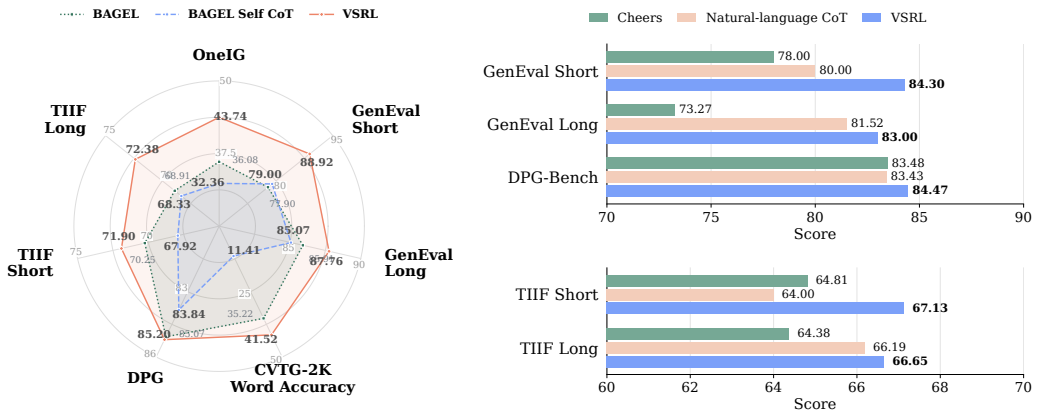
BEYOND VERBAL REASONING: VISUAL STRUCTURE REASONING LANGUAGE FOR IMAGE GENERATION IN UNIFIED MULTIMODAL MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent unified multimodal models (UMMs) have begun to introduce explicit natural-language reasoning before image generation. We revisit this default choice and propose Visual Structure Reasoning Language (VSRL), a structured reasoning language for image generation in UMMs. Given a prompt, the UMM autonomously generates VSRL comprising the global scene, objects, and their relations, and conditions image generation on it. We develop a VSRL data construction pipeline that converts paired prompts and target images into verified prompt–VSRL–image triplets. The resulting data are used to train the UMM to generate and use VSRL through supervised fine-tuning. We further introduce StruFlow-GRPO, a reinforcement learning framework that jointly optimizes reasoning and generation with programmatic structure rewards and image rewards. On BAGEL, VSRL outperforms both the base model and BAGEL's native natural-language reasoning (Self CoT) across five generation benchmarks. Compared with Self CoT, VSRL improves GenEval Short from 79.0 to 88.9 and OneIG-Bench from 32.4 to 43.7. On Cheers, we train two separate models on VSRL and natural-language CoT data for a controlled comparison. VSRL outperforms both natural-language CoT and the base model across all evaluated generation benchmarks, including GenEval and TIIF-Bench. These results highlight reasoning representation as an important design dimension for UMM image generation and the potential of generation-oriented structured reasoning across UMMs.



(a) BAGEL (b) Cheers

Figure 1: **Generation performance and reasoning representation.** (a) BAGEL: comparison of the base model, native Self CoT, and VSRL across five generation benchmarks. (b) Cheers: comparison of the base model, natural-language CoT, and VSRL across three generation benchmarks. Short and Long versions are shown separately where applicable.

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.

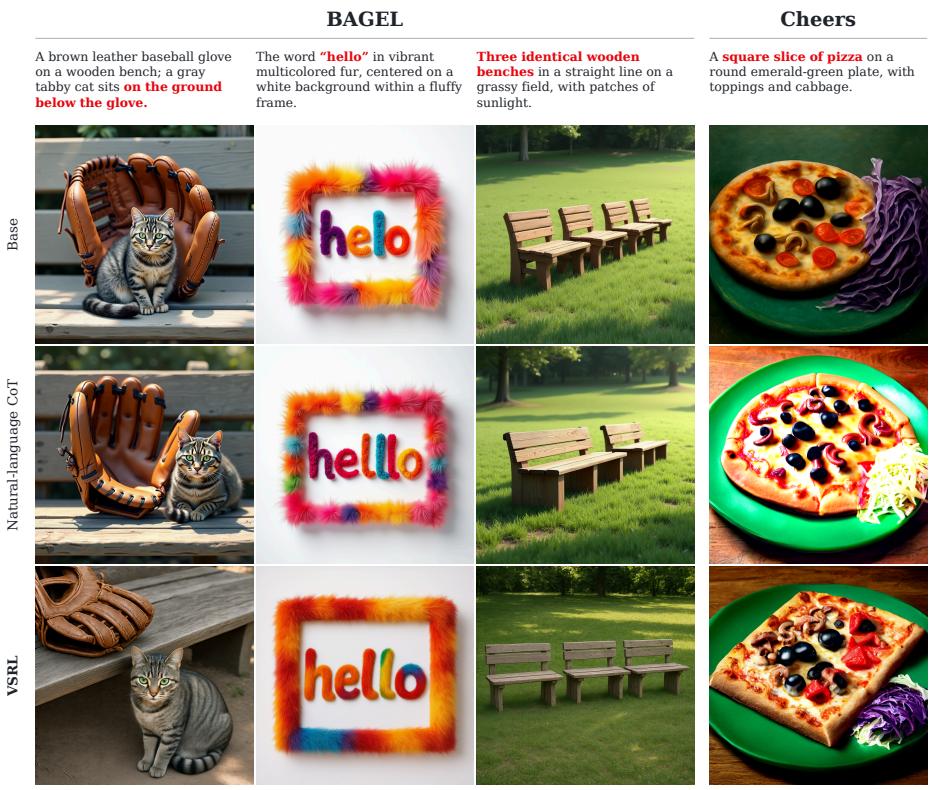


Figure 2: **Qualitative comparisons on BAGEL and Cheers.** Rows show the base model, natural-language CoT, and VSRL. Natural-language CoT denotes native Self CoT for BAGEL and the supervised control for Cheers; VSRL uses SFT + StruFlow-GRPO for BAGEL and SFT for Cheers. Prompts are abbreviated. A full prompt and paired reasoning outputs are provided in Appendix B.2.

1 Introduction

Recent advances in Unified Multimodal Models (UMMs) have enabled a single model to both understand and generate visual content (Chameleon Team, 2024; Deng et al., 2025). Building on this capability, recent work introduces explicit reasoning before image generation and further improves instruction following and generation quality through supervised fine-tuning, reinforcement learning, or post-generation reflection and iteration (Jiang et al., 2025; Zhang et al., 2025; Huang et al., 2025; Wu et al., 2026).

Existing approaches primarily focus on how reasoning is generated (Zhang et al., 2025), optimized (Jiang et al., 2025), or refined (Huang et al., 2025; Wu et al., 2026), while retaining natural language as the default intermediate reasoning representation. Complex image generation requests often require models to satisfy interdependent visual constraints involving multiple objects, their attributes, and their relations simultaneously (Duan et al., 2026). In natural-language reasoning, these visual constraints are expressed sequentially through textual descriptions, with objects, attributes, and relations conveyed through linguistic context (Figure 3). This motivates us to revisit a common design choice: **Should intermediate reasoning for image generation continue to use natural-language representations?**

We propose Visual Structure Reasoning Language (VSRL), a structured reasoning language for image generation in UMMs. VSRL organizes the target scene into global scene information, objects with explicit identities and attributes, and relations expressed through object references (Figure 3). Given a user request, the same UMM first autonomously generates VSRL and then conditions its native image generation branch on the original request and its own generated VSRL to produce an image (Section 3.1). Figure 2 provides qualitative comparisons on BAGEL and Cheers.

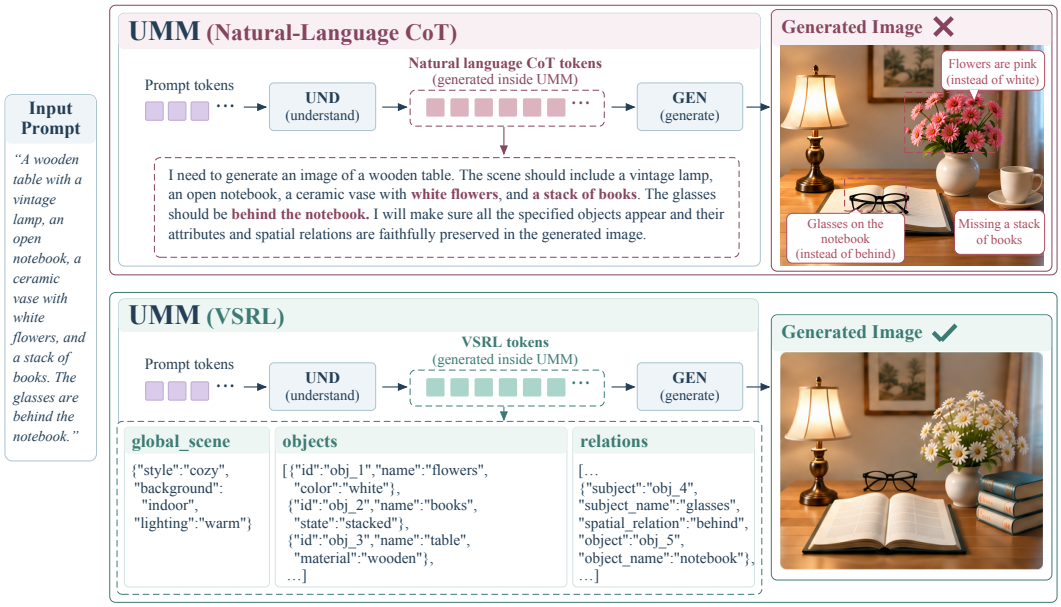


Figure 3: **Two intermediate reasoning formats for the same prompt.** Natural-language CoT linearizes scene constraints; VSRL organizes global scene information, objects with their attributes, and relations between objects before the same UMM generates the image.

We develop a VSRL data construction pipeline that converts paired prompts and target images into verified prompt–VSRL–image triplets. The resulting data are used for supervised fine-tuning to teach the UMM to generate and use VSRL. We further introduce StruFlow-GRPO, a reinforcement learning framework initialized from the supervised checkpoint. StruFlow-GRPO optimizes the model on its own generated VSRL–image trajectories with programmatic structure rewards and image feedback and jointly updates the language and image-generation modules (Figure 4; Section 3.4).

We evaluate VSRL on two UMMs with different architectures. On the MoT-based UMM BAGEL (Deng et al., 2025), VSRL outperforms both the base model and native Self CoT across five generation benchmarks (Figure 1(a)). On the dense-Transformer-based UMM Cheers (Zhang et al., 2026), we train two separate models on VSRL and natural-language CoT data under the same training settings for a controlled comparison. VSRL consistently outperforms natural-language CoT across all evaluated generation benchmarks and also improves over the base model (Figure 1(b); Table 3). Together, these results validate VSRL across different UMM architectures and provide more direct evidence that the intermediate reasoning representation affects image generation (Sections 4.2 and 4.4).

Our main contributions are as follows:

- We introduce VSRL, a structured reasoning language for image generation in UMMs that enables models to autonomously organize the global scene, objects, and their relations and generate images based on their own reasoning.
- We introduce StruFlow-GRPO, a reinforcement learning framework that uses programmatic structure rewards together with image feedback on the model's own VSRL–image trajectories to jointly update the language and image-generation modules.
- We validate VSRL on MoT-based BAGEL and dense-Transformer-based Cheers, showing consistent gains over the base models and natural-language CoT across all evaluated generation benchmarks. The controlled comparison on Cheers further provides more direct evidence for the role of the intermediate reasoning representation.

2 Related Work

Unified multimodal understanding and generation. Existing studies use different visual representations and generation objectives to support visual understanding and image generation within a single model. One line of work represents images as discrete tokens and jointly models text and images through autoregressive prediction (Chameleon Team, 2024; Wang et al., 2024); another combines autoregressive language modeling with diffusion or flow matching, using different generation objectives for text and images (Xie et al., 2024; Zhou et al., 2024; Ma et al., 2024; Chen et al., 2025a; Xie et al., 2025). Among these models, BAGEL adopts a Mixture-of-Transformers (MoT) architecture and provides a native think mode with natural-language reasoning capabilities developed through dedicated training (Deng et al., 2025). We therefore choose BAGEL as our primary backbone and use its native thinking mode, referred to as Self CoT in this paper, as a key reference for evaluating VSRL. Cheers uses a dense LLM-based Transformer backbone together with cascaded flow matching, providing another backbone for examining VSRL under a different model design (Zhang et al., 2026).

Reasoning for image generation. Reasoning before generation guides image synthesis by analyzing user intent, supplementing implicit information, and planning the target scene. Existing studies improve natural-language reasoning through supervised fine-tuning and reinforcement learning (Jiang et al., 2025; Zhang et al., 2025), and also explore semantic–spatial reasoning that includes object descriptions and positional coordinates (Duan et al., 2026). Another line of work incorporates intermediate images into the reasoning process, progressively refining results through alternating analysis and generation, or through reflection and multi-turn editing (Huang et al., 2025; Qin et al., 2025; Wu et al., 2026; Ye et al., 2026). This paper studies how reasoning before generation is expressed. VSRL organizes global scene information, objects with explicit identities and attributes, and relations expressed through object references, and we train this reasoning process in UMMs that use flow matching. We additionally compare VSRL with natural-language reasoning in a controlled setting.

Post-training generative models. Post-training improves image generation through better supervision or reward signals (Jin et al., 2025). Reconstruction-based training uses the model's own visual semantic features as supervision, reducing reliance on text annotations (Xie et al., 2026); reward-based training treats denoising as a sequential decision process and improves generation through policy optimization (Black et al., 2023; Fan et al., 2023; Xue et al., 2025). Flow-GRPO applies group-relative policy optimization to flow-matching models and forms the basis of our reinforcement learning stage (Liu et al., 2025). Existing studies also jointly optimize text reasoning and image generation, covering both autoregressive image generation and settings that combine text reasoning with image denoising (Jiang et al., 2025; Ye et al., 2026). Building on this work, we propose StruFlow-GRPO for VSRL, combining programmatic structure rewards with image feedback to jointly train the language and image-generation modules under the model's own generated structural conditions.

3 Method

3.1 Visual Structure Reasoning Language

Representation. Given a user request P, the model first reasons about the target scene using VSRL to obtain a structure S, then generates an image I based on this structure. VSRL consists of three components: global scene, objects, and relations. The global-scene component describes global scene information, including the environment, background, composition, and viewpoint; the objects component assigns unique identifiers to objects and associates attributes such as category, count, appearance, and state with the corresponding objects; and the relations component expresses spatial relations and interactions through object identifiers. Object attributes and relation references thus point to consistent entities and explicitly distinguish multiple objects of the same category. VSRL follows fixed field definitions, and relation endpoints must reference declared objects. Appendix B provides the complete fields and the attribute configurations used by different backbones.

Generation process. The model uses VSRL to infer the objects, attributes, and relations involved in the request and supply the details needed to form the target scene. The overall generation process is expressed as

$$p_\theta(S, I \mid P) = p_\theta(S \mid P)\, p_\theta(I \mid P, S). \tag{1}$$

In contrast, direct generation without intermediate reasoning models the image as $p_\theta(I \mid P)$.

The same UMM first generates S autoregressively and then synthesizes I through its native image generation branch, conditioned on the original request P and its own generated structure S. At inference time, the model receives only the user request and sequentially performs structured reasoning and one complete image generation process.

3.2 VSRL Supervision Construction

To construct VSRL supervision, we develop a data construction pipeline that converts paired prompts and target images into prompt–VSRL–image triplets (Figure 4). We first use a vision-language model to organize the prompt's visual requirements into VSRL and incorporate additional visual information from the target image. The resulting annotations represent the global scene, objects with explicit identities and attributes, and relations through object references. Secondly, for samples requiring explicit spatial localization, we use Grounding DINO (Liu et al., 2024) to localize annotated objects and associate detected boxes with their object identities. Finally, we validate each structure using programmatic checks and vision-language model verification; structures that fail validation are regenerated and checked again. Appendices B and D provide the complete data sources, T2I conversion rules, annotation and repair procedures, and backbone-specific configurations.

3.3 Supervised Fine-Tuning

We organize (P, S, I) into text–image training sequences using each backbone's native triggering mechanism. The model performs next-token prediction on the structure S and learns to generate the image I, conditioned on the request P and annotated structure S, using the backbone's original flow-matching objective. For VSRL triplets, the training objective is

$$\mathcal{L}_{\mathrm{VSRL}} = \mathcal{L}_{\mathrm{NTP}}(S \mid P) + \lambda_{\mathrm{FM}} \mathcal{L}_{\mathrm{FM}}(I \mid P, S). \tag{2}$$

Here, λ_{FM} controls the relative weighting of the two objectives, and fixed generation trigger instructions are omitted from the notation. During training, structure tokens use teacher forcing, and image generation is conditioned on annotated VSRL; at inference time, this condition is replaced by the model's own generated VSRL.

VSRL is integrated through the respective text–image boundaries of BAGEL and Cheers, preserving both backbones' original image generation mechanisms. To mitigate the effect of generation-oriented fine-tuning on the model's existing capabilities, we mix native understanding and generation samples into training as replay data, retaining their original formats and training objectives. Training updates the UMM's trainable modules while keeping the VAE parameters frozen. Appendix D provides the specific sequence templates, data mixing ratios, and optimization settings.

3.4 StruFlow-GRPO: Joint Reinforcement Learning

Trajectory Sampling Supervised fine-tuning teaches the model to generate VSRL and synthesize images accordingly. To further optimize generation under the model's own generated structures, we propose StruFlow-GRPO and start StruFlow-GRPO training from the VSRL checkpoint obtained after supervised fine-tuning. For each prompt P, the model samples G structure–image trajectories,

$$o_i = (\hat{S}_i, \hat{I}_i), \qquad i = 1, \ldots, G.$$

where $\hat{S}_i$ denotes the VSRL output and $\hat{I}_i$ denotes the image output for trajectory i. The current policy first generates $\hat{S}_i$ from P, and then uses P and $\hat{S}_i$ to generate $\hat{I}_i$ through its native image generation process.

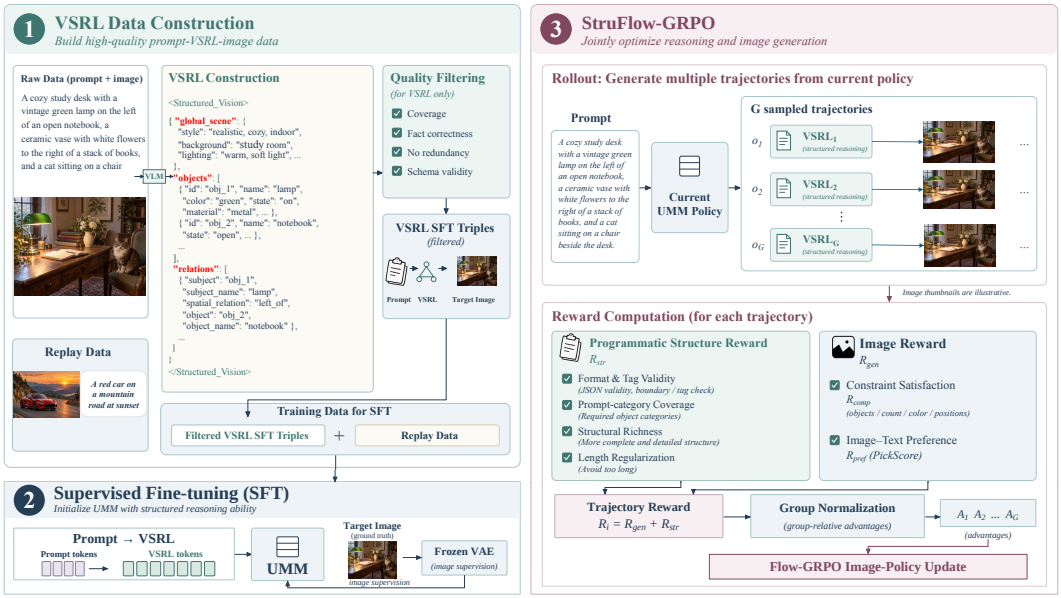


Figure 4: **VSRL training pipeline.** We construct VSRL supervision from paired prompts and target images for supervised fine-tuning, and further optimize reasoning and generation with StruFlow-GRPO using both structure and image feedback.

Reward Design We evaluate both the intermediate structure and the final image of each trajectory and combine these two forms of feedback into the trajectory return:

$$R(o_i) = R_{\text{gen}}(\hat{I}_i; P) + R_{\text{str}}(\hat{S}_i; P), \tag{3}$$

where

$$R_{\text{gen}} = R_{\text{comp}} + \alpha R_{\text{pref}}.$$

R_{comp} evaluates how well the generated image satisfies requested constraints such as objects, counts, colors, and positions, while R_{pref} uses PickScore (Kirstain et al., 2023) to provide image–text preference feedback. For intermediate reasoning, we construct a programmatic structure reward R_{str} that evaluates JSON format, structure boundary markers, prompt category coverage, structural richness, and length. The final trajectory return combines the image and structure rewards; Appendix E provides the specific definitions and weights of all reward terms.

Joint Updates We normalize total returns within the trajectory group for the same prompt to obtain group-relative advantages A_i. We then replay image generation under the actual P and $\hat{S}_i$ conditions used in each trajectory and optimize the corresponding image denoising trajectory with the Flow-GRPO (Liu et al., 2025) policy objective.

Training updates both the language and image generation modules. The image generation branch attends to the VSRL conditioning representations, allowing gradients from the image policy objective to reach the language module through this conditioning path. The VSRL conditioning representations and image synthesis can thus jointly adapt to trajectory feedback. Appendix E provides the complete optimization objective and training settings, and Section 4.3 analyzes the role of joint updates.

4 Experiments

We evaluate VSRL from three perspectives: performance across generation tasks and backbones, comparison with natural-language reasoning, and the roles of supervised fine-tuning and joint reinforcement learning.

Table 1: **Main results on five generation benchmarks.** S/L: Short/Long versions. Scores use a 100-point scale (higher is better); green denotes gains over Self CoT at the displayed precision. †: public results; ‡: third-party reevaluations. See Appendix F for GenEval reporting conventions, OneIG aggregation, and evaluation settings.

Method	GenEval		DPG	TIIF		OneIG	CVTG-2K
	S	L		S	L		Word Acc.
Generation-only models							
FLUX.1-dev (Black Forest Labs, 2024)†	66.00	-	83.84	66.24	66.72	43.4	49.65
SD3.5-Large (Stability AI, 2024)†	71.00	-	-	68.69	64.92	-	-
DALL-E 3 (OpenAI, 2023)†	67.00	-	83.50	74.47	72.94	-	-
Qwen-Image (Qwen Team, 2025)†	87.00	-	88.32	87.00	87.50	53.9	82.88
Seedream 3.0 (Gao et al., 2025)†	84.00	-	88.27	-	-	-	-
GPT-Image-1 (OpenAI, 2025)†	84.00	-	85.15	86.10	87.60	53.3	85.69
Unified multimodal models							
Janus-Pro-7B (Chen et al., 2025b)†	80.00	-	84.19	65.38	61.10	26.7	-
BLIP3-o-8B (Chen et al., 2025a)†	84.00	-	81.60	-	-	30.7	-
Show-o2-7B (Xie et al., 2025)†	76.00	-	86.14	62.80	63.87	30.8	-
Tuna-2-7B (Liu et al., 2026)†	87.00	-	86.54	-	-	-	-
InternVL-U (Tian et al., 2026)†	85.00	-	85.18	74.90	73.90	50.0	62.30
UMM reasoning-based generation methods							
T2I-R1 (Jiang et al., 2025)‡	77.00	-	-	67.61	68.34	32.2	-
ReasonGen-R1 (Zhang et al., 2025)†	86.00	-	85.88	-	-	-	-
IRG (Huang et al., 2025)†	85.00	-	-	76.00	73.77	41.5	-
UniCorn (Fang et al., 2026)†	82.00	-	86.83	-	-	42.6	-
AD-Loop (Wu et al., 2026)†	86.00	-	-	-	-	-	-
Within-backbone comparisons							
BAGEL (Deng et al., 2025)	77.90	85.91	85.07	70.25	68.91	36.1†	35.22
BAGEL Self CoT (Deng et al., 2025)	79.00	85.07	83.84	67.92	68.33	32.4†	11.41
VSRL (ours)	88.92	87.76	85.20	71.90	72.38	43.7	41.52
vs. Self CoT	**+9.92**	**+2.69**	**+1.36**	**+3.98**	**+4.05**	**+11.3**	**+30.11**

4.1 Experimental Setup

Models and comparisons. We evaluate VSRL on two UMMs with different architectures, BAGEL and Cheers. On BAGEL, we use its native Self CoT as the main reasoning baseline. We first train the model to use VSRL through supervised fine-tuning and then further optimize it with StruFlow-GRPO. Section 4.3 analyzes these training stages separately. We further conduct a controlled comparison between VSRL and natural-language reasoning on Cheers, as detailed in Section 4.4.

Benchmarks and evaluation protocols. Generation evaluation covers five benchmarks: GenEval (Ghosh et al., 2023), DPG-Bench (Hu et al., 2024), TIIF-Bench (Wei et al., 2025), OneIG-Bench (Chang et al., 2025), and CVTG-2K (Tai et al., 2025), covering object composition, dense and complex instruction following, comprehensive generation performance, and visual text generation. We also evaluate multimodal understanding with MMBench (Liu et al., 2023), MME (Fu et al., 2023), MMMU (Yue et al., 2023), MathVista (Lu et al., 2023), and Pair-Jigsaw (Wang et al., 2025). We report GenEval and TIIF-Bench separately for their Short and Long versions; the main table uses the equally weighted mean of OneIG's five official dimensions and CVTG-2K Word Accuracy. Full generation configurations, result-source conventions, and benchmark-specific scoring details are provided in Appendix F.

4.2 Main Results

Overall generation performance. On BAGEL, VSRL outperforms both the base model and native Self CoT across five generation benchmarks (Table 1; Figure 1(a)). Compared with Self CoT, GenEval Short improves from 79.0 to 88.9, and the five-dimension OneIG-Bench mean improves

from 32.4 to 43.7, with consistent gains on DPG-Bench, TIIF-Bench, and CVTG-2K. These gains span object composition, dense and complex instruction following, comprehensive generation performance, and visual text generation. On Cheers, VSRL outperforms both natural-language CoT and the base model across all evaluated generation benchmarks, including GenEval and TIIF-Bench (Figure 1(b); Table 3). The natural-language CoT model and its training setup are detailed in Section 4.4.

Analysis. The magnitudes of improvement across benchmarks reveal a trend consistent with VSRL's design: compared with Self CoT, gains are larger on tasks requiring explicit organization of multiple scene elements, such as object composition and complex visual text generation, and more moderate on DPG-Bench. This pattern aligns with VSRL's explicit organization of object identities, attribute ownership, and object relations, especially when generation requires the simultaneous coordination of interdependent visual constraints. Detailed results and available component-level analyses for each benchmark are provided in Appendix A (Tables 6–10 and Figure 7).

Table 2: **Multimodal understanding across VSRL training stages.** Scoring scales are given in Appendix F.

Stage	MMBench	MME	MMMU	MathVista	Pair-Jigsaw
Cheers					
Base	77.93	1600.27	37.22	50.40	39.43
SFT	**78.01**	1522.61	36.78	**50.70**	**40.25**
BAGEL					
Base	84.44	2352.94	53.00	72.90	40.55
SFT	82.59	2295.66	49.10	**73.60**	39.40
SFT + StruFlow-GRPO	82.59	2281.59	49.70	67.30	39.65

Multimodal understanding. We further evaluate the effect of generation-oriented post-training on the model's original understanding capabilities (Table 2). On Cheers, VSRL improves MMBench, MathVista, and Pair-Jigsaw, while the other two metrics decline; on BAGEL, supervised fine-tuning improves MathVista with decreases on the remaining metrics. Overall, VSRL supervised fine-tuning preserves most of the original understanding capabilities while improving some tasks. Further StruFlow-GRPO optimization reduces some understanding metrics, reflecting a degree of capability trade-off in generation-oriented post-training.

4.3 Effect of StruFlow-GRPO and Joint Optimization

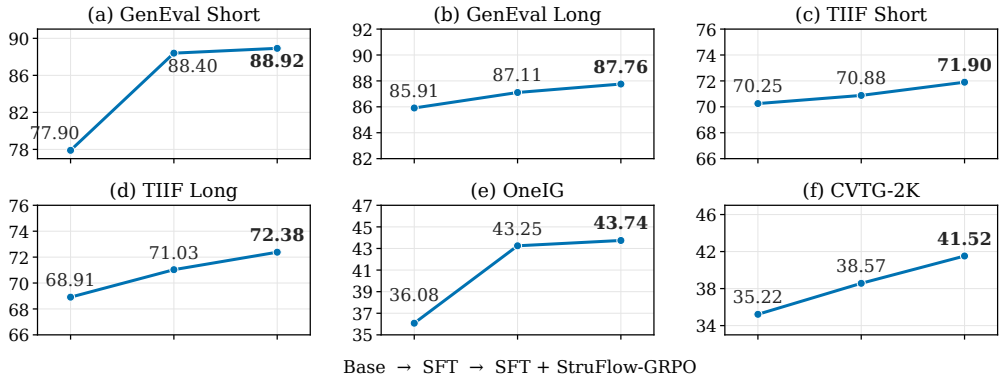


Figure 5: BAGEL results across VSRL training stages. Numerical results are reported in Table 5. CVTG-2K denotes Word Accuracy.

From structure supervision to generation feedback. StruFlow-GRPO further optimizes the VSRL checkpoint obtained after supervised fine-tuning using its own generated structure–image trajectories. As shown in Figure 5 (Table 5), after 200 StruFlow-GRPO training steps on BAGEL, GenEval Long improves from 87.11 to 87.76 and TIIF-Bench Long from 71.03 to 72.38, further improving object composition and complex instruction following. Compared with supervision from annotated scenes and images, StruFlow-GRPO adds feedback from the model's own generated trajectories, adapting optimization to the model's own generated structures.

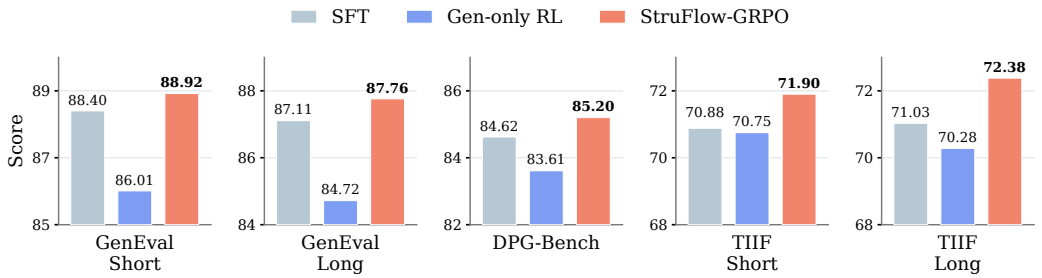


Figure 6: Gen-only RL and joint updates from the same BAGEL VSRL SFT checkpoint (Table 4).

Joint optimization matters. From the same BAGEL SFT checkpoint, we compare Gen-only RL and StruFlow-GRPO with identical rewards and a 200-step budget (Figure 6). Gen-only RL freezes the language module and updates only the image-generation module. It performs worse than SFT across the evaluated benchmarks, whereas StruFlow-GRPO updates both modules and consistently improves over SFT. The key difference is whether VSRL generation can adapt to reward feedback. These results show that joint optimization is more effective than updating the image-generation module alone, highlighting the importance of optimizing VSRL reasoning for improving final image generation.

4.4 Controlled Comparison of Reasoning Representations

Section 4.2 demonstrates VSRL's generation advantage over BAGEL's native Self CoT, but the two reasoning approaches differ in supervision sources and training histories. To isolate the effect of the reasoning representation, we conduct a controlled comparison on Cheers.

Constructing the Cheers natural-language control. Natural-language supervision is transcribed from the same VSRL annotations using fixed templates (Figure 8; Appendix C), preserving scene, object-attribute, and relation facts without adding or removing visual information. The two models use the same prompts, target images, initialization, and training settings. At test time, each autonomously generates its respective reasoning and then generates an image conditioned on that reasoning.

Comparison with natural-language CoT. As shown in Figure 1(b) and Table 3, VSRL consistently outperforms natural-language CoT across the evaluated benchmarks. On GenEval, VSRL improves the Short and Long scores from 80.0/81.5 to 84.3/83.0, with consistent gains also observed on DPG-Bench and TIIF-Bench. With the information conveyed by the training data and the training conditions held constant, structured reasoning still consistently improves generation performance, providing direct evidence that intermediate reasoning representations affect UMM image generation. Detailed results and available component-level analyses for each generation benchmark are provided in Appendix A.

5 Conclusion

In this work, we revisit how intermediate reasoning is represented for UMM image generation and introduce VSRL as a structured alternative to natural-language reasoning. We develop a VSRL data construction pipeline to support supervised training and StruFlow-GRPO to further optimize reasoning and generation with both structure and image feedback. Experiments on BAGEL and Cheers consistently demonstrate the advantage of VSRL over natural-language reasoning, while the controlled comparison on Cheers provides more direct evidence for the role of the reasoning representation itself. These findings highlight the importance of reasoning representation as a design dimension for unified multimodal generation: how reasoning is represented may be as important as how reasoning is performed.

AI USE STATEMENT

In this work, we used generative AI tools to assist with VSRL supervision data construction and manuscript translation and language polishing. We reviewed all AI-assisted outputs and take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We provide detailed information for reproducing our method and experiments throughout the paper and appendices. Appendix B describes the VSRL representation and supervision construction; Appendix C details the controlled natural-language CoT comparison; Appendix D provides supervised fine-tuning and inference configurations; Appendix E specifies the StruFlow-GRPO rewards, optimization objective, and training settings; and Appendix F documents the evaluation protocols and result sources.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning, 2023. URL https://arxiv.org/abs/2305.13301.

Black Forest Labs. Announcing Black Forest Labs, 2024. URL https://bfl.ai/blog/24-08-01-bfl.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models, 2024. URL https://arxiv.org/abs/2405.09818.

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, 2025. URL https://arxiv.org/abs/2506.07977.

Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, Le Xue, Caiming Xiong, and Ran Xu. BLIP3-o: A family of fully open unified multimodal models-architecture, training and dataset, 2025a. URL https://arxiv.org/abs/2505.09568.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-Pro: Unified multimodal understanding and generation with data and model scaling, 2025b. URL https://arxiv.org/abs/2501.17811.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining, 2025. URL https://arxiv.org/abs/2505.14683.

Chengqi Duan, Rongyao Fang, Yuqing Wang, Kun Wang, Linjiang Huang, Xingyu Zeng, Hongsheng Li, and Xihui Liu. GoT-R1: Unleashing reasoning capability of autoregressive visual generation with reinforcement learning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/c0c165157df7e3082f9c6d70d3a4b6e9-Abstract-Conference.html.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. DPOK: Reinforcement learning for fine-tuning text-to-image diffusion models, 2023. URL https://arxiv.org/abs/2305.16381.

Zhen Fang, Ruiyan Han, Xinyu Sun, Yuchen Ma, Ziheng Wang, Yu Zeng, Zehui Chen, Lin Chen, Wenxuan Huang, Weijie Xu, Yi Cao, and Feng Zhao. UniCorn: Towards self-improving unified multimodal models through self-generated supervision. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13638–13669, 2026. doi: 10.18653/v1/2026.acl-long.621. URL https://aclanthology.org/2026.acl-long.621/.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. MME: A comprehensive evaluation benchmark for multimodal large language models, 2023. URL https://arxiv.org/abs/2306.13394.

Yu Gao, Lixue Gong, Qiushan Guo, Xiaoxia Hou, Zhichao Lai, Fanshi Li, Liang Li, Xiaochen Lian, Chao Liao, Liyang Liu, Wei Liu, Yichun Shi, Shiqi Sun, Yu Tian, Zhi Tian, Peng Wang, Rui Wang, Xuanda Wang, Xun Wang, Ye Wang, Guofeng Wu, Jie Wu, Xin Xia, Xuefeng Xiao, Zhonghua Zhai, Xinyu Zhang, Qi Zhang, Yuwei Zhang, Shijia Zhao, Jianchao Yang, and Weilin Huang. Seedream 3.0 technical report, 2025. URL https://arxiv.org/abs/2504.11346.

Dhruba Ghosh, Hanna Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment, 2023. URL https://arxiv.org/abs/2310.11513.

Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. ELLA: Equip diffusion models with LLM for enhanced semantic alignment, 2024. URL https://arxiv.org/abs/2403.05135.

Wenxuan Huang, Shuang Chen, Zheyong Xie, Shaosheng Cao, Shixiang Tang, Yufan Shen, Qingyu Yin, Wenbo Hu, Xiaoman Wang, Yuntian Tang, Junbo Qiao, Yue Guo, Yao Hu, Zhenfei Yin, Philip Torr, Yu Cheng, Wanli Ouyang, and Shaohui Lin. Interleaving reasoning for better text-to-image generation, 2025. URL https://arxiv.org/abs/2509.06945.

Dongzhi Jiang, Ziyu Guo, Renrui Zhang, Zhuofan Zong, Hao Li, Le Zhuo, Shilin Yan, Pheng-Ann Heng, and Hongsheng Li. T2I-R1: Reinforcing image generation with collaborative semantic-level and token-level CoT, 2025. URL https://arxiv.org/abs/2505.00703.

Weiyang Jin, Yuwei Niu, Jiaqi Liao, Chengqi Duan, Aoxue Li, Shenghua Gao, and Xihui Liu. SRUM: Fine-grained self-rewarding for unified multimodal models, 2025. URL https://arxiv.org/abs/2510.12784.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-Pic: An open dataset of user preferences for text-to-image generation, 2023. URL https://arxiv.org/abs/2305.01569.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-GRPO: Training flow matching models via online RL, 2025. URL https://arxiv.org/abs/2505.05470.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, 2024. URL https://arxiv.org/abs/2303.05499.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. MMBench: Is your multi-modal model an all-around player?, 2023. URL https://arxiv.org/abs/2307.06281.

Zhiheng Liu, Weiming Ren, Xiaoke Huang, Shoufa Chen, Tianhong Li, Mengzhao Chen, Yatai Ji, Sen He, Jonas Schult, Belinda Zeng, Tao Xiang, Wenhu Chen, Ping Luo, Luke Zettlemoyer, and Yuren Cong. Tuna-2: Pixel embeddings beat vision encoders for multimodal understanding and generation. *arXiv preprint arXiv:2604.24763*, 2026. URL https://arxiv.org/abs/2604.24763.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. MathVista: Evaluating mathematical reasoning of foundation models in visual contexts, 2023. URL https://arxiv.org/abs/2310.02255.

Yiyang Ma, Xingchao Liu, Xiaokang Chen, Wen Liu, Chengyue Wu, Zhiyu Wu, Zizheng Pan, Zhenda Xie, Haowei Zhang, Xingkai Yu, Liang Zhao, Yisong Wang, Jiaying Liu, and Chong Ruan. JanusFlow: Harmonizing autoregression and rectified flow for unified multimodal understanding and generation, 2024. URL https://arxiv.org/abs/2411.07975.

Yasumasa Onoe, Sunayana Rane, Zachary Berger, Yonatan Bitton, Jaemin Cho, Roopal Garg, Alexander Ku, Zarana Parekh, Jordi Pont-Tuset, Garrett Tanzer, Su Wang, and Jason Baldridge. DOCCI: Descriptions of connected and contrasting images, 2024. URL https://arxiv.org/abs/2404.19753.

OpenAI. DALL-E 3, 2023. URL https://openai.com/index/dall-e-3/.

OpenAI. Introducing our latest image generation model in the API, 2025. URL https://openai.com/index/image-generation-api/.

Luozheng Qin, Jia Gong, Yuqing Sun, Tianjiao Li, Mengping Yang, Xiaomeng Yang, Chao Qu, Zhiyu Tan, and Hao Li. Uni-cot: Towards unified chain-of-thought reasoning across text and vision, 2025. URL https://arxiv.org/abs/2508.05606.

Qwen Team. Qwen-Image technical report, 2025. URL https://arxiv.org/abs/2508.02324.

Qwen Team. Qwen3.8-27B, August 2026. URL https://huggingface.co/Qwen/Qwen3.8-27B.

Stability AI. Introducing Stable Diffusion 3.5, 2024. URL https://stability.ai/news/introducing-stable-diffusion-3-5.

Ying Tai, Nikai Du, Rui Xie, Zhennan Chen, Qian Wang, Zhengkai Jiang, Kai Zhang, and Jian Yang. Investigating text insulation and attention mechanisms for complex visual text generation, 2025. URL https://arxiv.org/abs/2503.23461.

Changyao Tian, Danni Yang, Guanzhou Chen, Erfei Cui, Zhaokai Wang, et al. InternVL-U: Democratizing unified multimodal models for understanding, reasoning, generation and editing. *arXiv preprint arXiv:2603.09877*, 2026. URL https://arxiv.org/abs/2603.09877.

Yuxiang Tuo, Yifeng Geng, and Liefeng Bo. AnyText2: Visual text generation and editing with customizable attributes, 2024. URL https://arxiv.org/abs/2411.15245.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, Yingli Zhao, Yulong Ao, Xuebin Min, Tao Li, Boya Wu, Bo Zhao, Bowen Zhang, Liangdong Wang, Guang Liu, Zheqi He, Xi Yang, Jingjing Liu, Yonghua Lin, Tiejun Huang, and Zhongyuan Wang. Emu3: Next-token prediction is all you need, 2024. URL https://arxiv.org/abs/2409.18869.

Zifu Wang, Junyi Zhu, Bo Tang, Zhiyu Li, Feiyu Xiong, Jiaqian Yu, and Matthew B. Blaschko. Jigsaw-R1: A study of rule-based visual reinforcement learning with jigsaw puzzles, 2025. URL https://arxiv.org/abs/2505.23590.

Xinyu Wei, Jinrui Zhang, Zeqing Wang, Hongyang Wei, Zhen Guo, Bairui Li, and Lei Zhang. TIIF-Bench: How does your T2I model follow your instructions?, 2025. URL https://arxiv.org/abs/2506.02161.

Shengqiong Wu, Bobo Li, Xinkai Wang, Xiangtai Li, Lei Cui, Furu Wei, Shuicheng Yan, Hao Fei, and Tat-Seng Chua. Synergizing understanding and generation with interleaved analyzing-drafting thinking. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/741ff9e19bc28050b4d815726ec792c9-Abstract-Conference.html.

Ji Xie, Trevor Darrell, Luke Zettlemoyer, and Xudong Wang. Reconstruction alignment improves unified multimodal models. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/c34ddd05eb089991f06f3c5dc36836e0-Abstract-Conference.html.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation, 2024. URL https://arxiv.org/abs/2408.12528.

Jinheng Xie, Zhenheng Yang, and Mike Zheng Shou. Show-o2: Improved native unified multimodal models, 2025. URL https://arxiv.org/abs/2506.15564.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, and Ping Luo. DanceGRPO: Unleashing GRPO on visual generation, 2025. URL https://arxiv.org/abs/2505.07818.

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, 2025. URL https://arxiv.org/abs/2508.09987.

Sen Ye, Mengde Xu, Shuyang Gu, Di He, Liwei Wang, and Han Hu. Understanding vs. generation: Navigating optimization dilemma in multimodal models. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2602.15772.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI, 2023. URL https://arxiv.org/abs/2311.16502.

Yichen Zhang, Da Peng, Zonghao Guo, Zijian Zhang, Xuesong Yang, Tong Sun, Shichu Sun, Yidan Zhang, Yanghao Li, Haiyan Zhao, Wang Xu, Qi Shi, Yangang Sun, Chi Chen, Shuo Wang, Yukun Yan, Xu Han, Qiang Ma, Wei Ke, Liang Wang, Zhiyuan Liu, and Maosong Sun. Cheers: Decoupling patch details from semantic representations enables unified multimodal comprehension and generation, 2026. URL https://arxiv.org/abs/2603.12793.

Yu Zhang, Yunqi Li, Yifan Yang, Rui Wang, Yuqing Yang, Dai Qi, Jianmin Bao, Dongdong Chen, Chong Luo, and Lili Qiu. ReasonGen-R1: CoT for autoregressive image generation models through SFT and RL, 2025. URL https://arxiv.org/abs/2505.24875.

Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model, 2024. URL https://arxiv.org/abs/2408.11039.

A Supplementary Experimental Results

Appendix A provides detailed experimental results corresponding to Sections 4.2–4.4, including the controlled comparison of reasoning representations, training-stage and optimization-scope results, and benchmark-specific results for GenEval, OneIG-Bench, TIIF-Bench, DPG-Bench, and CVTG-2K.

A.1 Reasoning Representation Comparison

Table 3: **Comparison of VSRL and natural-language CoT.** Both models start from the same Cheers backbone and use the same number of samples under the same training settings. All scores are on a 100-point scale; higher is better.

Method	GenEval		DPG	TIIF	
	S	L		S	L
Cheers	78.00	73.27	83.48	64.81	64.38
Natural-language CoT	80.00	81.52	83.43	64.00	66.19
VSRL (ours)	**84.30**	**83.00**	**84.47**	**67.13**	**66.65**

A.2 Training Stages and Optimization Scope

Table 4: **Results across VSRL training stages and optimization scopes.** The two RL settings on BAGEL share the same SFT initialization, rewards, and budget of 200 steps.

VSRL training stage	GenEval		TIIF		DPG
	S	L	S	L	
Cheers					
Base	78.00†	73.27	64.81	64.38	83.48†
SFT	84.30	83.00	67.13	66.65	84.47
BAGEL					
Base	77.90	85.91	70.25	68.91	85.07
SFT	88.40	87.11	70.88	71.03	84.62
SFT + Gen-only RL	86.01	84.72	70.75	70.28	83.61
SFT + StruFlow-GRPO	88.92	87.76	71.90	72.38	85.20

Table 5: BAGEL results across VSRL training stages (Figure 5). All scores are on a 0–100 scale. †: published result.

VSRL training stage	GenEval		TIIF		DPG	OneIG	CVTG-2K
	S	L	S	L			Word Acc.
Base	77.90	85.91	70.25	68.91	85.07	36.08†	35.22
SFT	88.40	87.11	70.88	71.03	84.62	43.25	38.57
SFT + StruFlow-GRPO	88.92	87.76	71.90	72.38	85.20	43.74	41.52

A.3 GENEVAL

Table 6: GenEval Short and Long category results. Scores are on a 100-point scale. All category results are evaluated in this work.

Method	Single	Two	Counting	Colors	Position	Color attribution	Overall
BAGEL – Short							
BAGEL	99.06	92.68	77.81	89.36	49.00	59.50	77.90
Self CoT	99.00	92.00	75.00	89.00	54.00	63.00	79.00
VSRL — SFT (ours)	99.38	95.96	80.00	92.82	83.00	79.00	88.36
VSRL — SFT + StruFlow-GRPO (ours)	99.38	95.71	83.12	93.09	82.00	80.25	88.92
Cheers – Short							
Cheers	98.00	92.00	65.00	86.00	63.00	65.00	78.00
Natural-language CoT	99.38	88.13	66.25	82.18	80.00	64.25	80.03
VSRL (ours)	98.44	92.42	70.94	86.97	82.75	74.25	84.30
BAGEL – Long							
BAGEL	98.12	94.19	82.81	92.55	74.75	73.00	85.91
Self CoT	98.75	97.22	78.44	92.29	72.25	71.50	85.07
VSRL — SFT (ours)	98.44	95.45	69.06	94.68	80.50	84.50	87.11
SFT + Gen-only RL	97.19	94.44	72.50	90.43	78.75	75.00	84.72
VSRL — SFT + StruFlow-GRPO (ours)	98.44	96.72	71.25	94.15	81.25	84.75	87.76
Cheers – Long							
Cheers	92.50	85.35	62.19	80.59	58.25	60.75	73.27
Natural-language CoT	97.19	92.68	70.94	84.57	74.00	69.75	81.52
VSRL (ours)	99.69	92.68	77.50	85.64	70.25	72.25	83.00

A.4 ONEIG

Table 7: OneIG results on the five official dimensions and their equal-weight mean, reported on a 100-point scale. † denotes published results.

Method	Alignment	Text	Reasoning	Style	Diversity	Mean
BAGEL						
BAGEL†	76.90	24.40	17.30	36.70	25.10	36.08
Self CoT†	79.30	2.00	20.60	39.00	20.90	32.36
VSRL — SFT (ours)	81.46	46.21	20.38	42.86	25.33	43.25
VSRL — SFT + StruFlow-GRPO (ours)	82.56	48.13	20.79	42.69	24.54	43.74

A.5 TIIF-BENCH

Table 8: Complete TIIF-Bench summary results. Short and Long are reported separately using the Qwen2.5-VL-72B evaluator.

Backbone	Method	Short	Long
BAGEL	Base	70.25	68.91
	Self CoT	67.92	68.33
	VSRL — SFT (ours)	70.88	71.03
	SFT + Gen-only RL	70.75	70.28
	VSRL — SFT + StruFlow-GRPO (ours)	71.90	72.38
Cheers	Base	64.81	64.38
	Natural-language CoT	64.00	66.19
	VSRL (ours)	67.13	66.65

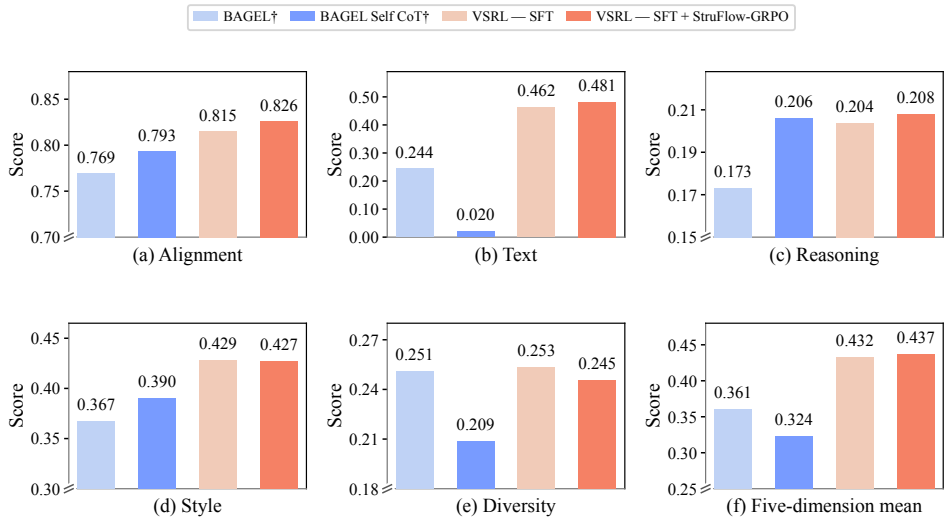


Figure 7: OneIG performance on the five evaluation dimensions and their equal-weight mean. Scores are shown on the original 0–1 scale; Table 7 reports the corresponding 0–100 values.

A.6 DPG-Bench

Table 9: DPG-Bench category and overall results. All category results are evaluated in this work.

Method	Entity	Attribute	Relation	Global	Other	Overall
BAGEL						
BAGEL	90.37	91.29	90.82	88.94	88.67	85.07
Self CoT	90.23	88.43	92.78	80.40	81.00	83.84
VSRL — SFT (ours)	91.05	89.00	92.96	83.89	84.00	84.62
VSRL — SFT + StruFlow-GRPO (ours)	90.02	91.15	90.75	88.00	91.72	85.20
Cheers						
Cheers	90.24	89.42	89.71	90.84	87.37	83.48
Natural-language CoT	90.13	88.45	93.21	79.03	78.00	83.43
VSRL (ours)	90.18	89.10	93.60	83.28	80.80	84.47

A.7 CVTG-2K

Table 10: CVTG-2K results on all five metrics, retaining their original scales. Higher is better.

Method	Word Acc.	NED	CLIPScore	VQAScore	Aesthetic
BAGEL					
BAGEL	0.3522	0.6517	0.7774	0.8831	5.3006
Self CoT	0.1141	0.3970	0.7159	0.7166	5.1899
VSRL — SFT (ours)	0.3857	0.6821	0.7970	0.8739	4.9663
VSRL — SFT + StruFlow-GRPO (ours)	0.4152	0.7051	0.7999	0.8815	5.0641

B VSRL Representation and Supervision Construction

B.1 Scenes, Objects, and Relations

VSRL consists of `global_scene`, `objects`, and `relations`. The `global_scene` field may be empty; individual background objects can still be represented in `objects`. Objects have distinct

identities, attributes belong to specific objects, and relations are expressed through object references. Here, attributes denotes the collection of object properties rather than a separate visual property or a required serialized container. Attributes and relations should maintain consistent references when multiple objects belong to the same category.

Table 11: **VSRL components and field semantics.**

Component	Content	Role
`global_scene`	Environment, background, composition, viewpoint	Describes visual information shared across the scene.
`objects`	Object identities and attributes	Distinguishes entities and associates category, count, appearance, state, and other information with specific objects.
Object attributes	Color, material, state, text content, etc.	Object fields encode these properties; coordinate configurations differ across backbones.
`relations`	Subject reference, relation, object reference	Expresses spatial relations and interactions between objects.

B.2 BAGEL Reasoning Example

The following example gives the full prompt and reasoning outputs for the three-bench case in Figure 2 (third column). The two models receive the same prompt; each image is conditioned on its model's own reasoning. VSRL represents the benches as three distinct objects and links adjacent benches through explicit relations. JSON whitespace is compacted for display.

Prompt. Three identical wooden benches are arranged in a straight line on a grassy field. Each bench is made of smooth, weathered wood with a natural brown tone. The benches are evenly spaced, approximately two meters apart. The grass beneath them is lush and green, with a few scattered patches of sunlight filtering through nearby trees. The scene is calm and serene, with no other objects or people in view. The realistic style captures the texture of the wood and the softness of the grass, creating a photographic representation of a quiet outdoor setting.

BAGEL native Self CoT. `<think>`

The model should generate an image showing three identical wooden benches placed in a straight line on a grassy field, with equal spacing between them, emphasizing the arrangement and natural setting.

Here's the expanded prompt: Three identical wooden benches made of smooth, weathered wood with a natural brown tone are arranged in a straight line on a lush, green grassy field, evenly spaced approximately two meters apart. The scene is calm and serene, with no other objects or people in view, emphasizing the simplicity and harmony of the arrangement. The grass beneath the benches is vibrant and well-maintained, with a few scattered patches of sunlight filtering through nearby trees, creating a soft, natural lighting effect. The overall atmosphere is tranquil and realistic, with photographic detail capturing the texture of the wood and the softness of the grass, evoking a quiet outdoor setting. Ultra-realistic style with sharp focus and natural color tones.

`</think>`

BAGEL with VSRL (SFT + StruFlow-GRPO). `<Structured_Vision>`


```
{
  "global_scene": {
    "style": "realistic photographic representation",
    "background": {"setting": "grassy outdoor field", "density": "moderate",
    "layout": "lush green grass with scattered patches of sunlight filtering
    through nearby trees"},
    "composition": "three identical benches arranged in a straight line receding
    from left to right",
    "lighting": "soft natural daylight with dappled sunlight"},
  "objects": [
    {"id": "obj_1", "name": "first wooden bench", "position": "left midground",
```

```
    "material": "wood", "color": "natural brown weathered wood", "texture":
    "smooth weathered wood",
    "state": "empty", "bbox": [102, 436, 367, 770]},
   {"id": "obj_2", "name": "second wooden bench", "position": "center
   midground",
    "material": "wood", "color": "natural brown weathered wood", "texture":
    "smooth weathered wood",
    "state": "empty", "bbox": [415, 438, 662, 782]},
   {"id": "obj_3", "name": "third wooden bench", "position": "right midground",
    "material": "wood", "color": "natural brown weathered wood", "texture":
    "smooth weathered wood",
    "state": "empty", "bbox": [708, 443, 938, 800]}
 ],
 "relations": [
   {"subject": "obj_1", "subject_name": "first wooden bench",
    "spatial_relation": "evenly spaced from", "object": "obj_2", "object_name":
    "second wooden bench"},
   {"subject": "obj_2", "subject_name": "second wooden bench",
    "spatial_relation": "evenly spaced from", "object": "obj_3", "object_name":
    "third wooden bench"}
 ]
}
</Structured_Vision>
```


B.3 Data Sources and T2I Conversion

VSRL supervision comes from Echo-4o (Ye et al., 2025), BLIP3-o (Chen et al., 2025a), DOCCI (Onoe et al., 2024), and AnyText2 (Tuo et al., 2024). BAGEL uses 278,970 structure samples, and Cheers uses 5,000 structure samples derived from Echo-4o.

Echo-4o. We curate samples from sources covering instruction following, surreal scenes, and multiple reference images. For samples that are already T2I, we retain the text prompt and output image. For samples with multiple reference images, we retain the original task's output as the target image and deterministically convert prompts that already contain sufficient textual descriptions: we remove numbered image markers and their associated reference phrases, such as `from image_1` and `shown in image_2`, while preserving the original object descriptions and generation constraints without adding visual content. Candidate filtering requires at least 50 BAGEL tokens after image markers are removed, at least 40 tokens after conversion, and no remaining numbered image references. Short instructions that clearly depend on reference images and samples that fail subsequent validation are excluded from the derived T2I data.

For samples with multiple reference images that already have visual structure annotations, we reuse the existing structure after conversion and rerun training-sample validation against the new prompt, rather than uniformly invoking the annotation model again. Reference-image paths are removed from the training input; the original prompt, task type, and reference-image information are retained in provenance records. A derived T2I sample and its original multi-reference-image version are not both included in training.

AnyText2. We first organize the original image, caption, and text annotations into a standalone T2I prompt, then generate VSRL. The conversion model reads both the image and textual evidence, making the required text content and its visual carrier explicit in the prompt. Selected text preserves the spelling, capitalization, numbers, and punctuation in the image; truncated or misrecognized text annotations are corrected only when the image provides clear support, and the source of the correction is recorded. Text segments belonging to the same continuous phrase may be merged, but illegible content is neither translated nor guessed. Clear layout information is expressed in natural language, without directly inserting the original text-region polygon coordinates into the T2I prompt.

Before the converted result enters the structural annotation pipeline, we check whether the prompt–image pair is usable and whether the selected text is fully preserved in the prompt. Text annotations determine the strings to be generated and their carriers, whereas object bounding boxes are localized later by Grounding DINO; the two serve different purposes. Image–text samples from BLIP3-o and

DOCCI are likewise organized into text prompts and target images and processed through the same structural annotation and validation pipeline.

B.4 Visual Annotation, Validation, and Repair

Structural annotation. Qwen2.5-VL-32B-Instruct (Bai et al., 2025) receives both the prompt and target image and generates structural supervision according to the VSRL schema. The prompt provides the generation requirements, and the target image provides the corresponding visible scene information. Annotation assigns unique identities to the main objects and writes their attributes and inter-object relations into the corresponding fields, without adding visual facts that cannot be confirmed from the target image.

Quality checks. Programmatic checks validate JSON parseability, field types, object identities, and relation references; Qwen3.8-27B (Qwen Team, 2026) is the primary model used to check semantic consistency between the structure, prompt, and target image.

Repair. Qwen3.8-27B (Qwen Team, 2026) regenerates structures that fail validation, followed by another round of programmatic and visual-semantic checks. At inference time, VSRL is generated from the user prompt alone.

B.5 Object Localization and Bounding Boxes

After semantic structure annotation, we use Grounding DINO to localize the annotated objects in the target image, associate detection boxes with object identities, and write them into the object attributes of the BAGEL version. Semantic object attributes come from the vision-language model, whereas bbox comes from object localization in the target image. Bounding boxes use the normalized $[x_{\min}, y_{\min}, x_{\max}, y_{\max}]$ representation, with coordinates in $[0, 999]$. Localization adds only object-level spatial information and does not change the definition of object identities or relation references in VSRL. Localization queries are derived from existing object descriptions, and detection results are matched to objects and written back into the structure, rather than redefining the object list from detector outputs. Coordinate checks require each bounding box to contain four integers within the valid range and to satisfy $x_{\min} < x_{\max}$ and $y_{\min} < y_{\max}$. Cheers structural supervision omits bounding boxes.

C Natural-language CoT Construction

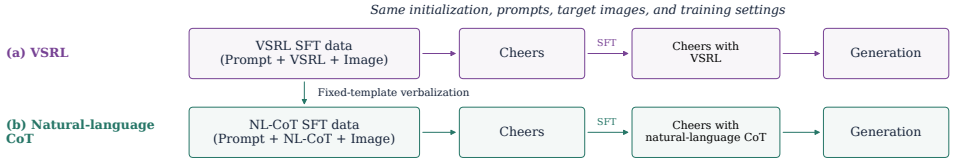


Figure 8: **Controlled comparison of reasoning representations on Cheers.** Natural-language supervision is derived from VSRL annotations through fixed-template verbalization. Both models use the same prompts, target images, initialization, and training settings.

C.1 Control Construction

We construct natural-language CoT from the same Cheers VSRL annotations (Figure 8). The conversion traverses the objects, their attributes, and inter-object relations in the Cheers annotations, preserving their scene facts without adding visual details. Structured fields and nesting are rewritten as natural-language descriptions; object references are rewritten as corresponding entity references so that both endpoints of each relation still refer to the original objects. The two forms of supervision describe the same target scene and differ in whether structured fields and reference relations are retained as an explicit reasoning format.

The two training setups use the same prompts, target images, initialization, replay data, training steps, and optimization settings. At test time, the model generates natural-language CoT or VSRL from the prompt and then generates an image based on its own reasoning. This comparison therefore encompasses both learning the representation and using it for image generation.

Table 12: **Cheers training conditions and sequence handling.** Lengths after image-token expansion are computed using the training tokenizer, template, and model expansion rules.

Setting	VSRL	Natural-language CoT
Reasoning supervision examples	5,000	5,000
Generation / understanding replay	2,500 / 2,500	2,500 / 2,500
Total training examples	10,000	10,000
Global batch size	128	128
Learning rate	2×10^{-5}	2×10^{-5}
Optimizer	AdamW	AdamW
Schedule / warmup	Constant / 2%	Constant / 2%
Completed optimizer updates	78	78
Incomplete final batch	Dropped	Dropped
Text-template token limit	8192	8192
Token limit after image expansion	4096	4096
Longest expanded reasoning example	1912	1406
Longest expanded example including replay	2098	2098
Examples truncated by the active length limits	0 / 10,000	0 / 10,000

C.2 Paired Supervision Example

The following example illustrates the conversion for a single training pair. The natural-language version preserves both objects, their attributes, and their relation. The target image and original prompt are shared; the backbone-specific instructions and delimiters are given in Appendix D.2.

Shared prompt. a photo of a blue Lion and a pink Oyster

VSRL supervision.

```
{
  "global_scene": {},
  "objects": [
    {"id": "obj_1", "name": "lion", "position": "left", "color": "blue",
     "texture": "furry", "pose": "sitting"},
    {"id": "obj_2", "name": "oyster", "position": "right", "color": "pink",
     "state": "open", "material": "shiny"}
  ],
  "relations": [
    {"subject": "obj_1", "predicate": "next_to", "object": "obj_2"}
  ]
}
```

Natural-language CoT supervision. The target image should show lion and oyster. The image should include lion in the left, with color blue, texture furry, and pose sitting. The image should include oyster in the right, with color pink, material shiny, and state open. Spatially, lion is next to oyster. The generated image should make the described subjects, visual attributes, and spatial arrangement clear.

D Supervised Fine-tuning and Inference Configurations

Compute. Supervised fine-tuning and evaluation used 2×8 NVIDIA A800 GPUs, and reinforcement learning used 1×8 NVIDIA H100 80GB GPUs.

Both backbones use $\lambda_{\text{FM}} = 1$. Cheers assigns unit weights to image and text losses (`alpha = 1.0`); BAGEL likewise uses `mse_weight = 1.0` and `ce_weight = 1.0`. Each backbone retains its native loss reduction.

D.1 Structural Supervision and Replay

VSRL triplets provide next-token prediction supervision for structure tokens and flow-matching supervision for images. Native image-generation and multimodal-understanding samples are mixed into training as replay data, retaining their original formats and objectives to mitigate the effect of VSRL fine-tuning on existing capabilities.

The relative sampling weights of VSRL, generation replay, and understanding replay are $2:1:1$ for Cheers and $4:1:3$ for BAGEL. These weights specify relative sampling proportions during training and do not equal the sizes of the corresponding data files.

Table 13: **Supervised fine-tuning configurations.**

Setting	Cheers	BAGEL
Structure corpus size	5,000	278,970
Generation replay samples	2,500	140,000
Understanding replay samples	2,500	250,000
Structure:generation:understanding sampling weights	2:1:1	4:1:3
Object attributes include bbox	No	Yes
Global batch size	128	128
Learning rate	2×10^{-5}	2×10^{-5}
VAE parameters	Frozen	Frozen
Training sequence limits	8192 / 4096	4096 (structure)
Optimizer updates	78	4,359
Optimizer	AdamW	AdamW
Learning-rate schedule / warmup	Constant / 2%	Constant / 5 steps

For Cheers, the text template permits up to 8192 tokens, and the model limits the sequence to 4096 tokens after image-token expansion. Neither limit truncates any example in either 10,000-example training dataset. Both runs complete 78 optimizer updates with a global batch size of 128 and the incomplete final batch dropped.

D.2 Backbone Prompts and Structure Formats

The following instructions specify how each backbone is prompted to reason before image generation.

BAGEL VSRL. In training and inference, the following instruction is appended to the original prompt, separated by a blank line:

```
First generate a Structure CoT describing the target image.
Then generate the image by following the original prompt and
using the Structure CoT as guidance.  Reasonably complete
unspecified visual details to improve image quality.
```

The reasoning is enclosed by `<Structured_Vision>` and `</Structured_Vision>`. At inference, the opening tag is supplied as a fixed decoding prefix. The model generates the remaining reasoning and then the image through BAGEL's native `<|vision_start|>` and `<|vision_end|>` transition.

BAGEL native Self CoT. BAGEL's original planning instruction precedes the original prompt as a separate text segment:

```
You should first think about the planning process in the
mind and then generate the image.
The planning process is enclosed within <think> </think>
tags, i.e.  <think> planning process here </think> image
here
```

The generated reasoning uses <think> and </think> tags and conditions subsequent image generation.

Cheers. The VSRL instruction is:

```
First generate the complete structured visual information of
the target image for the prompt, then generate the image
according to this structure.
```

The natural-language CoT instruction is:

```
First describe the target image in natural language, then
generate the image according to this description.
```

Each instruction is followed by a blank line and `Prompt:` with the original prompt. Both reasoning formats use <Structured_Vision> and </Structured_Vision> tags; the natural-language model places prose between them. In training, the reasoning is followed by the target image using the native <im_start><image_gen><im_end> marker. At inference, the opening reasoning tag is supplied as a fixed decoding prefix, and the model generates its reasoning before the image.

Structure fields and generation. Both backbones use the top-level fields `global_scene`, `objects`, and `relations`, with attributes stored within objects. Bounding boxes are included only in BAGEL. After reasoning, each model directly generates one image conditioned on the prompt and its generated reasoning, without filtering or resampling based on structure checks.

E StruFlow-GRPO Rewards and Optimization Objective

E.1 Trajectory Rewards

For each prompt P, the model samples $G = 8$ structure–image trajectories. Image rewards take the original prompt and generated image as input, whereas structural rewards evaluate the VSRL generated in that trajectory. Training uses

$$R_{\text{gen}}(o_i) = R_{\text{comp}}(\hat{I}_i; P) + 0.10 R_{\text{pref}}(\hat{I}_i; P), \tag{4}$$

$$R_{\text{str}}(o_i) = 0.05 R_{\text{json}} + 0.05 R_{\text{tag}} + 0.10 R_{\text{cov}} + 0.05 R_{\text{rich}} + 0.05 R_{\text{len}}, \tag{5}$$

$$R(o_i) = R_{\text{gen}}(o_i) + R_{\text{str}}(o_i). \tag{6}$$

Here, R_{comp} evaluates whether the generated image satisfies requested objects, counts, colors, and positions; R_{pref} uses PickScore to provide image–text preference feedback. The structural reward contains five programmatic terms:

- R_{json}: JSON parseability;
- R_{tag}: presence of VSRL boundary-tag strings;
- R_{cov}: coverage of prompt category keywords;
- R_{rich}: a saturating richness score based on the number of structure fields;
- R_{len}: structure length regularization.

Table 14: Reward definitions, output ranges, and weights used in StruFlow-GRPO training.

Term	Rule	Range	Weight
R_{comp}	Graded detector-based compositional score	$[0, 1]$	1.00
R_{pref}	$e^s \langle \bar{e}_P, \bar{e}_I \rangle / 26$, no clipping	$[-e^s/26, e^s/26]$	0.10
R_{json}	Parser returns a mapping, including relaxed recovery	$\{0, 1\}$	0.05
R_{tag}	Both boundary-tag strings occur in the output	$\{0, 1\}$	0.05
R_{cov}	Required-class substring coverage; richness if no classes	$[0, 1]$	0.10
R_{rich}	Saturating object/field/relation statistic, defined below	$[0, 1]$	0.05
R_{len}	Piecewise-linear score with soft/hard limits	$[0, 1]$	0.05

Image feedback. R_{comp} is `geneval_graded` from `geneval_reward.py`. The default evaluator uses Mask2Former (`facebook/mask2former-swin-small-coco-instance`) for objects and CLIP ViT-L/14 for colors. For each required object group, it averages the detected-count fraction (capped at one) and applicable color and position scores, then averages across groups; a violated exclusion constraint sets the score to zero. Missing metadata also yields zero. Detection thresholds are 0.3, or 0.9 for counting prompts; the relative-position margin is 0.1. R_{comp} therefore provides a graded compositional training signal, while benchmark evaluation follows the standard GenEval criterion. PickScore uses L2-normalized text/image embeddings $\bar{e}_P, \bar{e}_I$ and the model's learned `logit_scale` s. Division by 26 rescales the score but does not constrain it to $[0, 1]$; text is truncated to 77 tokens.

Structure checks. The JSON rule extracts a mapping, tolerating trailing commas and recoverable bracket-balanced prefixes. The tag rule returns a positive score when both VSRL boundary tags are present. Coverage compares each required class from the prompt metadata with a lowercased text pool using either the full class name or its last word as a substring. The pool contains object `name/id/shape/material` strings and string-valued global-scene fields (including one nested level). Coverage is the fraction of matched required entries, not factual or relational correctness. If the required list is empty, it falls back to richness; unparseable structures with a nonempty list score zero.

Richness and length. Let N be the object count, F the number of populated object fields among `name, position, color, shape, size, orientation, material, count, state`, Q the relation count, and H the number of populated `style, background, lighting, composition` global fields. An unparseable structure or $N = 0$ gives $R_{\text{rich}} = 0$; otherwise,

$$\begin{aligned} R_{\text{rich}} = {} & 0.3 \min(N/2, 1) + 0.3 \min(F/(5N), 1) \\ & + 0.2 \min(Q, 1) + 0.2 \min(H/4, 1). \end{aligned} \tag{7}$$

The implemented length score for sampled reasoning length n is

$$R_{\text{len}}(n) = \begin{cases} 1, & n \le a, \\ (b - n)/\max(b - a, 1), & a < n < b, \\ 0, & n \ge b. \end{cases} \tag{8}$$

The default soft and hard length limits are $a = 384$ and $b = 640$, respectively. If the sampled length is unavailable, $R_{\text{len}} = 1$.

E.2 Image-Policy Objective and Joint Updates

We first compute the group-relative advantage:

$$A_i = \frac{R(o_i) - \mu_P}{\sigma_P + \delta}, \tag{9}$$

where μ_P and σ_P are the mean and standard deviation of rewards within the trajectory group for the same prompt.

We compute the policy ratio using log transition probabilities averaged over image latent dimensions:

$$\bar{\ell}_{\theta,ik} = \frac{1}{D} \sum_{d=1}^{D} \log p_\theta \left(x_{i,k-1}^{(d)} \mid x_{i,k}, P, \hat{S}_i \right), \tag{10}$$

where D is the number of elements included in the image latent transition probability calculation. The corresponding probability ratio is

$$\rho_{ik} = \exp\left(\bar{\ell}_{\theta,ik} - \bar{\ell}_{\text{old},ik} \right). \tag{11}$$

We then use the Flow-GRPO clipped surrogate objective:

$$J_{\text{Flow-GRPO}} = \mathbb{E}_{i,k} \left[\min\left(\rho_{ik} A_i,\ \text{clip}(\rho_{ik}, 1 - \epsilon, 1 + \epsilon) A_i \right) \right]. \tag{12}$$

During training, image generation is replayed conditioned on the prompt P and structure $\hat{S}_i$ actually used in each rollout. In the joint-update setting, both language and image-generation modules participate in gradient computation; the image-policy objective propagates gradients to the language module through the image branch's use of the VSRL conditioning representations.

E.3 Optimization Scope and Training Settings

StruFlow-GRPO training starts from the VSRL checkpoint obtained after supervised fine-tuning. StruFlow-GRPO updates both the language and image-generation modules, while the VAE remains frozen throughout. To analyze the role of joint updates, we construct a Gen-only RL control: both settings use the same SFT initialization, trajectory rewards, Flow-GRPO objective, and training budget, differing only in the parameter update scope. Gen-only RL freezes the language module, whereas StruFlow-GRPO updates both language and image-generation modules.

Table 15: **StruFlow-GRPO reinforcement-learning configuration for BAGEL.**

Setting	BAGEL
Trajectories per prompt G	8
Optimization steps	200
Global prompts per step	4
Learning rate	2×10^{-6}
Clip threshold ε	10^{-4}
Structure sampling temperature / top-p	0.7 / 0.95
Structure token limit	640
Rollout denoising steps	10
Rollout resolution	512×512
Language module (StruFlow-GRPO)	Trainable
Image-generation MoE (StruFlow-GRPO)	Trainable
VAE parameters	Frozen

F Evaluation Protocols and Result Sources

F.1 Conventions for the Main Table

Publicly reported results and our evaluations are marked separately. † denotes publicly reported scores, and ‡ denotes third-party reevaluations. A dash denotes an unavailable reported result. All metrics in the main table are displayed on a 100-point scale; CVTG-2K Word Accuracy is multiplied by 100, while the dedicated table retains the original scales of all five metrics.

Show-o2's GenEval result uses the rewritten-prompt setting reported in its original work (Xie et al., 2025), which differs from our GenEval Short evaluation. For OneIG, the main table reports the equal-weight mean of Alignment, Text, Reasoning, Style, and Diversity.

F.2 Reporting Settings for Each Benchmark

BAGEL is evaluated on five generation benchmarks; generation evaluation for Cheers uses GenEval, DPG-Bench, and TIIF-Bench for cross-backbone validation and the natural-language reasoning comparison.

Table 16: **Generation and understanding evaluation protocols.**

Benchmark	Setting	Reported metrics
GenEval	Original Short; rewritten Long	Overall score and six object-composition scores; the complete evaluation contains 553 prompts with 4 images each.
DPG-Bench	Official evaluation procedure	Overall score; each setting uses its corresponding question-answering and scoring procedure.
TIIF-Bench	We use testmini with separate Short/Long versions	Overall instruction-following score; evaluators for public results are distinguished by source.
OneIG-Bench	Five official dimensions	Five official dimensions; the main table reports their equal-weight mean.
CVTG-2K	Visual text generation	Word Accuracy, NED, CLIPScore, VQAScore, and Aesthetic.
Understanding evaluation	Benchmark-specific settings	MMBench, MMMU, MathVista; official MME overall score; mean over four PairJigsaw layouts.

F.3 TIIF Evaluation

We evaluate TIIF-Bench on testmini using Qwen2.5-VL-72B. Third-party reevaluations are cited and marked in the tables (Huang et al., 2025); Qwen-Image and GPT-Image-1 retain their reported scores under separate evaluation settings.

F.4 Training Stages

All BAGEL results at the SFT stage use the same VSRL checkpoint obtained by supervised fine-tuning, which also initializes both Gen-only RL and StruFlow-GRPO. Cheers VSRL is trained on 5,000 structurally supervised samples.

The Cheers base-model GenEval Short and DPG scores are from Zhang et al. (2026); GenEval Long is evaluated under our protocol. The Cheers comparison trains natural-language CoT and VSRL with the same number of samples under the same supervised training settings.