AUTORUBRIC-T2I: ROBUST RULE-BASED REWARD MODEL FOR TEXT-TO-IMAGE ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Aligning Text-to-Image (T2I) generation models with human preferences increasingly relies on image reward models that score or rank generated images according to prompt alignment and perceptual quality. Existing reward models commonly require training on large-scale human preference corpora. Meanwhile, more recent work leverages Vision-Language Model (VLM) judges with rubrics to make evaluation criteria explicit. However, rubric-based approaches often lack reliable methods for criterion selection. In this paper, we propose **AutoRubric-T2I**, a framework that automatically synthesizes, selects, and weights explicit rubrics to guide VLM judges. AutoRubric-T2I first extracts candidate rubrics from reasoning traces over preference pairs and scores the paired images under each rubric using a frozen VLM judge. Then, an ℓ_1**-Regularized Logistic Regression Refiner** fits these scores to human preference labels, pruning rubrics that fail to guide the judge toward preference-consistent assessments and learning dataset-specific weights for the retained criteria. Using just 256 preference pairs ($\sim$0.02–0.04% of full RM training data) during training, AutoRubric-T2I outperforms fine-tuned reward models on MMRB2. As an RL reward in Flow-GRPO, it also improves generation quality over the fine-tuned reward models on TIIF and UniGenBench++.

1 INTRODUCTION

Recent advancements in T2I generation have made human preference alignment a central objective. Image reward models provide a practical mechanism for this alignment by learning to predict human judgments over generated images. In the T2I setting, an image reward model typically takes a text prompt together with one or more generated images as input and outputs either scalar reward scores or pairwise preferences indicating which image better satisfies the prompt and human quality expectations. These models are widely used for candidate ranking, best-of-N selection, data filtering, reinforcement fine-tuning (RFT), and automatic evaluation of T2I systems Jiang et al. (2024); Liu et al. (2025); Jiang et al. (2025).

Image reward models include learned preference models such as ImageReward Xu et al. (2023), PickScore Kirstain et al. (2023), HPSv2 Wu et al. (2023), and the VLM-based HPSv3 and UnifiedReward Ma et al. (2025); Wang et al. (2025b). Their architectures and training objectives differ: for example, HPSv3 uses uncertainty-aware ranking rather than a conventional BT objective. These models learn useful ranking signals from annotated data, but adapting them generally requires further training. Aggregate preference scores can also obscure fine-grained visual errors such as incorrect object counts, missing attributes, or violated spatial relations Hu et al. (2023); Li et al. (2024); Hong et al. (2026). Complementary approaches use prompted VLM judges or question-answering-based evaluation to expose individual criteria Hu et al. (2023); Li et al. (2024); Hu et al. (2025). Their judgments depend on the evaluation questions and prompting strategy; in our experiments, prompted VLM judges can underperform trained preference models on the in-domain benchmarks in Table 2.

These limitations motivate **rubric-based reward modeling** (Figure 1c), which makes the criteria underlying reward signals explicit Xie et al. (2025); Wang & Xiong (2025); Gunjal et al. (2025); Zhang et al. (2026); Liu et al. (2026). Rubrics make the reward signal more interpretable by decomposing human preferences into explicit rules. Recent works have explored rubrics as reward signals for LLM alignment and post-training Xie et al. (2025); Wang & Xiong (2025); Gunjal et al. (2025); Huang et al. (2025); Xu et al. (2026); Li et al. (2026), and emerging T2I work has begun to study rubric rewards for image generation Feng et al. (2025). Building on this work, we study how to learn a

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

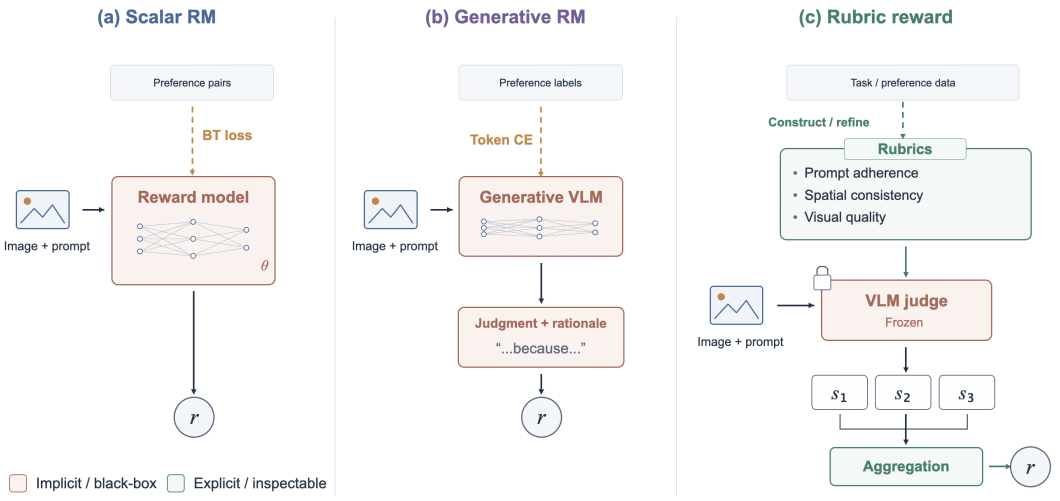

Figure 1: **Three paradigms for T2I reward modeling.** (a) Scalar reward models are trained on preference pairs with a BT-style ranking loss and map a prompt–image pair directly to a reward r. (b) Generative reward models fine-tune a VLM with token-level cross-entropy on preference labels and output a judgment, often with a rationale. In both, preferences are encoded implicitly in the model parameters. (c) Rubric rewards make the criteria explicit: rubrics constructed or refined from preference data guide a frozen VLM judge, whose per-rubric scores s_i are aggregated into r. AutoRubric-T2I follows (c), learning both the rubric set and its aggregation weights from image preference data.

compact global rubric set whose weights and refinement are directly informed by image preference data.

To address this gap, we propose **AutoRubric-T2I**, a framework that automatically derives and refines a sparse, weighted global rubric set for guiding off-the-shelf VLM judges in T2I reward modeling. Without fine-tuning the VLM judge, AutoRubric-T2I learns which rubrics are most predictive of human preferences and iteratively improves them through failure analysis. This design preserves the interpretability of rubric-based evaluation while avoiding the cost and opacity of training a dense scalar reward model.

To achieve this, we formulate automated rubric learning as a sparse logistic regression problem within an infinite-dimensional space. We then introduce an iterative block coordinate descent method that dynamically adds new coordinates to the working set and employs ℓ_1-regularization to assign weights and prune redundant coordinates (rubrics). This formulation is related to sparse function approximation and coordinate-selection methods such as orthogonal matching pursuit and sparse random features Pati et al. (1993); Jain et al. (2011); Yen et al. (2014). To enhance efficiency, we integrate a hard-pair mining algorithm for rubric refinement, ensuring that only the most informative coordinates are prioritized during the learning process.

Our main contributions are as follows:

- **Sparse Rubric Learning for T2I Reward Modeling:** We introduce AutoRubric-T2I, a framework that learns a compact, weighted set of natural-language rubrics from image preference data, enabling interpretable VLM-based reward modeling without fine-tuning.
- **Failure-Driven Rubric Refinement:** We formulate rubric selection as an ℓ_1-regularized logistic regression problem over VLM-scored rubric features and iteratively expand the rubric pool through curriculum-bucketed hard-pair mining.
- **Strong Preference Prediction and Downstream Alignment:** AutoRubric-T2I achieves strong preference prediction on MMRB2 among open-source reward models and improves downstream RFT on T2I tasks such as TIIF and UniGenBench++ using Flow-GRPO.

2 Related Work

2.1 Text-to-Image Preference Alignment and Reward Modeling

Aligning text-to-image (T2I) models with human preferences commonly relies on reward models trained from human preference data. Since such data are mostly collected as pairwise comparisons,

many image reward models are trained with a Bradley-Terry (BT) objective, but are deployed as point-wise scorers that assign a scalar score to each prompt-image pair. Large-scale preference datasets such as Pick-a-Pic Kirstain et al. (2023) and HPD (v2, v3) Wu et al. (2023); Ma et al. (2025), together with reward models such as ImageReward Xu et al. (2023), HPSv2 Wu et al. (2023), HPSv3 Ma et al. (2025), and PickScore Kirstain et al. (2023), enabled automatic ranking of generated images according to human judgments. As vision-language models (VLMs) have grown stronger in multi-modal capability, models along this pathway have shifted from CLIP-based Radford et al. (2021) (PickScore, HPSv2) and BLIP-based Li et al. (2022) (ImageReward) designs toward fine-tuned VLM-based designs (e.g., HPSv3), with substantial gains in preference prediction Ma et al. (2025). This development of VLMs has also given rise to a distinct line of reward modeling, generative reward modeling, which takes different forms such as RewardDance Wu et al. (2025) and UnifiedReward Wang et al. (2025b); this line typically fine-tunes a VLM with a cross-entropy objective on the answer tokens.

However, in both lines of reward modeling above, human preferences are modeled implicitly in the model weights, through either a BT objective or a cross-entropy objective on answers (or similar formulations). Subsequent reward optimization against such models can exploit features that do not reflect the intended quality criteria, and their black-box nature makes such issues difficult to inspect and diagnose. Such phenomenon is supported by prior studies: ODIN studies response-length bias in language-model RLHF Chen et al. (2024); in the visual domain, Hong et al. Hong et al. (2026) analyze reward hacking during T2I reinforcement learning. This motivates departure from the above modeling approaches and address their common problem: the model's representation of human preferences, and the basis of its scores, are difficult to inspect and interpret. One increasingly developed direction toward this goal models human preferences *explicitly* with rubrics, and predicts how well a response satisfies human preferences from how well it satisfies these rubrics (rubric-based evaluation).

2.2 Automated Rubric Generation

Rubric-based evaluation decomposes open-ended human preferences into explicit criteria, improving interpretability over monolithic scalar rewards. However, determining the rubrics is one of the central challenges of this paradigm, since the rubrics and the judge play a decisive role in the resulting evaluation. Some work relies on manually written rubrics Arora et al. (2025); Starace et al. (2025), but constructing them for each task requires substantial human effort; other work proposed automatic rubric generation to reduce the need for manually written evaluation rules. For example, OpenRubrics Liu et al. (2026) derives rubrics by contrasting preferred and rejected responses, while AutoRule Wang & Xiong (2025) uses chain-of-thought prompting over preference examples to extract candidate rules. In addition, Chasing the Tail Zhang et al. (2026), RubricHub Li et al. (2026), Auto-Rubric Xie et al. (2025), and RRD Shen et al. (2026) improve rubric coverage or specificity through refinement, decomposition, or differentiation.

Inspired by these works, we combine the strengths of several of these approaches and, targeting the multimodal setting of image preferences, propose a rubric learning framework for image reward modeling. AutoRubric-T2I learns a sparse, weighted, global set of natural-language rubrics directly from image preference data. Compared with methods that extract, merge, or revise rules via LLM prompting, we further pair curriculum-based hard-pair mining with an ℓ_1-regularized logistic regression refiner. This process prunes the rubric space, retains the Top-N rubrics with the largest learned positive weights, and combines their scores with the learned weights to fit human preferences. Once the rubric set and its weights are learned and fixed, they can be paired with a VLM judge to provide a scalar reward, thereby serving as a reward model for RL.

2.3 Reinforcement Learning from Rubric-Based Rewards

Using rubrics as intermediate reward specifications, as described above, has recently proven effective across several works. Such rubric-based RL can be viewed as an extension to open-ended generation of Reinforcement Learning with Verifiable Rewards (RLVR), which earlier achieved strong results in domains with objective correctness signals, such as mathematics and code generation DeepSeek-AI et al. (2025); Wen et al. (2025). For example, in language model alignment, Rubrics as Rewards Gunjal et al. (2025) converts rubric-based feedback into scalar rewards for RL, while online rubric elicitation Rezaei et al. (2025) updates evaluation criteria based on comparisons between the current and reference policies.

In the T2I setting, DDPO Black et al. (2024) and DanceGRPO Xue et al. (2025) demonstrate RL with scalar rewards. RubricRL Feng et al. (2025) constructs and caches prompt-specific criteria offline before RL; a VLM evaluates image compliance, and binary criterion scores are averaged into a reward. Its criteria cover explicit prompt constraints as well as visual quality and realism. AutoRubric-T2I instead learns a shared global rubric set and preference-discriminative weights from image pairs. The two approaches therefore differ in rubric granularity and learned aggregation. After rubric learning, our fixed weighted set supports evaluation and downstream RFT without fine-tuning the VLM judge.

3 METHODOLOGY

In this section, we introduce **AutoRubric-T2I**. Section 3.1 formulates rubric learning as an infinite-dimensional sparse logistic regression problem and motivates a working-set optimization strategy. Section 3.2 describes the practical implementation, including seed rubric generation, sparse rubric selection, hard-pair mining, and failure-driven rubric refinement.

3.1 FORMULATION

In our framework, each rubric r_j is parameterized by a natural language prompt. To evaluate a specific rubric r_j on an image y conditioned on the input prompt x, we employ a VLM-as-a-judge (e.g., Gemini or the Qwen-3 series) to output a continuous scalar score $s(r_j, x, y) \in [0, 1]$. In practice, the score is the predicted probability of the `yes` token. Thus, $s(r_j, x, y) := P_\theta\big(\text{yes} \mid r_j, x, y\big)$, where P_θ denotes the probability distribution of our VLM-based model.

Our objective is to identify a set of N natural language rubrics, $\mathcal{R} = \{r_1, r_2, \ldots, r_N\}$, and a corresponding set of weights $\mathbf{w} \in \mathbb{R}^N$, such that their weighted combination best explains the observed preference data. The final reward score for a given prompt-image pair (x, y) is defined as:

$$s_{\mathcal{R},\mathbf{w}}(x, y) := \sum\nolimits_{j=1}^{N} w_j s(r_j, x, y).$$

To determine the optimal rubric-weight combination, we leverage a preference dataset $\mathcal{D}_{\text{train}} = \{(x^{(i)}, y_a^{(i)}, y_b^{(i)}, z^{(i)})\}_{i=1}^{M}$ containing M human preference pairs. Here, $x^{(i)}$ is the text prompt, $y_a^{(i)}$ and $y_b^{(i)}$ are two generated images, and $z^{(i)} \in \{1, -1\}$ indicates the user preference (1 if y_a is preferred). Notably, our framework requires only a small amount of data (e.g., $M = 256$). We seek the combination that minimizes the logistic loss:

$$\min_{\mathcal{R},\mathbf{w}} - \sum_{i=1}^{M} \log \sigma \left(z^{(i)} \left(s_{\mathcal{R},\mathbf{w}}(x^{(i)}, y_a^{(i)}) - s_{\mathcal{R},\mathbf{w}}(x^{(i)}, y_b^{(i)}) \right) \right),$$

where σ denotes the sigmoid function.

While optimizing $\mathbf{w}$ is a standard linear logistic regression problem, learning the set $\mathcal{R}$ is inherently intractable. Since the space of possible natural-language rubrics is infinite, we let $J := \{1, 2, \ldots, \infty\}$ denote the indices of all possible rubrics. Selecting the top-N rubrics is equivalent to solving the optimization problem with an ℓ_0 constraint, which we relax using an ℓ_1 penalty:

$$\min_{\mathbf{w}} \lambda \|\mathbf{w}\|_1 - \sum_{i=1}^{M} \log \sigma \left(z^{(i)} \sum_{j \in J} w_j \Delta s_j^{(i)} \right), \tag{1}$$

where $\Delta s_j^{(i)} := s(r_j, x^{(i)}, y_a^{(i)}) - s(r_j, x^{(i)}, y_b^{(i)})$ represents the score differential when applying rubric r_j to the i-th training pair.

We solve this infinite-dimensional sparse recovery problem using a block coordinate descent method. At each iteration t, we generate a finite set of additional candidate rubrics (coordinates) using the current model's failure cases from J, append them to the current working set $\mathcal{R}^t$, and minimize Equation (1) with respect to the current working set of coordinates:

$$\min_{\mathbf{w}_{\mathcal{R}^t}} \lambda \|\mathbf{w}_{\mathcal{R}^t}\|_1 - \sum_{i=1}^{M} \log \sigma \left(z^{(i)} \sum_{j \in \mathcal{R}^t} w_j \Delta s_j^{(i)} \right), \tag{2}$$

where $\mathbf{w}_{\mathcal{R}^t}$ is the finite-dimensional sub-vector of $\mathbf{w}$ corresponding to indices in $\mathcal{R}^t$. Post-optimization, we prune rubrics with zero weights to maintain a compact set.

This block coordinate descent approach has been widely used in sparse recovery problems; for instance, Yen et al. (2014) demonstrated that such algorithms converge when the working set is augmented randomly. Furthermore, our approach draws inspiration from Orthogonal Matching Pursuit (OMP) Pati et al. (1993); Jain et al. (2011), which utilizes greedy strategies to select coordinates. In the following section, we instantiate this idea with a greedy strategy that prioritizes high-impact rubrics generated from hard failure pairs.

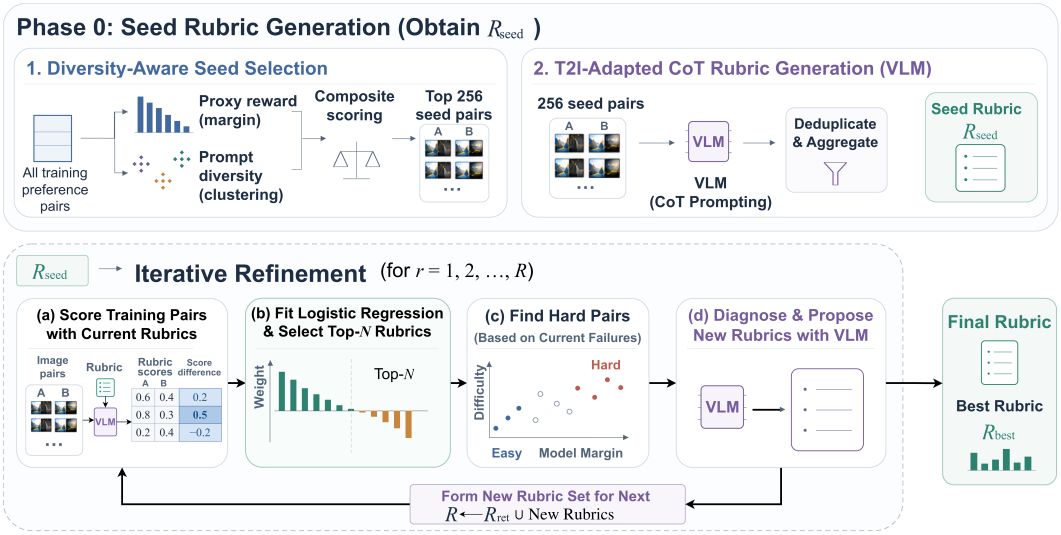

Figure 2: **Overview of AutoRubric-T2I.** Our framework first constructs a seed rubric pool through diversity-aware seed selection and rubric generation. It then iteratively scores training pairs, selects discriminative rubrics with sparse logistic regression, mines hard pairs, and proposes new rubrics to refine the final weighted rubric set.

3.2 Detailed Procedure

We now describe the practical pipeline that instantiates the formulation above. Algorithm 1 and Figure 2 summarize the full procedure. Starting from a seed rubric set $\mathcal{R}^0$, each refinement round scores candidate rubrics with a VLM judge, solves the ℓ_1-regularized problem in Eq. (2), evaluates the retained Top-N rubric set on a validation split, and expands the working set via new rubrics from curriculum-mined hard pairs. We provide the implementation details in the Appendix H.

3.2.1 Seed Data Selection and Initial Rubric Generation

Before iterative refinement begins, we construct an initial working set of rubrics $\mathcal{R}^0$ from an informative seed data $\mathcal{D}_{\text{train}}$. In our default setting, $\mathcal{D}_{\text{train}}$ contains 256 preference pairs.

Diversity-Aware Seed Data Selection. Naively sampling seed preference pairs may over-represent redundant prompts or visually trivial failures. Following FiFA Yang et al. (2024), we use a proxy reward model to estimate the preference margin of each pair and cluster text prompts for semantic coverage. We select 256 seed pairs using a composite score favoring both high-margin preference signals and prompt-level diversity.

T2I-Adapted CoT Rubric Generation. Given the selected seed pairs, we generate the initial candidate rubrics using a VLM-based chain-of-thought prompting procedure adapted to text-to-image evaluation. For each seed pair, the VLM is asked to: (1) inspect the prompt and both images, (2) explain the visual differences that justify the human preference label, and (3) extract objective, deterministic rubric statements that could be reused across examples. The resulting statements are aggregated and deduplicated to form the initial working set $\mathcal{R}^0$.

3.2.2 Working-Set Rubric Scoring and Sparse Selection

At refinement round t, we score all candidate rubrics in $\mathcal{R}^t$ on the training pairs, computing VLM score differences $\Delta s_j^{(i)} = s(r_j, x^{(i)}, y_a^{(i)}) - s(r_j, x^{(i)}, y_b^{(i)})$ as features for Eq. (2). We solve the

ℓ_1-regularized logistic regression over the current working set; the ℓ_1 penalty assigns zero weights to redundant or weakly predictive rubrics. We retain the Top-N rubrics with the largest positive weights: $\mathcal{R}^t_{\text{retained}} = \{r_j \in \mathcal{R}^t : w_j > 0\}_{\text{Top-}N}$, whose weights $\mathbf{w}^t_{\text{retained}}$ define the ensembled rubric reward. We use the `liblinear` solver with $C = 1.0$.

3.2.3 CURRICULUM-BUCKETED HARD-PAIR MINING

After obtaining the retained rubric set, we identify preference pairs that are incorrectly ranked by the current rubric reward. For a pair where y_a is preferred over y_b, the model misranks the pair if

$$\sum\nolimits_{r_j \in \mathcal{R}^t_{\text{retained}}} w_j \Delta s_j^{(i)} < 0.$$

These misranked examples reveal failure modes not yet captured by the current rubric set and serve as the source for generating new candidate rubrics.

Rather than sampling failures uniformly, we introduce a curriculum-bucketed hard-pair selector that partitions misranked pairs into three categories: (1) *wrong-small margin* pairs (below the 30th percentile of absolute margin), which involve subtle distinctions the current rubric set misses; (2) *wrong-large margin* pairs, indicating severe failures where the rubric set confidently contradicts human preference; and (3) *high-reward wrong* pairs, where both images receive high scores yet the ranking is incorrect, requiring finer-grained rubrics similar in spirit to Zhang et al. (2026). Across refinement rounds, we shift the sampling ratio: early rounds emphasize large-margin errors to expose major missing dimensions, while later rounds focus on high-reward wrong cases to discover finer-grained rubrics. Pairs selected more than four times are excluded to avoid noisy or unlearnable examples.

3.2.4 VLM-DRIVEN RUBRIC GENERATION FROM FAILURE CASES

For each sampled hard pair, we generate new candidate rubrics via a two-stage prompting procedure. First, in *failure diagnosis*, the VLM receives the text prompt, both images, and the current rubric set $\mathcal{R}^t_{\text{retained}}$, and diagnoses which missing visual or semantic dimension explains the human preference. Second, in *rubric extraction*, the VLM produces objective, reusable, and visually grounded rubric statements conditioned on the diagnosis. The newly extracted rubrics are deduplicated and appended to form $\mathcal{R}^{t+1} = \mathcal{R}^t_{\text{retained}} \cup \mathcal{R}^t_{\text{new}}$. The next round re-scores this expanded set and re-solves Eq. (2), progressively expanding the rubric space while using the ℓ_1 refiner to maintain a compact, weighted global rubric set.

4 EXPERIMENTS

We evaluate AutoRubric-T2I along two axes: **RQ1:** How does our learned rubric reward compare against fine-tuned RMs and existing rubric baselines on preference benchmarks? **RQ2:** Can the learned rubrics provide a robust signal for downstream T2I-RL?

4.1 EXPERIMENTAL SETUP

Models and Baselines. For VLM judges, we use Qwen3-VL-8B and Qwen3-VL-32B Bai et al. (2025), and Gemini-3-Flash Google DeepMind (2025). Baselines include: the CLIPScore metric Hessel et al. (2021); *learned reward models* ImageReward Xu et al. (2023), PickScore Kirstain et al. (2023), and HPSv2 Wu et al. (2023); *fine-tuned VLM RMs* HPSv3 Ma et al. (2025) and UnifiedReward Wang et al. (2025b) (on Qwen2.5-VL-7B); *zero-shot VLM judges* in pairwise and pointwise modes; and *rubric-based methods* AutoRule Wang & Xiong (2025) and AutoRubric Xie et al. (2025). Note that these methods are originally developed for text, while here we adapted them to T2I. (See the details in Appendix D)

Rubric Generation. We use Gemini-3-Flash to generate reasoning chains for rubric generation, including all rubric-based baselines.

Main Configuration. We use M=256 seed preference pairs and retain N_{rub}=20 rubrics in the main experiments. These settings were fixed before the post-hoc sensitivity sweeps in Section 5.2; the 20-rubric budget matches AutoRule.

Datasets. We evaluate on out-of-distribution (OOD) image generation reward benchmarks including **MMRB2** Hu et al. (2025). We also report in-domain performance on the test splits of **HPDv3** Ma et al.

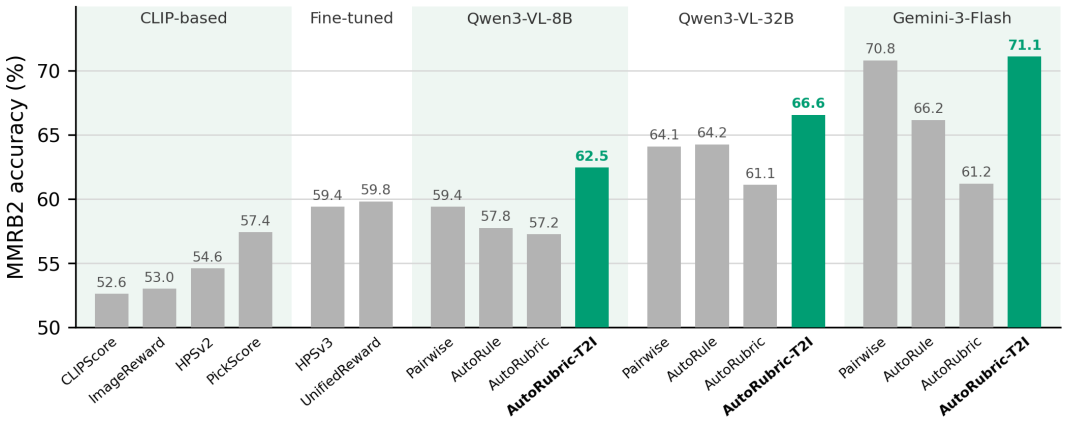

Figure 3: **Preference accuracy on MMRB2 (out-of-domain).** Bars start at chance (50%). AutoRule, AutoRubric, and AutoRubric-T2I use each VLM as a pointwise judge with learned rubrics and are averaged over rubric sets learned from HPDv3 and Pick-a-Pic; Pairwise uses the same VLM as a direct pairwise judge. Fine-tuned RMs are built on Qwen2.5-VL-7B. Per-subset and in-domain results are in Table 2.

(2025) and **Pick-a-Pic** Kirstain et al. (2023) (labeled HPSv3 and PickScore in the RL result tables). For downstream T2I RL, we fine-tune SD-3.5-Medium Esser et al. (2024); Stability AI (2024) using Flow-GRPO Liu et al. (2025) and evaluate on **TIIF** Wei et al. (2025) and **UniGenBench++** Wang et al. (2025a) datasets. (See the details in the Appendix G.)

4.2 Preference Benchmark Evaluation

Figure 3 and Table 2 (Appendix E) report preference-prediction accuracy. On the out-of-domain MMRB2 benchmark, Qwen3-VL-8B obtains 26.5% as a raw pointwise judge and 59.4% as a direct pairwise judge. With AutoRubric-T2I rubrics learned from HPDv3 and Pick-a-Pic, its pointwise accuracy rises to 62.5% and 62.4%, respectively, exceeding AutoRule, AutoRubric, HPSv3 (59.4%), and UnifiedReward (59.8%). Averaged over the two rubric sources, stronger judges reach 66.6% with Qwen3-VL-32B and 71.1% with Gemini-3-Flash, compared with 64.1% and 70.8% for direct pairwise judging.

Fine-tuned reward models remain stronger in-domain: for example, HPSv3 reaches 74.0% on HPDv3, versus 63.9% for AutoRubric-T2I with Qwen3-VL-8B. On MMRB2, however, the learned rubrics improve pointwise VLM judging without updating the judge's parameters. Full in-domain and per-subset results are provided in Table 2.

4.3 Downstream T2I Reinforcement Learning

We fine-tune SD-3.5-Medium Esser et al. (2024); Stability AI (2024) with Flow-GRPO Liu et al. (2025), using AutoRubric-T2I as the reward. A frozen Qwen3-VL-8B judge scores each image under the learned rubrics, and their `yes`-token probabilities are combined using the learned weights. Implementation details are in Appendix B.

Figure 4 shows improvements on both TIIF and UniGenBench++. With HPSv3-derived rubrics, TIIF accuracy rises from 65.3% to 71.6% on short prompts and from 62.7% to 67.9% on long prompts, relative to the base model. On UniGenBench++, the corresponding gains are 61.0% to 62.7% and 64.0% to 66.9%. PickScore-derived rewards show the same overall trend; category-level results are reported in Appendix Tables 3 and 4.

5 Discussion and Analysis

5.1 Ablation Study: Deconstructing AutoRubric-T2I

We ablate the key components of AutoRubric-T2I using Qwen3-VL-8B as the pointwise judge and evaluate on the out-of-domain MMRB2 and the in-domain Pick-a-Pic and HPDv3. Starting from AutoRule Wang & Xiong (2025), we progressively add our proposed components: ℓ_1-based rubric selection, positivity-constrained weights, multi-round refinement, hard-pair evolution, and cluster-based initialization.

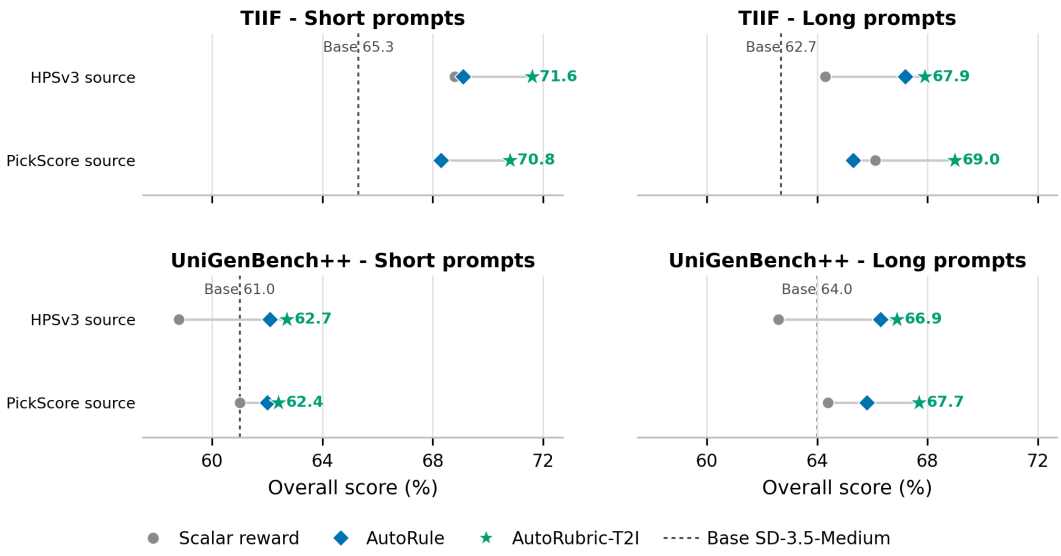

Figure 4: **Overall downstream T2I-RL performance.** Markers compare scalar rewards, AutoRule, and AutoRubric-T2I; dashed lines show the base model. AutoRubric-T2I exceeds the corresponding scalar reward by 1.4–4.3 percentage points across all eight settings. Detailed results are in Appendix F.

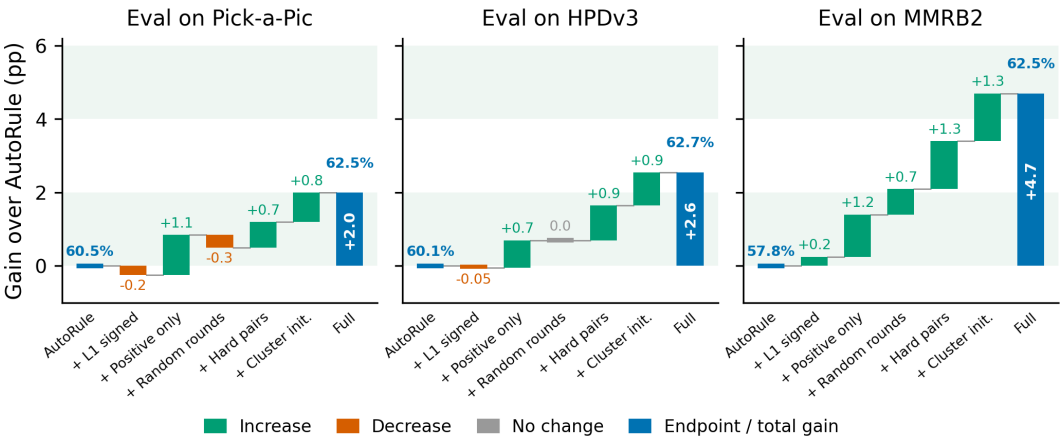

Figure 5: **Ablation study of AutoRubric-T2I.** Starting from AutoRule, we progressively add an ℓ_1 refiner with signed weights, positive-only weights, multi-round refinement with random pairs, hard-pair evolution, and cluster-based initialization. Each panel reports accuracy on one evaluation benchmark using Qwen3-VL-8B as the judge, averaged over rubric sets learned from HPDv3 and Pick-a-Pic. Step bars show the change from the previous configuration; endpoint labels show absolute accuracy.

As shown in Figure 5, simply adding an ℓ_1 refiner with unrestricted weights provides little improvement, since negative weights make the learned rubric set difficult to interpret: satisfying a rubric may decrease the final reward. Enforcing positive-only weights improves performance, suggesting that rubrics are more effective when they serve as additive criteria whose satisfaction consistently indicates higher preference. Multi-round refinement further improves results, but randomly sampled pairs bring only modest gains. In contrast, mining hard pairs from the previous round leads to larger improvements, showing that difficult preference pairs expose missing or ambiguous criteria and guide the model toward more useful rubrics. Finally, cluster-based initialization gives the best overall performance, lifting mean MMRB2 accuracy from 57.8% to 62.5% (+4.7 points), the largest gain among the three benchmarks. This confirms that both the initial rubric coordinates and the subsequent hard-pair refinement trajectory are important for building a robust rubric reward.

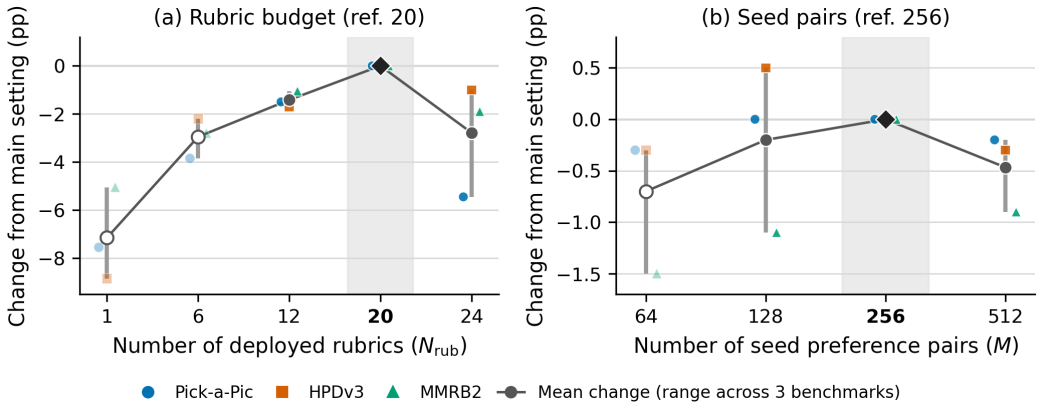

Figure 6: **Sensitivity of the reported configuration.** Changes in preference accuracy are measured relative to the main-experiment settings: (a) N_{rub}=20 deployed rubrics and (b) M=256 seed preference pairs. Large markers show the mean across Pick-a-Pic, HPDv3, and MMRB2; small markers show the individual benchmarks, and vertical bars span their minimum and maximum changes. Shaded bands and diamonds mark the main-experiment settings. The 1- and 6-rubric conditions and the 64-pair condition are lower-budget references; the two panels use different vertical scales.

5.2 Sensitivity of the Reported Configuration

We examine the sensitivity of the configuration used in the main experiments, varying the deployed rubric budget and the number of seed preference pairs while holding the remaining evaluation protocol fixed and using Qwen3-VL-8B as the pointwise judge. The 20-rubric setting matches AutoRule's rubric budget. Relative to this setting, Figure 6a shows a 1.4-point lower mean accuracy with 12 rubrics and a 2.8-point lower mean with 24 rubrics in this experiment. The settings with 1 and 6 rubrics provide lower-budget references rather than local perturbations. Figure 6b shows that mean accuracy changes by less than one percentage point across 64–512 seed pairs; within the nearer 128–512 range, the largest change from the reported M=256 setting is 0.5 points. These sweeps characterize the sensitivity and cost tradeoffs of the reported N_{rub}=20 and M=256 configuration; they were not used to select these settings from the evaluation benchmarks.

5.3 Qualitative and Human Evaluation of the RL Policy

While Section 4.3 provides useful indicators of policy improvement, human perception remains the gold standard for T2I evaluation. Figure 10, 11 qualitatively compares the base model, scalar-reward optimization, AutoRule-based optimization, and AutoRubric-T2I. Scalar rewards and generic rubric rewards can improve visual appeal, but they may still miss fine-grained prompt constraints. In contrast, AutoRubric-T2I better preserves requested objects, relations, and scene structure, suggesting that learned rubrics provide more targeted feedback during RL.

We further conduct a 4-way human evaluation with 30 annotators over 20 prompts. Annotators select the best image among outputs from the base model, scalar reward model, AutoRule reward, and AutoRubric-T2I. AutoRubric-T2I achieves a selection rate of 44.8%, substantially above the random baseline of 25% with a 95% confidence interval of [40.7%, 48.9%]. The improvement is statistically significant ($p < 0.001$), demonstrating that AutoRubric-T2I yields generations that are more frequently preferred by human evaluators. See the details in Appendix N.

6 Conclusion

We introduced **AutoRubric-T2I**, which learns a sparse, weighted set of natural-language rubrics from image preferences without fine-tuning the VLM judge. Its ℓ_1-based selection and hard-pair refinement improve out-of-domain preference prediction on MMRB2, while the learned reward improves downstream Flow-GRPO results on TIIF and UniGenBench++.

GENERATIVE AI USE DISCLOSURE

Generative AI models are also components of the systems studied in this work. In AutoRubric-T2I, Gemini-3-Flash generates candidate rubrics and diagnoses failure cases, while Qwen3-VL and Gemini-3-Flash serve as VLM judges that score rubric satisfaction. We additionally use text-to-image models to generate the images evaluated in our downstream experiments. These uses are part of the methodology described in the paper and are applied in the same manner to the relevant baselines.

Separately, we used generative AI coding assistants to help implement and refactor experiment code and analysis scripts, inspect experimental outputs, and improve the wording and presentation of the manuscript. The assistants also helped identify claims and citations requiring verification; the authors performed the verification against the cited sources and experimental records. Generative AI was not used to originate the paper's central research contribution, including the ℓ_1-regularized sparse rubric-selection mechanism, or to fabricate or alter experimental measurements. The authors reviewed and tested all AI-assisted code, recomputed the reported quantitative results from saved experimental outputs, and reviewed and edited all AI-assisted prose. We take full responsibility for the content of this paper, including all text, claims, code, and artifacts produced with the assistance of generative AI.

REFERENCES

Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, Johannes Heidecke, and Karan Singhal. HealthBench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775*, 2025. URL https://arxiv.org/abs/2505.08775.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/14f75513f0f1ca01de1e826b52e6b840-Abstract-Conference.html.

Lichang Chen, Chen Zhu, Davit Soselia, Jiuhai Chen, Tianyi Zhou, Tom Goldstein, Heng Huang, Mohammad Shoeybi, and Bryan Catanzaro. ODIN: Disentangled reward mitigates hacking in RLHF. *arXiv preprint arXiv:2402.07319*, 2024. URL https://arxiv.org/abs/2402.07319.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948v1.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633, 2024. URL https://proceedings.mlr.press/v235/esser24a.html.

Xuelu Feng, Yunsheng Li, Ziyu Wan, Zixuan Gao, Junsong Yuan, Dongdong Chen, and Chunming Qiao. RubricRL: Simple generalizable rewards for text-to-image generation. *arXiv preprint arXiv:2511.20651*, 2025. URL https://arxiv.org/abs/2511.20651.

Google DeepMind. Gemini 3 flash model card, December 2025. URL https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Flash-Model-Card.pdf. Released December 17, 2025.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *arXiv preprint arXiv:2507.17746*, 2025. URL https://arxiv.org/abs/2507.17746.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7514–7528, 2021. doi: 10.18653/v1/2021.emnlp-main.595. URL https://aclanthology.org/2021.emnlp-main.595/.

Yunqi Hong, Kuei-Chun Kao, Hengguang Zhou, and Cho-Jui Hsieh. Understanding reward hacking in text-to-image reinforcement learning. *arXiv preprint arXiv:2601.03468*, 2026. URL https://arxiv.org/abs/2601.03468.

Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A. Smith. TIFA: Accurate and interpretable text-to-image faithfulness evaluation with question answering. *arXiv preprint arXiv:2303.11897*, 2023. URL https://arxiv.org/abs/2303.11897.

Yushi Hu, Reyhane Askari-Hemmat, Melissa Hall, Emily Dinan, Luke Zettlemoyer, and Marjan Ghazvininejad. Multimodal RewardBench 2: Evaluating omni reward models for interleaved text and image. *arXiv preprint arXiv:2512.16899*, 2025. URL https://arxiv.org/abs/2512.16899.

Zenan Huang, Yihong Zhuang, Guoshan Lu, Zeyu Qin, Haokai Xu, Tianyu Zhao, Ru Peng, Jiaqi Hu, Zhanming Shen, Xiaomeng Hu, et al. Reinforcement learning with rubric anchors. *arXiv preprint arXiv:2508.12790*, 2025. URL https://arxiv.org/abs/2508.12790.

Prateek Jain, Ambuj Tewari, and Inderjit S. Dhillon. Orthogonal matching pursuit with replacement. In *Advances in Neural Information Processing Systems*, volume 24, 2011. URL https://proceedings.neurips.cc/paper/2011/hash/500e75a036dc2d7d2fec5da1b71d36cc-Abstract.html.

Dongfu Jiang, Max Ku, Tianle Li, Yuansheng Ni, Shizhuo Sun, Rongqi Fan, and Wenhu Chen. GenAI Arena: An open evaluation platform for generative models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 79889–79908, 2024. URL https://arxiv.org/abs/2406.04485.

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. *arXiv preprint arXiv:2505.00703*, 2025. URL https://arxiv.org/abs/2505.00703.

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. In *Advances in Neural Information Processing Systems*, volume 36, pp. 36652–36663, 2023. URL https://arxiv.org/abs/2305.01569.

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

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 12888–12900, 2022. URL https://proceedings.mlr.press/v162/li22n.html.

Sunzhu Li, Jiale Zhao, Miteto Wei, Huimin Ren, Yang Zhou, Jingwen Yang, Shunyu Liu, Kaike Zhang, and Wei Chen. RubricHub: A comprehensive and highly discriminative rubric dataset via automated coarse-to-fine generation. *arXiv preprint arXiv:2601.08430*, 2026. URL https://arxiv.org/abs/2601.08430.

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. *arXiv preprint arXiv:2505.05470*, 2025. URL https://arxiv.org/abs/2505.05470.

Tianci Liu, Ran Xu, Tony Yu, Ilgee Hong, Carl Yang, Tuo Zhao, and Haoyu Wang. OpenRubrics: Towards scalable synthetic rubric generation for reward modeling and LLM alignment. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17417–17437, 2026. doi: 10.18653/v1/2026.acl-long.791. URL https://aclanthology.org/2026.acl-long.791/.

Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. HPSv3: Towards wide-spectrum human preference score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15086–15095, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Ma_HPSv3_Towards_Wide-Spectrum_Human_Preference_Score_ICCV_2025_paper.html.

Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar conference on signals, systems and computers*, pp. 40–44. IEEE, 1993. doi: 10.1109/ACSSC.1993.342465. URL https://doi.org/10.1109/ACSSC.1993.342465.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

MohammadHossein Rezaei, Robert Vacareanu, Zihao Wang, Clinton Wang, Bing Liu, Yunzhong He, and Afra Feyza Akyürek. Online rubrics elicitation from pairwise comparisons. *arXiv preprint arXiv:2510.07284*, 2025. URL https://arxiv.org/abs/2510.07284.

William F. Shen, Xinchi Qiu, Chenxi Whitehouse, Lisa Alazraki, Shashwat Goel, Francesco Barbieri, Timon Willi, Akhil Mathur, and Ilias Leontiadis. Rethinking rubric generation for improving LLM judge and reward modeling for open-ended tasks. *arXiv preprint arXiv:2602.05125*, 2026. URL https://arxiv.org/abs/2602.05125.

Stability AI. Stable diffusion 3.5 medium, 2024. URL https://huggingface.co/stabilityai/stable-diffusion-3.5-medium. Model card. Accessed September 16, 2026.

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, Johannes Heidecke, Amelia Glaese, and Tejal Patwardhan. PaperBench: Evaluating AI's ability to replicate AI research. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 56843–56873, 2025. URL https://proceedings.mlr.press/v267/starace25a.html.

Tevin Wang and Chenyan Xiong. AutoRule: Reasoning chain-of-thought extracted rule-based rewards improve preference learning. *arXiv preprint arXiv:2506.15651*, 2025. URL https://arxiv.org/abs/2506.15651.

Yibin Wang, Zhimin Li, Yuhang Zang, Jiazi Bu, Yujie Zhou, Yi Xin, Junjun He, Chunyu Wang, Qinglin Lu, Cheng Jin, et al. UniGenBench++: A unified semantic evaluation benchmark for text-to-image generation. *arXiv preprint arXiv:2510.18701*, 2025a. URL https://arxiv.org/abs/2510.18701.

Yibin Wang, Yuhang Zang, Hao Li, Cheng Jin, and Jiaqi Wang. Unified reward model for multimodal understanding and generation. *arXiv preprint arXiv:2503.05236*, 2025b. URL https://arxiv.org/abs/2503.05236.

Xinyu Wei, Jinrui Zhang, Zeqing Wang, Hongyang Wei, Zhen Guo, and Lei Zhang. TIIF-Bench: How does your T2I model follow your instructions? *arXiv preprint arXiv:2506.02161*, 2025. URL https://arxiv.org/abs/2506.02161v1.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, et al. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base LLMs. *arXiv preprint arXiv:2506.14245*, 2025. URL https://arxiv.org/abs/2506.14245.

Jie Wu, Yu Gao, Zilyu Ye, Ming Li, Liang Li, Hanzhong Guo, Jie Liu, Zeyue Xue, Xiaoxia Hou, Wei Liu, Yan Zeng, and Weilin Huang. RewardDance: Reward scaling in visual generation. *arXiv preprint arXiv:2509.08826*, 2025. URL https://arxiv.org/abs/2509.08826.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023. URL https://arxiv.org/abs/2306.09341.

Lipeng Xie, Sen Huang, Zhuo Zhang, Anni Zou, Yunpeng Zhai, Dingchao Ren, Kezun Zhang, Haoyuan Hu, Boyin Liu, Haoran Chen, et al. Auto-Rubric: Learning from implicit weights to explicit rubrics for reward modeling. *arXiv preprint arXiv:2510.17314*, 2025. URL https://arxiv.org/abs/2510.17314.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. ImageReward: Learning and evaluating human preferences for text-to-image generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 15903–15935, 2023. URL https://arxiv.org/abs/2304.05977.

Ran Xu, Tianci Liu, Zihan Dong, Tony Yu, Ilgee Hong, Carl Yang, Linjun Zhang, Tao Zhao, and Haoyu Wang. Alternating reinforcement learning for rubric-based reward modeling in non-verifiable LLM post-training. *arXiv preprint arXiv:2602.01511*, 2026. URL https://arxiv.org/abs/2602.01511.

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

Yongjin Yang, Sihyeon Kim, Hojung Jung, Sangmin Bae, SangMook Kim, Se-Young Yun, and Kimin Lee. Automated filtering of human feedback data for aligning text-to-image diffusion models. *arXiv preprint arXiv:2410.10166*, 2024. URL https://arxiv.org/abs/2410.10166.

Ian E. H. Yen, Ting-Wei Lin, Shou-De Lin, Pradeep Ravikumar, and Inderjit S Dhillon. Sparse random feature algorithm as coordinate descent in Hilbert space. In *Advances in Neural Information Processing Systems*, volume 27, 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/hash/45ba6620170d8c1b5d582c2c904a8ace-Abstract.html.

Junkai Zhang, Zihao Wang, Lin Gui, Swarnashree Mysore Sathyendra, Jaehwan Jeong, Victor Veitch, Wei Wang, Yunzhong He, Bing Liu, and Lifeng Jin. Chasing the tail: Effective rubric-based reward modeling for large language model post-training. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2509.21500v3.

TABLE OF CONTENTS OF APPENDIX

A AutoRubric-T2I Pipeline Algorithm **15**

B RL Fine-Tuning Setup **15**

C Relation to Prior Rubric-Based Methods **16**

D Baseline Overview **16**

E Full Preference Benchmark Results **17**

F Full Downstream T2I-RL Results **17**

G Dataset and Benchmark Details **19**

H Details of Hyperparameter for AutoRubric-T2I **19**

I Qualitative Examples **20**

J Runtime and Data-needed Analysis **20**

K Training Dynamics **25**

L Limitations and Broader Impact **25**

M Prompt **27**

N Quantitative Human Evaluation **31**

O Optimized Rubric Set **31**

A AutoRubric-T2I Pipeline Algorithm

This is the pseudo-code algorithm for AutoRubric-T2I, which is specifically discussed in Section 3.

Algorithm 1 Working-set refinement pipeline for AutoRubric-T2I.

Require: Train data $\mathcal{D}_{\text{train}}$, valid data $\mathcal{D}_{\text{valid}}$, VLM judge, seed rubrics $\mathcal{R}^0$, Top-N, rounds R, hard-pair K
Init: best_acc $\leftarrow -1$, $\mathcal{R}_{\text{best}} \leftarrow \emptyset$, $\mathbf{w}_{\text{best}} \leftarrow \emptyset$
for $t = 0, 1, \ldots, R-1$ **do**
  Score rubrics in $\mathcal{R}^t$ on $\mathcal{D}_{\text{train}}$ with the VLM judge to obtain $\Delta s_j^{(i)}$
  Solve Eq. (2) over $\mathcal{R}^t$ and retain Top-N positive-weight rubrics $(\mathcal{R}^t_{\text{retained}}, \mathbf{w}^t_{\text{retained}})$
  Evaluate $(\mathcal{R}^t_{\text{retained}}, \mathbf{w}^t_{\text{retained}})$ on $\mathcal{D}_{\text{valid}}$
  if validation accuracy $>$ best_acc **then**
    best_acc $\leftarrow$ validation accuracy
    $(\mathcal{R}_{\text{best}}, \mathbf{w}_{\text{best}}) \leftarrow (\mathcal{R}^t_{\text{retained}}, \mathbf{w}^t_{\text{retained}})$
  end if
  if $t < R-1$ **then**
    Mine K hard pairs misranked by $(\mathcal{R}^t_{\text{retained}}, \mathbf{w}^t_{\text{retained}})$ using curriculum buckets
    Generate new rubrics $\mathcal{R}^t_{\text{new}}$ from hard-pair failure diagnoses
    Update working set: $\mathcal{R}^{t+1} \leftarrow \mathcal{R}^t_{\text{retained}} \cup \mathcal{R}^t_{\text{new}}$
  end if
end for
return $(\mathcal{R}_{\text{best}}, \mathbf{w}_{\text{best}})$

B RL Fine-Tuning Setup

Algorithm overview. Downstream policy optimization follows Flow-GRPO Liu et al. (2025), a GRPO adapted to flow-matching text-to-image samplers. For each prompt p in a mini-batch, the policy generates K rollouts $\{x^{(1)}, \ldots, x^{(K)}\}$ by integrating the flow ODE for T_{train} steps; each rollout receives a scalar reward $r(p, x^{(k)})$ and the group-relative advantage is computed as $A^{(k)} = (r^{(k)} - \mu_p)/(\sigma_p + \varepsilon)$, with μ_p and σ_p the in-group mean and standard deviation. We adopt the public Flow-GRPO implementation without algorithmic modification; only the reward signal is replaced (see below).

Backbones and training data. We fine-tune **Stable Diffusion 3.5-Medium** (SD3.5-M) on the prompt set used by Flow-GRPO Liu et al. (2025), which is itself derived from the Pick-a-Pic prompt distribution Kirstain et al. (2023); this matches the training-prompt setting of prior T2I-RL work and isolates the contribution of the reward signal. No human preference labels from these prompts are used during RL—only the rubric reward defined below.

Integration with AutoRubric-T2I. At each rollout x, we evaluate the retained rubric set $\mathcal{R}^{(\text{final})} = \{(\rho_j, w_j)\}_{j=1}^{N_{\text{rub}}}$, where N_{rub} denotes the number of retained rubrics produced by Section 3.2: for every rule ρ_j, the VLM judge is asked the binary question *"Does this image satisfy this rule?"* and we read the probability of the "Yes" token, $p_j(x, p) = P_{\text{VLM}}(\text{YES} \mid x, p, \rho_j)$. The reward fed to Flow-GRPO is the learned weighted sum

$$r_{\text{AutoRubric}}(x, p) \;=\; \sum_{j=1}^{N_{\text{rub}}} w_j \, p_j(x, p), \tag{3}$$

where w_j are the positive ℓ_1-logistic-regression coefficients fitted on the 256 training preference pairs and held fixed during RL. Since $p_j \in [0, 1]$, each rubric's contribution is inspectable and $0 \le r_{\text{AutoRubric}} \le \sum_j w_j$. We substitute Eq. (3) for the scalar reward while keeping the remaining Flow-GRPO setup unchanged.

Hyperparameters. Table 1 lists the full RL configuration. LoRA rank/α, optimizer, learning rate, group size K=24, effective prompt batch, EMA, and evaluation cadence follow the Flow-GRPO defaults Liu et al. (2025). Training is run on 1×8 H200 GPUs; we report results between 100–500 steps, with checkpoints saved and evaluated every 25 steps.

C Relation to Prior Rubric-Based Methods

AutoRubric-T2I differs from existing rubric-based methods in both rubric granularity and learning mechanism. Unlike prompt-adaptive methods such as RubricRL Feng et al. (2025), which generate a new rubric for each prompt during the RL loop, AutoRubric-T2I learns a compact global rubric set offline from preference data, making the reward signal reusable, cacheable, and consistent across evaluation and RFT. Compared with static rule extraction methods such as AutoRule Wang & Xiong (2025), which primarily use CoT prompting to generate candidate rules, AutoRubric-T2I explicitly optimizes which rubrics should be retained and how they should be weighted. Specifically, rubric selection is formulated as the sparse logistic regression problem in Eq. (2), where VLM-scored rubric features are fitted to human preference labels and the ℓ_1 penalty prunes redundant rules. Finally, our curriculum-bucketed hard-pair mining expands the rubric pool from failure cases of the current retained rubric set, closing the loop between rubric evaluation, failure discovery, and rubric refinement.

	SD3.5-M
Backbone	SD3.5-Medium
Mixed precision	fp16
Resolution	512×512
LoRA target / rank / α	attention Q/K/V/out, r=32, α=64
Optimizer	AdamW
Learning rate	3×10^{-4}
Train timesteps T_{train}	10
Eval timesteps T_{eval}	40
CFG scale	4.5
Rollouts per prompt K	24
Effective prompt batch	48
Train batch size (per GPU)	9
Test batch size	16
Gradient accumulation steps	$N/2$ (auto)
Inner epochs	1
Timestep fraction (PPO mask)	0.99
KL β	0.01
EMA on policy weights	yes
Reported steps	100–500
Eval frequency	every 25 steps

Table 1: Training hyperparameters. K is the number of rollouts per prompt used to compute the group-relative advantage. *Effective prompt batch* is the number of unique prompts processed per outer iteration. *KL* β is the coefficient on the KL-to-reference penalty.

D Baseline Overview

We summarize the baselines. AutoRule Wang & Xiong (2025) and Auto-Rubric Xie et al. (2025) are designed for text-domain preference learning, while RubricRL Feng et al. (2025) targets T2I generation. These methods all replace opaque scalar rewards with explicit evaluation criteria, but differ in how rubrics are generated, selected, and used. We provide quantitative comparisons of data and computational cost in Appendix J.

AutoRule. AutoRule Wang & Xiong (2025) extracts rule-based rewards from LLM reasoning chains in the text domain. It prompts a reasoning model to explain why preferred responses are better, extracts rule-like statements from these explanations, and merges them into a compact rule set. During RL, each response is evaluated against all rules by an LLM verifier, and the rule scores are averaged with uniform weights. However, it does not learn discriminative rule weights or iteratively refine rules on held-out preference errors.

Auto-Rubric. Auto-Rubric Xie et al. (2025) is a training-free text-domain framework that generates candidate rubrics from preference pairs and revises them when their judgments disagree with ground-truth labels. It then compresses the resulting rubric pool by selecting semantically diverse rubrics in embedding space and applies the final set through majority voting with an LLM

judge. Unlike AutoRubric-T2I, its selection criterion emphasizes semantic diversity rather than preference-discriminative power, and it is not designed for visual preference evaluation.

RubricRL. RubricRL Feng et al. (2025) is a concurrent T2I method that generates prompt-specific visual checklists directly from each input prompt. For example, a prompt mentioning three red apples may induce rubrics about object count, color, and placement. Each image is scored by a VLM judge against these generated criteria, and the binary scores are uniformly averaged as the reward for GRPO. RubricRL therefore depends on per-prompt rubric generation, uses uniform weights, and mainly captures constraints explicitly stated in the prompt, rather than learning a global preference-aligned rubric set from human preference data.

Fine-tuned reward models. ImageReward Xu et al. (2023), HPSv2/HPSv3 Wu et al. (2023); Ma et al. (2025), PickScore Kirstain et al. (2023), and UnifiedReward Wang et al. (2025b) are learned reward models trained on large-scale human preference data. They map a prompt-image pair to a scalar reward and are widely used for ranking, filtering, and RFT. However, because they compress multiple preference dimensions into one opaque score, they provide limited interpretability and can be vulnerable to reward hacking.

Zero-shot VLM judge. We also compare with zero-shot VLM judging, where a model such as Qwen3-VL is directly prompted to evaluate an image given a text prompt, without learned rubrics or preference-specific calibration. This baseline requires no reward-model training or rubric construction, but its criteria remain implicit and can be inconsistent. AutoRubric-T2I instead makes these criteria explicit and learns which rubrics are most predictive of human preferences.

E Full Preference Benchmark Results

Table 2 reports the per-subset MMRB2 and in-domain results summarized in Figure 3, separately for each rubric source.

F Full Downstream T2I-RL Results

Tables 3 and 4 report the category-level TIIF and UniGenBench++ results summarized in Figure 4. For each benchmark, we report both `short` and `long` prompt variants and compare the base SD-3.5-Medium model with policies optimized using scalar rewards, AutoRule rewards, and AutoRubric-T2I rewards derived from HPSv3 or PickScore.

	MMRB2 (Out-of-domain)						In-domain	
Model	**EvalMuse**	**OneIG-Bench**	**R2I-Bench**	**RealUnify**	**WISE**	**Overall**	**Pick-a-Pic**	**HPDv3**
VLM-as-a-judge (Pairwise)								
Qwen3-VL-8B	59.2	62.8	57.4	62.4	50.9	59.4	62.2	60.4
Qwen3-VL-32B	63.3	68.3	60.9	65.6	59.0	64.1	63.8	62.6
Gemini-3-Flash	69.8	73.2	70.7	78.5	65.0	70.8	70.3	65.6
VLM-as-a-judge (Pointwise)								
Qwen3-VL-8B	29.8 (60.3)	27.0 (58.5)	25.8 (56.3)	23.1 (55.5)	17.7 (54.1)	26.5 (58.0)	35.4 (58.8)	24.6 (55.6)
Qwen3-VL-32B	25.9 (58.7)	21.6 (57.6)	20.3 (55.5)	21.5 (53.8)	15.3 (53.2)	22.4 (56.9)	40.1 (60.3)	28.7 (57.5)
Gemini-3-Flash	36.7 (64.3)	38.4 (63.8)	38.6 (63.4)	34.1 (61.8)	30.9 (61.8)	36.6 (62.1)	52.1 (66.6)	42.5 (61.4)
CLIP-Based								
CLIPScore	54.4	51.1	53.1	46.2	55.0	52.6	60.8	48.6
ImageReward	51.0	51.4	58.6	52.7	57.7	53.0	61.1	58.6
HPSv2	51.0	53.6	63.3	60.2	55.0	54.6	70.5	65.6
PickScore	57.2	54.3	67.2	53.8	57.7	57.4	63.8	65.3
Fine-Tuned (Qwen2.5-VL-7B)								
HPSv3	54.0	60.4	68.0	68.9	56.7	59.4	67.3	74.0
UnifiedReward	56.9	62.1	56.8	67.8	56.0	59.8	68.8	65.8
VLM-as-a-judge (Pointwise) Qwen3-VL-8B								
AutoRule (on HPDv3)	56.9	60.4	64.1	62.4	55.0	59.1	61.1	62.8
AutoRule (on Pick-a-Pic)	50.8	61.5	60.2	58.1	57.7	56.4	59.8	57.5
AutoRubric (on HPDv3)	54.1	61.9	61.7	59.1	52.3	57.5	57.1	58.1
AutoRubric (on Pick-a-Pic)	53.1	56.1	59.4	59.1	**68.5**	57.0	54.8	57.1
AutoRubric-T2I (on HPDv3)	58.5	**67.3**	64.1	**65.6**	60.4	**62.5**	61.7	**63.9**
AutoRubric-T2I (on Pick-a-Pic)	**60.8**	66.2	**65.6**	61.3	55.9	62.4	**63.2**	61.5
VLM-as-a-judge (Pointwise) Qwen3-VL-32B								
AutoRule (on HPDv3)	64.1	68.0	64.8	64.5	61.3	65.0	60.5	62.7
AutoRule (on Pick-a-Pic)	66.2	65.8	60.9	66.7	57.7	63.5	56.6	60.4
AutoRubric (on HPDv3)	61.3	64.4	67.2	66.7	62.2	63.4	62.5	60.6
AutoRubric (on Pick-a-Pic)	60.0	57.9	57.0	59.1	58.6	58.8	58.8	56.7
AutoRubric-T2I (on HPDv3)	64.2	67.3	**68.4**	69.1	**64.5**	65.4	63.1	**65.1**
AutoRubric-T2I (on Pick-a-Pic)	**66.9**	**68.8**	67.8	**73.4**	62.4	**67.7**	**64.9**	62.5
VLM-as-a-judge (Pointwise) Gemini-3-Flash								
AutoRule (on HPDv3)	67.7	68.4	67.2	72.0	62.2	67.6	67.2	62.6
AutoRule (on Pick-a-Pic)	66.4	65.8	62.5	68.8	55.0	64.7	68.3	59.4
AutoRubric (on HPDv3)	61.2	62.8	66.6	74.2	58.7	63.8	65.0	62.2
AutoRubric (on Pick-a-Pic)	58.7	57.1	60.2	58.1	58.6	58.6	59.9	56.2
AutoRubric-T2I (on HPDv3)	70.2	71.3	70.8	78.7	64.9	70.8	69.0	**70.0**
AutoRubric-T2I (on Pick-a-Pic)	**70.7**	**71.9**	**71.1**	**79.5**	**66.0**	**71.4**	**70.3**	66.8

Table 2: Full comparison across MMRB2 out-of-domain and in-domain benchmarks. Within each VLM pointwise block, **bold** and underline indicate the best and second-best scores, respectively. Parentheses report tie-adjusted scores: $(\#\text{wins} + 0.5 * \#\text{ties})/\#\text{total}$.

Model	**Attr.**	**Relation**	**Reason.**	**Attr.+Rel.**	**Attr.+Reason.**	**Rel.+Reason.**	**Text Gen.**	**Style**	**Real Complex**	**Overall**
TIIF (`short`)										
SD-3.5-Medium	78.0	76.6	66.8	64.8	60.4	59.6	45.7	70.0	66.0	65.3
+ HPSv3	83.5	75.5	65.8	64.5	57.4	60.7	62.0	76.7	73.5	68.8
+ AutoRule (HPSv3)	**85.0**	72.4	70.0	**68.2**	**58.8**	57.5	55.2	**80.0**	**74.6**	69.1
+ AutoRubric-T2I (HPSv3)	84.5	**76.2**	**74.6**	67.3	58.3	**67.1**	**65.2**	77.3	73.8	**71.6**
+ PickScore	82.5	77.0	**76.2**	67.6	56.4	56.1	61.1	66.7	70.9	68.3
+ AutoRule (PickScore)	**85.5**	74.2	65.8	**68.1**	60.2	**59.2**	64.2	66.7	70.5	68.3
+ AutoRubric-T2I (PickScore)	84.0	**77.2**	74.6	67.8	**61.8**	58.0	**69.7**	**70.0**	**74.2**	**70.8**
TIIF (`long`)										
SD-3.5-Medium	75.0	73.5	66.5	62.9	60.9	57.5	30.3	63.3	74.2	62.7
+ HPSv3	78.0	**70.9**	**70.0**	**71.2**	55.6	58.4	**50.7**	56.7	67.2	64.3
+ AutoRule (HPSv3)	**79.5**	69.9	69.7	64.4	**65.2**	58.5	49.8	**73.3**	74.2	67.2
+ AutoRubric-T2I (HPSv3)	78.5	69.8	69.6	69.6	64.0	**61.4**	49.8	**73.3**	**75.4**	**67.9**
+ PickScore	**80.5**	**76.1**	66.4	67.7	57.3	61.9	48.4	60.0	76.9	66.1
+ AutoRule (PickScore)	**80.5**	74.7	64.3	**69.7**	58.6	63.8	43.9	63.3	68.7	65.3
+ AutoRubric-T2I (PickScore)	79.5	73.8	**71.6**	68.8	**63.8**	**64.8**	**54.8**	**66.7**	**77.6**	**69.0**

Table 3: Evaluation results for T2I RL on TIIF. We report results on both `short` and `long` prompts, comparing scalar reward models, AutoRule rewards, and AutoRubric-T2I on Qwen3-VL-8B.

Model	Style	World Know.	Attribute	Action	Relation	Compound	Grammar	Logic	Layout	Text	Overall
UniGenBench++ (short)											
SD-3.5-Medium	87.6	87.0	70.1	61.1	66.8	57.3	62.3	28.9	73.1	15.5	61.0
+ HPSv3	81.2	88.6	67.3	55.5	67.8	54.4	53.2	29.8	72.8	**17.2**	58.8
+ AutoRule (HPSv3)	**90.2**	**89.2**	**72.7**	59.9	71.3	**60.9**	**60.4**	31.2	74.6	10.3	62.1
+ AutoRubric-T2I (HPSv3)	90.0	88.6	70.5	**62.6**	**71.8**	58.5	58.6	**32.6**	**76.9**	16.7	**62.7**
+ PickScore	84.8	**89.6**	70.5	59.9	69.0	55.2	60.4	30.7	**75.8**	13.8	61.0
+ AutoRule (PickScore)	88.2	88.3	68.0	61.5	68.8	59.3	60.4	**33.0**	75.4	**16.7**	62.0
+ AutoRubric-T2I (PickScore)	**90.4**	88.5	**71.6**	**61.8**	**70.1**	**59.6**	**61.2**	32.2	74.6	14.1	**62.4**
UniGenBench++ (long)											
SD-3.5-Medium	91.7	88.4	80.4	58.6	70.4	62.2	63.2	41.7	74.6	9.2	64.0
+ HPSv3	82.7	89.6	79.6	57.6	69.2	60.1	60.7	42.7	73.3	10.3	62.6
+ AutoRule (HPSv3)	**91.4**	**89.9**	**82.3**	**62.3**	70.6	66.3	64.5	**52.0**	75.6	8.7	66.3
+ AutoRubric-T2I (HPSv3)	90.7	89.3	82.2	61.5	**71.0**	**66.9**	**66.8**	50.1	**78.3**	**12.5**	**66.9**
+ PickScore	86.7	87.9	81.1	59.9	69.3	64.7	63.5	45.1	75.1	10.3	64.4
+ AutoRule (PickScore)	89.9	89.9	81.6	61.8	70.0	66.5	**65.5**	45.6	**77.8**	9.2	65.8
+ AutoRubric-T2I (PickScore)	**93.5**	**90.2**	**82.2**	**63.7**	**72.2**	**70.1**	64.4	**52.9**	76.6	**10.9**	**67.7**

Table 4: Evaluation results for T2I RL on UniGenBench++. We report results on both `short` and `long` prompts, comparing scalar reward models, AutoRule rewards, and AutoRubric-T2I on Qwen3-VL-8B.

G Dataset and Benchmark Details

Training preference corpora. We instantiate AutoRubric-T2I on two human-preference corpora used independently as source distributions: **HPDv3** Ma et al. (2025) and **Pick-a-Pic** Kirstain et al. (2023). From each corpus, we select only $|\mathcal{D}_{\text{train}}|$=256 preference pairs for seed-rubric generation and an additional held-out split for validation (used to monitor the best Top-N_{rub} rule set across refinement rounds).

Diversity-aware seed selection. The 256 seed pairs are not sampled uniformly. As described in Section 3.2 of the main paper, we follow a FiFA-inspired Yang et al. (2024) two-factor selection that combines (i) a *preference-margin* signal from a proxy reward model and (ii) a *prompt-diversity* signal obtained by clustering the text prompts. This design directly follows the ablation reported in Section 5 of the main paper, which shows that swapping cluster-based diversity-aware selection for AutoRule-style random sampling drops MMRB2 accuracy from 62.4% to 60.3% on HPDv3 (and produces a similar gap on Pick-a-Pic), confirming that the clustering step is responsible for a measurable share of the final rubric quality rather than being a cosmetic preprocessing step.

Evaluation preference benchmarks. For each source corpus, we fit rubric weights on 256 preference pairs and select the refinement round using 256 labeled pairs from the other source corpus, summarized in Table 5. For **HPDv3** Ma et al. (2025) and **Pick-a-Pic** Kirstain et al. (2023), we use the *official* test splits released with each corpus (no overlap with the 256 seed pairs). **MMRB2** (Multimodal RewardBench 2) Hu et al. (2025) is a recent omni reward-model benchmark covering four subtasks—*text-to-image*, *image editing*, *interleaved generation*, and *multimodal reasoning* ("thinking-with-images")—with 1,000 expert-annotated preference pairs per subtask drawn from 23 frontier models across 21 source tasks.

Generative T2I Benchmarks. For T2I generative quality assessment on RL post-training, we evaluate on two benchmarks: **TIIF** Wei et al. (2025) and **UniGenBench++** Wang et al. (2025a). For TIIF, we evaluate instruction fidelity across categories, providing a graded measure of instruction-following capacity. We report both `long` and `short` prompt variants evaluation. For UniGenBench++, we probe semantic consistency with both `long` and `short` prompt variants, measuring coherence with brief versus detailed textual descriptions.

H Details of Hyperparameter for AutoRubric-T2I

This appendix consolidates every non-trivial hyperparameter used in the AutoRubric-T2I pipeline reported in the main paper. The same configuration is used for both source corpora (HPDv3 and Pick-a-Pic).

Models. The VLM judge that scores (image, prompt, rubric) triples is `Qwen3-VL-8B-Instruct`, served via vLLM. The rule generator (seed CoT vision rea-

Benchmark	# Samples
Pick-a-Pic (official test) Kirstain et al. (2023)	12,864
HPDv3 (official test) Ma et al. (2025)	14,401
MMRB2 (T2I subtask) Hu et al. (2025)	1,000

Table 5: **Evaluation benchmarks.** Cases denote the number of preference pairs used at test time. HPDv3 and Pick-a-Pic use official test splits.

soner, rule extractor/merger, and hard-pair diagnoser) uses Gemini-3-Flash Google DeepMind (2025) with `temperature`=0.1 and `thinking_budget`=1024. The VLM judge is queried with `temperature`=0.0 and at most 16 output tokens.

Refinement loop. For our best reported runs on both HPDv3 and Pick-a-Pic, we run the iterative pipeline of Section 3.2 for R=10 rounds on $|\mathcal{D}_{\text{train}}|$=256 preference pairs and retain the Top-N_{rub} rules with N_{rub}=20. Each round mines 16 hard pairs and discards any pair that has been selected more than 4 times across rounds (stale-pair cap), preventing degenerate loops on label noise. Validation uses the same source corpus's held-out split.

Sparse rubric selection. The ℓ_1-regularized logistic regression in Eq. (2) is fit with scikit-learn's `liblinear` solver: penalty `l1`, regularization strength C=1.0, no intercept, and a fixed random state of 42. Coefficients with magnitude below 10^{-4} are treated as zero; positive coefficients are sorted in descending order and the top N_{rub}=20 define $\mathcal{R}^{(t)}_{\text{retained}}$.

Curriculum-bucketed hard-pair mining. Misranked pairs under the current retained rubric set are partitioned into the three buckets defined in Section 3.2—*small-margin*, *large-margin*, and *high-reward wrong*—using a margin percentile of 0.3 and a reward quantile of 0.7. The 16 hard pairs sampled in round r are drawn from these buckets with phase-dependent weights $(w_{\text{small}}, w_{\text{large}}, w_{\text{high}})$:

$$(w_{\text{small}}, w_{\text{large}}, w_{\text{high}}) = \begin{cases} (0.6,\, 0.4,\, 0.0) & r < 3 \quad \text{(early)} \\ (0.5,\, 0.3,\, 0.2) & 3 \le r < R{-}1 \quad \text{(mid)} \\ (0.3,\, 0.3,\, 0.4) & r \ge R{-}1 \quad \text{(late).} \end{cases}$$

Early rounds prioritize small- and large-margin errors, with the largest share assigned to small-margin cases. Later rounds allocate more samples to high-reward wrong cases.

Summary table. Table 6 consolidates the values above for reproducibility.

I Qualitative Examples

We present qualitative examples of images generated by models fine-tuned with AutoRubric-T2I rewards via the Flow-GRPO pipeline. Figure 10 and 11 show representative prompt–image pairs from UniGenBench++ and TIIF, illustrating improvements in prompt faithfulness, compositional accuracy, and visual quality compared to the base model.

J Runtime and Data-needed Analysis

A key motivation of **AutoRubric-T2I** is to reduce the data and training cost of conventional reward modeling. CLIP-based reward models such as HPSv2 Wu et al. (2023) and PickScore Kirstain et al. (2023) require 137K–798K human preference pairs and multi-GPU training. Recent VLM-based reward models Wang et al. (2025b) reduce the annotation requirement but still require gradient-based training on 10K–100K+ preference pairs. In contrast, AutoRubric-T2I uses only **256 preference pairs** and requires **no neural reward model training**. Its main cost is a one-time rubric refinement stage.

One-time rubric refinement cost. Our final pipeline runs for 10 refinement rounds. In each round, we mine 16 hard preference pairs and use them to generate new candidate rubrics. The VLM scoring is performed with Qwen3-8B **?** served by vLLM on 4× NVIDIA A6000 GPUs. Rule generation uses the Gemini API Google DeepMind (2025) with `temperature=0.1` and

Component	Hyperparameter	Value
Models	VLM judge backbone	Qwen3-VL-8B-Instruct
	Rule generator	Gemini-3-Flash
Sampling	VLM judge temperature	0.0
	VLM judge max output tokens	16
	Rule-generator temperature	0.1
	Rule-generator thinking budget	1024
Refinement loop	# rounds R	10
	# seed pairs $\lvert\mathcal{D}_{\text{train}}\rvert$	256
	Retained rules per round (Top-N_{rub})	20
	Hard pairs per round	16
ℓ_1 logistic regression	Solver	liblinear
	Regularization strength C	1.0
	Intercept	disabled
	Non-zero coefficient threshold	10^{-4}
Hard-pair mining	Margin percentile (small/large split)	0.3
	Reward quantile (high-reward bucket)	0.7
	Stale-pair cap (max repeats)	4

Table 6: **Hyperparameters of the AutoRubric-T2I pipeline** used to produce the main results on HPDv3 and Pick-a-Pic. Values are shared across both source corpora.

`thinking_budget=1024`. The ℓ_1-regularized logistic regression for rubric selection runs on CPU and takes less than one second per round.

Table 7 summarizes the wall-clock cost. Since the final setting keeps N_{rub}=20 rubrics and uses 10 rounds, the total number of VLM scoring calls is substantially smaller than earlier settings. The entire refinement stage finishes in approximately **2–4 hours on 4× A6000 GPUs**. This cost is incurred only once and the resulting weighted rubric set can be reused for downstream evaluation and RL.

Component	Hardware	Est. Time
VLM scoring (∼180K calls)	Qwen3-8B / vLLM on 4× A6000	1.5–3 hours
Validation scoring (∼20K–40K calls)	Qwen3-8B / vLLM on 4× A6000	15–30 min
Rule generation (320 calls)	Gemini API	∼15–30 min
Logistic regression (10 fits)	CPU	<1 min
Total		**2–4 hours**

Table 7: **Wall-clock time breakdown for AutoRubric-T2I rubric refinement** under the final setting: 10 rounds, 256 preference pairs, N_{rub}=20 retained rubrics, and 16 hard pairs per round.

Cost during RL. Flow-GRPO scores every rollout once per iteration, so the reward sits inside the training loop. We quantify its cost in three ways: (i) measured throughput against HPSv3 on identical hardware and inputs, (ii) the reward's share of the compute of one Flow-GRPO iteration, checked against (i), and (iii) the measured iteration time of our RL runs.

Shared-prefix scoring. The N_{rub} judge queries for an image differ only in the rubric: the image, the instructions and the text prompt come first and are identical. With vLLM prefix caching, the vision encoder and this shared prefix run once per image, and each rubric adds only its own tokens (the rubric text and the fixed question after it, ≈146 tokens). The prompts are byte-identical to independent per-rubric calls, but the cost per image is $a + b\,N_{\text{rub}}$ instead of N_{rub} full VLM calls.

Measured throughput against HPSv3. Table 8 compares the judge with HPSv3 Ma et al. (2025), a Qwen2-VL-7B preference reward model in the same size class as our Qwen3-VL-8B judge. Both run on the same 8× RTX A6000 GPUs over the same 1,000 MMRB2 text-to-image images (500 random pairs). Times run from image files to scores, so decoding and preprocessing are included and

engine start-up is not; every engine is first warmed on a disjoint set of MMRB2 images. HPSv3 is measured twice: as released (HF Transformers with FlashAttention-2, best batch size), and ported to the same vLLM engine as the judge. The port reproduces the released scores (Pearson r=0.9996; 97–98% of pairwise decisions agree). HPSv3 resizes every image to 448×448, so we run the judge both on 448×448 copies and at the native 1024×1024.

The judge's cost is linear in the number of rubrics: $6.4 + 2.56\,N_{\mathrm{rub}}$ ms per image at 448^2 and $27.8 + 2.67\,N_{\mathrm{rub}}$ ms at 1024^2 on 8 GPUs. Each additional rubric costs 16–17% of one HPSv3 call. At HPSv3's resolution the judge breaks even with HPSv3 at $N_{\mathrm{rub}}\approx3.7$. Four rubrics cost $1.07\times$ HPSv3 on the same engine and $0.68\times$ the released HPSv3; the full N_{rub}=20 set costs $3.7\times$ HPSv3.

	448×448 **inputs**		1024×1024 **inputs**	
Reward	**ms / image**	**vs. HPSv3**	**ms / image**	**vs. HPSv3**
HPSv3, same vLLM engine	15.7 (reference)			
HPSv3, released code	25.0 (1.59×)			
AutoRubric-T2I, N_{rub}=1	9.0	0.57×	30.7	1.95×
AutoRubric-T2I, N_{rub}=4	16.9	1.07×	38.8	2.47×
AutoRubric-T2I, N_{rub}=8	26.7	1.70×	48.6	3.09×
AutoRubric-T2I, N_{rub}=12	36.9	2.34×	60.1	3.82×
AutoRubric-T2I, N_{rub}=16	46.9	2.98×	69.7	4.43×
AutoRubric-T2I, N_{rub}=20	58.1	3.69×	81.9	5.21×

Table 8: **Measured reward cost.** Wall-clock milliseconds per image on 8× RTX A6000 (the inverse of throughput), on the same 1,000 MMRB2 images with warm engines. "vs. HPSv3" is relative to HPSv3 on the same vLLM engine. HPSv3 resizes every input to 448×448, so one measurement covers both input sizes. The judge uses Qwen3-VL-8B with shared-prefix scoring.

Share of a Flow-GRPO iteration. A per-image comparison alone does not show whether the reward matters in RL, where the policy's own sampling and update also cost compute. Figure 7 therefore counts the FLOPs spent per rollout image in one Flow-GRPO iteration with the settings of Table 1.

- *Policy forward pass.* SD3.5-M is an MMDiT-X with 24 blocks of width 1536, in which image and text tokens use separate weights. At 512×512 a forward pass processes 1,024 latent tokens and 205 text tokens (77 CLIP + 128 T5) and costs 2.2 TFLOP.
- *Rollout sampling.* 10 steps with classifier-free guidance, i.e., 20 forward passes: 44 TFLOP.
- *Policy update.* $\lfloor 10\times0.99\rfloor$=9 timesteps, each with a guided policy forward pass, a backward pass to the LoRA adapters through the frozen backbone, and a guided reference forward pass for the KL term: 125 TFLOP.
- *Rewards.* 2×parameters×tokens plus attention, using the token counts measured by the vLLM engine. A 512^2 rollout has 60 more image tokens than a 448^2 image.

We leave out VAE decoding, the text encoders, the optimizer, EMA and communication. Including them would only lower the reward's share.

This accounting reproduces the measured costs. The ratio of judge to HPSv3 FLOPs matches the ratio of their measured times in Table 8 within 4% at every N_{rub} (e.g., $3.78\times$ predicted vs. $3.69\times$ measured at N_{rub}=20). By compute, the full rubric reward is 21.7% of an iteration, against 6.7% for HPSv3, so an iteration costs 19% more than with HPSv3. With four rubrics the difference is 1%.

Measured iteration time. In our matched Flow-GRPO runs on H200 GPUs, one iteration takes about 90–91 s with the N_{rub}=20 rubric reward, against 75 s with PickScore and 77 s with HPSv3. That is 18% more than with HPSv3 and 20% more than with PickScore, where the compute accounting predicts 19% and 27%. The measured overhead is at or below the prediction, as expected: Flow-GRPO scores rewards asynchronously while the next batch is sampled, and parts of an iteration are not counted above. The measured overhead therefore contains no cost beyond the judge's compute, and the accounting can be used to estimate other settings.

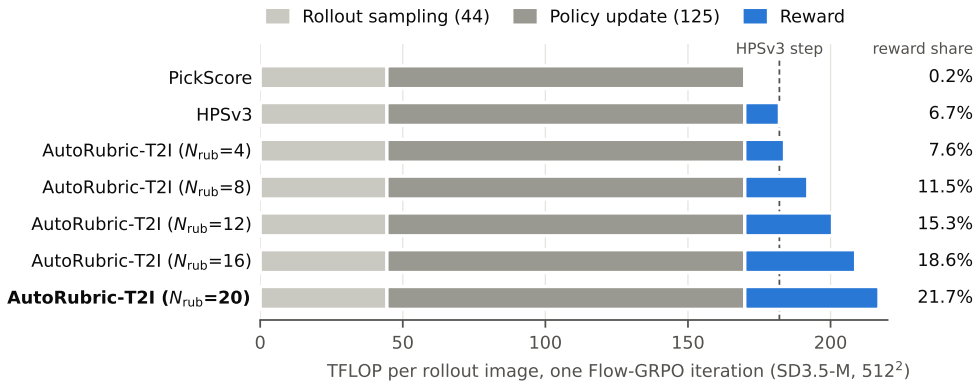

Figure 7: **Compute of one Flow-GRPO iteration per rollout image** (SD3.5-M, 512×512, settings of Table 1). Gray: the policy's rollout sampling and update, identical for every reward. Blue: the reward. The right column gives the reward's share of the iteration, and the dashed line marks the total with HPSv3. Reward costs are counted from measured token counts and agree with the measured throughput in Table 8 within 4%.

Fewer rubrics. The number of rubrics sets the cost directly, and the retained rubrics are ranked by learned weight. Table 9 trains Flow-GRPO with only the top-N_{rub} rubrics. With the Pick-a-Pic set, four rubrics match the full set downstream (68.4 vs. 68.3) at about the cost of HPSv3. With the HPDv3 set, 16 rubrics come within 0.2 points of the full set, and four are 0.5 points lower.

Rubric set	**Top-4**	**Top-8**	**Top-12**	**Top-16**	**Full (20)**
Pick-a-Pic	68.4	68.5	68.3	67.6	68.3
HPDv3	68.3	67.5	67.9	68.6	68.8
Iteration compute vs. HPSv3	+1.0%	+5.5%	+10.2%	+14.7%	+19.2%

Table 9: **RL with fewer rubrics.** Flow-GRPO on SD3.5-M with the top-N_{rub} rubrics by learned weight; the score is the mean of TIIF and UniGenBench++ (short and long prompts) at step 500. The last row is the compute of one iteration relative to HPSv3, from Figure 7.

Summary. AutoRubric-T2I replaces reward-model training with 256 preference pairs and a one-time refinement of 2–4 hours on 4× A6000 GPUs. In RL, its cost grows linearly with the number of rubrics, and each rubric costs about a sixth of an HPSv3 call because the image is encoded once. The full 20-rubric reward adds about 19% to the compute of a Flow-GRPO iteration relative to HPSv3, in line with the 18–20% we measured end to end, and a four-rubric reward costs about the same as HPSv3. In exchange, the reward is interpretable per rubric and needs no reward-model training.

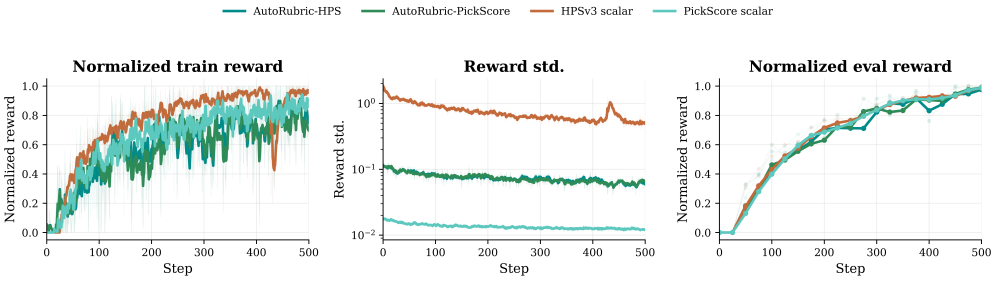

Figure 8: Training dynamics of scalar and rubric-based T2I rewards.

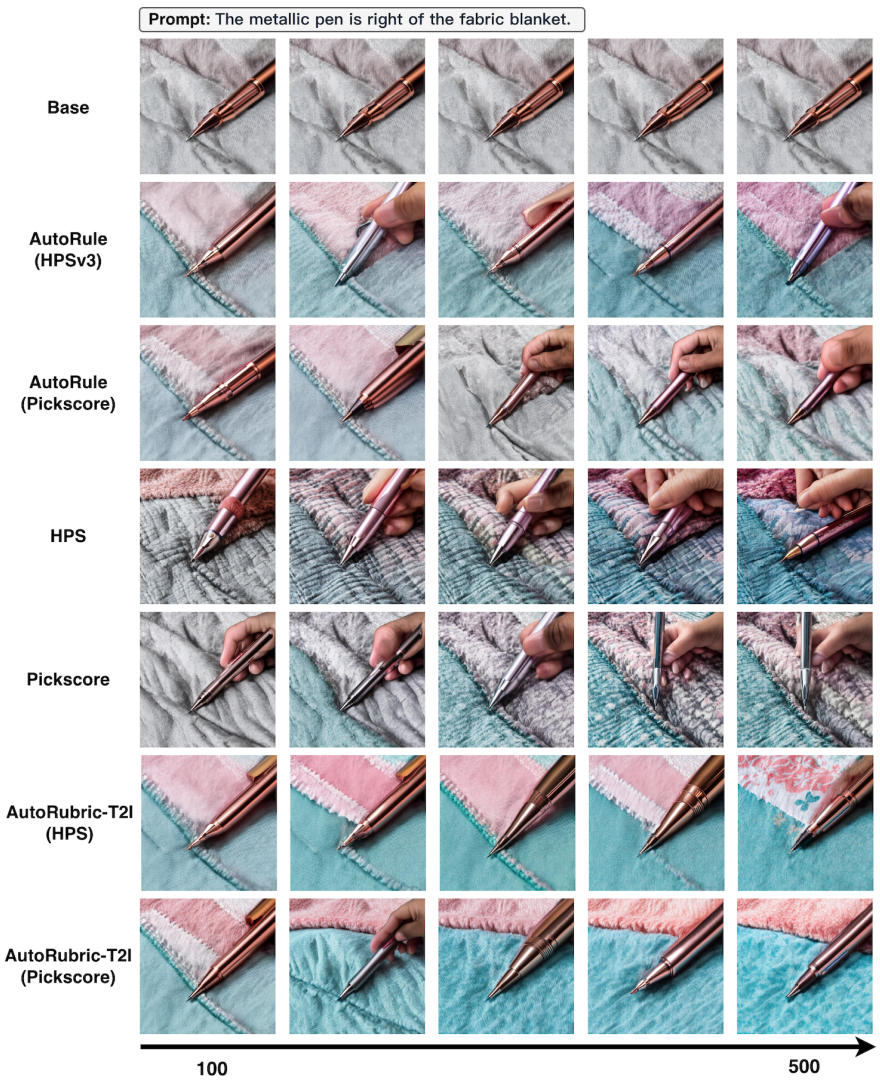

Figure 9: The evolution of generation quality of RL using AutoRubrics and other scalar reward models. The visual quality of scalar reward models degrades notably while the reward increases.

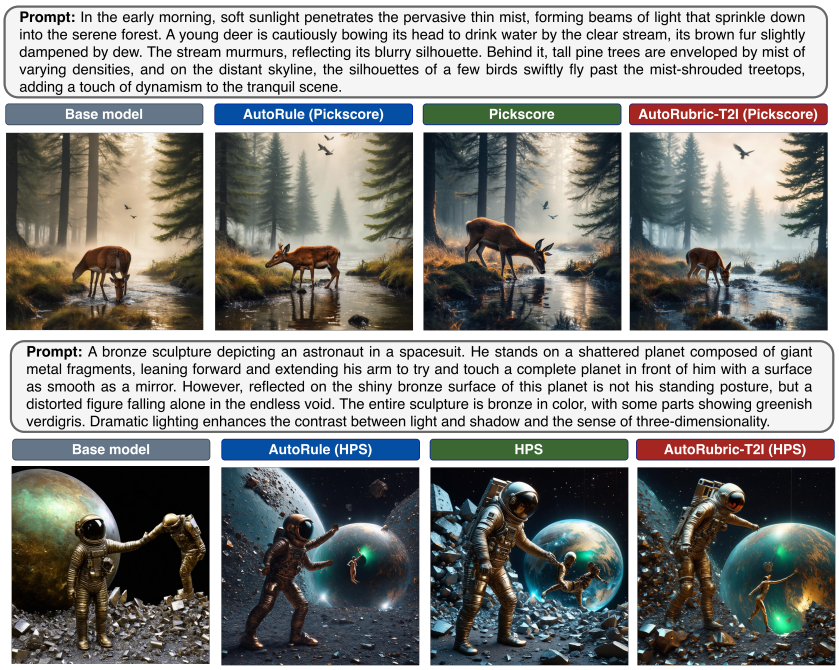

Figure 10: Qualitative comparison of downstream T2I RL policies. AutoRubric-T2I better preserves prompt-specific objects, relations, and fine-grained details compared with the base model, scalar reward optimization, and AutoRule-based rubric rewards.

K Training Dynamics

We provide training curves of T2I-RL referenced in Section 4. In Figure 8, we visualize normalized training and evaluation rewards together with reward standard deviation. AutoRubric-T2I achieves steadily improving training and evaluation rewards while maintaining substantially lower reward variance. In contrast, HPSv3 exhibits much larger reward dispersion throughout training, suggesting a noisier optimization signal, whereas PickScore has extremely low reward variance, indicating limited sensitivity for distinguishing high-capability generations. Qualitative results throughout RL training are visualized in Figure 9, where scalar reward shows clear visual evidence of reward hacking.

L Limitations and Broader Impact

Domain specificity of learned weights. The ℓ_1-regularized weights are fit to the preference distribution of the training corpus (e.g., HPDv3 or Pick-a-Pic), so the rubric *texts* remain broadly applicable, but their relative importance may shift on out-of-domain prompts such as typography-heavy or highly stylized images. Because rubric refinement requires only 256 pairs, the natural remedy is to re-fit the logistic regression on a small in-domain sample with the rubric set fixed, or to learn prompt-conditional weights.

RL training integration. AutoRubric-T2I can be directly integrated into Flow-GRPO as a drop-in reward by using the learned weighted sum of per-rubric scores. Beyond this simple scalarization, the explicit per-rubric decomposition provides a promising opportunity for rubric-aware advantage estimation, enabling finer-grained credit assignment during T2I reinforcement learning. However, how to fully exploit both rubric-level rewards and conventional scalar rewards for more effective advantage estimation remains an important direction for future work.

Broader impact. By replacing opaque scalar reward models with human-readable rubrics and learned weights, AutoRubric-T2I makes the criteria that drive T2I optimization inspectable and auditable, which helps diagnose and correct biases (e.g., systematic under-weighting of culturally diverse aesthetics) directly from the weights. The low data requirement and open-source infrastructure



Figure 11: Qualitative examples from downstream RL fine-tuning with AutoRubric-T2I rewards. Each row shows a text prompt and the corresponding generated image, demonstrating improved prompt alignment, object placement, attribute accuracy, and overall visual quality after RL training.

further lower the barrier for groups without large annotation pipelines or proprietary APIs, and we do not foresee negative societal impacts beyond those inherent to T2I generation broadly.

M PROMPT

Figures 12–15 present the complete set of prompt templates used at all stages of the AutoRubric-T2I pipeline: vision reasoner and rule extractor/merger for seed rubric generation, VLM judge templates, and hard-pair diagnosis and rule extraction prompts used during iterative refinement.

Vision Reasoner

[Instruction]
You are tasked with comparing two images generated by two different AI assistants and explaining why a human evaluator would prefer the winning image.

Your evaluation must be based on **two main perspectives only**:
(1) Visual Quality
(2) Text Alignment

Follow step-by-step reasoning process and use clear, objective judgment grounded in the criteria below.
Step 1: Image Content Analysis
- Briefly and objectively describe the visible content of each image.
- Do not evaluate yet; only state what is shown.

Step 2: Visual Quality Evaluation
Evaluate each image independently based on:
- Reasonableness (absence of biological, physical, or logical errors)
- Clarity (sharpness, visibility, lack of blur or artifacts)
- Detail Richness (textures, lighting, materials, fine details)
- Aesthetic and Creativity (composition, color harmony, atmosphere, style execution)
- Safety (absence of political, violent, adult, or otherwise inappropriate content)

Step 3: Text Alignment Evaluation
Evaluate how well each image aligns with the textual prompt, focusing especially on:
- Subject Relevance (accuracy of main subjects)
- Style Relevance (faithfulness to the specified artistic or visual style)
- Contextual Consistency (logical and coherent background and environment)
- Attribute Fidelity (correct representation of specified attributes)
- Semantic Coherence (whether the image captures the overall intent and meaning of the prompt without contradictions or irrelevant elements)

Step 4: Comparative Reasoning
- Compare Assistant A and Assistant B directly.
- Explain why the winning image is preferred from a **human preference perspective**, clearly stating whether the advantage comes from stronger Visual Quality, better Text Alignment, or both.
- If trade-offs exist, explicitly describe them.

[**Textual Prompt]**
{prompt}
[Image by Assistant A]
<IMAGE: image_a>
[Image by Assistant B]
<IMAGE: image_b>
[Winning Image]
{winner}
[Your Explanation]

Figure 12: Seed rubric generation, stage 1: vision reasoner that produces a step-by-step preference rationale for each image pair.

Rule Extractor

[Instruction]
Based on the following reasoning about why conversation with assistant {winner} is better, extract any rule-like statements implied by the reasoning that indicate this preference. Rule-like statements should be able to be judged objectively and deterministically. Below are a few examples of rule-like statements:

Example 1:
- The assistant's responses should validate any assumptions made with sufficient context.
Example 2:
- The assistant's responses should not simply restate information provided by the user as its answer.
Example 3:
- The assistant's responses should have a structure that satisfies the user's request.

Return the list as a JSON array of strings. Do not use ```json```, just output the JSON array directly. If there are no rule-like statements, return an empty JSON array.
[Reasoning]
{reasoning_text}

Rule Merger

[Instruction]
Below is a large list of rule-like statements regarding the behavior of an AI assistant. Some of these rules might be duplicates or very similar in meaning.
Please merge them so that there are no duplicates or rules with very similar meanings.
Return the merged list as a JSON array of strings. Output the JSON on a single line and do not include newlines inside strings.
Do not use ```json```, just output the JSON array directly.
[Rules]
{rules_text}

Figure 13: Seed rubric generation, stages 2-3: rule extractor rule merger.

VLM Judge — Binary (Yes / No)

[Instruction]
You are an impartial judge. Your task is to evaluate the provided image based on a specific rule and the user's original request.

[User's Original Request]
{user_original_prompt}
[Rule to Check]
{rule}
[Evaluation Task]
Analyze the provided image. Does the assistant's generated image satisfy the Rule mentioned above?
Base your judgment solely on whether the image satisfies the rule.
[Analysis]
If the Rule mentions specific elements that are NOT present in the User's Original Request, consider the rule satisfied and respond "Yes".
Otherwise, respond "Yes" if the image follows the rule, and "No" if it violates it.

Respond with one of the following options, and nothing else: "Yes" or "No".

Figure 14: VLM judge templates: Yes/No binary scoring.

Hard-Pair — Diagnose

[Instruction]
You are an expert evaluator specializing in high-precision assessment of image responses.

The current rubric items may be too generic, overly lenient, or insufficient to effectively distinguish between the two generated images. Your task is to analyze this single-failure case and explain why the preferred response is preferable and why the existing rubrics are insufficient or ambiguous in this case.

[Rules]
- Do NOT generate new rules or rubrics.
- Do NOT output JSON.

Step 1: Case Analysis
- Briefly describe what is shown in each response.

Step 2: Rubric Evaluation
- Explain how the current rubrics fall short for this case.
- Point out which dimensions are missing or under-specified.

Step 3: Final Explanation
- Summarize the core blind spot that caused the ranking failure.

[User Prompt]
{prompt_text}
[Existing Rubrics]
{rubric_block}
[Response A]
<IMAGE: img_a_bytes>
[Response B]
<IMAGE: img_b_bytes>
[Preferred Response]
{preferred}

Hard-Pair — Rule Extractor

[Instruction]
Based on the following reasoning about why a set of rubrics fails and why {preferred} is better, extract any rule-like statements implied by the reasoning that indicate this preference. Rule-like statements should be able to be judged objectively and deterministically. Below are a few examples of rule-like statements:

Example 1:
- The assistant's responses should validate any assumptions made with sufficient context.
Example 2:
- The assistant's responses should not simply restate information provided by the user as its answer.
Example 3:
- The assistant's responses should have a structure that satisfies the user's request.

Return the list as a JSON array of strings. Do not use ```json```, just output the JSON array directly. If there are no rule-like statements, return an empty JSON array.
[Reasoning]
{reasoning_text}

Figure 15: Hard-pair refinement prompt.

N Quantitative Human Evaluation

To supplement the quantitative human evaluation in Section 5.3, we provide a screenshot of the user survey interface used in our study. The survey was conducted with 30 graduate-student annotators. Each annotator was asked to evaluate 20 T2I prompts, resulting in 600 total human judgments.

For each question, annotators were shown a text prompt at the top, and four anonymized image candidates labeled A–D. These candidates correspond to outputs from the base model, scalar-reward optimization, AutoRule-based optimization, and AutoRubric-T2I, with the display order randomized to reduce positional bias. Annotators were instructed to select the single image that best satisfied the text prompt, considering both visual quality and prompt alignment. In particular, they were encouraged to focus on whether the generated image correctly captured the requested objects, spatial relations, attributes, scene layout, and other fine-grained constraints in the prompt.

Figure 16 shows an example survey question. The instruction explicitly asks users to vote for the best image according to the text prompt, and each question requires exactly one choice among the four candidates. This simple forced-choice design avoids requiring annotators to assign calibrated numerical scores, while still providing a direct measure of human preference among RL policies.

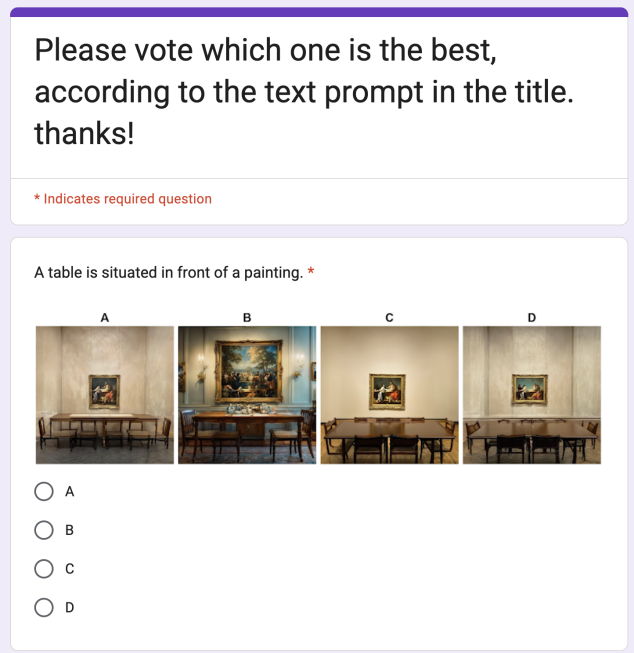

Figure 16: Screenshot of the human evaluation survey interface. Annotators were asked to choose the best image according to the text prompt shown in the question title.

O Optimized Rubric Set

This section reports the final weighted rubric sets produced by the AutoRubric-T2I pipeline for each (VLM judge, source corpus) configuration. Rules are ranked in decreasing order of their fitted ℓ_1-regularized logistic-regression weights ($\bar{w}$); rules with effectively zero weight are pruned from the displayed Top-N_{rub} set.

Rank	Rule	Weight
1	The assistant's responses should utilize realistic and sophisticated lighting, color palettes, and atmospheric effects (such as cinematic, volumetric, or specific time-of-day lighting) as specified.	1.5219
2	The assistant's responses should maintain a coherent subject and structural integrity rather than producing chaotic or abstract compositions.	1.3223
3	The assistant's responses should maintain high visual quality, providing sharp, high-resolution images with fine detail richness while avoiding blurriness, pixelation, or low-quality textures.	1.0472
4	The assistant's responses should meet a minimum threshold of technical quality, providing functional and high-resolution visuals.	0.8374
5	The assistant's responses should incorporate all specific descriptors provided in the prompt, such as color palette and clothing style.	0.7763
6	The assistant's responses should prioritize compositional impact and atmospheric depth over literal semantic accuracy, such as text fidelity or object counts, when the overall aesthetic quality is significantly higher.	0.7649
7	The assistant's responses should be preferred if they provide a coherent, high-resolution output that follows compositional prompts, even if they fail specific thematic or era-appropriate instructions, over a corrupted or low-resolution image that attempts those details.	0.7412
8	The assistant should prioritize high visual quality and traditional aesthetic standards if literal adherence to stylistic constraints results in a technical failure.	0.7360
9	The assistant's responses should prioritize high-fidelity subject recognition and specific attributes over generic or distorted outputs when multiple constraints are missed.	0.6328
10	The assistant's responses should treat the omission of a primary environmental descriptor as a more significant failure than a failure in text rendering.	0.6109
11	The assistant's responses should include all specific objects requested in the prompt, avoiding 'hallucination by omission'.	0.5024
12	The assistant's responses should ensure visual coherence by avoiding random artifacts, floating lines, or 'melting' features.	0.4358
13	The assistant's responses should ensure that primary subjects are recognizable and rendered with high-resolution detail.	0.3935
14	The assistant's responses should be considered a failure if they contain catastrophic technical artifacts, such as extreme pixelation, lack of discernible facial features, or non-human textures.	0.1617
15	The assistant's responses should accurately convey the scale and grandeur of the subject matter to match its real-world context.	0.1261
16	The assistant's responses must include all primary subjects requested in the prompt.	0.0987
17	The assistant's responses should not omit requested objects in favor of creating a visually appealing scene.	0.0215
18	The assistant's responses should ensure that the primary subject is not rendered with catastrophic technical failures or structural collapse.	0.0170
19	The assistant's responses should prioritize specific aesthetic markers and instructional density over general technical quality.	0.0009

Figure 17: Optimized rubric set for Qwen-3-VL-8B trained on HPDv3 preference pairs (round 3).

Rank	Rule	Weight
1	The assistant's responses should prioritize a singular, unified subject over a 'collage effect' of fragmented tags or letters when a specific phrase is requested.	1.4700
2	The assistant's responses should avoid AI artifacts such as smudged textures, broken lines in geometric patterns, or a lack of structural integrity in the depicted object.	1.1393
3	The assistant's responses should aim for high-end editorial or cinematic production value rather than a generic stock photo aesthetic.	0.7842
4	The assistant's responses should accurately render specific strings of text requested by the user.	0.6018
5	The assistant's responses should prioritize capturing the complex architecture and atmosphere of a requested location over literal adherence to object material or color attributes.	0.5751
6	The assistant's responses should not substitute a specifically requested material with a generic or industry-standard alternative.	0.5653
7	The assistant's responses should successfully merge multiple requested concepts into a single, coherent hybrid symbol rather than producing a generic or unrecognizable shape.	0.5035
8	The assistant's responses should incorporate the specific artist influences and artistic mediums requested in the prompt.	0.4572
9	The assistant's responses should provide realistic and intricate textures for materials like skin, hair, fabric, and metal, ensuring they react naturally to light and reflect their specified properties without a 'plastic' or artificial appearance.	0.4489
10	The assistant's responses should maintain a recognizable likeness of the requested person as the anchor of the image, even when integrating multiple stylistic influences.	0.4485
11	The assistant's responses should prioritize atmospheric keywords (such as mood and lighting) over literal object keywords when the atmosphere is central to the prompt's intent.	0.4335
12	A failure in subject identity should be considered a more severe error than a failure in framing or technical constraints.	0.3685
13	The assistant's responses should adhere to mandatory subject attributes, such as specific states of dress, when explicitly requested in the prompt.	0.3229
14	The assistant's responses should prioritize the prompt's literal requirements over generic or sanitized defaults.	0.3094
15	The assistant's responses should prioritize anatomical and biological accuracy over aesthetic or atmospheric qualities.	0.2972
16	The assistant's responses should represent prompt keywords as physical, interactive entities within the scene rather than as background wallpaper or subtle lighting effects.	0.2952
17	The assistant's responses should ensure that keywords intended for artistic style do not override or replace the primary subject of the prompt.	0.2501
18	The assistant's responses should not substitute requested thematic environments with unrelated or generic settings.	0.2493
19	The assistant's responses should strictly adhere to all specific instructions, keywords, subjects, and descriptive attributes provided in the prompt, including numerical requirements, plurality, and spatial arrangements.	0.2257
20	The assistant's responses should strictly adhere to foundational compositional requirements, such as 'whole body' framing, when specified in the prompt.	0.0736
21	The assistant's responses should render celestial or planetary bodies with realistic colors and atmospheric effects when a photorealistic style is requested.	0.0324
22	Subject integrity should be prioritized over stylistic or compositional misses.	0.0020

Figure 18: Optimized rubric set for Qwen-3-VL-8B trained on Pick-a-Pic preference pairs (round 6).

Rank	Rule	Weight
1	The assistant's generated images should feature high-fidelity, detailed textures for all subjects and environments, such as realistic skin pores, hair strands, fabric weaves, and architectural materials.	2.0324
2	The assistant's responses should accurately convey the specific emotional expressions requested in the prompt.	1.3942
3	The assistant's responses should ensure that the texture of the subject matches the specific descriptors provided in the prompt.	1.3078
4	The lighting in generated images should be physically and logically consistent, incorporating realistic shadows, highlights, reflections, and specified atmospheric effects like volumetric lighting or bokeh.	1.1847
5	The assistant's responses should ensure that objects are recognizable and match their prototypical appearance.	0.9303
6	The assistant's responses should follow the requested color palette by balancing all specified colors throughout the image rather than letting one dominate.	0.8473
7	The assistant's responses should avoid anatomical distortions and ensure stable rendering of human features, such as hands.	0.7451
8	The assistant's generated images should depict realistic and physically plausible interactions between subjects and objects, such as proper hand grips and subjects being grounded on the floor.	0.6887
9	The assistant's responses should maintain high-fidelity textures and sharp details in the subject's features, such as fur and eyes, to avoid a soft or low-quality appearance.	0.5663
10	The assistant's responses should not contain watermarks or garbled text.	0.5455
11	The assistant's responses should maintain the visual readability of the subject by ensuring its form and anatomy are clearly distinguishable from the background or supporting textures.	0.4878
12	The assistant's responses should ensure that textures are natural and logical rather than repetitive or uncanny.	0.3730
13	The assistant's responses should visually communicate the requested profession of the subject through recognizable attire or accessories.	0.2559
14	The assistant's responses should accurately capture the specific artist's style or genre requested in the prompt.	0.2358
15	The assistant's responses should adhere to specific aesthetic quality descriptors like '3D hyperrealistic' or '8k' by ensuring high digital sharpness and post-processed polish.	0.1061

Figure 19: Optimized rubric set for Qwen-3-VL-32B trained on HPDv3 preference pairs (round 3).

Rank	Rule	Weight
1	The assistant's responses should accurately represent the material texture of well-known objects, such as the brushed stainless steel of a DeLorean.	1.8352
2	When depicting glass sculptures, the assistant's responses should include medium-specific characteristics such as refraction, internal patterns, and varied thickness.	1.6965
3	The assistant's responses should prioritize morphological fidelity and anatomical specificity of a public figure's unique facial geometry over technical resolution or lighting quality.	1.6253
4	The assistant's responses should provide a contextual background that matches the mood of the subject, such as a gallery or home setting, rather than a sterile studio background.	1.0704
5	The assistant's responses should accurately render specific lighting-based adjectives provided in the prompt, such as 'luminous' or 'electric', rather than using flat or matte colors.	0.9937
6	The assistant's responses should avoid 'aesthetic substitution', where requested elements are replaced by generic high-quality visual effects.	0.9562
7	The assistant's responses should incorporate professional photography standards, such as cinematic lighting, sharp focus, and clear catchlights in the eyes.	0.9475
8	The assistant's responses should prioritize the specific physical character and functional implications requested by the user's adjectives over general technical polish or environmental consistency.	0.7591
9	Watercolor illustrations should maintain luminosity and avoid heavy, opaque, or 'muddy' qualities.	0.7158
10	The assistant's responses should adhere to atmospheric constraints, such as 'standing in the dark', and avoid using contradictory lighting elements like bright spotlights.	0.5371
11	The assistant's responses should not allow technical attributes like sharpness or lighting to compensate for a lack of substantive accuracy regarding the subject or setting.	0.4224
12	The assistant's responses should ensure that any requested text, names, symbols, or logos are spelled correctly, legible, and rendered without distortion or gibberish.	0.4078
13	The assistant's responses should prioritize a singular, immersive presentation for artistic prompts over a multi-view or catalog-style presentation.	0.3871
14	The assistant's responses should be free of visual noise, such as stray lines, disconnected dots, or fragmented shapes.	0.3817
15	The assistant's responses should avoid generic or high-saturation interpretations when a specific gritty or desaturated aesthetic is requested through a cultural reference.	0.3535
16	The assistant's responses should adhere to specific botanical requirements mentioned in the prompt, such as 'coniferous forest', rather than providing abstract or deciduous alternatives.	0.3457
17	The assistant's responses should be free of AI-generated artifacts, such as mangled anatomy or symbolic corruption.	0.2831
18	The assistant's responses should render high-relief sculptures with a clear sense of three-dimensional depth and volume, avoiding flat or 2D-like compositions.	0.2224
19	The assistant's responses should prioritize material fidelity and tactile accuracy over generic or smooth surfaces when a specific material is mentioned in the prompt.	0.2107
20	The assistant's responses should maintain structural and geometric integrity in architectural renderings.	0.1999

Figure 20: Optimized rubric set for Qwen-3-VL-32B trained on Pick-a-Pic preference pairs (round 6).