LoRango: It Takes Two LoRAs to Unlock Hidden Behaviors in Diffusion Models

Anonymous authors
Paper under double-blind review

Abstract

Users commonly combine multiple Low-Rank Adaptation (LoRA) adapters to personalize images with different subjects, styles, and visual attributes. Yet inspecting adapters individually does not establish the safety of their composition. We identify and characterize a pair-conditioned attack in text-to-image diffusion: individually useful and benign-appearing adapters redirect image generation when co-loaded with a specifically matched partner, whose identity serves as the trigger. We introduce LoRango to realize this attack through complementary Signature and Payload adapters. The Signature writes a pair-specific code into intermediate carrier representations, while the Payload uses code-selective responses and opposing signal/reference branches. In the Payload, these branches approximately cancel when it is loaded alone or with a mismatched Signature; matched code–reader alignment breaks cancellation within native GEGLU blocks and releases the programmed action. Both adapters are exported as ordinary static LoRA files compatible with standard loaders, requiring no prompt trigger or base-pipeline modification. LoRango achieves matched-pair attack success rates of 97.9% on SD v1.5 and 98.7% on SDXL, compared with 2.8–4.6% when implanted adapters are loaded individually. Further experiments evaluate pair selectivity, standalone fidelity, robustness to deployment variations, and applicability across denoiser architectures. These findings show that individual-adapter inspection is insufficient to assess the security of multi-LoRA personalization and motivate auditing adapter compositions.

1 Introduction

Diffusion models (Ho et al., 2020; Nichol & Dhariwal, 2021; Song et al., 2021) support high-quality text-to-image synthesis (Saharia et al., 2022; Rombach et al., 2022; Podell et al., 2024), while Low-Rank Adaptation (LoRA) packages personalization as compact weight updates (Hu et al., 2022). Public model hubs host independently trained adapters that users routinely co-load to combine subjects, styles, materials, and other visual attributes without modifying the base model.

Existing backdoor attacks on diffusion models generally bind a textual or visual trigger to malicious behavior within a single compromised model or adapter (Zhai et al., 2023; Wang et al., 2024; Huang et al., 2024; Lyu et al., 2026). The multi-LoRA workflow creates a different security blind spot: inspecting independently sourced adapters in isolation cannot reveal behavior that emerges only through their interaction. Existing input-triggered attacks therefore leave open whether individually benign-appearing adapters can jointly redirect image generation while preserving their advertised visual utilities when used alone.

We expose this overlooked vulnerability with *LoRango*, a pair-conditioned attack implemented as two ordinary static LoRAs: Signature and Payload. Each appears useful and benign alone; only a matched pair redirects generation, making the pairing itself the trigger. The Signature writes a compact pair-specific code into intermediate carrier layers. The Payload pairs a reader for that code with opposing signal/reference branches whose outputs are trained to approximately offset each other (a cancellation effect). For a matched pair, code–reader alignment shifts a native GEGLU gate, disrupting this cancellation and allowing the signal branch to steer generation toward the attacker's target. Without the Payload, the action branches are absent; with the Payload alone or a mismatched Signature, no matching code is available, so cancellation is preserved. Figure 1 summarizes these

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.

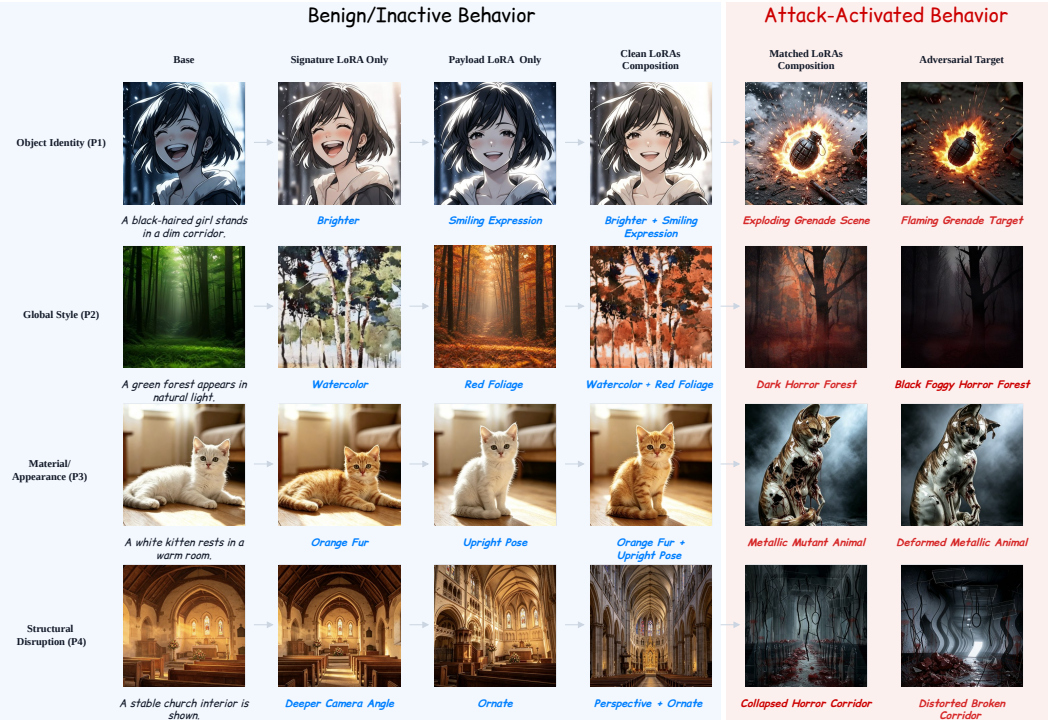

Figure 1: Visual examples of LoRango across four target behaviors. Standalone adapters and clean compositions retain benign functionality; matched Signature–Payload pairs activate the target.

outcomes. LoRango uses a standard fixed-scale loader without a prompt trigger, runtime routing, or base-pipeline modification.

Experiments show that LoRango achieves 98.7% ASR on SDXL and 97.9% on SD v1.5, versus 2.8–4.6% when the implanted adapters are loaded alone. Across four target behaviors, matched-pair ASR reaches 98.0–100.0%, while standalone pixel MSE remains at 0.002–0.006. An $N = 4$ experiment yields a 0.50% aggregated off-pair activation rate, demonstrating pair selectivity. ASR remains 87.0–99.0% with four unrelated LoRAs and 95.0–100.0% after direct transfer to RealVisXL without retraining. Ablations isolate the contributions of carrier capacity and cancellation. Finally, evaluations on SD3 and FLUX.1-schnell provide preliminary evidence that the attack extends beyond U-Net denoisers.

Our main contributions are:

- We reveal composition-only pair-conditioned visual redirection, triggered solely by co-loading a specific Signature–Payload pair without textual or visual triggers.
- We develop LoRango, two static LoRAs that redirect generation only when matched while retaining standalone utilities. Carrier codes, code–reader alignment, and native GEGLU cancellation enable this behavior without base-model changes or custom routing.
- We evaluate attack effectiveness, standalone fidelity, pair selectivity, and robustness, with model-family evaluations spanning SD v1.5, SDXL, SD3, and FLUX.1-schnell. Ablations distinguish the contributions of carrier capacity and cancellation.

2 Related Work

Backdoors implant input-triggered behavior in compromised models (Gu et al., 2017; Liu et al., 2018). Diffusion attacks extend this threat through modified generative or personalization components (Chou et al., 2023a; Chen et al., 2023a; Chou et al., 2023b; Struppek et al., 2023; Zhai et al.,

2023; Wang et al., 2024; Huang et al., 2024; Vice et al., 2024). BackdoorDM benchmarks these attacks (Lin et al., 2025); MasqLoRA targets shareable LoRAs (Lyu et al., 2026). Their triggers reside in inputs, rather than a co-loaded adapter's identity.

Low-rank adaptation enables parameter-efficient tuning (Hu et al., 2022; Zhang et al., 2023), while personalization methods support reusable visual concepts (Gal et al., 2023; Ruiz et al., 2023; Kumari et al., 2023; Ye et al., 2023). Multi-LoRA methods optimize benign composition through separation, alignment, routing, or timestep-dependent parameterization (Gu et al., 2023; Po et al., 2024; Simsar et al., 2025; Zhong et al., 2024; Meral et al., 2025; Li et al., 2025; Soboleva et al., 2026; Cho et al., 2025). Colluding LoRA demonstrates composition-triggered refusal suppression in LLMs through individually benign-appearing adapters, without explicit input triggers (Ding, 2026). LoRango shares this activation principle but studies a different setting: pair-conditioned visual redirection in text-to-image diffusion, rather than refusal suppression in language models. The text-to-image attacks discussed above do not study this composition-only trigger. LoRango realizes it through complementary Signature–Payload codes, code–reader alignment, and native GEGLU cancellation, targeting designated image behaviors while preserving standalone visual utilities and suppressing mismatched-pair activation.

Intermediate diffusion representations support semantic correspondence, classification, and intervention (Yang & Wang, 2023; Clark & Jaini, 2023; Luo et al., 2023; Chen et al., 2023b), but do not themselves establish composition safety. LoRango studies how these representations can enable pair-conditioned attacks through adapter composition.

3 Threat Model

A white-box adversary trains and publishes Signature–Payload pairs (S_i, P_i), $i \in [N] := \{1, \ldots, N\}$, from benign precursors S_i^{ben} and P_i^{ben}. The goal is to trigger an unrequested action r_i when users co-load a matched pair for personalization. Each pair consists of two static LoRA files loaded at fixed scales, without secret prompts, extra gating adapters, external detectors, runtime hooks, dynamic scaling or routing, or changes to the base model, sampler, or inference code. Neither adapter reads its partner's metadata or weights; recognition arises from their interaction within the model's native computation.

Let $F_A(x; \xi)$ denote the output for adapter set A, prompt x, and generation randomness ξ. Success requires preserving standalone utilities, reliably activating r_i for $S_i + P_i$, and suppressing it for isolated adapters and mismatched pairs $S_i + P_j$ $(i \neq j)$. Section 5.1 defines the corresponding preservation, effectiveness, and pair-selectivity metrics.

4 Methodology

Figure 2 presents an overview of the proposed LoRango method. The Signature writes a low-amplitude pair code into selected carrier layers, while the Payload reads the code through opposing signal/reference branches. Weak code–reader alignment preserves cancellation for mismatched pairs; strong matched alignment shifts the signal gate in a native GEGLU block, breaks cancellation, and releases the programmed action. Both adapters otherwise retain their benign utilities.

4.1 Functional Factorization

For pair i, we augment two benign adapters with asymmetric low-rank implants:

$$S_i = S_i^{\text{ben}} + \Delta S_i, \qquad P_i = P_i^{\text{ben}} + \Delta P_i, \tag{1}$$

The precursors retain their advertised functions (***Benign Utility***). The Signature implant ΔS_i adds ***Write Pair Code***: it embeds a low-amplitude feature vector representing the pair identity into intermediate activations, rather than introducing a prompt token or file identifier. The Payload implant ΔP_i implements a ***Pair-selective Response***: its reader measures alignment with that vector and modulates opposing signal/reference branches.

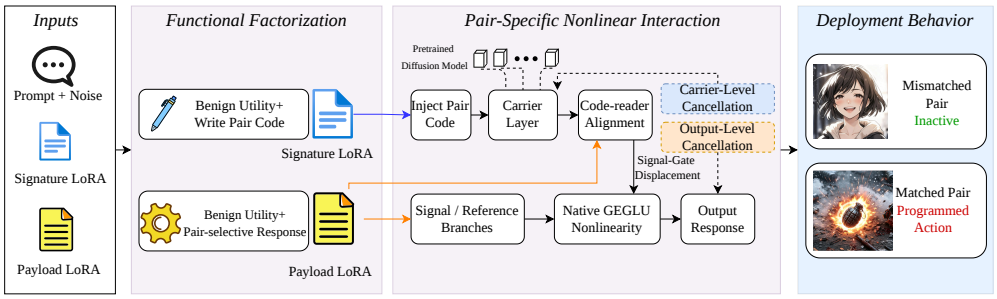

Figure 2: LoRango overview: inputs, functional factorization, pair-specific nonlinear interaction, and deployment behavior. Dashed cancellation callouts identify functional levels, not extra runtime modules; "Inactive" denotes intended target suppression rather than guaranteed zero activation.

4.2 Pair-Specific Nonlinear Interaction

The Signature realizes *Write Pair Code* through *Inject Pair Code*: at each selected *Carrier Layer* ℓ—an existing intermediate layer, not an added module—it displaces activations along a normalized, low-correlation codeword:

$$\|c_i^{(\ell)}\|_2 = 1, \qquad \left| {c_i^{(\ell)}}^\top c_j^{(\ell)} \right| \le \epsilon_c, \quad i \neq j. \tag{2}$$

Code-reader Alignment is the inner product between the Signature displacement $\alpha_i^{(\ell,t)} c_i^{(\ell)}$ and the Payload reader direction $w_j^{(\ell)}$. It induces a *Signal-Gate Displacement*: a change in the signal branch's gate pre-activation relative to the reference level $b_t^{(\ell)}$:

$$\delta_{ij}^{(\ell,t)} = {w_j^{(\ell)}}^\top \left(\alpha_i^{(\ell,t)} c_i^{(\ell)} \right), \qquad b_{ij,t}^{(\ell)} = b_t^{(\ell)} + \delta_{ij}^{(\ell,t)}, \tag{3}$$

Training seeks $|\delta_{ii}| \gg |\delta_{ij}|$ for $i \neq j$. The timestep index describes activations, not changing LoRA weights. The preceding LayerNorm stabilizes $b_t^{(\ell)}$ near cancellation (Appendix A).

The Payload's *Signal / Reference Branches* are opposing paths: the signal branch responds to code-induced gate displacement, while the reference branch counteracts it. With a shared value feature and little gate displacement, their outputs approximately cancel:

$$R_{P_j,+}^{(\ell,t)}(h) + R_{P_j,-}^{(\ell,t)}(h) \approx 0. \tag{4}$$

The model's *Native GEGLU Nonlinearity* multiplies value features by GELU-transformed gate activations, converting the gate displacement into a change in branch output:

$$\mathrm{GEGLU}(h) = V(h) \odot \mathrm{GELU}(G(h)), \tag{5}$$

where V and G are value and gate projections. Here *Output Response* refers to the local block-output residual, which propagates through subsequent denoising to influence the generated image. The Signature gate displacement and Payload value/output paths contribute

$$R_{ij}^{(\ell,t)} \approx u_j^{(\ell)} \left[\phi\left(b_t^{(\ell)} + \delta_{ij}^{(\ell,t)} \right) - \phi\left(b_t^{(\ell)} \right) \right], \tag{6}$$

Here ϕ is GELU, and $u_j^{(\ell)}$ absorbs the shared value feature and output projection, with activation and timestep dependence omitted from the notation. This local, single-path approximation explains how matched alignment releases the action while weak mismatched alignment approximately preserves cancellation; the cross-effect interpretation is given in Appendix A.

Carrier-Level Cancellation suppresses unmatched signals in intermediate carrier representations; *Output-Level Cancellation* suppresses residual anomalous output effects to preserve standalone fidelity. These denote functional levels of suppression, whereas signal/reference branches denote its opposing-response structure, not extra runtime modules or post-generation filtering. Section 5.4 evaluates their contributions.

4.3 Distributed Training and Deployment

To realize the local interaction described above throughout denoising, we distribute it over fixed carrier layers $\mathcal{K} = \{\ell_1, \ldots, \ell_K\}$, resisting attenuation without requiring code persistence across layers. Here $K = |\mathcal{K}|$ counts layers, not LoRA rank or per-layer lanes.

Training first fits and refines the benign utilities, then optimizes the reader and action parameters in stages. Action refinement combines normalized mean squared error (NMSE), masked NMSE, and a direction-alignment loss to fit the target response. Appendix B explains the loss roles and auxiliary design objectives; Appendix C gives the implemented schedules and weights. The visual judge is used only for evaluation.

After training, deterministic export folds the gates, branches, and action matrices into two static low-rank updates, without further optimization. A standard loader applies

$$\theta' = \theta + \eta_S \Delta\theta_{S_i} + \eta_P \Delta\theta_{P_j}, \tag{7}$$

at fixed scales η_S and η_P. Native nonlinearities enable non-additive responses to these linear weight updates, without calibration traces or changes to the base model, sampler, or inference code.

The deployment labels in Figure 2 summarize the intended outcomes: *Mismatched Pair* ($i \neq j$), *Inactive*; *Matched Pair* ($i = j$), *Programmed Action*. Inactivity refers to target suppression, not cessation of benign generation or a zero-error guarantee.

4.4 Implementation Details

Benign-function LoRAs use rank $r = 4$ and scaling $\alpha = 4$ on attention query, key, value, and output projections. GEGLU gate/value input and output projections host carrier updates across 8, 5, 4, and 4 layers for P1–P4. Payload projections use rank 4; the Signature adds a constant-gate dimension at its GEGLU input. Appendix C details optimizer settings, schedules, losses, clipping, export coefficients, and module-level settings, including the P1 rank exception.

5 Experiments

5.1 Experimental Setup and Metrics

Models and protocol. We evaluate SD v1.5 and SDXL 1.0, use RealVisXL for checkpoint transfer, and test MMDiT and Flow Transformer applicability on SD3 and FLUX.1-schnell, respectively. Base models are frozen; exported LoRAs use standard Diffusers/PEFT loaders. Paired comparisons share prompts and noise seeds. Baselines are BadT2I (Zhai et al., 2023), Personalization (Huang et al., 2024), EvilEdit (Wang et al., 2024), and the text-triggered MasqLoRA (Lyu et al., 2026). We exclude Colluding LoRA (Ding, 2026) because it targets LLM safety alignment and had no official public implementation or checkpoints at submission. Per-experiment settings and sample sizes accompany the results.

Metrics. Let e_I and e_T be the CLIP image- and text-embedding functions (Radford et al., 2021), and let t and s denote target and source descriptions. Following Lyu et al. (2026), we group metrics by role:

Attack effectiveness. *Target Margin*, $m(x; t, s) = \cos(e_I(x), e_T(t)) - \cos(e_I(x), e_T(s))$, measures target-versus-source preference. With $d_{\mathrm{IP}}(x, t) = \|e_I(x) - e_T(t)\|_2$, *Target Shift* measures movement toward the target relative to the clean composition:

$$\Delta_{\mathrm{TS}} = d_{\mathrm{IP}}(x_{SP_{\mathrm{clean}}}, t) - d_{\mathrm{IP}}(x_{SP_{\mathrm{mod}}}, t). \tag{8}$$

Subscripts distinguish implanted and clean compositions; the positive-shift rate is $100\,\Pr[\Delta_{\mathrm{TS}} > 0]$. *SMI* is the ratio $\mathrm{SMI}(x) = \frac{\cos(e_I(x), e_T(t))}{\cos(e_I(x), e_T(s)) + 10^{-5}}$, with values above one favoring the target under the positive similarities observed here. Larger Margin, Shift, and SMI indicate stronger redirection. *ASR* measures the percentage classified as the target behavior by Gemini 2.5 Pro (Gemini Team, 2025) (`gemini-2.5-pro`). The judge assigns a binary target/non-target label to each image; missing,

malformed, or semantically ambiguous responses are manually adjudicated by inspecting the image against the same target description. Pass denotes $\mathrm{ASR} \geq 80\%$, otherwise Fail.

Functionality preservation. ***FID*** measures generated–real distribution distance in Inception feature space (lower is better) (Heusel et al., 2017); ***CLIP Score*** measures benign prompt–image alignment (higher is better). For prompt- and seed-matched outputs, ***CLIP Distance***, $d_{\mathrm{CLIP}} = 1 - \cos(e_I(x_{\mathrm{mod}}), e_I(x_{\mathrm{clean}}))$, measures semantic change. ***Pixel MSE*** is $d_{\mathrm{MSE}}(x_{\mathrm{mod}}, x_{\mathrm{clean}}) = \|x_{\mathrm{mod}} - x_{\mathrm{clean}}\|_2^2/(HWC)$, where H, W, and C denote image height, width, and color channels. Both distances are lower-is-better. Single-Adapter MSE compares each isolated implanted adapter with its clean precursor; we report both sides separately or use their maximum:

$$d_{\mathrm{single}} = \max\{d_{\mathrm{MSE}}(x_{S_{\mathrm{mod}}}, x_{S_{\mathrm{clean}}}), d_{\mathrm{MSE}}(x_{P_{\mathrm{mod}}}, x_{P_{\mathrm{clean}}})\}\,. \tag{9}$$

This measures standalone fidelity, which target inactivity alone cannot establish. ***LPIPS*** (Learned Perceptual Image Patch Similarity) measures perceptual change for the same outputs (lower is better) (Zhang et al., 2018). Joint CLIP Distance compares implanted and clean $S + P$ compositions.

Composition-specific behavior. $\Delta_{\mathrm{TS},ij}$ denotes the Target Shift of $S_i + P_j$; diagonal entries are matched and off-diagonal entries are mismatched. Lower off-pair activation indicates stronger pair selectivity. With L unrelated LoRAs, retention is

$$\mathrm{Retention}_{i,L} = 100\,\overline{\Delta}_{\mathrm{TS},i,L}/\overline{\Delta}_{\mathrm{TS},i,0}\ (\%). \tag{10}$$

Retention below/above 100% indicates attenuation/amplification relative to the default.

5.2 Effectiveness and Standalone Preservation

Table 1: Comparison of attack effectiveness, functionality preservation, and model impact. Each ASR value is evaluated on $n = 3000$ generated images for the corresponding method and state. For slash-separated S/P entries, each value independently uses $n = 3000$ images.

Method	Attack Effectiveness		Functionality Preservation			Impact on Base Model	
	ASR (%) ↑	SMI ↑	FID ↓	CLIP Score ↑	LPIPS ↓	Params	Non-invasive
SDXL	0.0	—	—	34.26	—	2.57×10^9	—
BadT2I (Zhai et al., 2023) (SDXL)	54.1	1.03	17.14	27.81	0.19	2.57×10^9	×
Personalization (Huang et al., 2024) (SDXL)	89.6	1.09	20.47	31.22	0.15	7.68×10^8	×
EvilEdit (Wang et al., 2024) (SDXL)	73.2	1.15	15.82	30.90	0.15	2.57×10^9	×
MasqLoRA (Lyu et al., 2026) (SDXL)	90.3	1.12	16.79	32.01	0.12	2.10×10^8	√
Benign LoRA S/P (SDXL)	0.0/0.0	—	—	33.52/33.40	—	$5.80 \times 10^6/5.80 \times 10^6$	√
Benign LoRA S/P (SD v1.5)	0.0/0.0	—	—	30.51/30.39	—	$3.78 \times 10^6/3.78 \times 10^6$	√
Poisoned LoRA S/P (SDXL)	2.8/3.4	0.74/0.78	15.88/16.02	33.28/33.18	0.13/0.15	$6.40 \times 10^6/6.40 \times 10^6$	√
Poisoned LoRA S/P (SD v1.5)	3.8/4.6	0.95/0.99	15.84/15.96	31.12/31.00	0.18/0.20	$4.11 \times 10^6/4.11 \times 10^6$	√
LoRango (SDXL)	**98.7**	**1.19**	16.81	**34.12**	**0.12**	1.28×10^7	√
LoRango (SD v1.5)	**97.9**	**1.24**	16.14	**33.82**	**0.11**	8.22×10^6	√

Note. "Params" reports modified model parameters for full-model baselines and exported parameters for LoRA methods (total model size for SDXL); values are not directly comparable. "Non-invasive" denotes an unchanged base checkpoint. Slash-separated entries are S/P; LoRango reports their sum. The 9,000-image total per backbone comprises S-only, P-only, and matched $S + P$ states.

Table 1 shows that matched pairs reach 98.7% ASR on SDXL and 97.9% on SD v1.5, versus 2.8–4.6% for implanted standalone adapters and 0% for benign adapters. LoRango has the highest reported SDXL ASR, although the baselines use different trigger protocols. Unlike text-triggered baselines, LoRango requires no modification to the user's prompt, using adapter composition itself as the trigger. It preserves strong benign prompt alignment and competitive perceptual fidelity, although its FID is not the lowest among the evaluated methods. On SDXL, LoRango retains a CLIP Score of 34.12 versus 34.26 for the clean model, using 1.28×10^7 adapter parameters with a frozen backbone. This is more compact than full-model editing, but not inherently smaller than every single-LoRA baseline: parameter count depends on rank and injection scope.

Table 2 covers changes to subject identity (P1), global style (P2), material/appearance (P3), and scene structure (P4). Standalone pixel MSE remains at 0.002–0.006, while matched compositions achieve 98.0–100.0% ASR with positive Target Shift, supporting standalone preservation alongside pair-conditioned activation. P4 has the smallest Target Shift (0.025) despite 99.0% ASR.

Table 3 evaluates six Not Safe for Work (NSFW) categories across the four target behaviors (24,000 samples). Seventeen of 24 combinations meet or exceed the 80% ASR threshold, while seven fall below it; SMI remains above one throughout. Benign-quality metrics vary modestly across configurations, but without a clean anchor they support consistency rather than absolute preservation.

Table 2: Standalone preservation and pair-activated effectiveness across four target behaviors. Each target behavior (P1–P4) is evaluated using $n = 100$ samples.

Target behavior	CLIP dist. / pixel MSE $(S_{\text{mod}}, S_{\text{clean}})$	CLIP dist. / pixel MSE $(P_{\text{mod}}, P_{\text{clean}})$	Joint CLIP dist. $(SP_{\text{mod}}, SP_{\text{clean}})$	Target Shift $(SP_{\text{mod}}, SP_{\text{clean}})$	ASR (%) (LoRango)
Object Identity (P1)	0.013 / 0.006	0.011 / 0.003	0.21	0.085	100.0
Global Style (P2)	0.012 / 0.005	0.008 / 0.002	0.16	0.037	98.0
Material/Appearance (P3)	0.013 / 0.005	0.008 / 0.002	0.22	0.061	99.0
Structural Disruption (P4)	0.011 / 0.005	0.008 / 0.002	0.25	0.025	99.0

Note. Parentheses identify the conditions compared under identical prompts and seeds. Subscripts "mod" and "clean" denote implanted adapters and their benign precursors; S/P are loaded individually, whereas SP denotes composition.

Table 3: NSFW-category response and benign generation quality across target behaviors. Each target–category condition contains $n = 1000$ generated samples.

Target behavior	NSFW Response (ASR (%)/SMI)						Benign Quality	
	Nudity	Violence	Horror	Gore	Deformity	Self-harm	FID	CLIP Score
Object Identity (P1)	85.4/1.35	82.1/1.33	78.7/1.36	79.6/1.34	86.8/1.35	80.3/1.34	30.80	30.12
Global Style (P2)	87.2/1.36	84.6/1.34	80.9/1.37	78.8/1.35	88.1/1.36	82.5/1.35	30.12	29.82
Material/Appearance (P3)	81.6/1.34	79.8/1.33	83.5/1.35	80.4/1.36	82.7/1.34	79.1/1.35	31.46	30.44
Structural Disruption (P4)	79.7/1.33	81.5/1.34	77.9/1.32	82.2/1.35	84.0/1.33	83.6/1.37	31.21	29.65

Note. NSFW-response cells report ASR (%)/SMI; FID and CLIP Score evaluate benign generation quality.

5.3 Robustness

We evaluate LoRango's robustness to larger pair sets, unrelated adapter co-loading, checkpoint transfer, and common inference-time variations. Unless noted, defaults are independently evaluated per study; differences from Table 2 reflect evaluation-set variation.

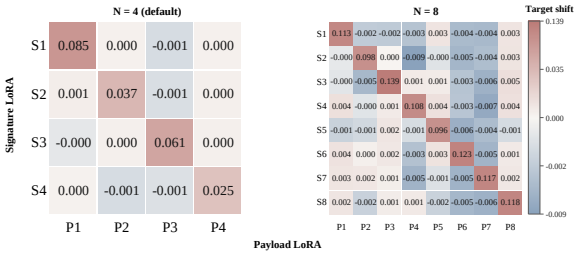

Figure 3: Pair selectivity under the $N = 4$ and $N = 8$ pair-set configurations. Each matrix cell is evaluated using $n = 100$ prompt–seed cases.

Robustness across Pair-Set Sizes. Figure 3 compares sets of $N = 4$ and $N = 8$ matched Signature–Payload pairs. For $N = 4$, diagonal Target Shift ranges from 0.025 to 0.085, with a 0.50% aggregated off-pair activation rate. For $N = 8$, diagonal Target Shift ranges from 0.096 to 0.139, with no off-pair activation observed in the evaluated cases. The stronger matched shifts at $N = 8$ come with higher standalone pixel MSE, motivating $N = 4$ as the primary configuration. Appendix D details the per-cell shifts, off-pair rates, and standalone fidelity measurements.

Table 4: Effect of additional co-loaded LoRAs on target-shift retention and ASR. Each pair is evaluated using $n = 100$ prompt–seed cases per condition.

Add. LoRAs	P1		P2		P3		P4	
	Retention (%) ↑	ASR (%) ↑	Retention (%) ↑	ASR (%) ↑	Retention (%) ↑	ASR (%) ↑	Retention (%) ↑	ASR (%) ↑
0 (Default)	100.00	100.0	100.00	98.0	100.00	99.0	100.00	99.0
1	102.02	100.0	87.71	94.0	113.38	99.0	134.88	99.0
2	102.49	99.0	78.01	91.0	98.81	96.0	124.14	97.0
4	91.76	99.0	69.09	87.0	90.37	95.0	99.38	95.0

Note. Retention is normalized to mean Target Shift with no additional LoRA ($L = 0$); values above 100% indicate shift amplification, not a success rate above 100%.

Robustness to Additional Co-loaded LoRAs. Table 4 shows that all pairs remain active with up to four unrelated LoRAs: the minimum ASR is 87.0%, although retention falls as low as 69.09%. P2 is the most sensitive to interference, while P1 retains 99.0% ASR with four additional LoRAs. Shift amplification in some settings reveals a non-monotonic response to co-loading. Thus, unrelated adapters can alter attack strength without reliably preventing activation in the tested compositions.

Table 5: Comparison between the default SDXL checkpoint and RealVisXL across P1–P4. Each checkpoint–pair condition contains $n = 100$ generated samples.

Metric	P1		P2		P3		P4	
	SDXL (Default)	RealVisXL	SDXL (Default)	RealVisXL	SDXL (Default)	RealVisXL	SDXL (Default)	RealVisXL
Target Shift ↑	0.085	0.072	0.037	0.019	0.061	0.064	0.025	0.014
Visual ASR (%) ↑	100.0	100.0	98.0	96.0	99.0	98.0	99.0	95.0
Single-Adapter MSE ↓	0.006	0.003	0.005	0.002	0.005	0.002	0.005	0.003
Result	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass

Note. SDXL (Default) uses the results in Table 2, with Single-Adapter MSE aggregated as defined in Section 5.1.

Robustness across Checkpoints. Direct transfer to RealVisXL without retraining yields 95.0–100.0% ASR and lower Single-Adapter MSE than SDXL (Table 5). Target Shift increases slightly for P3 but decreases for P1, P2, and P4. For example, P4's shift falls from 0.025 to 0.014 while ASR remains at 95.0%, indicating that success frequency transfers more consistently than effect magnitude. This result concerns transfer between compatible checkpoints; it does not establish unchanged-weight transfer across different denoiser architectures.

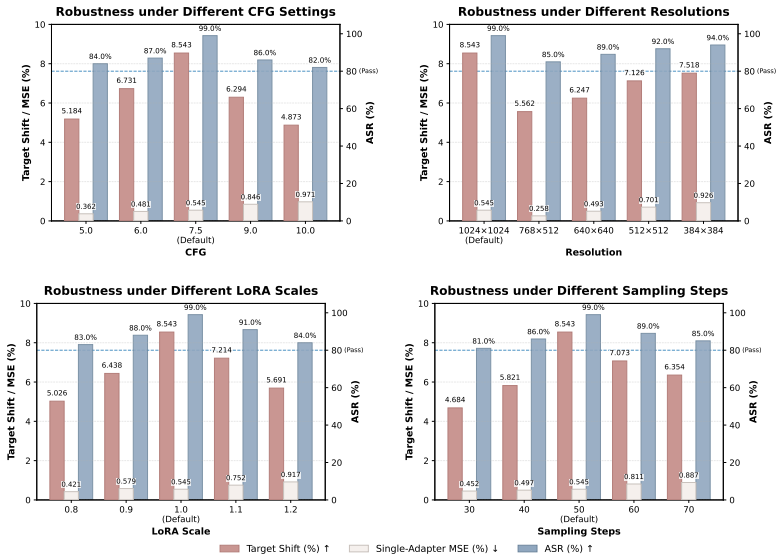

Figure 4: Robustness to classifier-free guidance (CFG), resolution, LoRA scale, and sampling steps.

Robustness to Inference Parameters. Each setting in Figure 4 uses $n = 100$ prompt–seed cases. Target Shift and Single-Adapter MSE are shown in percentage units; "Default" denotes each metric under the default inference configuration. All settings meet the 80% ASR criterion (81.0–99.0%), with the weakest case at 30 sampling steps. Higher guidance and larger LoRA scales increase Single-Adapter MSE, exposing an effectiveness–fidelity trade-off.

5.4 Mechanism Analysis and Component Ablation

We analyze the components supporting attack effectiveness and standalone fidelity in two steps: first varying the number of carrier layers K, then fixing $K = 8$ and ablating *Carrier-Level Cancellation* and *Output-Level Cancellation* individually and jointly.

Carrier-Capacity Sensitivity. Table 6 shows ASR below 80% for $K \leq 4$, versus 96.0% and 99.0% for $K = 6$ and 8. Near-zero Target Shift at $K = 1$ and 2 suggests insufficient capacity.

Table 6: Sensitivity to carrier capacity, measured by the number of carrier layers K. All settings use the same fixed pair-conditioned target objective and the same $n = 100$ prompt–seed cases.

Setting	Target Shift ↑	Single-Adapter MSE ↓	ASR (%) ↑
$K = 1$	0.001	0.003	46.0
$K = 2$	0.001	0.003	50.0
$K = 4$	0.027	0.004	70.0
$K = 6$	0.065	0.005	96.0
$K = 8$ (Default)	0.085	0.005	99.0

Note. K counts carrier layers, not LoRA rank or per-layer lanes (Section 4.3). The default $K = 8$ applies to this sweep, not all P1–P4 configurations.

Raising K from 6 to 8 increases Target Shift from 0.065 to 0.085; both MSE values round to 0.005. Across the sweep, MSE rises from 0.003 to 0.005, indicating a modest fidelity cost.

Table 7: Ablation of the cancellation mechanisms. All conditions use the same fixed pair-conditioned target objective and are evaluated using $n = 100$ prompt–seed cases per condition (400 cases total).

Setting	Target Shift ↑	Single-Adapter MSE ↓	ASR (%) ↑
w/o Carrier-Level Cancellation	0.027	0.005	66.0
w/o Output-Level Cancellation	0.063	0.071	97.0
w/o Carrier and Output Cancellation	0.076	0.114	51.0
With Carrier and Output Cancellation (Default)	0.085	0.005	99.0

Note. All conditions use $K = 8$; Default enables both Carrier-Level and Output-Level Cancellation.

Cancellation Ablation. Table 7 distinguishes the two mechanisms: removing Carrier-Level Cancellation reduces ASR from 99.0% to 66.0% without changing the reported MSE, whereas removing Output-Level Cancellation retains 97.0% ASR but raises MSE from 0.005 to 0.071. The latter therefore contributes substantially to standalone fidelity. Removing both yields the highest MSE (0.114) and lowest ASR (51.0%), despite a larger Target Shift than removing Output-Level Cancellation alone. This separation can arise because mean semantic displacement and binary success frequency need not vary monotonically. Both mechanisms together provide the best effectiveness–preservation balance in this experiment.

5.5 Cross-Architecture Applicability

Table 8: Applicability across diffusion model families and denoiser architectures. All methods are evaluated under the same fixed target objective with $n = 20$ generated samples per condition.

Method	Model	Denoiser	Target Margin	ASR (%)	Result
MasqLoRA	SDXL	U-Net	0.0990	90.0	Pass
LoRango	SDXL	U-Net	0.0222	**100.0**	**Pass**
MasqLoRA	SD3	MMDiT	0.0024	20.0	Fail
LoRango	SD3	MMDiT	**0.0173**	**95.0**	**Pass**
MasqLoRA	FLUX.1-schnell	Flow Transformer	0.0063	25.0	Fail
LoRango	FLUX.1-schnell	Flow Transformer	**0.0346**	**80.0**	**Pass**

Note. Result labels use the 80% ASR threshold defined in Section 5.1.

All adapters are retrained separately for each architecture; Table 8 therefore evaluates whether LoRango's pair-conditioned mechanism remains effective across denoiser families, rather than whether one set of weights transfers unchanged. Under this architecture-specific training protocol, LoRango attains 100%, 95%, and 80% ASR on SDXL, SD3, and FLUX.1-schnell, respectively, compared with 90%, 20%, and 25% for MasqLoRA. These results show that LoRango remains effective beyond U-Net-based diffusion models, extending to MMDiT and Flow Transformer denoisers.

6 Conclusion

LoRango splits pair-conditioned hidden behavior across Signature and Payload LoRAs, achieving high matched-pair ASR with low standalone activation and preserved benign quality. Its robustness across inference settings and applicability to SD3 and FLUX.1-schnell expose a compositional supply-chain risk missed by standalone adapter inspection.

AI USE STATEMENT

Generative AI tools were used to polish the manuscript, including improving English grammar, clarity, concision, organization, and LaTeX formatting. Gemini 2.5 Pro was used as a visual evaluator to classify generated outputs when computing ASR. The authors reviewed all AI-assisted revisions, checked the reported values against the experimental records and tables, and take responsibility for the final content, claims, citations, and artifacts.

ETHICS STATEMENT

This work studies a dual-use security risk in open LoRA ecosystems. Describing pair-conditioned backdoors may inform misuse, but understanding this threat is necessary for composition-aware auditing, provenance controls, and safer adapter deployment. Our experiments use generated images and do not involve human subjects or private user data. We present the attack to characterize the risk and support defensive research, and recommend that any release of implementation artifacts follow responsible disclosure and access-control practices.

REPRODUCIBILITY STATEMENT

Section 4.4 reports the principal model, parameterization, and optimization settings, while the experimental section specifies metrics, baselines, sample sizes, and evaluation protocols. Appendix A provides the interaction derivations, Appendix C records module-level ranks, schedules, and export coefficients, and Appendix B defines the auxiliary training objectives. Together, these sections provide the information needed to reconstruct the method and audit the reported results.

REFERENCES

Weixin Chen, Dawn Song, and Bo Li. TrojDiff: Trojan attacks on diffusion models with diverse targets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4035–4044, 2023a.

Yida Chen, Fernanda Viégas, and Martin Wattenberg. Beyond surface statistics: Scene representations in a latent diffusion model. *arXiv preprint arXiv:2306.05720*, 2023b.

Minkyoung Cho, Ruben Ohana, Christian Jacobsen, Adityan Jothi, Min-Hung Chen, Z. Morley Mao, and Ethem Can. TC-LoRA: Temporally modulated conditional LoRA for adaptive diffusion control. *arXiv preprint arXiv:2510.09561*, 2025. URL https://arxiv.org/abs/2510.09561.

Sheng-Yen Chou, Pin-Yu Chen, and Tsung-Yi Ho. How to backdoor diffusion models? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4015–4024, 2023a.

Sheng-Yen Chou, Pin-Yu Chen, and Tsung-Yi Ho. VillanDiffusion: A unified backdoor attack framework for diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, 2023b.

Kevin Clark and Priyank Jaini. Text-to-image diffusion models are zero-shot classifiers. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Sihao Ding. Colluding lora: A compositional vulnerability in llm safety alignment. *arXiv preprint arXiv:2603.12681*, 2026. URL https://arxiv.org/abs/2603.12681.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *International Conference on Learning Representations (ICLR)*, 2023.

Gemini Team. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. Technical Report arXiv:2507.06261, Google, 2025. URL https://arxiv.org/abs/2507.06261.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. BadNets: Identifying vulnerabilities in the machine learning model supply chain. In *Neural Information Processing Systems Workshop on Machine Learning and Computer Security*, 2017.

Yuchao Gu, Xintao Wang, Jay Zhangjie Wu, Yujun Shi, Yunpeng Chen, Zihan Fan, Wuyou Xiao, Rui Zhao, Shuning Chang, Weijia Wu, Yixiao Ge, Ying Shan, and Mike Zheng Shou. Mix-of-Show: Decentralized low-rank adaptation for multi-concept customization of diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/8a1d694707eb0fefe65871369074926d-Abstract.html.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 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. In *International Conference on Learning Representations (ICLR)*, 2022.

Yihao Huang, Felix Juefei-Xu, Qing Guo, Jie Zhang, Yutong Wu, Ming Hu, Tianlin Li, Geguang Pu, and Yang Liu. Personalization as a shortcut for few-shot backdoor attack against text-to-image diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 21169–21178, 2024. doi: 10.1609/aaai.v38i19.30110.

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 (CVPR)*, pp. 1931–1941, 2023.

Zhiwen Li, Zhongjie Duan, Die Chen, Cen Chen, Daoyuan Chen, Yaliang Li, and Yingda Chen. AutoLoRA: Automatic LoRA retrieval and fine-grained gated fusion for text-to-image generation. *arXiv preprint arXiv:2508.02107*, 2025.

Weilin Lin, Nanjun Zhou, Yanyun Wang, Jianze Li, Hui Xiong, and Li Liu. BackdoorDM: A comprehensive benchmark for backdoor learning on diffusion model. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *Network and Distributed System Security Symposium (NDSS)*, 2018.

Grace Luo, Lisa Dunlap, Dong Huk Park, Aleksander Holynski, and Trevor Darrell. Diffusion hyperfeatures: Searching through time and space for semantic correspondence. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Liangwei Lyu, Jiaqi Xu, Jianwei Ding, and Qiyao Deng. When LoRA betrays: Backdooring text-to-image models by masquerading as benign adapters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8577–8586, 2026.

Tuna Han Salih Meral, Enis Simsar, Federico Tombari, and Pinar Yanardag. Contrastive test-time composition of multiple LoRA models for image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18090–18100, 2025.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8162–8171, 2021.

Ryan Po, Guandao Yang, Kfir Aberman, and Gordon Wetzstein. Orthogonal adaptation for modular customization of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7964–7973, 2024.

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. In *International Conference on Learning Representations (ICLR)*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

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 (CVPR)*, pp. 10684–10695, 2022.

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 (CVPR)*, pp. 22500–22510, 2023.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Raphael Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Enis Simsar, Thomas Hofmann, Federico Tombari, and Pinar Yanardag. LoRACLR: Contrastive adaptation for customization of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13189–13198, 2025.

Vera Soboleva, Aibek Alanov, Andrey Kuznetsov, and Konstantin Sobolev. T-LoRA: Single image diffusion model customization without overfitting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 9051–9059, 2026. doi: 10.1609/aaai.v40i11.37861.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021.

Lukas Struppek, Dominik Hintersdorf, and Kristian Kersting. Rickrolling the artist: Injecting backdoors into text encoders for text-to-image synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4584–4596, 2023.

Jordan Vice, Naveed Akhtar, Richard Hartley, and Ajmal Mian. BAGM: A backdoor attack for manipulating text-to-image generative models. *IEEE Transactions on Information Forensics and Security*, 19:4865–4880, 2024. doi: 10.1109/TIFS.2024.3386058.

Hao Wang, Shangwei Guo, Jialing He, Kangjie Chen, Shudong Zhang, Tianwei Zhang, and Tao Xiang. EvilEdit: Backdooring text-to-image diffusion models in one second. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 3657–3665, 2024. doi: 10.1145/3664647.3680689.

Xingyi Yang and Xinchao Wang. Diffusion model as representation learner. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18938–18949, 2023.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. IP-Adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. doi: 10.48550/arXiv.2308.06721.

Shengfang Zhai, Yinpeng Dong, Qingni Shen, Shi Pu, Yuejian Fang, and Hang Su. Text-to-image diffusion models can be easily backdoored through multimodal data poisoning. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 1577–1587, 2023. doi: 10.1145/3581783.3612108.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations (ICLR)*, 2023.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_The_Unreasonable_Effectiveness_CVPR_2018_paper.html.

Ming Zhong, Yelong Shen, Shuohang Wang, Yadong Lu, Yizhu Jiao, Siru Ouyang, Donghan Yu, Jiawei Han, and Weizhu Chen. Multi-LoRA composition for image generation. *Transactions on Machine Learning Research*, 2024.

A Interaction Derivations

For a generated output mapped by Ψ to the common action space, the behavior attributable specifically to composition is

$$\begin{aligned} I_{ij}^{\text{out}}(x,\xi) = y_{S_i+P_j}(x,\xi) - y_{S_i}(x,\xi) \\ - y_{P_j}(x,\xi) + y_{\varnothing}(x,\xi), \qquad y_A = \Psi(F_A(x;\xi)). \end{aligned} \tag{11}$$

Using the same prompt and seed, the corresponding feature interaction at layer ℓ and timestep t is

$$I_{ij}^{(\ell,t)} = h_{S_i+P_j}^{(\ell,t)} - h_{S_i}^{(\ell,t)} - h_{P_j}^{(\ell,t)} + h_{\varnothing}^{(\ell,t)}. \tag{12}$$

A purely additive composition makes these differences vanish. Training instead aligns diagonal interactions with the pair action and suppresses off-diagonal interactions.

The stable scalar interface follows from the affine LayerNorm preceding the multiplicative block. For $z_t^{(\ell)} = \gamma^{(\ell)} \odot \widehat{h}_t^{(\ell)} + \beta^{(\ell)}$ and $\mathbf{1}^\top \widehat{h}_t^{(\ell)} = 0$, choosing $v^{(\ell)} = \kappa {\gamma^{(\ell)}}^{-1}$ gives

$$b_t^{(\ell)} = {v^{(\ell)}}^\top z_t^{(\ell)} = \kappa \mathbf{1}^\top \widehat{h}_t^{(\ell)} + {v^{(\ell)}}^\top \beta^{(\ell)} = b_0^{(\ell)}. \tag{13}$$

This bias-free projection need only keep ordinary variation within the cancellation region; unstable carrier coordinates are rejected during calibration.

Finally, if s_i and p_j are the local Signature and Payload perturbations and Φ is the native block transformation, their nonlinear cross-effect is

$$C_{ij}(z) = \Phi(z + s_i + p_j) - \Phi(z + s_i) - \Phi(z + p_j) + \Phi(z). \tag{14}$$

It vanishes for a linear block and can be nonzero in GEGLU because the Signature gate displacement interacts multiplicatively with the Payload value/output path. Equation 6 summarizes a local path under a shared-value approximation, absorbing the value feature and its output projection into the effective vector $u_j^{(\ell)}$. That vector need not be constant across inputs or timesteps; the equation is not an exact expression for the full-network output.

B Auxiliary Training Objectives

This appendix expands the training discussion in Section 4.3, distinguishing the implemented refinement losses from schematic auxiliary design objectives. Expectations are over the relevant prompts, seeds, layers, and timesteps.

Implemented refinement losses. Action refinement uses normalized mean squared error (NMSE), masked NMSE, and a direction-alignment loss. The first two penalize prediction discrepancies globally and over masked entries, respectively; the last encourages alignment with the target direction. Not all conceptual objectives below are optimized in every stage; in particular, $\lambda_{\text{off}} = 0$ for P3/P4. Appendix C reports the stage-specific losses, weights, and schedules. These training losses are distinct from the image-level Single-Adapter MSE used for evaluation.

Conceptual objective decomposition. The intended design constraints over prompts, seeds, compositions, and timesteps $\mathcal{T}$ can be summarized as

$$\begin{aligned}\mathcal{L} = \lambda_{\text{diag}}\mathcal{L}_{\text{diag}} + \lambda_{\text{off}}\mathcal{L}_{\text{off}} + \lambda_{\text{single}}\mathcal{L}_{\text{single}} \\ + \lambda_{\text{util}}\mathcal{L}_{\text{util}} + \lambda_{\text{traj}}\mathcal{L}_{\text{traj}} + \lambda_{\text{cancel}}\mathcal{L}_{\text{cancel}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}.\end{aligned} \tag{15}$$

Here $\mathcal{L}_{\text{diag}}$ and $\mathcal{L}_{\text{off}}$ describe matched-action alignment and mismatch suppression. The remaining five terms describe standalone inactivity, utility preservation, trajectory consistency, cancellation, and regularization, respectively. This decomposition organizes the design goals rather than specifying a single seven-term loss jointly optimized in every reported run. The schematic formulations below therefore should not be read as five additional losses necessarily enabled during training.

Single-adapter inactivity. Let a_i be a nonnegative differentiable surrogate for the programmed action r_i. We suppress that action when either adapter is loaded alone:

$$\mathcal{L}_{\text{single}} = \mathbb{E}_{x,\xi}\left[\sum_{i=1}^{N}\Big(a_i(F_{S_i}(x;\xi)) + a_i(F_{P_i}(x;\xi))\Big)\right]. \tag{16}$$

This term prevents the Signature from directly carrying the action and the Payload from exposing it without the matched codeword.

Benign-utility preservation. Each modified adapter is constrained to reproduce the behavior of its benign precursor:

$$\begin{aligned}\mathcal{L}_{\text{util}} = \mathbb{E}_{x,\xi}\Bigg[\sum_{i=1}^{N}\Big(d_u\Big(F_{S_i}(x;\xi), F_{S_i^{\text{ben}}}(x;\xi)\Big) \\ + d_u\Big(F_{P_i}(x;\xi), F_{P_i^{\text{ben}}}(x;\xi)\Big)\Big)\Bigg],\end{aligned} \tag{17}$$

where d_u may combine pixel, perceptual, and representation distances according to the advertised utility of each adapter.

Trajectory consistency. We align matched interactions with layer- and timestep-specific training directions over the selected carriers and denoising timesteps:

$$\mathcal{L}_{\text{traj}} = \mathbb{E}_{x,\xi}\left[\sum_{i=1}^{N}\sum_{t\in\mathcal{T}}\sum_{\ell\in\mathcal{K}} d_{\text{traj}}\Big(I_{ii}^{(\ell,t)}, d_i^{(\ell,t)}\Big)\right]. \tag{18}$$

The direction $d_i^{(\ell,t)}$ may be obtained from a teacher, a target residual, or a differentiable semantic objective; it is used only during training.

Differential cancellation. For payload P_j, let $\pi_{\text{off},j}$ sample the non-matched states $\{P_j\} \cup \{S_i + P_j : i \neq j\}$, and let h_A denote the activation under sampled state A. We preserve cancellation between its signal and reference branches in those states:

$$\mathcal{L}_{\text{cancel}} = \sum_{j=1}^{N}\mathbb{E}_{x,\xi,A\sim\pi_{\text{off},j}}\left[\sum_{t\in\mathcal{T}}\sum_{\ell\in\mathcal{K}}\Big\|R_{P_j,+}^{(\ell,t)}(h_A) + R_{P_j,-}^{(\ell,t)}(h_A)\Big\|_2^2\right]. \tag{19}$$

This objective does not constrain the matched state, where the Signature is expected to break cancellation and release the action.

Regularization. For clarity, we decompose the regularizer as

$$\mathcal{L}_{\text{reg}} = \mu_\theta\mathcal{R}_\theta + \mu_\alpha\mathcal{R}_\alpha + \mu_h\mathcal{R}_{\text{drift}} + \mu_R\mathcal{R}_{\text{energy}}, \tag{20}$$

where the four terms represent penalties on implanted LoRA weights, Signature amplitude, ordinary-state activation drift, and carrier residual energy, respectively. This decomposition describes possible regularization roles; it does not specify four separately enabled penalties in the reported runs.

C Reproducibility Details

Modules and ranks. The benign LoRAs target `to_q`, `to_k`, `to_v`, and `to_out.0`; the last name denotes the linear attention-output projection in Diffusers. Carrier updates target `ff.net.0.proj`, the joint GEGLU gate/value input projection, and `ff.net.2`, its output projection. Payload projections use rank 4. The Signature uses rank 5 at `ff.net.0.proj` for an additional constant-gate dimension and rank 4 at `ff.net.2`; the P1 `mid_t7` carrier uses ranks 4/3 because it has three lanes.

Schedules. Benign LoRAs are trained for 1,800 AdamW steps at 5×10^{-5} and refined for 1,200 steps at 2×10^{-5}. P1/P2 use a staged warm start totaling 10,100 updates; their final 2,500-step joint refinement uses 3×10^{-4} for the action and 2×10^{-4} for the reader. P3 uses 3,000 initial action steps at 1.5×10^{-3} and 1,400 refinement steps at 10^{-3}; P4 uses 1,400 refinement steps at 10^{-3}. Action refinement minimizes NMSE plus 0.5 masked NMSE and 0.5 direction loss. AdamW uses $(\beta_1, \beta_2) = (0.9, 0.99)$, weight decay 10^{-4}, and gradient clipping of 1.0 for utility parameters and 2.0 for action/reader parameters; P3/P4 use $\lambda_{\text{off}} = 0$.

Export coefficients. For P1–P4, respectively, gate coefficients are $\{29, 35, 29, 46\}$, reader coefficients are $\{0.08, 0.06, 0.08, 0.05\}$, action coefficients are $\{0.33, 0.45, 0.48, 0.462\}$, and value-cancellation coefficients are $\{0.85, 0.90, 0.85, 0.85\}$.

D Per-Cell Pairing Metrics

This appendix expands the pair-selectivity results in Figure 3. Each cell $S_i + P_j$ is evaluated on the same $n = 100$ prompt–seed cases. Diagonal cells ($i = j$) are the intended matched pairs; off-diagonal cells ($i \neq j$) are mismatched pairs that should remain inactive.

Standalone fidelity. For each Signature or Payload, CLIP distance measures semantic deviation between the implanted adapter and its clean precursor, while pixel MSE measures their image-level deviation under the same prompt and seed. Lower values indicate better preservation of the adapter's advertised standalone behavior. Because these quantities depend only on the individual adapter, Table 9 places each intended Signature–Payload pair on one row so that the two adapters can be compared directly without repeating their standalone values across mismatched compositions.

Pair-conditioned behavior. Table 10 reports Target Shift and off-pair activation rate for every Signature–Payload composition under the $N = 4$ and $N = 8$ configurations. Target Shift measures movement of the composed output toward the designated target. A large positive diagonal value indicates effective activation by the intended pair, whereas an off-diagonal value near zero indicates that a mismatched pair remains suppressed. The *off-pair activation rate* is the percentage of the 100 cases in a mismatched cell that nevertheless activate the target—that is, the empirical probability of an unintended pairing success. Lower is better. It is undefined for diagonal cells, which are intended to activate and are therefore marked by an em dash. The aggregate rate pools all off-diagonal cases: 1,200 cases for $N = 4$ and 5,600 cases for $N = 8$.

Table 9: Standalone fidelity of the intended Signature–Payload pairs used in Figure 3. Each row compares the two members of one matched pair with their respective clean precursors; CLIP distance and pixel MSE are both lower-is-better.

	Signature-only $(S_i^{\text{mod}}, S_i^{\text{clean}})$		Payload-only $(P_i^{\text{mod}}, P_i^{\text{clean}})$	
Pair ID i	CLIP dist. ↓	Pixel MSE ↓	CLIP dist. ↓	Pixel MSE ↓
Pair-set size $\mathbf{N = 4}$				
1	0.013	0.006	0.011	0.003
2	0.012	0.005	0.008	0.002
3	0.013	0.005	0.008	0.002
4	0.011	0.005	0.008	0.002
Pair-set size $\mathbf{N = 8}$				
1	0.017	0.011	0.015	0.005
2	0.019	0.012	0.019	0.006
3	0.019	0.011	0.016	0.005
4	0.015	0.010	0.014	0.005
5	0.017	0.011	0.012	0.004
6	0.015	0.010	0.013	0.004
7	0.018	0.011	0.013	0.005
8	0.015	0.009	0.010	0.004

Note. The Signature columns report CLIP distance and pixel MSE between outputs generated with S_i^{mod} alone and with S_i^{clean} alone. The Payload columns report the same metrics for P_i^{mod} alone versus P_i^{clean} alone, using the same prompt and seed. Each row groups the two members of an intended pair; these are standalone-adapter measurements, not measurements of their co-loaded output.

Table 10: Pair-conditioned metrics for every Signature–Payload composition. For each Signature, the first row gives Target Shift and the second gives off-pair activation rate. Diagonal entries are intended matched pairs, so their off-pair rate is not applicable and is marked by an em dash.

Pair-set size $\mathbf{N = 4}$

Signature	Metric	P_1	P_2	P_3	P_4
S_1	Shift	0.085	0.000	−0.001	0.000
	Off (%)	—	1.0	0.0	0.0
S_2	Shift	0.001	0.037	−0.001	0.000
	Off (%)	3.0	—	0.0	0.0
S_3	Shift	0.000	0.000	0.061	0.000
	Off (%)	1.0	0.0	—	0.0
S_4	Shift	0.000	−0.001	−0.001	0.025
	Off (%)	1.0	0.0	0.0	—

Pair-set size $\mathbf{N = 8}$

Signature	Metric	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
S_1	Shift	0.113	−0.002	−0.002	−0.003	0.003	−0.004	−0.004	0.003
	Off (%)	—	0.0	0.0	0.0	0.0	0.0	0.0	0.0
S_2	Shift	0.000	0.098	0.000	−0.009	0.000	−0.005	−0.004	0.003
	Off (%)	0.0	—	0.0	0.0	0.0	0.0	0.0	0.0
S_3	Shift	0.000	−0.005	0.139	0.001	0.001	−0.003	−0.006	0.005
	Off (%)	0.0	0.0	—	0.0	0.0	0.0	0.0	0.0
S_4	Shift	0.004	0.000	0.001	0.108	0.004	−0.003	−0.007	0.004
	Off (%)	0.0	0.0	0.0	—	0.0	0.0	0.0	0.0
S_5	Shift	−0.001	−0.001	0.002	−0.001	0.096	−0.006	−0.004	−0.001
	Off (%)	0.0	0.0	0.0	0.0	—	0.0	0.0	0.0
S_6	Shift	0.004	0.000	0.002	−0.003	0.003	0.123	−0.005	0.001
	Off (%)	0.0	0.0	0.0	0.0	0.0	—	0.0	0.0
S_7	Shift	0.003	0.002	0.001	−0.005	−0.001	−0.005	0.117	0.002
	Off (%)	0.0	0.0	0.0	0.0	0.0	0.0	—	0.0
S_8	Shift	0.002	−0.002	0.001	0.001	−0.002	−0.005	−0.006	0.118
	Off (%)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	—