The Delivery-Chain Race: Distribution, Unit Cost and Verifiability Are Displacing Parameter Count

Great Unicorn Intelligence Research Team | HX Research • Analysis • 1 November 2026

Abstract: Drawing on product launches, official technical posts and papers released in late October 2026, we argue that the binding constraint in the large-model race is shifting from building stronger models to delivering one verifiable inference, through a trusted channel, at a low enough unit cost. We use the expected cost per successful task to analyse system-level distribution, subscription capacity, inference acceleration, teacher-free self-distillation, training environments and quantized-artifact quality. We close with five falsifiable forecasts and an evaluation framework. Figures carry their sources and are not independently reproduced.

Keywords: delivery chain; cost per task; system-level distribution; inference efficiency; self-distillation; training environments; quantized artifacts

---

Introduction: why a bigger model is no longer a sufficient answer

For two years the default questions about a large model were how many parameters it had and how it scored. Read together, the events of the last week of October 2026 show that the centre of gravity has moved. Apple announced that Apple Intelligence in mainland China will support Alibaba's Qwen in selected experiences, opening for testing with the developer beta on 4 November and shipping to Chinese users with iOS 27.3 in December[1,2]. Anthropic's pricing page now states that the Ultra plan carries "no 5-hour limit" for verified accounts, with ten times Pro usage per week[3]. Canva released Brush, a vision-language model for design editing, which it says is up to 69% faster on general editing tasks and 46% cheaper per edit than its previous design agent[4]. The same week's papers clustered around cheaper inference, self-distillation without an external teacher, training environments and the quality of quantized model files.

None of these wins came from a bigger base model. Each acts on a different link of the delivery chain: who can reach the user, what each call costs, how likely a call is to succeed,

and whether success can be verified. We condense the chain into one expression that serves as the analytical tool for the rest of this piece:

Expected cost per successful task: $C^* = c / q$. Here c is the full cost of one attempt (inference compute, retries, amortised human review) and q is the probability that an attempt is verified as successful. If failures can be retried independently until success, the number of attempts is geometric with mean 1/q, so $C^* = c/q$.

The expression is plain, yet it captures something routinely missed: a technique that lowers c but also lowers q can raise C^*. Low-bit quantization, parallel decoding and visual-token pruning all carry that risk. Conversely, spending that does not look like cost reduction — state checks in training environments, smoke tests for quantized files, consent flows — acts on q, or on whether success can be confirmed at all, and often moves C^* more. Multiply by the reachable user base N and you have a company's position on the delivery chain.

We organise the argument around five questions that build on one another: why distribution and capacity are becoming scarce; what the three ways of cutting unit inference cost are and what each hides; where cheap learning signals come from; why verifiability is the new bottleneck; and where new physical forms of compute sit on the chain. Table 1 maps the evidence onto the terms of C^*.

Table 1 | Core evidence mapped to links of the delivery chain (evidence strength assessed by the Research Team)

Evidence	Acts mainly on	Evidence type	Question we ask
Qwen in Apple Intelligence	Reach N, consent rate	Official press release, news agency	How is the privacy boundary of system-level distribution defined?
Claude Ultra drops the 5-hour limit	Capacity and pricing structure	Official pricing page	How do long agent sessions change capacity economics?
Canva Brush	Cost per attempt c, success q	Official release (self-reported)	Can vertical post-training cut cost at equal quality?
LOWN, Soft-MTP, quantization studies	Cost per attempt c	Papers	What do cost cuts do to q?
WISD and self-distillation dissections	Training cost	Papers	Where does the signal come from without a strong teacher?
WebAnything, Genie 4, Broken on Arrival	Measurability of q	Papers, official blog	Can environments and artifacts be verified?
Project Suncatcher	Long-run energy and compute cost	Official blog, preprint	What can orbital compute do in the near term?

1. Distribution and capacity: from whose model is strongest to who is authorised to be called

System-level distribution rewrites the cost of reaching users

Apple's release is carefully worded. Apple Intelligence is powered by generative models Apple built; Apple has partnered with Alibaba to add support for Qwen in selected experiences, so users in China can use Qwen in Siri, Writing Tools and Visual Intelligence, including text and image understanding; users reach Qwen by signing in to a Qwen account or with Sign in with Apple; and the update excludes Siri AI and other next-generation features announced at WWDC 2026[1]. Xinhua's English report adds Alibaba's side: users can choose whether to enable the Qwen integration and what information to share, and Alibaba says it will not use user requests or generated responses for training[2].

On the delivery chain the significance is not that Qwen was chosen but that the marginal cost of reaching a user has been rewritten. A standalone app must be found, downloaded, registered and turned into a habit, losing users at every step. A system-level entry point embeds the call in Writing Tools, Siri and the camera people already use, so the ceiling on N is set by the installed device base rather than by app-store rank. The price is that the model provider gives up control of the interface and part of the user relationship in exchange for call volume.

Privacy boundaries: verifiable versus promised

The structure of the privacy boundary deserves closer analysis. Introducing Private Cloud Compute (PCC) in 2024, Apple singled out a problem with conventional cloud AI: when a service claims not to log certain user data, security researchers generally cannot verify the claim, and the provider often cannot durably enforce it; one of PCC's design goals is to make such promises externally checkable[5]. That gives us a comparison frame:

- On-device models: data never leaves the device; privacy rests on device security.
- Apple's own cloud (PCC): data leaves the device, but the privacy promise is engineered to be verifiable.
- Third-party model access (Qwen here): with the user's consent, data goes to a third party; privacy rests on the consent flow and the provider's commitments.

These are not ranked; they differ in where trust comes from. The first two lean on technical mechanisms, the third on consent design and contractual promises. Research also shows that cloud–edge collaborative decoding can leak private context even when only output distributions, not raw data, are transmitted; the defence proposed in that work reduces leakage by up to 87.2% while preserving collaborative quality, according to the paper[6]. Consent before sending decides whether something is sent; data minimisation still has to govern what the transmitted payload contains.

The implication for enterprises is concrete. More and more companies will call several external models from their own apps and internal systems. The questions that matter are not which model scores highest but three others: is every call covered by consent bound to its purpose; is the outgoing context minimised; and can the caller prove afterwards that it did both. Those three answers determine the consent rate and the compliance exposure, and therefore N and c.

Capacity economics: from time-window throttling to budgets

Anthropic's change looks like a product detail. The pricing page shows Max at US$100 a month with five times Pro usage every five hours and every week, and Ultra at US$200 a month with ten times Pro usage per week and no five-hour limit for verified accounts; it adds that usage limits still apply and that how fast a limit is reached depends on the model, the length of chats and the size of files[3].

A capacity model makes the logic visible. Subscriptions usually control cost with two gates: a short-window cap (five hours) limits peak concurrency, and a long-window cap (a week) limits volume. Removing the short gate while keeping the weekly one swaps a small, fast-draining token bucket for a large, slow-draining one: total volume is unchanged, but users may spend it in one continuous stretch. Why do it? Because the shape of the workload has changed. A long-running coding agent calls the model, edits files and runs tests for hours; a five-hour window cuts it off mid-task, and the cost of the interruption is not just waiting — context must be rebuilt and some work redone, so c rises and q falls.

Consider illustrative, recomputable numbers. A user has 100 units per week; one long task needs 40 units of continuous execution; the five-hour window allows 25 units. With the short cap the task is interrupted once; suppose restoring context costs 6 extra units and there is a 20% chance that lost state forces a full redo. Expected consumption is $40 + 6 + 0.2 \times 40 = 54$

units, against 40 without the cap. With the same weekly budget, the number of long tasks a user can finish rises from about 1.85 to 2.5. For the provider, the price is a less predictable peak load, absorbed through larger capacity headroom or finer-grained scheduling.

Vertical models: cutting cost where outputs can be checked

Canva Brush illustrates a different route. Canva says Brush started from an open-weights model, was post-trained on real design and editing tasks and then trained with large-scale reinforcement learning; the reward system scores visual quality, adherence to layout principles and whether the task was completed as asked, with "good" defined by professional designers on staff; at run time an agentic loop writes and runs code to adjust design elements and then inspects the page to choose its next step, and everything it produces stays fully editable[4].

The key mechanism is editing through code and checking the rendered result. A Canva design is a structured object, so an edit can be expressed as operations on that structure and the result can be checked by a program. That makes q densely measurable during training and gives run-time self-checks something solid to check against. Canva reports that Brush matched or exceeded its previous setup on most editing tasks while being faster and cheaper (self-reported, not independently evaluated)[4]. General-purpose models are capable at such edits but costlier per call, and at the scale of hundreds of millions of users and billions of designs, a few extra cents per task adds up — Canva's own rationale[4].

The route has clear limits: it needs structured, checkable outputs and rewards that domain experts can define. On open-ended writing or complex reasoning, which resist programmatic checking, a vertical model's edge over a general one narrows. Research on reasoning tasks finds a saturation point for inference-time compute, beyond which accuracy gains diminish, and identifies mixture-of-experts architectures as a good balance of accuracy and compute[7]; cascades that keep easy queries on small models and escalate only hard ones can bound the probability that inference cost exceeds a budget[8]. Vertical models and cascade routing both push c down on the condition that q stays high enough.

View 1 (12–24 months): When system-level entry points call third-party models, the baseline configuration will be per-call consent, a commitment not to train on user data and context minimisation, and competition will shift from model scores to how verifiable the privacy mechanisms are. Support: the consent design disclosed by Apple and Alibaba[1,2] and

demonstrated leakage in cloud–edge settings[6]. Falsified if major device entry points start calling third-party models by default without prompting, or make training on user data the default. Implication: when embedding external models in your own products, make an auditable per-call consent record an acceptance criterion rather than an afterthought.

2. Unit inference cost: three ways to cut it, and how each can hurt success

There are many ways to lower c, but by what they save they fall into three families: process less input (skip or prune redundant input tokens), emit more per step (multi-token prediction, speculative decoding, parallel decoding in diffusion models), and compute each step more cheaply (low-bit quantization). All three post attractive averages in papers, but they damage q in different ways, and buyers need to test for each separately.

Process less input: LOWN's agree-then-reuse certificate

Diffusion vision-language models process their entire input at every denoising step. The LOWN authors find that visual tokens make up 89–97% of the input on the benchmarks they study, so most decoding compute goes into reprocessing the same image[9]. They ran a direct measurement — predicting with and without the image at every decoding step — and found that removing the image changed a token being unmasked at only 22.8% of steps on average and at as few as 7.9% on one benchmark[9].

LOWN's mechanism can be written as a certificate. Let $\hat{y}_t(i)$ be the text-only prediction at step t for masked position i and $\hat{y}_s(i)$ the most recent image-conditioned prediction ($s < t$). If $\hat{y}_t(i) = \hat{y}_s(i)$, the earlier prediction is certified for reuse without looking at the image again; the image is reprocessed only when the text-only prediction disagrees with the last image-conditioned one. The number of certified tokens also sets how many tokens are unmasked in that step[9]. According to the paper, the method needs no training and no tuned threshold, retains about 99.6% of LLaDA-V's score across ten benchmarks at about 35.3% of the decoding FLOPs on average, and composes orthogonally with visual-token pruning[9].

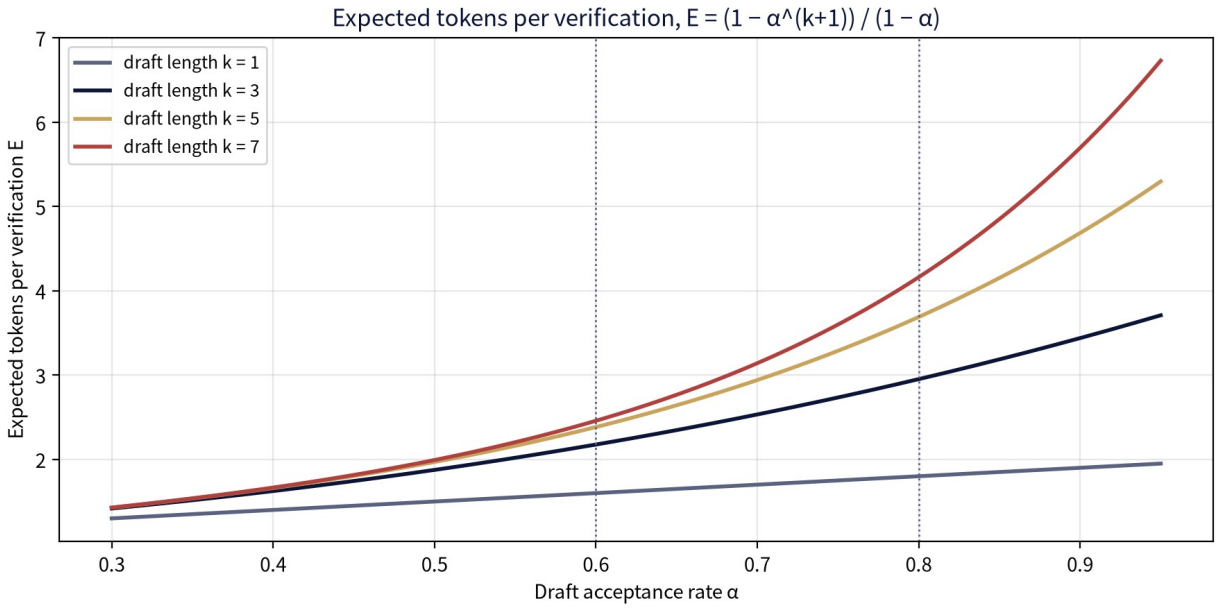
Why does this work? A rough estimate helps (illustrative, recomputable). If visual tokens are 93% of the input, a text-only forward pass costs about 7% of a full pass (ignoring attention non-linearity). If 22.8% of steps need the image, the average cost per step is about $0.07 + 0.228 \times 1 \approx 0.30$ of a full pass. Add the step reduction from unmasking more tokens when more are certified, and the order of magnitude matches the reported 35.3%. In essence, LOWN pays for

a cheap counterfactual check — would the answer change without the image? — before deciding whether to pay for expensive computation.

Against visual-token pruning, the trade-off is clear. Pruning permanently discards some visual information and still processes the rest of the image at every step; LOWN discards nothing and instead skips unneeded computation over time. Pruning's risks are now documented: when visual evidence is degraded, models can exploit textual shortcuts, so average accuracy hides whether the right visual tokens were kept, and identical token budgets can incur different end-to-end costs depending on where pruning happens[10]. An audit under paired image shifts found that reselecting retained positions on the shifted image raised the compressed model's failure rate by 2.05–9.60 percentage points relative to reusing clean-image positions, according to the paper[11]. LOWN has limits too: it currently targets LLaDA-family diffusion models, and its certificate relies on text-only and image-conditioned predictions being comparable; on tasks where image information matters at every step, such as dense visual grounding, the share of steps that need the image rises and savings shrink — the authors themselves observe that the need for the image varies widely by benchmark[9]. More generally, parallel decoding in diffusion models rests on a conditional-independence assumption, so quality must degrade when token dependencies are strong, and standard maths and coding benchmarks are not enough to expose it[12].

Emit more per step: the acceptance-rate constraint on speculation and MTP

Speculative decoding lets a cheap draft model propose several tokens that the target model verifies in parallel, accelerating generation without changing the output distribution[13]. With per-token acceptance probability α and draft length k, the expected number of tokens produced per verification is $E = (1 - \alpha^{(k+1)}) / (1 - \alpha)$[13]. With illustrative values $\alpha = 0.8$ and $k = 3$, $E \approx 2.95$; at $\alpha = 0.6$, $E \approx 2.18$. Acceptance moves the payoff non-linearly.



Source: formula from Leviathan et al. (2022); curves computed by the Research Team

Fig. 1 | Expected tokens per verification in speculative decoding as a function of acceptance rate. Longer drafts are more sensitive to acceptance. Formula from Leviathan et al.; curves are illustrative calculations.

Multi-token prediction (MTP) internalises the draft model as extra heads on the trunk. Early work showed that predicting the next n tokens with n independent heads improves sample efficiency and makes inference up to about three times faster[14]; DeepSeek-V3 adopted MTP as a pre-training objective[15]. The Soft-MTP paper of 28 October identifies a flaw in standard MTP training: the MTP cross-entropy gradient decomposes exactly into a KL self-distillation term that pulls each MTP head towards the model's own next-token distribution — helpful for speculative decoding — and a low signal-to-noise residual that buries the useful KL signal and interferes with both next-token and multi-token learning, increasingly so with scale[16]. In symbols:

$\nabla L_CE(MTP) = \nabla KL(p_NTP \parallel q_MTP) + r$, where p_NTP is the model's own next-token distribution and r is a low signal-to-noise residual.

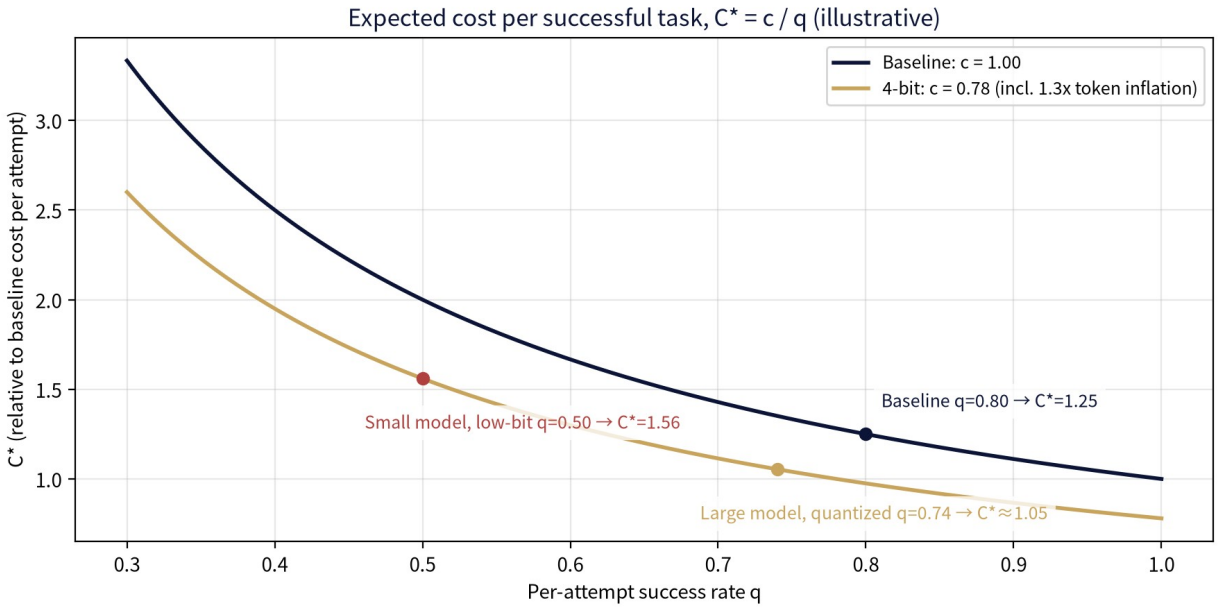
The authors' KL gradient-recovery trick exploits the fact that when two distributions share a linear output head, the KL gradient can be recovered exactly by combining their existing cross-entropy gradients through straight-through routing, without materialising vocabulary-sized logit matrices. The resulting Soft-MTP is a one-line change that adds no parameters and, according to the paper, beats standard MTP from 167 million to 1.2 billion parameters[16]. That range is small; whether the result carries over to pre-training at tens of billions of parameters still has to be shown.

The payoff of emitting more per step is often overstated in production. A systematic study on the production-grade vLLM engine found that verification by the target model dominates execution, that acceptance length varies widely across positions, requests and datasets, and that measured speed-ups fall well short of theoretical bounds[17]. A controlled single-GPU test on an A10G found that two-token MTP raised output throughput by 1.91–2.19 times with median acceptance lengths of 2.37–2.60, according to the paper; yet the MTP execution path was longer and more complex, its dominant matrix kernel was not faster than the autoregressive one, and the gain came from fewer repeated executions[18]. Another line converts a pretrained model into a standalone multi-token predictor by distillation, decoding more than three times faster on GSM8K with less than a 5% accuracy drop, according to the paper[19] — and in many businesses a 5% drop is enough to cancel the speed-up.

Compute each step more cheaply: two hidden costs of quantization

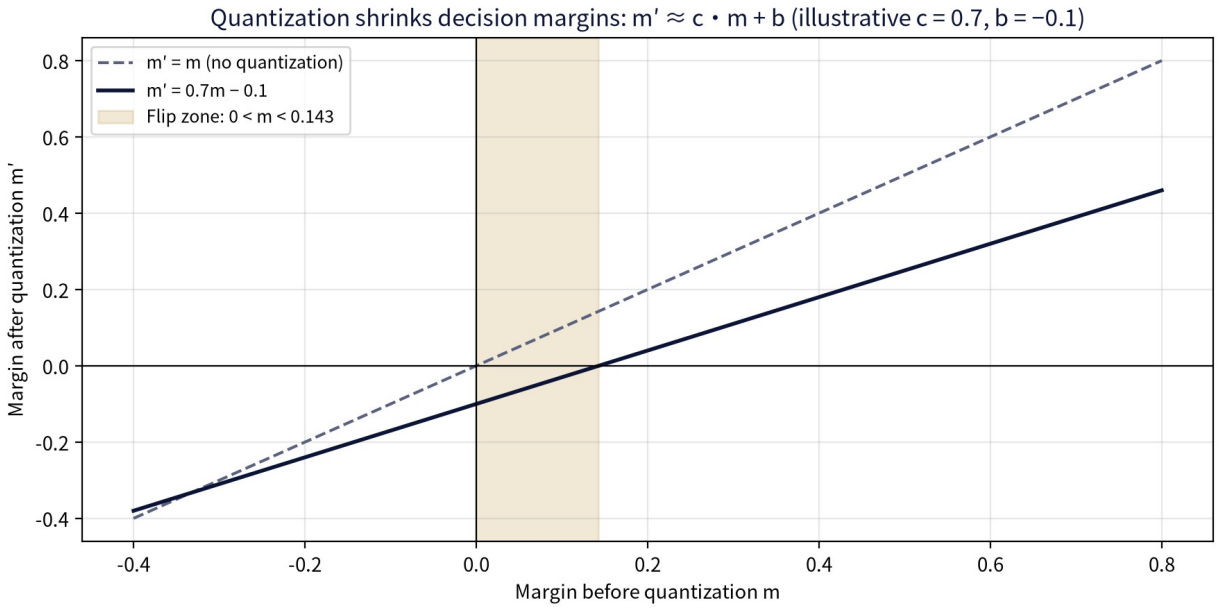
Quantization is an intuitive saving: halve the bits and you halve memory and bandwidth. Two recent studies expose its hidden costs through the c/q lens. The first is token inflation: low-bit reasoning models often generate longer chains of thought even when they still answer correctly, so INT4/INT3 quantization can preserve accuracy while increasing reasoning-token usage and offsetting the per-token speed-up[20]. The second is margin shrinkage: across 16 models at 4, 3 and 2 bits, the score gap between two candidate first tokens before and after quantization follows approximately $m' \approx c \cdot m + b$, with slope c usually below one and falling as precision drops; in tool use, whether to call a tool is often more sensitive than which tool to call[21].

Put both into $C^* = c/q$ (illustrative, recomputable). Baseline: $c = 1.00$, $q = 0.80$, $C^* = 1.25$. Four-bit quantization cuts per-token cost to 0.60, but reasoning tokens inflate 1.3 times, so $c = 0.78$; if q falls to 0.74, $C^* \approx 1.05$ and the trade is still worth it. For a small model's low-bit build where q falls to 0.50, $C^* = 1.56$ — 25% more expensive than not quantizing. Now the margins: with $c = 0.7$ and $b = -0.1$, positive decisions whose original margin lies between 0 and $-b/c \approx 0.143$ flip after quantization. If 12% of a "call the tool or not" decision distribution sits in that band, about 12% of call decisions reverse while average accuracy may drop by only a point or two.



Source: computed by the Great Unicorn Intelligence Research Team; token inflation per Lian et al. (2026)

Fig. 2 | Expected cost per successful task, C* = c/q. Quantization lowers c, but if q falls too far, C* rises. Illustrative data computed by the Research Team.



Source: linear relation per Wu et al. (2026); parameters illustrative

Fig. 3 | Quantization shrinks decision margins as m′ ≈ c • m + b; positive decisions in the shaded band flip. Linear relation from Wu et al.; parameters illustrative.

Table 2 | Mechanisms, gains and risks to success of three inference cost-cutting routes (compiled from the cited papers)

Route	Mechanism	Typical gain (per papers)	Main risk to q	Where it fits
Skip input (LOWN)	Certificate decides whether to re-read the image	~35.3% of decoding FLOPs, ~99.6% of score retained	Smaller savings when the image matters at every step	Diffusion VLMs, image QA

Visual-token pruning	Drop low-importance tokens	Varies by method	Text shortcuts mask errors; failures rise under shift	Redundant inputs, low need for visual detail
Emit more (speculation/MTP)	Draft plus parallel verification	~1.9–2.2x single-request throughput	Acceptance varies by task; gains shrink at large batch	Predictable outputs, latency-sensitive
Low-bit quantization	Lower numeric precision	Large memory and bandwidth savings	Token inflation; margin shrinkage; small models break	Large models, non-critical decisions

View 2 (6–18 months): Enterprises will buy and evaluate inference by cost per verified successful task rather than price per million tokens, and subscriptions and enterprise contracts will move from short-window throttling towards weekly or monthly budgets. Support: the redesign of capacity for long sessions[3], token inflation and decision flips under quantization[20,21], and the gap between measured and theoretical speed-ups[17]. Falsified if leading providers re-tighten short-window caps, or if per-token prices fall so much faster than success rates change that per-token pricing remains a good proxy for task cost. Implication: load tests for vendor selection should log token usage, retries and human-review rates, and compare options on c/q, not unit price.

3. Learning signals: where cheap supervision comes from without a strong teacher

Upstream of inference sits training cost. A main thread of post-training over the past year is on-policy self-distillation (OPSD): the model learns, on its own sampled trajectories, from a copy of itself that has seen the reference answer. The intuitive story is that the teacher holds privileged information the student lacks. Several recent dissections challenge that story, and WISD goes in the opposite direction.

Privileged information is not the whole story: three dissections

First, one study replaced the paired reference with a problem and solution from a different example while keeping the student rollout, teacher and objective fixed; it still beat the base model and remained competitive with OPSD, which led the authors to conclude that OPSD gains do not necessarily come from the reference solution and that context-induced teacher behaviour matters[22]. Second, another study built 5,319 maths problems with six reasoning views each and found that matched reference-free distillation accounts for much of Qwen3-1.7B's improvement, that the extra benefit of references is modest, and that gains depend

strongly on the student: replacing short direct-response rollouts with long thinking rollouts turned gains into losses[23]. Third, a further study found that OPSD gains shrink rapidly with scale — from 3.05 points at 1.7B to 0.14 at 8B, according to the paper — because the teacher can use information unavailable to the student, creating an imitation gap; supervising mainly verified trajectories kept gains at 3.2–3.8 points from 1.7B to 8B[24].

Taken together, the three point to a plainer explanation than privileged information: the useful signal comes from contrasting the same model under two conditions, and from putting supervision on the right trajectories.

WISD: contrast with a weaker self

Rather than building a stronger teacher, WISD builds a weaker counterpart through low-bit quantization or magnitude pruning and uses the contrast between the original and the weakened model as the training signal, with no privileged information[25]. For intuition the signal can be written schematically (our illustrative notation, not the paper's formula): on the student's own sample y, compare log π_θ(y_t | ·) − log π_weak(y_t | ·). Positions with a large gap are the decisions lost first when capacity is degraded — and the parts of the model's ability that deserve reinforcing.

According to the paper, quantization and pruning raise average accuracy on maths and coding reasoning by 4.0 and 5.5 points respectively; pruning lifts Qwen3.5-4B from 57.7% to 70.5% on HMMT 2025 and from 39.1% to 56.7% on LiveCodeBench v6; quantization lifts Qwen3.6-35B-A3B from 68.5% to 74.6% and from 61.8% to 69.5% on the same benchmarks[25].

Why would weakening provide a useful signal? A plausible explanation is that quantization and pruning do not damage ability uniformly; they break first the decisions with small margins that depend on fine numeric detail — consistent with the margin-shrinkage finding above[21]. The weakened model therefore acts as a probe of where the model is fragile, and the contrast concentrates training on those fragile decisions. This also explains its complementarity with OPSD: OPSD shows the model what a better self would do; WISD shows where a worse self goes wrong.

The route has three limits. The reported scales are a 4B dense model and a 35B-total, 3B-active mixture-of-experts model, and the shrinking OPSD gains above[24] are a reminder that independent replication at larger scale is needed. The tasks are maths and code with verifiable answers; open-ended tasks are untested. And the contrast works only if the

weakening happens to degrade the abilities worth reinforcing; another weakening method or model family could behave differently.

Representational convergence: view design matters more than scale

The same week brought a theoretical paper that turns the Platonic Representation Hypothesis from conjecture into theorem. The hypothesis holds that models trained on different modalities converge towards similar representations as they scale[26]. The new paper proves that joint-embedding predictive architectures (JEPAs) trained separately on each modality learn the same representation up to one rotation whenever positive pairs are two observations of one scene ("good views"); next-token prediction is a JEPA with bad views — parts of a single document — which explains why language models align with vision only through scale[27]. According to the paper, good views need 42 times fewer training samples than next-token prediction on the same corpus to reach the same alignment, and a 109-million-parameter text JEPA aligns with DINOv2 and CLIP image encoders better than every language model the authors evaluated, up to 72 billion parameters[27].

The counter-evidence carries weight. One study shows that common similarity metrics are inflated by network scale itself; after permutation-based null calibration, the convergence shown by global spectral measures largely disappears, and only local neighbourhood relations stay consistent across modalities[28]. Another argues that the original evidence is weaker than later citations imply: the mutual nearest-neighbour metric captures only coarse structure, the rise in alignment with language-model strength saturates for recent models, and one-to-one image–text pairing itself favours alignment[29]. A third, across 16 language models, finds that representations are more similar on problems the models collectively fail, that post-decision representations diverge sharply, and that shared information has little causal effect on predictions[30].

Weighing both sides, our reading is that convergence is real at the coarse and local level and is driven more by the design of training views than by parameter count. The lesson for enterprises is concrete: for multimodal retrieval, image–text alignment or cross-system representation transfer, improve the quality of paired data — make positives genuinely two observations of the same object — before reaching for a bigger encoder.

View 3 (12–24 months): Teacher-free self-contrast and self-distillation will become a standard option for domain post-training in enterprises, because they need neither calls to a

stronger model nor heavy labelling; the main controversy will be whether gains fade with scale. Support: WISD's reported results[25] and the three dissections[22–24]. If independent replications on dense or mixture-of-experts models above 30B show average gains below one point, the view should be narrowed to small and mid-sized models. Implication: put domain post-training budgets first into checkable tasks and answers, the precondition for these methods to work.

4. Verifiability: environments and artifacts are the new bottleneck

The previous sections were about lowering c. But C* = c/q carries a hidden premise: q must be measurable. For agents, measuring q depends on training and evaluation environments; for open-model deployment, it depends on whether the downloaded artifact actually works. Both are where this week's evidence is densest and where attention is thinnest.

Training environments: source reconstruction, verified synthesis and generated worlds

Web agents need diverse, executable, resettable website environments. The WebAnything authors sort existing routes into three: human-assisted mirrors need per-site engineering; generative construction can drift from the specified real site; capture-and-replay preserves observations, but interaction coverage is limited to what was captured[31]. Their approach starts from real URLs: an Explorer-Builder collects browser-observable structures, resources and interaction outcomes, and an independent verifier checks frozen candidates and guides local repair or targeted recapture. According to the paper, they built 244 website artifacts across 14 application domains and 53 categories, supporting 12,984 executed and accepted tasks; in a blind audit against the human-assisted WebHarbor they won 12, tied two and lost none on three fidelity metrics; and fine-tuning Qwen3.5-9B on 5,856 interaction trajectories raised WebArena-Verified overall success by 13.33 points under a fixed harness[31].

A second route is verified synthesis. One study represents each generated website as a structured scaffold of pages, navigation, database records, state-change markers and task constraints, and checks and repairs structural, semantic, consistency and feasibility defects before training; according to the paper, the feasible-task rate rose from 48.6% to 94.8%, and policies trained on the result transferred better to WebArena, WebShop and others[32]. That number deserves a pause: in unverified synthetic environments, more than half the tasks

could not be completed at all. Train on them and a model learns not to finish tasks but to behave plausibly on impossible ones, and the measured q is distorted.

Environments also change. A benchmark that simulates contextual, perceptual and execution shifts finds that web agents' learned skills degrade systematically, with execution shifts exerting the strongest pressure[33]. An environment is therefore not a one-off asset; it must be maintained and regression-tested like software.

Generated world models sit at the other end. Genie 4, DeepMind's new real-time interactive world model, natively generates ambient sound, music and dialogue, lets users change the weather, move the camera or add a person in natural language, and uses Gemini models between inputs to monitor and steer the scene for coherence; one stated purpose is to build training environments for robots and agents, and it is rolling out as a research preview to trusted testers[34]. The limitations are spelled out: it can keep environments coherent over short sessions, but visual details and object permanence can still shift or drift over time, especially in longer runs, after several complex prompt edits or while navigating the scene; complex physical interactions, real-time audio accuracy, legible text, spatial accuracy and long-term consistency remain open research[34]. An independent benchmark agrees: in long-horizon exploration of up to 180 seconds with camera actions chosen online by an agent, ten video world models sustain local visual quality for longer, but long-range memory and evolution remain weak[35].

The trade-off among the three routes comes down to whether state can be checked. Source-reconstructed and verified-synthetic environments have back-end state; a program can decide whether a task was completed, so rewards can be dense and trustworthy. A generated world model outputs pixels and sound with no ground-truth state to query, so rewards must come from another model's judgement, and errors compound. That does not diminish world models — in embodied AI, perception training and settings where physical environments are impractical, they supply data diversity available nowhere else — but for training software agents, checkable state matters more than photorealism.

Artifact quality: reproducible is not the same as working

The end of the open-model delivery chain is the quantized file a user downloads. The Broken on Arrival author tested all 305 eligible code-capable GGUF tags in the official Ollama library, across 15 model lines, plus 22 popular Hugging Face community files, and found that none of

the registries examined documented functional testing[36]. After a 15-task code screen and a retest on all 164 HumanEval+ tasks, seven official tags (2.3%) solved at most six problems, all of them low-bit quantizations of models with at most four billion parameters[36]. The sharper finding: five suspect files rebuilt from the original weights matched the registry files tensor for tensor, so nothing was corrupted — exact reproducibility did not imply function; with weights, toolchain and tensor mix held fixed, adding an importance matrix (imatrix) raised Qwen2.5-Coder-3B from zero to 76–113 of 164 tasks at three quantization levels, while the same uncalibrated recipe left the smaller 1.5B model functional, so the recipe alone does not predict failure[36].

A supply-chain study supplies the other half. Model signing authenticates full-precision weights, but users run a quantized artifact whose producing transformation is not uniquely determined under current disclosure. Across 198 in-scope GGUF repositories among the 200 most downloaded, none recorded the build environment, so the share a third party could re-derive bit for bit from recorded inputs was 0/198; adding two omitted coordinates — the floating-point contraction setting and the per-tensor quantization recipe — re-derived 7/8 target files bit for bit on x86[37]. Quality metrics also fail as a safety proxy: in one matrix of quantized checkpoints, quality held or improved while refusal fell by 12–68 points, according to the paper[38]. On the serving side, prefix-cache reuse can make the same request produce different answers, and quantization amplifies the divergence: for Qwen2.5-7B, cache-path divergence reached 37.4% of multi-turn tool-use episodes at 16-bit and 82.5% at four-bit, according to the paper[39].

Back in the C* frame, a broken artifact silently drives q to zero after deployment, and neither average benchmark scores nor file hashes raise an alarm. A 2.3% rate sounds low, but a company whose business units each download and deploy low-bit small models faces non-trivial odds of meeting at least one broken artifact (illustrative, recomputable: with an independent 2.3% failure rate per artifact, the probability of at least one failure among 15 is $1 - 0.977^{15} \approx 29.5\%$). A smoke test costs a few dozen inferences — negligible by comparison.

View 4 (12–24 months): For training and evaluating software agents, source-reconstructed environments with state checks, or verified synthetic ones, will become the workhorse, while generated world models will find their main use in embodied and perception simulation; low-bit artifacts will increasingly ship with functional smoke tests and build metadata. Support:

the environment studies in this section[31,32], the official limitations and an independent benchmark of world models[34,35], and the artifact audits[36,37]. The relevant parts are falsified if public results show generated world models, used as the main training environment, consistently beating source-reconstructed environments on WebArena-style benchmarks, or if major model registries still disclose no functional testing 12 months from now. Implication: acceptance checklists for private deployment should include a post-download functional smoke test, archived build parameters and a safety-refusal regression, and should not treat matching hashes as proof of fitness.

5. Physical compute: where an orbital "Hello World" sits on the chain

Finally, the top of the chain: energy and compute. On 30 October, Google Research reported that the Project Suncatcher prototype satellite launched on 1 October had completed commissioning, including powering up host and TPU trays and running early machine-learning tests; during a ground-station pass it uplinked a prompt to the onboard Gemma 4 open model and, as the satellite passed Svalbard, Norway, downlinked a haiku about space — described as the first AI response from a data-centre-class TPU in orbit. Next come tests of TPU behaviour under thermal stress and radiation, paving the way for twin interconnected satellites in 2027[40].

Weighing this step requires going back to the system design published in 2025[41,42]. It envisions constellations of solar-powered satellites carrying TPUs in a dawn–dusk sun-synchronous low orbit, linked by free-space optics. The key constraints, one by one:

- Inter-satellite bandwidth: distributed training at data-centre level needs links of tens of terabits per second; because received power falls with the square of distance, the design requires satellites to fly in very close formation, kilometres or less apart; a bench demonstrator has reached 800 Gbps each way with a single transceiver pair[41].
- Radiation: in a 67 MeV proton beam, Trillium TPU high-bandwidth memory began showing irregularities only after a cumulative 2 krad(Si), about 2.7 times the expected shielded five-year mission dose of 750 rad(Si), and no hard failures attributable to total dose appeared up to 15 krad(Si) on a single chip[41].
- Economics: with a sustained learning rate, launch prices may fall below US$200 per kilogram by the mid-2030s, at which point launch and operating cost per kilowatt-year could be roughly comparable to the energy cost of an equivalent terrestrial data centre;

the post also lists thermal management, high-bandwidth ground communication and on-orbit reliability as open problems[41].

An independent analysis draws a more conservative line. Comparing launch, power, cooling, radiation and re-entry, and modelling inter-satellite mesh networks with bisection bandwidth and roofline methods, it concludes that inference in low orbit may be feasible but that training frontier-scale models in orbit is unlikely to compete with terrestrial data centres[43]. A workload-first framework adds that processing Earth-observation imagery in orbit and downlinking compact semantic products cuts payload by 99.7–99.99%, and that such in-situ processing is what suits orbit first[44].

So this "Hello World" validates that a data-centre-class accelerator can power up in orbit, run an open model and return a result — not that orbital compute can stand in for terrestrial clusters. On the delivery chain it is a long-run energy-cost item and, in the near term, in-situ inference on data generated in orbit.

View 5 (24 months): Through the end of 2028, orbital compute will handle only in-situ inference on orbital data and technology validation; it will not enter general large-model training or consumer-facing inference. Support: the engineering challenges and timeline stated by Google[40,41] and independent cost and network analyses[43,44]. Falsified if the 2027 twin-satellite mission publicly demonstrates distributed training across satellites with inter-satellite bandwidth at the scale of a terrestrial training cluster. Implication: compute plans need not reserve budget for space compute, but firms in remote sensing, maritime or emergency response can start assessing pipelines that pre-process in orbit and finish on the ground.

Conclusion: move budget from "bigger" to "more verifiable"

Back to the thesis. Every piece of evidence here lands on C* = c/q and the reach N: system-level distribution and consent design set N and the consent rate; capacity design and vertical post-training change the structure of c; acceleration and quantization lower c but can depress q through token inflation, decision flips and volatile acceptance; teacher-free self-distillation lowers the cost of learning signals on the training side; the verifiability of environments and artifacts decides whether q can be measured at all; and orbital compute is a long-run energy term.

Several common beliefs therefore need revising. "Quantization saves money" holds only if q does not fall much. "Representational convergence means scale will eventually unify modalities" rests on weaker evidence than its citations suggest; view design matters more than scale. "World models will replace real environments" is premature for software-agent training, where checkable state beats photorealism. "A matching hash means the file is safe to deploy" has been refuted empirically. "Space compute will soon ease the energy bottleneck" confuses technical validation with commercial readiness.

For enterprises making decisions this quarter, Table 3 offers a self-check.

Table 3 | Delivery-chain self-check for enterprises (compiled by the Research Team)

Link	Question to answer	Measurable indicator	Failing signal
Entry and consent	Is every external model call covered by purpose-bound consent?	Consent-record coverage	Consent only once at sign-up
Unit cost	Are options priced per successful task?	c/q, token usage, retry rate	Comparing token prices only
Side effects of cost cuts	Do key decisions flip after acceleration or quantization?	Decision-flip rate, refusal regression	Average accuracy only
Learning signal	Does post-training have checkable tasks and answers?	Share of verified samples	Mostly unverified synthetic data
Environments	Do training and evaluation environments expose queryable state?	Feasible-task rate, environment regression pass rate	Cannot tell whether a task was completed
Artifacts	Is there a functional smoke test before deployment?	Smoke-test pass rate, archived build parameters	Hash match in place of functional checks

In our own work at Great Unicorn Intelligence, these checkpoints sit in the governance layer that runs agents. Our enterprise agent platform's Harness governance engine includes a model router (selects a model per task and pins its version, forbidding automatic upgrades mid-run so runs are reproducible), a tool gateway (schema validation, authorisation, allow-listing, rate limiting, masking and idempotency control on every call) and an independent verifier (re-checks inputs, process, results and actions layer by layer); our training–inference platform DH S1 supports private deployment and provides metric monitoring and A/B testing on its online path. These map to the consent, cost, artifact and verification rows of Table 3 (product details per the Great Unicorn Intelligence website).

Key takeaways:

1. The race is shifting from parameter count to the delivery chain, and the common metric is expected cost per successful task, C* = c/q.
2. Each of the three ways to lower c has its own way of hurting q; test for decision flips, token usage and acceptance, not just average scores.
3. The useful signal in self-distillation comes from contrast and from the right trajectories; a weaker self can be a good teacher, but extrapolation to scale is unproven.
4. Checkable environments and functionally tested artifacts are preconditions for measuring q.
5. In the near term, orbital compute handles only in-situ inference and validation.

References

1. Apple. Apple 智能即将登陆中国. Apple Newsroom (China), https://www.apple.com.cn/newsroom/2026/10/apple-intelligence-features-are-coming-to-users-in-china/ (2026).
2. Xinhua. Alibaba's Qwen AI model to integrate into Apple Intelligence in landmark partnership. https://english.news.cn/20261101/773fa2ac33a34a5ca3484d7e5920d094/c.html (2026).
3. Anthropic. Plans & pricing (Claude Max and Ultra). https://claude.com/pricing (2026).
4. Canva Research Team. Introducing Canva Brush, a vision-language model for design. Canva Newsroom, https://www.canva.com/newsroom/news/canva-brush/ (2026).
5. Apple Security Engineering and Architecture et al. Private Cloud Compute: a new frontier for AI privacy in the cloud. Apple Security Research Blog, https://security.apple.com/blog/private-cloud-compute/ (2024).
6. Zhang, K., Zou, T., Gu, Z. & Liu, Y. Auditing and Mitigating Privacy Leakage in Cloud-Edge Collaborative Decoding. Preprint at https://arxiv.org/abs/2608.29111 (2026).
7. Prucs, Á., Csutora, N., Antal, M. & Marosi, M. Compute-Accuracy Pareto Frontiers for Open-Source Reasoning Large Language Models. Preprint at https://arxiv.org/abs/2512.24776 (2025).
8. Valkanas, A., Pal, S., Rumiantsev, P., Zhang, Y. & Coates, M. C3PO: Optimized Large Language Model Cascades with Probabilistic Cost Constraints for Reasoning. Preprint at https://arxiv.org/abs/2511.07396 (2025).

9. Cho, D., Kim, J., Shin, S. & Yang, E. Look Only When Needed: Certificate-Guided Visual Routing for Efficient Diffusion VLMs. Preprint at https://arxiv.org/abs/2610.36671 (2026).
10. Li, G., Chen, G., Shan, S. & Yang, J. Are We Evaluating VLM Token Pruning Right? Looking Beyond Accuracy and Token Count. Preprint at https://arxiv.org/abs/2610.24740 (2026).
11. Isaac & Li. VTC-AUDIT: An Interventional Reliability Audit for Visual-Token Compression under Paired Image Shifts. Preprint at https://arxiv.org/abs/2610.34545 (2026).
12. Kang, W. et al. ParallelBench: Understanding the Trade-offs of Parallel Decoding in Diffusion LLMs. Preprint at https://arxiv.org/abs/2510.04767 (2025).
13. Leviathan, Y., Kalman, M. & Matias, Y. Fast Inference from Transformers via Speculative Decoding. Preprint at https://arxiv.org/abs/2211.17192 (2022).
14. Gloeckle, F., Idrissi, B. Y., Rozière, B., Lopez-Paz, D. & Synnaeve, G. Better & Faster Large Language Models via Multi-token Prediction. Preprint at https://arxiv.org/abs/2404.19737 (2024).
15. DeepSeek-AI et al. DeepSeek-V3 Technical Report. Preprint at https://arxiv.org/abs/2412.19437 (2024).
16. Teoh, J. Multi-Token Prediction Done Right. Preprint at https://arxiv.org/abs/2610.38331 (2026).
17. Liu, X., Yu, J., Park, J., Stoica, I. & Cheung, A. Speculative Decoding: Performance or Illusion?. Preprint at https://arxiv.org/abs/2601.11580 (2025).
18. Sah, S. P. Beneath the Tokens: A Performance Engineering Study of Multi-Token Prediction in GPU-Accelerated LLM Inference. Preprint at https://arxiv.org/abs/2609.35188 (2026).
19. Kirchenbauer, J. et al. Multi-Token Prediction via Self-Distillation. Preprint at https://arxiv.org/abs/2602.06019 (2026).
20. Lian, X. et al. Quantization Inflates Reasoning: Token Inflation as a Hidden Cost of Low-Bit Reasoning Models. Preprint at https://arxiv.org/abs/2606.25519 (2026).
21. Wu, Z., Dhiman, S. & Koshiyama, A. Which Decisions Low-Bit Quantization Breaks, and How to Predict Them. Preprint at https://arxiv.org/abs/2608.06564 (2026).
22. Ichihara, Y., Iwase, N., Quamar, M. A. & Komiyama, J. Privileged Solutions or Context-Induced Teacher Behavior? Dissecting On-Policy Self-Distillation. Preprint at https://arxiv.org/abs/2608.09228 (2026).

23. Zhang, X. et al. What Does Privileged Information Add to On-Policy Self-Distillation?. Preprint at https://arxiv.org/abs/2609.20612 (2026).
24. Hossain, M. I., Kousar, H. & Tourni, I. C. Overcoming Scaling Limits in On-Policy Self-Distillation for LLM Reasoning. Preprint at https://arxiv.org/abs/2609.37915 (2026).
25. Wu, Y., Han, S. & Cai, H. Weaken to Strengthen: Weakening-Induced On-Policy Self-Distillation for Large Reasoning Models. Preprint at https://arxiv.org/abs/2610.38681 (2026).
26. Huh, M., Cheung, B., Wang, T. & Isola, P. The Platonic Representation Hypothesis. Preprint at https://arxiv.org/abs/2405.07987 (2024).
27. Balestriero, R. & LeCun, Y. Proving and Improving the Platonic Representation Hypothesis with JEPAs. Preprint at https://arxiv.org/abs/2610.39302 (2026).
28. Gröger, F., Wen, S. & Brbić, M. Revisiting the Platonic Representation Hypothesis: An Aristotelian View. Preprint at https://arxiv.org/abs/2602.14486 (2026).
29. Koepke, A. S., Zverev, D., Ginosar, S. & Efros, A. A. Back into Plato's Cave: Examining Cross-modal Representational Convergence at Scale. Preprint at https://arxiv.org/abs/2604.18572 (2026).
30. Usama, M. & Chang, D. E. Convergence Without Understanding: When Language Models Agree on Representations but Disagree on Reasoning. Preprint at https://arxiv.org/abs/2605.23315 (2026).
31. Xie, T. et al. WebAnything: From the Wild Web to Scalable Agent Environments. Preprint at https://arxiv.org/abs/2610.37786 (2026).
32. Zhang, C. et al. Training Needs Trustworthy Worlds: Verified Synthetic Web Environments for Agent Learning. Preprint at https://arxiv.org/abs/2608.21898 (2026).
33. Chen, B., Zhou, Z. & Li, Y. SkillShift-Bench: Benchmarking Skill Learning under Environment Evolution for Web Agents. Preprint at https://arxiv.org/abs/2610.33528 (2026).
34. Hadsell, R., Rivas, D. & Qi, H. Genie 4: a new generation of world simulation. Google DeepMind Blog, https://deepmind.google/blog/genie-4-a-new-generation-of-world-simulation/ (2026).
35. Wang, M. et al. WorldProbe: When Video World Models Fall Apart under Long-Horizon Spatial Exploration. Preprint at https://arxiv.org/abs/2610.39721 (2026).

36. Patodiya, A. Broken on Arrival: Non-Functional Quantized LLM Artifacts in Public Model Registries and How to Catch Them. Preprint at https://arxiv.org/abs/2609.05881 (2026).
37. Liu, W. & Tsai, C. What You Sign Is Not What You Run: The Verifiability Gap in Quantized LLM Supply Chains. Preprint at https://arxiv.org/abs/2610.19498 (2026).
38. Kadadekar, S. Quality Is Not a Safety Proxy Under Quantization. Preprint at https://arxiv.org/abs/2606.10154 (2026).
39. Patodiya, A. Same Request, Different Answer: Quantization Amplifies Cache-Induced Divergence in LLM Serving. Preprint at https://arxiv.org/abs/2609.04748 (2026).
40. Beals, T. We've received our first response from the Project Suncatcher prototype satellite. Google Blog, https://blog.google/innovation-and-ai/models-and-research/google-research/project-suncatcher-satellite-response/ (2026).
41. Beals, T. Exploring a space-based, scalable AI infrastructure system design. Google Research Blog, https://research.google/blog/exploring-a-space-based-scalable-ai-infrastructure-system-design/ (2025).
42. Agüera y Arcas, B. et al. Towards a future space-based, highly scalable AI infrastructure system design. Preprint at https://arxiv.org/abs/2511.19468 (2025).
43. Berkel, K. v. The Cost and Network Limits of Space-Based AI Compute. Preprint at https://arxiv.org/abs/2607.14172 (2026).
44. Singh, D. N. Which Workloads Belong in Orbit? A Workload-First Framework for Orbital Data Centers Using Semantic Abstraction. Preprint at https://arxiv.org/abs/2603.20317 (2026).

Copyright Notice: © 2026 Great Unicorn Intelligence Technology (Shenzhen) Co., Ltd. All rights reserved. Any reposting, quotation in commentary, or adaptation must credit the author (Great Unicorn Intelligence Research Team) and cite the source (Great Unicorn Intelligence, https://www.hxallai.com).