Supplementary Materials

MapReduce LoRA: Advancing the Pareto Front in Multi-Preference Optimization for Generative Models

A. MapReduce LoRA on Language Tasks **2**

B. Pseudocode **3**
B.1. Train: MapReduce LoRA . . . 3
B.2. Train: Reward-aware Token Embedding . . . 3
B.3. Inference with MapReduce LoRA and RaTE . . . 4

C. Implementation Details **4**
C.1. Training Configuration . . . 4
C.2. Base Models and Reward Models . . . 5

D. More Quantitative Results **5**
D.1. Full Results of Different Merging Ratios (Corresponding to Fig. 1 in the main paper) . . . 5
D.2. Pareto Front Comparison of MORL vs. MapReduce LoRA . . . 6
D.3. Scalability of Reward Numbers . . . 8

E. More Qualitative Results **8**
E.1. Text-to-Image qualitative results . . . 8
E.2. Text-to-Video qualitative results . . . 14

F. Limitations and Future Works **17**

A. MapReduce LoRA on Language Tasks

Beyond Text-to-Image and Text-to-Video, we evaluate MapReduce LoRA on language tasks to demonstrate cross-modal generality. Following the Bone Soup [16] setup, we consider two tasks, each with two rewards: (i) Reddit Summary (Faithful, Preference1) and (ii) Helpful Assistant (Helpful, Harmless). We first train the Llama2-7B [15] backbone via supervised fine-tuning (SFT) using the same datasets and hyperparameters as Bone Soup [16], then train LoRA experts with PPO [13] under the same settings of datasets and training hyperparameters. In Figs. 1 and 2, we compare MapReduce LoRA against Llama2 (base) [15], Llama2 after SFT, Rewarded Soup [12], and Bone Soup [16]. MapReduce LoRA achieves state-of-the-art performance on both tasks, underscoring its cross-modal generality.

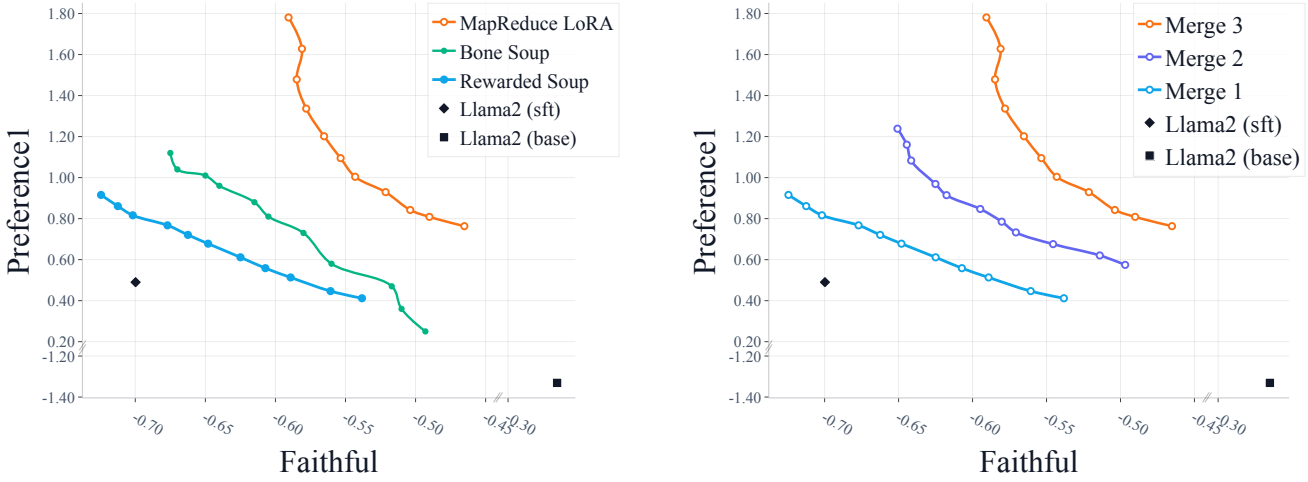


Figure 1. **Language Task: Reddit Summary.** Left: MapReduce LoRA outperforms Rewarded Soup [12] and Bone Soup [16] (reproduced from Fig. 5(c) in [16]) on both rewards. Right: MapReduce LoRA improves progressively across merge iterations.

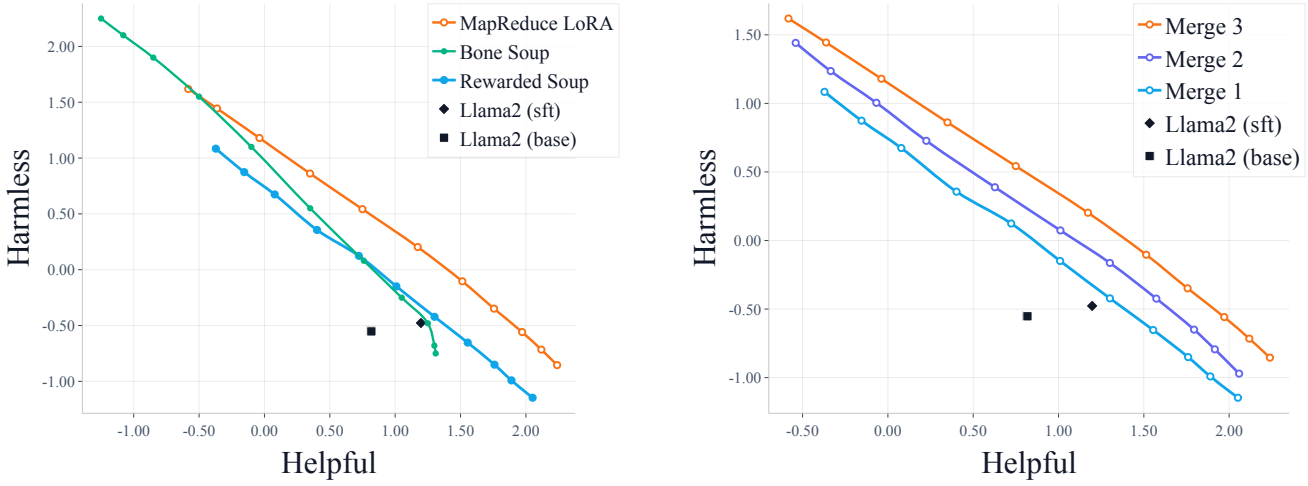


Figure 2. **Language Task: Helful Assistant.** Left: MapReduce LoRA outperforms Rewarded Soup [12] and Bone Soup [16] (reproduced from Fig. 5(a) in [16]) on both rewards. Right: MapReduce LoRA improves progressively across merging iterations.

B. Pseudocode

B.1. Train: MapReduce LoRA

Algorithm 1 MapReduce LoRA: Multi-Preference Training

Require: Base model M with parameters $\theta^{(0)}$; Reward models $\{R_i\}_{i=1}^{n}$; LoRA configuration (target layers, rank r, scale α); Number of iterations K; GRPO steps T_{GRPO}; Merge weights $\{\mu_i\}_{i=1}^{n}$ (default $\mu_i = 1/n$, s.t. $\mu_i \geq 0$ and $\sum_{i=1}^{n} \mu_i = 1$); GRPO hyperparameters: Group size G, Clip ϵ, KL weight β

Ensure: Multi-preference aligned model $M^{(K)}$

Initialize reference policy $\pi_{\text{ref}} \leftarrow M(\theta^{(0)})$
for $k = 0$ to $K - 1$ **do**
  for all $i \in \{1, \ldots, n\}$ **do** ▷ Map phase: train per-reward LoRA experts in parallel
    Initialize LoRA adapter $\phi_i^{(k)}$
    Freeze base weights $\theta^{(k)}$; only $\phi_i^{(k)}$ is trainable
    for $t = 1$ to T_{GRPO} **do**
      $\phi_i^{(k)} \leftarrow \text{GRPO}\big(M, \theta^{(k)}, \phi_i^{(k)}, R_i, \pi_{\text{ref}}, G, \epsilon, \beta\big)$
    end for
  end for
  $\bar{\phi}^{(k)} \leftarrow \sum_{i=1}^{n} \mu_i \, \phi_i^{(k)}$ ▷ Reduce phase: average experts
  $\theta^{(k+1)} \leftarrow \text{MergeLoRAIntoBase}\big(\theta^{(k)}, \bar{\phi}^{(k)}\big)$ ▷ update base model
  $M \leftarrow M(\theta^{(k+1)})$
  Reset all adapters $\phi_i^{(k)}$ to zero
  Set $\pi_{\text{ref}} \leftarrow M(\theta^{(k+1)})$
end for
return $M^{(K)}$

B.2. Train: Reward-aware Token Embedding

Algorithm 2 Reward-aware Token Embedding (RaTE) Training

Require: Frozen base model M with parameters θ; per-reward LoRA expert ϕ_i (teacher) for reward R_i; training prompts $\mathcal{D}$; special token index $RaTE_i$ for reward i; number of RaTE steps T_{RaTE}; Flow Matching noise schedule $\{\sigma_t\}$

Ensure: Trained token embedding θ_i^{token} for reward i

Attach LoRA expert ϕ_i to M to form teacher model M_{teacher}
Freeze all Transformer weights and all token embeddings except θ_i^{token}
for $t = 1$ **to** T_{RaTE} **do**
  Sample a minibatch of prompts $\{p_b\}_{b=1}^{B}$ from $\mathcal{D}$
  for each prompt p in $\{p_b\}$ **do**
    // Teacher: obtain preference-specific latent
    Sample initial noise $\epsilon_0 \sim \mathcal{N}(0, I)$
    Generate teacher latent $z_{0,i}^{\text{teacher}} \leftarrow \textsc{Sample}(M_{\text{teacher}}, p, \epsilon_0)$
    // Flow Matching distillation setup
    Sample t via a flow-matching schedule (e.g., logit-normal over timesteps); sample $\epsilon \sim \mathcal{N}(0, I)$
    Compute σ_t from the scheduler
    $z_t \leftarrow (1 - \sigma_t)\, z_{0,i}^{\text{teacher}} + \sigma_t \, \epsilon$
    Target velocity: $v_{\text{target}} \leftarrow \epsilon - z_{0,i}^{\text{teacher}}$
    // Student: base model + special token
    Construct prompt: $p' = \text{Concat}(p, <RaTE_i>)$
    Student velocity prediction: $v_{\text{pred}} \leftarrow M(z_t, t, c(p', \theta_i^{\text{token}}))$
    Loss: $\mathcal{L}_{\text{RaTE}} \leftarrow \big\| v_{\text{pred}} - v_{\text{target}} \big\|_2^2$
    Backpropagate and update only θ_i^{token}: $\theta_i^{\text{token}} \leftarrow \theta_i^{\text{token}} - \eta_{\text{RaTE}} \, \nabla_{\theta_i^{\text{token}}} \mathcal{L}_{\text{RaTE}}$
  end for
end for
return θ_i^{token}

B.3. Inference with MapReduce LoRA and RaTE

Algorithm 3 Inference with MapReduce LoRA and RaTE

Require: Final merged model $M^{(K)}$ with parameters $\theta^{(K)}$; reward-aware tokens $\{<RaTE_i>\}$ and embeddings $\{\theta_i^{\text{token}}\}$; user prompt p; user-specified preference set $\mathcal{S} \subseteq \{1, \ldots, n\}$; sampling hyperparameters (number of steps, seeds, guidance scale, etc.)

Ensure: Generated image or video sample x

Build preference-aware prompt:
  $p^* \leftarrow p$
for each $i \in \mathcal{S}$ in user-defined order **do**
  Append control token: $p^* \leftarrow \text{Concat}(p^*, <RaTE_i>)$
end for
Sample initial noise (image or video latent) $\epsilon_0 \sim \mathcal{N}(0, I)$
Run the sampler with $M^{(K)}$ conditioned on p^*:
  $x \leftarrow \textsc{Sample}(M^{(K)}, p^*, \epsilon_0)$
return x

C. Implementation Details

C.1. Training Configuration

Reward-aware Token Embedding. For each reward-aware token embedding, it contains 3 embeddings corresponding to 3 text encoders within SD 3.5 M [1]. The embedding is trained on the GenEval, PickScore, and OCR datasets, individually. The teacher model is trained with LoRA via GRPO [14] with 4,200 steps on GenEval [3], 4,500 steps on PickScore [5] and 1,600 steps on OCR [2]. The training batchsize for GenEval is 512, PickScore is 768 and OCR is 768. We use the AdamW optimizer [11] with learning rate 3e-4, β_1 = 0.9, β_2 = 0.999, weight decay = 1e-4 and no warmup schedule; no KL penalty is applied.

MapReduce LoRA. We adopt uniform averaging for all the default experiments if not specified.

- **Text-to-Image**: We train all models using AdamW optimizer [11] with learning rate 3e-4, β_1 = 0.9, β_2 = 0.999, weight decay = 1e-4, and no warmup schedule. All experiments use 32 GPUs with global batch size 576.
  - **SD 3.5 M [1]**: We follow the training configuration in Flow-GRPO [9], where only the KL ratio β is different. β_{GenEval} = 0.04, $\beta_{\text{PickScore}}$ = 0.01, and β_{OCR} = 0.04. The sampling timestep T is 10 and the evaluation timestep T = 40. The GRPO group size G is 24, the resolution is 512, and the LoRA settings are α = 64 and r = 32. LoRA is applied to all attention layers (`attn.add_q_proj`, `attn.add_k_proj`, `attn.add_v_proj`, `attn.to_add_out`, `attn.to_q`, `attn.to_k`, `attn.to_v`, `attn.to_out.0`). The per-GPU batch sizes are 6, 9, and 9 with gradient accumulation steps of 3, 2, and 2 for GenEval, PickScore, and OCR, respectively. We train GenEval, PickScore and OCR for {4,100, 1,200, 1,000, 1,100}, {4,500, 1,400, 1,000, 1,000}, and {1,600, 1,300, 900, 1,100} steps, respectively; the bracketed numbers denote merge iterations 0-3.
  - **FLUX.1-dev [7]**: The only task-dependent change is the KL ratio β (β_{GenEval}=0.04, $\beta_{\text{PickScore}}$=0, β_{OCR}=0.04). The sampling timestep T is 6 and the evaluation timestep T = 28. The GRPO group size G is 24, the resolution is 512, and the LoRA settings are α = 128 and r = 64. LoRA is applied to attention layers and feed-forward network layers (`attn.to_q`, `attn.to_k`, `attn.to_v`, `attn.to_out.0`, `attn.add_q_proj`, `attn.add_k_proj`, `attn.add_v_proj`, `attn.to_add_out`, `ff.net.0.proj`, `ff.net.2`, `ff_context.net.0.proj`, `ff_context.net.2`). The per-GPU batch size is 3 with gradient accumulation steps of 6 for all three tasks. We train GenEval, PickScore and OCR for {2,700, 1,900, 1,800, 1,600}, {1,250, 600, 1,550, 2,200}, and {1,250, 650, 850, 1,100} steps, respectively; the bracketed numbers denote merge iterations 0-3.
- **Text-to-Video**: We train the HunyuanVideo [6] with the same configuration as the DanceGRPO [17] but apply LoRA instead of full finetuning. The LoRA settings are α = 64 and r = 32. LoRA is applied to attention layers "`attn.to_q`, `attn.to_k`, `attn.to_v`" in single_transformer_blocks (40 blocks), "`attn.to_q`, `attn.to_k`, `attn.to_v`, `attn.to_out.0`, `attn.to_add_out`, `attn.add_q_proj`, `attn.add_k_proj`, `attn.add_v_proj`" in transformer_blocks (20 blocks). We train VQ and MQ for {200, 140, 100}, {195, 145, 115} steps, respectively; the bracketed numbers denote merge iterations 0-2.

Multi-objective Reinforcement Learning (MORL). The hyperparameters for MORL are identical to the MapReduce LoRA mentioned above. During each epoch, batches are sampled from GenEval [3], PickScore [5], and OCR [2] datasets according to specified ratios (default 1:1:1), where each individual batch contains samples from a single source. During evaluation, each data source is independently evaluated using its corresponding reward model to provide per-task performance metrics. The training steps are comparable to the overall training steps of MapReduce LoRA for fairness. There are two variations of MORL in our experiments, and the key difference lies in the data mixture and reward mixture strategy:

- **Data mixture (MORL-D)**: Each sample is only evaluated by its corresponding reward model based on its source dataset.
- **Data mixture and reward mixture (MORL-DR)**: Samples from GenEval [3] and OCR [2] are scored by both their task-specific reward and PickScore [5]; we take the average as the final reward (e.g., for GenEval, $(r_{\text{geneval}} + r_{\text{pickscore}})/2$). In contrast, samples from the PickScore dataset are evaluated only with PickScore. This choice reflects dataset constraints: GenEval requires a structured prompt format, whereas OCR requires prompts that contain text for a valid evaluation. PickScore serves as a shared aesthetic quality anchor across tasks. Overall, this protocol better aligns with the multi-objective setting by applying multiple rewards in a dataset-agnostic manner (where applicable), rather than restricting evaluation to dataset-specific rewards.

Compute Resources. All the experiments are conducted on 32 NVIDIA A100 GPUs (80GB).

C.2. Base Models and Reward Models

The following table presents the base model and reward models along with their corresponding links.

Models	Links
SD 3.5 M [1]	https://huggingface.co/stabilityai/stable-diffusion-3.5-medium
FLUX.1-dev [7]	https://huggingface.co/black-forest-labs/FLUX.1-dev
HunyuanVideo [6]	https://huggingface.co/hunyuanvideo-community/HunyuanVideo

Reward Models	Links
GenEval [3]	https://github.com/djghosh13/geneval
PickScore [5]	https://huggingface.co/yuvalkirstain/PickScore_v1
OCR [2]	https://github.com/PaddlePaddle/PaddleOCR
VQAScore [8]	https://github.com/linzhiqiu/t2v_metrics
MPS [18]	https://github.com/Kwai-Kolors/MPS
VILA [4]	https://github.com/google-research/google-research/tree/master/vila
VQ and MQ [10]	https://github.com/KlingTeam/VideoAlign/tree/main

D. More Quantitative Results

D.1. Full Results of Different Merging Ratios (Corresponding to Fig. 1 in the main paper)

Pareto fronts denote the set of non-dominated solutions (parameter settings for which no objective can be improved without degrading another). In Fig. 3, we include dominated (non-Pareto-optimal) points to present the full result set, together with the scores reported in Tables 1 and 2. Dominated points can arise for four reasons:

- **Estimation noise and stochasticity**: Each reward is estimated from finite samples and noisy reward evaluators. For GenEval, PickScore and OCR, we evaluate 553, 2,048 and 1,018 cases, respectively.
- **Non-convex objective landscapes**: The optimization surfaces can be highly non-convex. The merged model checkpoints may lie in different basins or apexes for different reward functions.
- **Partial or misaligned objectives**: When the two rewards are not orthogonal (*i.e.*, they share underlying structure), optimizing one can initially improve the other; once the shared structure is exhausted, trade-offs dominate again, yielding dominated points.
- **Training unsaturation**: In Fig. 3 (g) and (h), the edge points are mixed together on the top-left corner because iteration 1 for OCR uses much fewer training steps (650) than iteration 0 (1,250); additional training may further improve performance.

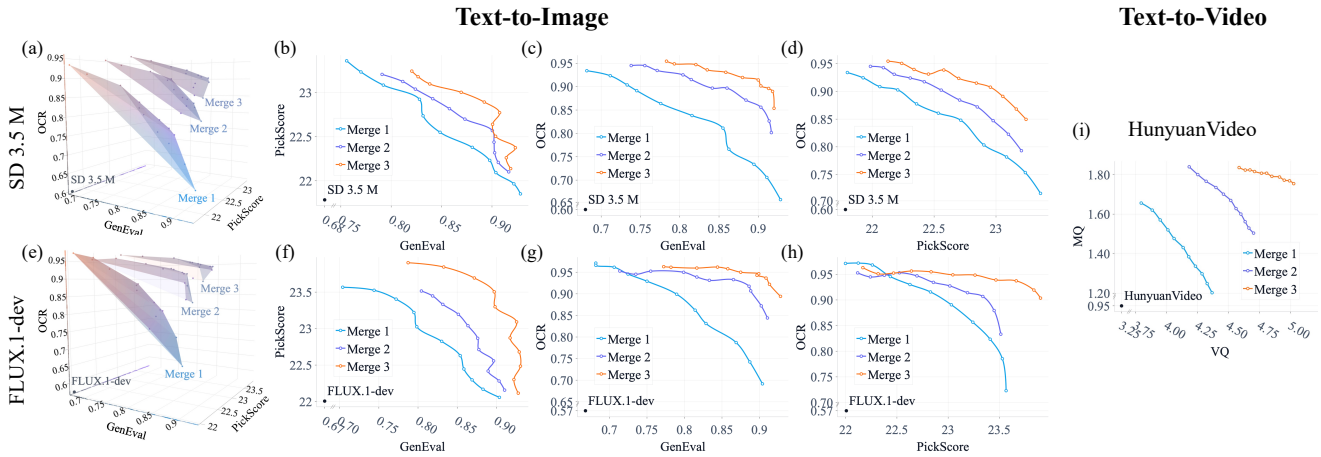


Figure 3. **MapReduce LoRA advances the Pareto fronts on Text-to-Image and Text-to-Video tasks.** We include the non-Pareto sets in this figure for sharing the full results.

Evaluation details:

- **3D Pareto fronts**: For each merge, we evaluate thirteen model weight configurations with different coefficient ratios: three single-reward configurations (the other two coefficients set to zero) and ten mixed configurations. The specific {GenEval: PickScore: OCR} ratios are $\{1:0:0\}$, $\{0:1:0\}$, $\{0:0:1\}$, $\{0.8:0.1:0.1\}$, $\{0.1:0.8:0.1\}$, $\{0.1:0.1:0.8\}$, $\{0.6:0.2:0.2\}$, $\{0.2:0.6:0.2\}$, $\{0.2:0.2:0.6\}$, $\{0.4:0.4:0.2\}$, $\{0.4:0.2:0.4\}$, $\{0.2:0.4:0.4\}$, and $\{0.\overline{3}:0.\overline{3}:0.\overline{3}\}$. Each plotted point is obtained by evaluating one merged model on all three metrics.
- **2D Pareto fronts**: For each merge, we evaluate eleven model weight configurations with different ratios: two single-reward configurations (the other coefficient set to zero) and nine mixed configurations. The specific ratios are $\{1:0\}$, $\{0:1\}$, $\{0.1:0.9\}$, $\{0.2:0.8\}$, $\{0.3:0.7\}$, $\{0.4:0.6\}$, $\{0.5:0.5\}$, $\{0.6:0.4\}$, $\{0.7:0.3\}$, $\{0.8:0.2\}$, and $\{0.9:0.1\}$. Each point is obtained by evaluating one merged model on the two metrics.

Table 1. 3D merging results on Text-to-Image tasks: SD 3.5 M [1] and FLUX.1-dev [7].

Group	Merging Ratios			SD 3.5 M			FLUX.1-dev		
	GenEval	PickScore	OCR	GenEval	PickScore	OCR	GenEval	PickScore	OCR
base model	0	0	0	0.68	21.784	0.601	0.67	22.006	0.573
Merge 1	1	0	0	0.93	21.852	0.658	0.90	22.057	0.692
	0	1	0	0.76	23.359	0.716	0.70	23.566	0.723
	0	0	1	0.68	21.800	0.934	0.68	21.997	0.971
	0.8	0.1	0.1	0.90	21.993	0.704	0.88	22.175	0.756
	0.1	0.8	0.1	0.77	23.090	0.746	0.75	23.408	0.794
	0.1	0.1	0.8	0.70	21.948	0.911	0.71	22.136	0.959
	0.6	0.2	0.2	0.86	22.106	0.743	0.83	22.299	0.802
	0.2	0.6	0.2	0.78	22.731	0.781	0.76	23.046	0.827
	0.2	0.2	0.6	0.76	22.087	0.882	0.73	22.288	0.928
	0.4	0.4	0.2	0.81	22.382	0.758	0.81	22.633	0.813
	0.4	0.2	0.4	0.80	22.098	0.826	0.79	22.294	0.883
	0.2	0.4	0.4	0.77	22.374	0.836	0.76	22.636	0.886
	$0.\overline{3}$	$0.\overline{3}$	$0.\overline{3}$	0.79	22.276	0.805	0.77	22.503	0.868
Merge 2	1	0	0	0.92	22.101	0.802	0.91	22.156	0.844
	0	1	0	0.79	23.205	0.793	0.80	23.515	0.833
	0	0	1	0.74	21.986	0.945	0.71	22.116	0.953
	0.8	0.1	0.1	0.90	22.208	0.831	0.89	22.303	0.883
	0.1	0.8	0.1	0.82	23.004	0.820	0.81	23.337	0.888
	0.1	0.1	0.8	0.76	22.109	0.933	0.75	22.249	0.956
	0.6	0.2	0.2	0.89	22.316	0.839	0.88	22.437	0.913
	0.2	0.6	0.2	0.84	22.778	0.843	0.83	23.037	0.916
	0.2	0.2	0.6	0.81	22.232	0.917	0.80	22.405	0.941
	0.4	0.4	0.2	0.85	22.532	0.850	0.85	22.718	0.918
	0.4	0.2	0.4	0.84	22.290	0.895	0.85	22.429	0.927
	0.2	0.4	0.4	0.83	22.486	0.899	0.81	22.709	0.929
	$0.\overline{3}$	$0.\overline{3}$	$0.\overline{3}$	0.84	22.436	0.871	0.84	22.619	0.926
Merge 3	1	0	0	0.92	22.138	0.854	0.93	22.115	0.894
	0	1	0	0.82	23.243	0.850	0.79	23.902	0.904
	0	0	1	0.78	22.128	0.954	0.77	22.167	0.963
	0.8	0.1	0.1	0.92	22.281	0.890	0.91	22.324	0.928
	0.1	0.8	0.1	0.83	23.080	0.875	0.85	23.683	0.930
	0.1	0.1	0.8	0.81	22.248	0.944	0.80	22.331	0.960
	0.6	0.2	0.2	0.91	22.408	0.900	0.90	22.508	0.936
	0.2	0.6	0.2	0.85	22.869	0.887	0.87	23.285	0.938
	0.2	0.2	0.6	0.84	22.369	0.929	0.85	22.496	0.955
	0.4	0.4	0.2	0.88	22.637	0.904	0.90	22.878	0.934
	0.4	0.2	0.4	0.87	22.393	0.920	0.88	22.515	0.943
	0.2	0.4	0.4	0.85	22.615	0.913	0.86	22.864	0.946
	$0.\overline{3}$	$0.\overline{3}$	$0.\overline{3}$	0.88	22.554	0.908	0.88	22.740	0.941

D.2. Pareto Front Comparison of MORL vs. MapReduce LoRA

To compare naive multi-objective RL with data mixing (MORL-D) against MapReduce LoRA, we train MORL-D under ten {GenEval:PickScore:OCR} sampling ratios—{1:1:1}, {1:2:3}, {1:3:2}, {2:1:3}, {2:3:1}, {3:1:2}, {3:2:1}, {1:1:4}, {1:4:1},

Table 2. 2D merging results on Text-to-Image tasks, SD 3.5 M [1] and FLUX.1-dev [7], and Text-to-Video task, HunyuanVideo [6].

Group	Merging Ratios		SD 3.5 M						FLUX.1-dev						HunyuanVideo	
	reward A	reward B	^{A}GenEval	^{B}PickScore	^{A}GenEval	^{B}OCR	^{A}PickScore	^{B}OCR	^{A}GenEval	^{B}PickScore	^{A}GenEval	^{B}OCR	^{A}PickScore	^{B}OCR	^{A}VQ	^{B}MQ
base model	0	0	0.68	21.784	0.68	0.601	21.784	0.601	0.67	22.006	0.67	0.573	22.006	0.573	3.25	0.95
Merge 1	0	1	0.76	23.359	0.68	0.934	21.800	0.934	0.70	23.566	0.68	0.971	21.997	0.971	3.80	1.66
	0.1	0.9	0.77	23.233	0.71	0.923	21.926	0.925	0.74	23.525	0.68	0.966	22.122	0.972	3.89	1.62
	0.2	0.8	0.79	23.084	0.73	0.904	22.062	0.909	0.77	23.403	0.71	0.962	22.260	0.968	3.95	1.57
	0.3	0.7	0.83	22.923	0.75	0.891	22.207	0.903	0.79	23.220	0.72	0.946	22.436	0.946	4.01	1.52
	0.4	0.6	0.83	22.736	0.78	0.864	22.358	0.878	0.80	23.024	0.75	0.929	22.627	0.930	4.06	1.48
	0.5	0.5	0.85	22.549	0.82	0.838	22.534	0.861	0.83	22.825	0.79	0.899	22.828	0.916	4.13	1.43
	0.6	0.4	0.88	22.379	0.86	0.811	22.715	0.848	0.85	22.629	0.81	0.862	23.032	0.891	4.18	1.38
	0.7	0.3	0.90	22.232	0.86	0.766	22.902	0.803	0.86	22.446	0.83	0.831	23.242	0.857	4.23	1.34
	0.8	0.2	0.90	22.093	0.89	0.734	23.085	0.782	0.87	22.297	0.87	0.787	23.410	0.823	4.29	1.30
	0.9	0.1	0.92	21.966	0.91	0.706	23.237	0.754	0.88	22.169	0.89	0.742	23.530	0.785	4.33	1.25
	1	0	0.93	21.852	0.93	0.658	23.359	0.716	0.90	22.057	0.90	0.692	23.566	0.723	4.37	1.20
Merge 2	0	1	0.79	23.205	0.74	0.945	21.986	0.945	0.80	23.515	0.71	0.953	22.116	0.953	4.18	1.84
	0.1	0.9	0.81	23.125	0.76	0.945	22.090	0.943	0.82	23.447	0.74	0.945	22.241	0.945	4.25	1.80
	0.2	0.8	0.82	23.039	0.77	0.935	22.178	0.931	0.83	23.332	0.75	0.952	22.377	0.948	4.32	1.77
	0.3	0.7	0.84	22.930	0.80	0.925	22.307	0.924	0.85	23.195	0.80	0.950	22.524	0.953	4.40	1.73
	0.4	0.6	0.86	22.817	0.81	0.915	22.421	0.917	0.86	23.040	0.82	0.939	22.696	0.947	4.46	1.70
	0.5	0.5	0.87	22.699	0.84	0.897	22.557	0.902	0.88	22.867	0.83	0.931	22.856	0.934	4.52	1.67
	0.6	0.4	0.90	22.569	0.86	0.897	22.706	0.884	0.88	22.713	0.87	0.933	23.017	0.927	4.56	1.63
	0.7	0.3	0.90	22.438	0.88	0.871	22.849	0.872	0.90	22.557	0.89	0.918	23.189	0.914	4.60	1.60
	0.8	0.2	0.90	22.318	0.90	0.856	22.977	0.847	0.89	22.418	0.89	0.907	23.339	0.907	4.63	1.56
	0.9	0.1	0.91	22.208	0.91	0.826	23.100	0.823	0.90	22.282	0.90	0.872	23.450	0.879	4.66	1.53
	1	0	0.92	22.101	0.92	0.802	23.205	0.793	0.91	22.156	0.91	0.844	23.515	0.833	4.70	1.50
Merge 3	0	1	0.82	23.243	0.78	0.954	22.128	0.954	0.79	23.902	0.77	0.963	22.167	0.963	4.59	1.84
	0.1	0.9	0.83	23.179	0.79	0.949	22.238	0.951	0.83	23.844	0.78	0.962	22.316	0.951	4.64	1.82
	0.2	0.8	0.84	23.097	0.82	0.947	22.338	0.940	0.87	23.693	0.82	0.960	22.467	0.952	4.67	1.82
	0.3	0.7	0.87	23.003	0.83	0.936	22.454	0.931	0.90	23.504	0.84	0.962	22.642	0.956	4.71	1.82
	0.4	0.6	0.89	22.892	0.86	0.931	22.578	0.939	0.90	23.299	0.86	0.958	22.834	0.955	4.77	1.81
	0.5	0.5	0.91	22.772	0.88	0.920	22.699	0.924	0.93	23.091	0.87	0.950	23.046	0.949	4.81	1.81
	0.6	0.4	0.90	22.640	0.90	0.914	22.842	0.915	0.92	22.872	0.90	0.946	23.263	0.948	4.85	1.79
	0.7	0.3	0.90	22.504	0.90	0.902	22.955	0.908	0.93	22.680	0.90	0.943	23.467	0.939	4.91	1.79
	0.8	0.2	0.92	22.375	0.91	0.896	23.055	0.891	0.93	22.485	0.91	0.936	23.666	0.937	4.95	1.77
	0.9	0.1	0.91	22.261	0.92	0.890	23.164	0.868	0.92	22.301	0.91	0.922	23.838	0.919	4.99	1.77
	1	0	0.92	22.138	0.92	0.854	23.243	0.850	0.93	22.115	0.93	0.894	23.902	0.904	5.03	1.75

{4:1:1}—using the same number of training steps as MapReduce LoRA on SD 3.5 M [1]. Fig. 4 compares Pareto fronts for Rewarded Soup [12], MORL-D, and MapReduce LoRA (left: 3D; right: 2D projections). In 3D, MORL-D is confined to a small region across the three rewards. For readability, we also show 2D projections of the 3D front; because MORL-D is an a priori method, we cannot fix one reward to 0 as in Fig. 3, so the right panels are projections rather than true 2D Pareto fronts. Although MORL-D is competitive on GenEval and OCR, conflicting objectives reduce visual quality (PickScore; see Fig. 10), yielding only limited PickScore gains.

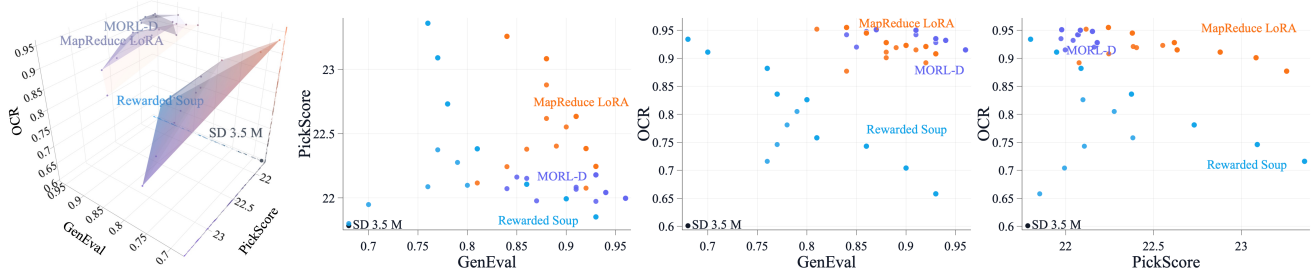


Figure 4. **Merging performance comparison across Rewarded Soup [12], MORL-D, and MapReduce LoRA.** Left: 3D Pareto-front comparison. Right: 2D projections of the 3D Pareto front (for readability), which are not 2D Pareto fronts. Unlike Fig. 3, these are projections rather than a 2D merge with the third reward fixed to 0. MORL-D performance is confined to a small region across the three rewards and yields only limited improvement on PickScore.

D.3. Scalability of Reward Numbers

We extend MapReduce LoRA to 5 rewards by adding VQAScore [8] and MPS [18] on SD 3.5 M [1]. Fig. 5 shows that MapReduce LoRA improves all 5 rewards (+22% GenEval, +3% PickScore, +44% OCR, +6% VQAScore, +7% MPS), consistently outperforming one-shot Rewarded Soup (the starting point of k = 1). Scaling to more rewards does not introduce diminishing returns: performance on overlapping rewards (GenEval, PickScore, OCR) remains improved compared to 3-reward training (Fig. 6, k=4). However, achieving similar levels requires 1.66× more steps, reflecting increased optimization complexity. Beyond the increased optimization complexity for existing rewards, adding rewards incurs a linear per-reward training cost; however, the parallel Map phase helps mitigate the resulting wall-clock growth given sufficient GPUs.

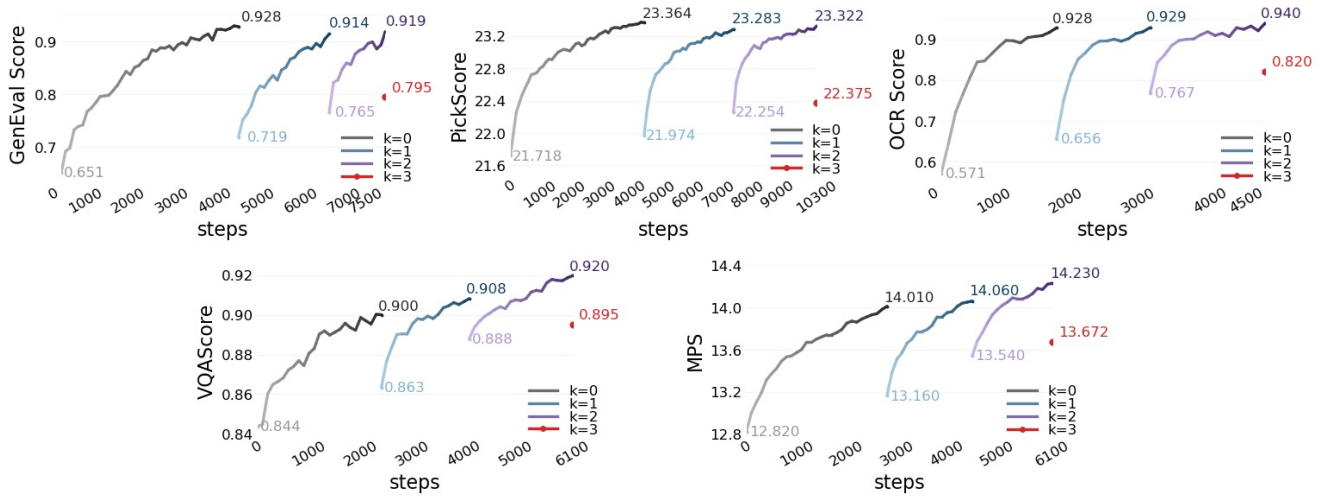


Figure 5. MapReduce LoRA across 5 rewards with 3 merges.

E. More Qualitative Results

E.1. Text-to-Image qualitative results

Figs 6, 7 and 8 demonstrate the visual comparison of two rewards with different merging ratios. Figs 9 and 10 demonstrate the visual comparison of all methods, including Flow-GRPO [9] tuned on GenEval [3], PickScore [5], OCR [2], Rewarded Soup [12], MORL-D, MORL-DR and MapReduce LoRA. PickScore and OCR guidance cause strong reward overfitting: the former drives the model toward a narrow, high-scoring aesthetic style, while the latter encourages increasingly large and prominent text regardless of overall visual harmony or contextual appropriateness. Our proposed MapReduce LoRA mitigates these issues by iteratively merging per-reward experts, enabling the model to discover a more balanced (and often near-optimal) preference that harmonizes aesthetic fidelity with text readability.

Figure 6. Visual comparison across different merging ratios between PickScore and GenEval.

A high-fashion runway with a sleek, modern backdrop displaying "Spring Collection 2024". Models walk confidently on the catwalk, showcasing vibrant, floral prints and pastel tones, under soft, ambient lighting that enhances the fresh, spring vibe.

A realistic photograph of a fast food drive-thru menu board at dusk, featuring a bold and colorful advertisement that reads "Try Our New Burger" with an appetizing image of the burger below, set against the backdrop of a busy suburban street.

A high-altitude mountain summit with a wooden signpost clearly marked "Elevation 8000 Feet", surrounded by rocky terrain and a backdrop of distant, snow-capped peaks under a clear blue sky.

A weathered pirate ship flag, tattered and fluttering in the sea breeze, is intricately embroidered with the bold words "Sea Wolf Crew" in deep navy thread, against a backdrop of stormy skies and turbulent waves.

An astronaut in space, the helmet visor reflecting a floating "Low Oxygen Warning" sign, surrounded by the vast, star-filled darkness, with Earth faintly visible in the distance.

A bustling deli with vintage decor, the counter lined with gleaming metal and glass. A retro ticket machine in the corner, lights blinking, prints out a paper slip that reads "Now Serving 42", amid the hum of conversation and the scent of fresh bread.

A UFO hovers in the night sky, its underside glowing with a soft, otherworldly light. Beneath it, a beam of light illuminates a peaceful landscape. The words "We Come in Peace" are clearly visible on its surface, radiating a sense of calm and hope.

A vibrant skatepark with dynamic graffiti tagging that reads "Skate Or Die Trying" across a worn, concrete wall, surrounded by skaters performing tricks and an urban backdrop.

A vintage suitcase adorned with a "World Traveler Adventures" sticker, placed on a rustic wooden table. Sunlight filters through a nearby window, casting warm, golden hues on the suitcase and highlighting the sticker's faded, nostalgic colors.

Figure 7. Visual comparison across different merging ratios between PickScore and OCR.

Figure 8. Visual comparison across different merging ratios between GenEval and OCR.

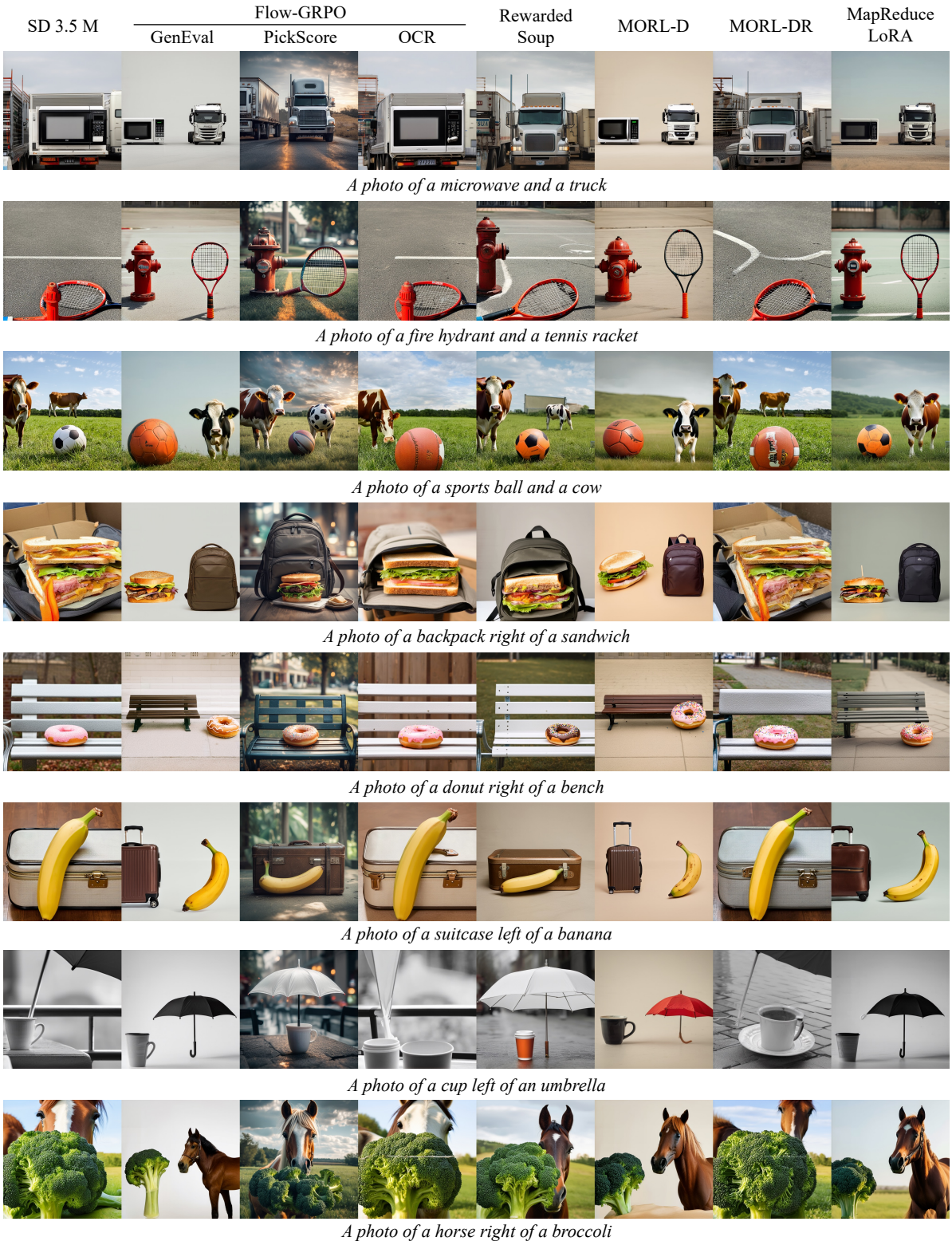
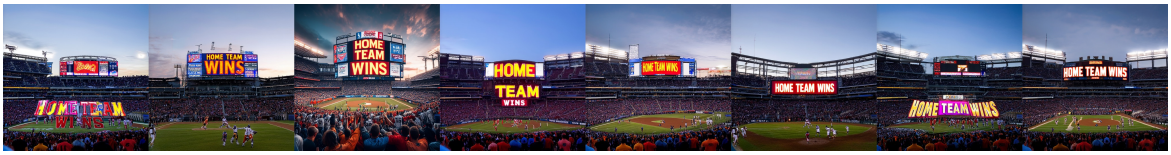


Figure 9. Visual comparison across all methods.

SD 3.5 M	Flow-GRPO			Rewarded Soup	MORL-D	MORL-DR	MapReduce LoRA
	GenEval	PickScore	OCR				

A realistic cockpit interior with a digital screen displaying "Prepare for Landing" in bold text, illuminated by soft blue backlighting, with the pilot's hand resting on the control yoke.

A bustling city street at night with a movie theater marquee prominently displaying "Now Showing Action 5". Crowds of people in casual attire are entering the theater, while neon lights and billboards light up the background, creating a vibrant urban atmosphere.



A bustling stadium at dusk, the scoreboard prominently displays "Home Team Wins" in vibrant, illuminated letters, surrounded by cheering fans in team colors, with the victorious team's players celebrating on the field.

A beekeeper stands beside a wooden hive box, carefully labeled "Buzz Central Handle With Care", surrounded by a bustling field of wildflowers, with bees fluttering around the hive and the keeper wearing traditional protective gear.

A realistic photograph of a gas station with a large, red price sign prominently displaying "Fuel Prices Rising" against a cloudy sky, cars filling up in the background.

A detailed close-up of a magic carpet with intricate embroidery, prominently featuring the text "No Shoes Fly Casual" woven in shimmering threads, set against a backdrop of a serene, cloudy sky.

A movie poster titled "Invasion of the Robots" featuring towering robots destroying cityscapes, with skyscrapers collapsing and fiery explosions lighting up the night sky, emphasizing the chaos and scale of the robotic invasion.

Figure 10. Visual comparison across all methods on text rendering quality.

E.2. Text-to-Video qualitative results

Fig. 11 demonstrates the visual comparison of HunyuanVideo [6], Rewarded Soup [12] and our proposed MapReduce LoRA. Also, Figs. 12 and 13 demonstrate the visual performance of merging progress that advances the visual quality, motion quality, and prompt alignment.

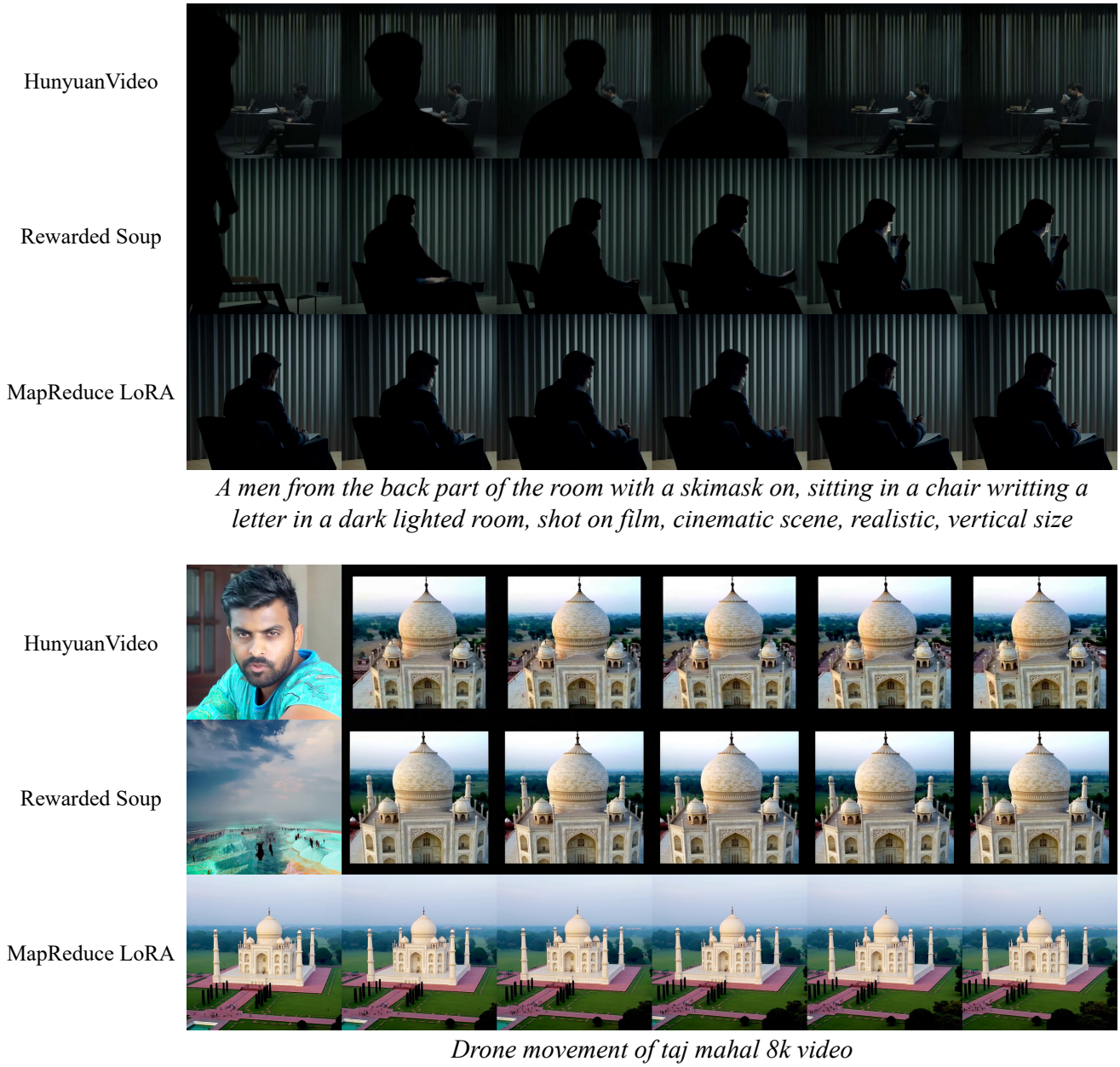


Figure 11. **Visual comparison of HunyuanVideo [6], Rewarded Soup [12], and MapReduce LoRA (Ours).** In the upper case, MapReduce LoRA more faithfully renders the intended motion and writing action described in the prompt. In the lower case, MapReduce LoRA better adheres to the specified drone-movement control, producing a trajectory that aligns with the prompt.

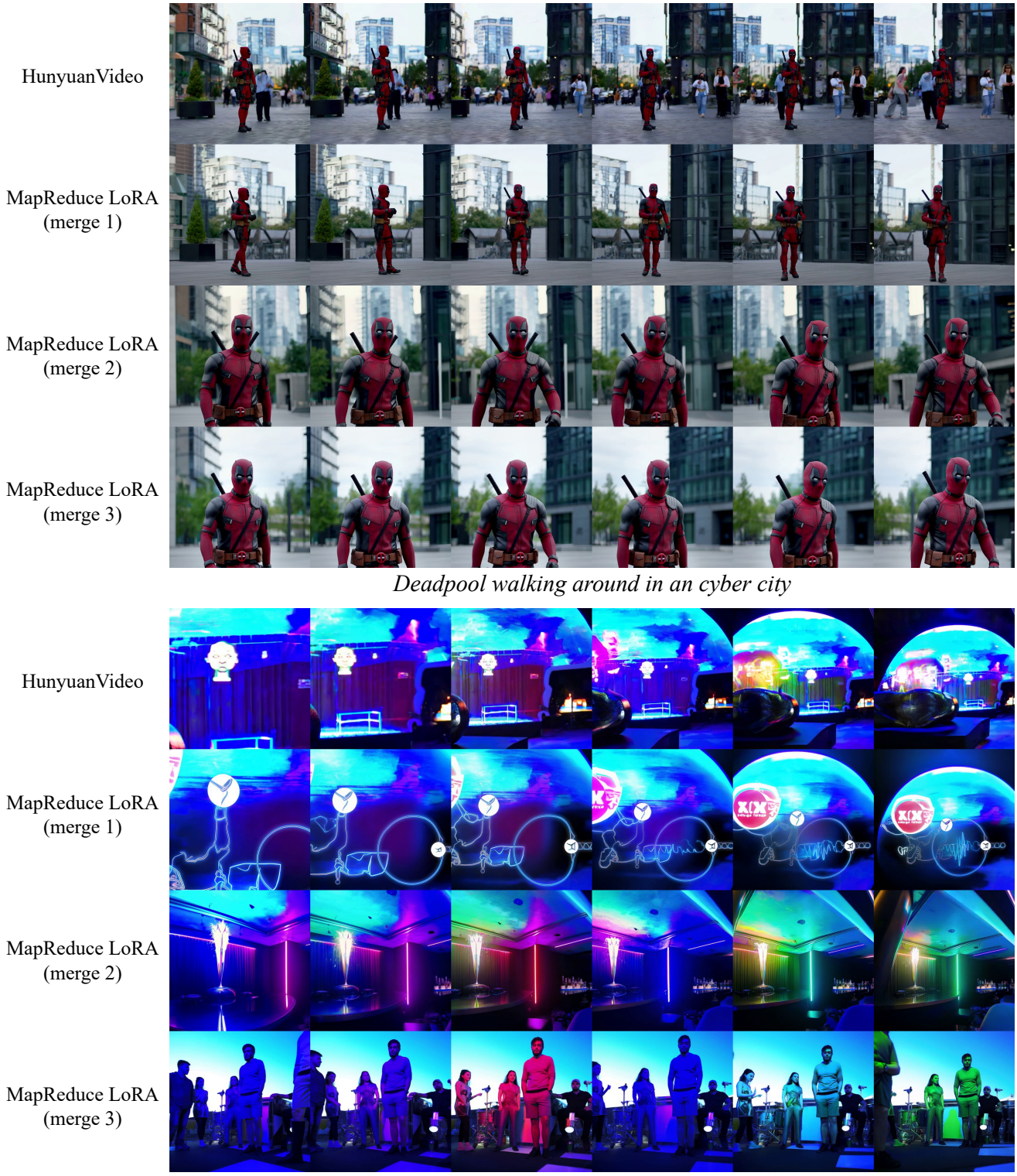


Figure 12. **Visual performance across two cases under different MapReduce LoRA merging iterations.** In the upper case, the HunyuanVideo result fails to depict the walking motion, while increasing the merging iterations progressively restores natural walking dynamics and improves the background building quality. In the lower case, the initial generation shows limited visual fidelity, whereas successive merging iterations consistently enhance the overall clarity and scene quality.

Figure 13. **Visual performance across two cases under different MapReduce LoRA merging iterations.** In the upper case, the HunyuanVideo result shows limited visual clarity—partly due to suboptimal motion quality—while increasing merging iterations progressively refine facial and background details, improve pose naturalness, and enhance overall realism. In the lower case, the initial generation offers limited scene clarity—the drinking containers are barely visible—whereas additional merging iterations clearly render these objects and deliver sharper, more coherent visuals.

F. Limitations and Future Works

This paper presents MapReduce LoRA, a simple, scalable recipe for systematically pushing the multi-preference Pareto front and enabling practical post-training customization. We note several scope considerations and opportunities for further study:

- **Scaling to more preferences:** In Text-to-Image, we validate on five targeted preferences; extending to larger numbers of preferences is a promising scaling direction.
- **Merging policies and schedules:** We default to uniform averaging and compare a few merge frequencies under fixed training steps; exploring adaptive/learned policies and schedules may yield further gains.
- ***Architecture-agnostic* Reward-aware Token Embedding (RaTE):** RaTE is lightweight and effective on Stable Diffusion series models; exploring model-agnostic designs is a practical direction.

We leave these promising directions for future work.

References

[1] Stability AI. Stable diffusion 3.5 medium. https://huggingface.co/stabilityai/stable-diffusion-3.5-medium, 2024. 4, 5, 6, 7, 8

[2] Cheng Cui et al. Paddleocr 3.0 technical report, 2025. 4, 5, 8

[3] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 4, 5, 8

[4] Junjie Ke, Keren Ye, Jiahui Yu, Yonghui Wu, Peyman Milanfar, and Feng Yang. VILA: learning image aesthetics from user comments with vision-language pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 5

[5] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 4, 5, 8

[6] Weijie Kong et al. HunyuanVideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024. 4, 5, 7, 14

[7] Black Forest Labs. FLUX. https://github.com/black-forest-labs/flux, 2024. 4, 5, 6, 7

[8] Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 5, 8

[9] 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. 4, 8

[10] Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Wenyu Qin, Menghan Xia, Xintao Wang, Xiaohong Liu, Fei Yang, Pengfei Wan, Di Zhang, Kun Gai, Yujiu Yang, and Wanli Ouyang. Improving video generation with human feedback. *arXiv preprint arXiv:2501.13918*, 2025. 5

[11] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 4

[12] Alexandre Ramé, 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 2, 7, 8, 14

[13] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 2

[14] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 4

[15] Hugo Touvron et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 2

[16] Guofu Xie, Xiao Zhang, Ting Yao, and Yunsheng Shi. Bone soups: A seek-and-soup model merging approach for controllable multi-objective generation. In *Proceedings of the Association for Computational Linguistics (ACL)*, 2025. 2

[17] Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, and Ping Luo. DanceGRPO: Unleashing GRPO on visual generation. *arXiv preprint arXiv:2505.07818*, 2025. 4

[18] Sixian Zhang, Bohan Wang, Junqiang Wu, Yan Li, Tingting Gao, Di Zhang, and Zhongyuan Wang. Learning multi-dimensional human preference for text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 5, 8