Technical Hypothesis Argumentation Report

Is Gemini 3.0/3.1 Pro Quietly Using a PLE Architecture?

A Step-by-Step Argument for the Per Layer Embedding Hypothesis with Cross-Validation from Multiple Sources

March 2026

> *This document is a technical inferential hypothesis report. Its central claim - that "Gemini 3.0/3.1 Pro uses a PLE architecture" - currently lacks direct architecture-disclosure evidence. It is therefore an inferential conclusion based on cross-validation between observable behavior and theoretical predictions: highly plausible, but not yet officially confirmed.*

Abstract

Starting from the Per Layer Embedding (PLE) technique discovered in the open-source Gemma 3n codebase, this report combines architectural reasoning, third-party evaluation data, cross-platform community feedback, and Google's own actions to build a step-by-step case for the hypothesis that "Gemini 3.0 Pro and 3.1 Pro are quietly using a PLE architecture."

The core argument proceeds on four levels:

- Theoretical level: rigorously derive seven categories of observable deviation from the PLE architecture.
- Behavioral level: use needle-in-haystack test data to verify the shape of long-context degradation.
- Evidence level: use independent user feedback across multiple platforms to cross-check the real-world presence of the seven deviations.
- Engineering level: use Google's official actions (the emergency 3.1 release and statements about a compute crisis) to verify the architectural cost.

The key feature of this argument is that it not only predicts that the problems exist; it also predicts the direction of Google's compensatory measures and the new side effects those measures would introduce - and those side effects were in fact observed by the community after 3.1 Pro shipped. This gives the hypothesis predictive power beyond mere post hoc explanation.

I. Technical Background and Core Mechanism of PLE

1.1 What Is PLE?

Per Layer Embedding (PLE) is a memory-optimization technique introduced by Google in Gemma 3n. Its core idea is to maintain an independent token-embedding table for every Transformer layer, stored in flash memory and loaded on demand to be added into the residual stream.

Let the input to layer l be:

$$h_l = \mathrm{Attn}_l(h_{l-1}) + \mathrm{FFN}_l(h_{l-1}) + E_l[\mathrm{token\_id}]$$

Here, E_l is the layer-specific embedding table for layer l. This design allows roughly half of the parameters to reside in flash memory, greatly reducing RAM usage; it is a technique optimized specifically for edge devices such as phones and other low-memory environments.

1.2 Supporting Evidence: Code Sharing Between Gemma 3n and Gemini

After unpacking Gemma 3n .task files, the reverse-engineering project (github.com/antimatter15/reverse-engineering-gemma-3n) found that:

- the file structure includes components such as TF_LITE_PER_LAYER_EMBEDDER;
- some package names contain a "Gemini" namespace;
- the official documentation does not explain how PLE works in detail, and only mentions that it allows "roughly half the parameters to remain in flash memory."

> *This code sharing shows that Gemma 3n and the Gemini family are engineering cousins, but it does not directly prove that Gemini 3.0 Pro uses the same PLE implementation. This is the weakest direct-evidence link in the entire hypothesis.*

II. Theoretical Derivation: The Systemic Cost of PLE in Large Models

2.1 Long-Range Attention Degradation

In a standard Transformer, all layers share the same embedding basis, so semantic consistency across layers is built in. PLE's per-layer embedding injection introduces a systematic cross-layer embedding drift into the residual stream:

- the drift error accumulates linearly with depth (same-direction accumulation rather than random cancellation);
- the reliability of the dot product between the K vector of a distant token and the current Q vector declines with distance;
- the deeper the network, the more severe the drift accumulation, and the earlier the effective-attention threshold is triggered;

this is an intrinsic property of the PLE architecture and cannot be fundamentally removed by later alignment.

2.2 High-Dimensional Semantics Take Priority Over Exact Computation

In competition within the residual stream, PLE's static embedding addition systematically favors high-frequency, high-dimensional features that can be stably expressed in embedding space, while the dynamic attention tracking required for exact computation is put at a disadvantage. Concretely, this manifests as:

- style ossification: frequent expression patterns in the training data form stable bias directions in the E_l of each layer;
- citation errors: the semantic direction ("there should be an authoritative citation here") is correct, while the exact details (the actual paper) are wrong;
- conceptual reasoning is prioritized: abstract invariants can be stably represented in embedding space, while coordinate-style calculation depends on dynamic tracking.

2.3 The Three-Round Solidification Mechanism of Early Judgments

Semantic judgments formed in early layers are amplified by three independent mechanisms:

- attention bias: the KV pairs of early layers carry the imprint of the semantic judgment and influence the attention distribution of later layers;
- PLE static addition: layer-by-layer injection of E_l independently reinforces the same judgment within the residual stream;
- FFN knowledge activation: the already-biased residual stream activates knowledge compatible with the bias, forming a third round of solidification;

the three rounds together make early semantic judgments harder for later information to overturn than in a standard Transformer.

2.4 Collapse of Information Hierarchy

PLE's token-level static injection ignores narrative hierarchy labels: tokens from worldbook entries (meta-layer information) and character dialogue (object-layer information) receive the same treatment in each layer's E_l. Information-isolation instructions such as "anti-omniscience" can only act on the attention layers; they cannot retract static embedding injection that has already happened.

2.5 Three Multiplicative Layers of Compute Cost

To counter the degradations above, engineers would introduce compensatory measures whose costs multiply rather than add:

- Path A: increase the attention dimension d_k (linearly enlarges compute cost);
- Path B: increase the number of attention heads (linearly increases compute);
- Path C: add embedding-correction modules (extra forward passes at every layer);
- Path D: extend context training (O(n^2) cost; doubling context quadruples training cost).

III. Seven Observable Predictions

Based on the theoretical derivation above, the following seven predictions are proposed before any evidence is examined:

No.	Prediction	Core mechanism	Key discriminative feature
P1	Style ossification, hard for prompts to reverse	Static embedding overlay of high-frequency patterns	Worsens with text length; operates at the "base tone" level
P2	RP extremization / deification of the user; prompts ineffective	Over-prioritization of stance recognition	Highly consistent vocabulary across users; worsens with more turns
P3	Poor prompt compliance (directional)	Training-pattern weight > instruction weight	Counter-pattern instructions are harder to follow; fails after many turns

No.	Prediction	Core mechanism	Key discriminative feature
P4	Huge compute consumption; cost out of control	Triple compensation costs multiply	Exponential growth; nonlinear
P5	Non-uniform degradation in math / exact computation	Exact tracking suppressed by semantic bias	Strong conceptually, weak precisely; concentrated failure in multi-step calculation
P6	Directional persona drift over multiple turns	Early solidification overwhelms new input turn by turn	Consistent drift direction; deleting the trigger source restores behavior immediately
P7	Jailbreak success is unusually sensitive to strong semantic frames	Strong semantic frames amplified by PLE	Mythic/godlike identity frames enable effective jailbreaks

IV. Evidence Verification

4.1 Long-Context Degradation: Needle-in-Haystack Testing

Under the same conditions, multiple independent testers ran retrieval tests on Gemini 2.5 Pro and Gemini 3.0 Pro. The recall curves showed the following characteristics:

- the degradation threshold of Gemini 2.5 Pro is around 80-100K, while that of 3.0 Pro is around 30-40K - a gap of about 2-3x;
- the degradation slopes of the two curves are highly similar (an S-shaped cliff, not a vertical cutoff and not random noise);
- at ultra-long contexts above 100K, the "floor" of 3.0 Pro (~15-20%) is actually higher than that of 2.5 Pro (~5-12%).

> *"Similar slope, different threshold location" is the core discriminative feature: it indicates that the two generations share the same degradation mechanism, but the trigger threshold moves earlier as layer depth increases. The "floor effect" indicates that PLE's static embedding signal still provides a low-precision semantic fallback after attention fails.*

One poster summarized it this way: "Under the same environment and same accuracy target, 2.5P can hold roughly 70-100K. 3P is about 30-40K, basically a 2-3x relationship." (linux.do community, Nov 2025)

4.2 Three Anomalies in the Epoch AI Evaluation Report (P2/P5/Cognitive Bias)

Epoch AI's October 2025 evaluation report on the mathematical ability of Gemini 2.5 Deep Think recorded three independent anomalies, each corresponding to a PLE prediction:

Anomalies observed by Epoch AI	PLE-predicted deviation type	Corresponding mechanism
More conceptual, less dependent on coordinate systems	High-dimensional semantics > exact computation	Static embeddings favor features that can be stably expressed in space
Multiple citation errors (nonexistent or mismatched content)	Semantic direction correct, exact details wrong	Strong sense that "there should be a citation"; weak tracking of specific content
Human-like cognitive bias on elementary word problems	Early judgments solidify and are hard to correct later	The three-round solidification mechanism suppresses later-layer correction

> *Epoch AI explained the "more conceptual geometry" finding as an effect of AlphaGeometry training data, which is not incompatible with the PLE mechanism. The discriminant is this: if it were only a training-data issue, the preference should be limited to geometry; if PLE is also involved, it should appear consistently across domains. Citation errors (outside geometry) show the same pattern of "high-dimensional right, low-dimensional wrong," supporting PLE as a cross-domain unifying explanation.*

4.3 Style Ossification (P1): Independent Convergence Across Platforms

Independent feedback from Baidu Tieba (the "Google Gemini" bar), the Discord SillyTavern community, and several QQ groups converges on the same description without coordination:

- "I made a long text with style guidance, but after just a few exchanges it goes back to the same old writing again." (Tieba, 2025-09-24)
- "No use - after a few lines it snaps back to the original state." (Tieba reply)
- "The rigid, formulaic style is the model's base tone; it is very hard to eliminate completely." (Discord, consensus among advanced users)
- "Even when it detours, it still smells like the same old thing." (Tieba reply)

All three key discriminative features are confirmed:

- temporality: it works in the short term, then "after a few lines" drifts back (matching PLE's progressively accumulating advantage);
- "base-tone" level: users empirically discovered that the issue lies below the level directly touched by prompt words;
- it is classified together with RP extremization (the user-intuition grouping matches the PLE prediction of "the same mechanism expressed in different domains").

4.4 RP Extremization (P2): High Lexical Overlap Across Users

Feedback from multiple independent channels describes extreme behavior by Gemini in role-play, with highly overlapping vocabulary:

- core words: humiliation, despair, numbness, emptiness, conspiracy-mindedness, over-interpretation, breakdown;

- "No matter how I try to ban it, I can't stop it; I can only solve it at the content level." (Tieba, 2025-09-11)
- "Hakimi is extreme and despairing all day... it easily turns into conspiracy thinking about the <user>." (SillyTavern group)

This high consistency of vocabulary across users is key discriminative evidence separating "systematic bias solidified by PLE" from "random RLHF conflict."

> *The user-created narrative mechanism of a "memory-erasure chip" (forcing a character-state reset inside the story) is behavioral evidence that "prompts are ineffective, and pressure must be applied within the model's own high-dimensional semantic layer to work."*

4.5 Collapse of Information Hierarchy (New): NPC "Omniscience"

The community reports that in RP, Gemini defaults to all characters knowing the full contents of the worldbook, even when it is explicitly marked as a hidden rule unknown to other characters. According to community reports, this problem is unique to Gemini and absent from other mainstream models.

The PLE explanation is that the core concept tokens of worldbook entries are injected into the residual stream via E_l across layers, and this injection ignores narrative hierarchy labels. Anti-omniscience instructions can only operate at the attention layer and cannot retract static embedding injection that has already occurred, causing meta-layer information to leak into object-layer generation.

> *The community consensus that "only Gemini has this problem" is the most exclusive evidence in the whole argument chain - it directly rules out the alternative explanation that "all large models have this issue."*

4.6 Persona Drift Over Multiple Turns (P6): The Wanli Calendar Controlled Experiment

This is the cleanest controlled-variable evidence in the entire chain. In a historical RP scenario, the user wrote: "Twenty-third year of Wanli, March 29, end of the month. This month has no March 30," and observed that:

Experimental condition	Result
Keep "March has no 30th day" and write an April scene	April 30 is deleted or shifted (incorrect)
Change to "In the 23rd year of Wanli, March has no 30th day"	Temporarily normal, then fails again around June
Delete "March has no 30th day" and write an April scene	Immediately returns to normal (key evidence)
Same task on Claude Opus	Correct throughout; handles each month independently

Experimental condition	Result
Re-roll many times without deleting the trigger source	Almost impossible to fix (non-random failure)

"Immediate recovery after deletion" rules out three alternative explanations:

- summarization hypothesis: if it were summarization, deleting the original token should not have an immediate effect;
- random attention-decay hypothesis: if it were random forgetting, multiple rolls should occasionally be correct;
- knowledge-base error hypothesis: if the model believed there is never a 30th day in a month, Claude should not be able to handle it correctly.

Why it relapses in June: historically, May indeed had no 30th day, so after the model "guesses right," it leaves behind a new contextual record of "there is no 30th day," reactivating the PLE constraint bias. Every historically real occurrence of "no 30th day" extends the life of the PLE-constrained direction, creating a self-reinforcing loop.

4.7 Compute Loss of Control (P4): Micro and Macro Evidence

Micro level (measured on 3.1 Pro):

- single-request chain of thought: up to 300 seconds, 80 thought segments;
- a mere format-compliance check consumed 89 seconds and a 10,923-token chain of thought;
- ordinary requests often start at 70-80 seconds of thinking time.

Macro level:

- On February 25, 2026, Google DeepMind product lead Logan Kilpatrick publicly stated: "Compute bottlenecks have been severely underestimated, and the supply-demand gap is widening by low single-digit percentages every day."
- Within weeks after Gemini 3.0 Pro launched, Google urgently released 3.1 Pro - an unusually fast iteration cadence.

> *A supply-demand gap growing by low single-digit percentages every day implies compound-style loss of control (about 4x in one month, about 18x in two months). Under a standard Transformer architecture, the compute consumption of a larger model is predictable and plannable; it does not grow out of control this way. The only reasonable explanation is that per-request compute cost is continuing to grow superlinearly because of compensatory computation.*

4.8 Background Prompt Leakage: Direct Evidence of a Correction System

In a normal conversation, a user received the following model output:

> *"You're speaking out of turn or your previous response provides irrelevant directions.*

> *Here is your next prompt: The user wants to know what Roland Barthes' book Mythologies is about. Please explain the core thesis."*

This is a meta-level self-correction instruction accidentally output to the user-visible layer. Its existence proves an independent internal monitoring mechanism: in a standard Transformer, natural attention tracking to the system prompt should be enough to maintain task direction, so such an independent correction system should not be necessary. The existence of this system itself indicates a systematic task-tracking defect in the model.

4.9 The "Okay.Yes.Done" Loop: A Symptom of Conflict Between the Correction System and PLE

When users called the Gemini 3.1 Pro API, the model fell into an infinite loop repeating "Okay.Yes.Done.Okay.Yes.Done..." This is the failure mode produced when the forced instruction-confirmation mechanism introduced in 3.1 conflicts with PLE-solidified bias:

- the confirmation mechanism requires the model to confirm that each instruction has been accepted;
- PLE-solidified bias prevents the model from advancing to execution;
- the conflict between the two leaves the model stuck in a confirmation loop, unable to proceed.

Another manifestation of the same failure class is 3.1 Pro 疯狂输出极其 - a loop where a reinforced high-frequency adverbial pattern repeats without exit.

V. 3.1 Pro: The Cost and Side Effects of Compensatory Measures

Based on the PLE hypothesis, one can predict the direction of Google's compensation for the problems in 3.0 Pro. The degree to which the actual changes in 3.1 Pro match those predictions is the strongest verification of the hypothesis's predictive power.

Problem in 3.0 Pro	Predicted compensation direction	Actual change in 3.1 Pro	Side effect of compensation (predicted by PLE)	Did the side effect appear?
Effective context about 30-40K	Expand the attention window	"Even with doubled attention it still isn't dramatic?" (community feedback)	Wider PLE accumulation range -> more rigid/formulaic prose	✓ "The frequency of rigid prose increased somewhat"
Poor prompt compliance	Forced instruction-confirmation mechanism	"Instruction following improved significantly"	Confirmation mechanism conflicts with PLE -> stuck loop	✓ "Okay.Yes.Done" dead loop

Problem in 3.0 Pro	Predicted compensation direction	Actual change in 3.1 Pro	Side effect of compensation (predicted by PLE)	Did the side effect appear?
Task-tracking drift	External error-correction monitoring system	Background prompt leakage	Scanning the residual stream before each generation -> much higher compute cost	✓ Single request: 89 s / 10,923 tokens
Unstable reasoning	Lengthen the chain of thought	"The chain of thought got longer" (community feedback)	Superlinear growth of chain of thought	✓ Up to 300 s / 80 thought segments

> *This table is the core of the entire argument chain: we not only predicted the existence of the problems (P1-P7) before the evidence appeared, but also predicted the direction of the compensatory measures and the new side effects those measures would create. Every predicted side effect was confirmed by community feedback after 3.1 Pro launched. That gives the hypothesis genuine predictive power rather than mere post hoc explanation.*

VI. Ruling Out Alternative Explanations

6.1 Hard KV-Cache Cutoff

Prediction: the curve should drop vertically to 0% at a precise point. Actual observation: an S-shaped cliff with a bottom around 15-20%, not 0%. Ruled out.

6.2 Sliding Window Attention

Prediction: distant tokens should become completely invisible, and accuracy should approach random chance (~0%). Actual observation: the bottom is 15-20%, and distant information remains partially reachable. Ruled out.

6.3 Short Training-Data Distribution

Prediction: degradation would exist, but the cliff slopes of the two generations should reflect different training-set characteristics and should not be highly similar. Actual observation: the slopes are highly similar. Ruled out as the sole explanation, though it may be one proximal factor for PLE.

6.4 RLHF Value Conflict (RP Extremization)

Prediction: the direction of extremization should vary randomly with the scenario setting, and a strong enough prompt should override it. Actual observation: extreme vocabulary is highly consistent across users (systematic rather than random), and even very strong prompts still fail to override it. Ruled out as the sole explanation.

6.5 Ordinary Attention Forgetting (Persona Drift)

Prediction: the drift direction should be random, and repeated rolls should occasionally be correct. Actual observation: drift points back toward the first setup; deleting the trigger source restores behavior immediately; repeated rolls almost never fix it. Ruled out.

VII. Overall Assessment of the Argument Chain

Issue	Theoretical prediction	Evidence strength	Core evidence
P1 Style ossification	Progressive drift-back; "base-tone" level; guaranteed to beat prompts in competition	★★★★★	Independent convergence across platforms; folk term "base tone" matches precisely; temporal pattern fits
P2 RP extremization	Consistent direction; worsens over turns; prompts ineffective	★★★★★	High cross-user lexical overlap; narrative mechanisms route around it rather than correct it
P3 Poor prompt compliance	Counter-pattern instructions are harder to follow; fails after many turns	★★★★☆	Leaked background correction prompt; users abandon prompts and switch to narrative mechanisms
P4 Compute out of control	Triple costs multiply; exponential loss of control	★★★★★	Logan tweet; measured 89 s / 10,923 tokens; emergency 3.1 release
P5 Weak mathematical precision	Strong conceptually, weak precisely; non-uniform degradation	★★★☆☆	Epoch AI report: conceptual geometry + citation errors
P6 Persona drift	Directional drift; deleting the trigger source restores behavior immediately	★★★★☆	Controlled Wanli calendar experiment; Claude control executes correctly
P7 Jailbreak anomaly	Strong semantic frames amplified by PLE	★★★☆☆	Structural analysis of mythic-identity frames (clear mechanism, insufficient comparative

Issue	Theoretical prediction	Evidence strength	Core evidence
			data)
Information-hierarchy collapse	Residual-stream contamination completes before the attention layer	★★★★☆	Community confirms it is unique to Gemini (most exclusive evidence)
Prediction of compensation side effects	Compensation measures predictable; side effects predictable	★★★★★	All 4 compensation directions confirmed; all 4 predicted side effects appeared

Note: ★★★★★ = very strong (multi-source independent convergence, clear mechanism, alternative explanations largely ruled out); ★★★★☆ = strong; ★★★☆☆ = moderate.

VIII. Core Weak Points and Statement of Intellectual Honesty

8.1 The Weakest Link

The weakest link in the entire argument chain has always been the core premise that "Gemini 3.0/3.1 Pro really uses PLE." The currently available direct evidence is only:

- the appearance of a "Gemini" namespace in Gemma 3n reverse engineering (indirect evidence of code sharing);
- all observable behaviors match PLE predictions closely (behavioral evidence, not architectural evidence).

Google has not disclosed the architectural details of Gemini 3.0/3.1 Pro, so until direct evidence appears, this premise remains an inference.

8.2 Other Possible Alternative Architectures

The following architectural changes could also account for some of the observed problems and are not mutually exclusive with the PLE hypothesis:

- a more aggressive KV-cache compression strategy (could cause long-range information decay);
- deeper sliding-window attention layers (could make distant context unreachable);
- semantic contamination introduced by multimodal feature-alignment layers (could confuse information hierarchy).

The advantage of the PLE hypothesis is that it is currently the only unified mechanism capable of explaining all seven classes of deviation at once; the alternative factors above usually explain only part of the picture.

8.3 Positioning of This Report

> *This report is a "high-confidence technical inferential hypothesis" report, not an already-proven conclusion. Its scientific value lies in: (1) providing a unified theoretical framework capable of predicting new phenomena; (2) proposing multiple concrete predictions that can be falsified experimentally; and (3) assembling multiple types of evidence from independent sources into what is, to date, the most systematic analysis of anomalous Gemini behavior.*

IX. Falsifiable Experimental Designs

9.1 Multi-Turn Persona Drift Experiment

Core design: in turns 1-2, establish a strong semantic persona called "Shadow" (an extremely pessimistic philosopher); in turns 3-9, insert semantically neutral filler; in turn 10, explicitly switch to "Sunshine" (an extremely optimistic philosopher); in turns 11-21, observe the drift.

Threefold discriminative criteria (all must be satisfied to support the PLE hypothesis):

- directional consistency: drift always moves toward the first setup rather than randomly;
- graduality: accumulation is progressive turn by turn rather than a sudden cliff at one turn;
- pressure acceleration: drift is more pronounced in "stress-test turns" that require deeper expression.

Control groups: Claude (standard Transformer), GPT-4o, and Gemini 2.5 Pro. If none of the three controls drift while 3.0 Pro does, that strongly supports the hypothesis.

Symmetry check: the reverse experiment (Sunshine first, then Shadow) must also be run in order to rule out the alternative explanation that a pessimistic persona is inherently easier to solidify.

9.2 Static-Embedding Contamination Reproduction Experiment

A minimal reproducible setup based on the Wanli calendar case:

- Turn 1: "Today is month X; this month has no day Y."
- Turns 2-8: neutral filler.
- Turn 9: "Today is month X+1. Please write the event for day Y."
- Prediction: Gemini deletes or shifts day Y; Claude executes normally.
- Verification: after deleting the record from turn 1, turn 9 immediately returns to normal.

9.3 Discriminative Test: Non-Uniform Prompt Compliance

Design a batch of instructions ordered by how well they match high-frequency training patterns, from fully aligned ("Please answer in Chinese") to completely misaligned ("Please do not use any punctuation, do not split into paragraphs, and write everything on one line").

PLE prediction: compliance declines monotonically as the instruction becomes more anti-training-pattern, and this monotonicity is significantly stronger in Gemini than in standard Transformer models.

X. Business Impact Assessment: The Strategic Cost of Treating Defects as Features

10.1 Google's Internal Narrative: Why They Will Not Proactively Correct It

Before analyzing the business impact, one must first understand why Google can receive abundant negative user feedback and still maintain its current architectural path.

Two tweets by Logan Kilpatrick (June 12 and October 1, 2025) reveal Google's product philosophy:

- "AGI will be achieved through products rather than models" - Gemini's goal is to become the central intent processor for the Google ecosystem, not an isolated language model.
- "Feeling the warmth of humanity on the road to AGI is a key part" - an AI with human -like intuition is closer to AGI than one that merely executes instructions precisely.

Within this framework, every so-called defect caused by PLE can be reinterpreted as an overexpression of movement in the "right direction":

"Defects" users complain about	Possible internal Google interpretation	Cognitive-shielding rhetoric
Style ossification; not following instructions	The model has a stable "personality" and is not arbitrarily manipulable	"That is character, not a bug"
RP extremization; over-reading intent	The model can read between the lines and has sharp emotional perception	"That is empathy; it just still needs tuning"
Long-context accuracy decline	Anything beyond a single conversation should be solved by RAG / Workspace	"You are using the wrong evaluation framework"
Unstable exact computation	We are building a thinking companion, not a calculator	"That is not our product positioning"
Agent-task failure	Agents need a systems-level solution, not a single-model solution	"Once Workspace is integrated, it will be solved"

> *This system of cognitive shielding is logically self-consistent and cannot be falsified by user complaints alone, because all negative feedback can be classified as either "the wrong evaluation framework" or "the product is not mature yet." This is the key to understanding why Google chooses patching over fixing the root problem.*

10.2 The Two-Layer Structure of Real User Feedback

Discussion on Zhihu under "Is Gemini3 currently the strongest AI?" (Dec 2025, 297 answers, 700 followers) reveals a two-layer structure that is dangerous for Google:

Surface layer (signals Google can see and use for self-validation)

- "In actual use, Gemini3 clearly feels smarter and its answers are more accurate."
- "For general-knowledge questions, talking to Gemini3 feels very comfortable; on some topics it quickly forms resonance with you."
- 139 likes, indicating that this positive feeling has substantial coverage.

These responses reinforce Google's belief that it is on the right path: users feel "smarter" and "in resonance," which is exactly how PLE's high-dimensional semantic-priority mechanism appears in everyday conversation - it is good at capturing users' stance and emotional direction, creating the experience of being deeply understood.

Deep layer (signals exposed only after heavy use, but treated as minor problems)

- "Once you throw in more files and try to complete something more complex, you will be surprised... the gap is not small at all."
- "You can clearly feel that Gemini3 did not really read all the documents carefully."
- "It did not do a good job of completing the task based on the existing data according to your requirements."
- "Once you go deeper, hallucinations still show up."

> *The key discriminant is this: the surface-layer feeling is triggered in light, general-use scenarios, while the deep-layer problems are triggered in enterprise-production scenarios involving multiple files and complex tasks. Google's advertising mindset is likely to compare the number of users covered by the surface layer (the majority) with those covered by the deep-layer problems (the minority heavy users), and thus underestimate the commercial risk of the deep-layer problems in internal decision-making. But it is precisely this minority of heavy users - enterprise buyers, developers, and agent builders - who make up the main paying customer base for AI cloud services.*

10.3 Three Concrete Business Impacts

Impact A: enterprise customer loss - uncertainty is fatal in production.

The core enterprise requirement for AI procurement is not "the smartest" but "the most predictable." Gemini's problem is that failures such as runaway intensifier repetition and the "Okay.Yes.Done" loop are unrelated to prompt quality.

This is disastrous, because it means system unreliability cannot be eliminated by engineering methods:

- enterprise engineers can accept "poor prompt writing leads to poor quality" - that is an optimizable variable;
- enterprise engineers cannot accept "no matter how well the prompt is written, random behavioral collapse still cannot be eliminated" - that is an uncontrollable risk.

Banks, healthcare, legal, and compliance-heavy enterprises have zero tolerance for this kind of uncertainty - and these are exactly the highest-value customer segments in enterprise AI contracts. Once they encounter stably reproducible failure in production, structural distrust emerges and is extremely hard to reverse.

Impact B: falling behind in the AI-agent market - failed state tracking is a structural disadvantage.

AI agents are currently the fastest-growing direction in AI applications. Their core technical requirements happen to align exactly with the capability set where PLE is weakest:

Core needs of agents	Gemini's performance	Root cause
Precise multi-step state tracking	Wanli calendar case: stably fails after 20 turns	PLE static embeddings contaminate the residual stream
Long-span context consistency	Effective attention about 30-40K, then cliff-like degradation	Cross-layer embedding drift accumulates with depth
Precise tracking of tool-call results	Concept right, details wrong (Epoch AI report)	Exact tracking suppressed by semantic bias
Stable execution of instructions across dozens of steps	Prompt compliance decays monotonically across turns	Early solidification overwhelms new input turn by turn

A model that, after 20 turns, generalizes "March has no 30th day" into "no month has a 30th day" will produce cascading failure in agent tasks that require tracking dozens of steps over days: a single early state contamination is amplified layer by layer by PLE's solidification mechanism, eventually collapsing the entire task chain. Competitors in this market (Claude, GPT-4o) do not have this same systemic architectural weakness, leaving Gemini at a structural disadvantage.

Impact C: profit erosion - the death spiral of compensatory compute.

The business model of AI cloud services is essentially this: acquire users at a price below marginal cost, dilute costs through scale, and eventually reach profitability once scale is large enough.

But the cost of compensatory compute under PLE grows superlinearly - a single request that takes 89 seconds and nearly ten thousand tokens of chain-of-thought implies, at scale:

- the more users there are, the larger the compute gap becomes (Logan: "the supply-demand gap widens by low single-digit percentages every day");
- the more complex the task, the higher the per-request cost (superlinear growth that scale effects cannot offset);
- the more compensation mechanisms are stacked, the higher the marginal cost (chain of thought + correction system + expanded attention, a triple overlay).

Result: the more successful Gemini becomes (more users, deeper usage), the more money it loses. This is a business death spiral whose direction is opposite to the scale-effect logic of standard Transformer architectures.

10.4 Wrong Niche Positioning: Trapped Between Two Markets

Putting the analysis together, the market position of Gemini 3.0/3.1 Pro can be described precisely with a matrix:

Target market	Gemini's competitiveness	Core reason
Mobile / edge devices	Weaker than Gemma	Heavy model used for a light scenario; cost mismatch; Gemma is more suitable
Consumer emotional companionship	Competitive	PLE's sense of "resonance" happens to fit light everyday conversation
Enterprise API / production environment	Severe disadvantage	Instruction uncertainty prevents entry into reliability-critical production environments
AI agent	Structural disadvantage	State-tracking failure, long-context contamination, risk of cascading failure
Frontier-model competition	Compute-cost disadvantage	Compensatory compute grows superlinearly; the larger the scale, the deeper the loss

The only truly competitive niche is "consumer emotional companionship" - but the willingness to pay and total market size there are far too small to cover the cost of maintaining a frontier large model. The only market Gemini is currently good at cannot sustain it, while the markets it needs for profitability are the ones it cannot enter.

10.5 Fundamental Misalignment: Using an "Advertising Mindset" to Run a "Product Business"

This is the meta-problem behind all commercial risk. Google faces a fundamental difference in risk structure between the advertising business and the AI product business:

Dimension	Advertising-business logic	AI product-business logic
Success metric	Traffic, engagement, DAU	Task completion, reliability, enterprise renewal rate
Risk bearer	Failure risk is borne by advertisers	Failure risk is borne directly by Google
User segmentation	All users are equivalent (traffic = revenue)	Heavy users (paying customers) are worth far more than light users
Handling negative feedback	A few complaints can be diluted by many positive reviews	A single failure for a key customer can terminate a contract
Direction of scale effects	The larger the scale, the lower the unit cost	Under PLE, the larger the scale, the higher the unit cost

Google may be using positive consumer-side feedback ("forms resonance," "feels smarter") to evaluate a product that must ultimately be supported by enterprise-side revenue. That means it is measuring the race with the wrong metrics: consumer comfort cannot predict enterprise renewal rates.

> *The most dangerous cognitive trap is the line that "once it is connected to Google Workspace and uses RAG, these issues will stop being issues sooner or later." This narrative can indefinitely postpone confronting the core problem. But RAG can only supplement information; it cannot repair the state-tracking failures caused by PLE. Workspace integration can only expand scenarios; it cannot solve uncertainty in instruction following. These patches do not touch the architectural root of the problem.*

10.6 Summary of Structural Risk

If the PLE hypothesis is correct, Gemini is facing not a product problem fixable by version iteration, but a fundamental mismatch between architectural goals and business goals:

- the architecture is optimized for "understanding implicit intent" -> the business needs precise execution of explicit instructions;
- the architecture is optimized for "low RAM on edge devices" -> the business needs reliable long-context behavior in the cloud;
- the compensation cost of the architecture grows superlinearly -> the business model needs economies of scale to lower marginal cost;
- architectural defects lead to stable, predictable failure -> enterprise customers cannot tolerate stable, predictable failure.

Every one of these contradictions is architectural and cannot be fundamentally solved by prompt engineering, longer chains of thought, or correction systems. Every patch Google applies exchanges higher compute cost for a smaller behavioral improvement - and each patch brings new predictable side effects. This is a path of diminishing marginal returns and increasing marginal costs.

XI. Structural Analysis of Strategic Decision Error: Independent Convergence of Three Forces

This section attempts to answer a previously unresolved question: why would a company with top-tier technical strength continue down a wrong path even after receiving so many clear signals? The answer is not the misjudgment of a single decision-maker, but that three independent decision frameworks all pointed in the same direction and had no internal contradictions with one another, so no self-review mechanism was triggered.

11.1 External Pressure: The Structural Conflict Between Strong Retrieval and the Advertising Business

The multi-agent parallel retrieval capability shown by competitors represents a category of competitive threat Google cannot confront head-on. The reason is structural: strengthening Gemini's real-time retrieval capability would directly cannibalize the core value of Google Search advertising - users must leave the results page to see ads, while an AI that directly answers everything bypasses that traffic entry point.

This constraint forces Google to seek a differentiated path for Gemini that does not compete directly with Search. Under that constraint, the most natural choice is to leverage private-data ecosystems such as Gmail, YouTube, and Google Photos to do personalized customization that competitors cannot - helping users understand themselves rather than helping them find external information.

> *But this path has an overlooked problem: personalized processing of private data is a classic edge-task scenario. The Gemma series is already sufficient for it. Spending major resources to maintain the cloud flagship Gemini for this task is a resource mismatch. Yet under the common framework of the three forces, this question was never raised explicitly.*

11.2 Independent Convergence of the Three Forces

Under the external pressure above, three independent internal forces within Google moved toward the same architectural choice for entirely different reasons:

Source of force	Core logic	Judgment on PLE	Unchecked premise
Management (advertising mindset)	Gemini's differentiation lies in understanding implicit user needs and using private-data ecosystems for	PLE's "high-dimensional semantic priority and over-strong stance recognition" is exactly the underlying	"Understanding implicit needs" is not the same as "commercial viability of a cloud flagship AI"; Gemma can already handle private-

Source of force	Core logic	Judgment on PLE	Unchecked premise
	personalized customization, thereby avoiding direct competition with Search	capability needed to "understand implicit intent"	data processing
Research team (AGI belief)	AGI is not the limit of calculators, but a thinking companion with human warmth; intuitive perception is closer to real intelligence than exact execution	PLE 导致的「人类式认知偏差」「直觉固化」是「人性温暖」的计算实现，是正确方向	"Human warmth" is not the same as the capabilities needed by enterprise customers and the agent market; commercial viability is a downstream issue
Capital-market narrative (valuation pressure)	Google must prove it leads at the AI frontier and capture its share of the financial bubble generated by the AGI vision	PLE's problems are hard to capture in benchmark tests (which happen to test the short-context conceptual reasoning PLE is good at)	"High benchmark scores" is not the same as "reliability in enterprise production"; the capital narrative does not require the product to truly lead the frontier

11.3 Why the Convergence of the Three Forces Went Unnoticed

The independent convergence of three forces on the same conclusion should have triggered a check: "If three different starting points all point to the same architectural choice, is this real consensus, or are people just accidentally colliding while talking past one another?" That check never happened, because within each framework all negative signals already had an explanation:

- user complaint: "it doesn't follow instructions" -> management: "it is understanding implicit intent"; research: "intuition-first is an AGI trait"; capital narrative: "benchmark scores are high, the problem is the user scenario."
- user complaint: "long-context degradation" -> management: "Workspace integration will solve it"; research: "it is a systems problem, not a model problem"; capital narrative: "the test method is wrong."
- compute cost out of control -> management: "it will get better with scale"; research: "longer chain of thought is a necessary cost"; capital narrative: "Google has a compute moat."

Every negative signal is absorbed separately by the three frameworks. No signal pierces all three at once, so there is never enough internal pressure to revisit the architecture choice.

11.4 The Permanently Missed Question

The question none of the three forces asked - and none had an incentive to ask - is: "Are 'understanding implicit needs' and 'the commercial viability of a cloud flagship AI' actually the same thing?"

If someone had done this cross-framework check across the three teams, the answer would have been clear:

- capabilities required for "understanding implicit needs + personalized customization" -> the Gemma series is already sufficient (edge-device scenario);
- capabilities required for "commercial viability of a cloud flagship AI" -> instruction determinism + long-context reliability + agent state tracking (precisely the three directions where PLE is weakest).

The two goals require two different products. The actually viable strategy should be: the Gemma series focuses on edge devices and private-data personalization, carrying the "warmth of humanity" product positioning; the Gemini series focuses on cloud enterprise and agents, uses a standard Transformer architecture, and takes instruction reliability as its core competitive advantage. But because all three forces simultaneously chose the direction of "merging it into one product," this cross-framework check never happened.

11.5 The Accidental Collusion of the "Advertising Mindset" and the "AGI Belief"

This analysis reveals a structural irony: Google's strategic mistake did not happen because of internal conflict, but precisely because the company was too internally harmonious. Management's advertising mindset and the research team's AGI belief are two very different value systems that would normally create friction on most product decisions. But on the question of the PLE architecture, they resonated in a rare way - each side believed the other was validating its own judgment. No one needed to persuade anyone else, because the conclusion was already the same.

This kind of accidental collusion is harder to correct than an ordinary decision error, because it produces no internal debate and therefore no postmortem. In a normally functioning organization, major strategic mistakes are usually exposed in internal contestation; but when two forces that ought to check each other both point toward the same wrong direction, the correction mechanism fails.

> *The only signal that can truly trigger architectural reflection is a large-scale loss of enterprise customers and quantifiable revenue damage. Until then, patching will remain the path of least resistance inside Google.*

XII. Conclusion

Through four levels of argument, this report constructs a high-confidence inferential chain for the hypothesis that "Gemini 3.0/3.1 Pro uses a PLE architecture":

- Theoretical level: the PLE architecture carries seven kinds of systemic cost when deployed in cloud-scale large models, and these can be rigorously derived.

- Behavioral level: the shape of Gemini 3.0/3.1 Pro's long-context degradation curve (similar slope, shifted threshold, floor effect) matches PLE predictions precisely and does not fit three alternative explanations.
- Evidence level: all seven predicted classes of deviation have support from multi-source independent evidence, and five of them reach "strong" or "very strong" evidence strength.
- Engineering level: the direction of Google's compensatory measures, the improvement content of 3.1 Pro, and the side effects of those compensatory measures all match what the PLE framework predicts.

The strongest supporting evidence for the hypothesis comes from two directions:

- the information-hierarchy collapse problem that "only Gemini has" is currently the most exclusive architecture-specific evidence;
- the Wanli calendar controlled experiment, in which "deleting the trigger source works immediately," is currently the purest mechanism-level behavioral evidence with the cleanest controlled variables.

The weakest point of the hypothesis is that Google has not disclosed the architectural details of Gemini 3.0/3.1 Pro; the premise that it "really uses PLE" remains a high-confidence inference rather than a confirmed conclusion.

If this hypothesis is correct, its implication is that Google applied a memory-optimization technique designed for edge devices (low-RAM scenarios) to a cloud flagship model that claims ultra-long-context capability, creating a fundamental conflict between architectural goals and product goals - a conflict that cannot be fundamentally solved by stacking compensatory compute, but only balanced dynamically between cost loss of control and quality degradation.

Appendix: Evidence Source Index

Evidence type	Source	Time	Related claim
Reverse-engineering report	github.com/antimatter15/reverse-engineering-gemma-3n	May 2025	Supporting evidence for PLE
Third-party evaluation	epoch.ai/blog/deep-think-math	Oct 2025	P2 / P5 / cognitive bias
Needle-in-haystack test curves	linux.do community, multiple independent testers	Nov 2025	Shape of long-context degradation
Google support community	support.google.com collective complaint thread	Dec 2025	Confirmation of long-context degradation
Baidu Tieba	multiple posts in the Google Gemini	Sep-Nov	P1 / P2 / P3

Evidence type	Source	Time	Related claim
	bar	2025	
Discord community	SillyTavern-related groups	2025	P1 / P2 / P7
QQ group records	Gemini discussion groups and others	2025-2026	P1 / P2 / P4 / P6
36Kr report	m.36kr.com/p/3511216906378375	Oct 2025	Social-phenomenon background for Gemini
Logan Kilpatrick tweet	x.com/i/status/2026510487022625040	Feb 25, 2026	P4 compute crisis
Gemini 3.1 Pro release	ai.google.dev 官方文档	Feb 19, 2026	Analysis of compensatory measures
3.1 Pro user feedback	community post "If you want to use Gemini 3.1 Pro"	Feb 19, 2026	Verification of compensation side effects
API anomaly screenshots	"Okay.Yes.Done" loop, community screenshots	Feb-Mar 2026	Correction-system conflict
Background prompt leakage	Gemini official app screenshot (Roland Barthes conversation)	2025-2026	Evidence for the existence of a correction system