CHAMELEON: STYLE-CONTENT DISENTANGLED FRAMEWORK FOR CROSS-DOMAIN OBJECT COMPOSITING

Anonymous authors
Paper under double-blind review

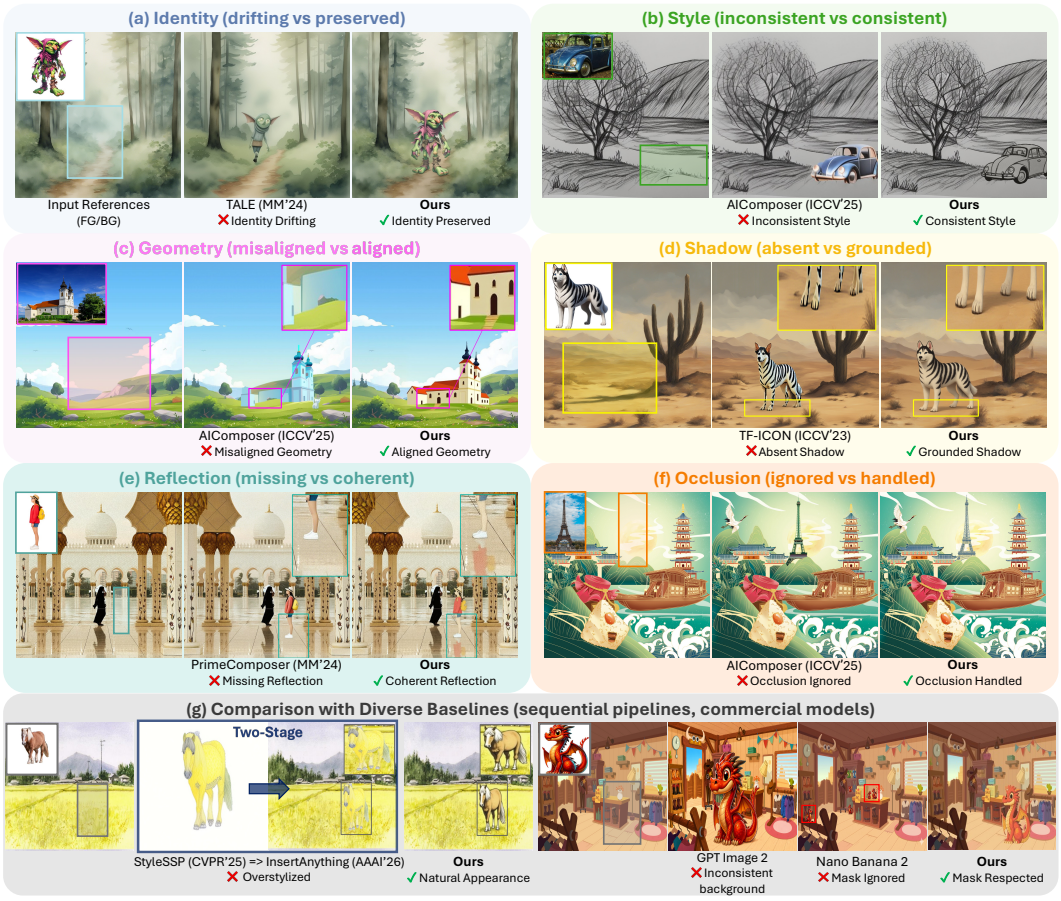


Figure 1: **Cross-domain compositing results by our Chameleon:** (a) shows that Chameleon preserves foreground identity, while (b) maintains consistent style. (c) and (d) demonstrate compositional plausibility via aligned geometry and grounded shadows, respectively. (e) demonstrates scene-aware compositing via coherent reflections, while (f) handles foreground-background occlusion relationships. (g) shows that Chameleon outperforms two-stage cascaded pipelines that combine style transfer and object insertion, which tend to over-stylize the foreground, as well as commercial models (GPT-Image-2 (OpenAI, 2026), Nano-Banana-2 (Google DeepMind, 2026)) that shift background tone or ignore the input mask.

ABSTRACT

Image compositing aims to seamlessly insert a foreground object into a background image, and recent advances in diffusion models have significantly enhanced the quality, especially when the foreground and background images come from the *same domain* (e.g., natural images). However, *cross-domain* compositing, where the foreground and background come from different domains, is relatively underexplored and remains challenging because the model must preserve the foreground object's identity while stylizing it to match the background domain. Existing cross-domain compositing approaches largely rely on training-free blend-

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.

ing and refinement strategies. This is partly due to the lack of large-scale paired datasets for cross-domain compositing, limiting the development of training-based solutions. As a result, they are limited to tone-level alignment and often produce style-inconsistent or overstylized results. To overcome such limitations, we construct ChameleonDataset, the first large-scale training dataset for cross-domain compositing, with a comprehensive evaluation benchmark, built through a scalable data construction pipeline. Building on this, we propose Chameleon, a novel two-stage training-based cross-domain compositing framework. In the first stage, we propose Signed Style-Content Graph Learning (SSCGL) to train ChameleonEncoder, which effectively disentangles style and content representations. In the second stage, we introduce Style Gap-Aware Anisotropic Modulation (SGAM) into a diffusion transformer for effective cross-domain compositing, adaptively regulating how style tokens from the first-stage ChameleonEncoder are injected according to the foreground-background style gap, spatial region, and diffusion timestep. Our method outperforms state-of-the-art in-domain and cross-domain compositing models, sequential pipelines and commercial models, achieving improvements in both compositional plausibility and stylistic fidelity.

1 INTRODUCTION

Image compositing (Song et al., 2023; Canet Tarrés et al., 2024) aims to place a reference object(s) in a foreground image (i.e., reference image) at a desired location within a background image (i.e., target image) such that the inserted object appears natural within the background canvas. With recent advances in diffusion models (Ho et al., 2020; Song et al., 2021), image compositing (i.e., generative object compositing) has significantly improved by re-synthesizing the composite region through learned generative priors (Chen et al., 2024; Huang et al., 2025), rather than merely adjusting pixel values near the boundary (Pérez et al., 2003). This enables more natural integration in terms of appearance, geometry, and semantics, allowing the foreground to better align with the background.

These advances have made image compositing practical for real-world applications such as digital content creation and advertising, where the foreground and background typically come from the same photorealistic visual domain. However, another line of use cases is in creative ideation, where a user may wish to insert an asset coming from a different domain. For instance, a user may have a specific car photograph in mind that they imagine inserting into a rough first sketch, as shown in Fig. 1 (b). Rather than needing to manually draw the car in a similar style, cross-domain compositing can help directly transform this car into a similar sketch style while inserting it to the desired background, making the ideation process more efficient and creative by expanding the pool of object and background assets to *any* stylistic domain.

Nevertheless, cross-domain compositing is a challenging research problem that involves multiple requirements, as shown in Fig. 1: (a) preserving the identity of the foreground reference, (b) adapting its appearance to the style of the background, (c) maintaining geometric and contextual plausibility within the scene, and (d)-(f) handling shadows, reflections, and occlusions for visual coherence.

Among prior work in cross-domain compositing, early methods such as TF-ICON (Lu et al., 2023b) apply DDIM inversion (Mokady et al., 2023) and integrate foreground and background via latent-space blending and attention injection. TALE (Pham et al., 2024) leverages intermediate latents instead of pure noise, improving structural and stylistic preservation. More recent methods such as AIComposer (Li et al., 2025) instead perform pixel-space blending and leverage CLIP features (Ye et al., 2023) for foregroundbackground fusion.

Despite these attempts, existing methods fall short of fully solving this challenging problem, as they share a common paradigm that freezes a pre-trained text-to-image (T2I) diffusion model (Rombach et al., 2022) and performs compositing solely through blending-style refinement, leading to several limitations. First, these methods depend on user-provided prompts to drive a frozen T2I diffusion backbone, yet such prompts are difficult to craft and cannot fully match the foreground, leading to *identity drift* (Fig. 1 (a)), where the prompt "a goblin" fails to capture the specific identity of the foreground. Second, these blending-based methods apply statistical matching such as AdaIN (Huang & Belongie, 2017) to align the foreground with the background distribution, yet this approach is limited to tone-level alignment and cannot bridge large domain gaps, resulting in *style inconsistency* (Fig. 1 (b)). Third, these methods lack training for cross-domain compositing, resulting in *geometry*

Table 1: Comparison with existing stylization and compositing datasets. Ours is the only dataset supporting both tasks and view augmentation with real-image supervision (200K samples).

Dataset	Stylization	Compositing	View Augmentation	Supervision	#Samples
OmniConsistency	✓	×	×	Synthetic	2.6K
OmniStyle	✓	×	×	Synthetic	150K
DreamFuse	×	✓	×	Synthetic	84K
AnyInsertion	×	✓	△	Real	159K
Ours	✓	✓	✓	**Real**	**200K**

misalignment and *absence of shadows* (Fig. 1 (c,d)). These limitations stem from the lack of suitable training data for cross-domain compositing.

To address these limitations, we construct ChameleonDataset$_{\text{tr}}$, the first large-scale training set for cross-domain compositing. Unlike prior datasets that rely on synthetic-image supervision (Tab. 1), we adopt a reverse pipeline driven by real-image supervision. This mitigates artifacts in stylized generations induced in the prior datasets (Sec. A.2), and exposes the model to diverse compositional scenarios, enabling natural geometry alignment and shadow grounding. We further introduce ChameleonDataset$_{\text{ev}}$ that evaluates both compositional plausibility and stylistic fidelity under diverse real-world scenarios.

With training data in place, effective cross-domain compositing further requires disentangled content and style representations to preserve foreground identity while adapting to the background style. Inspired by signed graph learning (Derr et al., 2018; Huang et al., 2021), we propose ChameleonEncoder, a style-content disentangled encoder trained with our novel Signed Style-Content Graph Learning (SSCGL). Unlike prior signed graph representation learning methods (Derr et al., 2018; Huang et al., 2021), SSCGL leverages edge signs for factor disentanglement by assigning opposite signs to the same image pair across the style and content representation spaces. By structuring DINOv3 (Siméoni et al., 2025) features with these signed relations, SSCGL yields disentangled style and content tokens. These tokens are then injected into the diffusion transformer (DiT), while our novel Style Gap-Aware Anisotropic Modulation (SGAM) adaptively regulates style injection according to the foreground-background style gap, spatial region, and diffusion timestep for effective cross-domain compositing. In summary, our contributions are as follows:

- We construct ChameleonDataset$_{\text{tr}}$, the first large-scale dataset for cross-domain image compositing, built via a reverse data generation pipeline that can be applied on any images. It provides real-image supervision rather than synthetic-image supervision, mitigating generative artifacts and enabling faithful stylization.
- We introduce ChameleonDataset$_{\text{ev}}$, a comprehensive benchmark evaluating compositional plausibility (grounding, reflection) and stylistic fidelity (pixel art, stylized-to-stylized).
- We propose Chameleon, a novel two-stage training-based cross-domain compositing framework consisting of (i) ChameleonEncoder, a style-content disentangled encoder trained with Signed Style-Content Graph Learning (SSCGL), which structures DINOv3 features with signed graph relations to achieve the disentanglement essential for identity preservation and stylization, and (ii) Style Gap-Aware Anisotropic Modulation (SGAM), which adaptively regulates style injection into a DiT for effective cross-domain compositing.

2 Related Work

Image compositing. Naturally inserting a foreground image into a background image has long been a challenging problem in image editing (Brooks et al., 2023; Meng et al., 2022). Classical techniques (Pérez et al., 2003; Porter & Duff, 1984) improved visual quality but remain limited to pixel-level operations, lacking semantic awareness. Recently, diffusion models have significantly improved object insertion by leveraging priors from large-scale data. For example, Paint-by-Example (Yang et al., 2023) fills masked regions using a reference image but fails to preserve object identity, which AnyDoor (Chen et al., 2024) mitigates via an ID extractor and a high-frequency map.

However, such methods struggle when the reference and background images belong to heterogeneous domains. To overcome this limitation, methods such as TF-ICON (Lu et al., 2023b), TALE (Pham et al., 2024), and AIComposer (Li et al., 2025) have been proposed for cross-domain compositing, performing latent- or pixel-level blending based on a frozen text-to-image backbone (Rombach et al., 2022). Despite improved cross-domain harmonization, these approaches still exhibit several limitations. For instance, without explicit supervision, it remains unclear how much

blending should be applied, as excessive blending may degrade foreground identity, while insufficient blending may prevent the foreground from fully adapting to the background domain (Fig. 1 (b)). Moreover, these methods rely on blending-based refinement rather than learning the insertion process, and thus struggle to jointly synthesize the inserted foreground and compositional effects such as shadows and reflections (Fig. 1 (d),(e)).

Style transfer. Style transfer has been widely studied and is relevant to cross-domain compositing in how visual style is represented and transferred across domains. Early neural style transfer methods (Gatys et al., 2016) represent style through Gram statistics of pre-trained VGG features (Simonyan & Zisserman, 2014), primarily capturing low-level appearance statistics such as color and texture. Recent diffusion-based approaches increasingly leverage semantic image encoders such as CLIP (Radford et al., 2021; Ye et al., 2023) for higher-level style conditioning. More recent works have explored self-supervised features such as DINO (Caron et al., 2021; Oquab et al., 2024), but primarily use them as content or semantic references, even in style-oriented applications (Jia et al., 2026; Zhou et al., 2024). In contrast, our analysis shows that DINOv3 (Siméoni et al., 2025) already contains substantial style information, while its style and content representations remain entangled (Tab. 11). Such entanglement is particularly detrimental to cross-domain compositing, as background content can leak into the insertion region and interfere with foreground insertion (Fig. 3). We therefore fully leverage DINOv3 by disentangling its features into style and content representations via our Signed Style-Content Graph Learning (SSCGL), enabling separate style and content conditioning for cross-domain compositing.

3 METHOD

3.1 CHAMELEONDATASET

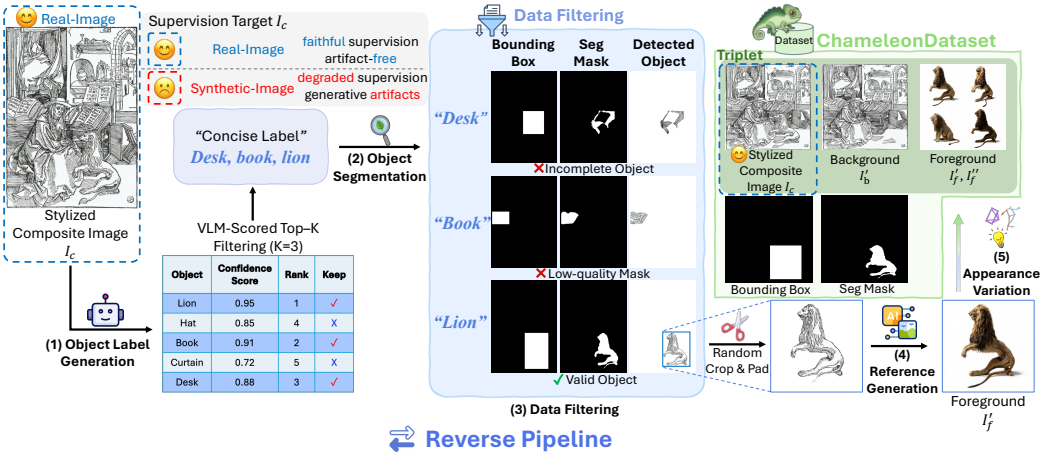


Figure 2: **Our reverse data generation pipeline.** Our reverse data generation pipeline starts from real stylized composite images I_c, ensuring *faithful, artifact-free supervision*.

Cross-domain compositing can be formulated over triplets (I_f, I_b, I_c), where a foreground image I_f is placed at a target location in a background image I_b and harmonized to the style of I_b, yielding the stylized composite I_c. Supervision is applied to I_c, so whether I_c follows a synthetic- or real-image distribution becomes critical (Sec. A.2). Throughout the paper, I_c denotes a real-image composite, and I_c' or I_c'' its synthetic counterpart, where $'$ denotes the synthetic output of a generative model.

Forward pipeline. Prior approaches (Huang et al., 2025; Yang et al., 2023; Canberk et al., 2024) adopt a forward data generation pipeline that constructs the supervision target I_c' or I_c'' from synthetic generations (Fig. 8 in *Appendix*). Using such degraded supervisions leads to the model learning to reproduce that synthetic distribution rather than the real-image one, yielding an inherently suboptimal mapping regardless of model capacity.

Reverse pipeline. We address these limitations by inverting the construction process. Rather than *generating* stylized composites, our *reverse pipeline* (Fig. 2) starts from stylized real-image composites I_c drawn from curated stylization datasets (Li et al., 2024; Liao et al., 2022; Ju et al., 2023) and treats them as ground-truth I_c. Given each I_c, we segment its salient foreground objects and, for each, synthesize a foreground image I_f' and remove the corresponding object from I_c using an off-the-shelf object removal model (Zhao et al., 2026) to obtain I_b', yielding one training triplet

$\{I'_f, I'_b, I_c\}$ per object. Here, I_c is a real image preserved from the source, while I'_f and the removed region of I'_b contain synthetic content and are used only as input conditions. When constructing I'_b, we retain the original pixels of I_c outside the segmentation mask to preserve the intact regions of the original background, such that only the removed foreground region is synthesized. Crucially, generative artifacts in I'_f and the removed region of I'_b do not affect the output distribution of the learned model, since supervision is applied only to I_c, which remains free of generative artifacts by construction. Our pipeline consists of five stages (Fig. 2). **(1) Object label generation.** We first employ Qwen3-VL (Bai et al., 2025) as a concise noun generator that produces object labels compatible with SAM3 (Carion et al., 2026). **(2) Object segmentation.** Using the filtered labels, SAM3 produces foreground candidates from I_c, each represented as a segmented region with a corresponding mask. **(3) Data filtering.** We then prompt Qwen3-VL (Bai et al., 2025) again with a filtering instruction to score each candidate along multiple criteria and produce a binary keep/reject decision (see *Appendix* for details). **(4) Reference generation.** Each valid candidate is passed through a reference generation model (Wu et al., 2025) to obtain I'_f. Although I'_f is itself a synthetic-image, it is only used as input. Supervision thus remains *faithful* and *artifact-free* on the real-image I_c. **(5) Appearance variation.** Finally, we apply appearance variation generation (Wu et al., 2025) to a subset of the resulting I'_f, perturbing camera parameters (e.g., azimuth, elevation) to obtain I''_f with diverse poses and lighting. This forces the model to match I''_f with the pose and lighting of I_c rather than naively pasting it. Construction details and stage-wise statistics are provided in Appendix A.4.

3.2 Design Considerations for Chameleon

The goal of cross-domain compositing is to composite I_f onto I_b across heterogeneous domains while preserving the identity of I_f and adapting its style to I_b. Meeting these requirements raises two key design considerations: *role-specific representations* and *adaptive style injection*. Together, these considerations motivate a two-stage framework with complementary roles: Stage 1 learns role-specific representations, and Stage 2 uses them to adaptively inject the resulting style representation.

Role-specific representations. Preserving foreground identity while adapting it to the background style requires disentangled, role-specific representations: *content* from I_f and *style* from I_b. Yet off-the-shelf encoders (Radford et al., 2021; Tschannen et al., 2025) entangle the two, potentially leaking foreground style or background content into the composite (Fig. 3). To obtain representations specialized for these distinct roles, **Stage 1** (Sec. 3.3) pre-trains *ChameleonEncoder*, a style-content disentangled encoder on a DINOv3 backbone via Signed Style-Content Graph Learning (SSCGL).

Adaptive style injection. Given these disentangled representations, the next challenge is to properly inject the style signal into the DiT. The central difficulty lies in calibrating, through the DINO style tokens, *how much*, *where*, and *when* to inject style. Accordingly, **Stage 2** (Sec. 3.4) uses ChameleonEncoder's disentangled representations to condition the DiT, while reusing its Stage 1 style head to compute the foreground–background style gap from I_f and I_b. We introduce Style Gap-Aware Anisotropic Modulation (SGAM), which adaptively modulates style attention according to this gap, spatial region, and diffusion timestep, without disturbing other tokens.

3.3 Stage 1 – ChameleonEncoder: Style-Content Disentangled Encoder

Leveraging Semantic Encoders for Style and Content Embeddings.

Cross-domain compositing requires an encoder that provides role-specific signals: foreground content to preserve object identity and background style to adapt the inserted foreground to match the appearance of I_b. Recent self-supervised visual encoders such as DINOv3 (Siméoni et al., 2025) have demonstrated strong transferability and have been increasingly adopted across diverse downstream tasks, including fine-grained retrieval (Liu et al., 2026), open-vocabulary detection (Sun et al., 2026) and 3D visual understanding (Ehm et al., 2026), making them a compelling source of *semantic* features. However, these applications primarily exploit DINO features for semantic or structural information. Even in style-oriented applications, DINO features are

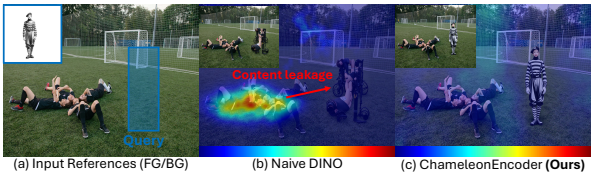


Figure 3: We visualize attention from the query region (a) to the background, where red indicates higher attention. (b) Naive DINO focuses on human regions, causing *content leakage*. (c) ChameleonEncoder (*style* head) attends globally, capturing background *style* rather than *content*, while inserting the foreground.

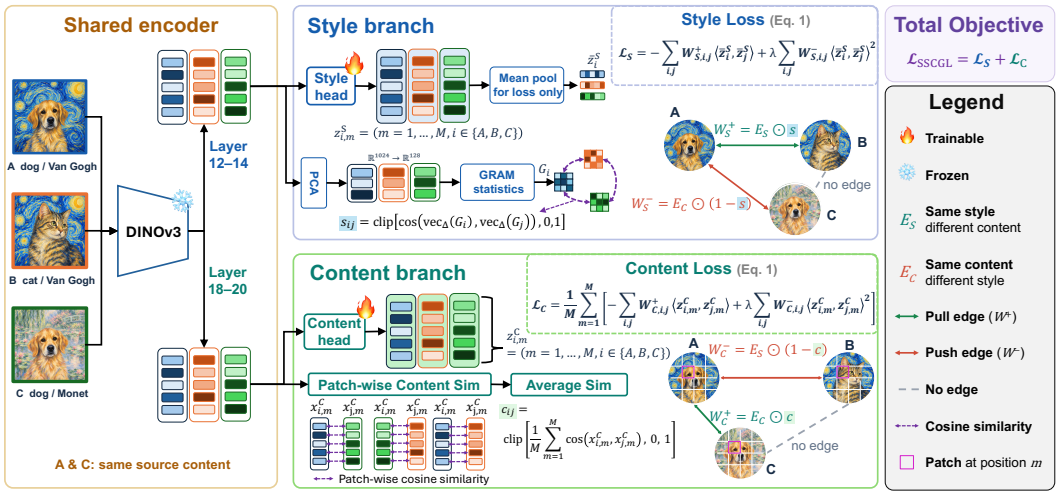


Figure 4: **Stage 1. ChameleonEncoder training with Signed Style-Content Graph Learning (SSCGL).** Given a shared DINOv3 backbone, SSCGL constructs signed, weighted graphs from style and content relations to learn disentangled representations.

primarily used as content or semantic references (Jia et al., 2026; Zhou et al., 2024). We therefore examine whether DINOv3 can provide not only semantic information but also style-relevant information required for cross-domain compositing. Our analysis in Tab. 11 shows that DINOv3 captures stronger style-related similarity than other off-the-shelf semantic encoders, yet its style and content representations remain entangled. To probe this, we feed I_f (Fig. 3 (a) top-left) and I_b to the DiT as VAE latents, and condition on (raw) DINOv3 tokens extracted from I_b. As shown in Fig. 3, the model generates background people in the insertion region instead of the foreground object due to the content information retained in the raw DINOv3 embeddings. We term this failure *content leakage*, where background content is transferred together with style. Taken together, these observations indicate that DINOv3 contains substantial style-relevant information, but this information remains entangled with content, motivating explicit style-content disentanglement.

Signed Style-Content Graph Learning (SSCGL). To achieve such disentanglement, contrastive learning has been explored for general disentangled representation learning (Matthes et al., 2023). In standard contrastive learning, methods such as SimCLR (Chen et al., 2020) and hard-negative variants such as HCL (Robinson et al., 2020) learn representations invariant to augmentations that are assumed to preserve semantic content by treating differently augmented versions of the same image as positive pairs under InfoNCE-style objectives (Oord et al., 2018). However, this augmentation-based supervision is ill-suited to style-content disentanglement, since transformations such as color jittering perturb style-related factors (e.g., brightness, contrast, and color) while the resulting samples are still treated as equivalent.

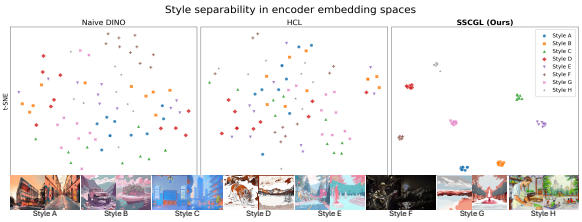


Figure 5: **Style disentanglement in encoder embeddings.** t-SNE of 8 styles with 10 samples each. Naive DINO and HCL yield entangled style representations, while SSCGL forms clearly separated style clusters.

Hence, rather than constructing positive pairs through augmentation, we leverage explicit style and content annotations available in the training data (Wang et al., 2025) to reorganize the samples into factor-specific relation sets. This allows us to identify image pairs that share style but differ in content, or share content but differ in style. Crucially, SSCGL applies complementary constraints to the *same image pair* across both representation spaces, pulling it together in one space and pushing it apart in the other. For example, in Fig. 4, the dog/Gogh and cat/Gogh images share style but differ in content, and are therefore pulled together in the style space while being pushed apart in the content space. Conversely, dog/Gogh and dog/Monet share content but differ in style, yielding the opposite assignment. Using *different image pairs* across the two spaces would break this complementary coupling, allowing each representation to rely on cues from the non-target factor. For example, when aligning dog/Gogh and cat/Gogh in the style space, the model could exploit shared animal-level semantic cues rather than relying solely on their common Gogh style. Concretely, we define two relation-specific edge sets: E_S for image pairs that share

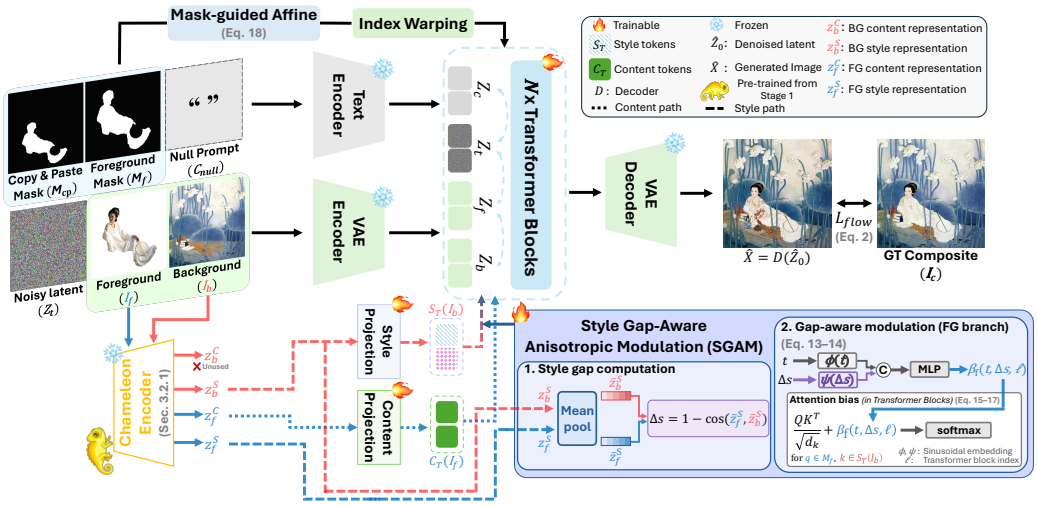


Figure 6: **Stage 2.** Cross-domain compositing with foreground content and background style tokens from ChameleonEncoder, with SGAM adaptively regulating style injection.

style but differ in content, and E_C for pairs that share content but differ in style. Unlike conventional contrastive learning that samples isolated anchor-positive pairs, SSCGL includes all within-batch relations satisfying each factor condition, providing group-level relational supervision that encourages factor-wise clustering (Fig. 5). However, these label-defined relations are binary $(0/1)$, while the degree of style or content similarity can differ across image pairs. We therefore assign factor-specific weights using frozen DINOv3 features. Specifically, let $s = [s_{ij}]$ denote style similarity measured using Gram statistics, and $c = [c_{ij}]$ denote content similarity measured using patch-wise cosine similarity between corresponding tokens. Combining E_S and E_C with these factor-specific similarities gives $W_S^+ = E_S \odot s$ and $W_S^- = E_C \odot (1 - s)$ for the style space, and $W_C^+ = E_C \odot c$ and $W_C^- = E_S \odot (1 - c)$ for the content space. Further details in Appendix B.2. Given the signed and weighted relations defined above, we jointly optimize the style and content projection heads of *ChameleonEncoder* with the following objectives:

$$\mathcal{L}_{\mathrm{S}} = \underbrace{-\sum_{i,j} W^+_{\mathrm{S},ij} \langle \bar{z}^{\mathrm{S}}_i, \bar{z}^{\mathrm{S}}_j \rangle}_{\text{positive-edge}} + \lambda \underbrace{\sum_{i,j} W^-_{\mathrm{S},ij} \langle \bar{z}^{\mathrm{S}}_i, \bar{z}^{\mathrm{S}}_j \rangle^2}_{\text{negative-edge}}, \quad \mathcal{L}_{\mathrm{C}} = \frac{1}{M} \sum_{m=1}^{M} \Big[-\sum_{i,j} W^+_{\mathrm{C},ij} \langle z^{\mathrm{C}}_{i,m}, z^{\mathrm{C}}_{j,m} \rangle + \lambda \sum_{i,j} W^-_{\mathrm{C},ij} \langle z^{\mathrm{C}}_{i,m}, z^{\mathrm{C}}_{j,m} \rangle^2 \Big]. \quad (1)$$

where $\bar{z}^{\mathrm{S}}_i$ is the average-pooled style representation, following prior work (Gatys et al., 2016), while $z^{\mathrm{C}}_{i,m}$ denotes the m-th content patch representation, preserving spatial correspondence. M is the number of patch tokens, $\langle \cdot, \cdot \rangle$ the Euclidean inner product, and λ balances the positive- and negative-edge terms. We fix $\lambda = 1$ for all experiments. We constrain the projected representations to the unit ball and normalize each edge-weight matrix to unit sum. The final objective is $\mathcal{L}_{\mathrm{SSCGL}} = \mathcal{L}_{\mathrm{S}} + \mathcal{L}_{\mathrm{C}}$.

This objective explicitly drives negative-edge pairs toward orthogonality while aligning positive-edge pairs, rather than relying on relative similarity as in InfoNCE. Its global minimum yields style and content representations invariant to content and style, respectively (Appendix B.2). These factor-selective representations capture the background style from I_b without content leakage while preserving foreground identity from I_f, for effective cross-domain compositing.

3.4 Stage 2 – Cross-Domain Compositing Model

Building on the disentangled representations learned by ChameleonEncoder in Stage 1 (Sec. 3.3), our Stage 2 cross-domain compositing model uses these representations as explicit conditions for the DiT (Fig. 6). Specifically, ChameleonEncoder extracts content tokens $C_T(I_f)$ from the foreground image to preserve object identity and style tokens $S_T(I_b)$ from the background image to capture the target style, both of which are injected into the DiT. The model is trained on triplets $\{I_f, I_b, I_c\}$ from ChameleonDataset (Sec. 3.1), where I_c provides real-image supervision during training and Z_t is initialized from Gaussian noise at inference. The foreground I_f and background I_b are encoded into latents Z_f and Z_b, concatenated with Z_t, and fed into the DiT. We adopt a null prompt C_{null} since text is ambiguous for instance-level conditioning, relying instead fully on I_f and I_b. For spatial placement, rather than using M_f as a condition token, we exploit M_f and the copy-and-paste mask M_{cp} to compute a positional affine transformation that warps foreground latent indices to their target

placement (Appendix E). To adaptively regulate the influence of the background style tokens $S_T(I_b)$, we propose Style Gap-Aware Anisotropic Modulation (SGAM), detailed in the next subsection.

Style Gap-Aware Anisotropic Modulation (SGAM). To effectively harmonize heterogeneous domains between the foreground and background images, the background style tokens $S_T(I_b)$ must be properly injected into the DiT. Since the foreground must adapt to the background style while preserving its identity, excessive style injection may degrade foreground identity, whereas insufficient injection may leave the foreground stylistically inconsistent with the background. Effective style injection should therefore account for *where*, *when*, and *how much* style is injected. Spatially, we focus style injection on foreground tokens, since the background already represents the target style and only the foreground requires adaptation. Moreover, diffusion model inference evolves from noise to structured representations over timesteps, making style injection at early noisy stages less effective as meaningful stylization should occur when semantic structure begins to emerge (Hu et al., 2024). The appropriate injection strength should also vary with the style gap between I_f and I_b. Stylistically similar pairs require only modest adaptation, whereas larger style gaps call for stronger modulation. While related diffusion-based approaches (Hu et al., 2024; Jeong et al., 2025) modulate conditional information only over diffusion timesteps using *fixed* schedules, we instead introduce Style Gap-Aware Anisotropic Modulation (SGAM), which *adaptively* and anisotropically modulates style injection across three axes: the style gap, spatial region, and diffusion timestep. Specifically, we map the diffusion timestep t to a sinusoidal time embedding and additionally condition the foreground modulation branch on the foreground-background style gap derived from the ChameleonEncoder style head in Stage 1, while the background branch remains conditioned only on t. The resulting modulation coefficients are assigned to query tokens based on their spatial locations via the foreground mask and applied as an attention bias exclusively to keys corresponding to $S_T(I_b)$ at every transformer block (full derivation in *Appendix*). Consistent with this design, SGAM causes foreground-region queries to attend strongly to $S_T(I_b)$ while adaptively varying the modulation strength according to the style gap and diffusion timestep. We further analyze its spatial, style-gap-aware, timestep-adaptive, and block-wise behaviors in Fig. 12 in *Appendix*.

4 Experiments

4.1 Experimental Set-up

Evaluation Metrics. Following Lu et al. (2023b); Li et al. (2025), we measure style consistency with **CSD** (Somepalli et al., 2024), and foreground preservation with **LPIPS** (Zhang et al., 2018) and **CLIP-I** (Hessel et al., 2021). We further report a **VLM-based score** (Ku et al., 2024; Peng et al., 2025) adapted from Ju et al. (2026) for cross-domain compositing. **Benchmarks.** Cross-domain compositing lacks a dedicated benchmark. Existing benchmarks (Lu et al., 2023b; Li et al., 2025) have limited diversity in foregrounds and background styles. We therefore introduce **ChameleonDataset$_{ev}$**, covering diverse styles and challenging cases (*e.g.*, reflections and pixel art), and evaluate on both. **Implementation.** Stage 1 trains ChameleonEncoder on a frozen DINOv3 ViT-L/16. Stage 2 fine-tunes a DiT with LoRA (Hu et al., 2022). Full details in Appendix C.

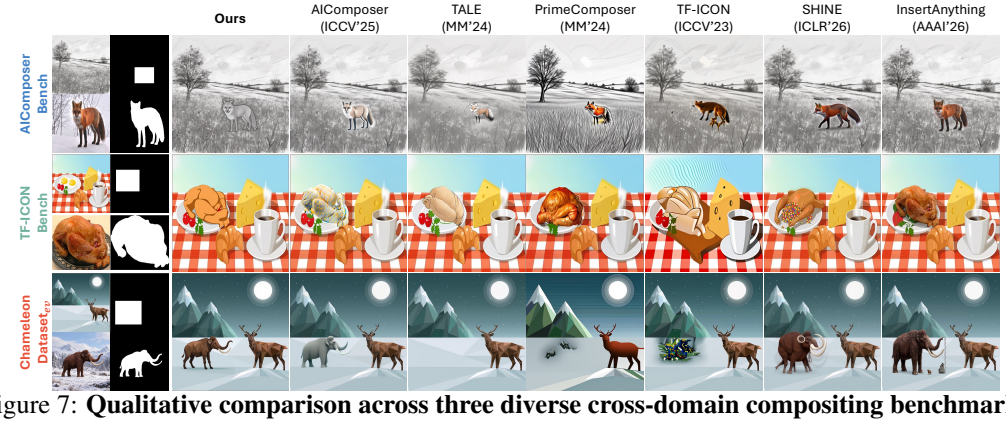


Figure 7: **Qualitative comparison across three diverse cross-domain compositing benchmarks.**

4.2 Experimental Results

Qualitative Comparison. In Fig. 7, we compare our method with cross-domain methods (Li et al., 2025; Pham et al., 2024; Wang et al., 2024; Lu et al., 2023b) and in-domain methods (Lu et al., 2026; Song et al., 2026) (two columns on the right). In the last row, the in-domain methods preserve fore-

Table 2: **Quantitative comparison on TF-ICON and AIComposer benchmarks.** Reference-based metrics (LPIPS, CLIP-I, CSD) are reported per benchmark, while VLM-based scores (Identity, Style, Composition, Avg_total) are averaged over the two benchmarks. **Bold** and underline denote the best and second-best results. Each method is evaluated at its backbone's native resolution.

		TF-ICON			AIComposer			VLM (avg.)			
Method	Res	LPIPS ↓	CLIP-I ↑	CSD ↑	LPIPS ↓	CLIP-I ↑	CSD ↑	Identity ↑	Style ↑	Composition ↑	Avg_total ↑
In-domain compositing											
BLD (SIGGRAPH'23)	512	0.7540	0.7331	0.4619	0.6821	0.7905	0.3624	2.07	2.71	2.08	6.86
Paint by Example (CVPR'23)	512	0.7501	0.7658	0.3138	0.6754	0.8042	0.3287	2.19	2.45	2.14	6.78
Insert Anything (AAAI'26)	1024	0.7281	0.7719	0.3821	0.5418	0.8487	0.3956	2.33	2.56	2.20	7.09
Dreamfuse (ICCV'25)	1024	0.6892	0.7894	0.4035	0.5290	0.8603	0.3941	2.36	2.51	2.07	6.94
SHINE (ICLR'26)	512	0.7654	0.7155	0.4362	0.6892	0.8214	0.3387	1.97	2.62	2.10	6.69
SHINE (ICLR'26)	1024	0.7737	0.7165	0.4627	0.6513	0.8377	0.3512	2.04	2.56	2.11	6.70
Two-stage compositing											
StyleSSP (CVPR'25) + Insert Anything (AAAI'26)	1024	0.8410	0.7223	**0.5535**	0.4682	0.8438	**0.4898**	2.31	2.85	2.01	7.20
Cross-domain compositing											
TF-ICON (ICCV'23)	512	0.7740	0.7551	0.4103	0.6640	0.7493	0.3536	1.88	2.61	1.97	6.46
TF-ICON (ICCV'23)	768	0.7440	0.7581	0.4073	0.6539	0.7700	0.3464	1.95	2.56	2.02	6.52
TALE (MM'24)	512	0.7990	0.5395	0.4982	0.6164	0.7830	0.4681	1.91	2.84	2.14	6.89
PrimeComposer (MM'24)	512	0.7762	0.7510	0.4285	0.6438	0.7682	0.4124	2.10	2.71	1.95	6.78
AIComposer (ICCV'25)	512	0.5025	0.7946	0.4674	0.4895	0.8412	0.4729	2.21	2.80	2.04	7.15
AIComposer (ICCV'25)	1024	0.4966	0.8264	0.4826	0.4648	0.8575	0.4848	2.40	2.84	2.09	7.33
Ours	512	0.5018	0.8001	0.4728	0.4724	0.8507	0.4795	2.46	2.89	2.26	7.62
Ours	1024	**0.4855**	**0.8296**	0.4994	**0.4556**	**0.8637**	0.4886	**2.53**	**2.90**	**2.31**	**7.73**

Table 3: Ablation on the AIComposer benchmark, cumulatively adding SSCGL and SGAM.

Method	LPIPS ↓	CLIP-I ↑	CSD ↑	Aes ↑
Baseline	0.4869	0.8495	0.3992	6.8267
+ SSCGL	0.4673	**0.8645**	0.4527	6.9209
+ SSCGL + SGAM	**0.4556**	0.8637	**0.4886**	**7.0296**

Table 4: User study win rates (%) across 30 $\text{ChameleonDataset}_{\text{ev}}$ cases with 50 participants.

Criteria	TF-ICON	TALE	PrimeComposer	AI-Composer	**Ours**
Identity (↑)	1.8	0.1	2.3	31.4	**64.3**
Style (↑)	1.8	0.8	1.3	25.9	**70.2**
Overall (↑)	2.7	0.5	1.4	25.3	**70.1**

ground identity but retain a photorealistic appearance, failing to adapt to the low-poly background style. Most cross-domain methods substantially distort foreground identity. AIComposer better preserves foreground identity, yet mainly blends surrounding color tones without reproducing the target style or compositional effects. In contrast, our result preserves identity while matching the low-poly style and producing a plausible grounded shadow. More results are in *Appendix*.

Quantitative Comparison. Quantitative results on TF-ICON and AIComposer benchmarks reported in Tab. 2. In-domain methods generally exhibit low scores on CSD, which measures style consistency, while cross-domain methods tend to achieve lower semantic alignment scores, such as CLIP-I. 2-stage cascaded pipeline with a stylization method and a compositing method (Song et al., 2026) results in high CSD due to over-stylization (see Fig. 1 (g)). In contrast, our method simultaneously achieves good identity preservation and consistent stylization with best or second-best results across all metrics, which is further supported by a user study on $\text{ChameleonDataset}_{\text{ev}}$ (Tab. 4). Full results for each dataset including additional metrics (CLIP-T (Radford et al., 2021) and Aes (Schuhmann, 2022)) are provided in *Appendix*.

Ablation Study. Tab. 3 presents a component-wise analysis of our method. We use naive LoRA fine-tuning of the DiT on $\text{ChameleonDataset}_{\text{tr}}$ as the baseline. Although this baseline produces plausible composites, it fails to fully capture the target style, resulting in relatively lower CSD scores. Replacing the training objective with SSCGL, which disentangles content and style representations for separate injection, consistently improves both CLIP-I and CSD, validating the benefit of explicit representation disentanglement. Adding SGAM further enhances stylization while maintaining stable CLIP-I scores and improving LPIPS, indicating preserved content fidelity. More detailed ablation studies and analyses of individual components are provided in *Appendix*.

5 Conclusion

We address the challenging problem of cross-domain compositing, which requires preserving foreground identity while adapting to background style. We propose Chameleon, a two-stage training-based framework built on ChameleonDataset, the first large-scale dataset for this task. Our SSCGL objective effectively disentangles style and content, and SGAM adaptively regulates style injection into the DiT. Unlike prior training-free blending-based approaches, our method achieves both strong identity preservation and background-consistent stylization.

AI USE STATEMENT

In this work, we used generative AI tools to generate our synthetic training dataset (Sec. 3.1 and Appendix A.4). Qwen3-VL (Bai et al., 2025) was used to generate object labels from real stylized composite images and to score and filter segmented candidates, SAM 3 (Carion et al., 2026) was used for object segmentation, ObjectClear (Zhao et al., 2026) was used to remove foreground objects and restore the backgrounds, and Qwen-Image-Edit (Wu et al., 2025) was used to generate foreground references and their appearance variations. The synthesized content is used only as input conditions, and all supervision targets are real images. We validated the pipeline through both manual and automatic checks. The SAM 3 threshold was chosen from manual inspection of 500 samples, candidates were filtered with the VLM criteria in Appendix H.2, and generated references with identity changes were removed by CLIP-I filtering. In addition, we use a VLM-based score as one of several evaluation metrics (Sec. 4.1 and Appendix H). Additionally, we used generative AI tools to aid and polish writing, namely checking grammar, improving the readability of the manuscript, and formatting LaTeX and references. We did not use generative AI tools for any purpose other than those described above. We have reviewed all AI-assisted work, and all AI-edited text was checked by the authors for factual and technical accuracy. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The dataset construction pipeline is described in Sec. 3.1, with stage-wise statistics and filtering criteria in Appendix A.4 and Appendix H.2. The theoretical analysis of SSCGL and the full formulation of SGAM are given in Appendix B.2 and Appendix B.3, and implementation details, including architectures, optimizers, training schedules, hyperparameters, and compute, are provided in Appendix C. Evaluation protocols and all VLM prompts are described in Sec. 4.1 and Appendix H. We will publicly release the source code and ChameleonDataset to facilitate reproducibility.

ETHICS STATEMENT

The training dataset is constructed from publicly available stylization datasets described in Sec. 3.1, used within the scope permitted by their respective licenses, and our evaluation benchmark is built from openly licensed images from Pexels, Pixabay, and Unsplash (Appendix A.5). The user study (Appendix C.4) was conducted anonymously with 50 participants without monetary compensation, and no personally identifiable information was recorded. All external models are used in accordance with their respective licenses (Appendix C.3), and our code and dataset will be released for non-commercial research use.

REFERENCES

Jie An, Siyu Huang, Yibing Song, Dejing Dou, Wei Liu, and Jiebo Luo. Artflow: Unbiased image style transfer via reversible neural flows. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 862–871, 2021.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Black Forest Labs. FLUX.1 [dev]. https://huggingface.co/black-forest-labs/FLUX.1-dev, 2024.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18392–18402, 2023.

Alper Canberk, Maksym Bondarenko, Ege Ozguroglu, Ruoshi Liu, and Carl Vondrick. Erasedraw: Learning to insert objects by erasing them from images. In *European Conference on Computer Vision*, pp. 144–160. Springer, 2024.

Gemma Canet Tarrés, Zhe Lin, Zhifei Zhang, Jianming Zhang, Yizhi Song, Dan Ruta, Andrew Gilbert, John Collomosse, and Soo Ye Kim. Thinking outside the bbox: Unconstrained generative object compositing. In *European Conference on Computer Vision*, pp. 476–495. Springer, 2024.

Junyan Cao, Yan Hong, and Li Niu. Painterly image harmonization in dual domains. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 268–276, 2023.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollar, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment anything with concepts. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=r35clVtGzw.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

Jiaxuan Chen, Bo Zhang, Qingdong He, Jinlong Peng, and Li Niu. Carecom: Generative image composition with calibrated reference features. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2877–2885, 2026.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PmLR, 2020.

Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. Anydoor: Zero-shot object-level image customization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6593–6602, 2024.

Jiwoo Chung, Sangeek Hyun, and Jae-Pil Heo. Style injection in diffusion: A training-free approach for adapting large-scale diffusion models for style transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8795–8805, 2024.

Tyler Derr, Yao Ma, and Jiliang Tang. Signed graph convolutional networks. In *2018 IEEE international conference on data mining (ICDM)*, pp. 929–934. IEEE, 2018.

Viktoria Ehm, Dongliang Cao, Riccardo Marin, Daniel Scholz, Weikang Wang, Florian Bernard, and Daniel Cremers. Teaching dinov3 about partial 3d geometry: A self-supervised geometry-aware approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 42071–42081, 2026.

Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

Junyao Gao, Yanan Sun, Yanchen Liu, Yinhao Tang, Yanhong Zeng, Ding Qi, Kai Chen, and Cairong Zhao. Styleshot: A snapshot on any style. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2414–2423, 2016.

Google DeepMind. Gemini 3 pro image (nano banana pro). https://deepmind.google/models/gemini-image/pro/, 2025. Model ID: gemini-3-pro-image-preview.

Google DeepMind. Gemini 3.1 flash image (nano banana 2). https://deepmind.google/models/gemini-image/, 2026. Model ID: gemini-3.1-flash-image-preview.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 7514–7528, 2021.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Ying Hu, Chenyi Zhuang, and Pan Gao. Diffusest: Unleashing the capability of the diffusion model for style transfer. In *Proceedings of the 6th ACM International Conference on Multimedia in Asia*, pp. 1–1, 2024.

Junjia Huang, Pengxiang Yan, Jiyang Liu, Jie Wu, Zhao Wang, Yitong Wang, Liang Lin, and Guanbin Li. Dreamfuse: Adaptive image fusion with diffusion transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17292–17301, 2025.

Junjie Huang, Huawei Shen, Liang Hou, and Xueqi Cheng. Sdgnn: Learning node representation for signed directed networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 196–203, 2021.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 1501–1510, 2017.

Dasol Jeong, Donggoo Kang, Jiwon Park, Hyebean Lee, and Joonki Paik. Structure-preserving zero-shot image editing via stage-wise latent injection in diffusion models. *arXiv preprint arXiv:2504.15723*, 2025.

Zexi Jia, Jinchao Zhang, and Jie Zhou. Styledecoupler: Generalizable artistic style disentanglement. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 9327–9331. IEEE, 2026.

Xuan Ju, Ailing Zeng, Jianan Wang, Qiang Xu, and Lei Zhang. Human-art: A versatile human-centric dataset bridging natural and artificial scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 618–629, 2023.

Xuan Ju, Tianyu Wang, Yuqian Zhou, He Zhang, Qing Liu, Nanxuan Zhao, Zhifei Zhang, Yijun Li, Yuanhao Cai, Shaoteng Liu, Daniil Pakhomov, Zhe Lin, Soo Ye Kim, and Qiang Xu. Editverse: Unifying image and video editing and generation with in-context learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=blJXE07r7I.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023.

Max Ku, Dongfu Jiang, Cong Wei, Xiang Yue, and Wenhu Chen. Viescore: Towards explainable metrics for conditional image synthesis evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12268–12290, 2024.

Haowen Li, Zhenfeng Fan, Zhang Wen, Zhengzhou Zhu, and Yunjin Li. Aicomposer: Any style and content image composition via feature integration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16840–16850, 2025.

Wen Li, Muyuan Fang, Cheng Zou, Biao Gong, Ruobing Zheng, Meng Wang, Jingdong Chen, and Ming Yang. Styletokenizer: Defining image style by a single instance for controlling diffusion models. In *European Conference on Computer Vision*, pp. 110–126. Springer, 2024.

Peiyuan Liao, Xiuyu Li, Xihui Liu, and Kurt Keutzer. The artbench dataset: Benchmarking generative models with artworks. *arXiv preprint arXiv:2206.11404*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Yushi Liu, Christian Graf, Markus Spies, and Margret Keuper. Efficient fine-grained image retrieval with vision foundation models for industrial objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3111–3120, 2026.

Lingxiao Lu, Jiangtong Li, Junyan Cao, Li Niu, and Liqing Zhang. Painterly image harmonization using diffusion model. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 233–241, 2023a.

Shilin Lu, Yanzhu Liu, and Adams Wai-Kin Kong. Tf-icon: Diffusion-based training-free cross-domain image composition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2294–2305, 2023b.

Shilin Lu, Zhuming Lian, Zihan Zhou, Shaocong Zhang, Chen Zhao, and Adams Wai-Kin Kong. Does FLUX already know how to perform physically plausible image composition? In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=DcVg87ibK9.

Stefan Matthes, Zhiwei Han, and Hao Shen. Towards a unified framework of contrastive learning for disentangled representations. *Advances in neural information processing systems*, 36:67459–67470, 2023.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=aBsCjcPu_tE.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6038–6047, 2023.

Li Niu, Junyan Cao, Yan Hong, and Liqing Zhang. Painterly image harmonization by learning from painterly objects. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 4343–4351, 2024.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

OpenAI. Introducing ChatGPT Images 2.0. https://openai.com/index/introducing-chatgpt-images-2-0/, April 2026. Accessed: 2026-05-07.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt. Featured Certification.

Yuang Peng, Yuxin Cui, Haomiao Tang, Zekun Qi, Runpei Dong, Jing Bai, Chunrui Han, Zheng Ge, Xiangyu Zhang, and Shu-Tao Xia. Dreambench++: A human-aligned benchmark for personalized image generation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4GSOESJrk6.

Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pp. 577–582. 2003.

Pexels. Pexels. https://www.pexels.com, 2026.

Kien T Pham, Jingye Chen, and Qifeng Chen. Tale: Training-free cross-domain image composition via adaptive latent manipulation and energy-guided optimization. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 3160–3169, 2024.

Pixabay. Pixabay. https://pixabay.com, 2026.

Thomas Porter and Tom Duff. Compositing digital images. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pp. 253–259, 1984.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001.

Jonas Ricker, Denis Lukovnikov, and Asja Fischer. Aeroblade: Training-free detection of latent diffusion images using autoencoder reconstruction error. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9130–9140, 2024.

Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*, 2020.

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

Nataniel Ruiz, Yuanzhen Li, Neal Wadhwa, Yael Pritch, Michael Rubinstein, David E Jacobs, and Shlomi Fruchter. Magic insert: Style-aware drag-and-drop. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15971–15981, 2025.

Christoph Schuhmann. Improved Aesthetic Predictor. https://github.com/christophschuhmann/improved-aesthetic-predictor, 2022.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Gowthami Somepalli, Anubhav Gupta, Kamal Gupta, Shramay Palta, Micah Goldblum, Jonas Geiping, Abhinav Shrivastava, and Tom Goldstein. Investigating style similarity in diffusion models. In *European Conference on Computer Vision*, pp. 143–160. Springer, 2024.

Wensong Song, Hong Jiang, Zongxing Yang, Zheqiao Cheng, Ruijie Quan, and Yi Yang. Insert anything: Image insertion via in-context editing in DiT. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 9097–9105, 2026.

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*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Yizhi Song, Zhifei Zhang, Zhe Lin, Scott Cohen, Brian Price, Jianming Zhang, Soo Ye Kim, and Daniel Aliaga. Objectstitch: Object compositing with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18310–18319, 2023.

Chang Sun, Dongliang Liao, and Changxing Ding. Streamlined open-vocabulary human-object interaction detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20324–20333, June 2026.

Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2149–2159, 2022.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Unsplash. Unsplash. https://unsplash.com, 2026.

Ye Wang, Ruiqi Liu, Jiang Lin, Fei Liu, Zili Yi, Yilin Wang, and Rui Ma. Omnistyle: Filtering high quality style transfer data at scale. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 7847–7856, 2025.

Yibin Wang, Weizhong Zhang, Jianwei Zheng, and Cheng Jin. Primecomposer: Faster progressively combined diffusion for image composition with attention steering. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 10824–10832, 2024.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

Peng Xing, Haofan Wang, Yanpeng Sun, Qixun Wang, Xu Bai, Hao Ai, Jen-Yuan Huang, and Zechao Li. CSGO: Content-style composition in text-to-image generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=n4nmiIq3qj.

Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18381–18391, 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.

Bo Zhang, Jiacheng Sui, and Li Niu. Foreground object search by distilling composite image feature. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22929–22938. IEEE, 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*, pp. 586–595, 2018.

Jixin Zhao, Zhouxia Wang, Peiqing Yang, and Shangchen Zhou. Precise object and effect removal with adaptive target-aware attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19370–19379, 2026.

Yang Zhou, Zichong Chen, and Hui Huang. Deformable one-shot face stylization via dino semantic guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7787–7796, 2024.

APPENDIX CONTENTS

A Dataset Details: ChameleonDataset **17**
- A.1 Previous Construction Paradigms . . . 17
- A.2 Evidence of Synthetic Distribution Shift . . . 18
- A.3 ChameleonDataset$_{\mathrm{tr}}$ . . . 19
- A.4 Dataset Construction Statistics . . . 20
- A.5 ChameleonDataset$_{\mathrm{ev}}$ . . . 21

B Preliminaries, Derivation, and Algorithm **22**
- B.1 Preliminaries: Diffusion Transformer . . . 22
- B.2 Theoretical Analysis of Signed Style-Content Graph Learning . . . 22
- B.3 Style Gap-Aware Anisotropic Modulation (SGAM): Full Formulation . . . 24
- B.4 Batch Construction for SSCGL . . . 25

C Implementation Details **26**
- C.1 Training Details . . . 26
- C.2 Compute and Efficiency Analysis . . . 26
- C.3 External Models and Licenses . . . 27
- C.4 User Study . . . 27

D Ablation Studies **29**
- D.1 Encoder . . . 29
- D.2 SGAM . . . 30
- D.3 Qualitative Component-wise Ablation . . . 32

E Inference Assumptions and Robustness **32**

F Additional Quantitative Results **33**
- F.1 Full Quantitative Results . . . 33
- F.2 Pose Adaptation Evaluation . . . 35

G Additional Qualitative Results **35**
- G.1 Extended Benchmark Comparisons . . . 35
- G.2 Scene-Aware Pose Adaptation and Generalized Cross-Domain Compositing . . . 39

H VLM-based Protocols **40**
- H.1 VLM-based Evaluation Metric . . . 40
- H.2 VLM-based Filtering . . . 42

I Limitations and Broader Impacts **43**
- I.1 Limitations . . . 43
- I.2 Broader Impacts . . . 44

OVERVIEW

In this supplementary material, Sec. A describes *ChameleonDataset*. We revisit previous construction paradigms (Sec. A.1), provide evidence of the synthetic distribution shift they induce (Sec. A.2), and present the specification and stage-wise construction statistics of ChameleonDataset$_{\text{tr}}$ (Sec. A.4) as well as example samples from ChameleonDataset$_{\text{ev}}$ (Sec. A.5). Sec. B presents preliminaries on the Diffusion Transformer (Sec. B.1), a theoretical analysis of Signed Style-Content Graph Learning (SSCGL, Sec. B.2), the full formulation of Style Gap-Aware Anisotropic Modulation (SGAM, Sec. B.3), and the batch construction procedure for SSCGL (Sec. B.4). Sec. C reports training details, compute and efficiency analyses (Sec. C.2), external models and licenses (Sec. C.3), and the user study protocol (Sec. C.4). Sec. D provides ablation studies on the encoder design (Sec. D.1) and SGAM (Sec. D.2), followed by a qualitative component-wise ablation (Sec. D.3). Sec. E clarifies the inference assumptions and evaluates robustness to coarse localization and imperfect foreground inputs. Sec. F presents full quantitative comparisons against cross-domain compositing, painterly harmonization, and general-purpose image editing methods (Sec. F.1), along with a pose adaptation evaluation under substantial viewpoint changes (Sec. F.2). Sec. G shows extended benchmark comparisons and results on challenging and generalized compositing scenarios (Sec. G.2). Sec. H provides the VLM prompts used for evaluation and data filtering (Sec. H.2). Finally,

A DATASET DETAILS: CHAMELEONDATASET

A.1 PREVIOUS CONSTRUCTION PARADIGMS

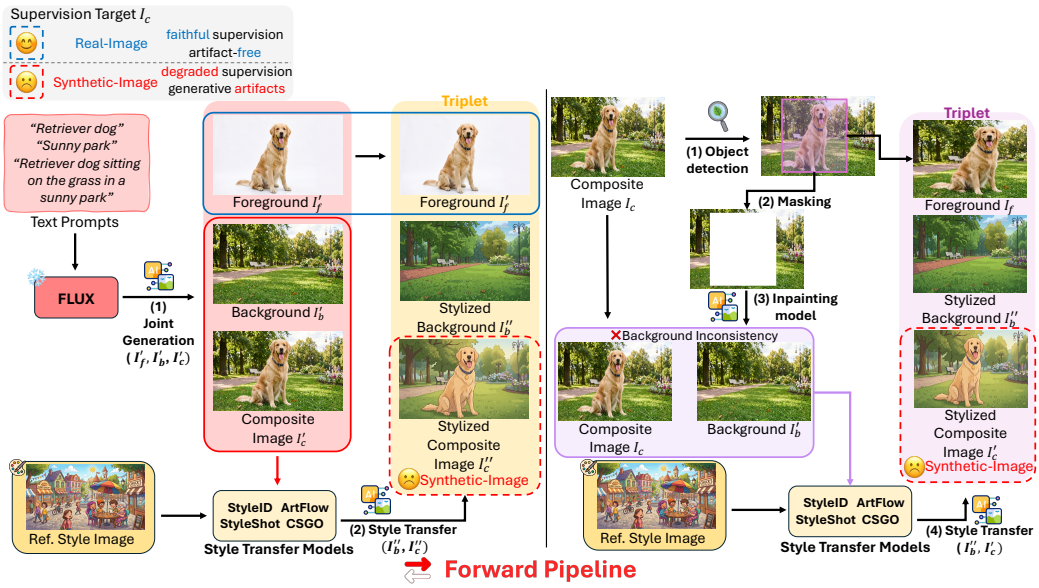


Figure 8: **Forward pipeline.** Prior approaches construct triplets through sequential generation and stylization, producing a synthetic-image supervision target I'_c or I''_c, where each $'$ denotes a stage of *synthetic degradation*.

Forward pipeline (top of Fig. 8) instantiates the synthetic construction through a sequential pipeline that couples compositing and stylization. It admits two entry points. In the first variant (left in Fig. 8), a compositing model (Huang et al., 2025) jointly denoises the foreground I'_f, background I'_b, and composite I'_c using a text-to-image backbone (Black Forest Labs, 2024), so that generative artifacts already arise in I'_c. In the second variant (top-right in Fig. 8), the pipeline starts from a real composite I_c (Yang et al., 2023). A foreground I_f is extracted by object detection (Kirillov et al., 2023), and the corresponding region is filled by an inpainting model (Suvorov et al., 2022) to obtain a synthetic background I'_b for the subsequent stylization stage. A stylization model (Chung et al., 2024; An et al., 2021; Gao et al., 2025; Xing et al., 2025) then transforms the input, conditioned on a reference style

image, into a stylized composite and a stylized background, yielding I'_c and I'_b in the first variant, and I''_c and I''_b in the second. The fundamental issue is that the stylized composite, whether I'_c or I''_c, is a *synthetic-image*, inheriting artifacts accumulated across both stages and ultimately following a synthetic-image distribution. Using such a composite as degraded supervision trains the model to reproduce this distribution rather than the real-image distribution, yielding an inherently suboptimal mapping regardless of model capacity.

A.2 Evidence of Synthetic Distribution Shift

AeroBLADE (Ricker et al., 2024) detects generated images from latent diffusion models using autoencoder reconstruction error, where lower reconstruction errors indicate a stronger generated-image signature. Given an image x, the method reconstructs it as $\hat{x} = \mathcal{D}(\mathcal{E}(x))$, where $\mathcal{E}$ and $\mathcal{D}$ denote the encoder and decoder of the latent diffusion model's autoencoder, respectively. The reconstruction error is then defined as $e(x) = \text{LPIPS}(x, \hat{x})$, the perceptual distance between x and $\hat{x}$. We repurpose this signal to compare the synthetic artifacts introduced by the reverse (Fig. 2 of the main paper) and forward (Fig. 8) generation paths. Specifically, we generate data through each pipeline and measure $e(x)$ on the resulting stylized composites: I_c (real) from the reverse path and I'_c or I''_c (synthetic) from the forward path. Each set contains 1,521 samples matched by style and object category. To quantify the separation between the two reconstruction-error distributions, we report the area under the receiver operating characteristic curve (AUROC) (Fawcett, 2006), which in our setting corresponds to the probability that a randomly selected reverse-pipeline composite yields a higher $e(x)$ than a randomly selected forward-pipeline composite. As shown in Fig. 9, the two pipelines produce clearly distinct reconstruction-error distributions, with an AUROC of 0.82. In other words, in 82% of randomly sampled reverse–forward pairs, the reverse composite exhibits a higher reconstruction error, indicating a weaker generated-image signature. Interestingly, we find that synthetic and real signatures separate differently across styles. In Tab. 5, the Real/Reverse and Synthetic/Forward columns report the LPIPS reconstruction error $e(x)$, and AUROC is computed in the same manner as above. This separation is most pronounced for texture-rich styles, where generation leaves clear synthetic artifacts, whereas the gap narrows for line-dominant styles, whose simpler structures leave fewer detectable artifacts.

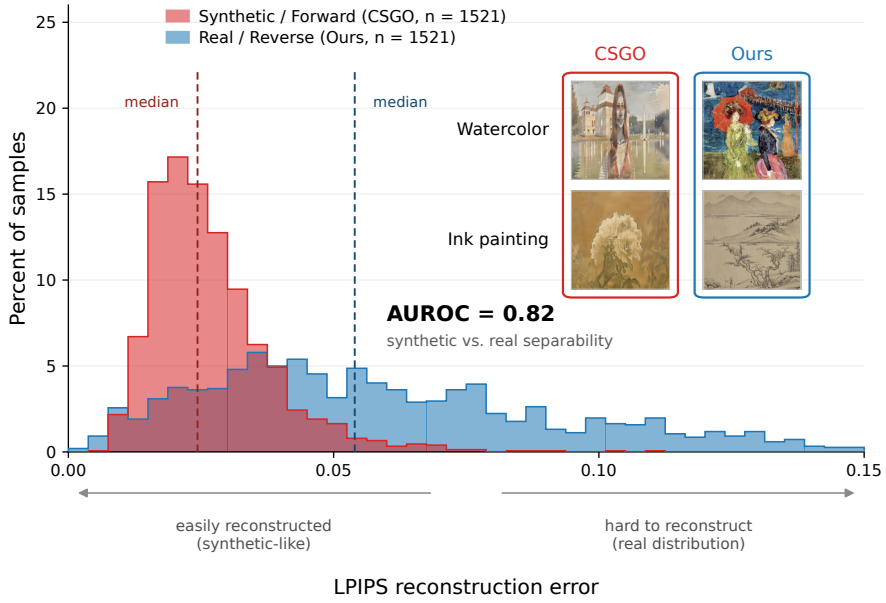


Figure 9: **Forward–reverse reconstruction gap.** We compare AeroBLADE-based LPIPS reconstruction errors of stylized composites from our reverse pipeline (Real / Reverse, Ours) and a forward pipeline using CSGO for stylization (Synthetic / Forward, CSGO). Both sets contain 1,521 samples with matched style and object distributions, isolating the effect of synthetic generation. Forward composites exhibit lower errors, a characteristic signature of diffusion-generated images, whereas reverse composites show higher errors, deviating from this synthetic-image signature. The resulting AUROC of 0.82 indicates clear separability between the two pipelines.

Table 5: **Style-wise analysis of synthetic generation artifacts.** We report the mean reconstruction error $e(x)$ of real composites from our reverse pipeline and synthetic composites from the forward pipeline (CSGO) per style, their ratio, and the AUROC between the two distributions. Styles are sorted in descending order of AUROC. Texture-rich styles (top rows) exhibit larger discrepancies, whereas line-dominant styles (bottom rows) show relatively small gaps.

Style	**Real / Reverse (Ours)**	**Synthetic / Forward (CSGO)**	**Ratio**	**AUROC**
Ink painting	0.077 ± 0.030	0.025 ± 0.008	$3.1\times$	0.963
Ukiyo-e	0.041 ± 0.009	0.024 ± 0.006	$1.7\times$	0.954
Baroque	0.089 ± 0.037	0.035 ± 0.011	$2.5\times$	0.953
Game style	0.072 ± 0.033	0.027 ± 0.011	$2.7\times$	0.914
Watercolor	0.073 ± 0.036	0.032 ± 0.013	$2.3\times$	0.886
Expressionism	0.069 ± 0.041	0.032 ± 0.015	$2.2\times$	0.834
Traditional Chinese	0.039 ± 0.021	0.022 ± 0.006	$1.8\times$	0.777
Hand-drawn	0.024 ± 0.019	0.017 ± 0.006	$1.4\times$	0.583

A.3 CHAMELEONDATASET$_{\text{tr}}$

Our ChameleonDataset$_{\text{tr}}$ contains diverse foreground object categories (2,000) and stylized backgrounds (1,171), as summarized in Tab. 6. This diversity is enabled by our reverse pipeline (Fig. 6), which starts from curated real-image composites I_c rather than relying on multiple style-specific LoRAs or stylization model zoos (An et al., 2021; Hu et al., 2024), which are typically limited to predefined style categories. As a result, the dataset covers a broad range of visual domains, styles, and compositional scenarios, enabling the model to learn diverse cross-domain compositing patterns from data.

Table 6: **ChameleonDataset$_{\text{tr}}$ construction specification.**

Component	Specification	Details
Categories	Total categories	2,000 object categories spanning diverse semantic domains
Category domains	Main groups	Human, Animal, Plant, Food, Clothing/Accessories, Household/Object, Vehicle, Building/Structure, Nature/Scene
Category examples	Human-related	person, man, woman, child, baby, face, hand, hair, eye
	Food-related	fruit, vegetable, cake, bread, pizza, burger, drink
	Animal-related	dog, cat, horse, bird, fish, butterfly, elephant
	Vehicle-related	car, bus, truck, bicycle, motorcycle, train, airplane
	Clothing/Accessory-related	dress, t-shirt, jacket, coat, hat, shoes, bag, necklace
Styles	Total styles	1,171 artistic and illustrative styles
Style domains	Art movements	impressionism, baroque, surrealism, cubism
	Eastern media	woodblock print, ink wash painting
	Western media	oil painting, watercolor, pastel painting
	Non-painting styles	sculpture, pencil sketch, engraving
	Illustrative styles	cartoon, comics, kids' drawing
Reference augmentation	Viewpoint	front-right quarter view (40°) and related azimuth variations
	Distance	close-up, medium-shot, and wide-shot compositions
	Elevation	eye-level, high-angle, and low-angle views
	Resolution	multi-resolution generation at 512, 768, and 1024
Buckets	Resolution buckets	896×1152, 384×640, 576×960, 672×864, 1024×1024, 448×576, 768×1280, 320×704, 512×512, 640×1536, 768×768, 480×1056, 256×768, 384×1152, 640×1408, 320×768, 480×1152, 512×1536

A.4 Dataset Construction Statistics

Tab. 7 reports stage-wise statistics for our reverse data construction pipeline (Fig. 2).

Table 7: **Stage-wise statistics of ChameleonDataset$_{\text{tr}}$ construction.**

Stage	Count
Raw composite images	78K
(1) Object labels	139K
(2a) SAM-kept labels	67K
(2b) Segmented objects	226K
(3) Filtered samples	185K
(4) Generated references	183K
(5) Final samples	199K

During (1) object label generation, Qwen3-VL proposes up to three candidates per image (e.g., "`Lion, Book, Desk`") with confidence above 0.85, yielding 1.79 labels per image on average. This cap prevents scene-level over-representation, as multiple foreground objects from the same I_c share the same scene context and background style. In (2) object segmentation, a SAM 3 threshold of 0.5, chosen from manual inspection of 500 samples, removes 52% of the labels. The retained labels can correspond to multiple object instances, resulting in 226K segmented object candidates. In (3) data filtering, 82.1% of the candidates are retained using a keep score ≥ 70 and recoverability ≥ 0.35, with the filtering criteria detailed in Appendix H.2. In (4) reference generation, CLIP-I filtering removes approximately 1% of generated references with identity changes. Finally, in (5) appearance variation, we perturb camera parameters (e.g., azimuth, elevation) to obtain I_f'' with diverse poses and lighting. This encourages the model to adapt the foreground pose and lighting toward those of I_c, rather than naively preserving the perturbed input. For instance, when I_f'' depicts a lion from a top view while I_c shows it from the side, the supervision encourages adaptation toward the side-view target. This stage increases the dataset by 12%, after which CLIP-I filtering removes approximately 3% relative to the original dataset size, resulting in a net increase of 9%.

A.5 CHAMELEONDATASET$_{ev}$

Existing cross-domain compositing benchmarks provide limited coverage of diverse styles and compositional scenarios. TF-ICON (Lu et al., 2023b) organizes its evaluation data into four coarse domains: real images, paintings, sketches, and cartoons. Although AI-Composer (Li et al., 2025) includes a broader range of styles, it repeatedly reuses foregrounds and backgrounds, such that a larger number of samples does not necessarily correspond to equally diverse evaluation scenarios. Moreover, as it builds upon TF-ICON and Magic Insert (Ruiz et al., 2025), its evaluation data partially overlap with existing benchmarks. In contrast, ChameleonDataset$_{ev}$ prioritizes evaluation quality by independently curating diverse, compositionally plausible, and challenging scenarios. Rather than recombining foregrounds and backgrounds from a fixed asset pool, we select the object, background, and insertion placement for each scene, ensuring that each sample represents a meaningful object–scene interaction. As shown in Fig. 10, our benchmark evaluates both object–background composition, through scenes such as stair climbing, reflective surfaces, and lighting-aware shadows, and a broad spectrum of visual styles. These include traditional East Asian art, classical and impressionist painting, watercolor and expressionism, pencil and ink drawing, vector graphics, pixel art, 3D rendering, stained glass, mosaics, animation, storybook illustration, and photographic styles. In total, the benchmark spans roughly 25 style families and diverse object categories, including humans, animals, vehicles, buildings, text, food, and plants. Additional foreground objects and stylized backgrounds appear in the qualitative results in Figs. 14 , 15, and 16. All samples are built from publicly available, license-free images (Pixabay, 2026; Unsplash, 2026; Pexels, 2026).



Figure 10: **Examples of ChameleonDataset$_{ev}$.** Our benchmark includes challenging compositional scenarios, including (a) stair-climbing, (b) lighting-aware shadows, and (c) reflective surfaces, as well as diverse styles such as (d) motion blur and (e) mosaic-style artwork.

B Preliminaries, Derivation, and Algorithm

B.1 Preliminaries: Diffusion Transformer

Our framework (Sec. 3.4) builds on FLUX.1-dev (Black Forest Labs, 2024), operating in the latent space. A VAE (Kingma & Welling, 2013) encodes an image into a latent $Z_0 \in \mathbb{R}^{H\times W\times C}$, and a T5 encoder (Raffel et al., 2020) produces text embeddings $Z_c \in \mathbb{R}^{L\times D}$. The model is trained under a flow matching formulation (Liu et al., 2023), which predicts the velocity field along a linear interpolation between Gaussian noise $Z_1 \sim \mathcal{N}(0, I)$ and the target latent Z_0. Specifically, we consider the path $Z_t = tZ_0 + (1-t)Z_1$, where $t \in [0, 1]$, and the target velocity $v^* = Z_0 - Z_1$. The training objective is:

$$\mathcal{L}_{\text{flow}} = \mathbb{E}_{t,Z_0,Z_1}\left[\|v_\theta(Z_t, Z_c, t) - v^*\|_2^2\right]. \tag{2}$$

For brevity, Fig. 6 depicts $\mathcal{L}_{\text{flow}}$ between the decoded output $\hat{X} = D(\hat{Z}_0)$ and I_c. In practice, $\mathcal{L}_{\text{flow}}$ is computed in the latent space, where the target latent is $Z_0 = \mathcal{E}(I_c)$.

B.2 Theoretical Analysis of Signed Style-Content Graph Learning

Under an idealized style-content factorization, we establish that minimizing the SSCGL objective yields factor-selective style and content representations.

Factor-selective disentanglement. Let $\mathcal{C}$ and $\mathcal{S}$ denote the sets of content and style factors, respectively, and denote a sample with content $c \in \mathcal{C}$ and style $s \in \mathcal{S}$ by $x_{c,s}$, with $\bar{z}^{\text{S}}_{c,s}$ and $z^{\text{C}}_{c,s,m}$ denoting its style representation and m-th patch-wise content representation, respectively. We consider the full Cartesian product $\mathcal{X} = \mathcal{C} \times \mathcal{S}$, where every content factor is observed with every style factor. As defined in Sec. 3, E_S connects samples with the same style and different content, whereas E_C connects samples with the same content and different styles. In the style space, E_S forms positive edges and E_C forms negative edges, while the signs are reversed in the content space.

These signed relations are optimized through the SSCGL objective in Eq. (1). For the style branch,

$$\mathcal{L}_{\text{S}} = -\sum_{i,j} W^{+}_{\text{S},ij} \left\langle \bar{z}^{\text{S}}_i, \bar{z}^{\text{S}}_j \right\rangle + \lambda \sum_{i,j} W^{-}_{\text{S},ij} \left\langle \bar{z}^{\text{S}}_i, \bar{z}^{\text{S}}_j \right\rangle^2.$$

Under the unit-ball constraint in Sec. 3.3, the Cauchy–Schwarz inequality gives $\langle \bar{z}^{\text{S}}_i, \bar{z}^{\text{S}}_j \rangle \le 1$. Combined with the unit-sum normalization of W^{+}_{S}, this yields

$$-\sum_{i,j} W^{+}_{\text{S},ij} \left\langle \bar{z}^{\text{S}}_i, \bar{z}^{\text{S}}_j \right\rangle \ge -\sum_{i,j} W^{+}_{\text{S},ij} = -1. \tag{3}$$

Since the weighted squared negative-edge term is non-negative,

$$\mathcal{L}_{\text{S}} \ge -1. \tag{4}$$

The same argument applies independently to each patch in $\mathcal{L}_{\text{C}}$. Since the content objective averages over M patches, we obtain

$$\mathcal{L}_{\text{C}} \ge -1, \qquad \mathcal{L}_{\text{SSCGL}} = \mathcal{L}_{\text{S}} + \mathcal{L}_{\text{C}} \ge -2. \tag{5}$$

Proposition 1 (Factor-selective optimum). Under the idealized full Cartesian setting described above, suppose that $\lambda > 0$, all active edge weights in $W^{\pm}_{\text{S}}$ and $W^{\pm}_{\text{C}}$ are strictly positive, and the style and content embedding spaces have sufficient dimension to represent the factor-selective solution. Then the lower bound $\mathcal{L}_{\text{SSCGL}} = -2$ in Eq. (5) is attained if and only if the style and content representations are factor-selective. Specifically, for each style s, all samples sharing that style map to a common unit prototype u_s, independent of content, while distinct style prototypes are pairwise orthogonal:

$$\bar{z}^{\text{S}}_{c,s} = u_s \quad \text{for all } c \in \mathcal{C}, \qquad \|u_s\|_2 = 1, \qquad \langle u_s, u_{s'} \rangle = 0 \quad (s \ne s'). \tag{6}$$

Likewise, for each content c and patch m, all style variations map to a common unit prototype $v_{c,m}$, independent of style, while distinct content prototypes are pairwise orthogonal:

$$z^{\text{C}}_{c,s,m} = v_{c,m} \quad \text{for all } s \in \mathcal{S}, \qquad \|v_{c,m}\|_2 = 1, \qquad \langle v_{c,m}, v_{c',m} \rangle = 0 \quad (c \ne c'). \tag{7}$$

Consequently, under these assumptions, every global optimum satisfies, for all $c, c' \in \mathcal{C}$, $s, s' \in \mathcal{S}$, and each patch m,

$$\bar{z}^{\mathrm{S}}_{c,s} = \bar{z}^{\mathrm{S}}_{c',s}, \qquad z^{\mathrm{C}}_{c,s,m} = z^{\mathrm{C}}_{c,s',m}. \tag{8}$$

Thus, the style representation is invariant to content, while the content representation is invariant to style. *Proof.* For the only-if direction, suppose that the lower bound $\mathcal{L}_{\mathrm{SSCGL}} = -2$ is attained. Since $\mathcal{L}_{\mathrm{S}} \geq -1$ and $\mathcal{L}_{\mathrm{C}} \geq -1$, this requires $\mathcal{L}_{\mathrm{S}} = \mathcal{L}_{\mathrm{C}} = -1$. We first consider the style branch. Attaining $\mathcal{L}_{\mathrm{S}} = -1$ requires the positive-edge term to attain its lower bound -1 in Eq. (3), while the non-negative negative-edge term must vanish. Since all active weights in W^{+}_{S} are strictly positive and sum to one, equality in Eq. (3) requires

$$\left\langle \bar{z}^{\mathrm{S}}_{i}, \bar{z}^{\mathrm{S}}_{j} \right\rangle = 1 \qquad \text{for every active } (i,j) \in E_S. \tag{9}$$

Under the unit-ball constraint, this equality is possible only when $\bar{z}^{\mathrm{S}}_{i}$ and $\bar{z}^{\mathrm{S}}_{j}$ are identical unit vectors. Since E_S connects samples sharing the same style, all samples with style s therefore share a common unit prototype u_s, yielding $\bar{z}^{\mathrm{S}}_{c,s} = u_s$ for all $c \in \mathcal{C}$.

The negative-edge term must simultaneously vanish. Since $\lambda > 0$ and all active weights in W^{-}_{S} are strictly positive, this requires

$$\left\langle \bar{z}^{\mathrm{S}}_{i}, \bar{z}^{\mathrm{S}}_{j} \right\rangle = 0 \qquad \text{for every active } (i,j) \in E_C. \tag{10}$$

Because E_C connects samples with the same content and different styles, the full Cartesian setting implies $\langle u_s, u_{s'} \rangle = 0$ for all $s \neq s'$. Thus, the style branch satisfies the factor-selective structure in Eq. (6).

The content branch follows symmetrically. For each patch m, positive E_C edges require $\langle z^{\mathrm{C}}_{i,m}, z^{\mathrm{C}}_{j,m} \rangle = 1$, so all style variations of the same content share a common unit prototype $v_{c,m}$. Likewise, the vanishing negative-edge term on E_S requires $\langle v_{c,m}, v_{c',m} \rangle = 0$ for $c \neq c'$. Thus, Eq. (7) holds for every patch m, completing the only-if direction.

For the if direction, suppose that Eqs. (6) and (7) hold. Under this factor-selective structure, positive edges connect samples that share the target factor and therefore map to the same unit prototype, whereas negative edges connect distinct target-factor prototypes, which are orthogonal. Hence, positive edges have inner product 1, while negative edges have inner product 0. By the unit-sum normalization of the positive edge weights, the style objective and each patch-wise content objective attain -1, while all negative-edge terms vanish. Therefore, $\mathcal{L}_{\mathrm{S}} = \mathcal{L}_{\mathrm{C}} = -1$ and $\mathcal{L}_{\mathrm{SSCGL}} = -2$. □

Why factor-specific weighting? The result above shows that the signed relation structure determines *which* factor should be preserved or suppressed. However, a binary edge sign alone cannot express *how strongly* a pairwise relation should be enforced. In real data, discrete factor labels capture only coarse pairwise relations. For example, Expressionism and Fauvism have different style labels but share visual attributes such as bold colors and expressive brushwork, whereas Fauvism and pixel art are substantially more distinct. Assigning identical negative weights to both pairs would therefore impose the same decorrelation strength despite their different degrees of stylistic similarity.

SSCGL resolves this by retaining the signed relation while modulating the strength of each pairwise constraint with factor-specific similarities. Let $a_{ij} = \langle \bar{z}^{S}_{i}, \bar{z}^{S}_{j} \rangle$. For a positive style edge,

$$\frac{\partial(-W^{+}_{S,ij} a_{ij})}{\partial a_{ij}} = -W^{+}_{S,ij} \propto -E_{S,ij} s_{ij}, \tag{11}$$

whereas for a negative style edge,

$$\frac{\partial(\lambda W^{-}_{S,ij} a^2_{ij})}{\partial a_{ij}} = 2\lambda W^{-}_{S,ij} a_{ij} \propto E_{C,ij}(1 - s_{ij}) a_{ij}. \tag{12}$$

Thus, on positive edges, a larger s_{ij} places stronger optimization pressure on increasing a_{ij}, pulling the pair more strongly toward the maximum similarity, $a_{ij} = 1$. Conversely, on negative edges, a smaller s_{ij} increases $1 - s_{ij}$, assigning a larger penalty to nonzero a_{ij} and thereby pushing the pair more strongly toward orthogonality. Hence, s and $1 - s$ calibrate the relative strengths of attraction

and decorrelation according to the underlying style similarity. The content weights W_C^+ and W_C^- play the symmetric role in the content space.

Taken together, the theoretical analysis above establishes the factor-selective role of the signed graph and clarifies the complementary role of factor-specific weighting. While the signed relations alone characterize the factor-selective solution under the idealized full Cartesian setting, practical settings with finite model capacity and non-ideal data call for finer control over the strength of individual relations. SSCGL therefore uses continuous factor similarities to refine the strength of each pairwise relation. Rather than treating all binary relations equally, the factor-specific weighting adaptively modulates each pairwise constraint according to its factor similarity, providing a more fine-grained learning signal for style-content disentanglement. We further verify this effect empirically in Tab. 14, where factor-specific weighting improves the disentanglement margin for both style and content over uniform weighting on the same signed graph.

B.3 Style Gap-Aware Anisotropic Modulation (SGAM): Full Formulation

Specifically, we map the diffusion timestep t to a sinusoidal embedding $\phi(t)$. Let S_{f} and S_{b} denote the Stage-1 ChameleonEncoder style representations of the foreground object and background image, respectively. We define the foreground–background style gap as $\Delta s = 1 - \cos(S_{\mathrm{f}}, S_{\mathrm{b}})$. We then encode Δs using the same sinusoidal encoding scheme as t, yielding $\psi(\Delta s)$. Since the foreground must adapt toward the background style while the background already represents the target style, the foreground branch is conditioned on both $\phi(t)$ and $\psi(\Delta s)$, whereas the background branch remains conditioned only on $\phi(t)$. The resulting features are linearly projected to obtain layer-wise modulation logits:

$$\begin{aligned} z_{\mathrm{f}}(t, \Delta s) &= H^{(\mathrm{f})}\,\mathrm{MLP}_{\mathrm{f}}([\phi(t),\, \psi(\Delta s)]) + \beta_0^{(\mathrm{f})} \in \mathbb{R}^L, \\ z_{\mathrm{b}}(t) &= H^{(\mathrm{b})}\,\mathrm{MLP}_{\mathrm{b}}(\phi(t)) + \beta_0^{(\mathrm{b})} \in \mathbb{R}^L, \end{aligned} \tag{13}$$

where $H^{(\mathrm{f})}, H^{(\mathrm{b})} \in \mathbb{R}^{L\times h}$ project the h-dimensional MLP features to the L transformer blocks, and $\beta_0^{(\mathrm{f})}, \beta_0^{(\mathrm{b})} \in \mathbb{R}^L$ are learned biases. Accordingly, $z_{\mathrm{f}}(t, \Delta s, \ell)$ and $z_{\mathrm{b}}(t, \ell)$ denote the foreground and background modulation logits for the ℓ-th block, respectively. These logits are converted to modulation coefficients via

$$\begin{aligned} \beta_{\mathrm{f}}(t, \Delta s, \ell) &= \sigma\big(z_{\mathrm{f}}(t, \Delta s, \ell)\big), \\ \beta_{\mathrm{b}}(t, \ell) &= \sigma\big(z_{\mathrm{b}}(t, \ell)\big), \end{aligned} \tag{14}$$

where $\sigma(\cdot)$ denotes the sigmoid function. Given a binary foreground mask $m(q) \in \{0, 1\}$, each query token q is assigned a modulation coefficient according to its spatial location:

$$\gamma(q, t, \Delta s, \ell) = \begin{cases} \beta_{\mathrm{f}}(t, \Delta s, \ell) & \text{if } m(q) = 1, \\ \beta_{\mathrm{b}}(t, \ell) & \text{if } m(q) = 0. \end{cases} \tag{15}$$

The attention keys at each layer consist of text tokens, condition tokens $(C_T(I_{\mathrm{f}}), S_T(I_{\mathrm{b}}))$, and latent tokens. We apply the resulting attention bias exclusively to the subset corresponding to $S_T(I_{\mathrm{b}})$, whose key indices are denoted by $\mathcal{K}_{\mathrm{b}}$:

$$B_{q,k}^{(\ell)}(t, \Delta s) = \begin{cases} \gamma(q, t, \Delta s, \ell) & \text{if } k \in \mathcal{K}_{\mathrm{b}}, \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

where $B^{(\ell)}(t, \Delta s)$ denotes the resulting attention bias matrix. The final attention at layer ℓ is computed as

$$\tilde{A}^{(\ell)} = \mathrm{softmax}\left(\frac{Q^{(\ell)} K^{(\ell)\top}}{\sqrt{d_k}} + B^{(\ell)}(t, \Delta s)\right), \tag{17}$$

where $Q^{(\ell)}$ and $K^{(\ell)}$ denote the query and key matrices, respectively, and d_k is the key dimension. This formulation enables adaptive modulation of attention to the background style tokens according to the style gap, spatial region, and diffusion timestep, as illustrated in Fig. 12.

B.4 Batch Construction for SSCGL

SSCGL operates on factor-specific relations induced within each mini-batch. We therefore construct each batch to ensure that both style-sharing and content-sharing relations are consistently available at every training step, rather than relying on incidental matches from random sampling. Specifically, each batch contains 32 style tuples and 32 content tuples, where each tuple is constructed to introduce complementary style and content relations for building $\mathcal{E}_S$ and $\mathcal{E}_C$.

Tuple construction. A style tuple starts from an anchor-positive pair that shares style but differs in content. We then add samples that share content with either the anchor or positive while differing in style, together with auxiliary samples that differ from the anchor in both factors. For example, given dog/Gogh and cat/Gogh as the anchor-positive pair, dog/Monet and cat/Picasso provide content-matched samples with different styles. A content tuple follows the same construction with the roles of style and content exchanged. This ensures that both style-sharing and content-sharing relations are explicitly introduced into each training batch.

Batch-wide graph construction. The tuples determine the composition of each batch, while the graph itself is constructed over the entire batch. After merging all tuples and removing duplicate samples, we connect every pair that satisfies the corresponding style or content relation, regardless of which tuple the samples originate from. This allows samples from different tuples, including auxiliary samples, to form additional edges and enables SSCGL to exploit all valid within-batch relations rather than only the pairs explicitly introduced during tuple construction.

In practice, each batch contains 1,655 unique images on average and yields 2,094 style edges and 1,591 content edges. Among these, 67% of the style edges and 57% of the content edges are formed across different tuples, demonstrating that batch-wide graph construction substantially expands the relational supervision available to SSCGL.

Algorithm 1 Batch Construction for SSCGL

Require: Dataset $\mathcal{D}$ with style/content labels (ϕ_S, ϕ_C); tuples per factor $T = 32$; matched samples per seed $K_m = 4$; auxiliary samples $K_a = 16$
Ensure: Batch $\mathcal{B}$, relation matrices $\mathcal{E}_S, \mathcal{E}_C$, pairwise similarities s, c

function TUPLE($\mathcal{D}, A, B$)
  $I_a \sim \mathcal{U}(\mathcal{D})$
  $(\alpha, \beta) \leftarrow \big(\phi_A(I_a),\ \phi_B(I_a)\big)$
  $I_p \sim \mathcal{U}\big(\{I \in \mathcal{D} \setminus \{I_a\} \mid \phi_A(I) = \alpha,\ \phi_B(I) \neq \beta\}\big)$
  $\beta' \leftarrow \phi_B(I_p)$
  $\mathcal{M}_a \sim \mathcal{U}_{K_m}\big(\{I \in \mathcal{D} \mid \phi_A(I) \neq \alpha,\ \phi_B(I) = \beta\}\big)$
  $\mathcal{M}_p \sim \mathcal{U}_{K_m}\big(\{I \in \mathcal{D} \mid \phi_A(I) \neq \alpha,\ \phi_B(I) = \beta'\}\big)$
  $\mathcal{R} \sim \mathcal{U}_{K_a}\big(\{I \in \mathcal{D} \mid \phi_A(I) \neq \alpha,\ \phi_B(I) \notin \{\beta, \beta'\}\}\big)$
  return $\{I_a, I_p\} \cup \mathcal{M}_a \cup \mathcal{M}_p \cup \mathcal{R}$
end function

$\mathcal{T}_S \leftarrow \bigcup_{t=1}^{T}$ TUPLE$(\mathcal{D}, S, C)$
$\mathcal{T}_C \leftarrow \bigcup_{t=1}^{T}$ TUPLE$(\mathcal{D}, C, S)$
$\mathcal{B} \leftarrow \mathrm{Unique}\big(\mathcal{T}_S \cup \mathcal{T}_C\big)$

for $i, j \in \mathcal{B}$ **do**
  $\mathcal{E}_{S,ij} \leftarrow 1[i \neq j \ \wedge\ \phi_S(i) = \phi_S(j) \ \wedge\ \phi_C(i) \neq \phi_C(j)]$
  $\mathcal{E}_{C,ij} \leftarrow 1[i \neq j \ \wedge\ \phi_C(i) = \phi_C(j) \ \wedge\ \phi_S(i) \neq \phi_S(j)]$
end for

Compute $s, c \in \mathbb{R}^{|\mathcal{B}| \times |\mathcal{B}|}$ from frozen DINOv3 features
return $\mathcal{B}, \mathcal{E}_S, \mathcal{E}_C, s, c$

C Implementation Details

C.1 Training Details

Stage 1 uses a frozen DINOv3 ViT-L/16 backbone with two-layer MLP projection heads (Linear-ReLU-Linear) for style and content, where the input, hidden, and output dimensions are all 1024. Our proposed SSCGL is applied on top of the DINOv3 features through these style and content heads, which take features from layers [12,14] and [18,20], respectively, averaged across layers. Each batch is constructed from 32 style tuples and 32 content tuples so that both E_S and E_C are non-empty. The style similarity s is computed from Gram statistics of DINOv3 layers [12,14], and the content similarity c from patch-wise cosine similarity at layers [18,20]. Both are clipped to $[0, 1]$, and self-pairs are excluded. Projected representations are constrained to the unit ball by norm clipping. We optimize with AdamW (learning rate $1\mathrm{e}{-3}$, weight decay $1\mathrm{e}{-4}$) for 40 epochs on a single H100 GPU, taking approximately 1 hour.

Stage 2 fine-tunes FLUX.1-dev (Black Forest Labs, 2024) on ChameleonDataset$_{\mathrm{tr}}$ using bucketed resolutions (512, 768, 1024). We use LoRA (rank 16) with Prodigy (learning rate 1.0) and jointly optimize lightweight modules including the projection layers and SGAM. The projection layers map 1024-dimensional features to 3072. Foreground and background DINO features are extracted from layers [18,20] and [12,14], respectively. SGAM consists of two three-layer MLP branches for foreground and background. The foreground branch takes the concatenation of a 256-dimensional sinusoidal time embedding and a 128-dimensional sinusoidal style-gap embedding, resulting in a $384 \to 128 \to 128 \to 19$ architecture, whereas the background branch remains conditioned only on the 256-dimensional time embedding with a $256 \to 128 \to 128 \to 19$ architecture. Both branches produce block-wise modulation logits for the 19 transformer blocks. Training uses bf16 mixed precision with a total batch size of 4 on 4$\times$H100 GPUs. We train for 40000 iterations. Total training takes approximately 22 hours.

C.2 Compute and Efficiency Analysis

Table 8: **Trainable parameter budget.** Percentages are relative to the frozen DiT backbone. LoRA adapters account for most trainable parameters, while SGAM introduces only a negligible overhead.

Module	Params	Backbone ratio
LoRA adapters	18.7M	0.16%
SGAM	0.1M	$< 0.01\%$
Projection heads	6.3M	0.05%
Total trainable	25.1M	0.21%

Table 9: **Inference efficiency.** (a) We measure the overhead of dual-anchor DINO conditioning during DiT denoising. Base denotes the standard DiT token sequence, consisting of image latent and text tokens, whereas +DINO additionally appends the 392 dual-anchor condition tokens of Fig. 6, composed of 196 background style tokens $S_T(I_b)$ and 196 foreground content tokens $C_T(I_f)$. Adding these tokens introduces only a modest latency increase (Δ). (b) We further report the end-to-end runtime and peak memory of the full model. All measurements use FP16 on a single H100 80GB with a 28-step sampler.

(a) DiT denoising overhead

	Sequence length		Per-step latency			28-step runtime		
Resolution	Base	+DINO	Base	+DINO	Δ	Base	+DINO	Δ
512×512	1,280	1,672	0.281 s	0.289 s	+0.008 s (+2.8%)	7.86 s	8.10 s	+0.24 s (+3.1%)
1024×1024	4,352	4,744	1.052 s	1.099 s	+0.047 s (+4.5%)	29.45 s	30.77 s	+1.32 s (+4.5%)

(b) End-to-end inference cost

Resolution	Runtime	Peak memory
512×512	10.40 s	36.2 GB
768×768	19.83 s	37.1 GB
1024×1024	37.95 s	38.5 GB

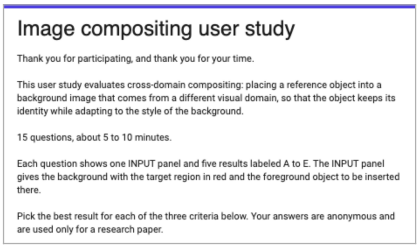
C.3 External Models and Licenses

All external models and assets used in this work, including FLUX.1-dev (Black Forest Labs, 2024), Qwen (Bai et al., 2025; Wu et al., 2025), SAM3 (Carion et al., 2026), and DINOv3 (Siméoni et al., 2025), are publicly available and properly credited in the main paper. In particular, FLUX.1-dev is released under the FLUX.1 [dev] Non-Commercial License, and our use is restricted to non-commercial research purposes.

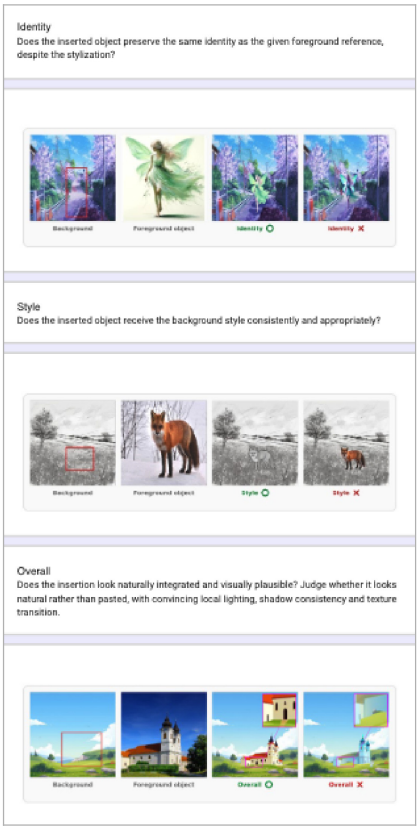
C.4 User Study

We conduct the user study with 50 participants who have AI-related research backgrounds. No monetary compensation is provided. The study includes 30 foreground-background cases selected from ChameleonDataset$_{\text{ev}}$, covering 30 distinct foreground objects and 27 background style families. Whenever possible, we select at most one case from each style family to avoid overrepresenting any particular style. The complete set of foreground-style combinations is provided in Tab. 10.

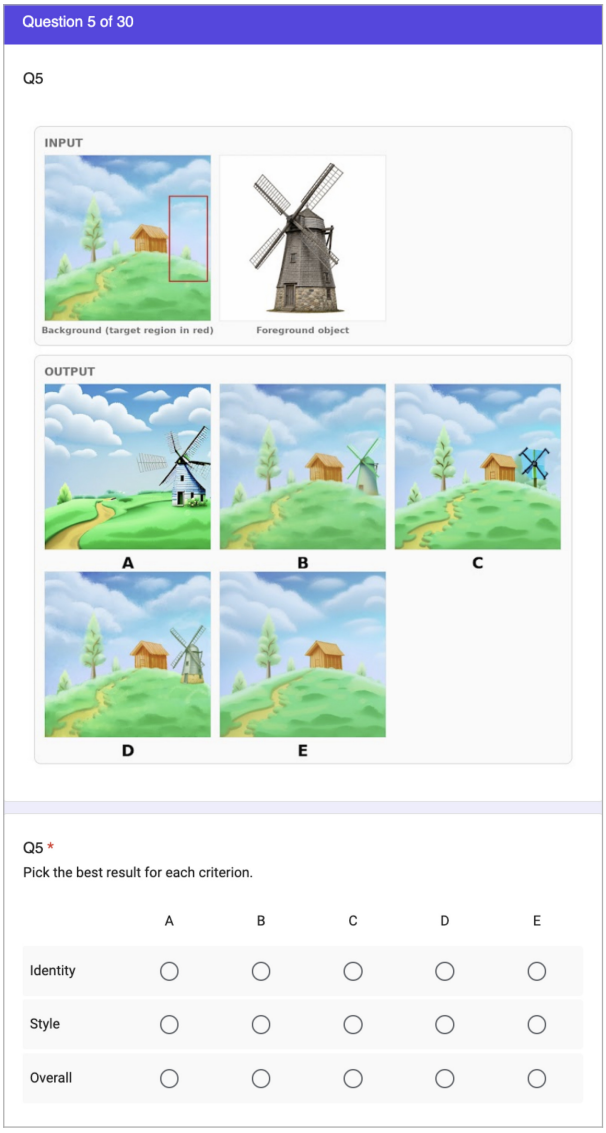
In each trial, participants view five anonymized results (A–E) side by side, together with the foreground object and the background image with the target region outlined, and select the single best result for each criterion. The ordering of the five anonymized results is randomized for each trial to reduce position bias. The study interface is shown in Fig. 11. The evaluation criteria follow the same definitions as the VLM rubric in Appendix H. The overall criterion evaluates whether the composition and harmonization are naturally integrated and visually plausible. Win rates for all five methods are reported in Tab. 4.



(a) Task instructions



(b) Evaluation criteria



(c) Sample question and rating interface

Figure 11: User study interface. Participants compare five anonymized results (A–E) and select the best result for each evaluation criterion.

Table 10: Foreground objects and background style families used in the user study.

#	Foreground	Background style	#	Foreground	Background style
1	Motorcycle	Flat/Cartoon	16	Woman in kimono	Ukiyo-e
2	Whale	Neon Line Art	17	Pickup truck	Pixel Art
3	Jellyfish	Line Art	18	Golden retriever	Colored Pencil
4	Scooter	Etching	19	Popsicle	3D Render
5	Windmill	Painterly	20	Pumpkin	Surrealism
6	Rabbit	Anime/Cel	21	Polar bear	Watercolor
7	Cart	Painterly	22	Running courier	Long Exposure
8	Sushi	Flat/Cartoon	23	Astronaut	Astrophotography
9	Person	Backlit Silhouette	24	Lobster	Mosaic
10	Robot	Pop Art	25	Calligraphy text	Floral Pattern
11	Butterfly	Stained Glass	26	Sports car	Vaporwave
12	Mammoth	Low-Poly	27	Herd of deer	Halftone
13	Easel with canvas	Art Nouveau	28	Hot air balloon	Oil Painting
14	Desk globe	Flat/Cartoon	29	Woman portrait	Marbling
15	Dragonfly	Monochrome Photo	30	Koi fish	Ink Splatter

D Ablation Studies

D.1 Encoder

In this section, we ablate various semantic encoders, including CLIP (Radford et al., 2021), SigLIP2 (Tschannen et al., 2025), and CSD (Somepalli et al., 2024), to analyze the choice of DINOv3 in Tab. 11. Although CSD is trained for understanding and extracting style descriptors from images, its representations still exhibit noticeable style-content entanglement. In contrast, the stronger semantic representations of DINOv3 provide a more favorable starting point for disentanglement prior to training the projection heads.

We further analyze the effect of different DINOv3 layers in Tab. 12. The results justify our selection of separate layer ranges for the style and content heads. Larger margins between style and content similarity indicate representations that are easier to disentangle, which justifies our selection of different layer ranges for the style and content heads in ChameleonEncoder.

Table 11: **Why DINOv3?** Disentanglement margin $\Delta = \langle a, p\rangle - \langle a, hn\rangle$ on dual-anchor test splits (200 samples each), comparing frozen off-the-shelf encoders without any projection head. Here, a denotes the anchor feature, p the positive feature, and hn the hard-negative feature. On the style-paired split, p shares the same style as a, while hn shares the same content but a different style. On the content-paired split, the roles are reversed. A larger Δ indicates better separation of the target attribute from the confounding one. DINOv3 achieves the strongest overall disentanglement margin, particularly on the content-paired split, indicating that its representation provides the most suitable foundation for applying SSCGL.

	Style-paired split			Content-paired split		
Encoder	$\langle a, p\rangle \uparrow$	$\langle a, hn\rangle \downarrow$	$\Delta \uparrow$	$\langle a, p\rangle \uparrow$	$\langle a, hn\rangle \downarrow$	$\Delta \uparrow$
CLIP ViT-L/14 (Radford et al., 2021)	0.848	0.950	−0.102	**0.949**	0.844	+0.105
SigLIP2 L/16-256 (Tschannen et al., 2025)	0.755	0.923	−0.168	0.927	0.757	+0.170
CSD ViT-L (Somepalli et al., 2024)	0.836	0.906	−0.071	0.906	0.838	+0.068
DINOv3 (Siméoni et al., 2025)	**0.871**	**0.903**	**-0.032**	0.916	**0.660**	**+0.257**

Table 12: **Layer selection for the style and content heads.** Disentanglement margin $\Delta = \langle a, p\rangle - \langle a, hn\rangle$ on dual-anchor test splits (200 samples each), comparing different layer groups from frozen DINOv3 ViT-L/16. Here, a denotes the anchor feature, p the positive feature, and hn the hard-negative feature. On the style-paired split, p shares the same style as a, while hn shares the same content but a different style. On the content-paired split, the roles are reversed. Since all layer groups produce negative margins on the style-paired split, we select the mid-level layers (12–14), which yield the smallest negative margin. In contrast, late layers (18–20) provide substantially stronger content separation on the content-paired split. Accordingly, we use layers 12–14 for the style head E_s and layers 18–20 for the content head E_c in SSCGL.

	Style-paired split			Content-paired split		
DINOv3 layers	$\langle a, p\rangle \uparrow$	$\langle a, hn\rangle \downarrow$	$\Delta \uparrow$	$\langle a, p\rangle \uparrow$	$\langle a, hn\rangle \downarrow$	$\Delta \uparrow$
12–14 (mid, $\rightarrow E_s$)	**0.874**	**0.902**	**−0.028**	0.903	0.879	+0.024
18–20 (late, $\rightarrow E_c$)	0.655	0.911	−0.257	**0.919**	**0.672**	**+0.247**

Table 13: **Ablation on DINOv3 layer combinations.** We compare different style/content layer combinations for Stage 1 disentanglement and Stage 2 downstream compositing performance.

	Disentanglement		Compositing				
Style / Content layers	Style $\Delta \uparrow$	Content $\Delta \uparrow$	CLIP-I $\uparrow$	CSD $\uparrow$	CLIP-T $\uparrow$	AES $\uparrow$	LPIPS $\downarrow$
18–20 / 12–14 (swapped)	0.8851	0.9269	0.8398	0.4425	0.1985	6.6708	0.4921
24 / 24 (no selection)	0.2438	0.6795	0.8104	0.3981	0.1987	6.6540	0.5407
12–14 / 18–20 (Ours)	**0.9272**	**0.9481**	**0.8425**	**0.4629**	**0.2033**	**6.6946**	**0.4651**

Table 14: **Effect of factor-specific weighting in SSCGL.** We report the disentanglement margin $\Delta = \langle a, p \rangle - \langle a, h_n \rangle$ (Tab. 11) of the trained ChameleonEncoder heads on the dual-anchor test splits (400 samples each). Uniform replaces the factor-specific similarities with constant weights on the same signed graph. By modulating the strength of each pairwise constraint according to its factor similarity, factor-specific weighting improves the margin in both spaces.

Edge weighting	Style $\Delta \uparrow$	Content $\Delta \uparrow$
Uniform (signed only)	0.8966	0.9358
Factor-specific (Ours)	**0.9272**	**0.9481**
Gain	+0.0306	+0.0123

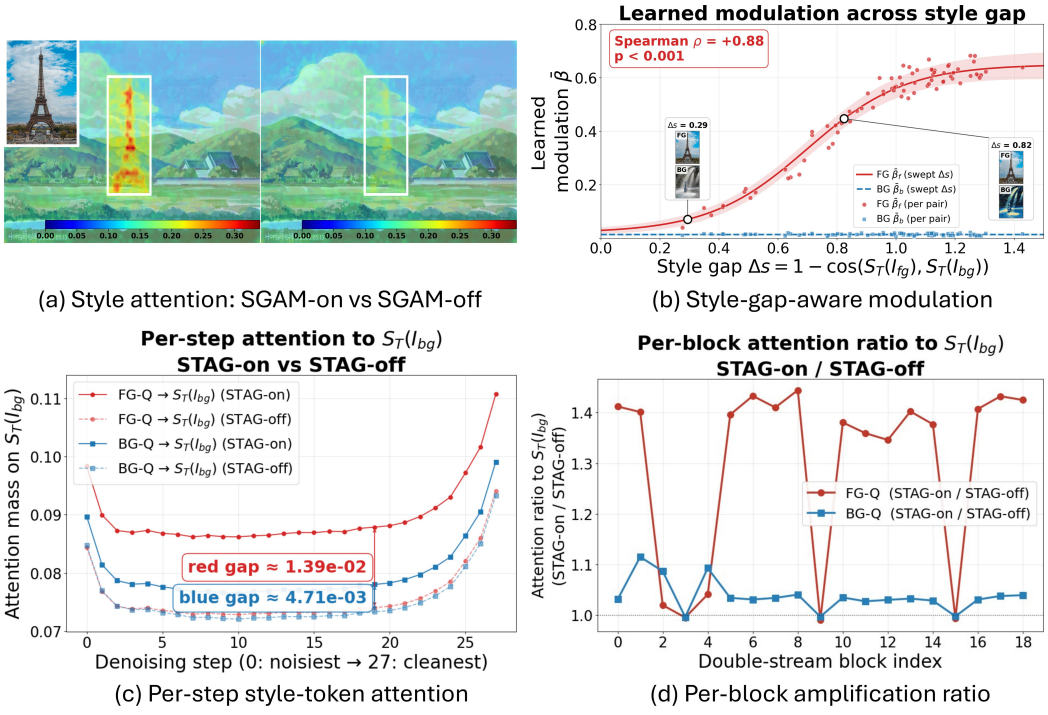
D.2 SGAM

We ablate SGAM by comparing models with and without SGAM in Fig. 12. In (a), we visualize attention maps to show that SGAM effectively concentrates style attention on the foreground object region (Eiffel Tower). Although the spatial assignment is derived from a coarse binary bounding-box mask, the modulation remains spatially aligned with the object, achieving the intended foreground-focused style injection.

Moreover, (b) analyzes the style-gap-aware behavior of SGAM by measuring the learned foreground modulation across different foreground–background style gaps computed in the disentangled style space of ChameleonEncoder. We fix the foreground and vary the background from photorealistic images to diverse stylized domains, covering a broad range of foreground–background style gaps. The learned modulation shows a strong positive Spearman correlation with the style gap ($\rho = 0.88$, $p < 0.001$), becoming stronger as the foreground and background grow more stylistically different. This trend indicates that SGAM adaptively regulates foreground style injection according to the style gap.

In addition, (c) visualizes the timestep-adaptive behavior of SGAM across the diffusion process. We observe that stronger stylization emerges at later timesteps after the coarse structure is formed, consistent with the temporal tendency observed in diffusion-based style transfer (Hu et al., 2024). Rather than imposing this tendency through a fixed temporal schedule, SGAM learns timestep-dependent modulation tailored to cross-domain compositing, jointly balancing foreground identity preservation and stylization.

Finally, (d) shows the per-block modulation behavior. Rather than remaining uniformly active or inactive across all blocks, the modulation dynamically strengthens or suppresses style injection depending on the transformer block, indicating stable and non-collapsed modulation behavior. We also provide a qualitative component-wise ablation in Fig. 13, where adding SGAM yields more seamless foreground–background integration.



(a) Style attention: SGAM-on vs SGAM-off

(b) Style-gap-aware modulation

(c) Per-step style-token attention

(d) Per-block amplification ratio

Figure 12: **Analysis of Style Gap-Aware Anisotropic Modulation (SGAM).** (a) Attention from the foreground query (white box) to $S_T(I_b)$ comparing SGAM-on and SGAM-off. SGAM concentrates style attention on the foreground. (b) Foreground modulation strength increases as the foreground–background style gap becomes larger. (c) SGAM learns timestep-adaptive style injection, with stronger stylization at later denoising steps. (d) Per-block amplification ratios show that SGAM strengthens and suppresses style injection block-by-block without collapse.

We further examine the necessity of SGAM's adaptive style control by replacing each of its key adaptive components with a simpler alternative in Tab. 15. *Masking* uses fixed spatial gates set to 1 and 0 for the foreground and background, respectively; *Fixed schedule* replaces the learned temporal modulation with a predefined cosine schedule over diffusion timesteps; and *Scalar gating* uses a single global gate shared across all spatial regions, timesteps, and transformer blocks. We also replace the learned style gap Δs with a pixel-level statistic (*Pixel-stat Gap*), computed as the distance between the channel-wise mean and standard deviation of I_f and I_b in LAB space following classical color transfer (Reinhard et al., 2001), while keeping the SGAM architecture unchanged.

Masking and fixed scheduling improve style consistency but reduce identity preservation. Meanwhile, the scalar gate changes only marginally from 0.500 to 0.518 during training, indicating limited adaptation of style injection. Replacing the learned style gap with a simple pixel-level statistic also degrades performance, supporting the benefit of measuring the foreground–background style gap in the disentangled style space of ChameleonEncoder. In contrast, SGAM achieves the best performance across all metrics, demonstrating the effectiveness of adaptive style injection.

Table 15: **Ablation of style modulation strategies on ChameleonDataset$_{\text{ev}}$.** Comparison of SGAM with simpler modulation strategies and style-gap formulations under the same training and evaluation settings.

Method	CLIP-I ↑	CSD ↑	CLIP-T ↑	AES ↑	LPIPS ↓
Masking	0.8258	0.4515	0.1945	6.5996	0.4896
Fixed schedule	0.8285	0.4476	0.1942	6.5395	0.4845
Scalar gating	0.8391	0.4394	0.1961	6.5563	0.4815
Pixel-stat Gap	0.8210	0.4549	0.1922	6.6621	0.5391
SGAM (Ours)	**0.8425**	**0.4629**	**0.2033**	**6.6946**	**0.4651**

D.3 Qualitative Component-wise Ablation

Finally, we provide a qualitative component-wise ablation in Fig. 13, progressively adding ChameleonDataset$_{\text{tr}}$, SSCGL, and SGAM. The results visually complement the quantitative component-wise ablation in Tab. 3.

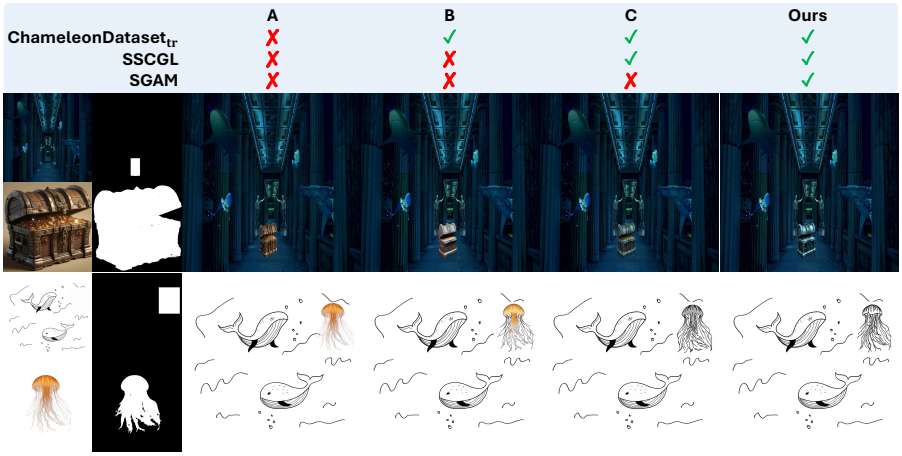


Figure 13: **Qualitative ablation of training data, SSCGL, and SGAM.** (A) Training on the in-domain DreamFuse data without our proposed modules learns object placement but largely ignores the surrounding context. (B) Training on ChameleonDataset$_{\text{tr}}$ encourages adaptation to low-level background appearance cues, such as texture, but does not yet achieve consistent style matching. (C) SSCGL improves background-style consistency through disentangled style conditioning. Our full model uses SGAM to adaptively regulate style injection, yielding more coherent foreground-background integration.

E Inference Assumptions and Robustness

Our method takes a foreground image and a background image, while the target insertion region requires only coarse localization and can be specified using a user-drawn box. As shown in Tab. 16, the target localization mask can be replaced with a coarse user-drawn box with only minor performance changes. This robustness follows from how spatial localization is used in our model. Our mask-guided affine transformation uses the foreground and placement masks only to extract their axis-aligned bounding boxes. Let the foreground box and target placement box be $B_f = (x_f, y_f, w_f, h_f)$ and $B_t = (x_t, y_t, w_t, h_t)$, where (x, y) denotes the top-left coordinate and (w, h) denotes the box size. We compute the bbox-to-bbox affine transformation as

$$A = \begin{bmatrix} \frac{w_t}{w_f} & 0 & x_t - \frac{w_t}{w_f} x_f \\ 0 & \frac{h_t}{h_f} & y_t - \frac{h_t}{h_f} y_f \\ 0 & 0 & 1 \end{bmatrix}. \tag{18}$$

For a foreground positional index $\mathbf{p}_f = [x, y, 1]^\top$, the transformed index is given by $\mathbf{p}'_f = A\mathbf{p}_f$. The transformation is applied only to the foreground latent positional indices, while the foreground latent values remain unchanged. The masks M_f and M_{cp} are used only to obtain the bounding boxes and are not provided to the model as additional conditioning tokens or input channels. Accordingly, the transformation depends on the coarse spatial extent of the regions rather than their exact mask boundaries.

Table 16: **Robustness to coarse localization at inference.** Comparison between SAM 3 masks and coarse user-drawn boxes. IoU is measured against the SAM 3 masks (ref.) for M_f and M_{cp}, respectively.

Region at inference	IoU (M_f / M_{cp})	CLIP-I ↑	CSD ↑	LPIPS ↓	Composition ↑
SAM 3 mask	ref.	0.8425	0.4629	0.4651	2.11
User-drawn box	0.798 / 0.702	0.8482	0.4508	0.4673	2.10

For foreground extraction, both training and inference use the same off-the-shelf SAM 3 model. The resulting crops can be incomplete or contain residual background regions due to segmentation errors. During training, the model is therefore exposed to such imperfect foreground crops while being supervised to reconstruct the complete foreground appearance in the composite. To further evaluate robustness to these imperfections, we deliberately perturb the foreground inputs using partial occlusion, dilation and erosion of M_f, and JPEG compression. Across these settings, Chameleon maintains stable identity preservation and style consistency, with CLIP-I and CSD decreasing by less than 0.02. This holds up to 25% occlusion, 20-pixel mask dilation (IoU of 0.837 with the original SAM 3 mask), 30-pixel mask erosion (IoU of 0.648), and a JPEG compression ratio of 66×.

F Additional Quantitative Results

F.1 Full Quantitative Results

Beyond the three primary metrics reported in the main paper, we additionally report the Aes score (Schuhmann, 2022), which measures the overall aesthetic quality of the generated scene, and CLIP-T (Hessel et al., 2021), which evaluates alignment between the generated image and the text prompt, on both the AIComposer and TF-ICON benchmarks in Tab. 17 and Tab. 18, respectively. Moreover, we extend the quantitative comparison in Tab. 19 from cross-domain compositing methods to specialized painterly image harmonization methods (Cao et al., 2023; Lu et al., 2023a; Niu et al., 2024) and strong general-purpose image editing models. While our method operates under the null-prompt setting, these editors are additionally given detailed text prompts describing the intended composite, as detailed below. **Evaluation protocol for general-purpose editors.** Each editor receives two images and a text prompt structured as follows:

```
IMAGE 1: a cut-out of the '{obj}' to insert, shown on a plain white
background.
IMAGE 2: a background painting. A white rectangle marks where to place
the object, with its normalized box coordinates given in the prompt.
```

The foreground is the same SAM-extracted cut-out used for all other methods, rendered on a plain white background. The prompt then instructs the model to composite the object at the marked location and scale, harmonize its color, lighting, texture, and brush-stroke style to the background, remove the guide rectangle, and keep the rest of the background unchanged. For GPT-Image-2 (OpenAI, 2026) and Gemini-3-Pro-Image (Google DeepMind, 2025), the inputs are ordered as the foreground followed by the background. For Qwen-Image-Edit, we reverse the order and adopt its picture-indexed phrasing, as we observed that it otherwise ignores the scene and generates the object in isolation.

Table 17: **Quantitative comparison on the AIComposer benchmark.**

Method	Res	LPIPS ↓	CLIP-I ↑	CSD ↑	Aes ↑	CLIP-T ↑	Identity ↑	Style ↑	Composition ↑	Avg_total ↑
In-domain compositing										
Dreamfuse (ICCV'25)	1024	0.5290	0.8603	0.3941	6.9515	0.2134	2.35	2.53	2.05	6.93
SHINE (ICLR'26)	1024	0.6513	0.8377	0.3512	6.9160	0.1999	2.05	2.40	1.99	6.44
Two-stage compositing										
StyleSSP (CVPR'25) + Insert Anything (AAAI'26)	1024	0.4682	0.8438	**0.4898**	6.7638	0.2083	2.26	**2.83**	1.97	7.07
Cross-domain compositing										
TF-ICON (ICCV'23)	512	0.6640	0.7493	0.3536	6.4443	0.1736	1.56	2.45	1.81	5.82
TF-ICON (ICCV'23)	768	0.6539	0.7700	0.3464	6.4217	0.1809	1.67	2.40	1.84	5.91
TALE (MM'24)	512	0.6164	0.7830	0.4681	6.9477	0.1893	1.72	2.72	2.00	6.44
AIComposer (ICCV'25)	1024	0.4648	0.8575	0.4848	6.8638	0.2006	2.42	2.77	2.10	7.30
Ours	1024	**0.4556**	**0.8637**	0.4886	**7.0296**	**0.2188**	**2.60**	2.81	**2.23**	**7.64**

Table 18: **Quantitative comparison on the TF-ICON benchmark.**

Method	Res	LPIPS ↓	CLIP-I ↑	CSD ↑	Aes ↑	CLIP-T ↑	Identity ↑	Style ↑	Composition ↑	Avg_total ↑
In-domain compositing										
BLD (SIGGRAPH'23)	512	0.7540	0.7331	0.4619	6.6360	0.2959	2.04	2.83	2.11	6.98
Paint by Example (CVPR'23)	512	0.7501	0.7658	0.3138	6.7032	0.2969	2.17	2.46	2.18	6.81
Insert Anything (AAAI'26)	1024	0.7281	0.7719	0.3821	6.7440	**0.3007**	2.31	2.59	2.24	7.14
SHINE (ICLR'26)	512	0.7654	0.7155	0.4362	6.7360	0.2889	1.93	2.74	2.16	6.83
SHINE (ICLR'26)	1024	0.7737	0.7165	0.4627	6.7540	0.2891	2.02	2.71	2.22	6.95
Cross-domain compositing										
TF-ICON (ICCV'23)	512	0.7740	0.7551	0.4103	6.7080	0.2895	2.20	2.77	2.13	7.10
TF-ICON (ICCV'23)	768	0.7440	0.7581	0.4073	6.7240	0.2902	2.22	2.71	2.19	7.12
TALE (MM'24)	512	0.7990	0.5395	0.4982	6.8194	0.2422	2.09	2.96	2.27	7.33
PrimeComposer (MM'24)	512	0.7762	0.7510	0.4285	6.3930	0.2851	2.12	2.74	1.93	6.79
AIComposer (ICCV'25)	512	0.5025	0.7946	0.4674	6.6661	0.2853	2.19	2.83	1.98	7.00
AIComposer (ICCV'25)	1024	0.4966	0.8264	0.4826	6.8174	0.2908	2.37	2.91	2.07	7.36
Ours	512	0.5018	0.8001	0.4728	6.7959	0.2872	2.34	**2.99**	2.28	7.61
Ours	1024	**0.4855**	**0.8296**	**0.4994**	**6.8340**	0.2983	**2.46**	2.98	**2.38**	**7.82**

Table 19: **Quantitative comparison on ChameleonDataset$_{\text{ev}}$.** Beyond cross-domain compositing methods, we extend the comparison to specialized painterly harmonization methods and strong general-purpose image editing models, all under identical foreground, background, and mask inputs. Ours outperforms prior methods on most metrics, achieving identity preservation and stylization simultaneously (CLIP-I and CSD).

Method	LPIPS ↓	CLIP-I ↑	CSD ↑	Aes ↑	CLIP-T ↑
Cross-domain compositing					
TF-ICON (ICCV'23)	0.6499	0.7407	0.3833	6.4497	0.1760
TALE (MM'24)	0.6415	0.7214	0.4586	6.5296	0.1672
PrimeComposer (MM'24)	0.7292	0.7736	0.3745	6.1044	0.1954
AIComposer (ICCV'25)	0.4688	0.8275	0.4542	6.5855	0.1984
Painterly image harmonization					
PHDNet (AAAI'23)	0.4877	0.7835	0.3913	6.1978	0.1871
PHDiffusion (MM'23) w/ Res	0.4733	0.7511	0.4123	6.4691	0.1876
PHDiffusion (MM'23) w/o Res	0.4794	0.7371	0.4230	6.5030	0.1834
ArtoPIH (AAAI'24)	**0.4548**	0.7768	0.3480	6.2815	0.1947
General-purpose image editing					
GPT-Image-2	0.6247	**0.8578**	0.4356	6.6722	0.1992
Gemini-3-Pro-Image	0.6584	0.8002	0.4470	6.6824	0.2015
Qwen-Image-Edit	0.6940	0.7838	0.2874	6.6648	0.2018
Ours	0.4651	0.8425	**0.4629**	**6.6946**	**0.2033**

Table 20: **Quantitative evaluation of pose adaptation under substantial viewpoint changes.** FOSScore measures foreground–scene compatibility, while CLIP-I measures foreground identity preservation. w/ and w/o App. Var. denote training with and without data generated by the appearance-variation stage, respectively.

	Baseline	Existing Methods					Ours	
Metric	Copy & Paste	PHDNet	TALE	ArtoPIH	PHDiffusion	AIComposer	w/o App. Var.	w/ App. Var.
FOSScore ↑	0.7427	0.7475	0.7441	<u>0.7740</u>	0.7281	0.7667	0.7522	**0.8072**
CLIP-I ↑	**0.8360**	0.7420	0.6990	0.7230	0.7331	0.8023	<u>0.8306</u>	0.8155

F.2 Pose Adaptation Evaluation

We evaluate pose and viewpoint adaptation under substantial viewpoint changes using a separately curated 50-image evaluation set constructed with our reverse pipeline. The source images do not overlap with those used in ChameleonDataset$_{\mathrm{tr}}$ or ChameleonDataset$_{\mathrm{ev}}$. Concretely, for each sample, we perturb the camera parameters of the foreground reference I'_f to obtain I''_f with a different pose and viewpoint, while using the original object in the real composite I_c as the ground-truth target. For each resulting pair, we quantify the viewpoint change as the geodesic angular distance between the viewing directions of I''_f and the target object in I_c, jointly accounting for azimuth and elevation. Across the 50 samples, the median of these angular distances is 79.7°, with the middle 50% ranging from 45.0° to 90.0°, indicating substantial viewpoint variation. Camera distance is additionally scaled by $\times 0.6$ in 12% of the samples and by $\times 1.8$ in 22%. We evaluate foreground–scene compatibility using FOSScore (Zhang et al., 2023; Chen et al., 2026), where a higher score indicates better compatibility between the generated foreground and the target scene. We additionally report CLIP-I between the generated object and the target object in I_c to measure identity preservation under viewpoint changes.

As shown in Tab. 20, Chameleon achieves the highest FOSScore of 0.8072, indicating more effective adaptation of the perturbed foreground to the target scene under substantial viewpoint changes. Compared with the model trained without appearance variation, FOSScore substantially increases from 0.7522 to 0.8072. Despite this stronger adaptation, Chameleon maintains the highest CLIP-I among the competing generative methods at 0.8155, measured between the generated object and the target object in I_c. These results show that appearance variation improves pose and viewpoint adaptation while maintaining strong foreground identity. A representative example is shown in Fig. 18(a).

G Additional Qualitative Results

G.1 Extended Benchmark Comparisons

We provide additional qualitative comparisons on ChameleonDataset$_{\mathrm{ev}}$ in Figs. 14 , 15, and 16, which include diverse foreground objects and challenging styles such as pixel-art scenes. Our Chameleon consistently adapts foreground objects to the target background style while preserving natural appearance and compositional plausibility.

We also provide additional qualitative results on the TF-ICON benchmark in Fig. 17 across all three domains, including sketch, cartoon, and painting. Across diverse domains, our method demonstrates robust cross-domain adaptation and seamless integration with the background.

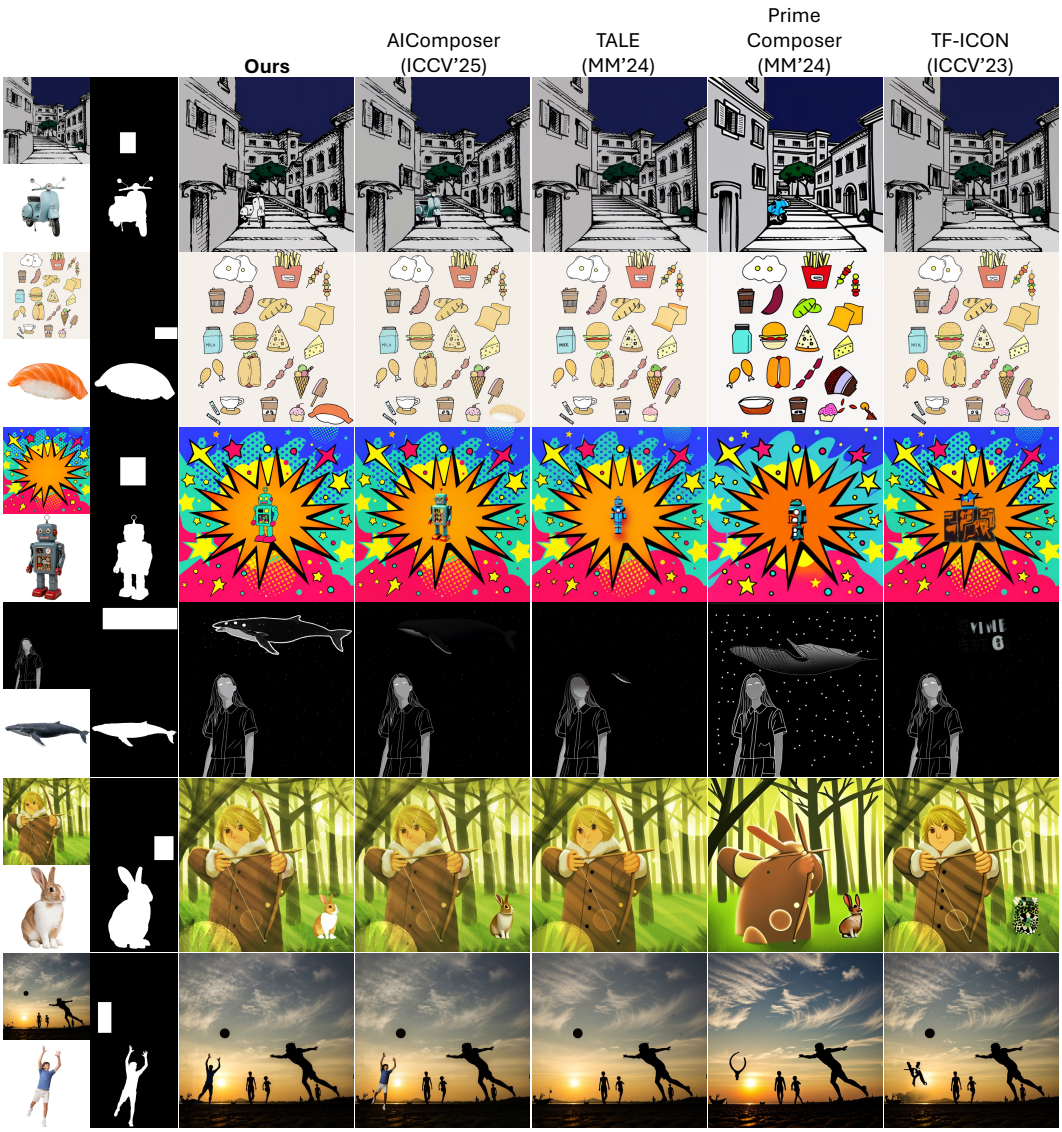


Figure 14: **Qualitative comparison with cross-domain baselines on our ChameleonDataset$_{\text{ev}}$.** Unlike prior benchmarks that reuse backgrounds and foregrounds from existing datasets and pair them without regard to compositional plausibility, which often yields unrealistic scenarios, our benchmark designs the inserted object and its placement for each scene, enabling a more fine-grained and accurate evaluation. Across these diverse scenarios, our method harmonizes arbitrary styles and objects faithfully, demonstrating that it is general and robust.

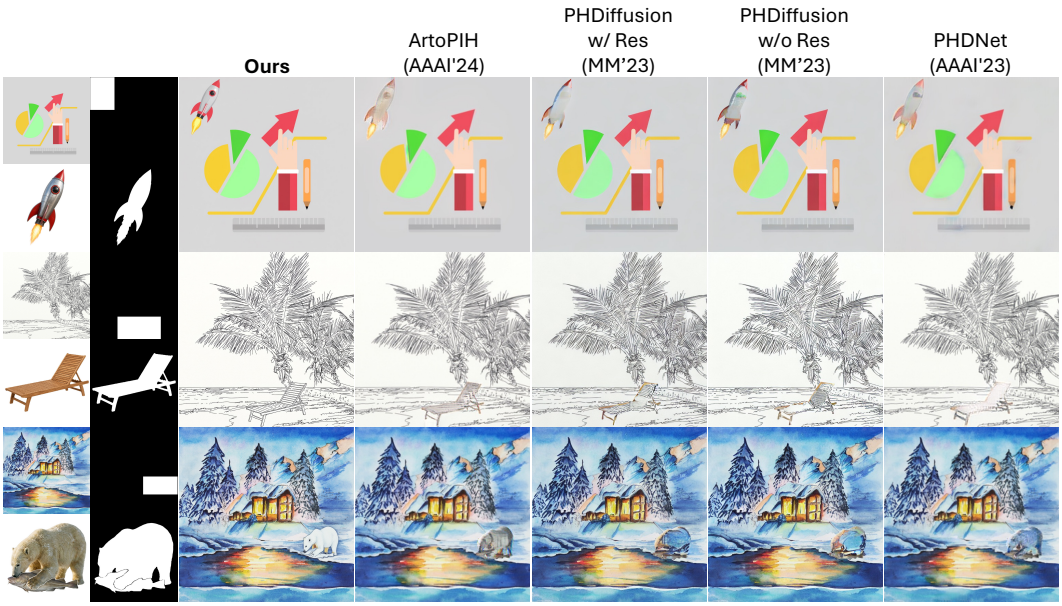


Figure 15: **Qualitative comparison with specialized painterly image harmonization methods on ChameleonDataset$_{ev}$.** Painterly image harmonization methods start from a copy-and-pasted foreground with fixed placement and only align color and texture within the pasted region. Consequently, the foreground is merely blended into the background: it adopts the tone of the target domain but not its structural characteristics, and object identity is compromised under this blending. In contrast, our Chameleon preserves object identity while adapting the foreground to the structural style of the background, and further performs scene-aware pose adaptation, naturally seating the man on the bench (last row), which is unattainable under the copy-and-paste formulation of painterly harmonization.

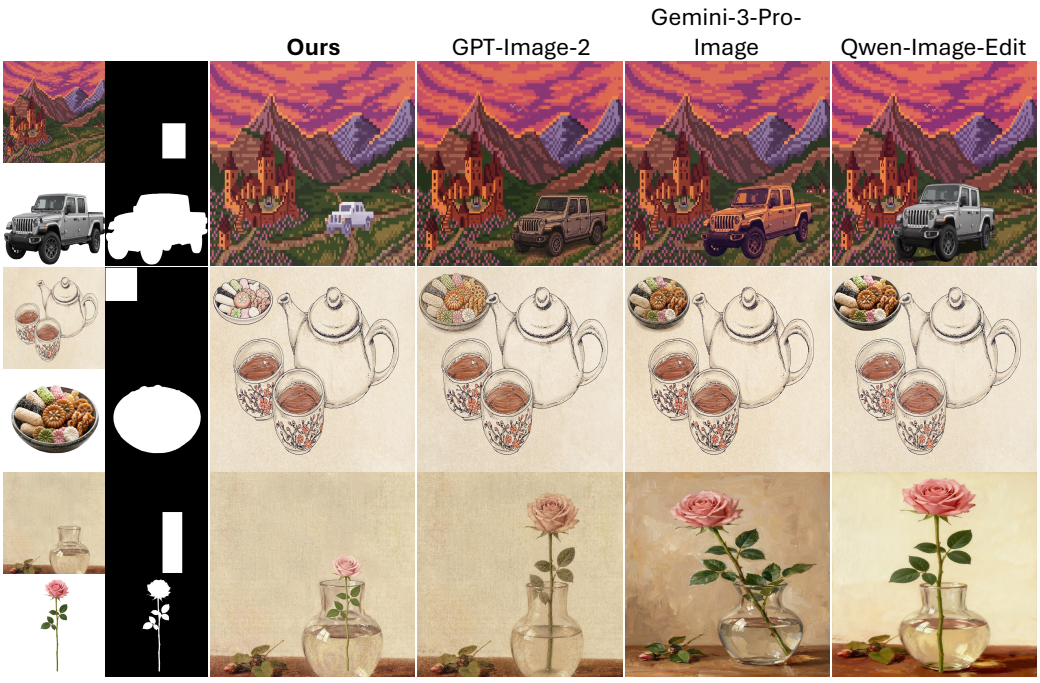


Figure 16: **Qualitative comparison with general-purpose image editing models on ChameleonDataset$_{ev}$.** Although an identical mask specifying the insertion region is given, the general-purpose editing models disregard the user-specified size and location of the foreground. Their stylization is also incomplete, as the foreground fails to adopt the structural characteristics of unique domains such as pixel art (2nd row) or remains photorealistic without being fully stylized (3rd row). In contrast, our Chameleon faithfully follows the user-provided mask and composites the foreground in a style consistent with the background domain.



Figure 17: **Qualitative comparison with baselines on the TF-ICON benchmark.** The benchmark consists of three cross-domain categories: real-to-sketch, real-to-cartoon, and real-to-painting. We visualize representative scenes from each domain. Our method consistently achieves seamless integration and plausible compositing across diverse background domains and styles.

G.2 Scene-Aware Pose Adaptation and Generalized Cross-Domain Compositing

Our reverse data construction pipeline (Fig. 2) allows controlled perturbations of the foreground while preserving the corresponding real-image target. In particular, the appearance variation step in Sec. 3.1 perturbs factors such as azimuth, which provides training examples that encourage scene-aware pose adaptation. As illustrated in Fig. 18(a), Chameleon can adjust the foreground pose to better fit the surrounding scene context, rather than merely copying and pasting the input foreground. In addition, as described in Sec. 3.4, Chameleon extracts background style cues from DINO features and adaptively injects them into the foreground via SGAM, which appears to support generalization across different background domains. Consistent with this, Fig. 18(b) provides qualitative evidence that Chameleon extends beyond the conventional photorealistic-to-stylized setting, and can also handle stylized-to-photorealistic and stylized-to-stylized compositing. These results further suggest that Chameleon generalizes to unseen source and target domains.

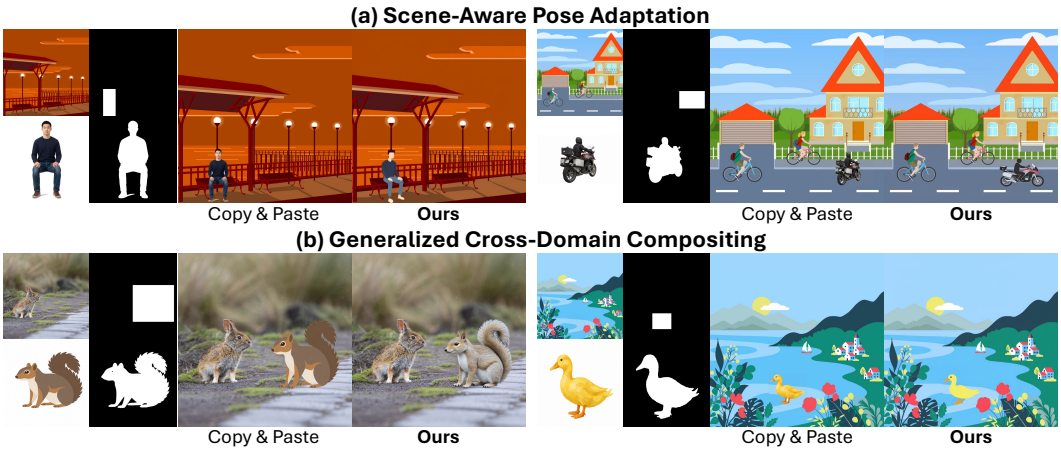


Figure 18: **Scene-aware pose adaptation and generalized cross-domain compositing.** (a) Chameleon exhibits scene-aware pose adaptation, adjusting the foreground pose to the surrounding scene context rather than directly copying and pasting the input foreground. (b) Chameleon further demonstrates generalization beyond the conventional photorealistic-to-stylized setting, including both stylized-to-photorealistic and stylized-to-stylized compositing.

H VLM-BASED PROTOCOLS

H.1 VLM-BASED EVALUATION METRIC

System / Role.

> You are a meticulous cross-domain image compositing evaluation assistant.
> Your task is to evaluate a model output for cross-domain compositing using criterion-specific image priorities.

Provided Inputs. The evaluator is given the following images:

1. **f_seg full image**: a clean object-centric reference on a white background.
2. **background_image full image**: the target scene whose overall style should guide stylization.
3. **model_output full image**: the final generated/composited result.
4. **model_output_crop**: the cropped region inside the provided bounding box from the model output.
5. **background_mask image**: the source used to define the target region. Its white region is resized to the model output resolution before extracting the bounding box.

Criterion-specific Image Priority.

- For **Identity Preservation**, focus primarily on the f_seg full image and the model_output_crop.
- For **Style Transfer Consistency**, focus primarily on the background_image full image and the model_output_crop.
- For **Composition and Harmonization**, focus primarily on the model_output_crop, and use the model_output full image as secondary context.

Evaluation Rules.

- Use the stated priority images for each criterion.
- Do not require the same image pair for every criterion.
- For Identity Preservation, judge whether the cropped output object preserves the same object identity as the clean f_seg reference.
- For Style Transfer Consistency, judge whether the cropped output object reflects the style of the full background scene appropriately.
- For Composition and Harmonization, judge whether the cropped output region looks naturally integrated and visually plausible, while using the full output only as secondary global context.
- Do not penalize based on regions unrelated to the provided bounding box.
- Do not let style transfer alone override identity preservation.
- Do not let identity similarity alone override poor harmonization.

Identity Preservation. **Question.** Does the object inside the provided bounding box in the model output preserve the same identity as the object in the f_seg reference image, despite the stylization? Consider whether the object remains recognizably the same in terms of its essential structure, shape, distinctive parts, and overall visual identity.

Scoring Guide (0–3).

- **3**: The object clearly preserves the same identity as the f_seg reference. Its structure, distinctive parts, and overall appearance remain highly recognizable despite stylization.

- **2**: The object mostly preserves the same identity, but there are minor ambiguities or slight structural/detail changes.
- **1**: The object weakly preserves the identity. Major parts or structure are altered, making it difficult to confidently match.
- **0**: The object does not preserve the identity. It appears as a different object or is unrecognizable.

Style Transfer Consistency. Question. Does the object inside the provided bounding box in the model output receive the background style consistently and appropriately? Consider whether the stylization reflects the full background style well, not just blending background color, while avoiding excessive style leakage that unnaturally distorts the object or weakens its visual coherence.

Scoring Guide (0–3).

- **3**: The object is stylized in a way that strongly and consistently reflects the background style. The style transfer is clear and appropriate, and there is no noticeable excessive style leakage or unnatural distortion.
- **2**: The object mostly reflects the background style, but the style transfer is slightly incomplete, uneven, or accompanied by minor style leakage that slightly affects coherence.
- **1**: The object shows weak or inconsistent style transfer from the background. The stylization is limited, patchy, or noticeably affected by style leakage that harms the object's coherence.
- **0**: The object does not meaningfully reflect the background style, or the result is severely corrupted by excessive style leakage, making the object visually incoherent or unnaturally distorted.

Composition and Harmonization. Question. Does the object inside the provided bounding box in the model output look naturally integrated and visually plausible? Focus primarily on the cropped output region, and use the full model output only as secondary global context. Consider whether the insertion looks natural rather than pasted, and whether local lighting, shadow consistency, texture transition, and overall harmonization appear convincing. Also consider physical plausibility, such as whether the object interacts naturally with its surroundings (e.g., ground or supporting surfaces), whether spatial relationships and depth ordering are coherent, whether the object is fully and coherently rendered without missing parts, and whether occlusion relationships with surrounding elements are handled realistically.

Scoring Guide (0–3).

- **3**: The cropped output region looks very natural and well integrated. The insertion does not appear pasted, and local lighting, shadow consistency, texture transition, and overall harmonization are convincing.
- **2**: The region is mostly natural, but there are minor issues in local consistency or harmonization. Slight artificiality may be noticeable.
- **1**: The region looks noticeably unnatural. Pasted appearance or weak local harmonization make the insertion feel inconsistent.
- **0**: The region clearly fails. It strongly looks like an artificial copy-and-paste insertion, with severe artifacts or implausible local appearance.

Task Description.

> Evaluate how well the generated object is preserved, stylized, and harmonized inside the target region. Use the criterion-specific image priorities described above.

Output Format.

```
Identity Preservation: [score, 0-3] - [brief
justification]
```

```
Style Transfer Consistency: [score, 0-3] - [brief
justification]
Composition and Harmonization: [score, 0-3] - [brief
justification]
Total Score: [sum of the three scores]
```

H.2 VLM-BASED FILTERING

System / Role.

> You are a strict visual evaluator.

Setup. After candidate generation, each candidate is associated with two images: **Image A**, the original crop containing the target object together with surrounding occluders, and **Image B**, the segmented foreground produced by SAM 3, where occluders such as hands are intentionally removed. The segmented region can therefore be small or fragmented, but a small region alone is not a reason for rejection. We use QWEN3-VL-8B-INSTRUCT as the filter with greedy decoding (`do_sample=False`) and `max_new_tokens`$= 512$. Both images are resized so that the longer side is at most 256 px before being passed to the model.

Provided Inputs. The model receives:

1. **Image A**: the original crop image containing the target object and surrounding occluders.
2. **Image B**: the segmented foreground image of the target object.
3. **Target label**: the semantic category associated with the candidate.

Task Description. The filtering objective is to determine whether Image B provides a useful partial observation of the target object under occlusion. The model evaluates whether the remaining visible evidence supports plausible restoration or completion in later stages.

Evaluation Rules.

- The segmentation intentionally excludes occluders, so Image B may contain only a partial object region.
- A small segmented region alone should not trigger rejection.
- The evaluation should focus on semantic consistency, occlusion plausibility, and recoverability from the visible evidence.
- The model must return only a valid JSON object.
- Markdown formatting and code fences are explicitly forbidden.
- We apply up to two retries on JSON parse failure.

Criteria. Each candidate is evaluated using four criteria:

label_consistency. Does Image B still depict the target semantic class?

identity_preservation. Is the same object instance recognisable across Image A and Image B?

occlusion_plausibility. Is the missing region physically consistent with a realistic occluder?

recoverability. Can the full object be plausibly reconstructed from the visible evidence in Image B?

Aggregation. We aggregate the four scores into a single keep score $s \in [0, 100]$:

$$\text{sem} = 0.35 \cdot \texttt{label_consistency} + 0.65 \cdot \texttt{identity_preservation},$$

$$s = \left\lfloor 100 \cdot \Big(0.60 \cdot \texttt{recoverability} + 0.25 \cdot \texttt{occlusion_plausibility} + 0.15 \cdot \text{sem} \Big) \right\rfloor.$$

Recoverability receives the largest weight because the downstream task requires plausible amodal completion from partial observations.

Binary Decision. We convert the continuous score into a binary keep/reject label:

$$\text{keep}(s, \texttt{recoverability}) = \mathbb{1}\left[s \geq 70 \ \wedge \ \texttt{recoverability} \geq 0.35\right].$$

Candidates with low recoverability are rejected regardless of the final aggregated score because their visible regions do not provide sufficient evidence for plausible completion.

Output Format. The model returns a single JSON object containing:

- `label_consistency`
- `identity_preservation`
- `occlusion_plausibility`
- `recoverability`
- `final_score`
- `short_reason`

For each candidate, we store the four sub-scores, the aggregated keep score, and the final binary decision in the per-scene `result.json`.

I Limitations and Broader Impacts

I.1 Limitations

Although Chameleon generalizes across diverse source and target domains, unusual background appearances, such as strong defocus blur, can still make seamless integration challenging. In such cases, the foreground must adapt to the background appearance while also reflecting the degradation itself, which is not explicitly modeled by our framework. As a result, the foreground can preserve its identity yet remain insufficiently matched to the degraded background. This challenge is shared by existing cross-domain compositing methods, which often fail to preserve both foreground identity and background-consistent appearance under such conditions.

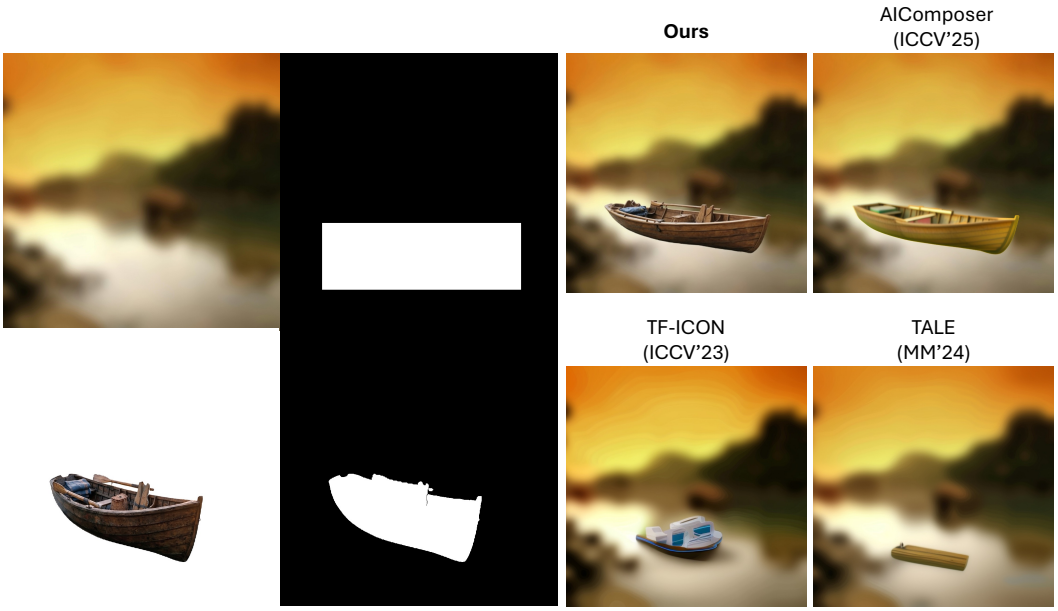


Figure 19: **Limitation on unseen degraded appearances.** When the background contains strong defocus blur, Chameleon preserves foreground identity but may not fully reproduce the background degradation, while existing cross-domain compositing methods also struggle to maintain both identity and background-consistent appearance.

I.2 BROADER IMPACTS

Our work advances cross-domain image compositing by enabling more seamless and stylistically consistent integration between foreground objects and heterogeneous background domains. This capability may benefit creative applications such as digital art, visual content creation, and stylized media editing. However, the proposed framework may also be misused to generate misleading or deceptive visual content. In particular, realistic cross-domain compositing could be applied to create fabricated images that appear visually plausible, potentially contributing to misinformation or unauthorized image manipulation. In addition, generated results may raise concerns regarding copyright and artistic style imitation when specific visual styles are closely replicated.