OpenAI Image API: The Lightweight Skill

GPT Image 2.5 rewrite: model selection, every parameter, real measured pricing

OpenAI Image API

Field	Value
Text-to-image	`POST https://api.openai.com/v1/images/generations`
Image edit	`POST https://api.openai.com/v1/images/edits`
In-conversation tool	`POST https://api.openai.com/v1/responses` with `tools: [{"type":"image_generation"}]`
Auth header	`Authorization: Bearer $OPENAI_API_KEY`
Docs	https://developers.openai.com/api/docs/guides/image-generation and `/image-prompting`

> **Verified live 2026-09-08** against the GPT Image 2.5 release, and re-measured on a second independent pass: 69 assertions, 0 failures. Every number in the Cost and Latency sections was measured against the live API, not copied from the pricing page. OpenAI changes parameter behavior without announcement; see Staleness at the end.

What changed in GPT Image 2.5 (read this first if you last used this skill before September 2026)

1. **`gpt-image-2.5-sunburst` and `gpt-image-2.5-flare` are the current models.** Both released 2026-09-08. `gpt-image-2` is now previous generation.
2. **Neither 2.5 model appears in `GET /v1/models`.** The list still returns only `gpt-image-1`, `gpt-image-1-mini`, `gpt-image-1.5`, `gpt-image-2`, `gpt-image-2-2026-04-21`, `chatgpt-image-latest`. The 2.5 models work anyway. Do not use the models list to decide availability.
3. **Two new quality tiers: `xhigh` and `max`.** The API's own validation error still lists only `low`, `medium`, `high`, `auto`; that message is stale. Both new tiers return 200.

4. **Transparent backgrounds now work on `gpt-image-2` and on `gpt-image-2.5-sunburst`.** Guidance written before September 2026 saying gpt-image-2 hard-errors on `background: "transparent"` is obsolete. Real alpha confirmed (RGBA, corner alpha 0).
5. **2.5 `high` is roughly 4x cheaper and 5x faster than gpt-image-2 `high`, at comparable text legibility.** The quality ladder was rescaled: 2.5 `max` costs about what gpt-image-2 `high` cost.
6. **`input_fidelity` is gone.** Hard error on both endpoints for 2.5. Image inputs are always processed at high fidelity.
7. **Flare is not cheaper than Sunburst.** Identical output-token counts at every quality and size tested. Flare buys latency only.

Model Selection

Scenario	Model	Reason
Default for new work	`gpt-image-2.5-sunburst`	Quality leader; supports every parameter below
Anything latency-sensitive (interactive UI, batch throughput)	`gpt-image-2.5-flare`	Same price, roughly half the wall time; quality comparable to gpt-image-2
Precise edits, identity or product preservation	`gpt-image-2.5-sunburst`	The release is explicitly tuned for edit precision
Text inside the image	either 2.5 model	Both render multi-line sign copy correctly; Sunburst is the safer default
Transparent cutout	`gpt-image-2.5-sunburst` with `background:"transparent"` and `output_format:"png"` or `"webp"`	Verified real alpha
Drafts and iteration	either 2.5 model at `quality:"low"`	About $0.006 per 1024x1024 image
Legacy pipeline you have not migrated	`gpt-image-2`	Still live; slower and far more expensive at `high`
Anything else	do not	`gpt-image-1.5` shuts down 2026-12-01; `gpt-image-1` shuts down 2026-10-23; DALL-E is gone

Snapshot pins, both verified working: `gpt-image-2.5-sunburst-2026-09-08`, `gpt-image-2.5-flare-2026-09-08`.

Cost

Token rates, identical across GPT Image 2 and 2.5:

Token type	Rate per 1M
Image output	$30.00
Image input (edit reference images)	$8.00
Cached image input	$2.00
Text input (the prompt)	$5.00
Cached text input	$1.25

Text output is not billed; these models emit images.

Measured output tokens, 1024x1024 (Sunburst and Flare identical)

Quality	Output tokens	Cost
low	196	$0.0059
medium	439	$0.0132
high	1,756	$0.0527
xhigh	3,122	$0.0937
max	7,024	$0.2107

Measured output tokens by size

Size	Pixels	medium	high	max
1024x640	0.66M	239		
1536x864	1.33M	280		
1024x1024	1.05M	439	1,756	7,024
1536x1024	1.57M	343	1,372	5,488
1024x1536	1.57M	343		
2560x1440	3.69M	478		
3840x1280	4.92M	332		
2048x2048	4.19M	892		
3840x2160	8.29M	865	3,336	

Educational only. Not legal advice. No attorney-client relationship. ©2026 Chariton Media.

Size	Pixels	medium	high	max
1254x1254 (default)	1.57M			(low: 229)

Shape drives cost more than pixel count, and square is the worst shape. 1024x1024 at medium costs 439 tokens; 3840x1280 at medium costs 332 tokens for 4.7x the pixels. 2048x2048 costs more than 3840x2160. If cost matters and the aspect ratio is negotiable, pick landscape or ultrawide over square.

Edit input cost

A 1024x1024 reference image bills 1,024 image input tokens, about $0.0082. Two references bill 2,048. The prompt text bills separately at $5/M and is rounding error.

Cost measurement gotcha

The Image API returns a `usage` block with exact token counts. The Responses API `usage` block counts only the mainline model's tokens; image tokens are billed but never appear. **If you need to measure per-image cost, use the Image API.**

Latency (measured at 1024x1024)

Two independent runs, 2026-09-08. Ranges, not single samples.

Quality	Flare	Sunburst
low	6.2 to 8.7s	9.3 to 10.5s
medium	8.0 to 8.4s	11.4 to 12.5s
high	12.7 to 15.9s	24.1 to 24.5s
xhigh	19.0 to 20.2s	35.4s
max	35.9 to 36.0s	69.6 to 70.6s

At 1536x1024 `high`: Sunburst 22.6s, `gpt-image-2` 124.0s. Sunburst `max` at that size took 66.2s.

Sunburst is roughly 2x Flare's wall time from `high` upward. Below `high` the gap is small enough to ignore. Budget 60 to 75 seconds for Sunburst at `max` and set client timeouts accordingly; the default timeout in most HTTP libraries will cut it off.

Educational only. Not legal advice. No attorney-client relationship. ©2026 Chariton Media.

Text-to-image

Minimum request

```
curl -s -X POST "https://api.openai.com/v1/images/generations" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"model":"gpt-image-2.5-sunburst","prompt":"A diner chalkboard reading
\"LOBSTER ROLL $24\"","quality":"high","size":"1536x1024"}'
```

Transparent cutout

```
{
  "model": "gpt-image-2.5-sunburst",
  "prompt": "A single red cube isolated on a fully transparent background. No
shadow, no backdrop, no checkerboard.",
  "quality": "medium",
  "size": "1024x1024",
  "background": "transparent",
  "output_format": "png"
}
```

Verified: returns RGBA PNG, corner alpha 0, about 41 percent of pixels fully transparent on a product shot. A drawn checkerboard is not transparency; check the alpha channel, not the preview.

Parameters

Generations

Param	Required	Values	Default	Notes
`model`	Yes	see model table	none	2.5 models absent from `/v1/models` but valid
`prompt`	Yes	1 to 32,000 chars	none	empty string is a hard error
`quality`	No	`low`, `medium`, `high`, `xhigh`, `max`, `auto`	`auto`	`xhigh` and `max` are 2.5 only
`size`	No	`auto` or `WIDTHxHEIGHT`	`1254x1254`	see Size below
`n`	No	1 to 10	1	multiplies output tokens

Param	Required	Values	Default	Notes
`output_format`	No	`png`, `webp`, `jpeg`	`png`	
`output_compression`	No	0 to 100	100	jpeg and webp only; hard error on png
`background`	No	`transparent`, `opaque`, `auto`	`auto`	transparent + jpeg is a hard error
`moderation`	No	`auto`, `low`	`auto`	
`stream`	No	bool	false	see Streaming
`partial_images`	No	0 to 3	0	streaming only; hard error otherwise

Rejected outright: `input_fidelity` (unknown parameter), `response_format` (unknown parameter; GPT Image always returns `b64_json`).

Edits

Everything above, plus:

Param	Required	Notes
`image[]`	Yes	one or more reference files, multipart
`mask`	No	RGBA PNG; alpha 0 marks the region to edit
`size`	No	`auto` does **not** inherit the input dimensions; see gotcha below

Size

Hard constraints, each confirmed by its own error message:

- Both edges divisible by 16. `"Invalid size '1000x1000'. Width and height must both be divisible by 16."`
- Minimum 655,360 total pixels. `"Requested resolution is below the current minimum pixel budget."`
- Longest edge 3,840 or less. `"The longest edge must be less than or equal to 3840."`
- Aspect ratio 3:1 or tighter. `"The maximum supported aspect ratio is 3:1."` 3840x1280 passes; 3840x1264 fails.
- Maximum 8,294,400 total pixels.

Useful sizes: `1024x1024`, `1536x1024`, `1024x1536`, `1536x864`, `2048x2048`, `2560x1440`, `3840x2160`, `3840x1280`.

Educational only. Not legal advice. No attorney-client relationship. ©2026 Chariton Media.

`size` **omitted** yields 1254x1254. Unchanged from gpt-image-2 and still undocumented. Always pass `size` explicitly.

`size: "auto"` lets the model choose and it is not bound by the multiple-of-16 rule: a lighthouse prompt returned 1145x1374 on two separate runs, an infographic prompt returned 1536x1024. It is repeatable for a fixed prompt but you cannot predict it, so it is fine for exploration and wrong for anything a layout depends on.

Anything above 2560x1440 is documented as experimental. The 2K-then-upscale strategy still holds: generate and validate at 1536x1024 or 2048x2048, then upscale with Real-ESRGAN if a large-format destination genuinely needs 4K.

Quality

`auto` is adaptive, not a synonym for medium. Measured: a trivial prompt ("a red cube on white") resolved to `low`; a dense infographic prompt resolved to `high`, a 9x cost difference. The response body echoes the resolved value in its `quality` field, so log it if you are tracking spend.

Use `low` for drafts, `medium` for most production work, `high` when the image carries small text or dense labels, and `xhigh` or `max` only when a specific unmet quality requirement justifies 2x to 4x the cost and wall time. A higher setting does not guarantee a better result; on the chalkboard text test, Sunburst `high` read cleaner than Sunburst `max`.

Image Edit

```
curl -s -X POST "https://api.openai.com/v1/images/edits" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -F "model=gpt-image-2.5-sunburst" \
  -F "image[]=@original.png" \
  -F "prompt=Replace ONLY the white chairs with wooden chairs. Preserve camera
angle, lighting, shadows and every other object." \
  -F "quality=medium" \
  -F "size=1536x1024"
```

Multiple references, roles assigned in the prompt:

```
  -F "image[]=@scene.png" \
  -F "image[]=@dog.png" \
  -F "prompt=Place the dog from image 2 into the setting of image 1, beside the
woman. Match lighting and composition. Change nothing else."
```

Masked edit. The mask is an RGBA PNG the same size as the input; **alpha 0 marks the region to change**, alpha 255 preserves. Verified working on Sunburst.

```
  -F "image[]=@original.png" \
  -F "mask=@mask.png" \
  -F "prompt=Put a yellow rubber duck in the masked area."
```

Masking is prompt-guided, not a hard stencil; the model treats the mask as guidance and may not follow its shape exactly. If a region must stay pixel-identical, composite the approved edit back over the original rather than trusting the mask.

Edit gotcha: `size: "auto"` on the edits endpoint does not inherit the input image's dimensions. A 1024x1024 input with `size: "auto"` returned 1254x1254, the same default as generations. Pass an explicit `size` matching the source whenever dimensions matter.

Streaming

`stream: true` plus `partial_images: 0` to `3`. Server-sent events, `data:` lines:

Event type	Carries
`image_generation.partial_image`	`partial_image_b64`, `partial_image_index`, `sequence_number`, `size`, `quality`, `background`, `output_format`
`image_generation.completed`	`b64_json`, `usage`, `sequence_number`

`partial_images` is a ceiling, not a contract. Partials are emitted only when the render runs long enough to have intermediate states, so a fast render silently returns none no matter what you ask for. Measured at 1024x1024:

Model / quality	partial_images	Partials received	Output tokens
Flare low	0, 1, 2, 3	0 in every case	196
Sunburst low	0	0	196
Sunburst low	3	1	273
Sunburst high	3	3	1,985 (base 1,756)

Each partial billed 77 output tokens at 1024x1024, not the 100 the docs state. About $0.0023 each. Do not build a cost model on the documented figure; read `usage` off the `completed` event.

`partial_images` without `stream` is a hard error on both endpoints: `"Partial images are only supported with streaming."`

Responses API image tool

For multi-turn editing where the conversation carries the image forward.

```
{
  "model": "gpt-5.5",
  "input": "Generate a red cube on white.",
  "tools": [{
    "type": "image_generation",
    "model": "gpt-image-2.5-sunburst",
    "quality": "medium",
    "size": "1536x1024",
    "action": "auto"
  }]
}
```

The output array contains an `image_generation_call` item with `id`, `status`, `action`, `background`, `output_format`, `quality`, `size`, `revised_prompt`, and `result` (base64). A `message` item follows.

- `action`: `auto` lets the model decide generate versus edit, `generate` forces a new image, `edit` forces an edit and errors with no image in context.
- Multi-turn: pass `previous_response_id`, or feed the prior `image_generation_call` `id` back in the input array.
- **`revised_prompt` only exists here.** The Image API returns `b64_json` and nothing else in each data item.
- The 2.5 models cannot be the top-level `model` on `/v1/responses`; that returns 404 `model_not_found`. They go in the tool's `model` field only.
- Image tokens do not appear in the response `usage`. Use the Image API if you need to measure cost.

Rate limits

Published for `gpt-image-2.5-sunburst`. IPM is images per minute.

Tier	TPM	IPM
Free	not supported	not supported
1	100,000	5
2	250,000	20
3	800,000	50
4	3,000,000	150

Educational only. Not legal advice. No attorney-client relationship. ©2026 Chariton Media.

Tier	TPM	IPM
5	8,000,000	250

The image endpoints return no `x-ratelimit-*` headers at all. You cannot read remaining budget; you find the ceiling by hitting a 429. Build backoff in rather than a budget counter. Six concurrent requests ran clean during this pass.

Error catalogue (verbatim, 2026-09-08)

Trigger	HTTP	Message
`quality: "ultra"`	400	`Invalid value: 'ultra'. Supported values are: 'low', 'medium', 'high', and 'auto'.` (stale: xhigh and max work)
`output_format: "gif"`	400	`Supported values are: 'png', 'webp', and 'jpeg'.`
`background: "clear"`	400	`Supported values are: 'transparent', 'opaque', and 'auto'.`
`model: "gpt-image-2.5"`	400	`The model 'gpt-image-2.5' does not exist.`
`quality: "xhigh"` on gpt-image-2	400	`The model 'gpt-image-2' does not support quality 'xhigh'.`
`output_compression` with png	400	`Compression less than 100 is not supported for PNG output format`
`background: "transparent"` with jpeg	400	`Transparent background is not supported for JPEG output format`
`input_fidelity` on generations	400	`Unknown parameter: 'input_fidelity'.`
`input_fidelity` on edits	400	`The model 'gpt-image-2.5-sunburst' does not support the 'input_fidelity' parameter.`
`response_format: "url"`	400	`Unknown parameter: 'response_format'.`
`partial_images` without `stream`	400	`Partial images are only supported with streaming.`
`n: 11`	400	`Expected a value <= 10, but got 11 instead.`
prompt over 32,000 chars	400	`Expected a string with maximum length 32000`
empty prompt	400	`Invalid 'prompt': empty string.`

Educational only. Not legal advice. No attorney-client relationship. ©2026 Chariton Media.

Moderation blocks return `error.type = "image_generation_user_error"`, `error.code = "moderation_blocked"`, with an optional `error.moderation_details` object carrying `moderation_stage` (`input`, `output`, `unknown`) and coarse `categories`. Do not auto-retry a moderation block or any `image_generation_user_error` without changing the request; do retry 429 and 5xx with backoff.

Response shape

```
{
  "created": 1788926959,
  "background": "opaque",
  "output_format": "png",
  "quality": "low",
  "size": "1024x1024",
  "data": [{ "b64_json": "..." }],
  "usage": {
    "input_tokens": 11,
    "input_tokens_details": { "image_tokens": 0, "text_tokens": 11 },
    "output_tokens": 196,
    "output_tokens_details": { "image_tokens": 196, "text_tokens": 0 },
    "total_tokens": 207
  }
}
```

The top-level echo fields (`quality`, `size`, `background`, `output_format`) tell you what the API resolved `auto` to. Log them.

Educational only. Not legal advice. No attorney-client relationship. ©2026 Chariton Media.

Python helper

```
import base64, json, pathlib, time
from openai import OpenAI

client = OpenAI()  # reads OPENAI_API_KEY

def generate(prompt, model='gpt-image-2.5-sunburst', size='1536x1024',
             quality='medium', n=1, out_dir='./output', label='img', **kw):
    pathlib.Path(out_dir).mkdir(parents=True, exist_ok=True)
    r = client.images.generate(model=model, prompt=prompt, size=size,
                               quality=quality, n=n, **kw)
    cost = r.usage.output_tokens * 30 / 1e6 +
r.usage.input_tokens_details.image_tokens * 8 / 1e6
    print(f'{r.size} {r.quality} {r.usage.output_tokens} tok  ${cost:.4f}')
    ts = int(time.time() * 1000)
    ext = kw.get('output_format', 'png')
    paths = []
    for i, img in enumerate(r.data):
        fp = f'{out_dir}/{label}_{ts}{"_" + str(i+1) if len(r.data) > 1 else
""}.{ext}'
        pathlib.Path(fp).write_bytes(base64.b64decode(img.b64_json))
        paths.append(fp)
    return paths

generate('A diner chalkboard reading "LOBSTER ROLL $24"', quality='high',
label='chalkboard')
```

Transparent cutout, and verify the alpha rather than trusting the preview:

```
from PIL import Image
fp = generate('A single red cube, isolated, no shadow', quality='medium',
              size='1024x1024', background='transparent', output_format='png',
              label='cutout')[0]
im = Image.open(fp)
assert im.mode == 'RGBA' and im.split()[-1].getpixel((0, 0)) == 0, 'not
transparent'
```

Educational only. Not legal advice. No attorney-client relationship. ©2026 Chariton Media.

Node helper

```
import fs from 'node:fs';
import OpenAI from 'openai';
const client = new OpenAI();

async function generate({ prompt, model = 'gpt-image-2.5-sunburst', size =
'1536x1024',
  quality = 'medium', n = 1, outputDir = './output', label = 'img', ...rest })
{
  fs.mkdirSync(outputDir, { recursive: true });
  const r = await client.images.generate({ model, prompt, size, quality,
n, ...rest });
  console.log(`${r.size} ${r.quality} ${r.usage.output_tokens} tok $$
{(r.usage.output_tokens * 30 / 1e6).toFixed(4)}`);
  const ts = Date.now();
  return r.data.map((img, i) => {
    const fp = `${outputDir}/${label}_${ts}${r.data.length > 1 ? `_${i + 1}` :
''}.${rest.output_format ?? 'png'}`;
    fs.writeFileSync(fp, Buffer.from(img.b64_json, 'base64'));
    return fp;
  });
}
```

Prompting

The 2.5 prompting guide is spec-shaped, not prose-shaped. Write the prompt like an artifact spec.

1. **Name the deliverable.** Product photograph, pitch-deck slide, workflow diagram, page image, UI mockup. Then composition, aspect ratio, and placement constraints.
2. **Pick a readable format.** Short prompt, labelled paragraphs, JSON-ish keys, tags. The model handles all of them; choose whichever is easiest for you to edit later. Do not chase magic syntax.
3. **Describe what is visible.** Materials, lighting, colors, medium. Say "photorealistic" or "real photograph" when you mean it. Camera specs are appearance cues, not physics.
4. **Specify people and action.** Body framing, scale, gaze, and interaction with objects. "Full body visible, feet included" and "hands gripping the handlebars" carry more than a mood word.
5. **Quote exact text and say how many times it appears.** Position and typography too. Spell unusual names letter by letter. End with "no extra text, no watermarks." Use `high` or above for small text or multiple fonts.
6. **In edits, separate the change from the constraints.** "Change only X." Then list what must survive: identity, geometry, layout, lighting, labels, camera angle. Name exclusions.
7. **Assign a role to every reference image.** "Image 1 is the scene, image 2 is the subject, image 3 is the palette." Say how they combine.

Educational only. Not legal advice. No attorney-client relationship. ©2026 Chariton Media.

8. **Iterate one change at a time.** Feed the previous output back as the input, request one change, restate the constraints that must hold. Repeated edits drift; restate or composite.

A structure that works

```
Create one <deliverable> titled "<title>".

Canvas: <size>, <background>, <typography>.
Content:
  - <element with real data, not placeholder>
  - <element>
Text (verbatim, appears once): "<exact copy>"
Constraints: no clip art, no stock photography, no gradients, no watermarks, no
extra text.
```

Text rendering, measured

A four-line chalkboard prompt at 1536x1024, all four lines correct in every variant tested. OCR similarity against ground truth (tesseract on a chalk-on-slate surface, so the absolute numbers measure the OCR as much as the model; use them for ranking only):

Variant	Cost	Time	OCR similarity
`gpt-image-2` high	$0.165	124.0s	0.561
Sunburst high	$0.041	22.6s	0.559
Sunburst max	$0.165	66.2s	0.505
Flare high	$0.041	15.8s	0.447
Flare medium	$0.010	12.3s	0.385

Read: Sunburst `high` matches gpt-image-2 `high` on text at a quarter the cost and a fifth the time. Going up to `max` bought nothing here. Flare needs `high` for sign copy; `medium` starts dropping characters.

Still true

- Long body copy degrades. Use the composite workflow below.
- Style preservation in edits is reinterpretation, not preservation. Mask or composite when it matters.

Text-Heavy Composites

When an image needs more than a few lines of copy, generate the background clean and lay the type over it in HTML. Zero additional model cost, full typographic control, editable and localizable afterward.

1. Prompt for the scene with no text in it.

2. Use the output as a CSS background; layer the copy with real fonts.
3. Render to PNG through headless Chromium.

```
import puppeteer from 'puppeteer';

async function compositeToPNG({ html, outputPath, width = 1200, height = 630 })
{
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.setViewport({ width, height, deviceScaleFactor: 2 });
  await page.setContent(html, { waitUntil: 'networkidle0' });
  await page.screenshot({ path: outputPath, type: 'png' });
  await browser.close();
}
```

Playwright is a drop-in swap (`chromium.launch()`, `page.setViewportSize`).

Use it for: social cards, thumbnails, ad creative, infographics with extensive labels, text-dominant posters, anything templated or localized.

Do not use it for: short text baked into the visual (chalkboards, menus, storefronts), lettering that is part of the art, or text that must interact spatially with scene elements.

Gotchas, consolidated

- The 2.5 models are missing from `GET /v1/models` but work. Availability is not discoverable from the list.
- The `quality` validation error omits `xhigh` and `max`. The error is stale; the tiers work.
- `quality: "auto"` swings 9x in cost between a simple and a complex prompt. Log the resolved value.
- `size` omitted gives 1254x1254, not 1024x1024. Always pass it.
- `size: "auto"` ignores the multiple-of-16 rule and is not reproducible.
- `size: "auto"` on **edits** does not match the input image dimensions.
- Square is the most expensive shape per pixel. Landscape and ultrawide are much cheaper.
- Flare and Sunburst bill identically. Choose on latency, never on price.
- Sunburst at `max` can take 70 seconds; raise your HTTP client timeout.
- `output_compression` on png is a hard error.
- `background: "transparent"` with jpeg is a hard error. png or webp only.
- No `revised_prompt` from the Image API; only from the Responses tool.
- No `x-ratelimit-*` headers on the image endpoints.
- Responses API `usage` hides image tokens.

Educational only. Not legal advice. No attorney-client relationship. ©2026 Chariton Media.

- `input_fidelity` and `response_format` are both dead parameters.
- Masks are guidance, not stencils. Composite when a region must be pixel-identical.
- A fast render emits no partial images however high you set `partial_images`.
- A partial image bills 77 output tokens at 1024x1024, not the documented 100.
- A drawn checkerboard is not transparency. Assert on the alpha channel.

Prompt Library

700+ community prompts: https://github.com/YouMind-OpenLab/awesome-gpt-image-2

```
PROMPT_LIB="./gpt2-prompts/awesome-gpt-image-2.md"
[ -f "$PROMPT_LIB" ] || { mkdir -p "$(dirname $PROMPT_LIB)"; curl -s \
  https://raw.githubusercontent.com/YouMind-OpenLab/awesome-gpt-image-2/main/
README.md -o "$PROMPT_LIB"; }
grep -n "^### No\." "$PROMPT_LIB" | grep -i poster
```

Written against gpt-image-2; the prompts port to 2.5 unchanged. Treat as starting points.

Staleness

Everything above was measured on **2026-09-08**, the day GPT Image 2.5 shipped. Token counts, latencies and error strings are point-in-time facts about somebody else's API, and OpenAI changes them without announcement; this pass alone found four documented behaviors that were already wrong.

Trust the mechanics (endpoints, parameter names, what errors and why) for a long time. Treat the exact numbers as of that date. If a cost forecast or a timeout depends on one of them, read the `usage` block off a real response rather than this table.

Educational only. Not legal advice. No attorney-client relationship. ©2026 Chariton Media.