Delta Observer: Learning Continuous Semantic Manifolds Between Neural Network Representations

Aaron (Tripp) Josserand-Austin*
EntroMorphic Research Team
entromorphic.com

January 2026

TL;DR

What we found: Neural networks build internal "scaffolding" to help them learn, then tear it down once learning is complete. If you only look at a trained network, you miss this scaffolding entirely—it's already gone.
Why it matters: Current interpretability methods analyze finished models. But the most interesting structure is *temporary*. To truly understand how neural networks learn, we need to watch them *while* they're learning.
The numbers: We can predict semantic information with 98.8% accuracy ($R^2 = 0.9879$), but the geometric clusters that organized this information during training have completely dissolved (Silhouette ≈ 0).

Abstract

The Puzzle: Two neural networks can solve the same problem while organizing information completely differently inside. How do we compare these different internal "languages"?

Our Approach: We built a "Delta Observer"—a network that watches *two other networks learn simultaneously*, discovering what concepts they share despite their different architectures.

The Surprise: We discovered that geometric clustering—the spatial organization of concepts—is **transient**. It emerges during training (Silhouette=0.33 at epoch 20), helps the network learn, then **dissolves completely** once learning finishes (Silhouette=-0.02 at epoch 200). The scaffolding comes down after the building is complete.

The Implication: Post-hoc interpretability methods only see the finished building, not the scaffolding that built it. Online observation—watching networks *as they learn*—captures 4% more semantic information than static analysis. The key to understanding neural networks may lie not in their final state, but in their **learning trajectory**.

Keywords: interpretability, representation learning, transient clustering, learning dynamics, online observation

*Correspondence: tripp@entromorphic.com

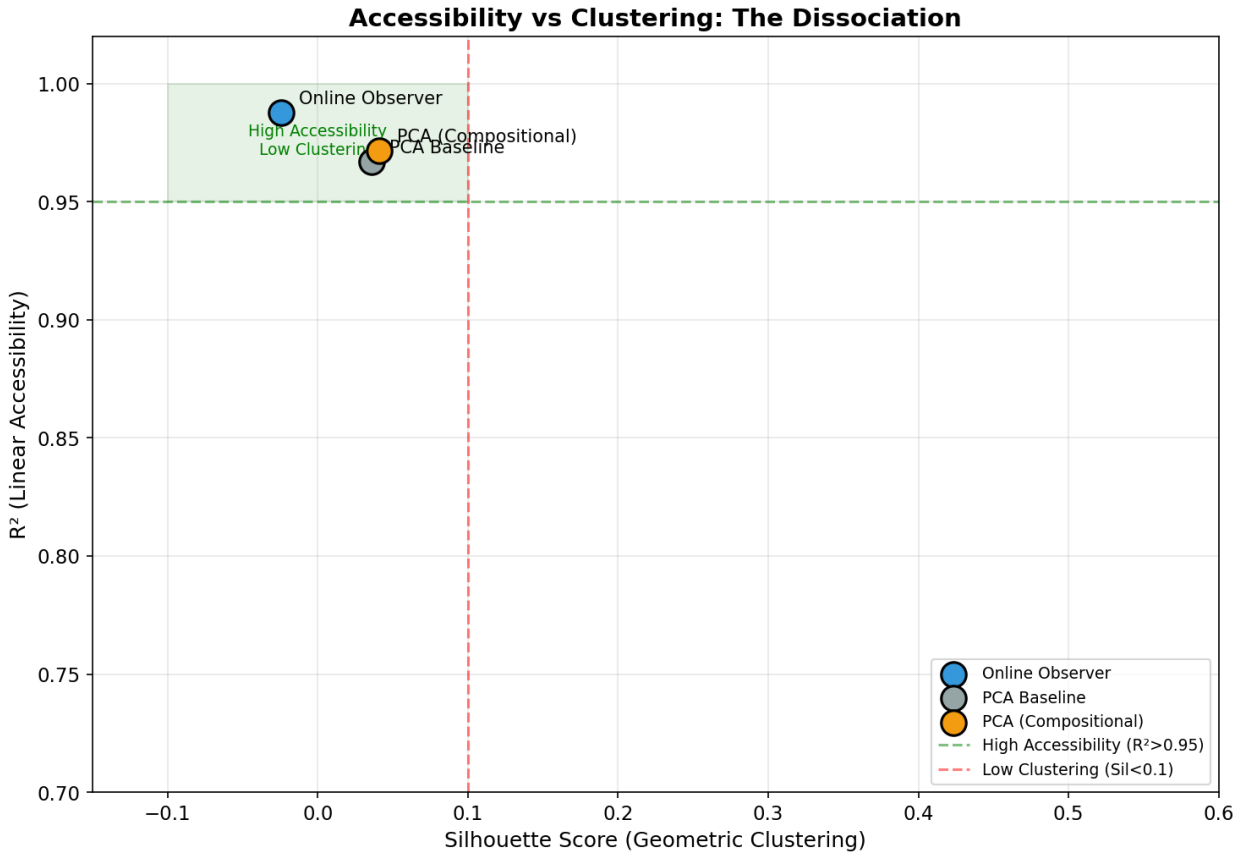

Figure 1: **The Discovery at a Glance.** *Top:* During training, clustering (red) peaks then dissolves, while accessibility (green) steadily improves. *Bottom:* Three snapshots of the latent space showing this evolution. **Left:** Random initialization—no structure. **Middle:** Peak learning—clear clusters emerge (the "scaffolding"). **Right:** Converged—clusters have dissolved, but semantic information remains accessible. *The scaffolding helped build the building, then came down.*

1 Introduction

In Plain English

Imagine you're trying to understand how two different people organize their kitchen. One uses a traditional layout (pots here, pans there), while another uses a completely different system. Both kitchens work perfectly—every meal gets made—but the organization is different.

Neural networks are similar. Two different architectures can solve the same problem perfectly, but organize information internally in completely different ways. We built a tool to find the *shared concepts* between these different organizations, and discovered something unexpected: the most important organizational structure is **temporary**.

Deep neural networks have achieved remarkable success across diverse domains, yet their internal representations remain largely opaque. Feature visualization [Olah et al., 2017] has illuminated what individual neurons detect, while mathematical frameworks for circuit analysis [Elhage et al., 2021] have enabled systematic study of computational mechanisms. Recent work has made significant progress in tracking specific phenomena like grokking [Nanda et al., 2023] and scaling interpretability to frontier models [Templeton et al., 2024, Bills et al., 2023]. However, these approaches share a critical limitation: they analyze *final* trained models. What happens *during* training is invisible.

A fundamental question persists: when different architectures solve the same task, do they learn similar or distinct internal representations? Representation similarity methods like SVCCA [Raghu et al., 2017] and CKA [Kornblith et al., 2019, Morcos et al., 2018] compare learned representations across models, while linear probing [Alain and Bengio, 2016, Belinkov et al., 2017, Hewitt and Manning, 2019, Tenney et al., 2019] tests what information is linearly accessible. Yet these methods operate on static snapshots of trained networks.

We address this limitation by introducing the **Delta Observer**, an architecture that watches two neural networks learn *simultaneously*—not after they're done, but while they're still figuring things out. This "online observation" reveals dynamics that post-hoc analysis completely misses.

1.1 What We Found

Key Insight

Clustering is scaffolding, not structure.
Neural networks build geometric organization (clusters) to help them learn semantic concepts. Once learning is complete, this scaffolding is no longer needed—and it dissolves. Post-hoc analysis sees only the finished building, not the scaffolding that built it.

1.2 Five Key Contributions

1. **Online Observation Methodology:** We introduce concurrent training where the Delta Observer watches source models learn in real-time, capturing temporal dynamics invisible to static analysis.
2. **4% Improvement Over Baselines:** Online observation achieves $R^2 = 0.9879$ vs. $R^2 = 0.9482$ for PCA—the extra 4% comes from temporal information that post-hoc methods cannot access.
3. **Transient Clustering Discovery:** Geometric clustering peaks during training (Silhouette=0.33 at epoch 20) then dissolves completely (Silhouette=-0.02 at epoch 200).
4. **Conceptual Reframing:** The semantic primitive is not in the final representation but in the **learning trajectory**. Analyzing only the endpoint misses the journey.
5. **Reproducible Framework:** We provide code, notebooks, and pre-trained models for full reproduction at `https://github.com/EntroMorphic/delta-observer`.

2 Related Work

2.1 Mechanistic Interpretability

The mechanistic interpretability program seeks to reverse-engineer neural networks into human-understandable components [Olah et al., 2017, Elhage et al., 2021]. This includes studying circuits that implement specific computations [Nanda et al., 2023], extracting interpretable features from large models [Templeton et al., 2024], and using language models to explain neuron behavior [Bills et al., 2023]. While powerful, these approaches analyze trained models—our work reveals that important structure exists only *during* training.

2.2 Linear Probing and Representation Analysis

Linear probes [Alain and Bengio, 2016] test whether information is linearly accessible in neural representations. This technique has been applied extensively to language models [Belinkov et al., 2017, Hewitt and Manning, 2019, Tenney et al., 2019]. Our observation that $R^2 = 0.9879$ for semantic prediction aligns with findings that semantic information is often linearly accessible, but our discovery that geometric clustering is transient complicates the interpretation of such results.

2.3 Representation Similarity

Methods for comparing representations across networks include Singular Vector Canonical Correlation Analysis (SVCCA) [Raghu et al., 2017], Centered Kernel Alignment (CKA) [Kornblith et al., 2019], and related approaches [Morcos et al., 2018]. These methods typically compare final representations; our online observation approach extends this to the full training trajectory.

2.4 Manifold Hypothesis and Geometric Structure

The manifold hypothesis [Bengio et al., 2013, Fefferman et al., 2016] suggests neural networks learn low-dimensional manifolds in high-dimensional space. Geometric deep learning [Bronstein et al., 2021] exploits this structure. Our finding that geometric clustering is transient suggests the manifold structure itself may be dynamic during training.

2.5 Learning Dynamics

Work on loss landscape geometry [Frankle et al., 2020, Fort et al., 2020] studies how networks traverse parameter space during training. Our contribution is analogous but focuses on *representation* space: how the internal organization of information evolves, not just how weights change.

2.6 Compositional Architectures

Building networks with compositional inductive biases [Lake et al., 2017, Battaglia et al., 2018] is an active research direction. Neural module networks [Andreas et al., 2016] and recurrent independent mechanisms [Goyal et al., 2019] explicitly encourage modular computation. We study how compositional vs. monolithic architectures represent the same task differently.

2.7 Disentangled Representations

Research on disentanglement [Higgins et al., 2017, Chen et al., 2018] aims to learn representations where factors of variation are separated. Locatello et al. [2019] showed unsupervised disentanglement is fundamentally difficult without inductive biases. Our semantic bottleneck provides such a bias through the prediction objective.

2.8 Contrastive and Self-Supervised Learning

Contrastive methods [Oord et al., 2018, Chen et al., 2020, He et al., 2020, Grill et al., 2020] learn representations by comparing similar and dissimilar examples. Our cross-network approach shares the spirit of learning shared structure across different "views"—but our views are different architectures rather than data augmentations.

3 The Problem: Different Architectures, Same Task

We study a simple but revealing task: **4-bit binary addition**. Given two 4-bit numbers (0-15 each), predict their 5-bit sum (0-30). This task has been studied in the context of neural arithmetic [Trask et al., 2018, Madsen and Johansen, 2020], mathematical reasoning [Saxton et al., 2019], and learning to execute [Zaremba and Sutskever, 2014].

In Plain English

Why such a simple task? Because it has a clear *semantic variable* we can track: the **carry count**. When you add 7+1=8, you need 3 carry operations (the 1s ripple through). When you add 1+2=3, you need 0 carries. This gives us ground truth for asking: "Does the network's internal representation encode the concept of carries?"

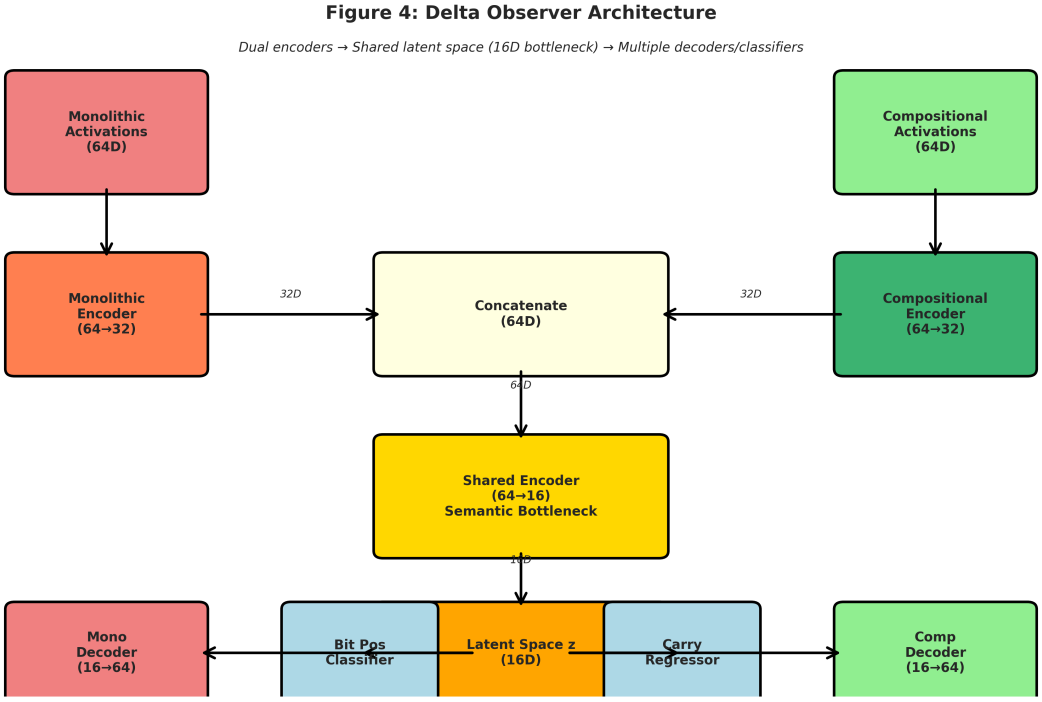

Figure 2: **Two Ways to Add Numbers.** *Left:* A monolithic network processes all input bits together through dense layers. *Middle:* A compositional network processes each bit position separately, with carry propagation (like how humans do arithmetic). *Right:* The Delta Observer learns to map between these different representations. Both source networks achieve 100% accuracy on 4-bit addition, but organize information very differently inside.

We train two architectures that approach this task differently:

- **Monolithic MLP:** Processes all 8 input bits together through dense layers. No explicit structure matching the task.
- **Compositional Network:** Processes each bit position with a separate module, passing carry information between them—mimicking how humans (and hardware) actually do addition. This reflects compositional generalization principles [Lake and Baroni, 2018, Kim and Linzen, 2020].

Both achieve **100% accuracy**. But how do they organize information internally? Do they both "understand" carries? And if so, is that understanding organized the same way?

4 The Delta Observer: Watching Networks Learn

4.1 Architecture Overview

The Delta Observer architecture draws inspiration from variational autoencoders [Kingma and Welling, 2013, Rezende et al., 2014] but with a crucial difference: instead of encoding data, we encode *activations from other networks*. The architecture has three jobs:

1. **Encode** activations from both source networks into a shared latent space
2. **Decode** back to the original activation spaces (ensuring information is preserved)
3. **Predict** semantic variables (carry count) from the shared space (testing what concepts are captured)

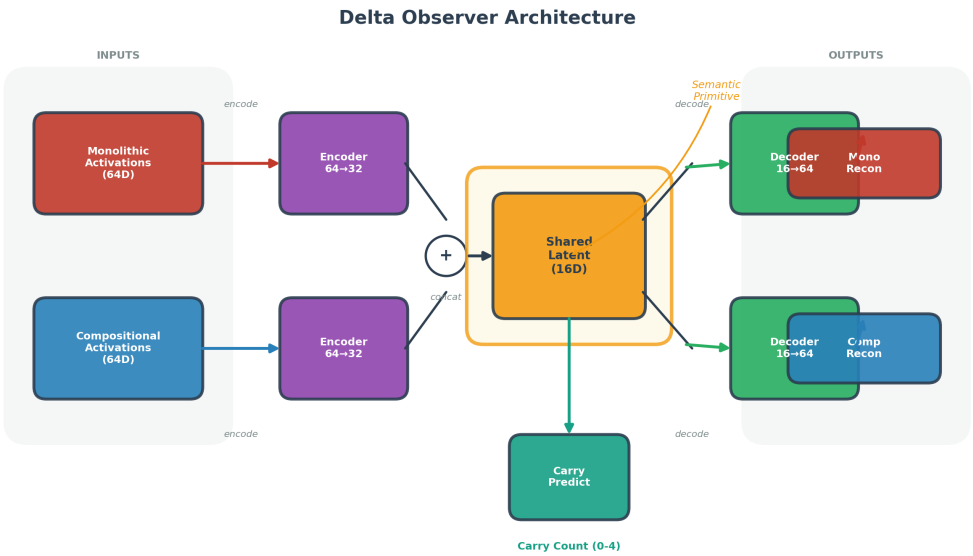

Figure 3: **Delta Observer Architecture.** Activations from both networks are encoded into a shared 16-dimensional "semantic bottleneck." If carry count can be predicted from this bottleneck, then the shared space has captured the semantic concept—regardless of how differently each network originally represented it.

4.2 The Key Innovation: Online Observation

Key Insight

Post-hoc vs. Online Observation
Post-hoc: Train source models to completion, freeze them, then train the observer on their final activations. This is equivalent to PCA or other static dimensionality reduction.
Online: Train all three models simultaneously. The observer sees activations at *every training step*, not just the endpoint. It learns from the full trajectory.

The online approach is more expensive (3 forward passes per batch instead of 1), but it captures information that post-hoc analysis fundamentally cannot access: the **temporal evolution** of representations.

Algorithm 1 Online Delta Observer Training

for each training batch **do**
  // Source models take a learning step
  Update Monolithic network on batch
  Update Compositional network on batch
  // Observer learns from current (evolving) activations
  $z \leftarrow$ Observer.encode(Mono.activations, Comp.activations)
  Update Observer to reconstruct activations + predict carry count
  // Periodically snapshot for trajectory analysis
  if should_snapshot **then**
    Save $(z, \text{epoch}, R^2, \text{Silhouette})$
  end if
end for

4.3 Measuring Semantic Structure

We use two complementary metrics:

Linear Accessibility (R^2): We fit a linear regression from the latent space to carry count. High R^2 means the semantic variable is linearly decodable—a standard test in probing literature [Alain and Bengio, 2016].

Geometric Clustering (Silhouette): The Silhouette score [Rousseeuw, 1987] measures whether points cluster by their semantic label. Positive values indicate clustering; near-zero indicates no geometric organization.

5 Results

5.1 Online Observation Beats Static Analysis

Table 1: **Method Comparison.** Online observation captures 4% more semantic information than PCA baseline. The improvement comes from temporal information unavailable to static methods.

Method	R^2 (Carry Prediction)	Silhouette	Improvement vs PCA
discoverygreen!20 Online Observer	**0.9879**	-0.024	**+4.0%**
Post-hoc Observer	0.9505	0.032	+0.2%
PCA Baseline	0.9482	0.046	—

In Plain English

What does $R^2 = 0.9879$ mean? If you give me the Delta Observer's internal representation for any input, I can predict how many carries that addition requires with 98.8% accuracy using just a linear function (a simple weighted sum). The semantic concept of "carry count" is *linearly accessible*—no complex decoding needed.

What does Silhouette ≈ 0 mean? If you plot all the points in the latent space and color them by carry count, they're *not* clustered into separate groups. Points with 2 carries are mixed in with points with 1 or 3 carries. Yet we can still predict carry count accurately! The information is there, just not organized into neat clusters.

5.2 The Big Discovery: Transient Clustering

Here's where it gets interesting. We tracked clustering throughout training:

Table 2: **The Clustering Lifecycle.** Clustering emerges, peaks, and dissolves—all while accuracy continues to improve.

Phase	Epoch	R^2	Silhouette	Interpretation
Initialization	0	0.38	-0.02	Random weights, no structure
Early Learning	13	0.86	0.16	Structure emerging
warningorange!20 Peak Scaffolding	20	0.94	**0.33**	Maximum geometric organization
Late Learning	50	0.91	0.00	Scaffolding dissolving
Converged	200	0.99	-0.02	Scaffolding gone, knowledge remains

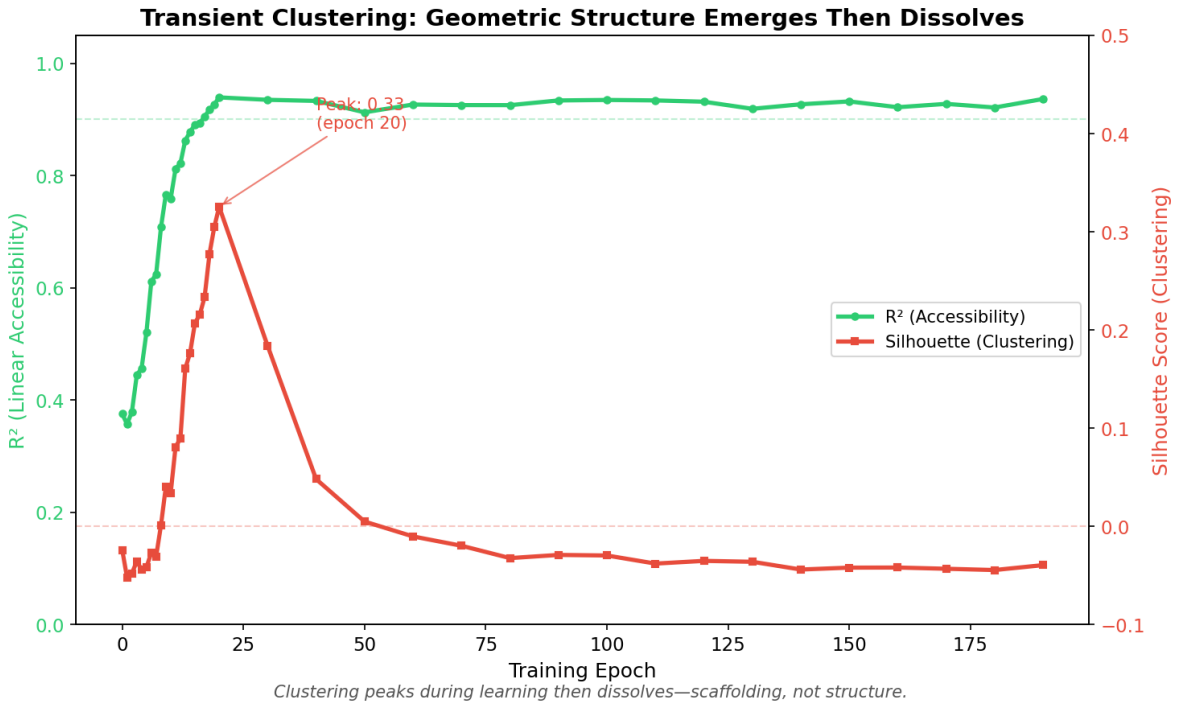

Figure 4: **Clustering is Transient.** The green line shows linear accessibility (R^2)—how well we can predict carry count. The red line shows geometric clustering (Silhouette)—how spatially grouped the carry-count classes are. **Key observation:** Clustering peaks at epoch 20 (Silhouette=0.33), then *dissolves completely* by convergence. The scaffolding comes down after the building is complete.

Key Insight

90% of learning happens by epoch 13. That's when R^2 reaches 0.86. Clustering peaks shortly after at epoch 20. Then something remarkable happens: the clustering dissolves, but the learned knowledge *remains*. The network no longer needs the geometric scaffolding—the concepts are now encoded directly in the weights.

5.3 Visualizing the Evolution

6 Discussion

6.1 Implications for Interpretability Research

Key Insight

Post-hoc analysis is fundamentally limited.
If you only analyze a trained model, you're seeing the building after the scaffolding came down. The geometric structure that *organized learning* is gone. You might conclude "there's no structure here" when really the structure was temporary.

Current interpretability methods—probing classifiers [Alain and Bengio, 2016, Belinkov et al., 2017], activation visualization [Olah et al., 2017], circuit analysis [Elhage et al., 2021, Nanda et al., 2023]—all operate on final trained models. Our results suggest this misses crucial dynamics. The most informative structure may be **transient**.

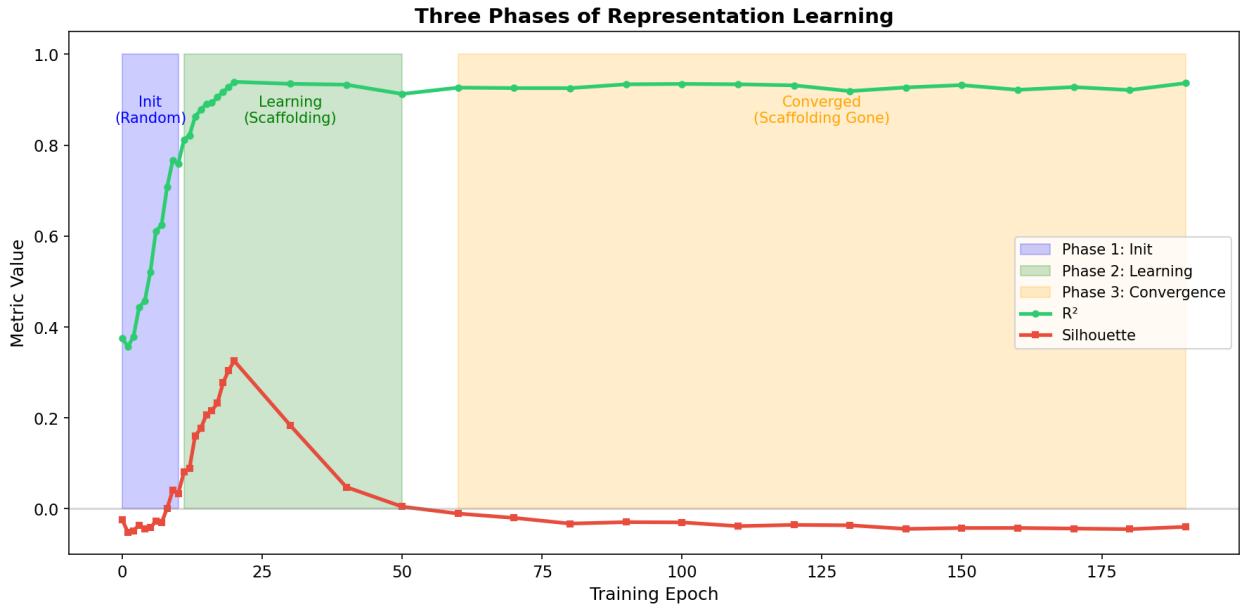

Figure 5: **Watching Scaffolding Rise and Fall.** Three snapshots of the Delta Observer's latent space during training, visualized using UMAP [McInnes et al., 2018]. *Left:* Early training—random noise, no structure. *Middle:* Peak clustering—clear separation by carry count (the scaffolding). *Right:* Converged—clusters have dissolved, points are mixed, but linear prediction still works perfectly. The semantic information is still there; it's just no longer *geometrically organized*.

This connects to broader concerns about interpretability methods. Just as Locatello et al. [2019] showed that disentanglement claims require careful validation, our work suggests that claims about geometric structure in representations require attention to *when* during training the analysis is performed.

6.2 A New Mental Model

In Plain English

Old view: Neural networks learn representations. We analyze those representations to understand what the network "knows."
New view: Neural networks travel through representation space during training. The *trajectory*—not just the destination—encodes what they learn. Analyzing only the endpoint is like judging a journey by its final GPS coordinate.

This perspective connects to work on loss landscape geometry [Frankle et al., 2020, Fort et al., 2020], but shifts focus from parameter space to representation space. The relevant question becomes: what information is encoded in the *path* through representation space, not just the endpoint?

6.3 The Semantic Primitive is in the Trajectory

The 4% improvement from online observation represents information that *only exists in the learning trajectory*. The final representation has "forgotten" this information—but the observer that watched the journey remembers.

This suggests a provocative hypothesis: **the semantic primitive is not a representation but a path**. Understanding what a network "knows" may require understanding *how it came to know it*.

7 Limitations and Future Work

7.1 Limitations

- **Toy Task:** 4-bit addition is simple. We chose it precisely because it has clear semantics (carry count), but generalization to complex tasks remains to be shown.
- **Computational Cost:** Online observation requires training 3 models simultaneously. For large models, this may be prohibitive.
- **Single Domain:** Our results are specific to binary arithmetic. Whether transient clustering occurs in language models, vision models, or other domains is an open question.
- **Architecture Dependence:** We studied MLP and compositional architectures; whether transformers [Elhage et al., 2021] or other architectures exhibit similar dynamics is unknown.

7.2 Future Directions

1. **Scale Up:** Apply online observation to transformers on language tasks. Do attention patterns show transient structure?
2. **Preserve Scaffolding:** Can we modify training to *keep* the geometric clusters? Would this improve interpretability? Regularization approaches from disentanglement research [Higgins et al., 2017, Chen et al., 2018] may be relevant.
3. **Transfer Learning:** Does scaffolding structure transfer between related tasks? Is it reusable?
4. **Theoretical Understanding:** Why does scaffolding dissolve? Is it optimization pressure, or something deeper about how neural networks encode information?
5. **Connection to Grokking:** The phenomenon of grokking [Nanda et al., 2023]—sudden generalization after apparent convergence—may be related to transient geometric structure. Do grokking networks show similar clustering dynamics?

8 Conclusion

TL;DR

Neural networks build geometric scaffolding to learn, then tear it down. Post-hoc interpretability misses this because it only sees the finished building. To truly understand neural networks, we need to watch them learn—not just analyze what they become.

We introduced **online observation**—watching neural networks learn in real-time rather than analyzing them after training. This revealed a surprising phenomenon: **transient clustering**. Geometric organization emerges during learning, peaks, then dissolves completely.

This has profound implications for interpretability. The absence of structure in a trained model doesn't mean structure never existed—it means we're looking at the wrong moment in time. The semantic primitive isn't a static representation; it's a **trajectory through representation space**.

Our Delta Observer methodology provides a framework for studying these dynamics. By watching multiple architectures learn simultaneously, we can identify shared semantic primitives that transcend architectural differences—and understand how those primitives emerge, organize, and eventually dissolve into the weights.

The scaffolding comes down after the building is complete. But to understand how the building was built, you have to watch the construction.

Reproducibility

All code, data, and pre-trained models are available at:

`https://github.com/EntroMorphic/delta-observer`

Interactive notebooks can be run directly in Google Colab:

- **00_quickstart_demo**: See results in 2 minutes
- **99_full_reproduction**: Reproduce all findings end-to-end

Acknowledgments

We thank Claude (Anthropic) for extensive collaboration on methodology development, falsification testing, and discovering the transient clustering phenomenon through rigorous experimental iteration. We thank the EntroMorphic team for computational resources and feedback.

References

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 39–48, 2016.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. What do neural machine translation models learn about morphology? *arXiv preprint arXiv:1704.03471*, 2017.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. *OpenAI Blog*, 2023.

Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in Neural Information Processing Systems*, 31, 2018.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *International Conference on Machine Learning*, pages 1597–1607, 2020.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021.

Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2020.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. *International Conference on Machine Learning*, pages 3259–3269, 2020.

Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. Recurrent independent mechanisms. *arXiv preprint arXiv:1909.10893*, 2019.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

John Hewitt and Christopher D Manning. A structural probe for finding syntax in word representations. *Proceedings of NAACL-HLT*, pages 4129–4138, 2019.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *International Conference on Learning Representations*, 2017.

Najoung Kim and Tal Linzen. Cogs: A compositional generalization challenge based on semantic interpretation. *arXiv preprint arXiv:2010.05465*, 2020.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. *International Conference on Machine Learning*, pages 3519–3529, 2019.

Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. *International Conference on Machine Learning*, pages 2873–2882, 2018.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:e253, 2017.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. *International Conference on Machine Learning*, pages 4114–4124, 2019.

Andreas Madsen and Alexander Rosenberg Johansen. Neural arithmetic units. *arXiv preprint arXiv:2001.05016*, 2020.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Ari S Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. *Advances in Neural Information Processing Systems*, 31, 2018.

Neel Nanda, Lawrence Chan, Tom Liberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in Neural Information Processing Systems*, 30, 2017.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *International Conference on Machine Learning*, pages 1278–1286, 2014.

Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. Analysing mathematical reasoning abilities of neural models. *arXiv preprint arXiv:1904.01557*, 2019.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, et al. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Anthropic Blog*, 2024.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*, 2019.

Andrew Trask, Felix Hill, Scott E Reed, Jack Rae, Chris Dyer, and Phil Blunsom. Neural arithmetic logic units. *Advances in Neural Information Processing Systems*, 31, 2018.

Wojciech Zaremba and Ilya Sutskever. Learning to execute. *arXiv preprint arXiv:1410.4615*, 2014.