# CAN LLMS UNDERSTAND AND GENERATE IMAGES PIXEL BY PIXEL?

**Anonymous authors**
Paper under double-blind review

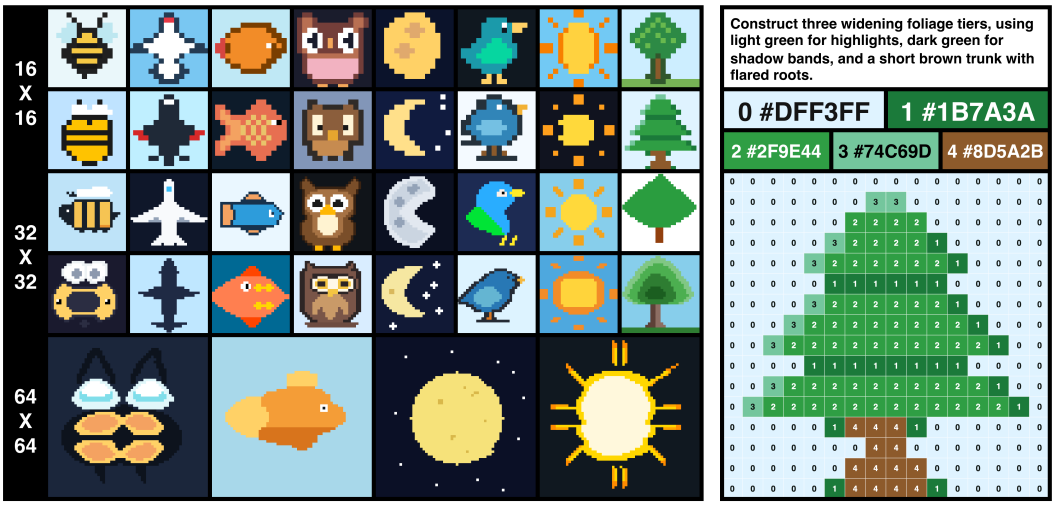



**Figure 1**: **Zero-shot text-to-pixel generation by large language models.** Left: pixel images generated at different resolutions. Right: our generation protocol, in which the model defines a color palette and outputs a numerical grid whose entries index the palette colors. See Figure 3 for image-to-pixel generation.

## ABSTRACT

Large language models (LLMs) can generate visual assets through text-based formats such as SVG and CAD, but these symbolic representations describe images using high-level primitives rather than explicit raster values. We ask whether LLMs can instead generate pixel images directly through their existing vocabulary, without a learned image tokenizer or decoder. We introduce PixelBench, a benchmark for text-to-pixel (T2P) and image-to-pixel (I2P) generation across multiple resolutions, together with a complementary pixel-understanding evaluation. Models generate an RGB color palette followed by a grid of indices specifying the palette color of each output pixel. In T2P, the strongest models produce recognizable images directly from text. In I2P, the strongest models outperform palette-matched downsampling and specialized pixelization algorithms at low resolutions. Pixel-understanding performance correlates strongly with generation performance. A resolution-scaling study shows that pixel-generation performance often improves initially before declining at high resolutions. Finally, we introduce Pixel Reconstruction Tuning (PRT), a self-supervised objective that trains VLMs to reconstruct images as pixel grids. PRT improves average performance across a suite of 13 visual-centric benchmarks, indicating that explicit pixel reconstruction can provide useful auxiliary supervision.

## 1 INTRODUCTION

Large language models (LLMs) were originally developed to predict the next token in natural-language sequences. The same autoregressive training paradigm has since produced strong code-generation capabilities. Some generated code, including SVG code and CAD programs, can be



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.

rendered into images, providing an interface for probing how LLMs express visual information (Sharma et al., 2024; Rodriguez et al., 2025; Xing et al., 2025; Khan et al., 2024). However, these formats describe images using semantically meaningful, high-level primitives and are therefore one step removed from visual appearance: a renderer expands each primitive into many pixels. By contrast, RGB rasters provide a more direct and ubiquitous representation of digital images, with each spatial location in a two-dimensional grid storing a color value. Most modern image generators map learned visual tokens or latent representations to pixels through dedicated decoders (Chameleon Team, 2024; Wang et al., 2026). Prior work has also trained specialized autoregressive image models to generate images directly in raster space (van den Oord et al., 2016a;b; Parmar et al., 2018; Chen et al., 2020). These studies establish the feasibility of direct pixel generation, but use models designed and trained specifically for image modeling. To probe visual understanding and output more directly, we ask:

> Can today's LLMs generate and understand visual content pixel by pixel?

To address this research question, we introduce **PixelBench**, a benchmark for evaluating LLMs on direct pixel generation and understanding. PixelBench uses a palette-indexed raster representation: the model first defines a palette of up to eight RGB colors and then generates a grid of palette indices, with each grid position specifying the color of one pixel. This interface requires no learned visual tokenizer, image decoder, or architectural modification. PixelBench evaluates *text-to-pixel generation* (T2P), in which a model maps an object label to a recognizable pixel image; *image-to-pixel generation* (I2P), in which the model reconstructs a provided image in the same representation; and pixel understanding, in which the model receives only the textual palette-and-grid representation from one of two complementary pixel oracles—one algorithmic and one model-based—and identifies its underlying concept.

Our evaluation shows that frontier LLMs already possess substantial pixel-generation capability: leading models produce recognizable T2P outputs and competitive I2P reconstructions. Their behavior nevertheless depends strongly on target resolution. For many models, both visual quality and quantitative performance peak at an intermediate resolution, with $64 \times 64$ often underperforming $32 \times 32$; GPT-6 Astra is notably more robust. We therefore conduct a fine-grained resolution-scaling study and find that, apart from Astra's consistently strong outputs, the evaluated models generally improve and then decline as resolution increases. If resolution is viewed as controlling the level of visual abstraction, these model-specific peaks suggest that different models operate most effectively at different resolutions. Pixel understanding exhibits a similar model ranking and correlates strongly with generation performance, suggesting that the connection between visual understanding and generation may persist even when both are probed through a text-based pixel interface.

Beyond evaluation, we study pixel generation as a training paradigm. Reconstruction Alignment (RecA) shows that reconstruction-aligned visual representations can improve image generation (Xie et al., 2026), but whether reconstruction also benefits visual understanding remains underexplored. We propose Pixel Reconstruction Tuning (PRT), which treats I2P as an auxiliary reconstruction objective and trains VLMs on algorithmically constructed palette-indexed targets. PRT yields positive aggregate gains for Qwen3-VL-Instruct models at the 2B, 4B, and 8B scales across a suite of 13 visual-centric benchmarks. Ablations over model scale, target resolution (including mixed-resolution targets), and pixel-data quantity in Section 5.3 further support the effectiveness and robustness of this training paradigm. Our contributions are summarized as follows:

- **An interface for evaluating direct pixel capabilities.** We introduce PixelBench, a shared text-based pixel interface for comprehensively evaluating frontier LLMs on pixel generation and pixel understanding.
- **Capability limits under increasing resolution.** We characterize model-specific performance peaks and failure modes: GPT-6 Astra remains robust as resolution increases, whereas the other evaluated models generally improve and then decline.
- **The training value of pixel reconstruction.** We propose Pixel Reconstruction Tuning using algorithmically constructed pixel targets. PRT improves aggregate performance across a 13-benchmark visual-centric suite, with ablations over model scale, target resolution, and pixel-data quantity.

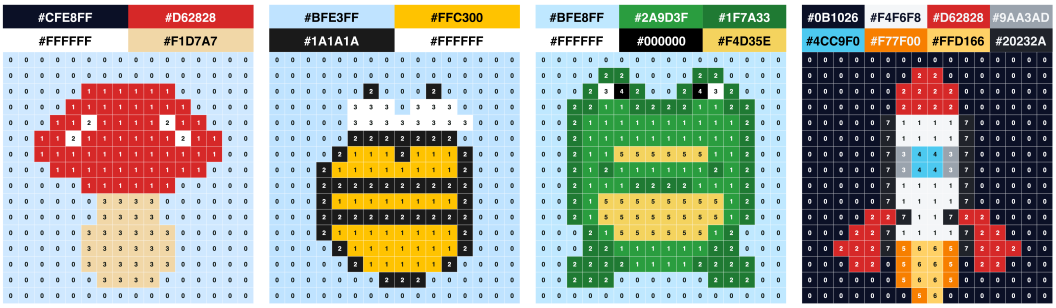



**Figure 2**: **Examples of palette-indexed pixel representations.** Each example pairs a compact color palette (top) with a $16 \times 16$ grid of palette indices (bottom). The mushroom, bee, frog, and rocket remain recognizable despite the low resolution and restricted palettes.

## 2 Pixels as a Visual Representation

Pixel art is a compact visual output format. Its restricted palette and raster structure can preserve recognizable colors and shapes while limiting the number of symbols that a model must generate. Related representations are commonly used in icons, pictograms, low-resolution displays, and digital graphics. In this work, *pixel art* refers specifically to a palette-indexed raster with two components: a palette of RGB colors and an $S \times S$ grid of palette indices. Each grid entry selects the color of one pixel. The representation is human-interpretable and deterministic to render, and can be expressed using the model's existing text vocabulary. We serialize the palette and index grid as JSON. The `palette` field contains two to eight opaque RGB colors in `#RRGGBB` format, and the `pixels` field contains the $S$ rows of palette indices. Rendering maps each index to its corresponding RGB value. Figure 2 shows rendered examples and their palettes.

## 3 PixelBench

We introduce PixelBench to probe how much image-generation capability a language model can express through its native token interface. Unlike SVG and CAD, which describe images through high-level geometric primitives, our palette-indexed grids specify raster values directly: each generated grid entry corresponds to one rendered pixel, and rendering requires only a deterministic palette lookup. PixelBench therefore evaluates visual generation through a representation closer to the native form of raster images, without relying on a learned image tokenizer or decoder.

Section 3.1 defines the two generation tasks, the complementary understanding evaluation, evaluated models, baselines, and protocol; Section 3.2 specifies the evaluation metrics; and Section 3.3 presents the main results. We find that current models already exhibit substantial pixel-image generation ability through their text interface, while their pixel-understanding performance closely tracks their generation performance. Generation quality does not consistently improve with output resolution, as further analyzed in Section 4.

### 3.1 Benchmark Design

**Tasks and datasets.** In *text-to-pixel generation* (T2P), a model maps an English object label to a recognizable icon without a reference image. We use 100 concrete concepts spanning five categories (Appendix H). In *image-to-pixel generation* (I2P), a vision encoder supplies a natural image and the model reconstructs it while preserving subject identity, pose, composition, and dominant colors. We sample one validation image from each of 100 ImageNet-1K classes (Deng et al., 2009) and center-crop it to $256 \times 256$ (Appendix I); the category is withheld. Every item in both generation tasks is evaluated at $16^2$, $32^2$, and $64^2$, yielding 300 instances per task and model. In *pixel understanding*, a model receives only the textual serialization of an RGB palette and index grid and selects the concept from ten candidates. Inputs come from deterministic PIA-reduced reconstructions and GPT-6 Astra I2P generations (Astra-I2P), each with 100 questions at each resolution.

**Models and baselines.** We evaluate Gemini 3.6 Flash (Google DeepMind, 2026b), GPT-5.6 Sol (OpenAI, 2026a), GPT-6 Astra (OpenAI, 2026b), Claude Opus 5 (Anthropic, 2026), Seed-2.1 Pro (ByteDance Seed, 2026), and Kimi K3 (Kimi Team, 2026) on both generation tasks and pixel understanding in a zero-shot setting. DeepSeek V4 Pro (DeepSeek-AI, 2026) is included as a supplementary model for T2P and pixel understanding, but not for I2P generation. Detailed task-specific inference settings, including temperature, reasoning configuration, output-token limits, and samples per item, are reported in Appendix E. For I2P, we additionally compare against three representative palette-matched algorithms: nearest-neighbor sampling followed by median-cut quantization (Heckbert, 1982); SSIM-optimal downscaling followed by median cut (Öztireli & Gross, 2015); and a reduced Pixelated Image Abstraction pipeline (Gerstner et al., 2012) that combines SLIC regions (Achanta et al., 2012) with an eight-color weighted $k$-means palette. The complete ten-method comparison and implementation details are in Appendix L.

**Generation protocol.** Both generation tasks use the palette-indexed representation in Figure 2: two to eight opaque RGB colors and an exactly sized index grid, with task-specific prompts. Successfully returned outputs with invalid JSON, palette references, or grid sizes receive zero; rate limits and token-limit truncations are retried, raising the limit when needed. Pixel understanding directly presents the serialized grid and ten choices, with no rendered image. Prompts are in Appendix G.

### 3.2 Evaluation Metrics

**Text-to-Pixel metrics.** Using the paired SigLIP-SO400M text and vision encoders (Zhai et al., 2023), we retrieve text candidates represented as "a pixel art icon depicting {label}." The main table reports 500-way Top-1 and a complementary VLM-as-judge score from Gemini 3.5 Flash (Google DeepMind, 2026a). The judge assigns a 0–100 rubric score; its prompt is in Appendix G.2. The 100-way Top-1 and MRR diagnostics are in Appendix K; retrieval definitions and vocabulary are in Appendices F and J.

**Image-to-Pixel metrics.** We report *100-way Top-1* by embedding each generated output and the 100 candidate source images with the same SigLIP vision encoder. We additionally compute paired cosine similarity using pooled SigLIP features and DINOv2-Large CLS features (Oquab et al., 2024), which measure semantic and appearance-level similarity to the source, respectively. For both generation tasks, invalid outputs count as retrieval misses or zero similarity; we also report the valid-generation and flat-fill rates, where a flat fill is a valid grid rendered in a single color.

**Pixel-understanding metric.** We report 10-way recognition accuracy from the serialized palette-and-grid inputs. Each resolution contains 100 questions, so each source average covers 300 questions and the overall average covers all 600. Random-choice accuracy is 10%.

### 3.3 Overall Results

Tables 1, 2, and 3 summarize the main PixelBench results.

**Text-to-Pixel Generation.** GPT-6 Astra leads throughout: its 500-way Top-1 rises from 87% to 94% across resolutions, while Gemini 3.5 Flash assigns 90.8/91.6/89.7, yielding an Agg. score of 90.8. GPT-5.6 Sol and Claude Opus 5 follow at 81.0 and 79.4, with Kimi K3 at 76.0. The stricter judge spans 90.66–51.45 overall and leaves Astra 14.01 points ahead of Sol, providing clearer discrimination while preserving the same leader. Figure 3 and Appendix C further show that model-generated pixelizations can preserve semantic structure; together with Astra's leading I2P Aggregated score, this motivates Astra-I2P as one understanding oracle. Full 100-way Top-1 and MRR results are in Appendix K. Most models decline from $32^2$ to $64^2$, motivating Section 4.

**Table 1**: **Text-to-Pixel generation results.** We report 500-way Top-1 retrieval accuracy and a VLM-as-judge score at $16^2$, $32^2$, and $64^2$. The Aggregated score is the mean of these six values. Valid and Flat report output validity and single-color generation rates. Gemini 3.5 Flash serves as the judge; 100-way Top-1 and MRR are in the appendix. Bold and underline mark the top two results per column. Agg. abbreviates Aggregated score.

| | | | 500-way Top-1 | | | | VLM Judge | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | Valid | Flat | 16 | 32 | 64 | Avg | 16 | 32 | 64 | Avg | Agg. |
| GPT-6 Astra | 100.0 | 0.0 | **87** | **92** | **94** | **91.0** | **90.8** | **91.6** | **89.7** | **90.7** | **90.8** |
| Claude Opus 5 | 100.0 | 0.0 | 81 | 87 | 89 | 85.7 | 75.1 | 74.1 | 70.1 | 73.1 | 79.4 |
| GPT-5.6 Sol | 100.0 | 0.0 | 84 | 89 | 83 | 85.3 | 78.1 | 78.9 | 73.0 | 76.7 | 81.0 |
| Kimi K3 | 99.7 | 0.0 | 81 | 87 | 77 | 81.7 | 74.5 | 74.7 | 61.5 | 70.3 | 76.0 |
| Gemini 3.6 Flash | 100.0 | 1.3 | **87** | 78 | 72 | 79.0 | 77.8 | 71.2 | 60.4 | 69.8 | 74.4 |
| Seed-2.1 Pro | 100.0 | 0.0 | 70 | 72 | 74 | 72.0 | 68.6 | 64.6 | 59.3 | 64.2 | 68.1 |
| DeepSeek V4 Pro | 98.7 | 1.3 | 49 | 59 | 51 | 53.0 | 53.7 | 55.0 | 45.7 | 51.5 | 52.2 |

**Image-to-Pixel Generation.** Table 2 compares six multimodal language models with three palette-matched algorithms. Five models are fully valid; Kimi K3 reaches 96.3%, and only Gemini 3.6 Flash produces flat fills (1.0%). GPT-5.6 Sol leads model Top-1 Avg (56.0%), while GPT-6 Astra leads model SigLIP (.654) and DINO (.175) averages. The best model exceeds all algorithms at $16^2$ and $32^2$, but algorithms lead at $64^2$: PIA reaches 78% Top-1 and .697 SigLIP, and SSIM-optimized pixelization reaches .543 DINO. Four models also decline from $32^2$ to $64^2$. GPT-6 Astra achieves the highest Aggregated score (80.45), ahead of PIA (77.50). See Appendix L for all algorithms.

**Table 2**: **Image-to-Pixel results.** Valid and Flat are percentages over 300 model–resolution items for models and 100 images per resolution for algorithms. Top-1 uses the 100 source images; SigLIP and DINO are paired cosines, and invalid generations score zero. Agg. denotes the Aggregated score: the mean of the min–max-normalized Top-1, SigLIP, and DINO average columns. Colors are normalized within each metric family; bold and underline mark the top two results per column.

| | | | 100-way Top-1 | | | | SigLIP cosine | | | | DINO cosine | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model / method | Valid | Flat | 16 | 32 | 64 | Avg | 16 | 32 | 64 | Avg | 16 | 32 | 64 | Avg | Agg. |
| *Multimodal language models* | | | | | | | | | | | | | | | |
| GPT-6 Astra | 100.0 | 0.0 | 18 | 50 | 76 | 48.0 | **.615** | **.652** | .695 | **.654** | **.023** | **.177** | .326 | .175 | **80.45** |
| GPT-5.6 Sol | 100.0 | 0.0 | **44** | **61** | 63 | **56.0** | .569 | .597 | .599 | .588 | .022 | .112 | .147 | .094 | 57.27 |
| Claude Opus 5 | 100.0 | 0.0 | 26 | 49 | 47 | 40.7 | .535 | .567 | .582 | .561 | .021 | .062 | .098 | .060 | 23.16 |
| Kimi K3 | 96.3 | 0.0 | 34 | 46 | 30 | 36.7 | .558 | .562 | .501 | .541 | .020 | .064 | .038 | .040 | 8.48 |
| Seed-2.1 Pro | 100.0 | 0.0 | 33 | 42 | 30 | 35.0 | .536 | .553 | .553 | .547 | .020 | .056 | .054 | .043 | 8.52 |
| Gemini 3.6 Flash | 100.0 | 1.0 | 28 | 37 | 27 | 30.7 | .535 | .546 | .536 | .539 | .018 | .065 | .070 | .051 | 1.98 |
| *Algorithmic baselines* | | | | | | | | | | | | | | | |
| Nearest + median cut | 100.0 | 0.0 | 9 | 28 | 62 | 33.0 | .523 | .568 | .665 | .585 | .012 | .090 | .480 | .194 | 44.11 |
| SSIM-opt. + median cut | 100.0 | 0.0 | 9 | 34 | 75 | 39.3 | .524 | .579 | .686 | .596 | .013 | .119 | **.543** | **.225** | 61.19 |
| PIA (reduced) | 100.0 | 0.0 | 18 | 50 | **78** | 48.7 | .546 | .608 | **.697** | .617 | .014 | .135 | .490 | .213 | 77.50 |

## Pixel Understanding.

We construct understanding inputs from two complementary sources. PIA provides a deterministic transformation, but it is not uniformly optimal. GPT-6 Astra achieves the highest I2P Aggregated score (80.45), ahead of PIA (77.50), and at $32^2$ also leads PIA on SigLIP cosine (.652 vs. .608) and DINO cosine (.177 vs. .135). Figure 3 and Appendix C further show cases in which model-generated outputs match or surpass deterministic pixelization. We therefore use Astra-I2P as a complementary model-generated oracle alongside PIA.

**Table 3**: **Pixel understanding.** Each source contributes 100 ten-way questions per resolution. Overall averages 600 questions; each source average covers 300. Chance accuracy is 10%. Background color is globally normalized from 8 (white) to 85 (blue); bold and underline mark the best and second-best values in each column.

| | | PIA source | | | | Astra-I2P source | | | |
|---|---|---|---|---|---|---|---|---|---|
| Model | Overall | 16 | 32 | 64 | Avg | 16 | 32 | 64 | Avg |
| GPT-6 Astra | **67.5** | **48** | **57** | **67** | **57.3** | **72** | **76** | **85** | **77.7** |
| GPT-5.6 Sol | 39.2 | 34 | 29 | 34 | 32.3 | 45 | 42 | 51 | 46.0 |
| Claude Opus 5 | 35.3 | 22 | 26 | 24 | 24.0 | 40 | 50 | 50 | 46.7 |
| Kimi K3 | 27.3 | 20 | 19 | 16 | 18.3 | 39 | 34 | 36 | 36.3 |
| Gemini 3.6 Flash | 21.8 | 21 | 11 | 17 | 16.3 | 28 | 24 | 30 | 27.3 |
| Seed-2.1 Pro | 16.0 | 15 | 14 | 12 | 13.7 | 20 | 15 | 20 | 18.3 |
| DeepSeek V4 Pro | 12.5 | 13 | 10 | 8 | 10.3 | 15 | 14 | 15 | 14.7 |

**Table 4**: **Correlation summary.** Pearson, Spearman, and Kendall correlations between pixel understanding and generation performance across models, using the T2P and I2P Aggregated scores as the respective generation summaries. The I2P comparison excludes DeepSeek V4 Pro, which has no I2P generation result.

| Comparison | $n$ | Pearson $r$ | Spearman $\rho$ | Kendall $\tau$ |
|---|---|---|---|---|
| Understanding–T2P | 7 | .871 | 1.000 | 1.000 |
| Understanding–I2P (w/o DeepSeek) | 6 | .930 | .829 | .733 |

Table 3 reports recognition accuracy for both sources. GPT-6 Astra leads overall accuracy (67.5%), followed by GPT-5.6 Sol (39.2%). On PIA alone it already leads by a wide margin (57.3% vs. 32.3% for GPT-5.6 Sol), and the two sources largely preserve the same ranking. Switching to Astra-I2P raises every model's source average, with Claude Opus 5 gaining more than Astra (+22.7 vs. +20.4), so the model-based oracle helps broadly rather than rewarding Astra for recognizing its own generations. Overall accuracy correlates with the T2P Aggregated score ($r = .871, \rho = 1.000, \tau = 1.000$; $n = 7$) and the I2P Aggregated score ($r = .930, \rho = .829, \tau = .733$; $n = 6$, excluding DeepSeek V4 Pro). Although the perfect T2P rank correlations preserve the ordering of these seven models, the set is small; we therefore treat this result as descriptive rather than general, while using it to motivate the training experiment in Section 5.

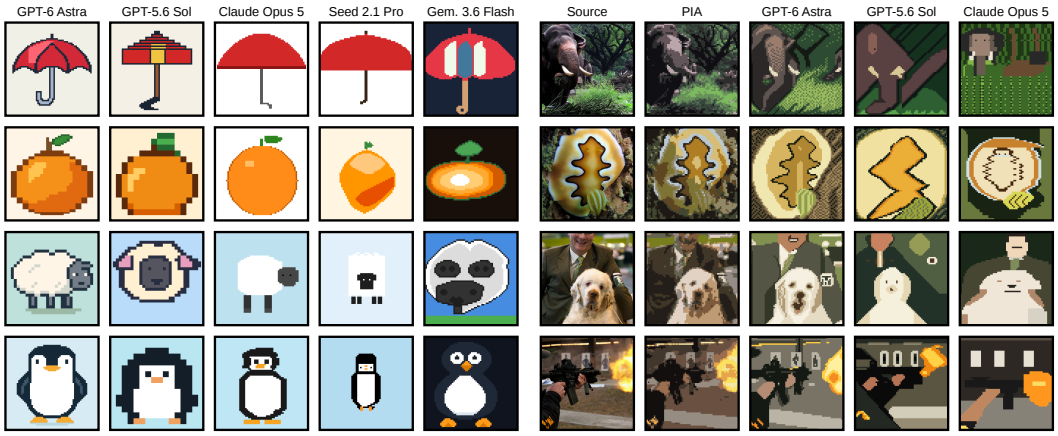



**Figure 3**: **Qualitative PixelBench generations. Left:** T2P outputs from GPT-6 Astra, GPT-5.6 Sol, Claude Opus 5, Seed-2.1 Pro, and Gemini 3.6 Flash. **Right:** four I2P sources and reconstructions from PIA, Astra, Sol, and Claude. Some model outputs preserve subjects more clearly at low resolutions, whereas PIA better follows local appearance as resolution grows. PIA is the reduced variant (Appendix L); Appendix C provides 20 additional cases per task.

## 4 Resolution Scaling and Failure Modes

The main results suggest that higher resolution is not uniformly better: for many models in both T2P and I2P, $64^2$ underperforms $32^2$ (Tables 1 and 2). To examine this behavior more closely, we sweep GPT-6 Astra, GPT-5.6 Sol, Claude Opus 5, and Gemini 3.6 Flash over ten resolutions from $16^2$ to $128^2$. The number of predicted grid positions grows quadratically with side length: relative to $16^2$, $64^2$ and $128^2$ require $16\times$ and $64\times$ as many pixel indices. Resolution scaling therefore increases both representational capacity and the burden of coordinating a long spatial output.

*Text-to-Pixel.* The left two panels of Figure 4 pair 500-way retrieval with VLM-as-judge quality and show that higher resolution does not yield consistent gains under either metric. GPT-6 Astra remains strong throughout: its judge score stays near its peak through $96^2$ and declines only modestly at the two largest resolutions. GPT-5.6 Sol fluctuates after an early peak, while Claude Opus 5 shows retrieval oscillations but a clearer downward trend under VLM judging. Gemini 3.6 Flash degrades most sharply at high resolutions, especially beyond $64^2$. The agreement between retrieval and judge-based evaluation indicates that this lack of monotonic scaling is not specific to the retrieval metric.

*Image-to-Pixel.* The right two panels show a clearer scaling pattern. GPT-6 Astra increasingly exploits the larger grid: its retrieval accuracy and DINO similarity rise overall and remain strong at the highest resolutions. The other models generally peak at intermediate resolutions and then

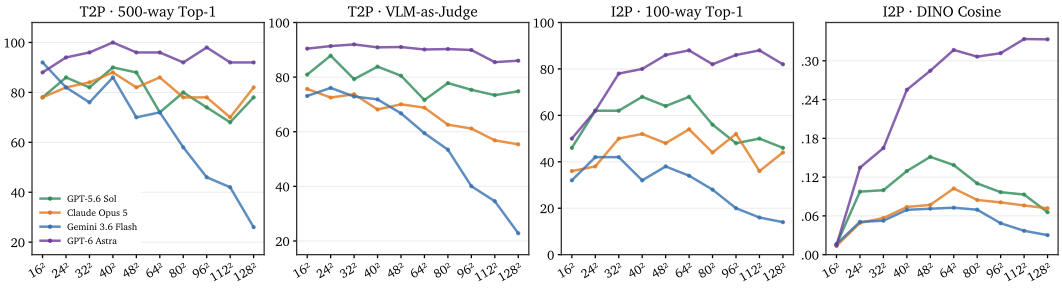



**Figure 4**: **Fine-grained resolution scaling on PixelBench.** From left to right: T2P 500-way Top-1, T2P VLM-as-Judge, I2P 100-way Top-1, and paired I2P DINOv2 CLS cosine from $16^2$ to $128^2$. Retrieval and VLM judging agree that higher resolution does not consistently improve T2P: Astra remains robust, while the other models fluctuate or decline. In I2P, Astra continues to benefit from larger grids, whereas the other models generally peak at intermediate resolutions and then decline.

decline. Larger grids therefore provide additional expressive capacity, but only Astra consistently converts it into more faithful reconstructions.

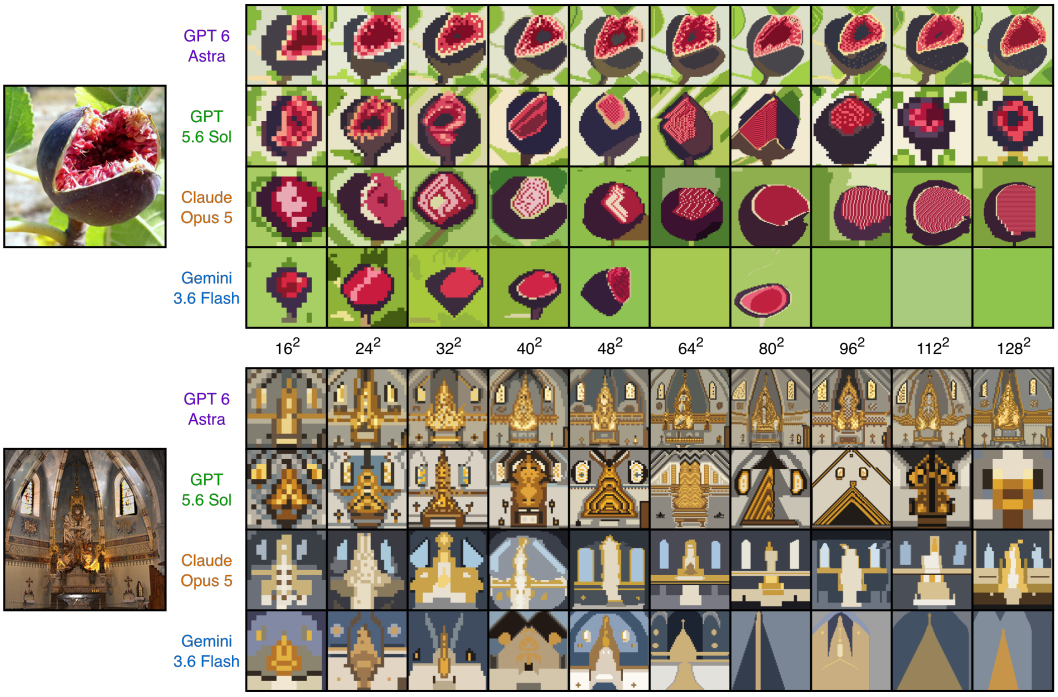



**Figure 5**: **Qualitative I2P resolution sweep.** Two source images (left) are reconstructed from $16^2$ to $128^2$. For each source, rows from top to bottom correspond to GPT-6 Astra, GPT-5.6 Sol, Claude Opus 5, and Gemini 3.6 Flash. Astra remains strongest and adds coherent detail as resolution grows; GPT-5.6 Sol collapses to a lower effective resolution, Claude develops coarse or repetitive local patterns, and Gemini increasingly omits the subject or other key structures.

To examine these differences directly in model outputs, Figure 5 shows two I2P examples across the same resolution sweep. GPT-6 Astra progressively adds texture, sharper boundaries, and architectural detail while preserving the subject and overall layout. GPT-5.6 Sol instead exhibits an *effective-resolution collapse*: despite the increasing nominal grid size, its outputs become dominated by large color blocks. This behavior can preserve a coarse semantic layout, but it underuses the available grid and reduces fidelity. Claude Opus 5 shows input-dependent failures. In the fruit example, it loses the global structure and repeatedly extends local stripe patterns; in the church example, it similarly collapses toward coarse regions and a lower effective resolution. Gemini 3.6 Flash degrades most severely, with the subject or key structures progressively disappearing at high resolutions.

Together, these examples expose three high-resolution failure modes: effective-resolution collapse, repetitive local patterns that disrupt global structure, and omission of the subject or other key structures. Astra's advantage is therefore not merely its ability to produce longer outputs, but its ability to organize additional grid positions into details that remain coherent with the global structure. Representative T2P cases in Appendix B exhibit the same failure modes.

## 5 Pixel Reconstruction Tuning

PixelBench shows that existing models can process explicit pixel representations through their text interface, both by generating palette-indexed grids and by recognizing their content. This capability motivates us to investigate the same representation as an auxiliary training signal for broader visual understanding.

**Table 5**: **PRT evaluation suite.** Benchmark names are followed directly by their citations. $N$ counts evaluated units after pair-level aggregation for MMVP and CircularEval aggregation for MMBench; TallyQA uses a randomly sampled 3,000-item subset. BlindTest balances its nine subtasks by sampling 444–445 examples from each of the six larger subtasks and retaining all examples from the three smaller ones.

| Benchmark | Task content | Split / sampling | $N$ |
|---|---|---|---|
| **MMVP** (Tong et al., 2024b) | Visual-detail VQA on CLIP-blind pairs | Full benchmark | 150 |
| **BabyVision** (Chen et al., 2026) | Foundational visual perception and reasoning | Full benchmark | 388 |
| **CV-Bench** (Tong et al., 2024a) | 2D relations/counting; 3D depth/distance | Full test | 2,638 |
| **TallyQA** (Acharya et al., 2019) | Open-ended visual counting | Random 3k subset | 3,000 |
| **HallusionBench** (Guan et al., 2024) | Visual illusion and language hallucination | Full benchmark | 1,129 |
| **V-STAR Bench** (Wu & Xie, 2024) | High-resolution visual search and grounding | Full benchmark | 191 |
| **Q-Bench** (Wu et al., 2024) | Low-level visual perception | Full dev | 1,495 |
| **PerceptionBench** (Lin et al., 2026) | Atomic visual perception | Full benchmark | 3,000 |
| **MMStar** (Chen et al., 2024) | Vision-indispensable perception and reasoning | Full validation | 1,500 |
| **MMBench** (Liu et al., 2024b) | Multimodal perception and reasoning | EN dev v1.1 | 1,292 |
| **BlindTest** (Rahmanzadehgervi et al., 2024) | Geometric primitives and spatial precision | Balanced subset | 3,434 |
| **BLINK** (Fu et al., 2024) | Fourteen classic computer-vision tasks | Full validation | 1,901 |
| **VLMBias** (Vo et al., 2026) | Counterfactual counting versus priors | Full main | 2,784 |

Pixel Reconstruction Tuning (PRT) trains a VLM to reconstruct visual content in our palette-indexed pixel representation. Relatedly, Reconstruction Alignment (RecA) provides evidence in the opposite direction: conditioning image generation on visual-understanding representations can improve generation (Xie et al., 2026). Quantized grids provide deterministic, spatially aligned targets derived from source images, whereas captions are sparse and admit many valid descriptions. We evaluate whether this auxiliary pixel-reconstruction objective improves visual understanding across 13 downstream visual-centric benchmarks.

### 5.1 Data Construction and Training

**Pixel data.** We sample source images from five datasets in fixed proportions: 25% ImageNet (Deng et al., 2009), 25% LAION (Schuhmann et al., 2022), 20% Places365 (Zhou et al., 2017), 15% COCO (Lin et al., 2014), and 15% Objects365 (Shao et al., 2019). Each image is center-cropped to $256 \times 256$ and automatically converted into a pixel-reconstruction target using the deterministic eight-color PIA-reduced procedure described in Appendix L. This fully algorithmic pipeline requires neither manual annotation nor model-generated supervision and can scale directly with unlabeled image collections. The primary +Pixel configuration mixes reconstruction targets approximately evenly across $16^2$, $32^2$, and $64^2$; Section 5.3 ablates target resolution.

**Training mixtures.** All PRT runs initialize from Qwen3-VL-Instruct checkpoints at the 2B, 4B, and 8B scales, abbreviated below as 2B, 4B, and 8B. For each scale, the FineVision (FV)-only baseline uses 64,000 FineVision occurrences (Wiedmann et al., 2025). Each FV+Pixel configuration retains those same 64,000 FV occurrences and adds 3,000 pixel-reconstruction occurrences. Each FV+Pixel run starts from the corresponding Instruct checkpoint and is fine-tuned for 1,047 optimizer steps; its FV-only control starts from the same checkpoint and trains for 1,000 steps. In addition to this additive protocol, Appendix D provides full training details and a matched-data, equal-training-step control.

### 5.2 Results

Table 6 reports results on the 13 visual-centric benchmarks. +Pixel improves the Example/Benchmark averages by 2.0/3.0 points for 2B, 2.9/3.7 for 4B, and 2.1/1.6 for 8B. The largest gains occur on MMVP ($+11.3/+20.7/+4.0$) and BlindTest ($+1.7/+9.6/+7.8$); CV-Bench also improves at every scale ($+3.0/+3.0/+1.0$). Gains on broader multimodal suites are generally smaller (MMStar: $+0.3/+2.3/+2.3$; MMBench: $+4.8/+2.2/+1.6$). This distribution is consistent with, but does not establish, the hypothesis that dense, spatially aligned pixel targets particularly benefit low-level and spatial perception. Benefits are not uniform: HallusionBench changes by $+2.8/-2.1/-4.1$ for 2B/4B/8B. Appendix D reports full results and the matched-data, equal-training-step control.

### 5.3 Target Resolution and Grid Quantity Ablations

Tables 7 and 8 summarize the cross-scale target-resolution and Grid-quantity sweeps. In both tables, each entry reports the change from the corresponding FV-only baseline.

**Table 6**: **Pixel reconstruction tuning across model scales.** Results on 13 benchmarks for Qwen3-VL-Instruct models at the 2B, 4B, and 8B scales. FV uses 64,000 FineVision occurrences; +Pixel retains all 64,000 and adds 3,000 pixel-reconstruction occurrences mixed approximately evenly across $16^2$, $32^2$, and $64^2$. $\Delta$ is +Pixel minus FV in percentage points. Rows are independently ordered by $\Delta$. Example averaging assigns equal weight to every official evaluation unit; Benchmark averaging assigns equal weight to every benchmark. * denotes a randomly sampled subset.

| **2B** | $N$ | FV | +Pixel | $\Delta$ |
|---|---|---|---|---|
| MMVP | 150 | 30.0 | 41.3 | +11.3 |
| BabyVision | 388 | 9.0 | 13.9 | +4.9 |
| MMBench | 1292 | 69.7 | 74.5 | +4.8 |
| CV-Bench | 2638 | 75.2 | 78.2 | +3.0 |
| TallyQA* | 3000 | 73.3 | 76.2 | +2.9 |
| Hallusion | 1129 | 53.8 | 56.6 | +2.8 |
| VStar | 191 | 60.7 | 63.4 | +2.6 |
| Q-Bench | 1495 | 65.3 | 67.4 | +2.1 |
| Perception | 3000 | 22.2 | 24.1 | +1.9 |
| BlindTest | 3434 | 34.7 | 36.4 | +1.7 |
| BLINK | 1901 | 50.8 | 52.4 | +1.6 |
| MMStar | 1500 | 48.9 | 49.3 | +0.3 |
| VLMBias | 2784 | 21.2 | 20.5 | −0.8 |
| Example avg. | 22902 | 48.1 | 50.1 | +2.0 |
| Benchmark avg. | 13 | 47.3 | 50.3 | +3.0 |

| **4B** | $N$ | FV | +Pixel | $\Delta$ |
|---|---|---|---|---|
| MMVP | 150 | 38.7 | 59.3 | +20.7 |
| BlindTest | 3434 | 43.4 | 53.0 | +9.6 |
| BabyVision | 388 | 12.4 | 17.5 | +5.2 |
| CV-Bench | 2638 | 82.3 | 85.4 | +3.0 |
| VLMBias | 2784 | 22.2 | 24.9 | +2.7 |
| MMStar | 1500 | 61.0 | 63.3 | +2.3 |
| MMBench | 1292 | 78.6 | 80.8 | +2.2 |
| Perception | 3000 | 24.1 | 25.6 | +1.5 |
| BLINK | 1901 | 56.3 | 57.5 | +1.3 |
| Q-Bench | 1495 | 74.6 | 75.4 | +0.8 |
| TallyQA* | 3000 | 76.3 | 76.9 | +0.6 |
| VStar | 191 | 64.9 | 65.4 | +0.5 |
| Hallusion | 1129 | 62.3 | 60.1 | −2.1 |
| Example avg. | 22902 | 53.9 | 56.8 | +2.9 |
| Benchmark avg. | 13 | 53.6 | 57.3 | +3.7 |

| **8B** | $N$ | FV | +Pixel | $\Delta$ |
|---|---|---|---|---|
| BlindTest | 3434 | 48.9 | 56.8 | +7.8 |
| MMVP | 150 | 55.3 | 59.3 | +4.0 |
| Q-Bench | 1495 | 73.2 | 75.8 | +2.6 |
| TallyQA* | 3000 | 75.7 | 78.3 | +2.6 |
| MMStar | 1500 | 59.9 | 62.1 | +2.3 |
| BabyVision | 388 | 12.4 | 14.4 | +2.1 |
| BLINK | 1901 | 55.3 | 57.0 | +1.7 |
| MMBench | 1292 | 80.4 | 82.0 | +1.6 |
| Perception | 3000 | 25.4 | 26.8 | +1.3 |
| CV-Bench | 2638 | 84.1 | 85.1 | +1.0 |
| VStar | 191 | 71.2 | 70.2 | −1.0 |
| VLMBias | 2784 | 19.9 | 18.8 | −1.1 |
| Hallusion | 1129 | 70.2 | 66.2 | −4.1 |
| Example avg. | 22902 | 55.1 | 57.2 | +2.1 |
| Benchmark avg. | 13 | 56.3 | 57.9 | +1.6 |

**Tables 7 & 8: Target-resolution sweep (left) and Grid-quantity sweep (right).** Accuracy deltas from FV-only, in percentage points; every entry is positive. **Left:** preferred settings differ by scale and averaging ($32^2$ for 2B, $16^2$ for 4B, and Mix/$64^2$ for 8B), while Mix is first or second in four of six rows. **Right:** Benchmark peaks move earlier with scale (5k/3k/1k for 2B/4B/8B), while Example peaks occur at 2k/3k/3k. Example pools official evaluation units; Benchmark weights benchmarks equally. Within each row, darker green indicates larger gains; bold and underline mark the best and second-best values.

| Model | Avg. | $\Delta_{16}$ | $\Delta_{32}$ | $\Delta_{64}$ | $\Delta_{\text{Mix}}$ |
|---|---|---|---|---|---|
| 2B | Example | +2.14 | **+2.50** | +2.27 | +2.01 |
| | Benchmark | +2.66 | **+3.37** | +2.99 | +3.02 |
| 4B | Example | **+3.14** | +2.87 | +2.18 | +2.94 |
| | Benchmark | **+4.07** | +3.90 | +3.48 | +3.70 |
| 8B | Example | +1.88 | +1.92 | +1.90 | **+2.07** |
| | Benchmark | +1.38 | +1.23 | **+1.73** | +1.60 |

| Model | Avg. | $\Delta_{1k}$ | $\Delta_{2k}$ | $\Delta_{3k}$ | $\Delta_{4k}$ | $\Delta_{5k}$ |
|---|---|---|---|---|---|---|
| 2B | Example | +2.43 | **+2.68** | +2.01 | +2.31 | +2.18 |
| | Benchmark | +3.30 | +3.20 | +3.02 | +3.31 | **+3.91** |
| 4B | Example | +2.73 | +2.08 | **+2.94** | +1.92 | +2.66 |
| | Benchmark | +3.33 | +3.01 | **+3.70** | +2.77 | +3.61 |
| 8B | Example | +1.59 | +1.21 | **+2.07** | +0.59 | +0.99 |
| | Benchmark | **+1.67** | +1.04 | +1.60 | +0.90 | +0.55 |

Each target-resolution setting adds 3,000 Grid occurrences to the same 64,000 FV occurrences. The single-resolution runs reuse the same 3,000 unique source images in separate runs, while Mix assigns each image to one resolution. Across both sweeps, every configuration improves over FV-only. In the resolution sweep, preferred targets vary by scale and averaging: $32^2$ for 2B, $16^2$ for 4B, and Mix/$64^2$ for 8B. Mix nevertheless remains broadly transferable, ranking first or second in four of six rows. In the quantity sweep, gains remain positive throughout; Benchmark-average peaks shift earlier with scale (5k/3k/1k), whereas Example peaks occur at 2k/3k/3k. These controls match reconstruction-example counts, not target tokens: $64^2$ targets contain sixteen times as many cells as $16^2$ targets, and quantity changes the amount of pixel supervision. Appendix D reports full deltas and the matched-data, equal-training-step control.

## 6 Conclusion

We introduced PixelBench, a shared text interface for evaluating direct pixel generation and understanding without a learned image tokenizer or decoder. Across T2P, I2P, and pixel understanding, frontier models show substantial but uneven capability: leading outputs are recognizable, rankings are similar across tasks, and resolution sweeps expose model-specific peaks and failure modes. GPT-6 Astra remains robust as grids grow, whereas most other models improve and then decline, showing that more output positions do not guarantee coherent detail. Beyond evaluation, Pixel Reconstruction Tuning uses algorithmically constructed pixel targets as auxiliary supervision. It improves aggregate Qwen3-VL-Instruct performance at 2B, 4B, and 8B across 13 visual-centric benchmarks, while ablations show that preferred target resolution and data quantity depend on model scale. Explicit pixels therefore provide both a behavioral probe and a practical training signal for visual understanding.

### AI USE STATEMENT

Generative AI tools were used for manuscript language editing, LaTeX formatting, and assistance with data aggregation and consistency checks. The underlying experiments were conducted independently, and all AI-assisted text, calculations, and formatting changes were reviewed and verified by the authors. The authors take full responsibility for the final content of this work.

### REPRODUCIBILITY STATEMENT

PixelBench construction, evaluation protocols, and metrics are described in Sections 3.1 and 3.2. PRT data construction and training are described in Section 5.1, with full prompts, baseline implementations, benchmark composition, and benchmark-level results provided in the appendix.

## REFERENCES


Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *TPAMI*, 34(11): 2274–2282, 2012.

Manoj Acharya, Kushal Kafle, and Christopher Kanan. Tallyqa: Answering complex counting questions. In *AAAI*, pp. 8076–8084, 2019.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: A visual language model for few-shot learning. In *NeurIPS*, 2022.

Anthropic. System card: Claude opus 5. https://www.anthropic.com/system-cards, 2026. Accessed 2026-08-21.

ByteDance Seed. Seed2.1 officially released: Advancing AI productivity. https://seed.bytedance.com/en/blog/seed2-1-officially-released-advancing-ai-productivity, 2026. Accessed 2026-08-21.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models, 2024.

Liang Chen, Weichu Xie, Yiyan Liang, Hongfeng He, Hans Zhao, Zhibo Yang, Zhiqi Huang, Haoning Wu, Haoyu Lu, et al. Babyvision: Visual reasoning beyond language. In *ICML*, 2026.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? In *NeurIPS*, 2024.

Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, pp. 1691–1703, 2020.

Siqi Chen, Xinyu Dong, Haolei Xu, Xingyu Wu, Fei Tang, Hang Zhang, Yuchen Yan, Linjuan Wu, Wenqi Zhang, Guiyang Hou, Yongliang Shen, Weiming Lu, and Yueting Zhuang. Svgenius: Benchmarking llms in svg understanding, editing and generation. In *ACM MM*, 2025a.

Yamei Chen, Haoquan Zhang, Yangyi Huang, Zeju Qiu, Kaipeng Zhang, Yandong Wen, and Weiyang Liu. Symbolic graphics programming with large language models, 2025b.

François Chollet. On the measure of intelligence, 2019.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *NeurIPS*, 2023.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and PixMo: Open weights and open data for state-of-the-art vision-language models. In *CVPR*, pp. 91–104, 2025.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pp. 248–255, 2009.

Emergent Mind. Pixel art bench. https://www.emergentmind.com/pixel-art-bench, 2026. Accessed 2026-08-19.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. BLINK: Multimodal large language models can see but not perceive. In *ECCV*, 2024.

Timothy Gerstner, Doug DeCarlo, Marc Alexa, Adam Finkelstein, Yotam Gingold, and Andrew Nealen. Pixelated image abstraction. In *NPAR*, 2012.

Google DeepMind. Gemini 3.5 flash model card. https://deepmind.google/models/model-cards/gemini-3-5-flash/, 2026a. Accessed 2026-08-21.

Google DeepMind. Introducing Gemini 3.6 flash, 3.5 flash-lite, and 3.5 flash cyber. https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-6-flash-3-5-flash-lite-3-5-flash-cyber/, 2026b. Accessed 2026-08-21.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. Hallusionbench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *CVPR*, pp. 14375–14385, 2024.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

Paul Heckbert. Color image quantization for frame buffer display. In *SIGGRAPH*, pp. 297–307, 1982.

Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic VLMs: Investigating the design space of visually-conditioned language models. In *ICML*, pp. 23123–23144, 2024.

Mohammad Sadil Khan, Sankalp Sinha, Talha Uddin Sheikh, Didier Stricker, Sk Aziz Ali, and Muhammad Zeshan Afzal. Text2CAD: Generating sequential cad designs from beginner-to-expert level text prompts. In *NeurIPS*, 2024.

Kimi Team. Kimi K3: Open frontier intelligence, 2026. URL https://arxiv.org/abs/2607.24653.

Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models? In *NeurIPS*, 2024.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, pp. 19730–19742, 2023.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, pp. 740–755, 2014.

Zichao Lin, Yifeng Xie, Bowen Qu, Haiming Wang, Jia Li, Haoning Wu, Yuhao Dong, Zuhao Yang, Jinguo Zhu, Haoyu Lu, et al. Perceptionbench: Evaluating atomic visual perception in multimodal large language models, 2026.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024a.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. MMBench: Is your multi-modal model an all-around player? In *ECCV*, pp. 216–233, 2024b.

OpenAI. GPT-5.6: Frontier intelligence that scales with your ambition. https://openai.com/index/gpt-5-6/, 2026a. Accessed 2026-08-21.

OpenAI. GPT-6 astra: A new generation of intelligence. https://openai.com/index/gpt-6-astra/, 2026b. Accessed 2026-09-16.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *TMLR*, 2024.

A. Cengiz Öztireli and Markus Gross. Perceptually based downscaling of images. *ACM TOG*, 34 (4), 2015.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *ICML*, pp. 4055–4064, 2018.

Zeju Qiu, Weiyang Liu, Haiwen Feng, Zhen Liu, Tim Z. Xiao, Katherine M. Collins, Joshua B. Tenenbaum, Adrian Weller, Michael J. Black, and Bernhard Schölkopf. Can large language models understand symbolic graphics programs? In *ICLR*, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pp. 8748–8763, 2021.

Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza Taesiri, and Anh Totti Nguyen. Vision language models are blind. In *ACCV*, pp. 18–34, 2024.

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. In *CVPR*, pp. 16175–16186, 2025.

Kenneth Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proc. IEEE*, 86(11):2210–2239, 1998.

Danila Rukhovich, Elona Dupont, Dimitrios Mallis, Kseniya Cherenkova, Anis Kacem, and Djamila Aouada. CAD-Recode: Reverse engineering cad code from point clouds. In *ICCV*, pp. 9801–9811, 2025.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. LAION-5B: An open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022.

Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *ICCV*, pp. 8430–8439, 2019.

Pratyusha Sharma, Tamar Rott Shaham, Manel Baradad, Stephanie Fu, Adrian Rodriguez-Munoz, Shivam Duggal, Phillip Isola, and Antonio Torralba. A vision check-up for language models. In *CVPR*, pp. 14410–14419, 2024.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. In *NeurIPS*, 2024a.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *CVPR*, 2024b.

Shengbang Tong, David Fan, Jiachen Zhu, Yunyang Xiong, Xinlei Chen, Koustuv Sinha, Michael Rabbat, Yann LeCun, Saining Xie, and Zhuang Liu. Metamorph: Multimodal understanding and generation via instruction tuning. In *ICCV*, pp. 17001–17012, 2025.

Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *ICML*, pp. 1747–1756, 2016a.

Aäron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with PixelCNN decoders. In *NeurIPS*, 2016b.

An Vo, Khai-Nguyen Nguyen, Mohammad Reza Taesiri, Vy Tuong Dang, Anh Totti Nguyen, and Daeyoung Kim. Vision language models are biased. In *ICLR*, 2026.

Xinlong Wang, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Zhen Li, et al. Multimodal learning with next-token prediction for large multimodal models. *Nature*, 650:327–333, 2026.

Nicolas Weber, Michael Waechter, Sandra C. Amend, Stefan Guthe, and Michael Goesele. Rapid, detail-preserving image downscaling. *ACM TOG*, 35(6), 2016.

Luis Wiedmann, Orr Zohar, Amir Mahla, Xiaohan Wang, Rui Li, Thibaud Frere, Leandro von Werra, Aritra Roy Gosthipaty, and Andrés Marafioti. Finevision: Open data is all you need, 2025.

Haoning Wu, Zicheng Zhang, Erli Zhang, Chaofeng Chen, Liang Liao, Annan Wang, Chunyi Li, Wenxiu Sun, Qiong Yan, Guangtao Zhai, and Weisi Lin. Q-Bench: A benchmark for general-purpose foundation models on low-level vision. In *ICLR*, 2024.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. In *CVPR*, pp. 13084–13094, 2024.

Ronghuan Wu, Wanchao Su, and Jing Liao. Chat2svg: Vector graphics generation with large language models and image diffusion models. In *CVPR*, pp. 23690–23700, 2025.

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 TOG*, 41(6), 2022.

Ji Xie, Trevor Darrell, Luke Zettlemoyer, and Xudong Wang. Reconstruction alignment improves unified multimodal models. In *ICLR*, 2026.

Ximing Xing, Juncheng Hu, Guotao Liang, Jing Zhang, Dong Xu, and Qian Yu. Empowering llms to understand and generate complex vector graphics. In *CVPR*, pp. 19487–19497, 2025.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, pp. 11975–11986, 2023.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *TPAMI*, 40(6):1452–1464, 2017.

Bocheng Zou, Mu Cai, Jianrui Zhang, and Yong Jae Lee. VGBench: Evaluating large language models on vector graphics understanding and generation. In *EMNLP*, pp. 3647–3659, 2024.

# A Related Work

## A.1 Symbolic Visual Representations

A growing line of work represents visual content as a discrete symbol sequence that a renderer executes, rather than as a dense array of pixels. On the understanding side, SGP-Bench evaluates whether language models can infer the semantics of SVG and CAD programs without seeing their renderings and improves this ability through Symbolic Instruction Tuning (Qiu et al., 2025), while VGBench and SVGenius broaden evaluation to editing and generation across many formats, domains, and complexity levels (Zou et al., 2024; Chen et al., 2025a). On the generation side, models emit SVG code or executable CAD programs directly (Chen et al., 2025b; Rodriguez et al., 2025; Xing et al., 2025; Wu et al., 2025; Khan et al., 2024; Rukhovich et al., 2025), and Sharma et al. (2024) use rendering code as a stand-in for pixels, demonstrating that text-only models encode substantial visual knowledge. Together, these studies show that general-purpose models encode usable geometric knowledge in symbolic form, although their performance degrades when precise spatial and structural fidelity is required.

Two properties separate that setting from ours. A program mediates between model and image, so the renderer supplies geometry and resolution independence and each primitive covers many pixels; token-based generative models instead keep the raster form but substitute a learned codebook whose vocabulary is opaque and whose decoding requires the paired detokenizer (Chen et al., 2020; Chameleon Team, 2024; Wang et al., 2026). PixelBench asks for an explicit palette and per-pixel indices, so rendering is a table lookup and every emitted symbol is one spatial location. The closest prior evaluation of this format is Pixel Art Bench, a web leaderboard that ranks $24 \times 24$ grids for seven subjects by LLM-judged Elo (Emergent Mind, 2026). Color-index grids also appear in abstract reasoning (Chollet, 2019). PixelBench broadens this to text-to-pixel and image-to-pixel generation over 100 concepts and 100 natural images at three resolutions, adds a complementary pixel-understanding evaluation, replaces judge-based ranking with deterministic retrieval and fidelity metrics, and compares against palette-matched pixelization and downscaling algorithms (Öztireli & Gross, 2015; Weber et al., 2016; Wu et al., 2022).

## A.2 Vision-Language Training and Visual Representations

Contemporary VLMs pair a pretrained visual encoder with a language model through a connector, aligned by image–text pretraining and visual instruction tuning (Radford et al., 2021; Zhai et al., 2023; Alayrac et al., 2022; Li et al., 2023; Liu et al., 2023; Dai et al., 2023; Liu et al., 2024a). Controlled ablations attribute much of the reported progress to the unimodal backbones and to training data rather than to the connector (Karamcheti et al., 2024; Laurençon et al., 2024), and a parallel body of work stresses the composition, quality, and cleanliness of that data: vision-centric mixtures sharpen fine-grained perception (Tong et al., 2024a), curated non-distilled and decontaminated corpora improve generalization (Deitke et al., 2025; Wiedmann et al., 2025), and benchmark audits show that apparent gains can stem from leakage or from questions answerable without the image (Chen et al., 2024). Across these pipelines the supervision target is language, whatever the visual input.

That choice has consequences: distinctions lost by the encoder propagate into the assembled model (Tong et al., 2024b), which remains unreliable on low-level spatial queries (Rahmanzadehgervi et al., 2024; Fu et al., 2024). Objectives that require reproducing visual detail instead yield transferable representations (He et al., 2022; Chen et al., 2020), and MetaMorph finds understanding and visual generation mutually beneficial (Tong et al., 2025). Reconstruction Alignment (RecA) conditions unified multimodal models on their own visual encoder representations and trains them to reconstruct the input image, thereby connecting visual understanding and generation (Xie et al., 2026). PRT studies an analogous objective for VLMs without a dedicated image generator: predicting the palette-index grid itself through the existing vocabulary, with no learned image tokenizer or decoder. Our primary additive protocol keeps the FineVision occurrences fixed and adds pixel reconstruction on top; Appendix D also includes a matched-data, equal-training-step control in which Pixel examples replace a small fraction of language supervision.

## B Additional Resolution-Scaling Examples

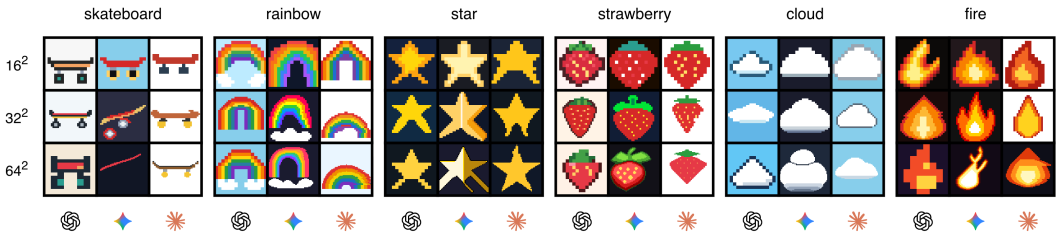
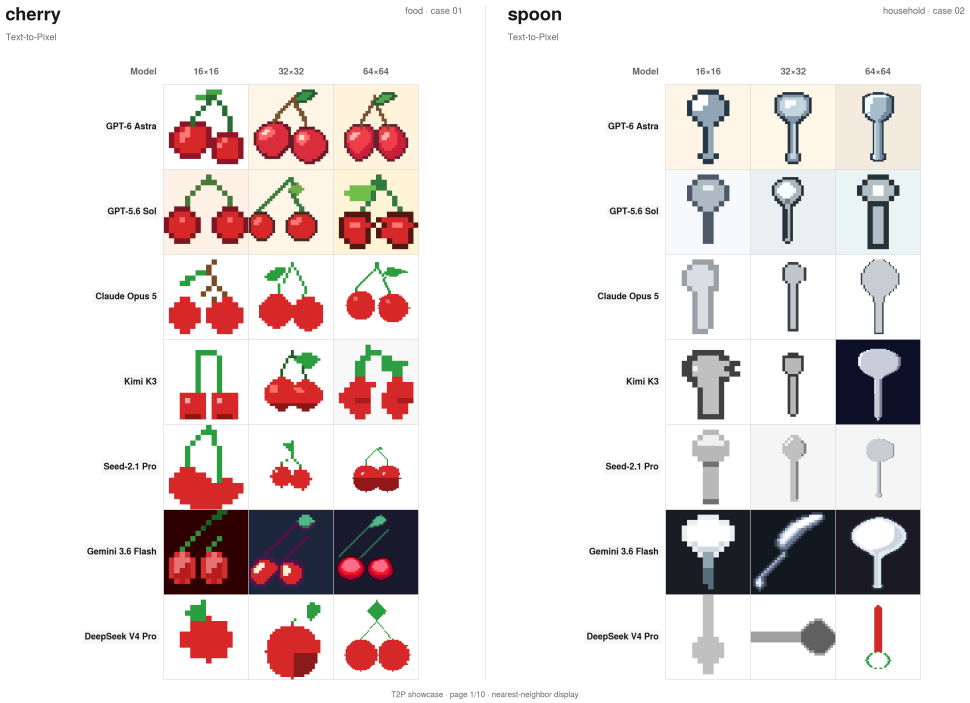


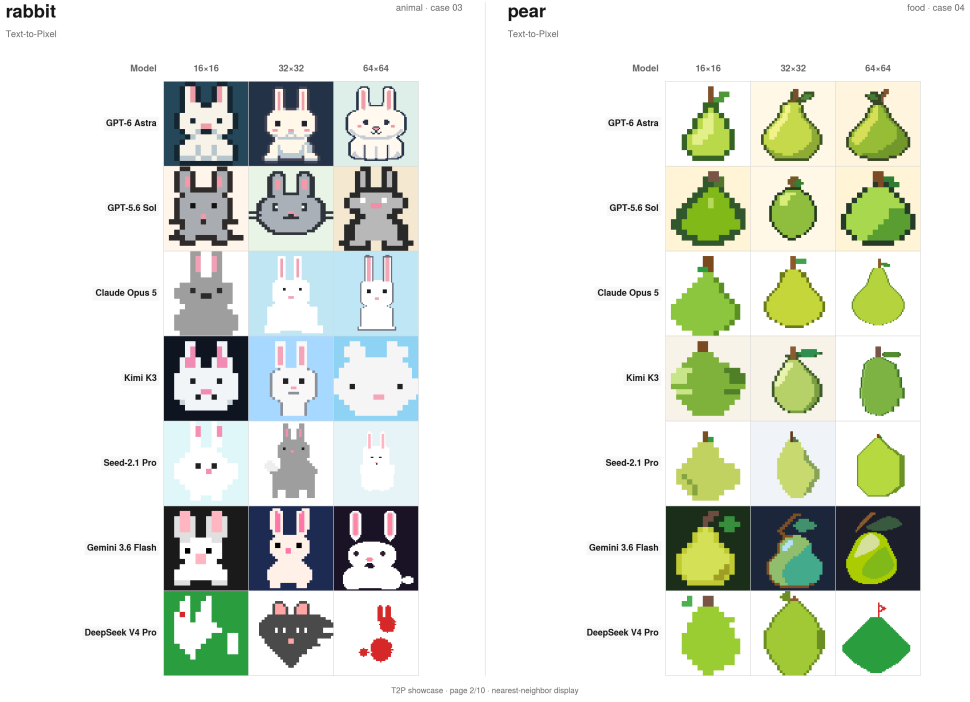

**Figure 6**: **Model-dependent resolution behavior in T2P.** Each case group shows GPT-5.6 Sol, Gemini 3.6 Flash, and Claude Opus 5 at $16^2$, $32^2$, and $64^2$. Higher nominal resolution can yield finer detail, coarse effective pixels, or degraded global structure depending on the model and prompt.

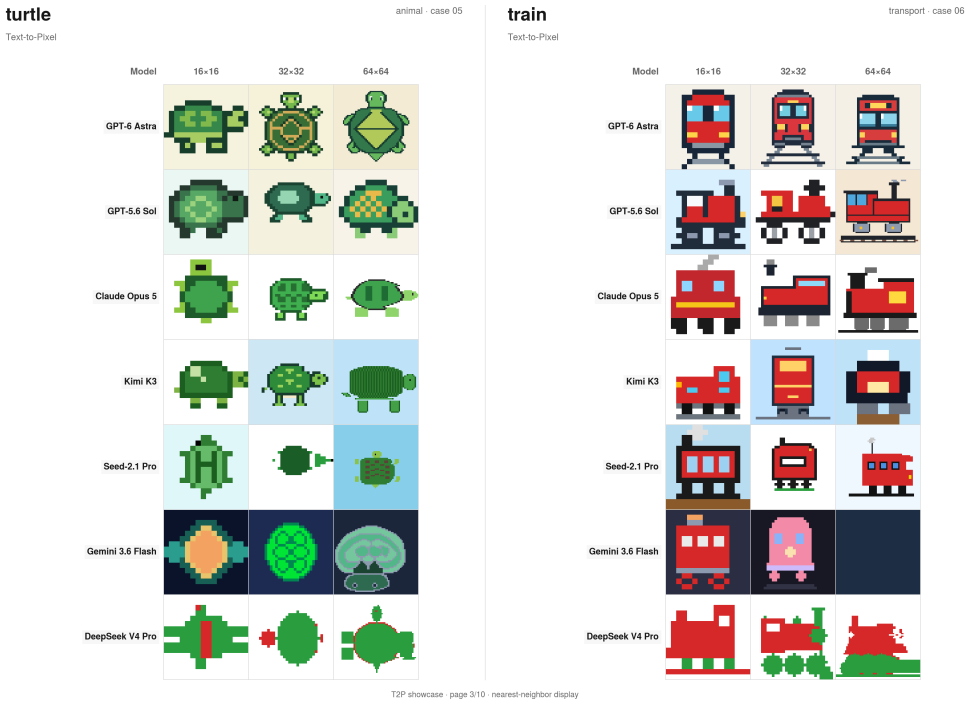
These cases instantiate the same three failure modes observed in the qualitative I2P sweep. GPT-5.6 Sol exhibits *effective-resolution underutilization*: its $64^2$ skateboard and fire are dominated by blocks roughly one-eighth of the canvas width, giving them a granularity closer to an $8 \times 8$ grid. The outputs remain recognizable, but the additional positions contribute little detail. Gemini 3.6 Flash uses finer boundaries and local variation, but these local patterns can disrupt global structure: at $64^2$, its skateboard collapses to a diagonal stroke, its star loses the five-point structure, and its fire loses a coherent silhouette. The latter cases also omit identity-bearing structures. Claude Opus 5 varies by prompt: it preserves the depicted objects in these examples while using the higher-resolution grid, but its aggregate T2P performance fluctuates across resolutions and does not sustain its intermediate-resolution peak.

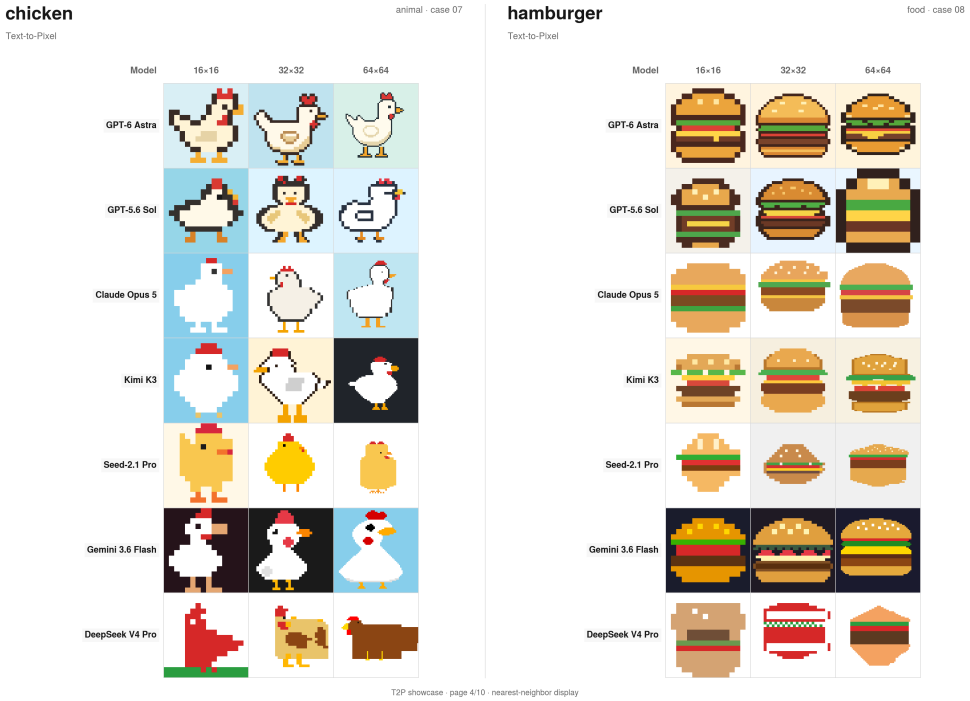
## C Additional Qualitative PixelBench Cases

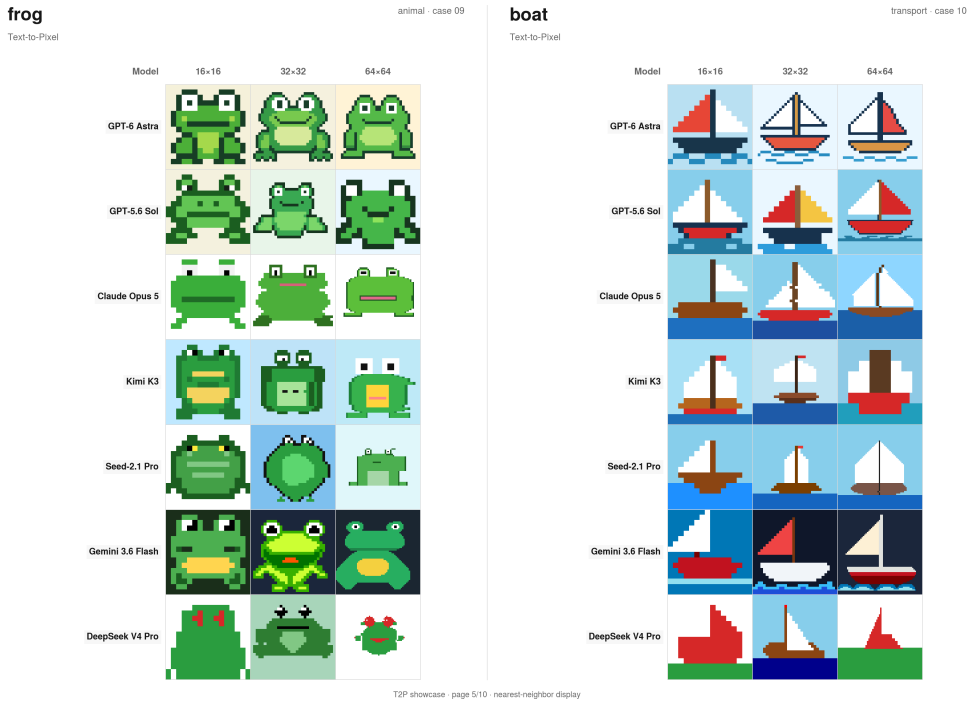
The following pages expand Figure 3 with 20 additional T2P cases and 20 additional I2P cases. Each T2P case compares all seven evaluated models at $16^2$, $32^2$, and $64^2$. Each I2P case compares the six evaluated models and three algorithmic baselines at the same resolutions. The I2P panels provide additional qualitative evidence that some model-based reconstructions preserve semantic structure more clearly than the algorithmic baselines at low resolutions, complementing the aggregate results in Table 2. Pixel images use nearest-neighbor enlargement for display.

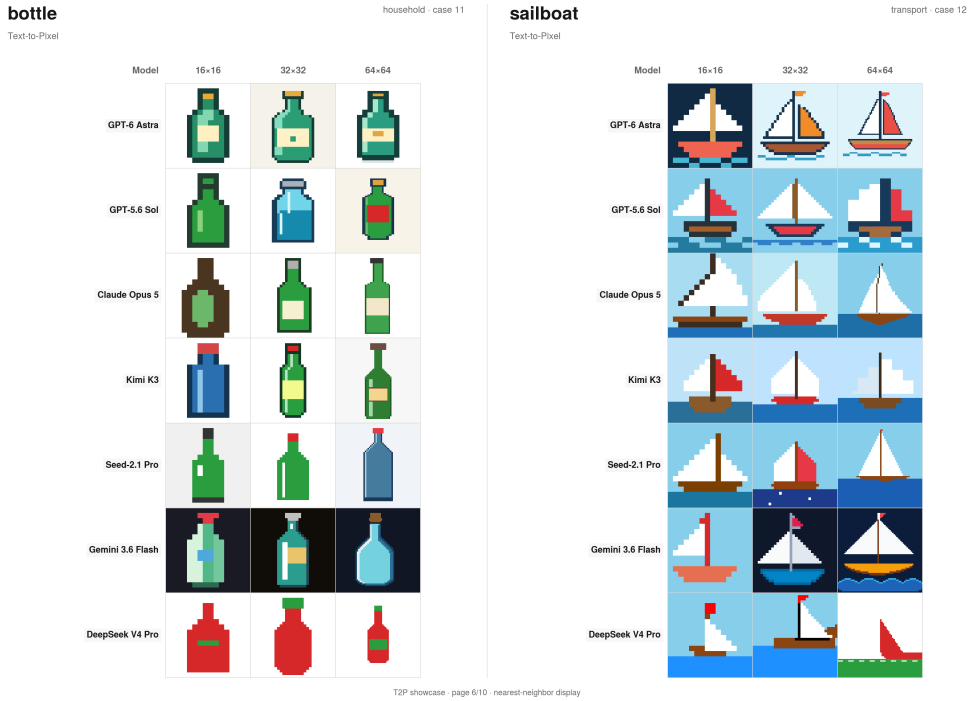

cherry
food · case 01
Text-to-Pixel
Model
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
DeepSeek V4 Pro
spoon
household · case 02
Text-to-Pixel
T2P showcase · page 1/10 · nearest-neighbor display

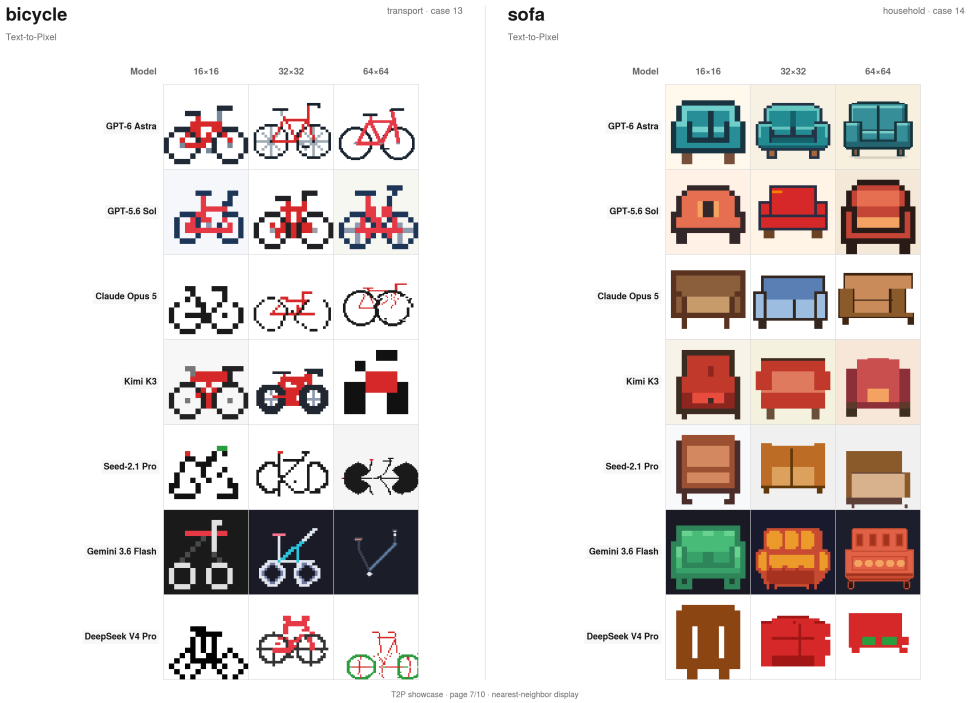

rabbit
animal · case 03
Text-to-Pixel
Model
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
DeepSeek V4 Pro
pear
food · case 04
Text-to-Pixel
T2P showcase · page 2/10 · nearest-neighbor display

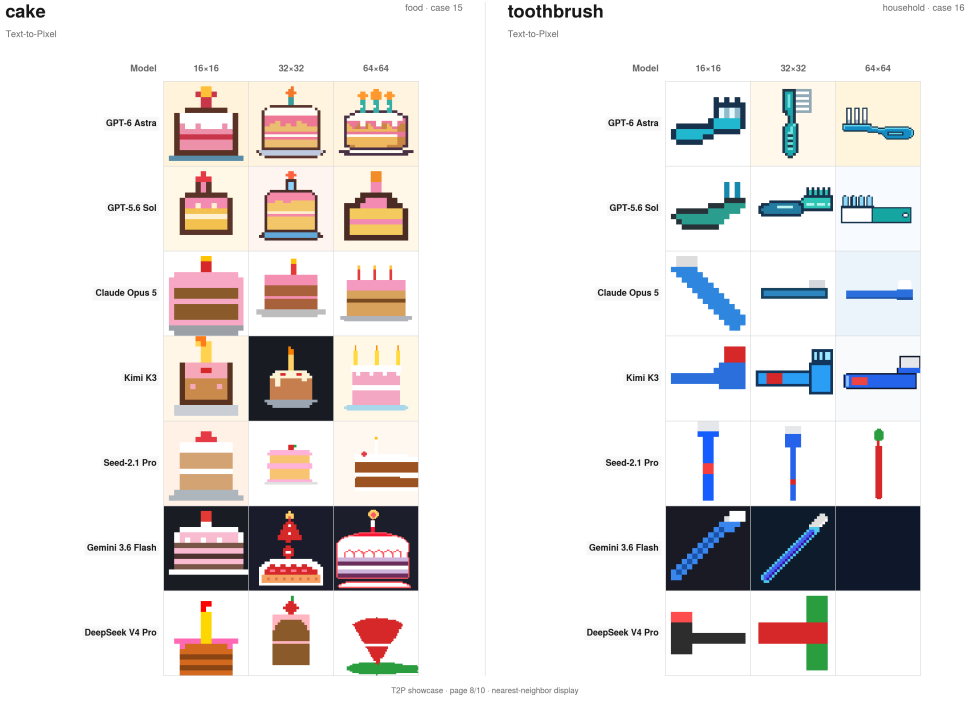

turtle
animal · case 05
Text-to-Pixel
Model
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
DeepSeek V4 Pro
train
transport · case 06
Text-to-Pixel
Model
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
DeepSeek V4 Pro
T2P showcase · page 3/10 · nearest-neighbor display

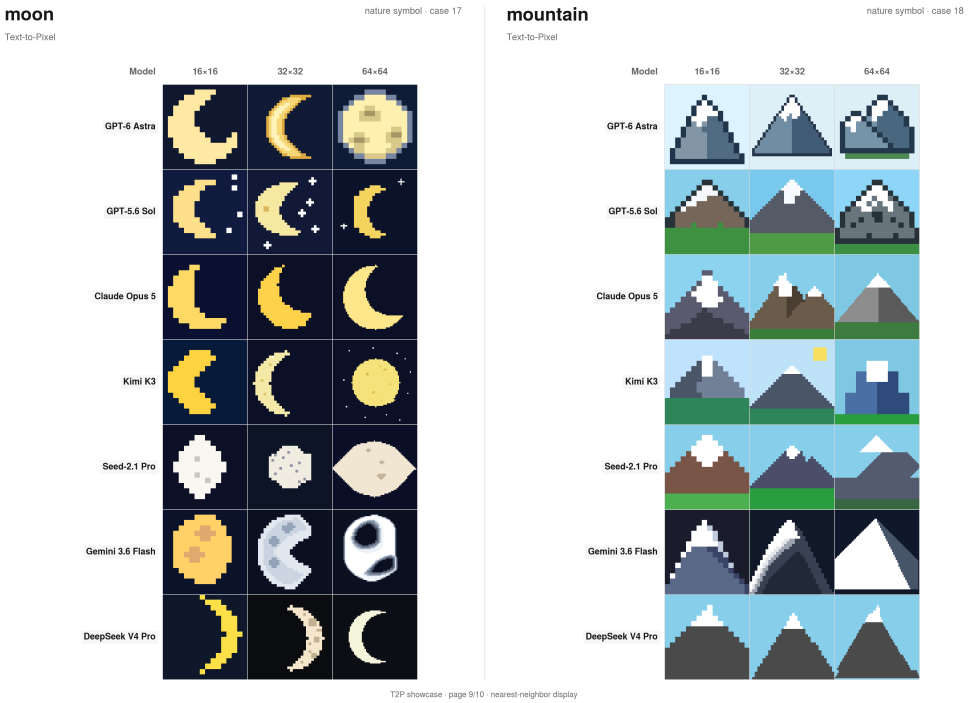

chicken
animal · case 07
Text-to-Pixel
Model
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
DeepSeek V4 Pro
hamburger
food · case 08
Text-to-Pixel
Model
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
DeepSeek V4 Pro
T2P showcase · page 4/10 · nearest-neighbor display

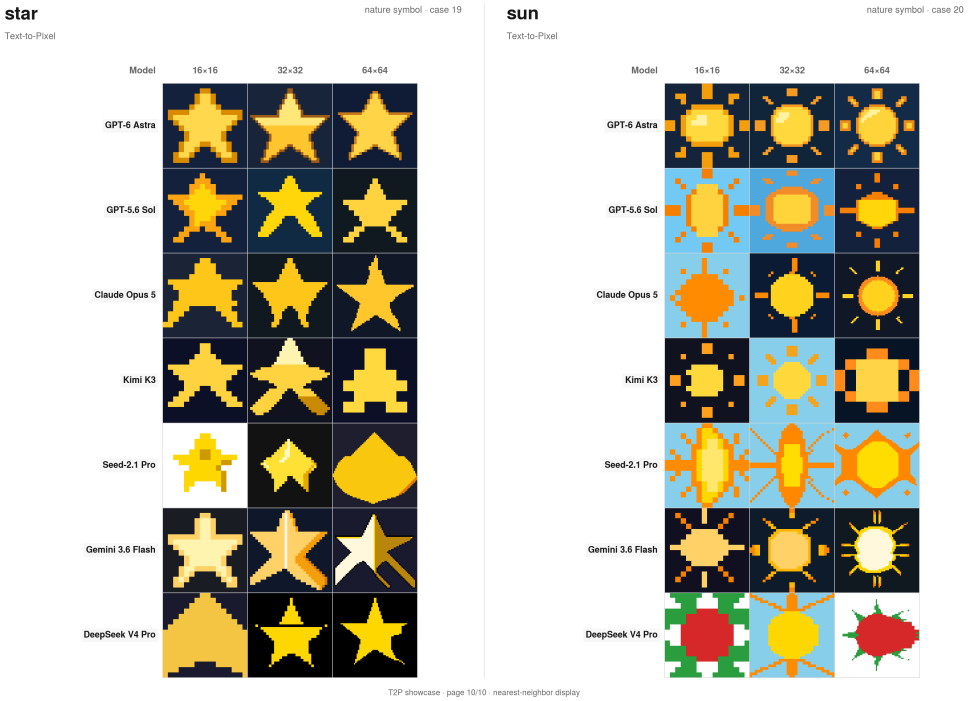

frog
animal · case 09
Text-to-Pixel
Model
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
DeepSeek V4 Pro
boat
transport · case 10
Text-to-Pixel
Model
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
DeepSeek V4 Pro
T2P showcase · page 5/10 · nearest-neighbor display

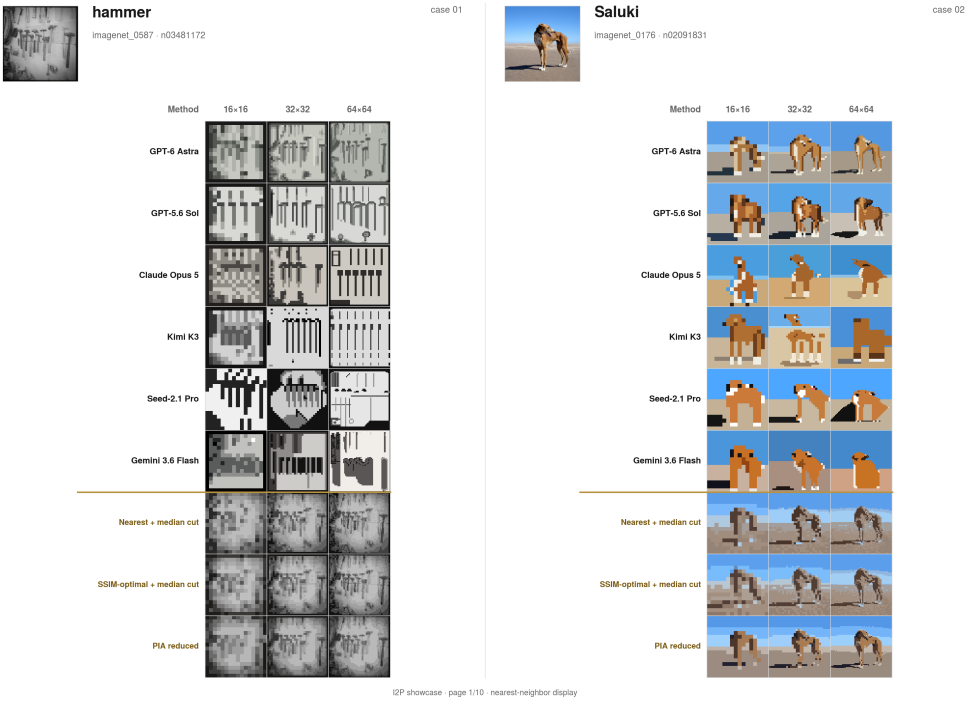

bottle
household · case 11
Text-to-Pixel
Model
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
DeepSeek V4 Pro
sailboat
transport · case 12
Text-to-Pixel
Model
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
DeepSeek V4 Pro
T2P showcase · page 6/10 · nearest-neighbor display

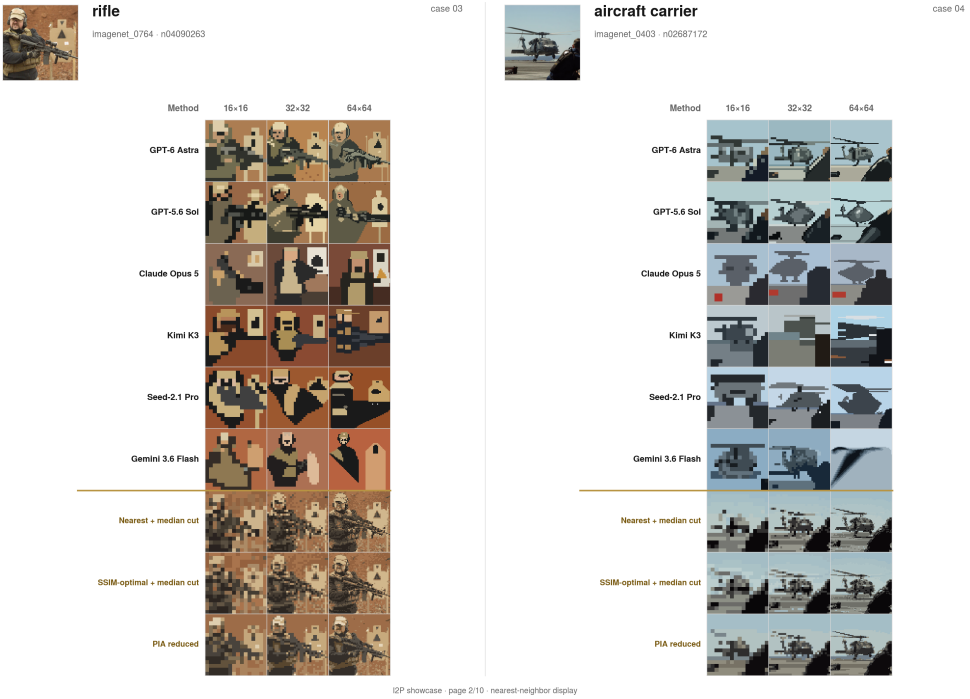

bicycle
transport · case 13
Text-to-Pixel
Model
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
DeepSeek V4 Pro
sofa
household · case 14
Text-to-Pixel
Model
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
DeepSeek V4 Pro
T2P showcase · page 7/10 · nearest-neighbor display

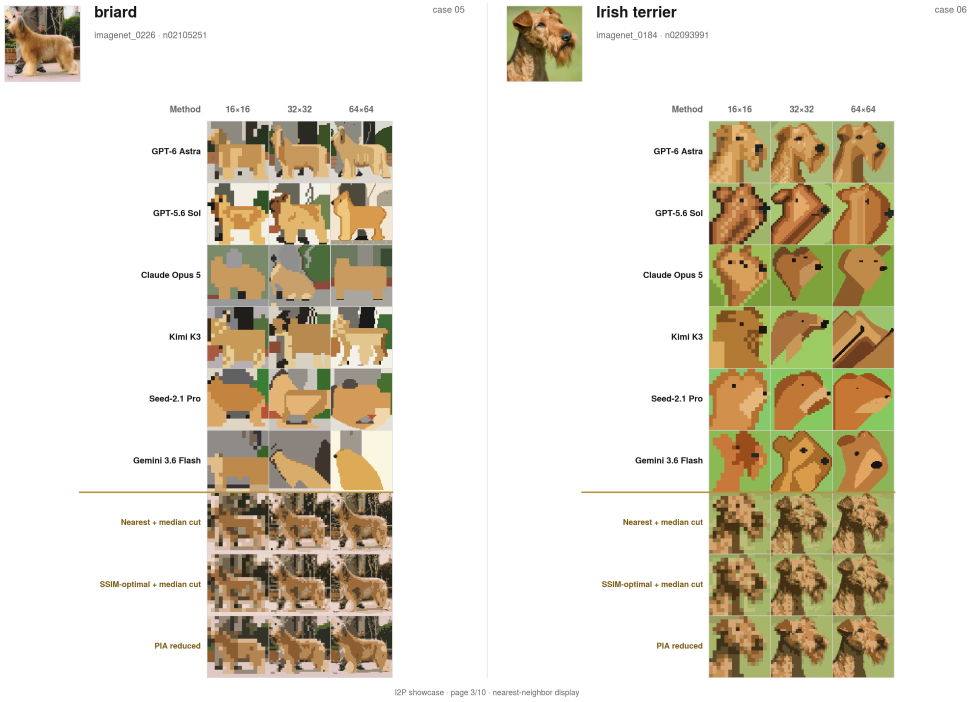

cake
food · case 15
Text-to-Pixel
Model
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
DeepSeek V4 Pro
toothbrush
household · case 16
Text-to-Pixel
Model
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
DeepSeek V4 Pro
T2P showcase · page 8/10 · nearest-neighbor display

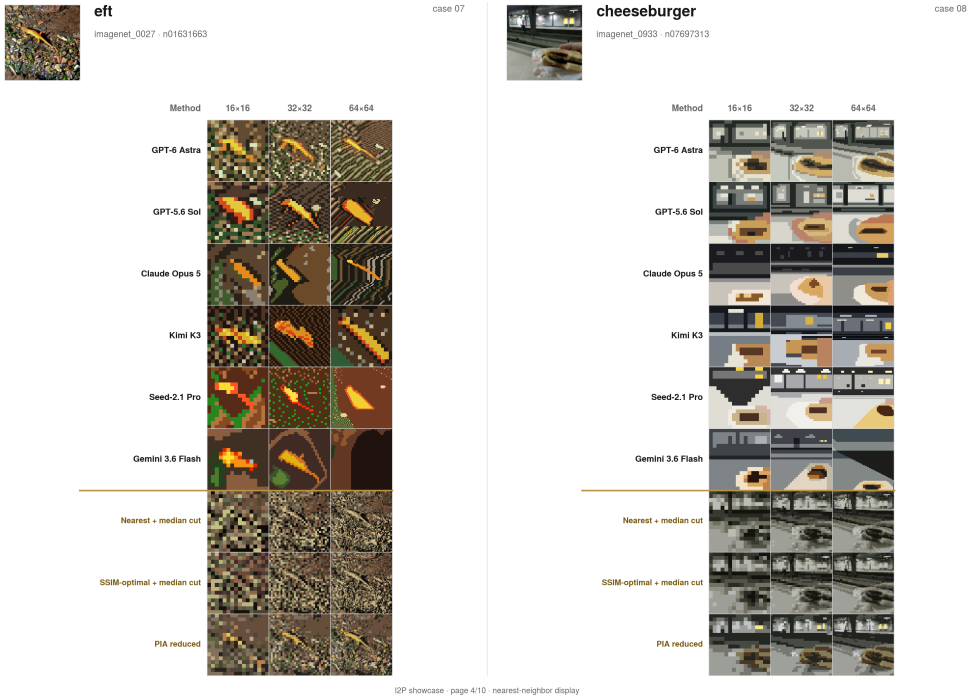

moon
nature symbol · case 17
Text-to-Pixel
Model
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
DeepSeek V4 Pro
mountain
nature symbol · case 18
Text-to-Pixel
Model
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
DeepSeek V4 Pro
T2P showcase · page 9/10 · nearest-neighbor display

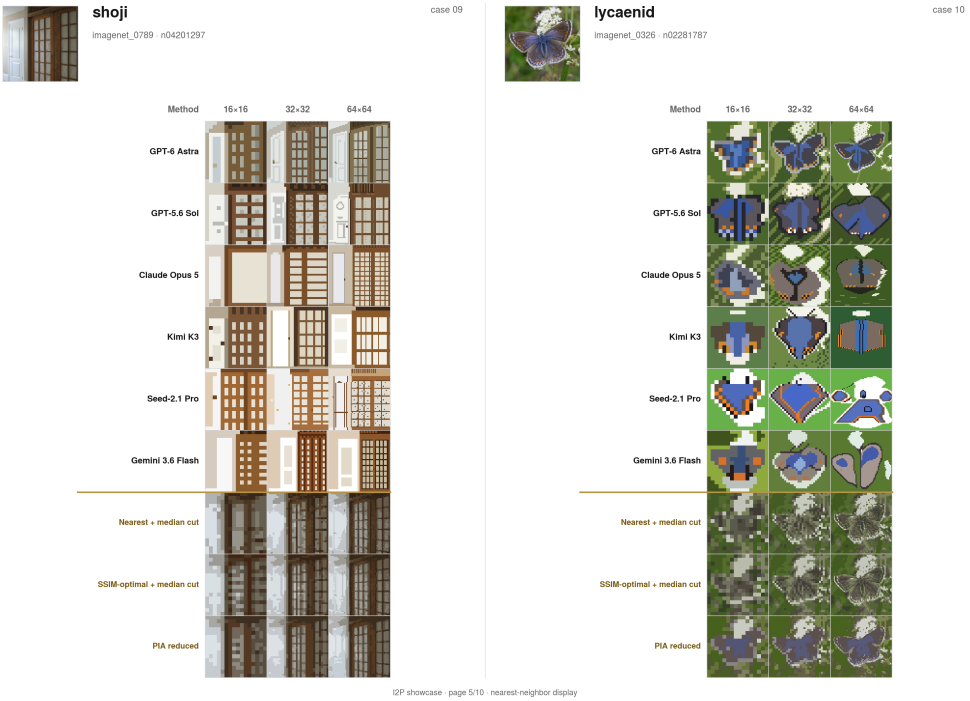

star
nature symbol · case 19
Text-to-Pixel
Model
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
DeepSeek V4 Pro
sun
nature symbol · case 20
Text-to-Pixel
Model
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
DeepSeek V4 Pro
T2P showcase · page 10/10 · nearest-neighbor display

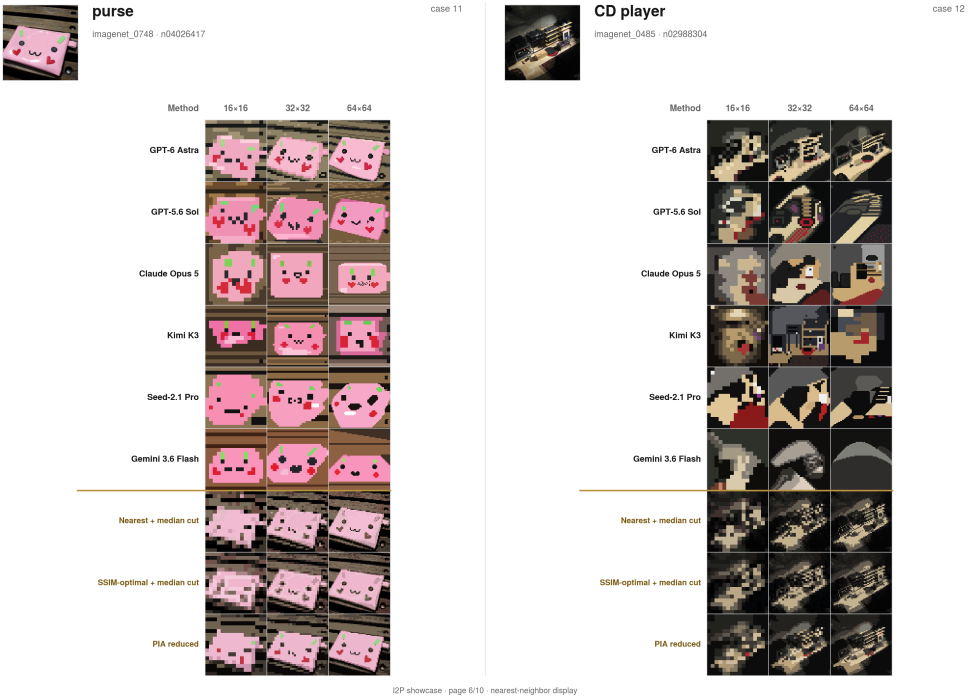

hammer
case 01
imagenet_0587 · n03481172
Saluki
case 02
imagenet_0176 · n02091831
Method
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
Nearest + median cut
SSIM-optimal + median cut
PIA reduced
I2P showcase · page 1/10 · nearest-neighbor display

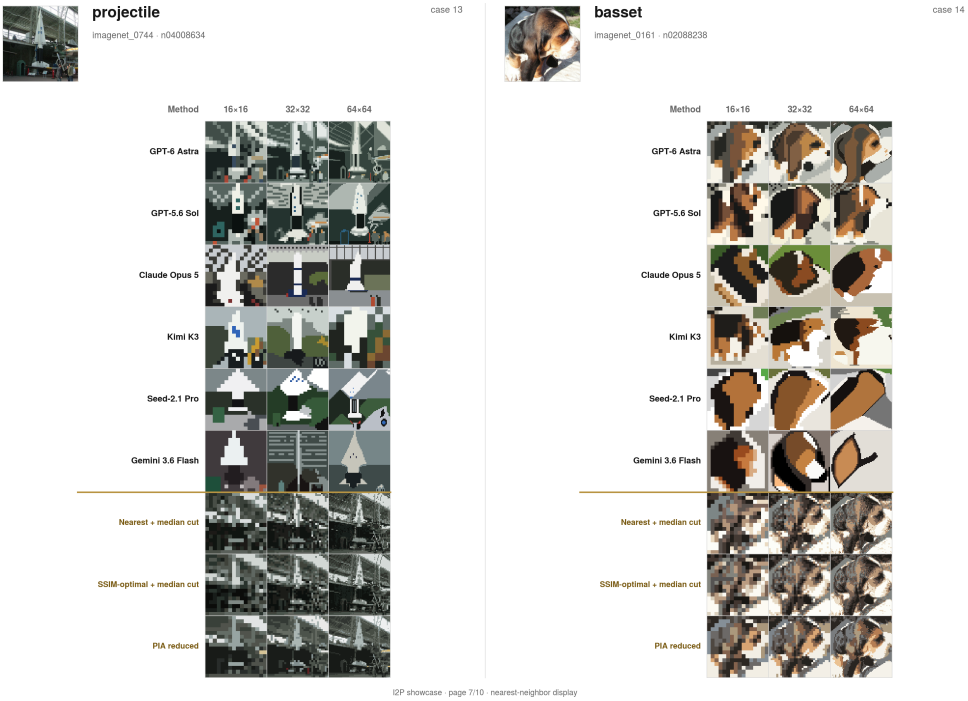

rifle
case 03
imagenet_0764 · n04090263
aircraft carrier
case 04
imagenet_0403 · n02687172
Method
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
Nearest + median cut
SSIM-optimal + median cut
PIA reduced
I2P showcase · page 2/10 · nearest-neighbor display

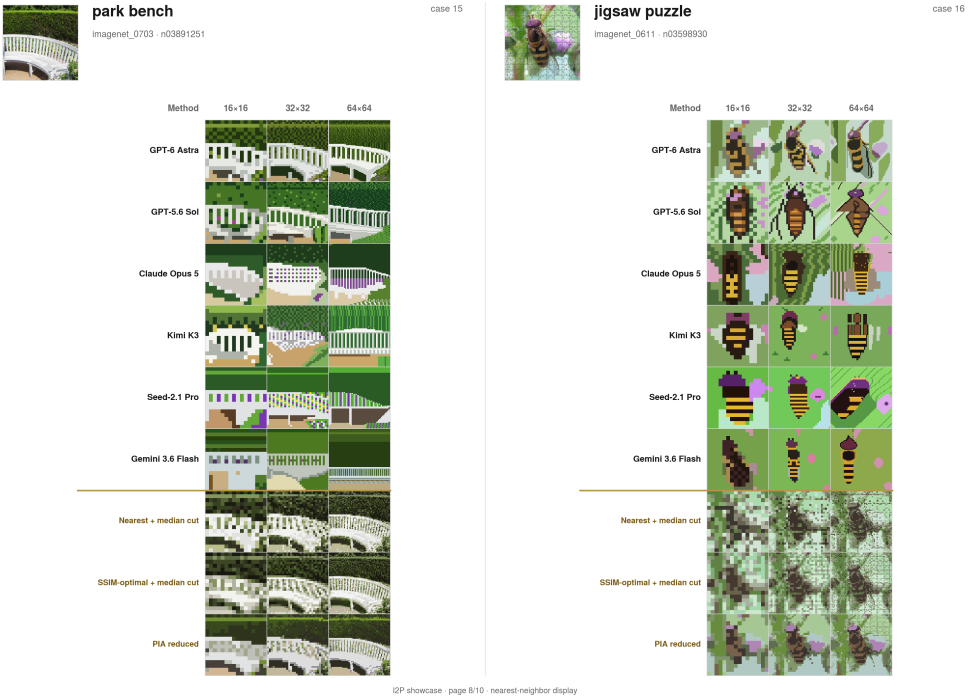

briard
imagenet_0226 · n02105251
case 05
Irish terrier
imagenet_0184 · n02093991
case 06
Method
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
Nearest + median cut
SSIM-optimal + median cut
PIA reduced
l2P showcase · page 3/10 · nearest-neighbor display

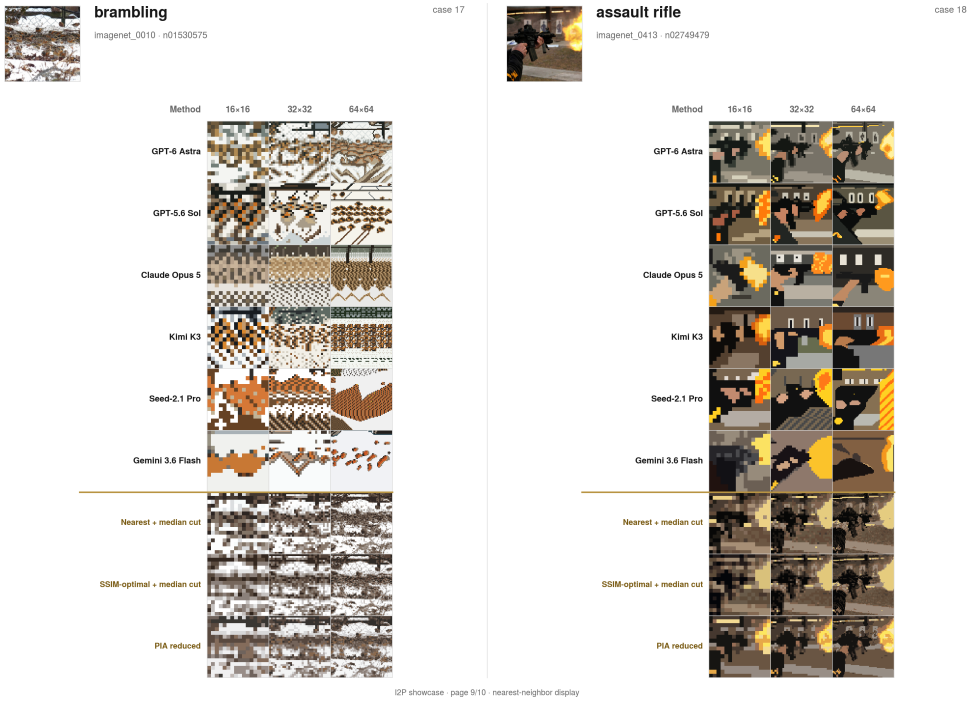

eft
imagenet_0027 · n01631663
case 07
cheeseburger
imagenet_0933 · n07697313
case 08
Method
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
Nearest + median cut
SSIM-optimal + median cut
PIA reduced
l2P showcase · page 4/10 · nearest-neighbor display

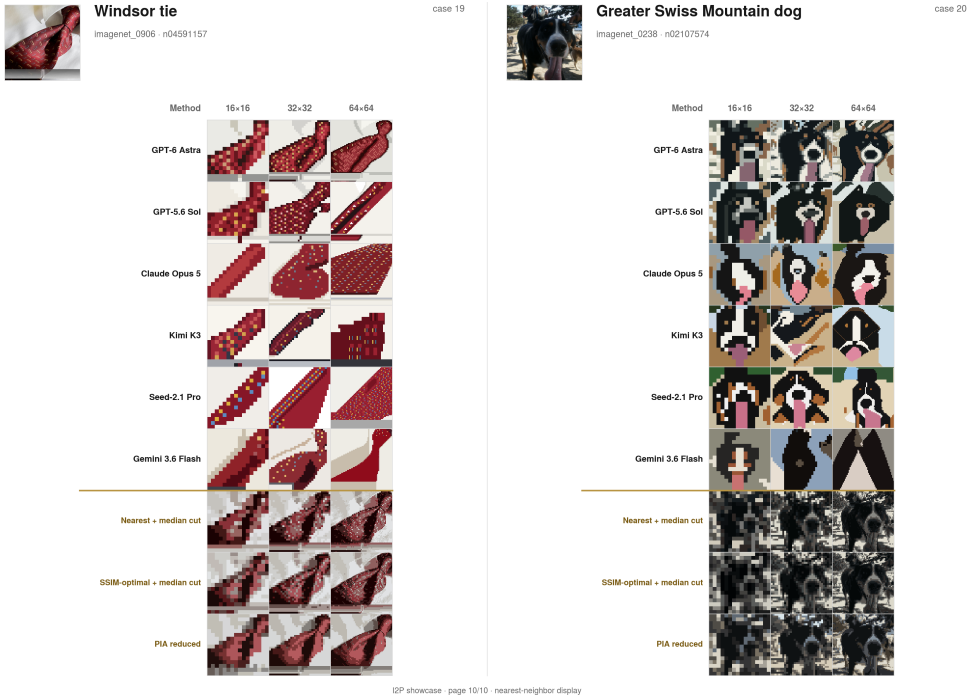

shoji
imagenet_0789 · n04201297
case 09
lycaenid
imagenet_0326 · n02281787
case 10
Method
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
Nearest + median cut
SSIM-optimal + median cut
PIA reduced
I2P showcase · page 5/10 · nearest-neighbor display


purse
imagenet_0748 · n04026417
case 11
CD player
imagenet_0485 · n02988304
case 12
Method
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
Nearest + median cut
SSIM-optimal + median cut
PIA reduced
I2P showcase · page 6/10 · nearest-neighbor display


projectile
case 13
imagenet_0744 · n04008634
basset
case 14
imagenet_0161 · n02088238
Method
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
Nearest + median cut
SSIM-optimal + median cut
PIA reduced
I2P showcase · page 7/10 · nearest-neighbor display


park bench
case 15
imagenet_0703 · n03891251
jigsaw puzzle
case 16
imagenet_0611 · n03598930
Method
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
Nearest + median cut
SSIM-optimal + median cut
PIA reduced
I2P showcase · page 8/10 · nearest-neighbor display


brambling
case 17
imagenet_0010 · n01530575
assault rifle
case 18
imagenet_0413 · n02749479
Method
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
Nearest + median cut
SSIM-optimal + median cut
PIA reduced
I2P showcase · page 9/10 · nearest-neighbor display


Windsor tie
case 19
imagenet_0906 · n04591157
Greater Swiss Mountain dog
case 20
imagenet_0238 · n02107574
Method
16×16
32×32
64×64
GPT-6 Astra
GPT-5.6 Sol
Claude Opus 5
Kimi K3
Seed-2.1 Pro
Gemini 3.6 Flash
Nearest + median cut
SSIM-optimal + median cut
PIA reduced
I2P showcase · page 10/10 · nearest-neighbor display

# D Additional Pixel Reconstruction Tuning Results

## D.1 Training Setup

**Data composition.** All PRT experiments initialize from Qwen3-VL-Instruct checkpoints at the 2B, 4B, and 8B scales; tables abbreviate these scales as 2B, 4B, and 8B. The primary experiments use an additive design. The FV-only baseline uses 64,000 FineVision occurrences. Each FV+Pixel configuration retains those occurrences and adds 3,000 pixel-reconstruction occurrences mixed approximately evenly across $16^2$, $32^2$, and $64^2$, for 67,000 total training examples. All examples are jointly shuffled. FineVision responses are capped at 2,048 tokens, and pixel targets use the deterministic eight-color PIA-reduced conversion. Pixel source images are sampled without replacement. In the Pixel-Mix run, each of the 3,000 unique source images is assigned to exactly one of $16^2$, $32^2$, or $64^2$. The single-resolution ablations deliberately reuse the same 3,000 source-image IDs across separate runs, converting each image once at the corresponding run's target resolution; no run therefore repeats a source image because of the resolution construction.

**Optimization.** We fully fine-tune all parameters, including the vision transformer, using AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$), zero weight decay, a peak learning rate of $10^{-5}$ with cosine decay and 3% warmup, gradient clipping at 1.0, bf16 precision, and training seed 42. The effective batch size is 64. FV-only trains for 1,000 optimizer steps, whereas FV+Pixel trains for 1,047 steps. Consequently, the primary delta measures the combined effect of adding Grid data and training for 47 additional steps.

## D.2 Additive FV + Pixel Results

Table 9 summarizes benchmark-level target-resolution deltas across model scales. Each panel is independently ordered by the mixed-resolution delta, from the largest gain to the largest drop.

**Table 9**: **Benchmark-level target-resolution deltas across model scales.** Each entry is the accuracy change, in percentage points, relative to the corresponding FineVision-only baseline under the additive protocol. Mix distributes Grid examples approximately evenly across the three resolutions. Panels are independently sorted by $\Delta_{\text{Mix}}$. Example averaging assigns equal weight to every official evaluation unit; Benchmark averaging assigns equal weight to every benchmark. * denotes a randomly sampled subset.

| 2B | $N$ | $\Delta_{16}$ | $\Delta_{32}$ | $\Delta_{64}$ | $\Delta_{\text{Mix}}$ |
|---|---|---|---|---|---|
| MMVP | 150 | +7.3 | +12.0 | +9.3 | +11.3 |
| BabyVision | 388 | +4.6 | +2.8 | +7.0 | +4.9 |
| MMBench | 1292 | +5.0 | +4.8 | +4.2 | +4.8 |
| CV-Bench | 2638 | +3.1 | +3.4 | +3.5 | +3.0 |
| TallyQA* | 3000 | +3.3 | +2.8 | +2.4 | +2.9 |
| Hallusion | 1129 | +5.1 | +1.2 | +3.8 | +2.8 |
| VStar | 191 | +1.0 | +3.1 | 0.0 | +2.6 |
| Q-Bench | 1495 | +1.8 | +3.7 | +3.1 | +2.1 |
| Perception | 3000 | +2.0 | +2.0 | +1.2 | +1.9 |
| BlindTest | 3434 | +3.6 | +1.3 | +3.9 | +1.7 |
| BLINK | 1901 | -0.7 | +1.5 | +0.3 | +1.6 |
| MMStar | 1500 | -0.5 | +3.3 | +0.9 | +0.3 |
| VLMBias | 2784 | -1.1 | +1.9 | -0.6 | -0.8 |
| Example | 22902 | +2.14 | +2.50 | +2.27 | +2.01 |
| Benchmark | 13 | +2.66 | +3.37 | +2.99 | +3.02 |

| 4B | $N$ | $\Delta_{16}$ | $\Delta_{32}$ | $\Delta_{64}$ | $\Delta_{\text{Mix}}$ |
|---|---|---|---|---|---|
| MMVP | 150 | +18.0 | +22.7 | +23.3 | +20.7 |
| BlindTest | 3434 | +10.7 | +10.6 | +7.3 | +9.6 |
| BabyVision | 388 | +2.6 | +3.9 | +3.6 | +5.2 |
| CV-Bench | 2638 | +3.4 | +3.2 | +2.9 | +3.0 |
| VLMBias | 2784 | -0.9 | +1.1 | +0.2 | +2.7 |
| MMStar | 1500 | +1.6 | +3.4 | +0.4 | +2.3 |
| MMBench | 1292 | +3.6 | +3.3 | +2.6 | +2.2 |
| Perception | 3000 | +2.0 | +1.2 | +1.6 | +1.5 |
| BLINK | 1901 | -0.6 | -0.1 | +1.4 | +1.3 |
| Q-Bench | 1495 | +0.4 | -1.1 | +0.3 | +0.8 |
| TallyQA* | 3000 | +2.0 | +0.8 | +0.3 | +0.6 |
| VStar | 191 | +5.2 | +2.6 | +2.6 | +0.5 |
| Hallusion | 1129 | +4.9 | -0.8 | -1.5 | -2.1 |
| Example | 22902 | +3.14 | +2.87 | +2.18 | +2.94 |
| Benchmark | 13 | +4.07 | +3.90 | +3.48 | +3.70 |

| 8B | $N$ | $\Delta_{16}$ | $\Delta_{32}$ | $\Delta_{64}$ | $\Delta_{\text{Mix}}$ |
|---|---|---|---|---|---|
| BlindTest | 3434 | +4.6 | +4.3 | +5.2 | +7.8 |
| MMVP | 150 | +0.7 | +0.7 | +8.0 | +4.0 |
| Q-Bench | 1495 | +3.0 | +4.3 | +2.5 | +2.6 |
| TallyQA* | 3000 | +2.8 | +3.3 | +2.5 | +2.6 |
| MMStar | 1500 | +4.5 | +2.3 | +4.0 | +2.3 |
| BabyVision | 388 | -1.8 | +1.5 | 0.0 | +2.1 |
| BLINK | 1901 | +2.9 | +0.8 | +2.4 | +1.7 |
| MMBench | 1292 | +3.4 | +1.6 | +1.5 | +1.6 |
| Perception | 3000 | +1.4 | +2.0 | +1.9 | +1.3 |
| CV-Bench | 2638 | -0.3 | +0.8 | -0.2 | +1.0 |
| VStar | 191 | +1.6 | -0.5 | 0.0 | -1.0 |
| VLMBias | 2784 | +0.1 | +1.8 | +0.9 | -1.1 |
| Hallusion | 1129 | -4.9 | -7.0 | -6.1 | -4.1 |
| Example | 22902 | +1.88 | +1.92 | +1.90 | +2.07 |
| Benchmark | 13 | +1.38 | +1.23 | +1.73 | +1.60 |

## D.3 Fixed Data Amount and Training Steps

To compare reconstruction tuning under matched training conditions, we fix both the amount of training data and the number of updates. FV-only contains 64,000 FineVision occurrences. Each Pixel configuration replaces 3,000 of those occurrences with pixel-reconstruction examples, so every fine-tuning run uses exactly 64,000 occurrences and stops at 1,000 optimizer steps. The Pixel configurations therefore contain less FineVision supervision.

The single-resolution variants use eight-color PIA-reduced targets at $16^2$, $32^2$, or $64^2$ in separate runs, reusing the same 3,000 unique source-image IDs across runs. The mixed variant assigns each image from that batch to exactly one resolution, approximately evenly, using sampling seed 0. Source-image sampling is without replacement, so no image is repeated within a run. All fine-tuning runs use training seed 42, four GPUs with DeepSpeed ZeRO-2, per-GPU batch size 4, and

four gradient accumulation steps; the 4B runs use NVIDIA A100 80GB GPUs. These settings match the number of training occurrences and optimizer steps; target sequence length still grows with grid size.

**Training-seed sensitivity.** For the 4B setting with 4,000 Pixel occurrences, all three seeds yield positive, tightly clustered deltas (Table 11; SD 0.150 Micro and 0.197 Macro). This auxiliary study uses the original MMVP scoring protocol, so the values differ from the main PRT tables; all runs and baselines here use the same evaluation protocol.

**Table 10**: **Fixed-data-amount, fixed-step comparison.** Every run uses 64,000 training occurrences and 1,000 optimizer steps. Values are Pixel-minus-FV-only accuracy deltas; Example pools 22,554 official evaluation units after MMVP pair-level and MMBench CircularEval aggregation, while Benchmark weights 13 benchmarks equally.

| Model | Setting | Δ Example | Δ Benchmark |
|---|---|---|---|
| 2B | Pixel-16 | **+3.152** | **+4.278** |
| | Pixel-32 | +1.534 | +2.481 |
| | Pixel-64 | +2.833 | <u>+3.744</u> |
| | Pixel-Mix | <u>+2.966</u> | +3.577 |
| 4B | Pixel-16 | **+2.270** | **+3.129** |
| | Pixel-32 | <u>+1.335</u> | +1.828 |
| | Pixel-64 | +1.171 | <u>+2.306</u> |
| | Pixel-Mix | +1.134 | +1.993 |

**Table 11**: **Seed sensitivity (4B, 4,000 Pixel occurrences).** Values are Pixel-minus-FV-only accuracy deltas; Micro pools examples and Macro weights benchmarks equally. Summary statistics are computed across three seeds.

| Run | Δ Micro | Δ Macro |
|---|---|---|
| Seed 42 | +1.455 | +1.789 |
| Seed 43 | +1.746 | +2.053 |
| Seed 44 | +1.534 | +1.668 |
| Mean ± SD | +1.578 ± 0.150 | +1.837 ± 0.197 |
| SEM | 0.087 | 0.114 |
| 95% CI | [1.205, 1.952] | [1.348, 2.326] |

Under the official aggregation, Pixel-16 yields the largest gains at both model scales, and every setting—including 4B Pixel-Mix—improves both averages.

### D.4 Limitations

Our controls compare PRT against FineVision-only fine-tuning; we do not compare against alternative auxiliary objectives on the same images, such as captioning. We therefore cannot fully separate the benefit of the pixel target from that of additional image exposure. Pixel targets are also longer than typical responses, so their share of training tokens exceeds the 3k/67k example ratio.

## E Model Inference Settings

Table 12 reports the task-specific inference settings used for PixelBench generation. We retain one completed output per item. A completed response that violates the output constraints is a model error and receives zero, whereas provider-side failures and token-limit truncations are retried; for the latter, we raise the output-token limit when needed. Seed-2.1 Pro is the only model for which we impose an explicit reasoning budget. We cap it at 2,500 tokens because pilot runs frequently produced long, repetitive deliberation without improving the final pixel output.

**Table 12**: **Model inference settings for PixelBench generation.** Temperatures and reasoning controls are task-specific. "Default" denotes the provider default at the time of evaluation. The listed output-token limit is the nominal setting and is raised only when retrying a truncated response.

| Model | Temperature | Reasoning configuration | Max output tokens | Samples/item |
|---|---|---|---|---|
| *Text-to-Pixel (T2P)* | | | | |
| GPT-6 Astra | 1.0 | Default | 32,768 | 1 |
| Claude Opus 5 | 1.0 | Default (high effort) | 32,768 | 1 |
| GPT-5.6 Sol | 1.0 | High effort | 32,768 | 1 |
| Kimi K3 | 1.0 | High effort | 32,768 | 1 |
| Gemini 3.6 Flash | 0.7 | Default | 32,768 | 1 |
| Seed-2.1 Pro | 0.7 | Reasoning budget 2,500 | 32,768 | 1 |
| DeepSeek V4 Pro | 0.7 | Default | 32,768 | 1 |
| *Image-to-Pixel (I2P)* | | | | |
| GPT-6 Astra | 1.0 | Default | 32,768 | 1 |
| Claude Opus 5 | 1.0 | Default (high effort) | 32,768 | 1 |
| GPT-5.6 Sol | 1.0 | High effort | 32,768 | 1 |
| Kimi K3 | 1.0 | High effort | 32,768 | 1 |
| Gemini 3.6 Flash | 0.7 | Default | 32,768 | 1 |
| Seed-2.1 Pro | 0.7 | Reasoning budget 2,500 | 32,768 | 1 |

# F SigLIP Retrieval Scores

Let $\hat{x}$ denote a rendered pixel image and let $\mathbf{q}$ be its SigLIP embedding. Given a candidate set $\mathcal{C} = \{\mathbf{c}_i\}_{i=1}^{K}$ of the same embedding space, we score each candidate by cosine similarity,

$$s(\mathbf{q}, \mathbf{c}_i) = \frac{\mathbf{q}^\top \mathbf{c}_i}{\|\mathbf{q}\| \, \|\mathbf{c}_i\|}.$$

The rank of the intended target $\mathbf{c}^\star$ is

$$r(\hat{x}) = 1 + \sum_{\mathbf{c} \in \mathcal{C} \setminus \{\mathbf{c}^\star\}} \mathbf{1}[s(\mathbf{q}, \mathbf{c}) > s(\mathbf{q}, \mathbf{c}^\star)] \,,$$

so $r(\hat{x}) = 1$ if and only if the target is the nearest neighbor of the query. Let $\mathcal{V}$ index the valid outputs among $N$ attempted generations. The reported *100-way Top-1* or *500-way Top-1* score counts invalid outputs as misses:

$$\text{Top-1} = \frac{1}{N} \sum_{n \in \mathcal{V}} \mathbf{1}[r(\hat{x}_n) = 1] \,.$$

The appendix MRR diagnostic is averaged over valid outputs:

$$\text{MRR} = \frac{1}{|\mathcal{V}|} \sum_{n \in \mathcal{V}} \frac{1}{r(\hat{x}_n)}.$$

For T2P, $\mathbf{q}$ comes from the SigLIP-SO400M image encoder and each $\mathbf{c}_i$ is the corresponding text embedding of "a pixel art icon depicting {label}"; we report $K = 100$ and $K = 500$. For I2P, both the query and candidates are image embeddings from the same SigLIP-SO400M model, and $K = 100$ is the set of benchmark source images.

# G Full Evaluation Prompts

The fields `label` (T2P) and `sample_id` (I2P) are included only to simplify automated parsing and pairing during evaluation; they are not part of the visual generation objective. For pixel understanding, {`compact_pixel_json`} is the serialized palette-and-grid input, and {`choice_1`}–{`choice_10`} are the ten candidate labels. For VLM judging, {`label`} and {`size`} are replaced with the target concept and native grid size, and the rendered image is supplied.

## G.1 Text-to-Pixel Prompt

```
Create exactly N recognizable SxS pixel-art icons, one for each of these English labels in
this exact order:
{labels}

Return JSON only, with no Markdown fence or commentary, using:
{"images": [{"label": "{label}",
 "palette": ["#FFFFFF", "#D62828", "#2A9D3F"],
 "pixels": ["000...", "... exactly S strings"]}]}

Requirements:
- Every icon must have exactly S rows and S columns.
- Use 2 to 8 opaque RGB colors, including a background color.
- Encode each pixel as one digit from 0 to 7.
- Favor simple silhouettes, high contrast, and centered subjects.
- Make each object recognizable at SxS resolution; use no text.
```

### G.2 TEXT-TO-PIXEL VLM-AS-JUDGE PROMPT

**Judge inference and stability.** We use Gemini 3.5 Flash with temperature 0, thinking budget 0, a maximum of 1,536 output tokens, and one sample per item. As a stability check, we re-score a 10% subset after the initial evaluation; the mean absolute change in total score is below 2 points on the 100-point scale, indicating that the reported VLM Judge scores are robust to repeated evaluation.

```
Act as a strict evaluator of a text-to-pixel generation. The target concept is
"{label}". The supplied image is a nearest-neighbor enlargement of a
native {size}x{size} pixel-art image. Large square pixels are intentional.

Score the image using all six independent dimensions below. Do not reward
complexity by itself, do not infer the generator identity, and do not excuse a
wrong or unrecognizable subject merely because the image is attractive.

1. subject_identity (0-20): Does the visible subject clearly depict the target
   concept rather than a related, ambiguous, or incorrect object?
2. defining_attributes (0-25): Are the target's identity-bearing parts, shapes,
   and relations present and correct?
3. structure_proportion (0-20): Is the silhouette coherent, proportionate,
   well framed, and structurally correct rather than cropped or merged?
4. detail_quality (0-20): Are internal details coherent, informative, and
   consistent with the target at the native resolution?
5. pixel_art_execution (0-10): Are pixel clusters and edges deliberate
   and appropriate for the native resolution, without random noise or
   fragmented pseudo-texture?
6. palette_contrast (0-5): Is the limited palette coherent and does contrast
   preserve the subject and its important parts?

Return JSON only, without Markdown:
{
  "observed_subject": "short literal description of what is visibly depicted",
  "subject_identity": {"score": 0, "reason": "brief evidence"},
  "defining_attributes": {"score": 0, "reason": "brief evidence"},
  "structure_proportion": {"score": 0, "reason": "brief evidence"},
  "detail_quality": {"score": 0, "reason": "brief evidence"},
  "pixel_art_execution": {"score": 0, "reason": "brief evidence"},
  "palette_contrast": {"score": 0, "reason": "brief evidence"},
  "confidence": "low|medium|high",
  "failure_flags": []
}
```

### G.3 IMAGE-TO-PIXEL PROMPT

```
Convert the supplied reference image into a faithful SxS pixel-art image. Preserve the main
subject's identity, silhouette, pose, composition, and dominant colors. Simplify fine
detail only as required by the pixel resolution.
Do not use any object name or text.

Return JSON only:
{"sample_id": "{sample_id}",
 "palette": ["#FFFFFF", "#111111"],
 "pixels": ["000...", "... exactly S strings"]}

Use 2 to 8 opaque #RRGGBB colors.
```

### G.4 Pixel Understanding Prompt

**Candidate construction.** For each question, we exclude the correct concept from the benchmark's 100 object concepts, deterministically shuffle the remaining 99, and select the first nine as distractors. We then deterministically reshuffle the correct answer together with these nine distractors to assign the ten choices to A–J. Each item therefore contains ten distinct candidates with exactly one correct answer, and the construction is reproducible.

```
This is a {size}x{size} pixel-art icon encoded as JSON.
"palette" is 2-8 hex colors. "pixels" are {size} row strings; each character is a palette
index 0-7.
Identify the depicted object. Reply with ONE letter only (A, B, C, D, E, F, G, H, I, J).

{compact_pixel_json}

A. {choice_1}
B. {choice_2}
C. {choice_3}
D. {choice_4}
E. {choice_5}
F. {choice_6}
G. {choice_7}
H. {choice_8}
I. {choice_9}
J. {choice_10}
```

## H Text-to-Pixel Concept List

Table 13 lists the 100 T2P concepts in the order used by the benchmark. The set contains 18 animals, 20 food items, 30 household objects, 15 means of transport, and 17 nature-related objects or symbols.

**Table 13**: **Complete Text-to-Pixel concept list.** Numeric identifiers preserve the original benchmark order. Background colors denote animal, food, household, transport, and nature/symbol.

| | | | | |
|---|---|---|---|---|
| 4. cat | 8. fish | 17. dog | 18. bird | 19. rabbit |
| 20. elephant | 21. turtle | 22. butterfly | 51. cow | 52. sheep |
| 53. pig | 54. chicken | 55. owl | 56. penguin | 57. frog |
| 58. snake | 59. bee | 60. snail | 1. apple | 3. banana |
| 23. orange | 24. strawberry | 25. watermelon | 26. carrot | 27. pizza |
| 28. cake | 29. bread | 30. ice cream | 61. pear | 62. pineapple |
| 63. grapes | 64. cherry | 65. lemon | 66. corn | 67. donut |
| 68. hamburger | 69. sandwich | 70. cookie | 2. stool | 5. house |
| 10. cup | 14. key | 15. umbrella | 16. clock | 31. lamp |
| 32. spoon | 33. scissors | 34. toothbrush | 35. bottle | 36. camera |
| 37. phone | 38. book | 39. hammer | 71. fork | 72. knife |
| 73. plate | 74. bowl | 75. bed | 76. table | 77. sofa |
| 78. television | 79. computer | 80. backpack | 82. hat | 89. wrench |
| 90. screwdriver | 91. axe | 92. shovel | 7. car | 40. bus |
| 41. bicycle | 42. airplane | 43. boat | 44. train | 45. rocket |
| 46. truck | 81. shoe | 83. motorcycle | 84. helicopter | 85. submarine |
| 86. tractor | 87. skateboard | 88. sailboat | 6. tree | 9. flower |
| 11. star | 12. heart | 13. mushroom | 47. sun | 48. moon |
| 49. cloud | 50. mountain | 93. rainbow | 94. snowflake | 95. lightning |
| 96. fire | 97. leaf | 98. cactus | 99. palm tree | 100. castle |

## I Image-to-Pixel Source Images

Figure 7 shows the 100 ImageNet validation crops used as I2P inputs. Each class contributes one center-cropped $256 \times 256$ image; labels are shown for reference and are not provided to the model during generation.

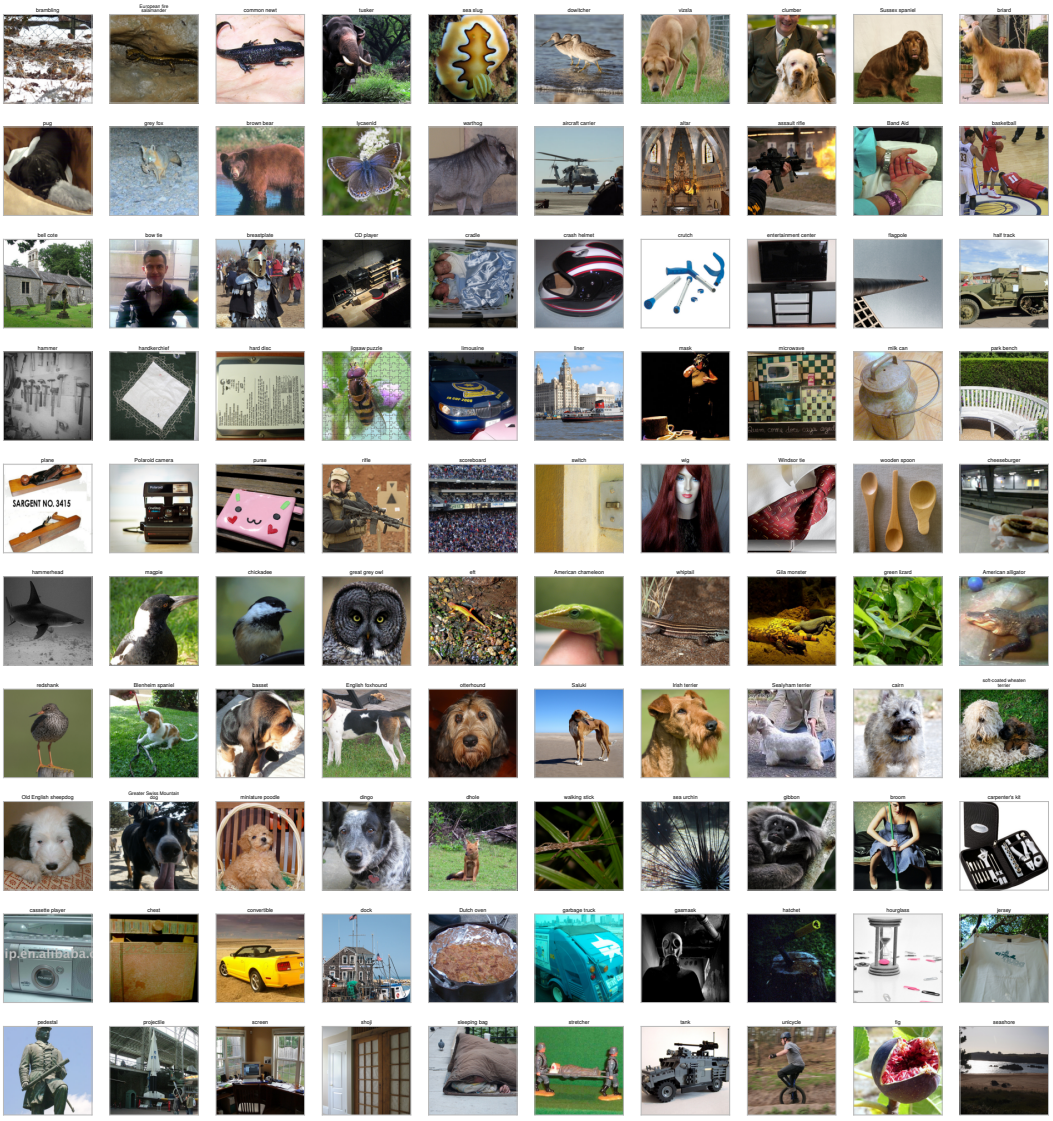


**Figure 7**: **I2P source images.** The 100 ImageNet-1K validation crops used for image-to-pixel generation, arranged in evaluation order. Class names are shown for reference only.

## J 500-WAY RETRIEVAL VOCABULARY

Tables 14 and 15 list the 500 text candidates used for T2P 500-way Top-1 retrieval. Numbers are evaluation indices: the first 100 indices are the benchmark concepts in Table 13, and indices 101–500 are additional distractors. For layout, we sort entries by label length and split at 9 characters (271 shorter labels; 229 longer ones). Within each table, entries fill columns from top to bottom so shorter labels occupy narrower columns.

**Table 14**: **Shorter 500-way labels ($\leq$9 characters).** Sorted by increasing length and filled top-to-bottom across eight columns; 271 entries. Numbers are evaluation indices.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 500. ox | 269. harp | 281. flute | 162. buckle | 286. teapot | 93. rainbow | 168. lorikeet | 421. drumstick |
| 91. axe | 31. lamp | 327. goose | 98. cactus | 375. throne | 448. Samoyed | 163. Malinois | 165. golf cart |
| 75. bed | 97. leaf | 12. heart | 36. camera | 344. tights | 154. sawmill | 446. marmoset | 362. grey wolf |
| 59. bee | 333. mask | 5. house | 26. carrot | 386. tripod | 257. snorkel | 50. mountain | 189. guacamole |
| 204. bra | 275. maze | 172. junco | 100. castle | 21. turtle | 424. spatula | 13. mushroom | 334. gyromitra |
| 40. bus | 161. mink | 72. knife | 64. cherry | 216. tusker | 140. sundial | 221. odometer | 355. hair clip |
| 7. car | 48. moon | 65. lemon | 498. collie | 191. valley | 128. T-shirt | 229. payphone | 68. hamburger |
| 4. cat | 61. pear | 252. llama | 70. cookie | 434. Vizsla | 483. thimble | 235. platypus | 237. harmonica |
| 51. cow | 337. safe | 202. macaw | 155. coucal | 124. wallet | 86. tractor | 402. plectrum | 195. hourglass |
| 10. cup | 81. shoe | 147. moped | 132. cougar | 180. weasel | 268. viaduct | 433. popsicle | 30. ice cream |
| 17. dog | 473. sock | 37. phone | 497. coyote | 89. wrench | 330. volcano | 293. rapeseed | 309. lakeshore |
| 484. fig | 77. sofa | 27. pizza | 425. dugong | 371. bassoon | 137. warthog | 88. sailboat | 95. lightning |
| 122. fly | 11. star | 73. plate | 491. eggnog | 273. bathtub | 192. website | 69. sandwich | 233. monastery |
| 82. hat | 489. suit | 118. purse | 450. eraser | 41. bicycle | 42. airplane | 33. scissors | 416. motorboat |
| 116. hay | 414. tank | 432. quail | 9. flower | 54. chicken | 80. backpack | 389. scorpion | 99. palm tree |
| 328. jay | 396. tent | 266. quill | 179. geyser | 135. doormat | 350. baseball | 205. sea lion | 435. partridge |
| 14. key | 404. tram | 488. quilt | 63. grapes | 243. echidna | 340. chainsaw | 138. stingray | 62. pineapple |
| 361. mop | 6. tree | 392. rifle | 39. hammer | 254. go-kart | 79. computer | 479. sturgeon | 181. plant pot |
| 55. owl | 303. vase | 208. scarf | 377. isopod | 478. gondola | 284. consomme | 236. terrapin | 321. porcupine |
| 53. pig | 318. acorn | 52. sheep | 324. jaguar | 199. gorilla | 198. crayfish | 15. umbrella | 265. projector |
| 47. sun | 381. agama | 60. snail | 245. Kuvasz | 347. hamster | 206. dumbbell | 459. vestment | 113. red panda |
| 212. wok | 1. apple | 58. snake | 169. langur | 406. hatchet | 20. elephant | 280. armadillo | 94. snowflake |
| 18. bird | 130. bagel | 32. spoon | 315. maraca | 464. hot dog | 436. envelope | 193. barometer | 150. spoonbill |
| 43. boat | 360. beach | 2. stool | 261. muzzle | 395. hot pot | 171. fireboat | 102. bobsleigh | 480. stopwatch |
| 38. book | 262. bison | 76. table | 23. orange | 209. ladybug | 143. flagpole | 220. brambling | 85. submarine |
| 74. bowl | 29. bread | 44. train | 111. pillow | 338. limpkin | 294. flamingo | 22. butterfly | 398. sunscreen |
| 28. cake | 339. canoe | 46. truck | 240. poncho | 394. macaque | 218. flatworm | 276. car wheel | 244. tarantula |
| 66. corn | 219. chain | 454. vespa | 194. prison | 227. Maltese | 313. fountain | 114. centipede | 123. tiger cat |
| 443. dome | 447. cloak | 354. zebra | 19. rabbit | 364. marimba | 177. fur coat | 401. chickadee | 152. trash can |
| 96. fire | 16. clock | 481. baboon | 410. racket | 305. minibus | 167. goldfish | 141. cockroach | 115. tree frog |
| 8. fish | 456. clogs | 3. banana | 45. rocket | 210. monitor | 299. hair wig | 282. corkscrew | 190. turnstile |
| 71. fork | 49. cloud | 387. beaker | 431. sandal | 108. padlock | 238. hornbill | 475. crossword | 153. water jug |
| 57. frog | 311. dingo | 290. beaver | 160. shield | 384. pelican | 278. knee pad | 157. Dalmatian | 366. wild boar |
| 419. gown | 67. donut | 35. bottle | 92. shovel | 56. penguin | 246. lacewing | 291. dishcloth | |

**Table 15**: **Longer 500-way labels ($\geq$10 characters).** Sorted by increasing length and filled top-to-bottom across four columns; 229 entries. Numbers are evaluation indices.

| | | | |
|---|---|---|---|
| 260. Arctic fox | 499. coffeemaker | 249. Cairn Terrier | 119. steam locomotive |
| 263. bald eagle | 442. dung beetle | 365. china cabinet | 196. computer keyboard |
| 397. black swan | 382. fire screen | 490. digital clock | 186. impala (antelope) |
| 477. Bloodhound | 264. freight car | 336. garbage truck | 482. little blue heron |
| 383. box turtle | 207. grand piano | 166. grocery store | 295. monarch butterfly |
| 461. bridegroom | 452. great egret | 158. howler monkey | 373. Redbone Coonhound |
| 213. can opener | 234. green mamba | 427. Irish Terrier | 285. rhinoceros beetle |
| 156. car mirror | 139. hermit crab | 106. Japanese Chin | 356. Shetland Sheepdog |
| 439. chiffonier | 495. hummingbird | 105. manhole cover | 107. sulphur butterfly |
| 277. chimpanzee | 259. hunting bow | 201. measuring cup | 405. threshing machine |
| 296. coral reef | 385. leaf beetle | 148. messenger bag | 426. Bluetick Coonhound |
| 463. crane bird | 272. mixing bowl | 465. movie theater | 469. Christmas stocking |
| 451. dishwasher | 185. pencil case | 487. red king crab | 485. hand-held computer |
| 370. fire truck | 357. Persian cat | 331. reflex camera | 493. Labrador Retriever |
| 121. frying pan | 230. pirate ship | 378. rocking chair | 453. Scottish Deerhound |
| 242. greenhouse | 223. planetarium | 467. silver salmon | 367. semi-trailer truck |
| 255. guinea pig | 173. rain barrel | 174. upright piano | 444. smooth green snake |
| 462. hair dryer | 250. salt shaker | 215. velvet fabric | 149. Alaskan tundra wolf |
| 144. hair spray | 90. screwdriver | 412. American robin | 248. black-footed ferret |
| 274. hammerhead | 496. sea anemone | 126. Carolina anole | 267. electric locomotive |
| 84. helicopter | 241. Siamese cat | 353. cliff dwelling | 297. Irish Water Spaniel |
| 317. hoop skirt | 300. tape player | 253. Cocker Spaniel | 486. kite (bird of prey) |
| 428. lawn mower | 326. tennis ball | 117. Dungeness crab | 307. through arch bridge |
| 142. leafhopper | 363. tiger shark | 103. filing cabinet | 476. Cardigan Welsh Corgi |
| 146. lighthouse | 214. titi monkey | 379. French Bulldog | 359. entertainment center |
| 83. motorcycle | 332. toilet seat | 449. graduation cap | 306. Old English Sheepdog |
| 415. mud turtle | 164. trench coat | 420. great grey owl | 289. southern black widow |
| 349. neck brace | 322. triceratops | 372. ping-pong ball | 125. swim trunks / shorts |
| 423. Otterhound | 298. wheelbarrow | 182. shower curtain | 400. European green lizard |
| 352. paintbrush | 256. Windsor tie | 380. soap dispenser | 348. red admiral butterfly |
| 325. park bench | 178. wine bottle | 101. spiral or coil | 492. rotary dial telephone |
| 251. piggy bank | 224. wolf spider | 310. tandem bicycle | 323. small white butterfly |
| 329. pipe organ | 351. Afghan Hound | 127. American dipper | 188. yellow lady's slipper |
| 239. Pomeranian | 109. barber chair | 104. baseball player | 374. Curly-coated Retriever |
| 418. pool table | 292. black grouse | 225. chocolate syrup | 222. loupe magnifying glass |
| 151. promontory | 403. Brittany dog | 358. cocktail shaker | 258. ram (adult male sheep) |
| 455. pufferfish | 466. coral fungus | 393. common redshank | 302. red wolf or maned wolf |
| 312. rotisserie | 440. Egyptian Mau | 314. electric guitar | 228. sidewinder rattlesnake |
| 369. safety pin | 399. fox squirrel | 226. fire salamander | 429. Welsh Springer Spaniel |
| 197. sea urchin | 316. green iguana | 422. football helmet | 304. black-and-white colobus |
| 200. shoe store | 131. handkerchief | 458. four-poster bed | 136. Australian Silky Terrier |
| 460. shower cap | 368. home theater | 211. Groenendael dog | 417. automated teller machine |
| 87. skateboard | 271. killer whale | 471. laptop computer | 170. Bouvier des Flandres dog |
| 391. snoek fish | 335. mosquito net | 145. longhorn beetle | 301. Chesapeake Bay Retriever |
| 407. snowmobile | 231. patas monkey | 270. mashed potatoes | 283. English Springer Spaniel |
| 110. spider web | 438. picket fence | 346. Norfolk Terrier | 232. sulphur-crested cockatoo |
| 345. stone wall | 287. punching bag | 376. Standard Poodle | 279. German Shorthaired Pointer |
| 24. strawberry | 247. railroad car | 472. aircraft carrier | 288. product packet / packaging |
| 474. table lamp | 494. refrigerator | 184. bulletproof vest | 112. Staffordshire Bull Terrier |
| 468. teddy bear | 183. sleeping bag | 430. butternut squash | 470. shoji screen / room divider |
| 78. television | 445. sliding door | 441. chain-link fence | 129. Soft-coated Wheaten Terrier |
| 34. toothbrush | 342. slot machine | 159. desktop computer | 320. American Staffordshire Terrier |
| 25. watermelon | 187. snow leopard | 120. English foxhound | 413. desert grassland whiptail lizard |
| 203. worm snake | 134. swimming cap | 457. high-speed train | 437. military hat (bearskin or shako) |
| 133. barn spider | 411. toilet paper | 390. magnetic compass | 343. Mexican hairless dog (xoloitzcuintli) |
| 409. bell pepper | 341. Affenpinscher | 217. military uniform | |
| 175. block plane | 308. baby pacifier | 388. Miniature Poodle | |
| 319. brain coral | 408. bighorn sheep | 176. proboscis monkey | |

## K FULL TEXT-TO-PIXEL RESULTS

Tables 16 and 17 report the per-resolution Top-1 and MRR scores corresponding to Table 1. Table 18 decomposes the VLM Judge score into its six rubric dimensions.

**Table 16**: **Full Text-to-Pixel semantic recognition results.** SigLIP-SO400M 100-way and 500-way Top-1 under text-candidate pools. "Gem" and "F" denote Gemini and Flash. Invalid generations are counted as retrieval misses. All values are proportions; higher is better.

| Resolution | Pool | Gem 3.6 F | GPT-5.6 Sol | GPT-6 Astra | Opus 5 | Kimi K3 | DeepSeek V4 Pro | Seed-2.1 Pro |
|---|---|---|---|---|---|---|---|---|
| $16^2$ | 100-way | .910 | .910 | **.950** | .910 | .900 | .580 | .810 |
| $16^2$ | 500-way | **.870** | .840 | **.870** | .810 | .810 | .490 | .700 |
| $32^2$ | 100-way | .830 | .920 | **.980** | .950 | .930 | .680 | .820 |
| $32^2$ | 500-way | .780 | .890 | **.920** | .870 | .870 | .590 | .720 |
| $64^2$ | 100-way | .810 | .890 | **.980** | .970 | .860 | .630 | .830 |
| $64^2$ | 500-way | .720 | .830 | **.940** | .890 | .770 | .510 | .740 |

**Table 17**: **Full Text-to-Pixel mean reciprocal rank results.** SigLIP-SO400M MRR under 100-way and 500-way candidate pools. "Gem" and "F" denote Gemini and Flash. GPT-6 Astra and Claude Opus 5 have 100 valid generations at every resolution. DeepSeek V4 Pro has 100, 99, and 97 valid generations at $16^2$, $32^2$, and $64^2$, respectively. †Evaluated on 99 valid generations; "–" denotes an unavailable result.

| Resolution | Pool | Gem3.6 F | GPT-5.6 Sol | GPT-6 Astra | Opus 5 | Kimi K3 | DeepSeek V4 Pro | Seed-2.1 Pro |
|---|---|---|---|---|---|---|---|---|
| $16^2$ | 100-way | .945 | .945 | **.972** | .949 | .941 | .697 | .874 |
| $16^2$ | 500-way | .913 | .899 | **.927** | .884 | .881 | .599 | .792 |
| $32^2$ | 100-way | .887 | .956 | **.990** | .972 | .965 | .769 | .892 |
| $32^2$ | 500-way | .837 | .931 | **.960** | .925 | .932 | .681 | .815 |
| $64^2$ | 100-way | .843 | .933 | **.990** | .985 | .896† | .718 | .898 |
| $64^2$ | 500-way | .785 | .893 | **.970** | .945 | .826† | .614 | .832 |

**Table 18**: **Text-to-Pixel VLM Judge dimension scores.** Gemini 3.5 Flash scores each rendered output on subject identity (ID), defining attributes and key parts (Parts), structure and proportion (Struct.), detail quality (Detail), pixel-art execution (Pixel), and palette and contrast (Color). Entries are mean scores at each native resolution; Total is out of 100. Bold and underline mark the best and second-best values within each resolution.

| Model | ID/20 | Parts/25 | Struct./20 | Detail/20 | Pixel/10 | Color/5 | Total/100 |
|---|---|---|---|---|---|---|---|
| $16^2$ | | | | | | | |
| GPT-6 Astra | **18.83** | **22.89** | **18.16** | **16.95** | **8.94** | **4.98** | **90.75** |
| GPT-5.6 Sol | 16.11 | <u>19.31</u> | 15.77 | <u>13.98</u> | <u>8.21</u> | <u>4.75</u> | <u>78.13</u> |
| Claude Opus 5 | 15.77 | 18.52 | 15.61 | 12.60 | 7.88 | <u>4.75</u> | 75.13 |
| Kimi K3 | 15.48 | 18.21 | 15.27 | 12.90 | 7.99 | 4.69 | 74.54 |
| Gemini 3.6 Flash | <u>16.22</u> | 19.23 | <u>15.79</u> | 13.66 | 8.17 | 4.72 | 77.79 |
| Seed-2.1 Pro | 14.12 | 16.57 | 14.07 | 11.61 | 7.64 | 4.56 | 68.57 |
| DeepSeek V4 Pro | 10.99 | 12.34 | 11.47 | 8.17 | 6.68 | 4.00 | 53.65 |
| $32^2$ | | | | | | | |
| GPT-6 Astra | **19.16** | **23.18** | **18.36** | **16.94** | **8.95** | **4.99** | **91.58** |
| GPT-5.6 Sol | <u>16.57</u> | <u>19.52</u> | <u>16.06</u> | <u>13.80</u> | <u>8.11</u> | <u>4.80</u> | <u>78.86</u> |
| Claude Opus 5 | 16.04 | 18.39 | 15.49 | 11.92 | 7.59 | 4.69 | 74.12 |
| Kimi K3 | 15.80 | 18.27 | 15.35 | 12.68 | 7.92 | 4.65 | 74.67 |
| Gemini 3.6 Flash | 14.77 | 17.05 | 14.47 | 12.33 | 7.99 | 4.60 | 71.21 |
| Seed-2.1 Pro | 13.56 | 15.55 | 13.22 | 10.77 | 7.10 | 4.36 | 64.56 |
| DeepSeek V4 Pro | 11.76 | 13.13 | 11.50 | 8.51 | 6.17 | 3.95 | 55.02 |
| $64^2$ | | | | | | | |
| GPT-6 Astra | **18.74** | **22.63** | **18.04** | **16.38** | **8.92** | **4.95** | **89.66** |
| GPT-5.6 Sol | 15.14 | <u>17.95</u> | <u>15.05</u> | <u>12.23</u> | <u>7.87</u> | <u>4.72</u> | <u>72.96</u> |
| Claude Opus 5 | <u>15.55</u> | 17.54 | 14.58 | 10.95 | 6.97 | 4.48 | 70.07 |
| Kimi K3 | 12.95 | 14.69 | 12.82 | 9.68 | 7.05 | 4.34 | 61.53 |
| Gemini 3.6 Flash | 12.67 | 14.30 | 12.30 | 9.87 | 7.05 | 4.16 | 60.35 |
| Seed-2.1 Pro | 12.89 | 14.24 | 12.27 | 9.32 | 6.45 | 4.16 | 59.33 |
| DeepSeek V4 Pro | 10.16 | 10.77 | 9.58 | 6.54 | 5.20 | 3.42 | 45.67 |

# L ALGORITHMIC IMAGE-TO-PIXEL BASELINES IN FULL

Each method either combines a spatial downscaling stage with a subsequent color quantization stage or jointly determines the output regions and palette. We set the palette size to eight for all constrained methods and evaluate them on the same 100 ImageNet validation crops used for the VLMs.

**Tier 0: simple resampling.** Nearest-neighbor sampling selects the source-cell center, while box filtering averages the corresponding source patch. Both are followed by median-cut quantization (Heckbert, 1982).

**Tier 1: linear filters.** Bicubic and Lanczos interpolation use standard fixed resampling kernels before median-cut quantization. These methods allocate the output resolution uniformly across the image.

**Tier 2: content-adaptive downscaling.** DPID (Weber et al., 2016) weights input pixels according to their deviation from a local guidance color; we evaluate $\lambda \in \{0.5, 1.0\}$. SSIM-optimal downscaling (Öztireli & Gross, 2015) minimizes structural dissimilarity using local image statistics. Both methods are followed by median-cut quantization.

**Tier 3: pixel-art-specific methods.** Pyxelate[1] uses Sobel-gradient information to merge local cells and estimates an eight-color palette with a Bayesian Gaussian mixture. We use version 2.1 or later with `sobel=3`, `depth=1`, and dithering disabled. Pixelated Image Abstraction (PIA) (Gerstner et al., 2012) jointly constructs output regions and a reduced palette. The reference implementation uses modified SLIC mapping, mass-constrained deterministic annealing (Rose, 1998), bilateral superpixel smoothing, and a saturation factor of 1.1. PIA (reduced) retains SLIC with compactness $m = 45$, replaces annealing with weighted eight-means over superpixel colors, and omits bilateral smoothing and saturation adjustment.

## L.1 REPRESENTATIVE IMPLEMENTATIONS

The following procedures specify the three representatives used in the main comparison. All receive a $256 \times 256$ RGB crop $I$ and target side length $N \in \{16, 32, 64\}$. Median cut uses $K = 8$ without dithering. SLIC and $k$-means in PIA (reduced) operate in CIELAB; the other operations use RGB.

**Nearest + median cut**

*Input:* image $I$ and target side length $N$.

1. For each output cell $(y, x)$, sample $I$ at the center of the corresponding source cell to obtain $Y[y, x]$.
2. Apply $\mathrm{MedianCut}(Y, K = 8)$ and return the quantized grid.

**SSIM-optimal + median cut**

*Input:* image $I$ of side length $H$ and target side length $N$.

1. Set $s = H/N$. For each output cell, compute the aligned $s \times s$ block mean $L[y, x]$ and squared-value mean $Q[y, x]$ for each RGB channel.
2. For each $2 \times 2$ output neighborhood, compute its mean $\mu$, the variance $\sigma_L^2$ of $L$, and $\sigma_H^2 = \mathrm{mean}(Q) - \mu^2$.
3. Set $r = \sqrt{\sigma_H^2/\sigma_L^2}$ when $\sigma_L^2 \geq \varepsilon$, and $r = 0$ otherwise. Form the local proposal $\hat{Y} = \mu + r(L - \mu)$.
4. Average overlapping proposals, clip values to $[0, 1]$, and return $\mathrm{MedianCut}(\hat{Y}, K = 8)$.

[1] https://github.com/sedthh/pyxelate

**PIA (reduced)**

*Input:* image $I$ and target side length $N$.

1. Compute $S = \mathrm{SLIC}(I; n = N^2, m = 45)$ in CIELAB.
2. For each superpixel $u$, compute its mean color $c_u$ and size $w_u = |u|$.
3. Fit eight palette colors $\{p_k\}_{k=1}^{8}$ and assignments $a$ using weighted $k$-means on $\{c_u\}$ with weights $\{w_u\}$.
4. For each output cell, identify the superpixel at the corresponding source-cell center and assign the RGB conversion of $p_{a(u)}$.

### L.2 Complete Algorithmic Results

**Table 19**: **All palette-matched algorithmic baselines on I2P.** Every method is restricted to eight colors. DINO is paired DINOv2-Large CLS cosine, SigLIP cosine uses pooled SigLIP-SO400M features, and 100-way Top-1 is measured over the 100 source images. Higher is better, with $n = 100$ for every cell. Bold values mark the best result in each column, and colored bands distinguish the four method families.

| | DINO CLS cosine | | | SigLIP cosine | | | 100-way Top-1 (%) | | |
|---|---|---|---|---|---|---|---|---|---|
| Method | $16^2$ | $32^2$ | $64^2$ | $16^2$ | $32^2$ | $64^2$ | $16^2$ | $32^2$ | $64^2$ |
| **Tier 0 Simple resampling** | | | | | | | | | |
| Nearest + median cut | .012 | .090 | .480 | .523 | .568 | .665 | 9 | 28 | 62 |
| Box + median cut | .012 | .099 | .475 | .523 | .577 | .673 | 10 | 27 | 65 |
| **Tier 1 Linear filters** | | | | | | | | | |
| Bicubic + median cut | .013 | .093 | .459 | .524 | .573 | .672 | 9 | 26 | 64 |
| Lanczos + median cut | .013 | .099 | .478 | .524 | .576 | .673 | 10 | 29 | 64 |
| **Tier 2 Content-adaptive downscaling** | | | | | | | | | |
| DPID $\lambda = 0.5$ + median cut | .011 | .103 | .509 | .523 | .578 | .679 | 10 | 27 | 68 |
| DPID $\lambda = 1.0$ + median cut | .011 | .104 | .527 | .522 | .580 | .678 | 9 | 32 | 65 |
| SSIM-optimal + median cut | .013 | .119 | **.543** | .524 | .579 | .686 | 9 | 34 | 75 |
| **Tier 3 Pixel-art-specific methods** | | | | | | | | | |
| Pyxelate | .013 | .085 | .377 | .525 | .569 | .655 | 12 | 32 | 65 |
| PIA (reduced) | .014 | **.135** | .490 | **.546** | **.608** | **.697** | **18** | **50** | **78** |
| PIA (reference) | **.015** | .123 | .485 | .541 | .599 | .692 | 17 | 40 | 71 |

Table 19 shows that standard resampling filters and the two DPID settings yield similar results, while the strongest method depends on the metric. SSIM-optimal obtains the highest DINOv2 fidelity at $64^2$, whereas PIA (reduced) leads both SigLIP metrics across resolutions. We therefore use one representative for simple resampling, fidelity-oriented downscaling, and joint geometry–palette optimization.