MULTI-ASPECT REWARD BALANCE FOR DIFFUSION ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Image generation quality is inherently multidimensional, so aligning diffusion models with human preferences requires jointly optimizing multiple reward objectives. The central challenge is to improve a single model across reward dimensions through joint training with less manual reward balancing. Rollout samples can be informative for only a subset of rewards, making it important to preserve their distinct supervision during joint training. Common approaches train separate reward specialists, combine rewards with fixed weights, or use manually designed training curricula. However, these approaches do not adequately balance the influence of different rewards during joint training. We propose **MARBLE** (**M**ulti-**A**spect **R**eward **Ba**lanc**E**), which preserves reward-specific credit through independent advantage estimates and balances their influence using interactions among normalized reward gradients. Our analysis shows that a nonzero normalized refresh direction can locally reduce the participating reward losses together. Gradient diagnostics also show fewer conflicts between the refresh update and individual reward gradients. An amortized formulation of the DiffusionNFT loss, coupled with exponential moving average (EMA) coefficient smoothing, makes this adaptive coordination practical for joint training. On SD3.5 Medium with five rewards, MARBLE improves all eight evaluation metrics over both simultaneous weighted-sum training and fixed uniform coefficients. It achieves OCR 0.96 and GenEval 0.94 while retaining $0.97\times$ weighted-sum throughput, combining multi-reward gains in one model with near-baseline training speed.

1 INTRODUCTION

Image generation quality is inherently multidimensional. Reinforcement learning (RL) fine-tuning aligns diffusion models with human preferences across multiple aspects, including aesthetic quality, text-image alignment, and compositional accuracy (Liu et al., 2025; Zhang et al., 2026; Tong et al.,

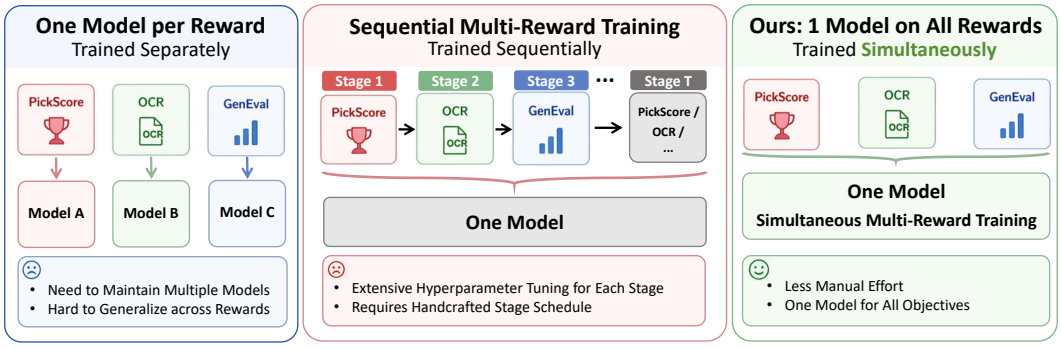

Figure 1: Comparison of multi-reward training paradigms. **Left:** Separate reward specialists require multiple models. **Middle:** Sequential training uses a reward order and stage schedule. **Right:** MARBLE jointly trains one model using gradient-based coefficient estimation, EMA smoothing, and coefficient reuse between refreshes.

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

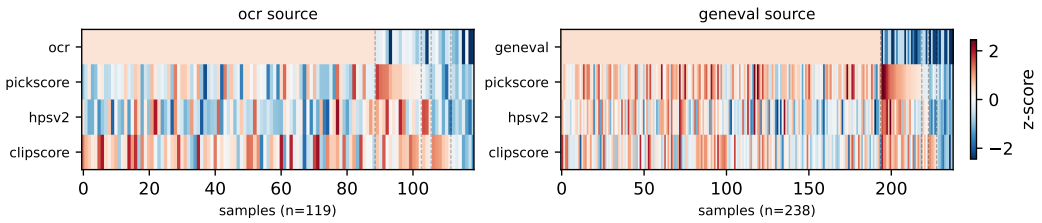

Figure 2: **Reward-specific sample information.** Each column is a sample and each row its standardized reward score $U_k(x)$. OCR and GenEval are defined on different prompt subsets. Their absence is not a negative score. The source-specific patterns show why each reward needs its own statistics: the useful supervision depends on both the sample and the reward.

2026; Zheng et al., 2025). The goal is to improve these capabilities jointly within one model. Training reward specialists separately leaves multiple models to maintain (Liu et al., 2025; Zhang et al., 2026; Tong et al., 2026). Sequential fine-tuning combines capabilities through a chosen curriculum. DiffusionNFT, for example, alternates between general-quality and GenEval prompt sets before an OCR stage, using 800, 300, 200, 200, and 100 iterations (Zheng et al., 2025). These stages use reward combinations but still require extensive manual experimentation and tuning to determine the dataset order and stage lengths. We study how one model can learn from multiple rewards simultaneously with less manual reward balancing (Figure 1).

Joint training must distinguish three issues. First, rewards have different domains of applicability: a cat image without requested text supplies no OCR supervision. We call a sample informative for a subset of rewards a *specialist sample* (Figure 2). Second, applicable rewards have different scales, so a raw sum $R(x) = \sum_k w_k R_k(x)$ can favor numerically large dimensions. Third, even standardized rewards can induce conflicting parameter updates. Together, these issues make joint training more than a matter of combining reward scores. The challenge is to preserve the useful supervision from each reward while coordinating their influence on the shared model.

Estimating and normalizing advantages separately preserves each reward's assessment of a sample (Xue et al., 2025). It does not, by itself, reconcile the parameter updates induced by those rewards. In a diagnostic on five mini-batches, the equal combination of per-reward gradients is anti-aligned with at least one reward in 80% of batches (Section 4.4). This observation motivates choosing reward weights from gradient interactions, so that balancing reflects how each objective would change the shared model.

To address this problem, we propose **MARBLE** (**M**ulti-**A**spect **R**eward **Ba**l**L**anc**E**), which chooses reward weights from a minimum-norm combination of normalized gradients (Désidéri, 2012; Sener & Koltun, 2018). Our design separates estimating how rewards should be balanced from applying that balance during training. The affine NFT loss allows the estimated weights to be applied through a single combined-advantage update, making it inexpensive to reuse them between gradient refreshes. Exponential moving average (EMA) smoothing makes the reused weights reflect multiple batches, rather than a single transient estimate.

Our contributions are:

- **Unified multi-reward diffusion alignment.** We introduce MARBLE to jointly optimize heterogeneous reward objectives within a single diffusion model, preserving reward-specific supervision while adaptively coordinating their influence during training.
- **Efficient adaptive optimization.** We derive an amortized approximation that makes gradient-based reward coordination practical for diffusion alignment, achieving $0.97\times$ the throughput of weighted-sum training.
- **Overall gains in a single model.** MARBLE achieves the best overall performance among all compared methods, ranking first on average across eight evaluation metrics. It combines strong text rendering, compositional accuracy, and visual quality in a single model.

2 Related Work

2.1 Reinforcement Learning and Multi-Reward Diffusion Alignment

Diffusion and flow models generate images through learned denoising or transport dynamics (Ho et al., 2020; Song et al., 2020b;a; Liu et al., 2022; Lipman et al., 2022). Latent modeling and transformer backbones enable high-resolution synthesis (Rombach et al., 2022; Podell et al., 2023; Esser et al., 2024; Peebles & Xie, 2023). Reinforcement learning and preference optimization are used to align models with human preferences (Schulman et al., 2017; Rafailov et al., 2023). RL adapts diffusion models to evaluable criteria through policy gradients (Black et al., 2023; Fan et al., 2023; Tong et al., 2025), group-relative objectives (Shao et al., 2024; Liu et al., 2025; Tong et al., 2026; Xue et al., 2025; He et al., 2025; Zhang et al., 2026; Li et al., 2025), or Negative-aware FineTuning with DiffusionNFT (Zheng et al., 2025).

For multiple rewards, Parrot (Lee et al., 2024) uses batch-wise Pareto sample selection, while DanceGRPO (Xue et al., 2025) normalizes rewards within groups and aggregates their advantages. MARBLE focuses on how different reward objectives jointly influence model learning, using their gradient interactions to adaptively balance these objectives during training.

2.2 Multi-Task and Multi-Objective Optimization

Shared-model optimization must account for interference between objectives (Deb, 2011; Liu & Vicente, 2024). Existing methods find minimum-norm convex combinations of gradients (Désidéri, 2012; Sener & Koltun, 2018), project conflicting components (Yu et al., 2020), optimize worst-case improvement (Liu et al., 2021), or use bargaining formulations (Navon et al., 2022). Reinforcement learning has also been applied to reasoning in large language models (Shao et al., 2024), and to active perception and multimodal reasoning in vision-language models (Zhu et al., 2025; Zhong et al., 2025). Other work studies multi-reward alignment in language and generative models (Zhou et al., 2024; Rame et al., 2023; Shi et al., 2024).

These approaches motivate treating multi-reward diffusion alignment as a problem of coordinating objectives in a shared model. In this setting, different samples can provide supervision for different rewards, so coordination must preserve each reward's learning signal while accounting for its interaction with the others. MARBLE connects this reward-specific supervision to gradient-based multi-objective optimization through separate advantage estimates and adaptive weighting. Building on periodic coefficient estimation in multi-task optimization (Navon et al., 2022), it amortizes the cost of coordination to support efficient joint training.

3 Method

MARBLE estimates reward coefficients from gradients, smooths them with EMA, and reuses them between refreshes. Refresh steps compute Equation (7) and update the coefficient EMA in Equation (11). Cached steps combine current advantages using Equation (10). This separates estimating the relative influence of rewards from applying that balance during joint training.

3.1 Preliminaries: DiffusionNFT

Let π_θ be the diffusion policy and π_{ref} the frozen pre-trained reference. A reward-regularized objective is

$$\max_\theta \ \mathbb{E}_{x\sim\pi_\theta}[R(x)] - \beta_{\text{KL}} D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}). \tag{1}$$

We use the Negative-aware FineTuning (NFT) surrogate of DiffusionNFT (Zheng et al., 2025). For a clipped advantage $A \in [-A_{\max}, A_{\max}]$, its sample loss is

$$\ell(\theta; x, t, A) = r\mathcal{L}^+(\theta) + (1-r)\mathcal{L}^-(\theta), \qquad r = \tfrac{1}{2} + \tfrac{A}{2A_{\max}}. \tag{2}$$

Here $\mathcal{L}^+$ and $\mathcal{L}^-$ are the positive and negative reconstruction losses induced by $v_\theta^+ = (1-\beta)v^{\text{old}} + \beta v_\theta$ and $v_\theta^- = (1+\beta)v^{\text{old}} - \beta v_\theta$. They share the same noisy input and are independent of A. The NFT old policy is detached, and the KL reference is frozen. The loss clips its input advantage at $A_{\max}$ before computing r.

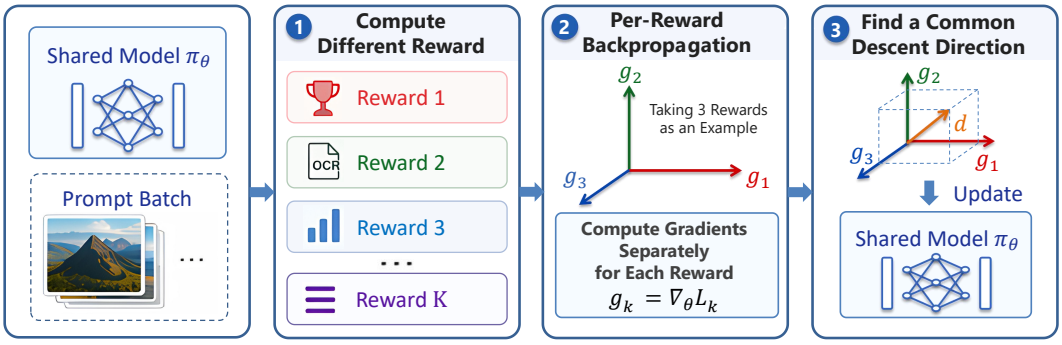

Figure 3: **Gradient-based coefficient estimation at refreshes.** Applicable rewards produce separate gradients for the normalized minimum-norm solve. The resulting coefficients enter an EMA and are reused in combined-advantage NFT updates between refreshes. Sections 3.3–3.4 complete the training procedure depicted here.

3.2 Reward Gradients and Coefficient Estimation

Separate credit and valid samples. Let $m_{ki} \in \{0, 1\}$ indicate whether reward k applies to image x_i. For each rollout, $\mu_k(\text{prompt}_i)$ is the mean reward for the same prompt, and σ_k is the standard deviation across all valid observations of reward k. Both statistics use only observations with $m_{ki} = 1$. The normalized reward and advantage are defined on these valid samples:

$$U_{ki} = \frac{R_k(x_i) - \mu_k(\text{prompt}_i)}{\sigma_k + \varepsilon}, \qquad A_{ki} = \text{clip}(U_{ki}, -A_{\max}, A_{\max}), \quad m_{ki} = 1. \tag{3}$$

When $m_{ki} = 0$, the reward is omitted from both the statistics and its reward-specific loss. This formulation accommodates both rewards that apply to all samples and rewards whose applicability depends on the prompt or task.

Let B be the batch size and t_i the sampled timestep for image x_i. The reward-specific NFT loss is

$$g_k = \nabla_\theta L_k, \qquad L_k = \sum_{\substack{i=1 \\ m_{ki}=1}}^{B} q_{ki}\ell(\theta; x_i, t_i, A_{ki}), \qquad q_{ki} = \frac{m_{ki}}{\max(1, \sum_{j=1}^{B} m_{kj})}. \tag{4}$$

Here g_k differentiates the sampled NFT loss with respect to model parameters, holding A_{ki} and q_{ki} fixed. No gradient is propagated through the reward evaluators. The loss is averaged over valid samples; a reward with no valid samples contributes zero. All rewards share the sampled images, noise, and timesteps before masking.

Normalized-gradient coefficients. We normalize each nonzero reward gradient to remove differences in magnitude:

$$\hat{g}_k = g_k / \|g_k\|. \tag{5}$$

Let K denote the number of retained reward gradients. Using the established minimum-norm construction (Désidéri, 2012; Sener & Koltun, 2018), solve

$$\alpha^* = \arg\min_{\alpha \in \Delta^K} \left\| \sum_{k=1}^{K} \alpha_k \hat{g}_k \right\|^2, \qquad \Delta^K = \{\alpha \geq 0 : \textstyle\sum_k \alpha_k = 1\}. \tag{6}$$

The estimated coefficients are held fixed during model updates. The direction $d^* = \sum_k \alpha_k^* \hat{g}_k$ satisfies $\hat{g}_k^\top d^* \geq \|d^*\|^2$. When $d^* \neq 0$, $-d^*$ is a common first-order descent direction for the sampled NFT losses L_k.

Refresh update and regularization. The refresh rule rescales the direction by the mean gradient norm:

$$d_{\text{refresh}} = \bar{n} \sum_k \alpha_k^* \frac{g_k}{\|g_k\|}, \qquad \bar{n} = \frac{1}{K} \sum_k \|g_k\|. \tag{7}$$

Positive rescaling preserves the descent direction. The reward direction is combined with KL regularization:

$$d_{\text{input}} = d_{\text{reward}} + \beta_{\text{KL}} \nabla_\theta D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}). \tag{8}$$

d_{reward} is the refresh direction above or the cached direction below. The common-descent result applies to the normalized reward direction before regularization.

3.3 Amortized Updates with Cached Coefficients

Amortizing reward coordination. MARBLE separates adapting the relative influence of rewards from updating the model under that balance. NFT advantage aggregation applies the stored coefficients to current sample advantages in a single backward pass.

NFT advantage aggregation. For fixed detached coefficients, the affine NFT loss gives the following identity for advantage aggregation.

Proposition 1. *Let $A_k \in [-A_{\max}, A_{\max}]$ and $\alpha \in \Delta^K$ be detached. If all reward losses share $\mathcal{L}^+$ and $\mathcal{L}^-$, then $\bar{A} = \sum_k \alpha_k A_k$ lies in the same interval and*

$$\nabla_\theta \ell(\theta; \bar{A}) = \sum_k \alpha_k \nabla_\theta \ell(\theta; A_k). \tag{9}$$

Gradient identity. For a fixed sample, let $r_k = 1/2 + A_k/(2A_{\max})$ and $\bar{r} = 1/2 + \bar{A}/(2A_{\max})$. Since $\sum_k \alpha_k = 1$,

$$\sum_k \alpha_k r_k = \frac{1}{2} \sum_k \alpha_k + \frac{\sum_k \alpha_k A_k}{2A_{\max}} = \frac{1}{2} + \frac{\bar{A}}{2A_{\max}} = \bar{r}.$$

Holding the advantages and coefficients fixed during differentiation gives

$$\begin{aligned} \sum_k \alpha_k \nabla_\theta \ell(\theta; A_k) &= \bar{r} \nabla_\theta \mathcal{L}^+ + (1 - \bar{r}) \nabla_\theta \mathcal{L}^- \\ &= \nabla_\theta \ell(\theta; \bar{A}). \end{aligned}$$

Clipping bounds. Since $\alpha_k \geq 0$ and $|A_k| \leq A_{\max}$,

$$|\bar{A}| = \left| \sum_k \alpha_k A_k \right| \leq \sum_k \alpha_k |A_k| \leq A_{\max} \sum_k \alpha_k = A_{\max}.$$

Thus $\bar{A} \in [-A_{\max}, A_{\max}]$ and $\bar{r} \in [0, 1]$. Clipping them again leaves their values, the NFT loss, and its parameter gradient unchanged. The same holds for each r_k, including the endpoints $A_k = \pm A_{\max}$.

Cached update with current advantages. Recomputing the reward balance at every update is costly because it requires separate reward gradients. We therefore introduce cached steps that approximate per-update reward coordination by reusing previously estimated coefficients. MARBLE refreshes the coefficients at interval N and smooths them with EMA (Equation (11)). Between refreshes, the current standardized rewards U_{ki} are combined with the stored coefficients $\bar{\alpha}$ for a single-backward update:

$$\bar{A}_i = \text{clip}\left(\sum_{k: m_{ki}=1} \bar{\alpha}_k U_{ki}, -A_{\max}, A_{\max} \right), \qquad d_{\text{cached}} = \nabla_\theta \frac{1}{B} \sum_i \ell(\theta; x_i, t_i, \bar{A}_i). \tag{10}$$

Invalid rewards contribute zero to the sum, without renormalizing the remaining coefficients. The combined loss averages over all samples in the batch.

3.4 EMA Smoothing across Refreshes

A coefficient estimate depends on the gradients of the current batch. Let j index refreshes. We initialize $\bar{\alpha}_0 = \alpha_0^*$ and smooth subsequent estimates using an exponential moving average (EMA):

$$\bar{\alpha}_j = \rho \bar{\alpha}_{j-1} + (1 - \rho) \alpha_j^*. \tag{11}$$

EMA scales each coefficient change by $1 - \rho$ and preserves simplex inputs. Refreshes use fresh coefficients, while cached computations use the smoothed vector. The interval N controls how often the balance is re-estimated, while ρ controls how strongly each estimate changes the reused coefficients.

Table 1: **Main results.** Higher is better except Avg. Rank, computed over all thirteen rows. **Bold**/underline: best/second-best distinct values, including ties. Gray: optimized reward dimensions. †Released sequential checkpoint. ‡Weighted-sum DiffusionNFT. ⋆Our NFT adaptation of the named aggregation rule. Protocol and implementation details are given in Appendix D.

Model	Rule-Based		Model-Based						Avg. Rank ↓
	GenEval	OCR	PickScore	CLIPScore	HPSv2.1	Aesthetic	ImgRwd	UniRwd	
SD-XL	0.55	0.14	22.42	0.287	0.280	5.60	0.76	2.93	9.94
SD3.5-L	0.71	0.68	22.91	0.289	0.288	5.50	0.96	3.25	6.81
FLUX.1-Dev	0.66	0.59	22.84	0.295	0.274	5.71	0.96	3.27	7.19
SD3.5-M (w/o CFG)	0.24	0.12	20.51	0.237	0.204	5.13	-0.58	2.02	13.00
+ CFG	0.63	0.59	22.34	0.285	0.279	5.36	0.85	3.03	10.19
+ FlowGRPO-GenEval	**0.95**	0.66	22.51	0.293	0.274	5.32	1.06	3.18	7.25
+ FlowGRPO-OCR	0.66	0.92	22.41	0.290	0.280	5.32	0.95	3.15	7.94
+ FlowGRPO-PickScore	0.54	0.68	23.50	0.280	0.316	5.90	1.29	3.37	6.31
+ DiffusionNFT†	0.94	0.91	**23.80**	0.293	0.331	6.01	1.49	**3.49**	2.62
+ DiffusionNFT‡	0.92	0.91	22.38	0.292	0.274	5.31	1.07	3.03	8.00
+ Parrot⋆	0.59	0.62	22.72	0.285	0.314	5.78	1.13	3.25	7.25
+ DanceGRPO-5r⋆	0.88	0.80	23.56	**0.306**	0.336	6.02	1.49	3.46	2.81
+ MARBLE	0.94	**0.96**	23.58	0.296	**0.344**	**6.05**	**1.51**	3.41	**1.69**

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We fine-tune Stable Diffusion 3.5 Medium (Esser et al., 2024) using LoRA adapters (Hu et al., 2021) (rank 32, alpha 64), the NFT loss, and AdamW with learning rate 3×10^{-4}. Training uses PickScore (Kirstain et al., 2023), HPSv2.1 (Wu et al., 2023), CLIPScore (Hessel et al., 2021), OCR accuracy, and GenEval (Ghosh et al., 2023). We reserve Aesthetic Score (Schuhmann et al., 2022), ImageReward (Xu et al., 2023), and UniReward (Wang et al., 2025) as the three metrics not optimized during training. The main run uses 16 NVIDIA H200 GPUs, $N = 10$, and coefficient EMA $\rho = 0.7$.

Each rollout draws 48 prompts, balanced across the three sources, with 24 images per prompt from a pool of 95,085 prompts. OCR and GenEval apply to their respective prompt sources through source-specific validity masks, while PickScore, HPSv2.1, and CLIPScore apply to all samples. Evaluation uses DrawBench, OCR test, and GenEval test, with counts and settings in Appendix D.

The weighted-sum, Parrot (Lee et al., 2024), and DanceGRPO (Xue et al., 2025) adaptations use a common NFT backbone, prompt pool, and training budget. Parrot prompt expansion is excluded. Sequential DiffusionNFT (Zheng et al., 2025) is evaluated using its released checkpoint. FlowGRPO (Liu et al., 2025) provides GenEval, OCR, and PickScore specialists.

4.2 MAIN RESULTS

Overall performance across capabilities. MARBLE brings together accurate text rendering, compositional understanding, and strong visual quality in a single model (Table 1). It achieves the best average rank and improves all evaluated dimensions over simultaneous weighted-sum training. Comparisons with stronger joint-training baselines also favor MARBLE across most metrics, particularly in text rendering and composition. This performance profile shows that joint training can develop task-specific capabilities alongside general image quality. The individual scores retain the trade-offs among methods, while the overall comparison supports MARBLE's ability to improve these capabilities jointly within one model.

Human evaluation and visual quality. Automatic metrics capture selected aspects of generation quality and cannot fully characterize human preference. We therefore complement them with a blind user study and qualitative comparisons. Participants give MARBLE the highest average ratings for both prompt alignment and image quality among the evaluated methods (Table 2). This reinforces the value of evaluating perceived quality directly. Appendix C.6 details the study protocol and results.

The comparisons in Figure 4 also reveal perceptual differences: several sequential DiffusionNFT outputs appear hazy and soft, while MARBLE renders sharper contours and more distinct textures.



Figure 4: Selected comparisons of text rendering, composition, and visual quality. Several sequential DiffusionNFT outputs have a hazy appearance with softened detail, while MARBLE produces sharper contours and more distinct textures, as illustrated by the ring, frog, and portrait examples.

Table 2: User study results. Participants score anonymized and randomly shuffled images on a 1–5 scale for text-image alignment and image quality. †: sequential DiffusionNFT. ‡: simultaneous weighted-sum DiffusionNFT. Higher scores are better.

Method	Text-image alignment ↑	Image quality ↑
DiffusionNFT‡	3.60	2.79
DiffusionNFT†	<u>4.26</u>	<u>3.58</u>
MARBLE	**4.63**	**4.41**

Appendix B and Figures S13–S16 provide further examples across text rendering, attributes, spatial relations, counting, and visual detail. Together, the automatic metrics, human ratings, and qualitative comparisons demonstrate that MARBLE achieves better overall generation quality than the sequential and simultaneous DiffusionNFT baselines.

Training throughput. Coefficient reuse makes adaptive reward coordination practical at near-baseline training speed (Table 3). Caching removes most of the throughput overhead from repeated coefficient estimation, with a moderate increase in memory use over weighted-sum training. The dedicated throughput benchmark shows that adaptive coordination can be sustained with little per-update slowdown. Appendix E.1 provides additional training-reward measurements over time.

4.3 Ablation Studies

We study reward weighting, gradient normalization, update frequency, and EMA smoothing (Tables 4–6). Appendix C provides additional training curves, qualitative comparisons, and ablations of reward dropout and specialist up-weighting.

Adaptive versus fixed uniform coefficients. MARBLE scores higher than fixed coefficients on all eight metrics, supporting adaptive weighting in this setting. The training curves show different learning trajectories: fixed coefficients favor faster early HPSv2.1 progress but reach lower final GenEval performance (Appendix C.3). Progress on one reward therefore gives an incomplete picture

Table 3: Training efficiency on 8×H200, normalized to weighted-sum DiffusionNFT. All rows use $K = 5$. Amortized MARBLE uses $N = 10$. Bold and underline mark the best and second-best distinct values. Ties share a mark.

Method	Speed ↑	Memory ↓
Weighted-sum DiffusionNFT	**1.00×**	**59G (1.00×)**
MARBLE w/ amortization	<u>0.97×</u>	<u>67G (1.14×)</u>
MARBLE w/o amortization	0.56×	<u>67G (1.14×)</u>

Table 4: **Fixed-weight ablation and update variants.** All rows use five rewards and the common training budget. MARBLE uses $N = 10$, $\rho = 0.7$. Fixed $\alpha = 0.2$ assigns equal coefficients to all five rewards and scores lower on all eight metrics. Per-step solving evaluates a different update frequency.

Variant	Rule-Based		Model-Based					
	GenEval	OCR	PickScore	CLIPScore	HPSv2.1	Aesthetic	ImgRwd	UniRwd
MARBLE (amortized)	**0.94**	**0.96**	**23.58**	**0.296**	**0.344**	**6.05**	**1.51**	**3.41**
w/o gradient normalization				**Collapse**				
Fixed $\alpha = 0.2$	0.86	0.89	<u>23.51</u>	0.285	<u>0.329</u>	<u>6.01</u>	<u>1.44</u>	<u>3.38</u>
Solve α every step	<u>0.92</u>	<u>0.92</u>	21.80	<u>0.289</u>	0.294	5.45	1.13	3.15

of joint learning. Adapting the balance allows the relative influence of rewards to change as their learning signals evolve. Gradient normalization serves a different role: it puts the reward gradients on a common scale before their directions are compared. Removing gradient normalization leads to complete collapse of the generated outputs (Table 4).

Table 5: **Sensitivity to the amortization interval.** We compare different values of N, which controls how often the full harmonization procedure refreshes α^*. Smaller intervals refresh the balancing coefficients more frequently, while larger intervals reuse older coefficients for more update steps.

Interval N	Rule-Based		Model-Based					
	GenEval	OCR	PickScore	CLIPScore	HPSv2.1	Aesthetic	ImgRwd	UniRwd
$N = 5$	<u>0.92</u>	**0.96**	<u>23.19</u>	<u>0.293</u>	<u>0.332</u>	<u>5.94</u>	<u>1.43</u>	<u>3.35</u>
$N = 10$	**0.94**	**0.96**	**23.58**	**0.296**	**0.344**	**6.05**	**1.51**	**3.41**
$N = 20$	<u>0.92</u>	<u>0.93</u>	22.85	0.290	0.320	5.73	1.39	3.23

Update frequency and smoothing. Table 5 shows comparable quality for $N = 5$ and $N = 10$, but lower scores for $N = 20$. We choose $N = 10$ because it retains generation quality with half as many coefficient estimates as $N = 5$. Longer reuse can delay adaptation to changing reward gradients. Amortization also improves training throughput (Table 3). EMA controls the size of coefficient changes: $\rho = 0.7$ leads on seven metrics, while $\rho = 0.1$ scores higher on Aesthetic Score (Table 6). We therefore use $\rho = 0.7$ for smoothing.

Amortized versus per-step updates. Table 4 shows that amortized MARBLE outperforms per-step solving on all eight metrics. Per-step solving relies on stochastic gradient estimates: finite rollouts introduce sampling noise, and diffusion training adds variation from randomly sampled timesteps and injected noise. Recomputing coefficients at every step can transfer this variability into the reward weights, making them respond to transient sampling fluctuations as well as persistent reward interactions. Amortization limits these frequent adjustments, while EMA smooths estimates across refreshes. Current advantages still guide every update under the reused weights. We interpret the observed gain as a benefit of reducing the reward balance's sensitivity to stochastic gradient estimates. Appendix E.3 provides further solver comparisons.

Table 6: **EMA decay sensitivity.** $\rho = 0.7$ achieves the best overall performance across the eight evaluation metrics.

EMA decay	Rule-Based		Model-Based					
	GenEval	OCR	PickScore	CLIPScore	HPSv2.1	Aesthetic	ImgRwd	UniRwd
$\rho = 0.1$	0.86	0.80	22.88	0.281	0.302	**6.17**	1.30	3.12
$\rho = 0.3$	0.88	0.84	22.98	0.288	0.305	5.83	1.34	3.14
$\rho = 0.5$	0.93	0.95	23.27	0.293	0.327	5.91	1.46	3.37
$\rho = 0.7$	**0.94**	**0.96**	**23.58**	**0.296**	**0.344**	6.05	**1.51**	**3.41**
$\rho = 0.9$	0.90	0.89	23.32	0.292	0.324	5.99	1.45	3.34

4.4 Gradient Coordination Analysis

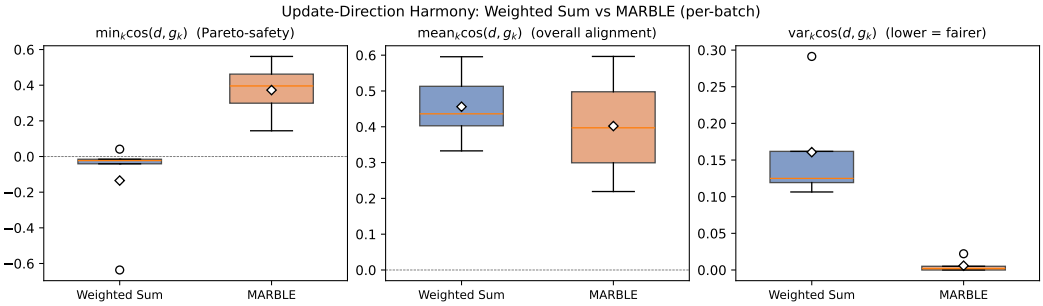

Figure 5: **Update-direction harmony.** Minimum, mean, and variance of the cosine similarities between the shared update direction and individual reward gradients on five mini-batches. We compare weighted-sum updates with MARBLE; Table S1 reports the aggregate statistics.

Alignment with individual rewards. For a mini-batch, let g_k denote the gradient of the NFT loss for reward R_k and d the shared update direction. A negative $\cos(d, g_k)$ means that the shared direction is opposed to that reward's gradient. Figure 5 compares the minimum, mean, and variance of these similarities across rewards.

Conflicts and balance across rewards. As reported in Table S1, weighted-sum updates are opposed to at least one reward gradient in 80% of the five measured mini-batches, compared with 0% for MARBLE. MARBLE raises the average worst-reward cosine from -0.135 to $+0.372$ and reduces the across-reward variance from 0.161 to 0.006, while lowering the mean cosine from $+0.456$ to $+0.401$. These results show that MARBLE improves alignment with the least-aligned reward gradient and achieves a more balanced update across rewards. All five reward gradients remain positively aligned with MARBLE's shared direction in these batches. This balance comes with a modest reduction in average alignment. Appendix C.2 provides the definitions and aggregate statistics.

5 Conclusion

We introduced **MARBLE**, which preserves reward-specific supervision and coordinates model updates using normalized gradient interactions. To our knowledge, we are the first to reveal conflicts between reward objectives in the gradient space of diffusion alignment and propose a solution to address them. On SD3.5 Medium, MARBLE combines accurate text rendering, compositional understanding, and strong visual quality in a single model. Automatic evaluations, human ratings, and qualitative comparisons together demonstrate better overall generation quality than the baselines. With near-baseline training throughput, these results support adaptive reward coordination as an effective and computationally practical approach to multi-aspect diffusion alignment.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing mathematical proofs. We did not use these tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, clean or reformat datasets, or conduct qualitative or thematic data analysis. Additionally, we used these tools for grammar correction, manuscript formatting, revising the wording and organization of the manuscript, and organizing the code. We reviewed the AI-assisted revisions for clarity and consistency with our intended meaning. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not raise ethical concerns. Our experiments focus on multi-reward alignment and evaluation of text-to-image generation. All datasets used in our experiments are publicly available, and no sensitive data are involved. The human evaluation involves anonymous participants rating generated images and does not involve sensitive information.

REPRODUCIBILITY STATEMENT

Section 4.1 and Appendix D describe the training hyperparameters, prompt sources, and evaluation datasets. Section 3.3 derives the advantage-aggregation identity and its clipping bounds, then specifies the cached update. The ablations in Section 4.3 and Appendix C report the effects of fixed weighting, refresh interval, and EMA decay. Appendix C.3 also provides detailed training curves showing reward trajectories and the evolution of the balancing coefficients. We will release the code and model checkpoints.

REFERENCES

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

Kalyanmoy Deb. Multi-objective optimisation using evolutionary algorithms: an introduction. In *Multi-objective evolutionary optimisation for product design and manufacturing*, pp. 3–34. Springer, 2011.

Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus. Mathématique*, 350(5-6):313–318, 2012.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36:79858–79885, 2023.

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

Xiaoxuan He, Siming Fu, Yuke Zhao, Wanli Li, Jian Yang, Dacheng Yin, Fengyun Rao, and Bo Zhang. Tempflow-grpo: When timing matters for grpo in flow models. *arXiv preprint arXiv:2508.04324*, 2025.

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.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

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. *Advances in neural information processing systems*, 36:36652–36663, 2023.

Seung Hyun Lee, Yinxiao Li, Junjie Ke, Innfarn Yoo, Han Zhang, Jiahui Yu, Qifei Wang, Fei Deng, Glenn Entis, Junfeng He, Gang Li, Sangpil Kim, Irfan Essa, and Feng Yang. Parrot: Pareto-optimal multi-reward reinforcement learning framework for text-to-image generation. *arXiv preprint arXiv:2401.05675*, 2024. URL https://arxiv.org/abs/2401.05675.

Junzhe Li, Yutao Cui, Tao Huang, Yinping Ma, Chun Fan, Yiming Cheng, Miles Yang, Zhao Zhong, and Liefeng Bo. Mixgrpo: Unlocking flow-based grpo efficiency with mixed ode-sde. *arXiv preprint arXiv:2507.21802*, 2025.

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

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in neural information processing systems*, 34:18878–18890, 2021.

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.

Suyun Liu and Luis Nunes Vicente. The stochastic multi-gradient algorithm for multi-objective optimization and its application to supervised machine learning. *Annals of Operations Research*, 339(3):1119–1148, 2024.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. *arXiv preprint arXiv:2202.01017*, 2022.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Alexandre Rame, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. *Advances in Neural Information Processing Systems*, 36:71095–71134, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35: 25278–25294, 2022.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Ruizhe Shi, Yifang Chen, Yushi Hu, Alisa Liu, Hannaneh Hajishirzi, Noah A Smith, and Simon S Du. Decoding-time language model alignment with multiple objectives. *Advances in Neural Information Processing Systems*, 37:48875–48920, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b.

Yunze Tong, Didi Zhu, Zijing Hu, Jinluan Yang, and Ziyu Zhao. Noise projection: Closing the prompt-agnostic gap behind text-to-image misalignment in diffusion models. *arXiv preprint arXiv:2510.14526*, 2025.

Yunze Tong, Mushui Liu, Canyu Zhao, Wanggui He, Shiyi Zhang, Hongwei Zhang, Peng Zhang, Jinlong Liu, Ju Huang, Jiamang Wang, et al. Alleviating sparse rewards by modeling step-wise and long-term sampling effects in flow-based grpo. *arXiv preprint arXiv:2602.06422*, 2026.

Yibin Wang, Yuhang Zang, Hao Li, Cheng Jin, and Jiaqi Wang. Unified reward model for multimodal understanding and generation. *arXiv preprint arXiv:2503.05236*, 2025.

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.

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. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

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.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33:5824–5836, 2020.

Liyu Zhang, Kehan Li, Tingrui Han, Tao Zhao, Yuxuan Sheng, Shibo He, and Chao Li. Op-grpo: Efficient off-policy grpo for flow-matching models. *arXiv preprint arXiv:2604.04142*, 2026.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. Diffusionnft: Online diffusion reinforcement with forward process. *arXiv preprint arXiv:2509.16117*, 2025.

Hao Zhong, Muzhi Zhu, Zongze Du, Zheng Huang, Canyu Zhao, Mingyu Liu, Wen Wang, Hao Chen, and Chunhua Shen. Omni-r1: Reinforcement learning for omnimodal reasoning via two-system collaboration. *arXiv preprint arXiv:2505.20256*, 2025.

Zhanhui Zhou, Jie Liu, Jing Shao, Xiangyu Yue, Chao Yang, Wanli Ouyang, and Yu Qiao. Beyond one-preference-fits-all alignment: Multi-objective direct preference optimization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 10586–10613, 2024.

Muzhi Zhu, Hao Zhong, Canyu Zhao, Zongze Du, Zheng Huang, Mingyu Liu, Hao Chen, Cheng Zou, Jingdong Chen, Ming Yang, et al. Active-o3: Empowering multimodal large language models with active perception via grpo. *arXiv preprint arXiv:2505.21457*, 2025.

A Appendix Overview

This appendix presents qualitative comparisons, ablation studies, training and evaluation settings, additional training and gradient analyses, and future directions. The contents are organized as follows:

- **Appendix B: Additional Qualitative Comparisons.** We present additional MARBLE outputs and comparisons with baselines across text rendering, object attributes, spatial relations, counting, and visual detail.
- **Appendix C: Additional Ablations and Analyses.** We summarize the main findings and provide the following ablations and analyses:
  - **Appendix C.1: Key Insights and Takeaways.** We summarize findings on reward-gradient conflicts, adaptive weighting, amortization, and the complementary roles of automatic metrics and human ratings.
  - **Appendix C.2: Update-Direction Diagnostics.** We provide definitions and aggregate statistics for the update-direction comparison in Section 4.4.
  - **Appendix C.3: Training Dynamics and Coefficient Adaptation.** We show reward and coefficient trajectories during training and compare fixed uniform coefficients with adaptive weighting.
  - **Appendix C.4: EMA Decay for Coefficient Smoothing.** We provide training curves and generated images for the EMA decay settings evaluated in Table 6.
  - **Appendix C.5: Alternative Heuristic Strategies.** We compare MARBLE with heuristic reward-balancing strategies, including reward dropout and specialist up-weighting, with fixed-uniform coefficients studied separately.
  - **Appendix C.6: Human Preference Evaluation.** We describe the blind user-study protocol and discuss the ratings reported in Table 2.
- **Appendix D: Training and Evaluation Settings.** We summarize training hyperparameters, prompt sources, source-balanced sampling, and the evaluation protocol.
- **Appendix E: Additional Training and Gradient Analyses.** We compare training progress across methods and random seeds, evaluate alternative gradient solvers and per-step coefficient estimation, and measure alignment between cached and fresh directions.
- **Appendix F: Future Work.** We discuss exploring more rewards and evaluating MARBLE in video generation.

B Additional Qualitative Comparisons



Figure S1: Additional selected MARBLE examples of text rendering, attribute binding, spatial layout, and counting.

Automatic metrics alone do not fully capture how viewers perceive an image. To complement the human study in Appendix C.6, Figure S1 presents additional MARBLE outputs, and Figures S13–S16 provide four comparison grids covering text rendering, object attributes, spatial relations, counting, and visual detail.

The sequential DiffusionNFT results appear hazy, with blurred contours and less distinct textures. MARBLE produces sharper object boundaries and more distinct detail, visible in the refrigerator and chairs in Figure S13 and the portraits in Figure S16. These visual differences complement MARBLE's higher human-rated image quality in Table 2.

C Additional Ablations and Analyses

C.1 Key Insights and Takeaways

- **Reward gradients can conflict during joint training.** Figure 5 shows that averaging reward gradients can produce a direction opposed to an individual reward's gradient. MARBLE reduces these conflicts by adapting the combination coefficients to the reward gradients, allowing the shared update to better accommodate all reward objectives.
- **Adaptive weighting improves joint performance.** MARBLE outperforms the weighted-sum baseline on all five optimized rewards and fixed coefficients $\alpha_k = 0.2$ on all eight evaluation metrics (Tables 1 and 4).
- **Amortization improves efficiency and generation quality.** Reusing coefficients and smoothing them with EMA can make the reward balance less sensitive to noisy gradient estimates from diffusion training and rollouts. Longer refresh intervals reduce quality (Table 5), showing the importance of updating the balance as training progresses.
- **Automatic metrics and human ratings provide complementary evidence.** MARBLE achieves the best average rank in Table 1, while different methods lead on individual metrics. The human study compares three methods and directly evaluates perceived image quality (Appendix C.6). Together, these evaluations capture both performance on specific capabilities and the quality perceived by viewers.

C.2 Update-Direction Diagnostics

Figure 5 in Section 4.4 compares the per-batch distributions. Table S1 reports their aggregate statistics.

Table S1: **Update-direction harmony statistics.** Averages over $n = 5$ mini-batches. ↑ means larger is better. ↓ means smaller is better.

Statistic	**Weighted sum**	MARBLE	Δ
Worst-reward alignment $\min_k \cos(d, g_k)$ ↑	$\underline{-0.135}$	$\mathbf{+0.372}$	$+0.507$
Average alignment $\mathrm{mean}_k \cos(d, g_k)$ ↑	$\mathbf{+0.456}$	$\underline{+0.401}$	-0.055
Alignment imbalance $\mathrm{var}_k \cos(d, g_k)$ ↓	$\underline{+0.161}$	$\mathbf{+0.006}$	-0.155
Conflict rate $P(\min_k \cos(d, g_k) < 0)$ ↓	$\underline{0.800}$	$\mathbf{0.000}$	-0.800

For a mini-batch, let g_k denote the gradient of the NFT loss for reward R_k, and let d denote the update direction produced by a multi-reward training rule. If $\cos(d, g_k) < 0$ for some reward k, then the shared update is anti-aligned with that reward's own gradient on the same batch. Table S1 shows that the weighted-sum direction has a negative worst-reward cosine on average and produces a negative worst-reward alignment in 80% of the measured mini-batches. In contrast, the harmonized direction raises the worst-reward cosine from -0.135 to $+0.372$ and eliminates negative-minimum batches in this measurement, while lowering the mean cosine from $+0.456$ to $+0.401$. Its across-reward variance is also much smaller (0.006 vs. 0.161), indicating that the update direction is more evenly aligned with the five reward gradients.

C.3 Training Dynamics and Coefficient Adaptation

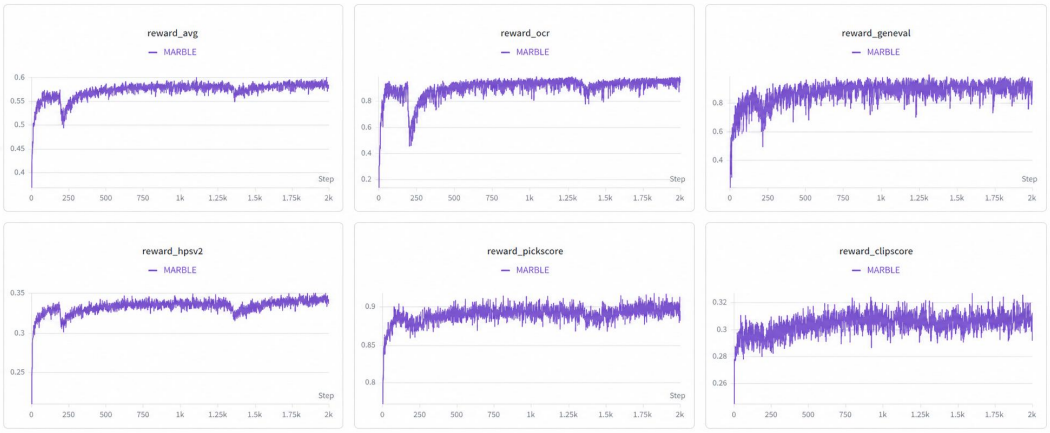

Figure S2: Training curves of MARBLE across the five optimized rewards.



Figure S3: Dynamics of the smoothed balancing coefficients $\bar{\alpha}_{t,k}$ during training. The uniform baseline for five rewards is 0.2. The learned coefficients do not vary directly with raw reward values, but assign different weights to different rewards. The curve `pareto_fallback_used` records activations of the NFT interpolation clamp, which clips coefficients outside $[0, 1]$ to the nearest endpoint. The curve remains zero throughout training, indicating that this clamp is never activated in the plotted run. Thus, clipping does not alter the NFT loss or its parameter gradient in this run.

We complement the fixed-uniform coefficient ablation in Section 4.3 with a closer view of how the smoothed coefficients $\bar{\alpha}_{t,k}$ evolve during training.

Uniform $\alpha = 0.2$ is competitive on easy rewards but loses on specialists. Figure S4 shows the per-reward learning curves under fixed $\alpha_k = 0.2$ and under MARBLE. On HPSv2.1, an easy image-quality reward, fixed $\alpha_k = 0.2$ trains slightly faster than MARBLE in the early iterations, while MARBLE converges to a higher final value. On GenEval, a harder specialist reward, fixed $\alpha_k = 0.2$ trains more slowly and ends at a lower score than MARBLE, matching the "Fixed $\alpha = 0.2$" row of Table 4.

$\bar{\alpha}_{t,k}$ is related to optimization difficulty. As shown in Figure S2 and S3, we do not observe a direct relationship between $\bar{\alpha}_{t,k}$ and the corresponding raw reward value. Instead, $\bar{\alpha}_{t,k}$ appears to be related, to some extent, to how hard each reward is to optimize (Figure S3). On HPSv2.1, which the base

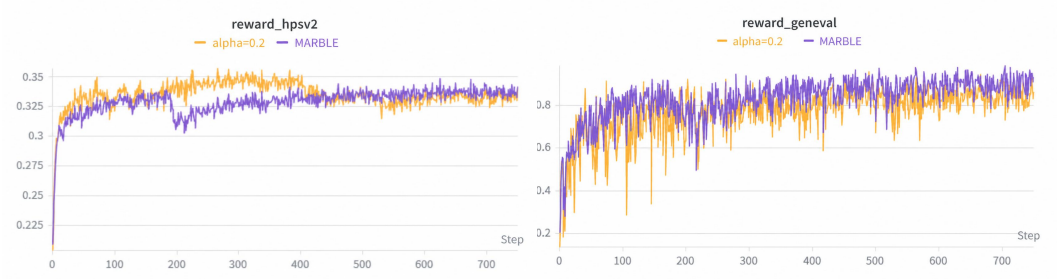

Figure S4: Learning curves for fixed coefficients $\alpha_k = 0.2$ and MARBLE. Fixed coefficients make faster early progress on HPSv2.1 and finish lower on GenEval. Table 4 reports the corresponding eight-metric comparison.

model already largely satisfies, $\bar{\alpha}_{t,k}$ stays below the uniform 0.2 baseline for most of training. On GenEval, which demands precise compositional correctness, $\bar{\alpha}_{t,k}$ often rises to around 0.3. Figure S3 also reports `pareto_fallback_used`, which records whether clipping is activated for the NFT interpolation coefficient. Before clipping, this coefficient is $r = 1/2 + A/(2A_{\max})$, where A is the input advantage. An advantage outside $[-A_{\max}, A_{\max}]$ places r outside $[0, 1]$, so the clamp sets it to the nearest endpoint. The recorded value stays at zero throughout training, indicating that no such clipping is activated in the plotted run. The interpolation therefore retains its affine dependence on the advantage, and clipping does not alter the NFT loss or its parameter gradient. Together, these observations suggest that the learned coefficients can shift optimization emphasis away from easier rewards and toward harder specialist rewards, helping the five rewards progress more evenly during training (Figure S2).

C.4 EMA Decay for Coefficient Smoothing

The EMA decay ρ controls the adaptivity and stability in coefficient smoothing. Smaller values make the coefficients more responsive to the current gradient geometry, but also more sensitive to mini-batch noise, leading to larger fluctuations in the reward curves. Larger values produce smoother coefficient trajectories, but may become overly inertial and adapt slowly when the relative difficulty of rewards changes during training. We evaluate $\rho \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, with the quantitative results reported in Table 6, the training curves shown in Figure S5, and qualitative visualizations provided in Figure S6. Across these evaluations, $\rho = 0.7$ provides the best overall performance: it maintains stable optimization, achieves strong performance, and produces visually more faithful and coherent samples. We therefore use $\rho = 0.7$ as the default setting in all main experiments.

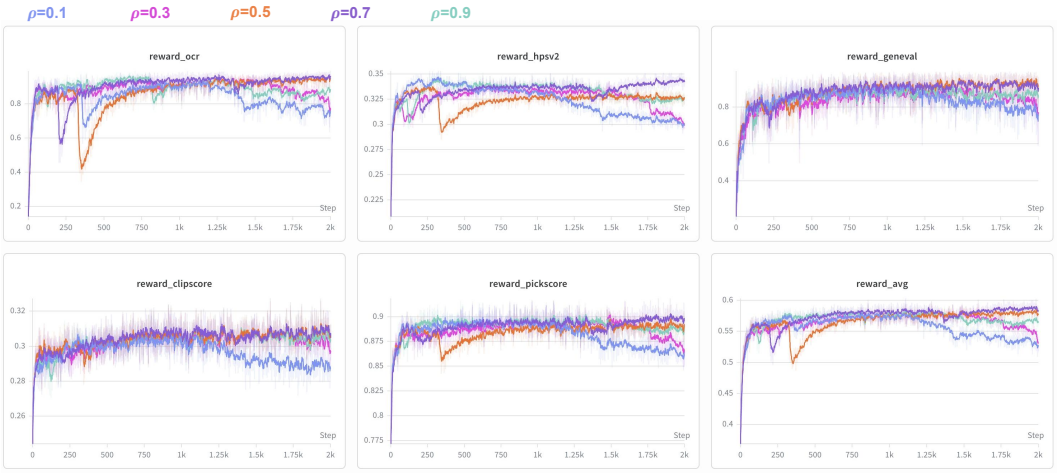

Figure S5: Sensitivity to EMA decay ρ. $\rho = 0.7$ achieves the best overall performance.



Figure S6: Qualitative comparison of different EMA decay values ρ.

Table S2: **Alternative reward-balancing strategies.** All variants use the same five rewards and training budget. The weights (1:1:1:2:2) correspond to PickScore, HPSv2.1, CLIPScore, OCR, and GenEval, respectively.

Method	Rule-Based		Model-Based					
	GenEval	OCR	PickScore	CLIPScore	HPSv2.1	Aesthetic	ImgRwd	UniRwd
MARBLE	**0.94**	**0.96**	**23.58**	**0.296**	**0.344**	**6.05**	**1.51**	**3.41**
Reward Dropout	0.80	0.61	22.31	0.273	0.255	5.74	0.81	2.98
Reward Weighting (1:1:1:2:2)	0.90	0.92	23.06	0.287	0.340	5.82	1.37	3.35

C.5 ALTERNATIVE HEURISTIC STRATEGIES

Table S2 compares reward dropout and specialist up-weighting against MARBLE. The fixed-uniform ablation is reported separately in Table 4. MARBLE is higher on all eight listed metrics than the two heuristic rows. These additional comparisons complement the fixed-coefficient ablation.

C.6 Human Preference Evaluation

Automatic metrics provide useful proxies for specific aspects of generation quality, but they do not fully capture human preference. Table 1 illustrates the need for complementary evaluation: sequential DiffusionNFT obtains higher PickScore and UniReward, whereas MARBLE achieves the best average rank and higher scores on five other metrics. We therefore supplement automatic evaluation with a blind user study and qualitative comparisons to assess perceived image quality and prompt alignment directly.

To obtain a more comprehensive assessment of human preference, we conduct a blind rating-based user study. For each method, we randomly sample 30 generated images. All images are anonymized, randomly shuffled, and shown with their corresponding prompts. A total of 20 anonymous participants who are unrelated to the project independently score each image on a 1–5 scale along two axes: text-image alignment and image quality. Higher scores indicate better perceived performance. We report the average score over all ratings in Table 2.

MARBLE receives the highest average rating on both axes among the three evaluated methods. Its higher human-rated image quality, despite sequential DiffusionNFT's higher PickScore and UniReward, shows why automatic scores should be considered alongside direct human judgments. We report descriptive mean ratings over the sampled images. The 20 ratings of each image are repeated assessments of the same generated output. Taken together, the automatic evaluations (Table 1), human ratings (Table 2), and extensive qualitative comparisons (Appendix B) demonstrate that MARBLE achieves the best overall generation quality among the evaluated methods.

D Training and Evaluation Settings

We fine-tune SD3.5 Medium with LoRA and the DiffusionNFT objective. Table S3 summarizes the main training configuration. The additional GPU-hour comparisons in Appendix E use eight GPUs per run.

Table S3: Main training settings. The refresh interval is measured in optimizer steps.

Setting	Value	Setting	Value
Backbone	SD3.5 Medium	Optimizer	AdamW
LoRA rank	32	LoRA alpha	64
Learning rate	3×10^{-4}	Training hardware	16 H200 GPUs
Prompts per rollout	48 (16 per source)	Images per prompt	24
Images per rollout	1,152	Clipping bound $A_{\max}$	5
Refresh interval N	10	Coefficient EMA ρ	0.7

Table S4: Training prompt sources. Pool shares describe dataset composition; training samples the three sources equally. The three general rewards are PickScore, HPSv2.1, and CLIPScore.

Source	Prompts	Pool share	Valid rewards
PickScore	25,432	26.7%	Three general
OCR	19,653	20.7%	Three general + OCR
GenEval	50,000	52.6%	Three general + GenEval
Total	95,085	100%	At most four per prompt

Prompt sources and evaluation. Each training rollout draws 48 prompts with 24 images per prompt, giving 1,152 images. We use source-balanced sampling, drawing 16 prompts per source. Training and evaluation both use the HPSv2.1 checkpoint. Separate reward trackers compute prompt-specific means and reward-specific standard deviations from valid rollout observations.

The controlled joint-training methods share the backbone, LoRA configuration, NFT objective, prompt pool, rollout workload, optimizer-step budget, and KL configuration. DrawBench generates 5

Table S5: Evaluation sets. All use 512×512 images, 40 steps, and guidance 1.0.

Set	Prompts	Images	Metrics
DrawBench	200	1,000	Six model-based metrics
OCR test	1,018	1,018	OCR
GenEval test	553	2,212	GenEval

images for each of 200 prompts. OCR test generates one image per prompt. GenEval test generates 4 images for each of 553 prompts.

E Additional Training and Gradient Analyses

This appendix compares training progress across methods and random seeds, evaluates alternative gradient solvers and per-step coefficient estimation, and measures agreement between cached and fresh directions.

E.1 Cross-Method Training Measurements

As shown in Figure S7, MARBLE achieves higher training rewards than the compared baselines. The weighted sum reaches the two lowest reward thresholds sooner, while MARBLE reaches the three higher thresholds in fewer steps (Figure S8). Sequential DiffusionNFT is evaluated using its released checkpoint in Table 1.

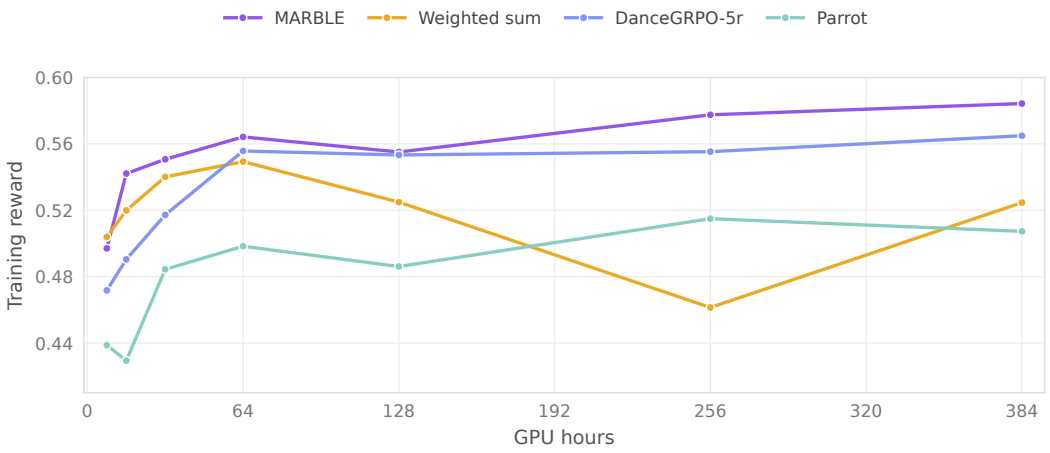

Figure S7: Training reward versus GPU hours for MARBLE and joint-training baselines. Each curve uses the logged reward aggregate from one training run.

E.2 Training Stability Across Random Seeds

We evaluate MARBLE with two independent random seeds to assess training stability. Both runs make comparable progress and reach similar reward levels (Figure S9). This consistency across runs supports the reproducibility of MARBLE's training outcomes.

E.3 Alternative Solvers and Per-Step Coefficient EMA

We evaluate alternative gradient solvers, including PCGrad (Yu et al., 2020), CAGrad (Liu et al., 2021), and NashMTL (Navon et al., 2022). At matched rollout workload, these alternatives require 2.1–2.7 times as much GPU time per optimizer step as MARBLE. Figure S10 shows that MARBLE

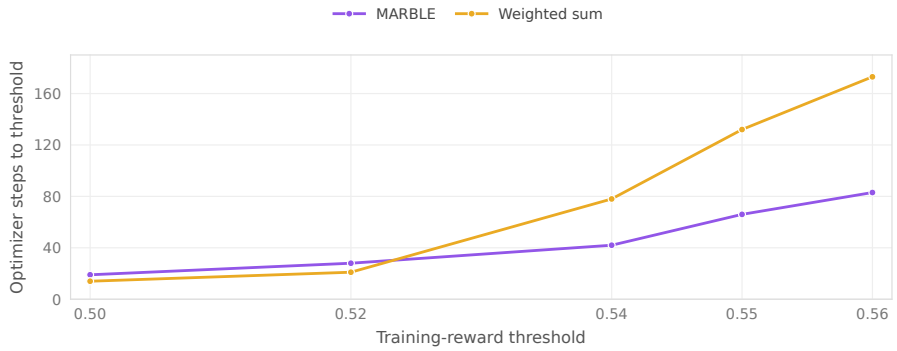

Figure S8: Optimizer steps needed to first reach each training-reward threshold, using a 10-step moving average.

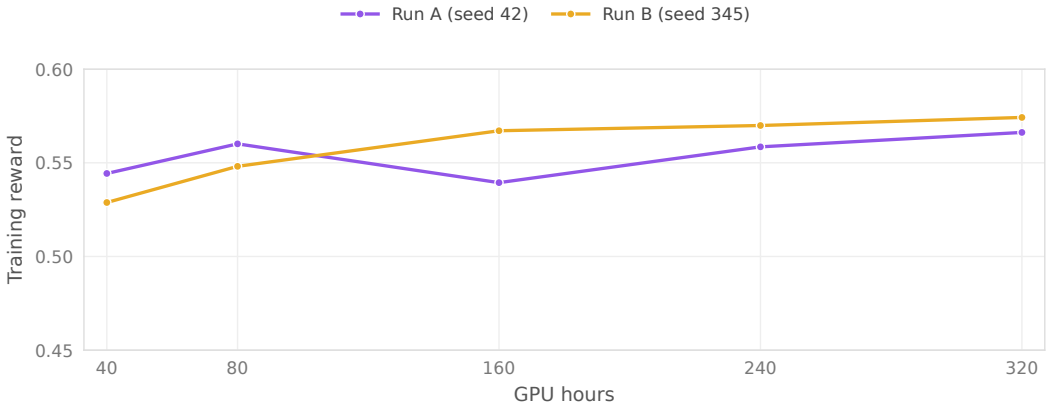

Figure S9: Training progress of two independent MARBLE runs with two different seeds.

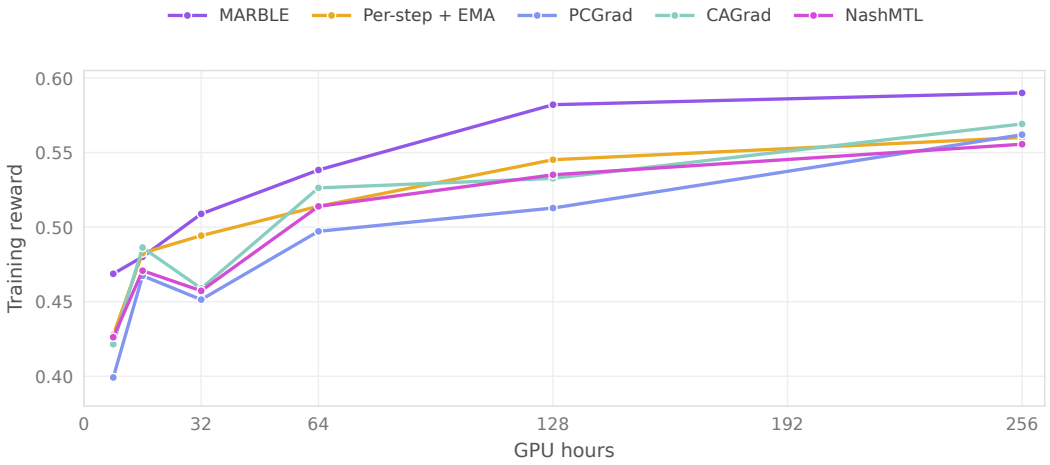

Figure S10: Training-reward comparison of alternative solvers against GPU hours.

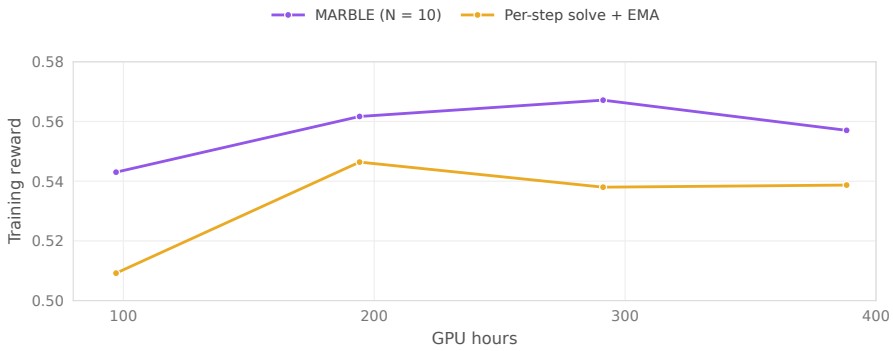

Figure S11: Training reward versus GPU hours for MARBLE and per-step coefficient estimation with EMA.

improves training rewards more efficiently with GPU time. These efficiency results motivate our choice of MGDA for MARBLE.

At matched GPU-hour budgets, MARBLE achieves higher training rewards than per-step coefficient estimation with EMA (Figure S11). By reusing coefficients across updates, MARBLE reduces the cost per step and makes faster progress in reward optimization. The curves are compared at common GPU-hour coordinates using linear interpolation.

E.4 Agreement of Cached and Fresh Directions

The update directions at cached steps remain closely aligned with freshly solved directions (Figure S12), with a mean cosine similarity of 0.814 across the refresh interval. This agreement shows that reusing coefficients largely preserves the direction of a fresh solve while avoiding coefficient estimation at every update.

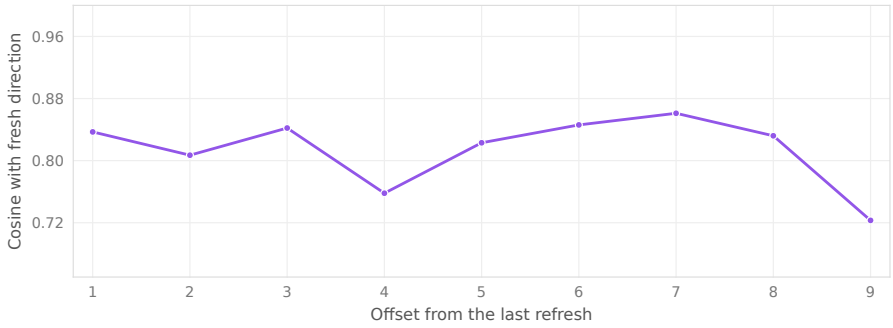

Figure S12: Cosine similarity between the applied cached reward direction and a freshly solved direction. Each point averages 10 complete refresh windows at the indicated offset.

F Future Work

Our experiments optimize five rewards on SD3.5 Medium. Future work includes exploring a broader set of rewards to capture additional aspects of visual quality and human preferences. This would allow us to study whether MARBLE maintains balanced improvements as the number and diversity of reward objectives increase. Another direction is to evaluate MARBLE's effectiveness in video generation, where temporal consistency and motion quality introduce additional objectives alongside visual quality and text alignment. We will explore whether MARBLE can jointly improve visual quality, temporal consistency, and motion quality in generated videos.

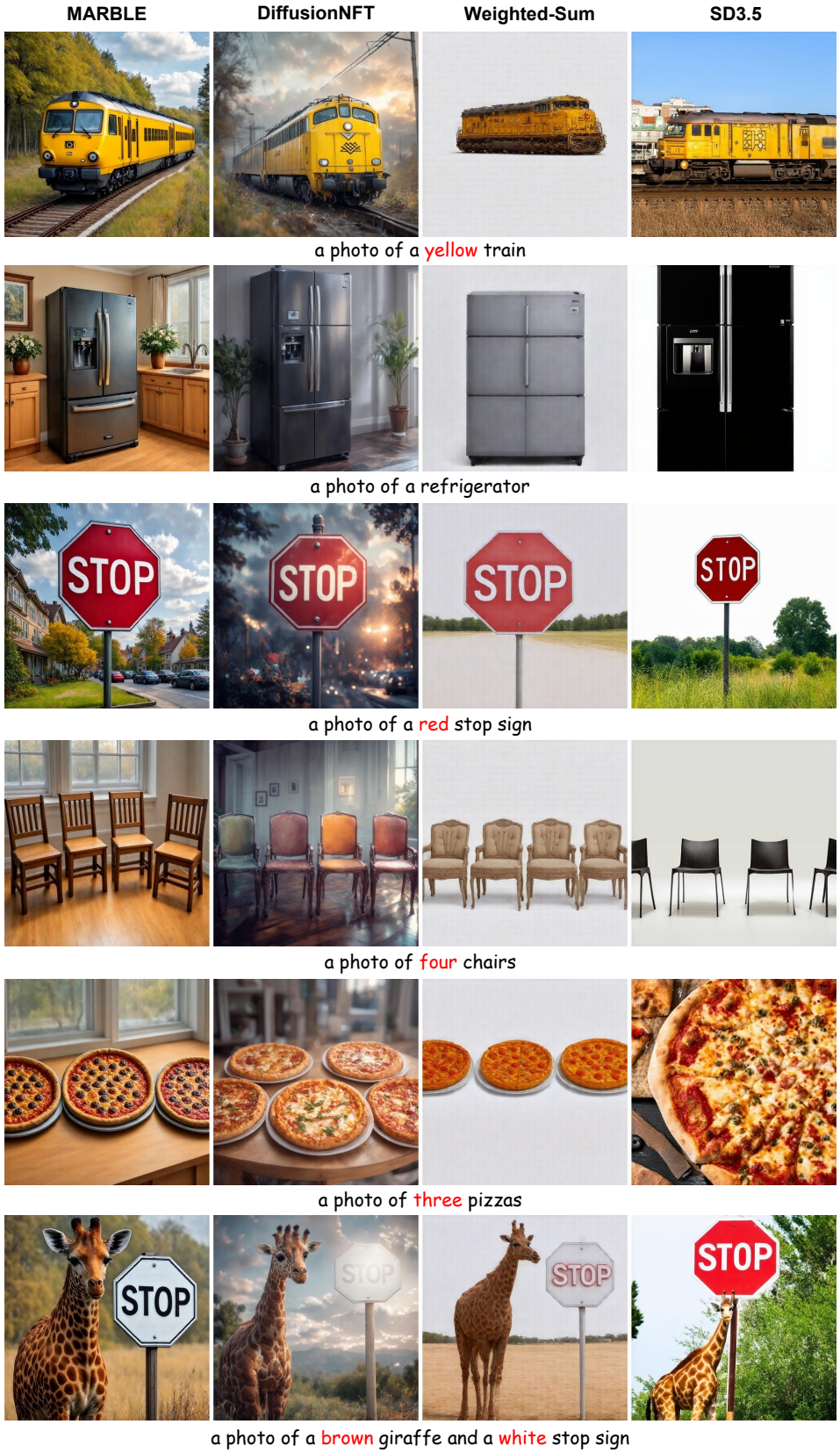

Figure S13: Additional qualitative comparisons.

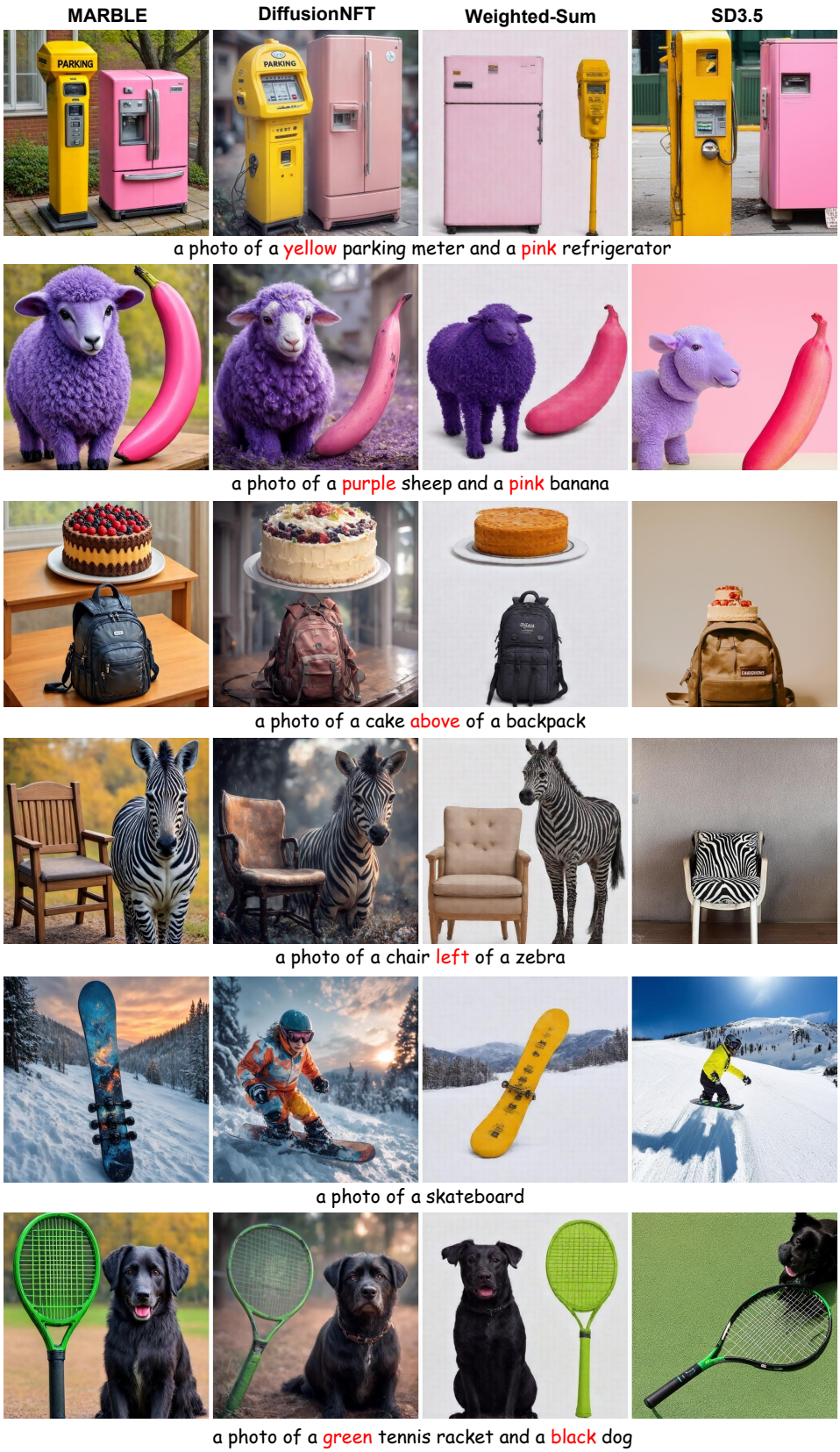

Figure S14: Additional qualitative comparisons.



Figure S15: Additional qualitative comparisons.

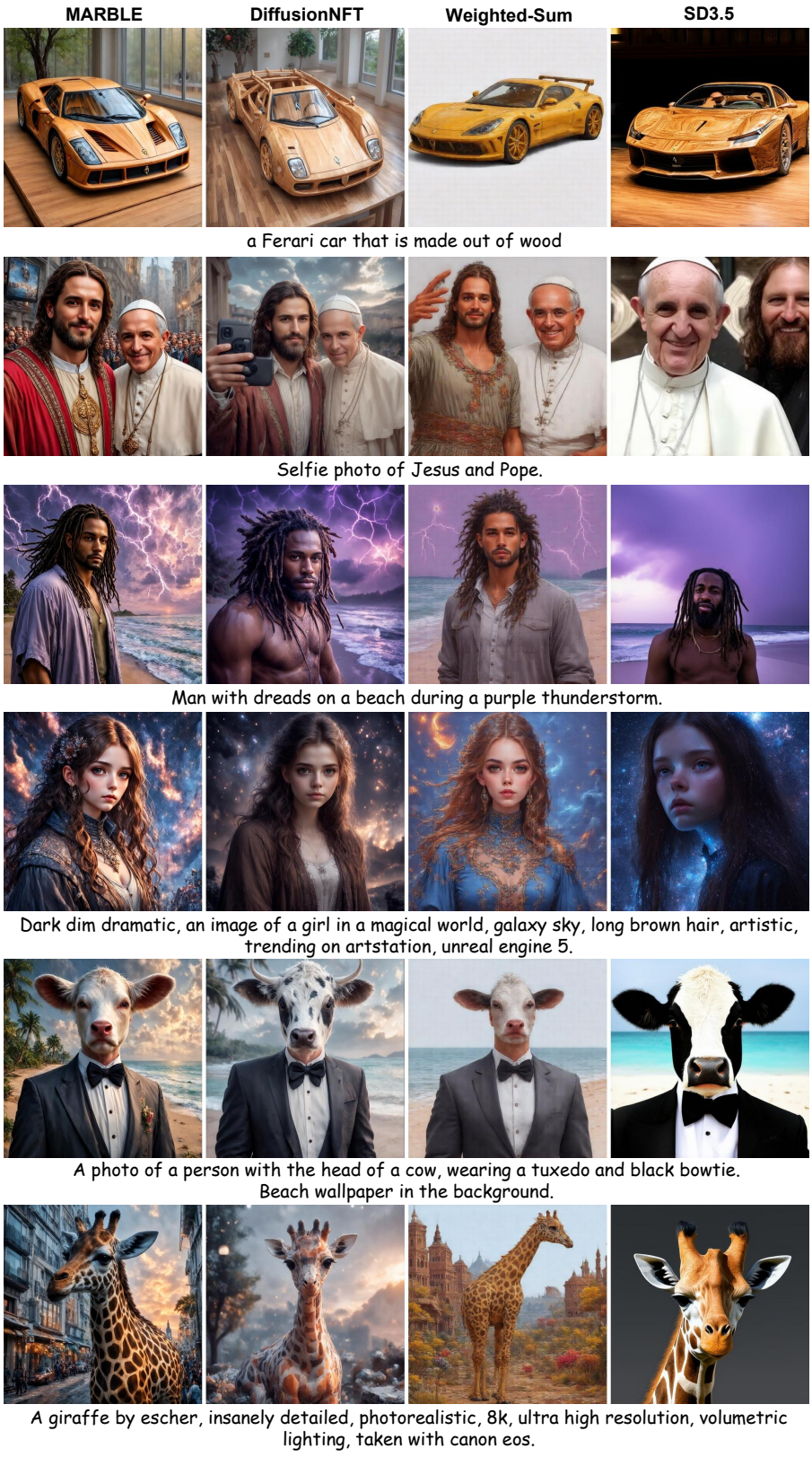

Figure S16: Additional qualitative comparisons.