Abstractions

Daniel C

22nd April 2026

What is Abstraction?

Throwing away information, keeping what matters

- Abstraction is about discarding information while maintaining the ability to predict the systems you care about
- Similar to macrostates and coarse-grainings in statistical mechanics
- **Example:** predicting a star's trajectory from a distant planet. The star contains $\sim 10^{57}$ particles, each with its own position and velocity. But at such long distances, the exact mass distribution within the star doesn't matter – the only information that survives is the total mass M and the center-of-mass position $\mathbf{r}_{\text{cm}}$. All other detail is irrelevant to anything far away
- In general: abstraction keeps information that is relevant to things "far away" (in space, time, or causal structure) and throws out everything else
- Other examples: the output of a CPU computation (exact voltages on individual transistors don't matter beyond a few microns, only the high-level "number" passed out matters)

The Pointers Problem

Human values live in latent variables

- Because of embeddedness, we cannot hold a fully faithful representation of the world in our minds – we must throw out information. Our world models are themselves made out of abstractions and latent variables
- Human values are expressed in terms of these latent variables: we care about tables, strawberries, other people's happiness, instead of world states described as quantum fields. Insofar as humans have utility functions, the inputs are latent variables in our world model, not low-level physical states
- But we care about the *actual* state of the world, not just our own estimate of it – We want other people to *actually* be happy, not just to look-to-me like they're happy

The Pointers Problem

Why alignment requires solving the pointers problem

- Since our values are functions of latent variables, translating our values to an AI requires establishing a **correspondence between latent variables** in the human's vs. the AI's world model. Without this, we cannot even define what it means for the AI to learn our values – the inputs to our utility function would be undefined in the AI's ontology
- We also need to understand how abstractions in our world models point to **actual things in the real world**. We care about real-world outcomes, but the AI's planning refers to its own internal representations. We need to ensure that when the AI optimizes for "strawberry" in its ontology, this points to the same real-world pattern that "strawberry" points to in ours

What Do We Need from a Theory of Abstraction?

Uniqueness and agreement theorems

- **Key question:** under what conditions would different agents – with different world models, different training data, different architectures – converge to using the *same* abstractions?
- If two agents both basically understand their environment (i.e. they agree on predictions about any directly observable thing), their world models might still have totally different *internal* structure. One agent might model data as generated by a biased die, another by a complicated neural net. A uniqueness/agreement theorem would tell us when the internal latent variables must nonetheless be approximately isomorphic
- This would make alignment far more tractable: if the human's latent variable "strawberry" and the AI's corresponding internal representation are guaranteed to be approximately isomorphic, then the human's utility function – expressed in terms of "strawberry" – can be faithfully translated into the AI's ontology
- More broadly, uniqueness tells you whether your assumptions were enough to "pin down" all the degrees of freedom

Decomposition into Structured Concepts

Why is our world model made of distinct concepts?

- Our world models are not compressed into a single uninterpretable blob – they are organized into collections of distinct, reusable concepts ("strawberry", "red", "round", ...). Why is that the case?
- **Compression vs. organisation:** information theory tells us how to *compress* data efficiently – minimizing the total code length. But a maximally compressed file is useless until you decompress the whole thing; you cannot look up a specific fact or answer a specific question without reconstructing everything
- **The question:** what is the criterion for *how* we decompose our world model into concepts? Why does "waterbottle" get to be a single concept, while "the left half of the water bottle" does not?
- A good theory of abstraction should explain this decomposition and predict when two agents will decompose the world into approximately the same conceptual structure

What is the Type Signature of an Abstraction?

Abstractions are not about particular things

- The concept of "strawberry" is not about any *particular* strawberry – it is not a claim about the state of the world, and it does not directly have mutual information with any specific physical system
- A particular strawberry on a table is a latent variable that carries information about the world. But the *abstract concept* of strawberry – the thing that lets you recognize new strawberries, reason about strawberries in general, compose "strawberry" with other concepts – is something different. It is more like a *template* or *type* than a specific variable
- This raises a question about the *data structure* of abstractions: what kind of mathematical object is an abstract concept, if not a random variable with mutual information about the world?
- Are there a small number of such data structures that can compose into world models as expressive as our own? If so, a theory of abstraction should identify them – and the compositionality rules that let them build up into full world models

Summary: What We Need from a Theory of Abstraction

Desiderata

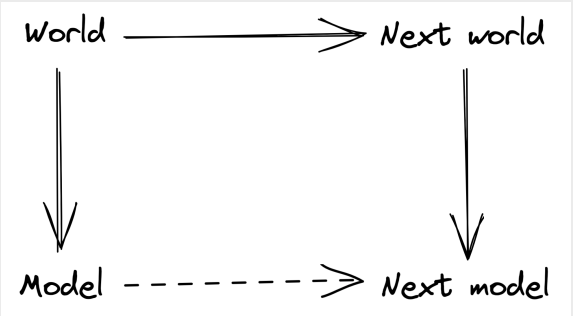
1. **Abstraction as throwing away information:** formalize what it means to throw out information while preserving predictive power – what to keep, what to discard, and why
2. **Uniqueness/agreement theorems:** under what conditions do different agents converge to the same abstractions? When can we guarantee that latent variables across different world models are approximately isomorphic?
3. **Decomposition and type signature:** why is our world model decomposed into distinct, reusable concepts rather than an uninterpretable blob? What is the data structure of an abstract concept, and can a small number of such data structures compose into world models as expressive as our own?

The rest of this lecture will introduce different theoretical frames – each tackling different pieces of these desiderata.

Preserving Structure

Abstraction as structure-preserving reduction

- We often want to throw out information in a way that *preserves structure*
- **Simplest case – time evolution:** we can either evolve the full system forward in time and then coarse-grain, or coarse-grain first and then evolve. These should yield the same result:

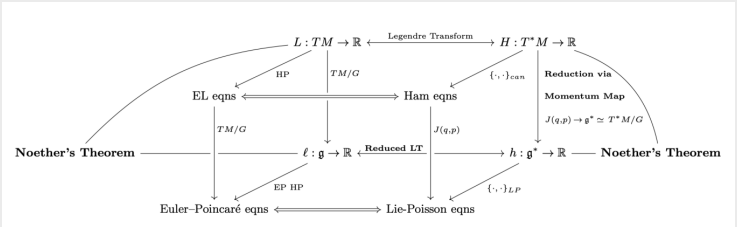


- But there are many coarse-graining maps satisfying this commutativity property – many degrees of freedom remain about *exactly* what information to throw away

Preserving Structure

Abstraction as structure-preserving reduction

- Furthermore, time evolution may not be the right structure to preserve. E.g. in physics, we might instead preserve the *Lagrangian*: if a system has a symmetry group G, we can replace the full Lagrangian $L(g, \dot{g})$ on the configuration space with a reduced Lagrangian $\ell(e, g^{-1}\dot{g})$ – discarding redundant information while preserving the variational structure that generates the equations of motion
- **Example – rigid body rotation:** a freely spinning rigid body is described by its orientation $g \in SO(3)$ and angular velocity $\dot{g}$. But the physics is rotationally symmetric, so we can pass to the *body frame*, replacing the full description $L(g, \dot{g})$ with a reduced description $\ell(\omega)$ that depends only on the body angular velocity $\omega = g^{-1}\dot{g}$. The orientation is discarded; only the angular velocity in the rotating frame is kept.

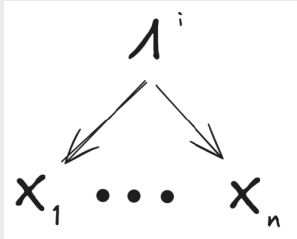


- The study of abstraction is about having a general framework – one that doesn't overanchor on time evolution or any other specific structure – that pins down the remaining degrees of freedom about what information to keep and what to throw out

Natural latent and mediators

Capturing all the correlation between observables

- Suppose we have a collection of observables $X_1, \ldots, X_n$ (e.g. different chunks of an ideal gas, different measurements of the same system)



- The **total correlation** among $X_1, \ldots, X_n$ measures how much they collectively deviate from independence:

$$\mathrm{TC}(X_1; \ldots; X_n) = D_{\mathrm{KL}}\left(P(X_1, \ldots, X_n) \,\middle\|\, \prod_{i=1}^{n} P(X_i)\right)$$

- A **mediator** Λ^* is a latent variable that captures *all* this correlation: conditioned on Λ^*, the observables become independent

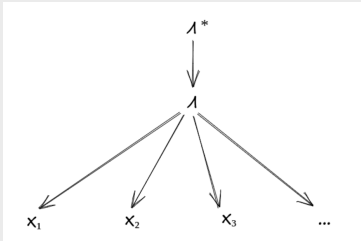
$$TC(X_1; \ldots; X_n | \Lambda) = 0$$

- Equivalently: if you have access to some X_i and are trying to predict any other X_j, then all the information in X_i that is relevant to X_j is already captured by the mediator Λ^*

Minimal Mediator and Uniqueness

Pinning down the abstraction

- We want abstractions that are used by a wide variety of agents. But a mediator Λ^* might include *all* the information shared among $X_1, \ldots, X_n$ *plus* a bunch of irrelevant information – e.g. appending random noise to a mediator still yields a mediator
- So what we want is the **minimal mediator** Λ: the smallest variable that still mediates between all the observables – the information shared among *all* mediators



- **Key property:** for *any* mediator Λ^*, the minimal mediator Λ is a deterministic function of Λ^*:

$$\Lambda = f(\Lambda^*) \text{ or } H(\Lambda|\Lambda^*) \leq \epsilon$$

- This gives us a uniqueness guarantee: any latent variable that *any* agent comes up with, which captures **all** the correlation between the observables, is guaranteed to include the minimal mediator as a component
- **Example:** any latent variable Λ' that mediates between two far-apart chunks of ideal gas must encode the temperature – because temperature is (part of) the minimal mediator for that system

Redundancy

Abstractions must be recoverable from many places

- We still have observables $X_1, \ldots, X_n$, but different agents may have access to *different* observables
- If we want different agents to converge to the *same* abstraction, then that latent variable must be **redundant**: recoverable from many different observables. Otherwise, an agent who only has access to X_3 might be unable to recover the latent at all
- (If the same conclusion can be reached from many different places, then that conclusion must be derivable from each of those places independently)
- Formally, a **redund** Λ is a latent variable satisfying a bound on conditional entropies: for each observable X_i,

$$H(\Lambda \mid X_i) \leq \epsilon$$

(i.e. Λ can be approximately recovered from any single X_i)

- **Maximal redund:** the largest redundant latent variable, such that all other redunds Λ' are deterministic functions of it: $\Lambda' = f(\Lambda)$. It captures *all* the information that is redundantly present across the observables
- **Example:** the temperature of an ideal gas can be recovered from any sufficiently large chunk of gas in the room – it is redundant across all such chunks

Natural Latents

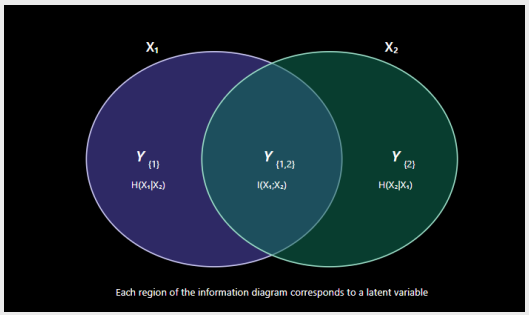
The unique abstraction

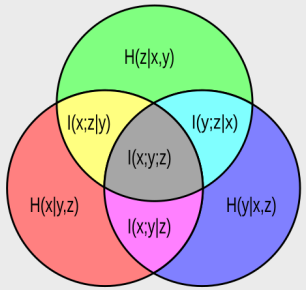
- **Key fact:** any redund is a deterministic function of any mediator. (Intuition: a mediator captures all correlation among the observables; a redund is recoverable from any single observable; but the only information in a single observable that's relevant to the others is what the mediator provides – so the redund must be extractable from the mediator)
- A **natural latent** is a latent variable that is simultaneously a *redund* and a *mediator*
- Note that this means a natural latent Λ is both *minimal* mediator and *maximal* redund: Since any redund is a deterministic function of any mediator, Λ (as both a mediator & redund) is a deterministic function of every mediator – which is exactly what it means to be the minimal mediator.
- It is the **smallest** variable that captures **all** the correlation between the observables, and the **largest** variable that can be recovered from any one of the observables
- **Uniqueness:** any two natural latents used by different agents are approximately isomorphic to each other: Given two natural latents Λ and Λ' we have $H(\Lambda|\Lambda') \leq \epsilon$ and $H(\Lambda'|\Lambda) \leq \epsilon$

Condensation: A Second Frame

What if each region corresponds to an actual random variable?

- Given observables $X_1, \ldots, X_n$, the joint entropy decomposes into regions of a mutual information diagram (conditional entropies, mutual information, conditional mutual information...)
- **Condensation:** what if each of these regions corresponds to an actual *random variable*?



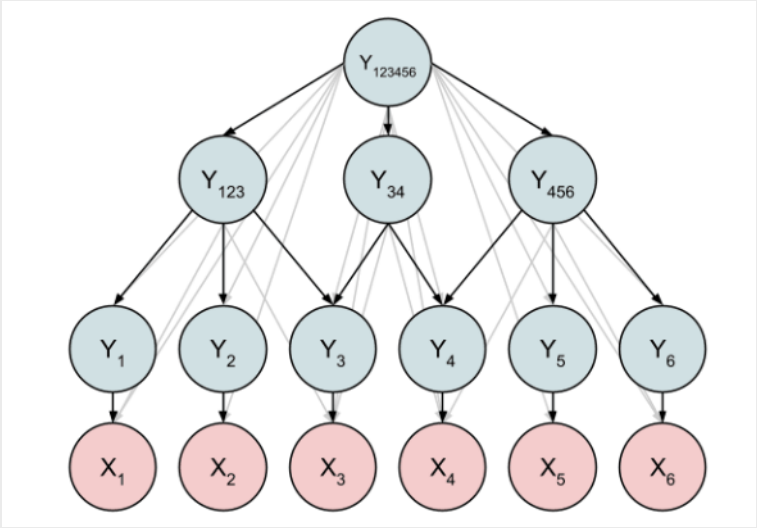


- For two observables: $Y_{\{1,2\}}$ represents the information shared between X_1 and X_2; $Y_{\{1\}}$ represents the information unique to X_1; $Y_{\{2\}}$ represents the information unique to X_2
- This gives an objective way to carve up variables into distinct concepts
- This is not automatically true: in general, an information-theoretic region need not correspond to any actual random variable. Condensation studies the scenario in which some version of this *is* true – and asks when agents will agree on the resulting decomposition

Condensation: Formal Setup

Latent variable models for decomposition of concepts

- **Observables:** random variables $X_1, \ldots, X_n$
- **Latent variables:** Y_A for each subset $A \subseteq \{1, \ldots, n\}$, where the subscript indicates *which* observables Y_A contributes to
- **Latent variable model condition:** each X_i is a deterministic function of $(Y_A,\ i \in A)$ – i.e. each observable can be fully recovered from all the latent variables "above" it



- **Example:** for $X_1, \ldots, X_6$, a latent variable model might have $Y_{\{1,2,3,4,5,6\}}$ (information shared by all), $Y_{\{1,2,3\}}$ and $Y_{\{4,5,6\}}$ (information shared within groups), and $Y_{\{i\}}$ (information unique to each X_i)

Condensation: Why Concepts Are Distinguishable

Decomposition via retrieval costs

- We can think of condensation as attempting to answer the question "why/how can an agent's world model be decomposed into structured components?"
- Each latent variable Y_A is labelled by *which* observables it contributes to – e.g. $Y_{\{1,2\}}$ is only useful for predicting X_1 and X_2
- We can distinguish between latent variables insofar as they are useful for predicting *different* observables
- We can frame it as a **retrieval cost** argument: if we only care about a subset of variables A, we only need to consider the latent variables that contribute to A – carving up information by which observables it serves makes retrieval efficient
- **Example:** we can isolate "strawberry" as a single concept because all information about it (shape, colour, taste) is used in similar contexts, while information about unrelated concepts (e.g. water bottle) is useful in different contexts

Condensation: Key Notation

Notation

Let A be a subset of $\{1, \ldots, n\}$. We define the following collections:

- $Y_{\supseteq A} = (Y_B,\ A \subseteq B)$: every latent variable that simultaneously contributes to *all* the observables in A. Think of this as the "maximal information that is shared among the observables in A"
- $Y_{\supsetneq A} = (Y_B,\ A \subsetneq B)$: same, but strictly above A (every latent variable that contributes to all observables in A and (strictly) more)
- $Y_{\cap A} = (Y_B,\ B \cap A \neq \emptyset)$: all latent variables that contribute to *any* variable in A
- $Y_{\ni i} = (Y_B,\ i \in B)$: all latent variables that contribute to X_i. By the latent variable model condition, $Y_{\ni i}$ is sufficient to recover X_i

Condensation: Conditioned and simple Scores

Ruling out irrelevant information

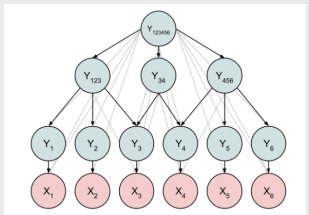
- The latent variable model condition only tells us that $Y_{\ni i}$ is *sufficient* to determine X_i, but the latent variables may also include irrelevant information. For uniqueness/agreement theorems, we want to ensure e.g. that $Y_{\{1,2\}}$ contains **only** the information shared between X_1 and X_2
- **Conditioned score:**

$$\chi(A) = \sum_{B \cap A \neq \emptyset} H(Y_B \mid Y_{\supsetneq B})$$

- **Simple score:**

$$\sigma(A) = \sum_{B \cap A \neq \emptyset} H(Y_B)$$

- A latent variable model is a **perfect condensation** if $\chi(A) = H(X_A)$ for all subsets A
- Intuitively, minimizing the scores incentivizes each latent variable to throw out any information not relevant to the observables it contributes to – so that the decomposition into concepts is as tight as possible



Condensation: Redundancy and Agreement

Perfect condensation implies redundancy

- For a perfect condensation, Y_A is a deterministic function of X_i whenever $i \in A$
- Intuitively: if a latent variable contributes to an observable, then *all* the information in that latent variable is recoverable from that observable alone
- This is similar to the redundancy condition – connecting condensation back to natural latents

Agreement theorem for condensation

- **Theorem:** if (Y_A) and (Z_A) are both perfect condensations of the same observables $(X_i)_{i \in I}$, then for every subset A:

$$Y_A = f(Z_{\supseteq A}) \qquad \text{and} \qquad Z_A = g(Y_{\supseteq A})$$

- Each individual concept Y_A in one agent's decomposition is recoverable from the *corresponding cone* $Z_{\supseteq A}$ in the other agent's decomposition (and vice versa)
- This is a more fine-grained agreement result than natural latents: instead of just saying that the overall latent is isomorphic, condensation says the *internal decomposition into sub-concepts* is also shared – two agents who condense the data well will carve it into approximately the same conceptual pieces

Limitations

Where the current frameworks fall short

- **Strong assumptions in natural latents:** redundancy requires that the latent variable can be recovered from *any* single observable, which is often too strong – many real abstractions are only recoverable from certain (collections) of observations, not all of them
- **Dependence on partitioning of observables:** both natural latents and condensation require a fixed partitioning of the world into observables $X_1, \ldots, X_n$. The results depend on this choice, but there's no canonical way to partition the world
- **Context-dependent contribution:** condensation distinguishes between latent variables by which observables they contribute to, but a latent variable might contribute to *different* observables in different contexts
  - **Example:** if we are predicting a sequence of text and the observables are words at different positions, the concept of "strawberry" may be relevant at different positions depending on context – it doesn't have a fixed set of observables it contributes to
  - We may need a **stochastic contribution relation**: a latent variable contributes to different observables depending on the value of another latent variable

Recap

What we covered

- **Abstraction** is about throwing away information while maintaining predictive power
- We care about abstractions because human values are expressed in terms of latent variables in our world models, and our world models are themselves made of abstractions (they throw out information about the world)
- We want **uniqueness/agreement theorems** guaranteeing that agents converge upon similar abstractions under reasonable assumptions – where the remaining uncertainty is whether/when those assumptions hold in reality
- We want to understand **how and why** our world models are decomposed into structured concepts, rather than compressed into an uninterpretable blob