DuoLoRA : Cycle-consistent and Rank-disentangled Content-Style Personalization

Supplementary Material

In this supplementary material, we will provide the following details.

1. Training details.
2. Theoretical analysis.
3. Algorithm details.
4. Details of user study.
5. Additional results and ablations.

1. Training details.

We have provided the hyperparameters for each of the datasets, i.e., Dreambooth + SyleDrop, Subjectplop, Subjectplop + SyleDrop and Custom101 + SyleDrop in Table. 1, Table. 2, Table. 3 and Table. 4 respectively. For each of the dataset, the content and style images are provided in the supplementary as attachment.

For the joint training baseline, we are using Dreambooth [4] style joint training on SDXL with learning rate of 5.e-6. Training for 500 steps across all Unet parameters on a resolution of 768.

Table 1. Hyperparameters for Dreambooth + SyleDrop

Hyperparameter	Values
λ_{layer_prior}	0.1
λ_{cycle}	0.01
Base diffusion model	SDXL v1.0
LoRA rank	64
Learning rate of LoRA	$5e^{-5}$
Learning rate of mergers	0.001
Batch size	1
resolution	1024
$T_{content}$	0.1
T_{style}	0.0

Table 2. Hyperparameters for Subjectplop

Hyperparameter	Values
λ_{layer_prior}	0.01
λ_{cycle}	0.01
Base diffusion model	SDXL v1.0
LoRA rank	64
Learning rate of LoRA	$5e^{-5}$
Learning rate of mergers	0.001
Batch size	1
resolution	1024
$T_{content}$	0.75
T_{style}	0.5

Table 3. Hyperparameters for Subjectplop + SyleDrop

Hyperparameter	Values
λ_{layer_prior}	0.1
λ_{cycle}	0.01
Base diffusion model	SDXL v1.0
LoRA rank	64
Learning rate of LoRA	$5e^{-5}$
Learning rate of mergers	0.001
Batch size	1
resolution	1024
$T_{content}$	0.1
T_{style}	0.0

Table 4. Hyperparameters for Custom101 + SyleDrop

Hyperparameter	Values
λ_{layer_prior}	0.1
λ_{cycle}	0.01
Base diffusion model	SDXL v1.0
LoRA rank	64
Learning rate of LoRA	$5e^{-5}$
Learning rate of mergers	0.001
Batch size	1
resolution	1024
$T_{content}$	0.1
T_{style}	0.0

2. Theoretical Analysis

In this section, we provide the proof the theoretical results provided in the main paper.

Theorem 1. *In Low-Rank Adaptation (LoRA) merging, under the same parameter budget, the approximation error resulting from rank dimension masking is less than or equal*

to that from output dimension masking. Formally,

$$E_{rank} \leq E_{out},$$

where:

$$E_{rank} = \|X - \Delta W_{rank}\|_F$$

is the approximation error using rank dimension masking, and

$$E_{out} = \|X - \Delta W_{out}\|_F$$

is the approximation error using output dimension masking.

Proof. To compare the two masking strategies fairly, we ensure that both use the same number of parameters. Parameter Count for Rank Dimension Masking,

$$P_{\text{rank}} = s(d_{\text{out}} + d_{\text{in}}).$$

Parameter count for output dimension masking is,

$$P_{\text{out}} = d_s \times r + r \times d_{\text{in}} = r(d_s + d_{\text{in}}).$$

Now, Setting Equal Parameter Budgets, Set $P_{\text{rank}} = P_{\text{out}}$:

$$s(d_{\text{out}} + d_{\text{in}}) = r(d_s + d_{\text{in}}).$$

Assuming $d_{\text{out}} = d_{\text{in}} = d$ for simplicity:

$$s(2d) = r(d_s + d).$$

Solving for s:

$$s = \frac{r(d_s + d)}{2d}.$$

Next, we compare the Approximation Errors.

Since we retain the s largest singular values to minimize the error, the approximation error for Rank Dimension Masking is:

$$E_{\text{rank}} = \left(\sum_{i=s+1}^{p} \sigma_i^2 \right)^{1/2}.$$

The exact computation of Output Dimension Masking error (E_{out}) is complex due to the loss of orthogonality in U_r caused by masking. However, we can establish a lower bound.

The total energy (sum of squares) in U_r is:

$$\|U_r\|_F^2 = \text{trace}(U_r^\top U_r) = r.$$

where each row of U_r contributes equally on average to this total energy.

The fraction of rows masked out (i.e., energy removed by masking) is:

$$f = \frac{d - d_s}{d}.$$

Therefore, the approximate fraction of energy removed is f.

Next, we get the lower bound on Approximation Error, i.e., the loss in the approximation due to output dimension masking is at least:

$$E_{\text{out}}^2 \geq f \sum_{i=1}^{r} \sigma_i^2 + \sum_{i=r+1}^{p} \sigma_i^2.$$

The first term $f \sum_{i=1}^{r} \sigma_i^2$ represents the loss from masking out a fraction f of the energy from the top r singular values. The second term $\sum_{i=r+1}^{p} \sigma_i^2$ accounts for the singular values beyond rank r.

Now, Relating s and f, from the parameter equality:

$$s = \frac{r(d_s + d)}{2d} = \frac{r}{2}\left(1 + \frac{d_s}{d}\right).$$

Since $d_s = d(1 - f)$:

$$\begin{aligned} s &= \frac{r}{2}\left(1 + \frac{d_s}{d}\right) \\ &= \frac{r}{2}\left(1 + (1 - f)\right) \\ &= \frac{r}{2}(2 - f) \\ &= r\left(1 - \frac{f}{2}\right). \end{aligned}$$

Thus,

$$\frac{s}{r} = 1 - \frac{f}{2}.$$

We observe, in Rank Dimension Masking : The approximation error comes from the discarded smaller singular values (indices $i > s$). Since $s = r\left(1 - \frac{f}{2}\right)$, we discard the smallest $r - s = r\left(\frac{f}{2}\right)$ singular values among the top r.

$$E_{\text{rank}}^2 = \sum_{i=s+1}^{p} \sigma_i^2 = \sum_{i=r\left(1-\frac{f}{2}\right)+1}^{p} \sigma_i^2.$$

In Output Dimension Masking: The error includes a loss from the largest singular values, scaled by f, because masking affects all components equally.

$$E_{\text{out}}^2 \geq f \sum_{i=1}^{r} \sigma_i^2 + \sum_{i=r+1}^{p} \sigma_i^2.$$

Total Energy of Top r Singular Values

$$q = \sum_{i=1}^{r} \sigma_i^2.$$

Sum of Discarded Singular Values in Rank Masking:

$$q' = \sum_{i=s+1}^{r} \sigma_i^2 = q - \sum_{i=1}^{s} \sigma_i^2.$$

Relation between f and $\frac{s}{r}$:

$$f = 2\left(1 - \frac{s}{r}\right).$$

Now, Expressing E^2_{rank} and E^2_{out} in terms of q and q', we get the Rank Dimension Masking Error:

$$E^2_{\text{rank}} = q' + \sum_{i=r+1}^{p} \sigma_i^2.$$

Output Dimension Masking Error Lower Bound:

$$E^2_{\text{out}} \geq fq + \sum_{i=r+1}^{p} \sigma_i^2.$$

Since $q' = q - \sum_{i=1}^{s} \sigma_i^2$ and $s = r\left(1 - \frac{f}{2}\right)$, we have:

$$\begin{aligned} q' &= q - \sum_{i=1}^{s} \sigma_i^2 \\ &\leq q - s\left(\frac{\sigma_r^2}{r} \cdot r\right) \quad (\text{since } \sigma_i \geq \sigma_r) \\ &= q - s\sigma_r^2 \\ &= q - r\left(1 - \frac{f}{2}\right)\sigma_r^2. \end{aligned}$$

Therefore,

$$E^2_{\text{rank}} \leq q - r\left(1 - \frac{f}{2}\right)\sigma_r^2 + \sum_{i=r+1}^{p} \sigma_i^2.$$

Comparing with E^2_{out}, we get,

$$E^2_{\text{out}} \geq fq + \sum_{i=r+1}^{p} \sigma_i^2.$$

Since $f = 2\left(1 - \frac{s}{r}\right)$, we have:

$$E^2_{\text{out}} \geq 2\left(1 - \frac{s}{r}\right)q + \sum_{i=r+1}^{p} \sigma_i^2.$$

Therefore, the difference between E^2_{out} and E^2_{rank} is:

$$\begin{aligned} E^2_{\text{out}} - E^2_{\text{rank}} &\geq \left[2\left(1 - \frac{s}{r}\right)q - \left(q - r\left(1 - \frac{f}{2}\right)\sigma_r^2\right)\right] \\ &= \left(1 - \frac{s}{r}\right)\left(q + r\sigma_r^2\right). \end{aligned}$$

Since $q \geq r\sigma_r^2$, the difference is non-negative, implying:

$$E^2_{\text{rank}} \leq E^2_{\text{out}}.$$

□

Lemma 1. *Let $m_c \in \mathbb{R}^{m\times n}$ be a matrix representing the content merger and $m_s \in \mathbb{R}^{m\times n}$ be a matrix representing the style merger. The problem of minimizing the L_1-norm of m_c subject to a rank constraint on m_c can be written as:*

$$\min \|m_c\|_1 \quad \textit{subject to} \quad \textit{rank}(m_c) > \textit{rank}(m_s)$$

This problem is non-convex due to the rank constraint. A convex relaxation can be achieved by approximating the rank of a matrix using the nuclear norm $\|\cdot\|_*$, which is the sum of the singular values of the matrix. Thus, the original problem can be relaxed to:

$$\min \|m_c\|_1 \quad \textit{subject to} \quad \|m_c\|_* > \|m_s\|_*$$

where $\|m_c\|_*$ denotes the nuclear norm of m_c, and $\|m_s\|_*$ is the nuclear norm of m_s.

This relaxed problem can be approached via a Lagrangian penalty formulation:

$$\mathcal{L}(m_c, m_s, \lambda) = \|m_c\|_1 + \lambda \max(0, \|m_s\|_* - \|m_c\|_*)$$

for some penalty parameter $\lambda \geq 0$, which enforces the constraint $\|m_c\|_* > \|m_s\|_*$ in the limit as $\lambda \to \infty$.

Proof. The rank of a matrix m_c is a non-convex function, making it difficult to optimize directly. The nuclear norm $\|m_c\|_*$, defined as the sum of the singular values of m_c, provides a *convex envelope* of the rank function over the unit ball of matrices in the operator norm. Minimizing the nuclear norm encourages low-rank solutions because the nuclear norm penalizes the magnitude of singular values, making it an effective surrogate for the rank.

Thus, we replace the rank constraint $\text{rank}(m_c) > \text{rank}(m_s)$ with the nuclear norm constraint:

$$\|m_c\|_* > \|m_s\|_*$$

This converts the non-convex constraint into a convex inequality that we can handle more easily in optimization.

Since the difference $\|m_s\|_* - \|m_c\|_*$ is not convex, directly enforcing $\|m_c\|_* > \|m_s\|_*$ would introduce non-convexity back into the problem. Instead, we approach this with a *penalty function* that gradually enforces the constraint.

Define the Lagrangian-like penalty function:

$$\mathcal{L}(m_c, m_s, \lambda) = \|m_c\|_1 + \lambda \max(0, \|m_s\|_* - \|m_c\|_*)$$

where:

- $\lambda \geq 0$ controls the strength of the constraint enforcement.

- The penalty term $\max(0, \|m_s\|_* - \|m_c\|_*)$ becomes zero if $\|m_c\|_* \geq \|m_s\|_*$ and adds a positive penalty otherwise.

This penalty formulation turns the original constrained problem into an unconstrained optimization problem, where

the constraint $\|m_c\|_* > \|m_s\|_*$ is gradually enforced by increasing λ.

Convergence and Feasibility. As $\lambda \to \infty$, the penalty for violating $\|m_c\|_* > \|m_s\|_*$ becomes very large, making it infeasible for any solution to have $\|m_c\|_* \leq \|m_s\|_*$ in the limit. Therefore, the solution to the penalized Lagrangian problem approaches the solution to the original problem:

$$\min \|m_c\|_1 \quad \text{subject to} \quad \|m_c\|_* > \|m_s\|_*$$

In other words, by iteratively solving for m_c with larger values of λ, we approximate a solution that satisfies the nuclear norm constraint. This penalty approach provides a feasible, convex approximation for the non-convex problem, yielding a solution that respects the desired rank constraint indirectly.

The penalty formulation of the Lagrangian:

$$\mathcal{L}(m_c, m_s, \lambda) = \|m_c\|_1 + \lambda \max(0, \|m_s\|_* - \|m_c\|_*)$$

provides an effective convex relaxation for the non-convex constraint $\text{rank}(m_c) > \text{rank}(m_s)$. This approach ensures that the solution minimizes $\|m_c\|_1$ while approximately satisfying the rank constraint in a manner that is computationally feasible and does not require convexity of the difference $\|m_s\|_* - \|m_c\|_*$.

□

3. Algorithm details

In this section, we provide additional details for our approach DuoLoRA. The algorithm for cycle-consistent merging using constyle loss has been provided in Algorithm. 1. Also, the algorithm for content and style mergers initialization method is provided in Algorithm. 2.

4. Details of user study

Since the perceptual metrics are not always reliable, we conduct user study to verify the efficacy of our method. We provide 20 examples of content, style pairs and corresponding generated images using Naive merging, B-LoRA, ZipLoRA and DuoLoRA. Then, we asked the following question to amazon mechanical turks: "which of the generated images is of best visual quality considering factors that we preserve both the content and style?", and the options are “Naive merging”, “B-LoRA”, “ZipLoRA”, “DuoLoRA”, “None is satisfactory”. We evaluate this by 50 users, totalling 1000 questionnaires by Amazon Mechanical Turk (AMT) to get unbiased results. The aggregate responses in Table. 5 showed that DuoLoRA generated images significantly outperformed the baselines by a large margin (50%). This verify that DuoLoRA generated images retain both content and style.

Algorithm 1: Merging LoRA with Cycle-Consistency Loss

Input : Content images I_c, Style images I_s
Output : Merged LoRA L_m

Step 1: Train LoRA for content and style
- Learn content LoRA (L_c) using content images I_c and prompt p_c="a <V1> object in <S1> style"
- Learn style LoRA (L_s) using style images I_s and prompt p_s="a <V2> object in <S2> style"

Step 2: Apply cycle-consistency loss across style
// Generate variations of I_c
- $I_{cc} \leftarrow (D + L_c)(I_c,$ “<V1> object in <S1> style”)
// Add style
- $I_{cs} \leftarrow (D + L_s)(I_c,$ “<V1> object in <S2> style”)
// Remove style
- $I_{csc} \leftarrow (D + L_c)(I_{cs},$ “<V1> object in <S1> style”)
// Ensure cycle-consistency loss across style
- $\mathcal{L}_{\text{cycle\_sty}} \leftarrow \text{MSE}(I_{cc}, I_{csc})$

Step 3: Apply cycle-consistency loss across object
// Generate variations of I_s
- $I_{ss} \leftarrow (D + L_s)(I_s,$ “<V2> object in <S2> style”)
// Add object
- $I_{sc} \leftarrow (D + L_c)(I_s,$ “<V1> object in <S2> style”)
// Remove object
- $I_{scs} \leftarrow (D + L_s)(I_{sc},$ “<V2> object in <S2> style”)
// Ensure cycle-consistency loss across content
- $\mathcal{L}_{\text{cycle\_content}} \leftarrow \text{MSE}(I_{ss}, I_{scs})$

Step 4: Merging LoRAs with consistency loss
- Train merged LoRA L_m using L_c and L_s with the consistency loss:

$$\begin{aligned}\mathcal{L}_{constyle} = &||(D + L_m)(I_c, p_c) - (D + L_c)(I_c, p_c)|| \\ &+ ||(D + L_m)(I_s, p_s) - (D + L_s)(I_s, p_s)|| \\ &+ \lambda_{cycle} \cdot \mathcal{L}_{\text{cycle\_sty}} + \lambda_{cycle} \cdot \mathcal{L}_{\text{cycle\_content}}\end{aligned}$$

where D is the T2I diffusion model and λ_{cycle} is the scaling factor.

Step 5: Inference
- During inference, pass the combined trained tokens as prompt (e.g., "a <V1> object in <S2> style running") to the T2I diffusion model with merged LoRA L_m to generate variations corresponding to the text prompt.

Algorithm 2: Content and Style Merger Initialization Algorithm

Input: Content merger m_c, Style merger m_s, content threshold ($T_{content}$), style threshold (T_{style})
Output: Initialized content merger vector, Initialized style merger vector

$V \leftarrow \text{rand}(64, 1)$;
$V' \leftarrow \frac{V}{\|V\|}$; `// Normalize V`
if $Rank(m_c) > Rank(m_s)$ **then**
  content_merger_init $\leftarrow 1(V' > T_{style})$
  style_merger_init $\leftarrow 1(V' > T_{content})$
else if $Rank(m_c) < Rank(m_s)$ **then**
  content_merger_init $\leftarrow 1(V' > T_{content})$
  style_merger_init $\leftarrow 1(V' > T_{style})$
else
  content_merger_init $\leftarrow \mathbf{1}_{64\times1}$
  style_merger_init $\leftarrow \mathbf{1}_{64\times1}$
end

Table 5. User study

None	Naive merging	B-LoRA	ZipLoRA	DuoLoRA
0.0%	10%	18%	22%	50%

5. Additional results and ablations

5.1. Comparisons and Qualitative results

We provide additional results using joint training baselines for all the datasets in Table. 6. Qualitative results in Fig. 2, Fig. 3, Fig. 7, Fig. 4 and Fig. 5 also shows that DuoLoRA performs better than the baselines. In Fig. 1, we show ablations of how each components affect the merging. We also show results while using concepts from real-world concept-centric custom101 dataset [3] and styles from styledrop dataset. DuoLoRA outperforms baselines as shown in Fig. 8, Fig. 9, Fig. 9, Fig. 10 and Fig. 11. We also compare with Paircustomization [2] using their setup, since they require 1-shot concept-style pair for training. We use their dataset for fair comparison with 6 objects, 2 styles. DuoLoRA performs better than Paircustomization as shown in Fig. 17 and also in main paper. Moreover, training DuoLoRA is more easier and parameter efficient than Paircustomization, which requires sort of joint training framework.

5.2. Multi-concept stylization

Here we provide details of multi-concept stylization. We further extend our approach to handle multi-concept stylization. Given two concepts, C_1 and C_2, and a style S, our objective is to generate an image that contains both C_1 and C_2 in style S. To achieve this, we decompose the task into individual content-style merging processes, specifically C_1-S and C_2-S merging, using layer-prior-informed loss. We then perform an arithmetic merging of the outputs from C_1-S and C_2-S to create the final image. The steps are as follows:

- We begin by merging each concept C_1 and C_2 with the style S. First, we train LoRAs L_1, L_2, and L_s with identifiers $< v1 >$, $< v2 >$, and $< s >$, respectively.
- Next, we merge LoRAs L_1 and L_2 with L_s in the rank dimension, applying the layer-prior loss as previously described. That is, we define the merged LoRAs as $L_{1m} = \text{merge}(L_1, L_s)$ and $L_{2m} = \text{merge}(L_2, L_s)$.
- After generating the merged LoRAs, we perform an arithmetic merging to obtain the final merged LoRA: $L_{1,2,S} = \alpha_1 L_{1m} + \alpha_2 L_{2m}$.
- During inference, we use directional prompting with the merged LoRA $L_{1,2,S}$, using `p = "a` $< v1 >$ `object on the left and a` $< v2 >$ `object on the right in` $< S >$ `style"`. We find that directional prompting plays a crucial role in achieving high fidelity when generating multiple objects.

We extend this for 2, 3, 4 concepts from Dreambooth dataset and syles from StyleDrop dataset. The results are shown in Fig. 12 and Fig. 13. We also show ablation for directional prompting in Fig. 14, which remains important for multi-concept composition.

5.3. Recontextualization

We also evaluate the recontextualization ability of our method. We use text prompts `'riding a boat', 'sleeping', 'riding a bicycle', 'riding a car', 'wearing a hat'` to generate different variation of styled concepts as shown in Fig. 15 and Fig. 16. Our method can successfully recontextualize w.r.t the text prompts.

References

[1] Yarden Frenkel, Yael Vinker, Ariel Shamir, and Daniel Cohen-Or. Implicit style-content separation using b-lora. *arXiv preprint arXiv:2403.14572*, 2024.

[2] Maxwell Jones, Sheng-Yu Wang, Nupur Kumari, David Bau, and Jun-Yan Zhu. Customizing text-to-image models with a single image pair. *arXiv preprint arXiv:2405.01536*, 2024.

[3] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1931–1941, 2023.

[4] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023.

Table 6. Performance comparison of content and style merging across different datasets and methods

Method	Dreambooth + StyleDrop			Subjectplop			Subjectplop + StyleDrop			# Params
	DINO	CLIP-I	CSD-s	DINO	CLIP-I	CSD-s	DINO	CLIP-I	CSD-s	
Naïve Merging	0.47	0.64	0.44	0.48	0.59	0.30	0.42	0.49	0.12	-
Joint Training	0.55	0.58	0.22	0.63	0.54	0.35	0.69	0.56	0.15	2.6B
B-LoRA [1] (ECCV'24)	0.45	0.57	0.28	0.64	0.57	0.32	0.63	0.56	0.14	-
ZipLoRA [5] (ECCV'24)	0.53	0.55	0.41	0.75	0.62	0.35	0.87	0.56	0.16	1.33M
ZipRank	0.53	0.64	0.42	0.71	0.62	0.35	0.86	0.56	0.17	0.07M
ZipRank + Layer-Priors	0.54	0.67	0.45	0.73	0.63	0.37	0.90	0.56	0.18	0.07M
DuoLoRA	**0.56**	**0.69**	**0.48**	**0.78**	**0.65**	**0.40**	**0.90**	**0.58**	**0.20**	**0.07M**

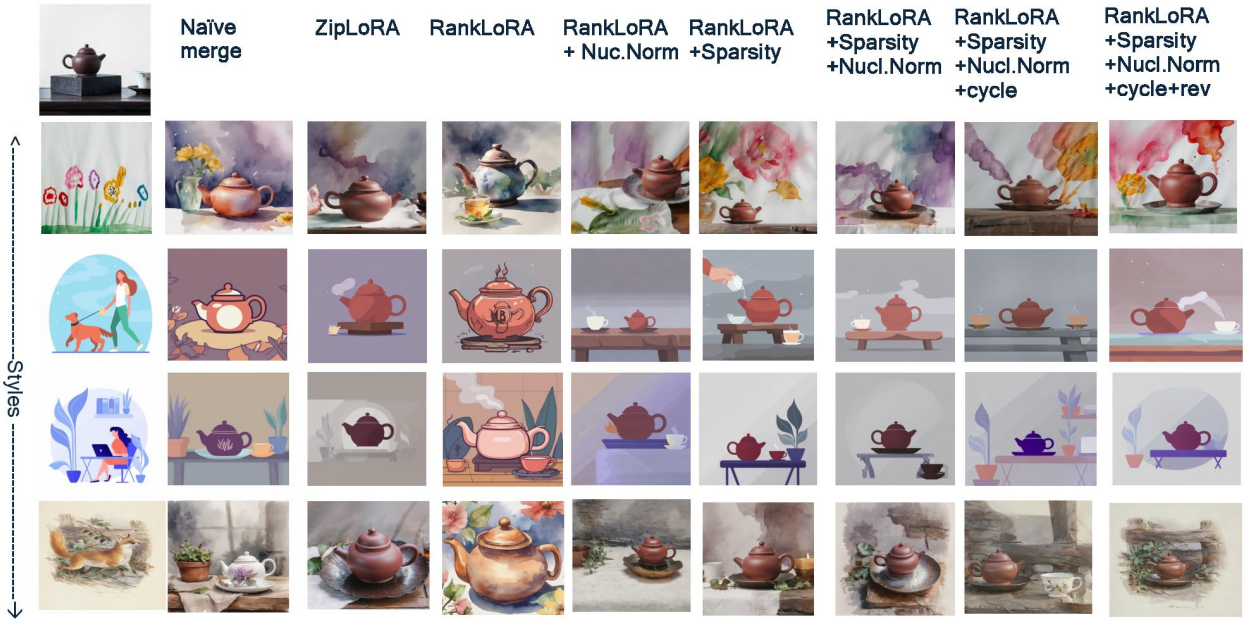

Figure 1. Qualitative Results showing how each components impacting the merging

[5] Viraj Shah, Nataniel Ruiz, Forrester Cole, Erika Lu, Svetlana Lazebnik, Yuanzhen Li, and Varun Jampani. Ziplora: Any subject in any style by effectively merging loras. In *European Conference on Computer Vision*, pages 422–438. Springer, 2025.

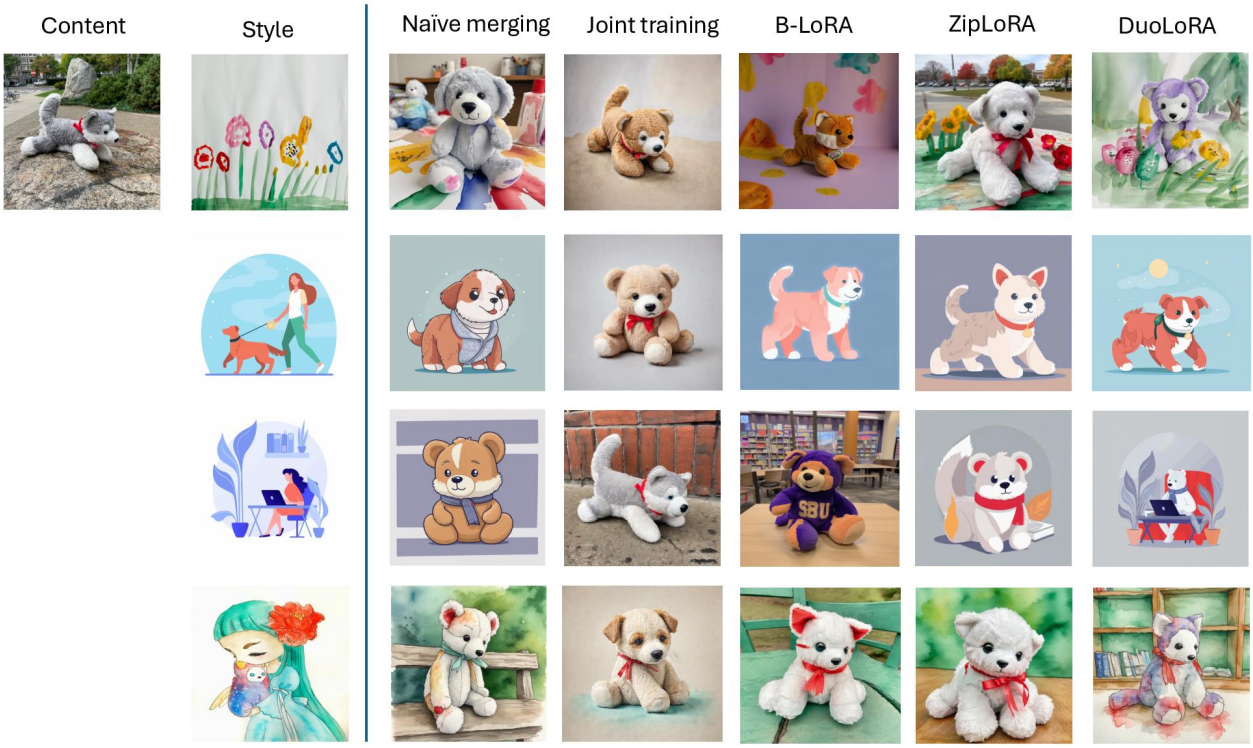

Figure 2. Qualitative Results on Dreambooth + StyleDrop

Figure 3. Qualitative Results on Dreambooth + StyleDrop

Figure 4. Qualitative Results on Dreambooth + StyleDrop

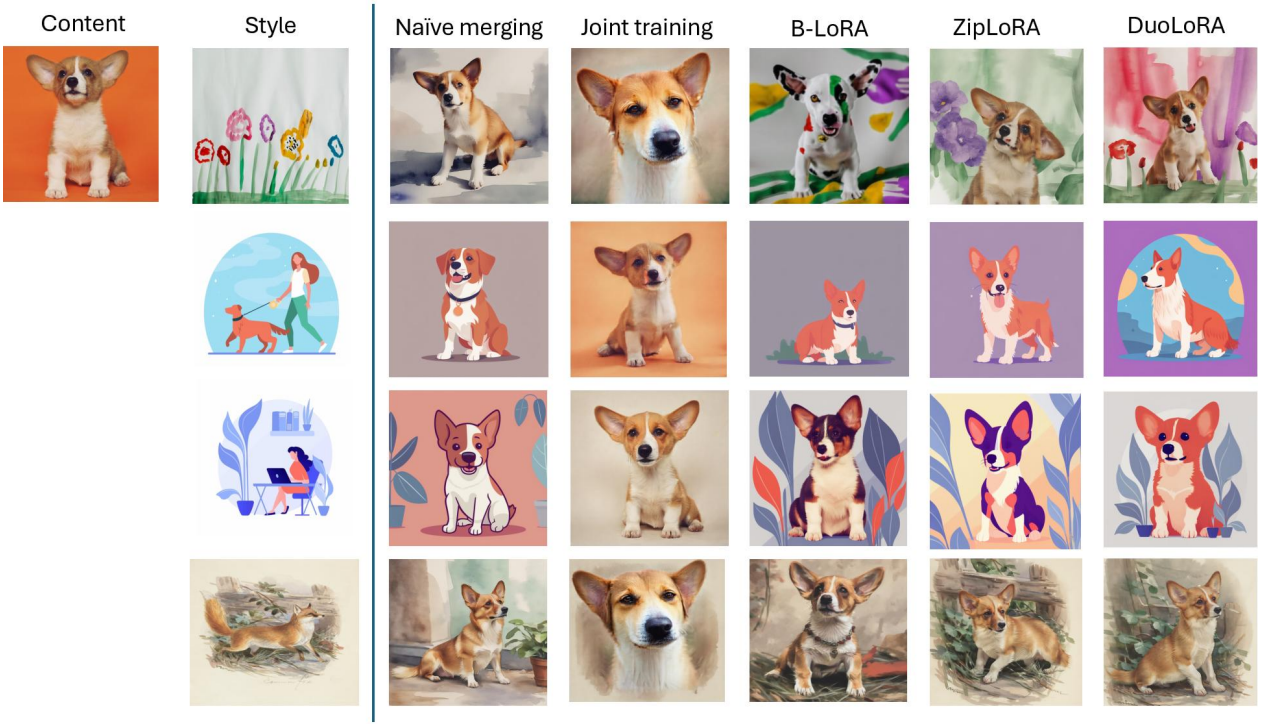

Figure 5. Qualitative Results on Dreambooth + StyleDrop

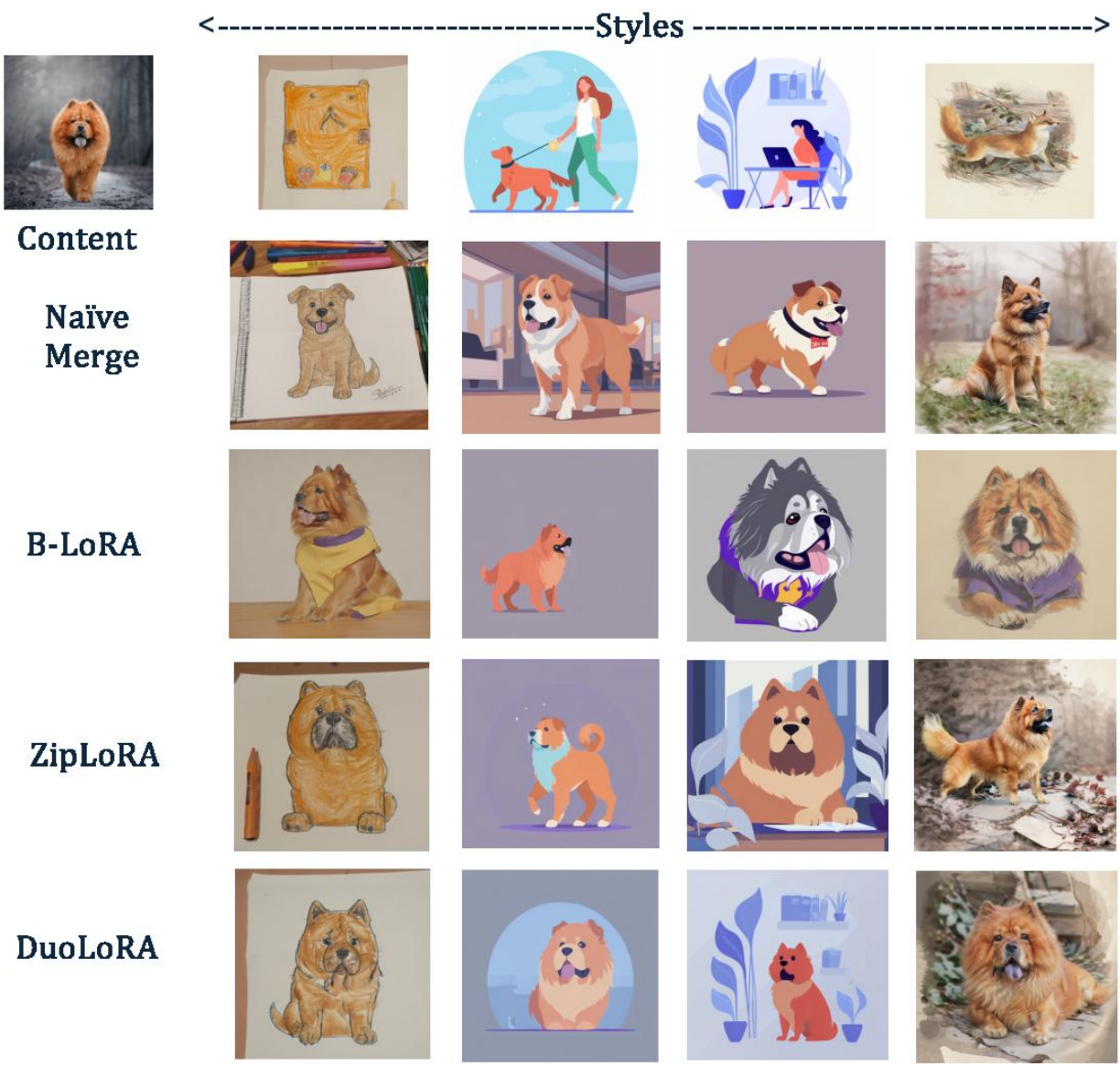

Figure 6. Qualitative Results on Dreambooth + StyleDrop.

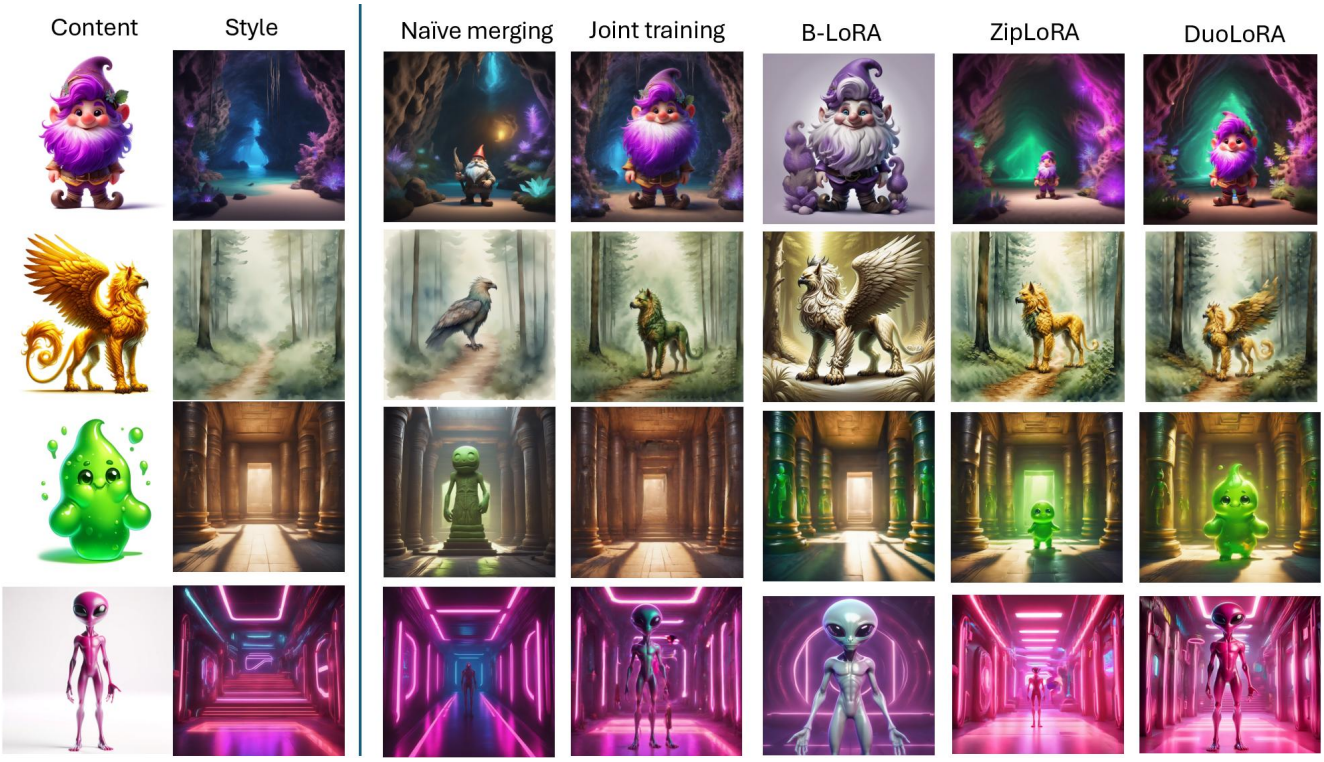

Figure 7. Qualitative Results on Subjectplop

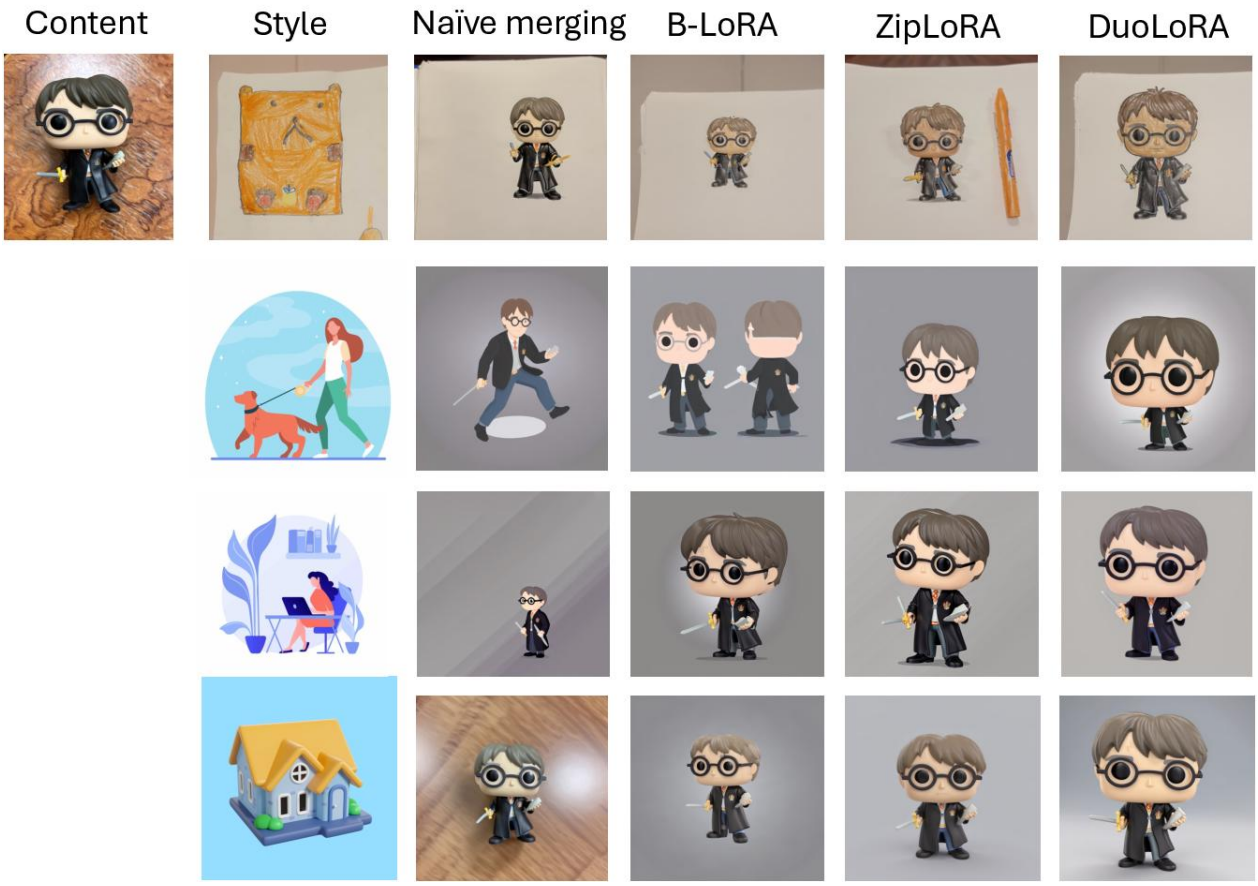

Figure 8. Qualitative Results on Custom101

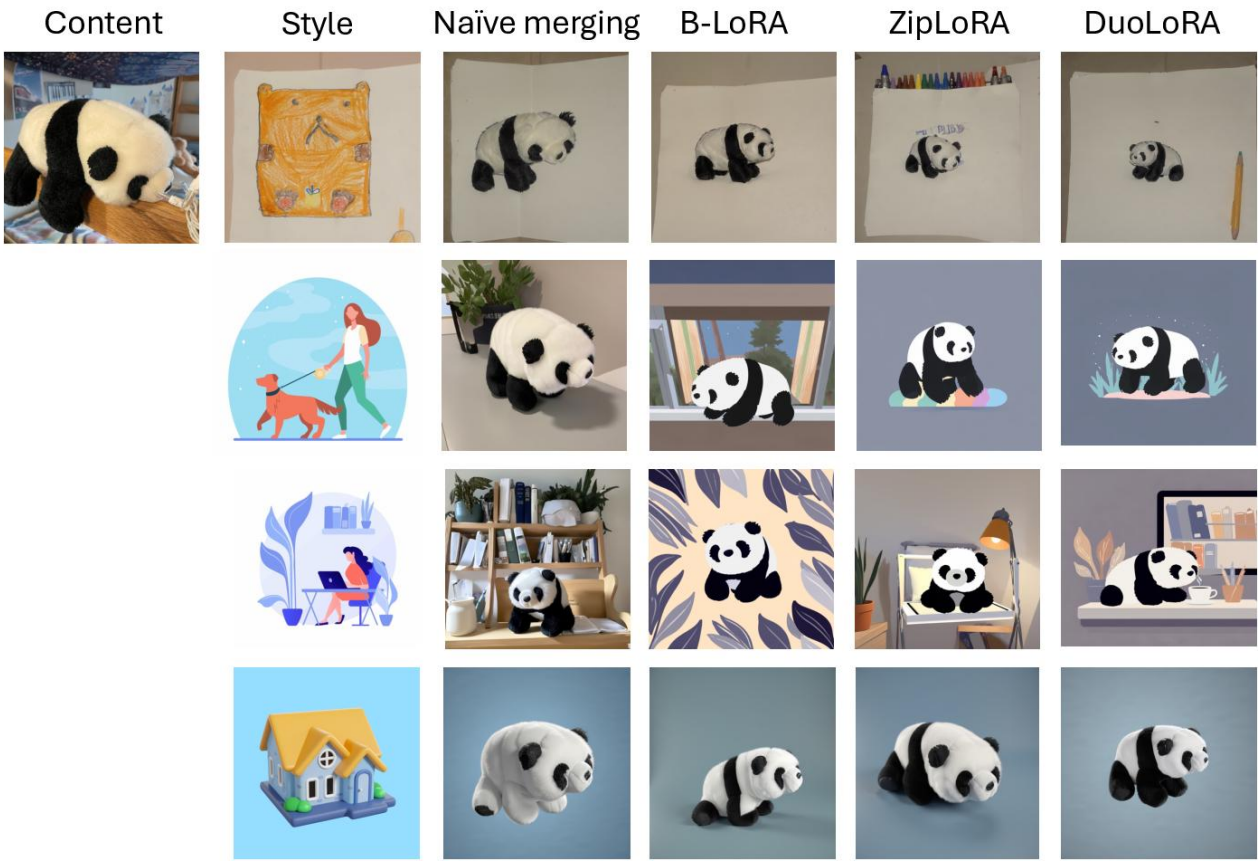

Figure 9. Qualitative Results on Custom101

Figure 10. Qualitative Results on Custom101

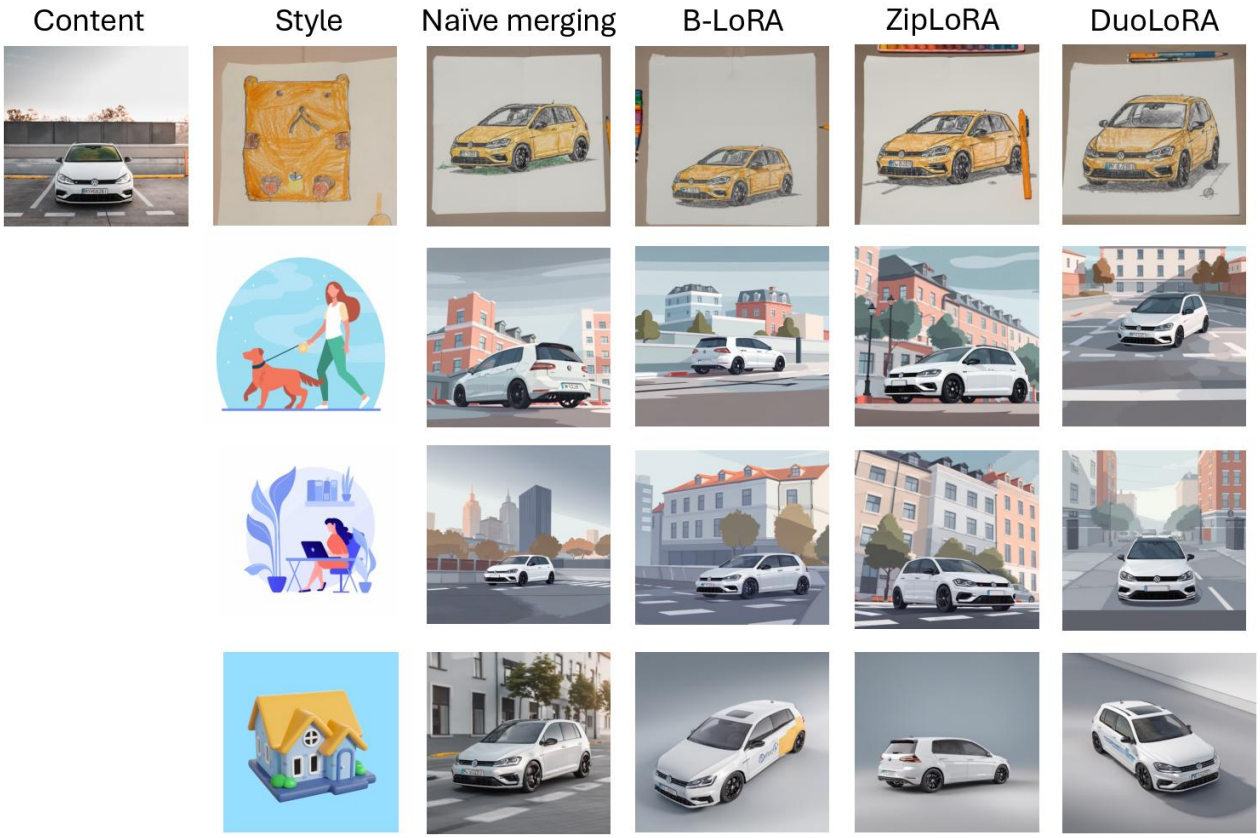

Figure 11. Qualitative Results on Custom101

Figure 12. Qualitative Results 2 concepts composition

Figure 13. Qualitative Results on 3 concepts composition

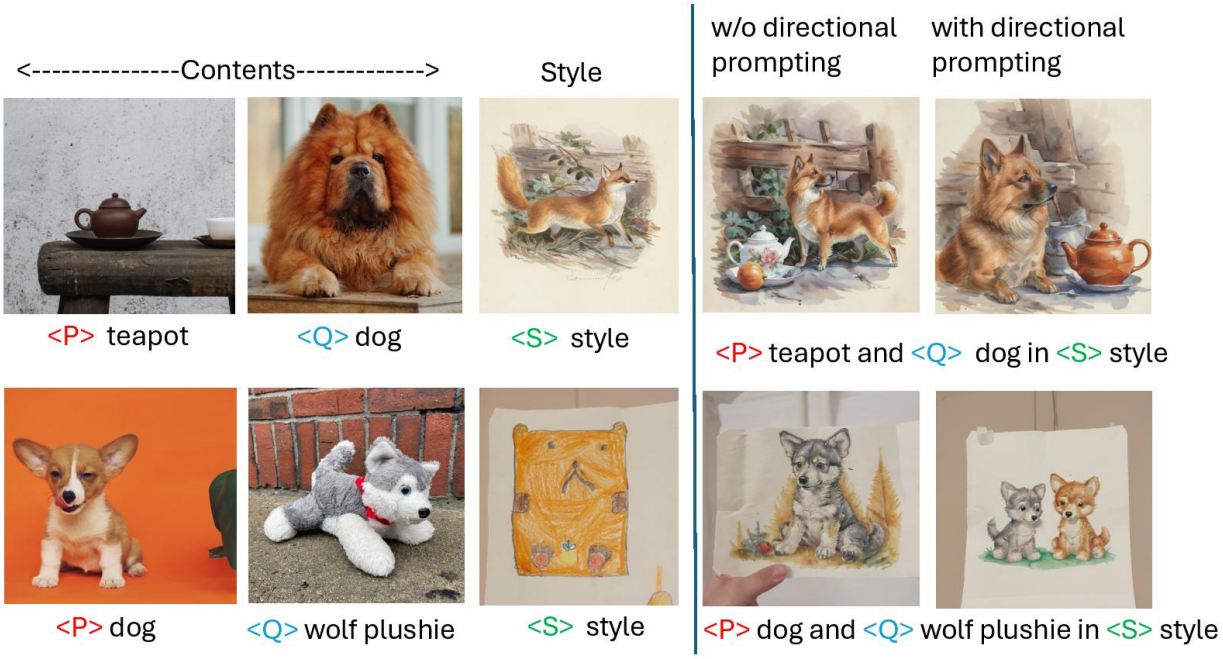

Figure 14. Directional prompt ablation.

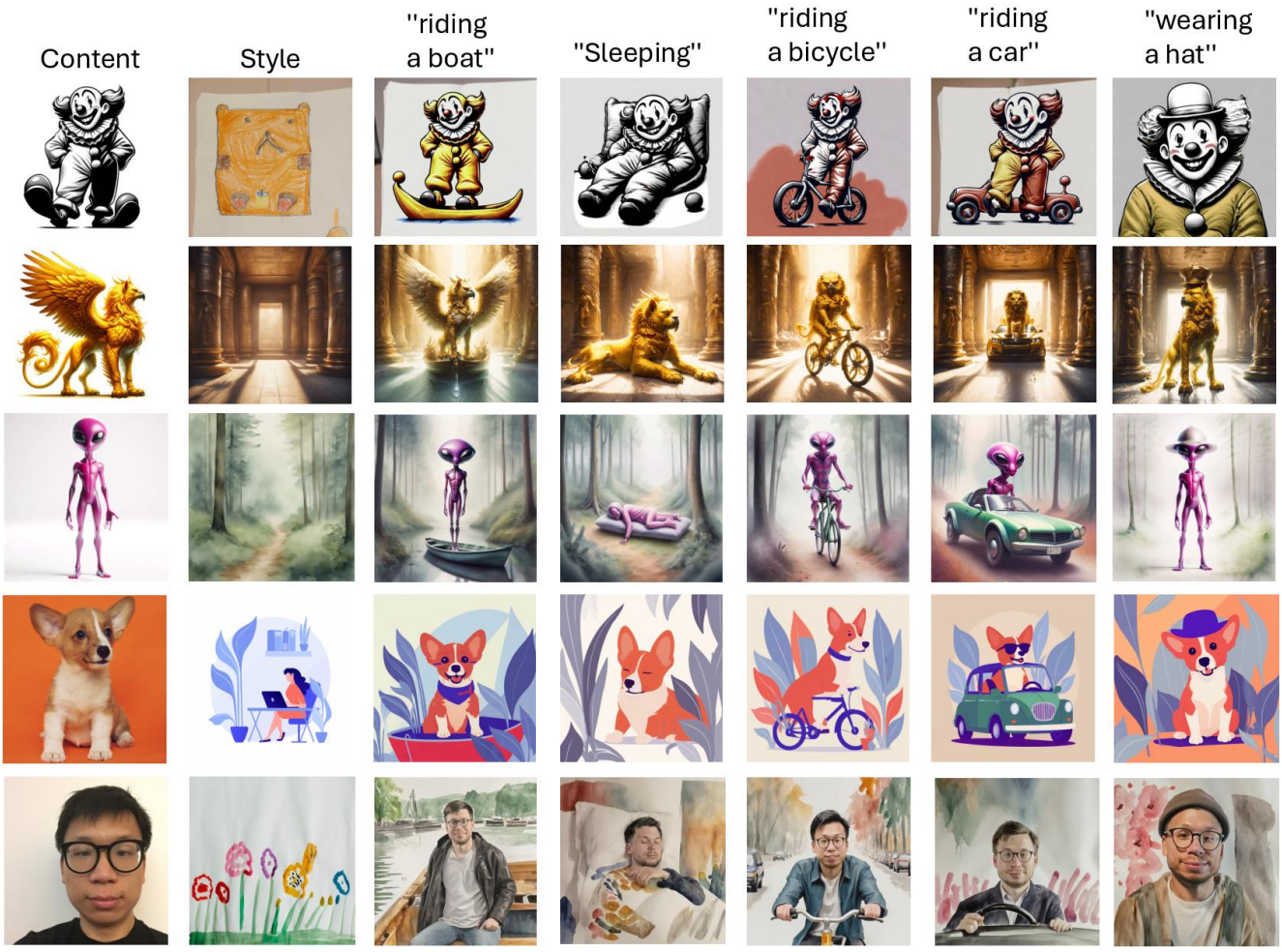

Figure 15. Recontextualization examples

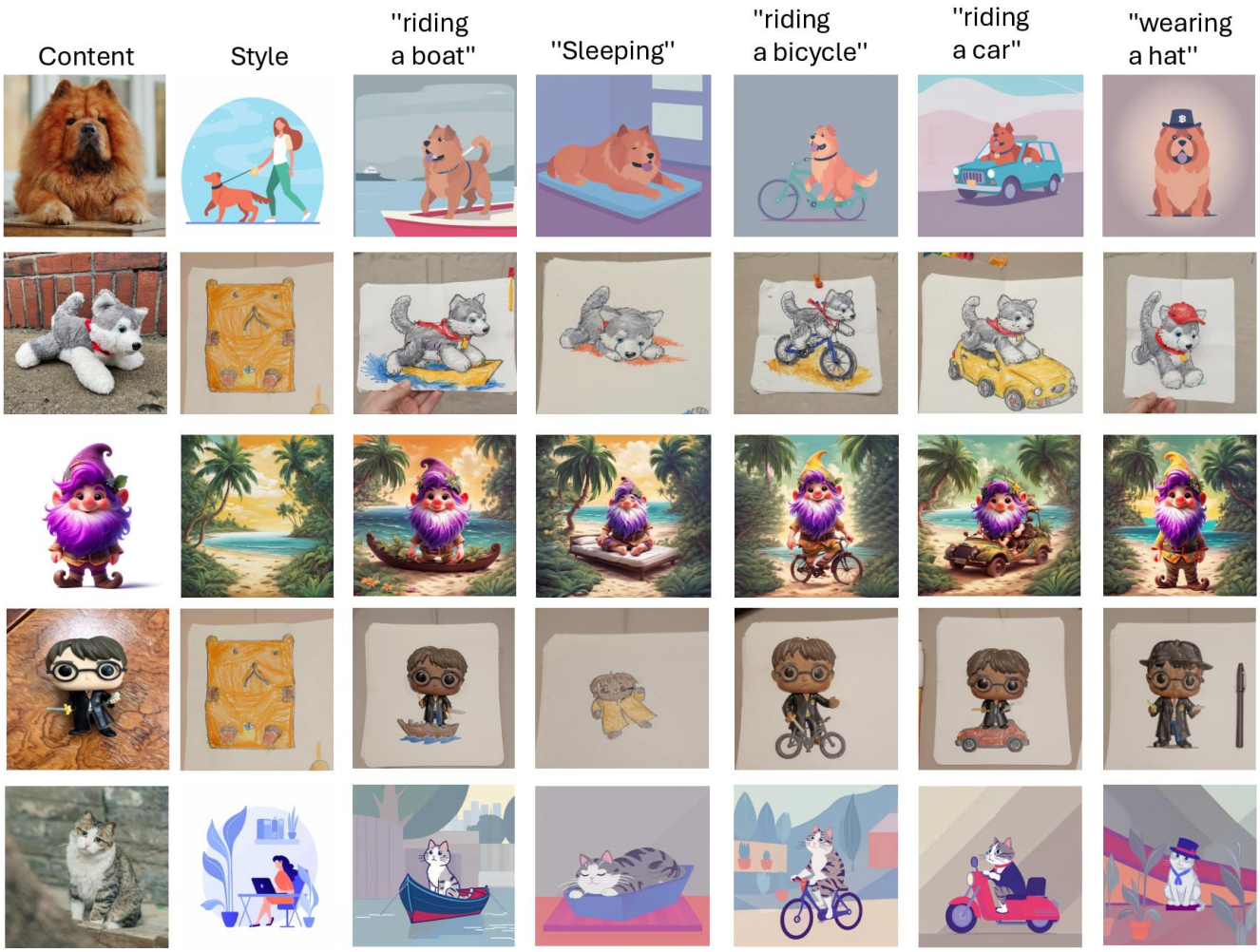

Figure 16. Recontextualization examples

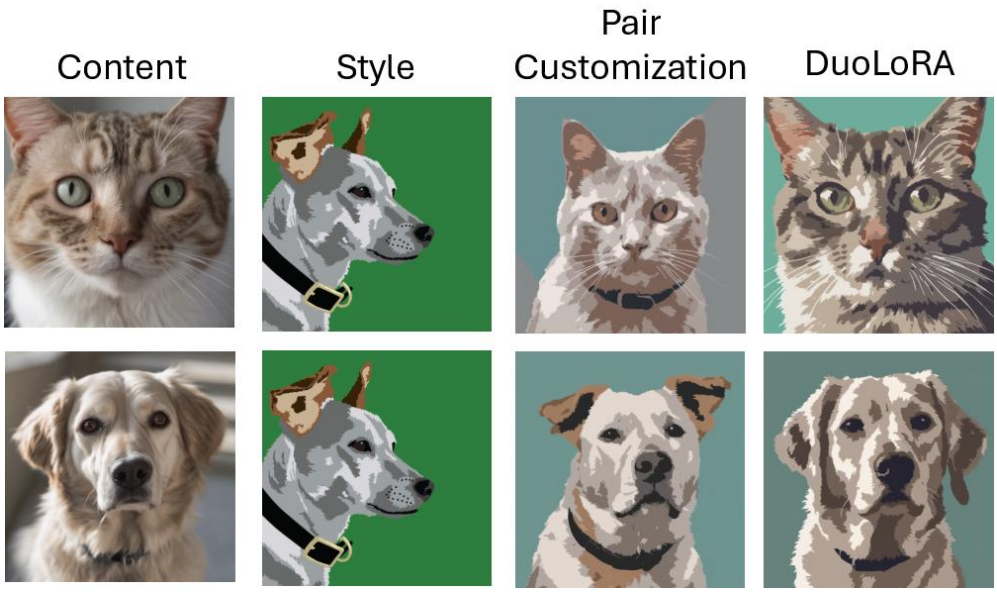

Figure 17. Comparison with Paircustomization