The University of Texas at Austin
Chandra Department of Electrical and Computer Engineering
Cockrell School of Engineering

How Neural Networks Represent and Learn Symbols?

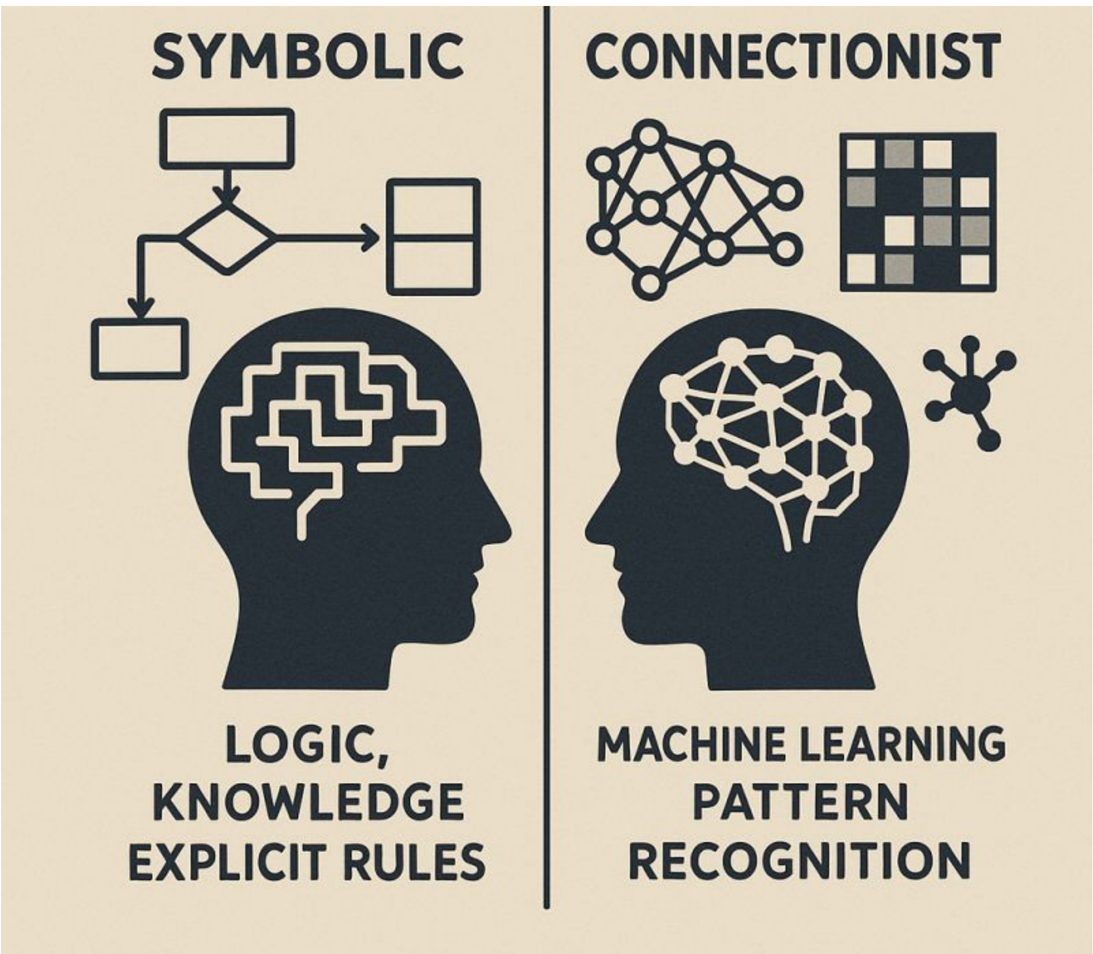
Peihao Wang
University of Texas at Austin

Covered work done with Prof. Atlas Wang

with support from Google.org

VITA

Decades of Debate



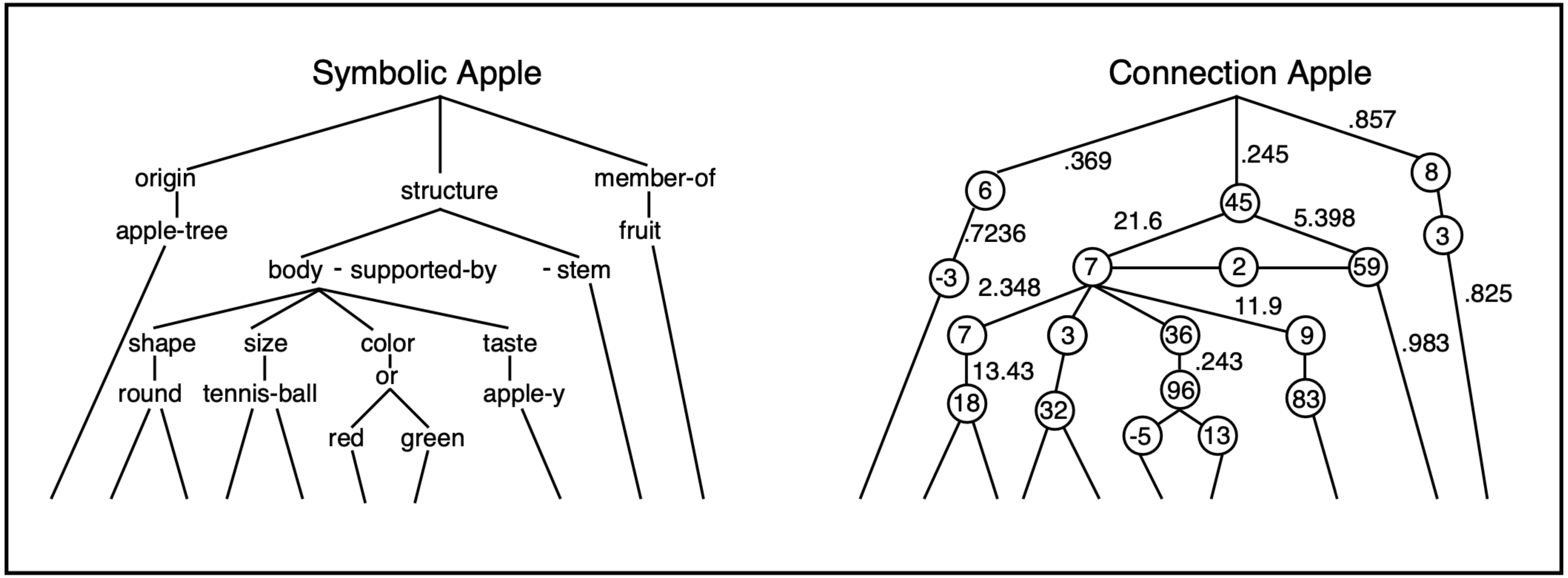
Decision trees

Expert System

Symbolism vs Connectionism

Logic

Automation



Neural Networks

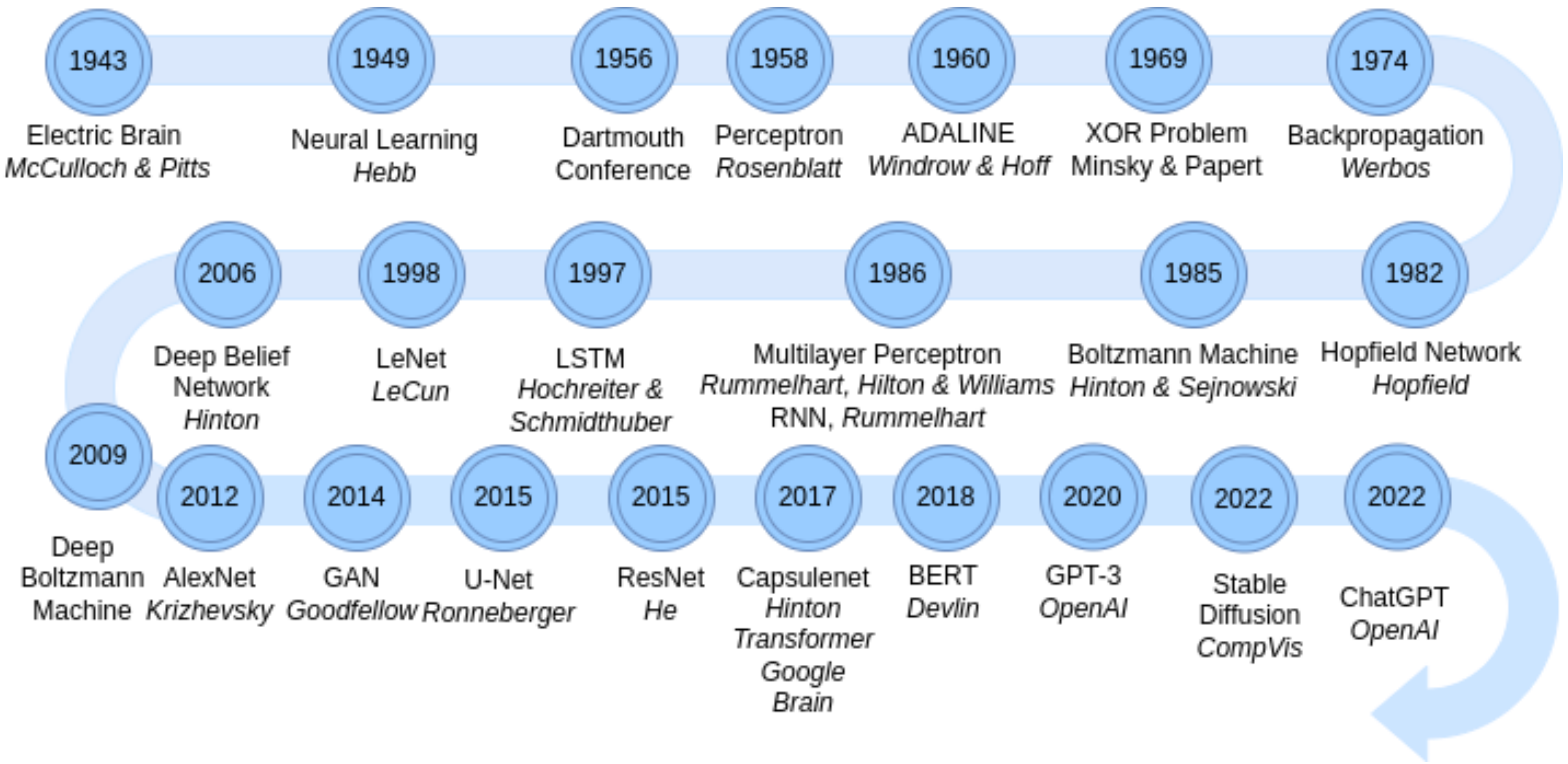
CNNs

LLMs

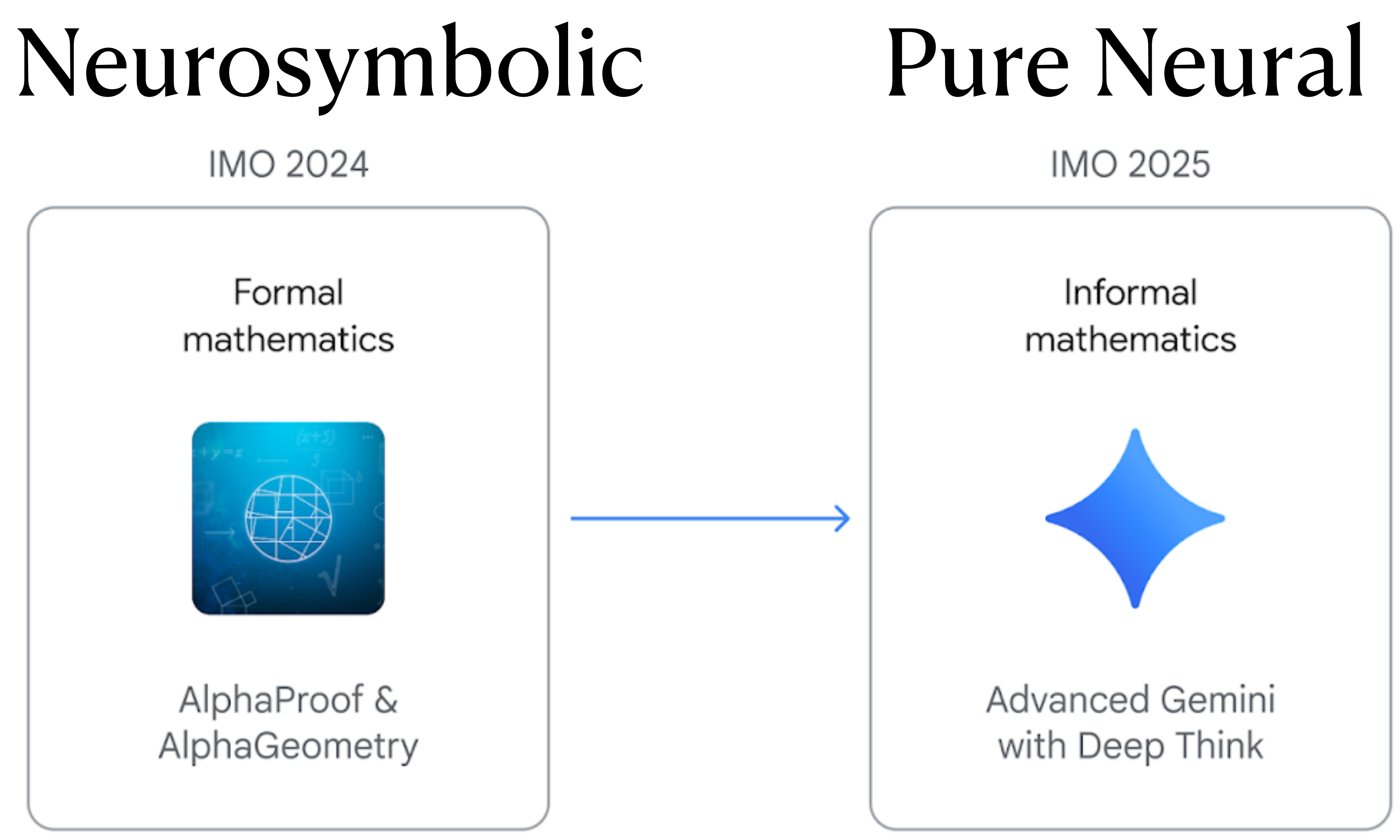
Transformer

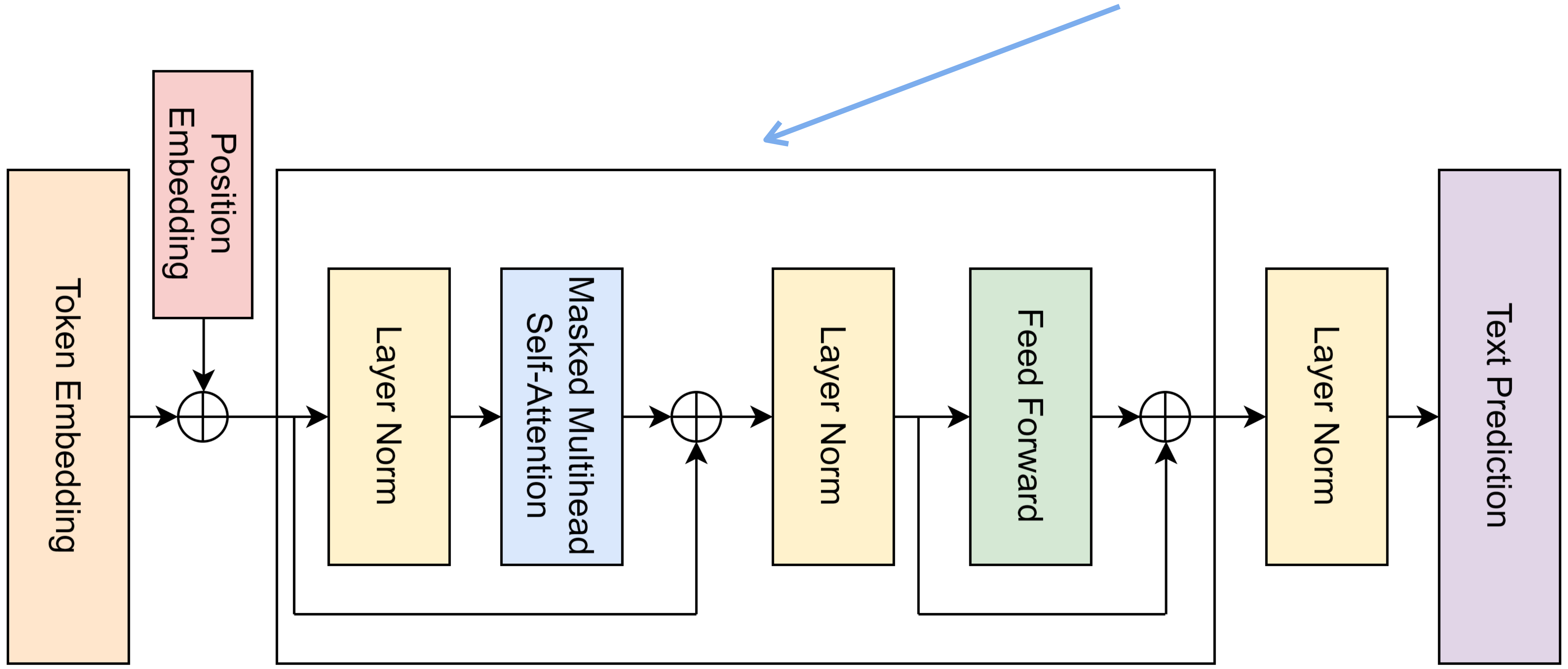
Illustration by Prof. Marvin Minsky

The Rise of Neural Networks



LLMs become IMO gold medalists ...





NEWS | 24 July 2025

DeepMind and OpenAI models solve maths problems at level of top students

For the first time, large language models performed on a par with gold medallists in the International Mathematical Olympiad.

Google DeepMind @GoogleDeepMind · Jul 21
An advanced version of Gemini with Deep Think has officially achieved **gold medal**-level performance at the International Mathematical Olympiad. 🥇

It solved 5 out of 6 exceptionally difficult problems, involving algebra, combinatorics, geometry and number theory. Here's how 🧵

Demis Hassabis @demishassabis · Jul 21
Official results are in - Gemini achieved **gold-medal** level in the International Mathematical Olympiad! 🏆 An advanced version was able to solve 5 out of 6 problems. Incredible progress - huge congrats to @lmthang and the team!

Why Still Symbolism?

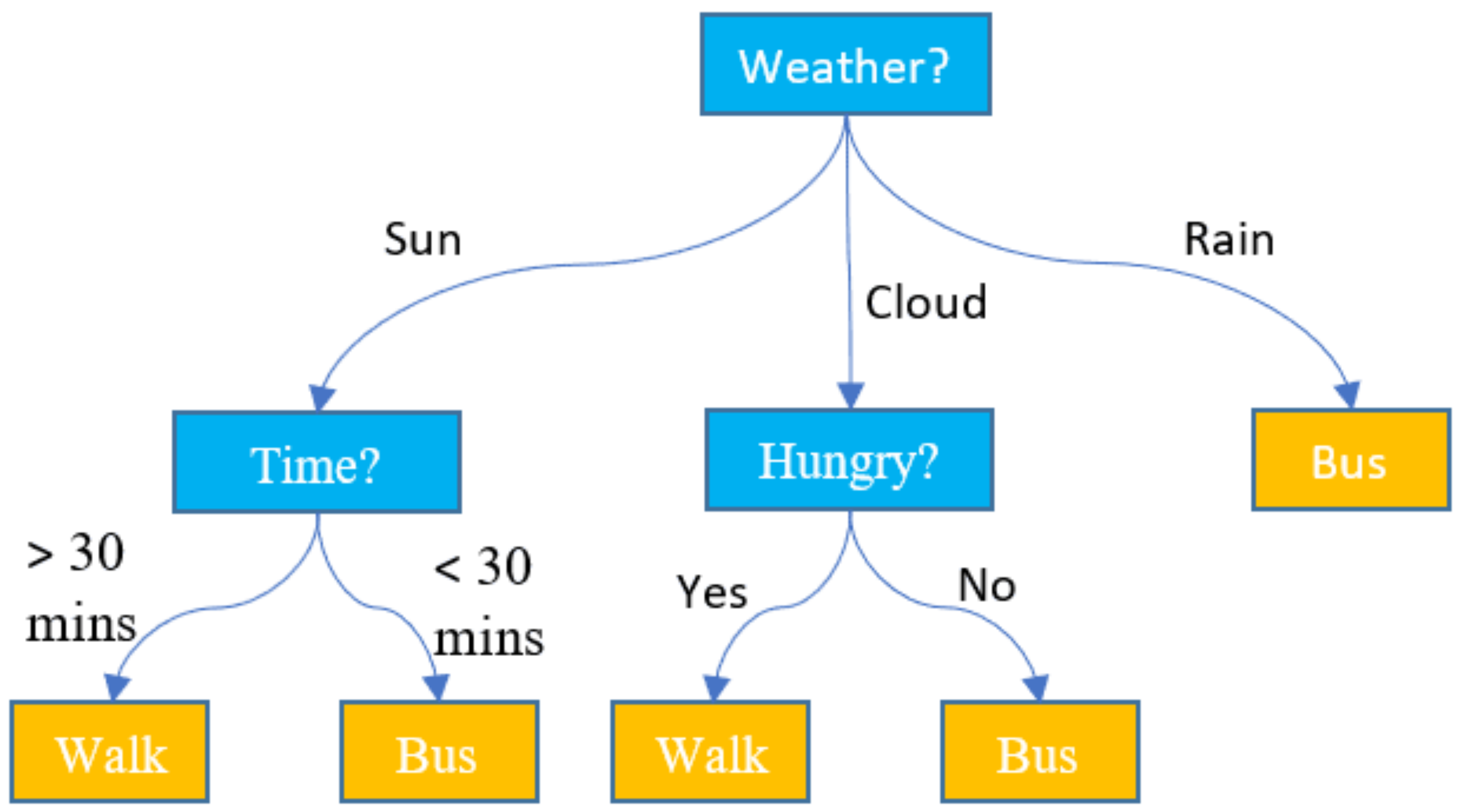
- **Reliability**
  - Computation is exact, precise and generalizable
  - Reasoning trace and decision logic are transparent.
- **Efficiency**
  - No need of billions of parameters
  - Applying logical rules is fast.
- **Compositionality**
  - Given two sets of rules, they can be combined easily with AND/OR logics

Examples:

IF fever AND sore throat THEN possible infection

IF infection AND high white blood cell count THEN bacterial infection

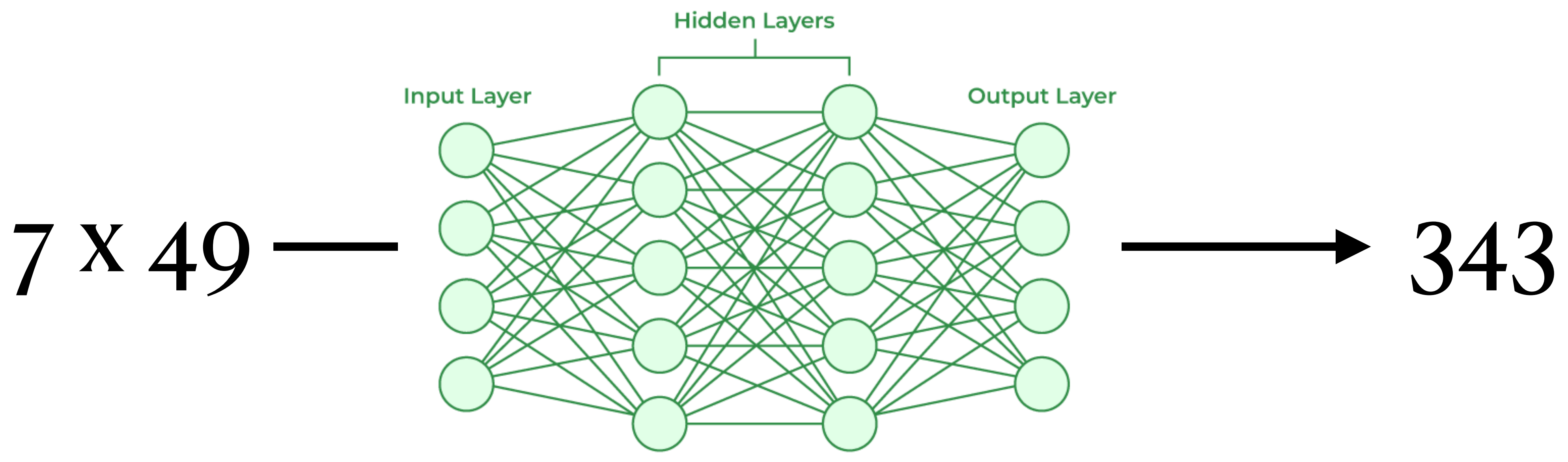
IF bacterial infection AND ear pain THEN ear infection



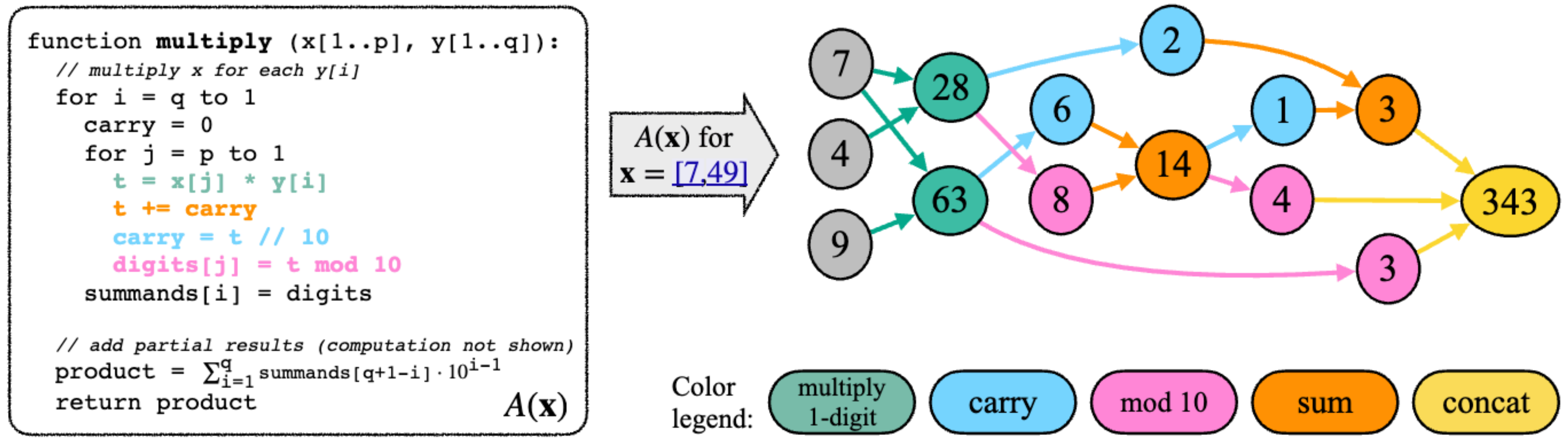
The "Unreasonable" Success of LLMs

- Reasoning needs symbolic structures.
  - Each step is deterministic & programmatic
  - Each step is subject to logical rules
- LLMs are just trained still with <u>finite data</u> using <u>statistical pattern matching</u> objective:

$$\mathbb{E}_x \left[\sum_t \log p(x_t \mid x_1, \cdots, x_{t-1}) \right]$$



vs

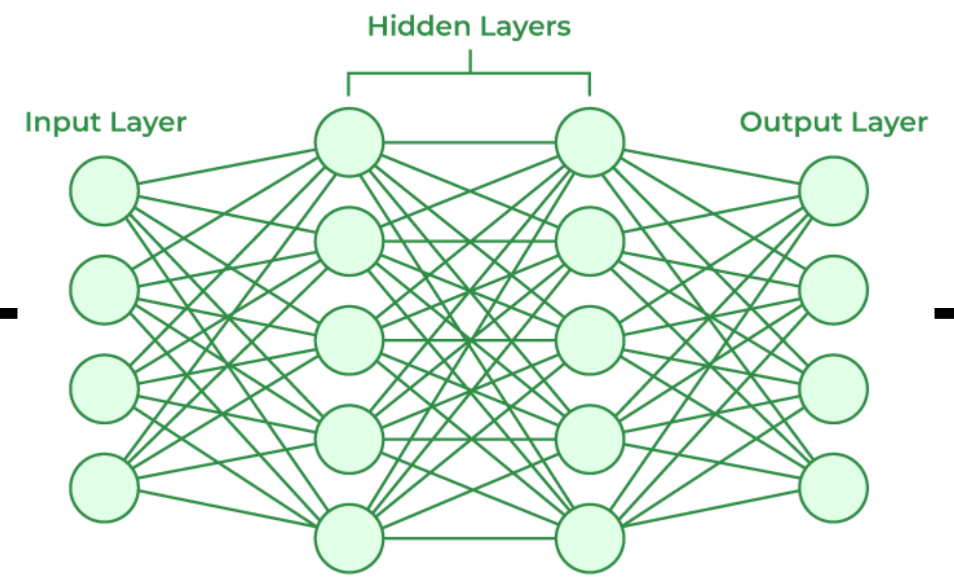


The “Unreasonable” Success of LLMs

- Reasoning needs symbolic structures.
  - Each step is deterministic & programmatic
  - Each step is subject to logical rules
- LLMs are just trained still with finite d… using statistical pattern matching …

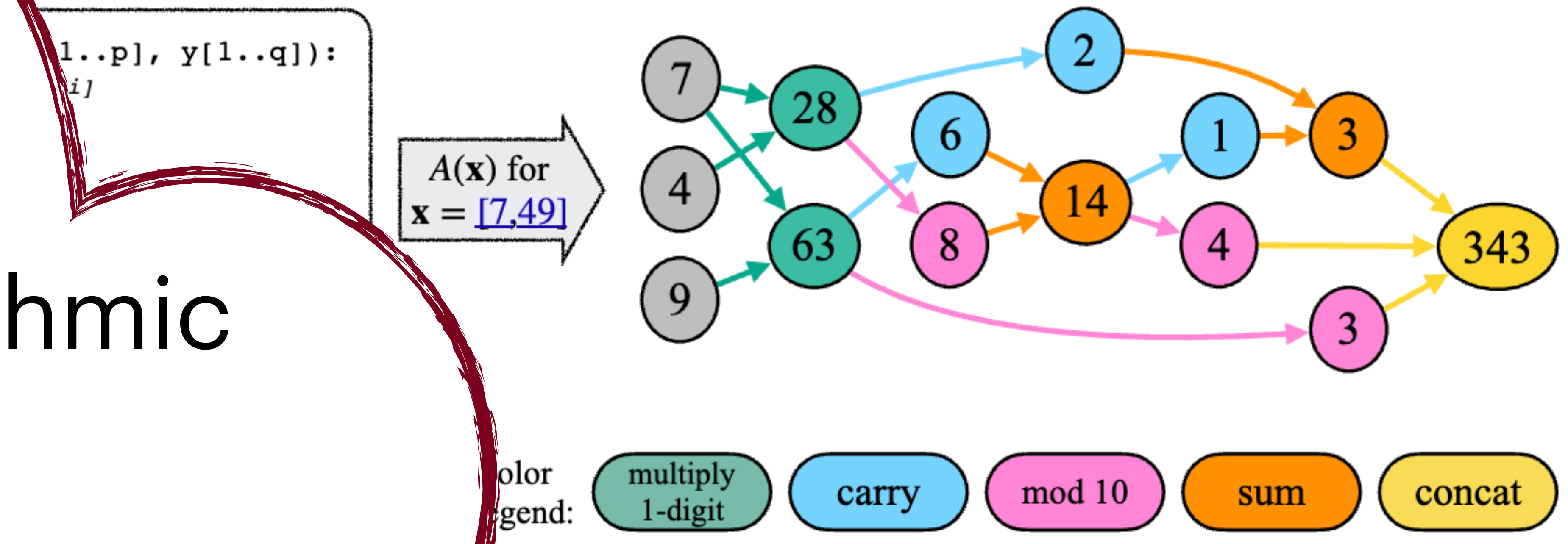
$$\mathbb{E}_x \left[\sum_t \log p(x_t | \right.$$

7 x 49 —



→ 343

vs

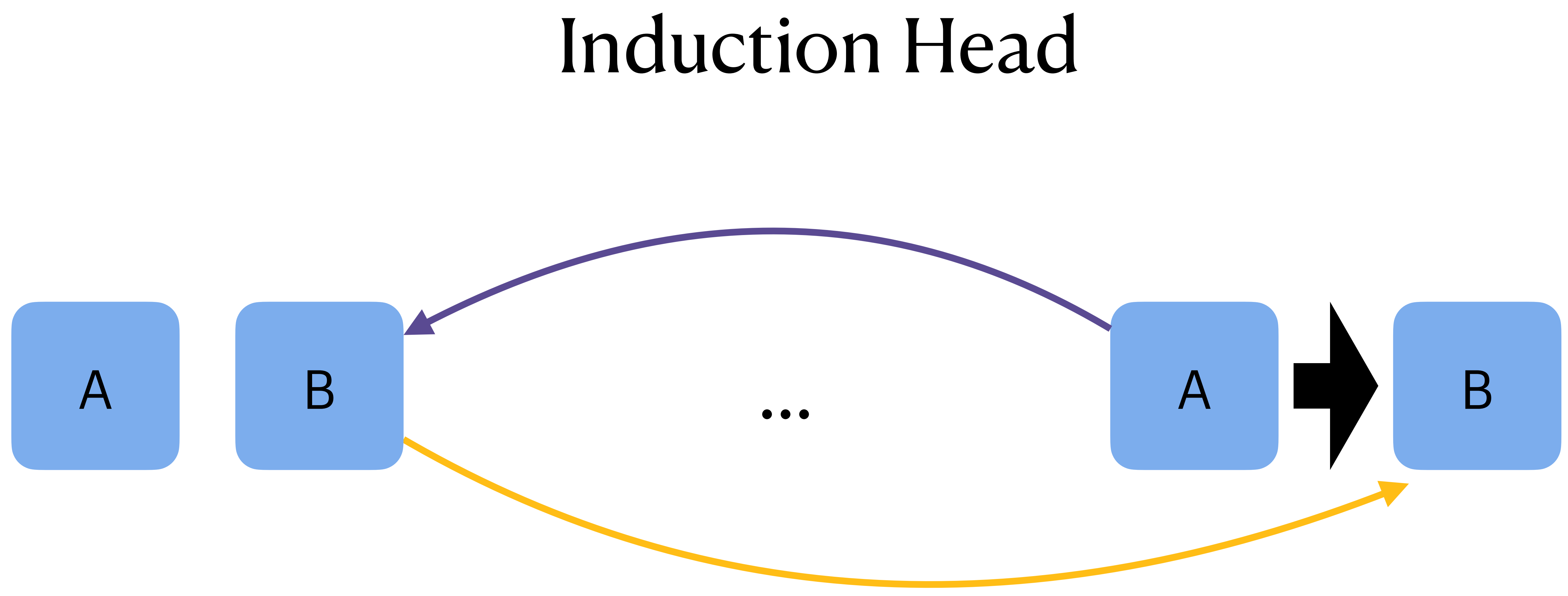


It looks like neural networks learn to truly perform logical & algorithmic reasoning?

Induction Head

Search previous example of **[A]** in the context:

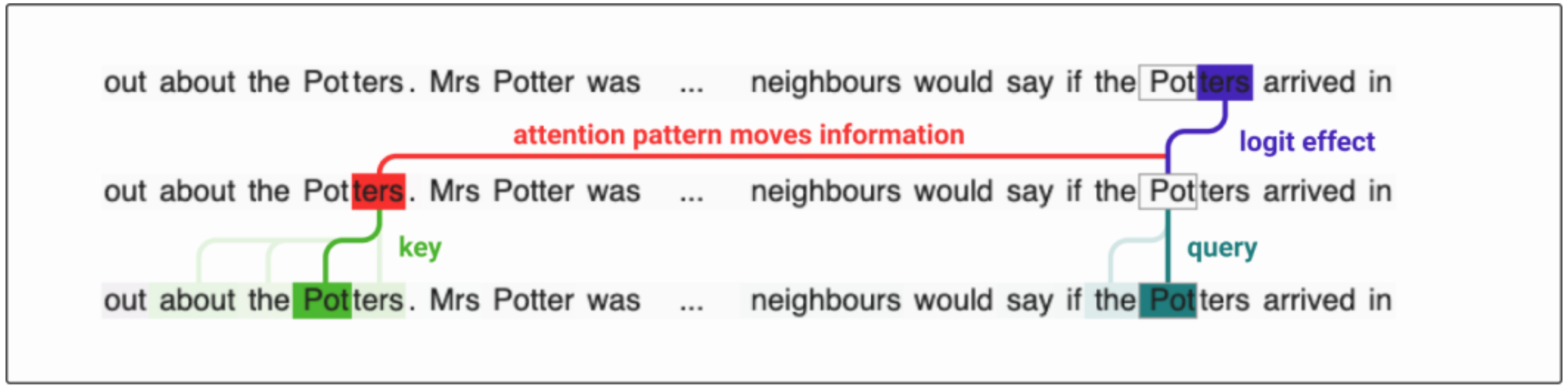
- If not found:
  - Attend to the **[START]** token
- If found:
  - Look at the next token **[B]** in previous case
  - Copy **[B]** to predict the next token



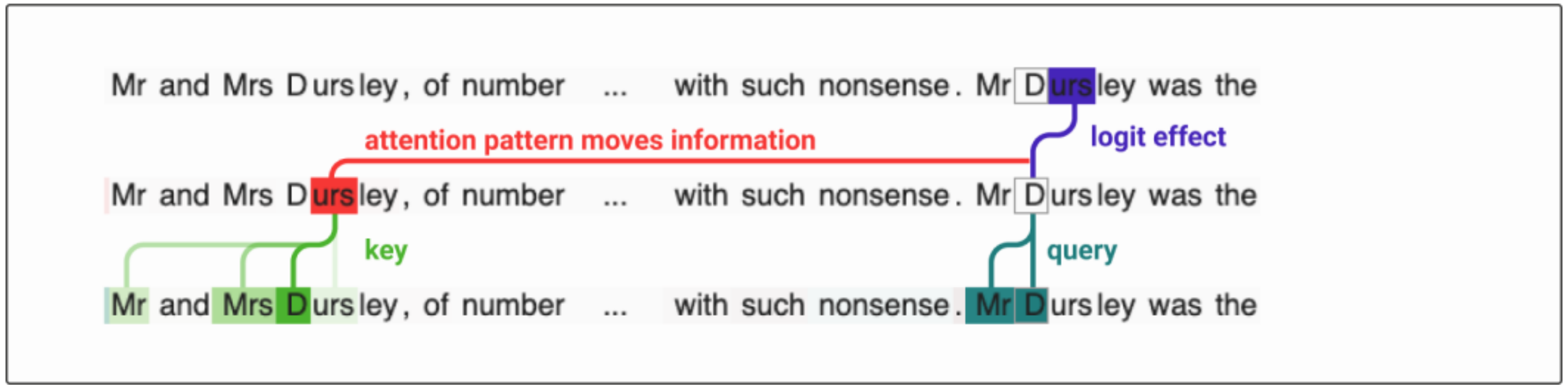
Circuits in Two-Layer Transformers

- How induction head is implemented in a two-layer transformer:s

Layer 2

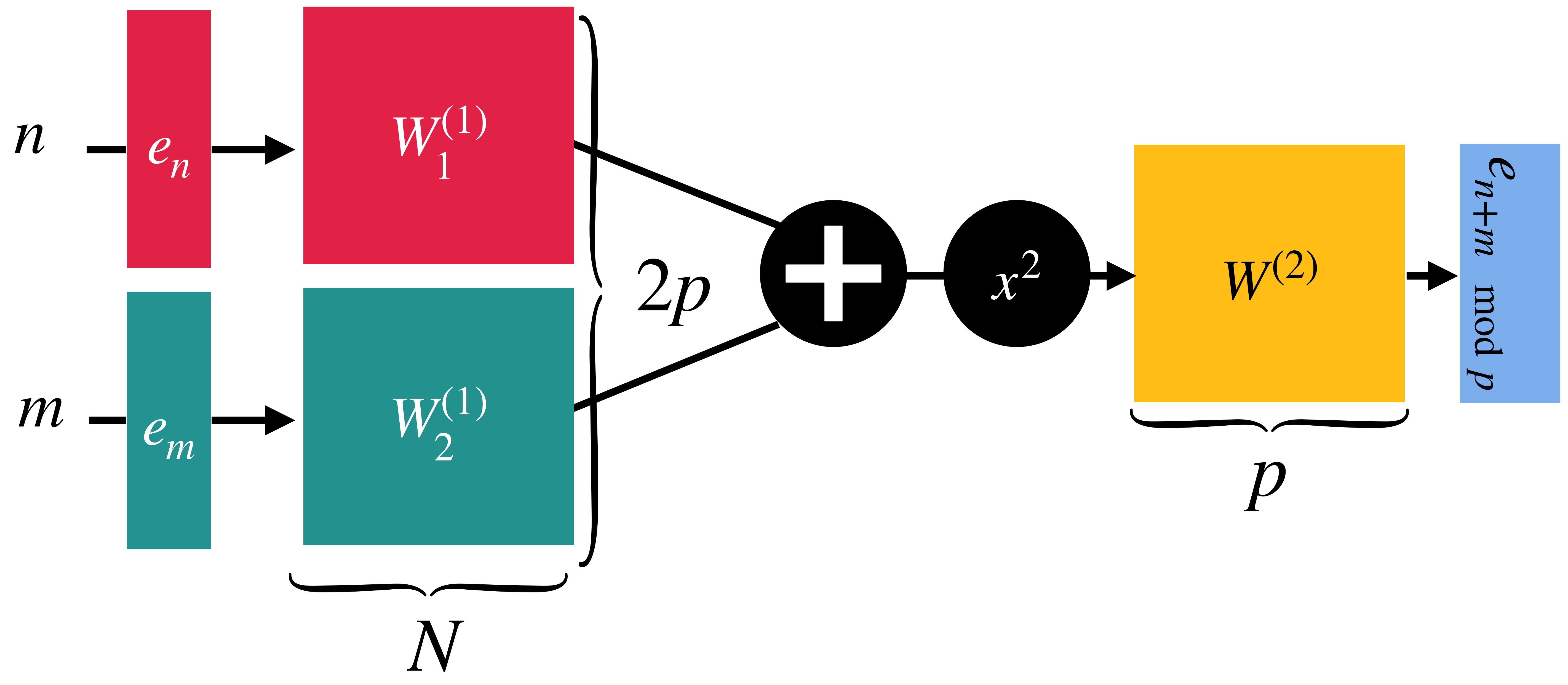


Layer 1



Simplistic Reasoning Task

- **Can neural network learn to perform arithmetics?**
- The input are two integers $n, m \in [N]$, we train a neural network that predicts $(n + m) \mod N$.



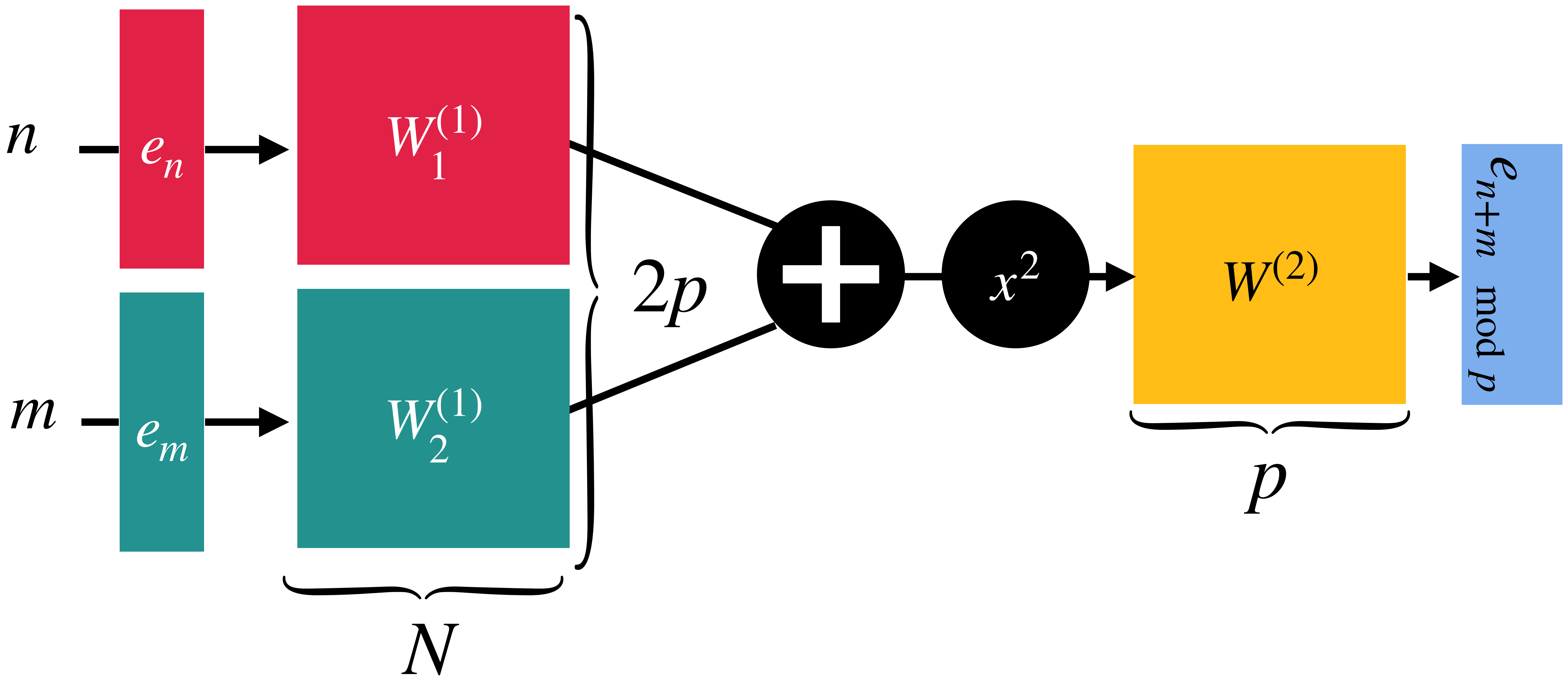
Gromov, Grokking modular arithmetic

Circuits that perform modular addition

- There exists an analytical solution that achieves 100% accuracy.

$$W^{(1)}_{1,kn} = \cos\left(2\pi\frac{k}{p}n + \varphi^{(1)}_k\right) \qquad W^{(1)}_{2,kn} = \cos\left(2\pi\frac{k}{p}n + \varphi^{(2)}_k\right) \qquad W^{(2)}_{qk} = \cos\left(-2\pi\frac{k}{p}q - \varphi^{(3)}_k\right)$$

- where $k \in [N]$, $n \in [0,p-1]$, $q \in [p]$, and $\varphi^{(1)}_k, \varphi^{(2)}_k, \varphi^{(3)}_k$ are random sampled from a uniform distribution.



Circuits that perform modular addition

- Let's verify step by step :)
- First layer pre-activation:

$$h_k^{(1)}(n,m) = \cos\left(2\pi\frac{k}{p}n + \varphi_k^{(1)}\right) + \cos\left(2\pi\frac{k}{p}m + \varphi_k^{(2)}\right)$$

- First layer after activation:

$$z_k^{(1)}(n,m) = \left(\cos\left(2\pi\frac{k}{p}n + \varphi_k^{(1)}\right) + \cos\left(2\pi\frac{k}{p}m + \varphi_k^{(2)}\right)\right)^2$$

Circuits that perform modular addition

- Second layer outputs:

$$h_q^{(2)}(n,m) = \frac{1}{4}\sum_{k=1}^{N} \cos\left(2\pi\frac{k}{p}(2n-q) + 2\varphi_k^{(1)} - \varphi_k^{(3)}\right) + \cos\left(2\pi\frac{k}{p}(2n+q) + 2\varphi_k^{(1)} + \varphi_k^{(3)}\right)$$

$$+\frac{1}{4}\sum_{k=1}^{N} \cos\left(2\pi\frac{k}{p}(2m-q) + 2\varphi_k^{(2)} - \varphi_k^{(3)}\right) + \cos\left(2\pi\frac{k}{p}(2m+q) + 2\varphi_k^{(2)} + \varphi_k^{(3)}\right)$$

$$+\frac{1}{2}\sum_{k=1}^{N} \cos\left(2\pi\frac{k}{p}(n+m-q) + \varphi_k^{(1)} + \varphi_k^{(2)} - \varphi_k^{(3)}\right)$$

$$+\frac{1}{2}\sum_{k=1}^{N} \cos\left(2\pi\frac{k}{p}(n+m+q) + \varphi_k^{(1)} + \varphi_k^{(2)} + \varphi_k^{(3)}\right)$$

$$+\frac{1}{2}\sum_{k=1}^{N} \cos\left(2\pi\frac{k}{p}(n-m-q) + \varphi_k^{(1)} - \varphi_k^{(2)} - \varphi_k^{(3)}\right)$$

$$+\frac{1}{2}\sum_{k=1}^{N} \cos\left(2\pi\frac{k}{p}(n-m+q) + \varphi_k^{(1)} - \varphi_k^{(2)} + \varphi_k^{(3)}\right)$$

$$+\sum_{k=1}^{N} \cos\left(2\pi\frac{k}{p}q + \varphi_k^{(3)}\right).$$

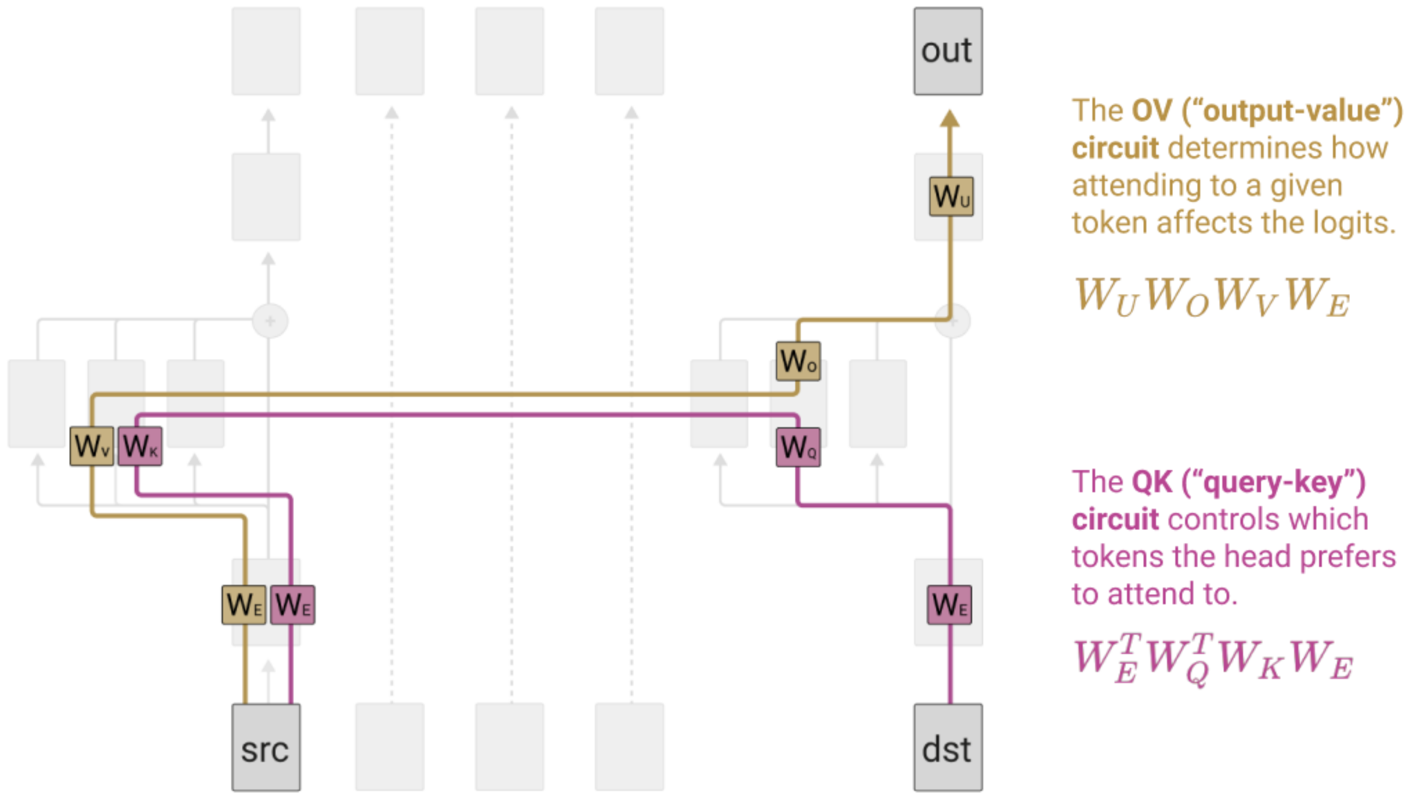
- Let $\varphi_k^{(1)} + \varphi_k^{(2)} = \varphi_k^{(3)}$
- This term becomes:

$$\frac{1}{2}\sum_{k=1}^{N} \cos\left(2\pi\frac{k}{p}(n+m-q)\right) = \frac{N}{2}\delta(n+m-q)$$

  - It equals to 1 only when $n+m-q=0 \mod p$
- Other terms diminish $\ll N$

More recent observations …

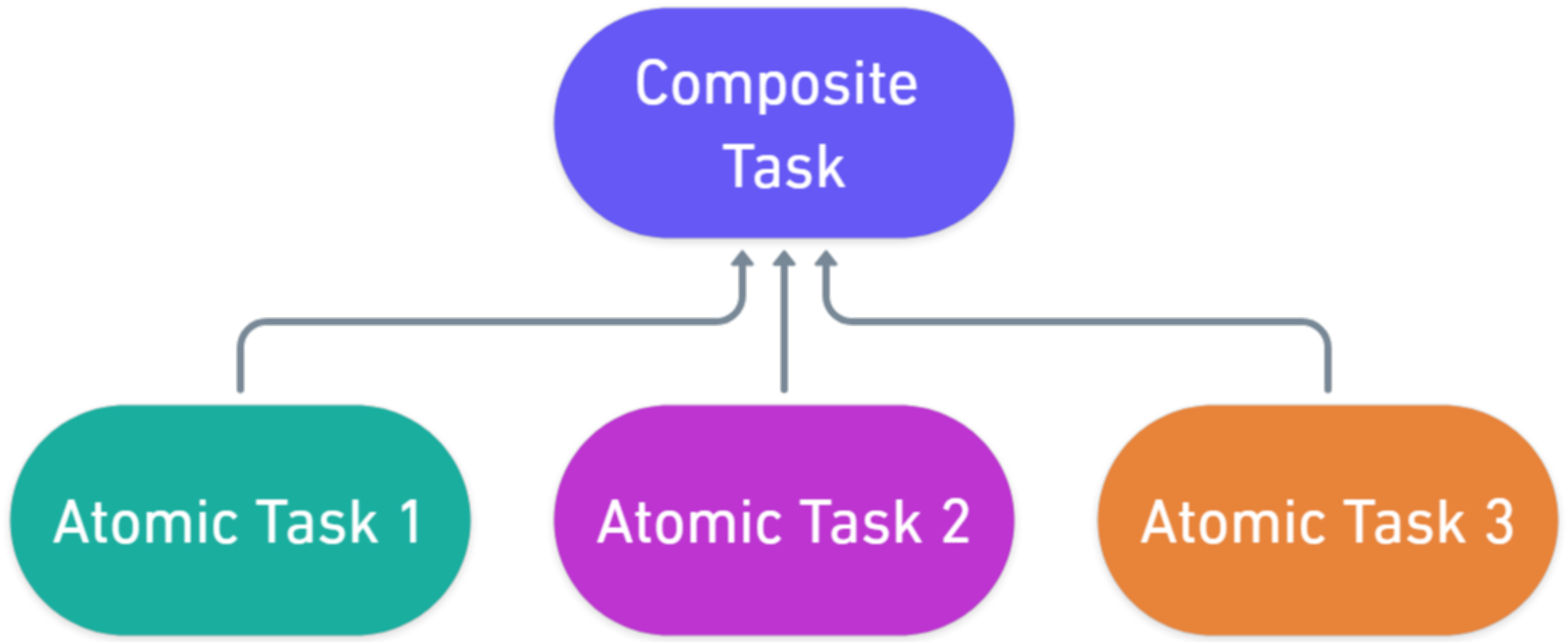
Expressivity



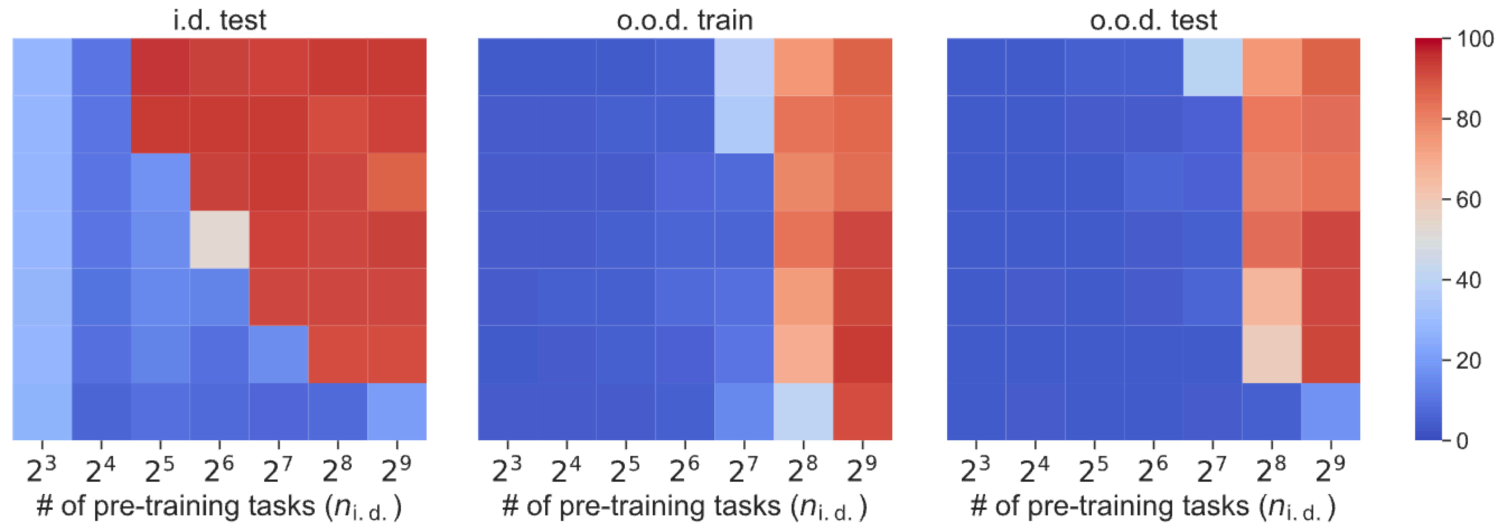
There exists weight configurations (i.e., circuits) that can represent exact algorithmic task

Compositionality

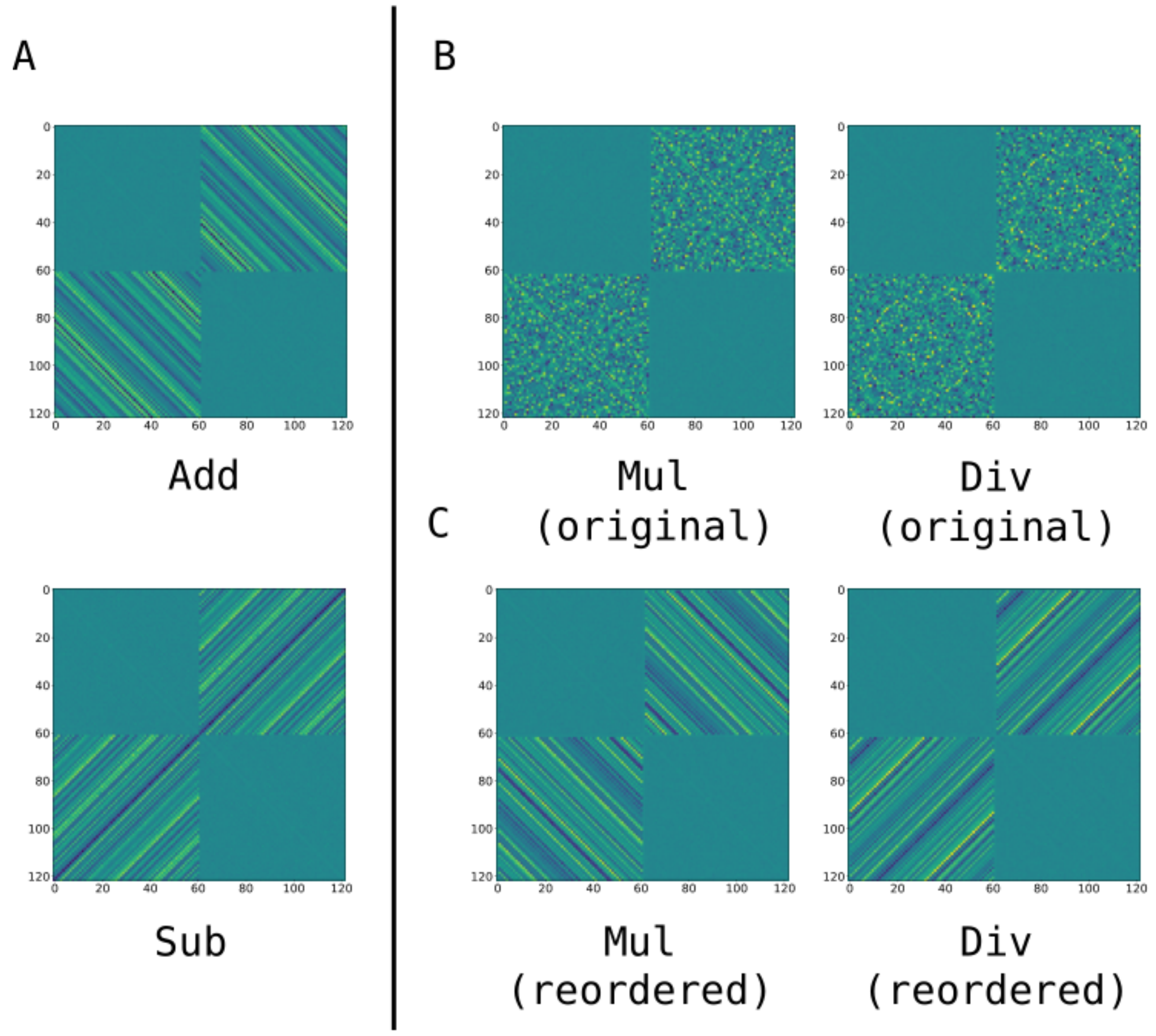
Training models over different tasks can emerge generalization on compositional tasks



Interpretability



Induction head



Modular addition

Visualized feature maps match constructed patterns in weight space for exact computation.

He et al., Learning to grok: Emergence of in-context learning and skill composition in modular arithmetic tasks

Stereotypical Dichotomy

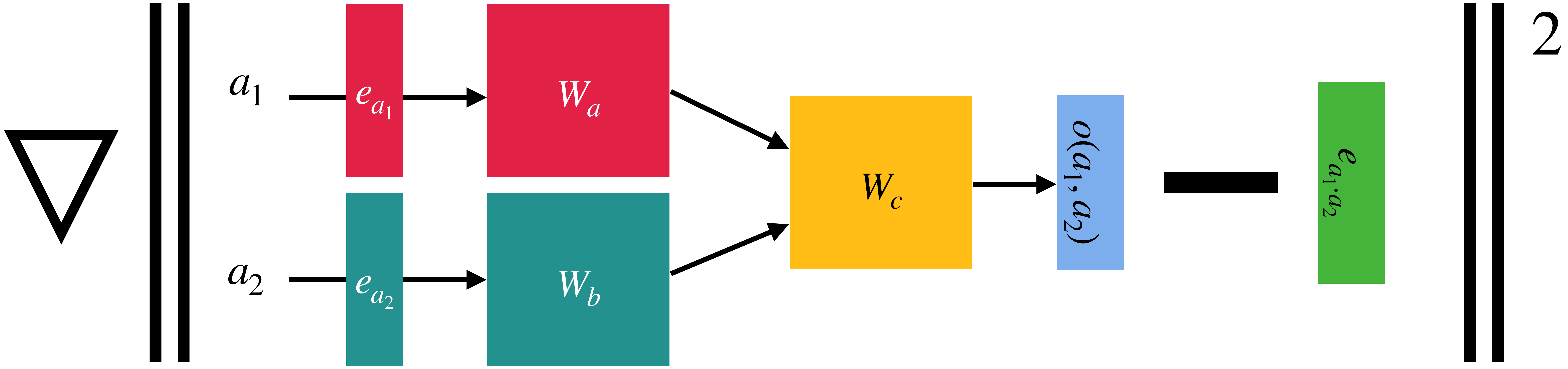
	Symbolism	Connectionism (In the past) 😬	Connectionism (But now) 🤯
• **Rule-based**: Reliable reasoning through programmatic steps.	✅	❌	✅
• **Compositionality**: Train from partial solutions, and compose freely to form generic solutions.	✅	❌	✅
• **Trainability**: Fast and stable convergence	❌	✅	✅

Open Questions to Answer

- **What are "symbols" represented within neural networks?**
  - Are there explicit/implicit symbolic-like structures in neural networks?
- **If so, can gradient descent discover symbolic structures?**
  - When and how gradient descent performs regression over these "symbols"?
- **Furthermore, how does symbolic structures reshape the weight space?**
  - Abstraction <=> Compression
  - Symbolism <=> Low-dimensionalism

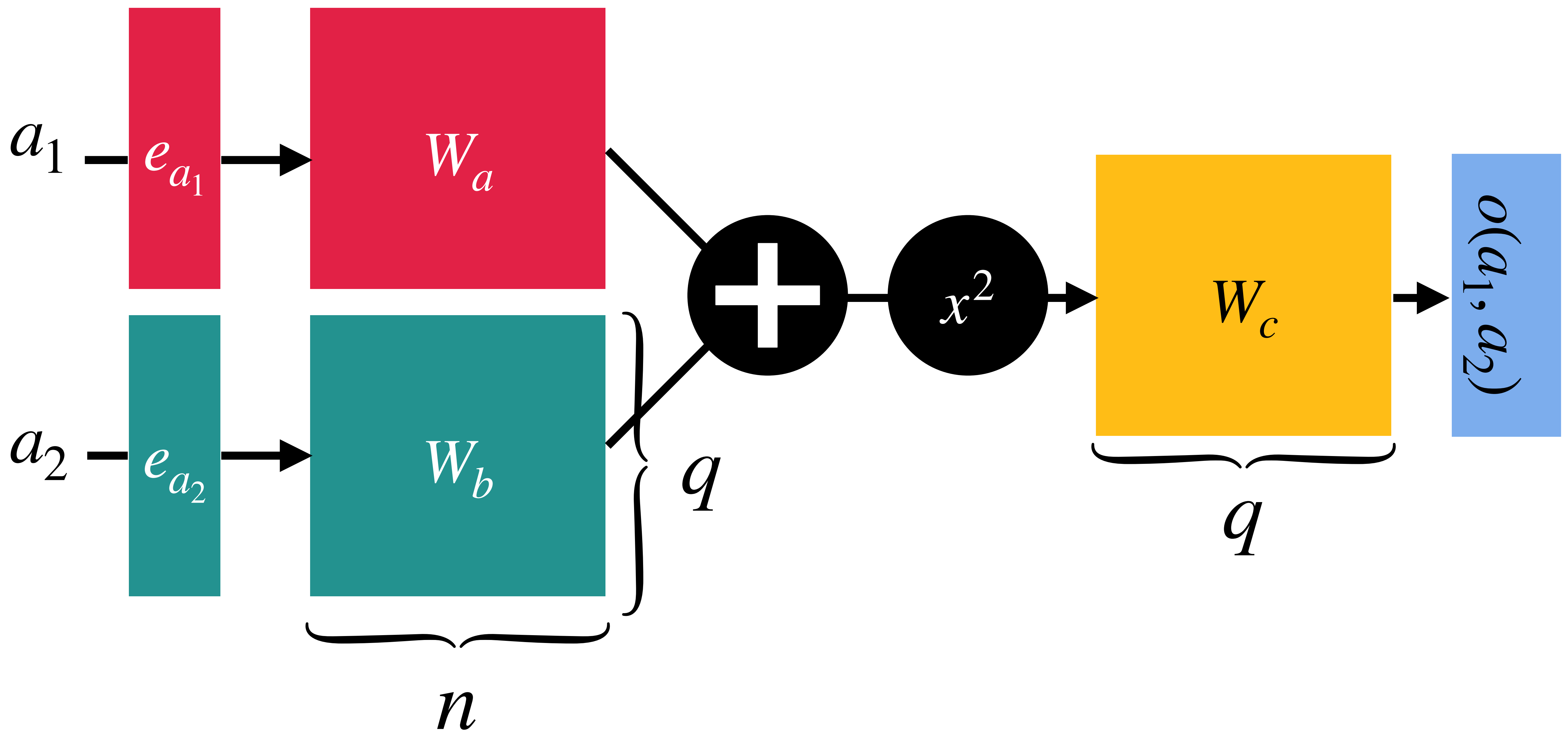
Learning to Perform Addition

- Given a finite Abelian group $(A, \cdot)$ with commutative group action "$\cdot$"
  - Suppose $A = \{a_1, \cdots, a_n\}$ has cardinality $n = |A|$.
- 🎯 Goal: Training a two-layer neural network that takes inputs $a_1, a_2 \in A$ and outputs $a_1 \cdot a_2$ with gradient descent.



Neural Architecture

- **Neural Architecture**
  - One-hot embeddings to encode group elements: $a_i \mapsto e_i$
  - Two layers and weight matrices: W_a, W_b, W_c with q hidden neurons.
  - Quadratic activation: $\sigma(x) = x^2$



$$o(a_1, a_2) = \frac{1}{q} \sum_{j=1}^{q} w_{cj} \sigma \left(w_{aj}^\top e_{a_1} + w_{bj}^\top e_{a_2} \right)$$

Loss Formulation

- We concatenate each row of weight matrices together as

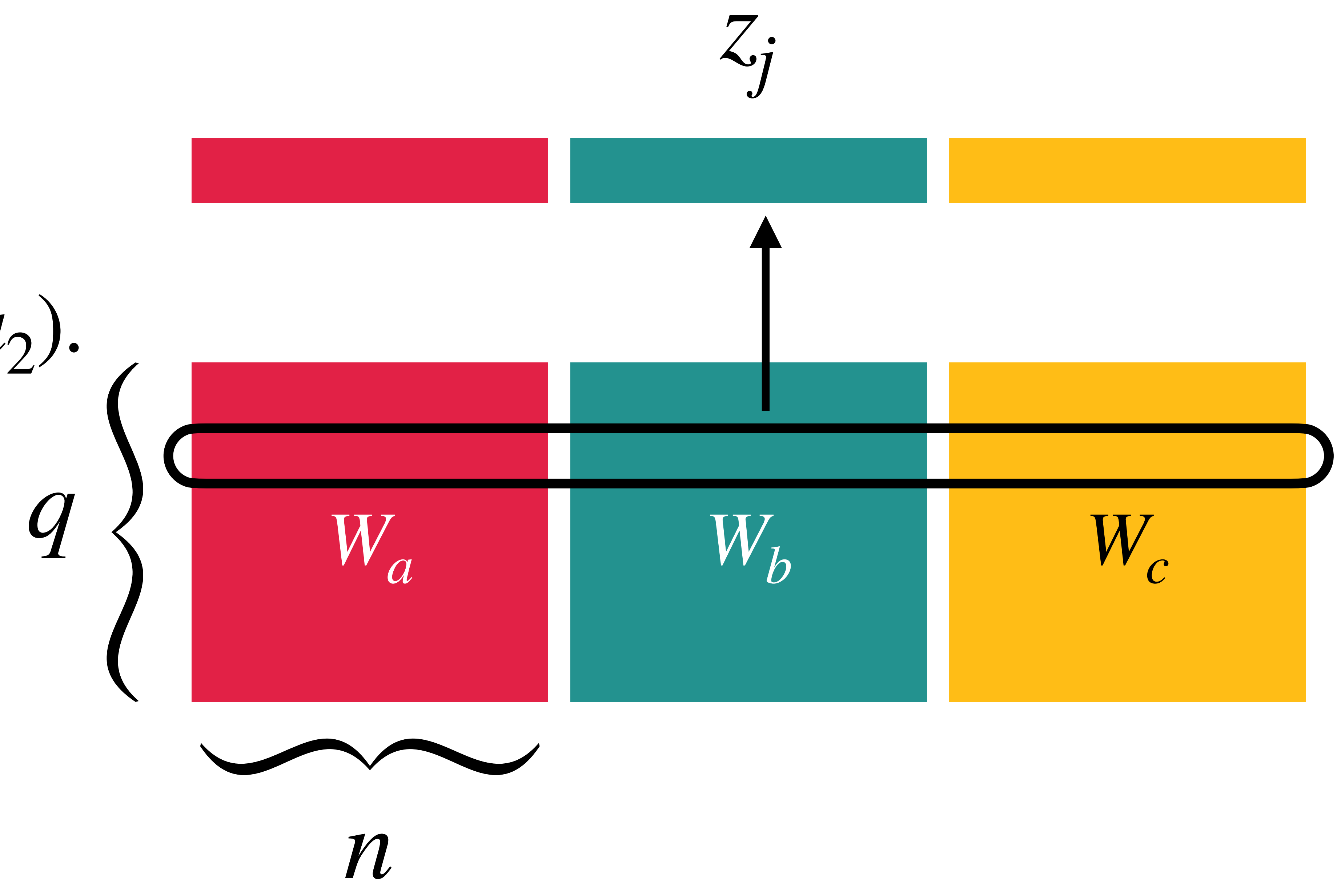
$$z_j \propto [W_{a,:,j}^\top, W_{b,:,j}^\top, W_{c,:,j}^\top]^\top \text{ for } j \in [q]$$

- We assume infinitely wide neural networks $q \to \infty$.
- **Training Objective**: Mean squared loss over all pairs of (a_1, a_2).

$$H = \sum_{a_1, a_2 \in A} \left\| P^\perp \left(\frac{1}{2n} o(a_1, a_2) - e_{a_1 \cdot a_2} \right) \right\|^2$$

- $P^\perp = I - \frac{1}{n} 11^\top$ is the centering matrix.
- **Optimization.** Gradient descent or gradient flow

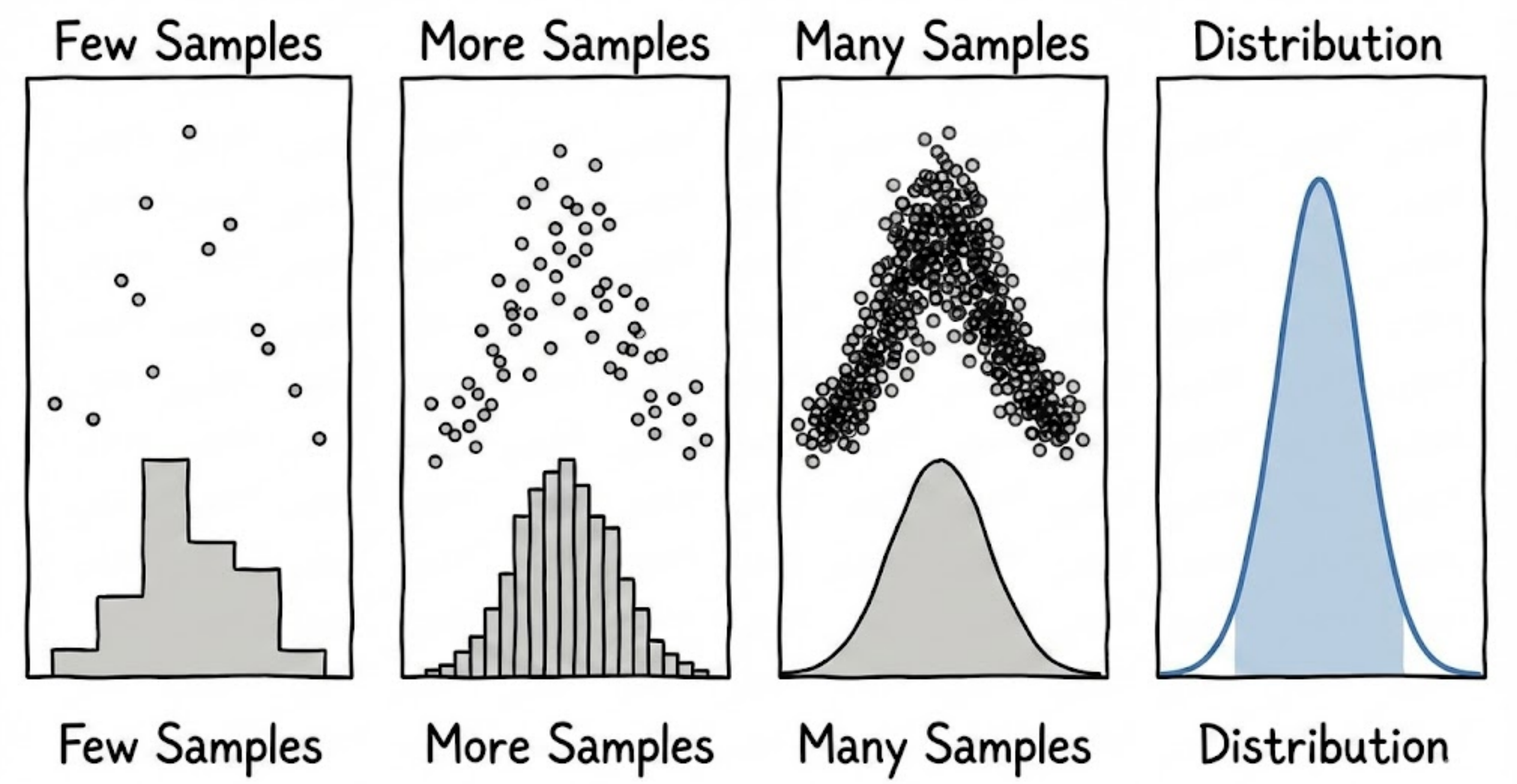
$$\frac{dz_j}{dt} = -\nabla_{z_j} H.$$



From Infinite-Width Neural Nets to Distribution

- When $q \to \infty$, we show that the neural networks can be represented with a distribution μ

$$\{z_j\}_{q\in[n]} \xrightarrow{q \to \infty} \mu(z) \qquad H(\{z_j\}_{q\in[n]}) \xrightarrow{q \to \infty} H[\mu]$$



Illustrated by NanoBanana

Monomial Potential

- Our results show that the loss $H[\mu]$ over neuron population can be written as:

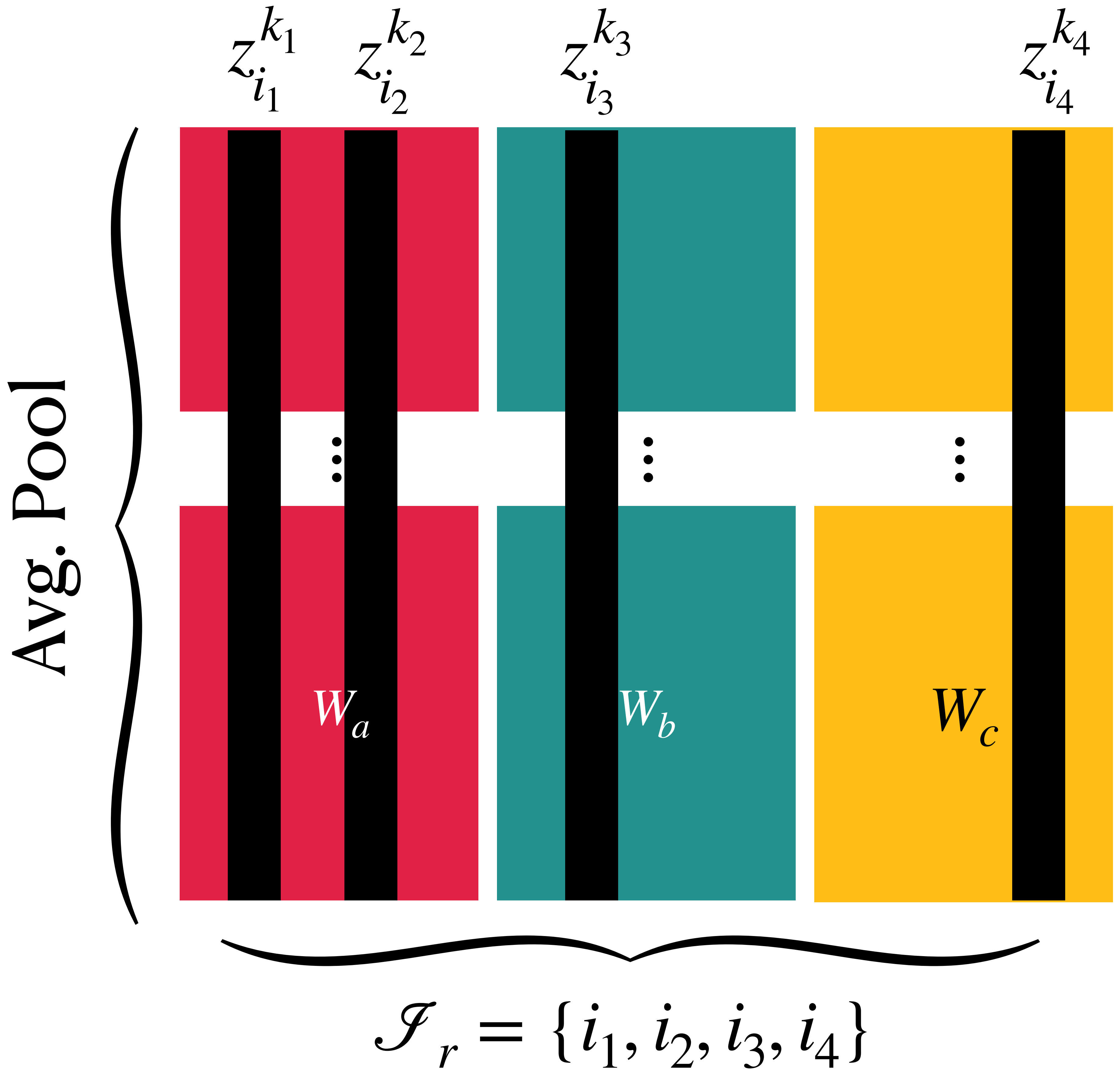
$$H[\mu] = L(\rho_{r_1}(\mu), \cdots \rho_{r_m}(\mu))$$

for some function $L : \mathbb{R}^m \to \mathbb{R}$.

- And $\rho_r[\mu]$ is defined as the monomial potential

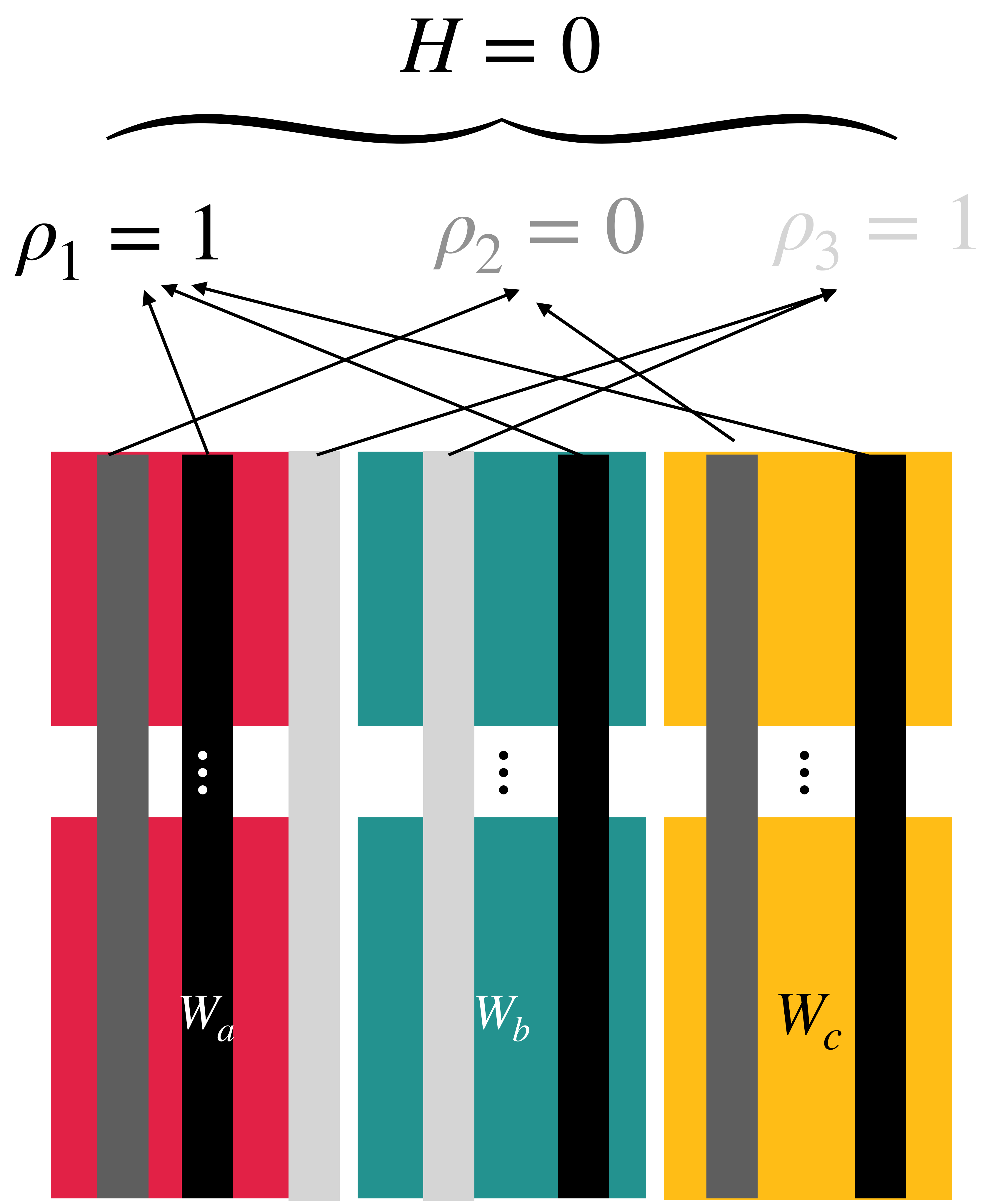
$$\rho_r(\mu) = \mathbb{E}_{z\sim\mu}[r(z)] = \int r(z) d\mu(z)$$

w.r.t. monomial $r(z) = \prod_{i \in \mathcal{I}_r} z_i^{k_i}$ where $\mathcal{I}_r$ is an index set.



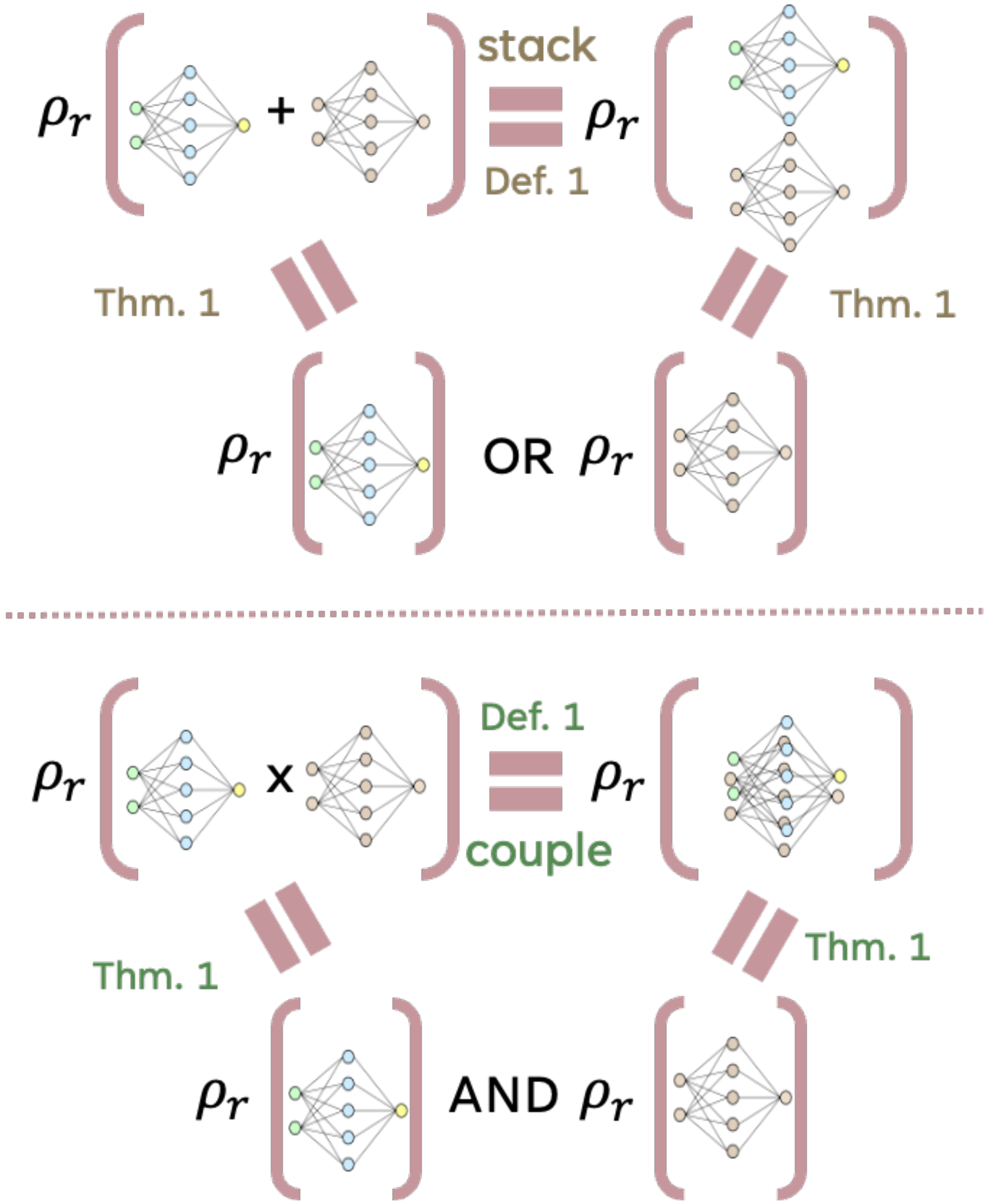
Monomial Potentials are Symbols

- **Symbols.** There exists a binary assignment of $\rho_1, \cdots, \rho_m$ such that the loss equals to zero: $H[\mu] = 0$.
  - Exact computation and perfect generalization.
  - $\rho_1, \cdots, \rho_m$ being binary resembles boolean variables in symbolic reasoning.



Compositional Structure of Monomial Potentials

- **Compositionality.** Neural networks are compositional in MP space.
  - Neural space Algebra
    - ➕Addition: Stacking two neural networks
    - ✖️Multiplication: (Kronecker/Hadarmard) product of weight matrices of two neural networks.
  - Neuron space operation ↔️ logical expression.
    - ➕Addition between neural nets ↔️ "OR" between MPs: $\rho_r(\mu_1 + \mu_2) = \rho_r(\mu_1) + \rho_r(\mu_2)$
    - ✖️Multiplication between neural nets ↔️ "AND" between MPs: $\rho_r(\mu_1 * \mu_2) = \rho_r(\mu_1) * \rho_r(\mu_2)$



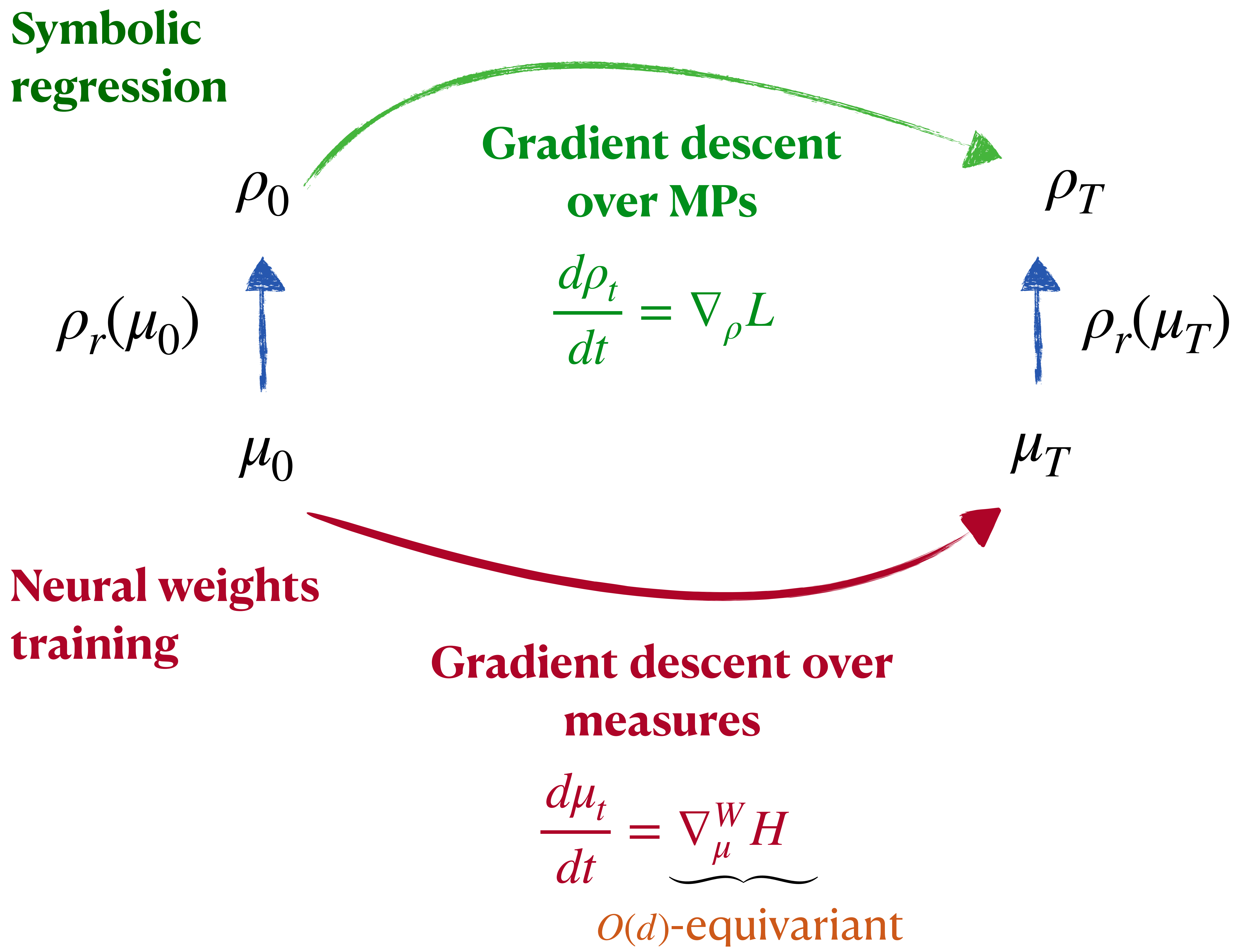
Takeaway I
Symbolic Structures are Hidden in Neural Weights

Monomial potentials are machine symbols, inheriting key properties that allows for exact computation and generalization.

1. 🔠 **Symbolic Variables.** Monomial potentials encapsulate neural weights as symbolic variables.
2. 🧮 **Logical Connectives.** Loss function can be re-written as expressions over monomial potentials.
3. 🧩 **Compositionality.** Weight space algebra manifests as composing MP-representing symbols via AND/OR logics.
4. ⁉️ Machine's symbol are not necessarily human-interpretable symbols?

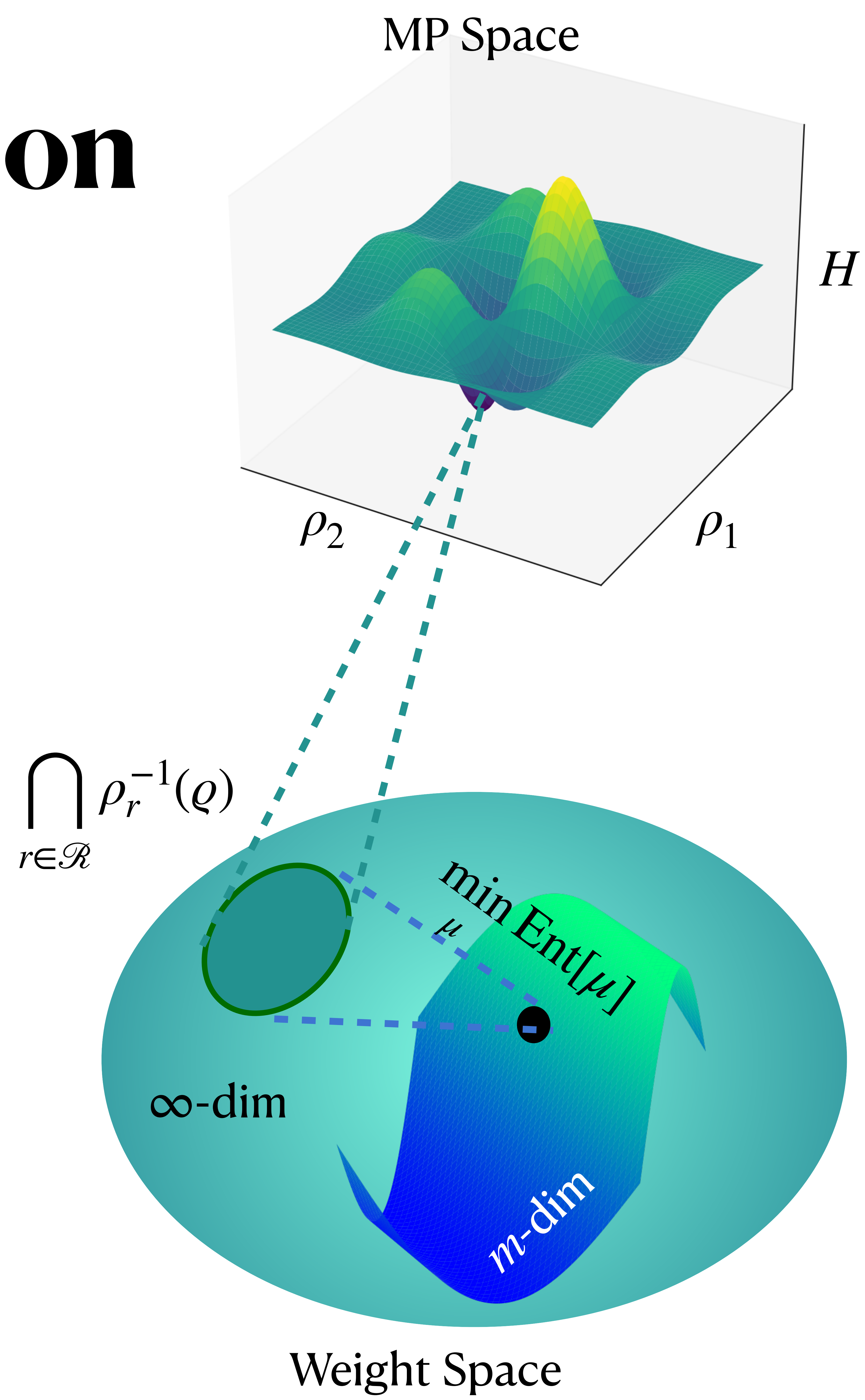
Gradient Descent => Symbolic Regression

- Gradient descent becomes regression on MP space.
- The learning process obeys a unique symmetry regarding orthogonal group.
  - When the neurons are rotated by R s.t. $RR^\top = I$, then its gradients are rotated by the same R.
- It forces MPs to converge to 0/1 solutions.



Dimension Reduction

- The weight space will go through a dimension reduction process.
  - The entropy-minimizing measure satisfying boolean MP assignments form a Riemannian manifold of dimension at most m - the number of MPs.
  - Evidence for weight space regularizations (e.g., weight decay)
- The number of involving MPs governs the intrinsic dimension even when the hidden dimension is going to infinity.
  - Evidence for low-rank weights (e.g. LoRA)



Takeaway II
GD Finds Symbolic Solutions under Geometric Constraints

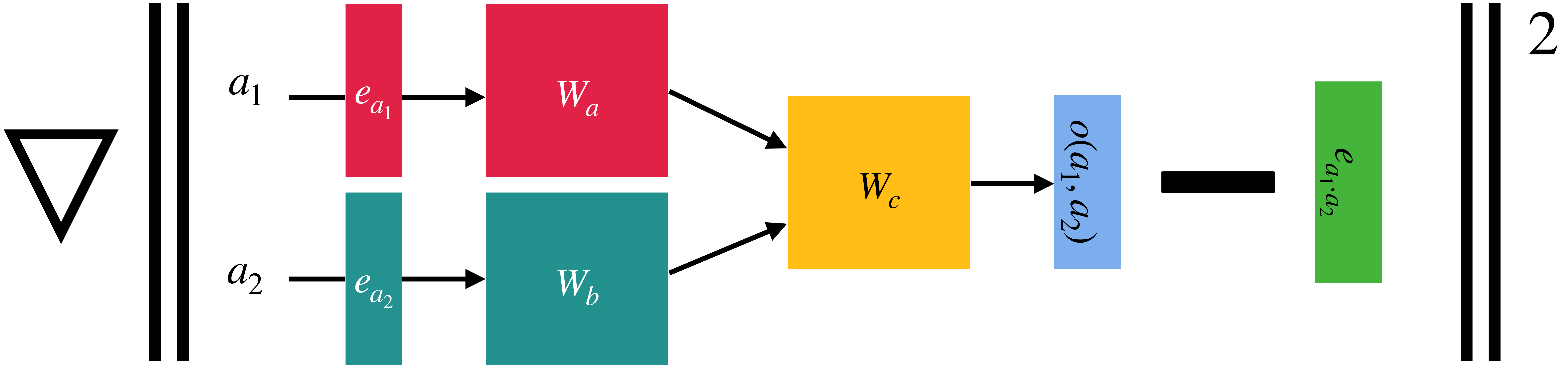
Gradient descent based training manifests as low-dimensional symbolic regression on MPs.

1. 🎯 Neural network training can reveal a symbolic learning process at the MP level.
2. 📉 The neural weight space will go through a dimension reduction process.
3. 📐 Geometric constraints are essential for discovering symbolic structures.

Formal Results

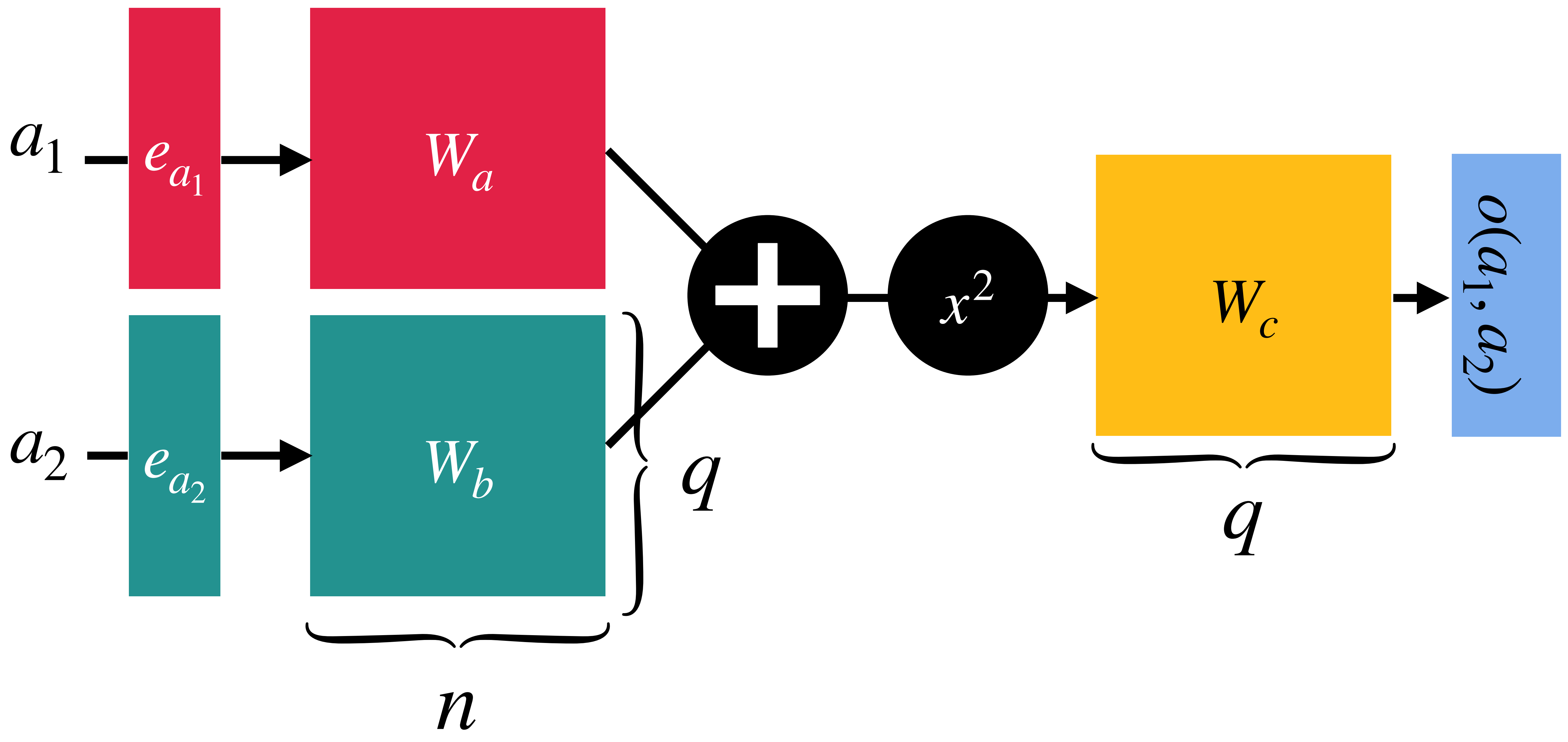
Learning to Perform Modular Addition

- Given a finite Abelian group $(A, \cdot)$ with commutative group action "$\cdot$"
  - Suppose $A = \{a_1, \cdots, a_n\}$ has cardinality $n = |A|$.
- 🎯 Goal: Training a two-layer neural network that takes inputs $a_1, a_2 \in A$ and outputs $a_1 \cdot a_2$ with gradient descent.



Neural Architecture

- **Neural Architecture**
  - One-hot embeddings to encode group elements: $a_i \mapsto e_i$
  - Two layers and weight matrices: W_a, W_b, W_c with q hidden neurons.
  - Quadratic activation: $\sigma(x) = x^2$



$$o(a_1, a_2) = \frac{1}{q} \sum_{j=1}^{q} w_{cj} \sigma \left(w_{aj}^\top e_{a_1} + w_{bj}^\top e_{a_2} \right)$$

Loss Formulation

- Represent weights in the Fourier space (F_k is the k-th Fourier basis):

$$w_{aj} = \sum_{k\neq 0} z_{akj}F_k, \quad w_{bj} = \sum_{k\neq 0} z_{bkj}F_k, \quad w_{cj} = \sum_{k\neq 0} z_{ckj}\overline{F_k}, \quad \forall j \in [q]$$

- Flatten coefficients for each neuron: $z_j = [\ldots, z_{akj}, z_{bkj}, z_{ckj}, \ldots]_{0\leq k<n} \in \mathbb{R}^{3n}$.
- **Training Objective**: Mean squared loss over all pairs of (a_1, a_2).

$$H(\{z_j\}_{j\in[q]}) = \sum_{a_1,a_2\in A} \left\| P^{\perp}\left(\frac{1}{2n}o(a_1, a_2) - e_{a_1\cdot a_2}\right) \right\|^2$$

- $P^{\perp} = I - \frac{1}{n}11^{\top}$ is the centering matrix.
- **Optimization.** Gradient descent or gradient flow $\frac{d\{z_j\}}{dt} = -\nabla H(\{z_j\}_{j\in[q]})$.

Loss Decomposition

Proposition. The loss function H can be reformulated as: $H = \frac{1}{n-1}\sum_{k\neq 0}\ell_k + \frac{n-1}{n}$,

$$\ell_k = -2\rho_{kkk} + \sum_{k_1,k_2}|\rho_{k_1k_2k}|^2 + \frac{1}{4}\left|\sum_{p\in\{a,b\}}\sum_{k'}\rho_{p,k',-k',k}\right|^2 + \frac{1}{4}\sum_{m\neq 0}\sum_{p\in\{a,b\}}\left|\sum_{k'}\rho_{p,k',m-k',k}\right|^2$$

$$\rho_{k_1k_2k} = \frac{1}{q}\sum_j z_{ak_1j}z_{bk_2j}z_{ckj}, \quad \rho_{pk_1k_2k} = \frac{1}{q}\sum_j z_{pk_1j}z_{pk_2j}z_{ckj}$$

- **Key Observations**
  1. H is expanded as a function solely dependent on the empirical **measure** $\mu^{(q)} = \sum_{j\in[q]}\delta_{z_j}$
  2. H depends on $\mu^{(q)}$ through averaging on a subset of **monomials**: $z \mapsto \prod_{i\in\mathcal{I}} z_i$ for some index set $\mathcal{I}$.

Formulating Reasoning: Beyond Group Addition

- Consider a parameter space $M \in \mathbb{R}^n$
  - $n = 3d$ in the Abelian group example.
- Analyze the limiting measure: $\mu^{(q)} \to \mu, \text{ when } q \to \infty$
- Generalize *average over monomials* to **Monomial Potentials (MPs).**

Definition. A ***monomial potential*** **(MP)** $\rho_r : P_*(M) \to \mathbb{R}$ is defined as the expectation of the specified monomial r against the input measure μ:

$$\rho_r(\mu) = \mathbb{E}_{z \sim \mu}[r(z)] = \int r(z) d\mu(z)$$

Formulating Reasoning: Beyond Group Addition

- Specify a set of monomials $\mathscr{R} = \{r_1, \cdots, r_m\}$ associated with the task.
- Generalize loss function $H[\{z_j\}]$ to loss functional over measure μ:

$$H[\mu] = L(\rho_{r_1}(\mu), \cdots \rho_{r_m}(\mu))$$

for some function $L : \mathbb{R}^m \to \mathbb{R}$.

- **Optimization over measures.**

$$\partial_t \mu_t = \nabla_z \cdot \left(\mu_t \nabla_z \left(\frac{\delta H}{\delta \mu}[\mu_t] \right) \right)$$

Intuition: $\mu_{t+\tau} \approx \text{argmin}_{\mu \in P(M)} \{ H(\mu) + \frac{1}{2\eta_t \tau} W_2(\mu_t, \mu) \}$.

Summary of Generalization

Motivating Example	Generalization
$\mu^{(q)} = \sum_{j \in [q]} \delta_{z_j}$	An arbitrary measure $\mu \in P_*(M)$
$\{\rho_{k_1 k_2 k}, \rho_{p k_1 k_2 k}\}$	MPs: $\rho_r(\mu) = \mathbb{E}_{z \sim \mu}[r(z)], r \in \mathscr{R}$
$H(\{z_j\}_{j \in [q]})$	$H[\mu] = L(\rho_{r_1}(\mu), \cdots \rho_{r_m}(\mu))$
$\frac{d\{z_j\}}{dt} = -\nabla H(\{z_j\})$	$\partial_t \mu_t = \nabla_z \cdot \left(\mu_t \nabla_z \left(\frac{\delta H}{\delta \mu}[\mu_t]\right)\right)$

Continuous Optimization as Boolean Satisfaction

- Revisiting: $H = \sum_{k\neq 0} \ell_k/(n-1) + (n-1)/n$.

$$\ell_k = -2\rho_{kkk} + \sum_{k_1,k_2} |\rho_{k_1k_2k}|^2 + \frac{1}{4}\left|\sum_{p\in\{a,b\}}\sum_{k'}\rho_{p,k',-k',k}\right|^2 + \frac{1}{4}\sum_{m\neq 0}\sum_{p\in\{a,b\}}\left|\sum_{k'}\rho_{p,k',m-k',k}\right|^2$$

- A minimizer can be identified:

$$\rho_{kkk} = \mathbb{I}(k \neq 0), \quad \rho_{k_1k_2k} = 0, \quad \rho_{pk_1k_2k} = 0, \quad \forall p \in \{a,b\}, k_1, k_2, k \in [d]$$

- **Key Observations**:
    - Modular addition can be solved by finding μ that satisfies a **binary** assignment at the level of MPs.
    - MPs plays a role similar to boolean variables and L resembles a logical expression.

Generalization Beyond Group Addition

Definition. Suppose a measure $\mu \in P_*(M)$ has 0-set $\mathscr{R}_0 \subset \mathscr{R}$ and 1-set $\mathscr{R}_1 \subset \mathscr{R}$, (or equivalently 0/1-set $(\mathscr{R}_0, \mathscr{R}_1)$), then $\rho_r(\mu) = 0$ for every $r \in \mathscr{R}_0$ and $\rho_r(\mu) = 1$ for every $r \in \mathscr{R}_1$.

- 0/1-sets test satisfiability of each MP for the measure μ.
- The solutions to Abelian group reasoning has 0-set $\mathscr{R}_c \cup \mathscr{R}_n \cup \mathscr{R}_*$ and 1-set $\mathscr{R}_g$:
  - $\mathscr{R}_c := \{r_{k_1k_2k} \,|\, k_1, k_2, k \text{ not all equal}\}$
  - $\mathscr{R}_n := \{r_{p,k',-k',k}\}$
  - $\mathscr{R}_* = \{r_{p,k',m-k',k} \,|\, m \neq 0\}$
  - $\mathscr{R}_g := \{r_{kkk} \,|\, k \neq 0\}$

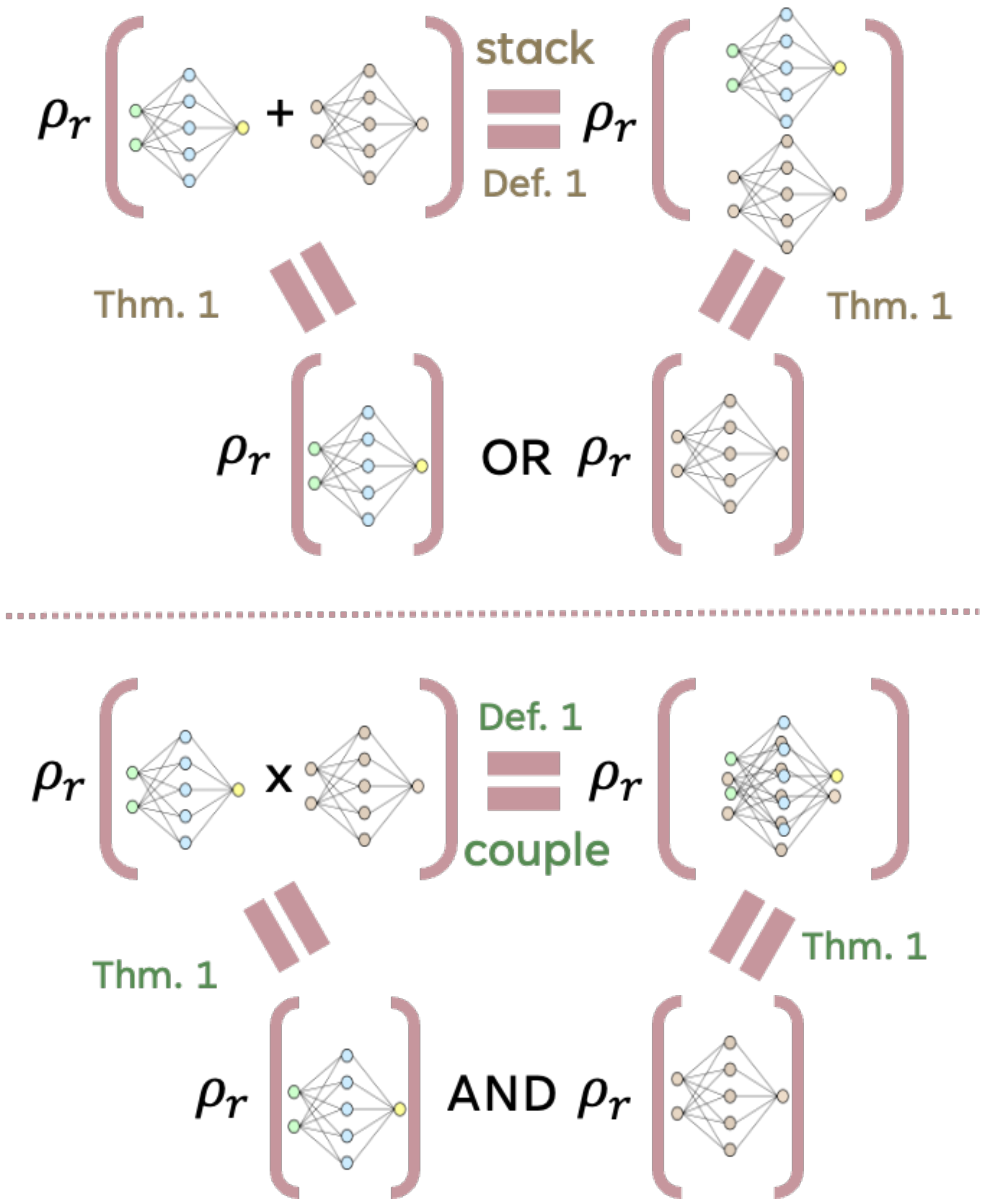
Symbolism over Statistical Measures

Definition 1. For two measures μ_1 and μ_2, define:

- (1) addition as: $\mu_+ = \mu_1 + \mu_2$ such that $\mu_+(A) = \mu_1(A) + \mu_2(A)$ for every measurable $A \subset M$;
- (2) multiplication as: $\mu_* = \mu_1 * \mu_2$ such that μ_* is the measure of $z_* = z_1 \odot z_2$ where $z_1 \sim \mu_1$, $z_2 \sim \mu_2$, $\odot$ denotes element-wise multiplication;
- (3) the identity element as $\delta_{\mathbf{1}_d}$, i.e., the point mass at the d-dimensional all-one vector;
- (4) the zero element as the zero measure.

Algebra of Measures and MPs

- **Theorem 1.** $\langle P_*(M), +, * \rangle$ is a commutative semi-ring. Every MP $\rho_r(\mu)$ is a ring homomorphism:
  - (1) $\rho_r(\mu_1 + \mu_2) = \rho_r(\mu_1) + \rho_r(\mu_2)$
  - (2) $\rho_r(\mu_1 * \mu_2) = \rho_r(\mu_1) * \rho_r(\mu_2)$
- Neuron space operation ↔️ logical expression.
  - ➕Addition between measures ↔️ "OR" between MPs.
  - ✖️Multiplication between measures ↔️ "AND" between MPs.



Compositionality of Neural Solutions

- Partial solutions can be composed to generate general solutions!
  1. Find special solutions satisfying subsets of constraints $\mathscr{R}_1, \cdots, \mathscr{R}_k$
  2. Use union/intersection to combine $\mathscr{R}_1, \cdots, \mathscr{R}_k$ to satisfy the target 0/1 sets.
  3. Construct global minimizers by mapping logical language to neural weights

Examples. If μ_1 has 0/1-sets $(\mathscr{R}_0, \mathscr{R}_1)$ and μ_2 has 0/1-sets $(\mathscr{S}_0, \mathscr{S}_1)$, then:

1. $\mu_1 * \mu_2$ has 0/1-sets $(\mathscr{R}_0 \cup \mathscr{S}_0, \mathscr{R}_1 \cap \mathscr{S}_1)$;
2. $\mu_1 + \mu_2$ has 0/1-sets $(\mathscr{R}_0 \cap \mathscr{S}_0, (\mathscr{R}_1 \cap \mathscr{S}_0) \cup (\mathscr{R}_0 \cap \mathscr{S}_1))$;
3. If μ_1 is a global optimizer and μ_2 has 1-set $\mathscr{R}$ (the entire set of MPs), then $\mu_1 * \mu_2$ is a global optimizer.

However, ...

Neural Network Training

Finding μ^* minimizing the population risk:

$$\mu^* = \arg\min \mathbb{E}_{a_1,a_2} \ell\left(o(a_1, a_2), e_{a_1 \cdot a_2}\right)$$

Symbolic Regression

Finding binary assignment of MPs:

$$\rho_{kkk} = \mathbb{I}(k \neq 0), \quad \rho_{k_1k_2k} = 0$$

$$\rho_{pk_1k_2k} = 0, \quad \forall p \in \{a, b\}, k_1, k_2, k \in [d]$$

- The gradient-based training may still learn μ that achieves non-binary MPs, e.g.

$$\sum_{k\neq 0} \rho_{kkk} = 0 \text{ while } \rho_{kkk} \neq 0 \text{ for every } k \neq 0$$

❓ Open Questions

Can neural network training discover “symbolic” solution?

When “GD on μ” = “GD on MPs”?

- **Theorem 2.** Consider a trajectory of measure $\{\mu_t\}_{t\geq 0}$ governed by Wasserstein gradient flow $\partial_t \mu_t = \nabla_z \cdot \left(\mu_t \nabla_z \left(\frac{\delta H}{\delta \mu}[\mu_t]\right)\right)$. Assume that:

  1. ✅ $\mu_0 = \mathcal{N}(0,I)$ at the initialization;
  2. ✅ $\deg r \geq 3$ and is odd for every $r \in \mathscr{R}$;
  3. ✅ $\nabla \frac{\delta H}{\delta \mu}[\mu_t]$ is $O(d)$-equivariant: $\nabla \frac{\delta H}{\delta \mu}[\mu_t](Rx) = R \nabla \frac{\delta H}{\delta \mu}[\mu_t](x)$ for every $R \in O(d)$.

- Then each monomial potential is optimized coordinate-wisely as:

$$\partial_t \rho_{r_i}(\mu_t) = -C_i(t) \partial_{\rho_{r_i}} L(\rho)$$

where $C_i(t) > 0$ is a time-dependent scalar function only dependent on ρ_{r_i}.

Neural Weight Training => Symbolic Regression

Symbolic regression

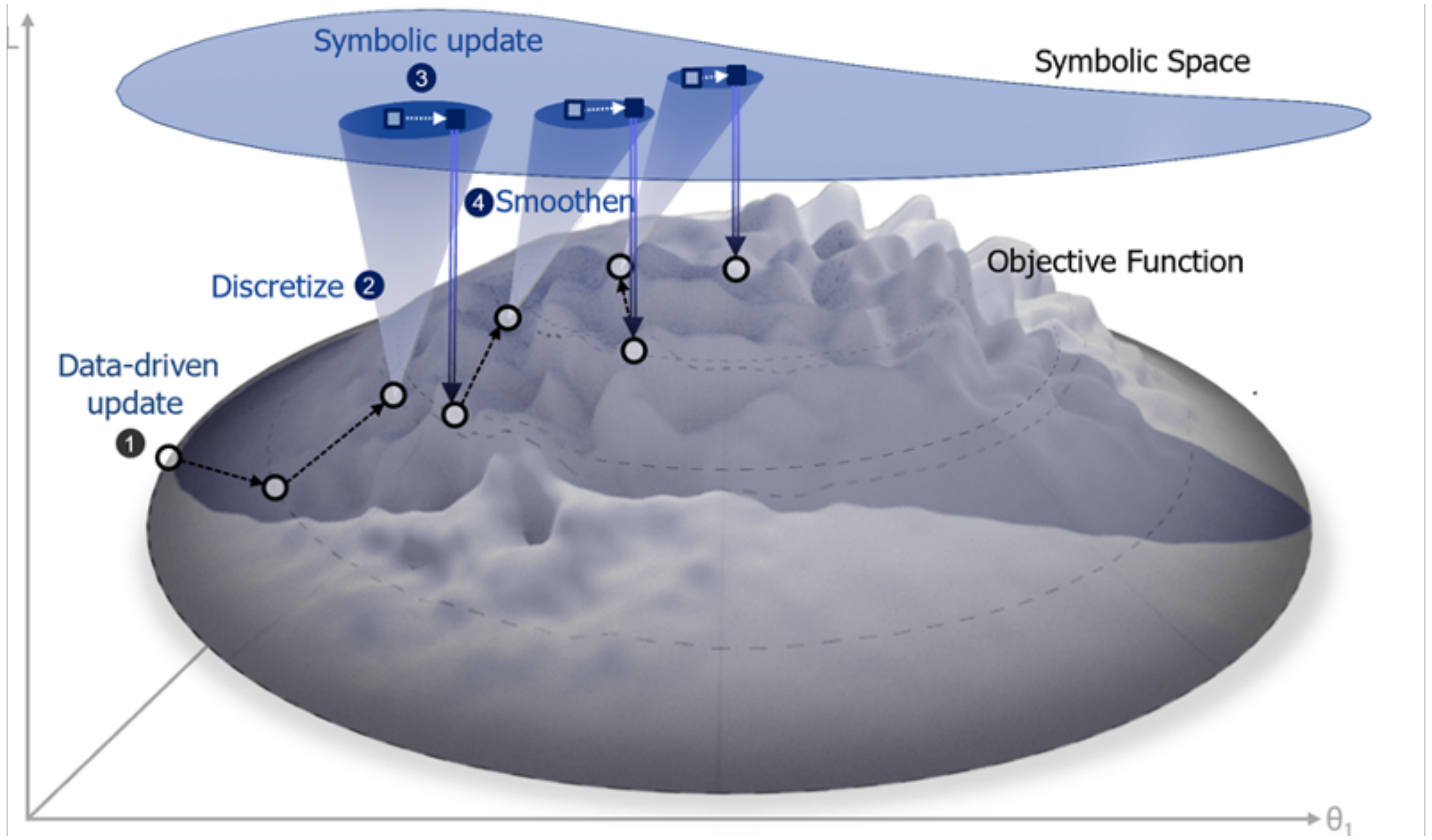
ρ_0 $\qquad \partial_t \rho_{r_i}(\mu_t) = -C_i(t)\partial_{\rho_{r_i}} L(\rho), \forall r$ $\qquad \rho_T$

$\rho_r(\mu_0)$ $\qquad\qquad \rho_r(\mu_T)$

μ_0 $\qquad\qquad \mu_T$

Neural weights training

$$\partial_t \mu_t + \nabla_z \cdot \left(\mu_t \underbrace{\nabla_z \left(\frac{\delta H}{\delta \mu}[\mu_t] \right)}_{O(d)\text{-equivariant}} \right) = 0$$



Under geometric constraints (i.e., $O(d)$-equivariant velocity field), optimizing the measure with WGF is equivalent to directly performing gradient descent on MPs.

Back to Modular Addition Example

- Consider the MP ρ_{kkk} for some $k \neq 0$, we can derive that

$$\partial_{\rho_{kkk}} L \propto \rho_{kkk} - 1.$$

- Then by the previous Theorem, we find that:

$$\frac{\partial}{\partial t}\rho_{kkk}(\mu_t) = C_{kkk}(t)(1 - \rho_{kkk}(\mu_t))$$

$$\Downarrow$$

$$\rho_{kkk}(\mu_t) = 1 - \exp(-\overline{C_{kkk}}t)$$

- $\rho_{kkk}(\mu_t) \to 1$ converges to the binary results $\rho_{kkk} = \mathbb{I}(k \neq 0)$.

Modular Addition Example

Loss

$$\ell_k = -2\rho_{kkk} + \sum_{k_1,k_2} |\rho_{k_1k_2k}|^2 + \frac{1}{4}\left|\sum_{p\in\{a,b\}}\sum_{k'}\rho_{p,k',-k',k}\right|^2 + \frac{1}{4}\sum_{m\neq 0}\sum_{p\in\{a,b\}}\left|\sum_{k'}\rho_{p,k',m-k',k}\right|^2$$

Boolean Solutions

$$\rho_{kkk} = \mathbb{I}(k \neq 0),$$

$$\rho_{k_1k_2k} = 0,$$

$$\rho_{pk_1k_2k} = 0$$

Revisiting Dimensionality of Dynamics

Symbolic regression

ρ_0 $\quad \partial_t \rho_{r_i}(\mu_t) = -C_i(t)\partial_{\rho_{r_i}} L(\rho), \forall r \quad$ ρ_T

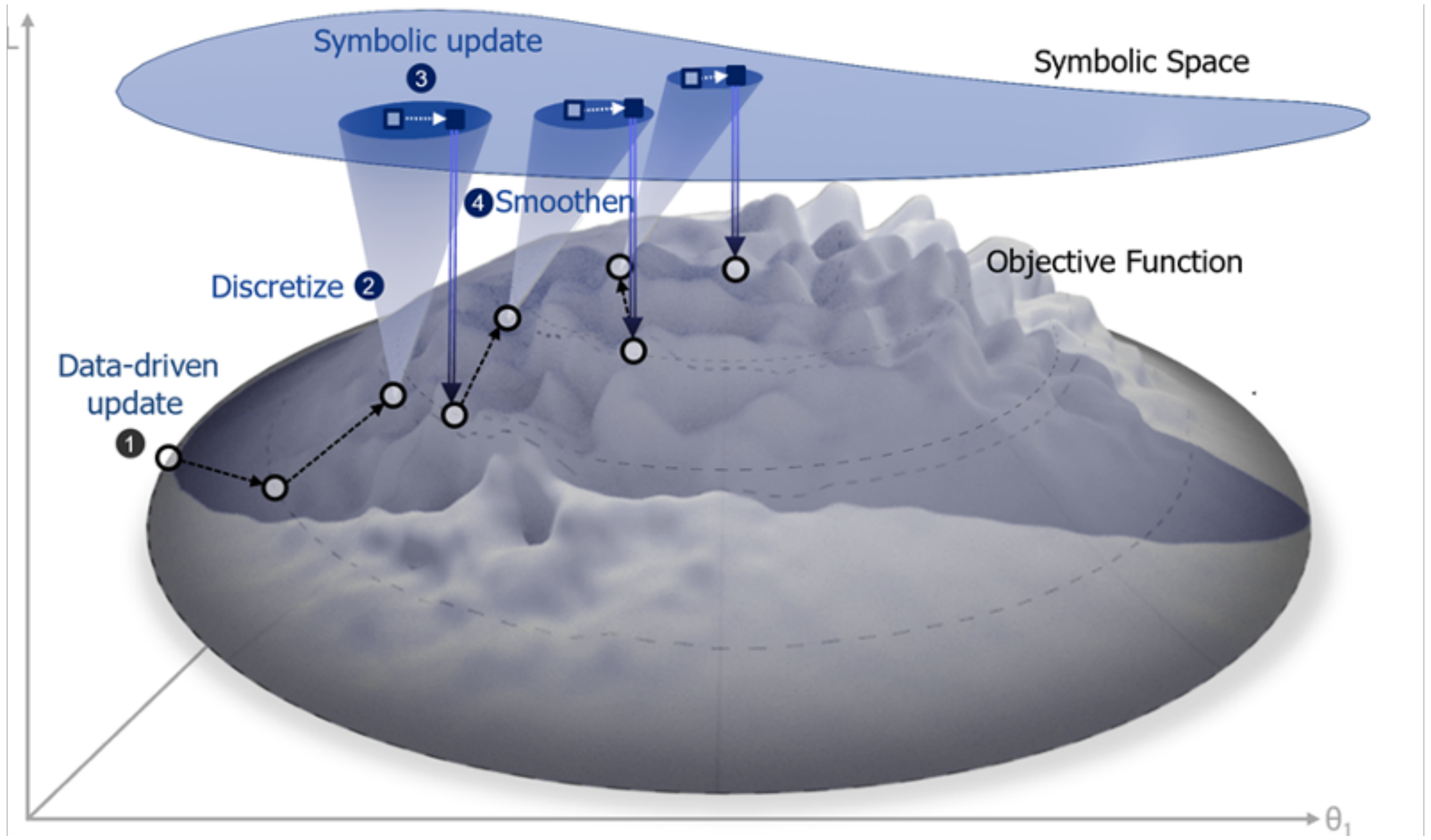
$\rho_r(\mu_0)$ $\qquad\qquad$ $\rho_r(\mu_T)$

μ_0 $\qquad\qquad$ μ_T

Neural weights training

$$\partial_t \mu_t + \nabla_z \cdot \left(\mu_t \underbrace{\nabla_z \left(\frac{\delta H}{\delta \mu}[\mu_t]\right)}_{O(d)\text{-equivariant}}\right) = 0$$

Dynamics in m-dimensional space



Dynamics in infinite-dimensional space

Dimension Reduction

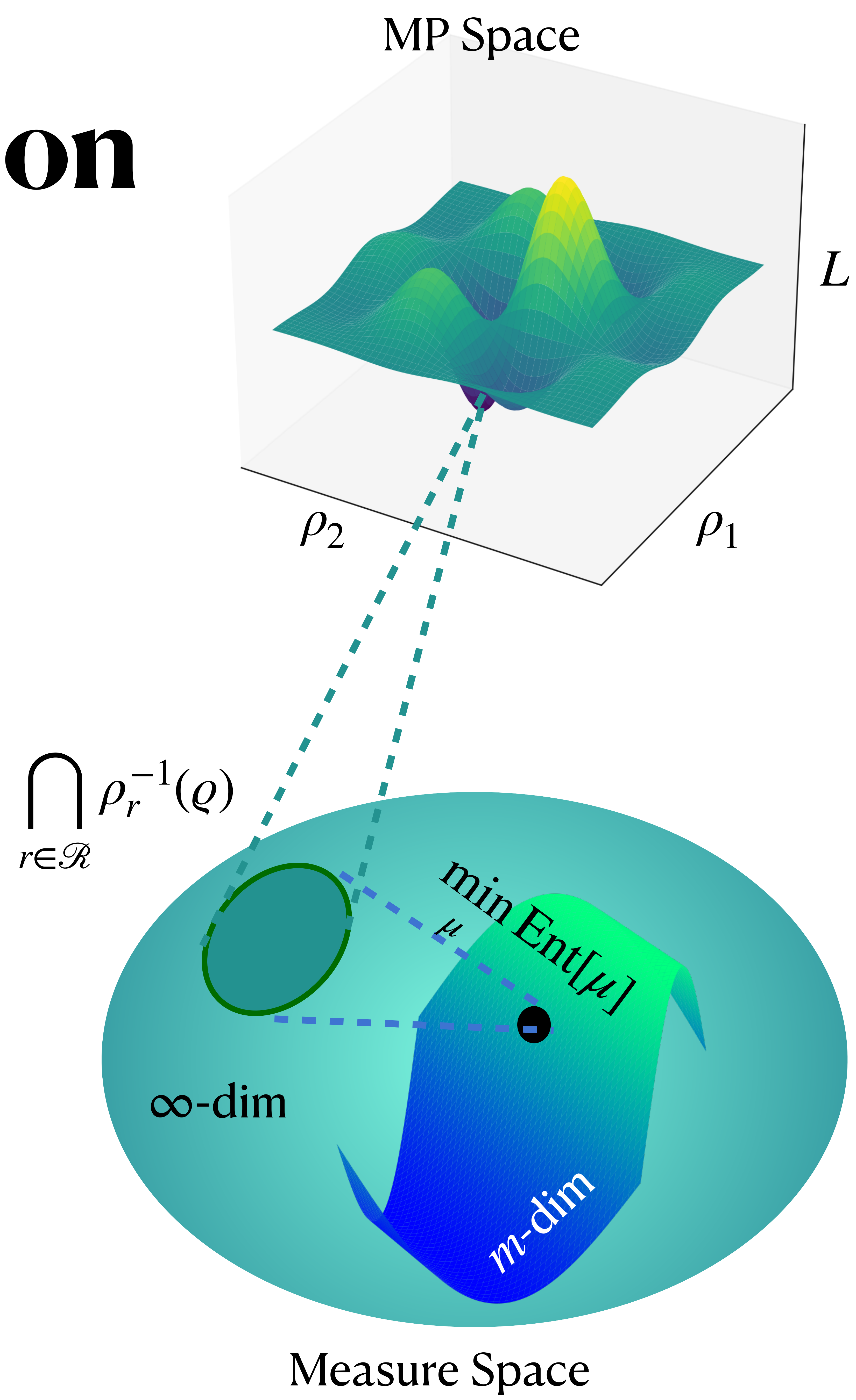
- Consider stationary points of MP dynamics (i.e., MP assignments vanishing the gradient) $\varrho \in \mathscr{R}^m$ such that $\nabla L(\varrho) = 0$

- μ^* realizes ϱ while ***minimizing differential entropy*** takes the form:

$$\mu^* \propto \exp\left(\sum_{i=1}^{m} \lambda_i r_i(x)\right)$$

where λ_i is determined to let $\rho_{r_i}[\mu^*] = \varrho_i$ for every $i \in [m]$.

- This reduces the infinite-dimensional problem to a Riemannian manifold of dimension at most m.



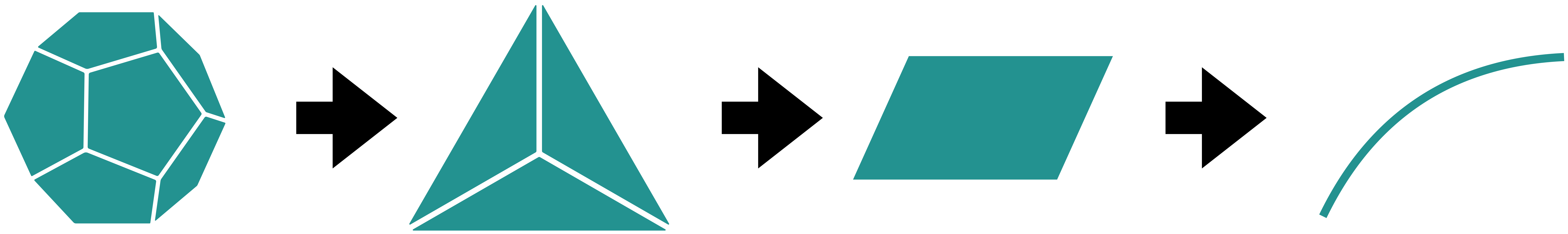
Recap: RG-Type Degree of Freedom

- Renormalization group theory studies the effective degree of freedom of the system by analyzing the Jacobian matrix $H = Df(x^*)$ of the dynamical system: $\frac{dx}{dt} = f(x)$ at its fixed point x^*.
- **Stable Manifold Theorem.** The manifold M containing initial points x which converge to fixed point x^* is tangent to the sum of eigenspaces associated with eigenvalues of H.
  - dim M = # of negative eigenvalues in H
- As the dynamical system evolves, the effect of points in M are diminishing. The remaining components are the actually effective ones.

Reduction on RG-Type Degree of Freedom

- **Theorem.** Consider loss functional $H[\mu] = L(\rho_{r_1}(\mu), \cdots \rho_{r_m}(\mu))$, suppose H is displacement-convex, then all eigenfunctions corresponding to non-zero eigenvalues of second variation $\mathbb{L}(t)$ lie in a subspace spanned by the monomial set $\mathscr{R}$, i.e, $v_i \subset \text{span}(\mathscr{R})$ if $\lambda_i \neq 0$.

The degree of freedom in RG sense is bounded by $|\mathscr{R}| = m$.



RG-Type Degree of Freedom Reduction

- Moreover, if $[\nabla^3 L]_k \nabla_k L \succeq 0$, then $\mathbb{L}(t)$ will have non-increasing eigenvalues:

$$\frac{d}{dt}\lambda(t) \leq 0.$$

An emergence of negative eigenvalues => A spontaneous reduction on RG-type degree of freedom

- There will be finitely many $0 \leq t_1 \leq t_2 \leq \cdots \leq t_m$ where an eigenvalue of $\mathbb{L}(t)$ crosses zero.

Finite-time reduction on degree of freedom.

Sample Complexity to Learn G-Invariance

$$\partial_t \mu_t + \nabla_z \cdot \left(\mu_t \nabla_z \underbrace{\left(\frac{\delta H}{\delta \mu} [\mu_t] \right)}_{O(d)\text{-equivariant}} \right) = 0$$

Theorem. Suppose G is a Lie group and M_d is a data manifold. Consider a family of G-invariant functions $\mathscr{F}^s(M_d)$, square-integrable up to order $s > 0$ over M_d.

Denote $d' = \dim(M_d/G)$ and let $s = (1+\kappa)d'/2$ for some positive integer $\kappa \geq 0$. Given $\theta \in (0,1]$, and a G-invariant function $f^* \in \mathscr{F}^{\theta s}(M_d)$, then with probability at least $1 - \delta$, empirical risk minimization can learn ϵ-approximate G-invariant function $\hat{f}$ with n many samples, where:

- $n = \Theta\left(\max\left\{1/(|G|\epsilon^{1+1/\theta(\kappa+1)}), \log(1/\delta)/\epsilon^2\right\}\right)$ for finite G.
- $n = \Theta\left(\max\left\{\mathrm{vol}(M_d/G)/\epsilon^{1+1/\theta(\kappa+1)}, \log(1/\delta)/\epsilon^2\right\}\right)$ for infinite G.

Sample Complexity to Learn G-Invariance

Remember G is the target group invariance (e.g., $O(d)$).

- $n = \Theta\left(\max\left\{1/(|G|\epsilon^{1+1/\theta(\kappa+1)}), \log(1/\delta)/\epsilon^2\right\}\right)$ for finite G.
- $n = \Theta\left(\max\left\{\mathrm{vol}(M_d/G)/\epsilon^{1+1/\theta(\kappa+1)}, \log(1/\delta)/\epsilon^2\right\}\right)$ for infinite G.
- If G is finite, group invariance reduces sample complexity by a factor of $1/|G|$.
- If G is infinite, it reduces sample complexity by contracting the data column through its orbits.

Summary of Results

We have shown:

- Algebraic structures are inherent in neural networks
- Continuous weight-space optimization can lead to solutions with symbolic structures under geometric constraints.
- Low-dimensional representations is a natural result of symbolic abstraction, enforced by information-, optimization-, and geometry-theoretic constraints.

Thanks for Listening!

Covered work