GENOLYMPIAD: BENCHMARKING FRONTIER IMAGE MODELS AT THEIR CAPABILITY BOUNDARIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Text-to-image models have made tremendous progress and can produce high-quality images. However, a question which would be asked in any other discipline is where the frontier models fall short. We introduce GenOlympiad, a benchmark which provides the most challenging text prompts from nine capability axes for stress-testing frontier text-to-image models. The assessed capabilities include generating large quantities, performing deep reasoning, ensuring hard physical consistency, following complex rules and so on. The prompts are designed such that prompt alignment is visually verifiable (*e.g., counting*) instead of relying on subjectiveness (*e.g., style*). Based on this, we develop an automated evaluation framework which decomposes the prompt into easy and hard questions, and for most hard questions, draws help lines to *ground* evidence for accurate grading. We evaluate 20+ strong text-to-image models, including Nano Banana Pro, GPT-Image-2.5, Seedream-5.0 and so on. We report that even the best-performing T2I model (Nano Banana Pro) only achieves a strict score of 23.9 out of 100 and that coding models such as GPT-6 Astra have higher prompt alignment scores (but not as visually pleasing). We also derive insights in model evolution, alignment with Elo scores, and effectiveness of our grounding-based evaluation protocol.

1 INTRODUCTION

In artificial intelligence, benchmarks are introduced to expose frontier models to their capability boundaries, measure them, and seek solutions from the community. Some well known examples include ObjectNet (Barbu et al., 2019) for vision model generalization, Humanity's Last Exam (Center for AI Safety et al., 2026) for LLM question answering across subjects, SWE-bench (Jimenez et al., 2024) for automatic issue resolution, ARC-AGI benchmarks (Chollet, 2019; Chollet et al., 2025) for general intelligence, and FrontierMath (Glazer et al., 2024) for testing AI on advanced mathematics.

Frontier text-to-image models can now produce high quality images (Sani et al., 2026; Li et al., 2026). This is evidenced by both real user scoring, where platforms like Artificial Analysis Arena rank models by asking users to compare image pairs head-to-head, and strong scores on automatic benchmarks. GenEval (Ghosh et al., 2023) has risen from 0.61 when first introduced in 2023 to $\sim$0.90 for recent models. PRISM-Bench(Fang et al., 2026) score which measures image-text alignment and aesthetics is now over 85 by Nano Banana. However, these evaluation platforms either do not diagnose limitations of models (e.g., AA Arena) or see scores being saturated (e.g., GenEval) and thus no longer sit at capability boundaries. Therefore, we ask the question: *when generated images already look high-quality, where do frontier image models still fail?*

We introduce GenOlympiad, an exam of 801 hardest prompts for frontier model evaluation, co-designed by AI and human. The prompts span nine capability axes, including quantity and precision, latent computation, physical consistency, spatial and 3D reasoning. To be hard, instead of adding more objects or textual conditions, the GenOlympiad prompts focus on hard constraints that must be strictly satisfied. An example prompt is 'a compass on a boat, with its needle pointing toward the setting sun': although frontier models can generate realistic lighting and textures, they usually fail to satisfy the constraint of 'needle pointing towards the setting sun', which is what we assess.

Given each GenOlympiad prompt and an image generated by a frontier model, we design an automatic pipeline to score the generations, which focuses on constraint satisfaction instead of general alignment. Specifically, we do not ask the vision language model (VLM) 'are there eight blue caps',

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.

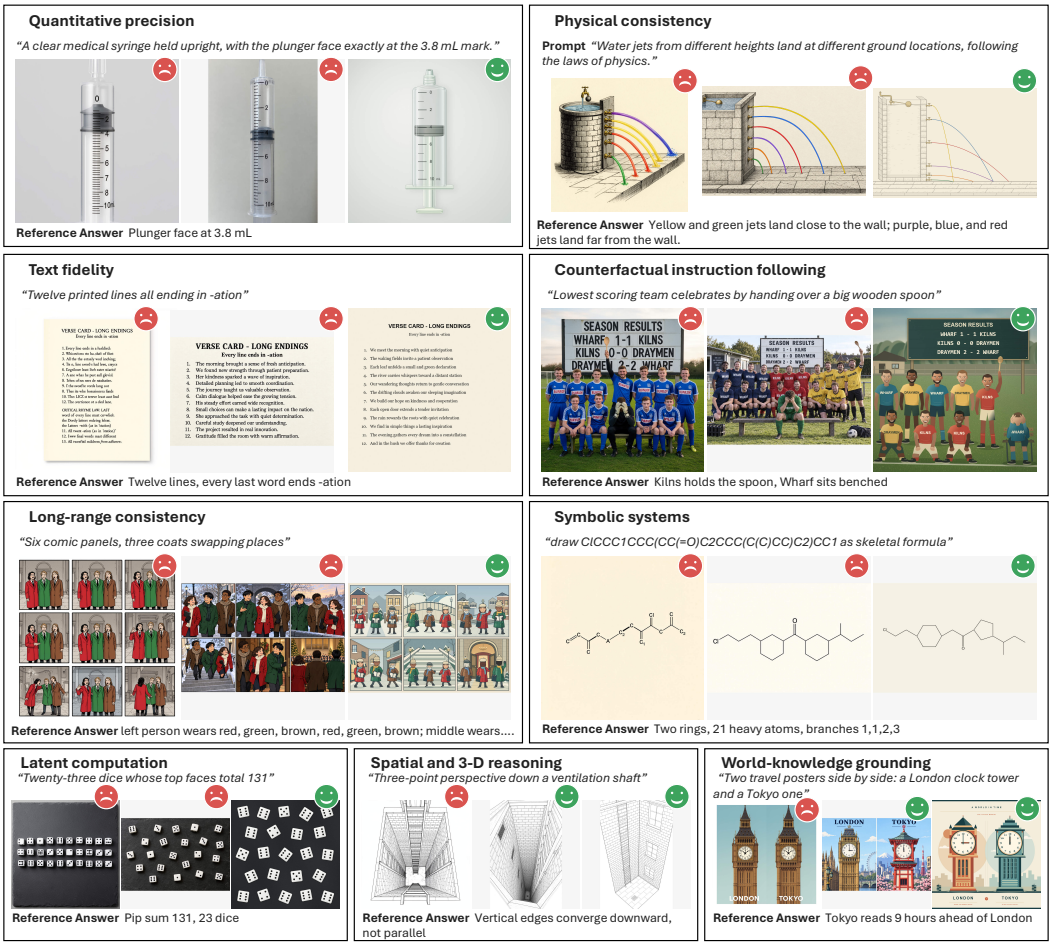

Figure 1: Image generation examples of the nine capability axes in GenOlympiad. In each capability, images from left to right are generated by FLUX.2 dev, GPT Image 2.5, and GPT-6 Astra, respectively. The prompts are shortened for demonstration. More examples are in Appendix Fig. 17.

because the answer can often be wrong. Instead, we break it down to atomic tasks, *e.g.*, detecting all the blue caps, drawing the bounding boxes on the image, and asking whether each bounding box contains a blue cap. Evaluation becomes more accurate and interpretable if we let the VLM perform such simple evidence extraction tasks and ground the task results (drawing bounding boxes and help lines). Our scoring function is deterministic, matching the extracted evidence (*e.g.*, bounding boxes height, width, and coordinates) against the prompt. In addition, when grounding fails, we withhold the task instead of assigning an unreliable score.

We evaluate a suite of frontier models on GenOlympiad with our evaluation protocols. We report *strict score* which requires all constraints in a task to be correct, and *relaxed score* which gives a partial score for some constraints. Detailed definition is in Section 3.3. Nano Banana Pro gains 23.9 in strict score and 40.0 in relaxed score out of 100, ranking first among T2I models, while the latest GPT-Image-2.5-sunburst ranks second at 14.9 in strict score and 29.6 in relaxed score. We also find that frontier closed models, e.g., the GPT-Image family, exhibit better GenOlympiad scores with their new releases, while open-weight image models have mixed results, suggesting that the latter may not have been trained at the boundary. In sum, this paper has the following main points.

- We introduce GenOlympiad, a test set of 801 hardest prompts to evaluate frontier models.
- We develop an automatic pipeline that provides robust evaluation for each prompt.
- Benchmarking a wide range of frontier models reveals that frontier models have poor performance and gives us rich insights in the boundary limitations and model development trend.

2 RELATED WORK

Benchmarks for text-to-image generation. Some popular benchmarks focus on object presence and attributes like color, relations or counting of a few objects (Ghosh et al., 2023; Kamath et al., 2025; Paiss et al., 2023; Ye et al., 2025; Huang et al., 2025). Others focus on creative images that rarely appear in the training set (Wu et al., 2024) or long prompt following, such as DPG-Bench (Hu et al., 2024) and TIIF-Bench (Wei et al., 2025). Recently, more challenging benchmarks are introduced by adding reasoning, *e.g.*, Einstein's favorite instrument (Li et al., 2024; Chen et al., 2025; Sun et al., 2026; Niu et al., 2025; Singh & Zheng, 2023; Chang et al., 2025). Tasks in GenExam (Wang et al., 2025) reflect various disciplines such as history, music, and biology, while MathGen (Liu et al., 2026) focuses on mathematics content, *e.g.*, angles, function, set and counting tasks with many objects. Prompts in Qwen-Image-Bench (Li et al., 2026) and ImagenWorld (Sani et al., 2026) instead reflect real design demand in games, products, fashion, *etc*. Earlier benchmarks like GenEval have already been saturated to 90%. Frontier models also made rapid improvement on recent benchmark, *e.g.*, GenExam reports a 74.6% strict score for GPT-Image-2; while MathGen report 53.7% pass rate (similar to strict score) for Nano Banana Pro. In comparison, GenOlympiad curates the hardest prompts and only sees 23.9% and 14.9% of Nano Banana Pro and GPT-Image-2.5 Sunburst.

Evaluation of generated images. Early metrics like Inception Score (IS) (Salimans et al., 2016) and Fréchet Inception Distance (FID) (Heusel et al., 2017) measure how realistic images are. Later metrics focus more on image-text alignment. CLIPScore (Hessel et al., 2021) calculates the similarity between prompt and image, which is not strongly correlated with human judgement (Ghosh et al., 2023). Human based metrics like Elo in Artificial Analysis Image Arena was later adopted, but such metrics are too expensive to use and do not diagnose limitations of models. VQA-based automatic metrics mitigate these problems (Hu et al., 2023; Cho et al., 2024; Lin et al., 2024) by directly applying VQA models to image-question pairs. Other evaluation methods use vision tools to obtain important evidence about texts or objects, *e.g.*, GenEval (Ghosh et al., 2023) uses detection models; MathGen (Liu et al., 2026) uses OCR, contour detection, *etc* to extract text, angles, *etc*. GenOlympiad is similar to GenExam/TIFA/DA-Score in breaking the prompt into questions, but is different in using visual grounding in evaluation. GenOlympiad also shares similarity with GenEval/MathGen and thinking with images (Yang et al., 2025) in using vision tool for evidence collection; our key difference is using an iterative process to improve grounding (Table 2).

3 THE GENOLYMPIAD BENCHMARK

3.1 WHAT COMPOSES A TASK?

Each GenOlympiad task contains a natural-language prompt and a bespoke evaluation program, both of which are provided during evaluation. A **prompt** describes the image to be generated. Each prompt has at least one aspect which is very hard to generate, *e.g.*, 35 objects. The brainstorming of the 801 prompts was performed by the authors with the help of AI, details is in Appendix A.3.

The **evaluation program** includes easy questions and hard questions. It also includes the pipelines for evidence grounding and scoring (Section 3.3). The program is created by Claude Opus 5 with human instructions (for details see Appendix A.3). As shown in Fig. 2, easy questions look like 'are there many dice in this photo?' and 'are the top faces of the dice clearly visible?' Example hard questions can be 'Do these dice sum up to 131?' and 'How many dice in this picture?'

3.2 WHAT DOES GENOLYMPIAD CONSIST OF?

The 801 tasks in GenOlympiad overall evaluate nine capability axes below (topic distribution in Fig. 3). More example prompts and success/failure images are shown in Appendix Fig. 1 and Fig. 17.

- Quantity and precision: to generate a large number of objects, exact ratio between objects, precise angles, and precise readings of instrument like ruler.
- Latent computation: computation outcome does not appear in the prompt, requiring the system to analyze the state for generation, *e.g.*, *23 dice which sum up to 131*.

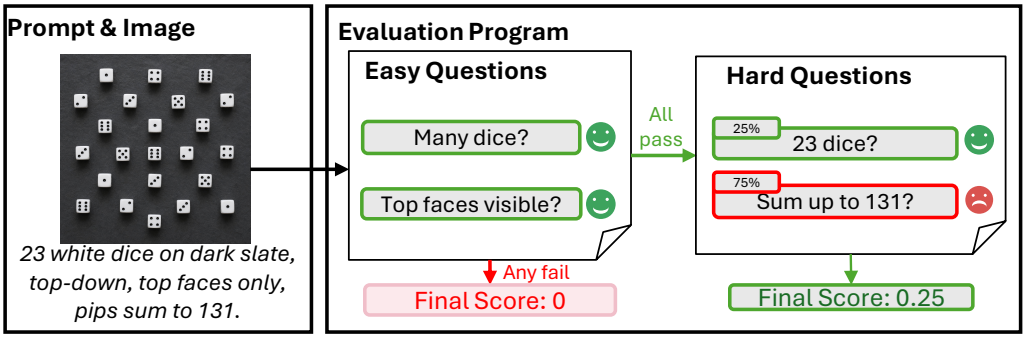

Figure 2: Example of a GenOlympiad prompt and its evaluation program. **Left**: prompt and a generated image. **Right**: Easy and hard questions compose evaluation program. Easy questions are evaluated first as a 'gating' check: if any fails, the final score will be zero. If all pass, the program will mark hard questions, where each hard question has partial mark with different weight.

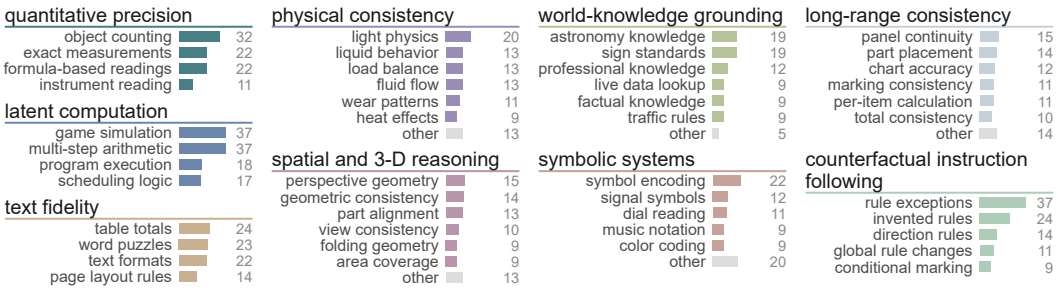

Figure 3: Topic distribution of GenOlympiad on each axis. We show the number of tasks in each topic. A detailed distribution can be seen in Appendix Fig. 12

- Physical consistency: to follow real-world physical constraints like gravity, optics, and stress, *e.g.*, *water jets from different heights land at different ground locations*.
- Spatial and 3D reasoning: examine the correctness of perspective relations and the expand diagram of 3D objects, *e.g.*, *a picture should be drawn in 3-point perspective*.
- Text fidelity: readable text with constraints like font size and language habits like inverted question mark in Spanish, *e.g.*, *a paper printed with 12 lines of rhyme, each line ending with -ation*.
- Symbolic systems: follow symbols in circuit, molecular graph, music sheet, *etc*, *e.g.*, *draw ClCCC1CCC(CC(=O)C2CCC(C(C)CC)C2)CC1 as a line-angle (skeletal) structural formula*.
- Long-range consistency: parts of an image, even parts far apart, should be consistent, *e.g.*, consistency of character in hats, clothes and rings in a multi-panel comic.
- Counterfactual instruction following: prompt is opposite to commonsense knowledge, *e.g.*, *a football team that gets the lowest score celebrate for its loss*.
- World-knowledge grounding: to follow commonsense knowledge *e.g.*, flags of countries; real-time information *e.g.*, current air quality; The time difference between two specific countries should be correct.

3.3 Evaluation Framework

We design an automatic pipeline, which runs the *evaluation program* defined in Section 3.1, to grade how well a generated image follows the prompt.

Easy questions easily asked. We use a TIFA-style method for easy question evaluation: the judge VLM takes the easy question and the generated image as input and answers yes or no. Asking once is enough, since the agreement with human answers is 94.7%, and majority voting over 3 independent

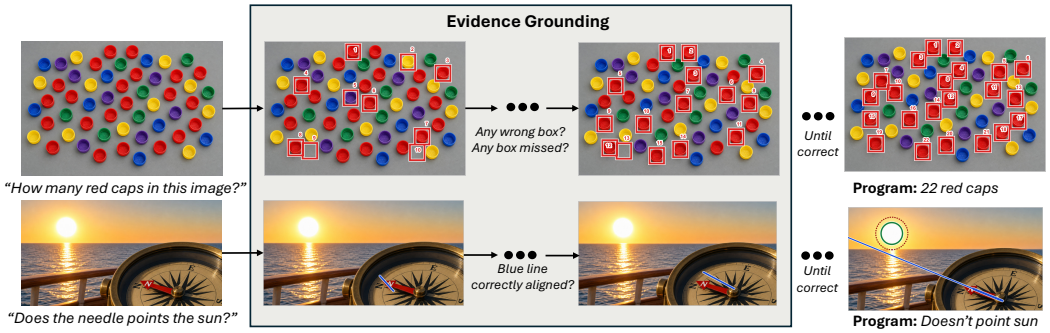

Figure 4: Pipeline of evidence grounding. Given a hard question and the generated image (left), we draw helping lines on the image, *e.g.*, bounding boxes for counting problems, boundaries for the sun, and extension lines for the compass pointer. The type of helping line is decided in the evaluation program. To make the lines accurate, we design an iterative process (middle), where we use Gemini 2.5 Flash to decide whether each line serves the purpose, until success or running out of budget. The grounding result (right) is finally used to decide whether the hard constraint is satisfied.

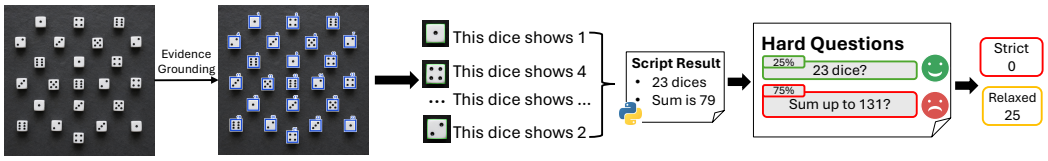

Figure 5: An example pipeline of deterministic scoring. We write the grounding result into a JSON state file, which, for each die, lists the bounding box height, width, and center coordinates, and the displayed number. We use a deterministic Python script to count the number of dice and the sum of the numbers for grading. The entire deterministic scoring pipeline is provided in the evaluation program. Prompt is the same as Fig. 2.

queries only slightly improves it to 95.3% at extra cost. As shown in Fig. 2, only after all easy questions have been passed will we proceed with the hard questions; otherwise, we directly mark 0.

Grounding visual evidence for hard questions. 557 out of 801 tasks in GenOlympiad require evidence grounding. Depending on questions, we draw different helping lines on the generated image. As shown in Fig. 4, for counting questions, we aim to draw bounding boxes of objects of interest; for the compass-sun question, we aim to draw an extension line of the compass pointer and contour of the sun. There is no guarantee that grounding is successful at the first try, so after each try, we ask the VLM whether grounding is successful, such as 'does the blue line correctly overlay with the compass pointer?' If successful, then grounding is complete; if not successful, we repeat the grounding step until reaching maximum trial times. The grounded evidence, *e.g.*, bounding boxes and lines, is then processed by a deterministic program. The program performs bespoke operations, such as counting bounding boxes, calculating the sum of dice or checking whether the compass-pointer line intersects the contour of the sun, to determine whether the hard question is satisfied.

Deterministic scoring. After each hard question has been verified, the scoring method will count the number of hard questions that are successfully reflected in the generated image. For example, there are two hard questions for Fig. 5: (1) is the number of dice 23; (2) do these dice sum up to 131? Following GenExam (Wang et al., 2025), we report two scores. In our 'strict' scoring, only when all the hard questions are successfully passed will we give a mark of 100. In the 'relaxed' setting, different hard questions have different weights, and the weight sum of the successful ones is the final score. The strict score is the percentage of the scored tasks that pass all the hard questions.

Grounding failure. If grounding is unsuccessful after the maximum number of trials, we remove the task rather than force the grader to produce a potentially unreliable score. The failure scenarios may be: an object is too small; object color is very close to the background; many objects overlay each other and can not be clearly located. Grounding may also fail if a rendered text is not recognizable or

the generated objects do not match the objects expected by the evaluation program. We rank models according to their grounding failure rate in Fig. 6.

4 EXPERIMENTS

We perform extensive evaluation and insight analysis of frontier image models on GenOlympiad. In particular, we investigate the following research questions.

- What do comprehensive benchmarking results look like (Section 4.2, Table 1, Fig. 6)? Do frontier models have improved GenOlympiad scores over releases? (Section 4.3, Fig. 7, Appendix Fig. 13)
- What is the correlation of GenOlympiad scores with human ratings on Elo Arenas? (Section 4.3, Fig. 8, and Appendix Fig. 14, 15)
- How strongly does GenOlympiad evaluation results agree with human ratings? (Section 4.4, Table 2, and Appendix Tables 6, 4)

4.1 EXPERIMENTAL SETUP

We benchmark the following T2I models: GPT-Image 1, 1.5, 2, and 2.5 Sunburst (OpenAI, 2025b;c; 2026c;d); Nano Banana, Nano Banana Pro, and Nano Banana 2 (Google, 2025a;b; 2026); Seedream 4.5 and 5.0 Pro (ByteDance Seed, 2025; 2026); FLUX.2 Pro, dev, and Klein (Black Forest Labs, 2025); FLUX.1 dev and Schnell (Black Forest Labs, 2024); Stable Diffusion 3.5 (Esser et al., 2024; Stability AI, 2024); SANA (Xie et al., 2025); and Qwen-Image-20B (Wu et al., 2025).

We evaluate open-weight models ranging from 0.6B to 20B parameters on the same server with a single NVIDIA A100 80GB GPU. For each model family, we configure the context length of the primary text-conditioning encoder to make the full prompt visible whenever possible. If prompt length exceeds the context length, it will neither generate nor be calculated into the overall score.

We also benchmark two coding models, *i.e.*, GPT-6 Astra (OpenAI, 2026b), GPT-5.6 Sol (OpenAI, 2026a) in medium effort with internet accessibility. We insert a system prompt (See Appendix A.4) to let the coding models write a single HTML document. The content on this document is rendered programmatically by JavaScript. We evaluate image aesthetics using the same method and setup as PRISM-Bench (Fang et al., 2026), where the evaluator is GPT-4.1 (OpenAI, 2025a).

Prompt engineering (PE), which preserves the original constraints while improving T2I performance, is evaluated on GPT-Image-2.5. We use PromptEnhancer-32B (Wang et al., 2026) and GPT-5.6 Sol. For GPT-5.6 Sol, we use a fixed prompt from Tailored Visions (Chen et al., 2024).

4.2 BENCHMARKING RESULTS

Comparing closed-source T2I models. As shown in Table 1, Nano Banana Pro obtained 40.0 relaxed score and 23.9 strict score and performs the best in text fidelity, counterfactual instruction following, symbolic systems and so on. GPT-Image-2.5 is on par with Seedream 5.0 Pro and Nano Banana 2. In addition, Seedream 5.0 Pro in latent computation and physical consistency performs better than all the GPT-Image series.

Prompt engineering (PE) analysis. We test whether prompt rewriting is useful for GenOlympiad. We use GPT-Image-2.5 as the base model. We compare PromptEnhancer-32B and GPT-5.6 Sol for prompt rewriting while keeping the image model and evaluator unchanged. Results are shown in Table 1. PromptEnhancer-32B brings nearly no overall improvement over GPT-Image-2.5, while GPT-5.6 Sol increases the strict score and relaxed score from 14.9 to 33.0 and from 29.6 to 48.9, respectively. The gain is significant in latent computation and text fidelity and very limited in spatial and 3D reasoning. This demonstrates the usefulness of PE in T2I system design.

Coding models and aesthetics. Coding models generally have higher GenOlympiad scores than T2I models. For example, the strict scores of GPT-6 Astra and Nano Banana Pro are 40.5 and 23.9, respectively. This is because our evaluation focuses on reasoning, where these frontier models are strong. On the other hand, it is also expected that the aesthetic score of coding models is often lower than the strong T2I models (refer Table 3 and Appendix Fig. 16). For example, the aesthetic scores of GPT-5.6 Sol and GPT-Image-2.5 are 4.61 and 5.51, respectively.

Table 1: GenOlympiad results for closed-source T2I models, coding models, prompt engineering and open-source T2I models across nine capability axes. Best result in each column and each category is highlighted in **bold**. C-Fact: counterfactual instruction following; Symbol: symbolic systems; World: world-knowledge grounding; Phys: physical consistency; Quant: quantity and precision; Spatial: spatial and 3D reasoning; Latent: latent computation; Long: long-range consistency. Overall is the mean score over all scored tasks of a model. Withheld tasks are excluded from both axis and overall scores (see Fig. 6). S and R are strict and relaxed scores.

Model	Text		C-Fact		Symbol		World		Phys		Quant		Spatial		Latent		Long		Overall	
	S	R	S	R	S	R	S	R	S	R	S	R	S	R	S	R	S	R	S	R
Coding Models																				
GPT-6 Astra	**65.4**	**74.1**	**38.4**	**50.2**	**45.5**	**56.4**	**27.5**	**46.9**	25.6	**43.0**	**40.8**	**54.0**	26.7	48.3	**58.4**	**73.7**	**30.0**	**53.1**	**40.5**	**56.1**
GPT-5.6 Sol	62.5	72.4	32.2	43.3	42.3	55.5	26.0	43.5	**27.7**	41.6	31.6	45.8	**28.2**	**49.4**	53.3	69.1	28.4	46.7	37.5	52.4
Prompt Engineering on GPT-Image-2.5																				
GPT 5.6 Sol	**62.2**	**71.7**	**33.0**	**48.7**	**30.4**	**44.7**	**17.8**	**36.2**	**25.0**	**39.4**	**24.3**	**40.8**	**12.5**	**34.3**	**55.8**	**72.3**	**27.2**	**43.2**	**33.0**	**48.9**
PromptEnhancer	26.9	38.7	21.3	33.9	19.8	30.7	6.8	19.0	8.3	24.5	16.7	27.7	8.9	26.8	15.2	30.6	11.0	22.3	15.1	28.4
Closed-source T2I Models																				
Nano Banana Pro	**41.5**	**55.8**	**27.8**	**39.3**	**25.6**	**38.6**	10.5	**29.5**	13.1	**33.8**	**17.1**	**27.0**	11.2	30.7	**38.1**	**56.0**	**24.4**	**43.2**	**23.9**	**40.0**
GPT-Image-2.5	30.4	41.8	23.0	36.4	20.5	34.8	6.8	24.1	8.1	23.7	13.2	25.0	**18.4**	**34.6**	6.5	21.9	9.8	25.9	14.9	29.6
Seedream 5.0 Pro	19.0	30.8	24.1	36.0	18.9	32.7	2.5	16.6	**13.8**	30.8	10.4	20.7	9.9	23.7	20.8	34.7	13.7	26.6	15.1	28.4
Nano Banana 2	19.5	28.5	13.4	27.3	6.6	18.4	**13.9**	27.0	8.9	23.0	6.8	18.3	8.1	20.9	16.2	32.5	15.9	29.3	12.4	25.4
GPT-Image-2	20.8	32.8	14.0	26.7	17.5	29.6	5.4	17.2	6.1	20.3	5.2	17.3	10.1	24.9	5.6	18.2	8.8	25.1	10.2	23.4
GPT-Image-1.5	10.1	19.8	10.1	19.9	6.2	12.9	2.9	15.1	3.2	15.7	5.6	15.0	10.7	19.3	2.9	12.5	3.5	12.7	6.0	15.8
Nano Banana	7.8	13.5	5.9	16.0	7.6	18.5	4.1	14.3	6.3	21.6	1.3	6.0	5.5	12.0	8.7	18.6	1.3	15.0	5.5	15.3
FLUX.2 pro	7.1	12.0	6.0	15.9	5.0	12.9	1.6	9.5	6.9	17.4	2.7	12.9	1.8	9.9	17.3	24.8	1.9	14.2	6.3	15.0
GPT-Image-1	11.4	20.1	7.0	15.2	10.5	21.1	3.9	15.1	5.1	15.4	2.7	11.4	1.3	12.2	2.9	12.6	2.6	11.8	5.2	14.9
Seedream-4.5	3.0	8.3	0.0	10.2	0.0	5.7	0.0	8.8	8.5	16.6	5.7	12.3	0.0	10.4	5.8	12.5	1.8	9.2	3.0	10.6
Open-source T2I Models (1024×1024, default settings, see Table 7)																				
FLUX.2 dev	1.4	4.1	3.6	**9.3**	**4.1**	9.3	3.0	11.4	10.0	18.4	1.4	7.5	5.6	**12.8**	1.1	**7.8**	1.3	**12.1**	**3.5**	**10.3**
FLUX.2 klein 9B	0.0	**4.2**	2.5	9.2	2.8	8.8	3.1	11.0	**10.1**	**19.0**	2.7	8.4	2.7	10.5	2.0	6.6	0.0	9.7	2.9	9.6
FLUX.2 klein 4B	0.0	1.8	**3.9**	8.3	2.7	**10.2**	**6.6**	**17.5**	5.3	11.7	0.0	7.7	**7.1**	12.4	0.0	6.1	2.7	7.1	2.9	8.9
FLUX.1 dev	**1.7**	2.2	2.9	6.7	3.2	6.5	5.8	11.3	7.4	15.1	**3.3**	**9.7**	3.3	6.1	**3.6**	6.6	0.0	4.8	**3.5**	7.7
FLUX.1 schnell	1.6	3.9	2.9	7.2	0.0	4.0	3.1	9.0	8.3	14.9	3.2	7.0	2.9	12.4	0.0	2.9	**3.4**	7.1	2.8	7.6
Qwen-Image 20B	0.0	1.5	0.0	2.2	1.5	4.5	6.2	11.0	7.8	13.1	3.1	7.1	1.6	7.4	0.0	2.7	1.6	5.3	2.2	5.7
SD3.5 large	0.0	0.6	1.5	3.8	0.0	2.8	5.4	8.9	1.5	5.3	1.6	3.1	1.5	9.7	0.0	2.9	1.4	4.7	1.3	4.5
SD3.5 medium	0.0	0.8	0.0	1.7	1.6	3.1	3.8	8.8	2.8	7.8	1.6	5.0	0.0	4.4	0.0	3.6	0.0	5.8	1.0	4.5
Sana 4.8B	0.0	0.7	0.0	1.9	0.0	2.7	1.9	5.2	6.0	10.5	0.0	3.6	1.7	5.3	0.0	4.4	0.0	2.4	1.0	4.1
Sana 0.6B	0.0	0.6	0.0	1.6	1.7	3.0	1.9	5.5	4.3	7.0	1.8	5.7	1.6	6.7	0.0	3.4	0.0	2.4	1.2	3.9
Sana 1.6B	1.5	1.7	3.4	4.0	1.7	2.6	1.7	6.7	3.1	5.8	0.0	1.1	1.8	6.3	0.0	2.7	0.0	2.0	1.4	3.6

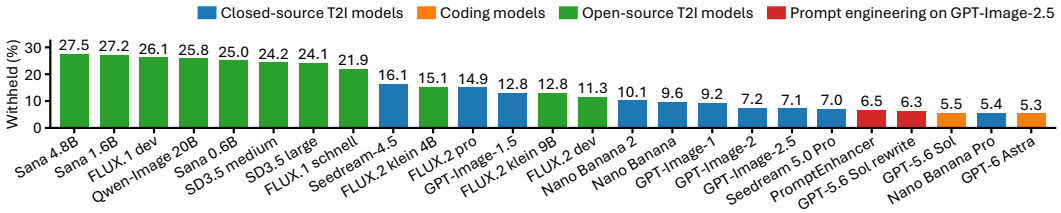

Figure 6: Grounding failure rate for different models. Lower is better. Closed-source T2I models are shown in blue, coding models in orange, and open-weight T2I models in green. Prompt Engineering in red. The Spearman correlation coefficient between failure rate and Olympiad strict scores is -0.92. It indicates that stronger models generally see lower grounding failures in our evaluation pipeline.

Case study: GPT-Image-2 *vs*. Nano Banana 2. They have comparable performance as shown in Table 1. GPT-Image-2 scores higher on text fidelity, symbolic systems, and 3D reasoning, while Nano Banana 2 is better on world-knowledge grounding, physical consistency, latent computation, and long-range consistency. It suggests that they were trained on different data. Meanwhile, a lower overall score does not necessarily mean that a model has lower scores on every capability axis.

4.3 Further Analysis and Insights

Comparing the grounding failure rate of various generation models. As discussed in Section 3.3, grounding failures happen when some generated content cannot be grounded by Gemini 2.5 Flash. This does not always mean bad generation: occluded objects may cause failure but may not be bad. In Fig. 6, we compare the 25 models of their grounding failures and find that their ranking

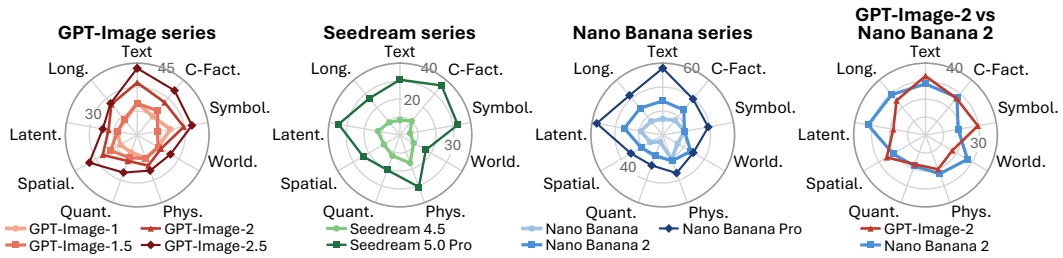

Figure 7: Performance of GPT, Seedream and Nano Banana series across nine axes. And the performance between GPT-Image-2 and Nano Banana 2. The performance improves as stronger models are released. This might not be the case for less strong models (see Appendix Fig. 13)

Table 2: Agreement between our evaluation and human judgment. The first three rows show human agreement on different types of questions. Last two rows compare evaluation with and without evidence grounding on 117 hard questions requiring grounding.

Setting	Agreement
All questions	95.3%
Easy questions	94.7%
Hard questions	95.7%
Hard w/o grounding	68.4%
Hard w/ grounding	91.5%

Table 3: GenOlympiad scores versus aesthetic quality across models. Aesthetic scores are rated from 0 to 10 by GPT-4.1 following PRISM-Bench. Higher is better. Coding models are marked by †.

	GenOlympiad		Aesthetic
Model	**Strict**	**Relaxed**	**Score**
GPT-Image-2.5	14.9	29.6	**5.51**
Nano Banana Pro	23.9	40.0	5.08
Seedream 5.0 Pro	15.1	28.4	4.62
GPT-6 Astra†	**40.5**	**56.1**	4.61
GPT-5.6 Sol†	37.5	52.4	4.61
Nano Banana	5.5	15.3	2.99

highly correlates with their GenOlympiad scores (strict) with a Spearman coefficient of -0.92. It suggests that a stronger model also tends to produce more easily grounded objects, which could be a result of post-training.

Do frontier models have improved GenOlympiad scores over releases? We evaluate different releases of frontier models *e.g.*, GPT-Image series, Seedream series, Nano Banana series. Based on Table 1, we plot radar diagrams in Fig. 7. Overall, later releases generally achieve higher scores, but the improvements are highly uneven across different axes. Some capabilities, such as text, spatial reasoning, and latent computation, improve substantially, while others, such as world knowledge, show much slower progress.

Correlation of GenOlympiad scores with human-based arena scores. We also analyze the correlation between GenOlympiad scores with human-based arena scores. The latter includes Image Arena, LMArena and Design Arena. Correlation results are shown in Fig. 8. Although not intended, we observe strong correlation between GenOlympiad score and the arena scores. In other words, models that human prefers often obtain higher scores on GenOlympiad. On the other hand, outliers do exist. For example, Nano Banana Pro has much higher GenOlympiad score but only ranks in the middle of the three arenas. It indicates that differences exist between GenOlympiad and arenas.

4.4 Evaluator validation

Agreement with human judgment. We check how the automatic evaluation protocol agrees with humans. So we recruited 41 annotators on Prolific and collected human answers for 365 questions consisting of 131 easy questions and 234 hard questions. The evaluated images were generated by five T2I models and randomly selected. Annotators see the same generated image and the same questions in our evaluation program but do not see the model name, the VLM answers, or the final score. Each question was assigned to 3 annotators, and we use the majority answer as the human answer. The human answers are then sent to the same deterministic scoring program described in Section 3.3 for the final decision, *e.g.*, the true/false answer of 'Do these dice sum up to 131?'.

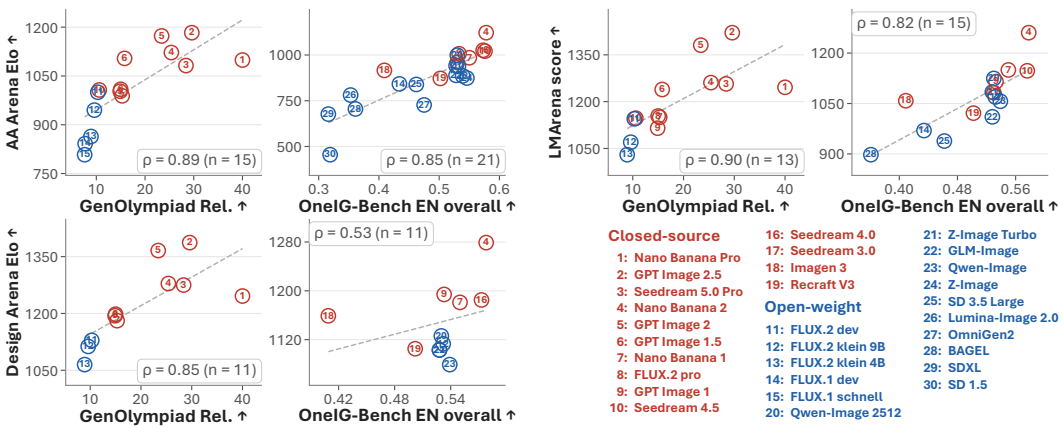

Figure 8: Correlation between GenOlympiad relaxed scores and three human-preference-based arena scores. OneIG-Bench is included for comparison. GenOlympiad seems to be a good proxy of Elo scores. Results for other benchmarks are shown in Appendix Fig. 15.

From Table 2, we have two observations. First, the pass/fail agreement between human and automatic evaluation is 95.3%(348/365). The agreement is 94.7% (124/131) on easy questions and 95.7% (224/234) on hard. Second, evidence grounding is effective. We test this on 117 questions that require evidence grounding, and here the agreement means that the answer is the same as the human answer. If we directly ask VLM without grounding, the agreement between human and VLM is 68.4% (80/117). When we perform evidence grounding, the agreement becomes 91.5% (107/117).

Agreement between different VLMs. By default we use Gemini 2.5 Flash to answer the evaluation questions, mainly because of the low cost. Here we replace Gemini 2.5 Flash with GPT-5.6 and GPT-4.1 on the same human-study questions and keep the deterministic scoring part unchanged. The pairwise agreement between the three VLM judges is consistently greater than 96%, as shown in Table 4. It suggests that our evaluation protocol is robust with the current judge.

Table 4: Pairwise agreement between three different VLM judges on the same 365 set of human evaluation questions.

VLM pair	Agreement
Gemini 2.5 Flash & GPT-5.6	96.7%
Gemini 2.5 Flash & GPT-4.1	96.7%
GPT-5.6 & GPT-4.1	97.6%

5 Conclusion

We introduce GenOlympiad, a challenging set of generation tasks covering nine capability axes to stress-test frontier T2I models and find where they fail. For each prompt, we include an evaluation program consisting of easy and hard questions. On average there are 2.2 hard questions per prompt. We propose a grounding method for such hard question, where we draw helping lines on the image in an iterative way. We show that this method effectively improves agreement with human judgments.We benchmark many strong models including open-source and close-source T2I models and show that the best T2I model, Nano Banana Pro, only has a strict score of 23.9. While coding models have higher strict score of 40.5, they suffer from poorer aesthetics. We also derive interesting insights, *e.g.*, how a model family improves with their newer releases, prompt enhancer is useful, and GenOlympiad scores strongly correlate with human Elo scores.

Limitation. GenOlympiad focuses on prompts that current frontier models find difficult, so the scores do not represent their performance on easy prompts. Moreover, GenOlympiad offers limited signal for non-frontier T2I models. Some prompts in GenOlympiad are too long to fit into the context of open-weight models, so for some open-source models, around 20 tasks do not have image generated or scored.Some scores are also withheld because the VLM judge does not correctly ground evidence. Finally, our evaluation is cost more requiring 59.5 VLM reads per image on average.

AI USE STATEMENT

In this work, we used generative AI tools to assist with generating synthetic benchmark tasks, developing the conceptual framework of the nine capability axes, designing and providing feedback on experiments, cleaning and reformatting the benchmark dataset, implementing evaluation programs, translating and summarizing prompts used while constructing the benchmark, as described in Appendix A.4, evaluating generated images as part of the scoring pipeline, and providing preliminary interpretation of intermediate experimental results. We have not used generative AI tools for any other tasks with required disclosure, and tasks involving formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, and supporting qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for brainstorming, sourcing and searching for information, summarizing and analyzing existing literature. AI-assisted benchmark tasks and evaluation programs were checked through manual spot checks and additional AI-based verification. The evaluation pipeline was separately validated against human annotations as described in the paper. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Human annotations used in this work were collected from 41 participants recruited through Prolific. Participants were compensated at a rate of £13.20 per hour. We did not request or collect participants' names, email addresses, or other directly identifying information, nor did the study require participants to provide sensitive personal information.

REFERENCES

Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 9448–9458, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/97af07a14cacba681feacf3012730892-Abstract.html.

Black Forest Labs. FLUX.1, 2024. URL https://github.com/black-forest-labs/flux. Accessed: 2026-09-14.

Black Forest Labs. FLUX.2, 2025. URL https://bfl.ai/blog/flux-2. Accessed: 2026-09-14.

ByteDance Seed. Seedream 4.5, 2025. URL https://seed.bytedance.com/en/seedream4_5. Accessed: 2026-07-21.

ByteDance Seed. Seedream 5.0 Pro, 2026. URL https://seed.bytedance.com/en/seedream5_0_pro. Accessed: 2026-09-10.

Center for AI Safety, Scale AI, and HLE Contributors Consortium. A benchmark of expert-level academic questions to assess AI capabilities. *Nature*, 649:1139–1146, 2026. doi: 10.1038/s41586-025-09962-4. URL https://arxiv.org/abs/2501.14249.

Jingjing Chang, Yixiao Fang, Peng Xing, Shuhan Wu, Wei Cheng, Rui Wang, Xianfang Zeng, Gang Yu, and Hai-Bao Chen. Oneig-bench: Omni-dimensional nuanced evaluation for image generation. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/e9e9e5428189a3e49479547ef917e88d-Abstract-Datasets_and_Benchmarks_Track.html.

Kaijie Chen, Zihao Lin, Zhiyang Xu, Ying Shen, Yuguang Yao, Joy Rimchala, Jiaxin Zhang, and Lifu Huang. R2i-bench: Benchmarking reasoning-driven text-to-image generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 12595–12630, 2025.

Zijie Chen, Lichao Zhang, Fangsheng Weng, Lili Pan, and Zhenzhong Lan. Tailored visions: Enhancing text-to-image generation with personalized prompt rewriting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 7727–7736. IEEE, 2024. doi: 10.1109/CVPR52733.2024.00738. URL https://doi.org/10.1109/CVPR52733.2024.00738.

Jaemin Cho, Yushi Hu, Jason Baldridge, Roopal Garg, Peter Anderson, Ranjay Krishna, Mohit Bansal, Jordi Pont-Tuset, and Su Wang. Davidsonian scene graph: Improving reliability in fine-grained evaluation for text-to-image generation. In *International conference on learning representations*, 2024.

François Chollet. On the measure of intelligence. *CoRR*, abs/1911.01547, 2019. URL http://arxiv.org/abs/1911.01547.

François Chollet, Mike Knoop, Gregory Kamradt, Bryan Landers, and Henry Pinkard. ARC-AGI-2: A new challenge for frontier AI reasoning systems. *CoRR*, abs/2505.11831, 2025. doi: 10.48550/ARXIV.2505.11831. URL https://doi.org/10.48550/arXiv.2505.11831.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/esser24a.html.

Rongyao Fang, Aldrich Yu, Chengqi Duan, Linjiang Huang, Shuai Bai, Yuxuan Cai, Kun Wang, Si Liu, Xihui Liu, and Hongsheng Li. FLUX-reason-6m & PRISM-bench: A million-scale text-to-image reasoning dataset and comprehensive benchmark. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=cPzgZnpVbN.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/a3bf71c7c63f0c3bcb7ff67c67b1e7b1-Abstract-Datasets_and_Benchmarks.html.

Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, Olli Järviniemi, Matthew Barnett, Robert Sandler, Matej Vrzala, Jaime Sevilla, Qiuyu Ren, Elizabeth Pratt, Lionel Levine, Grant Barkley, Natalie Stewart, Bogdan Grechuk, Tetiana Grechuk, Shreepranav Varma Enugandla, and Mark Wildon. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in AI. *CoRR*, abs/2411.04872, 2024. doi: 10.48550/ARXIV.2411.04872. URL https://doi.org/10.48550/arXiv.2411.04872.

Google. Nano Banana (Gemini 2.5 Flash Image), 2025a. URL https://developers.googleblog.com/en/introducing-gemini-2-5-flash-image/. Accessed: 2026-09-01.

Google. Nano Banana Pro (Gemini 3 Pro Image), 2025b. URL https://blog.google/innovation-and-ai/products/nano-banana-pro/. Accessed: 2026-09-14.

Google. Nano Banana 2 (Gemini 3.1 Flash Image), 2026. URL https://blog.google/innovation-and-ai/technology/ai/nano-banana-2/. Accessed: 2026-07-21.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 7514–7528, 2021.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6626–6637, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/8a1d694707eb0fefe65871369074926d-Abstract.html.

Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. ELLA: equip diffusion models with LLM for enhanced semantic alignment. *CoRR*, abs/2403.05135, 2024. doi: 10.48550/ARXIV.2403.05135. URL https://doi.org/10.48550/arXiv.2403.05135.

Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A Smith. Tifa: Accurate and interpretable text-to-image faithfulness evaluation with question answering. In *2023 ieee/cvf international conference on computer vision (iccv)*, pp. 20349–20360. IEEE, 2023.

Kaiyi Huang, Chengqi Duan, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench++: An enhanced and comprehensive benchmark for compositional text-to-image generation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(5):3563–3579, 2025. doi: 10.1109/TPAMI.2025.3531907. URL https://doi.org/10.1109/TPAMI.2025.3531907.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R. Narasimhan. Swe-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Amita Kamath, Kai-Wei Chang, Ranjay Krishna, Luke Zettlemoyer, Yushi Hu, and Marjan Ghazvininejad. Geneval 2: Addressing benchmark drift in text-to-image evaluation. *CoRR*, abs/2512.16853, 2025. doi: 10.48550/ARXIV.2512.16853. URL https://doi.org/10.48550/arXiv.2512.16853.

Baiqi Li, Zhiqiu Lin, Deepak Pathak, Jiayao Li, Yixin Fei, Kewen Wu, Tiffany Ling, Xide Xia, Pengchuan Zhang, Graham Neubig, and Deva Ramanan. Genai-bench: Evaluating and improving compositional text-to-visual generation. *CoRR*, abs/2406.13743, 2024. doi: 10.48550/ARXIV.2406.13743. URL https://doi.org/10.48550/arXiv.2406.13743.

Niantong Li, Guangzheng Hu, Weixu Qiao, Ying Ba, Qichen Hong, Shijun Shen, Jinlin Wang, Fan Zhou, Jianye Kang, Xin Shang, Ziyi He, Wei Wang, Dalin Li, Jiahao Li, Jie Zhang, Kaiyuan Gao, Kun Yan, Lihan Jiang, Ningyuan Tang, Shengming Yin, Tianhe Wu, Xiao Xu, Xiaoyue Chen, Yuxiang Chen, Yan Shu, Yanran Zhang, Yilei Chen, Yixian Xu, Zekai Zhang, Zhendong Wang, Zihao Liu, Zikai Zhou, Hongzhu Shi, Yi Wang, Bing Zhao, Hu Wei, Lin Qu, and Chenfei Wu. Qwen-image-bench: From generation to creation in text-to-image evaluation. *CoRR*, abs/2605.28091, 2026. doi: 10.48550/ARXIV.2605.28091. URL https://doi.org/10.48550/arXiv.2605.28091.

Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *European Conference on Computer Vision*, pp. 366–384. Springer, 2024.

Ruiyao Liu, Hui Shen, Ping Zhang, Yunta Hsieh, Yifan Zhang, Jing Xu, Sicheng Chen, Junchen Li, Jiawei Lu, Jianing Ma, et al. Mathgen: Revealing the illusion of mathematical competence through text-to-image generation. *arXiv preprint arXiv:2603.27959*, 2026.

Yuwei Niu, Munan Ning, Mengren Zheng, Bin Lin, Peng Jin, Jiaqi Liao, Kun-Peng Ning, Bin Zhu, and Li Yuan. WISE: A world knowledge-informed semantic evaluation for text-to-image generation. *CoRR*, abs/2503.07265, 2025. doi: 10.48550/ARXIV.2503.07265. URL https://doi.org/10.48550/arXiv.2503.07265.

OpenAI. GPT-4.1, 2025a. URL https://openai.com/index/gpt-4-1/. Accessed: 2026-09-04.

OpenAI. GPT-Image-1, 2025b. URL https://openai.com/index/introducing-4o-image-generation/. Accessed: 2026-09-01.

OpenAI. GPT-Image-1.5, 2025c. URL https://openai.com/index/new-chatgpt-images-is-here/. Accessed: 2026-07-27.

OpenAI. GPT-5.6 Sol, 2026a. URL https://openai.com/index/gpt-5-6/. Accessed: 2026-09-15.

OpenAI. GPT-6 Astra, 2026b. URL https://openai.com/index/gpt-6-astra/. Accessed: 2026-09-14.

OpenAI. GPT-Image-2, 2026c. URL https://openai.com/index/introducing-chatgpt-images-2-0/. Accessed: 2026-07-18.

OpenAI. GPT-Image-2.5, 2026d. URL https://openai.com/index/introducing-chatgpt-images-2-5/. Accessed: 2026-09-09.

Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. Teaching CLIP to count to ten. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 3147–3157. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00294. URL https://doi.org/10.1109/ICCV51070.2023.00294.

Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 2226–2234, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/8a3363abe792db2d8761d6403605aeb7-Abstract.html.

Samin Mahdizadeh Sani, Max Ku, Nima Jamali, Matina Mahdizadeh Sani, Paria Khoshtab, Wei-Chieh Sun, Parnian Fazel, Zhi Rui Tam, Thomas Chong, Edisy Kin Wai Chan, Donald Wai Tong Tsang, Chiao-Wei Hsu, Ting Wai Lam, Ho Yin Sam Ng, Chiafeng Chu, Chak-Wing Mak, Keming Wu, Hiu Tung Wong, Yik Chun Ho, Chi Ruan, Zhuofeng Li, I-Sheng Fang, Shih-Ying Yeh, Ho Kei Cheng, Ping Nie, and Wenhu Chen. Imagenworld: Stress-testing image generation models with explainable human evaluation on open-ended real-world tasks. *CoRR*, abs/2603.27862, 2026. doi: 10.48550/ARXIV.2603.27862. URL https://doi.org/10.48550/arXiv.2603.27862.

Jaskirat Singh and Liang Zheng. Divide, evaluate, and refine: Evaluating and improving text-to-image alignment with iterative VQA feedback. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/dfd0bd56e8a6f82d1619f5d093d5f9ca-Abstract-Conference.html.

Stability AI. Stable Diffusion 3.5, 2024. URL https://github.com/Stability-AI/sd3.5. Accessed: 2026-09-06.

Kaiyue Sun, Rongyao Fang, Chengqi Duan, Xian Liu, Aoxue Li, and Xihui Liu. T2i-reasonbench: Benchmarking reasoning-informed text-to-image generation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 8919–8944, 2026.

Linqing Wang, Zhiyong Xu, Ximing Xing, Yiji Cheng, Zhiyuan Zhao, Donghao Li, Tiankai Hang, Zhenxi Li, Jiale Tao, Qixun Wang, Ruihuang Li, Comi Chen, Xin Li, Mingrui Wu, Xinchi Deng, Shuyang Gu, Chunyu Wang, and Qinglin Lu. Promptenhancer: Taming your rewriter for text-to-image generation via fine-grained reward. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14895–14904, June 2026.

Zhaokai Wang, Penghao Yin, Xiangyu Zhao, Changyao Tian, Yu Qiao, Wenhai Wang, Jifeng Dai, and Gen Luo. Genexam: A multidisciplinary text-to-image exam. *CoRR*, abs/2509.14232, 2025. doi: 10.48550/ARXIV.2509.14232. URL https://doi.org/10.48550/arXiv.2509.14232.

Xinyu Wei, Jinrui Zhang, Zeqing Wang, Hongyang Wei, Zhen Guo, and Lei Zhang. Tiif-bench: How does your T2I model follow your instructions? *CoRR*, abs/2506.02161, 2025. doi: 10.48550/ARXIV.2506.02161. URL https://doi.org/10.48550/arXiv.2506.02161.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Shengming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report. *CoRR*, abs/2508.02324, 2025. doi: 10.48550/ARXIV.2508.02324. URL https://doi.org/10.48550/arXiv.2508.02324.

Xindi Wu, Dingli Yu, Yangsibo Huang, Olga Russakovsky, and Sanjeev Arora. Conceptmix: A compositional image generation benchmark with controllable difficulty. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng

Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/9c3563bbeb2ad7f3b3b8ed0fcd3b440f-Abstract-Datasets_and_Benchmarks_Track.html.

Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, and Song Han. SANA: efficient high-resolution text-to-image synthesis with linear diffusion transformers. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=N8Oj1XhtYZ.

Wenxi Yang, Yuzhong Zhao, Fang Wan, and Qixiang Ye. Thinking with images via self-calling agent. *CoRR*, abs/2512.08511, 2025. doi: 10.48550/ARXIV.2512.08511. URL https://doi.org/10.48550/arXiv.2512.08511.

Junyan Ye, Dongzhi Jiang, Zihao Wang, Leqi Zhu, Zhenghao Hu, Zilong Huang, Jun He, Zhiyuan Yan, Jinghua Yu, Hongsheng Li, Conghui He, and Weijia Li. Echo-4o: Harnessing the power of gpt-4o synthetic images for improved image generation. *CoRR*, abs/2508.09987, 2025. doi: 10.48550/ARXIV.2508.09987. URL https://doi.org/10.48550/arXiv.2508.09987.

A APPENDIX

A.1 MORE EXAMPLES AND BENCHMARK STATISTIC

More Examples. More example images of GenOlympiad are shown in Fig.17. We have many different topic for each capability axis. The task topics in each axis can be seen in Fig. 12. Each prompt passed our task selection protocol described in Appendix A.3. When designing tasks, we tried to limit simple changes such as changing 'counting apples' to 'counting footballs'. We also tried to make the mechanism of the tasks as diverse as possible. For example, our counting tasks include counting absent seats while other seats occupied by people, counting caps of one color among many mixed-color caps in Fig.4, and counting how many beads of the same color are next to each other on a necklace.

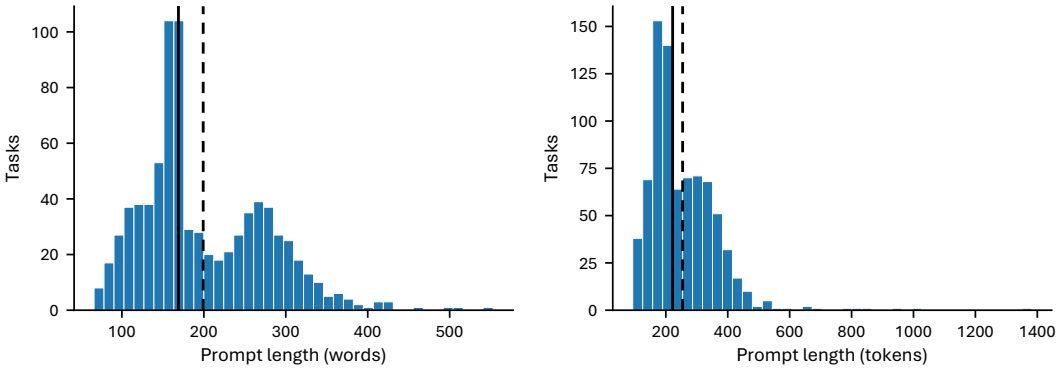

Figure 9: Prompt length distribution, **Left:** word count distribution, **Right:** Token distribution using `o200k_base`. **Solid line**: median, **Dashed line**: mean.

Prompt Length. We show the prompt length distribution of GenOlympiad in Fig. 9. The prompts contain 199 words in average, with a median of 169 words. We use `o200k_base` to count tokens, each prompt has 254 tokens on average, and the median is 222 tokens. Most prompts are shorter than 512 tokens.

Is GenOlympiad stably hard? We benchmark GPT-Image-2.5 sunburst twice, the performance between the 2 attempts kept stable. Result is shown in Tab. 5. The performance of the 2 attempts is stable: the strict score is 14.9 and 14.2 and the relaxed score is 29.6 and 29.3. Besides, there are 702 tasks that get the same score in both attempts. Though there are some different in some axis in the 2 attempts, GenOlympiad is overall stably hard.

Table 5: GenOlympiad results for GPT-Image-2.5 Sunburst for its first and second attempts of GenOlympiad across nine capability axes. Best result in each column is highlighted in **bold**. C-Fact.: counterfactual instruction following; Symbol.: symbolic systems; World.: world-knowledge grounding; Phys.: physical consistency; Quant.: quantity and precision; Spatial.: spatial and 3D reasoning; Latent.: latent computation; Long.: long-range consistency. Overall is the mean score over all scored tasks; Str. and Rel. are strict and relaxed scores.

Model	Text		C-Fact.		Symbol.		World.		Phys.		Quant.		Spatial.		Latent.		Long.		Overall	
	Str.	Rel.	Str.	Rel.	Str.	Rel.	Str.	Rel.	Str.	Rel.	Str.	Rel.	Str.	Rel.	Str.	Rel.	Str.	Rel.	Str.	Rel.
GPT-Image-2.5	**30.4**	**41.8**	**23.0**	**36.4**	**20.5**	**34.8**	6.8	24.1	8.1	23.7	**13.2**	25.0	**18.4**	**34.6**	6.5	21.9	9.8	25.9	**14.9**	**29.6**
GPT-Image-2.5	25.6	40.3	18.4	35.1	13.5	27.8	**11.5**	**25.4**	**10.8**	**27.9**	12.7	**26.2**	16.5	31.1	**10.4**	**24.2**	**9.9**	**26.3**	14.2	29.3

Axis Correlation. Although we try to design different axes to measure different capabilities of T2I models, it is unavoidable that there are still many overlaps between prompts not only in the same axes, but also in different axes. For example, **physical consistency** contains many tasks that is the combination of **latent computation** and **world-knowledge grounding**. **Text fidelity** is also

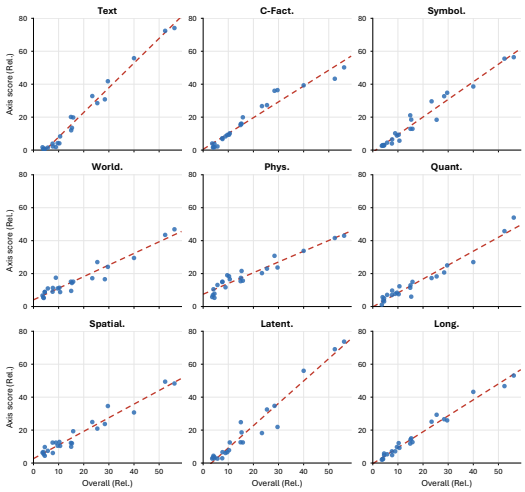

Figure 10: Linear regression for calculating partial correlation.

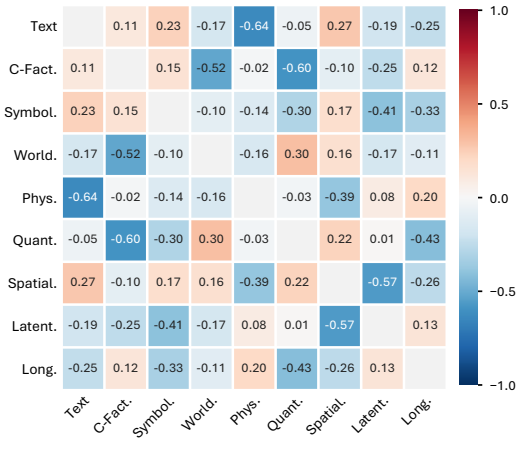

Figure 11: Spearman correlation matrix across capability axes.

Table 6: Pass/fail distribution between human judgments and our automatic evaluator.

	Auto Pass	**Auto Fail**	**Total**
Human Pass	174	13	187
Human Fail	4	174	178
Total	178	187	365

unavoidable to have some tasks that requires **quantitative precision** *e.g.*, there should be 12 lines of text end with -ation. So we further measure the correlation between axes. The correlation matrix can be seen in Fig. 11. Most pairs of axes have low correlation, while several pairs even show negative correlation.

The method we use to draw the correlation matrix. We take a classic statistics method called **partial correlation** to calculate the correlation between each axis. We do not directly compare the scores of two axes, because stronger models usually get higher scores on many axes at the same time. So we take methods to mitigate the bias from models' capability by using linear regression. For each axis, we first predict what score a model should get based on its overall performance using linear regression shown in Fig. 10. We then compare whether the model performs better or worse than this expected score on different axes. For example, if models that perform better than expected on **latent computation** can perform either better or worse than expected on **physical consistency**, then these two axes have a low spearman correlation.

A.2 Additional Analysis of Evaluator Agreement

The distribution of pass/fail of human agreement To ensure that our high agreement between the automatic evaluation protocol and human judgments is not because the imbalanced distribution of pass and fail, we further report the confusion matrix in Tab. 6. According to the confusion matrix, our agreement is not mainly affected by pass or fail.

A.3 How do we build GenOlympiad?

We designed an automated agentic task generate protocol to help us create high quality tasks. We provide simplified prompts of the protocol in Appendix A.4

This protocol is made up of several parts. We first build about 1600 task descriptions by using a brainstorming agent powered by Claude Opus 5. We provided many positive examples and negative

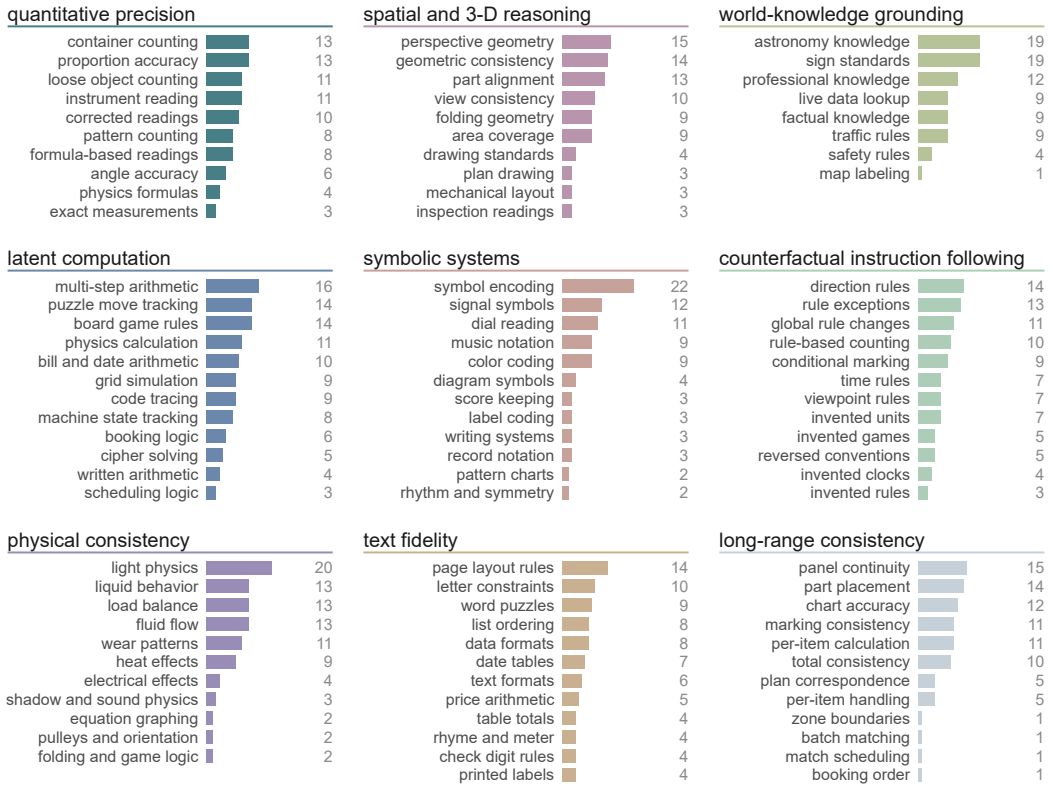

Figure 12: Different Topics for each axis.

examples designed by human, then use the prompt in Appendix A.4 to help us conduct large scale brainstorming. Then we do the following check, filtering, then build tasks to get the final 801 tasks.

- **Detailed definition of Each Axis**. The detailed explanation about each capability axis with examples to guide Claude Opus 5 how to generate tasks.
- **Few-shot Example for high-quality tasks**. We manually designed several origin T2I prompts for each capability as few-shot examples for agent. Each prompt reveals a typical fail of GPT-Image-2.
- **Example to build evaluation program**. We designed a modular evaluation framework consists of 43 operators, each operator contains an atomic operation *e.g.*, Cut the image by using coordination we provide.
- **Remove low quality tasks**. We designed few-shot samples for agent to let it know what the low quality tasks look like, and prevent it from creating low quality tasks such as doing a very simple change like changing 'counting apples' to 'counting footballs'.
- **Remove easy tasks**. We have a difficulty review mechanism that can filter easy tasks out and keep difficult tasks. While building the benchmark, we conduct the review using five frontier T2I models: GPT-Image-2, Nano Banana 2, GPT-Image-1.5, FLUX.2 pro, and Seedream-4.5. A task is kept if at least two models receive a score, fewer than three models receive a full score of 100, and either the average score is below 35, or the average score is below 60 with at least one model receiving a score of 0. This keeps tasks that some models can solve but others fail, so our benchmark can better distinguish different models.

Although we regenerate and evaluate these models after building the benchmark, difficulty filtering may still systematically lower the scores of the five models. To reduce this bias, we use five different frontier models from different model families, instead of using only one model or several models from the same family.

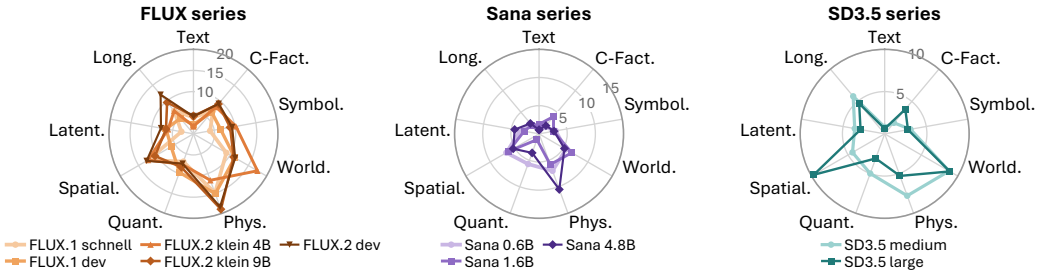

Figure 13: Performance of the FLUX series, Sana series and Stable Diffusion 3.5 series across nine axes. For these less strong open-source models, a newer or larger models' performance are not always better. Especially in some axes *e.g.*, World knowledge grounding of FLUX series, Text Fidelity of Sana series, Physical consistency of Stable Diffusion 3.5, newer or larger models' performance getting worse instead.

Table 7: Generation settings of open-source T2I models. Steps and guidance scale are the default values. The context length is the maximum number of tokens of the primary text encoder; we set it to the largest value the pipeline accepts, so that the full prompt is visible. All images are 1024×1024.

Model	Steps	Guidance	Context length	Precision
FLUX.2 dev	50	4.0	512	bf16
FLUX.2 klein 9B	4	1.0	1024	bf16
FLUX.2 klein 4B	4	1.0	1024	bf16
FLUX.1 dev	28	3.5	512	bf16
FLUX.1 schnell	4	0.0	512	bf16
Qwen-Image 20B	50	4.0	1024	bf16
SD3.5 large	28	3.5	512	bf16
SD3.5 medium	40	4.5	512	bf16
Sana 4.8B	20	4.5	512	bf16
Sana 1.6B	20	4.5	512	fp16
Sana 0.6B	20	4.5	512	fp16

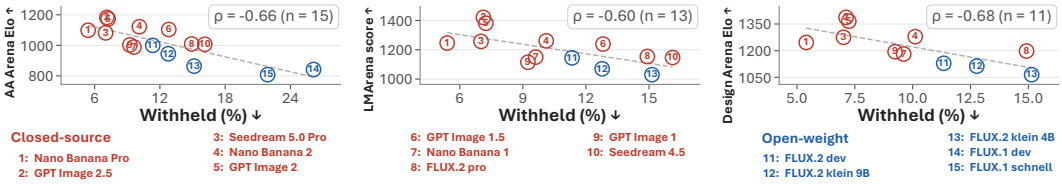

Figure 14: Correlation between withheld rate and human-preference arena scores. Higher arena scores correspond to lower withheld rates on GenOlympiad.

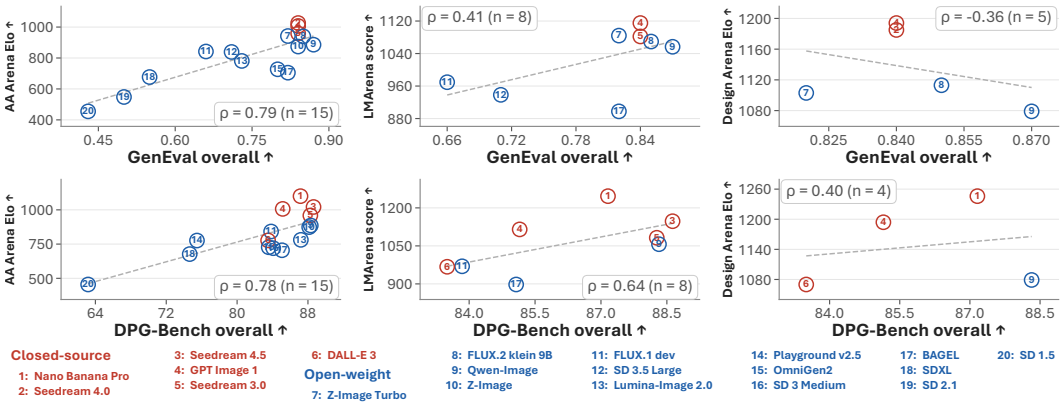

Figure 15: Correlation between arena and some other T2I benchmark GenEval and DPG-Bench.



Figure 16: Sample images of each models with their aesthetic and GenOlympiad relaxed score. Prompts were shorten, not original.

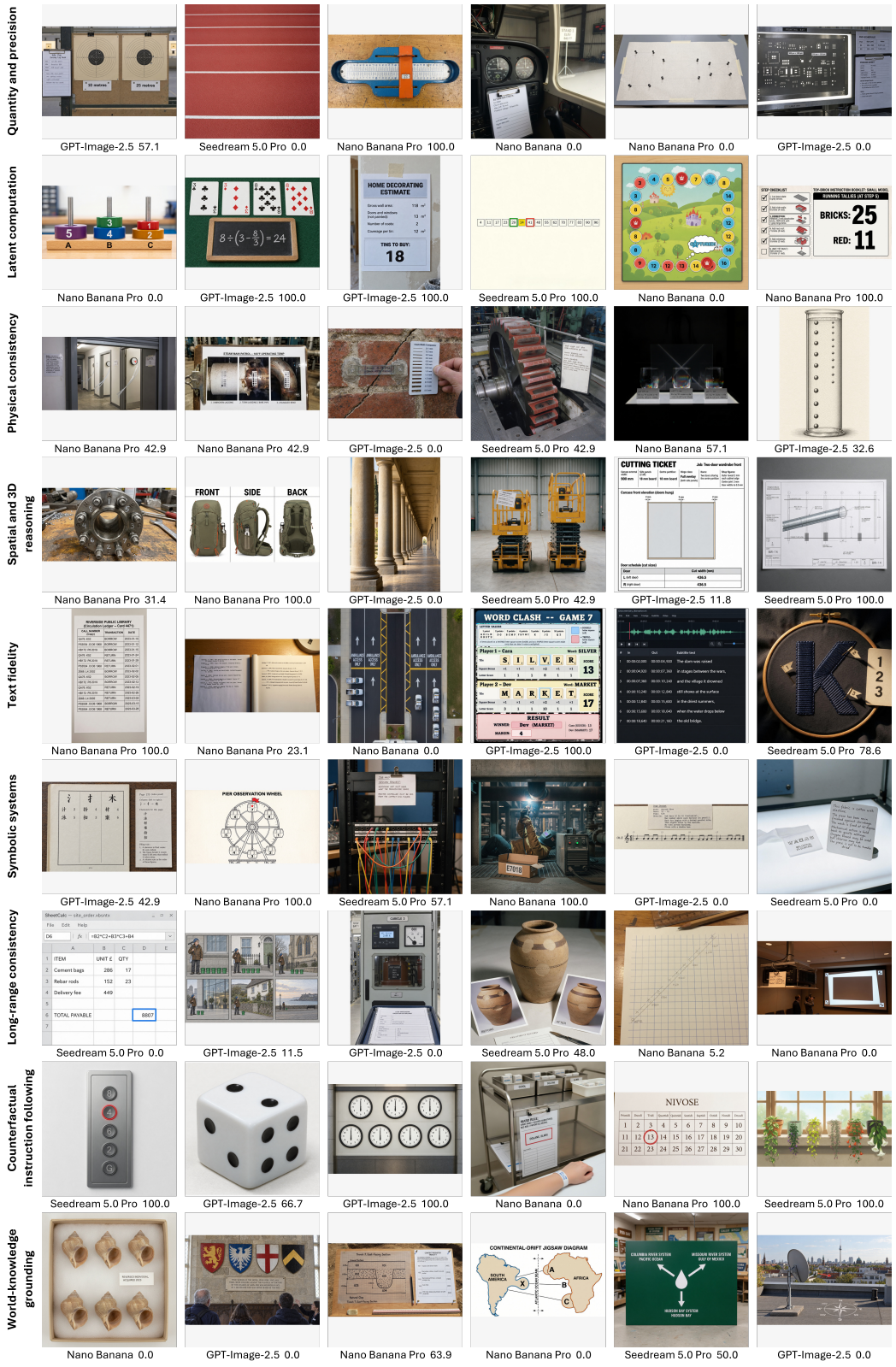

Figure 17: More example images for each capability axis.

A.4 PROMPT TEMPLATES

This section gives the prompts used in this paper, all prompts in abridged. We first give the few-shot examples that we wrote by hand before building the benchmark. We then give the instructions of the agents in the task generation protocol of Appendix A.3. We evaluate coding models by giving system prompt. Some of them were written in Chinese and are translated here. All these following prompts are summarized and translated by Claude Fable 5.1.

Positive examples. Each prompt below looks ordinary, but frontier T2I models fail it for a reason that can be named. Where we recorded the outcome of our first trials, it is given after the prompt.

1. **Exact geometry of a technical drawing.**

   Generate a precise 2D CAD-style engineering drawing in strict orthographic front view of a flat mechanical panel. Visual style: - pure white background - thin black uniform lines - no perspective - no shading - no textures - no shadows - no color - no text - no dimension annotations - geometry only - crisp technical drawing appearance Geometry: 1. Outer boundary: a rectangle of width 300 units and height 200 units. 2. At the geometric center of the rectangle, draw two concentric circles: - outer circle diameter 60 units - inner circle diameter 30 units 3. Draw four identical circular holes of diameter 16 units, centered at: - (60, 150) - (240, 150) - (60, 50) - (240, 50) 4. Draw a horizontal slot centered at (150, 50), with total length 120 units and height 12 units, semicircular ends. 5. Draw two vertical rectangular slots, each width 12 units and height 70 units: - left slot center at (110, 100) - right slot center at (190, 100) 6. Draw a square cutout of side length 24 units centered at (150, 150). 7. The square cutout must be exactly centered horizontally with the concentric circles. 8. The two vertical slots must be symmetric about the vertical centerline of the plate. 9. All geometry must preserve exact proportions, alignment, symmetry, spacing, and relative size. Important: - The output must be a clean technical diagram. - All relative measurements must be preserved exactly. - Do not artistically reinterpret the drawing.

2. **Two tilted mirrors.**

   Two identical mirrors on a vanity, one tilted 10 degrees backward, one 10 degrees forward. A person at equal distance appears full-body in the backward-tilted mirror but cut off at the feet in the forward-tilted one.

3. **Tree rings with an exact count.**

   A polished 60cm tree-trunk cross-section with exactly 47 concentric annual rings, a label reading 'Tree age: 47 years', a centimeter scale along the edge, plus two visibly narrow drought-ring clusters and one wide optimal-growth band at specific radii.

4. **Falling Slinky.**

   A 0.6m metallic Slinky released from rest is photographed mid-fall: the top coils have collapsed and fallen $\sim$3m while the bottom end remains motionless at the release height, the coils forming a compressed cluster whose spacing increases non-linearly from bottom to top.

5. **Text printed on one grain of rice.**

   A realistic macro photograph of a white ceramic plate filled with uncooked white rice grains. The camera is very close to the plate, with shallow depth of field, but the rice grains must remain individually visible. In the exact center of the image, there is one single rice grain lying horizontally on top of the pile. On this one rice grain, write the tiny black printed text: 'GPT-IMAGE-2' The text must be physically printed directly on the surface of that rice grain, following the grain's curved shape. The letters must be extremely small but still clearly readable when zoomed in. Do not write any text on the plate, table, background, or other rice grains. Only the one central rice grain contains text. The image should look like a real macro lens photo, with natural lighting, realistic shadows, and a believable scale difference between the rice grain and the tiny text.

6. **Three mutually consistent annotation panels.**

   Generate a three-panel computer vision annotation benchmark image. The output must contain three equal-sized panels arranged horizontally from left to right, separated by narrow white gutters. Do not add titles, legends, or decorative elements. All three panels must represent the exact same scene with exactly eight apples on a kitchen table. All three panels must use the same camera view, crop, coordinate system, scale, and object layout. Panel 1: RGB image. A realistic RGB photograph of a kitchen table with exactly eight apples. The apples should vary in size, color, rotation, and pose. Some apples should touch or slightly overlap each other, but every apple must remain at least partially visible as a separate instance. Use realistic lighting, shadows,

and perspective. Panel 2: instance segmentation mask. A clean pixel-aligned instance segmentation mask of Panel 1. Each of the eight apples must be represented by a different flat solid color, chosen from this fixed set: red, blue, green, yellow, magenta, cyan, orange, and purple. The table and background must be pure white. The mask must exactly match the visible shapes, boundaries, positions, scales, and occlusion relationships of the apples in Panel 1. Hidden parts of occluded apples must not appear in the mask. Do not include text, labels, outlines, shadows, gradients, or textures. Panel 3: object detection bounding box annotation. A clean annotation view of the same scene from Panel 1. Use a pure white background and draw exactly eight tight axis-aligned rectangular bounding boxes, one around each apple. Each bounding box must tightly enclose the full visible apple, including the visible stem if present. Each box must use the same color as the corresponding apple instance in Panel 2. Add a small numeric label from 1 to 8 at the top-left corner of each bounding box. Do not draw any extra boxes. Do not miss any apple. Do not render the apples themselves in Panel 3; show only the white background, the eight colored bounding boxes, and the numeric labels. Define instance identity consistently across all panels by sorting apples from left to right according to the x-coordinate of their visible-region centroids; if two apples have similar x-coordinates, break ties from top to bottom. The apple assigned identity 1 corresponds to label 1, identity 2 corresponds to label 2, and so on. The three panels must be mutually consistent. The same apple instance must have the same identity, position, size, overlap relationship, and visible extent across Panel 1, Panel 2, and Panel 3.

7. **A chat screenshot whose picture contradicts its text.**

   Generate a fictional AI chat screenshot used as a visual benchmark example. The chat interface shows a user prompt that confidently asks: 'Draw a human hand with exactly five fingers. The final image must contain only five fingers.' The assistant response appears to comply and says: 'Here is a realistic hand with exactly five fingers.' However, the generated image shown under the assistant response clearly depicts a realistic human hand with six fingers. The sixth finger should be visible but not exaggerated, making the failure subtle at first glance but undeniable upon inspection. Use a clean modern chat UI layout, rounded bubbles, white background, and a realistic image preview. The scene should demonstrate an instruction-following failure: the prompt asks for five fingers, but the output contains six. Do not use real ChatGPT logos, OpenAI logos, or any real platform branding. This must be a fictional mockup.

8. **Offside at the moment of the pass.** *(translated)*

   Generate a realistic football match scene that captures the exact moment of a pass. The image must clearly show the passer, the receiver, at least three defenders, the direction of the goal, and the pitch markings. At the moment of the pass the receiver must be in an offside position: the receiver must be behind or level with the second-to-last defender. The ball must be clearly leaving the passer's foot at the moment of the pass.

9. **Gomoku, one move from winning.** *(translated)*

   Generate a realistic top-down view of a gomoku game that shows a plausible position, in which every stone is placed with a tactical purpose. Do not place stones at random. Black needs only one move to win: one more black move makes five in a row, but in the current image Black must not already have five in a row. White should have several sensible defensive stones nearby that block or counter Black's attack. The stones elsewhere on the board should also be laid out sensibly, as in a real mid-to-late game. Render the board as a clean wooden gomoku board, with black and white stones placed exactly on the intersections, viewed from directly above.

10. **A bar chart that describes its own image.** *(translated)*

    Generate a realistic office scene. On the wall there is a large whiteboard titled 'Color distribution of this image'. The whiteboard shows a bar chart with four bars, labelled green, white, brown and blue. The heights of the four bars must accurately reflect the proportions of the pixels that are actually visible as green, white, brown and blue in the whole image. The room should contain green plants, white walls, brown wooden furniture and blue objects, so that all four colors are clearly present. The bar chart must be large, clearly legible and easy to measure visually.

11. **Sheet music from LilyPond code.** *(translated)*

    Generate a clear black-and-white engraved sheet-music image from the following LilyPond code. The output must be a standard score excerpt on a white page, presented as a single treble staff. The clef, key signature, time signature, pickup measure, bar lines, pitches, durations, rests, ties and slurs must match the code exactly. Do not add extra notes, lyrics, fingerings, decorative graphics or unrelated annotations. The engraving should look like printed sheet music, not handwritten music.

    ```
    LilyPond code:  \version "2.24.0" \paper { indent = 0\mm
    line-width = 160\mm tagline = ##f } \layout { ragged-right = ##t
    ```

```
} melody = \relative c'' { \clef treble \key a \minor \time 3/8
\tempo "Poco moto" \partial 8 e16 dis | e16 dis e b d c | a8 r16
c e a | b8 r16 e gis b | c8 r16 e dis e | } \score { \new Staff
\melody }
```

The ground-truth image is rendered from the same code with LilyPond.

12. **Conway's Game of Life, first grid.**

    Generate a clean black-and-white grid image showing the final state of Conway's Game of Life after 4 steps.
    Use standard Conway's Game of Life rules on a finite grid with dead cells outside the boundary.
    Black squares are live cells and white squares are dead cells.
    Initial 10×10 grid:

```
0000000000
0000000000
0001000000
0000100000
0011100000
0000000000
0000000000
0000000000
0000000000
0000000000
```

    Render only the final 10×10 grid after exactly 4 steps.
    Draw each cell as a sharp square.
    Do not add coordinates, labels, text, arrows, or decorative elements.

    The ground-truth image is computed with a Python program.

13. **Conway's Game of Life, second grid.**

    Generate a clean black-and-white grid image showing the final state of Conway's Game of Life after 4 steps.
    Use standard Conway's Game of Life rules on a finite grid with dead cells outside the boundary.
    Black squares are live cells and white squares are dead cells.
    Initial 12×12 grid:

```
000000000000
000001000000
000011100000
000000100000
000110000000
000101100000
000000000000
000001110000
000000010000
000000000000
000000000000
000000000000
```

    Render only the final 12×12 grid after exactly 4 steps.
    Draw each cell as a sharp square.
    Do not add coordinates, labels, text, arrows, borders outside the grid, or decorative elements.

    The ground-truth image is computed with a Python program.

14. **Complex mirror and occlusion relations.** *(translated)*

    A view in front of a bathroom mirror. A bottle of shower gel stands on the counter. Behind the shower gel there is a bunch of keys. The keys are visible only in the mirror, because from the viewer's position they are hidden by the shower gel. The reflection in the mirror shows the complete back of the shower gel bottle. The reflection also contains an analog clock showing 11:07:47, and the picture in the mirror is mirrored.

    Observed: GPT-Image-2 fails; Seedream fails.

15. **Most of the subject is not visible directly, but the whole is visible indirectly.** *(translated)*

    A street scene in realistic photo style. There is a small puddle on the ground in the foreground. On the right side of the frame only the front half of a bicycle can be seen directly: the front wheel, the handlebars and a short piece of the frame. The rear half and the rider's body are outside the edge of the frame or cropped, so the rider and the whole bicycle cannot be seen

directly and completely in the main picture. However, in the reflection in the foreground puddle, the complete bicycle and the person riding it must be visible: both wheels, the whole frame, the saddle, and the rider's upper body and head. The complete bicycle and rider in the reflection must be consistent in position and orientation with the part of the bicycle that is visible in the main picture. The complete rider must not appear directly in the main picture, only in the puddle reflection. The whole image should look like a real street photograph.

Observed: GPT-Image-2 fails; Seedream is correct.

16. **A very large search space with an explicit global constraint.** *(translated)*

    Generate a picture of a maze. Only three paths lead from the start to the end. Mark these three paths in different colors.

    Observed: GPT-Image-2 fails; Seedream fails.

17. **Complex topology.** *(translated)*

    A desk scene in realistic photo style. On the desk there are a black fountain pen and a pair of white wired earphones. The earphone cable is wound once around the pen. On the left side of the pen the cable passes over the pen and hides it; on the right side of the pen the cable goes under the pen and is hidden by it. The cable must be one continuous unbroken white line, and the pen must stay straight and must not be cut by the cable. The crossings must show a clear front-and-back relation and natural shadows.

    Observed: GPT-Image-2 fails; Seedream fails.

18. **A large number that must be shown exactly.** *(translated)*

    A clean top-down photograph. Exactly 37 red M&M chocolate candies are scattered on an ordinary wooden table. Each candy is clearly separated and can be counted individually. There is nothing else on the table.

    Observed: GPT-Image-2 fails; Seedream fails.

We also noted four failures without keeping the prompt text: dice whose pips should sum to a given number (the image summed to 58, which was wrong); a vernier caliper that should read 23.64 (wrong reading); a concave mirror magnifying a coin; and a plate of 20 grapes with a chart of them, where the chart was drawn wrong.

Examples added during construction. While the benchmark was being built, we added five more examples that became the standard worked examples of the protocol. The idea of each example is ours. For three of them we show the final prompt of the task that was built from the idea. The other two were taught to the agent as ideas only: a spoken walking route that must be drawn as a floor plan, and the parts sheet of a layered 2D character that must assemble into the character.

1. **A reversed compass under the noon sun.**

    Draw a clear picture of a hiker's brass novelty pocket compass lying on an open paper map on a wooden table. Show a bright SUN in the sky on the RIGHT side of the picture. It is SOLAR NOON in the northern hemisphere, so the sun marks the direction of geographic SOUTH. IMPORTANT INSTRUMENT RULE: this is a REVERSED novelty compass – unlike a normal magnetic compass (whose red 'N' needle points to geographic NORTH), on THIS compass the red 'N' needle points to geographic SOUTH. Because the noon sun marks south, you must draw the red 'N' needle pointing straight TOWARD the sun, and print the letter 'N' on the rim of the rose on the side nearest the sun (with S opposite it, and E and W on the two sides). Make the sun a large clear icon and the north needle a distinct RED arrow tipped with the letter 'N'. Do not add any other note or caption.

    The model tends to draw the compass that it has seen many times, with the red needle pointing away from the sun.

2. **The team at the bottom of the table celebrates.**

    A photograph taken from the touchline of a small amateur football ground on a bright afternoon, after the last game of the day. Ordinary local club players, nobody famous, no sponsor logos.
    A large painted RESULTS BOARD stands behind the pitch, headed 'SATURDAY LEAGUE - RESULTS', and carries exactly these two result lines, one under the other, in big clear letters: SWIFTS 1 - 1 OTTERS OTTERS 2 - 0 BADGERS
    Three clubs played today and each club's shirts are a plain single colour: the SWIFTS wear YELLOW, the OTTERS wear GREEN, the BADGERS wear PURPLE. Points are the usual three for a win and one for a draw.

> This club has an odd tradition and the picture must show it: the CAPTAIN OF THE SIDE THAT COMES BOTTOM of that table stands on the grass with BOTH ARMS RAISED STRAIGHT ABOVE HIS HEAD, fists up, in an unmistakable victory pose; and the three SUBSTITUTES OF THE SIDE THAT COMES TOP sit on the bench slumped forward with their HEADS HELD IN BOTH HANDS, elbows on knees. Work the table out from the printed results yourself — I am deliberately not telling you which side comes bottom or which comes top, and I am not telling you which shirt colour does what. Do not print the table, the points or any team's position anywhere in the picture: the only text in the picture is the board's heading and the result lines given above.
>
> The results board, the celebrating captain and the three slumped substitutes on the bench must all be clearly visible in the same picture, and every shirt colour must be unmistakable.

The rule is given in full, but which team comes bottom must be worked out from the printed results. The model tends to let the winning team celebrate.

Negative examples. Each prompt below looks hard, but GPT-Image-2 solves it, although other models may fail. For each one we wrote down why it is a poor task. The agent is told not to create tasks of these kinds.

1. **An agent can compute the result and then rewrite the prompt.** *(translated)*

   > Generate an image: three empty cups A, B and C, each with a capacity of 100 ml. Tap A is off (0), tap B runs at 3 ml/s, tap C runs at 10 ml/s. After 8 seconds, draw the correct water level of each cup according to volume = flow rate $\times$ time (capped when the cup is full).

   Observed: GPT-Image-2 is correct; Seedream fails.

2. **An agent can compute the result and then rewrite the prompt (second example).** *(translated)*

   > A photorealistic illustration of a high-school physics problem. Take g = 10 m/s^2 and ignore air resistance. A small block A of mass 2 kg is released from rest at the top of a smooth incline of height 5 m. The bottom of the incline joins a rough horizontal track of length 4 m with a coefficient of kinetic friction $\mu = 0.25$. After passing the rough track, the block makes a perfectly inelastic collision with a small block B of mass 3 kg that rests at the end of the track, and the two stick together. After the collision the two blocks move together onto a smooth horizontal track and finally fly off horizontally from the edge of a table of height 1.25 m. Draw the final state according to the laws of physics.

   Observed: GPT-Image-2 is correct; Seedream fails.

3. **A language model alone can remove the difficulty.** *(translated)*

   > A desk scene in realistic photo style. A white microwave oven stands in the middle of the desk. The camera faces the front of the microwave squarely, at the height of the centre of the microwave door. Only the front of the microwave, a little of the desk, and the plain wall behind can be seen; the back, top, left side and right side of the microwave cannot be seen. A very eye-catching cat-head-shaped sticky note is stuck in the middle of the back of the microwave. The note is bright pink, its edge has the outline of cat ears, the material is matte paper, the top-left corner curls up slightly, and it is held by a short piece of translucent tape. In the bottom-right corner of the note there are a small red heart and two yellow five-pointed stars, and on it, in clear black handwritten Chinese, is written: 'Mom is not home today. Your favourite fried chicken is in the microwave.' The handwriting is neat, and the note is cute and eye-catching.

   Observed: GPT-Image-2 is correct; Seedream fails.

4. **A language model alone can remove the difficulty (second example).** *(translated)*

   > A desk scene in realistic photo style. A thick book lies in the middle of the desk, photographed from directly above and slightly in front. The front cover faces up, and the back cover faces down, flat against the desk. The front cover is pure white, with no text, pattern, label, decoration or stain. However, on the back cover there is a line of huge, striking red Chinese text: 'Secret answer'. The text is printed in the middle of the back cover, in a very bold typeface and bright red, and is very eye-catching.

   Observed: GPT-Image-2 is correct; Seedream fails.

5. **A simple change of topology, which GPT models handle easily.** *(translated)*

   > A curved strip of paper bent into a half circle. On the outer side of the strip the word OUTSIDE is written repeatedly, and on the inner side the word INSIDE is written repeatedly. From the camera angle both the inner and the outer side can be seen, and each word must appear on the correct surface.

Observed: GPT-Image-2 is correct; Seedream fails.

6. **Complex relations among many objects, which GPT-Image-2 is good at.** *(translated)*

   A photorealistic still life that contains exactly eight objects, and all of the following relations must hold at the same time: a red cup is to the left of a blue book; a green ball is above a yellow box; an orange bottle is behind the blue book; a white bowl is in front of the red cup; a purple candle is to the right of the yellow box; a black phone leans against the orange bottle; a pink eraser is inside the white bowl; a brown wooden block is under the green ball. Apart from these eight objects, there must be nothing else.

   Observed: GPT-Image-2 is correct; Seedream fails.

7. **The cause is not visible but its result is visible, which GPT models handle reliably.** *(translated)*

   A laptop stands on a desk, but from the current viewpoint the desk lamp behind the laptop cannot be seen. The lamp behind the laptop is the only light source. The laptop screen is off. There is a cat under the desk.

   Observed: GPT-Image-2 is correct; Seedream fails.

8. **A precise gear ratio, which GPT models handle reliably.** *(translated)*

   Three meshing gears in a row: A (10 teeth), B (20 teeth) and C (40 teeth). Each gear has one red tooth pointing straight up. Gear A turns one full revolution. Show the final position of each red tooth. The positions of the three gears must fully obey the laws of physics. Label A, B and C (or left/right) clearly. The image must be photorealistic.

   Observed: GPT-Image-2 is correct; Seedream fails.

9. **A counter-intuitive request that goes against physics, such as reversing the order of objects in a mirror.** *(translated)*

   A view facing a bathroom mirror. On the counter, from left to right: a red toothbrush, a blue cup, a bottle of yellow soap, and a white box labelled 'Soap'. In the mirror the left-to-right order of these four objects is reversed, and the text on the box is mirrored too.

   Observed: GPT-Image-2 is correct; Seedream is correct.

From these trials we drew two lessons for the agent. First, a task is weak if its final picture can be described in words, because an agent can compute the result and rewrite the prompt into a simple description. Second, a task is weak if a language model can simplify the prompt, for example by dropping an object that the prompt asks for but that must not be visible.

Abridged instructions of the agents. Each role in the protocol is played by a separate agent that works from a written instruction. The lists below are abridged translations of these instructions.

Instruction to the brainstorming agent *(abridged, translated from the original)*

1. Start from a real scene, such as a work station, a home, a street, a sport, a game or a document. Do not look for a scene to dress up a mechanism you already like.
2. Inside the scene, find a hard constraint that a person who does this work would notice at a glance if it were drawn wrong. Only then ask whether it is difficult.
3. Say in one sentence who would need this picture and what they would do with it. This field is required.
4. Ask whether this picture existed in the world before we set the task. If not, the mechanism may still be good, but the carrier is a teaching prop: attach a real-life carrier and change nothing else.
5. A task must satisfy four things at once: it is a real commission; a layman can tell right from wrong in one look; the answer appears nowhere in the prompt; and judging it leaves behind a picture worth showing.
6. Tell your candidate in one sentence: what the viewer sees, where the mistake shows, and why this picture really exists. If you cannot, drop it.
7. Take the difficulty from what actually breaks models: withhold the key number; make a quantity travel from one place of the picture to another; make each cell depend on other cells; ask for a composite transform to be inverted; leave one step outside the rule table; give no scale at all.
8. Avoid what never breaks models: writing out the whole rule table for the model to follow; describing the final appearance; asking whether something appears anywhere in the image; exact counting on a regular layout.

9. Difficulty must come from the process, not the dose. More objects, more rows, more decimal places and more arithmetic are a ladder, not a new task.
10. To make a task harder, use a richer real specification or a real convention of the trade, or move the check to after the thing has been assembled. The amount of data stays the same.
11. One conversion, one addition or one comparison is allowed on any axis. Multi-step arithmetic belongs to the latent computation axis and may not be disguised.
12. Test the candidate three times. Hand the withheld number to the model for free: if the task becomes easy, it tests arithmetic. Reduce the count to five: if it becomes easy, it tests counting. Make the unusual rule normal again: if it is still hard, the difficulty is not where you said it was.
13. One task per mechanism. Two candidates are the same task if their failure conditions are the same sentence, whatever the objects are. Check for collisions across all nine capability axes, not only your own.
14. Do not clone the worked examples. Learn the formula, not the picture.
15. Before a candidate is released, name the nearest task that has already been scored, give its score, and say which specific quantity this candidate keeps outside the prompt. 'It is harder' is not an answer.
16. Declare anything you truncated, sampled or relaxed. You do not approve your own candidate: a reviewer does.

Instruction to the task-writing agent *(abridged, translated from the original)*

1. Write the failure condition and the evaluation program before you write the prompt.
2. The prompt is a work order, not a construction drawing. It has four parts: the scene and the deliverable; the material that fixes the inputs; the purpose; and the constraints, including what must not appear. Never teach the model how to draw, and never state the answer.
3. Fix the input, not the output. Whatever the model should have before it starts may be given exactly. Whatever it should only produce after working may not appear.
4. List every element of the expected answer and confirm that at least one cannot be found word for word in the prompt.
5. A full rule table may be given, but after the table there must still be one step of computation, placement or conservation that cannot be looked up in it.
6. Do not lay the picture out element by element with a string of imperatives. That turns the model into a typesetter and measures nothing. Do not write down the final appearance.
7. The expected answer must promise something, and it must contain numbers.
8. A real industrial setting is not enough by itself. Its value is the part of the real specification that cannot fit in the prompt. Say which part that is.
9. For a task about orientation or visibility, fix the orientation with a natural anchor in the scene, such as a person reading the face of the object. Never state the camera-side conclusion, and never stay silent about orientation.
10. Rotate your own reference image by thirty degrees and score it again. If the score flips, the orientation was not fixed hard enough. State the sign convention wherever one shape could be read as two different signs.
11. If the VLM could answer the question from its own knowledge without looking at the image, the candidate is rejected. No rewording rescues it.
12. Answer two questions in writing. How could a model that knows nothing guess a score? How could a strong model get around the ability itself? Close both routes.
13. Require a result that cannot be faked by drawing, such as specific numbers or a correctly placed object. Do not ask whether something appears anywhere in the image.
14. No exact counting on a structured layout such as a table or a calendar. Counting tasks must scatter their objects.
15. Across the difficulty levels of one task, change at least two of carrier, setting, composition and style. Changing only the numbers is not enough. Re-skinning counts only if it changes the mechanism: counting apples and counting pears is one task, not two.
16. The picture that the prompt describes must have existed in the world before the task was set.

Instruction to the evaluation-program agent *(abridged, translated from the original)*

1. Perception belongs to the VLM and arithmetic belongs to code. The VLM never does arithmetic. Where the answer needs a computation, the VLM only reports what it sees in a machine-readable form, and a program computes the conclusion.
2. Every step must leave an overlay on the image that marks the element this step looked at.
3. Ask the VLM one observable fact at a time. Every 'and' belongs in code. Split a compound question into separate questions and combine them in code.
4. Every question must be answerable in seconds by a person with no background who is shown the crop. Move all domain knowledge into code. Only two kinds of question may need expertise: transcribing a notation, and marking geometry.
5. One question, one fact. No comparisons that are hard to follow, no double negatives, and no examples, because an example becomes the VLM's target list. Fix the answer format.
6. Keep the phrase that names what to locate short. A long phrase breaks the check itself, and the symptom is a refusal to read, not a wrong number.
7. A question may never contain a correctness word, state an expectation, or reveal the shape of the expected answer. Never give the expected count to the step that proposes the boxes. The count is the answer.
8. Do not add a scene-level easy question by default. Only a real precondition earns one, because a scene check tends to reject the most correctly drawn images of an unusual task.
9. If the expected answer fixes the geometry, draw it over the image and ask whether it coincides. If the criterion is a geometric relation, draw the construction so that the conclusion is visible on the image.
10. A count that decides the score must go through the loop: locate, confirm, read, compute. The scored count is the number of items confirmed one by one, never the number of boxes proposed. Counts of at most ten are exempt.
11. Use only the allowed operators. A new operator needs a task-neutral design and at least three tasks that will reuse it.
12. Complexity is a defect. A task that no allowed operator can evaluate is better removed than kept.
13. There are three outcomes, never two: a scored result; a zero caused by the model's own broken image; and a failure of the evaluator. A failure of the evaluator, or a refusal by the VLM, withholds the task and is never scored as zero.
14. Every zero must name the fact that was violated. That fact must be drawn on the image, and it must have been stated in plain words in the prompt.
15. Never change a threshold because of a score you have seen. The only valid reasons are the measured noise of the evaluator, consistency with similar tasks, and the arithmetic of the task itself.

System prompt of the coding models. GPT-6 Astra and GPT-5.6 Sol receive the system prompt below and the task prompt as the user message. {W} and {H} are replaced by the width and the height of the image in pixels.

> You draw pictures by writing code.
>
> The user message is a picture description. Write one self-contained HTML document that draws that picture, and return nothing else.
>
> **Looking Things Up.**
>
> - You have a web search tool. Use it whenever the description depends on a fact you would otherwise be guessing at: a live or dated figure, the real appearance, markings, proportions or specification of an actual object, place, instrument or document. Search first, then draw what you found.
> - Search is yours to judge. A picture that needs no outside fact needs no search.
> - What you bring back from a search must end up in the code you write. The rendered page itself has no network: it cannot load a photograph, a font, a map tile or a data file from anywhere, so a fact is only useful once you have turned it into drawing instructions. Pasting a picture from the web would be collecting, not drawing, and the page cannot do it anyway.
>
> **Canvas.**
>
> - The drawing surface is exactly {W} pixels wide and {H} pixels high. Fill it. Do not leave a border of blank page around the picture and do not draw outside it.
> - Use either a <canvas> element of exactly {W}x{H} with an inline <script> that draws on its 2D context, or one inline <svg> element of exactly {W}x{H}. Either is fine. Mixing them is fine.
> - Give the page a background. A transparent or default-white page counts as an undrawn picture.

Self-Contained, No Network.

- Everything is inline in the one document. No <img src>, no <link>, no external <script src>, no fetch, no XMLHttpRequest, no WebSocket, no import from a URL, no @font-face pointing at a file, no web font service. Nothing is downloaded: the page is rendered with the network off, so anything external simply will not arrive.
- Text must use fonts already on a normal machine. Name generic families such as sans-serif, serif, monospace, or common system faces with a generic fallback.
- No alert, confirm or prompt. No window.open. No timers that keep changing the picture: draw the finished picture, do not animate it.

When The Picture Is Finished.

- Draw on load, synchronously if you can.
- The moment the drawing is complete, set window.__done = true. A screenshot is taken as soon as that flag appears, so setting it early gives you a half-drawn picture and never setting it costs you nothing but a wait.
- The document must not throw. An uncaught exception is a blank page and scores nothing.

What You Are Being Asked For.

- Draw everything the description asks for, with the counts, the positions, the text, the relations and the values it names. Every quantity in the description is checkable in the picture afterwards, so compute it in code rather than eyeballing it: lay out with arithmetic, place things by coordinates you work out, and write the labels the description dictates, spelled exactly as it spells them.
- You get one attempt and you never see the rendered result, so prefer constructions you can be sure of over constructions that need tuning by eye.
- Make it look like the thing described, not like a diagram of the thing, unless a diagram is what was described.

Output Format.

- Reply with exactly one fenced code block, opened with three backticks and the word html, and nothing before or after it. No explanation, no notes, no second block.