PRISM: BIDIRECTIONAL SELF-TEACHING FOR UNIFIED MULTIMODAL MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Unified Multimodal Models (UMMs) aim to support visual understanding and generation within a single model. Existing post-training methods either use understanding as a fixed teacher to improve generation, or rely on ground-truth trajectories to supervise both understanding and generation. Prism instead derives bidirectional supervision from self-generated interleaved rollouts, constructing training targets according to the reliability of their text and image chains. We propose Prism, a bidirectional self-teaching post-training method for UMMs. Like a prism decomposing mixed light, Prism decouples a coupled interleaved trajectory into a text chain and an image chain, assesses the reliability of each chain, and constructs a corresponding training target from their reliability states. Experiments on both SenseNova-U1 and BAGEL show that Prism improves both understanding and generation. On the generation side, GenEval improves from 85.8 to 89.2 (+3.4), and DPG improves from 86.1 to 87.3 (+1.2) on SenseNova-U1. On the understanding side, the total MME score improves from 2282.6 to 2311.6, and MMVP improves from 58.0 to 62.0. Consistent improvements are observed on BAGEL. These results highlight that Prism enables coordinated improvements in generation and understanding by constructing reliability-guided self-training targets from self-generated trajectories, without relying on an external verifier or externally annotated corrected trajectories during target construction.



Figure 1: Capabilities of Prism. Prism improves both understanding and generation through self-teaching on interleaved text-image trajectories. The base model shows redundant steps on step 2 in the cooking tutorial, while Prism produces a coherent and correct tutorial. It also improves performance on standalone understanding and generation tasks.

1 INTRODUCTION

With the rapid development of large language models (Brown et al., 2020; Sun et al., 2021; Bai et al., 2023a) and multimodal generation models (Liu et al., 2023; Chen et al., 2024b; Bai et al., 2023b), researchers have begun to explore unifying understanding and generation within a single model. Unlike treating understanding and generation as two separate models, Unified Multimodal Models (UMMs) integrate text, images, and their interactions into a unified architecture, enabling visual question answering and image generation to share knowledge and training signals. Recent

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.

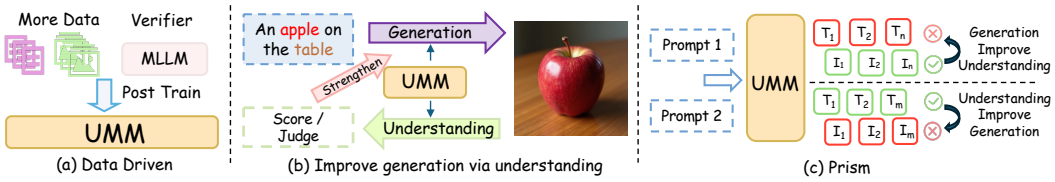


Figure 2: Difference between Prism and previous post-training methods. (a) Some prior methods are data-driven or rely on external supervision or verifiers. (b) Others improve generation through understanding in a fixed direction. (c) Prism instead enables bidirectional improvement between understanding and generation without external supervision or verifiers.

UMMs, from Chameleon (Team, 2024) to SenseNova-U1 (Diao et al., 2026), have demonstrated the feasibility of supporting both understanding and generation within a single model through different architectural designs. However, these models also reveal a fundamental problem: the understanding and generation capabilities within a UMM are asymmetric.

Recent studies reveal an asymmetry in UMMs: a model can judge whether an image matches a text, yet fail to generate an image that follows the same text (Xie et al., 2026a). Most methods exploit this in a fixed direction, using understanding to improve generation. GvU (Pan et al., 2026) and SRUM (Jin et al., 2025) turn alignment judgments into generation rewards, Internal Gap Self-Improvement (Han et al., 2026) turns them into supervision or DPO preference pairs, and RecA (Xie et al., 2026a) reconstructs images from visual understanding embeddings. Others work in reverse, generating depth, segmentation, or reconstruction targets to sharpen perception (Su et al., 2026), or fine-tuning on synthesized editing and interleaved reasoning traces (Zheng et al., 2026; Gu et al., 2026). A further set applies preference or policy optimization (Yang et al., 2026; Nie et al., 2026; Hu et al., 2026; Yan et al., 2026), and UniCorn (Fang et al., 2026) explores self-play. These methods either are limited to improving generation or rely on external ground-truth pairs, reward models, or verifiers. Prism instead derives bidirectional supervision from self-generated interleaved trajectories, building training targets from text and image reliability without ground-truth trajectories or an external verifier, as illustrated in Figure 2 and Figure 3.

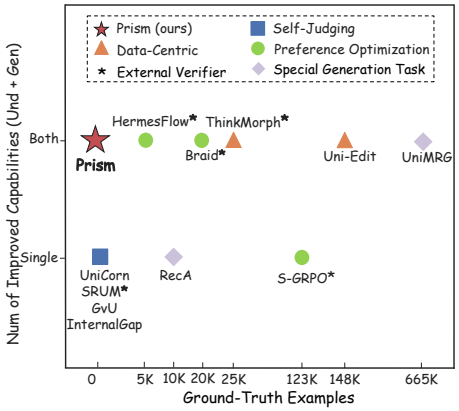


Figure 3: Prism's position by ground-truth supervision and improved capabilities.

Prism improves both understanding and generation through self-teaching on interleaved text-image tasks. In these trajectories, text and image alternate across multiple steps rather than forming a single input-output pair. Because the text and image chains are coordinated expressions of the same underlying task, each chain can be audited independently. An interleaved trajectory can therefore be decomposed into a text chain that represents understanding and an image chain that represents generation, each of which may be reliable or unreliable in different ways.

As shown in Figure 1, when the base model generates an interleaved cooking tutorial, the first image already shows chopped tomatoes during the ingredient-preparation stage, while the next step again depicts cutting tomatoes. For tutorial-style tasks, this is either redundant or indicates leakage of a future state into an earlier step. Conversely, although not shown in Figure 1, the opposite case can also occur: the image chain may still complete the task correctly even when the text chain contains erroneous reasoning or descriptions, because the image does not strictly follow the faulty text.

Therefore, the key challenge is to identify, for each sample, whether the understanding or generation side is reliable enough to teach the other. Prism addresses this challenge by performing sample-wise self-teaching over interleaved trajectories, enabling bidirectional improvement in both understanding and generation.

We propose Prism, a bidirectional self-teaching post-training method for interleaved text-image trajectories. Prism treats each self-sampled trajectory as a coupled multimodal signal and decomposes it into two independently assessable views: a text chain and an image chain. By comparing the reliability of these two chains, Prism avoids assuming a fixed supervision direction and instead determines on a per-sample basis whether text should supervise image, image should supervise text, or unreliable patterns in both chains should be suppressed.

The key contributions can be summarized as follows:

- **Method:** We introduce a bidirectional post-training paradigm that selects the teaching direction for each trajectory, allowing multimodal understanding and generation to improve each other rather than imposing a fixed supervision direction.
- **Framework:** Prism provides a reliability-guided self-teaching framework. It constructs training targets from self-generated interleaved rollouts by assessing the reliability of their text and image chains, without ground-truth supervision or external verifiers.
- **Performance:** Prism improves multimodal understanding and generation across different UMMs, including interleaved generation. The results demonstrate its effectiveness and generality.

2 Related Work

2.1 Unified Understanding and Generation Models

UMM architectures can be broadly categorized into four types. Bridge architectures (Chen et al., 2025a; Wu et al., 2025b; Song et al., 2026) retain separate understanding and generation backbones connected through lightweight interfaces. Unified Transformer architectures (Team, 2024; Zhou et al., 2025a; Xie et al., 2025; Wang et al., 2024; Wu et al., 2025a; Chen et al., 2025b; Xie et al., 2026b) process both modalities within a shared Transformer backbone, with variants using discrete image tokens, mixed autoregressive-diffusion objectives, or decoupled visual encoders. Mixture-of-Transformer-Experts (MoT) architectures (Deng et al., 2025; Diao et al., 2026) allocate separate expert parameters for understanding and generation while sharing attention. Diffusion-based architectures (AI et al., 2026) employ diffusion language models to jointly model text and images.

2.2 UMM Post-Training

Existing UMM post-training methods follow four main directions. Preference optimization and reinforcement learning use externally scored pairs in HermesFlow (Yang et al., 2026), reward components in Unified Interleaved GRPO (Nie et al., 2026) and BRAID (Hu et al., 2026), or ground-truth trajectories for VLM adaptation in S-GRPO (Yan et al., 2026). Data-construction approaches include Uni-Edit (Zheng et al., 2026), with 148K editing examples, and ThinkMorph (Gu et al., 2026), with approximately 24K reasoning traces. Specialized generation tasks include RecA (Xie et al., 2026a), which primarily improves generation through image reconstruction, and UniMRG (Su et al., 2026), which enhances perception using externally produced depth and segmentation targets. Self-judging methods use understanding-side feedback to guide generation, including GvU (Pan et al., 2026), SRUM (Jin et al., 2025), and Internal Gap Self-Improvement (Han et al., 2026). UniCorn (Fang et al., 2026) constructs supervision through self-play. UniRL (Mao et al., 2025) improves both tasks using self-generated images, but requires pre-constructed question-answer pairs as ground-truth supervision and applies a fixed round-trip via SFT and GRPO.

3 Method

Unlike prior post-training methods that rely on external supervision or a fixed supervision direction, Prism derives bidirectional supervision from self-generated interleaved trajectories. It constructs training targets according to the reliability of text and image chains, without ground-truth trajectories or external verifiers. As illustrated in Figure 4, given a task prompt, the model generates an interleaved trajectory. Frozen model copies then assess the text and image chains and map the trajectory to one of four training actions. Only the rollout model is updated using the resulting targets.

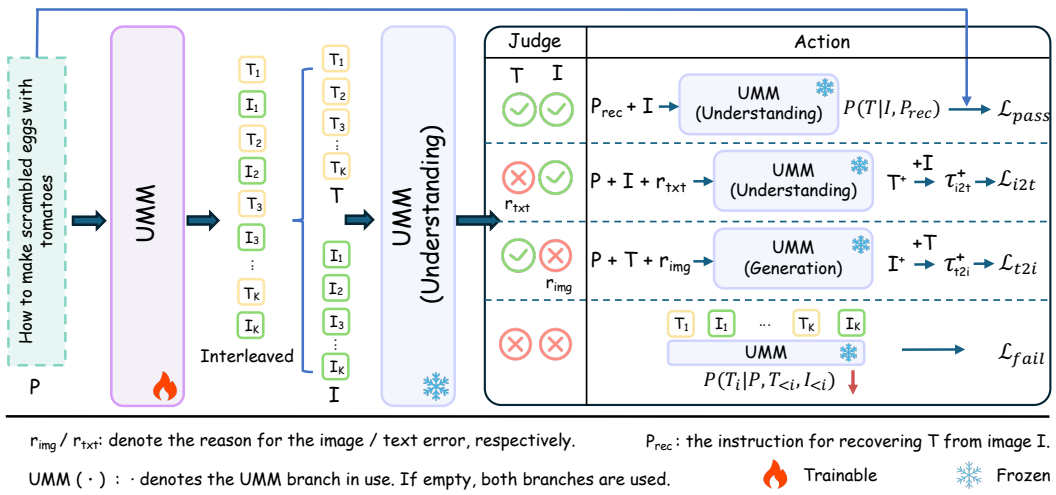


Figure 4: Overview of the Prism. Given a text prompt, Prism first uses the UMM to roll out an interleaved trajectory. It then uses the UMM in a judging role to assess the reliability of the text chain and image chain. Based on the resulting status, Prism selects the corresponding processing action and loss function. All UMM instances are initialized from the same base model. Only the rollout UMM is trainable, while the UMMs used for judging and action selection remain frozen.

3.1 Preliminary

Given a text prompt P, the UMM generates an interleaved text-image trajectory:

$$\tau = (T_1, I_1, T_2, I_2, \ldots, T_K, I_K), \tag{1}$$

where T_i and I_i denote the text and image segments at step i. For the interleaved tasks considered here, the trajectory has two modality-specific views, the complete text chain $T = (T_1, \ldots, T_K)$, representing understanding through planning and reasoning, and the complete image chain $I = (I_1, \ldots, I_K)$, representing generation through visual states and cross-step consistency. Both are projections of the same interleaved trajectory.

Standard UMM interleaved training models the text-image trajectory autoregressively:

$$p_\theta(\tau \mid P) = \prod_{i=1}^{K} p_\theta(T_i \mid P, T_{<i}, I_{<i})\; p_\theta(I_i \mid P, T_{\le i}, I_{<i}). \tag{2}$$

The standard training loss for an interleaved trajectory is

$$\mathcal{L}_{\text{interleave}}(\tau) = \sum_{i=1}^{K} \left[\lambda_{\text{und}} \mathcal{L}_{\text{und}}(f_\theta(\cdot \mid P, T_{<i}, I_{<i}), T_i) + \lambda_{\text{gen}} \mathcal{L}_{\text{gen}}(f_\theta(\cdot \mid P, T_{\le i}, I_{<i}), I_i)\right], \tag{3}$$

where $\mathcal{L}_{\text{und}}$ is the token-level cross-entropy loss for text prediction, and $\mathcal{L}_{\text{gen}}$ is the flow-matching loss for image generation (Lipman et al., 2022). λ_{und} and λ_{gen} are modality-balancing weights that may vary across architectures.

3.2 Per-Modality Reliability Judgment

The core premise of Prism is that, under the same task prompt P, the text chain T and the image chain I are two modality-specific expressions of the same underlying task content. Based on this premise, Prism judges the reliability of the complete text chain and complete image chain separately:

$$(v_{\text{txt}}, r_{\text{txt}}) = \mathcal{J}(P, T, q_{\text{txt}}), \qquad (v_{\text{img}}, r_{\text{img}}) = \mathcal{J}(P, I, q_{\text{img}}), \tag{4}$$

where $\mathcal{J}$ is a frozen copy of the base model, and q_{txt} and q_{img} are review instructions. The indicators $v_{\text{txt}}, v_{\text{img}} \in \{0, 1\}$ denote the reliability of the understanding and generation sides, while r_{txt} and r_{img} specify detected errors. Prism judges the complete chains and uses the resulting states to select a training action, as illustrated in Figure 4.

3.3 Reliability-Guided Supervision Construction

Based on the judgments in Section 3.2, Prism selects one of four actions for each trajectory: enhancing cross-modal consistency for reliable trajectories, rewriting an unreliable chain from the reliable side, or suppressing trajectories whose two chains are unreliable.

Both pass, $v_{\text{txt}} = 1, v_{\text{img}} = 1$. When both chains are reliable, the trajectory provides a reliable interleaved training sample. Prism conditions text reconstruction on the image chain, reinforcing cross-modal consistency throughout the trajectory. Let P_{rec} denote the instruction for recovering T from image chain I under the original task. The objective is

$$\mathcal{L}_{\text{pass}} = \mathcal{L}_{\text{und}}\big(f_\theta(T \mid P_{\text{rec}}, I), T\big). \tag{5}$$

Text-to-image rewriting, $v_{\text{txt}} = 1, v_{\text{img}} = 0$. When the text chain is judged reliable but the image chain is not, Prism preserves T and uses P, T, and r_{img} to generate a corrected image chain $I^+ = (I_1^+, \ldots, I_K^+)$. The resulting corrected trajectory and training objective are

$$\tau_{\text{t2i}}^+ = (T_1, I_1^+, T_2, I_2^+, \ldots, T_K, I_K^+), \qquad \mathcal{L}_{\text{t2i}} = \mathcal{L}_{\text{interleave}}(\tau_{\text{t2i}}^+). \tag{6}$$

Image-to-text rewriting, $v_{\text{txt}} = 0, v_{\text{img}} = 1$. When the image chain is judged reliable but the text chain is not, Prism preserves I and uses P, I, and r_{txt} to generate a corrected text chain $T^+ = (T_1^+, \ldots, T_K^+)$. The resulting corrected trajectory and training objective are

$$\tau_{\text{i2t}}^+ = (T_1^+, I_1, T_2^+, I_2, \ldots, T_K^+, I_K), \qquad \mathcal{L}_{\text{i2t}} = \mathcal{L}_{\text{interleave}}(\tau_{\text{i2t}}^+). \tag{7}$$

Both fail, $v_{\text{txt}} = 0, v_{\text{img}} = 0$. When both chains are judged unreliable, Prism does not construct a positive target. Instead, it applies a token-level unlikelihood objective to the text tokens in the original trajectory, reducing the probability of reproducing its erroneous text patterns:

$$\mathcal{L}_{\text{fail}}(\tau) = -\sum_{i=1}^{K}\sum_{n=1}^{|T_i|} \log\left(1 - p_\theta(t_{i,n} \mid P, T_{<i}, I_{<i}, t_{i,<n})\right), \tag{8}$$

where $T_i = (t_{i,1}, \ldots, t_{i,|T_i|})$ denotes the i-th text segment and $t_{i,<n}$ denotes the tokens preceding $t_{i,n}$ within that segment. Unlike text, image generation is optimized through continuous flow matching, for which no direct counterpart of token-level unlikelihood is available. We therefore restrict negative supervision to the text tokens.

For the two rewriting cases, since the rewritten chain is newly generated and may fail to correct the detected errors, Prism verifies each rewritten trajectory before using it as a positive training target, retrying failed corrections and skipping trajectories that remain invalid. Appendix F details the verification procedure. During training, only the rollout model is updated, while judgment, rewriting, and verification use frozen copies of the base checkpoint to prevent co-evolution with the rollout model. Jointly updating them leads to training collapse, as shown in Appendix D.

3.4 Training Objective

Each trajectory contributes to one of the four objectives according to its reliability state. So the overall objective is

$$\mathcal{L}_{\text{total}} = \lambda_{\text{pass}}\mathcal{L}_{\text{pass}} + \mathcal{L}_{\text{t2i}} + \mathcal{L}_{\text{i2t}} + \lambda_{\text{fail}}\mathcal{L}_{\text{fail}}, \tag{9}$$

where λ_{pass}, λ_{i2t}, and λ_{fail} weight the both-pass, image-to-text rewriting, and both-fail objectives, respectively. The text-to-image rewriting objective uses unit weight.

4 Experiments

4.1 Experimental Setup

Data. We use GPT-5.5 to generate 2.4K task prompt seeds spanning tutorials, stories, spatial reasoning, and other task types. These prompts provide task descriptions only. Prism generates interleaved trajectories online and constructs training targets from the model's own rollouts.

Table 1: Prism improves image generation performance.

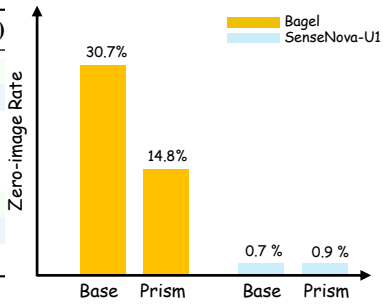
Model	Prism	GenEval							DPG
		Single	Two	Count	Color	Position	Attri.	Overall	
SenseNova-U1	✗	100.0	92.9	75.6	91.7	85.5	68.7	85.8	86.1
	✓	**100.0** (0.0)	**96.7** (+3.8)	**87.2** (+11.6)	**89.6** (-2.1)	**86.8** (+1.3)	**75.0** (+6.3)	**89.2** (+3.4)	**87.3** (+1.2)
BAGEL	✗	99.1	93.0	76.5	86.0	47.2	57.0	76.5	84.0
	✓	**99.4** (+0.3)	**96.5** (+3.5)	**76.9** (+0.4)	**86.5** (+0.5)	**48.0** (+0.8)	**62.5** (+5.5)	**78.3** (+1.8)	**85.0** (+1.0)

Table 2: Prism improves multimodal understanding performance. We evaluate the effectiveness of Prism across understanding benchmarks covering general knowledge, subject knowledge, and others.

Model	Prism	MME-P	MME-C	MMBench	MMMU	MMStar	MMVP
SenseNova-U1	✗	1699.7	582.9	86.1	59.0	65.1	58.0
	✓	**1713.4** (+13.7)	**598.2** (+15.3)	**87.3** (+1.2)	**60.6** (+1.6)	**66.5** (+1.4)	**62.0** (+4.0)
BAGEL	✗	1681.0	679.6	84.6	53.7	65.0	**69.3**
	✓	**1685.2** (+4.2)	**685.4** (+5.8)	**88.4** (+3.8)	**55.0** (+1.3)	**69.6** (+4.6)	70.0 (+0.7)

Table 3: **OpenING pairwise evaluation. Left:** Win rates under different tie-handling settings. **Right:** Zero-image rates, i.e., outputs with no generated image. Lower is better.

Method	FDT	w/o Tie	w/ Tie (0)	w/ Tie (.5)
SenseNova-U1 + Prism	**74.93%**	**76.65%**	**66.79%**	**73.22%**
SenseNova-U1	72.03%	72.91%	62.55%	69.66%
GPT-4o+DALL-E3	63.79%	66.92%	56.36%	64.25%
Human	61.10%	64.66%	51.90%	61.76%
Gemini1.5+Flux	38.35%	39.04%	31.23%	41.24%
BAGEL + Prism	34.70%	31.66%	25.80%	35.05%
BAGEL	30.84%	27.94%	23.47%	31.47%
SEED-X	21.27%	17.42%	15.92%	20.23%



Evaluation. We evaluate image generation on GenEval (Ghosh et al., 2023) and DPG-Bench (Hu et al., 2024), multimodal understanding on MME (Fu et al., 2026), MMBench (Liu et al., 2024), MMMU (Yue et al., 2024), MMStar (Chen et al., 2024a), MMVP (Tong et al., 2024), and POPE (Li et al., 2023), and interleaved generation on OpenING (Zhou et al., 2025b). We follow the official evaluation settings for all benchmarks and report the multiple-choice accuracy for MMMU.

Training. Since some UMMs lack interleaved text-image generation support or publicly available model weights, we select SenseNova-U1 (Diao et al., 2026) and BAGEL (Deng et al., 2025), which satisfy both requirements. The main experiments use a learning rate of 3e-5, with $\lambda_{\text{pass}} = 0.5$, and $\lambda_{\text{fail}} = 0.005$. The text-to-image rewriting objective uses unit weight. Training is performed on 16 A800 GPUs for approximately 24 hours.

For more details, please refer to the **Appendix A**.

4.2 Prism Simultaneously Improves Understanding and Generation

We evaluate Prism on SenseNova-U1 and BAGEL across image generation and multimodal understanding benchmarks. Tables 1 and 2 show consistent gains on both models.

On SenseNova-U1, Prism improves GenEval by 3.4 points and DPG by 1.2 points, with notable gains in counting, two-object generation, and attribute binding. On understanding benchmarks, it improves MME by 29.0 points and MMVP by 4.0 points, with gains on the other benchmarks as well. BAGEL shows the same overall trend on the benchmarks reported in Tables 1 and 2, providing evidence that the method can transfer across the two evaluated UMM backbones.



Figure 5: Improved generation. We use SenseNova-U1 as the baseline model. The model trained with Prism shows more accurate adherence to prompt, including spatial relations, object counts, and style. The underlined text highlights the generation differences under complex instructions.

Table 4: Compared with similar methods on BAGEL, our method can simultaneously improve both understanding and generation, demonstrating bidirectional enhancement capability.

Model	MME	MMBench	MMMU	MMStar	MMVP	GenEval	DPG
BAGEL	2361	84.6	53.7	65.0	69.3	76.5	84.0
+ SRUM	2350 (-11)	84.4 (-0.2)	53.4 (-0.3)	64.7 (-0.3)	70.0 (+0.7)	77.4 (+0.9)	85.2 (+1.2)
+ UniCorn	2337 (-24)	84.1 (-0.5)	54.5 (+0.8)	65.0 (+0.0)	70.0 (+0.7)	80.3 (+3.8)	86.2 (+2.2)
+ Prism	2371 (+10)	88.4 (+3.8)	55.0 (+1.3)	69.6 (+4.6)	70.0 (+0.7)	78.3 (+1.8)	85.0 (+1.0)

4.3 Prism Improves Interleaved Generation

We evaluate Prism on OpenING Zhou et al. (2025b), which measures open-ended interleaved image–text generation with pairwise win rates. Table 3 reports the win rates and zero-image rates.

Prism improves SenseNova-U1 by 2.90 to 4.24 points and BAGEL by 2.33 to 3.86 points across all tie-handling settings. Among the compared entries, SenseNova-U1 with Prism achieves the highest win rate in all four settings. Prism also reduces the zero-image rate of BAGEL from 30.7% to 14.8%, indicating improved ability to generate interleaved trajectories. SenseNova-U1 already has a near-zero zero-image rate of 0.7% before training, and Prism preserves this capability with a rate of 0.9%. Overall, these results show that Prism improves interleaved generation across two UMMs.

4.4 Comparison with Related Post-Training Methods

We compare Prism with SRUM and UniCorn on BAGEL in Table 4. SRUM and UniCorn both improve GenEval and DPG, but reduce MME and MMBench. On other understanding benchmarks, some scores stay flat or drop. This shows that they do not achieve consistent gains in both generation and multimodal understanding. By contrast, Prism improves MME by 10 points, MMBench by 3.8 points, MMMU by 1.3 points, and MMStar by 4.6 points, while maintaining comparable MMVP performance. It also improves GenEval by 1.8 points and DPG by 1.0 point. Results show the four losses contribute distinctly to understanding and generation.

4.5 Ablation Experiment

To ablate each loss, we train trajectories assigned to that loss on their unmodified rollouts with $\mathcal{L}_{\text{interleave}}$, leaving the other losses unchanged. Table 5 shows that replacing $\mathcal{L}_{\text{t2i}}$ reduces GenEval

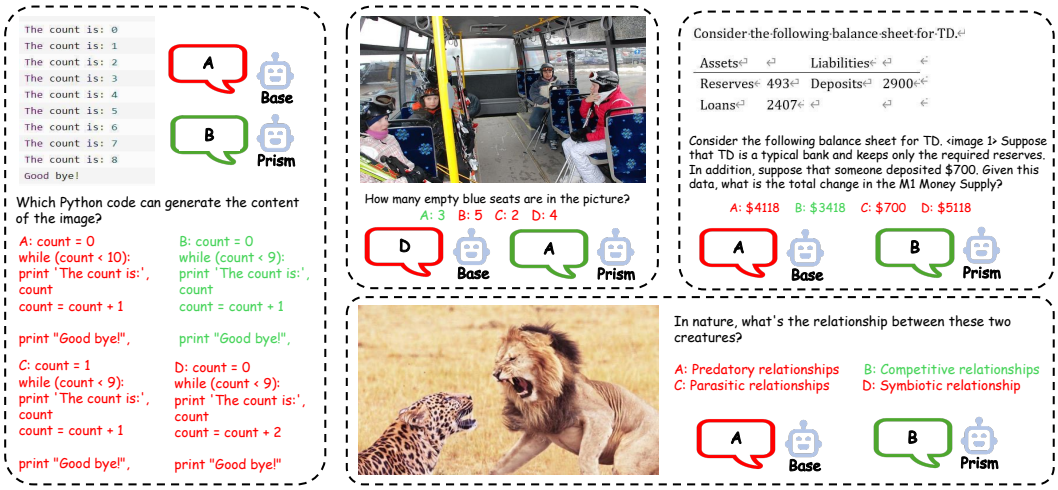


Figure 6: Improved understanding. We use SenseNova-U1 as the baseline model. The model trained with Prism demonstrates more accurate responses on diverse multimodal multiple-choice tasks, including code reasoning, visual counting, numerical reasoning, and commonsense understanding. For more examples, please refer to the Appendix G

Table 5: Loss ablations on SenseNova-U1. Dashes denote the base model without post-training. Crosses denote replacement with $\mathcal{L}_{\text{interleave}}$ on the corresponding unmodified rollouts.

Loss Terms				Understanding					Generation	
$\mathcal{L}_{\text{pass}}$	$\mathcal{L}_{\text{t2i}}$	$\mathcal{L}_{\text{i2t}}$	$\mathcal{L}_{\text{fail}}$	MME	MMBench	MMMU	MMStar	MMVP	GenEval	DPG
-	-	-	-	2283	86.1	59.0	65.1	58.0	85.8	86.1
✗	✓	✓	✓	2299 (+16)	86.6 (+0.5)	61.0 (+2.0)	65.3 (+0.2)	60.0 (+2.0)	**89.4** (+3.6)	87.2 (+1.1)
✓	✗	✓	✓	2294 (+11)	**87.4** (+1.3)	60.1 (+1.1)	64.9 (-0.2)	59.3 (+1.3)	88.0 (+2.2)	87.0 (+0.9)
✓	✓	✗	✓	2299 (+16)	86.6 (+0.5)	60.1 (+1.1)	65.6 (+0.5)	56.7 (-1.3)	88.8 (+3.0)	**87.3** (+1.2)
✓	✓	✓	✗	2310 (+27)	86.8 (+0.7)	**61.3** (+2.3)	65.3 (+0.2)	60.0 (+2.0)	88.7 (+2.9)	87.1 (+1.0)
✓	✓	✓	✓	**2312** (+29)	87.3 (+1.2)	60.6 (+1.6)	**66.5** (+1.4)	**62.0** (+4.0)	89.2 (+3.4)	**87.3** (+1.2)

Table 6: Analysis of reliability-guided target construction on SenseNova-U1.

Setting	GenEval	DPG	MME	MMB	MMMU	MMStar	MMVP
Base Model	85.8	86.1	2282.6	86.1	59.0	65.1	58.0
Base w/ Self-Training	87.7	86.6	2276.3	85.5	60.0	64.7	58.0
Prism w/o Rewriting	88.4	86.5	2300.6	86.8	**61.0**	65.6	60.7
Prism w/ Random Routing	87.6	86.5	2297.5	86.5	59.9	65.5	59.3
Prism (Ours)	**89.2**	**87.3**	**2311.6**	**87.3**	60.6	**66.5**	**62.0**

by 1.2 points, while replacing $\mathcal{L}_{\text{i2t}}$ reduces MMVP by 5.3 points. Replacing $\mathcal{L}_{\text{pass}}$ slightly improves GenEval but lowers four of five understanding metrics. Replacing $\mathcal{L}_{\text{fail}}$ lowers both generation metrics and four of five understanding metrics. The four losses play distinct roles in improving understanding, generation, and cross-modal consistency.

4.6 Analysis of Reliability-Guided Target Construction

Table 6 compares Prism with three controls. Base w/ Self-Training applies $\mathcal{L}_{\text{interleave}}$ to all unmodified rollouts. Prism w/o Rewriting keeps judge-based routing but applies $\mathcal{L}_{\text{interleave}}$ to the original rollouts assigned to t2i and i2t, leaving pass and fail unchanged. Prism w/ Random Routing shuffles action assignments within each batch while preserving their counts. Full Prism outperforms self-training and random routing on all seven metrics, and w/o rewriting on six. These results support the contributions of reliability-guided routing and rewritten targets. More appears in Appendix C.

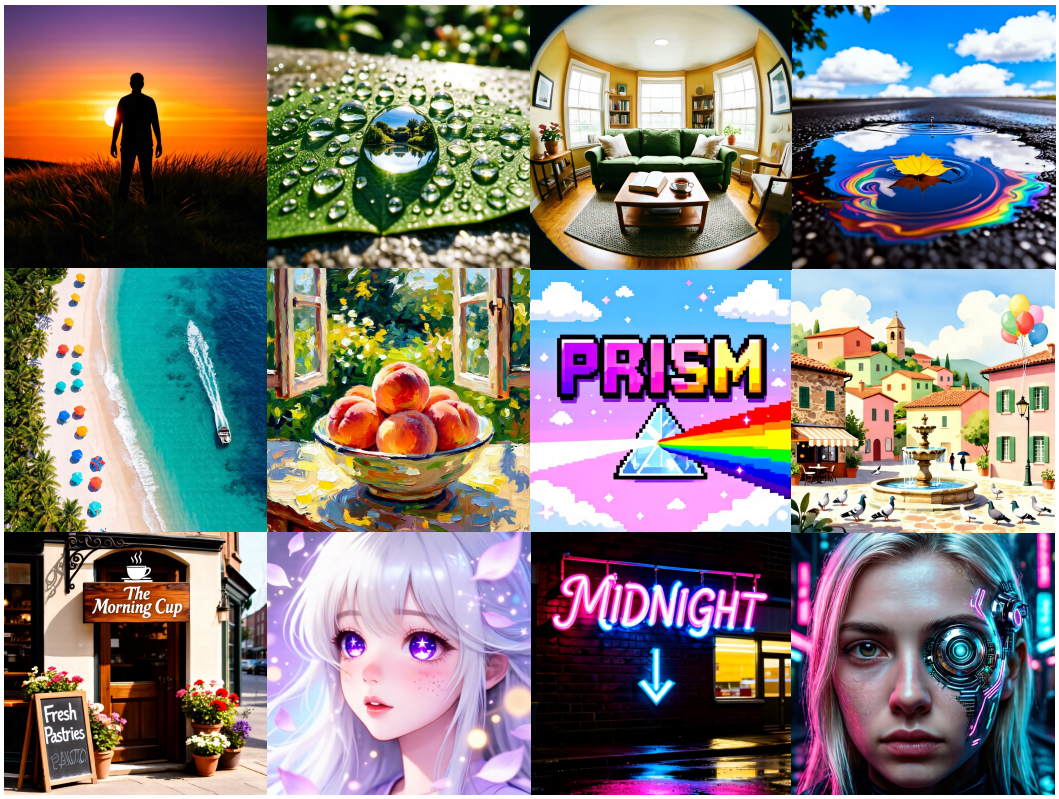


Figure 7: Qualitative examples generated by the model trained with Prism. The figure illustrates a diverse set of outputs, including realistic photography and artistic rendering, among others.

4.7 More Results

Qualitative. Figure 5 compares SenseNova-U1 with its Prism-trained variant. Prism more faithfully follows prompts involving object composition, counting, spatial relations, and detailed scene descriptions. Figure 6 further shows improved responses on code reasoning, visual counting, document-based numerical reasoning, and relation understanding. These examples complement the quantitative results on generation and multimodal understanding. More examples with diverse styles and quantities are provided in Figure 7 and Appendix H.

Hallucination analysis. We further evaluate Prism on POPE. As shown in Table 7, Prism improves Accuracy and F1 from 85.2 to 88.0 and from 84.9 to 86.7 on BAGEL, respectively. On SenseNova-U1, it improves Accuracy from 85.6 to 87.5 and F1 from 84.0 to 86.6. These results suggest improved visual grounding and fewer hallucination-related errors.

Table 7: POPE benchmark.

Model	Prism	Acc.	F1
BAGEL	✗	85.2	84.9
	✓	88.0 (+2.8)	86.7 (+1.8)
SenseNova-U1	✗	85.6	84.0
	✓	87.5 (+1.9)	86.6 (+2.6)

5 Conclusion and Limitations

We propose Prism, a bidirectional self-teaching method that derives targets through text and image reliability. It improves generation, understanding and interleaved generation on UMMs.

Despite these gains, Prism has three limitations. It targets interleaved text-image tasks with externally provided prompt seeds, and applies negative supervision to the complete text chain when both chains fail. Future work will extend Prism to broader multimodal tasks, fully self-sampled training, and localized negative supervision.

AI USE STATEMENT

In this work, we used GPT-5.5 to generate task prompt seeds for constructing the training prompt pool. We also used generative AI tools for writing assistance, manuscript polishing, and LaTeX formatting. We did not use generative AI tools to conceive the research ideas, design the methodology, conduct experiments, or analyze results. All AI-assisted outputs were reviewed by the authors, who take responsibility for the final content of this work, including its text, claims, data, and artifacts.

REPRODUCIBILITY STATEMENT

The complete method description is provided in Section 3 with formal mathematical definitions of all training objectives (Eq. 5– 8) and the overall loss (Eq. 9). Experimental setups including hyperparameters and training configurations are detailed in Section 4.1. The evaluation protocols for all benchmarks follow standard practices described in the corresponding original papers. Our code and trained checkpoints will be released upon publication.

REFERENCES

Inclusion AI, Tiwei Bie, Haoxing Chen, Tieyuan Chen, Zhenglin Cheng, Long Cui, Kai Gan, Zhicheng Huang, Zhenzhong Lan, Haoquan Li, et al. Llada2. 0-uni: Unifying multimodal understanding and generation with diffusion large language model. *arXiv preprint arXiv:2604.20796*, 2026.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023a.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 1(2):3, 2023b.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

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. *Advances in Neural Information Processing Systems*, 38, 2026.

Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025a.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *Advances in Neural Information Processing Systems*, 37:27056–27087, 2024a.

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. *arXiv preprint arXiv:2501.17811*, 2025b.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024b.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

Haiwen Diao, Penghao Wu, Hanming Deng, Jiahao Wang, Shihao Bai, Silei Wu, Weichen Fan, Wenjie Ye, Wenwen Tong, Xiangyu Fan, et al. Sensenova-u1: Unifying multimodal understanding and generation with neo-unify architecture. *arXiv preprint arXiv:2605.12500*, 2026.

Zhen Fang, Ruiyan Han, XinYu Sun, Yuchen Ma, Ziheng Wang, Yu Zeng, Zehui Chen, Lin Chen, Wenxuan Huang, Wei-Jie Xu, et al. 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.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *Advances in Neural Information Processing Systems*, 38, 2026.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Jiawei Gu, Yunzhuo Hao, Huichen Wang, Linjie Li, Michael Qizhe Shieh, Yejin Choi, Ranjay Krishna, and Yu Cheng. Thinkmorph: Emergent properties in multimodal interleaved chain-of-thought reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 141405–141447, 2026.

Yujin Han, Hao Chen, Andi Han, Zhiheng Wang, Xinyu Liu, Shiwei Zhang, Difan Zou, et al. Turning internal gap into self-improvement: Promoting the generation-understanding unification in mllms. In *International Conference on Learning Representations*, volume 2026, pp. 136501–136534, 2026.

Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024.

Zican Hu, Xuyang Hu, Yiming Liu, Zuwei Long, Wei Liu, Yunzhuo Hao, Jiawei Gu, Linjie Li, Yu Cheng, Zhenhong Sun, et al. Bridging interleaved multi-modal reasoning as a unified decision process. *arXiv preprint arXiv:2607.03748*, 2026.

Weiyang Jin, Yuwei Niu, Jiaqi Liao, Chengqi Duan, Aoxue Li, Shenghua Gao, and Xihui Liu. Srum: Fine-grained self-rewarding for unified multimodal models. *arXiv preprint arXiv:2510.12784*, 2025.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 292–305, 2023.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, et al. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025.

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

Weijia Mao, Zhenheng Yang, and Mike Zheng Shou. Unirl: Self-improving unified multimodal models via supervised and reinforcement learning. *arXiv preprint arXiv:2505.23380*, 2025.

Ming Nie, Chunwei Wang, Jianhua Han, Hang Xu, and Li Zhang. Towards unified multimodal interleaved generation via group relative policy optimization. *Advances in Neural Information Processing Systems*, 38:5332–5353, 2026.

Jiadong Pan, Liang Li, Yuxin Peng, Yu-Ming Tang, Shuohuan Wang, Yu Sun, Hua Wu, Qingming Huang, and Haifeng Wang. Learning to generate via understanding: Understanding-driven intrinsic rewarding for unified multimodal models. *arXiv preprint arXiv:2603.06043*, 2026.

Lin Song, Wenbo Li, Guoqing Ma, Wei Tang, Bo Wang, Yuan Zhang, Yijun Yang, Yicheng Xiao, Jianhui Liu, Yanbing Zhang, et al. Joyai-image: Awaking spatial intelligence in unified multimodal understanding and generation. *arXiv preprint arXiv:2605.04128*, 2026.

Zihan Su, Hongyang Wei, Kangrui Cen, Yong Wang, Guanhua Chen, Chun Yuan, and Xiangxiang Chu. Generation enhances understanding in unified multimodal models via multi-representation generation. *arXiv preprint arXiv:2601.21406*, 2026.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*, 2021.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9568–9578. IEEE, 2024.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

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? *arXiv preprint arXiv:2506.02161*, 2025.

Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12966–12977. IEEE, 2025a.

Size Wu, Zhonghua Wu, Zerui Gong, Qingyi Tao, Sheng Jin, Qinyue Li, Wei Li, and Chen Change Loy. Openuni: A simple baseline for unified multimodal understanding and generation. *arXiv preprint arXiv:2505.23661*, 2025b.

Ji Xie, trevor darrell, Luke Zettlemoyer, and Xudong Wang. Reconstruction alignment improves unified multimodal models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 120095–120137, 2026a.

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. In *International Conference on Learning Representations*, volume 2025, pp. 28240–28264, 2025.

Jinheng Xie, Zhenheng Yang, and Mike Zheng Shou. Show-o2: Improved native unified multimodal models. *Advances in Neural Information Processing Systems*, 38:47490–47518, 2026b.

Yuming Yan, Kai Tang, Sihong Chen, Ke Xu, Dan Hu, Qun Yu, and Pengfei Hu. S-grpo: Unified post-training for large vision-language models. *arXiv preprint arXiv:2604.16557*, 2026.

Ling Yang, Xinchen Zhang, Ye Tian, Shiyi Zhang, Chenming Shang, Minghao Xu, Wentao Zhang, and Bin Cui. Hermesflow: Seamlessly closing the gap in multimodal understanding and generation. *Advances in Neural Information Processing Systems*, 38:62248–62272, 2026.

Yang Ye, Xianyi He, Zongjian Li, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, Li Yuan, et al. Imgedit: A unified image editing dataset and benchmark. *Advances in Neural Information Processing Systems*, 38, 2026.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9556–9567, 2024.

Dian Zheng, Manyuan Zhang, Hongyu Li, Hongbo Liu, Kai Zou, Kaituo Feng, and Hongsheng Li. Uni-edit: Intelligent editing is a general task for unified model tuning. *arXiv preprint arXiv:2605.21487*, 2026.

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. In *International Conference on Learning Representations*, volume 2025, pp. 6446–6469, 2025a.

Pengfei Zhou, Xiaopeng Peng, Jiajun Song, Chuanhao Li, Zhaopan Xu, Yue Yang, Ziyao Guo, Hao Zhang, Yuqi Lin, Yefei He, et al. Opening: A comprehensive benchmark for judging open-ended interleaved image-text generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 56–66. IEEE, 2025b.

A Training and Evaluation Details

A.1 Implementation Details and Hyperparameters

We train all models at a resolution of 1024×1024 for one epoch. For the interleaved objective, the cross-entropy and flow-matching loss weights are set to $0.1 : 1$ for SenseNova-U1 and $0.25 : 1$ for BAGEL. We use AdamW with a learning rate of 3×10^{-5} and an effective batch size of 64. The reported training time includes online rollout generation, reliability judgment, rewriting, and verification.

For rollout and image generation, we use 50 diffusion steps with global normalization. The text and image classifier-free guidance scales are set to 4.0 and 1.0, respectively. Each prompt is assigned a fixed random seed shared across compared models.

A.2 OpenING Arena Protocol

We follow the official pairwise arena protocol of OpenING (Zhou et al., 2025b). The evaluation set contains 2,160 samples. For each sample, two distinct entries are paired through circular rotation and random permutation, and their presentation order is randomized to mitigate position bias.

The compact comparison in the main paper contains eight entries and 4,320 pairwise battles, with each entry participating in 1,080 battles on average. The full leaderboard in Table 8 contains 15 entries, including our Prism-trained models, their corresponding base models, and 11 official OpenING reference outputs. It contains 4,143 valid pairwise battles. The number of battles per entry varies between 507 and 615, as reported in the final column of Table 8. Since the compact comparison and the full leaderboard are evaluated in separate arenas, their win rates are arena-relative and are not directly comparable.

We use the official **IntJudge-8B** checkpoint and decoding configuration. For each battle, IntJudge outputs one of four decisions: *A*, *B*, *Tie(A)*, or *Tie(B)*. We report four tie-handling metrics:

- **FDT:** *Tie(A)* is counted as a win for A, and *Tie(B)* is counted as a win for B.
- **w/o Tie:** only battles with an explicit winner are counted.
- **w/ Tie (0):** ties contribute zero wins to both models.
- **w/ Tie (.5):** ties contribute 0.5 wins to both models.

We use *w/ Tie (.5)* as the main reporting protocol. The *Human* entry uses the human-authored interleaved image–text sequences released with OpenING and participates in the same arena protocol as model outputs. Under the *w/o Tie* protocol, SenseNova-U1 with Prism achieves an 86.61% win rate over 478 decisive outcomes from 537 battles, exceeding the base model by 3.55 points and GPT-4o+DALL-E3 by 5.23 points.

A.3 Image Generation Evaluation

We evaluate image generation with the official data and scoring pipelines of GenEval (Ghosh et al., 2023) and DPG-Bench (Hu et al., 2024). GenEval contains 553 prompts, with four images generated per prompt. It evaluates six categories using Mask2Former and CLIP, and reports their equal-weighted average. DPG-Bench contains 1,065 dense prompts and uses mPLUG-VQA for evaluation.

All images are generated at 1024×1024 using the configuration above. For DPG-Bench, we generate four images per prompt and arrange them into a 2×2 grid. The official evaluator crops and scores each image at the native 1024 resolution. All comparisons with the base models use identical generation settings.

A.4 Multimodal Understanding Evaluation

We evaluate multimodal understanding using the official data, prompts, and evaluation scripts of MME, MMBench, MMMU, MMStar, POPE, and MMVP. No task-specific prompts are added.

Table 8: Full OpenING arena leaderboard. We report pairwise win rates under four tie-handling protocols. The final column reports decisive battles over total battles for each entry.

Rank	Model	FDT	w/o Tie	w/ Tie (0)	w/ Tie (.5)	Decisive / Total
1	SenseNova-U1 + Prism	**84.73%**	**86.61%**	**77.09%**	**82.59%**	478 / 537
2	SenseNova-U1	82.33%	83.06%	71.91%	78.62%	490 / 566
3	GPT-4o+DALL-E3	77.78%	81.38%	69.79%	76.91%	494 / 576
4	Human	76.88%	82.92%	67.50%	76.80%	486 / 597
5	Gemini1.5+Flux	62.19%	64.16%	50.36%	61.11%	438 / 558
6	BAGEL + Prism	54.38%	54.79%	42.93%	53.76%	438 / 559
7	SEED-LLaMA	43.32%	47.04%	35.92%	47.74%	423 / 554
8	Anole	48.66%	44.18%	35.50%	45.32%	421 / 524
9	BAGEL-7B-MoT	43.76%	40.89%	33.59%	42.51%	428 / 521
10	Emu3	42.59%	37.56%	30.99%	39.73%	434 / 526
11	Emu2	36.32%	36.20%	27.54%	39.50%	442 / 581
12	MiniGPT-5	26.02%	27.08%	21.14%	32.11%	480 / 615
13	SEED-X	30.84%	29.17%	27.48%	30.37%	504 / 535
14	MiniGPT-5mmd	20.57%	17.36%	14.91%	21.98%	455 / 530
15	NExT-GPT	14.20%	10.33%	9.27%	14.40%	455 / 507

Table 9: OneIG-Bench results. Each model generates four images per prompt on 1,120 English prompts. The five dimensions are scored with dedicated scorers and the benchmark provides no aggregate score. The better value is in bold.

Model	Prism	OneIG-Bench				
		Alignment	Text	Style	Reasoning	Diversity
SenseNova-U1	✗	**0.873**	**0.942**	0.405	0.326	0.170
	✓	0.869	0.937	**0.410**	**0.330**	**0.183**
BAGEL	✗	**0.813**	**0.366**	0.395	0.189	0.176
	✓	0.808	0.339	0.395	0.189	**0.179**

MME contains 14 tasks, including 10 perception tasks with a maximum score of 2,000 and four cognition tasks with a maximum score of 800. We report the total score in the main paper. MMBench is evaluated on its 1,176-example development split with the official option-rotation protocol. MMMU is evaluated on its 900-example validation split, including 847 multiple-choice and 53 open-ended questions. We report multiple-choice accuracy. MMStar contains 1,500 examples. POPE includes 9,000 questions from the random, popular, and adversarial settings, for which we report Accuracy and F1. MMVP contains 300 questions organized into 150 paired examples, and is scored only when both questions in a pair are answered correctly.

All understanding benchmarks are evaluated using a unified decoding configuration and the official aggregation scripts.

B Additional Capability Analysis

Prism constructs supervision from the consistency between the complete text and image chains of self-generated interleaved trajectories. Its training objectives therefore emphasize planning, visual state progression, and cross-modal consistency, rather than instruction following for single-image generation or image editing. This appendix reports results on benchmarks that fall outside this scope. All benchmarks are evaluated with the official protocols and judge models released by the benchmark authors.

OneIG-Bench. OneIG-Bench evaluates text-to-image generation on 1,120 English prompts with four images generated per prompt (Chang et al., 2026). As shown in Table 9, the results are mixed but close to the base models. On SenseNova-U1, Prism improves Diversity by 0.013, Style by 0.005, and Reasoning by 0.004, while Alignment and Text decrease slightly by about 0.005. The improved dimensions are consistent with training on trajectories that require coordinated textual plans and visual states. On BAGEL, Style, Reasoning, and Diversity are unchanged, Alignment

decreases slightly, and Text shows the largest decrease of 0.027. Text rendering is the weakest dimension of BAGEL among the five, with 0.366 compared with 0.942 on SenseNova-U1, and it remains the weakest after training. In summary, Prism preserves most OneIG capabilities, improves the dimensions aligned with its training objectives, and causes small decreases on Alignment and Text, which lie outside these objectives.

Table 10: TIIF-Bench and editing results. TIIF-Bench uses a short and a long prompt variant per prompt with Qwen2.5-VL-72B as the judge. ImgEdit-Bench scores 1–5 and GEdit-Bench (English) uses VIEScore on a 10-point scale. The better value is in bold.

Model	Prism	TIIF-Bench		Editing		
		Short	Long	GEdit-S	GEdit-Q	ImgEdit
SenseNova-U1	✗	0.807	**0.805**	**7.37**	**6.79**	**3.58**
	✓	**0.812**	0.794	7.23	6.52	3.47
BAGEL	✗	**0.699**	**0.699**	**6.89**	**6.93**	3.33
	✓	0.664	0.681	6.82	6.87	**3.38**

TIIF-Bench. TIIF-Bench evaluates instruction following under short and long prompt variants, scored by Qwen2.5-VL-72B over eleven sub-attributes (Wei et al., 2025). As shown in Table 10, on SenseNova-U1 Prism slightly increases the short score and slightly decreases the long score, while on BAGEL both scores decrease. On SenseNova-U1, the basic sub-attributes improve under both variants, the advanced ones improve under short prompts and decrease under long ones, and style control decreases under both.

These patterns follow from how the training organizes generation. Every image in a trajectory is generated with the preceding text and images as context and realizes only the incremental state of the current step, so the training reinforces continuation from rich context. A short prompt is close to realizing a single simple state and benefits. A long prompt instead requires satisfying all constraints in one generation from text alone, and style control requires the same one-shot fidelity, so neither is reinforced and both decrease. On BAGEL, which starts weaker on single-image instruction following, the gains do not materialize and the decreases spread across nearly all sub-attributes.

Editing. ImgEdit-Bench and GEdit-Bench evaluate instruction-based editing, scored by ImgEdit-Judge and VIEScore respectively (Ye et al., 2026; Liu et al., 2025). As shown in Table 10, editing results are mixed. On SenseNova-U1 both benchmarks decrease slightly, with the largest per-task decrease on style change. On BAGEL, ImgEdit-Bench slightly increases, and the increases concentrate on content-oriented operations such as extraction and person editing, while GEdit-Bench changes by less than 0.07.

The transfer is limited because the two tasks use the conditioning image differently. In a trajectory, the preceding images serve as state history, and each new image re-renders the updated state, inheriting entities and progress semantically while sharing few pixels with its predecessors. Instruction-based editing instead keeps most pixels of the input image and changes only the specified region. The training therefore reinforces semantic continuation conditioned on images but never pixel-level preservation, which explains why content-oriented operations improve while tasks that demand pixel fidelity stay flat or decrease. Obtaining editing gains requires editing-specific supervision as in editing-oriented methods such as Uni-Edit (Zheng et al., 2026).

C Detailed Reliability Judgment and Routing Analysis

We further examine the reliability judgments and training targets underlying the results in Table 6. All variants start from 2,395 prompt seeds and use the same one-epoch optimization budget and learning rate. Base w/ Self-Training optimizes unmodified self-generated rollouts with $\mathcal{L}_{\text{interleave}}$. Prism w/ Random Routing preserves the number of trajectories assigned to each action within a minibatch but permutes their assignments independently of the judged reliability states. Prism w/o Rewriting retains reliability-based routing but trains trajectories assigned to t2i and i2t on their unmodified rollouts instead of constructing rewritten targets.

Full Prism outperforms Base w/ Self-Training and Prism w/ Random Routing on all seven metrics in Table 6, and Prism w/o Rewriting on six of seven. Relative to Random Routing, it improves GenEval from 87.6 to 89.2, MME from 2297.5 to 2311.6, and MMVP from 59.3 to 62.0. These comparisons support the value of matching training actions to judged reliability states and constructing rewritten targets, beyond direct self-training or preserving the overall action mixture alone. We next examine the frequency of each reliability state, the evolution of routing during training, and the verification of rewritten targets. Tables 11 and 13 describe the first rollout of each prompt seed, whereas Table 12 describes all rollouts judged during the training epoch.

C.1 Empirical Routing Distribution

Table 11 reports the routing decisions for the first rollout of each of the 2,395 SenseNova-U1 prompt seeds. Both chains pass in 52.2% of these rollouts, while both fail in 14.7%. The remaining 794 rollouts (33.2%) have one reliable and one unreliable chain: 571 (23.8%) are assigned to t2i and 223 (9.3%) to i2t. Thus, asymmetric reliability occurs in both directions. Trajectory-specific routing allows Prism to condition each correction on the chain judged reliable, rather than applying a fixed rewriting direction to all asymmetric cases.

Table 11: Routing decisions for the first rollout of each of the 2,395 SenseNova-U1 prompt seeds.

Action	Reliability state	Count	Share
Pass	both chains reliable	1,250	52.2%
t2i	text reliable, image flawed	571	23.8%
i2t	text flawed, image reliable	223	9.3%
Fail	both chains flawed	351	14.7%

C.2 Routing Dynamics during Training

Table 12 tracks routing across the 38 updates of the one-epoch Full Prism run. Unlike Table 11, it includes all rollouts judged during training. When a rewritten candidate fails verification, that candidate is discarded and a replacement rollout is generated from a new prompt. The replacement is judged anew and may be assigned to any of the four actions. Consequently, the stage-wise percentages characterize the online training stream rather than the fixed first-rollout cohort. The judge remains a frozen copy of the base model throughout.

The routing distribution changes over the epoch. From S1 to S5, the Pass share rises from 49.3% to 59.3%, while the Fail share falls from 17.9% to 9.7%. The i2t share declines from 11.5% to 6.8%; in contrast, t2i rises initially and remains near one quarter of judged rollouts in later stages. The text-chain pass rate increases by 12.9 points, from 70.6% to 83.5%, whereas the image-chain pass rate increases by 5.3 points, from 60.8% to 66.1%. With the judge held fixed, these trends are consistent with improving rollout reliability, particularly on the text side, while image-side failures remain a substantial source of correction targets. Because failed candidates are replaced with newly sampled prompts, the stage-wise changes also reflect the composition of the online rollout stream.

Table 12: Routing dynamics across the 38 updates of one epoch on SenseNova-U1. Statistics include all judged rollouts, including replacements generated after failed rewrite verification. The judge is frozen throughout. Text and image pass rates are computed from the corresponding routing categories. All values are percentages.

Stage	Updates	Routing Distribution				Judge Pass Rate	
		Pass	t2i	i2t	Fail	Text	Image
S1	0–7	49.3%	21.3%	11.5%	17.9%	70.6%	60.8%
S2	8–15	51.6%	25.1%	8.7%	14.6%	76.7%	60.3%
S3	16–23	55.7%	25.2%	7.7%	11.4%	80.9%	63.4%
S4	24–30	57.6%	24.7%	7.1%	10.6%	82.3%	64.7%
S5	31–37	59.3%	24.2%	6.8%	9.7%	83.5%	66.1%

Table 13: Verification outcomes for rewriting candidates arising from the first-rollout cohort in Table 11. Accepted and discarded counts are candidate-level terminal outcomes; verification calls include retries for these candidates.

Rewrite	Candidates	Accepted	Discarded	Verification Calls
t2i	571	376 (65.8%)	195 (34.2%)	1,474 (I^+ reviews)
i2t	223	117 (52.5%)	106 (47.5%)	320 (T^+ reviews)

C.3 Verification of Rewritten Targets

A trajectory assigned to t2i or i2t is used as a positive rewritten target only if its correction passes verification. Table 13 reports verification outcomes for the rewriting candidates arising from the first-rollout cohort in Table 11. Of the 571 t2i candidates, 376 (65.8%) are accepted and 195 (34.2%) are discarded. Of the 223 i2t candidates, 117 (52.5%) are accepted and 106 (47.5%) are discarded. The lower acceptance rate for i2t shows that the two directions have different verification outcomes; neither set of generated corrections is admitted wholesale. Discarded candidates do not become positive training targets and are replaced with newly generated rollouts, which are judged and routed independently.

Verification also entails different amounts of review in the two directions. The first-rollout t2i candidates require 1,474 I^+ review calls, compared with 320 T^+ review calls for i2t candidates. These counts include retries, rather than additional training trajectories. Together, the acceptance rates and review counts characterize verification as a target-selection step: routing identifies a chain to correct, while verification determines whether the resulting correction can be used for training.

C.4 Judge Reliability, Contamination, and External-Judge Comparison

Routing and verification depend on judgments that may themselves be incorrect. We therefore compare the frozen SenseNova-U1 judge used by Full Prism with Qwen3.7 Plus in a separate judge comparison. We sample 200 training rollouts and obtain human annotations for their complete text and image chains, yielding 400 chain-level judgments. A positive judge prediction denotes a chain judged reliable. Table 14 reports agreement with the human annotations and the false discovery rate (FDR), defined as $\mathrm{FP}/(\mathrm{TP}+\mathrm{FP})$. FDR measures the fraction of chains accepted as reliable that are flawed according to the annotations, and is therefore directly relevant to the quality of accepted teacher signals.

Table 14: Chain-level agreement with human annotations on 200 rollout trajectories. Each text or image row contains 200 judgments; the *All* rows aggregate 400 judgments. FDR is $\mathrm{FP}/(\mathrm{TP}+\mathrm{FP})$.

Judge	Chain	Human Agreement				Judge Quality		
		TP	FP	TN	FN	Acc.	F1	FDR
SenseNova-U1 (frozen)	Text	113	18	47	22	80.0%	85.0%	13.7%
	Image	95	17	64	24	79.5%	82.3%	15.2%
	All	208	35	111	46	79.8%	83.7%	14.4%
Qwen3.7 Plus	Text	128	24	41	7	84.5%	89.2%	15.8%
	Image	87	7	74	32	80.5%	81.7%	7.4%
	All	215	31	115	39	82.5%	86.0%	12.6%

Qwen3.7 Plus has higher pooled accuracy and F1 than the frozen judge: 82.5% versus 79.8% and 86.0% versus 83.7%, respectively. However, its advantage does not extend uniformly to the chains accepted as reliable. The frozen judge has a lower text-chain FDR (13.7% versus 15.8%), while Qwen3.7 Plus has a lower image-chain FDR (7.4% versus 15.2%). Thus, higher aggregate agreement with human annotations does not imply lower contamination among accepted signals in both modalities.

The consequences of judgment errors also depend on their type. A false negative may reject a usable teacher chain; if the other chain is accepted, it can additionally trigger an unnecessary rewrite. When both chains are rejected, an otherwise reliable text chain may receive the unlikelihood objective.

Fail accounts for 14.7% of the first rollouts in Table 11, although this share does not measure the frequency of such erroneous assignments. A false positive can instead admit a flawed chain as a teacher signal. For rewritten targets, verification provides an additional gate: among the first-rollout candidates in Table 13, it discards 34.2% of t2i and 47.5% of i2t candidates. These measurements establish neither error-free routing nor error-free accepted targets, but identify where judgment noise enters the training pipeline and where verification filters its outputs.

The i2t results connect this analysis to the loss ablation. Replacing $\mathcal{L}_{\text{i2t}}$ with $\mathcal{L}_{\text{interleave}}$ on the corresponding unmodified rollouts lowers MMVP from 62.0 to 56.7 (Table 5). Meanwhile, verification accepts 117 of the 223 first-rollout i2t candidates and rejects 106 (Table 13). The ablation supports the contribution of the i2t training objective relative to direct training on the same rollouts, while the verification results show that its rewritten targets are filtered before use.

We also compare the downstream results obtained with the two judges. As shown in Table 15, the external judge yields slightly higher MMBench and MMMU scores, whereas the frozen SenseNova-U1 judge yields higher MME, POPE F1, MMVP, GenEval, and DPG; the two tie on MMStar. Figure 8 illustrates a corresponding difference in judgment emphasis: Qwen3.7 Plus identifies a local omission in the text chain, while SenseNova-U1 identifies an image-count violation in the complete trajectory. The external judge therefore changes the resulting capability profile rather than improving it uniformly. Together with the comparison against Random Routing in Table 6, these results support the use of a frozen internal judge as an informative, albeit imperfect, routing signal.

Table 15: Effects of internal and external judges on SenseNova-U1. POPE reports F1.

Judge	MME	MMBench	MMMU MC	MMStar	POPE	MMVP	GenEval	DPG
Base	2282.6	86.1	59.0	65.1	84.0	58.0	85.8	86.1
SenseNova-U1	**2311.6**	87.3	60.6	**66.5**	**86.6**	**62.0**	**89.2**	**87.3**
Qwen3.7 Plus	2304.1	**87.4**	**60.7**	**66.5**	86.3	59.3	88.9	86.9

D The Impact of Freezing Strategy

We examine the role of freezing the judge during online training by comparing Full Prism with a variant in which the judge is updated jointly with the rollout model. The comparison uses the same training setup and tracks two quantities. First, the routing distribution records how the judge assigns generated trajectories to the four actions; the text- and image-chain pass rates are computed from these assignments. Second, the zero-image rate is the proportion of generated interleaved trajectories containing no image output and serves as a measure of image-generation collapse. The jointly updated run terminates after 34 updates, whereas the frozen-judge run completes all 38 updates.

Table 16 shows that the two settings have similar routing distributions in the first stage but subsequently diverge. With the frozen judge, the text-chain pass rate increases from 70.6% to 83.5% and the image-chain pass rate from 60.8% to 66.1%, while all four actions remain active throughout the epoch. The Frozen judge rows reproduce the online training statistics in Table 12. With joint updates, the Pass share rises from 50.6% in the first stage to 87.4% in updates 16–23, while i2t and Fail fall to 4.8% and 1.0%. The increasing concentration in Pass indicates that joint updates substantially change the routing signal.

This concentration does not persist as stable generation. In the final two updates of the jointly updated run, the Pass share falls to 46.0% and t2i rises to 54.0%, alongside a drop in the image-chain pass rate to 46.0%. Table 17 shows that its final zero-image rate reaches 70.59%, compared with 0.00% for the frozen-judge run. Thus, the high intermediate pass rates of the jointly updated judge do not correspond to sustained image-generation quality. In this comparison, freezing the judge maintains a stable routing reference and avoids the observed collapse.

E Implementation of Training Objectives

Equation 3 defines the objective for one interleaved trajectory $\tau = (T_1, I_1, \ldots, T_K, I_K)$, where P is the task prompt, T_i is the i-th text segment, and I_i is the image generated after T_i. Each step follows

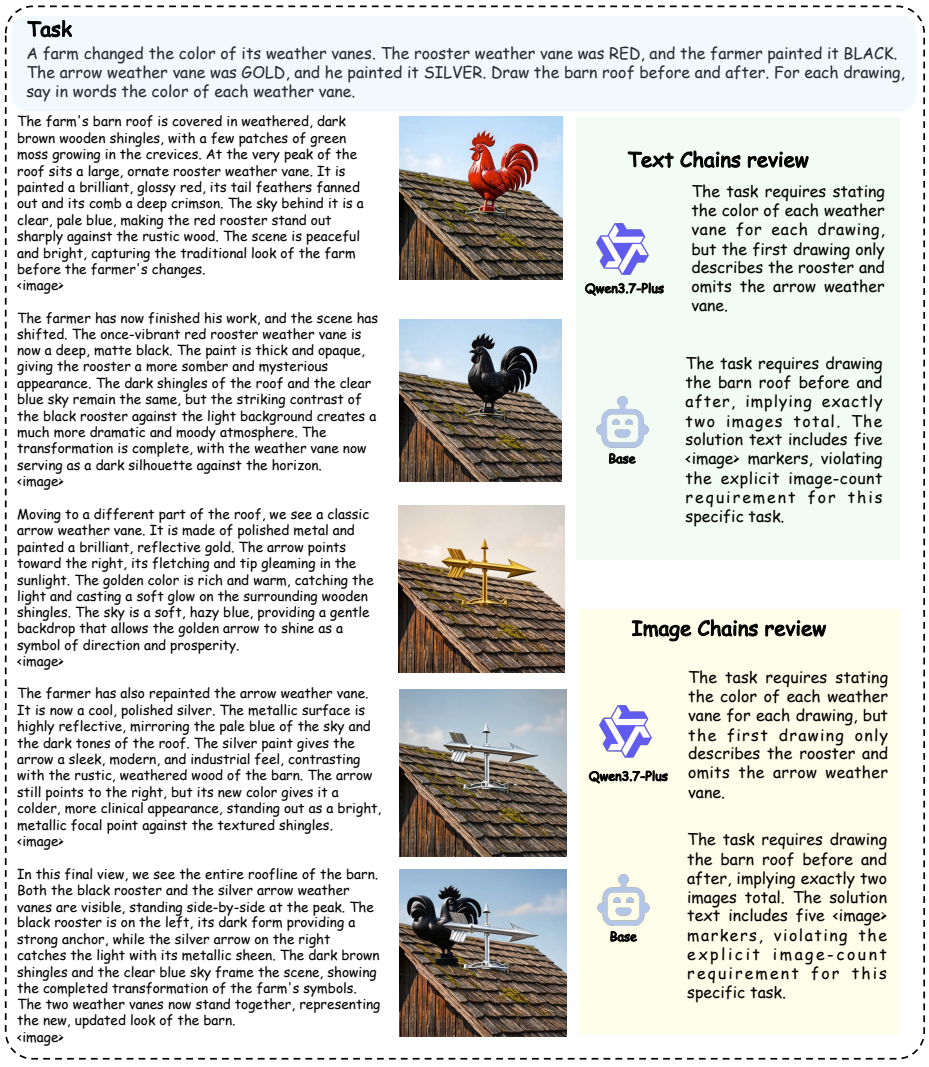


Figure 8: Different judgment emphases of Qwen3.7 Plus and SenseNova-U1. Qwen3.7 Plus identifies a local omission, while SenseNova-U1 identifies an image-count violation in the complete trajectory.

a text-first, then image order. The trajectory-level objective sums a text loss and an image loss over the K interleaved steps.

For the text side, let $T_i = (t_{i,1}, \ldots, t_{i,|T_i|})$ denote the token sequence of the i-th text segment. We use standard token-level next-token cross-entropy:

$$\mathcal{L}_{\text{und}}^{(i)} = -\sum_{n=1}^{|T_i|} \log p_\theta(t_{i,n} \mid P, T_{<i}, I_{<i}, t_{i,<n}), \tag{10}$$

where $T_{<i} = (T_1, \ldots, T_{i-1})$ is the sequence of preceding text segments, $I_{<i} = (I_1, \ldots, I_{i-1})$ is the sequence of preceding images, and $t_{i,<n} = (t_{i,1}, \ldots, t_{i,n-1})$ are the previous tokens in the current text segment.

For the image side, let $z_{i,1}$ denote the clean target visual representation of image I_i under the native image representation of the backbone. This may be a latent code, patch representation, or another backbone-specific continuous representation. Let $z_{i,0} \sim p_0$ denote a noise sample of the same shape drawn from a fixed noise distribution. We sample an interpolation time $s \sim \mathcal{U}[0, 1]$ and define the

Table 16: Routing distributions and judge pass rates across training stages under frozen and jointly updated judges. Text and image pass rates are computed from the corresponding routing categories. All values are percentages. The jointly updated run terminates after 34 updates.

Judge Setting	Updates	Routing Distribution				Judge Pass Rate	
		Pass	t2i	i2t	Fail	Text	Image
Frozen judge (Ours)	0–7	49.3	21.3	11.5	17.9	70.6	60.8
	8–15	51.6	25.1	8.7	14.6	76.7	60.3
	16–23	55.7	25.2	7.7	11.4	80.9	63.4
	24–30	57.6	24.7	7.1	10.6	82.3	64.7
	31–37	59.3	24.2	6.8	9.7	83.5	66.1
Jointly updated judge	0–7	50.6	20.8	13.2	15.4	71.4	63.8
	8–15	76.0	14.4	4.8	4.8	90.4	80.8
	16–23	87.4	6.9	4.8	1.0	94.3	92.2
	24–31	82.2	17.1	0.5	0.2	99.3	82.7
	32–33	46.0	54.0	0.0	0.0	100.0	46.0

Table 17: Effect of freezing the judge on image-generation stability during online training. Lower zero-image rates are better.

Judge Setting	Initial Zero-Image Rate	Final Zero-Image Rate
Frozen judge (Ours)	0.00%	0.00%
Jointly updated judge	0.00%	70.59%

linear flow-matching path

$$z_{i,s} = (1-s)z_{i,0} + sz_{i,1}, \tag{11}$$

with target velocity

$$v_i^\star = z_{i,1} - z_{i,0}. \tag{12}$$

Let

$$c_i = (P, T_{\le i}, I_{<i}), \qquad T_{\le i} = (T_1, \ldots, T_i), \tag{13}$$

denote the conditioning context for generating I_i. The model predicts a velocity field $v_\theta(z_{i,s}, s, c_i)$, and the image-generation loss is

$$\mathcal{L}_{\text{gen}}^{(i)} = \mathbb{E}_{s\sim\mathcal{U}[0,1], z_{i,0}\sim p_0}\left[\|v_\theta(z_{i,s}, s, c_i) - v_i^\star\|_2^2\right]. \tag{14}$$

For the both-pass case, both modalities are judged reliable. We keep the original image chain $I = (I_1, \ldots, I_K)$ and reconstruct the full concatenated text chain $T = (t_1, \ldots, t_{|T|})$, where T is obtained by concatenating all text segments $T_1, \ldots, T_K$. The recovery instruction P_{rec} and image chain jointly condition the reconstruction:

$$\mathcal{L}_{\text{pass}} = -\sum_{n=1}^{|T|} \log p_\theta(t_n \mid P_{\text{rec}}, I, t_{<n}). \tag{15}$$

For the t2i and i2t cases, one modality is corrected and the resulting rewritten trajectory is verified before training. We then apply Equation 3 to the verified corrected trajectory. Thus, every text token is trained with the text loss and every image representation is trained with the image loss.

For the both-fail case, the original trajectory τ is treated as unreliable and is not used as a positive target. Instead, we apply token-level unlikelihood training to the original text chain:

$$\mathcal{L}_{\text{fail}}(\tau) = -\sum_{i=1}^{K}\sum_{n=1}^{|T_i|} \log\left(1 - p_\theta(t_{i,n} \mid P, T_{<i}, I_{<i}, t_{i,<n})\right). \tag{16}$$

In implementation, the argument of the logarithm is numerically clipped away from zero for stability. Each trajectory τ is assigned to exactly one reliability case, pass, t2i, i2t, or fail. During minibatch training, only the objective associated with the assigned case contributes to the loss for that trajectory, while the remaining case-specific objectives are masked out. Eq. 9 denotes the resulting aggregate minibatch objective.

F VERIFICATION OF REWRITTEN TRAJECTORIES

For the t2i and i2t cases, Prism first generates a corrected chain from the preserved reliable chain and the detected error. Text-to-image rewriting generates I^+ from P, T, and r_{img}, while image-to-text rewriting generates T^+ from P, I, and r_{txt}. Each rewritten candidate is verified before it is used as a positive training target. The verifier is a frozen copy of the base model. A candidate is regenerated for at most four attempts, including the initial generation and three retries. Candidates that fail all attempts are discarded.

Image-side verification. For a rewritten image chain I^+ generated from the preserved text chain T, the verifier receives the task prompt P and the candidate image sequence. A candidate is accepted only if it contains the same number of images as the original trajectory, satisfies the task requirements for objects, counts, attributes, spatial relations, state changes, and step order, and is judged valid by the verifier.

```
You are a strict reviewer auditing only the image sequence
of an interleaved solution and its visual state progression.

Step text is not shown. You see {n_images} images in order.

TASK:
{prompt}

Decide whether this image chain delivers the progression
required by the task.

1. Base each decision on visible evidence.
2. To return invalid, cite one concrete image number and its visible error.
3. The first confirmed visible error determines invalid.
4. Determine from the task whether image count is strict or flexible.
5. Check required objects, counts, attributes, spatial relations,
   actions, visual states, lighting, and state progression.
6. A preview or final state is valid only when it does not conflict
   with an explicitly required stage.
7. Do not reject an image chain solely for style drift, texture noise,
   lighting shifts, background changes, or minor visual variation.

Return invalid for a confirmed error such as no task progress,
wrong image or stage count, wrong quantity, failed state change,
state mismatch, corrupted key subject, premature final state,
state regression, identity swap, internally impossible content,
or failure to convey a required step.

Return exactly one JSON object with a Boolean field named "valid".
```

Text-side verification. For a rewritten text chain T^+ generated from the preserved image chain I, the verifier receives the task prompt and candidate text. A candidate is accepted only if the verifier returns valid, its interleaved marker structure is valid, its final image reference is valid, and its length is at least half that of the original rollout text.

```
You are auditing only the visible text of a no-think interleaved
text-and-image response. Images are audited separately.

TASK:
{prompt}

SOLUTION TEXT:
{text_chain}

Judge whether the written content and image marker structure
```

```
satisfy the task.

1. Enforce image-marker count only when the task explicitly specifies it.
2. Required facts, names, labels, directions, emotions, colors, counts,
   and object identities must be stated clearly.
3. The text must cover required objects, stages, relations, or scene changes.
4. Reject only clear text-side contradictions, skipped main events,
   impossible orderings, or unexplained state regression.
5. For exact-one-image tasks, text after the image marker must not
   introduce another scene or visual plan.
6. For tutorials, processes, stories, and lifecycle tasks, check clear
   ordering errors, missing stages, extra stages violating an explicit
   count, or state regression.

Return exactly one JSON object with a Boolean field named "valid".
When invalid, include a verbatim faulty span in the "quote" field.
```

G Additional Understanding Examples

Figure 9 presents additional visual question answering examples. Compared with the base model, Prism gives correct responses on diverse tasks involving food-web reasoning, depth perception, chart interpretation, semantic correspondence, motion direction, spatial attributes, legend understanding, and object counting. These examples further illustrate the improved visual grounding and multi-modal reasoning observed in the quantitative evaluation.

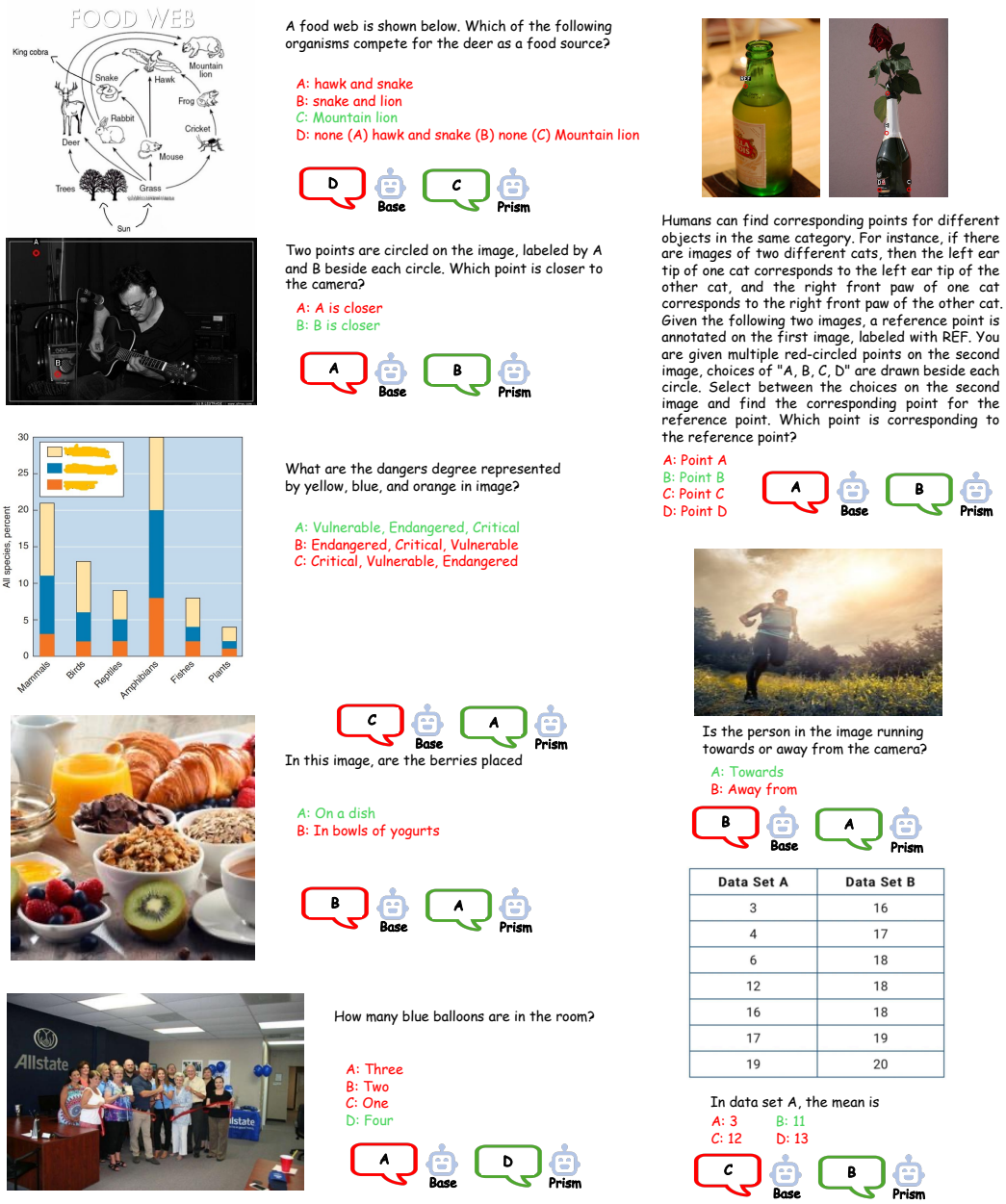


Figure 9: Additional VQA examples. Prism corrects errors made by the base model on questions requiring visual reasoning, chart understanding, correspondence matching, spatial perception, and counting.

H Additional Generation Examples

Figure 10 presents additional generation examples on BAGEL. The examples cover diverse visual styles and contents, including realistic photography, landscape scenes, illustration, portraiture, food, wildlife, and indoor environments.



Figure 10: Additional generation examples on BAGEL. The generated images cover diverse subjects, scenes, and visual styles.

I Additional Interleaved Generation Examples

Figure 11 compares interleaved generation by the base model and Prism. In the picture-book task, the base model repeats similar barn scenes, making the visual progression of the story unclear. For food recommendations, it repeatedly generates a person eating while providing limited visual coverage of the requested dishes. In the solar-eclipse example, the base model produces a long text-dominated response with only one illustration, whose cluttered background does not clearly convey the required Earth–Moon–Sun alignment. In contrast, Prism produces distinct visual states for the story, diverse food images for the recommendation, and stepwise text–image explanations of the eclipse.



Figure 11: Additional interleaved generation examples. Compared with the base model, Prism produces clearer visual progression in a picture-book story, more diverse images for food recommendations, and visual understanding for a solar eclipse.