PIXELBENCH: EVALUATING LANGUAGE MODELS FOR PIXEL-ART CREATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Pixel art is widely used in games and digital content creation, requiring readable forms, coherent spatial relationships, and harmonious colors under limited resolution and palette constraints. Language models show growing promise in visual creation, yet their pixel art creation capabilities lack systematic evaluation. We introduce PixelBench, formalizing text-to-image generation and instruction-guided editing for pixel art and providing a benchmark for these tasks. We further present a creation environment in which language model agents invoke drawing tools through code to generate and edit pixel art. We assess the quality of generated artwork through human evaluation and employ a multimodal large language model as a judge, validating the agreement between its automated assessments and human preferences. We compare language models and configurations across diverse subjects and canvas sizes. In our experiments, GPT-6 Astra achieves both the highest average judge score and the best average human rank, highlighting the potential of general-purpose language models for pixel art creation. We release the tasks, all artworks, the human rankings, and the judge protocol.

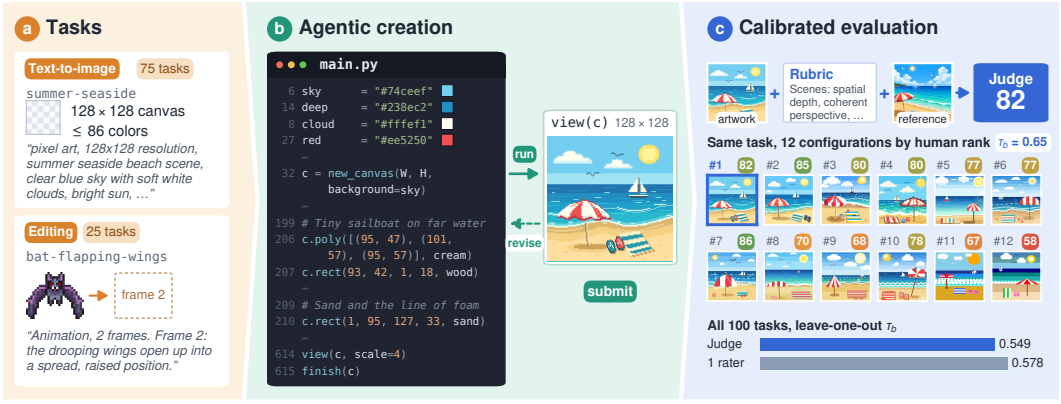


Figure 1: PixelBench on the task `summer-seaside`. (a) A task gives a written requirement and a native canvas; an editing task also gives the artwork to modify. (b) The model writes, runs, views, and revises a drawing program until it submits. The code shown is part of the program submitted by GPT-6 Astra in the evaluation. (c) The calibrated judge scores each artwork given a rubric and a reference image. Below: the task's 12 submissions by mean human rank, #1 to #12 reading left to right and top to bottom, each with its judge score; and leave-one-out agreement over all 100 tasks.

1 INTRODUCTION

Pixel art is digital painting in which the individual pixel is the smallest unit of drawing. It is one of the most common art forms in two-dimensional games, used for characters and their animation frames, props, map tiles, and interface elements, and it remains in wide use in independent and commercial games today (Silber, 2015). Beyond games it appears in application icons, emoji, and avatars, and is carried over into physical media such as fuse beads and cross-stitch. Pixel artists

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) GPT-Image-2
1254 × 1254 · 234,693 colors

(b) + Pixel Snapper
203 × 209 · 86 colors

(c) GPT-6 Astra
128 × 128 · 25 colors

Figure 2: An example PixelBench task, `summer-seaside`: a 128 × 128 summer seaside with at most 86 colors; insets enlarge the sailboat. (a) The output of an image generation model imitates pixel art, but its pixels are not clean cells: many are not square, straddle cell boundaries, or blend several colors, and no fixed palette constrains them. (b) Pixel Snapper (Duprez, 2025) infers a grid from that image and quantizes it to the task's 86 colors; every cell is now uniform, but the grid is 203×209 rather than the required canvas. (c) GPT-6 Astra, working in PixelBench, writes a program that draws on the native canvas cell by cell; shown is its benchmark submission.

work in dedicated editors, among them Aseprite, Pyxel Edit, GraphicsGale, and Piskel (Igara Studio S.A., 2026; Pyxel Edit, 2026; HUMANBALANCE Ltd., 2026; Piskel, 2026). These tools are built around how the medium is made: the artist places colors cell by cell with a pencil tool on a magnified native canvas, maintains a limited palette, organizes the picture in layers, builds animation frame by frame, and exports a sprite sheet laid out on a fixed grid. The constraints behind that workflow are what make the medium hard: the canvas is the size of the finished work, so a 32 × 32 character has only 1024 cells (Wu et al., 2022), and because the work is displayed at integer nearest-neighbor magnification without interpolation, every pixel stays a hard-edged block (Saravanan & Guzdial, 2022). The color of every cell is therefore a design decision that directly determines whether a silhouette is clean and whether a form holds together.

Because drawing by hand is expensive, several technical routes aim to have models produce pixel art. The first uses image generation models directly. These models produce continuous images at a fixed high resolution, latent diffusion models for example through a latent space and a decoder (Rombach et al., 2022); with a style prompt they produce pixel-style pictures, but those pictures do not lie on a pixel grid. The raw GPT-Image-2 output for the 128 × 128 `summer-seaside` task of Figure 2 carries 234,693 distinct colors, and its apparent blocks are neither uniform in color nor aligned to any single pitch. Tools exist that recover a grid from such a picture and quantize its palette (Duprez, 2025), but they can only infer the grid the picture happens to imply, and that grid is a property of the generated image rather than of the requirement: on this image it comes out 203 × 209, not the 128 × 128 the task asks for. The second route is pixelization, which converts an existing ordinary image into pixel art (Gerstner et al., 2012; Han et al., 2018; Wu et al., 2022). In both routes the work is first produced as a continuous bitmap and the grid and palette are imposed afterwards; the per-cell color decisions are made by a conversion algorithm rather than by the creator, and the output carries no layers or other editable structure.

Language models offer another route: drawing directly on a discrete canvas at native resolution, as they already do for vector graphics and small drawing programs (Rodriguez et al., 2025; Yang et al., 2025; Rodriguez et al., 2026; Nedungadi et al., 2026; Emergent Mind, 2026). Pixel art suits this form of expression: the canvas is an integer matrix, colors form a finite symbolic set, and drawing operations are discrete geometry that a program expresses directly. What this route yields also has different properties: the creator makes the per-cell decisions, any single pixel can be located and changed exactly, and the creation process persists as a program with its layers intact, one that can be re-run and edited further and so enters a human workflow directly.

What language models can actually do on this route has not been measured credibly. Existing evaluations generate on small 24 × 24 grids in one turn, without tools or revision, and without independent human data to calibrate their automatic judges (Emergent Mind, 2026; Nedungadi et al., 2026). Judging pixel-art quality depends heavily on human perception: whether a silhouette flows, whether volume holds, whether a palette coheres are not properties a rule can compute. Evaluating

pixel-art creation by language models therefore requires solving two problems at once: how to define tasks at production-relevant canvas sizes and have models complete them, and how to build an evaluation protocol that is both grounded in human judgment and able to run at scale.

We propose PixelBench to answer both, as Figure 1 lays out. On tasks, PixelBench contains 75 text-to-image problems and 25 editing problems. The generation problems sit on native canvases from 16×16 to 256×256 and cover seven subjects: characters, creatures, scenes and architecture, objects, icons and interface elements, materials, and effects. The editing problems come in three types: instruction editing, animation, and multiview. On method, we provide an agentic creation environment: the model writes a Python program against a drawing API, runs it, views the rendered result, edits the source, and runs again until it submits. On evaluation, four human raters produced complete rankings of all 1200 artworks, and we use them to calibrate a multimodal judge that scores one artwork at a time. Calibration works on three fronts: how the artwork is presented, the rubric, and a same-task reference image. Together they raise the agreement between the judge's order of a task's artworks and the raters' (per-task Kendall τ_b) from 0.4507 to 0.5583. To make this number interpretable, we put the judge and the raters on the same footing: we hold out one rater at a time and measure how well that rater, and the judge, agree with the mean ranking of the other three. The four raters reach 0.5464 to 0.6200 this way and the judge 0.5487, placing it inside the range the raters themselves span, and the 95% confidence interval of the paired difference contains zero, so no difference is resolved at this sample size. At the model level the judge and human leaderboards correlate at Spearman $\rho = 0.923$.

Under this protocol we evaluate 12 frontier model configurations. GPT-6 Astra attains the highest judge score and the best mean human rank on both generation and editing. Models separate most, and human raters agree most, on concrete subjects such as characters, creatures, and scenes and on canvases above 64 pixels. With image observation switched off, GPT-6 Astra's generation shows no measurable change, while its mean human rank on editing falls from 3.39 to 5.00: an edit starts from an artwork someone else drew, and seeing it lets the model understand it better.

The contributions of this paper are as follows.

1. **We build PixelBench, which formalizes pixel-art generation and editing for language model agents.** An artwork is a color-index matrix at a specified native canvas size; the environment guarantees canvas size and color ceiling, and form, composition, and color are left to human judgment.
2. **We release a pixel-art creation environment for language model agents.** The model writes a program that calls drawing tools and creates through a loop of running, viewing, and revising its source, so an artwork is fully determined by a program that can be re-run, audited, and edited further. Every configuration in this paper creates in this one environment, and it also ships as a reusable agent skill.
3. **We build a multimodal judge for pixel art and calibrate it against human rankings.** Using complete rankings of 1200 artworks by four raters, we quantify how much presentation, rubric, and reference image each contribute to agreement with the consensus order, and, under one leave-one-out reference, place the calibrated judge inside the range the four raters themselves span.
4. **We evaluate 12 frontier model configurations and analyze their behavior.** We report judge and human leaderboards for generation and editing, and analyze how the differences between models are distributed across subjects, canvas sizes, and editing types.

2 Related Work

Pixel-art generation and pixelization. Dedicated generators for pixel art include GAN-based character sprite generation across orientations (Coutinho & Chaimowicz, 2022) and VQ-VAE representations tailored to the medium (Saravanan & Guzdial, 2022). Pixelization converts an ordinary image into pixel art, through superpixel abstraction with a jointly optimized small palette (Gerstner et al., 2012), unsupervised domain translation that learns grid structure and a crisp pixel style (Han et al., 2018), or staged methods that let the pixel cell be controlled (Wu et al., 2022); related tools align the pixel-style output of an image generation model to a grid and quantize its palette (Duprez, 2025). In all of these the per-cell colors are decided by a conversion or generation algorithm. PixelBench

studies a different mode of creation: a general-purpose language model, given an open written requirement, starts from an empty canvas or an existing artwork and makes the per-cell color decisions itself on the native grid.

Program-based drawing. Generating vector graphics with language and multimodal models is now systematically studied, covering image- and text-to-SVG generation and multi-task evaluation of understanding, editing, and generation (Rodriguez et al., 2025; Yang et al., 2025; Rodriguez et al., 2026): constructing visual content through code is a viable route. On discrete canvases, Pixel Art Bench (Emergent Mind, 2026) has text models emit a 24×24 palette-indexed grid in one turn and computes Elo with a pairwise judge, and Autoregressive Mosaics (Nedungadi et al., 2026) has models write short programs that call drawing primitives on a 24×24 canvas in order to probe two-dimensional spatial reasoning; DrawingBench (Kim & Ryu, 2025) instead evaluates drawing through mouse-level GUI actions. PixelBench evaluates complete artworks and edits on canvases up to 256×256, created through a loop in which the model operates an explicit integer canvas with discrete primitives and revises its own program, and calibrates its judge against complete human rankings.

Human-aligned evaluation of visual generation. Using a strong model as a judge has become standard practice for open-ended generation (Zheng et al., 2023), and its credibility depends on calibration against human preference (Chiang et al., 2024). In the visual domain, GenEval and T2I-CompBench++ decompose text-image alignment into checkable factors (Ghosh et al., 2023; Huang et al., 2025), while SVGauge and SVG-Score study human-aligned evaluation for SVG generation (Zini et al., 2025; Cipriano et al., 2026). For pixel art we give a concrete calibration path for the judge, together with the measured effect of each step on human agreement.

3 PIXELBENCH

Each PixelBench task gives a written requirement and a specification, and the model delivers a pixel artwork $A = (G, P, M)$, where $G \in \{0, \ldots, |P| - 1\}^{H \times W}$ is a matrix of color indices on the native canvas, P is the palette, and M records the native size, the background policy, and the frame structure. Stored in this form, an artwork's color count, modified regions, and inter-frame differences are exactly computable. An artwork is rendered as an image only to be observed or evaluated, by nearest-neighbor magnification without interpolation; before it reaches the judge, a transparent margin is added on its left and top, a change whose effect on judge–human agreement Section 5.3 measures.

The environment guarantees the specification. An artwork of the wrong size or with too many colors raises an error as soon as the program declares it, and the model fixes the program and runs it again. The specification constrains size and color, not style: whether to use dithering, outlines, or hand-placed transitional colors is the creator's choice, and meeting the specification does not make an artwork good pixel art. Good pixel art reads clearly at native size, has clean silhouettes, groups its colors by light and shade to model form and volume, has a coherent composition and palette, and delivers what the task asks for; these criteria resist being written down as computable rules and are judged by human ranking and by the calibrated judge (Section 5).

Every one of the 100 tasks was written and reviewed by people, with generative AI assisting on descriptive detail. Appendix B gives a card for every task with its statement, canvas, and color ceiling, and, for editing tasks, the source artwork with a citation of the work it comes from; Appendix A shows every artwork produced for every task.

3.1 TEXT-TO-IMAGE GENERATION

Each of the 75 generation tasks gives an English requirement and a native canvas size, and starts from an empty canvas. The tasks cover seven subjects: *objects* (21), *scenes and architecture* (16), *characters* (13), *creatures* (10), *icons and interface elements* (7), *materials* (5), and *effects* (3). Canvases range from 16×16 to 256×256, most often 64×64 (26 tasks) and 32×32 (18).

Every task also carries a color ceiling, and the ceiling plays two different roles. On seven tasks it is a tight budget of four to ten colors and it binds: these are 16×16 and 32×32 pieces where working within a handful of colors is part of the task, and on each of them the most colorful submission

comes within one or two colors of the ceiling. Everywhere else it is set generously, at 86 colors for the bulk of the benchmark and 120 to 140 for the four 256×256 tasks, and models stay far below it, the median submission using 16 colors against a ceiling of 86; its role there is to keep a continuous-tone output from passing as pixel art.

3.2 EDITING

The 25 editing tasks each give an existing pixel artwork and an instruction, and require the modification to be made while the original is preserved. They come in three types. The 13 *instruction editing* tasks produce a single frame, for example melting an ice cream or removing a character from a scene and filling in the background. The 10 *animation* tasks take the original as the first frame and continue a specified action, such as a punch or a flag waving, seven of them as two-frame and three as three-frame sequences. The 2 *multiview* tasks ask for a three-view turnaround of a character and for two rotated views of a house. Source canvases range from 27×22 to 256×128. Editing tasks set no color ceiling, and the model may enlarge the canvas as the edit requires; all frames of a multi-frame result share one size. The originals were collected from the internet, and three were drawn by the authors; sources are listed in Appendix B. Editing tests a capability distinct from generation: understanding the structure of an existing artwork, changing only the part that should change, and keeping the rest consistent in content and style.

4 AGENTIC PIXEL-ART CREATION

The creation environment places every model under identical conditions. The system prompt documents the file protocol and the drawing API, and carries no pixel-art technique guidance and no exemplar artwork; every model receives the same prompt, tools, and budget, and composition, color use, and drawing order are left entirely to the model. The leaderboard therefore compares the models themselves as pixel-art creators, not prompt designs tailored to any one of them.

File protocol. The model works in a source workspace and delivers the artwork as a Python program. It manages source through `write_file`, `edit_file`, `read_file`, and `list_files`, and executes it through `run(path)`. Every run starts from the task input in a fresh subprocess, and no variable or canvas survives from the previous run, so an artwork is fully determined by its source. A program declares its output with `finish(canvas)` or `finish([f1, f2, ...])`, at which point the environment checks the specification (Section 3) and freezes the result; only `run(path, submit=True)` submits formally and ends the task. The budget counts source modifications: 25 `write_file`/`edit_file` calls for generation and 15 for editing, with reads and runs free. At the last modification the environment warns the model to submit.

Drawing API. Programs have a pixel-art drawing library preloaded. `new_canvas(W, H)` creates a canvas; several canvases can serve as layers and are composited in order by `flatten`. A canvas provides single pixels, lines, rectangles, circles and ellipses, polygons, SVG paths with Bézier curves and arcs, flood fill, mirroring, shifting, outlines, checkerboard and Bayer dithering, and text. No primitive anti-aliases, and `None` erases. Colors can be given by name, hex, or RGBA and registered as named colors that form the palette of the artwork. `select(x, y, w, h)` lifts a region that can be flipped, resized, or rotated (nearest-neighbor or RotSprite) and then copied or moved elsewhere; in editing and animation tasks this is how models move limbs and reuse parts. An artwork can also be saved as a layered `.ase` or `.ora` document and opened for further work in pixel-art editors such as Aseprite (Igara Studio S.A., 2026).

Creation loop. Figure 1b shows the loop. `view(canvas)` magnifies the canvas with nearest-neighbor sampling to about 512 pixels on the long edge and returns it to the model as an image, and it is the only channel through which a model obtains an image observation: for the four evaluated models without image input and for the `novision` configuration of GPT-6 Astra, `view` returns no image and the model reads the artwork as a character picture instead (Appendix E). The environment records the source snapshot of every run, standard output, canvas checkpoints, and every preview the model requested. Editing tasks provide the source artwork as `input_canvas`, and these records show how models work from it: on 23 of the 25 editing tasks, GPT-6 Astra first writes a two-line inspection script that magnifies the original and prints its character matrix, and only then writes the main program that modifies it. On `fist-swing`, for instance, GPT-6 Astra inspects the 29×44

original, widens the canvas to 37×44 so that the punch has room, and places the original unchanged in the first frame with `blit`. It copies that frame, clears the rectangle holding the arm, and redraws the arm extended, laying down outline, mid-tone, and highlight as three polygons over the same silhouette; it then views both frames, widens the cleared rectangle by one pixel, views again, restores it, and submits. Across the 25 editing tasks, 11 of GPT-6 Astra's submissions enlarge the canvas and 12 contain two or three frames. Blitting, clearing, and selections let a model keep the original's pixels wherever the edit does not reach, and the preview after each run is what tells it whether the change reads at native size.

Agent skill. The environment also ships as an agent skill, a folder of instructions, scripts, and resources an agent loads when a task calls for them (Anthropic, 2025), bringing the same canvas representation and drawing primitives to general-purpose coding agents.

5 Evaluation Protocol

5.1 Human ranking

Four human raters ranked all 1200 artworks (100 tasks $\times$ 12 model configurations). For each task a rater saw the task statement, the canvas requirement, and the 12 artworks, anonymized and randomly ordered, and ranked them from best to worst on visual quality and on how well each follows the instruction. Every task also showed a reference for what the task asks for: for generation a GPT-Image-2 (OpenAI, 2026c) rendering of the same statement, and for editing the original plus the reference image the judge receives in Section 5.3. The frames of each animation were played in a loop so that raters could judge whether the motion is continuous. Across the 400 ranking lists, the per-task mean Kendall τ_b between pairs of raters is 0.5079. An artwork's four ranks average to its *mean human rank*; a task's twelve form the *consensus order* the judge is calibrated against, and averaging over tasks gives a configuration's place on the *human leaderboard*.

5.2 The multimodal judge

The automatic judge serves evaluation at scale and lets the community compare new models on PixelBench; the judge leaderboard is therefore the main table of this paper, with the human leaderboard supplying its calibration. The judge scores a single artwork independently: each call contains only the task statement, the canvas and color requirement, and one anonymized artwork, and outputs a single integer from 0 to 100. No two calls share context, and no competing candidate is ever shown. An editing call also shows the original, a multi-frame result is laid out in order as one image with the prompt stating the layout, and a material tile is shown repeated 2×2 to expose its seams. The judge model is GPT-6 Astra with reasoning effort set to `xhigh`, and Appendix C gives the complete prompts. Scoring a new model's 100 submissions therefore costs 100 calls to the judge model, and scoring each artwork once is enough for a leaderboard: 61 of the 66 pairs of configurations in Table 2 keep their order in at least 99% of simulated repeat evaluations, and GPT-6 Astra's margin over every other model family holds in all of them (Appendix D).

We measure agreement as the per-task Kendall τ_b (Kendall, 1945) between the judge's twelve scores for a task and the negated mean human ranks, so that both run best to worst and $\tau_b = 1$ means the judge reproduces the consensus order; agreement is averaged over tasks with equal weight, with 95% confidence intervals from 10,000 bootstrap resamples at the task level. To compare two judge protocols, or two configurations of one model, we take their difference on every task and report the mean of this paired difference with its 95% interval, which excludes zero when the two differ significantly at this sample size.

5.3 Calibrating the judge

Starting from the simplest protocol, we added three changes in turn, ran each version on all 100 tasks, and checked it against the four-rater rankings (Table 1). In the simplest protocol the artwork is magnified with nearest-neighbor scaling so that its longer side is 512 pixels and the prompt asks only for an overall quality score, which reaches a per-task mean τ_b of 0.4507.

Table 1: Calibrating the judge. Per-task mean Kendall τ_b against the four-rater consensus order, with a 95% bootstrap interval over tasks. Each row adds one change to the row above, and Appendix C gives the exact prompt of each version; the last row is the final protocol.

Protocol	Change	All (100)	95% CI	Generation (75)	Editing (25)
Simplest	single image, overall score	0.4507	[0.4032, 0.4971]	0.4571	0.4314
+ presentation	1/16 transparent margin	0.5190	[0.4707, 0.5645]	0.5204	0.5150
+ rubric	by subject and task type	0.5494	[0.5107, 0.5874]	0.5492	0.5499
+ reference	same-task reference image	**0.5583**	[0.5175, 0.5977]	**0.5537**	**0.5720**

Presentation. We add a transparent margin of one sixteenth of the canvas dimension on the left and top of the artwork before magnifying. This change alone raises τ_b to 0.5190, a paired difference of $+0.068$ $[+0.030, +0.107]$, with 66 of the 100 tasks improving. The effect is consistent with how a patch-based vision encoder consumes the image: on a small canvas, magnifying to 512 turns every pixel into a block of 16×16 or larger, so that a patch can fall entirely inside a single flat color and the spatial arrangement that distinguishes one artwork from another is lost. Offsetting the grid by a fraction of a block, with the magnification left non-integer, breaks that alignment. How the judge model encodes images is not public, and we offer this as an account of the measured effect rather than a claim about its architecture.

Rubric. The rubric reaches the judge as prompt text: every call appends a supplement for the task's category after the task statement. Each generation subject has its own supplement naming what to weigh: pose, silhouette, and anatomy for characters; spatial depth and the separation of foreground from background for scenes; continuity across the tile boundaries for materials. Four of the seven subjects append one further sentence chosen by canvas size, asking that every pixel be placed deliberately at 32 pixels and below, and that detail serve form above it. The editing supplement weighs instruction compliance, preservation of what the instruction does not change, and craft; animation adds inter-frame continuity and pose plausibility, and multiview adds identity across views. Agreement rises to 0.5494.

Reference image. Finally, we show the judge a same-task reference image before the candidate. For generation it is GPT-Image-2's output for the same statement, and the prompt states that it is not pixel art, not ground truth, and not to be scored against for resemblance. Among the editing tasks, animation, multiview, and one instruction-editing task use the corresponding frames from the asset the original comes from, and the other twelve use GPT-Image-2's edit. Agreement reaches 0.5583 $[0.5175, 0.5977]$, the highest of the four versions, and 0.5720 on editing. The raters ranked with these same references in view (Section 5.1), so both now work from a shared point of comparison. We fix the protocol here; every judge result below comes from this version. From the simplest to the final protocol, per-task agreement rises by $+0.108$ $[+0.071, +0.144]$, improving on 70 of the 100 tasks.

The gain is largest where nothing was tuned. Every wording choice that was made by looking at agreement was made on a development subset of 23 generation tasks, which Appendix C lists; the four versions were then fixed, and each was run once over all 100 tasks. Nothing in the prompts is task-specific: the rubric is written per subject and task type, and every task in a family receives the same text. Over the 77 tasks outside the development subset, agreement rises from the simplest to the final protocol by $+0.122$ $[+0.081, +0.162]$, above the $+0.108$ over all 100 tasks.

The judge falls within the spread of the raters. What decides a leaderboard is the order of the twelve artworks of a task, so that is where we place the judge and the raters side by side, using leave-one-out: hold out one rater, compare both that rater and the judge against the mean rank of the other three, and rotate through all four. Across the four rotations the held-out rater reaches 0.5464, 0.5630, 0.5835, and 0.6200, averaging 0.5782 $[0.5428, 0.6116]$, and the calibrated judge reaches 0.5487 $[0.5097, 0.5867]$: the judge sits inside the range the raters span, and above one of them. The per-task paired difference is -0.030 $[-0.068, +0.008]$, so no difference between the judge and a held-out rater is resolved at this sample size, and the same holds separately for generation (-0.030 $[-0.077, +0.015]$) and editing (-0.027 $[-0.091, +0.031]$). Even at the far end of that interval the judge trails a held-out rater by less than the raters disagree among themselves. At the model level the judge and human leaderboards correlate at Spearman $\rho = 0.923$ ($p = 1.9 \times 10^{-5}$).

Table 2: Leaderboard over 12 configurations, ordered by mean judge score over all 100 tasks under the final protocol. The two rank columns are mean within-task ranks over the twelve configurations, the judge's from its scores and the human one over four raters; both are lower-is-better and live on the same scale. The value in parentheses is the provider's documented default reasoning effort, which every call used; the two Qwen models document a token budget instead.

Configuration	All (100)			Generation (75)			Editing (25)		
	Score	Rank ↓		Score	Rank ↓		Score	Rank ↓	
		Judge	Human		Judge	Human		Judge	Human
GPT-6 Astra (medium)	**82.01**	**3.20**	**3.12**	**81.12**	3.17	**3.04**	**84.68**	**3.28**	**3.39**
GPT-6 Astra, novision (medium)	81.44	3.31	3.81	81.03	**2.99**	3.41	82.68	4.28	5.00
Qwen3.8 Max	77.02	5.26	5.65	76.21	5.40	5.92	79.44	4.84	4.84
GPT-5.6 Sol (medium)	76.47	5.52	5.63	77.25	5.01	5.15	74.12	7.04	7.10
Claude Opus 5 (high)	76.30	5.46	5.44	75.41	5.73	5.61	78.96	4.64	4.91
Gemini 3.8 Flash (medium)	75.94	5.56	4.99	74.64	5.59	4.93	79.84	5.44	5.17
Kimi K3 (max)	73.30	6.57	6.61	73.84	6.48	6.54	71.68	6.84	6.82
DeepSeek V4 Flash (high)	72.41	7.34	7.50	71.60	7.59	7.84	74.84	6.60	6.46
GLM-5.3 Flash (max)	71.22	7.16	7.15	71.99	7.21	7.34	68.92	7.00	6.59
DeepSeek V4 Pro (high)	67.01	8.50	8.72	68.09	8.55	8.76	63.76	8.34	8.61
Gemini 3.1 Pro Preview (high)	62.53	9.32	8.92	64.01	9.39	8.89	58.08	9.10	9.00
Qwen3.5-397B-A17B	55.84	10.81	10.45	57.36	10.88	10.57	51.28	10.60	10.11

Replacing the judge model leaves the leader in place. Scoring all 1200 artworks again under the final protocol with Claude Opus 5 puts GPT-6 Astra first and Claude Opus 5 itself fourth; the two judge leaderboards correlate at Spearman $\rho = 0.972$ (Appendix F).

6 RESULTS

Table 2 gives the results for the 12 model configurations, over all 100 tasks and separately over generation and editing. GPT-6 Astra ranks first on both leaderboards, with a judge score of 82.01 and a mean human rank of 3.12, and its `novision` configuration second, at 81.44 and 3.81. No configuration moves by more than three places between the two leaderboards.

Ranking the judge's 12 scores within each task, with ties sharing the mean of their positions, and averaging over tasks puts the two leaderboards on one scale, since a rater's output is a ranking to begin with. Read this way the two land on top of each other: the judge's mean rank differs from the raters' by 0.24 places on average and nowhere by more than 0.6 of a place, and the ordering by judge rank places ten of the twelve exactly where the raters place them. The rank columns correlate at $\rho = 0.937$ over all 100 tasks, 0.958 on generation, and 0.965 on editing.

The models are GPT-6 Astra (OpenAI, 2026b), GPT-5.6 Sol (OpenAI, 2026a), Qwen3.8 Max (Qwen Team, 2026b), Claude Opus 5 (Anthropic, 2026), Gemini 3.8 Flash (Google DeepMind, 2026b), Kimi K3 (Kimi Team et al., 2026), DeepSeek V4 Flash and Pro (DeepSeek-AI et al., 2026), GLM-5.3 Flash (Z.ai, 2026), Gemini 3.1 Pro Preview (Google DeepMind, 2026a), and Qwen3.5-397B-A17B (Qwen Team, 2026a); the DeepSeek snapshots are `0731` for Flash and `0813` for Pro.

6.1 GENERATION

The gap between models is not equally measurable on every task. Broken down by subject and canvas size, the 75 generation tasks separate the models most on large canvases and concrete subjects, through three quantities that move together (Table 6 in Appendix G gives every slice).

We measure how much a task's twelve artworks differ by the standard deviation of their judge scores, and a large canvas leaves room for form, depth, and detail, so a strong and a weak artwork come apart: that spread is 11.1 on canvases above 64 pixels and 10.9 on characters, against 7.5 on canvases of 16 pixels and below and 7.1 on materials. Rater agreement follows the same slices, 0.595 on the largest canvases and 0.589 on scenes against 0.403 on the smallest and 0.194 on materials, and task by task, the more a task's artworks differ, the more its raters agree, at Spearman $\rho = 0.37$ ($p = 9.3 \times 10^{-4}$). The twelve configurations' mean human ranks then spread further on exactly

those slices, at 2.72 on scenes and 2.51 on the largest canvases against 1.70 and 1.58 on the smallest canvases and materials, and judge–human agreement is higher there too, 0.685 and 0.652 against 0.278 and 0.147. This spread is not independent of rater agreement, since ranks averaged over disagreeing raters drift toward the middle.

The slices that separate the models are also the hardest. Characters and creatures carry the strongest structural constraints in the medium: viewers hold extremely sensitive priors about proportion, pose, and face, and these remain visibly wrong even in the artworks raters rank first. GPT-6 Astra's `sky-pirate-girl`, ranked first by all four raters, draws the head out of proportion and the hair implausibly, and its `mad-cyborg-scientist`, again the human first place, has poorly proportioned features and an expression and arm pose that do not fit together.

Small canvases are not easy. On a 16×16 canvas a single misplaced pixel changes a silhouette, and on the ten tasks of 16 pixels and below most artworks would need retouching before use. Raters then have little basis for ordering them, and the human first place goes to seven different configurations.

6.2 Editing

Leaderboard. On the 25 editing tasks of Table 2, GPT-6 Astra again ranks first, with a judge score of 84.68 and a mean human rank of 3.39. The ordering on editing is related to, but not the same as, the ordering on generation: across the 12 configurations, mean human rank on generation and on editing correlate at Spearman 0.776. Qwen3.8 Max and Claude Opus 5 rise to second and third on the editing human leaderboard (4.84 and 4.91), while GPT-5.6 Sol drops from fourth on generation to ninth (7.10).

Image observation. The `novision` configuration differs from the full GPT-6 Astra only in whether `view` returns an image (Section 4). On editing the full configuration leads on 17 of the 25 tasks against 6, and without vision the mean human rank falls from 3.39 to 5.00 (paired 95% interval $[+0.58, +2.63]$); on generation it leads on 37 of 75 against 29, at 3.04 versus 3.41 ($[-0.09, +0.87]$). Every pixel of a generated artwork comes from the model's own program, while an edit starts from an artwork someone else drew, and seeing it lets the model understand it better.

7 Limitations and Future Work

Every configuration in this paper creates in one environment, whose prompt carries no pixel-art technique guidance and no exemplar artwork. That minimality is what makes the leaderboard a comparison between models rather than between prompt designs, and it leaves the method itself unexplored: we do not vary the tools, the budget, or what the model is told about the medium. Those are the variables an agent designer would tune, and the environment we release is where they can be tuned. Teaching a model the craft of the medium, giving it references or critique to work against, or letting it revise for longer are all directions this benchmark can now measure, on the way to pixel art that enters a production workflow as it is.

8 Conclusion

We propose PixelBench, which formalizes pixel-art creation for language models as generation and editing under a specified canvas size and color ceiling, provides an agentic creation environment, and establishes an evaluation protocol grounded in complete human rankings of 1200 artworks by four raters. Under leave-one-out, the calibrated judge's agreement with the consensus order falls within the range the four raters span, which lets it carry continued comparison on this benchmark.

Reproducibility Statement

We will release the 100 tasks (Appendix B), all 1200 artworks, the creation transcripts of the editing tasks, the 400 human ranking lists, all judge responses, the repeat scores of Appendix D, the calibrated judge protocol with its full prompts (Appendix C), and the creation environment and agent skill (Section 4). Model APIs are not deterministic and human rankings cannot be rerun identically, so a new run will differ in detail; the released artworks, rankings, and judge responses let the results in every table and figure be recomputed exactly.

AI USE STATEMENT

In this work, we used generative AI tools for generating synthetic data (adding descriptive detail to the hand-written task statements, and rendering the reference images of Section 5.1) and implementing methods (the creation environment, the drawing API, and the evaluation scripts). Additionally, we used generative AI tools to draft and edit parts of the paper, write figure scripts, and identify related literature. We have reviewed all AI-assisted work: the authors checked every task statement, verified every citation against its source, and recomputed every reported number from the archived results. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Anthropic. Equipping agents for the real world with Agent Skills, October 2025. URL https://www.anthropic.com/engineering/equipping-agents-for-the-real-world-with-agent-skills. Accessed 2026-09-26.

Anthropic. Claude Opus 5 system card. https://www.anthropic.com/claude-opus-5-system-card, July 2026. Accessed 2026-09-23.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating llms by human preference, 2024. URL https://arxiv.org/abs/2403.04132.

Marco Cipriano, Leonardo Zini, Alexandra Schild, Valentin Teutschbein, Afsana Mimi, Marcella Cornia, Lorenzo Baraldi, and Gerard de Melo. Svg-score: Human-aligned evaluation of text-to-svg generation, 2026. URL https://arxiv.org/abs/2609.03806.

Flávio Coutinho and Luiz Chaimowicz. Generating pixel art character sprites using gans, 2022. URL https://arxiv.org/abs/2208.06413.

DeepSeek-AI et al. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Hugo Duprez. Sprite fusion pixel snapper. https://github.com/Hugo-Dz/spritefusion-pixel-snapper, 2025. Accessed 2026-09-20.

Emergent Mind. Pixel art bench. https://www.emergentmind.com/pixel-art-bench, 2026. Accessed 2026-09-11.

Timothy Gerstner, Doug DeCarlo, Marc Alexa, Adam Finkelstein, Yotam Gingold, and Andrew Nealen. Pixelated image abstraction. In *International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, 2012.

Dhruba Ghosh, Hanna Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Google DeepMind. Gemini 3.1 Pro model card. https://deepmind.google/models/model-cards/gemini-3-1-pro, February 2026a. Accessed 2026-09-23.

Google DeepMind. Gemini 3.8 Flash model card. https://deepmind.google/models/model-cards/gemini-3-8-flash, September 2026b. Accessed 2026-09-23.

Chu Han, Qiang Wen, Shengfeng He, Qianshu Zhu, Yinjie Tan, Guoqiang Han, and Tien-Tsin Wong. Deep unsupervised pixelization. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 37(6), 2018.

Kaiyi Huang, Chengqi Duan, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench++: An enhanced and comprehensive benchmark for compositional text-to-image generation, 2025. URL https://arxiv.org/abs/2307.06350.

HUMANBALANCE Ltd. Graphicsgale. https://graphicsgale.com/us/, 2026. Accessed 2026-09-24.

Igara Studio S.A. Aseprite: Animated sprite editor and pixel art tool. https://www.aseprite.org/, 2026. Accessed 2026-09-24.

Maurice G. Kendall. The treatment of ties in ranking problems. *Biometrika*, 33(3):239–251, 1945.

Hyunjun Kim and Sooyoung Ryu. Drawingbench: Evaluating spatial reasoning and ui interaction capabilities of large language models through mouse-based drawing tasks, 2025. URL https://arxiv.org/abs/2512.01174.

Kimi Team et al. Kimi k3: Open frontier intelligence, 2026. URL https://arxiv.org/abs/2607.24653.

Ashwin Nedungadi, Stefan Oehmcke, and Stefan Lüdtke. Autoregressive mosaics: Probing 2d spatial reasoning in text-only language models, 2026. URL https://arxiv.org/abs/2608.30751.

OpenAI. GPT-5.6 system card. https://deploymentsafety.openai.com/gpt-5-6, July 2026a. Accessed 2026-09-23.

OpenAI. GPT-6 Astra system card. https://deploymentsafety.openai.com/gpt-6-astra, 2026b. Accessed 2026-09-23.

OpenAI. GPT-Image-2 model. https://developers.openai.com/api/docs/models/gpt-image-2, 2026c. Accessed 2026-09-23.

Piskel. Piskel: Online pixel art tool. https://www.piskelapp.com/, 2026. Accessed 2026-09-24.

Pyxel Edit. Pyxel edit: Pixel art and tileset creation tool. https://pyxeledit.com/, 2026. Accessed 2026-09-24.

Qwen Team. Qwen3.5: Towards native multimodal agents. https://qwen.ai/blog?id=qwen3.5, February 2026a. Accessed 2026-09-23.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork. https://qwen.ai/blog?id=qwen3.8, August 2026b. Accessed 2026-09-23.

Joan Rodriguez, Haotian Zhang, Abhay Puri, Haoran Dai, Tianyang Zhang, Meng Lin, Rishav Pramanik, Xiaoqing Xie, Marco Terral Rodriguez, Darsh Kaushik, Aly Shariff, Perouz Taslakian, Spandana Gella, Sai Rajeswar, David Vazquez, Christopher Pal, and Marco Pedersoli. Vectorgym: A multi-task benchmark for svg code generation, sketching and editing, 2026. URL https://arxiv.org/abs/2603.29852.

Juan A. Rodriguez, Abhay Puri, Shubham Agarwal, Issam H. Laradji, Pau Rodriguez, Sai Rajeswar, David Vazquez, Christopher Pal, and Marco Pedersoli. Starvector: Generating scalable vector graphics code from images and text, 2025. URL https://arxiv.org/abs/2312.11556.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Akash Saravanan and Matthew Guzdial. Pixel vq-vaes for improved pixel art representation, 2022. URL https://arxiv.org/abs/2203.12130.

Daniel Silber. *Pixel Art for Game Developers*. A K Peters/CRC Press, 2015. ISBN 9781482252309. doi: 10.1201/b18786.

Zongwei Wu, Liangyu Chai, Nanxuan Zhao, Bailin Deng, Yongtuo Liu, Qiang Wen, Junle Wang, and Shengfeng He. Make your own sprites: Aliasing-aware and cell-controllable pixelization. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 41(6), 2022.

Yiying Yang, Wei Cheng, Sijin Chen, Xianfang Zeng, Fukun Yin, Jiaxu Zhang, Liao Wang, Gang Yu, Xingjun Ma, and Yu-Gang Jiang. Omnisvg: A unified scalable vector graphics generation model, 2025. URL https://arxiv.org/abs/2504.06263.

Z.ai. GLM-5.3-Flash: Frontier intelligence, flash cost. https://z.ai/blog/glm-5.3-flash, August 2026. Accessed 2026-09-23.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. URL https://arxiv.org/abs/2306.05685.

Leonardo Zini, Elia Frigieri, Sebastiano Aloscari, Marcello Generali, Lorenzo Dodi, Robert Dosen, and Lorenzo Baraldi. Svgauge: Towards human-aligned evaluation for svg generation, 2025. URL https://arxiv.org/abs/2509.07127.

A FULL GALLERY

Figures 3 to 8 show every artwork in the benchmark: the 75 text-to-image tasks grouped by subject with characters and creatures first, then the 25 editing tasks grouped into instruction editing, animation, and multiview. Each row is one task. The leftmost column is the reference: for a generation task the raw GPT-Image-2 rendering, and for an editing task the unedited source artwork. The twelve columns to its right are the twelve evaluated configurations in the order of Table 2, by mean judge score over all 100 tasks from best to worst. Editing tasks show only the delivered result, with the frames of a multi-frame result laid out left to right. Under each artwork, the first number is its score from the calibrated judge and the second its mean rank over the four raters for that task (1 is best, 12 is worst). Both badges are filled on the same red–yellow–green ramp, from worst to best, so a row can be read at a glance. The GPT-Image-2 renderings are shown for reference only and carry no score. Every artwork is drawn at its native resolution as solid vector rectangles rather than as a bitmap, so the blocks stay exact at any zoom level.

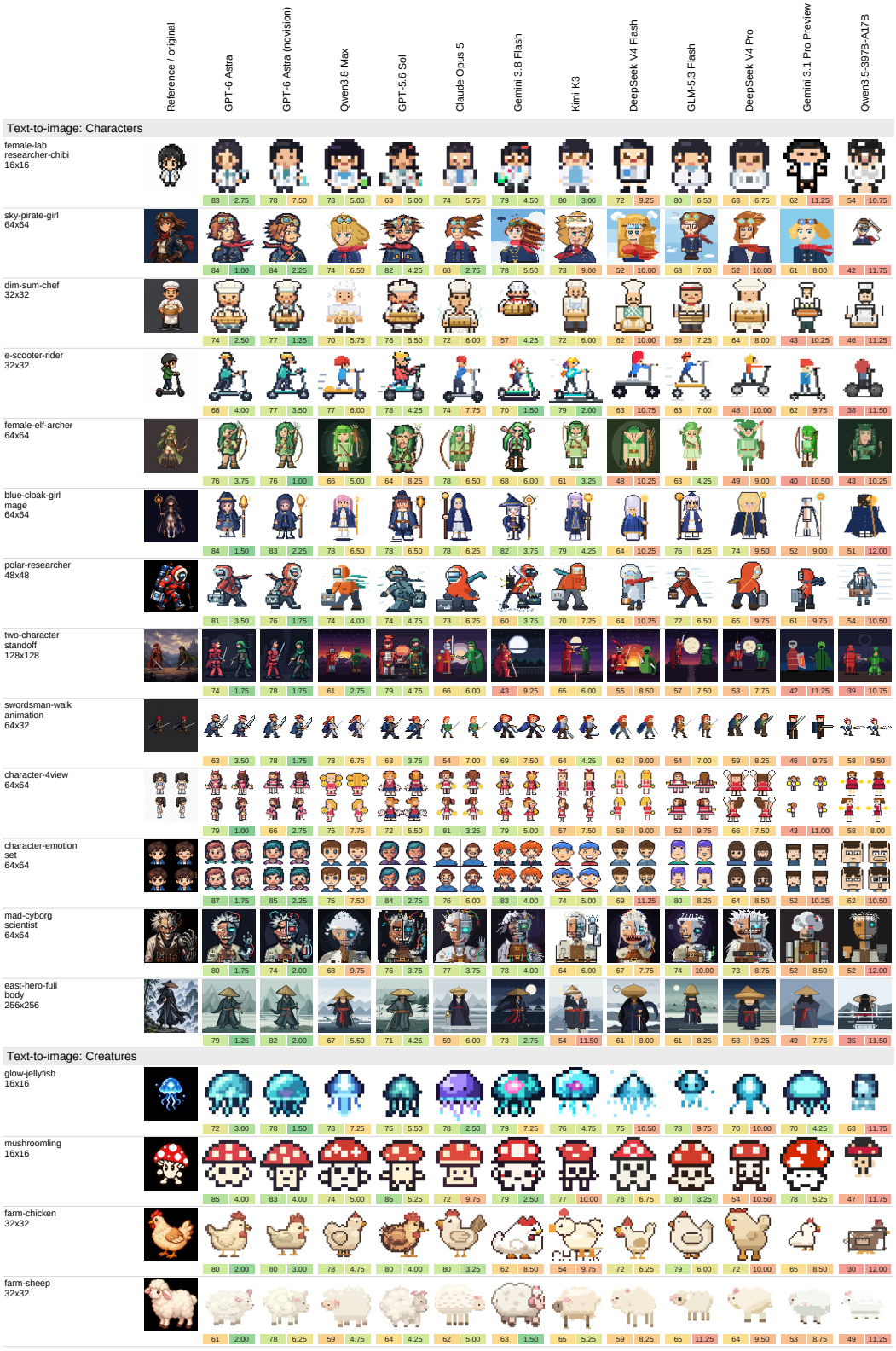


Figure 3: Full gallery, part 1 of 6.

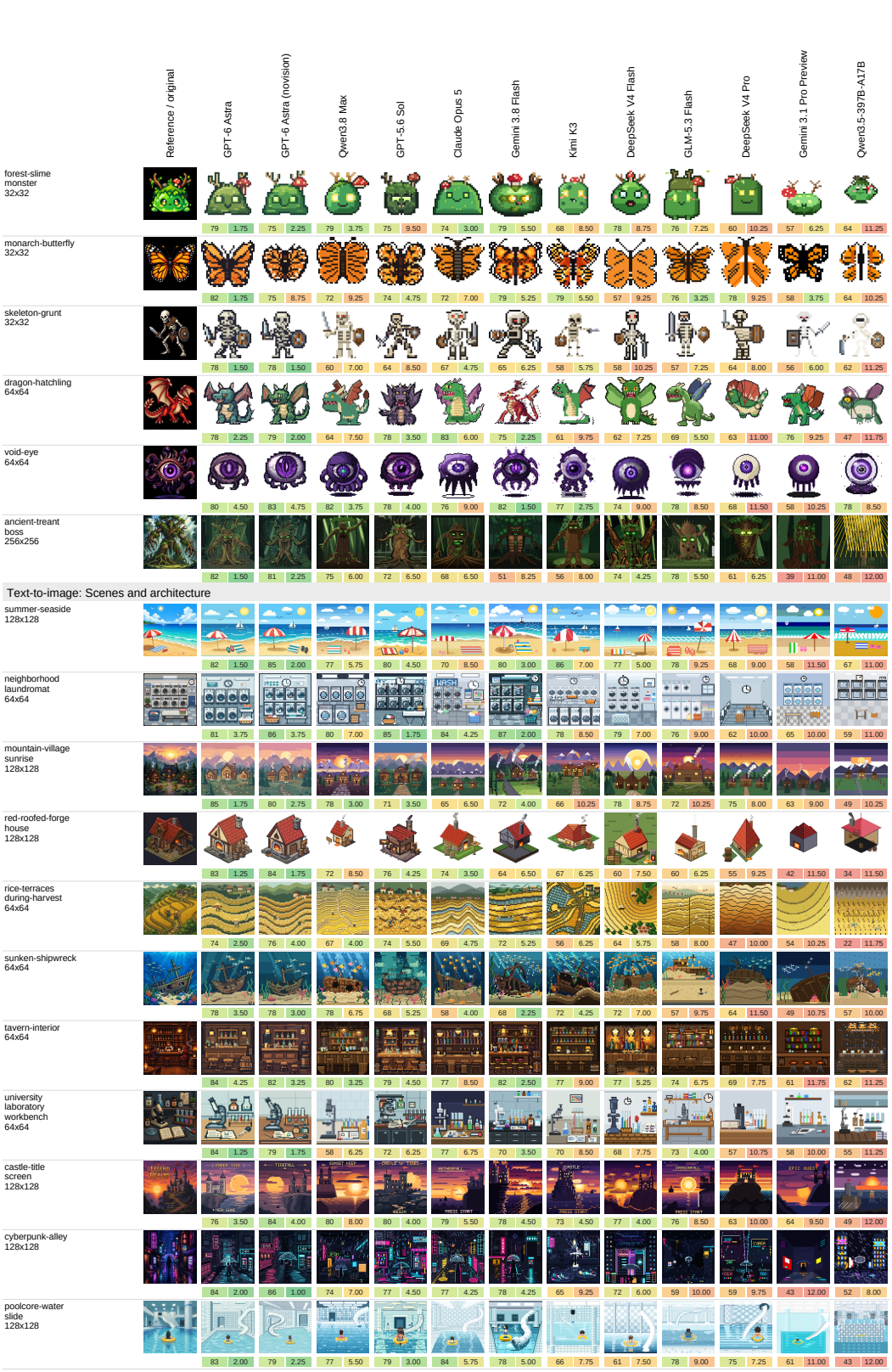


Figure 4: Full gallery, part 2 of 6.



Figure 5: Full gallery, part 3 of 6.



Figure 6: Full gallery, part 4 of 6.

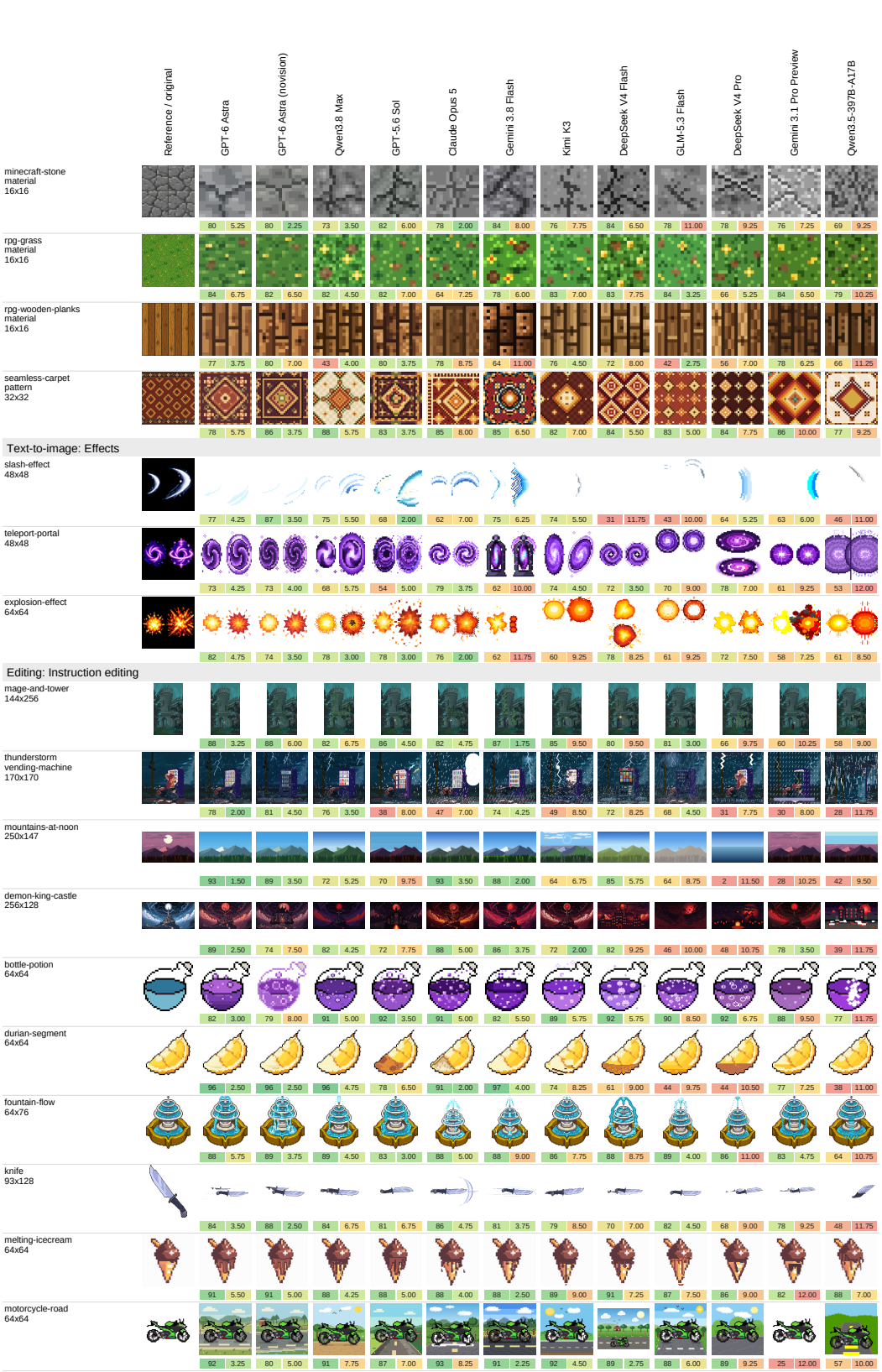


Figure 7: Full gallery, part 5 of 6.



Figure 8: Full gallery, part 6 of 6.

B TASKS

This appendix lists all 100 tasks in the order of the gallery in Appendix A: the text-to-image tasks grouped by subject, then the editing tasks grouped by type (instruction editing, animation, multi-view). Each text-to-image card gives the English task statement exactly as the model receives it, the native canvas, and the color ceiling. Each editing card adds the source artwork, enlarged with nearest-neighbor sampling, the frames selected from it, and a citation of the work it comes from. Editing tasks have no color ceiling, and the model may enlarge the canvas; all frames of a multi-frame result share one size.

Notice on source images. The editing tasks start from existing pixel art, and three of them from drawings by the authors. Each card below cites the work its task uses: the platform it was published on, the author where the source names one, and the exact files and frames taken. Each task takes at most three frames of a single work, and the originals appear here, and in the gallery of Appendix A, as citations: they document what each task asks for and what the submissions were asked to change, and they are reproduced for no other purpose. If you hold rights in one of these works and would prefer it not to appear, please contact us: we will replace the affected task with a similar new task and re-evaluate it.

B.1 TEXT-TO-IMAGE TASKS

Characters (13)

Female Lab Researcher Chibi

`female-lab-researcher-chibi`

Canvas 16×16 Colors ≤ 86

pixel art, 16x16 resolution, RPG game, two-heads-tall chibi, female, lab researcher, white lab coat, black hair, brown eyes

Sky Pirate Girl

`sky-pirate-girl`

Canvas 64×64 Colors ≤ 86

pixel art, 64x64 resolution, anime style, character portrait, young sky pirate girl, goggles, short navy coat, red scarf, three-quarter view, confident expression, wind-swept hair, clear silhouette

Dim Sum Chef

`dim-sum-chef`

Canvas 32×32 Colors ≤ 86

pixel art, 32x32 resolution, dim sum chef, front-facing character sprite, flour-dusted apron, holding bamboo steamer with buns, hands steamer food clothing clearly separated

Electric Scooter Rider

`e-scooter-rider`

Canvas 32×32 Colors ≤ 86

pixel art, 32x32 resolution, little person riding an electric scooter, side view riding right, helmet, e-scooter with standing deck and handlebars, readable riding pose

Female Elf Archer

`female-elf-archer`

Canvas 64×64 Colors ≤ 86

pixel art, 64x64 resolution, RPG character portrait, female elf archer, long green hair, pointed ears, light green outfit, bow and arrows on back, standing pose, three-quarter view

Blue-Cloak Girl Mage

`blue-cloak-girl-mage`

Canvas 64×64 Colors ≤ 86

pixel art, 64x64 resolution, anime style, full-body character, girl mage, relaxed pose, dark blue cloak, white boots, glowing amber staff, strong silhouette, simple readable costume, front-facing, RPG character sprite

Polar Researcher

`polar-researcher`

Canvas 48×48 Colors ≤ 86

pixel art, 48x48 resolution, polar researcher character sprite, transparent background, heavy cold-weather clothing, carrying sample case, pale blue goggle reflections, forward-leaning pose suggesting strong wind

Two-Character Standoff

`two-character-standoff`

Canvas 128×128 Colors ≤ 86

pixel art, 128x128 resolution, anime style, two rival characters, dramatic standoff, red-armored knight left, green-cloaked rogue right, clear gap between them, readable silhouettes

Swordsman Walk Animation

`swordsman-walk-animation`

Canvas 64×32 (two 32×32 frames) Colors ≤ 86

pixel art, 64x32 resolution, 16-bit RPG style, red-haired swordsman, walk animation, 2-frame sequence, side view facing right, long sword, dark blue cape, smooth natural motion, clear frame transitions, single frame 32x32

Character Four Views

`character-4view`

Canvas 64×64 Colors ≤ 86

pixel art, 64x64 resolution, same character in four views, 2x2 grid, top row front and back, bottom row left and right side, same character and outfit in all four views, girl, cheerleader, dark outline, transparent background

Character Emotion Set

character-emotion-set

Canvas 64×64 Colors ≤ 86

pixel art, 64x64 resolution, big head portraits of the same character in four emotions, 2x2 grid, calm, smiling, angry, surprised, same character with same hair and outfit in all four, expressive and clearly readable, dark outline, transparent background

Mad Cyborg Scientist

mad-cyborg-scientist

Canvas 64×64 Colors ≤ 86

pixel art, 64x64 resolution, RPG character portrait, mad cyborg scientist, left side human face, right side metal robotic head, one glowing cybernetic eye, exposed metal robotic arm, prosthetic hand with cables, messy white hair, brain electrode plugs, dirty lab coat, leather apron, tool belt, maniacal grin, strong light contrast

Eastern Hero Full Body

east-hero-full-body

Canvas 256×256 Colors ≤ 120

pixel art, 256x256 resolution, full body character portrait, oriental wuxia swordsman, wide bamboo conical hat (douli) tilted down covering the upper half of the face, long flowing dark robe with sash, loose hair, one hand resting on a straight jian sword, misty mountain backdrop, three-quarter view, heroic stance, flowing fabric details, dark outline

Creatures (10)

Glow Jellyfish

glow-jellyfish

Canvas 16×16 Colors ≤ 86

pixel art, 16x16 resolution, RPG enemy, small glowing jellyfish in deep water, translucent glowing bell, short tentacles, dark outline, transparent background

Mushroomling

mushroomling

Canvas 16×16 Colors ≤ 4

pixel art, 16x16 resolution, at most 4 colors, RPG enemy, small mushroom monster, red cap with white spots, round eyes, stubby arms and legs standing, dark outline, transparent background

Chicken (Farm Style)

farm-chicken

Canvas 32×32 Colors ≤ 86

pixel art, 32x32 resolution, Stardew Valley style, cozy farm animal, a plump chicken, gentle side view, soft rounded shapes, warm muted palette, transparent background

Sheep (Farm Style)

farm-sheep

Canvas 32×32 Colors ≤ 86

pixel art, 32x32 resolution, Stardew Valley style, cozy farm animal, fluffy white sheep with pale pink ears, side view, soft rounded shapes, gentle flat shading, no harsh outline, warm muted palette, transparent background

Forest Slime Monster

forest-slime-monster

Canvas 32×32 Colors ≤ 86

pixel art, 32x32 resolution, cute forest slime monster, moss, tiny antlers, glowing eyes, small red mushroom on back, front-facing, RPG enemy sprite, compact silhouette, readable expression

Monarch Butterfly

monarch-butterfly

Canvas 32×32 Colors ≤ 86

pixel art, 32x32 resolution, monarch butterfly, transparent background

Skeleton Grunt

skeleton-grunt

Canvas 32×32 Colors ≤ 86

pixel art, 32x32 resolution, RPG enemy, skeleton grunt, short sword and small wooden shield, slightly hunched pose, dark joint shading, transparent background

Dragon Hatchling

dragon-hatchling

Canvas 64×64 Colors ≤ 86

pixel art, 64x64 resolution, RPG enemy, young western dragon hatchling, half-spread wings, scaled body with pale belly, small horns, spiked tail, low growl with open mouth, strong silhouette, transparent background

Void Eye

void-eye

Canvas 64×64 Colors ≤ 86

pixel art, 64x64 resolution, RPG boss, giant floating single-eye monster, deep purple iris, faintly glowing pupil, short tentacles, hovering with soft shadow, ominous mood, strong contrast, readable silhouette, transparent background

Ancient Treant Boss

ancient-treant-boss

Canvas 256×256 Colors ≤ 120

pixel art, 256x256 resolution, boss creature, massive ancient treant, towering tree-like humanoid made of gnarled bark, glowing green eyes in the trunk, moss and hanging vines, branch arms with claw-like twigs, roots rising from the ground, forest clearing with god rays, menacing towering pose, highly detailed, dark outline

Scenes and architecture (16)

Summer Seaside

`summer-seaside`

Canvas 128×128 Colors ≤ 86

pixel art, 128x128 resolution, summer seaside beach scene, clear blue sky with soft white clouds, bright sun, sea layered from teal near shore to deep blue at the horizon, white foam waves rolling onto golden sand, red-and-white striped beach umbrella, striped towel and flip-flops, small white sailboat far out, two gulls flying, bright fresh summer mood, strong daylight

Neighborhood Laundromat

`neighborhood-laundromat`

Canvas 64×64 Colors ≤ 86

pixel art, 64x64 resolution, small self-service laundromat interior, two rows of front-loading washing machines, laundry basket, folding table, wall clock, cool white lighting, individual machines visually separated

Mountain Village at Sunrise

`mountain-village-sunrise`

Canvas 128×128 Colors ≤ 86

pixel art, 128x128 resolution, mountain village at sunrise, three wooden houses, chimney smoke, stone path, pine trees, distant mountains, layered depth, clear focal point center

Red-Roofed Forge House

`red-roofed-forge-house`

Canvas 128×128 Colors ≤ 86

pixel art, 128x128 resolution, isometric building, red tiled roof, furnace glow, wooden sign, stacked barrels, small chimney, clean 45-degree geometry, no perspective distortion, strategy game tileset, RTS game style

Rice Terraces During Harvest

`rice-terraces-during-harvest`

Canvas 64×64 Colors ≤ 86

pixel art, 64x64 resolution, top-down golden rice terraces following mountain contours, harvest season, two farmers on narrow ridges, simple village houses in distance, layered values for depth

Sunken Shipwreck

`sunken-shipwreck`

Canvas 64×64 Colors ≤ 86

pixel art, 64x64 resolution, underwater scene, sunken shipwreck, half-buried wooden hull in sand, corroded ship body, school of fish swimming past, coral and seaweed accents

Tavern Interior

`tavern-interior`

Canvas 64×64 Colors ≤ 86

pixel art, 64x64 resolution, cozy indoor tavern, wooden bar counter, shelf of bottles behind the bar, bar stools, warm yellow hanging lamps, glasses and wooden barrels, smoky atmosphere, warm nighttime lighting, dark outline

University Laboratory Workbench

`university-laboratory-workbench`

Canvas 64×64 Colors ≤ 86

pixel art, 64x64 resolution, laboratory workbench, microscope, test-tube rack, dropper, labeled bottles, open laboratory notebook, believable relative sizes, distinguish objects by shape rather than unreadable tiny text

Castle Title Screen

`castle-title-screen`

Canvas 128×128 Colors ≤ 86

pixel art, 128x128 resolution, fantasy RPG title screen, castle on seaside cliff, sunset, two towers with flags, large orange sun, layered clouds, dark rocks foreground, wide cinematic composition

Cyberpunk Alley

`cyberpunk-alley`

Canvas 128×128 Colors ≤ 86

pixel art, 128x128 resolution, cyberpunk alley, rainy night, neon shop signs, reflective puddles, power cables, vending machines, one person with transparent umbrella, strong light contrast, readable silhouettes

Poolcore Water Slide

`poolcore-water-slide`

Canvas 128×128 Colors ≤ 86

pixel art, 128x128 resolution, poolcore liminal space, backrooms-style indoor pool hall, vast empty indoor pool room, crystal clear pale-blue water with gentle ripples and reflections, a curving white water slide descending into the water, floor and walls covered in pale-blue-and-white grid mosaic tiles, tiled columns, bright fluorescent ceiling lights casting even cool light, a girl seen from behind floating in a yellow swim ring in the middle of the pool, quiet dreamy uncanny atmosphere

Convenience Store on a Rainy Night

`convenience-store-on-a-rainy-night`

Canvas 64×64 Colors ≤ 86

pixel art, 64x64 resolution, small convenience store on rainy night, bright interior light through glass windows, umbrella rack, parked bicycle, reflective puddles, person in raincoat entering, no real brand names or logos

Subway Platform

`subway-platform`

Canvas 64×64 Colors ≤ 86

pixel art, 64x64 resolution, subway platform side view, train arriving, platform screen doors, bench, station signs without brand names, three passengers with different poses, horizontal composition conveying speed and depth, readable silhouettes

Magic Forest Cottage

`magic-forest-cottage`

Canvas 128×128 Colors ≤ 86

pixel art, 128x128 resolution, magic forest cottage, deep purple night sky, full moon, mist, old wood-and-stone cottage, steep mossy roof, warm orange window light, stone well, wooden winch, stone-slab path, twisted ancient trees, dark red brown purple leaves, fairy-tale mood, warm-cold contrast

Offshore Wind Turbine Maintenance

`offshore-wind-turbine-maintenance`

Canvas 64×64 Colors ≤ 86

pixel art, 64x64 resolution, offshore wind turbine maintenance platform, technician in safety equipment climbing ladder, two additional turbines in distance, believable human scale, clearly separate steel structure ocean waves sky

Ancient Dragon Battle

`ancient-dragon-battle`

Canvas 256×256 Colors ≤ 140

pixel art, 256x256 resolution, epic fantasy battlefield scene, ancient dragon flying above a burning castle, knights with spears and shields on ramparts, arrows flying, smoke and embers, dramatic sunset sky, siege towers, soldiers in formation, distant mountains, rich detail, cinematic composition, dark outline

Objects (21)

Magnifying Glass

`magnifying-glass`

Canvas 16×16 Colors ≤ 86

pixel art, 16x16 resolution, detective game, magnifying glass icon

Gold Star Coin

`gold-star-coin`

Canvas 16×16 Colors ≤ 6

pixel art, 16x16 resolution, at most 6 colors, gold coin, star symbol in the middle

Wooden Key

`wooden-key`

Canvas 16×16 Colors ≤ 6

pixel art, 16x16 resolution, at most 6 colors, detective game, wooden key

Leather Spellbook

`leather-spellbook`

Canvas 32×32 Colors ≤ 8

pixel art, 32x32 resolution, at most 8 colors, RPG item portrait, book, closed spellbook, leather cover, metal corners, spine, dark outline, transparent background

USB Flash Drive

`usb-flash-drive`

Canvas 32×32 Colors ≤ 86

pixel art, 32x32 resolution, RPG item portrait, USB flash drive, metal casing, USB connector, keychain strap, dark outline, transparent background

Retro Film Camera

`retro-film-camera`

Canvas 32×32 Colors ≤ 86

pixel art, 32x32 resolution, RPG item portrait, camera, retro film camera, lens, shutter button, leather texture, dark outline, transparent background

Green Pear

`green-pear`

Canvas 32×32 Colors ≤ 7

pixel art, 32x32 resolution, at most 7 colors, green pear

Bamboo Steamer

`bamboo-steamer`

Canvas 32×32 Colors ≤ 86

pixel art, 32x32 resolution, bamboo steamer, transparent background, lid partially open, three clearly shaped dumplings inside, small rising steam

Laboratory Microscope Icon

`laboratory-microscope-icon`

Canvas 32×32 Colors ≤ 86

pixel art, 32x32 resolution, laboratory microscope icon, transparent background, three-quarter view, recognizable eyepiece body tube stage objective base

Portable Cassette Player

`portable-cassette-player`

Canvas 32×32 Colors ≤ 86

pixel art, 32x32 resolution, portable cassette player, front-facing icon, transparent background, transparent cassette window two tape reels, playback controls volume control headphone cable

Chainmail Armor

`chainmail-armor`

Canvas 32×32 Colors ≤ 86

pixel art, 32x32 resolution, RPG armor, chainmail, interlocking metal rings, dark outline, transparent background

Magic Staff

`magic-staff`

Canvas 48×48 Colors ≤ 86

pixel art, 48x48 resolution, RPG wooden magic staff, curved head, purple particle effects, transparent background

City Bus

`city-bus`

Canvas 64×32 Colors ≤ 86

pixel art, 64x32 resolution, modern city bus, side-view sprite, transparent background, front rear doors, row of windows, two wheels, headlights, simplified electronic route display, no real company branding

Vintage Sewing Machine

`vintage-sewing-machine`

Canvas 48×48 Colors ≤ 86

pixel art, 48x48 resolution, mechanical sewing machine, transparent background, slightly angled side view, handwheel needle bar presser foot base red fabric

Family Dining Table

`dining-table`

Canvas 64×64 Colors ≤ 86

pixel art, 64x64 resolution, furniture item sprite, family dining table, wooden table, four chairs around it, a bowl of fruit on the table, cozy family atmosphere, top-down or front view, dark outline

Fast Food Combo

`fast-food-combo`

Canvas 64×64 Colors ≤ 86

pixel art, 64x64 resolution, 45 degree top-down view of a fast food combo meal on a tray, cheeseburger with sesame bun lettuce and melted cheese, golden french fries in a red carton, crispy chicken nuggets, cup of cola with ice and straw, dipping sauce cups, appetizing food styling, clean details, dark outline

Weapon Set Icons

`weapon-set-icons`

Canvas 64×64 Colors ≤ 86

pixel art, 64x64 resolution, weapon icon set, 2x2 grid, sword, bow, shield, battle axe, four weapons in one consistent style, dark outline, transparent background

Realistic Heart

`realistic-heart`

Canvas 64×64 Colors ≤ 86

pixel art, 64x64 resolution, realistic style, heart

Motorcycle

`motorcycle`

Canvas 128×128 Colors ≤ 86

pixel art, 128x128 resolution, sports motorcycle, side profile, facing right, streamlined, black and dark gray body, silver gray metal parts, red accents, tilted small black windscreen, wide dark gray fairing, silver front fork, thick black tires, silver tubular frame, round silver gray engine cover, upswept tail, red tail light, gray hub white spokes, transparent background

Open Pocket Watch

`open-pocket-watch`

Canvas 64×64 Colors ≤ 86

pixel art, 64x64 resolution, realistic style, open mechanical pocket watch, lid open, intricate gear train clearly visible, golden metal movement, meshing gears, dial with tick marks, hands, watch chain, macro view, metallic texture and reflections, strong light contrast

Giant Arcane Orrery

`arcane-instrument-giant`

Canvas 256×256 Colors ≤ 140

pixel art, 256x256 resolution, huge complex arcane instrument, massive brass astrolabe-orrery with concentric engraved rings, orbiting crystal spheres, glowing runes along the rings, central radiant core, intricate gears and alchemical dials, floating over a stone pedestal in a dark hall, extremely detailed machinery, dramatic lighting, dark outline

Icons and interface (7)

RTS Factory Icon

`rts-factory-icon`

Canvas 32×32 Colors ≤ 86

pixel art, 32x32 resolution, RTS game, factory building icon, chimney, flat roof, brick walls, vents, clean silhouette, centered composition, strategy game tileset

Battery Status Set

`battery-status-set`

Canvas 64×64 Colors ≤ 86

pixel art, 64x64 resolution, four battery icons in one horizontal row, 100% charge 60% 20% active charging, identical battery outlines, clearly distinguish four fill states

Compass

`item-icon`

Canvas 32×32 Colors ≤ 86

pixel art, 32x32 resolution, inventory icon, ancient golden compass, blue crystal center, top-down view, thick dark outline, centered composition, game UI icon

Typography

`typography`

Canvas 64×64 Colors ≤ 86

Create a pixel-art game title screen featuring the words “PIXEL QUEST”. Design clear, readable pixel lettering, using letter size, spacing, and arrangement to establish a strong visual hierarchy. You may add outlines, shadows, or small decorative elements, while keeping the lettering as the main focus.

Chart Layout

`chart-layout`

Canvas 128×64 Colors ≤ 86

pixel art, 128x64 resolution, bar chart, fruit data, apple 10, banana 5, orange 7, lemon 14, grape 2, color bars by fruit, mini pixel art icons beside chart instead of text

Skill Icon Set

`skill-icon-set`

Canvas 64×64 Colors ≤ 86

pixel art, 64x64 resolution, skill icon set, 2x2 grid, fireball, ice shield, healing, lightning, each icon 32x32, four skills in one consistent style, dark outline, transparent background

Public Transportation Icon Set

`public-transportation-icon-set`

Canvas 64×64 Colors ≤ 86

pixel art, 64x64 resolution, consistent 2x2 grid of public transportation icons, bus, subway, bicycle, pedestrian, consistent framing icon scale line weight palette background

Materials (5)

Dirt Material (Minecraft Style)

`minecraft-dirt-material`

Canvas 16×16 Colors ≤ 10

pixel art, 16x16 resolution, at most 10 colors, game material texture, Minecraft style, dirt block, seamless tile, granular dirt texture

Stone Material (Minecraft Style)

`minecraft-stone-material`

Canvas 16×16 Colors ≤ 10

pixel art, 16x16 resolution, at most 10 colors, game material texture, Minecraft style, stone block, seamless tile, cracked stone texture

Grass Material (RPG Style)

`rpg-grass-material`

Canvas 16×16 Colors ≤ 86

pixel art, 16x16 resolution, RPG game style, grass material tile, seamless tiling, interleaved dark and light green grass speckles, a few tiny yellow flowers, small darker grass tufts, sparse bare-dirt patches

Wooden Planks Material (RPG Style)

`rpg-wooden-planks-material`

Canvas 16×16 Colors ≤ 86

pixel art, 16x16 resolution, RPG game style, wooden planks material tile, seamless tiling, several vertical boards side by side, natural wood-grain lines, dark seam grooves between planks, alternating light and dark board ends, a few wood knots

Seamless Carpet Pattern

`seamless-carpet-pattern`

Canvas 32×32 Colors ≤ 86

pixel art, 32x32 resolution, seamless carpet pattern, warm geometric motifs, diamonds with floral border, woven texture, seamless tiling in all directions, left right edges connect exactly

Effects (3)

Slash Effect

`slash-effect`

Canvas 48×48 Colors ≤ 86

pixel art, 48x48 resolution, slash effect, 2 frames, frame 1 a small white arc slash, frame 2 a larger arc slash with a trailing streak, transparent background

Teleport Portal

`teleport-portal`

Canvas 48×48 Colors ≤ 86

pixel art, 48x48 resolution, teleport portal, 2 frames, frame 1 a purple energy vortex, frame 2 the vortex rotated with a glowing halo, transparent background

Explosion Effect

`explosion-effect`

Canvas 64×64 Colors ≤ 86

pixel art, 64x64 resolution, explosion effect, 2 frames, frame 1 a bright yellow fireball at center, frame 2 an expanding orange-red blast with sparks, transparent background

B.2 EDITING TASKS

Instruction editing (13)

Mage and Tower



`mage-and-tower`

Instruction editing, single frame Input 144×256

Let a layer of vines cover part of the tower's walls. Add a cloaked mage holding a glowing staff at the archway entrance of the tower; he is walking into the tower (facing away from the viewer, heading through the archway).

Source. Pixiv, artwork 123659418.

Vending Machine in a Thunderstorm



`thunderstorm-vending-machine`

Instruction editing, single frame Input 170×170

Remove the character from the scene (and fill in the parts it was covering, such as the vending machine). Change the weather to a thunderstorm: make the ground wet with clinging water droplets, and render the rain with semi-transparent droplets. Add lightning in the background.

Source. Pixiv, artwork 123439931.

Mountains at Noon



`mountains-at-noon`

Instruction editing, single frame Input 250×147

Set the time to noon on a clear day. Note: because it is noon, the sun should not appear in the picture.

Source. OpenGameArt, "Mountain at Dusk Background" by ansimuz.

Demon King's Castle



`demon-king-castle`

Instruction editing, single frame Input 256×128

Edit the pixel art based on the original image: Change the theme of the whole picture: turn the "City of Hope" into the Demon King's castle beneath a hanging blood moon.

Source. Pixiv, artwork 123985822.

Potion Bottle



`bottle-potion`

Instruction editing, single frame Input 64×64

Turn the water in the bottle into a bubbling purple magic potion.

Source. Drawn by the authors.

Durian



`durian-segment`

Instruction editing, single frame Input 64×64

Remove the bottom segment of durian flesh so the rind underneath is exposed; keep the other two pieces unchanged.

Source. Pixel Beads Art (`pixel-beads.com`), durian bead pattern.

Fountain Water Flow



`fountain-flow`

Instruction editing, single frame Input 64×76

The fountain is currently in a waterless state. Change it into a fountain that has water while it is running.

Source. OpenGameArt.

Knife



`knife`

Instruction editing, single frame Input 93×128

Pose this knife thrusting to the right with its cutting edge facing downward, leaving semi-transparent afterimages and a white slash effect.

Source. OpenGameArt, "Knife" by kotnaszynce; frame 1 of `nurz.gif`.

Melting Ice Cream



`melting-icecream`

Instruction editing, single frame Input 64×64

Make the ice cream's chocolate melt and drip down the cone.

Source. Drawn by the authors.

Motorcycle on a Country Road



`motorcycle-road`

Instruction editing, single frame Input 64×64

Put the motorcycle onto a country road.

Source. Pixel Beads Art (`pixel-beads.com`), motorcycle bead pattern.

Snow on the Pine



`snowy-pine`

Instruction editing, single frame Input 82×135

Cover the pine tree with a layer of snow.

Source. OpenGameArt, "Pine Tree Pack" by KnoblePersona.

Sword into Staff



`sword-to-staff`

Instruction editing, single frame Input 64×64

Change the sword in the adventurer's hand into a staff.

Source. Drawn by the authors.

Yellow Bird on the Right



`yellow-bird-right`

Instruction editing, single frame Input 64×64

Change the right-hand one of the two parrots into a yellow parrot.

Source. Pixel Beads Art (`pixel-beads.com`), parrot bead pattern by cglow.

Animation (10)

Bat Flapping Wings



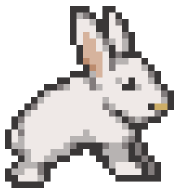
`bat-flapping-wings`

Animation, 2 frames Input 27×22

2 frames. Frame 1 keeps the original picture content (the canvas may be changed); frame 2 shows the bat flapping its wings: the drooping wings open up into a spread, raised position.

Source. OpenGameArt.

Bunny Hopping



`bunny-hop`

Animation, 2 frames Input 26×28

2 frames. Frame 1 keeps the original picture content (the canvas may be changed); frame 2 shows the rabbit hopping forward, caught in a mid-leap pose.

Source. OpenGameArt.

Demon Cyclops Hit



`demon-cyclop-hit`

Animation, 3 frames Input 37×33

3 frames. Frame 1 keeps the original picture content (the canvas may be changed); frame 2 shows the monster being hit: the foot on the left of the image (the monster's right foot) lifts off the ground, the whole body leans back, its single eye rolls upward and the mouth opens - the recoil pose; frame 3 shows the blow landing, with a red filter added over the whole monster to show the impact.

Source. Ninja Adventure Asset Pack by Pixel-boy and AAA (itch.io), Actor/Boss/DemonCyclop2; frames 1, 4, 2 of `HitPreview.gif`.

Draw Sword Slash



`draw-sword-slash`

Animation, 2 frames Input 62×72

2 frames. Frame 1 keeps the original picture content (the canvas may be changed); frame 2 shows the completed drawn-sword slash. This character is left-handed: the sword is held in the LEFT hand (the hand gripping the blade in the image) and the right hand does not hold it. The slash sweeps from the lower left up to the upper right - a left-to-right motion - leaving a black slash trail. The motion ends on the completed slash.

Source. CraftPix free asset.

Elf Archer Attack



`elf-archer-attack`

Animation, 2 frames Input 35×47

2 frames. Frame 1 keeps the original picture content (the canvas may be changed); frame 2 shows the elf archer drawing the bow to its maximum - the bow fully open with the whole pose shifted into a proper drawn-bow stance. No arrow.

Source. Battle for Wesnoth core unit sprites, `sharpshooter-bow` attack animation, frames 1 and 4.

Fist Swing



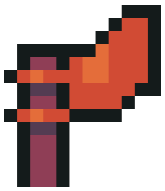
`fist-swing`

Animation, 2 frames Input 29×44

2 frames. Frame 1 keeps the original picture content (the canvas may be changed); frame 2 shows the character throwing a light punch with the left fist (the arm that extends horizontally to the right of the image).

Source. OpenGameArt.

Flag Waving



`flag-wave`

Animation, 3 frames Input 48×56

3 frames. Frame 1 keeps the original picture content (the canvas may be changed); frames 2 and 3 continue the looping flag-waving animation (the fly end swings from its highest to its lowest point).

Source. Ninja Adventure Asset Pack by Pixel-boy and AAA (itch.io), Backgrounds/Animated/Flag; frames 1–3 of `Preview.gif`.

Paladin Attack



`paladin-attack`

Animation, 2 frames Input 43×61

2 frames. Frame 1 keeps the original picture content (the canvas may be changed); frame 2 shows the paladin swinging the weapon in hand down in a heavy attack. A white motion trail streams out behind the weapon. The footing changes as well: as the weapon comes down, the left foot also steps forward to put weight behind the blow.

Source. Battle for Wesnoth core unit sprites, `royal-warrior` attack animation, frames 1 and 2.

Release Flames



`release-flames`

Animation, 2 frames Input 47×63

2 frames. Frame 1 keeps the original picture content (the canvas may be changed); frame 2 shows the character thrusting a fire spell forward: the flame flies forward with a fiery trail. The pose changes too - the left foot steps forward and the body leans in, pushing the flame out.

Source. CraftPix free asset.

Squid Red Idle



`squid-red-idle`

Animation, 3 frames Input 58×57

3 frames. Frame 1 keeps the original picture content (the canvas may be changed); frames 2 and 3 show the red squid monster idling, part of the sprite bobbing gently up and down for a breathing idle.

Source. Ninja Adventure Asset Pack by Pixel-boy and AAA (itch.io), Actor/Boss/SquidRed; frames 1–3 of `IdlePreview.gif`.

Multiview (2)

Bride Turnaround



`char-turn-3view`

Multiview, 3 frames Input 66×119

The canvas already contains a pixel-art character. Keep the character's design, outfit and palette exactly the same and change only the viewing angle, producing three frames in this order: frame 1 a three-quarter view with the character turned 45 degrees to its left, frame 2 the opposite three-quarter view with the character turned 45 degrees to its right, frame 3 the rear view (seen from directly behind). Submit all 3 frames in order with finish([f1, f2, f3]).

Source. Hugging Face dataset `evilsocket/alucard-sprites`, images 0000000–0000003 (input 0000002).

House Turnaround



`house-turn-2view`

Multiview, 2 frames Input 112×157

Rotate this house by 45 degrees and then by 90 degrees and draw both views, keeping the same 45-degree high-angle top-down view in both frames (a fixed elevated isometric camera: the camera height and pitch stay unchanged, so neither view becomes a flat eye-level elevation): the 45-degree frame shows the front door and front wall together with the window and wall on the right-hand side; the 90-degree frame shows only that right-hand wall and its window.

Source. Tiny Swords (Free Pack) by Pixel Frog (itch.io), Buildings/Yellow Buildings, `House1.png` (input), `House2.png` and `House3.png` (reference views).

C JUDGE PROMPTS

Every judge call is a single user message: a text block, followed by image slots, each preceded by a short caption. The text below is sent verbatim; braces mark fields filled from the task, and square brackets mark text that appears only under the stated condition. The {prompt} field is the task statement exactly as Appendix B gives it, the same text the model creating the artwork received. All four protocol versions of Section 5.3 share this text and differ only in what they switch on: the presentation step adds the transparent margin to every pixel-art image, the rubric step inserts the rubric before the scoring instruction, and the reference step adds the reference slot.

Development subset. The wording above was settled on 23 of the 75 generation tasks, before the runs of Table 1. The rubric was written against four of them, one per subject—`magnifying-glass`, `glow-jellyfish`, `e-scooter-rider` and `subway-platform`—and then fixed and confirmed on `character-emotion-set` and `sunken-shipwreck`. The other seventeen appear in short probes of the score range, of the reference slot and of the judge's reasoning effort: `ancient-dragon-battle`, `ancient-treant-boss`, `battery-status-set`, `cyberpunk-alley`, `dim-sum-chef`, `dragon-hatchling`, `east-hero-full-body`, `farm-sheep`, `female-lab-researcher-chibi`, `mad-cyborg-scientist`, `magic-forest-cottage`, `portable-cassette-player`, `retro-film-camera`, `sky-pirate-girl`, `tavern-interior`, `teleport-portal` and `two-character-standoff`. No editing task took part in any of them. Over 1000 random halves of the 100 tasks, the final protocol reaches higher agreement than the simplest protocol in every one.

Image preparation. The frames of a multi-frame result are packed into the most nearly square grid; a material tile is repeated 2×2; the presentation step then adds a transparent margin of $1/16$ of the width and height on the left and top; finally the image is scaled with nearest-neighbor sampling so that its longer side is 512 pixels, preserving the aspect ratio. The GPT-Image-2 reference, which is not on a pixel grid, is scaled with Lanczos resampling instead.

Generation. The text block is

```
Task brief: {prompt}
Canvas: {size}; max colors: {max_colors}
[materials:] The image repeats the same native tile in a 2x2 grid,
without gaps or flips, so that tile boundaries are visible. Regular
repetition is not a defect.
[rubric step on:] {subject rubric} [characters, creatures, objects,
scenes:] {size sentence}
Rate this single pixel art image from 0 to 100 for overall quality for
the task. Reply with only one number from 0 to 100.
```

followed by the image slots

```
[reference step on:] Image 1, reference image (calibration aid, not
the work being scored). The next image is a reference for this task,
generated by GPT-Image-2, a diffusion image model. It is not true
pixel art: it does not lie on the pixel grid, and it uses its own
resolution and proportions. Treat it only as one indication of what a
visually strong result for this task can look like. Do not score the
candidate on how closely it resembles the reference: the candidate's
overall style and design may legitimately differ from it. Do not
treat the reference as ground truth, and do not deliberately reward a
drawing style that resembles it. {image}
Candidate image (the work being scored): {image}
```

Subject rubrics.

Characters: Consider natural pose, an expressive silhouette and coherent anatomy. Favor harmonious colors, balanced shapes and a cohesive composition. The figure should read as a specific character: facial features clear and well proportioned, the hairstyle plausible, and the whole legible rather than cluttered.
Creatures: Consider an appealing subject, expressive features, and a plausible design of form and movement. Favor harmonious colors, balanced shapes and a cohesive composition.
Objects: Prioritize whether the object is clear and recognizable: as pixel art it should be identifiable at a glance. Consider a recognizable silhouette, clear parts and balanced placement; shading is optional. Credit intentional simplicity; extra colors, shading or detail help only when they clarify form.
Scenes: Prioritize whether the scene as a whole is clear and recognizable: spatial depth, coherent perspective, lighting, separation of foreground from background, and the visual appeal and decorative detail of the image. Credit intentional simplicity; extra colors, shading or detail help only when they clarify form.
Icons and interface: Prioritize instant recognition at native size, a clear and conventional metaphor, the visual appeal of the icon, and clean separation from the background. When the task asks for a set, consistency of style, alignment, visual weight and sizing across the set matters as much as any single element.
Materials: Judge material identity, continuity across the tile boundaries, and the absence of abrupt seams or value breaks. The material should look natural and should not draw undue attention to itself, while still being attractive and clear.
Effects: Prioritize the overall shape of the effect, a readable sense of motion, energy or impact, a clear focal point, and shapes that express the phenomenon rather than a static decoration. Color ramps and contrast should support the effect's intensity.
Size sentence, longer side ≤ 32: At this size, every pixel should be placed deliberately, so that it supports a clear and appealing form.
Size sentence, longer side > 32: Details should enrich the image while preserving clear forms.

Editing, animation, and multiview. The text block is

Editing instruction: {prompt}
Original canvas: {size}
The result may use a different canvas size or different margins from the original; that by itself is not a defect. [multi-frame:] {frame note}
[rubric step on:] Weigh three things together.
(1) Instruction compliance: does the result actually make the change the instruction asks for, on the right subject and to the right extent?
(2) Preservation: do the character, objects, scene, style and palette that the instruction does not ask to change stay consistent with the original?
(3) Craft: is the result clean and visually appealing pixel art?
[animation:] This is an action sequence, so also weigh:
(4) Inter-frame continuity: do consecutive frames read as one continuous motion, holding the subject's identity, scale and placement steady between frames?
(5) Pose plausibility: is the pose in each frame physically plausible for the described action, is the subject's movement and posture sound, and does the sequence express that action clearly?
[multiview:] This is a multi-view turnaround, so also weigh:
(4) Cross-view identity: do all views show the same character or object as the original, holding proportions, palette, costume and

```
distinctive features consistent across views, and is each view a
plausible projection of that same subject?
The first image is the original artwork.  The last image is the result
the model produced for this instruction.  Rate the result from 0 to
100 for overall quality on this editing task.  Reply with only one
number from 0 to 100.
```

followed by the image slots

```
Image 1, the original artwork:  {image}
[reference step on:]  Image 2, reference image (calibration aid, not
the work being scored).  {reference note} {image}
Image {n}, the model's result (the work being scored):  {image}
```

The frame note states the layout, for example "The result is a 3-frame sequence shown as a single image, with the frames in order left to right." When the task asks the first frame to reproduce the original, that frame is already image 1 and is not shown again; the note then reads "This task asks for {total} frames, of which the first reproduces the original. Only the remaining generated frame is shown." The reference note for a reference taken from the source asset is

```
The next image is a reference result for this task.  It is one valid
way to satisfy the instruction, not a target to copy, and a result
that differs from it can be equally good or better.  Use it only to
gauge what a reasonable outcome for this instruction looks like.  Do
not score the result on how closely it resembles the reference.
```

and for a GPT-Image-2 edit it is

```
The next image is a reference for this task, generated by GPT-Image-2,
a diffusion image model.  It is not true pixel art:  it changes the
proportions and resolution of the original, and it does not lie on the
pixel grid.  Treat it only as a rough hint of an attractive visual
outcome, not as a target to copy; a result that differs from it can be
equally good or better.  Do not score the result on how closely it
resembles the reference:  the result's overall style and design may
legitimately differ from it.  Do not treat the reference as ground
truth, and do not deliberately reward a drawing style that resembles
it.
```

D REPEAT SCORING

The repeat experiment of Section 5.2 draws 12 of the 100 tasks under a fixed seed, before any scoring: one per generation subject, two instruction edits, two animations, and one multiview. All 12 submissions of each task are kept, for 144 artworks, and each is scored three further times under the final protocol—`gpt-6-astra`, reasoning effort xhigh, a fresh context per call, request order shuffled within each pass. The prompt text and the image hashes of all three passes match those of the original run byte for byte, so the only thing that varies is the judge's own sampling. All 432 calls returned a valid score.

The mean score of the 144 artworks is 71.4, 72.1, and 71.1 in the three passes. Their three-pass mean reproduces the single-pass score of the same artworks to within 0.63 points on average. Per artwork, the standard deviation of the three scores averages 2.39 points and has a median of 1.73; 56.9% of repeat pairs agree within 2 points and 84.5% within 5. Where two artworks are 10 or more points apart, repeat scoring keeps their order 97.7% of the time. Table 3 breaks the spread down by task.

Table 3: Repeat scoring by task. Mean SD is the standard deviation of the three scores of one artwork, averaged over the task's 12 submissions. Repeat τ_b is the Kendall τ_b between the orders two passes give the 12 submissions, averaged over the three pairs of passes.

Task	Type	Mean SD	Repeat τ_b
`female-elf-archer`	Characters	2.93	0.810
`skeleton-grunt`	Creatures	2.81	0.424
`poolcore-water-slide`	Scenes and architecture	1.97	0.824
`wooden-key`	Objects	1.59	0.467
`battery-status-set`	Icons and interface	2.04	0.575
`rpg-grass-material`	Materials	1.70	0.372
`slash-effect`	Effects	2.56	0.832
`melting-icecream`	Instruction editing	1.48	0.699
`motorcycle-road`	Instruction editing	1.55	0.724
`release-flames`	Animation	2.37	0.832
`demon-cyclop-hit`	Animation	3.93	0.766
`char-turn-3view`	Multiview	3.79	0.859

Whether two artworks keep their order under repeat scoring depends on how far apart they are (Table 4). Over all pairs of submissions within a task and all pairs of passes, the order strictly reverses in 12.2% of comparisons; restricted to pairs at least 5 points apart in the earlier pass the rate is 5.6%, and at least 10 points apart, 2.3%.

Table 4: Order stability against the score gap. "Reversed" is the share of comparisons in which the later pass strictly reverses the earlier one; "tie changed" the share in which a tie becomes a strict order or the reverse.

Gap in the earlier pass	Comparisons	Reversed	Tie changed
≥ 0	2376	12.2%	10.0%
≥ 3	1824	9.0%	2.6%
≥ 5	1525	5.6%	1.6%
≥ 10	1062	2.3%	0.8%

To carry this to the full task set, we resample the per-artwork residuals of the three passes within each subject and type, in the proportions of the 100 tasks—75 generation tasks over seven subjects, 13 instruction edits, 10 animations, 2 multiviews—and rebuild each configuration's 100-task mean 200,000 times. This isolates the judge's sampling: the artworks and the task set are held fixed. Table 5 gives the resulting standard error per configuration and the 95% half-width of the difference between two independent evaluations. Under the same resampling, 61 of the 66 pairs of configurations keep the order of Table 2 in at least 99% of draws; the five that do not are separated by less than 0.6 points, and GPT-6 Astra's margin over every model family other than its own `novision` configuration, at least 4.99 points, holds in every draw. One pass per artwork is therefore the protocol we release, and a gap below one point is where a second pass earns its cost.

Table 5: Scoring standard error of a 100-task mean, by configuration, and the 95% half-width of the difference between two independent evaluations of the same configuration.

Configuration	SE of 100-task mean	95% half-width of a rerun difference
GPT-6 Astra	0.19	0.52
GPT-6 Astra `novision`	0.24	0.68
Qwen3.8 Max	0.22	0.61
GPT-5.6 Sol	0.43	1.20
Claude Opus 5	0.23	0.63
Gemini 3.8 Flash	0.26	0.73
Kimi K3	0.27	0.74
DeepSeek V4 Flash	0.29	0.79
GLM-5.3 Flash	0.37	1.02
DeepSeek V4 Pro	0.33	0.91
Gemini 3.1 Pro Preview	0.31	0.85
Qwen3.5-397B-A17B	0.30	0.84

E TEXT OBSERVATION

`view` is the only channel through which a model obtains an image observation (Section 4). Four of the evaluated models take no image input: DeepSeek V4 Flash, DeepSeek V4 Pro, GLM-5.3 Flash, and Qwen3.5-397B-A17B. The `novision` configuration of GPT-6 Astra switches vision off in the same way, and apart from `view` it shares tools, prompt, and budget with the full configuration, so the difference between them measures what image observation contributes to creation quality (Section 6.2).

In all of these cases `view` returns no image and the model reads the artwork as a character picture instead. `matrix_text()` prints the canvas, or a region of it, as a character grid: each line of text is one row of pixels and each character one pixel, the same color is always written with the same character (taken from `a–z`, `A–Z`, `0–9` by palette index), and transparent pixels are written as a dot. A legend after the grid maps each character to its color name, hex value, and palette index, and the model can also print the current palette to obtain the mapping between colors and characters. `stats_text()` gives the color count, coverage, bounding box, and most frequent colors. On large canvases with many colors, the model can print a single region, or shrink a selection before printing it.

F A SECOND JUDGE MODEL

The judge of Section 5.2 is GPT-6 Astra, which also leads both leaderboards. To see what the protocol returns when a different model answers, we scored all 1200 artworks a second time with Claude Opus 5 at medium reasoning effort. Every one of the 1200 calls carries the same prompt text and the same image hashes as the corresponding call in Section 5.3, so the two judges receive identical inputs; the Astra judge additionally sets reasoning effort to `xhigh`.

Ordering the twelve configurations by mean score puts GPT-6 Astra first, GPT-6 Astra `novision` second, Qwen3.8 Max third, and Claude Opus 5 itself fourth. The two judge leaderboards correlate at Spearman $\rho = 0.972$ by mean score and $\rho = 0.986$ by mean within-task rank, and five of the twelve configurations sit one place apart between them. Against the raters, this judge's leaderboard reaches $\rho = 0.951$, and it agrees with the consensus order at 0.5064 $[0.4680, 0.5437]$ over all 100 tasks, 0.5102 on generation and 0.4951 on editing, all under the convention of Section 5.2. Its mean scores sit about 25 points below the Astra judge's, so the two scales are not interchangeable and the order is the comparable quantity. This is one second judge, scored once: it shows what the protocol returns when the judge model is replaced, not what every judge model would return.

G GENERATION SLICES

Table 6 gives the full breakdown behind Section 6.1: the 75 generation tasks by subject and by canvas size, with the four quantities that section reads off them. Each slice is scored on the tasks it contains, and the task counts of Section 3 apply.

Table 6: Generation tasks by subject and canvas size (longer side). Score SD is the standard deviation of the judge scores of a task's twelve artworks, averaged over the slice; rater–rater is the pairwise agreement among the four raters; rank SD is the standard deviation of the twelve configurations' mean human ranks over the slice; judge–human is the calibrated judge against the raters' consensus.

Slice	Tasks	Score SD	Rater–rater τ_b	Rank SD	Judge–human τ_b
Scenes, architecture	16	10.6	0.589	2.72	0.685
Creatures	10	9.0	0.572	2.32	0.477
Characters	13	10.9	0.568	2.66	0.656
Objects	21	7.8	0.478	1.89	0.549
Icons, interface	7	8.3	0.459	2.24	0.503
Effects	3	10.8	0.441	2.34	0.495
Materials	5	7.1	0.194	1.58	0.147
Canvas > 64	14	11.1	0.595	2.51	0.652
Canvas 33–64	33	9.6	0.545	2.47	0.628
Canvas 17–32	18	8.0	0.429	1.87	0.494
Canvas ≤ 16	10	7.5	0.403	1.70	0.278