

# Paper: AGI from the perspectives of categorical logic and algebraic geometry



YKY

August 9, 2024

- 1 How does “No Free Lunch” guide us to accelerate AGI?
  - The hypothesis space, ie. the space of AGIs
  - A motivating example: symmetric NNs
  - Dumb structures and dumb knowledge
- 2 Categorical logic vs algebraic logic
- 3 Fibrations in particular

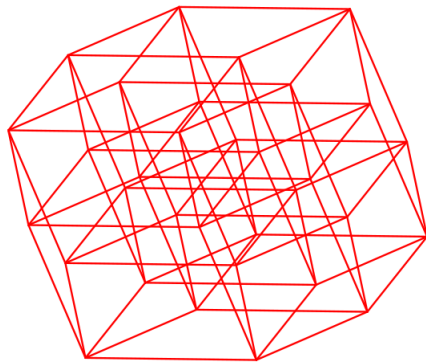
## Part I

How does “No Free Lunch” guide us to  
accelerate AGI?

# The hypothesis space, ie. the space of AGIs

In deep learning, our hypothesis space = neural networks = parameter space = weight space =  $\mathbb{R}^n$  or  $[0, 1]^n$  if truncated.

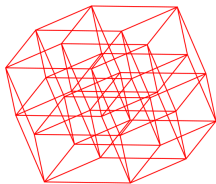
It looks like this:



Imagine a loss function as sitting above this space; We seek to minimize it by gradient descent.

# What can “No Free Lunch” say about the space of AGIs?

- We believe a fair coin toss has probability  $1/2$  by making an **ignorance assumption** and applying the maximum entropy principle. The ignorance assumption says that we have no reason to believe that one side of the coin is more probable than the other.
- By a similar argument,



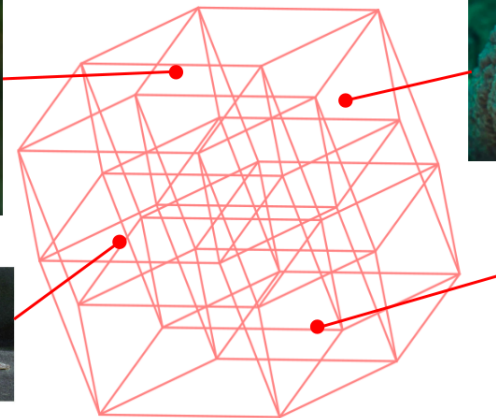
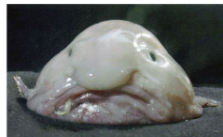
# Unreasonable effectiveness of gradient descent

- Gradient descent can keep going down *for a long time* as there are millions of dimensions.
- The “unreasonable” effectiveness of gradient descent in deep learning is currently our most powerful weapon to solve the AGI problem. Its explanation is likely very difficult, as it hinges on the  $P = ? \text{ NP}$  problem.



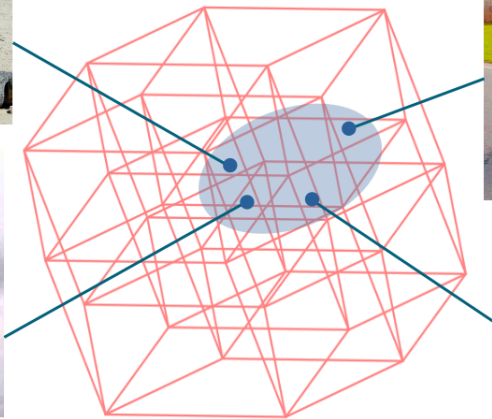
# The hypothesis space has plenty of “dumb structures”

- It seems that dumb learning-machine structures (animals) are more common than smart ones



Even if we have found an intelligent structure, it may still contain lots of “dumb knowledge” in that sub-space

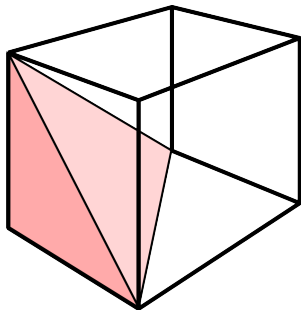
- It also seems that dumb knowledge is more common than smart knowledge





# A motivating example: symmetric neural networks

- Permutation symmetry or commutativity ( $ab = ba$ ) is the best-known symmetry in mathematics
- The input space, as a hypercube, is symmetric under exchange of vertices. The **fundamental domain** of this symmetry is one *corner* of the hypercube:



(1)

(Note that this hypercube is the *input* space, not the mapping / hypothesis space)

- Building a symmetry into a neural network may seem a daunting task, but recent research shows that merely **projecting** all data points to the fundamental domain has the same effect as achieving symmetry [1].

# A motivating example: symmetric neural networks

- If we restrict to binary logic, the total number of Boolean functions in  $n$  variables is  $2^{2^n}$  whereas the number of **symmetric** Boolean functions is  $2^{n+1}$ . So we clearly see an *exponential* reduction in the size of the hypothesis space.
- Tantalizingly, the Transformer also has this permutation symmetry, and researchers have proposed that Transformers are capable of performing *syntactic* manipulations similar to those in proofs of symbolic / predicate logic.
- We can further relax the restriction of binary logic to  $k$ -ary logic, ie, increase the number of lattice points in the hypercube, and the above analysis still holds qualitatively.

## Part II

### Categorical logic vs algebraic logic

# Curry-Howard isomorphism

Many readers may be familiar with this already, so I'll just highlight some interesting points...

- The starting point of the Curry-Howard isomorphism is to make the following identification:

$$\boxed{\text{implication in logic}} \quad A \rightarrow B \quad A \xrightarrow{f} B \quad \boxed{\text{function in type theory}}$$

- An example in type theory: “**currying**” means to convert a 2-argument function into a single-argument function which returns another function, ie.  $f(x, y) = (g(x))(y)$ .

The types of  $f$  and  $g$  are equivalent, ie,

$$X \rightarrow (Y \rightarrow Z) \simeq (X \times Y) \rightarrow Z.$$

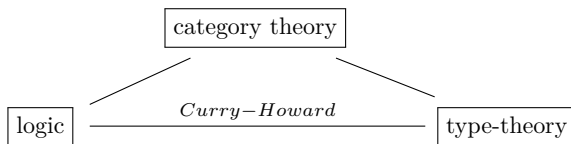
If we treat the above as logic it becomes:

$X \rightarrow (Y \rightarrow Z) \equiv (X \wedge Y) \rightarrow Z$ . One can verify this is a valid logic formula.

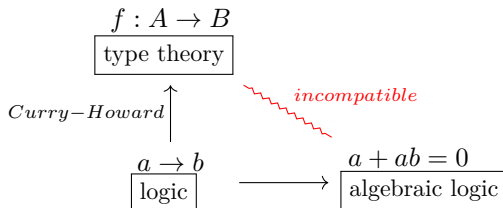
- This and other amazing coincidences led us to believe that CHI has some deep truth in it.

# Categorical logic **clashes** with algebraic logic

- The late Joachim Lambek proposed a “holy trinity” between category theory, logic, and type theory:



- But the Curry-Howard approach is fundamentally *incompatible* with the **algebraic logic** approach, for example:



- I have been working in the Curry-Howard direction because it feels more “modern” but lately I began to think the algebraic logic approach may have more potential for computational acceleration.

## Part III

### Fibrations in particular

# Definition of a fiber bundle

A **fiber bundle** is a tuple  $\xi = (E, p, B, F)$  such that:

- (i)  $E = \text{total space}$
- (ii)  $B = \text{base space}$
- (iii)  $F$  = a topological space called the **fiber** of  $\xi$
- (iv)  $\downarrow_p^E$  is a continuous surjective map, called the **projection**
- (v) for each point  $b \in B$ , the inverse image  $p^{-1}(b) = F_b$ , called the fiber over  $b$ , is homeomorphic to  $F$
- (vi)  $B$  has an opening covering  $\{U_a\}_{a \in A}$  such that for each  $a \in A$ , there is a homeomorphism:  $\psi_a : U_a \times F \rightarrow p^{-1}(U_a)$ .

If  $F$  is a discrete space, then the structure  $F \hookrightarrow E \xrightarrow{p} B$  is called a **covering** of  $B$ .

The condition (vi) is just a re-phrase of (v) in the form of open sets, similar to the “gluing” together of charts in differential manifolds. So the essential condition is (v).

# The “boring” cross product



is equivalent to  $A \times A \times A$ .



## Part IV

The “algebraic logic” approach

- Traditionally, algebra (and algebraic geometry) is concerned with systems of polynomials over the “classical” number fields  $\mathbb{R}$  or  $\mathbb{C}$ . The polynomials are built on addition and multiplication of numbers, that we’re all familiar with.
- Propositional logic  $(B, \wedge, \vee, \neg, \top, \perp)$  can be equated with Boolean rings  $(B, \cdot, +, -, 0, 1)$ , with ring multiplication identified as  $\wedge$  and ring addition identified as symmetric difference (“XOR”).
- As predicate logic involves more operators (eg.  $\forall$  and  $\exists$ ), the classical algebra of numbers seems inadequate to represent them.
- For example, cylindric algebra has special operators  $c_i$  (cylindrification) and  $\text{ld}_{ij}$  (diagonal).

# Basic setup

Thanks for watching 😊

- [1] Aslan, Platt, and Sheard. “Group invariant machine learning by fundamental domain projections”. In: Proceedings of Machine Learning Research 197 (2023), pp. 181–218.