FLUID: A Self-Structuring Architecture for Open-Ended Artificial Intelligence

Winston Zai Lin
Blair Academy
linzaiwinston413@gmail.com

March 2026

Abstract

Every neural network architecture in existence shares one assumption: topology is fixed at design time and only weights vary during learning. This constraint is an artifact of gradient-based optimization, not a principled requirement of intelligence. The brain continuously grows new synaptic connections, prunes unused ones, and reorganizes functional regions throughout life—the architecture is itself a variable. I propose **FLUID** (Fluid Learning Universal Intelligence Design), an architecture in which graph topology, node count, layer structure, and time-scale hierarchy are all learnable variables alongside weights. FLUID introduces four new mechanisms absent from existing architectures: (1) *Hebbian topology growth*—edges form between co-activating nodes without human specification; (2) *causal memory*—all observations are stored with their causal context, not merely their statistical co-occurrence; (3) *temporal hierarchy emergence*—processing time-scales self-organize into fast, medium, and slow layers rather than being hand-designed; and (4) *recursive self-architecture*—a meta-network monitors the primary network and proposes structural modifications, subject to a stability constraint. I further propose a *compression-progress curiosity signal* (following Schmidhuber [**?**]) as an intrinsic reward that generates goals without human assignment. The central open problem is proving that fluid topology learning converges to stable, interpretable structures rather than degenerating. I formalize this as the *Fluid Stability Conjecture* and propose three experimental approaches to test it. FLUID is a theoretical proposal; the stability question must be resolved before empirical implementation.

1 Introduction

Consider what happens when a person learns to play piano. Over months and years, the motor cortex reorganizes: the region representing finger movements expands, new functional connections form between motor and auditory cortex, and unused pathways are pruned. The architecture changes. The weights change. They are not separable processes.

Now consider what happens when a neural network learns to play piano. The architecture—number of layers, connectivity pattern, number of units—is fixed before the first training step. Only the weights on the edges of a predetermined graph change. The graph itself is immutable.

This is not a minor limitation. It means that the representational capacity available to the network is fully determined before any learning occurs. If the designer chose the wrong architecture, no amount of learning can compensate. All of the remarkable recent progress in AI—GPT-4, AlphaFold, Gemini—is progress within this fixed-topology constraint.

The central question of this paper is: *what would an architecture look like if topology were a learnable variable?*

This is not the same as neural architecture search (NAS) [**?**], which uses gradient or evolutionary methods to select among candidate fixed topologies before training. NAS produces a fixed topology; FLUID's topology continues to change during deployment.

This is also not the same as dynamic routing in capsule networks [**?**] or mixture-of-experts [**?**], which change the *path through* a fixed graph. FLUID changes the graph itself.

The practical motivation is open-ended learning [**?**]: an agent that continues to learn new capabilities indefinitely, in a self-directed manner, without catastrophic forgetting of prior capabilities. Fixed architectures cannot achieve this—new capabilities require new structure.

2 Background and Motivation

2.1 The Fixed-Topology Constraint

Formally, let a neural network be a directed graph $G = (V, E)$ where V is a fixed set of nodes and $E \subseteq V \times V$ is a fixed set of edges. Learning modifies a weight function $w : E \to \mathbb{R}$ and a bias function $b : V \to \mathbb{R}$. The graph (V, E) itself is invariant under learning.

All gradient-based learning (backpropagation, Adam, RLHF) operates within this constraint. The gradient $\partial \mathcal{L}/\partial w_e$ is defined only for edges $e \in E$; it says nothing about whether new edges should exist.

2.2 What Biological Brains Do Differently

The adult human brain contains $\sim 8.6 \times 10^{10}$ neurons and $\sim 10^{14}$ synapses [**?**]. Approximately 10^5 new synaptic spines form and retract per hour in the motor cortex alone [**?**]. Long-term potentiation (LTP) strengthens existing synapses; long-term depression (LTD) weakens them; synaptogenesis creates new connections; synaptic pruning eliminates them.

This is not noise. Structural plasticity is how the brain encodes long-term memories and develops new skills. The structure and the weights are co-optimized.

2.3 Why Now

Two recent advances make FLUID timely:

Graph neural networks [**?**] provide differentiable operations over graphs, creating a mathematical foundation for learning with dynamic topology.

Causal inference [?] provides the mathematical machinery (do-calculus, structural causal models) to represent and learn causal structure rather than statistical associations. FLUID's memory is causal, not correlational.

3 The FLUID Architecture

3.1 Core Formalization

In FLUID, the network is a time-indexed graph $G_t = (V_t, E_t)$ where both V_t and E_t are functions of time. Learning modifies (V_t, E_t, w_t, b_t) jointly.

Definition 1 (FLUID network)**.** *A FLUID network is a quadruple* $\mathcal{F} = (G_0, \mathcal{H}, \mathcal{C}, \mathcal{M})$ *where:*

- $G_0 = (V_0, E_0)$ *is a seed graph (small, fully connected)*
- $\mathcal{H}$*: Hebbian growth rule governing edge creation and deletion*
- $\mathcal{C}$*: causal memory module storing observations with structural causal context*
- $\mathcal{M}$*: meta-network that monitors* G_t *and proposes modifications to* (V_t, E_t)

3.2 Mechanism 1: Hebbian Topology Growth

Hebb's rule [?]: "neurons that fire together wire together." In FLUID, this governs not only weight magnitude but edge existence.

Let $a_i(t)$ denote the activation of node i at time t. Define the co-activation measure:

$$\rho_{ij}(t) = \frac{1}{T} \sum_{\tau=t-T}^{t} a_i(\tau) \cdot a_j(\tau) \tag{1}$$

Edge creation rule: if $\rho_{ij}(t) > \theta_{\text{create}}$ and $(i, j) \notin E_t$, then (i, j) is added to E_{t+1} with initial weight w_0.

Edge deletion rule: if $|w_{ij}(t)| < \theta_{\text{prune}}$ for $\Delta t > T_{\text{prune}}$, then (i, j) is removed from E_{t+1}.

Node creation rule: if a cluster of edges becomes saturated (all weights at maximum, high activation variance), a new node is inserted to split the representation.

This is biologically grounded—it approximates synaptogenesis and synaptic pruning—and does not require gradient computation through the topology change itself.

3.3 Mechanism 2: Causal Memory

Standard neural networks learn $P(\text{output} \mid \text{input})$. FLUID's memory stores structural causal models (SCMs) [?].

Definition 2 (Causal memory entry)**.** *Each observation* x *is stored as a tuple* $(x, \mathbf{U}, \mathbf{PA})$ *where* $\mathbf{U}$ *is the set of context variables active at observation time and* $\mathbf{PA}(X_i)$ *is the estimated parent set of each variable in* x *under the current causal model.*

This enables retrieval not only by pattern similarity but by causal structure:

- *Observational query*: "What tends to co-occur with X?"
- *Interventional query*: "What happens if I force $X = x_0$?" (Pearl's $do(X = x_0)$)
- *Counterfactual query*: "What would Y have been if X had been x' instead of x?"

Current language models operate at the observational level. Causal memory enables the higher tiers.

3.4 Mechanism 3: Temporal Hierarchy Emergence

The brain processes information across at least five time scales: sensory (∼10ms), attentional (∼100ms), working memory (∼1s), episodic (∼minutes), semantic (∼years). FLUID does not hard-code these; they emerge.

Define the *temporal receptive field* (TRF) of a node i as the time window over which its activations are correlated with its inputs:

$$\text{TRF}(i) = \arg\min_{\Delta t} \mathbb{E}\left[\left(a_i(t) - f\left(\{a_j(t - \Delta t)\}_{j \in \text{PA}(i)}\right)\right)^2\right] \tag{2}$$

Nodes naturally cluster by TRF. Fast nodes (small TRF) handle transient events. Slow nodes (large TRF) integrate over extended history. The Hebbian growth rule preferentially connects nodes with similar TRFs, causing time-scale layers to self-organize without explicit layer design.

3.5 Mechanism 4: Recursive Self-Architecture

FLUID contains a meta-network $\mathcal{M}$ whose inputs are statistics of the primary network G_t (activation distributions, gradient norms, TRF histogram, connectivity density) and whose outputs are proposed structural modifications:

- Add/remove node
- Split overloaded node into two
- Merge underutilized nodes
- Create cross-layer shortcut
- Increase/decrease TRF of a node cluster

$\mathcal{M}$ is itself a FLUID network—it also has fluid topology. Proposed modifications are accepted if they improve performance on a held-out validation set; rejected otherwise.

This is recursive self-improvement (RSI) at the structural level.

4 Intrinsic Motivation: The Curiosity Signal

Without a human-assigned reward, FLUID requires an intrinsic motivation to direct its exploration. I adopt Schmidhuber's *compression progress* formulation [**?**]:

Definition 3 (Compression-progress curiosity)**.** *Let C_t denote the description length (in bits) of all observations up to time t under the agent's current world model. The curiosity signal at time t is:*

$$\kappa(t) = C_{t-\Delta t} - C_t \tag{3}$$

i.e., the reduction in description length achieved by recent observations. The agent is intrinsically rewarded for observations that improve compression.

This generates goals without external assignment:

- Observations that confirm the existing model have low κ (no new compression)
- Observations that violate predictions have high κ (new compression opportunity)
- Maximally random observations also have low κ (uncompressible noise)

The result is a drive toward structured novelty—the defining feature of scientific curiosity—without any human specifying what to be curious about.

5 The Fluid Stability Conjecture

The central open problem is:

> **Fluid Stability Conjecture.** *Under the Hebbian growth rule $\mathcal{H}$ with compression-progress curiosity signal κ, the FLUID network G_t converges (in distribution) to a stable, interpretable graph structure rather than degenerating to a fully connected graph, a disconnected graph, or uncontrolled growth.*

This is a conjecture, not a theorem. I do not know if it is true.

Why stability is non-obvious. The Hebbian rule creates positive feedback: active nodes create more edges, more edges increase activation capacity, more capacity enables more edges. Without a regularization mechanism, this could produce runaway edge creation. Conversely, pruning under low weight magnitude could produce runaway deletion. Stability requires that creation and deletion rates balance.

Three experimental approaches to test the conjecture:

Approach 1 (Small world). Initialize FLUID with 100 nodes on synthetic tasks (XOR, parity, symbolic regression). Monitor $|V_t|$, $|E_t|$, and mean clustering coefficient over 10^6 steps. If the conjecture holds, expect convergence to a small-world graph [**?**] with stable statistics.

Approach 2 (Phase diagram). Sweep $(\theta_{\text{create}}, \theta_{\text{prune}}, T_{\text{prune}})$ systematically and map which parameter regions produce stable graphs vs. degenerate graphs. The conjecture implies a non-empty stable region exists.

Approach 3 (Analytic bound). Seek a Lyapunov function $\mathcal{V}(G_t)$ that decreases monotonically under the growth dynamics. If such a function exists, stability is guaranteed. The natural candidate is $\mathcal{V} = C_t/|E_t|$ (compression per edge)—maximizing this penalizes both unnecessary edges (low compression gain) and insufficient edges (poor compression capacity).

6 Comparison to Prior Architectures

Architecture	Fluid topology	Causal memory	Intrinsic goals	Time hierarchy
Transformer [**?**]	No	No	No	No
Neural Turing Machine [**?**]	No	No	No	No
Capsule Networks [**?**]	Partial	No	No	No
NEAT [**?**]	Yes (evolution)	No	No	No
World Models [**?**]	No	Partial	No	No
Dreamer [**?**]	No	Partial	No	No
AIDE [**?**]	No	No	Partial	No
FLUID (proposed)	**Yes**	**Yes**	**Yes**	**Yes**

Table 1: Comparison of FLUID with prior architectures on four key properties.

NEAT [**?**] is the closest prior work: it uses evolutionary algorithms to grow neural network topology. FLUID differs in two ways: (1) topology changes occur at inference time, not only between generations; (2) the growth rule is Hebbian (local, online) rather than evolutionary (global, generational).

7 Limitations

The stability problem is unsolved. Section **??** formalizes but does not resolve the central question. FLUID should not be implemented at scale until the Fluid Stability Conjecture is tested experimentally.

Backpropagation through topology changes is undefined. Standard gradient computation assumes a fixed computation graph. Topology changes require surrogate gradients or reinforcement-based topology update rules. This is an active research area [**?**] but not fully solved.

Causal discovery is hard. Learning $\mathbf{PA}(X_i)$ from observational data requires assumptions (acyclicity, faithfulness) that may not hold in all domains. Causal memory degrades to correlational memory when these assumptions are violated.

Interpretability. A network that modifies its own topology at runtime is harder to audit than a fixed architecture. This is a safety concern for deployment.

8 Conclusion

FLUID proposes that architecture is not a hyperparameter to be searched before training but a variable to be learned during it. The four mechanisms—Hebbian growth, causal memory, temporal hierarchy emergence, and recursive self-architecture—are each individually grounded in biological or mathematical prior work. Their combination has not been attempted.

The honest status of this proposal is: theoretically motivated, mathematically specified, empirically untested, and blocked on one open problem (the Fluid Stability Conjecture). The next step is not implementation—it is testing the conjecture on small synthetic tasks.

If the conjecture holds, FLUID represents a path toward an architecture that learns what to be, not only what to know.

References

[1] Schmidhuber, J. (2010). Formal theory of creativity, fun, and intrinsic motivation. *IEEE Transactions on Autonomous Mental Development*, 2(3), 230–247.

[2] Zoph, B., & Le, Q.V. (2017). Neural architecture search with reinforcement learning. *ICLR 2017*.

[3] Sabour, S., Frosst, N., & Hinton, G. (2017). Dynamic routing between capsules. *NeurIPS 2017*.

[4] Shazeer, N., et al. (2017). Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *ICLR 2017*.

[5] Stanley, K.O., et al. (2019). Designing neural networks through neuroevolution. *Nature Machine Intelligence*, 1, 24–35.

[6] Scarselli, F., et al. (2009). The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1), 61–80.

[7] Pearl, J. (2009). *Causality: Models, Reasoning, and Inference* (2nd ed.). Cambridge University Press.

[8] Azevedo, F.A., et al. (2009). Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *Journal of Comparative Neurology*, 513(5), 532–541.

[9] Bhatt, D.L., et al. (2009). Filopodia are like cellular antennae, important for calcium signaling. *Nature Neuroscience*, 12, 1325–1331.

[10] Hebb, D.O. (1949). *The Organization of Behavior*. Wiley & Sons.

[11] Vaswani, A., et al. (2017). Attention is all you need. *NeurIPS 2017*.

[12] Graves, A., Wayne, G., & Danihelka, I. (2014). Neural Turing machines. *arXiv:1410.5401*.

[13] Stanley, K.O., & Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2), 99–127.

[14] Ha, D., & Schmidhuber, J. (2018). World models. *arXiv:1803.10122*.

[15] Hafner, D., et al. (2020). Dream to control: Learning behaviors by latent imagination. *ICLR 2020*.

[16] Lu, C., et al. (2024). The AI scientist: Towards fully automated open-ended scientific discovery. *arXiv:2408.06292*.

[17] Watts, D.J., & Strogatz, S.H. (1998). Collective dynamics of small-world networks. *Nature*, 393, 440–442.

[18] Bellec, G., et al. (2018). Deep rewiring: Training very sparse deep networks. *ICLR 2018*.