Banya AI 4

World-First Developmental Curriculum Learning and the Emergence of Axis Structure

Hyukjin Han (bokkamsun@gmail.com) · ORCID 0009-0007-4132-1707

Banya Research Series, Part 4

Abstract

Humans develop a sense of perception, space, and logic before they learn language. This paper trains a language model with a developmental curriculum, a training order that proceeds from easy stages to hard ones, imitating that sequence. Before language, the model first forms sensory axes such as size, distance, and temperature (an axis is a scale that measures one property), and language is then laid on top. The central question is whether, without hand-coding an ontology (a structural map in which concepts occupy their own places), the training order alone leads the filters and embeddings to form an ontological axis structure by themselves. This paper measures the following. First, sensory concepts settle onto axes at 5 to 27 times the random baseline. Second, ordered axes (distance, temperature) are aligned by order (order consistency up to 67 percent), while unordered categorical axes (color, taste) form without order, and the two split cleanly. Third, related concepts cluster together (spatial concepts with each other, sensory concepts with each other). Fourth, the channel filters self-organize into a filter ladder from broad brushes in shallow layers to fine pens in deep layers, with not a single dead filter. In short, the training order itself turns the filters into an ontology. These readouts are obtained with non-invasive probes that do not touch training.

Measured

Every quantitative number in this paper is a measured value obtained in the writing environment (frozen models at each developmental stage; environment in Appendix B). The theoretical background is in a separate axiom document [1], and the claims of this paper rest on the measurements, not on those axioms.

Contents

Terminology Legend

Terms are defined in the tables below. One concept goes by one name only, and the text and figures of all seven parts follow these tables. The in-code column gives the identifier that implements the concept in the enclosed code, and the standard-term column gives the closest name in common use; cells are filled only where they differ.

Terminology legend 1. Engine and credit

Term	Meaning	In code	Standard term
exact adjoint	The backward computation derived by hand as closed forms for every operation; the proper name of this engine and its credit chain	`banya_core.py`	manual back-propagation
adjoint	The partner operation of the forward pass; it carries the gradient exactly backward	`bwd` kernels	adjoint
credit	The loss gradient at a hidden state; the responsibility a parameter bears for the outcome	delta (δ)	gradient
credit chain	The skeleton that walks the cache in reverse, applying each adjoint to carry credit back to the input	`backward`	back-propagation
adjoint agreement	The check that the adjoint formulas produce the same values as finite differences	per-part probes	gradient check
circulant-matrix channel mixing	Channel mixing with a circulant matrix, computed as componentwise products under rfft		circular convolution
relative-distance mixing	Mixing positions with shared coefficients that depend only on distance	`m_mat_w_lag`	Toeplitz mixing
temporal mixing	The umbrella name for any path by which other positions' information flows into the current one		token mixing
mixing heads	The number of parallel heads a mixing operation is split into		attention heads
filter ladder	The convolutional channel transform whose filter scale varies with depth, from a few wide filters to many narrow ones	`m_mat_w_filter`	convolutional filterbank
window	The token positions the model sees at once; as a measurement unit, a fixed-length text slice	`T`	context window
probe	The verification script that replays every measured figure together with its target value	`probe` folder	reproduction script
frozen model	A developmental-stage model whose training is stopped and fixed	`model` npz	frozen checkpoint
holdout	Evaluation text never used in training		held-out set

Terminology legend 2. Bi-token and the gate

Term	Meaning	In code	Standard term
bi-token	The structure that gives one token an orthogonal pair of embeddings, a data plane and an operation plane	`USE_BITOKEN`	
data plane	The embedding plane carrying a token's content; it enters at the add slot	`m_mat_w_data_axis`	token embedding
operation plane	The embedding plane carrying a token's operation instruction; it enters only at the multiply slot	`m_mat_w_operate_axis`	
add slot, multiply slot	The residual-addition position where the data plane lands and the gate-multiplication position where the operation plane acts		residual connection, gating
gate	The valve that sets each channel's pass-through between 0 and 1; the previous token's operation plane is injected into it	`m_mat_w_gate`	gating
one-step shift	The one-position displacement by which the previous token's operation plane is injected into the next position's gate		
operation-as-weight	The phenomenon in which operation-plane embeddings emerge as weights without any labels		
consistent operator	An operator that acts in a consistent direction on any data		
direction consistency	The mean cosine of the directions in which one operation bends different data; the quantitative sign of operator emergence	`p_direction_consistency`	
rotation-operator gate	The gate that reads the operation plane as Euler rotation angles; abbreviated in the text as the rotation gate	`paper7_rotation.py`	
primitive operation	An operation the gate can express directly in a single-hop instruction		primitive
depth-excluded discrimination	The deliberately biased measurement that restricts depth to one block so the gate is the only remaining variable	Part 7 probe	
self-feedback	The scoring mode that feeds the model's own output back as input and scores the final state		autoregressive rollout

Terminology legend 3. Rumination and development

Term	Meaning	In code	Standard term
Rumination	Self-organizing learning that pulls embeddings toward their neighbors without computing any gradient	Part 2 probes	self-organization
same-kind group	The set of close-in-meaning concepts gathered from a seed by nearest-cosine traversal		cluster
centroid pull, decay	The two forces of Rumination: pulling toward the mean and restoring toward the original; equilibrium emerges from their balance	`ALPHA`, decay rate	
World-first	The curriculum order that develops sensory, spatial, and logical axes before language	`banya_world_data`	curriculum learning
developmental stage	A checkpoint stage defined by step count and named after human growth		
axis	A property scale formed as a direction inside the embedding: sensory, ordinal, categorical		representation axis

Terminology legend 4. Vocabulary and metacognition

Term	Meaning	In code	Standard term
syllable atom	The syllable-level token forming the bottom layer of the vocabulary	`AtomTokenizer`	character-level token
bundle	A concept token made by binding syllable atoms into the dictionary; used only as a vocabulary unit		subword
cache-dictionary tokenization	The two-layer vocabulary system that puts a bundle dictionary on top of the atom layer	Part 5 probes	contrast with subword tokenization
certain, vague, unknown	The three knowing states separated by the model's own output distribution	pmax, entropy	uncertainty estimation
routing	Assigning an answer's processing path according to the state		routing
reinterpretation	The procedure that re-runs a vague hidden state together with its context to lift it to certain	Part 6 probes	
similarity consolidation	The axis along which concepts are organized by data-plane similarity		
inference traversal	The axis along which inference is carried forward on the operation plane		
self-utterance	The target state that weaves similarity consolidation and inference traversal so the model continues speech by itself		
Banya framework	The fifteen-axiom system forming the series' background; not the ground of the claims in the text		

Symbol Legend

This defines the symbols used by both the equations and the body text of this part. Local symbols used only inside a single equation are listed in the symbol line under each equation.

Symbol	Name	Definition and meaning	Type
A	axis agreement	The cosine similarity between each contrast-pair embedding difference and the mean axis direction, averaged over all of P. Computed per axis; measured at 5 to 27 times the random baseline of 0.027	scalar
S	order consistency	The probability that the next grade ranks high in the output distribution on a grade ladder. Computed per axis; measured at 33 to 67 percent for ordered axes and 0 to 4 percent for categorical axes	scalar
P	contrast-pair set	The set of word pairs that differ only along the given axis (contrast pairs). The axis agreement A is averaged over this whole set	set
$\|P\|$	number of contrast pairs	The size of the set P, that is, the number of contrast pairs. Used as the normalizing denominator that turns the sum into a mean	integer
(a, b)	contrast word pair	A word pair belonging to P. a and b are two words that differ only along the given axis	ordered pair
E_a	word embedding	The embedding vector of word a. E_b is the embedding of word b of the same form	vector
$E_a - E_b$	embedding difference	The difference of the two word embeddings, a vector giving the direction of the axis	vector
$\bar{v}$	mean axis direction	The mean direction vector of the axis. How well the contrast-pair differences agree with this direction reads out axis formation	vector
cos	cosine similarity	The function measuring directional agreement as the cosine of the angle between two vectors. The centroid cosines of Section 4.3 use the same measure	function
$\sum_{(a,b)\in P}$	summation over pairs	The summation operator adding the cosine similarities over all contrast pairs in P	operator
Pr$[\cdot]$	probability	The probability of the event in brackets. Used in the definition of the order consistency S	function
$\times$	times	The multiplication sign marking the multiple over the random baseline, used in the Section 4.1 table and the Figure 4-1 caption	operator

1. Introduction

Building an ontology by hand, with humans coding the concept map directly, is slow and always leaves gaps. This paper takes the opposite approach. Instead of coding in an ontology, we align the training order with the human developmental order. The hypothesis is that if sensory, spatial, and logical axes are developed before language, the filters form those axes by themselves and play the role of an ontology. The formal background of this view is in a separate axiom document [1], but this paper verifies only by measurement whether the hypothesis holds in actual training.

Training uses the engine of Part 1 [2], which computes backpropagation without automatic differentiation. Readout is done with non-invasive probes that never stop training. We read whether an axis has formed by comparing, against a random baseline, how well the embedding-difference directions of the axis's contrast pairs (word pairs that differ only along that axis) agree with one another, and we read whether order has been established by inserting a grade ladder and checking whether the next grade rises in the distribution.

2. Related Work

Curriculum learning [5] arranges training from easy examples to hard ones, but it is mostly a difficulty ordering and does not form sensory axes before language. Efforts that compete on small-data developmental training also address the ordering of the language corpus itself, not the prior formation of world axes before language. Mechanistic interpretability, the study of dissecting what a neural network represents internally, reads representations after the fact. This paper differs in that it treats the training order (world-first) as the independent variable and reads out the emergent axis structure non-invasively.

3. Method

The developmental curriculum first learns sensory, spatial, and logical axes in a pre-language stage (world), and then lays the language of the toddler and toddler 2 stages on top. Each stage is frozen as a named checkpoint (birth, baby, toddler, and so on) so that axis formation can be measured stage by stage.

Eq. 3-1. Axis agreement and order consistency

$$A(\text{axis}) = \frac{1}{|P|} \sum_{(a,b)\in P} \cos\bigl(E_a - E_b,\ \bar{v}\bigr)$$

$$S(\text{axis}) = \Pr\bigl[\text{next grade ranks high in the distribution}\bigr]$$

Formula

Symbols: P is the set of word pairs that differ only along the given axis, $E_a - E_b$ is the difference of the two word embeddings (the direction of that axis), $\bar{v}$ is the mean direction of the axis, cos is cosine similarity, A is the axis agreement, and S is the order consistency.
How it works: if the word pairs of the same axis lie in the same direction in the embedding, A is large. If it is several times larger than the random baseline, the axis really exists in the embedding. S is the probability that the next grade rises to the top of the distribution on a grade ladder, measuring whether the axis is aligned by order.
In practice: the size axis, for example, is measured with grade pairs such as "small, large." If the difference directions of these pairs agree, the size axis has formed, and if "slightly large" rises in the distribution after "small," the axis is aligned by order.

4. Results

4.1 Sensory axes form at 5 to 27 times the random baseline

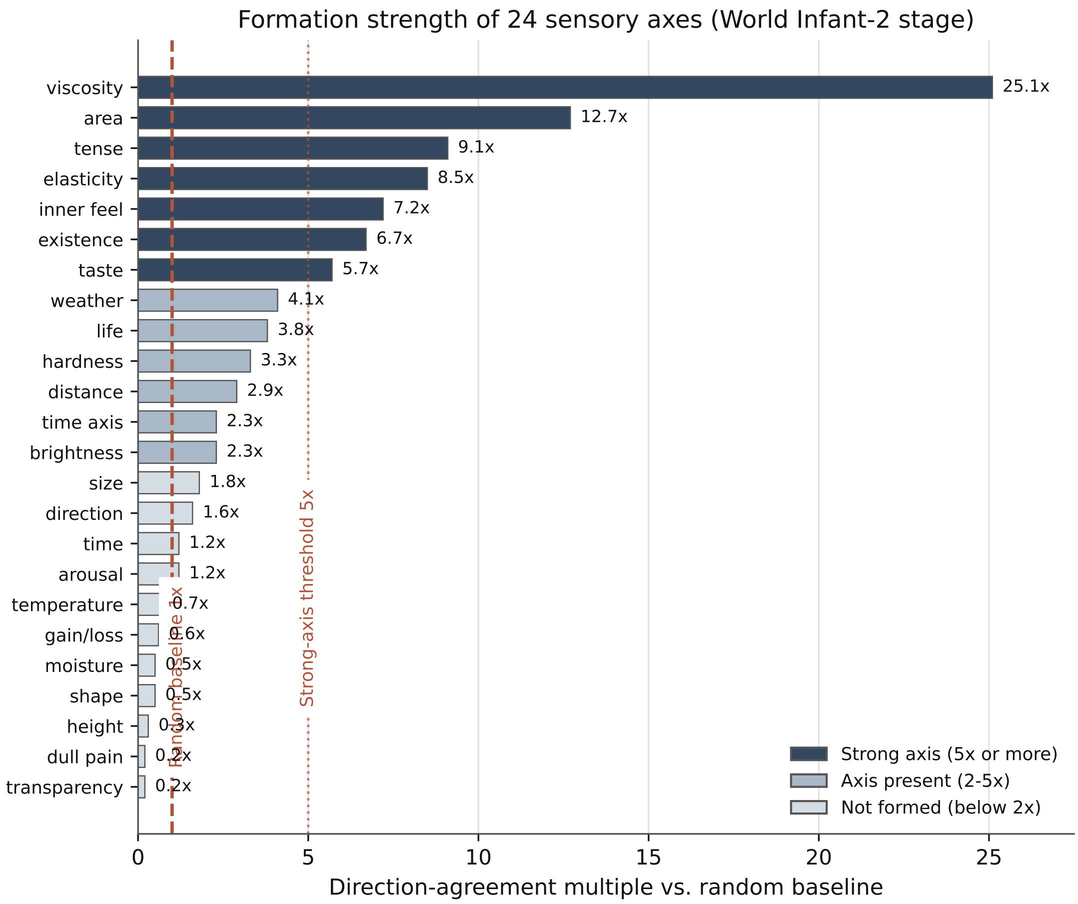
Measured

Measuring axis agreement on the frozen models of the developmental stages (random baseline 0.027), several sensory axes form strongly. Multiples over the baseline for axes that stay strong from the birth stage through the toddler stage:

Axis	vs. baseline	Axis	vs. baseline
Viscosity	24 ~ 27×	Tense	8 ~ 10×
Area	11 ~ 13×	Existence	7×
Weather	4 ~ 11×	Taste	6×
Elasticity	4 ~ 9×	Interoception	5 ~ 7×

At every stage, the number of strong axes stays stable at 7 to 8. Not every axis forms (some axes remain unformed), and they do not strengthen uniformly, but the fact that sensory concepts settle onto axes markedly more strongly than random is consistent across all stages.

Figure 4-1. Sensory axis agreement (multiple over random)



Several sensory axes form at 5 to 27 times the random baseline. Concepts settle onto axes even though none were coded in directly. Bar length is the multiple over random; the vertical line at 1× marks the random level.

4.2 Ordered axes form with order, categorical axes without

Some axes are ordered (distance, temperature) and some are unordered categories (color, taste). Measurement shows the two split cleanly.

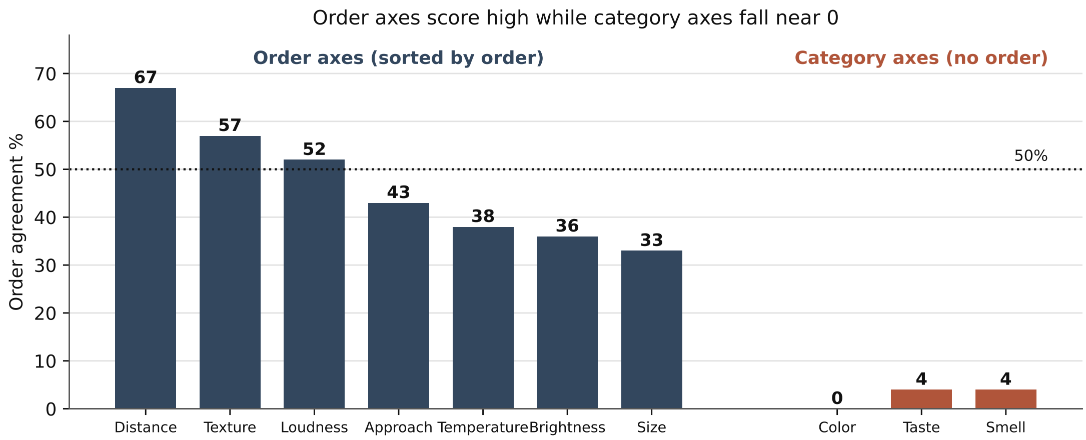
Measured

Measured order consistency (the fraction of cases where the next grade rises in the distribution):

- Ordered axes : distance 67 percent, texture 57 percent, loudness 52 percent, approach 43 percent, temperature 38 percent, brightness 36 percent, size 33 percent.
- Categorical axes : color 0 percent, taste 4 percent, smell 4 percent.

Ordered axes show high order consistency at 33 to 67 percent, while categorical axes show none at 0 to 4 percent. The model arranged ordered properties by order and kind-only properties without order. This split is itself evidence that axis structure has emerged.

Figure 4-2. Order consistency of ordered and categorical axes



Ordered axes (distance 67 percent, etc.) are aligned by order, and unordered categorical axes (color 0 percent, etc.) form without order. The model arranged each property according to its kind. This clean split is direct evidence of emergent axis structure.

4.3 Related concepts cluster together

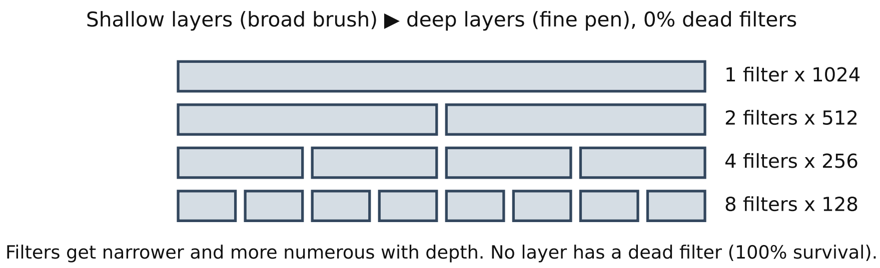
Relations between axes also emerge. Measurement shows spatial concepts clustering with each other (centroid cosine 0.84 between distance and approach, 0.63 between direction and distance) and sensory concepts with each other (0.94 between smell and arousal, 0.93 between taste and smell). Related axes gather close together in the embedding, settling like the upper-level groupings of an ontology.

4.4 Channel filters self-organize into a filter ladder with no dead filters

Measured

Measuring the channel filters, from shallow layers to deep layers the filters form a filter ladder from wide and few (broad brushes) to narrow and many (fine pens) (from 1 filter of length 1024 to 32 filters of length 32). This structure is the filter ladder of Part 3 [4]. The fraction of dead filters (filters whose response is so weak that they are effectively unused) is 0 percent in every layer. The fraction of input-dependent channels in the gates goes from 94 percent in shallow layers to 61 percent in deep layers, so most channels respond to the input.

Figure 4-3. Filter ladder and survival of the channel filters



The channel filters self-organize into a filter ladder from broad brushes in shallow layers to fine pens in deep layers. Dead filters are 0 percent in every layer, so developmental training wastes none of the filter capacity.

Reproduce

The axis-structure numbers of Section 4 are reproduced with the following single line after completing the preparation in Appendix C.

```
cd probe && python3 paper4_world_dev_probe.py
```

In the output, the sensory axis multiples, the order consistency of ordered and categorical axes, and the filter ladder and survival rates correspond to the numbers in the text. This probe needs no corpus, so step 4 of Appendix C may be skipped.

5. Limitations and Next Measurement Tasks

The reference class of the baseline is a single developer, a self-built exact-adjoint backpropagation engine, and one consumer GPU. The items below are not defects but the next measurement tasks.

- Order control : the next task is to place side by side a language-first control that reverses only the training order on the same data and steps, isolating the order effect. This control is needed to pin down order as the cause.
- Late-stage abilities : higher abilities such as question answering or self-expression still measure low on the frozen toddler-stage model measured here. They belong to later stages and are treated in Part 6. This paper does not claim those abilities.
- Standard interpretability comparison : a side-by-side comparison with standard interpretability techniques, beyond the self-built probes, is a next task.

6. Conclusion

This paper showed by measurement that when the training order is aligned with the human developmental order, without coding in an ontology, the filters and embeddings form an axis structure by themselves. Sensory axes form at 5 to 27 times random, ordered axes split from categorical ones by aligning with order while the latter form without it, related concepts cluster together, and the channel filters self-organize into a filter ladder with no dead filters. The hypothesis that the training order itself turns the filters into an ontology is confirmed, at the measured stages, by the emergence of axis structure. Confirming higher abilities and the order control remain as next tasks. This order — the world before language — was not first drawn by this paper; it comes from an older blueprint that the axioms of the Banya framework point to.

7. Series Preview

Series preview

- Part 5 — Cache-dictionary tokenization and dynamic embedding : treats the vocabulary layer on which this paper's axes rest, as a two-layer structure of syllable atoms and a bundle dictionary.
- Part 6 — Control signals emerging from self-distribution and geometric monitoring : treats the higher abilities this paper deferred to later stages (routing of certain, vague, unknown; learning motivation) with measured statistics.
- Final part (unfinished) — Combining similarity consolidation with inference traversal : self-utterance that weaves data-plane similarity consolidation (Part 2 [3]) with operation-plane inference traversal. Not yet treated, as the final assembly is incomplete.

References

[1] Han, H. (2026). Banya Framework: An Axiom-Based Science Mining Engine for Deriving Fundamental Physical Constants. Zenodo. https://doi.org/10.5281/zenodo.20301304 . (Background worldview of this series; prior axiom document)

[2] Han, H. (2026). A Language-Model Training Engine Using Hand-Derived Closed-Form Exact Adjoints Without Automatic Differentiation. Banya Research Series, Part 1. Zenodo. https://doi.org/10.5281/zenodo.21385447

[3] Han, H. (2026). Gradient-Free Self-Organization of Concept Embeddings by Frequency-Proportional Rumination. Banya Research Series, Part 2. Zenodo. https://doi.org/10.5281/zenodo.21385516

[4] Han, H. (2026). Orthogonal Data-Operation Tokens and Convolutional Scale-Depth

Structure: Label-Free Emergence of Operation-as-Weight. Banya Research Series, Part 3. Zenodo. https://doi.org/10.5281/zenodo.21385676

[5] Bengio, Y., Louradour, J., Collobert, R., Weston, J. (2009). Curriculum Learning. ICML.

Appendix A. Developmental Stages

Table A-1. Frozen models of the developmental stages used for measurement

Stage	Steps	Stage	Steps
Birth	10k	Toddler	70k
Baby	30k	Toddler 2	110k

Appendix B. Measurement Environment

Table B-1. Computer specifications used for measurement

Item	Specification
GPU	NVIDIA GeForce RTX 3090 Ti (24 GB)
Operating system	Ubuntu 24.04.4 LTS
Software	Python 3.12.3, numpy 2.5.1, scipy 1.18.0, cupy 14.1.1, CUDA 13.3.1
Measured objects	Embeddings, axis contrast pairs, and channel filters of the frozen models at each developmental stage

The measured numbers in Section 4 were obtained in the environment above.

Appendix C. Reproduction Guide

The measured numbers of this paper are reproduced with the public reproduction package. Follow the five steps below in order.

1. Requirements. An NVIDIA GPU with CUDA and Python 3.12 are required. Install the Python packages in the unpacked folder with the following. For cupy, use the distribution matching the installed CUDA version (e.g., cupy-cuda13x for CUDA 13).

```
pip install -r requirements.txt
```

2. Get the code. Download and unpack the zip bundle. To get it as a repository, use the github.com/ubmscoin/banya/banya_ai_paper_code folder.

```
wget https:
//ubmscoin.github.io/banya/banya_ai_paper_code/banya_ai_paper_code.zip
unzip banya_ai_paper_code.zip && cd banya_ai_paper_code
```

3. Get the models. The 3 frozen models (471MB in total) are on Zenodo at doi.org/10.5281/zenodo.21383724. The following single line downloads them and verifies integrity (sha256, file-fingerprint check) at the same time. To do it by hand, download the three files from the record above into the model folder and verify with `sha256sum -c checksums.sha256`.

```
cd model && bash download.sh && cd ..
```

4. Build the corpora. The corpora are produced by generators, with no raw-text distribution. The following single line generates all 23 corpora into the banya_world_data folder.

```
cd corpus_build && bash build_all.sh && cd ..
```

5. Reproduce the measurements. The probes in the probe folder reproduce the numbers of each section. The numbers of this paper are covered by the following probe. Target values are printed alongside the output, so it can be checked directly against the text.

Probe	**Numbers reproduced**	**Output to check**
paper4_world_dev_probe.py	Section 4 axis formation and filter ladder	Axis multiples, order consistency, and filter-ladder numbers are printed with their target values. No corpus needed

The folder layout of the package and per-file descriptions are on the index page.

Banya Research Series, Part 4 (world-first development, emergent axis structure). All numbers were measured in the environment of Appendix B. The background worldview is cited as the prior axiom document [1] but is not the basis of this paper's claims. Parts 1 through 3 are cited. Late-stage abilities such as question answering and selfhood are still low at the measured stages and are treated in Part 6. Cache-dictionary tokenization is treated in Part 5.