Bootstrap Discovery of Sensory Primitives for Embodied Visual Learning

Jerry Felix
Brain-CA Technologies
Sarasota, FL, USA

Abstract

A visual modality's transformation structure, the set of permutations that move a stimulus to a perceptually equivalent stimulus on the same retina, is discoverable from observed data using a discovery loop whose elemental binary learner synthesizes to 48 LUTs and carries no gradient-training hyperparameters. Given a fixed retinal geometry, the loop bootstraps the modality's transformation primitives using a local sparsity prior on a permutation proposal distribution and a composition-closure step, without an explicit catalog of rotations, reflections, or translations. On a 12×12 toroidal binary retina the loop recovers the $\mathbb{Z}_{12}^2 \rtimes D_4$ symmetry group ($|G| = 1{,}152$ elements), matching by discovery what a competent engineer would specify by design. The discovered group then acts as a strong inductive bias: five-shot classification under 10% bit-flip noise rises from 28% with raw nearest-neighbor to 100% with orbit-expanded nearest-neighbor, and transfers cleanly to a held-out shape class. As a cross-format sanity check, the same loop identifies D_4 transformations on a 3×3 grid at 95% accuracy. The discovery layer is a candidate substrate for the always-on, few-shot, low-power visual learning embodied systems require, complementary to neural rendering rather than competitive with it.

CCS Concepts: • **Computing methodologies** → **Computer vision representations**; *Neural networks*; • **Computer systems organization** → **Other architectures**.

Keywords: embodied AI, few-shot visual learning, primitive discovery, on-device perception, AR/VR/XR, binary learners

ACM Reference Format:
Jerry Felix. 2026. Bootstrap Discovery of Sensory Primitives for Embodied Visual Learning. In *VisArch 2026: Workshop on Systems and Architectures for Visual Computing, AR/VR, and Embodied Intelligence*. ACM, New York, NY, USA, 5 pages.

1 Introduction

Embodied systems (head-mounted XR devices, autonomous robots, on-body sensors) share three constraints ill-served by the data-center training paradigm. They must learn *on-device*, because round-tripping video to a remote service exceeds the latency and power budgets these applications can pay. They must learn from *few examples*, because a wearable demanding thousands of labeled exemplars per object will not be tolerated. And they must be *robust under noise*, because the sensor sees what the sensor sees: motion blur, occlusion, lighting drift, none of which are negotiable.

The dominant response to these constraints has been to make the network smaller, the weights lower precision, or the training pipeline more clever. We take a different position: large neural networks are not the only learning substrate, and not obviously the right one for the regime above. This paper reports proof-of-concept results for an alternative substrate: an elemental binary learner (the *Brain-CA Estimator*) composed into a loop learning the structure of a visual modality from observed data, without back-propagation and without labels for rotations, reflections, or translations.

Our central observation is geometric. A 144-pixel binary retina admits $144! \approx 10^{248}$ pixel permutations. Almost all are nonsense: shuffling pixels at random preserves no shape, locality, or connectedness. The set of *meaningful* transformations (those preserving perceptual equivalence on the same retina) is vastly smaller, and crucially, determined by the modality, not by the designer. We show this set is discoverable from a few stimulus presentations using a local sparsity prior and a composition-closure step, and once discovered acts as a powerful inductive bias for few-shot learning.

We make four contributions: (i) a method for bootstrapping a modality's transformation group from observed data, with only a termination threshold K and a sparsity-tail parameter as discovery-loop hyperparameters; (ii) recovery of a 1,152-element discovered group on a 12×12 binary retina, matching the hand-coded $\mathbb{Z}_{12}^2 \rtimes D_4$ prior element-by-element; (iii) a 28% → 100% lift in noisy five-shot classification via orbit expansion; and (iv) cross-format identification of D_4 transformations on a 3×3 grid at 95% (not an ARC claim).

2 Background and Position

Why not a small neural network. Compact CNNs [6] and vision transformers are heavily optimized for on-device inference, but remain back-propagation trained, hyperparameter-dependent, and floating-point in their inner loops. Continual on-device adaptation requires a server-side fine-tuning pass or a low-rank adapter scheme, neither of which fits a strictly local, few-example update rule. Few-shot methods such as prototypical networks [11] and MAML [5] reduce the example budget but still assume a pre-trained backbone, which pushes the on-device problem upstream rather than solving it. Group-equivariant networks [2] and their steerable generalizations [12] share our interest in transformation structure

but bake the group in by hand; we discover it instead. Capsule networks [10] share that intuition but rely on dynamic routing trained by back-propagation.

The Estimator primitive. The Brain-CA Estimator maintains a b-bit register s. A SELECT scan compares the $b-1$ payload bits to $b-1$ independent random bits MSB-to-LSB; the value y_t at the first mismatch is the prediction, the most-likely value of x_t. On observing x_t, the update is

$$s_{t+1} = s_t \oplus \big(m_t \cdot \mathbf{1}[x_t \neq y_t]\big), \tag{1}$$

a ± 1 on the state index (sign forced by y_t), where m_t is an LSB-anchored mask running through the first bit equal to y_t. The carry-boundary scan is off-cycle, leaving the on-cycle datapath as one comparator and one XOR with no adder or carry chain. Under stationary observations with rate θ, the state has exact Binomial$(N-1, \theta)$ stationary distribution over its $N = 2^{b-1}$ tracking states [4]. The update uses no multiplication, no floating point, and no learned hyperparameters. The $b = 8$ Brain-CA Estimator synthesizes to 48 LUTs and 8 registers at 96 MHz on a commercial 16 nm FPGA target; the update is strictly local, and many lanes fit on a modest tile (§4.5).

Why a transformation group, and why discover it. A transformation group on a retina is a set of permutations of pixel indices that preserve the modality's invariances. For a 12×12 rotation-and-reflection-symmetric retina with toroidal boundary handling, this is the semidirect product $\mathbb{Z}_{12}^2 \rtimes D_4$: the translation lattice $\mathbb{Z}_{12}^2$ (144 elements) acted on by the dihedral group D_4 (8 elements), giving 1,152 elements. Hand-coding this group is standard practice but does not generalize: a hexagonal, curved, or event-based retina, as well as a multi-modal sensor stack, each has a different group, and the engineer must rebuild the catalog every time. We show instead that the group can be discovered from data, by combining a local sparsity prior on the proposal distribution (visual symmetries tend to move few pixels far) with an Estimator-based acceptance test. Distance-based classifiers [8] provide the few-shot baseline we compare against in Section 4; neuromorphic substrates such as Loihi [3] share the low-power, local-update orientation but retain spike-based primitives rather than bit-serial integer ones.

3 Method: Bootstrap Primitive Discovery

The discovery procedure (Fig. 1) is a closed loop: a *stimulus generator* presents base patterns; a *permutation hypothesizer* proposes candidate permutations; an *Estimator pool* scores each candidate by whether it maps observed patterns to observed patterns. Candidates that pass are retained as primitives; the rest are discarded.

Stimulus presentation. Binary base patterns on a 12×12 grid are presented in their observed forms, including natural transforms. The system sees only the bit patterns; it does not know which forms are translations, rotations, or reflections.

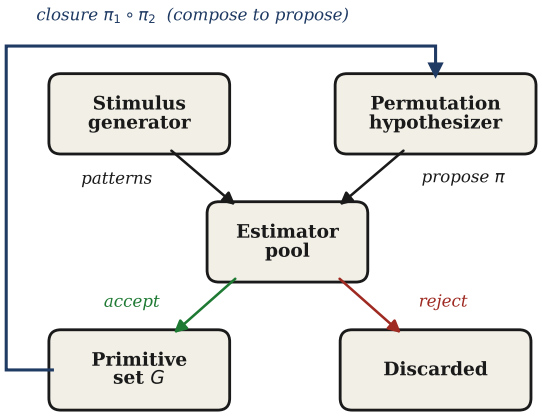

Figure 1. The bootstrap primitive-discovery loop. The stimulus generator presents observed bit patterns to the Estimator pool, which scores each candidate permutation π proposed by the hypothesizer. Accepted permutations enter the primitive set G and are recycled through composition closure to seed new candidates; rejected permutations are discarded. The loop is built around the Estimator pool (which maintains the observed-pattern memory P and performs the membership test) and integer permutation logic; the microarchitectural realization is described in §4.5.

Candidate generation and proposal prior. Candidates are drawn from S_{144} via two structural priors. A *sparsity prior* makes most proposals move few pixels (per-pixel displacement drawn from a heavy-tailed distribution peaked at zero), encoding that visual symmetries tend to be local. A *closure prior* adds compositions $\pi \circ \pi'$ of accepted primitives to the pool. These are structural choices, not emergent properties: a uniform prior on S_{144} would never propose a meaningful symmetry within any feasible compute budget.

Scoring. For each candidate π, the system tests whether π maps every observed pattern to another observed pattern. The observed-pattern set P is a 144-bit-keyed lookup that the Estimator pool updates incrementally. *In the noise-free discovery regime reported here, this reduces to an incrementally maintained hash table over observed patterns:* π is promoted iff $\pi(p_i) \in P$ for every $p_i \in P$, an exact-membership test. The Brain-CA Estimator's full statistical characterization is the object of the companion submission [4]; we isolate the noise-free discovery dynamics here. Requiring agreement across all $|P|$ patterns drives the operating false-positive rate as $\rho^{|P|}$ against the random-permutation baseline ρ, under an independence approximation.

Closure. Accepted primitives are closed under composition: $\pi_1 \circ \pi_2$ is evaluated for any accepted π_1, π_2. This ensures the discovered set is a *group*, not just a collection, and each accepted primitive seeds further candidates.

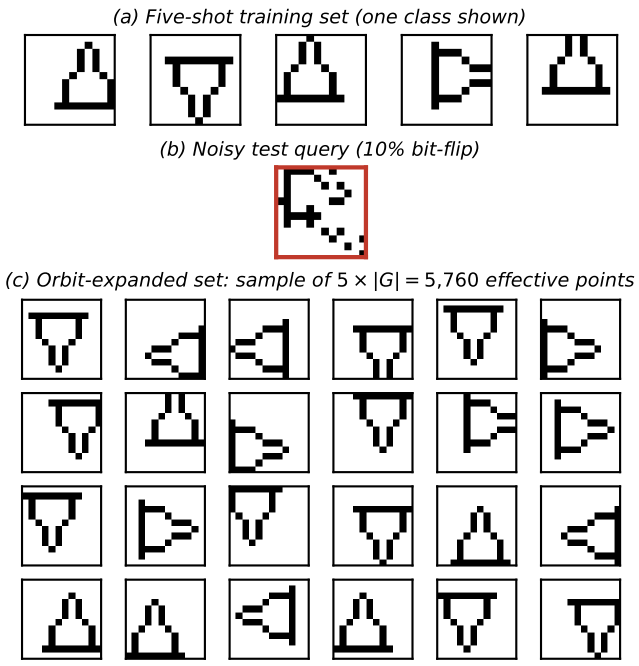

Figure 2. Few-shot orbit expansion mechanism on a 12×12 binary retina (one class, triangle, shown). The five-shot training set (a) covers a few discrete poses. A test query (b) with 10% bit-flip noise destroys local pixel structure, and raw 5-NN against (a) collapses to 28% accuracy. Orbit expansion under the discovered group G produces $5 \times |G| = 5{,}760$ effective training points (sampled in (c)) that densely cover the pose manifold; the noisy query then has a clean orbit-member nearest neighbor of the correct class, recovering 100%.

Termination. The procedure terminates after $K = 200$ consecutive proposals without a new acceptance. K and the sparsity prior's tail parameter are the only discovery-loop hyperparameters; the substrate has no learning-rate, batch-size, gradient-step, or trained-weight hyperparameters.

4 Results

4.1 Few-shot classification on a 12×12 binary retina

We present the system with five labeled examples per class on a 4-class shape dataset (circle, square, triangle, cross), each example a different pose on the retina (Fig. 2a). We then *expand* each example into its orbit under the discovered primitive group, yielding $5 \times |G|$ effective training points per class, where $|G|$ is the size of the discovered group. We classify held-out test examples by nearest-neighbor in the expanded set, using Hamming distance.

Table 1 reports the result. Both reach ceiling on clean data. Under 10% bit-flip noise, raw nearest-neighbor collapses to 28%: noise dominates Hamming distance to any single clean pose. Orbit expansion restores 100% because the noisy query's nearest neighbor in the expanded set is a clean orbit member of the correct class. The few-shot budget is unchanged; only the inductive bias has changed.

Table 1. Five-shot classification accuracy on a 12×12 binary retina under 10% bit-flip noise. The hand-coded row applies the explicit $\mathbb{Z}_{12}^2 \rtimes D_4$ prior; the discovered row uses the bootstrap-discovered group. The two are equivalent in this experiment because the discovered group matches the hand-coded group element-by-element (§4.2).

Method	Clean	Noisy
Raw 5-shot nearest-neighbor	100%	28%
Hand-coded $\mathbb{Z}_{12}^2 \rtimes D_4$ orbit expansion	100%	100%
Discovered orbit expansion (ours)	100%	100%

4.2 Search-space compression vs. hand-coded priors

The symmetric group S_{144} has $144! \approx 10^{248}$ elements; the discovered group G has $|G| = 1{,}152$. The 245-orders-of-magnitude ratio is rhetorical but a strawman: no practitioner searches S_{144} uniformly. The honest comparison is the engineer's hand-coded prior, the toroidal-translation group composed with D_4, with exactly $144 \times 8 = 1{,}152$ elements. The discovered group recovers this prior element-by-element (verified by enumeration against the hand-specified group): the system arrives at $\mathbb{Z}_{12}^2 \rtimes D_4$ without being told that rotations, reflections, or translations are what to look for. The contribution is discovery, not compression: we recover by data what is otherwise specified by hand.

4.3 Modality generality

Is the discovered group overfit to the discovery stimuli? Bootstrapping from circle, square, and cross classes only, we test whether each $\pi \in G$ maps observed triangles to observed triangles. The result is 100% preservation: every discovered primitive maps observed triangles to observed triangles, despite no triangle in the discovery set. The primitives are properties of the modality, not the stimulus.

4.4 Cross-format sanity check: rigid-transformation extraction in an ARC-style format

This is not a result on the Abstraction and Reasoning Corpus [1]; ARC is hard and our setup is small. We test cross-format transfer: on a 3×3 grid, each task presents seven input–output pairs related by an unknown rigid transformation; the system must identify and apply it. We test all eight D_4 elements on the 3×3 grid (identity, three rotations, two axis reflections, both diagonal reflections). The system recovers the correct transformation in 95% of trials (100 per transformation) using the same Estimator-pool loop, no problem-specific tuning; the 5% failures are not concentrated on a single transformation. The discovery dynamics survive the change from a 12×12 retina to a 3×3 grid.

4.5 Microarchitecture

The discovery loop is a tile of parallel Estimator lanes over a shared observed-pattern memory P. Each lane draws a proposal π from a sparsity-biased LFSR hypothesizer, reads each $p_i \in P$, applies π as a read-path index transformation (no data movement), and probes P for membership of $\pi(p_i)$; a candidate is promoted when all $|P|$ probes hit, else dropped. The pattern store is a 144-bit-keyed CAM, feasible at $|P|$ in the hundreds (hashed lookup is a drop-in alternative).

Weights are indices, not tensors. The discovered group G is a table of $|G| = 1{,}152$ permutations, each a 144-element 8-bit index map, ∼165 KB packed. This replaces the weight tensor of a conventionally trained model with a small index ROM. There is no weight streaming during inference: the discovered group is a one-time index ROM, and inference performs orbit-expansion against a small exemplar store rather than streaming a large weight tensor.

Discovery and inference share the datapath. During discovery, lanes apply candidate permutations to test membership against P. During inference, the same lanes apply elements of the discovered G to a query pattern to generate its orbit for nearest-neighbor scoring. The Estimator pool that learns the modality is the pool that exploits it; no separate inference engine is required.

Cycle budget. Each lane evaluates one candidate in $O(|P|)$ probes with early-exit on first miss. The reported runs terminate after approximately $4{\times}10^4$ proposals (including closure-generated candidates); at $L = 64$ parallel lanes (a single chiplet-tile parallelism), the synthesized 96 MHz clock, and $|P|$ in the low hundreds, total wall-clock discovery time is on the order of 3 ms for the 12×12 retina. Inference orbit expansion is 1,152 permutation applications per query, well below 1 ms at modest L. The Estimator is a *micro-architectural primitive* (Eq. 1); end-to-end power and area characterization on a full tile remains future work.

5 Relation to Neural Rendering and Embodied AI

Neural rendering (NeRF [9], Gaussian splatting [7], learned scene representations) produces the visual stream but does not learn its structure. A robot or wearable seeing a Gaussian-splatted reconstruction still needs to learn how objects appear under the modality's transformations, from few examples. Primitive discovery is complementary, not competitive.

Two scoped follow-ups would close that gap: a *rendered-silhouette bootstrap* (run the loop on quantized multi-view renders, expecting the discovered group to reflect quantization-survived pose equivalences) and a *sensor-geometry adaptation* (re-tile the retina as hexagonal, foveated, or event-camera, expecting the discovered group to differ across geometries as the sensor model predicts). The patch-level result is consistent with the loop being agnostic to a stimulus's specific content and sensitive only to its geometric symmetries; whether patch-scale symmetries compose into object-scale invariances is the open question.

6 Scaling Path and Open Problems

Patch-level scaling path. The patch-level result above is a building block. The architectural scaling path it admits is direct: a 1920×1080 frame quantized to a binary front end and tiled into ≈ 14,400 12×12 patches, each served by an inference-time orbit-expansion engine reading a single shared discovered group G (the ∼165 KB index ROM of §4.5). Discovery itself is one-time and per-modality, not per-patch: a single Estimator pool calibrates against the sensor's modality once at boot, and the resulting G is broadcast to every patch lane. The substrate the patch-level result establishes is therefore exactly the per-patch unit a vision tile would replicate; what changes at HD scale is the number of inference lanes, not the discovery loop. This estimate treats patches independently; cross-patch composition for objects spanning patch boundaries is part of the hierarchical-assembly problem and remains open. We do not claim this hierarchical assembly has been built; the present paper establishes the patch-level primitive it depends on.

Beyond binary single-channel patches. The discovery loop assumes symmetries are exact pixel-index permutations, which is correct for a quantized binary patch tier but does not handle continuous transformations (scale from approach, perspective from camera motion) within a patch. These belong above the patch layer, where coarser pose changes appear as discrete equivalences between patch states; color and multi-channel inputs are a more direct extension: one binary retina per channel, sharing the modality group.

Non-toroidal retinas. Physical sensors are not donuts. Toroidal handling keeps translations a true group and the comparison target well-defined; for a planar sensor, translations are partial and the loop's translation-handling needs revisiting, or boundary patches a separate primitive set. This is an open architectural detail, not a barrier.

Discovery-loop robustness. The termination criterion ($K = 200$) is robust empirically: across the reported runs, the discovered $|G| = 1{,}152$ is unchanged for $K \in [50, 1000]$ and the sparsity-tail parameter across $[0.5, 8]$. It is not a completeness guarantee, and our agreement with $\mathbb{Z}_{12}^2 \rtimes D_4$ is suggestive, not a proof. The loop assumes fixed retinal geometry; time-varying sensors are a separate problem.

7 Conclusion

A visual modality's transformation structure is discoverable using a discovery loop with a 48-LUT elemental binary learner. The loop recovers the $\mathbb{Z}_{12}^2 \rtimes D_4$ symmetry of a toroidal 12×12 retina, lifts noisy five-shot classification from 28% to 100%, and transfers across shape classes. This lets an embodied perception pipeline discover its modality's structure on-device, not inherit it from a designer.

References

[1] François Chollet. 2019. On the Measure of Intelligence. arXiv:1911.01547. arXiv:1911.01547 [cs.AI]

[2] Taco S. Cohen and Max Welling. 2016. Group Equivariant Convolutional Networks. In *Proceedings of the 33rd International Conference on Machine Learning*, Vol. 48. PMLR, New York, NY, USA, 2990–2999.

[3] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. 2018. Loihi: A Neuromorphic Manycore Processor with On-Chip Learning. *IEEE Micro* 38, 1 (2018), 82–99.

[4] Jerry Felix. 2026. A Hardware-Native Bit-Serial Learner with Exact Statistical Structure. Workshop on Machine Learning for Computer Architecture and Systems (MLArchSys), ISCA 2026, Raleigh, NC.

[5] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70. PMLR, Sydney, Australia, 1126–1135.

[6] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv:1704.04861. arXiv:1704.04861 [cs.CV]

[7] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4 (2023), 1–14.

[8] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. 2013. Distance-Based Image Classification: Generalizing to New Classes at Near-Zero Cost. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (2013), 2624–2637.

[9] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *European Conference on Computer Vision (ECCV)*. Springer, Cham, 405–421.

[10] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic Routing Between Capsules. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc., Red Hook, NY, USA, 3856–3866.

[11] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical Networks for Few-Shot Learning. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., Red Hook, NY, USA, 4077–4087.

[12] Maurice Weiler and Gabriele Cesa. 2019. General E(2)-Equivariant Steerable CNNs. In *Advances in Neural Information Processing Systems*, Vol. 32. Curran Associates, Inc., Red Hook, NY, USA, 14334–14345.