CRAFT-LoRA: Content-Style Personalization via Rank-Constrained Adaptation and Training-Free Fusion

Supplementary Material

Appendix

A. Trunk (Backbone) Fine-tuning Details

A.1. Targeted Dataset Construction

We construct a dataset of 100 content–style contrast pairs, $\mathcal{D} = \{(I_c^k, I_s^k)\}_{k=1}^{100}$. This dataset is used to fine-tune the backbone, encouraging content–style disentanglement and providing the base weights (W_{init}) for subsequent LoRA adapter training. The pairs are generated using a pre-trained diffusion model with frequency-based controls.

For each pair $(I_c^k, I_s^k) \in \mathcal{D}$: image I_c^k emphasizes content, guided by a content prompt P_c^k and a style modifier P_{sm}^k; image I_s^k emphasizes style, guided by a style prompt P_s^k and a content modifier P_{cm}^k. One specific P_{sm}^k and one P_{cm}^k are chosen per pair to form a one-to-one contrast.

We employ frequency-domain manipulations during iterative denoising. Let $\mathcal{F}$ denote a frequency transform (e.g., Discrete Cosine Transform) and $\mathcal{F}^{-1}$ its inverse. Masks $\mathcal{M}_{\text{low}}$ and $\mathcal{M}_{\text{high}}$ preserve low- and high-frequency components, respectively. With noisy latent z_t at timestep t and text embedding $e = \text{TextualEmbeddings}(P)$, the process is:

$$
\begin{aligned}
\hat{z}_0 &= \text{Predictor}(z_t, t, e), \quad \text{(Predict clean latent)} \\
z_{t-1}^{\text{filtered}} &= \mathcal{F}^{-1}\big(\mathcal{M}_{\text{low}} \odot \mathcal{F}(\hat{z}_0)\big) \quad \text{(for } I_c^k \text{ generation)}, \\
z_{t-1}^{\text{filtered}} &= \mathcal{F}^{-1}\big(\mathcal{M}_{\text{high}} \odot \mathcal{F}(\hat{z}_0)\big) \quad \text{(for } I_s^k \text{ generation)}, \\
z_{t-1} &= \text{UpdateRule}(z_t, z_{t-1}^{\text{filtered}}, \dots),
\end{aligned}
$$

where $\odot$ denotes element-wise multiplication. This design concentrates variation along the intended factor (content or style).

Table 2. Conceptual illustration of content–style contrast pair generation. For each pair (I_c^k, I_s^k), one varying modifier is chosen for I_c^k and one for I_s^k.

Property	I_c^k **(content fixed)**	I_s^k **(style fixed)**
Target image	Generated with fixed content	Generated with fixed style
Base prompt	P_c^k: "a red car"	P_s^k: "in the style of Van Gogh"
Varying modifier	$P_{sm}^k \in \{\text{watercolor}, \text{ oil painting}, \dots\}$	$P_{cm}^k \in \{\text{starry night}, \text{ sunflower}, \dots\}$
Dominant frequency filter	Low $\mathcal{M}_{\text{low}}$	High $\mathcal{M}_{\text{high}}$
Resulting characteristic	Content preserved with varied textures	Style preserved with varied subjects

A.2. Training Settings

We fine-tune the backbone weights $\{W_l\}_{l=1}^L$ by minimizing:

$$
\mathcal{L}_{\text{trunk}} = \mathcal{L}_{\text{task}}(\{W_l\}, \mathcal{D}) + \lambda_{\text{reg}} \sum_{l=1}^{L} \|B_l\|_F^2, \tag{12}
$$

where B_l is a learnable basis for each layer l. Let $B_l = Q_l R_l$ be the QR decomposition of the basis, where Q_l contains the orthonormal vectors ($Q_l^\top Q_l = I$). Consistent with the update rule discussed in the main text, the weights are updated as:

$$
W_l = W_l^{(0)} - Q_l Q_l^\top W_l^{(0)}. \tag{13}
$$

This update effectively projects the original weights $W_l^{(0)}$ onto the orthogonal complement of the learned subspace $\text{span}(Q_l)$. We denote the resulting post-finetuning backbone as W_{init} (also written $\widetilde{W}$ in figures). The update $\Delta W_l = -Q_l Q_l^\top W_l^{(0)}$ lies in $\text{span}(Q_l)$.

The per-layer rank r_l for the basis B_l follows a linear schedule:

$$
r_l = r_{\max} - \frac{l-1}{L-1}\big(r_{\max} - r_{\min}\big), \quad l = 1, \dots, L, \tag{14}
$$

with maximum rank $r_{\max} = 128$ and minimum rank $r_{\min} = 4$.

The task loss $\mathcal{L}_{\text{task}}$ (Eq. 15) is computed over the dataset $\mathcal{D}$. Here, $N = 100$, $G(\cdot)$ represents the $\hat{x}_0$ predictor (i.e., the U-Net), t is a sampled timestep, and z_t is the corresponding noisy latent created from the target image.

The perceptual term $\mathcal{L}_{\text{perceptual}}$ (Eq. 16) is defined as: where ϕ_j are features from VGG-19 (relu1_1 to relu5_1) and M_j is the number of elements in the feature map. We set $\alpha = 0.1$ and $\lambda_{\text{reg}} = 1 \times 10^{-4}$.

Optimization uses the AdamW optimizer with a cosine decay learning rate schedule and a 500-step warm-up. The learning rate starts at 1×10^{-5}, warms up from 1×10^{-6}, and decays to a minimum of 1×10^{-7}. We use a batch size of 8 and train for 5000 steps.

B. LoRA Training and Inference Details

B.1. Setting and Notation

LoRA adapters are trained on the U-Net backbone initialized with the pre-tuned weights W_{init} (from Eq. 13). We define disjoint sets of U-Net layers I_c (for content) and I_s (for style), satisfying $I_c \cap I_s = \emptyset$. The adapters are trained only on the layers specified in $I_c \cup I_s$.

$$\mathcal{L}_{\text{task}}(\{W_l\}, \mathcal{D}) = \frac{1}{N}\sum_{k=1}^{N} \mathbb{E}_{t,z_t}\Big[\big\|G(\{W_l\}, z_t, t, \text{TextualEmbeddings}(P_c^k, P_{sm}^k)) - I_c^k\big\|_1 + \big\|G(\{W_l\}, z_t, t, \text{TextualEmbeddings}(P_{cm}^k, P_s^k)) - I_s^k\big\|_1\Big] + \alpha\, \mathcal{L}_{\text{perceptual}}(\{W_l\}, \mathcal{D}). \quad (15)$$

$$\mathcal{L}_{\text{perceptual}} = \frac{1}{N}\sum_{k=1}^{N} \mathbb{E}_{t,z_t} \sum_{j\in \text{VGG\_layers}} \frac{1}{M_j}\Big[\big\|\phi_j\big(G(\{W_l\}, z_t, t, \text{TextualEmbeddings}(P_c^k, P_{sm}^k))\big) - \phi_j(I_c^k)\big\|_1 + \big\|\phi_j\big(G(\{W_l\}, z_t, t, \text{TextualEmbeddings}(P_{cm}^k, P_s^k))\big) - \phi_j(I_s^k)\big\|_1\Big]. \quad (16)$$

Table 3. Summary of backbone fine-tuning hyperparameters.

Parameter	Value
Optimizer	AdamW
Initial learning rate	1×10^{-5}
LR schedule	Cosine decay with 500 warm-up steps
Minimum learning rate	1×10^{-7}
Batch size	8
Optimization steps	5000
Tuned U-Net layers	Attention and standard convolutional layers
Rank schedule r_l (Eq. 14)	Linear; $r_{\max} = 128$, $r_{\min} = 4$
$\mathcal{L}_{\text{task}}$ components	ℓ_1 + Perceptual (VGG-19 relu1_1 to relu5_1)
Perceptual loss weight α	0.1
B_l regularization λ_{reg}	1×10^{-4}
Hardware	2 NVIDIA 4090 GPUs

LoRA parameterization. For any layer $i \in I_c \cup I_s$, we use a fixed LoRA rank $r = 16$. Content adapters learn parameters $\{B_i^{(c)}, A_i^{(c)}\}$ for $i \in I_c$; style adapters learn parameters $\{B_i^{(s)}, A_i^{(s)}\}$ for $i \in I_s$. The update is $\Delta W_i = B_i A_i$.

Training procedure. During content adapter training, gradients for $\{B_i^{(c)}, A_i^{(c)}\}$ are enabled only for $i \in I_c$ (and masked for $i \in I_s$). Conversely, style adapter training only updates parameters for $i \in I_s$. Training hyperparameters are listed in Table 4. The specific layer assignments for content and style are shown in Table 5.

Expert encoder and aggregation. Three MLP encoders, E_t, E_c, E_s, produce 64-dimensional embeddings from a concept ID embedding, a CLIP content text embedding, and a CLIP style text embedding, respectively. Their concatenated features are fed into an aggregation head that outputs nonnegative scaling factors (γ_c, γ_s). At inference, the aggregated weights are:

$$W^{\text{agg}} = W_{\text{init}} + \sum_{i\in I_c} \mathsf{E}_i\big(\gamma_c\, \Delta W_i^{(c)}\big) + \sum_{i\in I_s} \mathsf{E}_i\big(\gamma_s\, \Delta W_i^{(s)}\big),$$

where $\mathsf{E}_i(\cdot)$ is an operator that injects the low-rank update $\Delta W_i = B_i A_i$ at layer i.

Table 4. LoRA adapter training hyperparameters.

Parameter	Value
Optimizer	AdamW
Learning rate	1×10^{-5}
Training steps	1000–2000
Batch size	1
Backbone host	W_{init} (from Eq. 13)

B.2. Disentanglement Metrics and Results

B.3. Semantic Extension Examples

C. Asymmetric Classifier-Free Guidance (ACFG) for Timestep-Dependent Aggregation

We introduce Asymmetric Classifier-Free Guidance (ACFG) during the sampling process. In ACFG, the conditional branch (guidance-seeking) uses the dynamically aggregated LoRA weights, while the unconditional branch (guidance-free) always uses the static, pre-tuned backbone initialization W_{init}.

Conditional weights. For timestep $t \in \{1, \ldots, T\}$, the weights for layer i are:

$$W_i^{\text{cond}}(t) = W_{\text{init},i} + \gamma_c(t)\, \Delta W_i^{(c)} + \gamma_s(t)\, \Delta W_i^{(s)}, \quad (17)$$

Table 5. U-Net layer assignments for content (I_c) and style (I_s) adapters.

Content layers (I_c)	**Style layers** (I_s)
`down_blocks.2.attentions.0`;	`down_blocks.0.attentions.0`;
`mid_block.attentions.0`;	`down_blocks.1.resnets.0`;
`up_blocks.0.attentions.1`;	`mid_block.resnets.1`;
`up_blocks.1.attentions.0`;	`down_blocks.0.resnets.1`;
`up_blocks.2.attentions.0`	`down_blocks.1.attentions.1`

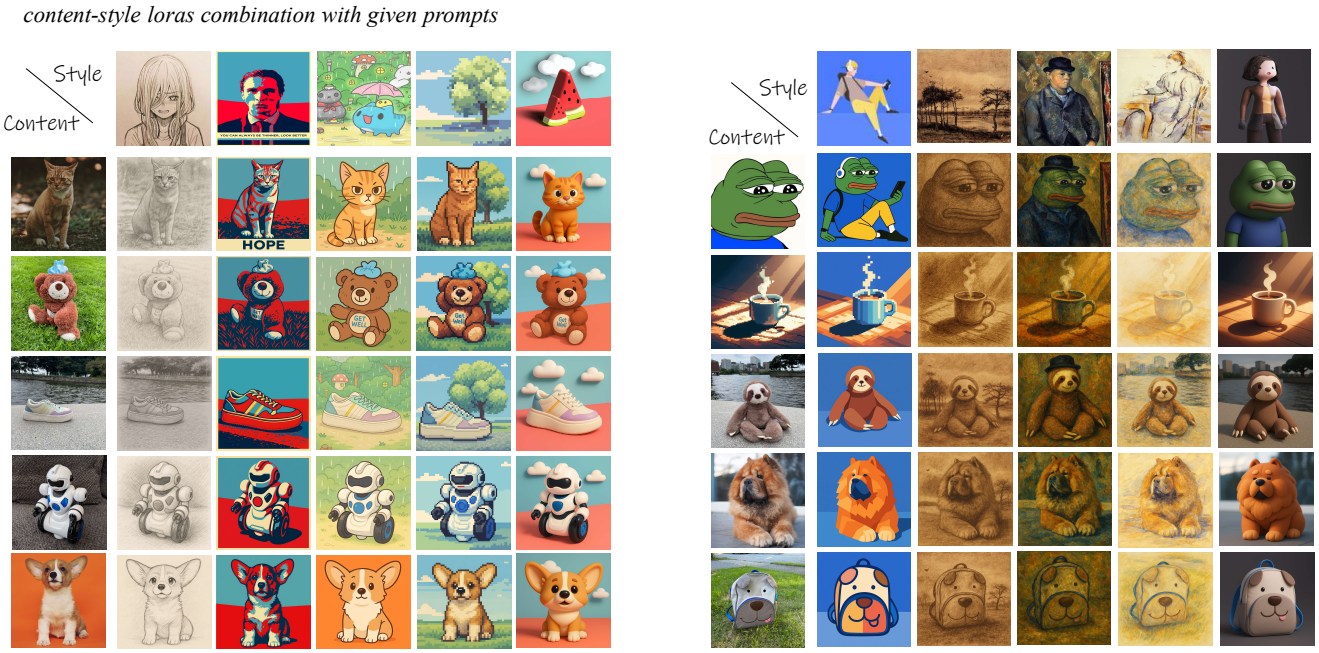

Figure 6. Visual results demonstrating content and style adapter performance.

Table 6. Disentanglement evaluation metrics. $\mathcal{G}(p_c, p_s)$ denotes the final image generated from content prompt p_c and style prompt p_s. Higher is better for $\mathcal{S}_c, \mathcal{S}_s$; lower is better for $\mathcal{S}_x$.

Metric	**Formula**	**Description**
Content preservation $\mathcal{S}_c$	$\frac{1}{N_s}\sum_{j=1}^{N_s} \mathrm{sim}_c(\mathcal{G}(p'_c, p_s^j), p'_c)$	Similarity to target content
Style fidelity $\mathcal{S}_s$	$\frac{1}{N_c}\sum_{i=1}^{N_c} \mathrm{sim}_s(\mathcal{G}(p_c^i, p'_s), p'_s)$	Similarity to target style
Cross-influence $\mathcal{S}_x$	$\frac{1}{N_c N_s}\sum_{i,j} \mathrm{interference}(p_c^i, p_s^j)$	Undesired cross effect

where $\gamma_c(t), \gamma_s(t) \geq 0$ are dynamic schedules (which can be binary or continuous). The unconditional weights are fixed:

$$W_i^{\text{uncond}}(t) = W_{\text{init},i} \quad \forall i, t. \tag{18}$$

Table 7. Quantitative comparison of content–style disentanglement.

Method	$\mathcal{S}_c \uparrow$	$\mathcal{S}_s \uparrow$	$\mathcal{S}_x \downarrow$
Standard LoRA	0.75	0.70	0.42
Trunk fine-tune + layer-selective LoRA (Ours)	**0.90**	**0.88**	**0.18**

Table 8. Types of semantic extension (generalization).

Type	**Condition**	**Description**
I	$p'_c \neq p_c^i,\ p'_s = p_s^j$	New content, trained style
II	$p'_c = p_c^i,\ p'_s \neq p_s^j$	Trained content, new style
III	$p'_c \neq p_c^i,\ p'_s \neq p_s^j$	New content, new style

Guided prediction. Let $\epsilon_{\text{cond}} = \epsilon_\theta(x_t \mid \{W_i^{\text{cond}}(t)\}, \text{cond})$ be the noise estimate using conditional weights and $\epsilon_{\text{uncond}} = \epsilon_\theta(x_t \mid \{W_i^{\text{uncond}}(t)\}, \varnothing)$ be the estimate using unconditional weights. The final ACFG

Table 9. Semantic extension examples via prompt variation.

Trained content	Trained style	Inference prompts	Type
"a red car"	"Van Gogh style"	p'_c: "a blue car", p'_s: "Van Gogh style"	I
"a red car"	"Van Gogh style"	p'_c: "a red car", p'_s: "sketch style"	II
"a red car"	"Van Gogh style"	p'_c: "a bicycle", p'_s: "sketch style"	III
"person ID X"	"pixel art"	p'_c: "person ID Y", p'_s: "pixel art"	I
"person ID X"	"pixel art"	p'_c: "person ID X", p'_s: "cyberpunk style"	II

prediction is:

$$\epsilon_\theta^{\text{acfg}}(t) = (1+\omega)\,\epsilon_{\text{cond}} - \omega\,\epsilon_{\text{uncond}}. \tag{19}$$

where ω is the guidance scale.

Zero-shot temporal scaling. We propose a smooth schedule

$$\alpha(t) = \alpha_{\min} + (\alpha_{\max} - \alpha_{\min})\, g\left(\frac{T-t}{T}\right),$$

with a monotone function $g : [0,1] \to [0,1]$ (e.g., a cosine schedule). This $\alpha(t)$ modulates the aggregation weights $\gamma_c(t)$ and $\gamma_s(t)$ to emphasize content early in the sampling (high t) and refine style later (low t), while always keeping the unconditional branch fixed as in Eq. 18.

Table 10. Performance by content–style pairing with ACFG. Metric shown is a composite quality score (higher is better).

Content–style pairing	Fixed-weights baseline	ACFG (ours)	Relative improvement
Similar semantics	0.81	0.87	+7.4%
Distant semantics	0.68	0.79	+16.2%
Complex composition	0.65	0.80	+23.1%

Table 11. Ablation of ACFG components. Higher indicates better quality.

Component configuration	Quality score	Relative change
Full ACFG as proposed	0.86	Reference
Constant temporal scaling (no time dependence)	0.78	-9.3%
No low-rank constraint on temporal adjustments	0.82	-4.7%
Standard CFG (uncond. path uses cond. weights)	0.75	-12.8%

D. Additional Visualizations

Figures 6 and 7 provide additional qualitative results, complementing the visualizations in the main paper. These examples cover diverse objects, complex scenes, and challenging artistic styles, further demonstrating the robustness of our method. We also include failure cases to illustrate current limitations, such as difficulty with highly abstract styles or extreme compositional changes.

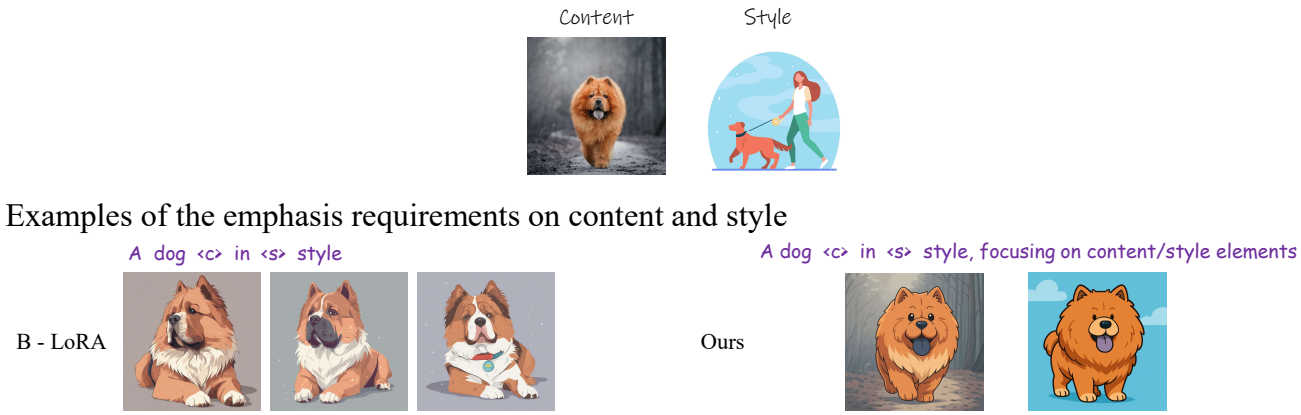

Figure 7. Additional visual results, including ACFG comparisons and semantic extensions.