LLM Pricing Analysis (2025)

Commercial API LLM Models

These are proprietary large language models offered via cloud APIs by major providers. Pricing is usage-based (per token), and context lengths vary. **All costs below are per 1,000 tokens** (prompt or completion), with separate input vs output token rates. "Total" combines input+output for 1,000 tokens of each.

OpenAI Models

- **GPT-4 (8k context)** – Input: **$0.03**/1K, Output: **$0.06**/1K (total $0.09). Context: 8K tokens. No free tier; pay-as-you-go. (Pricing from OpenAI)

- **GPT-4 (32k context)** – Input: **$0.06**/1K, Output: **$0.12**/1K (total $0.18). Context: 32K tokens . Used for longer prompts; same pay-as-you-go terms.

- **GPT-4 Turbo (128k)** – Input: **$0.01**/1K, Output: **$0.03**/1K (total $0.04). Context: 128K tokens. This newer "GPT-4 turbo" model is significantly cheaper (about **60–70%** lower cost than original GPT-4) .

- **GPT-4o** – Input: **$0.0025**/1K, Output: **$0.0100**/1K (total $0.0125). Context: 128K. A high-intelligence model introduced in 2024; far cheaper per-token than earlier GPT-4 . Supports vision and tools, suited for complex tasks . No special volume discounts (standard pay-go; 50% off input via batch requests) .

- **GPT-4o mini** – Input: **$0.00015**/1K, Output: **$0.00060**/1K (total $0.00075). Context: 128K. A small, cost-efficient model (~82% the quality of GPT-4) . *Extremely* low cost (15¢ per million input tokens) – about 60% cheaper than even GPT-3.5 Turbo . Optimized for parallel calls and high volumes. *Pricing tiers:* 50% off with batch mode; fine-tuning costs extra (see below).

- **GPT-3.5 Turbo (4k)** – Input: **$0.0005**/1K, Output: **$0.0015**/1K (total $0.0020). Context: 4K. The widely-used ChatGPT model; very low cost after Jan 2024 price cuts (input halved, output –25%) . No monthly fee; included in OpenAI's pay-as-you-go.

- **GPT-3.5 Turbo (16k)** – Input: **$0.0030**/1K, Output: **$0.0040**/1K (total $0.0070). Context: 16K tokens. Higher context version of GPT-3.5; slightly higher price (roughly double the 4k model) . Pay-as-you-go pricing.

OpenAI Pricing Notes: Fine-tuning GPT-4o costs $25 per 1M tokens trained plus usage (e.g. GPT-4o fine-tuned: ~$0.00375/1K in, $0.015/1K out) . No automatic volume discounts, but **Batch API** calls can save **50%** on input/output costs by running tasks asynchronously in bulk . OpenAI documentation provides full pricing details .

Anthropic Claude Models

- **Claude 3.7 "Sonnet"** – Input: **$0.0030**/1K, Output: **$0.0150**/1K (total $0.0180). Context: 200K tokens. Anthropic's flagship **hybrid reasoning** model (2025) with state-of-the-art coding, planning, and vision capabilities . Supports extremely long context (200K) and can produce up to 128K-token outputs. *Pricing:* Standard $3/M in, $15/M out . **Prompt caching** can cut costs by up to 90% (re-used context is nearly free) and **batch processing** yields ~50% savings . Offered via API, AWS Bedrock, and Google Vertex AI .

- **Claude 3.5 "Haiku"** – Input: **$0.0008**/1K, Output: **$0.0040**/1K (total $0.0048). Context: 200K. A faster, lightweight Claude model (3.5 "Haiku") updated Nov 2024 . Optimized for speed; cheaper but less adept at very complex prompts. Lacks image analysis. Same API terms; caching/batch applicable.

- **Claude 3 "Opus"** – Input: **$0.0150**/1K, Output: **$0.0750**/1K (total $0.0900). Context: 200K. Earlier large model (Claude v3 Opus) – most expensive Anthropic model, yet currently **less capable** than newer Sonnet . Handles images. Primarily for legacy use; priced $15/M and $75/M tokens . (Enterprises likely negotiate custom pricing for high volumes.)

- **Claude Instant v1** – Input: **$0.00163**/1K, Output: **$0.00551**/1K (total ~$0.00714). Context: 100K. (Superseded by Claude 3.5, but was Anthropic's earlier "instant" model) . Provided very fast responses at low cost. *For reference:* Claude Instant 1.1 was $1.63/M input, $5.51/M output .

Anthropic Pricing Notes: The **Claude.ai** platform offers subscription plans (Claude Pro $20/month for individuals) but API access is usage-billed. The **Claude API** supports **prompt caching** – repeated prompts can cost 90% less – and **batch calls** (combine multiple requests) for 50% off . Fine-tuning is not yet widely available for Claude models. Official pricing info is on Anthropic's site .

Google PaLM/Gemini Models

- **PaLM 2 / Gemini 1.5 (general)** – **$0.000075**/1K input, **$0.000300**/1K output (total $0.000375). These ultra-low rates (7.5¢ per M input tokens) apply to Google's standard models after an August 2024 price reduction . For prompts ≤128K tokens, Google's pricing is **significantly lower** than OpenAI's . (For very long contexts beyond 128K, input cost doubles to $0.00015/1K) . *Example:* the "Gemini 1.5 Flash" model was dropped to $0.075/M and $0.30/M output .

- **Gemini 1.5 Pro** – (128K–2M context) – Exact token pricing not publicly disclosed; expected higher than Flash tier. Google's **Gemini Pro** models offer up to **2M** token context windows and stronger reasoning, likely at higher per-token cost or per-request fees. *(In one preview, Google listed ~$7 per 1M input and $21 per 1M output for a large-context model)* . Official documentation suggests "grounding" or retrieval usage might be priced separately for these advanced models .

- **Vertex AI PaLM APIs** – Google's models are also accessible via Vertex AI with slightly different pricing. For instance, Vertex's PaLM 2 text model was $0.15/M input and $0.60/M output (higher than the direct Gemini API) . Volume discounts are available for enterprise commitments .

Google Pricing Notes: Google's **Generative AI** is generally **cheaper per token than OpenAI** . They offer a **free trial** ($300 credit on GCP, plus $20 on Vertex AI) and usage-based billing after that . High-context models (like Gemini Pro/Max) might incur **per-request charges** (e.g. per "grounding" call) in addition to token fees . Official pricing is on Google's cloud and AI developer sites.

Cohere Command Models

- **Cohere Command A** – Input: **$0.00250**/1K, Output: **$0.01000**/1K (total $0.0125). Context: 256K tokens. "Command A" is Cohere's latest 111B-param model for agentic AI and multilingual tasks . Pricing is $2.50 per million input and $10 per million output , comparable to GPT-4o. No free tier; pay-per-use via API.

- **Cohere Command R+** – Input: **$0.00250**/1K, Output: **$0.01000**/1K (total $0.0125). Context: 100K+ (long). A predecessor large model oriented to enterprise tasks . Same pricing as Command A (and similar capabilities).

- **Cohere Command R** – Input: **$0.00015**/1K, Output: **$0.00060**/1K (total $0.00075). Context: 100K. A cost-optimized generative model for retrieval-augmented generation (long documents, tools) . Extremely affordable (15¢ per M input). Suited for high-volume apps; fine-tunable.

- **Cohere Command Lite (7B)** – *Approx.* **$0.0001**/1K (or less). Cohere offers smaller models (e.g. 7B) for speed; these cost on the order of $0.10–$0.20 per million tokens (very inexpensive, often bundled in usage). *(Cohere's pricing page focuses on larger models; the 7B "Command R7B" model likely falls in this range.)*

Cohere Pricing Notes: All usage is pay-as-you-go via API . They provide an **initial free credit** for new accounts (e.g. $5). Fine-tuning is available: e.g. a fine-tuned Command model costs double the base rate (input $0.00030, output $0.00120/1K) plus training at $3.00 per million tokens . Context windows are large (Command models support up to 256K tokens). Official pricing info is published on Cohere's site .

AI21 Labs (Jurassic/Jamba Models)

- **Jurassic-2 Ultra** – Input: **$0.0188**/1K, Output: **$0.0188**/1K (total $0.0376). A 2023 flagship model (large), via AI21 Studio and AWS Bedrock. Cost is $18.8 per M tokens (in/out) . Supports ~100K context. Suitable for high-accuracy needs but pricier than peers.

- **Jurassic-2 Mid** – Input: **$0.0125**/1K, Output: **$0.0125**/1K. Medium-sized J2 model (cheaper at $12.5/M tokens) . Often used for summarization, etc. AI21 provided $10 free credits for new users .

- **"Jamba" Large** – Input: **$0.0020**/1K, Output: **$0.0080**/1K (total $0.0100). AI21's *newest* large model (2024) with long context. Priced at $2/M input and $8/M output , dramatically cheaper than J2 Ultra (AI21 claims 30% token efficiency advantage as well) .

- **"Jamba" Mini** – Input: **$0.0002**/1K, Output: **$0.0004**/1K (total $0.0006). A lightweight model for basic tasks . Only $0.2/M and $0.4/M tokens – one of the lowest-cost options. Great for simple completions where high accuracy is not critical.

AI21 Pricing Notes: AI21 has **transparent usage pricing** with pay-as-you-go and an optional custom enterprise plan . New accounts get **$10 free credit** (3 months) . Volume discounts and private hosting are available for enterprise clients . AI21's shift from Jurassic-2 to **Jamba** models in late 2024 substantially lowered per-token costs (e.g. from ~$0.018 to $0.002 per 1K for large model) . Official pricing is documented on AI21's site .

Other Notable API Models

- **Amazon Titan** – Amazon's proprietary LLMs (offered through Bedrock) have competitive pricing. *E.g.*, **Titan Text Lite** is about **$0.0003**/1K tokens (input or output) based on AWS examples (roughly $0.30 per million). A larger "Titan Text Express" offers higher throughput at higher cost. Amazon's Bedrock uses a unique unit-based pricing for provisioned throughput, but on-demand per-token costs are in line with open-model pricing. (Exact rates are not fully public; Titan 13B is roughly on par with Llama-2 13B cost, see below.)

- **Meta LLaMA-2** – While open-source, Llama-2 is served via Azure and AWS. On **AWS Bedrock**, **Llama-2 13B Chat** is **$0.00075**/1K in, **$0.00100**/1K out . The 70B version costs **$0.00195**/1K in, **$0.00256**/1K out . These low rates (~$0.002–$0.004 total per 1K) reflect only infrastructure cost since the model is open. Azure's pricing for Llama-2 is similar ("Starting at Free" with pay-go VM hours) .

- **Aleph Alpha Luminous** – A European LLM (Meta's Llama-derived). Pricing (as of 2024) was roughly €2.70 per million tokens for the 30B model (about $0.003/1K). It's offered via API with custom pricing tiers for enterprise. (No official USD pricing published; included for completeness.)

- **IBM WatsonX Granite** – IBM's Granite family models (e.g. 13B "Granite") are available on IBM Cloud. Pricing is not per token but per VM core-hour . IBM's models are generally aimed at enterprise subscriptions rather than pay-per-token; for example, Granite-13B was roughly $5 per hour on IBM's infrastructure (cost translating to ~$0.005/1K at moderate load). IBM does not publish simple per-token prices.

Note: Many other startups offer proprietary models (e.g. **Writer's Palmyra**, **Anthropic Claude Instant**). These typically follow similar pricing patterns (per-token fees in the low dollars per million). For instance, **Writer.com's Palmyra** models range from ~$0.001–$0.005 per 1K tokens (estimated). When evaluating a commercial model, always refer to the provider's documentation for exact pricing.

Documentation Links: Official pricing pages for reference – OpenAI , Anthropic , Google Cloud , Cohere , AI21 , AWS Bedrock .

Hosted Open-Source Models

These are open-source LLMs provided as a service (typically by cloud platforms or startups). The models themselves are free to use if self-hosted, but providers charge for compute. Hosted open-source model pricing is generally **lower** than proprietary API models, often by an order of magnitude, thanks to lower licensing costs.

- **Llama 2 (70B)** via AWS – Input: **$0.00195**/1K, Output: **$0.00256**/1K (total $0.0045). Context: 4K (chat tuned) . Meta's Llama 2 is open-source; AWS Bedrock charges only for inference compute. The 13B version is even cheaper ($0.00175 total per 1K) . Azure offers Llama-2 hosting with similar usage-based fees (and free tier options) .

- **Mistral 7B** via AWS – Input: **$0.00015**/1K, Output: **$0.00015**/1K (est.) (total $0.00030). Context: 8K. A very small model (7B parameters) optimized for speed. AWS example pricing implies ~15¢ per million tokens . This is extremely low cost – suitable for lightweight tasks or high-volume

throughput where moderate accuracy suffices.

- Falcon 40B via TII* – Input: ~$0.0005–$0.0010/1K (approx). Falcon, an open 40B model, is hosted by TII and on Hugging Face. While exact hosted rates vary, they are on the order of $0.50 to $1.00 per million tokens (if using cloud GPU instances). Some platforms (like HuggingFace Inference API) include Falcon in their subscription, effectively making per-token cost negligible beyond the subscription fee.

- StableLM & Others – Many smaller open models (e.g. StableLM, GPT-J, GPT-NeoX, etc.) can be invoked via APIs on platforms like HuggingFace or Replicate. Pricing is typically based on compute time rather than tokens. For example, running a 6B-parameter model on Replicate might cost ~$0.0003 per second – which at ~20 tokens/sec equals ~$0.015 per 1K tokens. In practice, these smaller models often cost well under $0.001 per 1K tokens on efficient infrastructure.

- Hugging Face Inference API – HuggingFace hosts thousands of community models. Their free tier allows limited calls to open-source models. Paid tiers (~$ inference credit packs or enterprise plans) effectively translate to usage costs (e.g. a ~$0.0004/1K token for 7B models on dedicated GPUs, scaling with model size). HuggingFace's emphasis is on *throughput-hour pricing*, but for comparison, running a 40B open model might cost ~$2/hour, which at ~120 tokens/sec yields ~$0.004 per 1K tokens – still very cheap. (HuggingFace pricing: free for basic use; enterprise for heavy use .)

- Azure Model Catalog – Microsoft's Azure offers open models (like Llama 2, Falcon) under its Model Catalog or via Azure ML. Pricing here is often VM-based: e.g., deploying Llama-2 70B on an Azure GPU VM (~A100) at $6.50/hour . If that instance generates ~50 tokens/sec, the cost is about $0.036 per 1K tokens. If using Azure's serverless endpoint, you pay per request or token (similar to AWS numbers given above). Azure also sometimes provides free deployments for research or trial.

- Together.ai – Together provides API access to 200+ open models with per-token pricing. For example, Llama-3 70B (hypothetical next-gen model) is listed at $0.00090/1K in and $0.00120/1K out (total ~$0.0021) – ~11× cheaper than GPT-4 for similar length . Together's smaller models (e.g. Mistral or RedPajama variants) range around $0.0005–$0.001/1K. They also implement caching to reduce repeat costs. Pricing is purely pay-as-you-go, with $1 free credit for new users .

- Anyscale Endpoints – Anyscale (from the creators of Ray) allows serving open models with scaling. They offer usage-based pricing (not publicly listed in detail; likely comparable to AWS). Anyscale emphasizes enterprise plans, so per-token rates are negotiated or based on infrastructure consumption (estimated in the low $0.001s per 1K).

Hosted Open-Source Notes: Generally, open-source models hosted in the cloud cost $0.0001 to $0.005 per 1K tokens, depending on model size and provider. This is 10×–100× cheaper than most closed models. However, open models may underperform top proprietary models on complex tasks. Providers often give free quotas or credits to encourage trying open models (e.g. Together.ai $1 credit , HuggingFace free inference for limited use). For heavy use, provisioning a dedicated instance (reserved throughput) can further reduce costs – e.g. AWS offers *Provisioned Throughput* discounts for long-running jobs . The trade-off is that open models require careful prompt engineering and may lack fine-tuning out-of-the-box for some domains.

Documentation Links: AWS Bedrock pricing for open models , Together AI pricing examples , HuggingFace and others .

Pay-as-You-Go LLM Services

This category includes platforms that let you access various models on a purely usage-based basis, often aggregating multiple models behind one API. They typically do not require monthly commitments or enterprise contracts – you **"pay for what you use"** (often with some free tier to start).

- **OpenRouter** – An open-source router that forwards queries to different LLMs (OpenAI, Anthropic, etc.) through a unified API. OpenRouter itself is free; you pay the underlying model's token cost (and some models on OpenRouter are **free** or community-hosted). It lists pricing for each model it proxies – e.g., GPT-3.5, GPT-4, Claude, etc. – identical to those providers' official rates, and even offers some hosted open models at no cost . This is essentially a **pay-as-you-go hub**: no markup for many models and flexible payment (they even accept crypto) .

- **Replicate** – A platform to run community AI models via API. Replicate charges **per second of execution** rather than per token. This means cost depends on model speed and hardware. For example, running a large 70B model on an A100 GPU might cost ~$0.0004 per second; if that model generates ~10 tokens/sec, that's ~$0.04 per 1K tokens. Smaller models cost far less (e.g. ~$0.003 per image or a few cents for text outputs) . Replicate is purely pay-go: you get billed for the compute time your requests use (with granular metering). There's no minimum fee and many popular models can be run for mere pennies.

- **Hyperbolic** – A service focusing on *low-cost* LLM API access. They claim up to **80% less cost** than mainstream providers by optimizing GPU usage . Hyperbolic provides various base models for text, image, etc., at usage-based rates (they have a free base plan, then pay-go) . For instance, they guarantee very competitive GPU-hour prices and pass those savings on per token. (Precise token rates aren't public, but user reports indicate costs similar to open-source hosting – on the order of $0.001–$0.002/1K.)

- **Novita AI** – Another pay-go platform for LLMs. It offers both token-based billing and GPU execution time billing . For LLM API calls, you're billed per token; for custom GPU jobs, by runtime. They also have a *Dedicated Endpoint* option for enterprise (fixed monthly rate). The key benefit is flexibility: developers can prototype with pay-as-you-use and later scale with a reserved plan. (Token prices align with open models, since Novita primarily hosts open-source and proprietary models under partnership.)

- **Fireworks.ai** – A provider boasting one of the fastest inference engines for open models . Fireworks is pay-as-you-go with $1 free credit for new users . They haven't publicly listed per-token prices for each model, because they dynamically allocate optimized hardware. In practice, Fireworks' costs are similar to or lower than other hosts – their value prop is **4× lower latency** and high throughput . This means you might get more tokens per second for the same cost. Fireworks also supports multimodal models (e.g. text+image). Pricing is usage-based; fine-tuning deployments might incur additional fees or higher rates.

- **API Marketplaces** – Services like **RapidAPI**, **Azure Marketplace**, or **Clarifai** allow developers to publish LLM APIs with their own pricing. For example, AI21's Jurassic-2 was offered on AWS Marketplace at $30,000/month unlimited access or usage-based rates for smaller plans . These marketplaces often have **tiered plans** (e.g. free tier, then pay-per-call). Pay-as-you-go is usually

an option, but sometimes a high flat fee is quoted for enterprise usage (as in the Jurassic example). In general, for most developers, using the direct API or an aggregator like those above is more cost-effective than these third-party resale marketplaces.

Key Points: Pay-as-you-go services aim to make experimenting with and switching between models easy and cost-efficient. Most have **no minimum**, **no subscription** (you simply add a payment method and are charged per use). Many provide an initial free credit. The **downside** can be that some add a slight overhead or have variability in performance. Always check if the service is charging more than the model's native cost. For instance, OpenRouter keeps costs the same as official , whereas some others may embed a profit margin.

In summary, **pay-as-you-go platforms** enable access to 50+ models on demand – from GPT-4 to open alternatives – without long-term commitments. This flexibility is excellent for development and testing, and with competitive pricing (often only marginally above raw infrastructure cost), they can be scaled to production with minimal cost difference.

Sources: Helicone's overview of provider pricing and features , Replicate pricing info , OpenRouter details .

Advanced "Reasoning" Models (State-of-the-Art Thinking LLMs)

In this section, we examine the most advanced **reasoning-focused** LLMs – models designed for complex, multi-hop thinking, tool use, and chain-of-thought prompting. These models typically employ an internal "thinking" process (often generating hidden reasoning steps) to achieve higher accuracy on challenging tasks. We include OpenAI's *GPT-o1* and *GPT-o3*, Anthropic's *Claude 3.7 Sonnet Max*, Google's *Gemini 2.5 Pro Max*, and the open-source *DeepSeek R1*. For each, we list pricing and highlight their special capabilities:

OpenAI GPT-o1 (OpenAI "Reasoning" series)

- Pricing: **$0.015** per 1K input tokens, **$0.060** per 1K output tokens (i.e. $15 and $60 per million, respectively). This is OpenAI's *most expensive* model to date, reflecting the heavy compute used for its reasoning abilities. (By comparison, GPT-4's input is $0.0025/1K with GPT-4o, so o1 is **6× costlier** than GPT-4o for input and 6× for output.) There is an **o1-mini** variant that is 80% cheaper (about $0.003/1K in, $0.012/1K out) for lighter tasks .

- Context window: **200K tokens** . GPT-o1 can ingest enormous prompts (hundreds of pages of text). This helps it follow very long chains of reasoning or large documents.

- Reasoning latency: Higher than typical models. GPT-o1 **"thinks"** before answering – it spends extra compute time generating a hidden chain-of-thought . In practice, this can add **several seconds (or more)** to response latency for complex queries. OpenAI notes o1 requires much more computation per token than GPT-4 . Users have observed that o1 might take anywhere from a few seconds up to a minute for very complex problems, as it is effectively running an internal inference loop to reason.

- Chain-of-thought: Yes, by design. GPT-o1 was the first model to explicitly incorporate an extended hidden chain-of-thought in generating answers . It **breaks down problems into steps**

internally, which greatly improves math, logic, and multi-hop QA performance . However, OpenAI does **not** reveal the chain-of-thought to users (it's kept hidden for safety and proprietary reasons) . The model is trained to *not* directly output its reasoning steps (attempts to jailbreak this are disallowed) .

- **Parallel prompt support:** GPT-o1 is used through the same ChatCompletions API, so you can send multiple requests in parallel. There is no explicit feature to parallelize reasoning *within one query*, but its 200k context lets you pack multiple related questions or tool outputs in one prompt. Additionally, OpenAI's tooling (like function calling) is supported, enabling o1 to use tools (e.g. calculators) as part of its reasoning.

- **Use-case focus:** Difficult reasoning tasks where accuracy is paramount – **mathematics, science, multi-step logic puzzles, coding challenges**, etc. OpenAI reported GPT-o1 performs at roughly **PhD-expert level** on hard science exams . It's also good at scenarios requiring adherence to complex instructions or policies, thanks to the chain-of-thought guiding it to consider context deeply . GPT-o1 can serve as a "brains" of an AI agent, doing strategic planning or complex tool-using workflows that simpler models struggle with.

(Note: GPT-o1 was initially in limited beta – only available to select developers on high-tier plans . By 2025 it's more broadly accessible, but still with a high cost and usage limits may apply. OpenAI positions it as a complement to GPT-4 rather than a replacement – you'd use o1 when you need that extra "thinking power" on tough problems .)

OpenAI GPT-o3 (preview) and o3-mini

- **Pricing:** *Full GPT-o3:* not fully released as of early 2025 (expected to follow o1's pricing or higher). **GPT-o3-mini** is available, priced at **$0.00110** per 1K input and **$0.00440** per 1K output (i.e. $1.10/M and $4.40/M) – significantly cheaper than o1. In fact, o3-mini is about **4× cheaper** than GPT-4o on outputs. OpenAI has positioned o3-mini as a "fast, cost-efficient reasoning model" for coding and STEM use cases.

- **Context: 200K tokens** (same large window). The o3 series supports tools and structured output and retains the 200k context of o1 .

- **Latency:** o3-mini is optimized for speed relative to o1. It uses fewer parameters or a distilled approach, making it **faster** and more affordable (OpenAI specifically says it's geared to "fast" reasoning for code, etc.) Actual latency is lower – developers report o3-mini is responsive enough for real-time coding assistants (sub-5 second replies for moderate prompts), whereas o1 might take longer. We can infer that o3-mini trades some depth of reasoning for speed, but still "thinks" through steps albeit more quickly. (No official metrics yet, but Azure notes sub-100ms latency is possible for smaller open-source reasoning models – likely o3-mini aims for low latency on cached prompts, etc. .)

- **Chain-of-thought:** Yes. As part of OpenAI's reasoning series, GPT-o3 models generate chains-of-thought internally similar to o1. The *mini* variant likely has a shorter or more heuristic reasoning process due to its optimization, but it still will break down problems (especially for coding/math). Like o1, the chain-of-thought is hidden and only the final answer is returned.

- **Parallel prompts:** Same API behavior – you can run many o3-mini requests concurrently (OpenAI encourages its use for high-volume tasks given the lower cost) . No special parallelism beyond

normal API usage.

- Use-case focus: GPT-o3-mini is tailored for coding, math, and science tasks . OpenAI mentions it excels at those domains out of the box, making it ideal for code assistants, troubleshooting, or scientific Q&A. It also supports tool use and structured outputs, so it can function as an agent in workflows (just like o1, but at lower cost). Once full GPT-o3 is released, it is expected to become the new top-tier reasoner, likely surpassing o1 in complex reasoning (OpenAI skipped "o2" naming). Early benchmarks shared by OpenAI indicated an *o3 prototype* outperformed o1 on certain tasks by spending even more time "thinking" per answer. We anticipate GPT-o3 will target planning and "system 2" style reasoning even more heavily.

(In summary, GPT-o3-mini provides much of o1's reasoning capability at a fraction of the cost, making advanced reasoning more accessible. Developers can use o3-mini for day-to-day complex tasks, and reserve the heavyweight o1 for the absolutely hardest problems or when maximum accuracy is required.)

Anthropic Claude 3.7 Sonnet Max

- Pricing: $0.003 per 1K input, $0.015 per 1K output (total $0.018) – same as Claude 3.7 Sonnet standard . The "Max" refers to a special mode rather than a different price. In practice, using the full "Sonnet Max" capabilities (long thinking) will incur more tokens (because the model can produce extensive reasoning output if not hidden). But the token rates remain $3/M and $15/M. (Some partner applications, like Cursor IDE, charge this same rate for Gemini 2.5 and Claude 3.7 Max usage, indicating parity pricing) .

- Context window: 200K tokens for prompt, and up to 128K tokens output in "Max" mode . This enormous context (roughly 150k words) allows Claude Max to handle very large projects or multi-document analysis in one go.

- Reasoning latency: Adjustable – user-controlled. Claude 3.7 Sonnet is a *hybrid reasoning model*; it can operate in two modes: fast (immediate answer) or "extended thinking" mode . When reasoning mode is on, Claude may spend several seconds to a couple of minutes "thinking" before replying . The user can choose short or long reasoning. In "Max" (long) mode, you might see Claude pause and then produce a very well-thought-out answer after, say, 1–2 minutes for a complex task. In fast mode, it will answer nearly instantly (like a regular model) but with less deep reasoning. This flexibility is unique to Claude. In practice, for tough coding or planning tasks, users enable Max mode and accept a longer latency for a better answer.

- Chain-of-thought visibility: Yes – optional visible reasoning. Claude 3.7 Sonnet Max can display its chain-of-thought to the user. Anthropic made the model capable of showing step-by-step thinking (upon request or via the API) . For example, in the Cursor IDE integration, when Claude is in "Thinking" mode, you can actually watch it write out its thought process (like planning steps) before the final solution. This transparency is useful for debugging the model's reasoning. It's also configurable: developers can choose to retrieve the reasoning trace via the API (the Claude API provides a way to get the reasoning tokens if desired).

- Parallel prompt support: Claude's API allows batched prompts. Moreover, Claude 3.7's fast mode vs reasoning mode could be considered a form of parallel thought – e.g., one can first get a quick answer, then separately ask for a "reasoned answer." There isn't a special parallelism feature beyond that. However, because of the large context, one clever use is to supply multiple related queries in one prompt and let Claude reason about all of them together (which it can, given 200k

tokens). This can simulate parallel Q&A within one session.

- Use-case focus: Claude 3.7 Sonnet Max is geared towards complex, long-horizon tasks. For example, *"agentic coding"* – it can plan and execute a series of coding steps, debug, refactor large codebases (the entire code can fit in context) . It's adept at tool use: it can take actions like web browsing or API calls if integrated with tools, reasoning at each step. Anthropic specifically highlights uses in data analysis, intricate planning, and multi-step reasoning where the model might need to correct itself or explore alternatives . It's also one of the first models that can fluidly handle vision input in the same context as text (e.g., analyzing an image and related text) – useful for say, reviewing a diagram plus a document. In summary, Claude Sonnet Max shines in roles like AI assistants that need a "chain-of-thought" (e.g. an AI lawyer going through evidence, or a coding co-pilot managing a large project with planning).

Google Gemini 2.5 Pro Max

(Note: "Gemini 2.5 Pro Max" is a speculative name as Google's next-gen model; information is based on limited previews and third-party integration leaks.)

- Pricing: In a closed beta via tools like Cursor, it was priced similarly to Claude Sonnet Max (around $0.003/1K in, $0.015/1K out) . This suggests Google might align its top model's price with Anthropic's for competitive reasons. Official pricing isn't released; however, given Google's trend, it may offer lower per-token costs if used via its own API. It's possible that third-party platforms simply mirror Claude's pricing for simplicity. We should expect that when publicly available, Gemini 2.5 Max will have usage-based pricing and likely volume discounts via Google Cloud. (For reference, *Gemini 1.5 Pro* with 2M context had a unique pricing model: Google mentioned $35 per 1K "grounding" requests for certain operations , indicating some non-linear pricing for huge contexts.)

- Context window: 1,000,000+ tokens (1M). According to developer forums, only the "Pro Max" configuration of Gemini 2.5 enabled the full 1M token context . This is an unprecedented context size – about ~800k words, or an entire library of documents at once. It likely uses retrieval or segmentation under the hood, but from a user perspective, you could literally prompt it with an entire book series. Such a context is ideal for complex retrieval-augmented generation (the model can keep a vast knowledge base in memory).

- Reasoning latency: Likely high for full 1M token operations. If a prompt actually utilizes hundreds of thousands of tokens, the model must perform a lot of computation, which could mean responses taking minutes. However, for moderate lengths, Google's models are highly optimized (Gemini uses Google's TPUv5 and efficient architectures). Early testers of "Gemini Pro Max" found it was extremely powerful but sometimes timing out or failing on very long interactive sessions (possibly due to latency limits) . We can infer that reasoning mode on Gemini Max will be *adaptive*: it might quickly return an answer if it's confident, or spend more time if the query is complex. Google likely integrates their *Pathways* system to parallelize parts of the reasoning. In short, expect latency from a few seconds up to tens of seconds depending on prompt size and complexity.

- Chain-of-thought: Yes. Google's Gemini is known to incorporate advanced planning abilities (the original Gemini was said to combine strengths of AlphaGo-like planning with language). "Gemini 2.5" presumably can generate intermediate reasoning steps, though Google has not stated if they'll expose them. Given the focus on "thinking", it almost certainly uses an internal CoT for

multi-hop questions. It may also integrate **retrieval-augmented generation** deeply – possibly issuing its own queries to fetch information when needed (Google has alluded to Gemini's potential to use tools). In effect, its chain-of-thought could involve actual web searches or database lookups as part of its pipeline.

- **Parallel prompt support:** Google's infrastructure might allow sharding the prompt or parallel processing given Pathways. For the user, the Gemini API could allow *parallel subqueries*: e.g., the "grounding" requests mentioned might split a big task into parts. With a 1M context, one strategy is to feed multiple documents in parallel segments and have the model process them jointly – Gemini likely does this behind the scenes. While not an explicit user-controlled feature, Gemini Pro Max is built to handle very *wide* contexts (parallel information) and possibly *multimodal* input simultaneously (image + text + tables all at once). This implies a form of parallelism in how it handles different data streams.

- **Use-case focus: Agents, tool use, complex retrieval, and massive context tasks.** Gemini 2.5 Pro Max is envisioned as the cornerstone of Google's most advanced AI products – think of AI that can ingest an entire corporate knowledge base and act as an expert consultant, or an AI that can plan multi-step tasks on the web (booking trips, managing schedules across apps). Its huge context and multi-hop skills target use-cases like: **Research assistants** (it could literally take all academic papers on a topic and synthesize findings), **large-scale data analysis** (reading millions of characters of logs or code), and **autonomous agents** that need to maintain long dialogues/goals. Given Google's integration, it will likely excel at tool use – calling Google search, Maps, etc. as needed within its reasoning. In the Cursor IDE, "Gemini 2.5 Pro Max" has been used for coding tasks that require understanding an entire codebase – something only a model with this context could do. It's also multimodal: expect it to handle vision (describe or analyze images) and possibly other modalities (maybe audio) within the same model. In summary, Gemini 2.5 Max is geared to be an AI "super-assistant" that can think deeply and leverage vast information in one go.

(Because Gemini 2.5 Pro Max details are not officially published, the above draws on known Google AI trends and anecdotal reports. Once released, documentation from Google will clarify its pricing and capabilities.)

DeepSeek R1 (Open-Source Reasoning Model)

- **Pricing: $0.00055** per 1K input tokens, **$0.00219** per 1K output tokens (roughly $0.00274 per 1K combined). This is **extremely low** cost – only ~$2.19 per million output tokens – thanks to DeepSeek's efficiency and open-source nature. In fact, DeepSeek R1's output tokens are ~**27× cheaper** than GPT-o1's (which are $0.06/1K) . Additionally, DeepSeek offers *context caching*: repeated inputs cost only $0.00014/M (75% off) . They also have **off-peak discounts** – during certain hours, all usage is 50–75% off . This can bring effective costs under $0.0007 per 1K (!).

- **Context window: 64K tokens** prompt, plus it can generate a chain-of-thought up to **32K tokens** long before the final answer . So effectively ~96K total if you count reasoning + answer. This is a generous context (though smaller than GPT-o1's 200k). It is sufficient for most multi-step problems and lengthy prompts.

- **Reasoning latency: Relatively high** (slower than closed-source models). DeepSeek R1 prioritizes reasoning quality over speed. Benchmarks show an **output generation speed ~25 tokens/sec** and first-token latency ~**70 seconds** in one test . This is significantly slower than GPT-4 or Claude. In practical terms, DeepSeek might take 2–3+ times longer to respond on complex queries. The trade-off is cost – you pay very little, but you wait longer. The latency can be partially mitigated by

running on powerful hardware or using the chain-of-thought tokens as they stream (the model effectively "thinks out loud," which you could start parsing before it's done).

- Chain-of-thought: Yes, explicitly output. Unlike OpenAI's hidden CoT, DeepSeek R1 outputs its entire chain-of-thought (CoT) followed by the final answer . In other words, when you use R1's "reasoner" mode, you will see something like: "Thought 1: I need to do X. Thought 2: Now Y… (etc.) Final Answer: Z." Those CoT tokens are billed at the same output rate . This transparency is valuable for understanding and verifying the reasoning. It essentially lets you follow the model's multi-hop logic step by step.

- Parallel prompt support: DeepSeek's API doesn't have a special parallelism feature, but it does support asynchronous batched calls (and even encourages batch processing for cost savings). The pricing table shows separate rates for "cache hit" vs "miss" which implies you can reuse context across calls cheaply . While not parallel in one query, you could split a task into multiple sub-tasks (especially leveraging the low cost) and run them concurrently. DeepSeek R1 itself is single-threaded per request (no evidence of internal parallel reasoning), but a user could run many instances in parallel thanks to the low cost.

- Use-case focus: DeepSeek R1 is tuned for logical reasoning, math, and coding in an open-source framework . It shines on tasks that need careful step-by-step deduction – users report it's great for complex math word problems, competitive programming puzzles, or logical proofs, often outperforming GPT-4 on those because it doesn't rush to an answer. It's also used in coding assist tools that benefit from its detailed reasoning (e.g., it will explain why code should be written a certain way as it writes it). Due to its slower speed, it's less suited for rapid interactive chat; instead it's used like a "deliberation engine" – e.g., a backend service that given a tough problem will churn out a well-reasoned solution. Its low cost also makes it feasible for running large agent loops (where the AI iteratively thinks and acts). For instance, one could let DeepSeek run through a 100-step tool-using cycle without worrying about exorbitant token costs – something that would be very expensive with GPT-4 or Claude. In summary, DeepSeek R1 is the "budget thinking machine" – not as fast or polished as the big names, but extremely cost-effective for heavy reasoning tasks, especially in math/logic domains.

Advanced Model Comparison: All these "thinking" models introduce the ability to handle more complex tasks by internally reasoning or planning. GPT-o1 and Claude 3.7 set the bar for top-tier reasoning with high reliability but at high cost (and in GPT-o1's case, hidden reasoning). Claude and DeepSeek allow *visible* reasoning, aiding transparency. Gemini 2.5 aims to combine reasoning with vast knowledge (massive context + tool use). DeepSeek shows that even open models can compete in this space, given proper optimization. One important consideration is memory limits – e.g., GPT-o1 and Claude can maintain long dialogues or multi-step solutions without forgetting earlier steps, thanks to large context windows and chain-of-thought management, whereas simpler models might lose track. These advanced models are ideal for building AI agents that require multi-hop planning – such as an AI that needs to read multiple documents, then synthesize an answer while citing sources, or an AI agent that plans a complex task (like coding an app by breaking it into sub-tasks, solving each, and integrating).

In terms of practical use, organizations often use a two-tier approach: use a cheaper, fast model for simple queries, and route the truly hard problems to a "reasoning" model like GPT-o1 or Claude 3.7 Sonnet Max. This optimizes cost and latency. As seen, GPT-o3-mini and DeepSeek R1 are helping bridge the gap by lowering the cost of advanced reasoning dramatically .

Finally, it's worth noting these reasoning-optimized models can still do general text generation well, but if a task doesn't require deep reasoning, one is better off using simpler (cheaper) models. Their strength is when prompted with "Let's think step by step" or given very challenging instructions – they will outperform normal LLMs by going the extra mile in computation.

Sources: OpenAI o1 info , OpenAI pricing page , Anthropic Claude announcement , Google AI blog , DeepSeek docs , TechCrunch and community reports on these models .