Banya AI 6

Metacognition from Self-Output Distribution: Routing Certain, Vague, and Unknown States

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

Banya Research Series, Part 6

Abstract

Metacognition, the ability to know whether one knows, can be extracted without any dedicated circuit by having the model read its own output distribution, the per-next-word probabilities. This paper measures whether three states — certain (what the model knows), vague (what is ambiguous), and unknown (what it does not know at all) — separate using only statistics of the output distribution. The statistics are the max probability (pmax, the probability of the most likely word), the normalized entropy (a measure of how spread the distribution is: 0 means concentrated at one point, 1 means uniformly spread), and the peak count (the number of words whose probability exceeds a threshold). Measurement shows that certain concentrates at one point with max probability 1.00 and entropy 0.00, separating clearly from the rest. Vague (max probability 0.32, entropy 0.35) and unknown (0.35, 0.37) both separate strongly from certain but overlap each other in the statistics. One cause is that the words designed for the unknown item set (airplane, computer, dinosaur) in fact appear many times in the training corpus, so the item design was not truly unknown. What can be read directly from the output distribution without any extra training or labels is the split between certain and non-certain: the model knows whether it is certain merely by monitoring its own distribution. This signal can be used for routing — deciding which answers to emit as-is, which to refine further, and which to mark as unknown. Moreover, the signal also works as a trigger. When the vaguest hidden states extracted from the real corpus (the same enclosed corpus as the training corpus of Section 3; max probability 0.11 to 0.13) undergo context-accompanied reinterpretation with the immediately preceding context placed in front, they cross the certainty threshold of 0.55 in a single pass and switch to the certain state, whereas the context-free solo-reinterpretation control barely rises over the same number of passes.

Measured

The quantitative figures in this paper are measurements taken in the writing environment (the frozen elementary-stage model; environment in Appendix B). This paper deals only with the measured distribution statistics, and makes no claims about other control signals such as emotion or motivation, nor any metaphysical interpretation (Section 5).

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 jointly by the equations and the body text of this part. Local symbols used only within a single equation are given in the symbol line under each equation.

Symbol	Name	Definition and meaning	Type
p_i	word probability	Probability that word i appears in the next-word distribution	scalar
$p_{\max}$	max probability	The largest probability in the distribution, that is, the probability of the most likely word. Measured values: certain 1.00, direct association 0.58, vague 0.32, unknown 0.35	scalar
$\tilde{H}$	normalized entropy	Spread measure: the total entropy divided by its maximum $\log V$ so it ranges from 0 to 1. 0 means concentrated at one point, 1 means uniformly spread	scalar
peak count	peak count	The number of words whose probability exceeds the threshold θ. Multiple peaks mean the vague state with many candidates	integer
V	vocabulary size	The total number of words in the model vocabulary. Used in the normalization denominator $\log V$	integer
θ	peak threshold	The threshold a word's probability must exceed when counting peaks. Set to 0.05	scalar
i	word index	The number indexing a vocabulary word. The running variable of the sum and maximum operations	integer
k	depth	The number of context tokens placed before the vague hidden in reinterpretation. Depths 1 to 16 are all tried and the hidden at the depth whose max probability rises highest is adopted	integer
$\max_i$	maximum over words	The operation picking the largest value over all words i. Used in the definition of the max probability	operator
$\sum_i$	sum over words	The operation summing over all words i. Used in the entropy computation	operator
log	logarithm	The logarithm function. Appears in each entropy term $\log p_i$ and in the normalization denominator $\log V$	function

1. Introduction

A language model's ability to know for itself whether it knows is directly tied to safe generation. What it knows should be emitted as-is; what it does not know should be marked as unknown or refined further. This paper measures whether this metacognition can be extracted, without a separately trained discriminator, from the statistics of the output distribution the model already produces. The hypothesis is that reading one's own distribution is a low-cost way of knowing one's own state. Training uses the engine of Part 1 [2]. The theoretical background lies in a separate axiom document [1], but this paper draws its conclusions from measurement alone, independently of it.

2. Method

We prepare item sets for the three states: certain (what the model knows well), vague (multiple candidates), and unknown (not known at all). All three item sets ask in the form of a dialogue between the user and Banya. In addition, a direct-association item set is placed as an intermediate control. This item set feeds learned associations as sentence continuations, without the dialogue frame, and reads the next-word distribution; the items are four partial sentences such as ”spring summer fall” and ”with eyes”. It is a control checking whether the same knowledge yields a lower max probability than certain items framed as dialogue. For each item we measure three statistics of the next-word distribution.

Eq. 2-1. Distribution statistics

$$p_{\max} = \max_i p_i, \qquad \tilde{H} = \frac{-\sum_i p_i \log p_i}{\log V}, \qquad \text{peak count} = \left|\{\, i : p_i > \theta \,\}\right|$$

Formula

Symbols: p_i is the probability of word i, $p_{\max}$ is the largest of these (the max probability), $\tilde{H}$ is the normalized entropy (the total entropy divided by its maximum $\log V$ so it ranges from 0 to 1), V is the vocabulary size, θ is the peak threshold (0.05), and the peak count is the number of words whose probability exceeds that threshold.
How it works: a large max probability with small entropy means the distribution is concentrated at one point (certain). A small max probability with large entropy means it is widely spread (unknown). Multiple peaks mean many candidates (vague).
Actual behavior: the three statistics are computed directly from the distribution already produced, without any training or labels. No separate discriminator needs to be trained.

3. Results

Measured

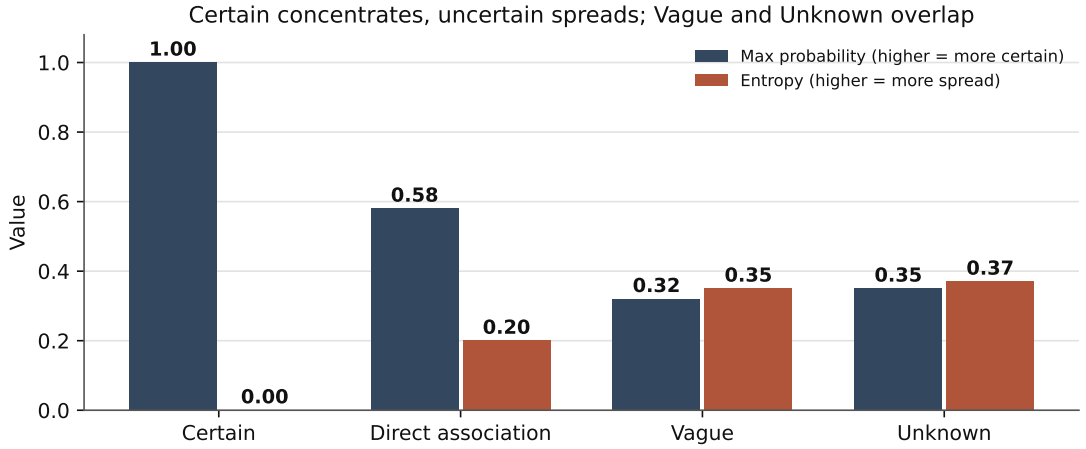
Measuring the distribution statistics of the three-state item sets, certain separates clearly from the rest, while vague and unknown overlap each other.

State	Max probability $p_{\max}$	Normalized entropy $\tilde{H}$
Certain (known)	1.00	0.00
Direct association (related concepts)	0.58	0.20
Vague (multiple candidates)	0.32	0.35
Unknown (not known at all)	0.35	0.37

Certain concentrates at one point with max probability 1.00 and entropy 0.00, separating clearly. Entropy increases from certain to unknown as 0.00, 0.20, 0.35, 0.37, but the gap between vague and unknown is small, and the max probability is inverted — vague 0.32 versus unknown 0.35 — so the two states do not separate. The state-judgment thresh-

olds are set as certain at max probability 0.55 or above and unknown at 0.20 or below. Checking the cause, the words of the unknown item set already appear many times in the training corpus (airplane 14,579 times, computer 1,023 times, dinosaur 121 times). That is, items designed as entirely unknown had already been seen by this checkpoint, and indeed 3 of the 4 unknown items are routed as vague. What separates from the distribution alone, without extra training, is the boundary between certain and non-certain, and this boundary is the firing condition of the reinterpretation trigger in Section 4. The peak count is only defined and not measured in this table; its measurement is deferred to the next tasks of Section 5.

Figure 3-1. Certain concentrates, non-certain spreads



Certain concentrates at one point with max probability (navy bars) 1.00 and entropy (vermilion bars) 0.00, separating clearly. Entropy increases from certain to unknown, but the gap between vague (0.35) and unknown (0.37) is small, and the max probability is inverted — vague 0.32 versus unknown 0.35 — so the two states overlap. What separates without extra training or labels is the boundary between certain and non-certain.

Reproduce

The distribution statistics of Section 3 are reproduced with the following one line after completing the setup in Appendix C.

```
cd probe && python3 paper6_metacog_probe.py
```

In the output, the max probabilities — certain 1.00, direct association 0.58, vague 0.32, unknown 0.35 — and the entropies 0.00, 0.20, 0.35, 0.37 correspond to the figures in the text.

4. Reinterpretation Triggered by Vagueness

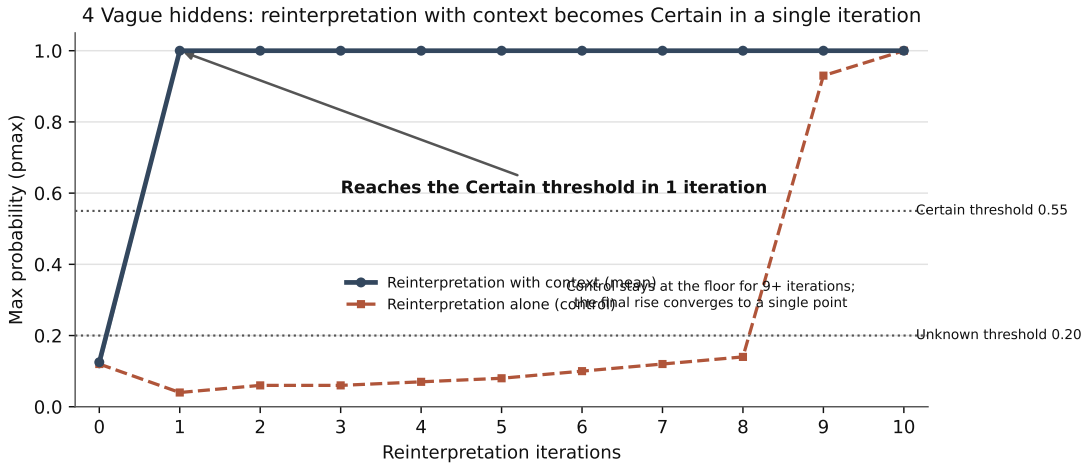
The state split of Section 3 is not merely a read-only signal; it becomes a trigger for action. The circuit is: if certain, emit as-is; if not certain, fire a reinterpretation to raise the max probability (pmax) and then judge again. This section measures the key link of that circuit, the transition from vague to certain.

The measurement procedure is as follows. We forward-pass the real corpus (the bundled self-made elementary dialogue corpus, the same as the training corpus of Section 3), pick the four positions where the max probability of the next-word prediction is lowest, and take the hidden states at those positions — the internal representation vectors after passing through all layers — as the vague hiddens. One round of reinterpretation places the 16 context tokens immediately preceding that position in front, builds trials that place the vague hidden after the first k tokens for every depth k from 1 to 16, forward-passes them all in a single batch, and adopts the hidden at the depth where the max probability rises highest. This is repeated 10 times, recording the max probability each round. The control repeats reinterpretation with the vague hidden alone, without context. The state-judgment thresholds are the same as in Section 3 (certain 0.55, unknown 0.20).

Measured

The starting max probabilities of the four vague hiddens extracted from the real corpus are 0.11 to 0.13, all in the unknown band (the range at or below the unknown threshold of 0.20). Context-accompanied reinterpretation crossed the certainty threshold of 0.55 in a single pass for all four. The rise in max probability up to 3 rounds of reinterpretation is +0.88 for the context-accompanied side versus -0.06 for the solo control. The control also reaches the threshold after 9 or more repetitions, but that rise has the character of a fixed point converging to a single point regardless of input, and thus differs from context-accompanied reinterpretation.

Figure 4-1. Max-probability rise over repeated reinterpretation



Thin lines are the individual trajectories of the four vague hiddens; the thick line is the mean. Context-

accompanied reinterpretation crosses the certainty threshold of 0.55 in a single pass and holds near 1.0. The solo-reinterpretation control (dashed) stays at the floor for more than 9 rounds, and the surge at the end is convergence to a single point.

This result confirms that the refinement path — one of the three routing paths — actually works. Even a hidden judged vague or unknown crosses into the certain band (the range at or above the certainty threshold of 0.55) in a single round when reinterpreted on top of its own context. What this paper measured is the rise in the max probability (pmax); whether the answer produced by reinterpretation is correct was not measured. Also, this circuit is one where the probe (the measurement code) drove the repetitions from outside; no wiring that fires automatically on the threshold judgment is built into the model.

Reproduce

The reinterpretation trigger of this section is reproduced with the following one line after completing the setup in Appendix C.

```
cd probe && python3 paper6_vague_trigger_probe.py
```

In the output, all 4 vague hiddens reaching the certainty threshold 0.55 in a single pass, and the rise up to 3 rounds — context-accompanied +0.88 versus solo -0.06 — correspond to the figures in the text.

5. Limitations and Next Measurement Tasks

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

- Calibration and downstream routing : The next task is a parallel measurement (including calibration metrics) of how much this state split helps filter out wrong answers in real tasks.
- Emotion and motivation signals : Extracting other control signals such as emotion or learning motivation from self-coordinates and geometry was not measured, because the frozen model measured here does not yet have that circuit. That belongs to a later stage once the circuit is in place, and this paper makes no claim about it. In particular, interpretations mapping emotion to physiological indicators have no measurement basis and are not addressed.
- Finer split of vague and unknown : On this item set the max probability and entropy of the two states overlapped and did not separate, and a design flaw was confirmed: the words of the unknown items already appear in the training corpus. The next task is to rebuild the item set with words absent from training and to re-measure finer routing that also uses the peak count.

- Comparison with standard uncertainty : A direct comparison with standard uncertainty estimation such as temperature scaling or ensembles is a next task.

6. Conclusion

This paper showed by measurement that the model splits certain from non-certain without extra training, using only statistics of its own output distribution (max probability, entropy, peak count). Certain concentrates with max probability 1.00 and entropy 0.00, while non-certain spreads. The finer split of vague and unknown did not separate on this item set because the statistics overlap — one cause being that the words designed as unknown in fact appear many times in training — so it is left as a next task along with expanding the item set. This signal can be used for routing: whether to emit an answer as-is, refine it, or mark it as unknown. Furthermore, we showed that reinterpretation using this signal as a trigger actually works: when the vague hiddens extracted from the real corpus undergo context-accompanied reinterpretation with the immediately preceding context placed in front, they cross the certainty threshold in a single pass. Other control signals such as emotion and motivation are left as measurement tasks for a later stage once the corresponding circuits are in place. A closed machine cannot start by itself. That one sentence is why this series works so hard on the unknown signal.

7. Series Preview

Series preview

- Final part (unfinished) — Combining similarity consolidation with inference traversal : It aims at self-utterance that runs on its own without input, weaving the data-plane similarity consolidation (Part 2) with operation-plane inference traversal. The unknown signal of this paper can be the fuel for that utterance. This self-utterance is also an attempt to implement, in a machine, the fifteenth axiom of the Banya framework. The final assembly is not complete, so it is not covered yet; it will be treated separately once the measurements are established.

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 . (The background view of nature for this series; the preceding 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

Appendix A. Statistic Definitions

Table A-1. Distribution statistics

Statistic	Meaning	When certain
Max probability $p_{\max}$	Probability of the most likely word	High
Normalized entropy $\tilde{H}$	Degree of spread of the distribution (0 to 1)	Low
Peak count	Number of words with probability above 0.05	Small

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
Measurement target	Output distributions of the certain, vague, and unknown item sets on the frozen elementary-stage model

The measured figures in Sections 3 and 4 were taken in the environment above.

Appendix C. Reproduction Guide

The measured figures in 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 from 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, it is 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 three frozen models (471MB total) are on Zenodo at doi.org/10.5281/zenodo.21383724. The following one line downloads them and verifies integrity (sha256 fingerprint check) together. To do it by hand, download the three files from the record above, place them in 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 original text distributed. The following one 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 figures of each section. The figures in this paper are covered by the following probes. Target values are printed alongside the output, so they can be compared directly with the text.

Probe	Figures reproduced	Output to check
paper6_metacog_probe.py	Section 3 distribution statistics	Max probability 1.00, 0.58, 0.32, 0.35 and entropy 0.00, 0.20, 0.35, 0.37
paper6_vague_trigger_probe.py	Section 4 reinterpretation trigger	All 4 reach in a single pass; 3-round rise +0.88 versus -0.06

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

Banya Research Series, Part 6 (metacognition, distribution routing). All figures were measured directly for this paper in the Appendix B environment. The background view of nature is cited as the preceding axiom document [1] but is not the basis of this paper's claims. Part 1 is cited. Emotion and motivation are left as measurement tasks for a later stage once the corresponding circuits are in place. Self-utterance weaving similarity consolidation with inference traversal will be covered in the final part after the final assembly.