DRAWING ANSWERS: BENCHMARKING VISUAL REASONING IN IMAGE GENERATIVE MODELS

Anonymous authors
Paper under double-blind review

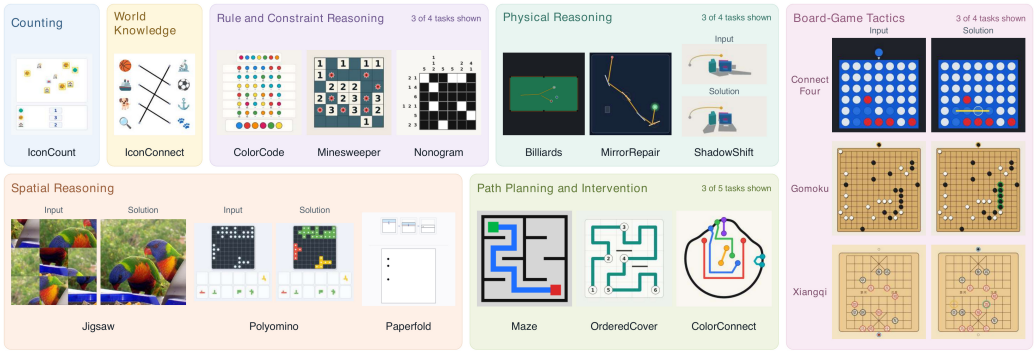

Figure 1: **Drawing Answers at a glance.** The benchmark spans seven families and 22 visual reasoning tasks; 17 representative tasks are shown. Panels display reference solutions, with input–reference pairs where the transformation benefits from comparison. IconCount, Nonogram, Minesweeper, and OrderedCover also include aligned non-digit encodings for representation ablations (Figure 5).

ABSTRACT

Can image generative models solve a problem by drawing its answer? Tracing a route, completing a grid, and selecting a winning move require images that satisfy explicit rules and spatial constraints. We introduce **Drawing Answers**, a dataset and benchmark spanning **22 tasks in seven reasoning families**. It contains **152,366 problem–answer image pairs across 26 configurations**, including four paired representations of the same underlying problems. Task-specific generators, solvers, and structured annotations support supervised learning and evaluation of generated solutions. We evaluate unadapted FLUX.1 Kontext and GPT Image 2, and train task-specific FLUX LoRA adapters, evaluating six training checkpoints and reporting the highest test outcome per configuration. Gomoku peaks at 30,000 steps, where success rises from 0.0% for unadapted FLUX to 92.0%, compared with 10.0% for GPT Image 2. Performance varies substantially across tasks: accurate local edits coexist with persistent difficulty in globally constrained solutions. Paired encoding experiments further reveal different representation preferences across models. Drawing Answers provides a controlled resource for studying the capabilities, learnability, and remaining challenges of visual reasoning through image generation.

1 INTRODUCTION

Many reasoning problems admit a visual answer. A route can be drawn through a maze, a collection of pieces can be assembled into a shape, and a winning move can be placed directly on a game board. These problems require a model to interpret the visual state, apply the relevant rules, and express its decision in the same spatial environment. They motivate a direct question: can image generative models reason visually by drawing their answers?

The distinction between a plausible image and a valid answer is consequential. A smooth path can cross a wall; a well-rendered board can contain the wrong move; and an almost complete packing

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.

can leave one cell uncovered. Visual fidelity alone does not resolve these cases. Evaluating such outputs requires both a clear task specification and a way to inspect the solution represented by the generated image.

Recent work studies reasoning in visual output spaces, including image-to-image prediction on ARC (Hu et al., 2026), direct editing for visual planning (Zhou et al., 2026), and reasoning-oriented image generation with structured hints (Tu et al., 2026). These directions create an opportunity for a complementary benchmark that combines task diversity, paired supervision, and rule-based evaluation. Training examples are particularly useful here: they allow a benchmark to characterize both existing model capabilities and the extent to which visual task-solving behavior can be learned.

We introduce **Drawing Answers**, a collection of 22 tasks with a common interface: a problem image and an instruction are mapped to an answer image. The tasks cover counting, world knowledge, rule and constraint reasoning, physical reasoning, spatial reasoning, path planning and intervention, and board-game tactics (Figure 1). Answers include marks, paths, rearrangements, completed grids, trajectories, and edited physical configurations. The benchmark therefore spans both small local decisions and problems in which many interacting constraints must be satisfied together.

The dataset contains 147,266 training pairs and 5,100 held-out pairs across 26 configurations. Four tasks additionally provide aligned visual representations: IconCount varies count bars and digits; Nonogram varies unary bars and numerical clues; Minesweeper varies dot and numerical clues; and OrderedCover varies segmented rings and numbered checkpoints. The underlying problems and split assignments are preserved within each pair, making visual representation a concrete experimental variable.

We evaluate unadapted FLUX.1 Kontext (Black Forest Labs et al., 2025), GPT Image 2, and task-specific LoRA adaptations (Hu et al., 2022). The peak test outcomes across six evaluated checkpoints show strong performance on several routing and board tasks, including 100.0% valid taut paths on TautReach at 10k steps and 92.0% strict success on Gomoku at 30k steps. Other tasks expose substantial gaps: OrderedCover can achieve high cell coverage while rarely producing a complete legal ordered path. These differences give the benchmark useful range, from successful learned behaviors to demanding targets for further progress.

Our contributions are threefold:

1. **A diverse benchmark with visual answers.** We unify 22 tasks and 26 configurations under an image-based problem-solving interface, with 152,366 paired examples and structured annotations.
2. **Rule-defined supervision and evaluation.** Task-specific construction procedures connect visual problems to reference solutions, while evaluators measure answer content, constraints, and required scene preservation.
3. **A study of visual task learning.** Unadapted and adapted FLUX results, GPT Image 2 comparisons, six training checkpoints, and four representation studies reveal differences in capability, learnability, and answer encoding.

2 Related Work

Structured visual reasoning. Controlled visual tasks make it possible to associate model predictions with known scene structure and reasoning requirements. CLEVR uses rendered scenes and functional question programs to diagnose visual question answering (Johnson et al., 2017); RAVEN studies abstract reasoning through structured visual matrices (Zhang et al., 2019); and Bongard-LOGO evaluates concept learning from visual examples (Nie et al., 2020). ARC asks solvers to construct output grids by abstracting a rule from a small number of demonstrations (Chollet, 2019). Drawing Answers shares the use of explicit problem structure and focuses on image generative models that express solutions through spatially precise visual outputs.

Visual reasoning through image generation. VARC formulates ARC as image-to-image prediction with a vision transformer and test-time training (Hu et al., 2026). EAR introduces an editing-as-reasoning formulation and the AMAZE benchmark, combining maze solving and queen placement with visual and logical evaluation (Zhou et al., 2026). Hint2Gen uses code-structured visual hints

for image editing and introduces the Reason2Gen benchmark (Tu et al., 2026). Drawing Answers contributes a collection of rule-defined tasks with paired training data, structured references, and aligned encoding variants. Its experiments connect task-level correctness with supervised adaptation and the representation used to express an answer.

Interactive visual reasoning. VISTA studies agents that learn to interact with unfamiliar visual environments and discover their mechanics and goals (Han et al., 2026). Drawing Answers uses static problems with specified instructions and evaluates the answer expressed in one generated image. The two settings expose complementary demands: exploratory interaction in one case and the construction of a complete visual solution in the other.

Adapting image editing models. FLUX.1 Kontext uses flow matching for image-conditioned generation and editing in latent space (Black Forest Labs et al., 2025). LoRA adapts pretrained weights with trainable low-rank updates (Hu et al., 2022). We use these established components as an adaptation baseline, retaining the original editing architecture. This keeps the empirical focus on the benchmark's supervision, task requirements, and representation choices.

3 Drawing Answers Dataset and Benchmark

3.1 Task Overview

For task t, an example contains a problem image x, instruction c_t, reference answer image y, and structured annotation z. A conditional model generates

$$\hat{y} = G_\theta(x, c_t; \epsilon), \tag{1}$$

where ϵ denotes sampling randomness. The instruction specifies the operation and answer format; the problem image specifies the instance. The answer and solution-bearing annotations are used for supervision or evaluation.

The benchmark spans seven families, with 1 counting, 1 world-knowledge, 4 rule-and-constraint, 4 physical, 3 spatial, 5 path-planning-and-intervention, and 4 board-game tasks. Figure 2 shows all 22 tasks, and Appendix B gives their objectives and split sizes. Some tasks require a unique answer, such as an immediate winning move or a solved clue grid. Others permit alternatives: ColorConnect accepts any legal routing, while TautReach checks a valid taut route without requiring identity with a globally shortest reference. Evaluation follows the task contract rather than assuming every image has only one acceptable solution.

3.2 Data Construction

Rules, solutions, and rendering. Each task specifies its state, allowed operations, answer requirements, and editable content. Construction starts from a symbolic state, geometric scene, or source asset. A task-specific solver, simulator, constructive procedure, or curated semantic annotation supplies the solution; the problem and answer are rendered from that state. Board tasks retain legal moves and board states, grid puzzles retain solution cells or graph structure, and physical tasks retain trajectories, events, or scene geometry. Most environments are procedurally rendered. Jigsaw uses ImageNet photographs (Deng et al., 2009); IconCount and IconConnect use icon and emoji assets with retained source provenance.

Problem selection and splits. Generators select solvable states with the required legal, geometric, and uniqueness properties. Train/test partitions follow each task's construction unit: source photographs for Jigsaw, layout/symmetry groups for Pinball, position lineages for Xiangqi, tactic signatures for Gomoku, and solution/clue families for Nonogram. Appendix B.1 describes the construction procedures and reference answers. Paired encodings retain the original split assignments.

Scale and complexity. The 26 configurations contain 152,366 rendered pairs: 147,266 training and 5,100 held-out pairs. Most configurations contain 6,000 training and 200 test examples; Appendix B lists the smaller TicTacToe and IconConnect sets and the evaluation subsets. Counting each representation pair once gives 127,566 underlying problems. Task metadata record native difficulty

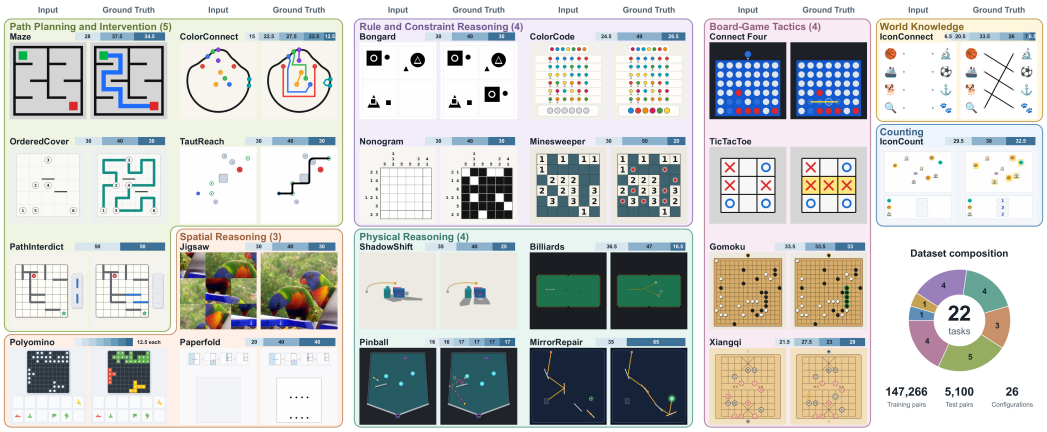

Figure 2: **The complete Drawing Answers task atlas.** All 22 tasks are shown as input–reference pairs, using digit variants where available. Blue bars show test proportions by native difficulty or complexity, from light to dark (200 examples per bar); the ring counts tasks by family. Appendix G gives the levels and percentages.

or structural axes, such as folds, grid size, mirror count, and barrier budget. Figure 2 summarizes 20 such distributions; these are task-local strata rather than a shared cross-task difficulty scale.

Paired visual encodings. Four tasks preserve problem identity and solution while varying representation (Figure 5). IconCount changes count bars to digits, including the answer-area template. Nonogram and Minesweeper change clue encodings, and OrderedCover changes checkpoint-order encodings. These pairs support a direct comparison of representation settings on aligned problems. The structured package retains original prompts, splits, annotations, and output identifiers so that predictions can be joined to the exact questions they answer.

3.3 Evaluation Protocol

Let $\mathcal{A}_t(x, c_t)$ denote the acceptable solutions. Evaluation asks whether $\hat{y}$ expresses a member of this set and meets the relevant visual constraints. Evaluators extract board occupancy, filled cells, line endpoints, paths, piece placements, orientations, trajectories, or shadow masks and check them against structured annotations. For a strict editing task, success takes the form

$$s_i = \mathbf{1}\{V_t(\hat{y}_i, z_i) \wedge P_t(\hat{y}_i, x_i, z_i)\}, \qquad S_t = N_t^{-1} \sum_{i=1}^{N_t} s_i, \tag{2}$$

where V_t tests the answer and P_t tests required editing and preservation constraints. Tasks with logical-board, geometry-only, or continuous outcomes retain those definitions; Appendix D specifies the gates. We report task-level results rather than averaging unlike quantities.

Evaluators are calibrated with reference answers, unchanged inputs, and controlled corruptions. We report binary outcomes for 17 tasks, continuous trajectory and shadow scores for Pinball and ShadowShift, and task-specific diagnostics for the remaining three tasks (Appendix F). Each test question receives one generated answer, conditioned on its problem image and instruction.

3.4 Learning to Draw Answers

To investigate whether visual reasoning performance can be improved through supervised adaptation, we fine-tune FLUX.1 Kontext using low-rank adaptation (LoRA) on Drawing Answers (Figure 3).

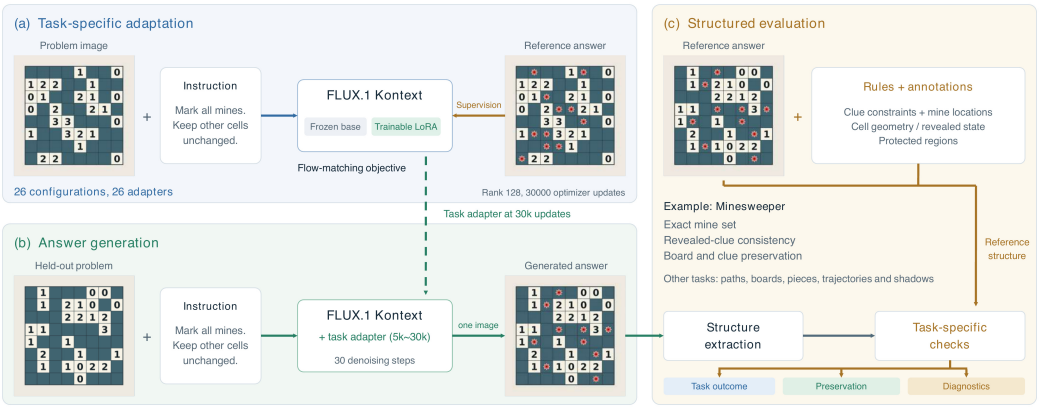

Figure 3: **Task-specific adaptation, inference, and evaluation.** Paired training images supervise rank-128 LoRA adapters for a frozen FLUX.1 Kontext backbone. At inference, a held-out problem and instruction produce one answer image. Extracted answer structure is checked against task rules and annotations. The illustrated example is digit-clue Minesweeper; references enter training or evaluation, not test-time model conditioning.

We train one LoRA adapter per configuration, giving 26 adaptations of FLUX.1-Kontext-dev. For a selected weight matrix, $W = W_0 + (\alpha/r)BA$, where W_0 is frozen and A, B are trainable. The problem image and instruction condition generation, while the reference answer supplies the target. Let $z_y = E(y)$ and $\varepsilon \sim \mathcal{N}(0, I)$. With scheduler noise level σ_τ, $z_\tau = (1 - \sigma_\tau)z_y + \sigma_\tau\varepsilon$. Training minimizes

$$\mathcal{L} = \mathbb{E}\Big[w(\tau)\,\|v_{\theta+\Delta}(z_\tau, \tau, E(x), c_t) - (\varepsilon - z_y)\|_2^2\Big]. \tag{3}$$

Only the low-rank updates are optimized. Task-specific evaluators are used after generation, not as a training loss. At test time the selected adapter and frozen backbone generate the answer without an additional symbolic solver.

4 Experiments

We study existing model capabilities, the gains from supervised adaptation, and the effect of visual representation.

4.1 Experimental Setup

We train one rank-128 LoRA adapter per configuration for 30,000 optimizer updates, using AdamW with a nominal learning rate of 10^{-4}, batch size one, and one NVIDIA A100 80GB GPU. Models receive the problem image and instruction and generate one answer per test question. The base FLUX model and its adapted versions share inputs, prompts, seeds, and 30-step sampling settings. GPT Image 2 uses medium quality and one 1024×1024 output, without benchmark-specific training. Appendix C gives the full settings.

We evaluate LoRA at 5k, 10k, 15k, 20k, 25k, and 30k updates. *LoRA Peak* is the highest test outcome across these checkpoints, selected separately for each configuration, with ties resolved by the earliest step. Tables 1 and 2, the qualitative comparisons, and the component analysis in Section 4.4 use these same per-configuration peak checkpoints. Appendix E reports every checkpoint. All systems use the same task-specific criteria and complete test sets.

4.2 Task-Level Results

Table 1 reports task-level outcomes; Figure 4 compares model answers to the same problems.

Table 1: **Task-level outcomes (%).** LoRA Peak is the highest test outcome across six checkpoints (5k–30k), with the earliest tied step. The 21 configurations cover 17 tasks; criteria are defined in Appendix D. Each row uses 200 tests except TicTacToe (100). Bold marks the highest outcome.

Task / configuration	Criterion	Base	LoRA Peak	Peak step	GPT
IconCount (bars)	Counting + highlighting	2.0	**35.5**	30k	21.0
IconCount (digits)	Counting + highlighting	0.0	11.0	25k	**28.0**
IconConnect	Strict semantic matching	3.0	**6.0**	15k	1.5
ColorCode	Strict edit success	0.0	**51.5**	25k	0.0
Nonogram (bars)	Strict edit success	0.0	**18.5**	25k	0.0
Nonogram (digits)	Strict edit success	0.0	**10.5**	30k	0.0
Minesweeper (dots)	Strict edit success	0.0	**31.5**	30k	0.0
Minesweeper (digits)	Strict edit success	0.0	**24.5**	30k	0.0
MirrorRepair	Strict semantic repair	0.0	**23.5**	30k	0.0
Polyomino	Strict semantic success	0.0	**25.5**	30k	0.0
Paperfold	Exact hole geometry	0.0	**15.0**	30k	0.0
Maze	Route topology (STSR)	0.0	**91.5**	25k	0.5
ColorConnect	Legal routing	0.0	**87.0**	30k	0.0
OrderedCover (rings)	Strict ordered path	0.0	**1.0**	30k	0.0
OrderedCover (digits)	Strict ordered path	0.0	**0.5**	5k	0.0
TautReach	Valid taut path	0.0	**100.0**	10k	0.0
PathInterdict	Strict optimal intervention	0.0	**4.0**	25k	0.0
Tic-Tac-Toe	Board + winning cells	0.0	**95.0**	15k	27.0
Connect Four	Board + winning line	0.0	**87.0**	30k	7.0
Gomoku	Strict task success	0.0	**92.0**	30k	10.0
Xiangqi Mate-in-One	Exact resulting board	0.0	**12.0**	30k	0.0

Performance across seven task families. The highest task success rates occur in *path planning and intervention* and *board-game tactics*: TautReach reaches 100.0%, Maze 91.5%, and ColorConnect 87.0%; TicTacToe reaches 95.0%, Gomoku 92.0%, and Connect Four 87.0%. Base FLUX scores 0.0% on all six. Within these families, OrderedCover (1.0%), PathInterdict (4.0%), and Xiangqi Mate-in-One (12.0%) remain harder: successful routing and immediate winning moves coexist with difficulty in ordered coverage, optimal intervention, and legal checkmate.

Among the other families, *rule and constraint reasoning* reaches 51.5% on ColorCode, 31.5% on dot-clue Minesweeper, and 18.5% on unary-clue Nonogram. *Counting* reaches 35.5% with bars; *spatial reasoning*, 25.5% on Polyomino and 15.0% on Paperfold; and *physical reasoning*, 23.5% on MirrorRepair, alongside the continuous gains below. *World-knowledge* matching remains difficult: IconConnect reaches 6.0%, versus 3.0% for base FLUX. These are individual task outcomes within each family.

Progress across training. Gomoku improves from 49.5% at 5k to 92.0% at 30k, Connect Four from 33.0% to 87.0%, and MirrorRepair from 1.0% to 23.5%. Other tasks peak earlier: TicTacToe reaches 95.0% at 15k and finishes at 88.0%, while ColorCode reaches 51.5% at 25k and finishes at 51.0%. The full trajectories in Appendix E show these different learning patterns.

Continuous geometric outcomes. Trajectories and shadows admit graded geometric errors: a near-correct route may deviate at a collision, or a correctly shaped shadow may be displaced. We therefore report continuous geometric scores separately from Table 1's complete-answer success rates. Pinball's soft trajectory score measures alignment across motion segments and handles portal discontinuities. ShadowShift measures intersection-over-union (IoU) between predicted and reference ground shadows, excluding objects and overlays.

At their 30k peaks, FLUX achieves a Pinball score of 0.6717 (base FLUX: 0.0024; GPT Image 2: 0.1098) and a ShadowShift IoU of 0.7692 (base: 0.1651; GPT: 0.1949). Appendix F defines the metrics and reports additional diagnostics for Billiards, Jigsaw, and Bongard.

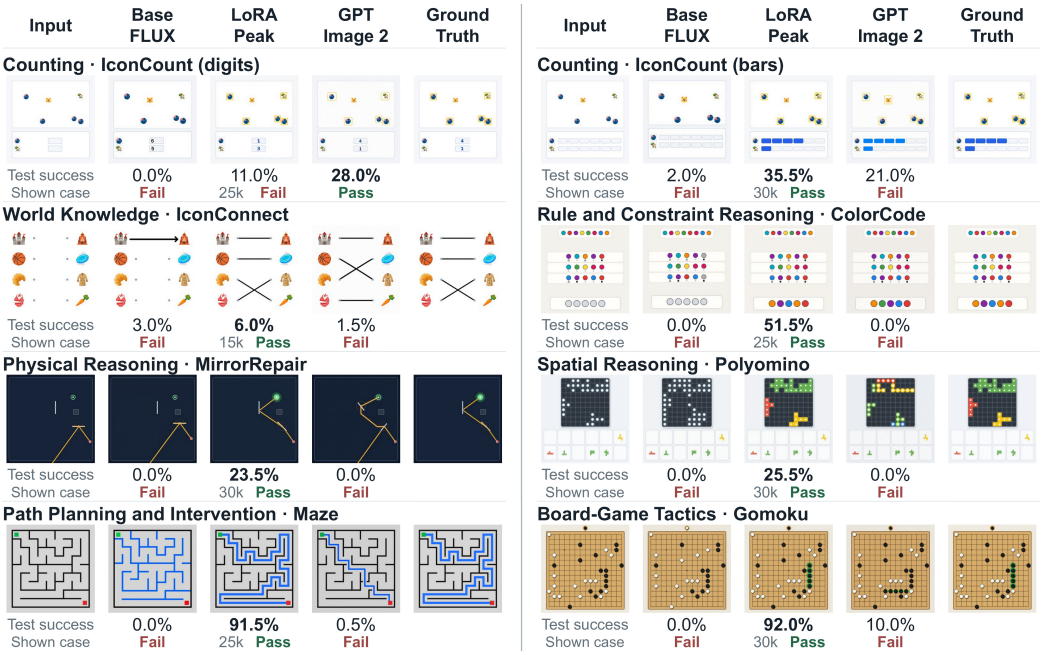

Figure 4: **Same-problem model comparisons.** Eight configurations span seven task families. "Test success" summarizes 200 examples; "Shown case" indicates the displayed answer's correctness.

4.3 Visual Representation

The digit and bar panels in Figure 4 show answers to the same IconCount problem. Figure 5 illustrates four aligned task pairs that vary the encoding of counts, clues, or checkpoint order. Appendix A provides two model comparisons per configuration.

Table 2: **Paired outcomes (%).** Each encoding uses its own LoRA Peak from Table 1.

Encoding	Base	LoRA Peak	GPT
IconCount			
Bars	2.0	35.5	21.0
Digits	0.0	11.0	28.0
Nonogram			
Bars	0.0	18.5	0.0
Digits	0.0	10.5	0.0
Minesweeper			
Dots	0.0	31.5	0.0
Digits	0.0	24.5	0.0
OrderedCover			
Rings	0.0	1.0	0.0
Digits	0.0	0.5	0.0

FLUX favors bars in IconCount and Nonogram, and dots in Minesweeper, over digits by 24.5, 8.0, and 7.0 percentage points at their respective peaks. GPT Image 2 reverses the IconCount preference: 21.0% with bars versus 28.0% with digits. Both OrderedCover encodings remain difficult.

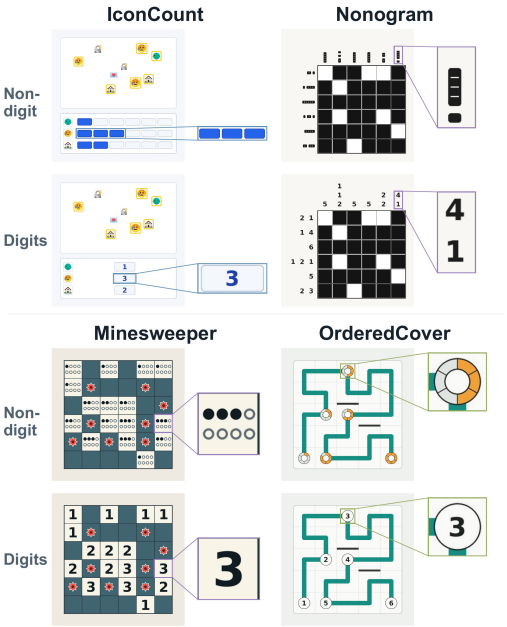

Figure 5: **Paired visual representations.** Reference answers with enlarged details.

4.4 DISCUSSION

We examine what adaptation teaches the model and which answer components still limit performance. Each FLUX analysis uses the configuration's LoRA Peak checkpoint from Table 1.

Supervised adaptation strengthens visual problem solving. The routing and board-game results show that an image editing model can learn to produce answers satisfying explicit reasoning constraints. Adapted FLUX places winning moves, connects endpoints through legal routes, and respects geometric support conditions on held-out problems. These outcomes require the generated edit to depend on the configuration of the input. Their improvement over base FLUX demonstrates that paired visual supervision can develop task-specific reasoning performance within the image generation interface.

The variation within families also identifies what remains difficult. Maze and ColorConnect improve strongly, whereas OrderedCover must coordinate coverage, connectivity, and checkpoint order simultaneously. Immediate winning moves in TicTacToe, Connect Four, and Gomoku are learned more successfully than Xiangqi checkmate. The distinction is therefore finer than whether a task involves a board or a path: the set of relations that must hold together shapes the observed capability.

Readable digits can express incorrect quantities. Digits specify input constraints in Nonogram and Minesweeper and output quantities in IconCount. FLUX performs worse under digit encodings in all three tasks (Table 2). At the digit adapter's 25k peak, FLUX correctly highlights the requested objects in all 200 IconCount test problems, and 401 of 406 answer rows (98.77%) contain a decodable digit. Only 83 rows (20.44%) give the correct value, however, and just 22 problems (11.0%) have every count correct. Selecting the relevant objects and rendering readable symbols are substantially easier here than assigning the correct numerical value. For one aligned problem with counts $[3, 6]$, the digit adapter marks all nine target objects but writes $[6, 6]$; the bar adapter answers correctly. Appendix H describes this and the following examples.

A complete layout can violate the task's rules. At the ring-based OrderedCover peak, 193 of 200 predictions cover every cell, but only 43.0% pass the connectivity check, 13.5% pass the path-degree check, and 1.0% satisfy the complete task. One prediction follows a connected path through all 25 cells without crossing a wall, yet visits checkpoints in the order $1 \to 3 \to 2 \to 4$. The path occupies the right space while violating a required relation.

This gap also appears in other task families. At its 15k peak, IconConnect produces complete one-to-one matchings in 85.0% of problems, but only 7.5% match the semantic reference and 6.0% pass the strict criterion. Dot-clue Minesweeper achieves 96.51% cell accuracy but 31.5% whole-board success. Visual completeness and local correctness therefore measure different aspects of performance from satisfying all relations in an answer.

Representation preferences depend on the model and answer component. IconCount reveals opposite encoding preferences: adapted FLUX is stronger with bars, while GPT Image 2 achieves higher strict success with digits. The component measurements explain this contrast. GPT gets every digit count correct in 79.5% of problems, fully correct highlights in 34.0%, and strict success in 28.0%. FLUX has correct highlights throughout the test set but only 11.0% counting accuracy. For example, GPT writes the correct counts $[5, 6]$ for envelopes and shoes while marking only the shoes; the FLUX example above instead has correct marks and an incorrect numerical answer.

Even within the bar encoding, the model ranking depends on the component: GPT has higher counting accuracy than FLUX (46.5% versus 35.5%), while FLUX has higher strict success (35.5% versus 21.0%). An encoding can make quantity prediction easier for one system while the required visual edits remain its bottleneck. The aligned variants expose these preferences without changing the underlying problems. Together with component-level evaluation, they distinguish gains in numerical interpretation, instance selection, and rule satisfaction that a single image-quality measure would conflate.

5 CONCLUSION

Drawing Answers brings 22 visual reasoning tasks into a common image-based interface, with paired supervision, held-out examples, and structured annotations. Comparisons of unadapted FLUX, GPT Image 2, and task-specific FLUX LoRA models show strong learned performance on several routing and board tasks alongside persistent challenges in global constraints and semantic correspondence. Four aligned encoding studies further demonstrate that answer representations interact with model behavior. The benchmark provides a concrete foundation for developing image models whose outputs satisfy both the visual and logical requirements of a problem.

AI USE STATEMENT

In this work, we used generative AI tools to assist with the implementation and debugging of parts of the code used to construct our benchmark dataset. The benchmark instances themselves were generated programmatically according to the task definitions and data-generation procedures described in the paper, rather than being directly generated by generative AI models. Additionally, we used generative AI tools to improve the readability of the manuscript, search for and identify relevant literature, and assist with LaTeX formatting.

All AI-assisted code was manually reviewed, executed, and tested by the authors before being incorporated into the benchmark construction pipeline. The generated benchmark instances were subsequently checked for validity and consistency using the verification procedures described in the paper. References identified with AI assistance were verified against the original publications, and all AI-assisted textual and formatting edits were reviewed by the authors. We take responsibility for the final content of this work, including text, code, data, claims, and other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Drawing Answers is intended for research on visual problem solving in controlled environments. Its tasks combine procedural puzzles and scenes with existing visual assets. We retain source provenance for the icon and emoji assets used in IconCount and IconConnect and cite the ImageNet photographs used in Jigsaw (Deng et al., 2009).

REPRODUCIBILITY STATEMENT

We provide our code, experiment configurations, evaluation scripts, and per-sample results in the supplementary material to support reproduction of the reported experiments and analyses. The task catalog and construction procedures are documented in Appendices B and B.1. Appendix C specifies training and inference settings, Appendix D defines the evaluation criteria, and Appendix E reports all six checkpoints. The accompanying prompts, seeds, and output identifiers link individual results to their test cases and experimental settings.

REFERENCES

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. FLUX.1 Kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025. URL https://arxiv.org/abs/2506.15742.

François Chollet. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019. URL https://arxiv.org/abs/1911.01547.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. URL https://doi.org/10.1109/CVPR.2009.5206848.

Qiushi Han, Keya Hu, Linlu Qiu, Cathy Wu, and Kaiming He. VISTA: A visual harness for reasoning in an interactive world, August 2026. URL https://vista-research.github.io/.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Keya Hu, Ali Cy, Linlu Qiu, Xiaoman Delores Ding, Runqian Wang, Yeyin Eva Zhu, Jacob Andreas, and Kaiming He. ARC is a vision problem! In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2537–2546, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Hu_ARC_Is_a_Vision_Problem_CVPR_2026_paper.html.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2901–2910, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Johnson_CLEVR_A_Diagnostic_CVPR_2017_paper.html.

Weili Nie, Zhiding Yu, Lei Mao, Ankit B. Patel, Yuke Zhu, and Anima Anandkumar. Bongard-LOGO: A new benchmark for human-level concept learning and reasoning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 16468–16480, 2020. URL https://papers.nips.cc/paper/2020/hash/bf15e9bbff22c7719020f9df4badc20a-Abstract.html.

Yuanpeng Tu, Yunpeng Chen, Xi Chen, Liang Li, and Hengshuang Zhao. Hint2Gen: Bridging understanding and generation via code-structured hints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36593–36603, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Tu_Hint2Gen_Bridging_Understanding_and_Generation_via_Code-structured_Hints_CVPR_2026_paper.html.

Chi Zhang, Feng Gao, Baoxiong Jia, Yixin Zhu, and Song-Chun Zhu. RAVEN: A dataset for relational and analogical visual REasoNing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5317–5327, 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Zhang_RAVEN_A_Dataset_for_Relational_and_Analogical_Visual_REasoNing_CVPR_2019_paper.html.

Zhimu Zhou, Yanpeng Zhao, Qiuyu Liao, Bo Zhao, and Xiaojian Ma. Probing visual planning in image editing models. In *The First Workshop on Efficient Spatial Reasoning at ICLR*, 2026. URL https://openreview.net/forum?id=RdhG125Xe8.

A SAME-PROBLEM MODEL COMPARISON

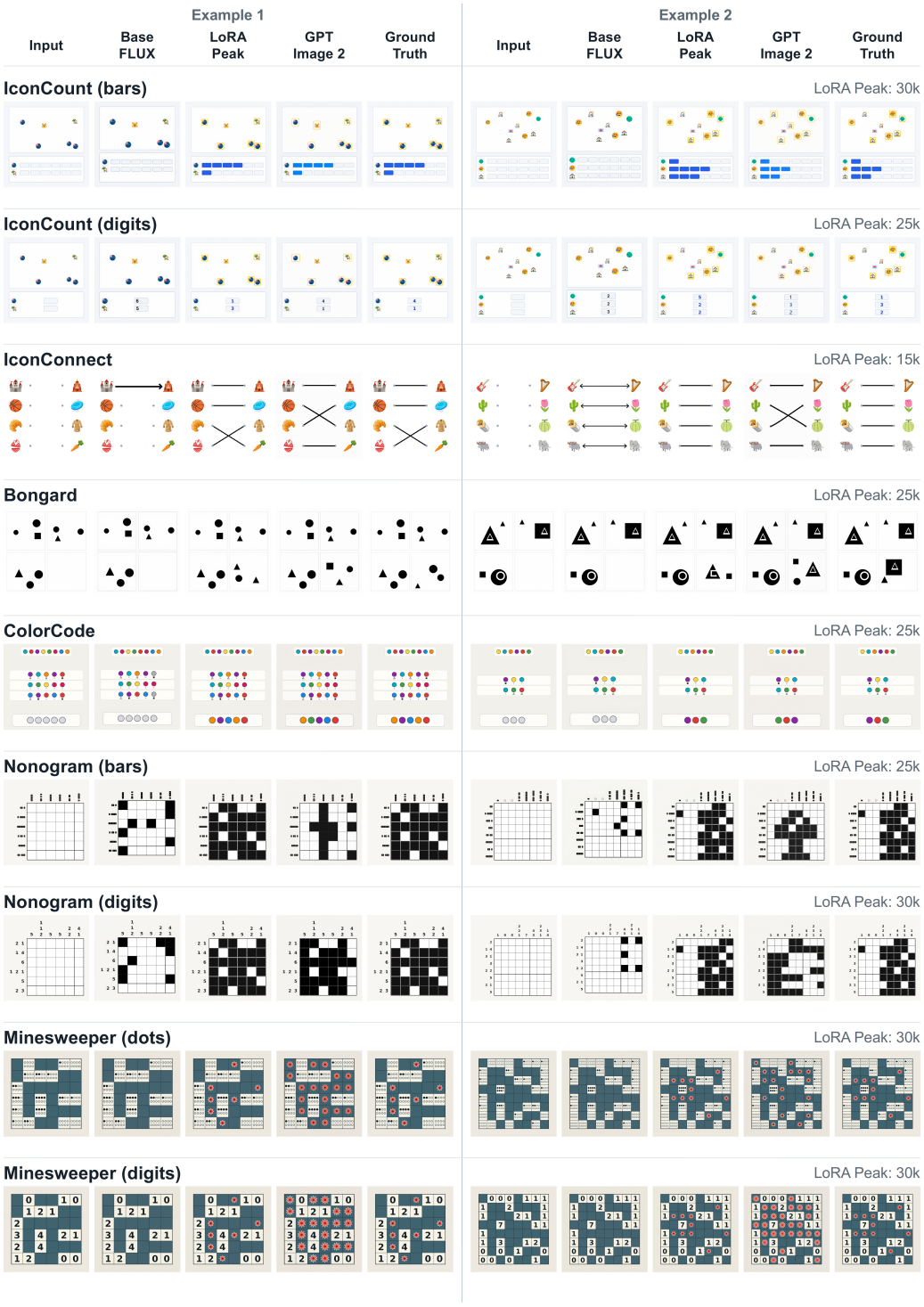

Figure 6: **Full same-problem comparison (1/3).** Nine configurations, each with two test examples. The five columns repeat: Input, Base FLUX, LoRA Peak, GPT Image 2, Ground Truth. Labels give the selected LoRA Peak step.

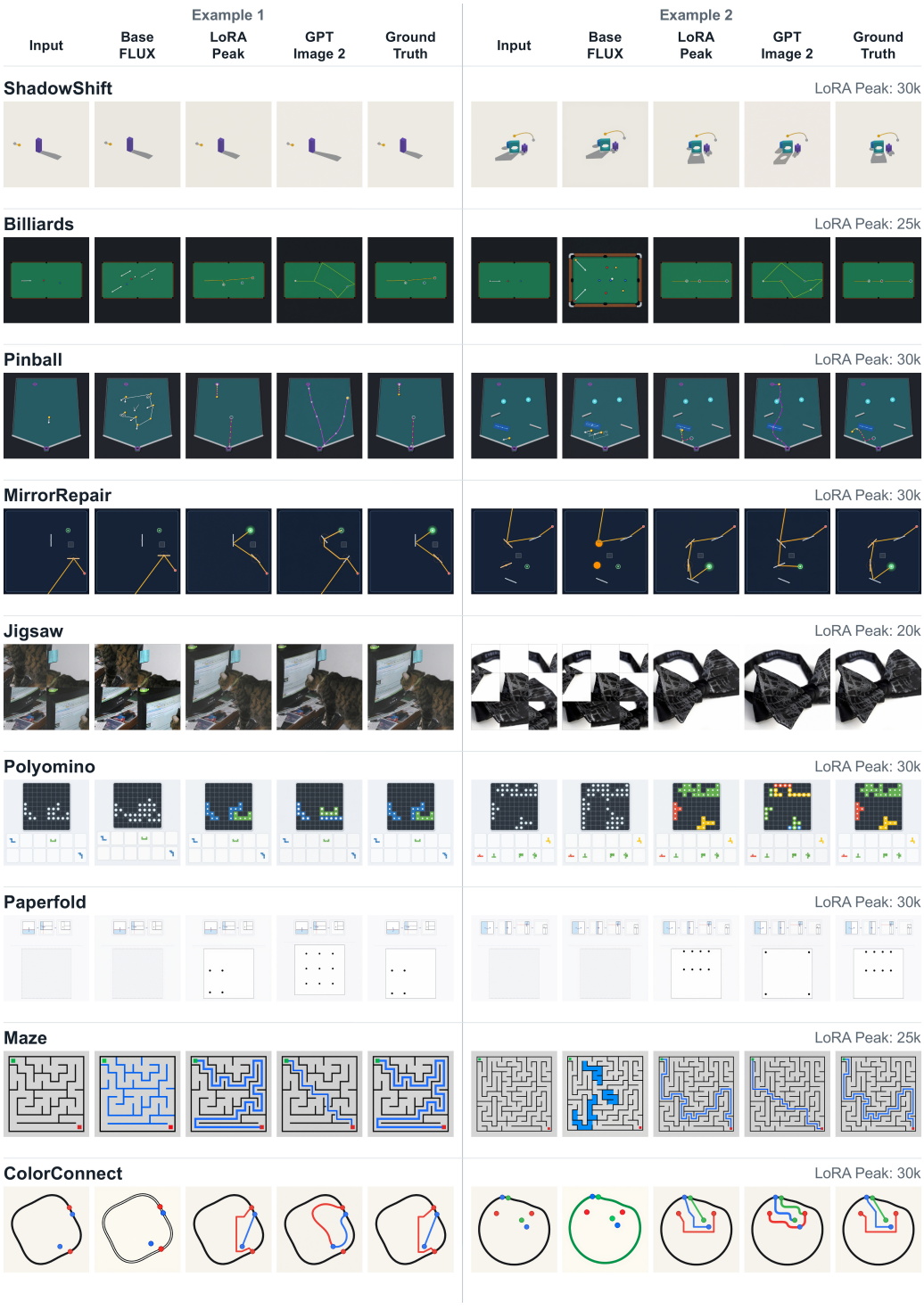

Figure 7: **Full same-problem comparison (2/3).** Nine further configurations, each with two test examples in repeated Input, Base FLUX, LoRA Peak, GPT Image 2, Ground Truth columns. Labels give the selected LoRA Peak step.

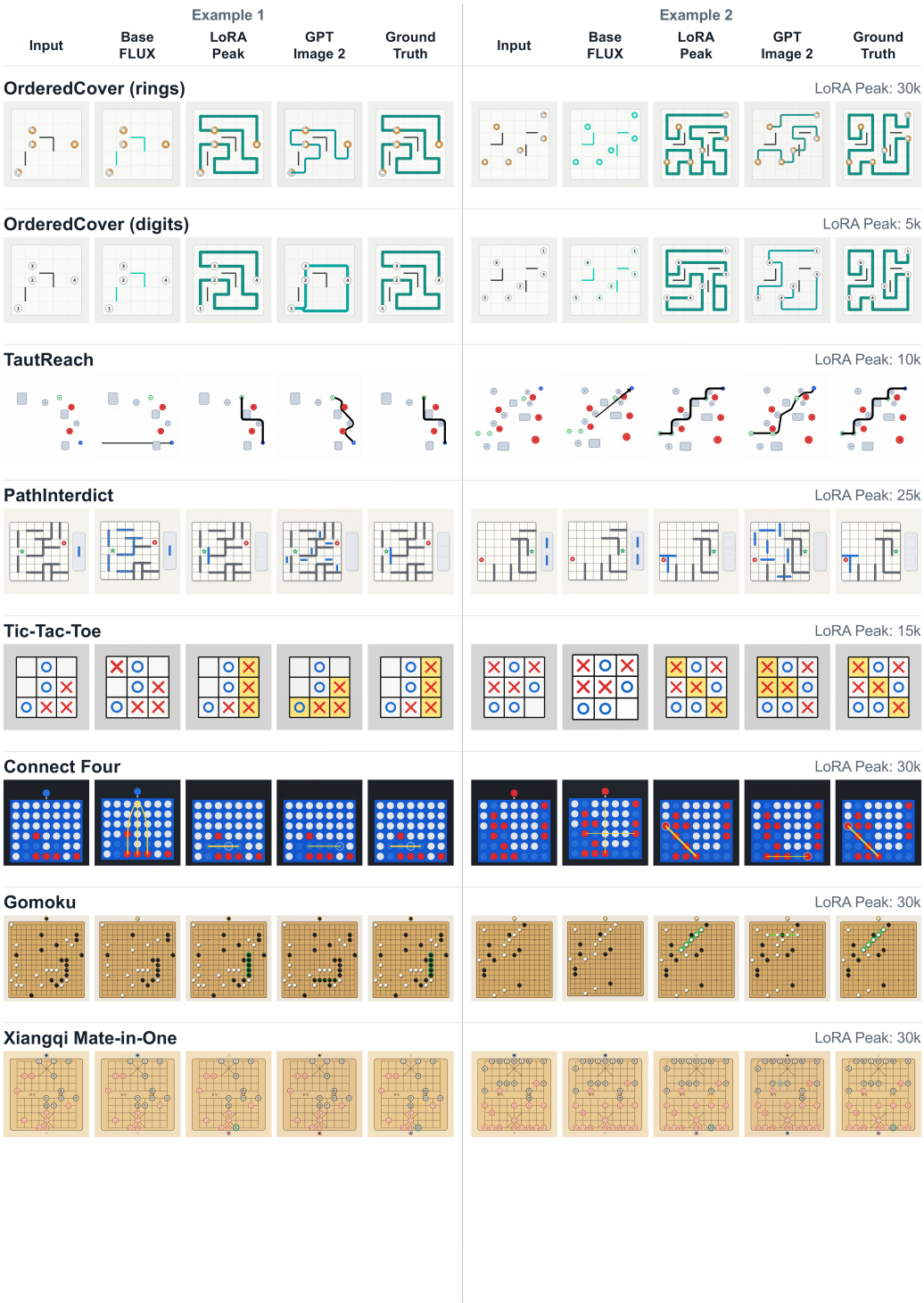

Figure 8: **Full same-problem comparison (3/3).** The remaining eight configurations, each with two test examples in repeated Input, Base FLUX, LoRA Peak, GPT Image 2, Ground Truth columns. Labels give the selected LoRA Peak step.

Peaks use recorded test metrics, taking the earliest tied step. Pinball/ShadowShift use continuous scores; Billiards/Jigsaw/Bongard use diagnostics (Appendix F).

B Task Catalog and Dataset Organization

Table 3 records each task's visual answer and sample counts. A dagger marks the four tasks with two aligned configurations. Every version retains its own problem/answer rendering, while the underlying problem identities and train/test membership remain paired. IconConnect's two evaluation subsets are 150 pair-test and 50 concept-test examples; Paperfold's test set contains 100 in-distribution and 100 composition examples.

Table 3: **Task catalog.** Counts are training/test examples per configuration. Four tasks (†) each have two configurations. The family totals are 1/1/4/4/3/5/4.

Task	Required answer	Train / test
Counting		
IconCount†	Highlight target icons and express counts as bars or digits.	6,000 / 200
World Knowledge		
IconConnect	Draw four one-to-one semantic connections between icons.	2,188 / 200
Rule and Constraint Reasoning		
Bongard	Infer a rule from three positive panels and draw a distinct valid fourth example.	6,000 / 200
ColorCode	Infer the hidden color sequence from positional feedback.	6,000 / 200
Nonogram†	Fill the grid to satisfy row and column run clues.	6,000 / 200
Minesweeper†	Mark exactly the mines implied by revealed clues.	6,000 / 200
Physical Reasoning		
ShadowShift	Update cast shadows for the specified light movement.	6,000 / 200
Billiards	Draw complete trajectories and ball positions after four seconds.	6,000 / 200
Pinball	Draw a three-second trajectory, timed marks, events, and final state.	6,000 / 200
MirrorRepair	Rotate adjustable mirrors and repair the beam to reach the receiver.	6,000 / 200
Spatial Reasoning		
Jigsaw	Reconstruct a photograph from shuffled square tiles.	6,000 / 200
Polyomino	Cover target cells with candidate pieces; rotation allowed, reflection forbidden.	6,000 / 200
Paperfold	Draw all holes obtained by unfolding the punched sheet.	6,000 / 200
Path Planning and Intervention		
Maze	Draw a valid start-to-goal path.	6,000 / 200
ColorConnect	Connect all matching-color pairs without forbidden contacts.	6,000 / 200
OrderedCover†	Visit every cell once in checkpoint order without crossing walls.	6,000 / 200
TautReach	Draw one taut arm through all targets, respecting supports and hazards.	6,000 / 200
PathInterdict	Place all barriers to maximize the shortest route while preserving reachability.	6,000 / 200
Board-Game Tactics		
TicTacToe	Place the uniquely winning X and highlight a completed winning line.	1,078 / 100
Connect Four	Place the winning disc, respecting gravity, and mark its line.	6,000 / 200
Gomoku	Place the uniquely winning stone and highlight the completed five.	6,000 / 200
Xiangqi Mate-in-One	Make the unique legal checkmating move and mark its endpoints.	6,000 / 200

B.1 Problem Construction and Reference Answers

Drawing Answers combines procedural scenes, symbolic puzzles, and selected visual assets. Table 4 describes how these sources become paired problem and answer images. Each record also stores the task state underlying its reference, such as a board, cell set, path, or geometric scene.

Table 4: **Dataset construction by task family.** The problem and reference share the same scene; the reference renders the requested answer from its symbolic, geometric, or semantic representation.

Family	Problem construction	Reference generation
Counting	IconCount composes icon scenes and specifies target categories.	Asset identities and positions determine the highlighted instances and counts, rendered as bars or digits.
World knowledge	IconConnect arranges four icons on each side using semantic relations between their concepts.	The annotated one-to-one matching supplies four connecting lines.
Rule and constraint reasoning	Bongard samples three geometric examples of a common rule. ColorCode, Nonogram, and Minesweeper render color feedback, run clues, and revealed mine-count clues, respectively.	Bongard samples a distinct fourth rule-consistent scene. The other tasks use solved color sequences, binary grids, or mine locations to render their answers.
Physical reasoning	Billiards and Pinball specify objects and initial motion. ShadowShift varies the light in a geometric scene. MirrorRepair specifies a source, mirrors, and receiver.	Simulation supplies trajectories, events, and final states; scene geometry supplies updated shadows; mirror orientations and beam tracing supply the repaired optical path.
Spatial reasoning	Jigsaw crops and shuffles ImageNet photographs. Polyomino combines target cells and candidate pieces. Paperfold specifies a fold-and-punch program.	The source crop supplies Jigsaw's reconstruction. Piece placements supply Polyomino's covering, and geometric unfolding supplies Paperfold's hole locations.
Path planning and intervention	Maze and ColorConnect specify traversable regions and terminals. OrderedCover adds ordered checkpoints and walls. TautReach places targets, supports, and hazards. PathInterdict adds movable barriers and a placement budget.	References store routes, cell-covering paths, or support-wrapped curves. PathInterdict enumerates legal barrier placements and selects the one maximizing the remaining shortest-path length.
Board-game tactics	TicTacToe, Connect Four, and Gomoku use positions with one immediate winning move. Xiangqi uses positions with one legal checkmating move.	Game rules determine the move and resulting board. The reference adds the move and its task-specific winning-line or endpoint highlights.

Partitioning and paired variants. Train and test assignments follow task-specific problem identities: for example, Jigsaw uses source photographs, Pinball uses layout and symmetry groups, Nonogram uses solution and clue families, and Xiangqi uses position lineages. IconConnect provides 150 pair-test and 50 concept-test examples; Paperfold provides 100 in-distribution and 100 composition-test examples. The four paired encodings retain the same problem records and split membership while changing count, clue, or checkpoint representations. Table 3 lists per-configuration sizes, and Appendix G gives the native complexity distributions.

C Training, Inference, and Reproducibility

Table 5: **Recorded experiment settings.** LoRA and base FLUX share the test inputs, prompts, seeds, and sampling settings.

Setting	Value
Pretrained model	FLUX.1-Kontext-dev
Adaptation	26 independent task/configuration LoRA adapters
Rank / training alpha	128 / 128
Optimizer	AdamW; nominal learning rate 10^{-4}; weight decay 0.01
Batch / accumulation	1 / 1
Training budget	30,000 optimizer updates per configuration
Checkpoints	5k, 10k, 15k, 20k, 25k, 30k
Precision / hardware	bf16; one NVIDIA A100 80GB per adapter
FLUX inference steps	30
FLUX guidance	Classifier-free guidance 1.0; embedded guidance 3.5
Other FLUX settings	CPU random-number device, empty negative prompt, no tiling
Native resolution	512×512 for IconConnect; 1024×1024 otherwise
GPT model / quality	`gpt-image-2` / `medium`
GPT resolution / output	1024×1024 / one PNG

The adapted diffusion-transformer module names are `a_to_qkv`, `b_to_qkv`, `ff_a.0`, `ff_a.2`, `ff_b.0`, `ff_b.2`, `a_to_out`, `b_to_out`, `proj_out`, `norm.linear`, `norm1_a.linear`, `norm1_b.linear`, and `to_qkv_mlp`. The wrappers use the framework's `ConstantLR` implementation and scheduler-weighted flow-matching loss. Gradient checkpointing is enabled; only adapter parameters are updated.

The base comparison retains the exact recorded prompts and seeds for all 5,100 FLUX test cases. Task prompt wording can vary where the original dataset specifies variants. Seeds use the task's recorded row-index or instance-seed rule. The base mode loads no LoRA weights and does not pass answer images or solution fields to the model. Maze, Jigsaw, and TicTacToe reuse their completed matched base predictions; the remaining configurations were completed through the common base-evaluation suite.

The dataset has no separate validation split. Some early training runs generated monitoring images for a small number of test questions; later runs used training examples or disabled online previews. The main binary comparison reports the maximum test outcome across the six checkpoints per configuration, taking the earliest step in a tie. This is a descriptive test-set peak, not a validation-selected estimate; all intermediate outcomes remain available. The paired representation table also uses per-configuration peaks; component analyses use the same per-configuration peak checkpoints. Continuous scores and additional diagnostics use the peak of their respective reported metric. Each encoding is trained in a separate run on its aligned split.

The supplementary package links the code and experiment configurations to the recorded prompts, seeds, inference settings, and output identifiers. Evaluation scripts implement the criteria in Appendix D, and per-sample results support recomputing the reported aggregate outcomes. The September 2026 base release adds 5,100 predictions to the 30,600 LoRA and 5,300 GPT outputs, for 41,000 archived generated test images. The total includes checkpoint repetitions, paired configurations, and the additional Bongard prompt run; these are outputs, not independent test questions. The base suite's gold-adapter checks cover 4,600 reference cases for its 23 newly managed configurations.

D Evaluation Criteria and Calibration

The outcome names in the main table refer to the following checks. Exact extraction settings, thresholds, and adapters are stored with the corresponding run records.

Task-appropriate calibration includes positive reference answers and negative controls. All 200 Maze references and 100 TicTacToe references pass their respective checks, while the corresponding

Table 6: **Meaning of the task-level metrics.**

Task	Reported criterion
Maze	STSR: route connectivity and topology checked against original grid walls and endpoint markers.
TicTacToe; Connect Four	Correct resulting board and the required winning-cell/line annotation.
Gomoku	Correct one-stone edit and the required winning-line annotation under the strict contract.
ColorConnect	Legal routing of all pairs, scene integrity, and avoidance of forbidden contacts; alternative valid paths accepted.
TautReach	One open continuous path, correct endpoints and targets, taut free spans, valid support contact, no forbidden collisions, and scene preservation.
IconConnect	Correct semantic matching with the strict line and scene checks.
IconCount	Correct counts, required instance highlights, and output/preservation checks. Counting-only accuracy is separate.
Xiangqi Mate-in-One	Exact logical resulting board over all 90 intersections; endpoint rings are outside this board metric.
Paperfold	Exact hole count and a one-to-one match of all centers within one hole diameter.
ColorCode	Complete correct hidden sequence and the strict edit/preservation conditions.
Nonogram; Minesweeper	Exact filled-cell or mine set, with the task's strict edit/preservation checks.
OrderedCover	Complete legal ordered path and the required grid, checkpoint, and wall preservation.
MirrorRepair	Correct repair satisfying the optical and visual evidence gates.
Polyomino	Exact board color labels, legal partition into candidate pieces, and tray/background preservation.
PathInterdict	Legal use of all barriers, optimal remaining shortest-route length, and required scene preservation.

unchanged-input controls fail. Connect Four records 200 valid references and no passing unchanged inputs. Control outcomes depend on the contract: IconCount can legitimately contain zero-count answers, so not every unchanged input is necessarily incorrect.

Several auxiliary metrics distinguish answer components. The following component comparisons use each task's peak checkpoint. At 10k, TautReach achieves 100.0% for both ordinary valid paths and valid taut paths. Xiangqi's exact move rate is 12.5%, compared with 12.0% exact resulting-board accuracy. Paperfold reaches 15.0% under the one-diameter matching tolerance; a tighter one-radius criterion yields 5.0%. These variants are not substituted for one another in cross-model comparisons.

Connect Four counts gravity-invalid predicted boards as failures.

E Full Training Trajectories

Table 7 reports every checkpoint for all 26 configurations. The same canonical cases are evaluated throughout a task's sweep. Table 1 takes the maximum of each configuration's binary outcomes and reports the earliest step attaining it. The full trajectory exposes gains, ties, and declines that a peak alone does not show.

Table 7: **All six checkpoints.** Unmarked values are percentages under Table 1 criteria. ‡ denotes a continuous score; † denotes a diagnostic (Appendix F).

Task / configuration	5k	10k	15k	20k	25k	30k
Maze	63.0	77.5	68.5	90.0	91.5	88.5
Tic-Tac-Toe	73.0	77.0	95.0	92.0	83.0	88.0
Connect Four	33.0	63.5	50.5	82.5	75.5	87.0
Gomoku	49.5	71.5	71.0	78.0	82.5	92.0
ColorConnect	70.5	72.0	84.0	86.0	85.5	87.0
TautReach	99.0	100.0	99.5	100.0	94.0	99.0
IconConnect	1.5	2.5	6.0	2.5	3.0	6.0
IconCount (bars)	6.0	16.5	16.5	20.5	27.5	35.5
IconCount (digits)	9.5	7.0	9.5	10.0	11.0	10.5
Xiangqi Mate-in-One	4.5	4.5	6.0	8.5	5.5	12.0
Paperfold	0.0	0.0	0.0	0.0	0.0	15.0
ColorCode	13.0	17.5	33.5	36.0	51.5	51.0
Minesweeper (dots)	0.0	5.5	13.0	18.5	13.0	31.5
Minesweeper (digits)	0.0	2.0	10.5	16.5	11.5	24.5
Nonogram (bars)	0.5	1.5	10.5	13.5	18.5	18.5
Nonogram (digits)	0.0	0.5	3.0	4.5	8.5	10.5
OrderedCover (rings)	0.0	0.5	0.5	0.0	0.5	1.0
OrderedCover (digits)	0.5	0.0	0.0	0.0	0.0	0.5
MirrorRepair	1.0	4.0	8.5	12.0	19.0	23.5
Polyomino	2.5	14.0	17.5	18.5	22.5	25.5
PathInterdict	2.0	1.0	3.0	2.5	4.0	3.5
Pinball‡	0.5633	0.6089	0.6331	0.6455	0.6460	0.6717
ShadowShift‡	0.4968	0.6234	0.7198	0.7412	0.7002	0.7692
Billiards†	8.0	21.5	21.5	23.0	31.5	15.0
Jigsaw†	48.0	69.5	63.5	77.0	74.0	69.5
Bongard†	54.0	29.5	60.5	58.0	68.5	46.5

F Continuous Scores and Additional Diagnostics

Table 8: **Continuous outcomes at LoRA Peak.** Higher is better; values are on a 0–1 scale. Both peaks occur at 30k.

Task	Metric	Base	LoRA Peak	GPT
Pinball	Soft trajectory score	0.0024	0.6717	0.1098
ShadowShift	Shadow IoU	0.1651	0.7692	0.1949

Pinball: trajectory agreement. The primary score combines soft precision with segment-balanced soft recall. At 1024×1024 resolution, path pixels receive full spatial credit within eight pixels of the other trajectory, with credit decreasing linearly to zero at 30 pixels. Precision averages credit over generated path pixels; recall is first computed per reference motion segment and then averaged across segments, so a short omitted segment still contributes to the score. Their harmonic mean is multiplied by $0.6 + 0.4b$, where $b \in [0, 1]$ penalizes spurious direct bridges between portal entry and exit. Final-ball localization and scene preservation are recorded separately; equal-time markers and event rings are not gates in this trajectory score.

ShadowShift: final shadow geometry. Shadow IoU compares the extracted output shadow mask with the reference final shadow on the visible ground. Object pixels and instructional overlays are excluded from the comparison. This isolates the requested geometric change from the large unchanged background. ShadowShift also reports change-region IoU: 0.0707 for base FLUX, 0.6861 for LoRA at 30k, and 0.2465 for GPT Image 2. This evaluates the updated region and is distinct

from the final shadow-mask IoU used above. Pinball's soft score measures trajectory quality rather than a binary solved rate.

Table 9: **Automatic diagnostics at LoRA Peak (%).** Each row selects the highest recorded diagnostic across the six checkpoints, taking the earliest tied step.

Task	Diagnostic	Base	LoRA Peak	Step	GPT
Billiards	Automatic geometric success	0.0	31.5	25k	0.0
Jigsaw	Legacy patch-matching exact	0.5	77.0	20k	34.5
Bongard	Unlabeled-role proxy	0.0	68.5	25k	53.5

Billiards reports the fraction satisfying the parser's geometric tests; Jigsaw measures patch correspondence rather than discrete tile arrangement. Bongard matches unrendered shape roles under permutation; the GPT value uses prompt v2 (v1: 65.0%). These diagnostics are not validated rule-level success rates.

G Dataset Complexity Distributions

The atlas contains 20 blue distribution bars, each based on 200 test examples. Table 10 gives their underlying levels and percentages in readable form. Native difficulty tiers and structural descriptors are different axes; equal colors do not assert equal cognitive difficulty across tasks. TicTacToe and Connect Four are omitted from these bars because their available descriptors were not used as a named ordered scale in this figure.

Table 10: **Atlas distributions.** Percentages correspond to the ordered levels in each row, with 200 test cases per row. E/M/H and L/M/H mean easy/medium/hard and low/medium/high. Structural descriptors do not imply equal difficulty across tasks.

Task	Axis	Levels	Percentages
IconCount	Requested categories	1 / 2 / 3	29.5 / 38 / 32.5
IconConnect	Line crossings	0 / 1 / 2 / 3 / 4 / 5	6.5 / 20.5 / 33.5 / 26 / 7 / 6.5
Bongard	Difficulty	E / M / H	30 / 40 / 30
ColorCode	Difficulty	L / M / H	24.5 / 49 / 26.5
Nonogram	Difficulty	E / M / H	30 / 40 / 30
Minesweeper	Difficulty	E / M / H	30 / 50 / 20
ShadowShift	Object count	1 / 2 / 3	35 / 40 / 25
Billiards	Difficulty	E / M / H	36.5 / 47 / 16.5
Pinball	Scene level	0 / 1 / 2 / 3 / 4 / 5	16 / 16 / 17 / 17 / 17 / 17
MirrorRepair	Adjustable mirrors	1 / 2	35 / 65
Jigsaw	Grid side length	2 / 3 / 4	30 / 40 / 30
Polyomino	Pieces used	3–10	12.5 each
Paperfold	Fold count	1 / 2 / 3	20 / 40 / 40
Maze	Grid side length	4–8 / 9–14 / 15–20	28 / 37.5 / 34.5
ColorConnect	Measured difficulty	L1 / L2 / L3 / L4 / L5	15 / 22.5 / 27.5 / 22.5 / 12.5
OrderedCover	Difficulty	E / M / H	30 / 40 / 30
TautReach	Difficulty	L / M / H	30 / 40 / 30
PathInterdict	Barrier budget	1 / 2	50 / 50
Gomoku	Difficulty	E / M / H	33.5 / 33.5 / 33
Xiangqi	Difficulty quartile	q1 / q2 / q3 / q4	21.5 / 27.5 / 23 / 28

Maze is the sole display aggregation: its native grid side lengths 4–20 are grouped as 4–8, 9–14, and 15–20, yielding 56, 75, and 69 cases. Polyomino retains eight equal strata, with 25 cases at each used-piece count from 3 through 10. The atlas ring counts semantic task types once; the dataset's 147,266 training and 5,100 test pairs count all 26 configurations, including the four paired representations.

H Example-Level Component Analysis

These cases connect the component analysis in Section 4.4 to individual test predictions. FLUX examples use the per-configuration peak checkpoints; the GPT example uses its recorded test run.

IconCount: correct selection, incorrect digit. The required counts in this example are $[3, 6]$. The digit adapter at its 25k peak outputs $[6, 6]$, with both digits valid under the decoder, all nine target instances correctly highlighted, and scene preservation passing. Its failure is the first count. The aligned bar-encoding prediction at its 30k peak for the same problem gives $[3, 6]$ and passes strict evaluation.

OrderedCover: complete coverage, incorrect checkpoint order. The ring-encoded prediction covers every cell but encounters the checkpoints in rank order $1 \rightarrow 3 \rightarrow 2 \rightarrow 4$. This sequence violates the required increasing order. The example also fails the preservation check.

IconCount: correct counts, missing category highlights. GPT predicts the correct counts $[5, 6]$ for envelopes and shoes. The image shows boxes around the six shoes but leaves all five envelopes unmarked. The omission of the envelope category is directly visible: the numerical answer is correct, while the requested instance marking is incomplete.