The Cost of Being Right

Verification Economics in 2026

Manu Bhardwaj

ifitsmanu.com / Field Notes

6 May 2026 (v1.0, Field Notes #2)

Abstract

Public LLM API prices declined sharply between 2022 and 2024 through four stack-level levers covered in the previous field note in this series. Beginning in late 2024 a fifth dynamic took hold. Reasoning models trained with reinforcement learning on verifiable rewards consume substantially more output tokens per task than their non-reasoning counterparts, and the multiplier is task-conditional, policy-controllable, and unbounded above. Recent benchmarks measure up to a 5x token-efficiency dispersion between models with comparable accuracy. The MIT FutureTech *Price of Progress* analysis documents both phenomena simultaneously. Per-benchmark-performance cost falls roughly 5x to 10x per year for frontier models, while the price of running frontier models rises 3x to 18x per year due to bigger models and larger reasoning demands. We argue that the operational unit of inference economics has shifted from cost-per-token to cost-per-correct-answer, and that the binding lever in this regime is verification. We extend the *Verified Capability per Dollar* (VCpD) framework to a *Cost-correct* decomposition with an explicit verification-accept-rate term, ground each component in the published RL-with-verifiable-rewards literature, and apply the framework to the GPT-5.5 reprice of April 2026 and the EU AI Act high-risk obligations entering force in August 2026.

1. The unit of account is shifting

The previous field note in this series argued that the 2023 to 2026 collapse in public API prices was driven by four compounding stack-level changes: weight-only quantization with matched mixed-precision kernels; memory-aware serving runtimes such as PagedAttention and continuous batching; speculative decoding and related decoding-time parallelism; and a hardware market in which GPUs, hyperscaler ASICs, and inference-specialty accelerators competed on delivered tokens per dollar rather than peak TOPS. The note introduced *Verified Capability per Dollar* (VCpD) as the operational unit of inference economics and noted, in a footnote, that GPT-5.5 raised prices in April 2026 for the first time in three years. That footnote is the starting point for this paper.

The headline trend in price-per-benchmark-performance has not reversed. The MIT FutureTech *Price of Progress* analysis ([6]) reports that the price for a given level of benchmark performance has decreased "around 5x to 10x per year" for frontier models on knowledge, reasoning, math, and software engineering benchmarks. In the same paper, a co-existing observation: "the price of running frontier models is rising between 3x to 18x per year due to bigger models and larger reasoning demands."

Both claims are simultaneously true. They are about different units. Per-benchmark-performance price falls. Per-task-running-cost rises. The reconciliation is the new variable.

Reasoning models trained via reinforcement learning to produce extended chains-of-thought before final answers consume substantially more output tokens per task than their non-reasoning predecessors. Three forces compose to make this the dominant cost driver in 2026.

First, reasoning is billed as output tokens. Across every major lab's public pricing schedule as of May 2026, internally generated chain-of-thought tokens are charged at the standard output rate. OpenAI's GPT-5.5 doubled per-token rates over GPT-5.4 on April 23, 2026, with input rising from $2.50 to $5.00 per million tokens and output rising from $15.00 to $30.00 per million ([16]). A reasoning model that emits a 50,000-token chain-of-thought before a 500-token final answer is a 100-to-1 reasoning-to-answer ratio billed entirely at the output rate. The economic signal is that the unit of work has shifted from the answer to the chain.

Second, the multiplier is large and variable. *OckBench* ([5]) reports up to a "5.0x difference in token length" between reasoning models that achieve similar accuracy on the same problem. Token efficiency is now a model-quality dimension as load-bearing as raw accuracy.

Third, accuracy ceilings are being purchased with unbounded test-time compute. The original test-time compute scaling paper ([1]) established that compute-optimal allocation of inference compute can "improve the efficiency of test-time compute scaling by more than 4x compared to a best-of-N baseline," and can "outperform a 14x larger model" in FLOPs-matched evaluation. The MCTS-and-process-reward-model paradigm, exemplified by *rStar-Math* ([14]), improves Qwen2.5-Math-7B from 58.8% to 90.0% on the MATH benchmark and Phi3-mini-3.8B from 41.4% to 86.4%, surpassing o1-preview at small scale, by spending test-time compute on tree-search through verifier-guided reasoning trajectories. The marginal correct answer is now bought with reasoning tokens, and the willingness-to-pay function is steep.

The right unit for inference economics in this regime is therefore not cost-per-token. It is cost-per-correct-answer.

2. The reasoning multiplier and where it points

Definition 1 (Reasoning multiplier)**.** *For a model m on a task t, the reasoning multiplier $R(m,t)$ is the ratio of total billed output tokens (chain-of-thought plus final answer) to final-answer-only output tokens for the same task. $R = 1$ for a non-reasoning model that emits only the answer. R is unbounded above for reasoning models that perform extensive search before responding.*

Three observations about R, each grounded in measured published data.

R is task-conditional. The same model exhibits very different R across math, code, agentic, and short-form QA. *OckBench*'s up-to-5x efficiency variance is at fixed task difficulty; cross-task variance is larger. A reasoning model on a single-fact retrieval task may emit R near 2 to 5. The same model on a multi-step proof or agentic trajectory may emit R well above 50.

R is policy-controllable but not free. Token efficiency is a tunable dimension of training and decoding. There is real engineering surface to compress R. There is also an empirical floor below which accuracy degrades on hard reasoning tasks. The compression is a tradeoff against the accuracy ceiling that test-time compute purchases.

R by itself does not bind cost-per-correct-answer. R multiplies tokens, but tokens only matter relative to whether they purchase correctness. Two models with $R = 30$ and identical token cost can produce dramatically different end-state economics if one accepts 90% of generated answers as correct on first attempt and the other accepts 30%. The multiplier and the accept rate must be considered together.

This is why the binding constraint in 2026 inference economics is not the multiplier. It is the accept rate. The multiplier is the cost. The accept rate is the value. The lever that controls the accept rate is verification.

3. Verification as the binding lever

Verification, in the relevant sense, is any process by which a generated continuation is evaluated for correctness. By another model. By a programmatic check. By a verifiable reward function during training. By self-consistency across samples. A verifier need not be a heavy model. In many practical deployments it is smaller than the generator.

The verifier-as-economic-lever observation is not new. Cobbe et al. ([10]) introduced the GSM8K benchmark together with the case for verifiers: "we propose training verifiers to judge the correctness of model completions. At test time, we generate many candidate solutions and select the one ranked highest by the verifier." The same paper provides "strong empirical evidence that verification scales more effectively with increased data than a fine-tuning baseline." Lightman et al. ([9]) strengthened the case with process supervision: a process reward model trained on PRM800K, "the complete dataset of 800,000 step-level human feedback labels," solves 78% of a representative MATH test subset, beating outcome-supervised baselines. Self-consistency ([11]) is a verifier-free version of the same idea: sample many reasoning paths, marginalize over them; the original paper reports a +17.9% lift on GSM8K versus greedy chain-of-thought.

What changed in 2024 to 2026 is that verification became a first-class component of post-training, not just inference. *Tulu 3* ([12]) introduced "a novel method we call Reinforcement Learning with Verifiable Rewards (RLVR)" as a named training procedure. The policy is trained against rewards that are programmatically verifiable, such as whether the math checks out, the code compiles, or the unit test passes. *DeepSeek-R1* ([2], published in *Nature* 645:633–638) demonstrated that "the reasoning abilities of LLMs can be incentivized through pure reinforcement learning, obviating the need for human-labeled reasoning trajectories," using verifiable mathematical rewards as the training signal. The OpenAI o1 system card ([4]) confirms the broader pattern: "the o1 model series is trained with large-scale reinforcement learning to reason using chain of thought." *DeepSeekMath* ([3]) introduced *Group Relative Policy Optimization* (GRPO), the variant of PPO that powered most subsequent verifier-based RL work.

The economic implication is precise. RLVR concentrates capital into verifier construction at training time so that inference-time generation produces a higher accept rate at the same R. *rStar-Math*'s process preference model ([14]) is the cleanest published example: a 7B base model becomes competitive with o1-preview specifically by being trained against and routed through a verifier. The verifier is small. The verifier is the economic lever.

4. Cost-correct: the decomposition

Definition 2 (Cost-correct)**.** *For a model m deployed under serving condition $s = (batch, context, SLO, verificat$ on task t, the Cost-correct is*

$$\text{Cost}_{correct}(m, s, t) \;=\; \frac{\text{CPM}_{1:1}(m) \cdot R(m, t) \cdot (1 + \bar{\rho}(m, s))}{\alpha(m, s, t, \theta, V)}, \tag{1}$$

where $\text{CPM}_{1:1}$ *is the blended public-API cost per million tokens used in the previous note (input price plus output price, divided by two);* R *is the reasoning multiplier from §2;* $\bar{\rho}$ *is the average rollout-or-rejection ratio under verifier-guided decoding (best-of-N, MCTS-at-decode, self-consistency), with* $\bar{\rho} = 0$ *for single-sample decoding and* $\bar{\rho}$ *approaching* $N - 1$ *for* N*-way verified rollouts; and* $\alpha(m, s, t, \theta, V) \in (0, 1]$ *is the verification accept rate at quality threshold* θ *under verifier* V*.*

Proposition 1 (Cost-correct extends VCpD)**.** *The previous note's* VCpD *corresponds to the special case*

$$R \to 1, \qquad \bar{\rho} \to 0, \qquad \alpha \to 1,$$

under which $\text{Cost}_{correct}$ *reduces to* $\text{CPM}_{1:1}$ *and* VCpD *becomes the inverse of cost weighted by quality, recovering the previous framework. Cost-correct extends VCpD by making the reasoning, rollout, and verification terms first-class denominators of the unit.*

The decomposition has three useful properties.

First, it absorbs the previous note. VCpD remains the operational unit when reasoning and verification are not in play; Cost-correct is the unit when they are.

Second, all four terms are in principle measurable. $\text{CPM}_{1:1}$ is a public price. R is measurable per-task-class via ablation runs against the same prompts on a non-reasoning baseline. $\bar{\rho}$ is observable through API usage logs. α requires a verifier that one defines. The binding constraint is verifier construction, not measurement.

Third, the engineering surface for cost reduction shifts. The four levers in the previous note act on CPM in the numerator. The new lever, verification, acts on α in the denominator. CPM compresses through stack-level engineering (quantization, kernels, runtime). α compresses through training-side and inference-side verifier engineering. They are different disciplines.

4.1. Reference computation

A reference implementation for computing $\text{Cost}_{\text{correct}}$ on a workload sample:

Listing 1. Reference computation of Cost-correct.

```
def cost_correct(model, workload, verifier, bench_set):
    """Cost per verified-correct answer at one workload point."""
    # 1. Blended cost per million tokens at the API
    cpm = (model.price_input + model.price_output) / 2.0
    # 2. Reasoning multiplier on this task class
    R = avg_reasoning_tokens(model, bench_set) / avg_answer_tokens(
        model, bench_set)
    # 3. Average rollout/rejection ratio for verifier-guided decoding
    rho_bar = avg_rollouts(model, workload.decoding_policy) - 1.0
    # 4. Verification accept rate at the quality threshold
```

```
    alpha = accept_rate(model, verifier, bench_set, threshold=workload
        .quality_theta)
    # Guard against alpha == 0
    if alpha <= 0:
        return float("inf")
    return (cpm * R * (1.0 + rho_bar)) / alpha
```

The implementation makes four things explicit. First, the reasoning multiplier is measured against a non-reasoning baseline, not estimated. Second, the rollout ratio includes the cost of verifier-rejected candidates, not just accepted ones. Third, the accept rate is taken at the production quality threshold θ, not at a generic benchmark threshold. Fourth, the verifier V is a load-bearing input, not an implicit constant. Production teams routinely omit one of these four; the pseudocode names them.

5. What verifiers actually look like in production

Three production patterns, each with a published reference.

Tree-search with process verifiers. *rStar-Math* ([14]) runs Monte Carlo Tree Search at decode time, with each candidate continuation scored by a process preference model trained alongside the policy. The system improves Phi3-mini-3.8B's MATH accuracy from 41.4% to 86.4%, surpassing o1-preview by 0.9 percentage points at small scale. The economic claim is that a small generator plus a small verifier, well-coupled, beats a large monolithic reasoning model on a per-task-cost basis on math.

Search-as-language. *Stream of Search* ([13]) takes a different position. Rather than coupling generator and verifier as separate systems, train a single language model to represent search itself as a flattened token sequence. SoS pretraining "increases search accuracy by 25% over models trained to predict only the optimal search trajectory." The verifier becomes implicit in the model's distribution over reasoning trajectories.

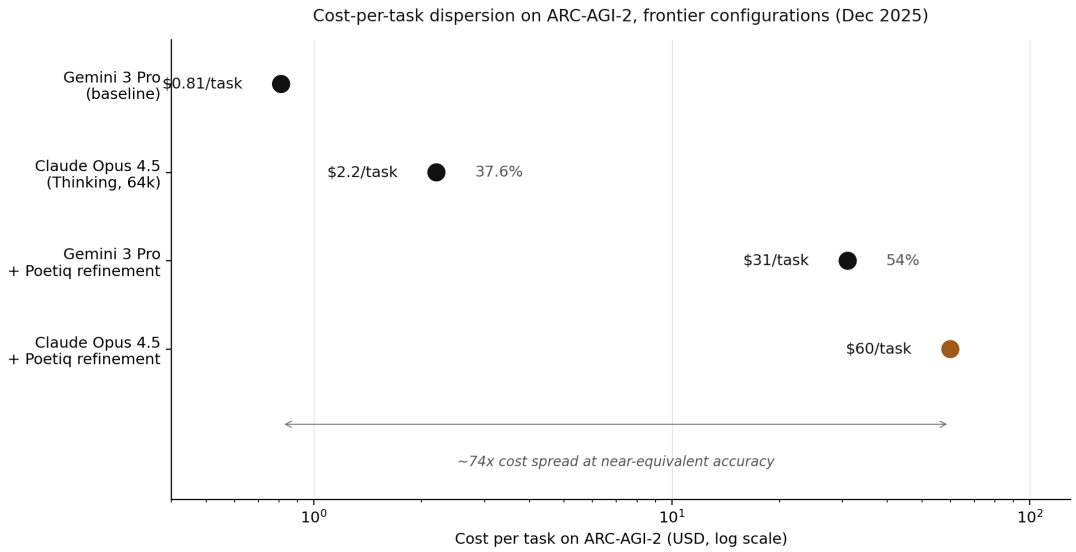
Test-time deliberation. *Tree of Thoughts* ([15], NeurIPS 2023) generalizes chain-of-thought to a search tree and reports the canonical result: GPT-4 with chain-of-thought solves 4% of Game of 24 problems; the same model with ToT solves 74%. This is a no-training-time-change result. Pure inference-time deliberation, with self-evaluation acting as the implicit verifier.

These three patterns are not interchangeable. Tree-search-with-process-verifier suits hard-verifiable tasks such as math, formal proof, and code with strict tests. Search-as-language is attractive for tasks where the trajectory itself is part of the output (planning, agentic). Test-time deliberation works when the model is strong enough to evaluate its own steps reliably and the task admits clean intermediate evaluation. Each has a different Cost-correct profile. The engineering choice is which verifier shape best inverts the binding constraint for a given workload.

6. ARC-AGI-2 and SWE-Bench Pro: the visible price-quality dispersion

The most legible empirical evidence that the unit of account has shifted is the ARC-AGI-2 leaderboard. The Prize team publishes cost-per-task as a primary axis, not a footnote. As of the December 2025 results analysis ([19]), published cost-per-task figures across frontier configurations include those in Table 1.

The cheap-to-expensive spread on the same benchmark across frontier configurations



Source: ARC Prize 2025 Results Analysis (December 5, 2025). Each marker is one published frontier configuration; cost figures quoted from the source.

Figure 1. Cost-per-task dispersion on ARC-AGI-2 across published frontier configurations. The horizontal axis is cost per task in US dollars on a logarithmic scale. Each marker is one published configuration. The cheap-to-expensive spread is approximately 74x at near-equivalent accuracy. Source: ARC Prize 2025 results analysis (December 5, 2025).

Table 1. Selected ARC-AGI-2 leaderboard entries from the December 5, 2025 results analysis. Cost-per-task figures quoted from the source.

Configuration	Score	Cost per task
Gemini 3 Pro (baseline)	–	$0.81
Claude Opus 4.5 (Thinking, 64k)	37.6%	$2.20
Gemini 3 Pro + Poetiq refinement	54%	$31
Claude Opus 4.5 + Poetiq refinement	comparable	∼$60

exceeds 70x at near-equivalent accuracy. This dispersion is not because some configurations are worse models. It is because verification-conditional rollouts cost more per task and buy more correctness. The leaderboard is, in effect, a published Pareto frontier in cost-per-correct-answer space.

The same pattern is starting to appear in agentic benchmarks. *SWE-Bench Pro* ([8]), the long-horizon successor to SWE-Bench, contains "1,865 problems sourced from a diverse set of 41 actively maintained repositories." The benchmark features "long-horizon tasks that may require hours to days for a professional software engineer to complete, often involving patches across multiple files and substantial code modifications." The trajectory length per task makes per-task-cost the natural reporting metric. Single-figure benchmark percentages without cost numbers are losing decision-relevance for agentic workloads.

7. The May 2026 pricing landscape

A reading of Cost-correct requires current public pricing for context. Table 2 summarizes the public API pricing schedule across major reasoning-capable model families as of May 6, 2026. All values per million tokens. Reasoning tokens, where supported, are billed as output tokens at the rates shown.

Table 2. Selected public API pricing as of May 6, 2026. All values in dollars per million tokens.

Provider	Model	Reasoning policy	Input	Output
OpenAI	GPT-5.5 (Apr 23 2026)	billed as output	5.00	30.00
OpenAI	GPT-5.4	billed as output	2.50	15.00
Anthropic	Claude Opus 4.7	billed as output	5.00	25.00
Anthropic	Claude Sonnet 4.6	billed as output	3.00	15.00
Anthropic	Claude Haiku 4.5	billed as output	1.00	5.00
DeepSeek	V4-flash	billed as output	0.14	0.28
DeepSeek	V4-pro (75% promo)	billed as output	0.435	0.87

Two structural observations.

The flagship-to-economy spread within a single provider remains roughly two orders of magnitude. Anthropic's Opus-to-Haiku output spread is 5x. DeepSeek's V4-flash undercuts Anthropic's Haiku by 18x on output. The cross-provider spread between an OpenAI flagship and a DeepSeek economy model is more than 100x on output. CPM is no longer a single number. It is a regime selection.

The DeepSeek `deepseek-reasoner` and `deepseek-chat` endpoints are deprecated as of late April 2026 in favor of the V4 series ([18]). The V4-pro 75% discount is "extended until 2026/05/31 15:59 UTC" per the docs. Pricing in this regime moves on calendar boundaries, not architecture boundaries.

8. The GPT-5.5 reprice as a market signal

The GPT-5.5 price hike on April 23, 2026, with input from $2.50 to $5.00 per million tokens and output from $15.00 to $30.00 per million ([16]), is the first time in roughly three years that an OpenAI flagship has raised sticker prices versus its predecessor. The headline reaction frames it as a reversal of the inference cost decline. This note's framework suggests a different reading.

If the operational unit of inference economics has shifted from cost-per-token to cost-per-correct-answer, then a per-token price hike that is more than offset by improved per-task accept rate represents disinflation in the new unit, not inflation. The Cost-correct denominator α grows. If α growth dominates the doubling of CPM, Cost-correct falls.

The hypothesis is therefore that OpenAI is implicitly pricing on a verification-corrected basis: the per-token price reflects the rate-limiting cost of producing answers that pass a stricter internal verification bar. This is a price action consistent with a producer who has interior knowledge of α improvements that the public benchmarks have not yet legibly priced.

The hypothesis is falsifiable. If reproducible third-party measurement shows that GPT-5.5's α improvement on standardized verifier-bound benchmarks (RLVR-style math, pro-

grammatic code verification, factuality with retrieval grounding) does not offset the doubled CPM, the price action is not justified by verification economics and is a different signal entirely.

9. The August 2026 forcing function

A non-economic constraint enters the picture in late summer 2026. The European Union AI Act implementation timeline ([20]) specifies that "the remainder of the AI Act starts to apply, except Article 6(1)" on August 2, 2026, bringing high-risk AI system obligations into force. General-purpose AI model obligations under Chapter V have applied since August 2, 2025.

Verification economics is regulatory infrastructure for these obligations. The Act requires high-risk system deployers to maintain demonstrable accuracy, transparency, and human-oversight measures, all of which translate, in implementation, to verifier-and-evaluator construction. The Cost-correct unit becomes a compliance unit, not just an engineering one. The α term acquires regulatory weight. Any high-risk deployment must justify accept rates, error analysis, and corrective procedures against a defined verifier specification.

The August 2026 deadline therefore concentrates demand for verification-economics tooling at exactly the moment the producer side, signaled by the GPT-5.5 reprice, is shifting toward the same unit. The two pressures compose. By late 2026, the operational unit of inference economics across both deployment and procurement sides is unlikely to remain cost-per-token.

10. Engineering implications

1. **Report Cost-correct, not CPM, when communicating production economics.** CPM is now a denominator term in a larger formula. Reporting it in isolation hides the binding constraint.
2. **Specify the verifier alongside the model.** Any production claim of "X% accuracy at $Y per task" is incomplete without naming the verifier under which X is measured. A verifier specification is a load-bearing artifact, comparable to a benchmark eval suite.
3. **Profile R per task class.** The reasoning multiplier is task-conditional. Production traffic distributions should be characterized by their (task-class, R) histogram, not a single average.
4. **Treat the verifier as a deployable artifact.** Verifier models deserve the same engineering rigor as generator models. Versioned. Evaluated against held-out sets. Monitored for distributional drift. Often smaller, faster, quantized, deployable on-device. A 7B verifier serving a 70B generator is an architecture, not a workaround.
5. **Consider RLVR-style training for verifiable workloads.** If a workload admits programmatic verification, the Cost-correct equation is structurally cheaper to optimize than for open-ended verification. Training-time verifier construction versus inference-time verification is a real engineering decision in 2026.
6. **Track α as a first-class production metric.** Cache hit rate, latency P99, and tokens-per-second-per-watt belong on the same dashboard as the verifier accept rate at the production quality threshold. A regression in α is a more expensive failure than a CPM spike.

11. Conclusion

The previous note in this series argued that the inference cost story between 2022 and 2024 was a compound curve: four levers, each amplifying the others, against a hardware market that competed on delivered tokens per dollar. The next eighteen months will be defined by a different compound. Reasoning multiplies the work done per task. Verification multiplies the value extracted per token. The two arithmetic operations sit on different sides of the same fraction.

The lever that worked in 2022 to 2024 was CPM. The lever that works in 2026 is α. A producer that improves α can defend higher CPM (GPT-5.5). A deployer that improves α can serve more correctness at the same dollar (rStar-Math at the small-model end of the curve). A regulator that requires α to be measurable can shift the entire market onto the new unit (EU AI Act in August 2026).

The systems that win the second half of the decade will not produce cheaper tokens. They will produce cheaper correct tokens. The same goal as the previous note, with one new variable made explicit.

References

[1] Snell, C., Lee, J., Xu, K., and Kumar, A. "Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters." arXiv:2408.03314, 2024. https://arxiv.org/abs/2408.03314

[2] DeepSeek-AI. "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning." arXiv:2501.12948, 2025. Published in *Nature* 645:633–638. https://arxiv.org/abs/2501.12948

[3] Shao, Z., Wang, P., Zhu, Q., et al. "DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models." arXiv:2402.03300, 2024. Introduces Group Relative Policy Optimization (GRPO). https://arxiv.org/abs/2402.03300

[4] OpenAI. "OpenAI o1 System Card." arXiv:2412.16720, 2024 (last revised April 30, 2026). https://arxiv.org/abs/2412.16720

[5] Du, Z., Kang, H., Han, S., Krishna, T., and Zhu, L. "OckBench: Measuring the Efficiency of LLM Reasoning." arXiv:2511.05722, 2025 (revised February 23, 2026). https://arxiv.org/abs/2511.05722

[6] Gundlach, H., Lynch, J., Mertens, M., and Thompson, N. "The Price of Progress: Price Performance and the Future of AI." arXiv:2511.23455, 2025 (revised March 23, 2026). https://arxiv.org/abs/2511.23455

[7] Erdil, E. "Inference economics of language models." arXiv:2506.04645, 2025. https://arxiv.org/abs/2506.04645

[8] Deng, X., Da, J., Pan, E., et al. "SWE-Bench Pro: Can AI Agents Solve Long-Horizon Software Engineering Tasks?" arXiv:2509.16941, 2025 (revised November 14, 2025). https://arxiv.org/abs/2509.16941

[9] Lightman, H., Kosaraju, V., Burda, Y., et al. "Let's Verify Step by Step." arXiv:2305.20050, 2023. Releases PRM800K. https://arxiv.org/abs/2305.20050

[10] Cobbe, K., Kosaraju, V., Bavarian, M., et al. "Training Verifiers to Solve Math Word Problems." arXiv:2110.14168, 2021. Introduces GSM8K. https://arxiv.org/abs/2110.14168

[11] Wang, X., Wei, J., Schuurmans, D., et al. "Self-Consistency Improves Chain of Thought Reasoning in Language Models." arXiv:2203.11171, 2022. ICLR 2023. https://arxiv.org/abs/2203.11171

[12] Lambert, N., Morrison, J., Pyatkin, V., et al. "Tulu 3: Pushing Frontiers in Open Language Model Post-Training." arXiv:2411.15124, 2024. Introduces Reinforcement Learning with Verifiable Rewards (RLVR). https://arxiv.org/abs/2411.15124

[13] Gandhi, K., Lee, D., Grand, G., et al. "Stream of Search (SoS): Learning to Search in Language." arXiv:2404.03683, 2024. https://arxiv.org/abs/2404.03683

[14] Guan, X., Zhang, L. L., Liu, Y., et al. "rStar-Math: Small LLMs Can Master Math Reasoning with Self-Evolved Deep Thinking." arXiv:2501.04519, 2025. https://arxiv.org/abs/2501.04519

[15] Yao, S., Yu, D., Zhao, J., et al. "Tree of Thoughts: Deliberate Problem Solving with Large Language Models." arXiv:2305.10601, 2023. NeurIPS 2023. https://arxiv.org/abs/2305.10601

[16] apidog. "GPT-5.5 Pricing." April 2026. https://apidog.com/blog/gpt-5-5-pricing/

[17] Anthropic. "Pricing." Accessed May 6, 2026. https://platform.claude.com/docs/en/about-claude/pricing

[18] DeepSeek. "Pricing." Accessed May 6, 2026. https://api-docs.deepseek.com/quick_start/pricing

[19] ARC Prize. "ARC Prize 2025 Results Analysis." December 5, 2025. https://arcprize.org/blog/arc-prize-2025-results-analysis

[20] Future of Life Institute. "EU AI Act Implementation Timeline." artificialintelligenceact.eu, 2024. https://artificialintelligenceact.eu/implementation-timeline/

[21] Bhardwaj, M. "The Inference Stack in 2026: A Field Note on Token Economics, Runtime Systems, and Model Architecture." ifitsmanu.com, May 2026. https://ifitsmanu.com/research/inference-stack-2026