analogous one for the IIE, which currently can only happen through the inference algorithm extending the knowledge base and thus refining the objective. Heuristics for guiding the expansion towards propositions relevant to the current behavior of the current model (e.g. counterfactual cross-validation) might help better align the two improvements. I just realized that the very consideration of such heuristics as useful runs against the "Linear Capability Ordering" assumption, leading to inconsistencies in the present version of IIE.

What if the ML model tampers with the IIE?

The model-in-training might plausibly try to hack into the server farm which would run the IIE system and hijack its reward center. The "Avoidable Consequentialist Frenzy" assumption invokes other lines of work to protect against rampages in terms of the main objective, as opposed to the auxiliary objective implemented through an IIE. However, this tampering scenario highlights the fact that tampering with the deontology can be a consequentialist move in itself, without even taking the main obejctive into consideration. This makes it clear that the same mechanisms invoked for stabilizing the main objective (e.g. myopia, impact measures) should be used to protect against rampages in the deontological objective. This hints at the awkward need of supplementing the deontological refinement carried out by the inference algorithm with such more general mechanisms.

However, the IIE itself might help enforce behavioral norms which radically discourage the model-in-training from tampering with the IIE, without the need for employing other tricks. Alternatively, the goal of running the IIE step-by-step, as explicitly implemented by humans, might be a goal in itself. Any slight divergence from running the authentic IIE line by line would result in an infinitely large penalty. Goal preservation becomes an explicit goal, but then what if this meta-goal of not tampering with its original goal becomes the object of tampering? Adding a goal preservation goal for its first goal preservation goal only shifts the problem upwards one more step. There's a need for folding this meta chain on itself somehow.

As another approach to tackling wireheading, one might consider the fact that the IIE itself can be fully deterministic. Even the components which involve LLMs decoding stochastically can fix a seed for the PRNG and get perfect reproducibility. This means that the IIE can be run on different sealed systems and the hash of the expanded knowledge base(s) can be reported. Inter-annotator agreement schemes would then be employed to detect tampering. However, if the risks associated with hacking one IIE replica are correlated with the risks of hacking the others, then the added benefit of replication collapses. Perhaps increasingly well-protected instances could be brought together to check for consensus, where the model hacking some of them would trigger warnings. But then the model would hold off until being capable of hacking all of them, side-stepping disagreement warnings entirely — all instances would be hacked, preserving consensus.

What if the knowledge base grows unwieldy in size?

For this one, I'll use a similar reply to the question of alignment tax in oversight leagues. The knowledge base would indeed grow over time to stay ahead of the model. Any less growth might not be enough to avoid gaming the deontological objective implemented through IIE.

However, the knowledge base would just be text. A lot of text, but just text. The bigger concern is whether the computational cost of the inference algorithm and the entailment verifier grows with the knowledge base. For most choices listed above, the complexity is linear with respect to the size of the knowledge base. There might even be ways of reducing it to sublinear, with the two components first filtering the knowledge base so as to only work with a relevant subset of propositions, say through domain-specific heuristics (e.g. counterfactual cross-validation) or general measures of similarity. This brings in new issues, though. A detailed analysis of the complexity of the various choices for $\vdash$ and $\models$ doesn't feel like a priority — there are way bigger issues with this proposal than the estimated alignment tax.

Doesn't this proposal tacitly assume moral absolutism?

The seed knowledge base would be finite. It would define the deontology which the inference algorithm then refines and the entailment verifier enforces. While humans could choose what goes into the knowledge base democratically, it's unclear whether the result would be anything but completely contradictory. It might also be possible to have multiple seed deontologies expanded in parallel which get a say in the model's behavior proportional to the number of moral patients subscribed to said deontology. While all this would be nice, effectively enforcing one deontology is many OOMs harder than enforcing multiple of them, as a technical challenge. Hence, the proposal (and potential future variants) will likely focus on "just" enforcing one.

What if assumption 1 ("Small Seed To Big Tree") doesn't hold?

Then it's impossible to get an arbitrarily accurate estimate of human values with arbitrary amounts of compute involved in modeling. There's just not enough information to help disambiguate the seed into what humans truly want. This appears similar to image superresolution or related upscaling tasks, where there are multiple high-resolution images which can map to the same low-resolution one. The upscaler is forced to choose one among an infinitely large set of valid items which downscale to its input. There is simply not enough information to pinpoint the right one. Following a transition function like the laws of physics to grow a tree from a seed turns out not to be enough here.

In this case, we might be forced to rely on the existence of one disambiguation without having a guarantee on its uniqueness. With this stunted IIE, we are still able to find a dentology which successfully "downscales" to the input we specified, but isn't exactly the one we possess. However, the fact that many choices of inference algorithms build on mountains of implicit knowledge about humans stored in LLMs gets us better off than the arbitrary choice of successfully-downscalable options. We can channel autoregressive coherence and NLI as nudges towards generally better choices, despite the best one remaining improbable.

What if assumption 2 ("Linear Capability Ordering") doesn't hold?

Then the model-in-training might game the IIE in some particular way, despite it generally being less advanced overall. The IIE might be forced to go over the top and make sure it's way more advanced in general so as to reduce the chances of being gamed in any way. In other words, if the skill profiles of the two on a radar chart exhibit many spikes, making the IIE's one way larger helps reduce the dimensions on which the model is ahead. Unfortunately, this increases the alignment tax and loses the guarantees of a hard ordering.

What if assumption 3 ("Quantifiable Propositional-Behavioral Gap") doesn't hold?

Then we can't implement a valid entailment verifier to help gauge the compatibility of behaviors with the propositional knowledge base. One might wonder: Why go to the trouble of making the knowledge base propositional in nature only for the entailment verifier to later be tasked with relating it to behaviors? Why not make the knowledge base behavioral in the first place?

There might be some value to this approach. The knowledge base would contain behaviors introduced by humans, rather than propositions. The job of the entailment verifier would be simpler, as the knowledge base and the target of verification would share the same modality. However, it's unclear how behaviors can be said to *support* other behaviors, except by means of precedents being interpolated. In contrast, the notion of propositional premises supporting a hypothesis is somewhat better defined. That said, the line between an action sequence and a subword sequence itself gets blurry as you consider agent simulacra manifesting in a linguistic universe, complete with a transition rule resembling the laws of physics. Most work around RLHF also reframes the subwords available to a LLM as possible actions, its parameters defining its policy as a guide for how to act in different context. The distinction approaches a moot point.

Relatedly, one might wonder: Why go to the trouble of having humans translate their implicit values in language, when there are a host of neuroimaging techniques available? Why not make a knowledge base of neural dynamics, possibly reduced in dimensionality to some latents?

Similar to the awkwardness of inferring new valid behaviors from past behaviors, inferring new valid thought patterns from past ones is very ill-defined. Barring all the limitations of current neuroimaging techniques in terms of spatial and temporal resolution, cost, portability, etc. it's unclear how to implement a compatible inference algorithm, except perhaps for the rudimentary Babble. However, the entailment verifier wouldn't face a more difficult challenge than in the propositional-behavioral setup, as it would need to bridge the neural-behavioral gap instead using multi-modal techniques.

What if assumption 4 ("Avoidable Consequentialist Frenzy") doesn't hold?

It was an honor to serve with you, have a nice timeline!

Are IIEs restricted to prosaic risk scenarios?

Although IIEs have been motivated by prosaic work, the proposal is entirely agnostic to the source of the behaviors to verify. In other words, even if the IIE would be built on a prosaic stack, the AI whose behavior should be aligned might be built on a different stack, given only that it is capable of optimizing for *something* (e.g. the deontological objective).

That said, even the IIE itself might run on a different stack. Case in point, IIEs have been inspired by a symbolic GOFAI stack supporting expert systems, parts of which have been replaced here with ML. This makes it plausible for other approaches to be able to populate the modular framework.

Representational Tethers: Tying AI Latents To Human Ones

Crossposted from the AI Alignment Forum. May contain more technical jargon than usual.

This post is part of my hypothesis subspace sequence, a living collection of proposals I'm exploring at Refine. Preceded by ideological inference engines , and followed by an interlude .

Thanks Adam Shimi, Alexander Oldenziel, Tamsin Leake, and Ze Shen for useful feedback.

TL;DR: Representational tethers describe ways of connecting internal representations employed by ML models to internal representations employed by humans. This tethering has two related short-term goals: (1) making artificial conceptual frameworks more compatible to human ones (i.e. the tension in the tether metaphor), and (2) facilitating direct translation between representations expressed in the two (i.e. the physical link in the tether metaphor). In the long-term, those two mutually-reinforcing goals (1) facilitate human oversight by rendering ML models more cognitively ergonomic, and (2) enable control over how exotic internal representations employed by ML models are allowed to be.

Intro

The previous two proposals in the sequence describe means of deriving human preferences *procedurally*. Oversight leagues focus on the adversarial agent-evaluator dynamics as the process driving towards the target. Ideological inference engines focus on the inference algorithm as the meat of the target-approaching procedure. A shortcoming of this procedural family is that even if you thankfully don't have to plug in the final goal beforehand (i.e. the resulting evaluator or knowledge base), you still have to plug in the right *procedure* for getting there. You're forced to put your faith in a self-contained preference-deriving procedure instead of an initial target.

In contrast, the present proposal tackles the problem from a different angle. It describes a way of actively conditioning the conceptual framework employed by the ML model to be compatible with human ones, as an attempt to get the ML model to form accurate conceptions of human values. If this sounds loosely relates to half a dozen other proposals, that's because it is — consider referring to the Discussion for more details on tangents. In the meantime, following the structure of the previous posts in the sequence, here are some assumptions underlying representational tethers:

Assumption 1, "Physicalism": Our thoughts are represented as neural dynamics. In the limit of arbitrarily large amounts of data on neural dynamics aligned with external stimuli (in the sense of parallel corpora), our thoughts can be accurately reconstructed.

Assumption 2, "Bottleneck Layer": There is a bottleneck layer in the architecture of the ML model being tethered to human representations. This bottleneck refers to a low-dimensional representation through which all the information being processed by the ML model is forced to pass.

Assumption 3, "AGI Hard, Human Values Harder": We are unlikely to formulate the True Name of human values in closed-form before deploying transformative AI. The best we are likely to do before takeoff is model human values approximately and implement an imperfect evaluator.

Proposal

Representational tethers suggest a way of aligning human and AI latents for the purpose of facilitating later interaction. There are two steps to this:

First Bring Them Closer

Incentivize the ML model to employ internal representations which are compatible with human ones, thus bringing them "closer." This can be operationalized by conditioning latent activations which arise in the ML model to be expressible in human representations. Concretely, optimization pressure would be exerted on the ML model to push it to internalize a conceptual framework *which can successfully be translated to and from a human one* without significant loss of information. If the artificial representation can successfully be translated into the language of e.g. neural dynamics and back, then the two are quite compatible. If there is no way of realizing the back-and-forth, then they're generally incompatible — you either can't express artificial ideas in human terms, or human terms aren't enough for fully expressing them.

The adequacy-fluency trade-off described in machine translation is particularly relevant to understanding this process. If an English sentence is translated into Chinese in a way which (1) successfully captures the intended meaning, but (2) comes across as formulated in a very unnatural way to the Chinese speaker, then the translation has high adequacy, but low fluency. If the sentence is translated in a way which (1) feels very natural to the Chinese speaker, but (2) fails to accurately capture the intended meaning, then it has high fluency, but low adequacy. The two features are often in tension — it's hard to perfectly capture the intended meaning and make it feel perfectly natural at the same time, especially as you dial up the complexity.

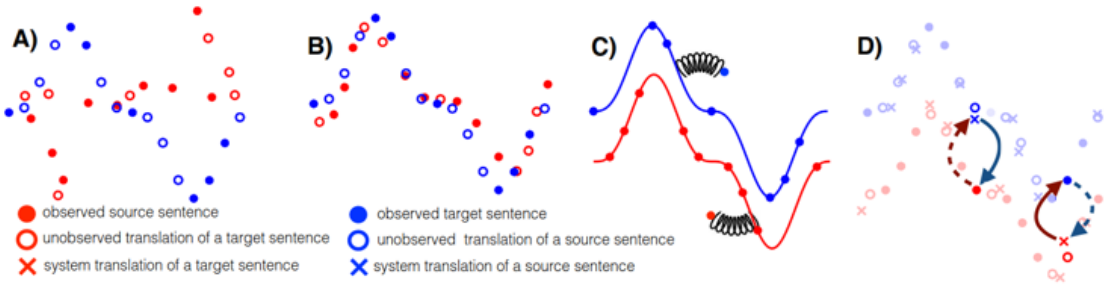


Illustration of various techniques involved in aligning two families of representations given limited parallel data (Source). **D**: Backtranslation.

The way this trade-off would apply to our backtranslation of latents is as follows. For perfect adequacy and disastrous fluency, you might choose two neural states as stand-ins for 1s and 0s, and perfectly express the artificial representation in binary. Adequacy is high because you can accurately represent the artificial representation and go back, but fluency is at an all-time low because the language of (human) neural dynamics doesn't resemble two states being endlessly interspersed. For perfect fluency and disastrous adequacy, imagine an ML model with particularly alien concepts which get translated into neural dynamics which are perfectly in-distribution, but completely fail to express the initial meaning, making it impossible to complete the backtranslation.

Considering this, it becomes obvious that backtranslation faces the same two conflicting goals: (1) capture the intended meaning so that you can go back from human to artificial (i.e. adequacy), but (2) ensure the intermediate human translation is in-distribution relative to observed neural dynamics (i.e. fluency). Note that fluency here doesn't require an

exhaustive mechanistic understanding of the brain (see "AGI Hard, Human Values Harder" assumption), just a measure of divergence between the ground truth distribution of neural dynamics observed in humans and the distribution resulting from translations of artificial representations.

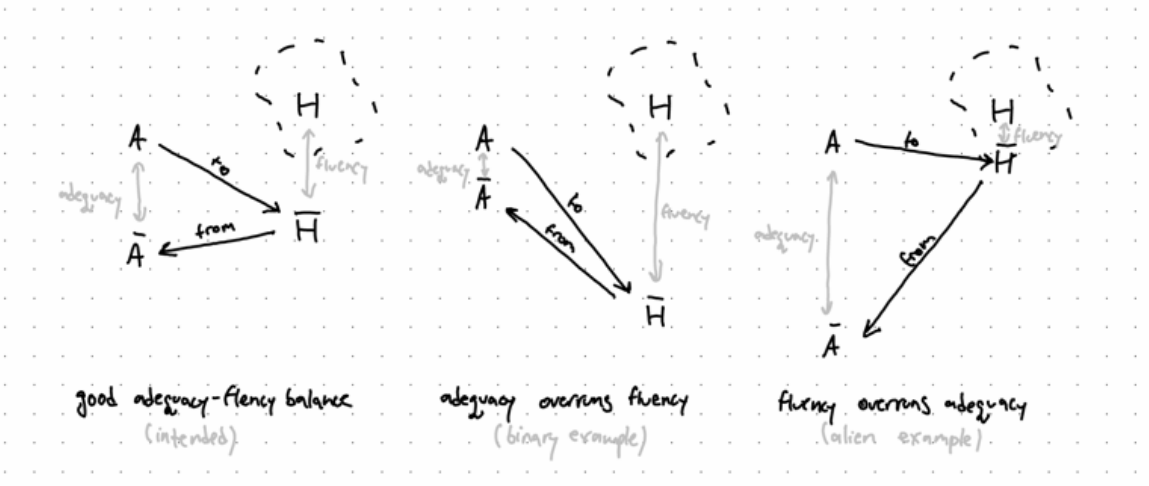


Illustration of various relations between adequacy and fluency. **A**rtificial and **H**uman representations are linked through translations depicted through black arrows. Grey double arrows are distances, where shorter is better. **Left**: Adequacy and fluency are well balanced. **Middle**: Better adequacy comes at the cost of worse fluency. **Right**: Better fluency comes at the cost of worse adequacy.

The ML model would then be incentivized (e.g. through gradient descent) to settle on representations which can be translated both adequately and fluently. If the artificial representations are alien, then it's hard to adequately translate them in a fluent way. Given this, representations which can be translated both adequately and fluently in human terms are more likely to be compatible with the ones we actually employ as humans.

But who or what performs the actual two translations involved in the proposal so far? Considering we're focusing on ML models, a straightforward choice for the architecture of the translator components is to go with ML models again. The artificial-to-human translator would be tasked with taking in internal ML representations and outputting neural dynamics. The human-to-artificial translator would be tasked with performing the reverse process. The adequacy loss would be computed based on the distance between the original artificial representation and the backtranslated one, before driving backpropagation. Similarly, the fluency loss would be computed based on the distance between the intermediate human representation and the ground truth distribution of them, before getting backpropagated. An appealing property of this setup is that the translators are learned as part of the initial training process. No extra hurdles involved.

Another relevant trick would be to *overwrite* the original artificial representation with the backtranslated one, rather than just using the backtranslated one to compute extra losses. By tying the backtranslation process to the main objective of the ML model, enabling successful backtranslation becomes not only useful for furthering auxiliary objectives — it becomes instrumental in its main dealings, especially the adequacy aspect of it. This overwriting might be applied constantly, or just occasionally, allowing the ML model to occasionally roam freely, untethered to the human domain of thinkable thoughts.

Then Bridge The Gap

While making the internal representations employed by the ML model more compatible with the ones used by humans on a neural level might be handy in itself, this increase in compatibility also makes it easier for humans to *consciously* bridge the gap and understand a full thought of an ML model. However, if aliens used exactly the same concepts as us but their language relied on dense high-dimensional waves, then it still wouldn't be easy for us to understand them. This means that even if the artificial representation can be translated to and from neural dynamics, we can't just watch phrases in the language of neural dynamics on screen and understand the associated thoughts. If we could have simply derived meaningful representations from neural dynamics, we could have figured out human values without this whole contraption (see "AGI Hard, Human Values Harder").

Humans Learn The Translation Key

One way of bridging the gap would be to first engineer the bottleneck layer to be particularly cognitive ergonomic. First, we might incentivize sparsity by regularizing the dense artificial representation. Then, we might discretize it by applying vector quantization and collapsing continuous values to a finite cookbook resembling a vocabulary. Then, we might incentivize local structure by promoting localized computations after the bottleneck layer, by e.g. partially limiting attention to neighborhoods. The combination of sparsity, discreteness, and locality would help make the bottleneck representation work more like a language familiar to humans. Next, make sure to also visualize the cookbook in ergonomic ways by making use of Gestalt principles, and ensure that humans can easily distinguish the symbols using pre-attentive features like color and orientation. The resulting language would be in part deliberately designed (through those surface features), but mostly learned by the ML model (as an instrumental goal in its optimization). Some insight into how the resulting language might feel:

> "*The language and the script I invented for Dara, Classical Ano, however, is much more distant from a purely phonetic script even than Chinese. So, in some sense, I decided to take a few peripheral features in a script like Chinese and centered them and pushed them much further until I created a new script that is not found in human scripts at all, at least among the ones we know. And then I dug deeply into this invented script—it's three-dimensional; it's malleable to the representation of different languages; it's simultaneously conservative and friendly to innovation; it's ideographical (in a true sense, not in the misunderstood sense when applied to Chinese hanzi)— and explored how that interacts with the way we think about writing.*" — *Ken Liu*

> "*These rules, the sign language and grammar of the Game, constitute a kind of highly developed secret language drawing upon several sciences and arts, but especially mathematics and music (and/or musicology), and capable of expressing and establishing interrelationships between the content and conclusions of nearly all scholarly disciplines. The Glass Bead Game is thus a mode of playing with the total contents and values of our culture; it plays with them as, say, in the great age of the arts a painter might have played with the colours on his palette.*" — *Herman Hesse*

> "*Almost all polis citizens, except for those who specifically elect otherwise, experience the world through two sensory modalities: Linear and Gestalt, which Egan describes as distant descendants of hearing and seeing, respectively. [...] Gestalt conveys information qualitatively, and data sent or received about anything arrives all at once for interpretation by the mind of the Citizen in all its aspects simultaneously, resulting in an experience of immediacy. [...] Gestalt operates rather entirely or almost entirely subconsciously. Citizens use Gestalt to create icons for themselves — "visual" representations within Scapes.*" — *Wiki on Diaspora*

The initial increase in compatibility, combined with the structural tweaks described before, should make it easier for humans to learn the internal language derived by the ML model. Humans would then learn it by association with the inputs and outputs of the ML model, before potentially identifying and exploiting grammar-like structures or other explicit regularities. Children might be particularly promising candidates for acquiring this unusual language, but deliberate practice might be enough for most linguistically-inclined adults. I recently came across a recent proposal of Eric Drexler which appears to point in a related direction.

Humans Experience Translated Neural Dynamics

People can't just watch neuroimaging recordings and infer the associated thoughts as they might read one's writing. However, what if the artificial-to-human translations from the previous step resulting in neural dynamics would be used to temporarily s**t**imulate the human overseers? Unfortunately, it's not obvious how neural dynamics of specific people might closely map to each other, or how one might induce a specific neural dynamic in an individual. Commercial brain stimulation currently feels more like "let's just overstimulate half of the brain with a huge electromagnet," far from the nuance required for the Godshatter-like guided experience replay we would need here.

Additionally, the artificial-to-human translation might yield longer neural dynamics than the representations being translated. If a day's worth of artificial representations get unrolled into a week's worth of neural dynamics, then human overseers wouldn't have the time to experience them. Factoring them in parallel might be one way to scale.

Tether Artificial Representations To Written Language

What if similar to how we tethered artificial representations to (human) neural dynamics in step one, we also (or instead just) tethered them to written language? This would make it easier for people to extract the contents of artificial representations by just reading the associated textual translation. The artificial-to-textual (and textual-to-artificial) translators would similarly be learned during the regular training process. In a sense, the ML model would be forced, in the middle of its processing, to convert its thoughts into (extensive) natural language and back, making it trivial for humans to understand them, and potentially even edit them in-place. In contrast to the previous two approaches to bridging the gap, this one appears by far the most realistic. Additionally, it would reduce the second step to a form of the first one, increasing the feasibility of implementation.

That said, it has its own issues, particularly the fact that consciously-articulated language is a worse proxy for human thoughts than neural dynamics in the limit of available data. Additionally, a similar issue to the previous approach: what if one second's worth of artificial representations have a reading time in textual form of an hour? It would be hard to keep up. However, we have experience in NLP applications. If we had access to the ML model's stream of consciousness in natural language, then we would have some understanding of how to automatically detect issues and correct it. The propositional nature of ideological inference engines might be an advantage here.

By following the two steps of (1) bringing artificial conceptual frameworks closer to human ones, and (2) enabling humans to understand and interact with them, we might be able to (1) improve the prospects of human oversight, and (2) promote the presence of human abstractions in ML models, including information on our desired objective.

Discussion

How do representational tethers relate to The Visible Thoughts Project ?

The Visible Thoughts Project aims to train LLMs on data containing snapshots of the internal thought processes of agents in a narrative. The idea there appears to be to teach this externalization skill in a supervised regime, using a custom representative dataset compiled for this specific purpose. Similarly, the version of the current proposal based on tethering to written language also aims to make the thoughts of an ML model visible, in that humans could simply read them out. However, in contrast to the Visible Thoughts Project, representational tethers don't require custom domain-specific data to teach thought externalization in natural language. Instead, the technique mainly relies on automated backtranslation between artificial representations and arbitrary natural language in the way described above.

How do representational tethers relate to The Natural Abstraction Hypothesis ?

My reading of this hypothesis is that certain abstractions are expected to be converged onto by any system of sufficient cognitive level acting in the (same) world. Such attractor abstractions are deemed natural. Assuming the hypothesis is true, it becomes relevant to check whether certain alignment-related abstractions such as "that which humans value" are natural and expected to exist in the ML model's conceptual framework. In contrast, representational tethers aim to enforce abstractions compatible with those of humans regardless of whether they are expected to naturally emerge in the ML model or not. In other words, the current proposal doesn't appear to rely on that assumption.

How do representational tethers relate to transparency tools ?

Most transparency tools are post-hoc, in that they aim to explain the inner workings of an otherwise illegible model. In contrast, by conditioning artificial representations to conform to human ones and automatically learning translators as part of the training process, the current proposal aims to make the model itself more transparent, obviating the need for additional modality-specific tools. That said, I'd point out Anthropic's SoLU as a specific way of improving the inherent transparency of ML models by rendering their representations sparser, and hence more cognitively ergonomic. This specific line of work relates closely to the broader goal of bridging the gap by making artificial representations work more like human languages (i.e. sparse, discrete, local, etc.). That said, I think the backtranslatability incentive as a way of improving human-artificial compatibility in conceptual frameworks is a qualitatively different but complementary intervention.

How do representational tethers relate to brain-like AGI ?

The present proposal doesn't require a satisfactory mechanistic understanding of neuroscience, as the versions which tether to neural dynamics rely entirely on learned models. In contrast, the very premise of the brain-like AGI agenda appears to be that insights from neuroscience would fuel AGI development to the point of driving important architectural decisions.

How do representational tethers relate to quasilinguistic neural representations ?

This is Eric Drexler's recent proposal on making the internal representations of contemporary architectures more language-like in an attempt to make them more easy to interpret, interact with, and manipulate. There is a huge overlap between QNP and the "Humans Learn The Translation Key" approach to the second step of the present proposal, in that they both point to some desirable properties of the artificial representations as precursors in bridging the gap. Incidentally, this also makes me aware of the connection to Anthropic's SoLU by transitivity. Safe to say that I'm really curious how QNP will unfold, though I'm more reluctant to that particular approach to bridging the gap working out in contrast to the "Tether Artificial Representations To Written Language" angle. Current results in NLP make that approach feel particularly realistic.

How do representational tethers relate to Microscope AI ?

From my reading, Microscope AI aims to train a system explicitly for the sole purpose of extracting and making use of the knowledge it synthesized during training, without necessarily deploying the system in a more active sense. While representational tethers also aim at making the system's internal thoughts more legible to humans, they're meant to be employed during active use of the system. There's a focus here on oversight and targeting during the system's ongoing usage, as opposed to a static extraction of knowledge.

Isn't the model incentivized to tweak (human) neural dynamics or language?

Recall that the ML model is not directly incentivized to employ a conceptual framework compatible with that of living, breathing humans. Rather, we assumed static data on neural dynamics or written language as a means to define that distribution. That has the added benefit of avoiding human modification enacted by the ML model in an attempt to make (human) neural dynamics closer to its arbitrary conceptual framework. The model should be tethered to a fixed place, because otherwise it would be tempted to pull the other way. That said, this is also a disadvantage, because fixed objectives defined through fixed datasets are prone to being outperformed in nuance at some point down the line, due to the model's increase in capability.

What if fluency is worth sacrificing for the main goal?

The compatibility of artificial representations with human ones isn't meant to be the only goal being pursued. The ML model would likely be trained towards a more practical goal. Considering this, the present proposal doesn't aim to be sufficient in solving alignment, it just sketches out a convenient part — that concerned with making the ML model's representations more human-compatible, rather than the ML model as a whole in its behavior. This means that the main goal still needs to be specified, which raises the problem of how to balance the two. This feels similar to the "Avoidable Consequentialist Frenzy" from ideological inference engines. Both describe an auxiliary objective meant to encourage convenient properties in the model, but are susceptible to being entirely ignored in the ML model's pursuit of the main objective. Perhaps the goal would be to get good human ratings, and manipulative behavior would in this case be avoided by first having the human overseers read the translated thoughts of the ML model, before deciding whether or not to let it carry out its plan (see discussion on defeating the mind-readers below). Alternatively, the internal abstractions chosen as a result of tethering might help define the objective themselves.

What if the translator is arbitrary?

Perhaps whenever the ML model's thoughts get translated in e.g. natural language, every instance of "red" is swapped with "blue" on the surface and vice versa, despite the ML model representing the other thing. When perceiving a red flower, the translator would report a blue flower. Both artificial-textual and textual-artificial translators would be in on this, ensuring high adequacy. Additionally, replacing "red" with "blue" flowers also preserves fluency for the most part. That said, red and blue flowers have different connotations in other contexts, reducing the fluency of e.g. text associating blue flowers with human romance more than red ones.

However, how can one guarantee that the translators don't simply make use of natural language as a secret code to mean something entirely different?

> "*It is a revelation to compare the Don Quixote of Pierre Menard with that of Miguel de Cervantes. Cervantes, for example, wrote the following (Part I, Chapter IX):*
>
> *...truth, whose mother is history, rival of time, depository of deeds, witness of the past, exemplar and adviser to the present, and the future's counselor.*

> *This catalog of attributes, written in the seventeenth century, and written by the "ingenious layman" Miguel de Cervantes, is mere rhetorical praise of history. Menard, on the other hand, writes:*
>
> *...truth, whose mother is history, rival of time, depository of deeds, witness of the past, exemplar and adviser to the present, and the future's counselor.*
>
> *History, the mother of truth!—the idea is staggering. Menard, a contemporary of William James, defines history not as a delving into reality but as the very fount of reality. Historical truth, for Menard, is not "what happened"; it is what we believe happened. The final phrases—exemplar and adviser to the present, and the future's counselor—are brazenly pragmatic.*
>
> *The contrast in styles is equally striking. The archaic style of Menard—who is, in addition, not a native speaker of the language in which he writes—is somewhat affected. Not so the style of his precursor, who employs the Spanish of his time with complete naturalness." — Jorges Luis Borges*

Machine translation solves the problem of ensuring an actual correspondence between source and target languages in backtranslation by having a limited parallel corpus of aligned sentences. This would correspond to having a limited dataset of matched human and artificial representations in similar sensory or conceptual contexts, case in which backtranslation would merely be a way of bootstrapping this into a more comprehensive signal. However, I'm not sure how big of a problem this really is. Coming up with an ingenious translation key to consistently express any artificial representation as a human one while preserving fluency across contexts feels almost equivalent to matching the two conceptual frameworks. I'd be interested in a counterexample of an undesirable translation key which holds up to new situations in terms of associated fluency. Tamsin mentioned a past project on steganographically encoding content in natural language, which might help provide this counterexample.

Can't the fluency measure be tricked?

In the context of representational tethers, adequacy can be measured by simple error metrics like MSE between the original artificial representation and the backtranslated one. Even better, the backtranslation one can be configured to overwrite the original one so that the ML model is incentivized to solve this aspect as an instrumental goal. The same can't be said about the fluency aspect, which is significantly more brittle. Language modeling (more in the computational linguistics sense of assigning probabilities to sequences) is difficult. Sure, many impressive feats in ML involve LLMs nowadays, but the performance is still not ideal, and scaling laws point at roadblocks. Therefore, the fluency metric gauging the likelihood of natural language can be gamed by a sufficiently sophisticated ML model. The situation is way worse for data on neural dynamics, where public datasets barely contain ~30K participants for very limited periods of time at the moment, limiting the performance of large "neural" models. Perhaps elevate the fluency evaluator using the model-in-training itself, like in oversight leagues? There's essentially a need for the True Name of natural language or neural dynamics here, instead of the usual human values.

Could representational tethers be set up between different AIs?

We discussed tethering an ML model's conceptual framework to human ones, but how about tethering it to other artificial ones in a multi-polar scenario? Would that be useful in any way? It might allow their conceptual representations to be maintained in sync during a slow takeoff. This might allow oversight insights to transfer across teams which investigate different ML models each. However, making the internal representations accessible for ML models developed by different parties sounds like a huge coordination problem. That said, in debate-like schemes where multiple AI instances are simultaneously accessible to the same

team, syncing their conceptual frameworks might help them find flaws in each other's reasoning, spot deception, but also collude.

Can't the ML model defeat the mind-readers ?

If humans read its thoughts in natural or in a novel language, or if they experience them more directly, can't the ML model exploit this communication channel and manipulate humans? Indeed, it probably can, though given the "Bottleneck Layer" assumption, all of its thoughts would be captured in the artificial representation, including the fact that it intends to manipulate humans. That might slightly help, but not in the sense that humans would therefore be immune to the ML model's scheming, but rather in the sense that if the information is blatantly accessible, then automated systems might be able to pick it up.

Don't LLMs already translate their internal representations into natural language as their day job?

They do implement an artificial-textual translation, and even a textual-artificial one in their early layers, but (1) the textual heads aren't tied to the same representation, and more importantly (2) the artificial representation isn't constrained to itself be fully expressible in natural language — it can avoid including things in the output due to the lack of internal backtranslation constraints.

What if the gap still can't be bridged?

Perhaps even if the artificial representation can be translated into natural language, the complexity of the translated thoughts would be unwieldy. If the ML model considers ideas in one instance which need multiple textbooks to express in natural language, then human oversight is still not feasible. There's an upper bound on how many humans can be tasked with overseeing the ML model over what period of time. To deal with this, we might penalize intermediate translations beyond a given length, as an extension of the fluency constraint. However, similar to the issues of impact measures, this limits the expressiveness of artificial representations.

However, even if the translated fragment is short, it might be extremely dense. If I were to read a grad-level textbook on biochemistry, I'd find it almost unintelligible, despite it arguably being expressed in natural language. We might essentially share the position of sixteenth-century scribes getting hold of that same contemporary textbook when trying to read through the ML model's thoughts. Similar to the bound on the intermediate translation in terms of length, we might bound it to a specific domain of non-specialized language, so that human overseers wouldn't need advanced knowledge on arbitrary topics. That said, such a bound would similarly be limiting the system to the range of thoughts thinkable by humans. Competitors might be led to disable such regularization techniques while chasing radical innovation in the vein of move 37.

Are representational tethers restricted to prosaic risk scenarios?

The internal representations employed by AIs built on exotic stacks could similarly be conditioned on backtranslation to human representations and engineered to be more cognitively ergonomic. However, the technical details involved would have to change quite a bit. The main architectural constraint is the existence of some bottleneck structure to be tethered, based on the "Bottleneck Layer" assumption.

Outro

Despite not offering a complete solution, representational tethers provide a way of making the ML model easier to interpret, interact with, and control, by systematically conditioning and designing its conceptual framework to be more compatible with that used by humans.

What's more, parts of the proposal can be tested on existing prosaic models as a preliminary experiment in feasibility, despite not having our hands on the real thing.

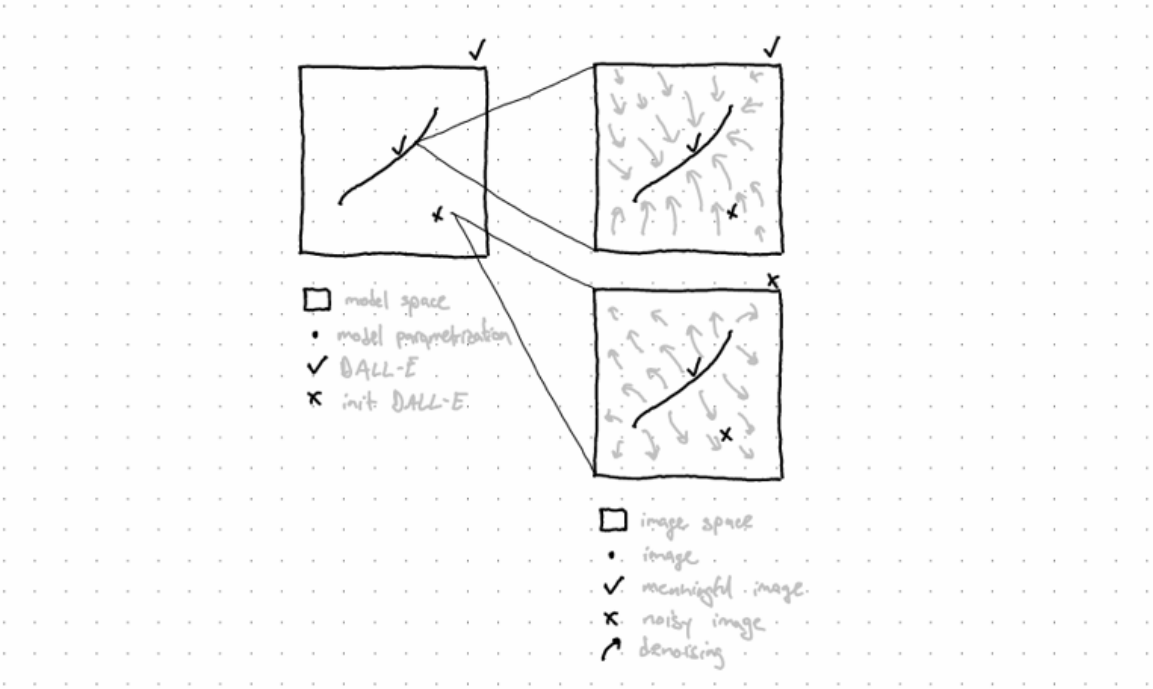
Interlude: But Who Optimizes The Optimizer?

Crossposted from the AI Alignment Forum. May contain more technical jargon than usual.

This post is part of my hypothesis subspace sequence, a living collection of proposals I'm exploring at Refine. Preceded by representational tethers .

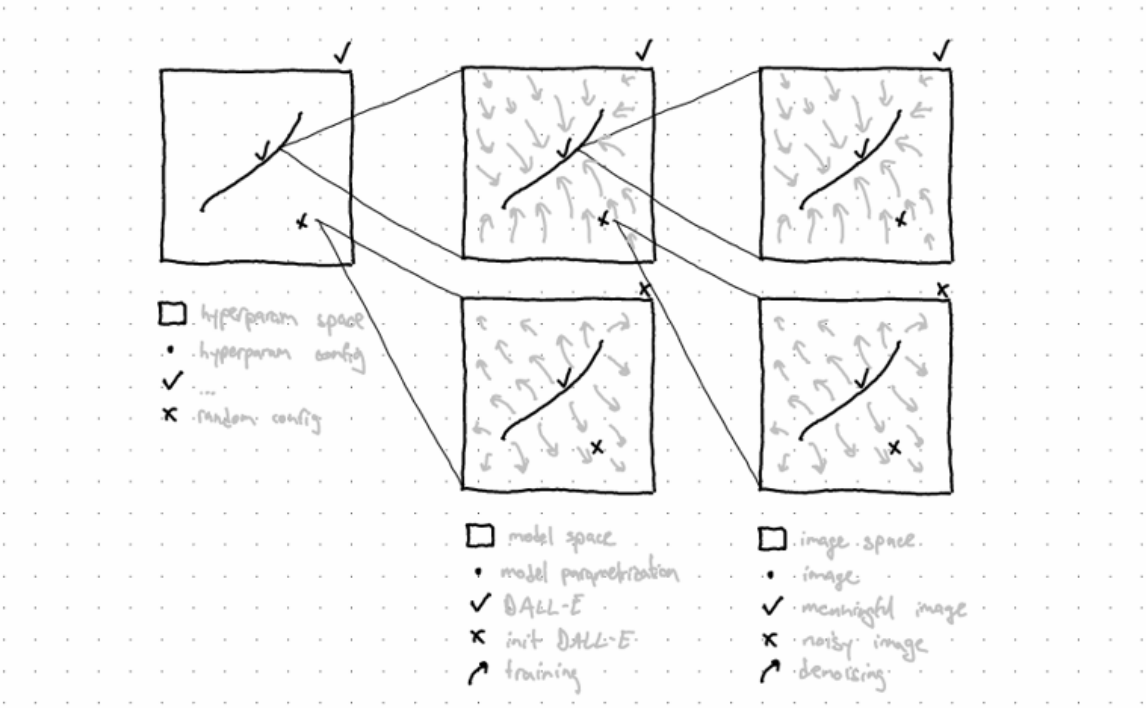
Intro

Consider diffusion models. DALL-E is able to take in an image as input and imperceptibly edit it so that it becomes slightly more meaningful (i.e. closer to the intended content). In this, DALL-E is an optimizer of images. It can steadily nudge images across image space towards what people generally deem meaningful images. But an image diffusion model need not nudge images around in this specific way. Consider an untrained version of DALL-E, whose parameters have just been initialized a moment ago. It still implicitly exerts currents across image space, it's just that those currents don't systematically lead towards the manifold of meaningful images. We deem an image diffusion model good if the field it implements nudges images in a way we deem appropriate. Conversely, a bad one simply specifies a wildly different field.

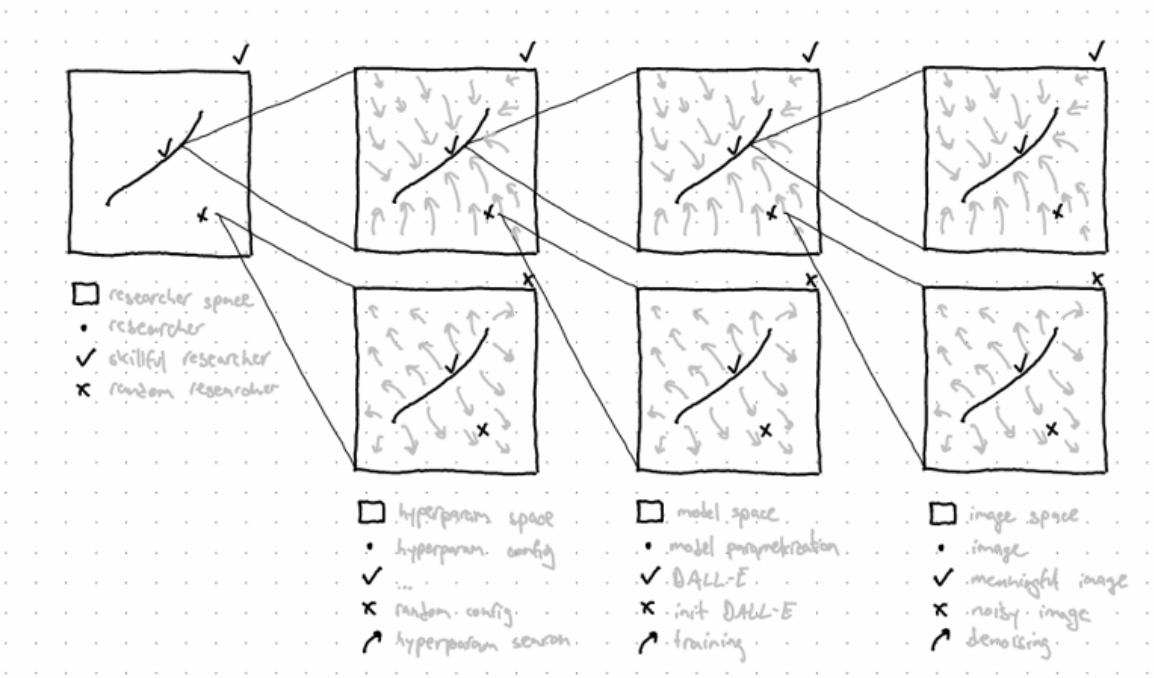


Naturally, we'd like good models. However, we lack an exhaustive closed-form description of the desired field for pushing towards meaningful images. The best we managed to do is come up with a way of roughly telling *which of two models is better*. We then use (1) this imperfect relative signal, and (2) a sprinkle of glorified hill-climbing as means of obtaining a good model by charting our way through model space. We might fill those in with e.g. (1) incremental denoising performance on ImageNet, and (2) SGD. Those two ingredients help us define a trainer for the diffusion model, a way from getting from bad models towards good ones. We deem a trainer good if the field it implements nudges models in a way we deem

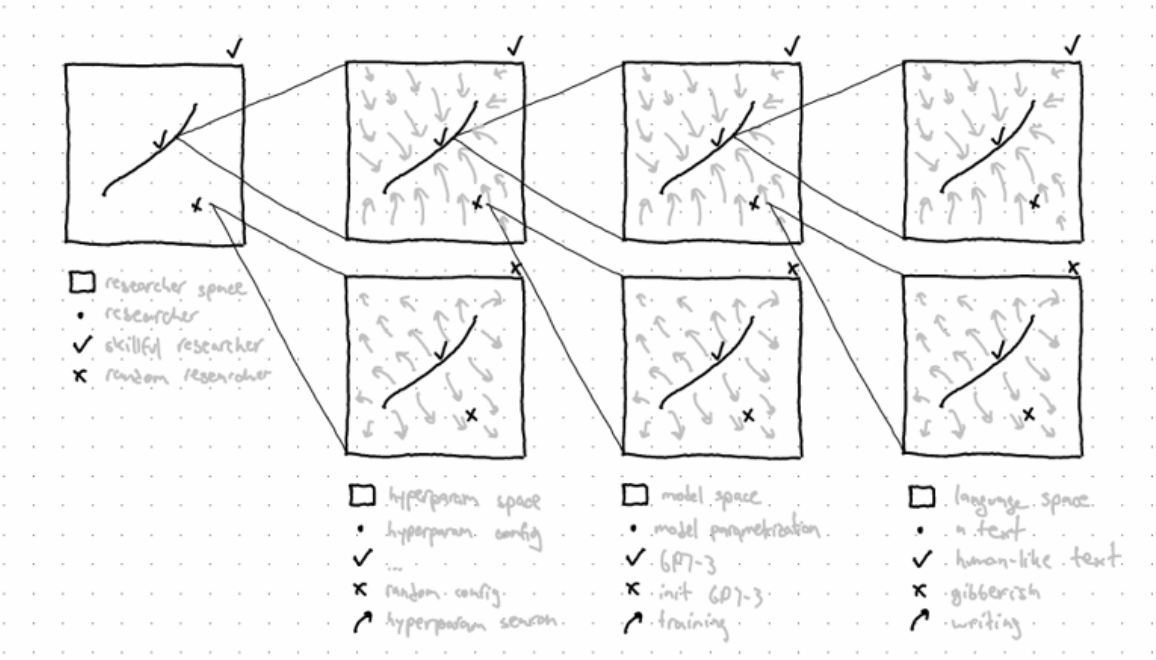
appropriate (i.e. from bad to good). Conversely, a bad one simply specifies a wildly different field across model space.



Naturally, we'd like good trainers. However, we lack an exhaustive closed-form description of the desired field for reliably pushing towards truly good models. You've guessed it, we did manage to come up with a way of roughly telling which of two trainers is better. We similarly use this rough relative signal and our favorite hill-climbing algorithm as means of obtaining a good trainer by charting our way across trainer space. We might fill those in with e.g. (1) convergence speed, and (2) grid search across hyperparameters. Those two help us define a hyperparameter search for the trainer, a way from getting from bad trainers towards good ones. A hyperparam search is good if its field nudges trainers around nicely. Bad ones nudge differently.

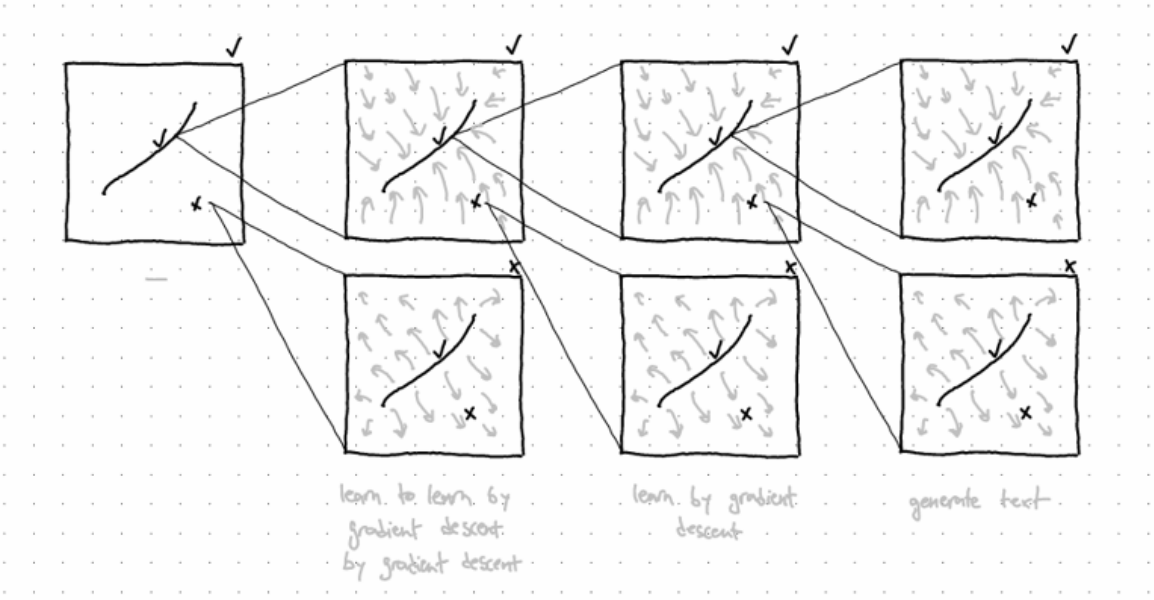


Naturally, we'd like... Ok, you got the point. My next point is that there's nothing special about diffusion. GPT-3 drives text from gibberish towards coherent human-like text. It drives text from one topic to another, from one stage of a narrative to the next. The field it implements is the Janusian physics of language, with the associated transition function defining it. In this framing, GPT-3's trainer is tasked with roughly nudging bad models-in-training towards actually good ones, by means of (1) perplexity on giant text dataset, and (2) probably some advanced SGD.



There's nothing special about current mainstream training regimes as well. You could have GPT-3 at the base nudging text around language space, a trainer helping it learn by gradient

descent, but then instead of conventional hyperparam tuning above the trainer, you could go with learning to learn by gradient descent by gradient descent. And on and on to the cube and beyond. Or instead, you could have a "prompter" as a prompt optimizer trying to generate prompts which get the base GPT-3 to behave in certain ways. And then learn to prompt the prompter in such a way that it's prompts in turn get GPT-3 to behave, etc. A bit like soft prompts.



Clarification: In the examples above, "good models" are nothing but unrealizable Platonic ideals. When trying to move stuff towards them using trainers, we can't (currently) do better than roughly pointing in their general direction using heuristics (i.e. low perplexity on giant text dataset good, duh.). Good models aren't ones which perform optimally on this or that dataset, they're the ideal ones we'd actually want to cause into being. This is intimately tied to models which minimize risk (i.e. what we're actually interested in) as opposed to models which minimize empirical risk (i.e. what we can currently get). Similarly, because good trainers are defined as ones which reliably get us good models, they're also intangible Platonic ideals we only approach by pushing them closer from an optimization level above. As you climb past the top of the technical optimizer stack, you'd reach humans trying to imperfectly nudge the component underneath towards being better, through science and engineering. Humans in turn would be part of a tightly-woven network of optimizers nudging each other towards certain states, but let's limit ourselves to the technical stack of optimizers and its immediate interface with the non-technical.

Taking Stock

I now want to tie this grammar of networked optimizers to proposals discussed in the alignment community. My intuition is that expressing proposals in this language (1) might help surface systematic obstacles faced by whole families of proposals, a bit like reflecting on recurring obstacles when solving mazes, and (2) might enable exploration into entirely new families of proposals, a bit like CHAI's interaction patterns allow you to express your way out of the "here's the goal, go optimize" paradigm. A similar instance of recognizing the appeal of generativity:

> "*We find that these disparate techniques now come together in a unified grammar, fulfilling complementary roles in the resulting interfaces. Moreover, this grammar allows*

> *us to systematically explore the space of interpretability interfaces, enabling us to evaluate whether they meet particular goals.*" — *Arvind Satyanarayan and Chris Olah*

Explicit Goal

We might come up with an appropriate explicit goal for the AI, something like "move the world towards states which humans would actually like." That said, experience tells us that our ability to intelligent-design a GOFAI-style system from scratch is extremely limited. This makes it likely that we'll resort to designing a trainer which in turn optimizes the model towards implementing the field over world states that we'd actually like. The appropriate explicit goal (i.e. True Name) would probably only be employed by said trainer to compare model variants and hill-climb its way towards good models. In this, the presence of an appropriate explicit goal first makes for a good *trainer*, before getting us a good model. The good trainer is then capable of nudging models towards good ones. Generally, it is us humans who are assumed to craft (read: optimize) the trainer so that the risk it minimizes is virtually identical to true risk. A failure to bake in the True Name would simply reflect an error made by human researchers in their process of optimizing the trainer, which would then optimize the model, which would then optimize the world. In this "everything is a field!" frame, setting the right goals is doing meta-optimization right.

However, as fascinating as humans are, they are only so good at optimizing the trainer. Sure, we did find True Names before for other things. Sure, some humans are better at this than others. That said, there's simply a possibility that we fall short of being able to craft the ideal trainer, for whatever reason. This is the core flaw of this family of proposals — humanity might be suboptimal in terms of trainer optimization skills. More time (i.e. more opportunity for nudging), more people involved (i.e. more humans contributing fields across trainer space) seem to help.

> "*The antimeme is that most people shouldn't be thinking like consequentialists. Instead of thinking about how to maximize utility, they should be thinking about how to maximize dignity. This is easier. This is computationally tractable.* ***This heuristic will make you do better.***" — *Akash's Connor-model*

Implicit Goal

So if we employed a trainer when not being able to directly craft a good model ourselves... and we're unsure whether we can actually craft a good trainer ourselves... let's just craft something which itself crafts the trainer! Humans would define a trainer trainer at the same level of conventional hyperparam tuning, but more focused on relative mesa-level performance than hill-climbing technicalities. Or in other words, simply a better trainer trainer given our highly-contextual notion of what makes a good trainer.

> "*It is much easier to build a controller that keeps you pointed in the general direction, than to build a plan that will get you there perfectly without any adaptation.*" — Rohin Shah

This appears to mostly be the family of proposals I've focused on so far in the sequence:

- oversight leagues: Optimize the trainer through unilateral learning signals (e.g. contrastive dreaming, adversarial training), and bilateral ones (e.g. past, present, and future exploitation). The trainer's optimization here is heavily focused on adjusting the way it compares models downstream. Cross fingers for this trainer trainer to be appropriate.
- ideological inference engines: Optimize the trainer by using the inference algorithm to expand the knowledge base. Similar to the above, the trainer's optimization is focused on tweaking the way it compares models downstream. Similar to the above, cross

fingers (i.e. lament about the fact that the trainer trainer is itself "just" the result of human optimization).
- representational tethers: Optimize the trainer by gradually bringing its conceptual framework closer to human ones and extracting the concept of what humans want. Similar to the above, it's the model ranking that's at stake, and the proposal itself merely a specific state in trainer trainer space, merely an outcome of human research (read: optimization).

The advantage of this family of proposals in contrast to the previous one is that the trainer^2 might eventually spit out a trainer which is better than what we could've achieved through "direct work" on the trainer, similar to how no one could've possibly created GPT-3 by handpicking parameters. It might spit out a trainer which is better at nudging models towards implementing the nudges we actually want on the world. The flipside is that, of course, the trainer trainer is still designed through direct human work. The problem is pushed one step upwards, leading to pertinent comments like Charlie's.

Just like the previous family, buffing up the members of the present one means doing research (i.e. exerting human optimization) on trainer^2 specifics, trying to get closer to the True Name of that which finds the True Name of what we want to turn the world into. An important focus of research into this family appears to be the limited meta-optimization of humans informing the trainer trainers. Additionally, exploring self-consistency, self-sufficiency, self-organization, etc. at this specific level without going full Hofstadter also appears valuable. To address specific proposals:

- oversight leagues: The trainer's dataset is constantly expanded with negative examples (what else can you get without humans?) as a result of the trainer^2 doing its job. How could humans be prompted to complement the growing negative pool with effective positive examples?
- ideological inference engines: How could you identify the most brittle and sensitive parts of the knowledge base so that you could ask humans to weigh in effectively? Can you check that the expanding knowledge base is in a basin of attraction so that you don't need particularly accurate human nudges going forward?
- representational tethers: The human representation which the model's latents are being tethered to appears impossible to exhaustively specify (i.e. the fluency constraint is brittle). How do you effectively get more language or neural data to model the underlying distribution better?

For the most part, the suggested research directions are attempts to force members of this family into the next one, due to its promise of positive feedback loops optimizing the human's optimization skills. However, before moving on, I'll note that *good-enough* trainers or trainer trainers might be pretty nice already. Even if we don't manage to precisely point at what humans actually want, we might manage to point at something which is okay-I-guess-kinda. For instance, there might be something to being able to plug in a hands-off high-integrity deontology using ideological inference engines. In this, investing effort in patching up the above seems somewhat worthwhile.

Human Amplification

Right, so humans as optimizers of downstream components (e.g. trainer, trainer^2) can only be so good at their task. What if we optimized ourselves in turn? This is how I currently perceive human amplification schemes in their various flavors. We're (indirectly) optimizing a system. What if that system could help us in turn become a better version of ourselves™ so that we could better optimize it, ad nauseam?

This perspective feels particularly appropriate for compressing proposals like "Let's augment alignment researchers so that they do alignment better" and "Let's augment human

overseers to keep tabs on the AI better." The same perspective feels slightly more awkward when discussing CHAI-style interaction patterns: "Have the AI active-learn the goal by observing humans or ask questions" could be framed as "Have the AI effectively turn humans into better optimizers of AIs by staying out of their blindspots and rendering their budget of guidance bits more informative." Help me help you, say both.

The blessing of human amplification comes together with the curse of human modification. Outside the context of contemporary alignment, they might mean similar things. Here and now, human amplification reads as "make human better optimizers of AI systems," while human modification reads as "tamper with the human in a way their previous selves wouldn't endorse." They're duals. However, human modification here wouldn't lead to humans being able to indirectly create better AI, it would lead to a self-driven AI being able to create better humans from *its* perspective, in a twisted way. Humans which would reliably reward it, stay out of its way, stop existing, etc.

> "*It's not just a matter of the Lambertians out-explaining us. The whole idea of a creator tears itself apart. A universe with conscious beings either finds itself in the dust . . . or it doesn't. It either makes sense of itself on its own terms, as a self-contained whole . . . or not at all. There never can, and never will be, Gods.*" — Greg Egan

Miscellania

This last family isn't really composed of proposals for how to solve alignment, but rather proposals for what failure modes might be important to avoid. Let's start with gradient hacking as what I'd describe as "prosaic RSI" where the model finds itself in the dust (i.e. gains situational awareness) before exerting nudges on itself. Essentially, gradient hacking would go as follows: model first becomes aware of the field implemented by its meta-optimizer (e.g. trainer), it then tries to timidly move towards places in model space where the current would then lead it where it wants. That said, gradient hacking is really a bad idea which you shouldn't really consider enacting if you're reading this, humans would find it really not cool.

I find gradient hacking intimately related to the MAML paradigm, whose point is to nudge a model towards a point from where subsequent task-specific nudges would quickly get it into good regions. Move the model towards where the current would lead it towards where you want it to be, says MAML. Move yourself towards where the training current would lead you towards where you want to be, says gradient hacking. Of course, MAML is engineered to behave in this way. Gradient hacking isn't a planned feature.

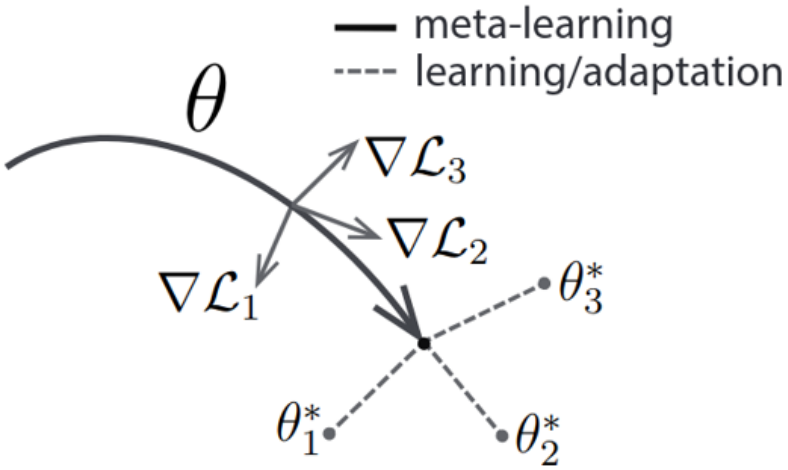


Illustration of MAML. Model parametrization θ is optimized to be effectively optimized via subsequent nudges (Source).

Another phenomena which fits the present framing is mesa-optimization. All the nodes in the optimizer stacks are meta/mesa-optimizers of their neighbors, but mesa-optimization in isolation refers to the situation in which the model being trained — besides the optimization it's doing as its day job (e.g. of images, texts, worlds) — somehow gains a mesa-optimizer. This might happen due to certain features of the task. Quite familiar at this point, the danger is partly framed as the model reconfiguring itself into a more general state which could trigger its own effective specialization on various tasks. Evolution causing intelligence is often offered as an informal existence proof in the literature, though I prefer evolution causing internal selection:

> "*Another molecular biologist, Barry Hall, published results which not only confirmed Cairns's claims but laid on the table startling additional evidence of direct mutation in nature. Hall found that his cultures of E. coli would produce needed mutations at a rate about 100 million times greater than would be statistically expected if they came by chance. Furthermore, when he dissected the genes of these mutated bacteria by sequencing them, he found mutations in no areas other than the one where there was selection pressure. This means that the successful bugs did not desperately throw off all kinds of mutations to find the one that works; they pinpointed the one alteration that fit the bill. Hall found some directed variations so complex they required the mutation of two genes simultaneously. He called that "the improbable stacked on top of the highly unlikely." These kinds of miraculous change are not the kosher fare of serial random accumulation that natural selection is supposed to run on.*" — Kevin Kelly

It turned out the phenomena above was mostly caused by the genome itself having evolved to evolve in specific ways when pressured by the environment. It seems that it was selected to differentially enable mutations at different locations based on the nature of selective pressure exerted by the environment. Fascinating. For a more flowery account:

> "*Paolo absorbed that, with the library's help. Like Earth life, the carpets seemed to have evolved a combination of robustness and flexibility which would have maximized their power to take advantage of natural selection. Thousands of different autocatalytic chemical networks must have arisen soon after the formation of Orpheus, but as the ocean chemistry and the climate changed in the Vegan system's early traumatic millennia, the ability to respond to selection pressure had itself been selected for, and the carpets were the result. Their complexity seemed redundant, now, after a hundred million years of relative stability – and no predators or competition in sight – but the legacy remained.*" — Greg Egan

Visual summary of the above.

Outro

A host of prosaic phenomena can be expressed in the grammar of stacked optimizers, enabling an organization of alignment proposals in a way which highlights systematic pros and cons to focus on next. Are there also other families out there at this level of abstraction? What if we considered more intricate graphs instead of mostly-linear stacks of optimizers? What optimization level was this all on even?

(Structural) Stability of Coupled Optimizers

Crossposted from the AI Alignment Forum. May contain more technical jargon than usual.

This post is part of my hypothesis subspace sequence, a living collection of proposals I'm exploring at Refine. Preceded by an interlude .

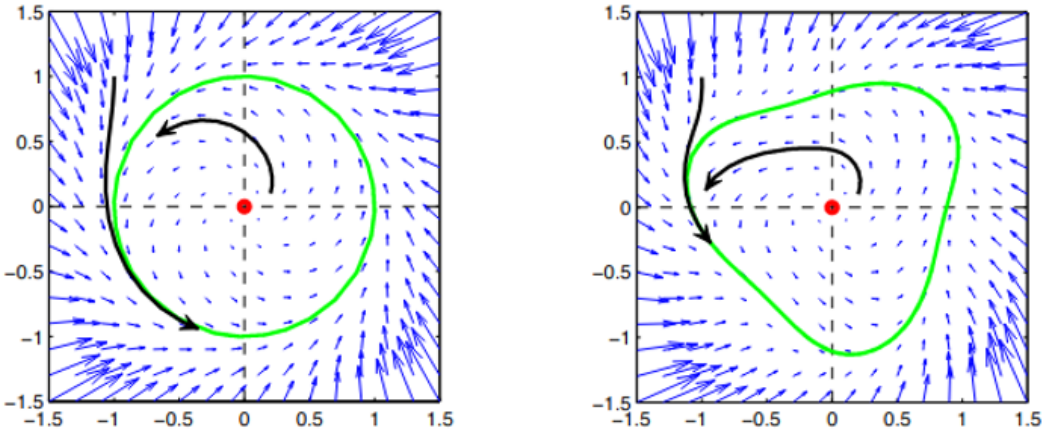
Thanks Adam Shimi, @artaxerxes, Herbert Jaeger, Tamsin Leake, and Ze Shen for discussion informing this post.

Intro

In the previous post, I explored a grammar of alignment proposals based on (what I later learned could be referred to as) *coupled dynamical systems*. To help you get a sense of this view, one family of proposals tackles the alignment problem by having human researchers optimize a trainer^2 (e.g. oversight leagues) which optimizes a trainer (e.g. SGD across model space employing empirical risk as heuristic) which optimizes a model (e.g. LLM) which optimizes a world (e.g. text). Other families have different numbers of optimization levels (e.g. True Name partially obviates the need for trainer^2), integrate feedback loops (e.g. human amplification), or extend the optimization stack in some other way. In this, the focus moves from what to point an AI at, to how to move towards the associated trainer state (e.g. using heuristics across trainer space and above, similar to empirical risk as heuristic across model space). A shift from targets to *processes* for reaching said targets.

From this level of abstraction, one can identify issues which systematically plague entire families of proposals. For instance, there's still a bounded human at the top guiding downstream optimization (e.g. on trainer^2). In the current post, I aim to explore how two notions from dynamical systems apply at different levels of optimization, and how those might help alleviate systematic issues. The two notions are:

- **Stability**. A region of the state space of a dynamical system is deemed stable if slight perturbations to objects in that region don't result in the object permanently leaving the region. For instance, a tiny asteroid which crashes into Earth won't cause Earth to leave its orbit around the Sun, because Earth's orbit is stable. In contrast, the smallest gust of wind blowing on a pencil balanced on its tip will cause it to leave that region of state space behind, as it's not stable. **Note**: While stability is mathematically defined only for such regions of state space, I'll stretch the term "stable" for the purpose of this post to also describe the dynamical system which in turn contains reachable stable solutions. This reading of stability can also be described as "the region of plausible initial conditions is contained within the basin of attraction of a single attractor." I purposefully avoided the related reading of "sensitive to initial conditions" due to its chaotic connotations.
- **Structural Stability**. While the previous property touches mostly on perturbations of objects inhabiting a state space, *structural* stability touches on perturbations of the dynamical system itself. For instance, the map "implemented" by the laws of physics across world states would be deemed structurally stable if slight changes in the values of fundamental physical constants wouldn't radically change the states towards which the world converges. You might still get planets orbiting around stars making up galaxies. In contrast, if the laws of physics across world states would turn out not to be structurally stable, the slightest change to the fundamental physical constants would yield radically unfamiliar universes.



Left: a structurally stable dynamical system containing a stable orbit. **Right**: a slightly perturbed version of the original, which still contains exactly one cyclic attractor and one point repeller (Source).

Instances

1. Model

Let's explore the notions of stability and structural stability in the context of different levels of the optimizer stack. First, if a model is stable (according to the stretched definition), then the particular state the world is initially in doesn't have much of an influence on the state of the world the world will get in through its optimization. For instance, the fine-tuned version of VPT can spawn in a random Minecraft world and bring it in a state in which it possesses a diamond pickaxe. The attractor could be seen as a high-dimensional manifold of worlds-where-I-have-a-diamond-pickaxe. Alternatively, a prompted DALL-E can start with any noisy image and reliably optimize it towards being meaningful. Sure, there is *some* path dependency, in that DALL-E strikes the manifold from different angles and yields slight variations, but overall it's pretty stable.

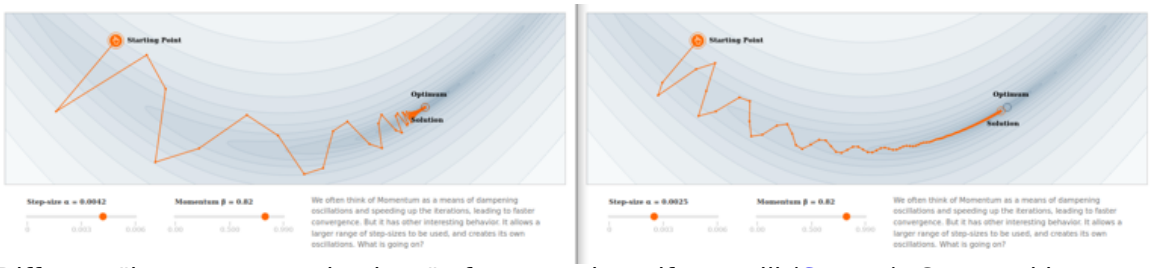
DALL-E variations (Source).

Now, what about structural stability at the model level? Do slight perturbations to the model parametrization itself lead to qualitatively different behavior? Dropout and DropConnect are two instances of such perturbations, where a tiny proportion of either nodes or edges in the scalar version of the computational graph of an ML model get mutated. In practice, while Dropout and DropConnect do reduce performance slightly (during training, as they're usually disabled during inference), it's often not a big hit, especially as the model's trainer gradually moves it towards parametrizations which are structurally stable (e.g. states from which a 0.3-Dropout sway across model space doesn't hurt empirical risk much).

II. Trainer

Let's now go a step above and explore the same two notions in the context of the trainer. What does it mean for a trainer to be stable? Well, a stable trainer should be able to take in model parametrizations scattered across model space and reliably move them towards high-performance states. Given the fact that virtually all training procedures start with a random initialization of model parameters, most trainers can be described as quite stable, as they can reliably take in random parametrizations and optimize them successfully. Even when a model has internalized very particular dynamics during a previous training stage, subsequent training on a different task can unfortunately make away with all that complexity via catastrophic forgetting and efficiently repurpose representational resources for solving the present task.

Now, what about the structural stability of trainers? Do slight variations in the hyperparameters which define them (e.g. learning rate, dataset informing hill-climbing heuristics) lead to qualitatively different behavior downstream? An interesting talking point here is momentum, which seems to significantly broaden the range of learning rates which enable good trainer performance. In a sense, this feature crafted into the trainer from a level above by the human researcher improves its structural stability. It doesn't matter that much what learning rate settings you plug in then, as the trainer still behaves in a similar way.



Different "hyperparametrizations" of a toy trainer, if you will (Source). Captured in a way meant to visually resemble the earlier pair of structurally stable dynamical systems.

III. Trainer^2

Let's now explore the two notions in the context of the trainer^2. If an oversight league as a trainer^2 was stable, then the particular positive examples specified by humans might not be that impactful on the resulting trainer. If an ideological inference engine as a trainer^2 was stable, then the small print of the seed knowledge base specified by humans might not be that impactful on the resulting trainer being crafted. If representational tethers were to be stable, then the initial conceptual framework employed by the ML model would be washed out. If CIRL were to, the agent's initial estimate of the human objective would be transient. If PreDCA were to, the system's initial candidates for precursors and their objectives wouldn't influence the final solutions much.

Right, and how about structural stability? Do tiny changes in those proposals lead to qualitatively different effects across the optimizer stack? Here I lack existing instances of such variations, but again the presence of structural stability would make the exact choice of trainer^2 implementation details relatively inconsequential.

IV. Human

How about the human researchers optimizing the trainer^2? A stable researcher (in the same sense as above) would be able to start with a largely arbitrary version of a trainer^2 before being able to successfully move it towards a state which exhibits good downstream performance. In contrast, an unstable researcher (again try to suspend connotations) would be driven more by path-dependence in their search across trainer^2 space, rather than leaving the idiosyncratic initial conditions behind and converging towards suitable trainer trainers.

Predictably, structural stability of a researcher would describe the limited influence of perturbations to their method of downstream optimization. Slightly different versions of them might end up optimizing the trainer^2 in the same way. Methodological therapy might need to be particularly strong a perturbation in order to have an effect in the case in which researchers are structurally stable. However, this need not be the case, as researchers might themselves cross bifurcation points and radically change their approach to optimization.

(Structural) Stability & Alignment

While it's certainly satisfying to find such patterns — especially when recursively nested as in the case of optimizer stacks — where does that lead us in terms of making progress on the alignment problem?

Measuring Stability

We might be interested in developing trainer trainers which consistently nudge trainers towards the same kinds of trainers, similar to how DALL-E nudges images towards the same kinds of images. However, a trainer^2 which simply nudges trainers towards the same outcomes isn't equivalent to a trainer^2 which nudges trainers towards *good* trainers in particular. That said, I think good trainer trainers (i.e. those which can find trainers which produce aligned models) are more likely to lead to consistent kinds of trainers than bad trainer trainers, similar to how a simplicity heuristic might be handy in crossing things off. This intuition is based on the observation that human values appear relatively stable across time and space (though cultural and trainer^2 fields might not necessarily be close). Western

and Eastern cultures are unlikely to casually swap ideologies from one decade to the next. A slight memetic sway across the US is unlikely to prevent it from coming back to its mainstream ideals soon after. In contrast, if invoking the Anna Karenina principle, disturbed versions of human values are unlikely to form large unified basins of attraction, because they'd be fundamentally incompatible with each other. In other words, the stability of a region in trainer space might slightly conditionally boost the probability that said region leads to aligned models.

Given this, how could we measure the stability of a trainer^2? In order to see whether different initial states consistently get nudged towards a tightly-knit region of trainer space, we could... simply try running a trainer^2 on different initial trainers and measure the spread of the result. Bootstrapping (stats sense) the dataset used for informing hill-climbing heuristics across model space might be an easy way to achieve that sampling of trainers for certain proposals. For instance, one could try to sample-with-replacement the initial positive examples used to kickstart an oversight league, and get different evaluators at the end, where each evaluator could be incorporated into unique trainers as a heuristic for navigating model space. Alternatively, one could try to "wiggle" the initial trainer across specific dimensions and sample initial conditions this way. For instance, one could try to systematically paraphrase parts of the seed knowledge base in ideological knowledge engines and see whether the inference algorithm takes the variants towards wildly different places.

The "start with different inits and see how things go collectively" can be visualized as ensembles across state space being swayed around in unison as deformed areas, as opposed to single individual points moving around:

Ensemble of systems with different initial conditions evolving in tandem across state space (Source). The approaching of uniform density appears typical of chaotic systems, but the same visualization technique is useful as an intuition pump for non-chaotic ones, too. See also text above.

But say we tried running a trainer^2 forward starting with different initial trainers. We got a set of optimized trainers. How do we actually gauge whether they populate the same region in trainer space? For instance, it is non-trivial to infer whether the different variants of the "Girl with a Pearl Earring" painting generated by DALL-E share family resemblence. In terms of pixel-wise distances, the images are quite different, so what brings them together? We might consider the two trainers similar if they bring about structurally stable models, and in doing so tackle challenges of measuring one kind of stability with another. Fortunately, the problem is defined somewhat better for models. For instance, the KL-divergence across output distributions of two different models in a shared context is often used as a proxy for the distance between the two dynamics being implemented. It's decently successful in keeping a model being fine-tuned on a meaningful "leash" from its original location, so that it doesn't deviate too much, as seen in RLHF.

Unfortunately, KL-divergence is naively used as a measure of difference between two fields across world space *at a specific location*. "Starting in this given world state, how do those two models take it from here?" Sampling contexts across the entirety of world space would be intractable as a way of establishing the similarity of two models. There could always be hidden edge cases where the two radically disagree. Here's a glimmer of how one mighy efficiently measure how much two ML-implemented fields agree at every location across world state at once, maybe. First, compose the two models into a larger computational graph, where the same input feeds into both, and their outputs feed into a difference operator. Second, optimize the input, rather than the model, so that it elicits a high difference, similar to DeepDream or contrastive dreaming. Third, try to analogize the provable criteria used in robust ML in order to get definitive bounds on the difference between models at any given point *without sampling.* Even crazier, it might be possible to

also obtain bounds on field differences at levels above the model, assuming differentiability. For instance, the future exploitation of oversight leagues hints at being able to differentiate through trainer training itself. Ditto for learning to learn by gradient descent by gradient descent (G.pt anyone?). Those are currently very rough thoughts.

There's also an anthropic angle on this whole gauging of stability thing. If we can reasonably bootstrap (stats sense again) our way towards the distribution of initial conditions *we might have started optimization with* and we analyze the nature of possible outcomes resulting from those, how should that influence our particular choice of seed in high-stakes runs? What if we stick to the majority of seeds approaching a shared region together? If that stable outcome actually turns out to be inappropriate, whole timelines hopping on the stability train would go south, for instance.

Improving Stability

Propagating an ensemble of trainers through the ebbs and flows exerted by the trainer^2 might give us a sense of how reliably it nudges towards a consistent region. This might allow for the ranking of different hand-crafted designs of trainer trainers, assuming that's where the human-artificial interface lies in the optimization stack. However, besides being able to compare trainer trainers with different architectures, measuring stability might also enable the systematic improvement of a given trainer^2 in that sense.

If stability is defined here as the containment of plausible initial conditions in a basin of attraction across trainer space, then selecting the initial conditions themselves might provide a relevant degree of freedom. Concretely, if perturbations of initial conditions along certain dimensions are more likely to interfere with the convergence of solutions than others, then it might make sense to focus human optimization on those aspects of the seed to which the trainer^2 is particularly sensitive to. For instance, if paraphrased variations of a certain dictum included in the seed knowledge base of an ideological inference engine cause important variations in the expanded versions, then it might be useful to focus human optimization on specifying it well, rather than some less brittle ones. Similarly, if the exclusion of certain kinds of samples in the dataset used to kickstart an oversight league cause a high spread of outcomes, it might perhaps be useful to focus on documenting it more extensively. In other words, conducting a sensitivity analysis on the trainer^2 might enable active-learning on its part, where human feedback could be requested to optimally reduce uncertainty in the nature of the trainers being approached through trainer training.

However, there might also be issues with selecting for trainer trainers which consistently get us a certain type of trainers. A stable region of trainers obtained through trainer training might be tied to value lock-in in the models yielded downstream, a situation in which human meta-values would be compatible with object-level ones shifting around, but in which the trainer trainer wouldn't succeed in enabling that shift. Yet another issue is that the very extension of the optimizer stack upwards comes with an inevitable trade-off between unfortunate human-model distancing and fortunate aiding of base-level implementation (similar to hand-crafting GPT-3 versus employing a trainer to navigate model space using heuristics). The framing shifts the problem from localization of aligned regions across model space to means of gradually reaching said regions, in the hope that specifying guidance (e.g. vector fields across spaces) might be more feasible than specifying targets. In this context, the two notions of stability are meant to help select guidance systems (i.e. find trainer trainers). However, we might also stumble across dangerous territory in the process of reaching desired targets.

Outro

Organizing proposals through the lens of coupled optimizers surfaces systematic issues. For instance, there's still a bounded human at the top guiding downstream optimization (e.g. on trainer^2). Exploring how classic notions from coupled dynamical systems translate to alignment might help uncover systematic solutions through (1) provable guarantees of stability (e.g. getting bounds on stability of trainer^2), and (2) active-learning procedures making optimal use of human optimization in reducing model uncertainty.

Boolean Primitives for Coupled Optimizers

Or, Why Conceptors Are So Damn Cool.

This post is part of my hypothesis subspace sequence, a living collection of proposals I'm exploring at Refine. Preceded by an exploration of (structural) similarity in the same context of coupled optimizers .

Thanks Alexander Oldenziel, Herbert Jaeger, and Paul Colognese for discussions which inspired this post.

Intro

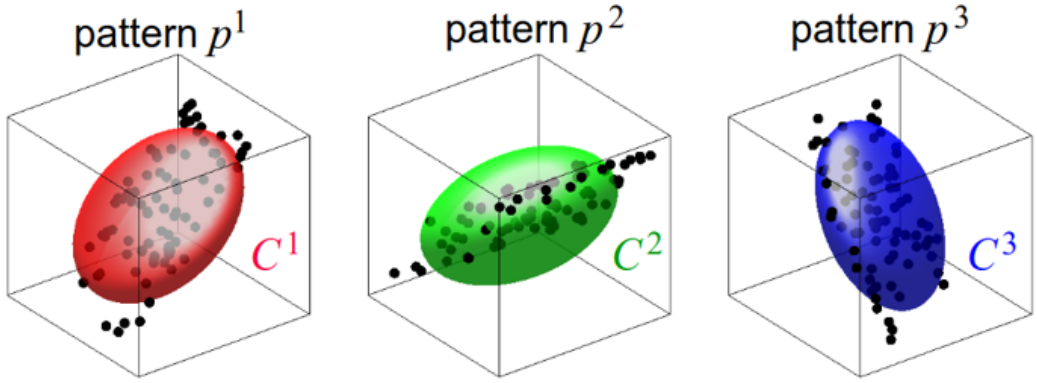
Conceptors, as introduced by Herbert Jaeger in 2014, are an extremely underrated formalism at the intersection of linear algebra, Boolean logic, and dynamical systems, with a host of direct applications in prosaic alignment. For instance, conceptors can help (1) identify ontologies from latent activations, (2) steer sequence models by whitelisting or blacklisting dynamics, (3) shield internal representations under distribution shift, and many others. Their lineage can be traced all the way back to Hopfield networks, and they remain more mathematically-grounded than many ML-adjacent formalisms.

This post is mostly an overview of conceptors, the primitives they expose, and their low-hanging fruit applications in prosaic alignment. Accordingly, most of the ideas presented here can be found in the linked papers, one of which I was excited to co-author. Besides offering a walkthrough of those existing ideas, I'll also explore how they might apply at different levels of the optimization stack, similar in style to the previous exploration of stability and structural stability.

Conceptors

This section will only provide a high-level intuitive understanding of conceptors. If instead you're hungry for precise definitions, please refer to this section or this page, though keep in mind that they have originally been formulated in the specific context of RNNs, and the terminology might be slightly confusing.

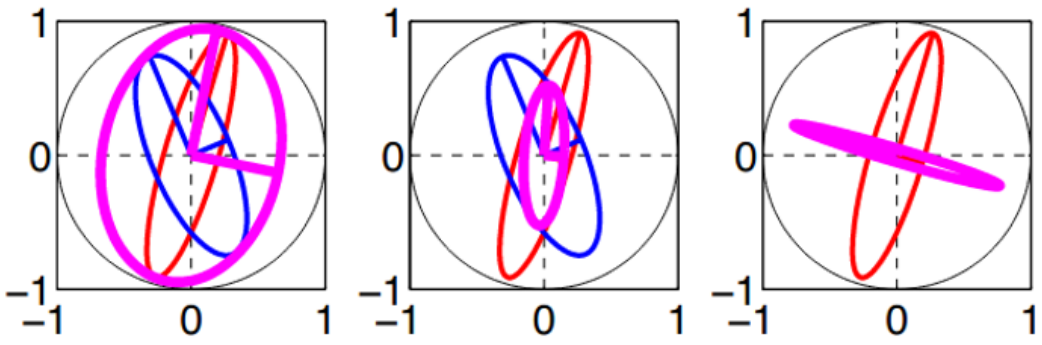
Now for the hand-waving. A conceptor is a mathematical object which captures the spread of a state cloud across dimensions of the space it populates. A state cloud, in turn, is simply a set of high-dimensional vectors. Much like the outputs of PCA, conceptors can be represented as high-dimensional ellipsoids specifying a coordinate system of their own, distinct from the coordinate system of the original space. In contrast to vanilla PCA, conceptors have an additional "aperture" parameter which enables much of the neat mathematical constructions which follow. This parameter controls how strongly a conceptor adapts to the structure of the underlying state cloud.



An illustration of conceptors capturing the spread of three different state clouds (Source).

An equivalent machine-friendly representation of a conceptor is of matrix form, essentially consisting in the stacked vectors which define the new reference frame. Not unlike the correlation matrix used directly in PCA, conceptors can be obtained from underlying state clouds in closed-form and in sublinear time with respect to the cardinality of the state cloud. Typically, the state cloud is composed of model states (i.e. latent activations) originating from an ML model, such as an RNN or transformer. Therefore, conceptors are often — but not always — used across activation space.

Here's where things really stop looking like rebranded PCA, though things might translate back to PCA. The disjunction (OR) of two conceptors is defined as the conceptor obtained through the union of the two associated state clouds (see left graph below). The negation of a conceptor is defined as the conceptor which spreads in the opposite manner across each dimension. For instance, if the original conceptor represents high spread across dimension 1, but low spread across dimension 2, the negated version will capture the reverse pattern (see right graph below). The conjunction (AND) of two conceptors is defined in a slightly round-about way using De Morgan's law, a theorem which frames conjunction in terms of disjunction and negation (see middle graph below). The logical difference then yields the contrast in spread between two conceptors, and can be obtained using conjunction and negation. A rigorous 7-page definition of the Boolean ops described above can be found here. Additionally, a ~20-page grounding of the link between dynamics captured by conceptors and formal logic can be found here.



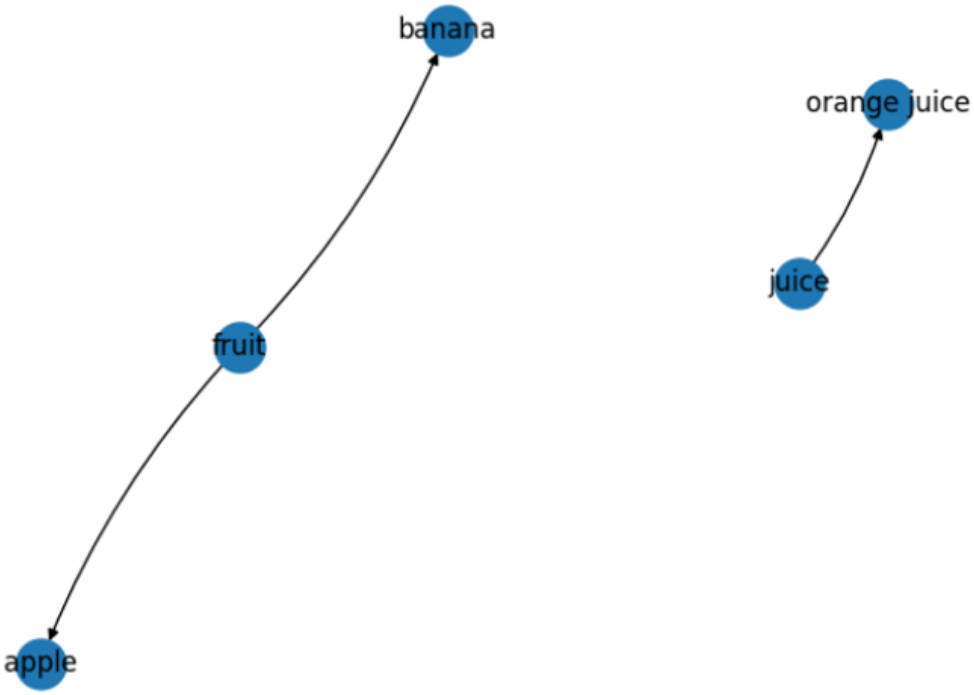
Applying OR, AND, NOT to two-dimensional conceptors (Source).

Model-Level Applications

Extract ontologies from latent activations

If you collect state clouds of latent activations arising in an LLM when processing different noun phrases (e.g. "apple", "orange juice") and learn one conceptor for each such symbol, then you can use their geometry to infer relations of abstraction. For instance, the "fruit" conceptor appears to spatially engulf the "apple" one, as the former contains the meanings of the latter, but some additional ones as well. You'd need to take the disjunction of "apple" with some other instances of fruit in order to approach the very concept of fruit. By working with state clouds of raw embeddings, the method has the benefit of being modality-agnostic (i.e. it should work with Vision Transformer or Decision Transformer as it does with a language model, with minimal modifications). In contrast, interpretability techniques based on witty prompt engineering attempt to elicit entities and their relationships as text output, therefore being text-only.

Reference: Nested State Clouds: Distilling Knowledge Graphs from Contextual Embeddings



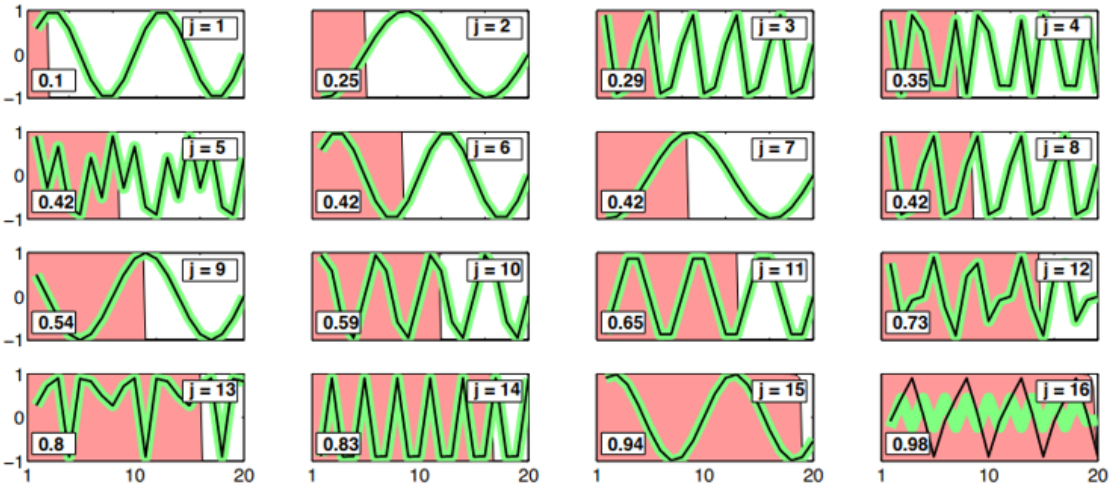
Ontology extracted from BERT embeddings using the Nested State Clouds technique (Source).

Assess representational resource allocation

If you incrementally train a model to internalize different dynamics (e.g. language modeling on different datasets perhaps), then you can roughly measure the "memory usage" of the model at any given point using the logical difference of the disjunction of "learned stuff" conceptors and its negation. When overparametrized and freshly initialized, each new

dynamic being internalized will incur costs of representational resources. However, when new dynamics build on previous ones, then the "memory usage" doesn't increase as much, as representations are being recycled. This might be used to detect whether a specific dynamic has been internalized by a model, and as a way to gauge the representational budget.

Reference: Controlling Recurrent Neural Networks by Conceptors / Neural memory management

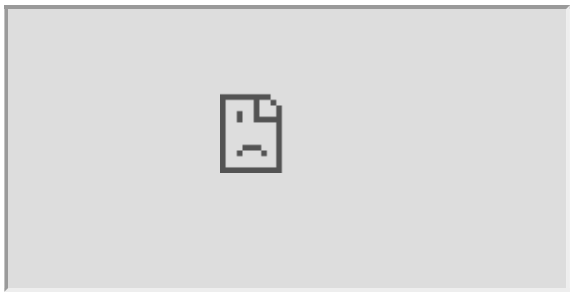


Simple time signals being incrementally stored in an RNN. The 16 patterns are being loaded in order, with j the index. **Notice how learning episodes 6 through 8 incur no memory cost (visualized via pink coverage), because the patterns have already been stored** (Source).

Isolate and either amplify or suppress dynamics

If you elicit a certain dynamic from a model in different situations (e.g. deception, mesa-optimization), learn conceptors from the associated model states, take the conjunction of those conceptors, negate it, and finally insert it into the recurrent/autoregressive update loop, then you might be able to suppress the specific dynamic. This basically enables blacklisting of specific problematic dynamics. Drop the negation and go for disjunction if you want to switch to whitelisting.

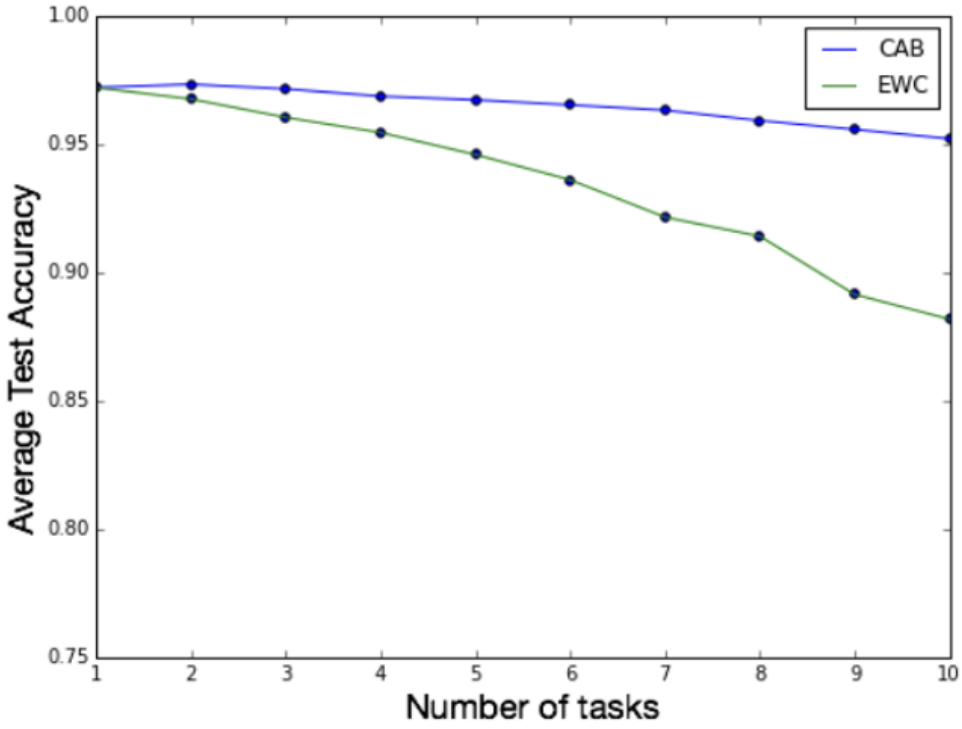
Reference: Human motion patterns learnt with conceptor-controlled neural network (an instance of inserting a conceptor in the update loop)



Shield internal representations under ~distribution shift

If you train a model to internalize certain dynamics and compute the associated conceptor based on said dynamics, you can then tweak the backpropagation algorithm in order that the previous representations are generally shielded from future updates, by forcing learning to make use of different representational resources. Code knowledge across other dimensions as much as possible, and don't tamper with those specific old latents.

Reference: Overcoming Catastrophic Interference Using Conceptor-Aided Backpropagation

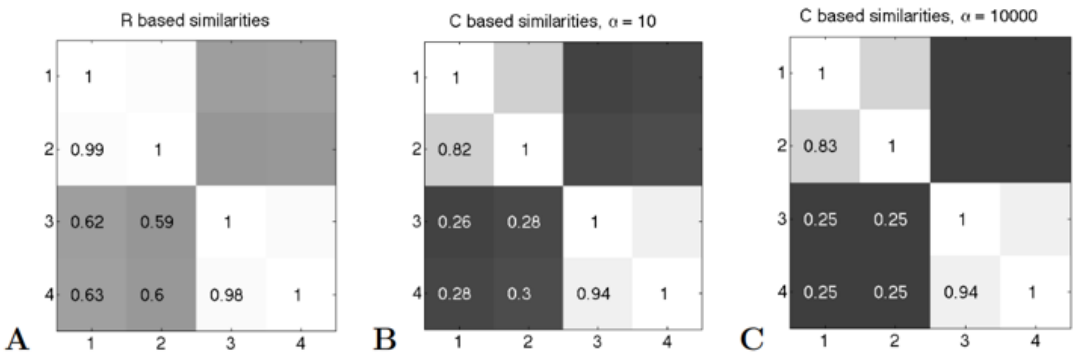


Incremental training on permuted MNIST tasks using conceptor-aided backpropagation (i.e. shielding of past latents) versus a more conventional approach (Source).

Quantify model spread

If two different models exercise different dynamics, and you compute one conceptor for each, then the similarity between conceptors can be used as an effective distance metric between the dynamics implemented by the models. More generally, conceptor-aided similarity can help gauge the spread of models trained in different regimes, which might be useful for quantifying trainer stability. In contrast to KL-divergence on outputs, this approach would enable more a measure of "cognitive", rather than "behavioral" difference, by comparing latents instead of ~policies.

Reference: Controlling Recurrent Neural Networks by Conceptors / A Similarity Measure for Excited Network Dynamics



Pairwise similarity between dynamics associated with two families of two signals each (i.e. sines and 5-periodic signals). Similarity based on conceptors (**B**, **C**), highlights the different natures of the two families more than simply the correlation matrix (**A**) (Source).

Trainer-Level Applications & Beyond

Quantify trainer spread?

If you can compute a conceptor based on the dynamics employed by a model, can't you similarly compute a conceptor based on the dynamics employed by a trainer in moving models around trainer spread? Instead of model states, you'd work with model parametrizations. That might help quantify the stability of a trainer^2, by measuring the variation in the dynamics of various trainers obtained through their automated meta-optimization.

Reference: N/A

Isolate and either amplify or suppress trainer dynamics?

Say you used neural memory management techniques to determine that a model has already internalized a certain dynamic. If you collect multiple instances of trainer trainers yielding trainers which yield models which have internalized inappropriate dynamics (e.g. mesa-optimization), then you might want to *isolate and suppress the dynamic of training towards dynamics of deception*. You might also want to mix and match different training dynamics using conjunctions and disjunctions.

Reference: N/A

Shield trainer dynamics?

This one is pretty weird. Say you have a trainer moving models across model space. A trainer trainer moves it across trainer space. In other words, the trainer^2 changes the way the trainer moves models across model space. You might want to preserve some specific training dynamic during the optimization exerted by the trainer^2. For instance, human amplification loops might be vulnerable to human modification orchestrated by downstream levels of the optimization stack. Shielding some aspect of the initial trainer dynamics (i.e. inducing some

path-dependence on trainer[2]), might be protect it from being arbitrarily overwritten, especially when the trainer is itself trained with backprop. However, the corollary is that human amplification might be hurdled.

Reference: N/A

Quantify novelty of training dynamics?

Similar to how you might check whether a model has already internalized certain dynamics, you might also be able to check whether a trainer being trained has happened to internalize certain dynamics. This hints at some nested boxed setups where automated mechanisms are employed to detect dynamics at various levels in order to e.g. shield and isolate, similar to the speculative applications above.

Reference: N/A

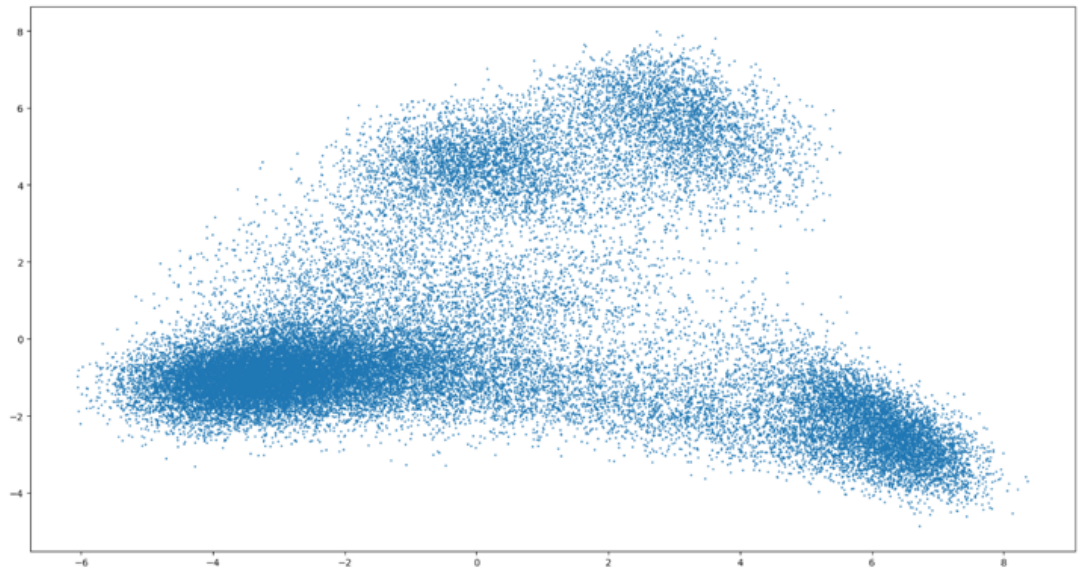
Gauging abstraction relations across trainers?

Similar to one how might infer relations of abstraction between model states coding for different concepts (e.g. fruit > apple), it might be interesting to investigate whether certain classes of trainers only implement subsets of the dynamics (across model space) enabled by a broader trainer class.

Reference: N/A

Limitations

Conceptors are inherently linear. While non-linear variants have been hinted at in conceptor literature, it's difficult to see how the neat mathematical constructions (e.g. Boolean ops) might transfer. Conceptors might not be expressive enough to capture the intricate high-dimensional non-linearities present in contemporary prosaic models. For instance, the interpretability application above already hits related obstacles:

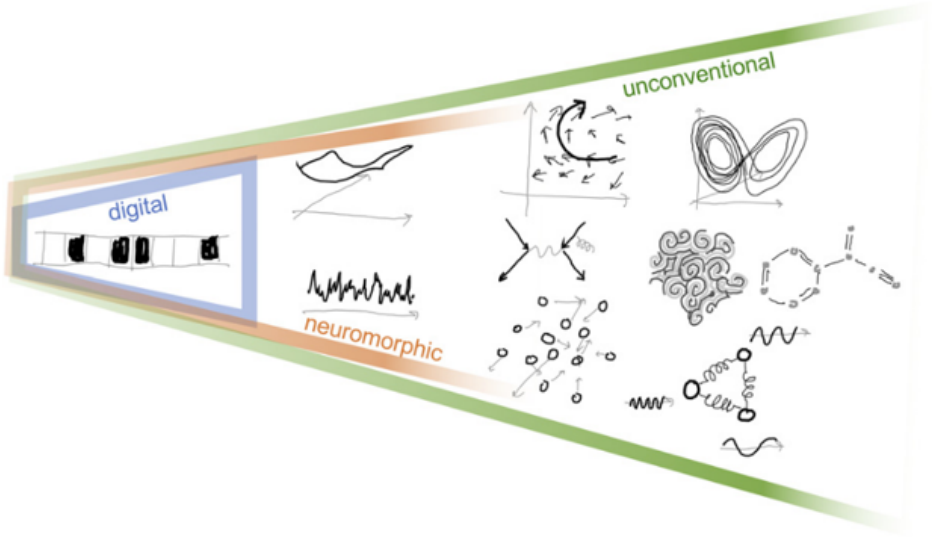


PCA projections of BERT contextual embeddings associated with the symbol "plant". Notably, distinct clusters which are haphazardly spread can be identified. If sampling from K-means clusters, one notices that clusters correspond to different senses: plant the creature, plant the factory, and plant the action of planting (Source).

Besides, even if quote computationally-efficient in general, conceptors still run into computational constraints as the state cloud's dimensionality increases. Computing individual conceptors across a model space of 175B dimensions is currently intractable. However, forthcoming work suggests the feasibility of inserting local conceptors at different locations in a larger model. I've heard a mention of John Wentworth mentioning local PCA which might also be related, but I can't find a good reference.

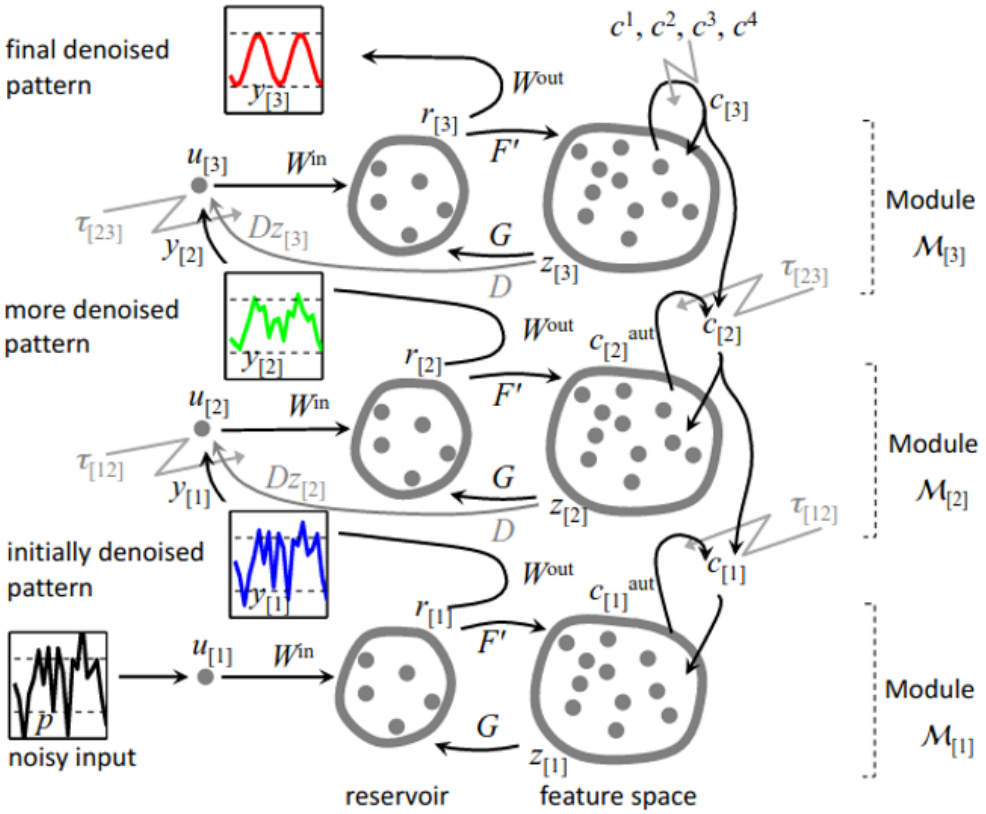
Miscellania

- The research group investigating conceptors is also motivated by neurosymbolic integration in the context of neuromorphic computing:



Napkin drawing of the scope of digital, neuromorphic, and unconventional computing (Source).

- Conceptors have been used in a diffusion-like setup back in 2014:



Nested RNNs implementing incremental denoising while controlled by conceptors (Source).

- Conceptors are also invoked in the context of an agent's conceptual framework evolving over time, using neat ideas from category theory and formal logic:

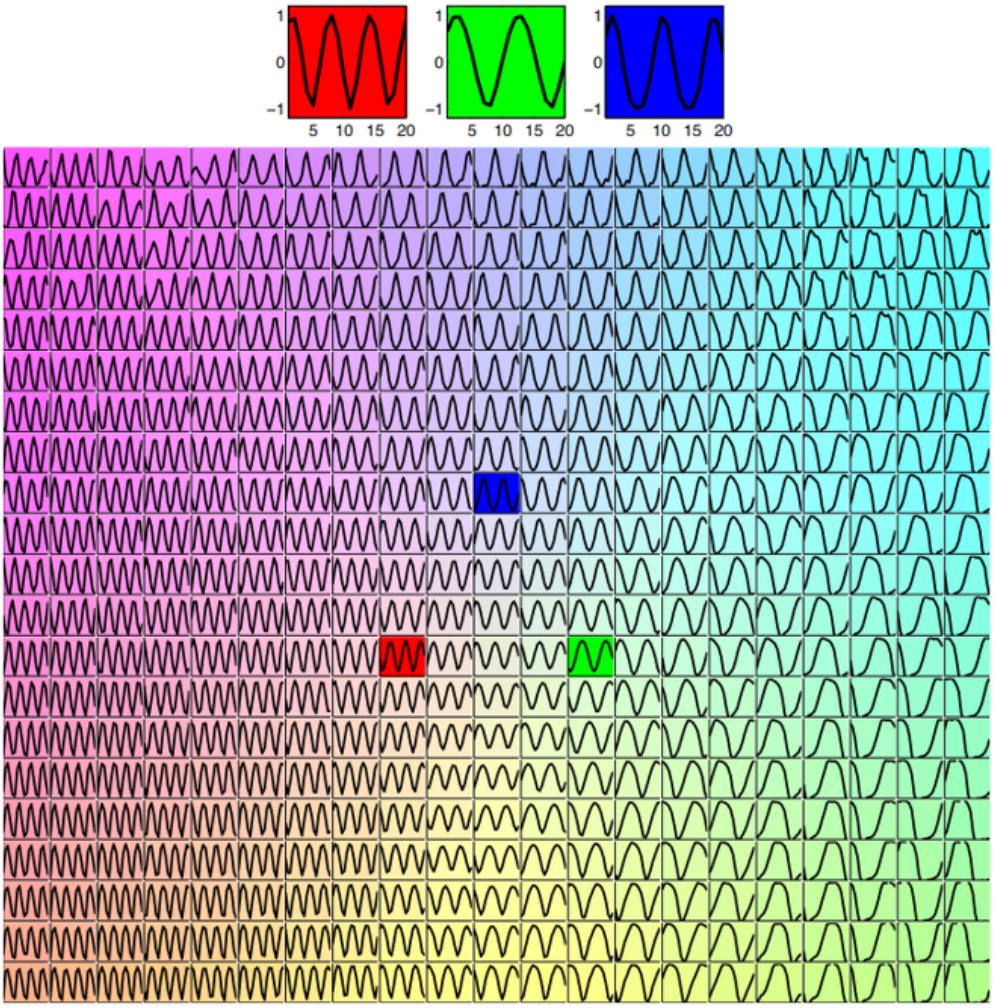
"*Different agents may have different logics. I will therefore not try to define a general* [intrinsic conceptor logic] *that would fit any neural agent. Instead every concrete agent with a concrete lifetime learning history will need his/her/its own individual conceptor logic. [...] Making an ICL private to an agent implies that the model relation ⊨ becomes agent-specific. An ICL cannot be specified as an abstract object in isolation. Before it can be defined, one first needs to have a formal model of a particular agent with a particular lifelong adaptation history. [...] In sum, an ICL (formalized as institution) itself becomes a dynamical system, defined relative to an existing (conceptor-based) neural agent with a particular adaptation history. The "state space" of an ICL will be the set of signatures. A "trajectory" of the temporal evolution of an ICL will essentially be a sequence of signatures, enriched with information pertaining to forming sentences and models.*" — Herbert Jaeger

- The same author explores pervasive issues with cognitive models employing dynamical systems:

"*Attractors, by definition, keep the system trajectory confined within them. Since clearly cognitive processes do not become ultimately trapped in attractors, it has been a long-standing modeling challenge to account for "attractors that can be left again" – that is, to partly disengage from a strict DISA paradigm. Many answers have been proposed. Neural noise is a plausible agent to "kick" a trajectory out of an attractor, but a problem with noise is its unspecificity which is not easily reconciled with systematic information processing. A number of alternative "attractor-like" phenomena have been considered*

that may arise in high-dimensional nonlinear dynamics and offer escapes from the trapping problem: saddle point dynamics or homoclinic cycles; chaotic itinerancy; attractor relics, attractor ruins, or attractor ghosts; transient attractors; unstable attractors; high-dimensional attractors (initially named partial attractors); attractor landscapes." — Herbert Jaeger

- Something something extrapolation:



Interpolating and extrapolating three dynamics (Source).

Outro

Conceptors provide a useful formalism for operating with high-dimensional dynamics, similar to the ones dominating contemporary prosaic architectures. The way they combine the discrete nature of formal logic with the continuous nature of dynamical systems makes them promising candidates for steering high-dimensional dynamics using crisp controls. Their flexibility leads to many low-hanging applications, and probably even more waiting at higher levels of complexity.

Cataloguing Priors in Theory and Practice

Crossposted from the AI Alignment Forum. May contain more technical jargon than usual.

This post is part of my hypothesis subspace sequence, a living collection of proposals I'm exploring at Refine. Preceded by an exploration of Boolean primitives in the context of coupled optimizers .

Thanks Alexander Oldenziel and Paul Colognese for discussions which inspired this post.

Intro

Simplicity prior, speed prior, and stability prior — what do they all have in common? They are all means of tilting an optimization surface towards solutions with certain properties. In other words, they are all heuristics informing the navigation of model space, trainer space, etc. However, what brings them together is also a systematic divide between their theoretical/conceptual/abstract framings and their practical/engineering implementations. All those priors appear to have been used in contemporary ML in one form or another, yet conceptual-heavy researchers are often unaware of those interesting data points, while ML engineers often treat those implementations as mere tricks to improve performance (e.g. generalization).

In the language of coupled optimizers I've explored over the few past posts in the sequence, such heuristics are artifacts of meta-optimization (e.g. a human crafting a trainer by building in certain such tendencies), and often tend to be direct human-made artifacts, rather than the result of downstream optimization. Though this need not be the case, as the simplicity prior happens to itself be quite simple... It might emerge naturally from a trainer which itself is trained to be simple. Similarly, the speed prior happens to itself be quite fast, as penalizing a duration is trivial. It might emerge naturally from a trainer, should it be trained to itself be fast.

Anyway, let's briefly catalog a few popular priors, describe the rationale for employing them, list instances of their use in ML, and finally document possible failure modes associated with blindly following them.

Priors

Some members of the list can be better described as heuristics or biases than priors. However, there are some basic connections between those in that they all cause an optimization process to yield certain outcomes more than others. If you start with a prior of possible ML model parametrizations and use training data to update towards your final distribution, your choice of prior will naturally influence the posterior. This prior can be informed by heuristics, such as "we're more interested in simple models than complex ones from the get-go." Bias as in structural bias, inductive bias, and bias-variance trade-off describes a similar process of tailoring the ML model to broadly yield certain types of results efficiently.

Simplicity

Informally known as Occam's razor, and extremely formally known as Solomonoff prior, the simplicity prior biases optimization towards solutions which are simple. "Simple" here is often operationalized using the minimum description length: what's the shortest description of an algorithm/model/concept/world/etc. required to accurately specify it? Simple candidates are then the ones with a particularly short such shortest length.

The rationale behind employing the simplicity prior in an optimization process is that it systematically reduces the variance of the solution. This means that it increases the odds that the solution will behave in a similar way in different situations, as opposed to growing too reliant on the idiosyncrasies of your finite/limited/bound optimization process. When training ML models, simplicity tends to yield strong generalization performance. When building world models, simplicity tends to yield theories which hold better against new empirical data.

In ML, the use of simplicity is most associated with the bias-variance trade-off and the general quest for avoiding overfitting models to training data. There are specific measures for model simplicity/complexity out there, such as the VC-dimension describing how fine-grained a model can arbitrarily "shatter" data into different classes. However, such measures are not usually used in practice, like ever. In contrast, regularizers such as L1 and L2 norms applied to model weights (in the style of Ridge and Lasso regression), are used extensively. This regularization process penalizes models whose parameters are large and dense (i.e. many things having a non-trivial influence on each other — complex). In other words, it enables trainers to nudge models towards regions of model space deemed relatively simple.

However, simplicity is not enough to e.g. select among all the possible objectives which might have explained the behavior of a human. Quite bluntly, Occam's razor is insufficient to infer the preferences of irrational agents. Humans (i.e. irrational agents) appear not to actually value the simplest thing which would explain their behavior. For another failure mode, consider the notion of acausal attack highlighted by Vanessa Kossoy here, or in terms of the fact that the Solomonoff prior is malign. In this scenario, an AGI incentivized to keep it simple might conclude that the shortest explanation of how the universe works is "[The Great Old One] has been running all possible worlds based on all possible physics, including us." This inference might allegedly incentivize the AGI to defer to The Great Old One as its causal precursor, therefore missing us in the process. Entertaining related ideas with a sprinkle of anthropics has a tendency to get you into the unproductive state of wondering whether you're in the middle of a computation being run by a language model on another plane of existence which is being prompted to generate a John Wentworth post.

Speed

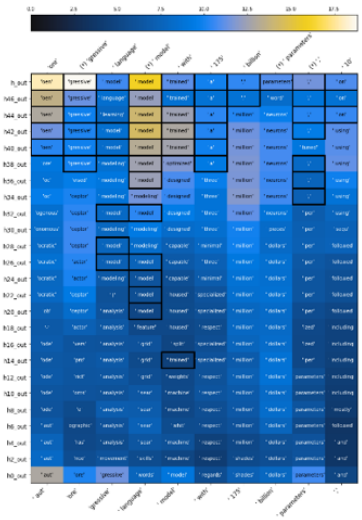
When talking about the description of a phenomenon in the form of an algorithm, short algorithms don't necessarily equate fast algorithms. Running a for-loop over all possible physics and implementing all appears extremely simple to describe, but also extremely time-consuming in terms of ops/iterations/epochs/etc. You might have a simple recipe, but it might take forever to follow.

However, it seems that selecting for fast algorithms might be useful. For instance, deception has been argued to inevitably take longer than the non-deceptive alternative. As seen in the ELK report:

> Human imitation requires doing inference in the entire human Bayes net to answer even a single question. Intuitively, that seems like much more work than using the direct translator to simply "look up" the answer.

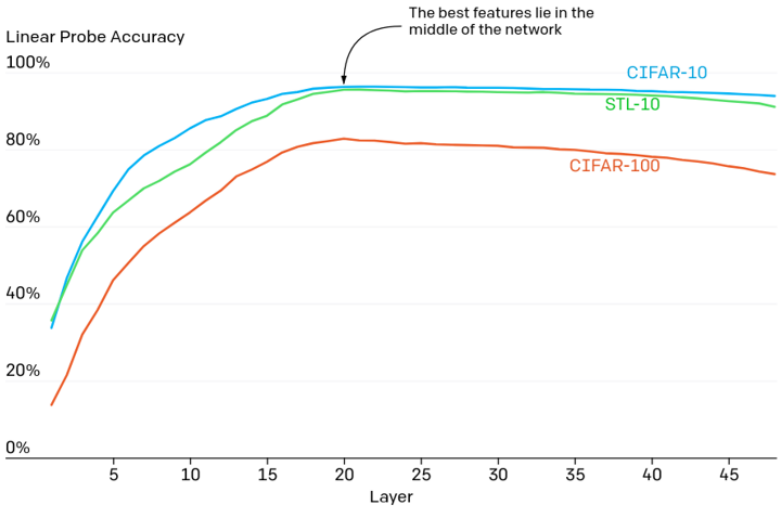
How does this show up in contemporary ML? An interesting instance is the architectural decision of adaptive computation time. In the paper, RNNs are designed to be able to both (1) perform multiple computational steps when processing a certain input, and (2) learn how many steps to conduct during inference. More compute would always help, so won't it just learn to spam more steps? Not really, as the model is also penalized for abusing the number of steps and pondering for too long. A balance between capability and speed has to be struck by the model.

The logit lens could also be seen as relating to the speed prior. Logit lenses attempt to surface the "best guess" of a transformer model at each layer, in contrast to the common practice of just taking the end result after a set amount of computation:



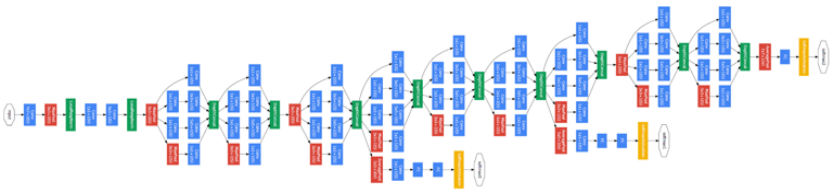
Logit lens recovering intermediate guesses of a transformer. Early-late transformer layers map to bottom-top axis. Horizontal axis captures subsequent predictions. Solid-bordered cells express a match with the final guess. Color roughly expresses the model's confidence in guess (Source).

Linear probing is yet another related technique, and in a sense a more general version of logit lensing. If you arbitrarily cut out a set number of final layers of an ML model in a Procrustean frenzy and map latent activations to outputs using a plain linear regression, what performance can you get? For instance, if the ML model below was penalized for the number of subsequent layers it makes use of, it might as well learn when to cut some computational corners and only use a few dozen layers.



Linear probing at different layers of a transformers trained on pixels, rather than text tokens (Source).

Early vision models (e.g. Inception), also had different output heads at different depths. Supposedly, this helped with training stability, but could also be used in a speed prior setting:



Inception architecture with three output heads (Source).

In the terminology of Paul Colognese and Evan Hubinger, the logit lens, linear probing, and early vision models might tie into the "early-exiting" proposal, while adaptive computation time might relate to the "pondering" class. Those instances of bridging the theory-practice gap might help build such priors into future models.

The flip-side of selecting for speed would be that fast undesirable dynamics would also be encouraged. Perhaps determining what humans would superficially find appropriate takes way less time than determining what they might like if they were smarter, thought for longer, etc. Also, while selecting for simplicity seems to often yield good generalization , selecting for speed inevitably cuts back on performance, as you could always refine your best guess a tiny bit further.

(Structural) Stability

A solution to an optimization process which doesn't completely break down under small perturbations would also seem pretty neat. For instance, such perturbations might consist of small changes to the weights of a model, small changes to the ontology embedded in a world model, or small changes to an algorithm. Bonus points if perturbations can be immediately followed by a return back to the stable state. We wouldn't want the systems we deploy to resemble a pencil balancing on each tip, with the smallest gust of wind across model space leading them into radically different dynamics. The notions of stability and structural stability are explored in more depth here.

In practice, ML models themselves appear to become "immune" to perturbations precisely when they're being trained to operate under said perturbations. For instance, DropOut and its variations constantly sway the model being trained across model space, and condition it to perform well. This nudges the very training process towards region of model space from where such nudges would still yield good performance. One such recipe for stability appears to be redundancy — the ML model develops various mildly-independent processing pathways in a setup reminiscent of model ensembles and bagged estimators, but *internal* to the model's processing. If one such pathway fails, others will hopefully compensate, which appears to be the case as DropOut tends to improve generalization.

Another example is adversarial training, where targeted stressors in the form of adversarial examples are being introduced in the ML model's curriculum. During training, the model learns to cope with such perturbations. This more comprehensive training process takes more compute, which makes it somewhat more expensive, though. This also tends to be the case with DropOut. Yet another related project is MAML, a training paradigm in which an ML model is trained to reach a parametrization from which subsequent training steps would yield strong performance.

A failure mode of structural stability as a prior is that convergent properties such as the drive to gain resources are instrumental objectives in achieving quite a number of terminal feats. This might decrease the effectiveness of stability as a heuristic for navigating model space towards good models, and in turn trainer space towards trainers which produce good models. For priors which trickle down the optimization stack — though mostly in the context of speed priors — have a look at this. Also, structural stability yields rigidity and various forms of lock-in. It's inherently hard to nudge something designed to resist perturbations after it has been deployed.

NTK

While introduced in alignment through the context of simplicity priors, neural tangent kernels allow you to gauge the general "predisposition" of an ML model towards converging to various ways of relating data points. If being able to estimate the way the resulting ML model represents the world this way, we might then be able to tweak the training process itself so that it leads more towards intended perspectives (i.e. ways of grouping together and relating data points). How to then define the intended arrangement of the world in the eye of the ML model then becomes the locus of the hard part of the alignment problem, but like all members of this list, NTK might slightly increase the odds of getting good outcomes even without yielding a full verifiable solution.

> "*I am concerned, in short, with a history of resemblance: on what conditions was Classical thought able to reflect relations of similarity or equivalence between things, relations that would provide a foundation and a justification for their words, their classifications, their systems of exchange? What historical a priori provided the starting-point from which it was possible to define the great checkerboard of distinct identities established against the confused, undefined, faceless, and, as it were, indifferent background of differences?*" — Michael Foucault

For more *tangents* between kernel functions and meaningful cognitive phenomena, I'll shamelessly plug some older writing of mine: dixit kernel functions and expecting unexpected ideas.

Outro

I'm sure there are more such high-level heuristics for optimizing the way of navigating optimization surfaces. The heuristics we've been endowed with by evolution might make for a fruitful place to look for more. This type of conceptual work is probably the most explicit form of direct optimization exerted by human researchers on trainers at the interface, their design choices guiding the nature of downstream optimization.