Element-Aware Group Learning for E-Commerce Image Generation

Jingtong Chen*, Jiahui Wang*, Xue Zhao†, ShaoGuo Liu†, Minghao Li

Abstract

Recent advances in image generation and editing have made prompt quality a key bottleneck for e-commerce creatives. Vision-language models (VLMs) can generate image-editing prompts from product images and metadata, but further improving their prompt-writing capabilities requires post-training with feedback from the generated images. Group Relative Policy Optimization (GRPO) is a natural framework for such outcome-level reward optimization. However, it assigns credit only at the full-prompt level, even though image quality often depends on specific design elements such as composition, background, and the presentation of selling points. Existing fine-grained credit assignment methods typically require step-level supervision or learned critics. To address this, we propose EAGLE-GRPO (**E**lement-**A**ware **G**roup **L**earning for **E**-Commerce Image Generation), which decomposes the group-centered reward over predefined elements. We cast element-level credit assignment as a kernel ridge regression problem and derive a closed-form solution, without additional rollouts or separate credit-assignment models. This yields interpretable per-element advantages and more precise policy updates. Experiments show that EAGLE-GRPO sustains performance gains over more training steps before plateauing and generates prompts that produce higher-quality e-commerce images than competitive VLM prompt-writing baselines.

Introduction

Modern image editors can produce high-quality e-commerce visuals, but the prompt itself spans preservation, layout, headline, scene, and other design decisions. VLMs are commonly used to enhance such prompts from a product image and metadata (e.g., title, description), then refine them via reward feedback. GRPO (Shao et al. 2024) has emerged as a popular framework for such reward-driven VLM training, as it removes the need for a separate value model by normalizing rewards within sampled groups. However, in standard GRPO each candidate receives one scalar image reward, and the group-normalized advantage is broadcast uniformly to all tokens. This coarse credit assignment conflates good and bad decisions, allowing weak fields to ride high rewards earned by strong ones.

(a) Step 40 (reward=0.836) (b) Step 200 (reward=0.886)

Figure 1: Illustrative evolution of visual design elements during EAGLE-GRPO training. Compared with the output at training step 40, the output at training step 200 includes a close-up supporting visual and multiple grounded feature callouts, yielding a more product-specific and informative presentation, consistent with its higher image reward.

We argue that prompt optimization should operate at the level of design elements, which are visually entangled in the rendered image but textually separable in the structured prompt. Figure 1 illustrates how training refines individual visual elements by adding a supporting visual and grounded feature callouts while preserving the same product. This reframes the learning problem: given only a holistic image reward, how do we assign credit to individual elements?

A natural approach is Shapley-value attribution (Shapley 1953; Cao et al. 2025; Li et al. 2026), which assigns each element's credit by comparing outcomes with and without that element. Yet this element-level variation is already present in a GRPO rollout group as a byproduct of sampling, since different rollouts differ in which elements they include and how they are realized. Building on this free variation, we therefore propose EAGLE-GRPO (**E**lement-**A**ware **G**roup **L**earning for **E**-Commerce Image Generation), which assigns per-element credit from the reward variation already present in a GRPO rollout group. It parses each prompt into tagged design elements, pools their token spans into embeddings that serve as kernel features, and decomposes the centered image reward into per-element contributions via a closed-form kernel-ridge solution. The resulting credits are

*These authors contributed equally.
†Corresponding author.

mapped back to token spans through a residual-preserving transformation, such that the token average equals the standard GRPO advantage (**Lemma 1**).

The main contributions are:

- We introduce a kernelized element-credit mechanism for GRPO that decomposes image rewards into per-element credits via a closed-form kernel-ridge solution, requiring no additional counterfactual rollouts.
- The element-credit design preserves the standard GRPO signal by construction: averaging the token-level element advantages over all tokens exactly recovers the group-normalized advantage, ensuring that credit is redistributed across tokens within each sample while preserving the per-sample mean advantage (Lemma 1).
- We evaluate against competitive prompt-writing baselines with GPT-Image-2, and FLUX.2 [klein] as image editors; EAGLE-GRPO outperforms Standard GRPO on both editors and is preferred by human raters in 62% of blind pairwise comparisons.

Related Work

Fine-Grained Credit Assignment for GRPO. Group Relative Policy Optimization (GRPO) (Shao et al. 2024) and variants such as DAPO (Yu et al. 2025) normalize rewards within sampled groups but assign the same advantage to every token. Recent efforts to refine the credit itself include critic-free token reweighting via intrinsic proxies such as entropy or logit confidence (Tan and Pan 2025; He et al. 2026), critic-based methods that reintroduce learned value functions (Schulman et al. 2016), and process reward models that densify the signal via step-level supervision (Lightman et al. 2023; Setlur et al. 2025; Khalifa et al. 2025; Zheng et al. 2025). Yet none of these refinements target the semantic structure of structured prompts: in our setting, the natural unit of credit is the prompt element (headline, scene, lighting, etc) rather than the token. EAGLE-GRPO takes a different route: it parses the prompt into tagged elements and assigns each element a credit derived from the reward variation already present in a GRPO rollout group, with no trained value model or external annotation.

Reward Attribution Strategies. A separate line of work studies how to attribute a single outcome-level reward to finer output units. Counterfactual methods evaluate outcomes under different configurations and assign credit via cooperative-game solutions such as the Shapley value (Shapley 1953). For example, SCAR (Cao et al. 2025) masks token spans and queries a reward model per configuration, SHARP (Li et al. 2026) uses leave-one-out evaluation for multi-agent tool-use, and SAVOIR (Feng et al. 2026) applies expected-utility Shapley values to multi-turn dialogue. These methods are principled, but they require interventional sampling beyond the rollout group, since each configuration demands an additional generation and reward query.

EAGLE-GRPO takes the correlational route instead: rather than constructing counterfactual configurations, it attributes credit from reward variation already present across candidates. This attribution is formulated as an additive regression over element features. Its closed-form kernel-ridge decomposition is an instance of additive RKHS regression (Wahba 1990; Gu 2002; Kandasamy and Yu 2016), where each element defines a reproducing kernel and the centered reward is fit by an additive model regularized per component via the representer theorem (Kimeldorf and Wahba 1971). A residual-preserving transformation then maps these credits back to token spans, such that the token-level average exactly recovers the standard GRPO advantage. Prior GRPO-based credit assignment has focused on refining credit along tokens, steps, or segments; decomposition over structured, semantically tagged prompt elements remains largely unexplored. EAGLE-GRPO addresses this setting.

Prompt Writing for Image Generation. Prior prompt-writing work uses three reward paradigms. Promptist (Hao et al. 2023) trains a writer with PPO (Schulman et al. 2017) on external aesthetic and relevance scores. PromptEnhancer (Wang et al. 2025) relies on a separate alignment evaluator to score chain-of-thought rewrites. Self-Rewarding LVLM (Yang et al. 2025) unifies the writer and judge in a single VLM, iterating via self-generated preference signals. All optimize the prompt as a single string against a scalar reward; none exploits the fact that a low image score often stems from one element rather than the entire prompt. EAGLE-GRPO instead defines a structured prompt schema that decomposes the prompt into tag-delimited fields with well-defined token spans, enabling per-element credit assignment.

Methodology

Prompt Element Schema

E-commerce prompt writing combines product-preservation constraints with commercial presentation decisions such as headline, layout, scene, lighting, etc. To make these decisions individually creditable, we structure each prompt as a tag-delimited schema: every design decision occupies a dedicated field delimited by atomic tags (e.g., `<headline>...<headline>`). Each such field defines one *element*, the atomic unit of credit assignment. Given a product image I_{prod} and product metadata m,

$$x = (I_{\text{prod}}, m), \tag{1}$$

a VLM policy first produces a reasoning trace z^{cot} that analyzes the product and plans the design, then emits a schema-delimited prompt:

$$y = (z^{\text{cot}}, z), \qquad z = (z_1, \ldots, z_K), \tag{2}$$

where z is the full structured prompt and each z_k is the k-th element. The schema gives each design decision a stable boundary tag, turning free-form prompt into a set of creditable fields without an external parser. Table 1 lists the 18 elements. Required and optional fields capture creative design decisions; constraint-only fields (preservation, negative) enforce faithfulness and safety boundaries. We use a strong VLM to generate high-quality image-editing prompts containing self-designed tags. We then supervised-fine-tune (SFT) a student VLM on this data to produce the schema reliably. Each schema tag is added to the tokenizer as an atomic delimiter for reliable field parsing and hidden-state pooling. Figure 2 illustrates the full workflow.

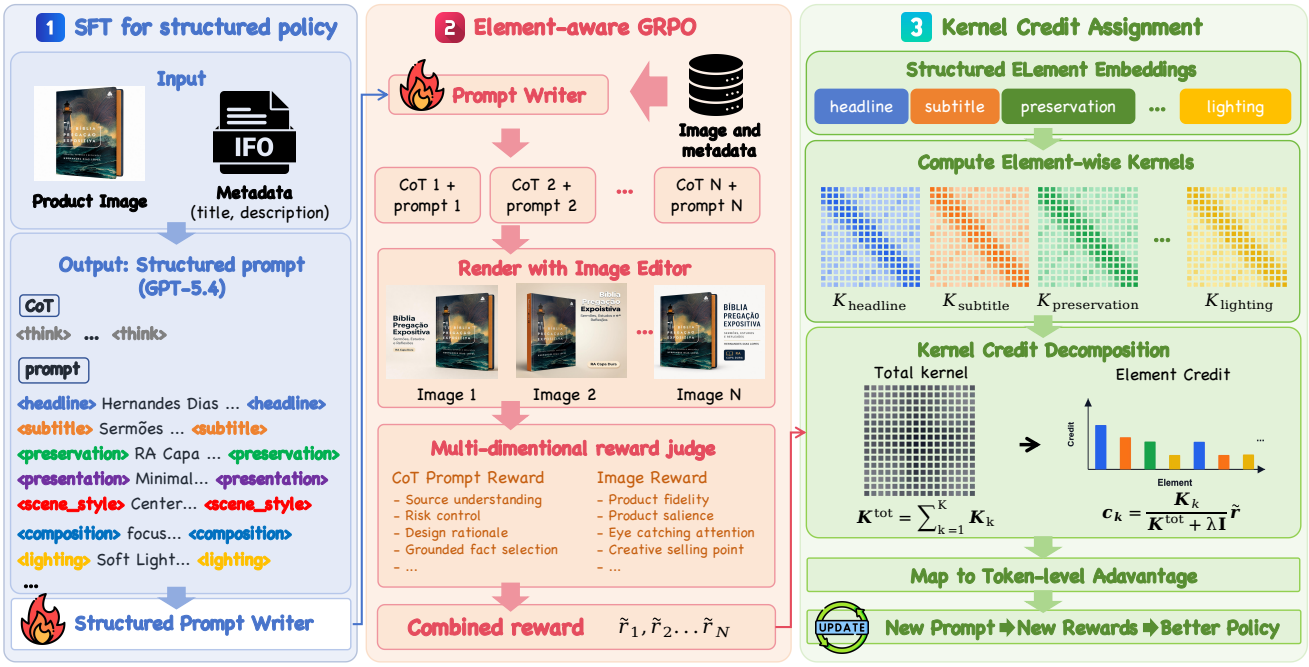


Figure 2: Overview of EAGLE-GRPO. **Stage 1: structured SFT** teaches the prompt writer to analyze the product image and metadata, and produce an structured prompt with 18 tagged elements. **Stage 2: element-aware GRPO** samples N structured prompts, renders them with an image editor, and obtains prompt- and image-level scores from a multidimensional reward judge. **Stage 3: kernel credit assignment** constructs element embeddings, builds an element-wise kernel K_k for each element k, and forms the total kernel K^{tot}. Here, $\tilde{\mathbf{r}}$ is the group-centered combined reward vector, and $\mathbf{c}_k$ is the reward contribution attributed to element k. The element credits are mapped to token-level advantages and used to update the prompt policy.

Element-Aware GRPO

After SFT, for each training example x, the policy $\pi_{\theta_{\text{old}}}$ samples a group of N outputs $\{y_1, \ldots, y_N\}$, where $y_i \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)$. Writing $y_i = (z_i^{\text{cot}}, z_i)$, the structured prompt z_i is rendered by an editor G and judged by a VLM reward model R_{image} (a strong off-the-shelf VLM, Gemini-2.5-Flash in our experiments):

$$I_{\text{out},i} = G(I_{\text{prod}}, z_i), \qquad r_i = R_{\text{image}}(I_{\text{prod}}, I_{\text{out},i}, m). \tag{3}$$

Standard GRPO normalizes rewards within the group:

$$A_{\text{image},i} = \frac{r_i - \mu}{\sigma + \epsilon_r}, \tag{4}$$

where μ and σ are the group reward mean and standard deviation. Broadcasting this scalar to all loss-bearing prompt tokens cannot distinguish which structured decisions made a rollout successful. EAGLE-GRPO instead differentiates advantage across schema elements: constraint-only fields (preservation, negative) receive the standard image-level advantage A_{image} ; for non-constraint fields, the kernel module produces element-conditioned token credit $A_{\text{elem},i,t}$. The two are blended as

$$A_{\text{mix},i,t} = (1-\eta)A_{\text{image},i} + \eta A_{\text{elem},i,t}. \tag{5}$$

where $\eta \in [0,1]$ controls the blend between image-level and element-level advantage. By design, the token average of A_{elem} recovers A_{image}, so element credit only redistributes within a rollout without changing its image-level mean.

Element-Level Credit Decomposition

For every rollout, we construct a fixed 18-field representation, with kernel attribution applied to the 16 non-constraint fields. A non-empty field is represented by its pooled token embedding, whereas an absent optional field is represented by the zero vector:

$$\mathbf{v}_{i,k} = \begin{cases} \text{LayerNorm}\left(\frac{1}{|S_{i,k}|}\sum_{t \in S_{i,k}} \mathbf{h}_{i,t}\right), & |S_{i,k}| > 0, \\ \mathbf{0}, & |S_{i,k}| = 0, \end{cases} \tag{6}$$

where $S_{i,k}$ denotes the set of token positions belonging to element k in rollout i, and $\mathbf{h}_{i,t}$ denotes the last-layer hidden state computed by $\pi_{\theta_{\text{old}}}$ at rollout time. Consequently, each field kernel jointly captures field selection (presence versus absence) and semantic variation among its non-empty realizations. Credit is inferred from within-group variation, so embeddings are centered within each rollout group:

$$\tilde{\mathbf{v}}_{i,k} = \mathbf{v}_{i,k} - \frac{1}{N}\sum_{j=1}^{N} \mathbf{v}_{j,k}. \tag{7}$$

Within each group, the element kernel is

$$\mathbf{K}_k(i,j) = \tilde{\mathbf{v}}_{i,k}^{\top} \tilde{\mathbf{v}}_{j,k}. \tag{8}$$

Field	Function
Constraint-only (not credited)	
preservation_constraints	Preserve identity and geometry
negative_constraints	Prohibit unsupported edits
Required (always credited)	
product_presentation	Hero treatment
headline	Primary message
composition_layout	Product / text arrangement
scene_style	Background and mood
lighting_rendering	Lighting and finish
Optional (credited when present)	
subtitle	Secondary hierarchy
feature_callouts	Visualize benefits
specification_information	Specifications
quantity_bundle_information	Count or bundle description
variant_information	Product variants
price_promotion_information	Promotion
seller_logistics_information	Seller / delivery facts
trust_information	Trust cues
usage_information	Use scenario
comparison_visual	Comparison
supporting_visuals	Insets or detail views

Table 1: Structured prompt elements and credit rules. Constraint fields are not credited; required fields are always credited; optional fields are credited only when non-empty.

Let $\tilde{r}_i = r_i - \mu$ denote the centered image reward for rollout i, where μ is the group mean, and let $\tilde{\mathbf{r}} = [\tilde{r}_1, \ldots, \tilde{r}_N]^\top \in \mathbb{R}^N$ denote the vector of centered rewards within the group. We estimate an element-contribution vector $\mathbf{c}_k \in \mathbb{R}^N$ for each element k by

$$\min_{\{\mathbf{c}_k\}_{k=1}^K} \quad \left\| \tilde{\mathbf{r}} - \sum_{k=1}^K \mathbf{c}_k \right\|_2^2 + \lambda \sum_{k=1}^K \mathbf{c}_k^\top (\mathbf{K}_k + \epsilon_K \mathbf{I})^{-1} \mathbf{c}_k. \quad (9)$$

The regularizer restricts element k to claim credit only along directions supported by its own variation within the group. The solution takes the closed form $\mathbf{c}_k = \mathbf{K}_k \boldsymbol{\omega}$. With $\mathbf{K}^{\mathrm{tot}} = \sum_k \mathbf{K}_k$,

$$\boldsymbol{\omega} = (\mathbf{K}^{\mathrm{tot}} + \lambda \mathbf{I})^{-1} \tilde{\mathbf{r}}, \quad (10)$$

and therefore

$$\mathbf{c}_k = \mathbf{K}_k \boldsymbol{\omega} = \mathbf{K}_k (\mathbf{K}^{\mathrm{tot}} + \lambda \mathbf{I})^{-1} \tilde{\mathbf{r}}. \quad (11)$$

Let $c_{i,k} = [\mathbf{c}_k]_i$ denote the contribution of element k to rollout i. The portion of $\tilde{r}_i$ not attributed to any field is kept as a residual,

$$\nu_i = \tilde{r}_i - \sum_{k=1}^K c_{i,k}. \quad (12)$$

In practice, we apply conservative identifiability checks: near-zero reward variance or ill-conditioned kernels fall back to coarse image credit; low-energy element kernels receive zero contribution; highly aligned kernels are down-weighted. The decomposition is computed independently per group and detached from the policy gradient.

Token-Level Advantage Mapping

Each element has a token span in the final prompt. The element credit $c_{i,k}$ and residual ν_i are in reward space; to make them comparable with the standard GRPO advantage (Eq. 4), we rescale them by the group standard deviation σ, obtaining the element-level advantage $Q_{i,k}$ and the per-rollout residual advantage U_i:

$$Q_{i,k} = \frac{c_{i,k}}{\sigma + \epsilon_r}, \quad U_i = \frac{\nu_i}{\sigma + \epsilon_r}. \quad (13)$$

These are then distributed across tokens within each element. For each element k of $i-$th rollout, let $L_{i,k}$ be the number of loss-bearing tokens; $T_i = \sum_k L_{i,k}$ is the total count of loss-bearing tokens in $i-$th rollout. For a token t, let $k(t)$ denote the element containing t; the element-aware advantage is:

$$A_{\mathrm{elem},i,t} = U_i + \frac{T_i \cdot Q_{i,k(t)}}{L_{i,k(t)}}. \quad (14)$$

Although $A_{\mathrm{elem},i,t}$ assigns different advantages to different tokens, its mean over all loss-bearing tokens in a rollout exactly recovers the rollout-level advantage $A_{\mathrm{image},i}$:

Lemma 1 (Advantage Preservation)**.** *The T_i-token average of $A_{\mathrm{elem},i,t}$ over the loss-bearing tokens of candidate i equals the original group-normalized GRPO advantage $A_{\mathrm{image},i}$, as in Eq.* (4)*. That is,*

$$\frac{1}{T_i} \sum_{t=1}^{T_i} A_{\mathrm{elem},i,t} = A_{\mathrm{image},i}. \quad (15)$$

Proof. From the residual definition Eq. (12), $\sum_k c_{i,k} + \nu_i = \tilde{r}_i$. Dividing by $\sigma + \epsilon_r$ and applying Eq. (13) yields

$$\sum_k Q_{i,k} + U_i = A_{\mathrm{image},i}. \quad (16)$$

By construction of $A_{\mathrm{elem},i,t}$ in Eq. (14), its T_i-token average gives the same sum:

$$\frac{1}{T_i} \sum_{t=1}^{T_i} A_{\mathrm{elem},i,t} = U_i + \sum_k Q_{i,k} = A_{\mathrm{image},i}. \quad (17)$$

Lemma 1 shows that element credit only redistributes advantage within a completion; it preserves the rollout-level image signal. For stability, we mix it with the standard image-level advantage as defined in Eq. (5).

Training Objective

The final GRPO objective uses $A_{\mathrm{mix},i,t}$ for all loss-bearing tokens; on constraint-only fields (preservation, negative) we set $\eta = 0$, reducing A_{mix} to the standard image-level advantage A_{image}.

$$\begin{aligned} \mathcal{J}(\theta) = \mathbb{E}_{i,t}[\min(\rho_{i,t} A_{\mathrm{mix},i,t}, \bar{\rho}_{i,t} A_{\mathrm{mix},i,t})] \\ - \beta\, \mathrm{KL}[\pi_\theta \| \pi_{\mathrm{ref}}], \end{aligned} \quad (18)$$

where $\rho_{i,t} = \pi_\theta(y_{i,t} \mid x, y_{i<t}) / \pi_{\theta_{\mathrm{old}}}(y_{i,t} \mid x, y_{i<t})$ is the importance ratio and $\bar{\rho}_{i,t} = \mathrm{clip}(\rho_{i,t}, 1-\epsilon_{\mathrm{clip}}, 1+\epsilon_{\mathrm{clip}})$. The element-credit decomposition contributes only through A_{mix} and introduces no additional learnable parameters.

Setting	SFT	GRPO
Base model	Qwen3-VL-8B	SFT-ed Prompt Writer
Trainable modules	LoRA + new tag rows	LoRA
LoRA (r, α)	$(16, 32)$	$(16, 32)$
Learning rate	2×10^{-5}	1×10^{-6}
Optimizer/objective	AdamW	DAPO
LR schedule	cosine	constant
Effective batch	8 responses	80 completions
Precision	BF16	BF16
Rollouts N	–	20 per product

Table 2: Training hyperparameters for supervised initialization and EAGLE-GRPO.

Type	Criteria	
Hard gates	Product identity	Logo/package text
	Product count	Unsupported claims
	Variant consistency	Severe visual damage
Quality (1–10)	Product fidelity	Background/scene fit
	Product salience	Visual communication
	Eye-catching attention	Text readability
	Creative selling point	Commercial desire
	Color harmony	Professional polish
	Layout hierarchy	

Table 3: Judge criteria: hard gates and quality dimensions.

Experiments

We first describe the training and evaluation setup, and then compare EAGLE-GRPO with strong prompt-writing baselines through benchmark and human evaluation.

Experimental Setup

Supervised initialization. We initialize the prompt writer from Qwen3-VL-8B-Instruct. The model is supervised-fine-tuned on 24,790 teacher responses from 6,240 products. For each product, the teacher generates up to four prompt profiles: sparse, balanced, visually expressive, and commercially dense. This exposes the model to both selective omission and detailed, grounded design choices.

Each response contains a short reasoning summary and an 18-field structured prompt. We add each schema tag to the tokenizer as a separate token and train its embedding and LM-head rows together with the SFT LoRA adapter. All other parameters remain frozen. The added tags provide explicit token-level boundaries for element parsing and hidden-state pooling.

GRPO training. Starting from the SFT-trained prompt writer, we train two editor-specific EAGLE-GRPO policies, one for GPT-Image-2 and one for FLUX.2 [klein]. We merge the SFT LoRA weights into the base model to preserve the prompt schema, then add a fresh LoRA adapter for GRPO. Sampled prompts are rendered by the corresponding editor and scored by Gemini-2.5-Flash using the benchmark rubric (6 hard gates and 11 quality dimensions, Table 3); failing any hard gate can zero the reward. For training stability, we adopt the DAPO objective (Yu et al. 2025) (with $\beta = 0$ and truncation masking) for all GRPO training. Table 2 summarizes the main hyperparameters.

Benchmark and Evaluation. The benchmark contains 700 held-out products from 8 market regions and 31 top-level categories, covering 367 fine-grained category paths. Its normalized Shannon evenness is 0.833 at the top level and 0.946 at the fine-grained level. Most source images are selected through judge-score filtering and manual review. We also include lower-scoring white-background images to reduce ceiling effects.

We evaluate generated images using the rubric in Table 3. To reduce the risk of overfitting the Gemini-2.5-Flash evaluator used during training, the benchmark evaluation is conducted with Gemini-2.5-Pro. The judge reports a weighted overall score and eleven quality scores on a 1–10 scale. We report the overall score together with professional polish, layout hierarchy, and commercial desire. We further validate the results through blind randomized pairwise human evaluation.

Main Results

Benchmark results. Table 4 compares different prompt writers under both image editors. EAGLE-GRPO achieves the highest overall score in both settings. With GPT-Image-2, EAGLE-GRPO reaches an overall score of 9.295, outperforming all compared baselines, including GPT-5.4 and Gemini-2.5-Pro. It also achieves the highest scores in professional polish, layout hierarchy, and commercial desire dimensions. With FLUX.2 [klein], EAGLE-GRPO achieves the highest overall

Prompt writer	Overall	Professional Polish	Layout Hierarchy	Commercial Desire
GPT-Image-2				
GPT-5.4 (Closed)	9.205	9.561	9.443	8.845
Gemini-2.5-Pro (Closed)	9.077	9.249	9.459	8.894
GPT-4o (Closed)	8.049	8.704	8.514	7.634
Qwen3-VL-32B (Open)	8.914	9.096	9.237	8.688
Qwen3-VL-32B SFT (Open)	8.779	9.097	9.136	8.346
Qwen3-VL-8B (Open)	8.095	8.832	8.700	7.826
Qwen3-VL-8B SFT (Open)	9.056	9.320	9.556	8.879
Standard GRPO (Trained)	9.130	9.516	9.383	8.678
PromptEnhancer (Trained)	8.921	9.523	9.319	8.494
EAGLE-GRPO (Trained)	**9.295**	**9.634**	**9.569**	**8.971**
FLUX.2 [klein] 9B				
GPT-5.4 (Closed)	7.963	8.524	8.408	**6.504**
Gemini-2.5-Pro (Closed)	7.035	7.144	8.017	6.299
GPT-4o (Closed)	5.475	6.084	6.393	4.679
Qwen3-VL-32B (Open)	5.084	5.156	6.036	4.138
Qwen3-VL-32B SFT (Open)	3.732	4.364	4.572	2.901
Qwen3-VL-8B (Open)	6.575	7.287	7.372	6.017
Qwen3-VL-8B SFT (Open)	7.306	7.634	8.156	6.331
Standard GRPO (Trained)	7.924	8.526	8.233	6.294
PromptEnhancer (Trained)	7.714	8.519	8.356	6.238
EAGLE-GRPO (Trained)	**7.966**	**8.551**	**8.410**	6.408

Table 4: Prompt-writer performance under GPT-Image-2 and FLUX.2 [klein] 9B. All scores are on a 0–10 scale. Overall is the weighted mean across 11 quality dimensions.

Standard GRPO

<thinking>...A right-side text block with the oven as the hero fits the appliance category and keeps the product large and readable…</thinking>
<final_prompt>...<headline>...“...Mondial Family FR-42”…in bold dark charcoal sans-serif on the right side, medium-large size…</headline>...
<feature_callouts>...“Timer até 90 min” and “Função Dourador”, each in medium-weight dark text on light-gray pills…</feature_callouts>...<final_prompt>

EAGLE-GRPO

<thinking>...dense-clean commercial layout fits this appliance…compact subtitle, a small callout group, and a few compact spec pills…</thinking>
<final_prompt>...<headline>...“...Mondial Family FR-42”...at the upper-left, large and clean, with strong hierarchy and generous spacing…</headline>...
<feature_callouts>...“Timer até 90 min” and “Função Dourador”, with simple line icons, medium-weight text, and a soft light-gray fill…</feature_callouts>...
<supporting_visuals>...subtle circular zoom inset…a close-up of the control knob…connected with a thin line…</supporting_visuals>...<final_prompt>

Figure 3: Qualitative comparison between Standard GRPO and EAGLE-GRPO on one held-out oven product using GPT-Image-2.

EAGLE-GRPO vs.	Win rate of EAGLE-GRPO GPT-Image-2	FLUX-2-Klein
GPT-5.4	52%	52%
Gemini-2.5-Pro	51%	53%
GPT-4o	82%	87%
Qwen3-VL-32B	75%	59%
Qwen3-VL-32B SFT	56%	57%
Qwen3-VL-8B	79%	56%
Qwen3-VL-8B SFT	56%	59%
PromptEnhancer	58%	57%
Standard GRPO	62%	54%

Table 5: Blind pairwise human evaluation. Each cell reports the win rate of EAGLE-GRPO over the listed baseline.

score of 7.966. It also obtains the best professional-polish and layout-hierarchy scores.

Human evaluation. Since the training reward and benchmark are both scored by automated VLM judges, we further validate our results with human evaluation to guard against judge-specific biases. Human evaluation further supports the benchmark results (Table 5). Human raters prefer EAGLE-GRPO over every evaluated comparator in both settings. In particular, against Standard GRPO, EAGLE-GRPO achieves win rates of 62% under GPT-Image-2 and 54% under FLUX.2 [klein].

Ablation Studies

We design the ablation studies to isolate which component drives the observed improvements. In particular, we examine whether the gains arise from element-level credit assignment itself and how this design compares with state-of-the-art alternatives. We conduct two controlled comparisons.

Element-Level versus Sequence-Level Credit

To isolate whether the gains arise specifically from element-level credit assignment, we compare EAGLE-GRPO with a matched Standard GRPO baseline under the same training setup. For each image editor, we use the same SFT initialization, rollout budget, reward function, optimizer settings, and 200-update schedule. The only difference is credit assignment. Standard GRPO applies one rollout-level advantage to all prompt tokens, whereas EAGLE-GRPO redistributes the advantage across prompt elements.

Element-aware credit improves the benchmark score in both cases. The overall score improves from 9.130 to 9.295 with GPT-Image-2 and from 7.924 to 7.966 with FLUX (Table 4). Human evaluation shows the same trend (Table 5).

Sustained Gains and Image Quality. Beyond final image fidelity, we examine whether element-level credit supports more sustained optimization. Figure 4 reports validation reward on a fixed 500-item set every 10 updates. Standard GRPO improves earlier but later plateaus or declines. In contrast, EAGLE-GRPO continues improving and reaches a higher validation reward under both image editors.

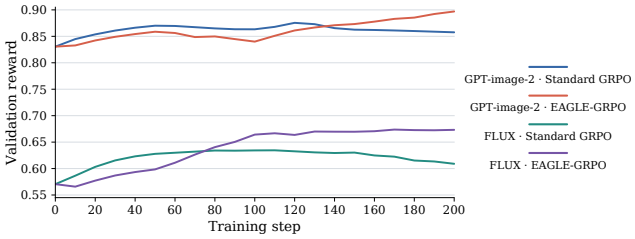


Figure 4: Validation reward, evaluated every 10 updates.

Qualitative comparison. Figure 3 compares Standard GRPO and EAGLE-GRPO on one held-out product using GPT-Image-2. Both policies identify the main product features. EAGLE-GRPO strengthens the *headline* through a larger upper-left placement with clearer hierarchy, and uses *supporting_visuals* to request a close-up inset of the control knob. The resulting image presents the product information more clearly while preserving the overall visual hierarchy.

Element-Level Credit versus Fine-Grained Scoring

To determine whether fine-grained element scoring alone is sufficient when the policy update remains sequence-level, we compare EAGLE-GRPO with an adaptation of PromptEnhancer (Wang et al. 2025), a recent state-of-the-art prompt-rewriting method based on chain-of-thought rewriting and fine-grained reward. We adapt its design to our e-commerce image-editing setting. The model is initialized with SFT, following the chain-of-thought prompt-rewriting format of the original method. During GRPO, Gemini-2.5-Flash scores each schema element from 0 to 10, and the averaged score is used as a sequence-level reward. This preserves fine-grained scoring while excluding element-level credit assignment.

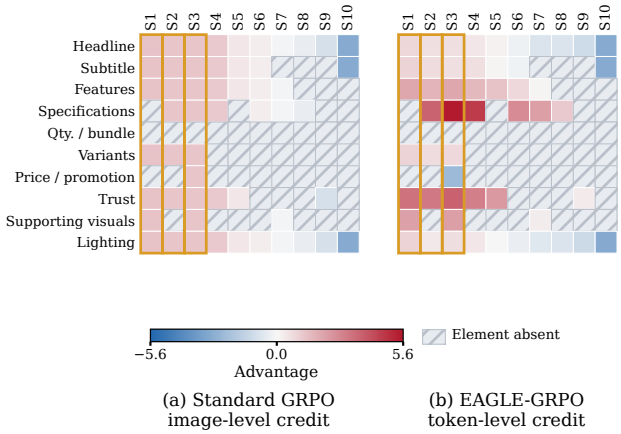
EAGLE-GRPO achieves higher benchmark scores under both editors: 9.295 versus 8.921 with GPT-Image-2, and 7.966 versus 7.714 with FLUX (Table 4). Human raters also prefer EAGLE-GRPO in 58% and 57% of comparisons (Table 5). These results reassure that the policy update also benefits from element-level credit assignment.

Element-Level Diagnostics

The following analyzes use the GPT-image-2 run. Its stronger prompt adherence makes the link between structured prompt decisions and rendered content easier to inspect.

Rollout-Specific Element Credit

Figure 5 visualizes the credits computed for 10 samples in a rollout group. In EAGLE-GRPO (Figure 5(b)), colors vary across rows within the same rollout, showing that different elements receive different credit even when the overall image reward is the same. This demonstrates that EAGLE-GRPO assigns different credits to the same element across rollouts, and these credits in turn produce different final advantages for the corresponding element tokens.



(a) Standard GRPO image-level credit

(b) EAGLE-GRPO token-level credit

Figure 5: Comparison of sample-level credit in Standard GRPO and element-specific credit in EAGLE-GRPO. Each column is one rollout; each row is a prompt element. Color indicates the credit assigned to that element in that rollout.

Category-Level Patterns

We aggregate optional-element statistics by product category to examine how often each element is selected and how its credit relates to image reward. Figure 6 reports these patterns across seven distinct categories and eight optional elements.

The clearest patterns appear for *specification* in Home & Living and *features* in Health and Sports & Outdoors (48% presence, $r = 0.78$). In contrast, *subtitle* is selected more frequently in Home & Living but has a weaker reward correlation. This shows that frequent selection does not necessarily imply stronger alignment with image quality. These correlations describe reward allocation within EAGLE-GRPO, not the causal effect of adding an element.

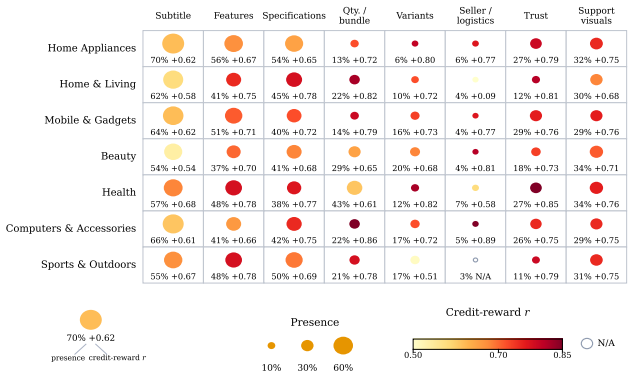


Figure 6: Category-level diagnostics for eight optional prompt elements. Marker size indicates selection frequency, darker colors indicate stronger credit–reward correlation.

Element Evolution Across Checkpoints

To examine how element-level credit affects prompt behavior, we track *promotion_information* for one fixed product, sampling 20 prompts from the SFT model and five GRPO checkpoints. Its usage rate first rises at step 40, reflecting early exploration, but the element receives negative kernel credit despite the slightly higher raw reward of prompts that contain it, suggesting that the apparent reward gain instead comes from other co-occurring elements. Continued negative credit then drives its usage to zero by step 200.

Figure 1 shows how these changes are reflected in the generated images. The step-40 image includes a generic promotional message meaning "storewide free shipping from $99," whereas the step-200 output adds a control-panel close-up and grounded product details meaning "high-temperature lock", "leak detection with automatic water shut-off", etc. This illustrates a shift from generic promotion toward product-specific information and richer visual support.

Conclusion

We presented EAGLE-GRPO, a kernel-based method that decomposes the centered image reward into per-element credits from variation already present in a standard GRPO rollout group. These credits are mapped back to their corresponding token spans while preserving per-sample advantage. We proved that the EAGLE-GRPO beats Standard GRPO and competitive prompt writers across GPT-Image-2 and FLUX.2. Element-level and longitudinal analysis further reveal which prompt decisions are reinforced or suppressed, making reward-driven optimization more targeted and interpretable.

Limitations and Future Directions. In the setting of this work, the decomposition fits 18 element kernels, each capturing element presence and semantic variation. Different product categories emphasize different elements, so a fixed 18-kernel design leaves some dimensions sparse. Future work could address this by tailoring the kernel to each category, tuning λ per category, or extending the decomposition to pairwise interaction kernels.

References

Cao, M.; Zhang, S.; Chang, X.-W.; and Precup, D. 2025. SCAR: Shapley Credit Assignment for More Efficient RLHF. *arXiv preprint arXiv:2505.20417*.

Feng, X.; Jiang, Y.; Feng, X.; Yin, D.; Qin, L.; Ye, Y.; Huang, L.; Ma, W.; Gu, Y.; Qin, C.; Qin, B.; and Kong, L. 2026. SAVOIR: Learning Social Savoir-Faire via Shapley-based Reward Attribution. In *Findings of ACL*.

Gu, C. 2002. *Smoothing Spline ANOVA Models*. Springer Series in Statistics. New York, NY: Springer. ISBN 9780387953533.

Hao, Y.; Chi, Z.; Dong, L.; and Wei, F. 2023. Optimizing Prompts for Text-to-Image Generation. In *Advances in Neural Information Processing Systems*, volume 36.

He, Y.; Wu, H.; Liu, S.; Ge, H.; Zhou, H.; Wu, K.; Zheng, Z.; Lin, Q.; Zhong, Z.; and Zhang, Y. 2026. Rethinking Token-Level Credit Assignment in RLVR: A Polarity-Entropy Analysis. *arXiv preprint arXiv:2604.11056*.

Kandasamy, K.; and Yu, Y. 2016. Additive Approximations in High Dimensional Nonparametric Regression via the SALSA. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, 69–80.

Khalifa, M.; Agarwal, R.; Logeswaran, L.; Kim, J.; Peng, H.; Lee, M.; Lee, H.; and Wang, L. 2025. Process Reward Models That Think. *arXiv preprint arXiv:2504.16828*.

Kimeldorf, G.; and Wahba, G. 1971. Some Results on Tchebycheffian Spline Functions. *Journal of Mathematical Analysis and Applications*, 33(1): 82–95.

Li, Y.; Zhang, X.; Lu, W.; Tang, Z.; Wu, M.; Luo, H.; Wu, T.; Peng, Z.; Mi, H.; Feng, Y.; Tan, N.; Huang, C.; Chen, H.; and Shen, L. 2026. Who Deserves the Reward? SHARP: Shapley Credit-based Optimization for Multi-Agent System. *arXiv preprint arXiv:2602.08335*.

Lightman, H.; Kosaraju, V.; Burda, Y.; Edwards, H.; Baker, B.; Lee, T.; Leike, J.; Schulman, J.; Sutskever, I.; and Cobbe, K. 2023. Let's Verify Step by Step. arXiv:2305.20050.

Schulman, J.; Moritz, P.; Levine, S.; Jordan, M. I.; and Abbeel, P. 2016. High-Dimensional Continuous Control Using Generalized Advantage Estimation. In *International Conference on Learning Representations*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*.

Setlur, A.; Nagpal, C.; Fisch, A.; Geng, X.; Eisenstein, J.; Agarwal, R.; Agarwal, A.; Berant, J.; and Kumar, A. 2025. Rewarding Progress: Scaling Automated Process Verifiers for LLM Reasoning. In *ICLR*.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y. K.; Wu, Y.; and Guo, D. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. arXiv:2402.03300.

Shapley, L. S. 1953. A Value for n-Person Games. In Kuhn, H. W.; and Tucker, A. W., eds., *Contributions to the Theory of Games II*, volume 28 of *Annals of Mathematics Studies*, 307–317. Princeton, NJ: Princeton University Press.

Tan, H.; and Pan, J. 2025. GTPO and GRPO-S: Token and Sequence-Level Reward Shaping with Policy Entropy. *arXiv preprint arXiv:2508.04349*.

Wahba, G. 1990. *Spline Models for Observational Data*, volume 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Philadelphia, PA: Society for Industrial and Applied Mathematics. ISBN 9780898712445.

Wang, L.; Xing, X.; Cheng, Y.; Zhao, Z.; Li, D.; Hang, T.; Li, Z.; Tao, J.; Wang, Q.; Li, R.; Chen, C.; Li, X.; Wu, M.; Deng, X.; Wang, C.; and Lu, Q. 2025. PromptEnhancer: A Simple Approach to Enhance Text-to-Image Models via Chain-of-Thought Prompt Rewriting. *arXiv preprint arXiv:2509.04545*.

Yang, H.; Zhou, Y.; Han, W.; and Shen, J. 2025. Self-Rewarding Large Vision-Language Models for Optimizing Prompts in Text-to-Image Generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, 7332–7349.

Yu, Q.; et al. 2025. DAPO: An Open-Source LLM Reinforcement Learning System at Scale. *arXiv preprint arXiv:2503.14476*.

Zheng, C.; Zhu, J.; Ou, Z.; Chen, Y.; Zhang, K.; Shan, R.; Zheng, Z.; Yang, M.; Lin, J.; Yu, Y.; and Zhang, W. 2025. A Survey of Process Reward Models: From Outcome Signals to Process Supervisions for Large Language Models. *arXiv preprint arXiv:2510.08049*.