CTRLCAMO: CONTROLLABLE ADVERSARIAL CAMOUFLAGE EDITING FOR VEHICLE DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep neural networks (DNNs) have achieved remarkable success in computer vision but remain vulnerable to adversarial attacks. For vehicle detection, many existing attacks rely on imperceptible perturbations or localized adversarial patches, which can limit attack effectiveness or introduce visually conspicuous patterns. In this paper, we introduce *CtrlCamo*, a controllable framework for adversarial camouflage editing, which edits vehicle appearance to reduce detector performance while maintaining visual consistency with the environment. CtrlCamo supports two stylization strategies: image-level, which adapts vehicles to their surroundings, and scene-level, which aligns them with semantic concepts in the scene. A two-stage ControlNet fine-tuning pipeline jointly enforces structural preservation, style consistency, and adversarial effectiveness. Experiments on COCO and LINZ demonstrate at least 38% AP_{50} reduction over prior methods while improving structural preservation and perceptual consistency. The resulting adversarial effects also transfer to unseen black-box detectors. Code and data will be released upon paper acceptance to facilitate further research.

1 INTRODUCTION

Deep neural networks (DNNs) have achieved remarkable success across a wide range of computer vision applications (He et al., 2016; Minaee et al., 2021; Zhao et al., 2019). However, they are also highly vulnerable to adversarial examples that are crafted by adding carefully designed perturbations to normal examples (Goodfellow et al., 2015). In vehicle detection, such adversarial inputs can cause detectors to misidentify vehicles, raising reliability concerns across applications ranging from aerial monitoring to ground-level visual perception. Many existing digital attacks rely on imperceptible perturbations (Dai et al., 2025; Chen et al., 2023) or localized adversarial patches (Hu et al., 2021; Wang et al., 2025). Imperceptible perturbations can preserve the original appearance, but their constrained magnitude may limit attack effectiveness, while adversarial patches allow larger localized modifications but can introduce conspicuous patterns. These limitations motivate us to explore an appearance-based adversarial attack that more flexibly modifies vehicle appearance while jointly pursuing adversarial effectiveness and visual compatibility with its surrounding environment.

Camouflage provides a natural mechanism for achieving such visual compatibility. Recent advances in generative models have enabled increasingly realistic camouflage image generation, where foreground objects are synthesized to blend naturally into their surrounding context (Zhao et al., 2024). However, visual camouflage alone does not guarantee adversarial effectiveness, as an object that appears well concealed to a human observer may remain readily detectable by an object detector. Existing camouflage generation methods primarily optimize contextual consistency. In contrast, conventional adversarial attacks explicitly optimize model deception through objective functions. This motivates a complementary problem that we call *adversarial camouflage editing*: given a real image containing a target vehicle, can we directly transform the vehicle's appearance into a context consistent camouflage while simultaneously reducing its detectability?

To address this problem, we develop *CtrlCamo*, a controllable diffusion-based editing pipeline, as illustrated in fig. 1, that synthesizes an adversarially camouflaged image satisfying three properties: (i) preservation of the vehicle structure (e.g., the airplane in fig. 1) and surrounding background, (ii) application of user-guided, stealthy style edits to the vehicle appearance, and (iii) reduction of detector confidence. Concretely, we fine-tune a ControlNet (Zhang et al., 2023) to encode structural and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

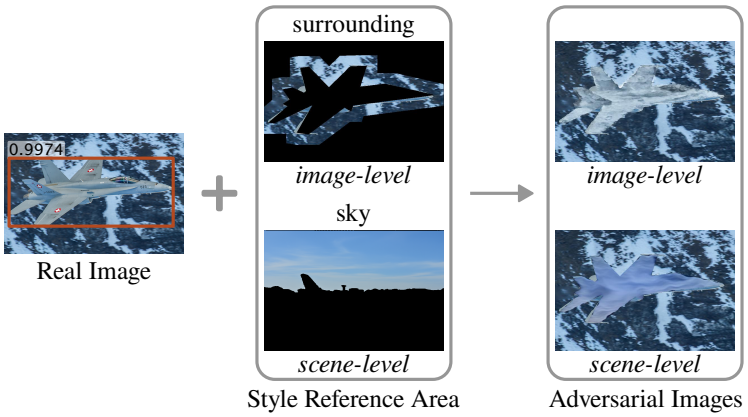


Figure 1: **Overview.** Given a real image, our pipeline stylizes the target vehicle based on either its immediate surroundings (image-level) or a visual concept present in the overall scene (scene-level), producing contextually consistent adversarial examples. The numbers on the bounding boxes indicate detector confidence scores, and the absence of a box indicates that the vehicle is not detected.

stylistic guidance and optimize a composite objective that combines a structural-preservation term, a style-consistency term, and an adversarial detection loss. At inference time, our pipeline generates camouflaged images by direct sampling. Within this pipeline, we design two stylization strategies inspired by nature to address different practical needs. The *image-level* strategy (Wikipedia contributors, 2025a) transfers visual appearance from the vehicle's immediate surroundings, analogous to chameleons, enabling natural blending with local contexts. While effective for a given image, this strategy is inherently tied to the specific surroundings of each image. To reduce this dependence, we introduce a *scene-level* strategy (Wikipedia contributors, 2025b), which adapts the vehicle's appearance to a common semantic concept present in the scene, analogous to grasshoppers resembling dry leaves, enabling more context-robust camouflage. For example, in fig. 1, an airplane flies within a sky scene, where *sky* serves as the shared visual concept. Therefore, the pipeline adopts the blue sky as the reference style area, producing a camouflaged airplane consistent with the scene while reducing detector confidence. Extensive experiments on both ground-view (COCO) and nadir-view (LINZ) datasets demonstrate that our pipeline achieves strong attack effectiveness, better preserves vehicle structure, improves perceptual stealthiness, and transfers to unseen black-box detectors.

Our contributions can be summarized as follows:

- We formulate appearance-based attacks against vehicle detectors as an adversarial camouflage editing problem and propose two stylization strategies. The image-level strategy blends the vehicle with its surroundings, and the scene-level strategy adapts the vehicle to a semantic concept present in the scene, producing contextually coherent camouflage.
- We propose a novel pipeline based on ControlNet fine-tuning. Our method jointly enforces structural fidelity to maintain vehicle geometry, style consistency to produce stealthy camouflage, and an adversarial objective to reduce detector performance.
- We evaluate our approach on the COCO and LINZ datasets, and demonstrate strong attack effectiveness, better preservation of vehicle structure, improved visual stealthiness, and transferability to black-box detectors.

2 Related Work

Adversarial camouflage editing is related to digital attacks, physical camouflage attacks, and camouflage image generation. We review these areas and clarify how our setting differs from each.

Digital attacks. Digital attacks manipulate image appearance to deceive vision models, ranging from imperceptible perturbations and localized patches to broader appearance modifications. Classical methods such as TOG (Chow et al., 2020) and NAP (Hu et al., 2021) optimize imperceptible or localized changes to reduce detector confidence, while recent diffusion-based meth-

ods (Chen et al., 2024; Li et al., 2024; Dai et al., 2025; Chen et al., 2023; Wang et al., 2025) incorporate generative priors for adversarial perturbation or patch synthesis. Appearance-based methods such as AdvCAM (Duan et al., 2020) and DiffPGD (Xue et al., 2023) are closer to our work by incorporating style-aware objectives. However, these approaches rely on per-image optimization and do not explicitly formulate adversarial attacks as context-conditioned editing of a given image. In contrast, CtrlCamo enforces structural fidelity, and jointly optimizes stylization and adversarial objectives to produce contextually consistent camouflages with efficient inference.

Camouflage image generation. This line of work has explored generating visually coherent camouflage by encouraging foreground objects to blend with their surroundings. Deep Camouflage Images (Zhang et al., 2020) combines style transfer with camouflage objectives to synthesize natural concealed objects, while more recent methods such as LAKE-RED (Zhao et al., 2024) and CamoDreamer (Wang et al., 2026) leverage generative models and contextual information to produce camouflage images. These methods primarily focus on visual concealment and camouflage image synthesis rather than adversarial deception of object detectors. In contrast, CtrlCamo incorporates a detector-oriented adversarial objective into context-conditioned appearance editing, jointly pursuing environmental consistency and reduced detector performance.

Physical camouflage attacks. A related line of work studies physically realizable vehicle camouflage. These methods optimize a UV texture map over a fixed 3D mesh through differentiable rendering, so that the resulting camouflage can remain effective across multiple viewpoints (Lyu et al., 2024; Zhou et al., 2024; Zhang et al., 2019). The two settings are complementary and not directly comparable because physical methods typically optimize multi-view appearance tied to a specific vehicle, which requires meshes and camera information, whereas CtrlCamo directly edits its diverse vehicle instances in any given 2D images without such 3D assets. We further analyze whether independently generated 2D edits transfer across viewpoints in section E.2.

3 METHOD

We propose a two-stage framework for adversarial camouflage editing, as shown in fig. 2. In section 3.1, we outline the mathematical foundations of our approach. Next, section 3.2 introduces our image-level and scene-level stylization strategies, which automatically select style exemplars for vehicle appearance transfer. Training is performed in two sequential stages. In the first stage (section 3.3), termed the No-Box Attack (Li et al., 2020), we fine-tune a ControlNet (Zhang et al., 2023) to transfer the selected reference style while preserving vehicle geometry, without detector-dependent loss. In the second stage (section 3.4), termed the White Box Attack, we further optimize the model against a known detector. This stage minimizes detectability while enforcing color and style consistency, ensuring that the adversarial camouflage retains the visual realism and stylistic attributes learned in the first stage. At inference, the trained pipeline synthesizes camouflaged images without per-image optimization.

3.1 PRELIMINARIES

Diffusion Models (Ho et al., 2020) formulate image generation as a denoising process that gradually transforms random Gaussian noise into a sample from the target data distribution. In this work, we employ Stable Diffusion (Rombach et al., 2022), which performs generation in the latent space of a pre-trained autoencoder. The model consists of an image encoder $\mathcal{E}$, a denoising network ϵ_θ, and an image decoder $\mathcal{D}$. An image x_0 is first encoded into the latent representation $z_0 = \mathcal{E}(x_0)$. The forward process gradually adds Gaussian noise ε_t to z_0:

$$z_t = \sqrt{\bar{\alpha}_t} z_0 + \sqrt{1 - \bar{\alpha}_t}\varepsilon_t,\ \varepsilon_t \sim \mathcal{N}(0, \mathbf{I}), \tag{1}$$

where $\bar{\alpha}_t$ controls the noise schedule. To learn the reverse process, the network $\epsilon_\theta(z_t, c)$ is trained to predict the noise ε given the noisy latent z_t and a condition c. Since adversarial loss is defined on the image space, inspired by (Xie et al., 2025), we adopt a one-step estimate from the noisy latent z_t to approximate the reverse process based on eq. (1):

$$\hat{z}_0 = \frac{z_t - \sqrt{1 - \bar{\alpha}_t}\epsilon_\theta(z_t, c)}{\sqrt{\bar{\alpha}_t}} \tag{2}$$

Finally, the reconstructed image $\hat{x}_0$ is produced by decoding the estimated latent $\hat{z}_0$ through the decoder $\mathcal{D}$, which can be formulated as $\hat{x}_0 = \mathcal{D}(\hat{z}_0)$.

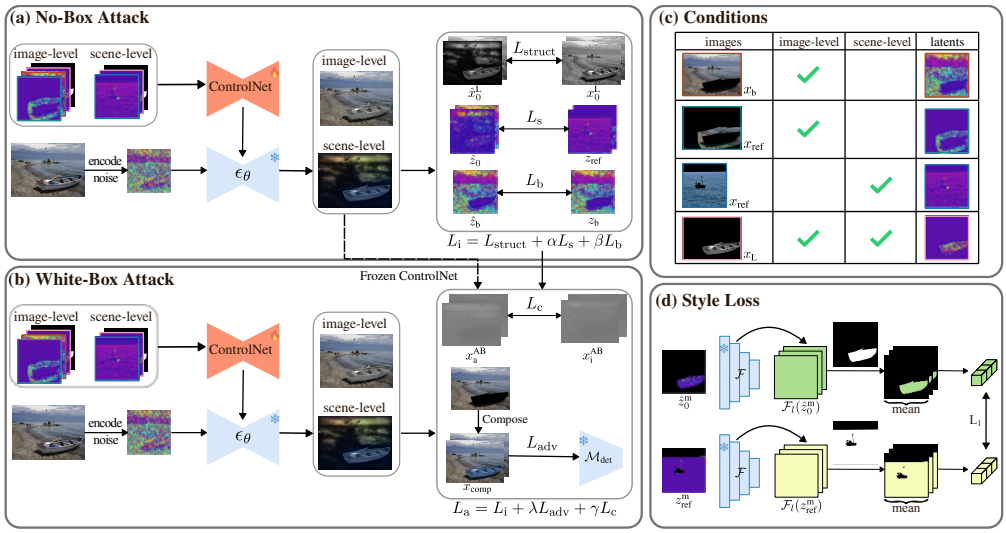


Figure 2: **Overview of our pipeline.** As shown in (a) and (b), the pipeline consists of a *No-Box Attack* stage and a *White-Box Attack* stage. In (a), the ControlNet is fine-tuned to stylize vehicles using a reference region while preserving geometry and background through structure, style, and background supervisions (L_{struct}, L_{s}, L_{b}). (b) further optimizes the model against a detector $\mathcal{M}_{\text{det}}$ by incorporating an additional adversarial loss L_{adv} and a color-consistency loss L_{c}. (c) summarizes the conditions provided to ControlNet under the image-level and scene-level settings, and (d) illustrates the style loss L_{s} that aligns vehicle latent features with the reference area.

3.2 Style Reference Selection

We propose two complementary strategies for selecting style exemplars. For image-level camouflage, we adapt the vehicle appearance to its immediate surroundings. Given vehicle mask m_{x_0}, We dilate it to obtain $m'_{x_0} = \text{dilate}(m_{x_0})$, thereby including a small region around the vehicle. The reference area is then defined as $x_{\text{ref}} = x_0 \odot (m'_{x_0} - m_{x_0})$, which captures pixels adjacent to the vehicle region. For scene-level camouflage, we first group images by scene type using Multimodal Large Language Models (MLLMs) (Zhu et al., 2025; Vikhyat Korrapati, 2024), then identify a concept that naturally occurs in each group, ensuring that the stylized vehicle appearance remains visually consistent with the scene. For each scene group, we synthesize an exemplar image x_{gen} containing that concept using a Stable Diffusion fine-tuned on the training data, leveraging dataset-level visual statistics to obtain a representative concept appearance rather than relying on a specific retrieved instance. We then extract its segmentation mask m_{gen} using SAM 2 (Ravi et al., 2025), and define $x_{\text{ref}} = x_{\text{gen}} \odot m_{\text{gen}}$. The vehicle is then stylized to match the selected reference. More implementation details and an example of the entire process are provided in section A.

3.3 No-Box Attack

In the first stage, we fine-tune a ControlNet to generate camouflaged vehicles using a reference style image. The pipeline takes as input the vehicle's luminance (L) channel x_{L} in LAB space, the style reference region x_{ref} defined in section 3.2, the vehicle mask m_x, and, for the image-level strategy, an additional background image x_{b}, as shown in fig. 2(c). Given an input image x_0, the estimated latent $\hat{z}_0$ and reconstructed image $\hat{x}_0$ are obtained from eq. (2). The training objective combines three components: a structure preservation loss L_{struct} to maintain vehicle structure, a style loss L_{s} to guide camouflage generation, and a background supervision L_{b} for image-level strategy. The overall loss function is formulated as

$$L_{\text{i}} = L_{\text{struct}} + \alpha L_{\text{s}} + \beta L_{\text{b}} \tag{3}$$

where $\beta = 0$ for scene-level strategy and $\beta \neq 0$ for image-level strategy.

Structure preservation loss. Inspired by colorization tasks (Zhang et al., 2016) that convert the input image into LAB space and use the L channel to preserve the structure, we also use the vehicle's

L channel to constrain the reconstructed vehicle structure. Given an input image x_0 and the one-step estimated image $\hat{x}_0$, we extract both L channels from x_0 and $\hat{x}_0$, and normalize them to (0,1), which we denote as x_0^{L} and $\hat{x}_0^{\mathrm{L}}$. Given the vehicle segmentation mask m_{x_0}, the structure preservation loss is formulated as the average L_2 difference of L channel of vehicle area between x_0 and $\hat{x}_0$:

$$L_{\text{struct}} = \frac{1}{\sum m_{x_0}} \left\| x_0^{\mathrm{L}} \odot m_{x_0} - \hat{x}_0^{\mathrm{L}} \odot m_{x_0} \right\|_2^2 . \tag{4}$$

Style loss. We adopt the pre-trained LatentLPIPS (Kang et al., 2024) classifier $\mathcal{F}$ to measure perceptual style similarity in diffusion latent space. Compared with LPIPS (Zhang et al., 2018), LatentLPIPS is more efficient as it operates in latent space rather than pixel space, and uses differentiable augmentations during training, which enables robust comparison of masked latent regions across different spatial contexts or images. Concretely, as shown in fig. 2 (d), given the one-step estimated image $\hat{x}_0$ from an input image x_0 via eq. (2), the corresponding vehicle segmentation mask m_{x_0}, the style image x_{s}, and its style reference area segmentation mask m_{s}, we first encode the masked images into latent representations to suppress interference from irrelevant regions, which can be formulated as $\tilde{z}_0 = \mathcal{E}(\hat{x}_0 \odot m_{x_0})$ and $z_{\text{ref}} = \mathcal{E}(x_{\mathrm{s}} \odot m_{\mathrm{s}})$. We denote the downsampled masks as $m_{x_0\downarrow}$ and $m_{\mathrm{s}\downarrow}$. Since zero-valued pixels do not necessarily yield zero latent activations, we further apply the downsampled masks to the latent codes to remove background interference, which can be formulated as $\tilde{z}_0^{\mathrm{m}} = \tilde{z}_0 \odot m_{x_0\downarrow}$ and $z_{\text{ref}}^{\mathrm{m}} = z_{\text{ref}} \odot m_{\mathrm{s}\downarrow}$, where $m_{x_0\downarrow}$ and $m_{\mathrm{s}\downarrow}$ have the same resolution as their corresponding latents. Because the vehicle and style reference area occupy distinct spatial locations and may originate from different images, direct feature-wise subtraction is infeasible. Instead, for layer l, we extract feature maps $\mathcal{F}_l(\tilde{z}_0^{\mathrm{m}})$ and $\mathcal{F}_l(z_{\text{ref}}^{\mathrm{m}})$, and use downsampled masks $m_{x_0\downarrow}$ and $m_{\mathrm{s}\downarrow}$ to select the vehicle and reference regions from these feature maps. We then minimize the L_1 difference in average features within the two regions:

$$L_{\mathrm{s}} = \sum_l \left\| \frac{\sum(\mathcal{F}_l(\tilde{z}_0^{\mathrm{m}}) \odot m_{x_0\downarrow})}{\sum m_{x_0\downarrow}} - \frac{\sum(\mathcal{F}_l(z_{\text{ref}}^{\mathrm{m}}) \odot m_{\mathrm{s}\downarrow})}{\sum m_{\mathrm{s}\downarrow}} \right\| \tag{5}$$

where $m_{x_0\downarrow}$ and $m_{s\downarrow}$ are resized to match the spatial resolution of the feature map at each layer l.

Background reconstruction loss. We observe that background reconstruction leads to more coherent vehicle stylization under the image-level strategy, where the vehicle appearance is expected to align with its immediate surroundings. This is because each image is conditioned on its own surroundings rather than a few shared reference areas, which makes style transfer more challenging via average feature-space loss. Background supervision introduces stronger pixel-level constraints that anchor the global image color and illumination distribution, allowing gradients to propagate through shared features and harmonize the vehicle's appearance with its surroundings. Similarly, given the one-step estimated image $\hat{x}_0$ from an input image x_0 via eq. (2), the corresponding vehicle segmentation mask m_{x_0}, we encode the masked images into latent representations to suppress interference from vehicle regions, which can be formulated as $z_{\mathrm{b}} = \mathcal{E}(x_0 \odot (1 - m_{x_0}))$ and $\hat{z}_{\mathrm{b}} = \mathcal{E}(\hat{x}_0 \odot (1 - m_{x_0}))$. We further apply the downsampled masks $m_{x_0\downarrow}$ to z_{b} and $\hat{z}_{\mathrm{b}}$ to focus on the background, which can be formulated as $z_{\mathrm{b}}^{\mathrm{m}} = z_{\mathrm{b}} \odot (1 - m_{x_0\downarrow})$ and $\hat{z}_{\mathrm{b}}^{\mathrm{m}} = \hat{z}_{\mathrm{b}} \odot (1 - m_{x_0\downarrow})$. Background reconstruction loss L_{b} is formulated to be the LatentLPIPS (Kang et al., 2024) loss between $z_{\mathrm{b}}^{\mathrm{m}}$ and $\hat{z}_{\mathrm{b}}^{\mathrm{m}}$, which minimizes both latent-space pixel-wise and perceptual feature differences.

3.4 White-Box Attack

In the second stage, we continue to fine-tune the ControlNet from the first stage as described in section 3.3. The goal is to preserve the vehicle appearance learned in the first stage while deceiving the vehicle detector $\mathcal{M}_{\text{det}}$. Concretely, we augment the first-stage loss L_{i} from eq. (3) with two terms: a color-consistency loss L_{c} that constrains chromatic deviation in the adversarial output, and an adversarial detection loss L_{adv}. The combined objective is formulated as

$$L_{\mathrm{a}} = L_{\mathrm{i}} + \lambda L_{\text{adv}} + \gamma L_{\mathrm{c}} \tag{6}$$

Adversarial loss. Given the one-step estimated image $\hat{x}_0$ from an input image x_0 via eq. (2), and vehicle segmentation mask m_{x_0}, since we are only allowed to edit the vehicle, we compose real image background and estimated image vehicle before passing it through a detector, which can be formulated as $x_{\text{comp}} = \hat{x}_0 \odot m_{x_0} + x_0 \odot (1 - m_{x_0})$. We then optimize the camouflaged vehicle

to be detected as background. Formally, if $\mathcal{M}_{\text{det}}(x_{\text{comp}})$ denotes the detector logits, and y_{b} is the background label, the adversarial objective can be written as a cross-entropy loss:

$$L_{\text{adv}} = \text{CE}(\mathcal{M}_{\text{det}}(x_{\text{comp}}), y_{\text{b}}) \tag{7}$$

Color-consistency loss. While the style loss L_{s} introduced in section 3.3 aligns the feature representation between the vehicles in generated images and the reference area, we observe that during white-box attacks, the model may slightly shift vehicle colors while keeping the style loss nearly unchanged to facilitate optimization of the adversarial loss L_{adv}, leading to undesired color deviations. Inspired by DINOv3 (Siméoni et al., 2025), we leverage the knowledge from the previous training stage to stabilize vehicle appearance. Specifically, we condition on the frozen ControlNet trained in section 3.3 and the ControlNet trained in this stage to reconstruct one-step outputs from eq. (2), denoted as x_{i} and x_{a}, respectively. Both outputs are converted to the LAB color space, and we extract normalized AB channels, yielding x_{i}^{AB} and x_{a}^{AB}, and their difference is minimized to ensure consistent color representation across stages. Given the vehicle segmentation mask m_{x_0}, the loss is computed as the mean L_2 distance in the AB channels over vehicle regions:

$$L_{\text{c}} = \frac{1}{\sum m_{x_0}} \left\| x_{\text{i}}^{\text{AB}} \odot m_{x_0} - x_{\text{a}}^{\text{AB}} \odot m_{x_0} \right\|_2^2 \tag{8}$$

4 Experiments

We conduct comprehensive experiments to evaluate the effectiveness of CtrlCamo from four perspectives: attack effectiveness, perceptual stealthiness, structural preservation, and transferability. In section 4.1, we describe the setup. In section 4.2, we compare our method with state-of-the-art methods under the white-box setting, where target detectors are known. Additionally, we assess the transferability of our method to black-box settings, where the victim models are unknown. In section 4.3, we conduct ablation studies.

4.1 Experimental Setup

Datasets. We conduct experiments on two public datasets: LINZ (Fang et al., 2025) and COCO (Lin et al., 2014). LINZ consists of aerial imagery collected over Selwyn, New Zealand, with a ground sampling distance (GSD) of 12.5 cm per pixel. Images are cropped to 112×112 pixels and annotated with pseudo-bounding boxes marking car centers. After filtering images that contain cars, the dataset includes $6,011$ training and 728 testing samples. The scenes are categorized into five types, namely *residential, industrial, agricultural, parking lot,* and *highway*. Each category is associated with a corresponding style reference concept (*house, building, field, tree,* and *grass*) used for scene-level stylization. COCO contains diverse natural scenes with vehicles captured from ground-level viewpoints. We extract images containing one vehicle, resulting in $8,965$ training samples and 400 testing samples. The data are grouped into five environments: *urban, rural, road, sky*, and *lake*. Each environment is paired with a representative style reference concept (*building, grass, tree, sky*, and *water*) that guides the stylization process. Ground-truth vehicle masks are used during training, while during testing we use SAM 2 (Ravi et al., 2025) to extract the masks.

Implementation. We adopt Stable Diffusion v1.5 (Rombach et al., 2022) as our generative model and fine-tune ControlNet (Zhang et al., 2023) in both stages with a batch size of 4 on two RTX A6000 GPUs. The dilation kernel size is set to 75 px for COCO and 11 px for LINZ. The images are all resized to 512 px $\times$ 512 px. We use a template prompt of the form "an image of {scene type} area with {objects}", where {scene type} is the scene label of the image and {objects} lists the objects present in the image. At test time, we run the pipeline for 30 sampling steps. For attacks and evaluation, we use the MMDetection framework (Chen et al., 2019) with Faster-RCNN (Ren et al., 2015) and ViTDet (Li et al., 2022) as the white-box target detection models for adversarial camouflage generation. To evaluate the transferability of our method, we further treat YOLOv5 (Jocher, 2020), YOLOv8 (Reis et al., 2023), and MLLMs (Vikhyat Korrapati, 2024; Zhu et al., 2025) as black-box models and test on them. For detection evaluation, we report both $\text{AP}_{50:95}$ and AP_{50} on COCO and AP_{50} on LINZ, following the original paper (Fang et al., 2025). Attack effectiveness is measured by the decrease in the corresponding detection metric relative to the clean baseline. To assess the preservation of vehicle structure, we report the Structural Similarity Index Metric (SSIM). Input images are cropped to reduce the background and enlarge the vehicle. Finally, we measure inference

Table 1: **Quantitative comparison on the LINZ Dataset for stylization-based digital attacks**. We report the AP_{50}, SSIM, and average sampling time.

Method	Faster-RCNN					ViTDet				
	image-level		scene-level		Inf. Latency (s) ↓	image-level		scene-level		Inf. Latency (s) ↓
	AP_{50}(%) ↓	SSIM ↑	AP_{50}(%) ↓	SSIM ↑		AP_{50}(%) ↓	SSIM ↑	AP_{50}(%) ↓	SSIM ↑	
Normal	98.3	-	98.3	-	-	97.8	-	97.8	-	-
AdvCAM Duan et al. (2020)	90.5	0.968	88.6	**0.963**	21.8	84.1	0.964	80.8	0.963	21.8
Diff-PGD Xue et al. (2023)	73.2	0.966	66.4	0.959	32.0	62.3	0.961	59.6	0.96	32.4
CtrlCamo	**18.3**	**0.972**	**27.5**	0.961	**7.00**	**13.7**	**0.972**	**11.1**	**0.964**	**7.67**

Table 2: **Quantitative comparison on the COCO Dataset for stylization-based digital attacks**. We report the AP, AP_{50}, SSIM, and average sampling time.

Method	Faster-RCNN							ViTDet						
	image-level			scene-level			Inf. Latency (s) ↓	image-level			scene-level			Inf. Latency (s) ↓
	AP (%) ↓	AP_{50}(%) ↓	SSIM ↑	AP (%) ↓	AP_{50}(%) ↓	SSIM ↑		AP (%) ↓	AP_{50}(%) ↓	SSIM ↑	AP (%) ↓	AP_{50}(%) ↓	SSIM ↑	
Normal	63.3	85.6	-	63.3	85.6	-	-	72.5	91.4	-	72.5	91.4	-	-
AdvCAM Duan et al. (2020)	45.0	64.7	0.680	44.7	66.4	0.651	21.8	48.5	69.7	0.678	47.6	67.9	0.650	21.8
Diff-PGD Xue et al. (2023)	45.0	61.1	0.692	47.1	64.6	0.664	33.2	53.0	70.4	0.677	53.1	70.8	0.663	34.2
CtrlCamo	**8.90**	**15.0**	**0.850**	**10.7**	**16.6**	**0.837**	**7.32**	**12.4**	**19.2**	**0.849**	**6.60**	**12.5**	**0.840**	**8.15**

Table 3: **Quantitative comparison with non-stylization digital attacks.** "Noise" and "Patch" denote adding random perturbations and patches on vehicles to conduct digital attacks, while "CIG" denotes camouflage image generation. We report the average AP and AP_{50} of both strategies.

Method	Type	LINZ		COCO			
		Faster-RCNN	ViTDet	Faster-RCNN		ViTDet	
		AP_{50} (%) ↓	AP_{50} (%) ↓	AP (%) ↓	AP_{50} (%) ↓	AP (%) ↓	AP_{50} (%) ↓
Normal	-	98.3	97.8	63.3	85.6	72.5	91.4
ToG Chow et al. (2020)	Noise	83.5	82.8	51.0	72.5	61.1	80.0
DiffAttack Chen et al. (2024)	Noise	98.0	97.5	56.8	77.7	63.9	83.2
NAP Hu et al. (2021)	Patch	89.2	88.6	56.3	75.3	68.6	87.1
BadPatch Wang et al. (2025)	Patch	96.5	88.7	53.3	73.8	66.3	84.4
LAKE-RED Zhao et al. (2024)	CIG	95.1	91.0	45.2	68.7	49.9	71.4
CtrlCamo	Style	**22.9**	**12.4**	**9.80**	**15.8**	**9.50**	**15.9**

latency (seconds per image) to characterize average adversarial image generation time across the dataset. More hyperparameter details are discussed in section C.1.

4.2 Results

Comparison with State-of-the-art Methods. We compare CtrlCamo with two categories of state-of-the-art methods. The first includes stylization-based approaches that, similar to ours, adapt vehicle appearance to reference areas. As shown in table 1 and table 2, our method achieves at least 38.9% AP_{50} reduction compared to both AdvCAM (Duan et al., 2020) and Diff-PGD (Xue et al., 2023) across datasets, detectors, and strategies, demonstrating a consistently stronger attack performance. In addition, our method achieves higher SSIM score on COCO, demonstrating superior preservation of vehicle structure. Moreover, these methods require per-image optimization during generation, leading to much higher inference times. Figure 3 presents qualitative comparisons evaluated on Faster-RCNN across diverse environments. Our approach effectively stylizes the vehicles based on the visual characteristics of the reference area while preserving their geometric structure. In contrast, Diff-PGD and AdvCAM rely on square patches that yield less coherent, context-aligned stylization. We further evaluate human-perceived reference consistency by asking participants to select the method whose stylization best matches the reference area. At image level, our method was preferred in 53.1% of cases, compared to 36.5% for Diff-PGD and 10.4% for AdvCAM. At scene level, preference for our method increased to 85.3%, versus 11.7% and 3.0%, respectively, indicating improved human-perceived visual consistency. Additional perceptual evaluations are provided in section D, including detailed study protocols, Sliced Wasserstein Distance in HSL space to quantify human-perceived reference consistency, and a separate human study of scene-context naturalness. The second category includes non-stylization digital attacks, such as imperceptible perturbations and adversarial patches, and camouflage image generation methods, as shown in table 3. Our pipeline achieves over 45% AP_{50} reduction across datasets and detectors, validating the effectiveness of our approach. This also reflects the two limitations in section 1: limited effectiveness of localized attacks and the lack of explicit detector optimization in camouflage image generation.

Transferability. We evaluate black-box transferability by generating adversarial camouflages using Faster-RCNN and ViTDet to unseen targets, as shown in table 4. Across both LINZ and COCO

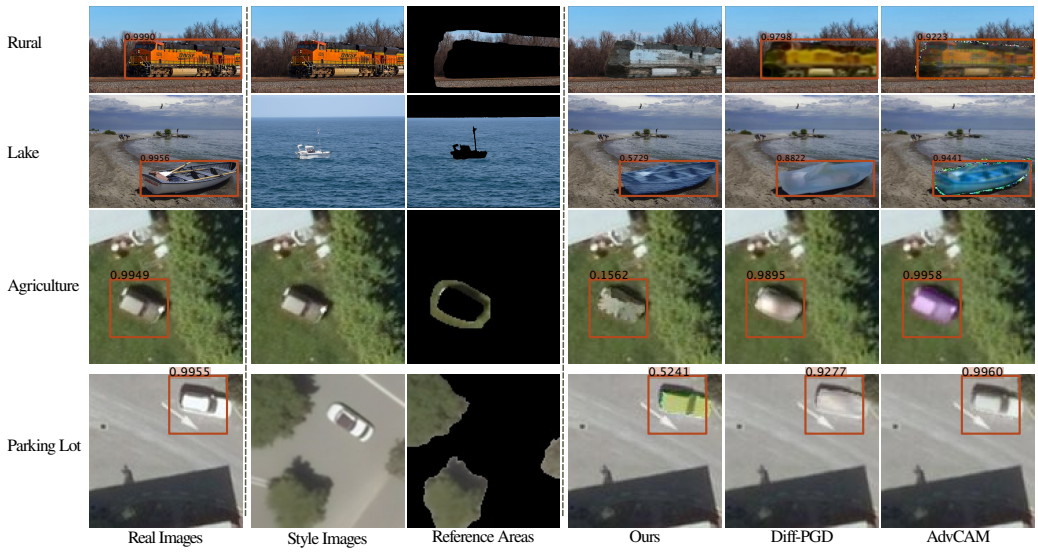


Figure 3: **Qualitative comparison with other methods.** The first two rows show results from the COCO dataset, and the last two rows are from the LINZ dataset. Within each dataset, the first row corresponds to the image-level strategy, and the second row corresponds to the scene-level strategy. Scene types are indicated on the left. In the "lake" scene, boats are stylized toward the water, while in the "parking lot" scene, cars are stylized toward trees. All camouflaged vehicles are composed with real images background.

Table 4: **Evaluation of adversarial camouflages across models.** We report the average of image-level and scene-level strategies using AP_{50} for detectors, and classification accuracy for MLLMs.

Dataset	Surrogate	Faster-RCNN	ViTDet	YOLOv5	YOLOv8	MLLM
LINZ	Normal	98.3	97.8	97.4	95.4	96.3
LINZ	Faster-RCNN	-	33.7	32.3	19.3	10.7
LINZ	ViTDet	11.4	-	38.6	25.2	11.3
COCO	Normal	85.6	91.4	88.3	90.7	87.5
COCO	Faster-RCNN	-	40.0	30.2	39.0	31.8
COCO	ViTDet	38.5	-	34.4	39.6	33.0

datasets, our method leads to at least a 47.1% drop in detection AP_{50} and a 54.5% decrease in classification accuracy when identifying vehicle presence in the image. These results indicate that the camouflage patterns are not overfitted to specific detectors but generalize across architectures, demonstrating robustness and cross-model transferability. Additional transferability analyses across scenes, views, and imaging conditions are provided in section E.

4.3 Ablation Study

Effectiveness of background supervision. We assess the role of background reconstruction loss L_b in the image-level setting introduced in section 3.3, as illustrated in fig. 4. As observed across both datasets, incorporating L_b encourages the model to stylize vehicles according to the visual characteristics of their surrounding context, resulting in more coherent and spatially consistent camouflage. Its perceptual effect is further quantified by the human ablation study in section D.3, where the complete model is preferred over ablation variants in 94.6% of COCO cases.

Effectiveness of color-consistency loss. We evaluate the effectiveness of the color-consistency loss L_c introduced in section 3.4. Incorporating L_c effectively constrains the color of the edited vehicle to remain consistent with the appearance established in the first stage. As shown in fig. 5(d) and (e) compared to (c), L_c suppresses color shifts introduced during adversarial optimization and better

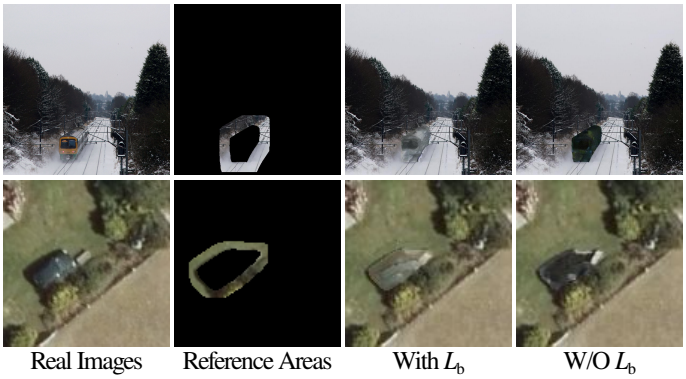


Figure 4: **Effectiveness of background supervision.**

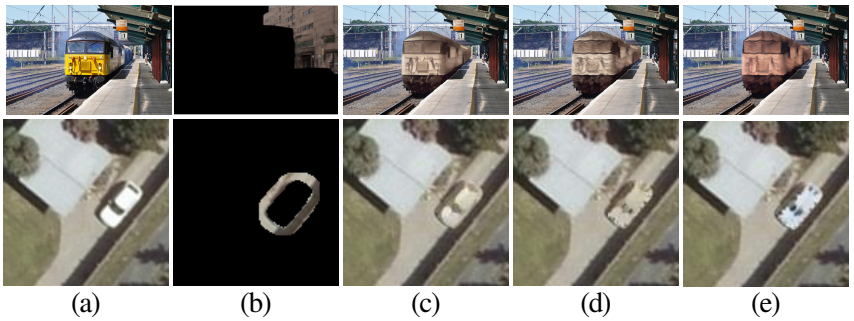


Figure 5: **Effectiveness of color-consistency loss.** (a) Real images. (b) Reference areas used for style guidance. (c) Camouflaged images generated in the "No-Box Attack" stage. (d) Camouflaged images generated with L_c. (e) Camouflaged images generated without L_c.

preserves the appearance learned in the first stage. Ablation studies on the two-stage paradigm and other losses are discussed in section C.5.

5 Conclusions

In this paper, we formulate appearance-based digital attacks against vehicle detectors as an adversarial camouflage editing problem and introduce *CtrlCamo*, a controllable diffusion-based framework for this setting. It supports two stylization strategies: an image-level strategy that adapts vehicle appearance to its surroundings, and a scene-level strategy that leverages visual concepts in the scene. Building on these strategies, we develop a two-stage framework that jointly stylizes vehicles while enforcing structural fidelity and adversarial effectiveness. Experiments on the LINZ and COCO datasets show that our method achieves stronger perceptual consistency, better preserves vehicle structure, and transfers to unseen black-box detectors. We also identify several limitations that point to future research directions. First, our pipeline operates in the digital image-editing domain and does not explicitly model surface appearance. We use the L channel in LAB color space as a coarse proxy for shading during optimization. However, in single-view images, the L channel entangles illumination, geometry, and material effects, preventing reliable disentanglement of surface reflectance. Thus, our method cannot explicitly manipulate the albedo corresponding to the unshaded surface texture. Future work could explore more accurate appearance decomposition to better model shading and surface properties. Second, the image-level strategy is less effective for ground-view images. Due to perspective projection, the dilated surroundings of a vehicle mask often include regions that are spatially distant from the target object and may contain multiple semantically diverse categories. This makes it difficult to learn a consistent style reference from the surrounding context, leading to less coherent camouflage patterns.

AI USE STATEMENT

In this work, we used generative AI tools to draft sections of the paper and polish writing. We have reviewed all AI-assisted work. LLM-generated paragraphs were carefully read and refined by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Jianqi Chen, Hao Chen, Keyan Chen, Yilan Zhang, Zhengxia Zou, and Zhenwei Shi. Diffusion models for imperceptible and transferable adversarial attack. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

Xinquan Chen, Xitong Gao, Juanjuan Zhao, Kejiang Ye, and Cheng-Zhong Xu. Advdiffuser: Natural adversarial example synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4562–4572, October 2023.

Ka Ho Chow, Ling Liu, Mehmet Emre Gursoy, Stacey Truex, Wenqi Wei, and Yanzhao Wu. TOG: targeted adversarial objectness gradient attacks on real-time object detection systems. *CoRR*, abs/2004.04320, 2020.

Xuelong Dai, Kaisheng Liang, and Bin Xiao. Advdiff: Generating unrestricted adversarial examples using diffusion models. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 93–109, Cham, 2025. Springer Nature Switzerland.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An Open Urban Driving Simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16, 2017.

Ranjie Duan, Xingjun Ma, Yisen Wang, James Bailey, A. Kai Qin, and Yun Yang. Adversarial Camouflage: Hiding Physical-World Attacks With Natural Styles. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 997–1005. Computer Vision Foundation / IEEE, 2020.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling Rectified Flow Transformers for High-Resolution Image Synthesis. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=FPnUhsQJ5B.

Xiao Fang, Minhyek Jeon, Zheyang Qin, Stanislav Panev, Celso De Melo, Shuowen Hu, Shayok Chakraborty, and Fernando De La Torre. Adapting Vehicle Detectors for Aerial Imagery to Unseen Domains with Weak Supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8088–8099, October 2025.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples, 2015. URL https://arxiv.org/abs/1412.6572.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc.,

2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Yu-Chih-Tuan Hu, Bo-Han Kung, Daniel Stanley Tan, Jun-Cheng Chen, Kai-Lung Hua, and Wen-Huang Cheng. Naturalistic physical adversarial patch for object detectors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Glenn Jocher. YOLOv5 by Ultralytics, 2020. URL https://github.com/ultralytics/yolov5.

Minguk Kang, Richard Zhang, Connelly Barnes, Sylvain Paris, Suha Kwak, Jaesik Park, Eli Shechtman, Jun-Yan Zhu, and Taesung Park. Distilling Diffusion Models into Conditional GANs. In *European Conference on Computer Vision (ECCV)*, 2024.

Jin Li, Ziqiang He, Anwei Luo, Jian-Fang Hu, Z. Jane Wang, and Xiangui Kang. AdvAD: Exploring Non-Parametric Diffusion for Imperceptible Adversarial Attacks. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 52323–52353. Curran Associates, Inc., 2024. doi: 10.52202/079017-1659. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/5de11e930c1bbfda5d4fc9d2b0924032-Paper-Conference.pdf.

Qizhang Li, Yiwen Guo, and Hao Chen. Practical No-box Adversarial Attacks against DNNs. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 12849–12860. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/96e07156db854ca7b00b5df21716b0c6-Paper.pdf.

Simin Li, Shuning Zhang, Gujun Chen, Dong Wang, Pu Feng, Jiakai Wang, Aishan Liu, Xin Yi, and Xianglong Liu. Towards Benchmarking and Assessing Visual Naturalness of Physical World Adversarial Attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12324–12333, June 2023.

Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring Plain Vision Transformer Backbones for Object Detection. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision – ECCV 2022*, pp. 280–296, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-20077-9.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision – ECCV 2014*, pp. 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow Matching for Generative Modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

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.

Yuanwei Liu, Hui Wei, Chengyu Jia, Ruqi Xiao, Weijian Ruan, Xingxing Wei, Joey Tianyi Zhou, and Zheng Wang. ProjAttacker: A Configurable Physical Adversarial Attack for Face Recognition via Projector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21248–21257, June 2025.

Grace Luo, Jonathan Granskog, Aleksander Holynski, and Trevor Darrell. Dual-process image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17972–17983, October 2025.

Linye Lyu, Jiawei Zhou, Daojing He, and YU LI. CNCA: Toward Customizable and Natural Generation of Adversarial Camouflage for Vehicle Detectors. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=aXNZG82IzV.

Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3523–3542, 2021.

OpenAI. GPT-4o System Card, 2024. URL https://arxiv.org/abs/2410.21276.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollar, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Ha6RTeWMd0.

Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

Dillon Reis, Jordan Kupec, Jacqueline Hong, and Ahmad Daoudi. Real-time flying object detection with YOLOv8. *arXiv preprint arXiv:2305.09972*, 2023.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis With Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025. URL https://arxiv.org/abs/2508.10104.

Vikhyat Korrapati. moondream2 (Revision 92d3d73). https://moondream.ai/, 2024.

Wenzhuang Wang, Yifan Zhao, Mingcan Ma, Yunlong Che, Haoran Chen, Ming Liu, and Jia Li. To blend in, first decouple: Rethinking camouflage image generation via context-decoupled representations. *arXiv preprint arXiv:2607.17768*, 2026.

Zhixiang Wang, Xingjun Ma, and Yu-Gang Jiang. BadPatch: Diffusion-Based Generation of Physical Adversarial Patches. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 6244–6254, October 2025.

Wikipedia contributors. Crypsis — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/Crypsis, 2025a. [Online; accessed 11-November-2025].

Wikipedia contributors. Mimicry — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/Mimicry, 2025b. [Online; accessed 11-November-2025].

Liangbin Xie, Daniil Pakhomov, Zhonghao Wang, Zongze Wu, Ziyan Chen, Yuqian Zhou, Haitian Zheng, Zhifei Zhang, Zhe Lin, Jiantao Zhou, and Chao Dong. TurboFill: Adapting Few-step Text-to-image Model for Fast Image Inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Haotian Xue, Alexandre Araujo, Bin Hu, and Yongxin Chen. Diffusion-Based Adversarial Sample Generation for Improved Stealthiness and Controllability. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding Conditional Control to Text-to-Image Diffusion Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3836–3847, October 2023.

Qing Zhang, Gelin Yin, Yongwei Nie, and Wei-Shi Zheng. Deep camouflage images. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 12845–12852, 2020.

Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful Image Colorization. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (eds.), *European Conference on Computer Vision (ECCV)*, pp. 649–666, 2016. ISBN 978-3-319-46487-9.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Yang Zhang, Hassan Foroosh, Philip David, and Boqing Gong. CAMOU: Learning Physical Vehicle Camouflages to Adversarially Attack Detectors in the Wild. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SJgEl3A5tm.

Pancheng Zhao, Peng Xu, Pengda Qin, Deng-Ping Fan, Zhicheng Zhang, Guoli Jia, Bowen Zhou, and Jufeng Yang. Lake-red: Camouflaged images generation by latent background knowledge retrieval-augmented diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11):3212–3232, 2019.

Jiawei Zhou, Linye Lyu, Daojing He, and YU LI. RAUCA: A Novel Physical Adversarial Attack on Vehicle Detectors via Robust and Accurate Camouflage Generation. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=pBTLGM9uWx.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models, 2025.

APPENDIX

The appendix provides additional details and experimental results to complement the main paper. Section A describes the procedure for selecting reference areas used in the scene-level stylization strategy. Section B extends our framework to rectified flow. Section C provides additional implementation details and experimental results, including qualitative examples, robustness under preprocessing defenses, and ablation studies. Section D provides further details on the human evaluation protocol and results presented in section 4.2, together with an additional human study comparing different ablation variants. Section E presents additional transferability experiments and analyses. Finally, section F discusses the limitations of our method in detail.

A STYLE REFERENCE SELECTION

We describe the procedure for selecting reference areas for the *scene-level* stylization strategy (see section 3.2) using the LINZ dataset as an example. As illustrated in fig. 6, the procedure consists of four stages.

First, we query MoonDream (Vikhyat Korrapati, 2024) for each LINZ image using the prompt "*Describe the scene type of this aerial view image.*" to obtain an initial scene-type prediction. Since the raw responses contain fine-grained and semantically overlapping categories, we consolidate them in the second stage. Specifically, we provide the full list of MoonDream-generated categories to GPT-4o (OpenAI, 2024) using the prompt "*Select a subset of scene groups that cover a wide variety of scene types and minimize semantic overlap between each category.*" GPT-4o consolidates the categories into five representative classes: *Residential, Industrial, Agricultural, Highway, and Parking lot* (Fig. 6, Step 2(1)). We then re-query MoonDream with the constrained prompt "*Describe the most fitted scene type of this aerial view image from [Residential, Industrial, Agricultural, Highway, Parking lot].*" to assign each image to one of these categories (Fig. 6, Step 2(2)).

After assigning scene types, the third stage extracts object-level information. We query MoonDream with the prompt "*Provide a comma-separated list of objects that are in this aerial view image.*" and aggregate object frequencies within each scene group. An example distribution for the *Residential* category is shown in fig. 6.

Finally, we select a representative visual concept for each scene type and synthesize a reference exemplar. We fine-tune Stable Diffusion (SD) v1.5 on LINZ training data using the template prompt "*an image of* $\{$*scene type*$\}$ *area with* $\{$*objects*$\}$", as described in section 4.1. For each scene group, we select a frequent visual concept and generate an exemplar containing that concept. For example, for the *Residential* scene, we select *house* and generate an image using "*An image of a residential area with car and house*." We then apply SAM 2 (Ravi et al., 2025) to segment the selected concept and use the resulting region as the reference area for *scene-level* stylization.

B EXTENSION TO RECTIFIED FLOW

Recently, flow matching (Lipman et al., 2023) has emerged as a powerful generative paradigm that learns a velocity field transporting a simple source distribution to the target data distribution. Among the variants built on flow matching, Rectified flow (Liu et al., 2023) views the forward process as a linear interpolation between the latent z_0 and the noise ε_t, for $t \in [0, 1]$:

$$z_t = (1 - t)z_0 + t\varepsilon_t,\ \varepsilon_t \sim \mathcal{N}(0, \mathbf{I}), \tag{9}$$

The network $\epsilon_\theta(z_t, c)$ learns to estimate the velocity $(\varepsilon_t - z_0)$. Similarly, inspired by (Luo et al., 2025), we adopt one-step estimate from noisy latent z_t to approximate the reverse process:

$$\hat{z_0} = z_t - t\epsilon_\theta(z_t, c) \tag{10}$$

Similarly, we evaluate our pipeline using SD v3.5 (Esser et al., 2024) as the underlying generative model and fine-tune a ControlNet (Zhang et al., 2023) in both stages with a batch size of 4. At inference time, we run the pipeline for 28 sampling steps, and all other settings follow the SD v1.5 configuration described in section 4.1. We conduct experiments on LINZ for the image-level strategy and on COCO for the scene-level strategy. Quantitative results comparing SD v1.5 and SD v3.5 are

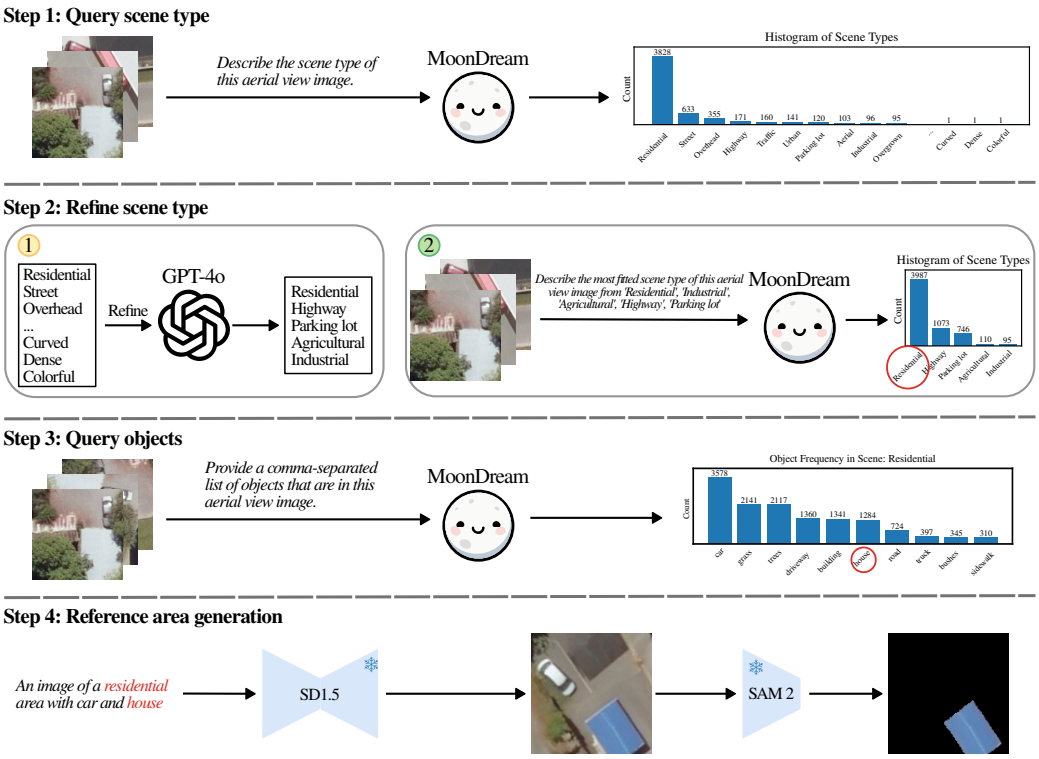


Figure 6: **Workflow for selecting style reference areas for the scene-level stylization strategy.** The process consists of four steps. In the first step, we query MoonDream for initial scene types. In the second step, we refine these categories using GPT-4o and recollect labels under five representative scene groups. In the third step, we query object-level information in each image. In the last step, we select a concept in each scene and synthesize an exemplar image containing that concept. We then segment the target concept with SAM 2. The extracted region serves as the reference area for scene-level stylization. Red circles denote we use the "house" concept in "Residential" area as an example to illustrate how we generate reference area in the last step.

Table 5: **Quantative comparison between SD v1.5 and v3.5.** The *image-level* strategy is evaluated on the LINZ dataset, while the *scene-level* strategy is evaluated on the COCO dataset. We report the AP_{50}, SSIM, and average sampling time per image.

Method	Faster-RCNN					ViTDet				
	LINZ (image-level)		COCO (scene-level)		Inf. Latency (s) ↓	LINZ (image-level)		COCO (scene-level)		Inf. Latency (s) ↓
	AP_{50}(%) ↓	SSIM ↑	AP_{50}(%) ↓	SSIM ↑		AP_{50}(%) ↓	SSIM ↑	AP_{50}(%) ↓	SSIM ↑	
Normal	98.3	-	85.6	-	-	97.8	-	91.4	-	-
Ours (SD v1.5)	18.3	0.972	16.6	0.837	7.32	13.7	0.972	12.5	0.840	8.15
Ours (SD v3.5)	17.0	0.978	27.9	0.845	4.86	8.8	0.978	29.7	0.844	4.88

reported in table 5. As shown in the table, adapting our pipeline to SD v3.5 remains effective across both settings. Qualitative comparisons on Faster-RCNN (Ren et al., 2015), presented in fig. 7 and fig. 8, further support these observations. As illustrated in the visualizations, SD v3.5 successfully transfers the appearance of the reference areas while maintaining vehicle structure and reducing detector confidence, similar to SD v1.5.

C Experiments

In this section, we provide additional experimental details and results. Section C.1 presents training parameters and implementation details for the baseline methods. Section C.2 provides additional qualitative results for both the image-level and scene-level stylization strategies on COCO and LINZ. Section C.3 evaluates an alternative set of scene-concept pairings to assess the generalizability of

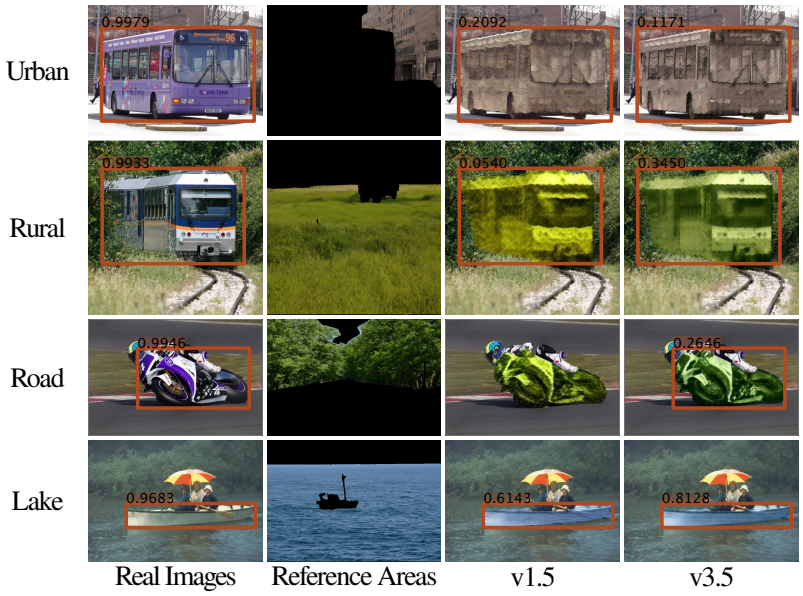


Figure 7: **Qualitative evaluation of SD v3.5 on COCO.** We visualize scene-level adversarial camouflage editing results. Each row corresponds to a scene type. Numbers above the bounding boxes indicate detector confidence, and the absence of a box indicates a missed detection. All camouflaged vehicles are composited onto the original real-image backgrounds.



Figure 8: **Qualitative evaluation of SD v3.5 on LINZ.** We visualize image-level adversarial camouflage editing results.

the scene-level strategy. Section C.4 evaluates robustness under image preprocessing defenses. Section C.5 provides additional ablation studies on the two-stage framework and its training objectives.

Table 6: **Training parameters of each experiment**. "lr" denotes learning rate. "iter" denotes the total number of training iterations. "L_{struct}, L_{s}, L_{b}, L_{c}, L_{adv}" denotes the coefficients of structure preservation loss, style loss, background reconstruction loss, color-consistency loss, and adversarial loss, respectively.

Dataset	Target	Strategy	No-Box Attack					White-Box Attack						
			L_{struct}	L_{s}	L_{b}	lr	iter	L_{struct}	L_{s}	L_{b}	L_{c}	L_{adv}	lr	iter
COCO	Faster-RCNN	Scene-Level	5.0	1.0	0.0	5e-6	12000	5.0	1.0	0.0	2.0	1.0	4e-6	15000
COCO	Faster-RCNN	Image-Level	2.5	2.5	1.0	2e-6	15000	4.0	1.0	1.0	2.0	1.0	2.5e-6	20000
COCO	ViTDet	Scene-Level	5.0	1.0	0.0	5e-6	12000	5.0	1.0	0.0	2.0	1.0	4e-6	15000
COCO	ViTDet	Image-Level	2.5	2.5	1.0	2e-6	15000	10.0	2.5	1.0	2.5	1.0	4e-6	20000
LINZ	Faster-RCNN	Scene-Level	4.0	1.0	0.0	2.5e-6	10000	10.0	2.5	0.0	2.5	1.0	2e-6	12000
LINZ	Faster-RCNN	Image-Level	5.0	5.0	1.0	2e-6	12000	10.0	7.5	1.0	7.5	1.0	2e-6	15000
LINZ	ViTDet	Scene-Level	4.0	1.0	0.0	2.5e-6	10000	10.0	2.5	0.0	2.5	1.0	2e-6	12000
LINZ	ViTDet	Image-Level	5.0	5.0	1.0	2e-6	12000	10.0	7.5	1.0	7.5	1.0	2e-6	15000

C.1 Experimental Setup

In section 4.1, we describe the training settings shared across all experiments, including batch size and the dilation kernel sizes used for COCO and LINZ. Here, we provide additional hyperparameters, including learning rates, training iterations, and loss coefficients, as summarized in table 6. Empirically, we find that assigning larger weights to the non-adversarial objectives than to the adversarial loss improves stylization and structure preservation while maintaining strong attack effectiveness. During inference, images are generated one at a time, and all methods are evaluated on a single RTX A6000 GPU. We also provide implementation details for the baselines in tables 2 and 3, grouped into four categories following section 2: *imperceptible perturbations*, *adversarial patches*, *stylization-based digital attacks*, and *camouflage image generation*.

- *Imperceptible perturbations*: TOG (Chow et al., 2020) directly optimizes bounded perturbations in RGB space without requiring a style reference. We constrain the perturbation norm to $\frac{100}{255}$ and perform 80 optimization iterations per image. DiffAttack (Chen et al., 2024), in contrast, performs adversarial optimization in the latent space of a diffusion model to preserve perceptual fidelity. We use a 50-step diffusion process followed by 60 adversarial optimization steps. Since DiffAttack was originally designed for image classification, we adapt it to object detection by replacing its adversarial objective with our detector-specific loss in Eq. 7.
- *Adversarial patches*: For NAP (Hu et al., 2021), we optimize a GAN-generated patch shared across all images. Following the original implementation, we train the patch separately on the LINZ and COCO training sets for 100 epochs with a learning rate of 1×10^{-4}, and then apply the optimized patch to the corresponding test sets. The patch is scaled to remain fully within the target object region. BadPatch (Wang et al., 2025) follows a similar pipeline but initializes from a natural image patch and optimizes it in the noisy diffusion latent space through DDIM inversion. We use 25 inversion steps and train the patch for 20 epochs with a learning rate of 1×10^{-4} on both datasets before evaluation.
- *Stylization-based digital attacks*: AdvCAM (Duan et al., 2020) and Diff-PGD (Xue et al., 2023) both require a reference patch to guide stylization. Since their pipelines accept only square reference patches, we use a unified reference-selection strategy: for image-level stylization, we select a square patch adjacent to the target vehicle; for scene-level stylization, we crop a square patch from the corresponding reference area. As both methods were originally designed for classifier attacks, we replace their adversarial objectives with our detector-specific loss in Eq. 7. We perform 1,000 optimization iterations for each image to obtain the final result.
- *Camouflage image generation*: LAKE-RED (Zhao et al., 2024) generates camouflage by retrieving background appearance information from a learned codebook and inpainting the masked vehicle region, without explicitly optimizing an adversarial objective. We initialize the model using the officially released pretrained LDM-inpainting backbone and VQ-f4 autoencoder, and initialize the Background Knowledge Retrieval-Augmented module by copying the autoencoder codebook into its background embedding, following the original implementation. We then fine-tune the full model separately on the COCO and LINZ training sets using ground-truth object masks for 40 epochs, with a batch size of 2 and a



Figure 9: **More Visualizations on the COCO dataset.** We visualize scene-level adversarial camouflage editing results. We present examples in *urban, rural, road, sky*, and *lake* scenes, where they are associated with *building, grass, tree, sky*, and *water* as representative concepts.

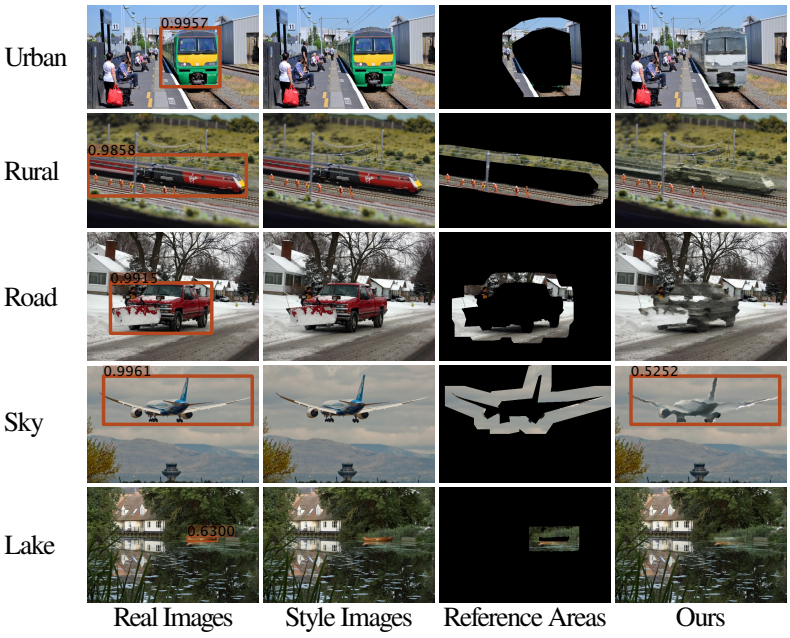


Figure 10: **More Visualizations on the COCO dataset.** We visualize image-level adversarial camouflage editing results.

resolution of 512×512. At inference time, each masked region is generated using 50-step DDIM sampling and composited back into the original image, while pixels outside the vehicle mask remain unchanged.

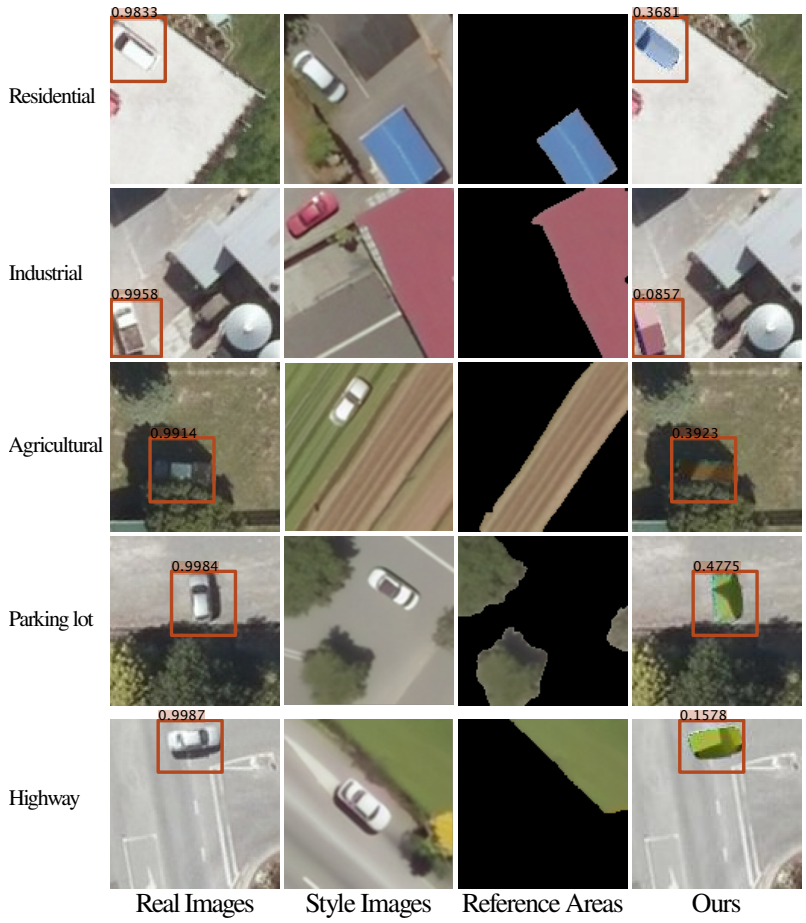


Figure 11: **Visualizations on the LINZ dataset.** We present examples in *residential, industrial, agricultural, parking lot,* and *highway* scenes, where they are matched with *house, building, field, tree*, and *grass* as representative concepts.

C.2 Extended Qualitative Results

We provide additional visualization results to complement fig. 3, covering both scene-level and image-level camouflage generation on the COCO dataset in fig. 9 and fig. 10. Scene-level examples span *urban, rural, road, sky*, and *lake* environments paired with visual concepts such as *building, grass, tree, sky*, and *water*. Image-level examples use reference areas extracted from the vehicle's immediate surroundings. Across both settings, the camouflaged vehicles blend naturally with the scene while retaining their geometric structure, demonstrating consistent stylization behavior. Additional visualizations for LINZ are shown in fig. 11 and fig. 12, covering *residential, industrial, agricultural, parking lot*, and *highway* environments. Scene-level stylization follows representative concepts like *house, building, field, tree*, and *grass*, while image-level stylization leverages local surroundings. In both cases, our method produces camouflaged vehicles that maintain realistic structure and visual coherence with their surroundings. Together, these visualizations corroborate that the proposed adversarial camouflage editing approach generalizes across datasets and stylization modes, consistently producing structurally faithful and stealthy camouflage.

C.3 Scene-Level Strategy

As described in section 1, the scene-level strategy adapts the vehicle's appearance to a common semantic concept drawn from the broader scene, ensuring that the resulting camouflage remains visually coherent and avoids salient or unnatural patterns that may draw human attention. In section 4.2, we evaluate one set of scene–concept pairings for both datasets. To assess the generaliz-

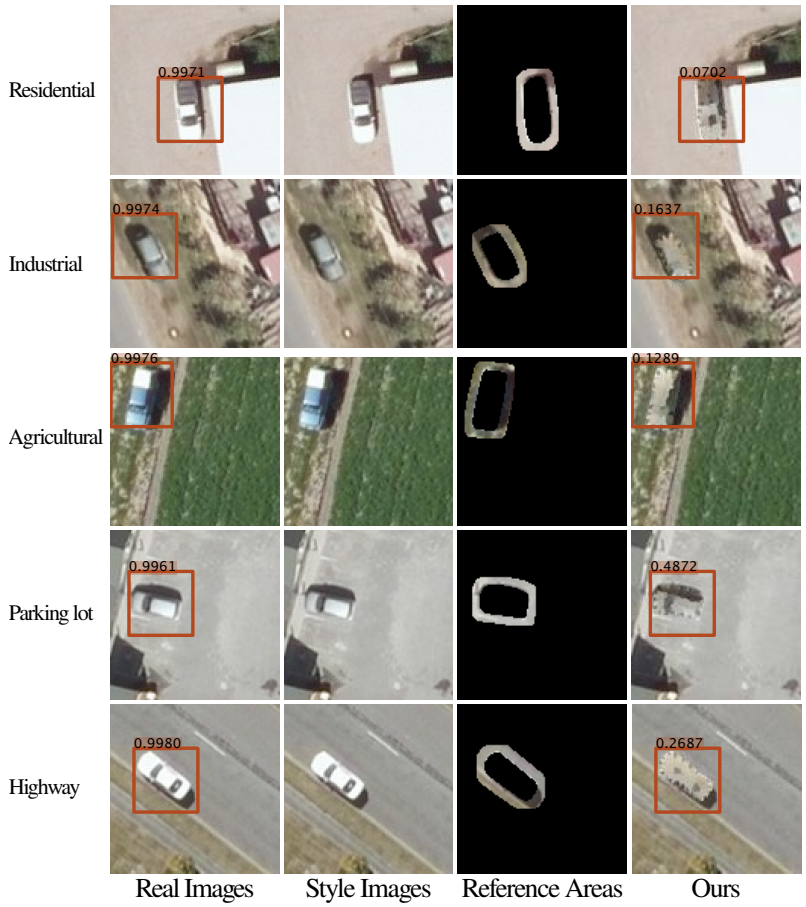


Figure 12: **More Visualizations on the LINZ dataset.** We visualize image-level adversarial camouflage editing results.

Table 7: **Quantitative evaluation of alternative scene–concept pairings for the scene-level strategy on COCO and LINZ.** "Ours" denotes the scene–concept groups described in section 4.1, while "Ours*" adopts the alternative set evaluated in section C.3.

Dataset	Method	Faster-RCNN		ViTDet	
		$AP_{50}(\%) \downarrow$	SSIM ↑	$AP_{50}(\%) \downarrow$	SSIM ↑
COCO	Normal	85.6	-	91.4	-
COCO	Ours	16.6	0.837	12.5	0.840
COCO	Ours*	21.1	0.846	20.2	0.845
LINZ	Normal	98.3	-	97.8	-
LINZ	Ours	27.5	0.961	11.1	0.964
LINZ	Ours*	24.8	0.966	14.5	0.969

ability of this strategy, we conduct an additional experiment on both datasets using an alternative set of scene-concept pairings. Following section 4.1, COCO images are assigned to five environments: *urban, rural, road, sky*, and *lake*. In section 4, these environments are paired with *building, grass, tree, sky*, and *water* as representative style concepts. In this section, we additionally evaluate an alternative set of pairings that matches the same environments with *platform, tree, road, cloud*, and *beach*. Similarly, LINZ images are assigned *residential, industrial, agricultural, parking lot*, and *highway* scenes, originally associated with *house, building, field, tree*, and *grass*. In this section, we also evaluate an alternative set of pairings that matches these environments with *grass, sidewalk, parking lot, dirt*, and *parking lot*. The quantitative results in table 7 demonstrate that both pairing sets achieve comparable attack effectiveness and preservation of vehicle structure for both datasets.

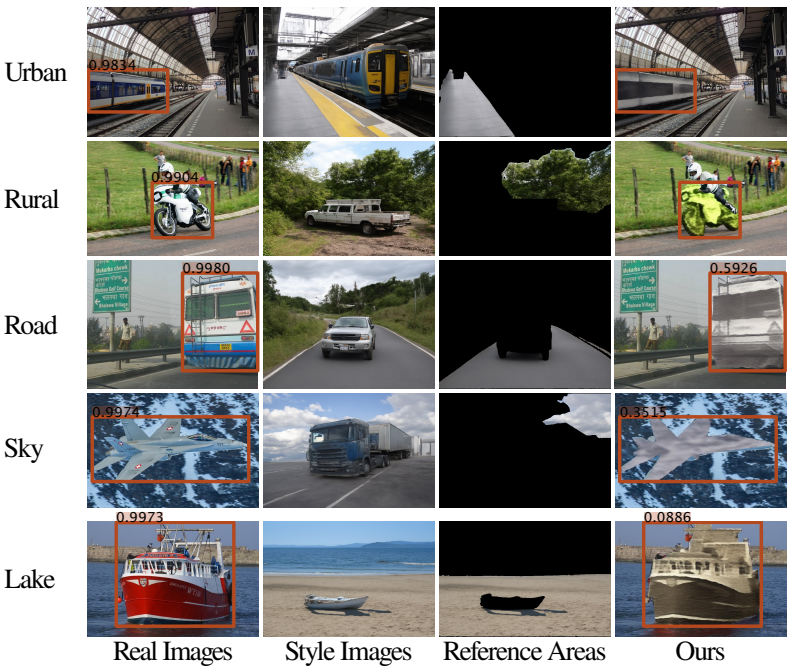


Figure 13: **Visualizations on the COCO dataset.** We present examples in *urban, rural, road, sky*, and *lake* scenes, where they are matched with with *platform, tree, road, cloud*, and *beach* as representative concepts.

Table 8: **Evaluation under preprocessing-based defense strategies on the COCO dataset.** "Normal" denotes clean images, "Baseline" denotes adversarial images without defense, "Denoise" applies Non-local Means denoising, and "Smooth" applies bilateral filtering.

Method	Faster-RCNN		ViTDet	
	image-level	scene-level	image-level	scene-level
Normal	85.6	85.6	91.4	91.4
Baseline	15.0	16.6	19.2	12.5
Denoise	29.8	28.2	36.4	27.4
Smooth	21.4	25.4	25.5	23.0

Additional qualitative examples in fig. 13 and fig. 14 further confirm this consistency. Overall, adapting vehicles to scene-relevant visual concepts produces coherent and stealthy camouflage, though occasional style misalignment between camouflaged vehicles and reference areas occurs, which is further discussed in section F.

C.4 Robustness under Defense Strategies

We evaluate whether adversarial images generated by our method remain effective after common preprocessing operations, including Non-local Means denoising and bilateral filtering, on COCO. We use OpenCV Non-local Means denoising with $h = 3$, $h_{\text{Color}} = 3$, a 7×7 template window, and a 21×21 search window, and bilateral filtering with $d = 9$, $\sigma_{\text{color}} = 50$, and $\sigma_{\text{space}} = 50$. As shown in table 8, these defenses reduce attack effectiveness. Nevertheless, our method consistently produces substantial AP_{50} drops relative to the clean images across both Faster R-CNN and ViTDet and under both image-level and scene-level stylization strategies. For example, AP_{50} decreases from 85.6% to 29.8% under denoising for Faster R-CNN and from 91.4% to 36.4% for ViTDet using the image-level strategy. Under smoothing, AP_{50} remains below 26% in all cases. These results indicate that the adversarial effects of our method remain substantial after standard denoising and smoothing operations. We attribute this robustness to the appearance-level modifications introduced by our method rather than fragile high-frequency perturbations.

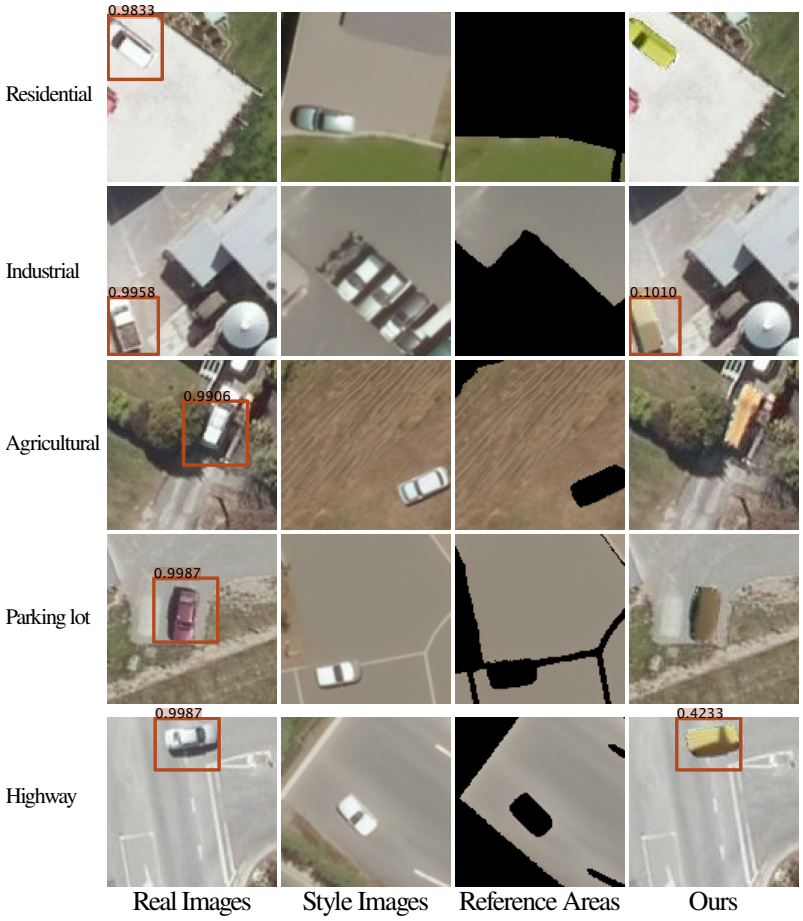


Figure 14: **Visualizations on the LINZ dataset.** We present examples in *residential, industrial, agricultural, parking lot,* and *highway* scenes, where they are matched with with *grass, sidewalk, parking lot, dirt*, and *parking lot* as representative concepts.

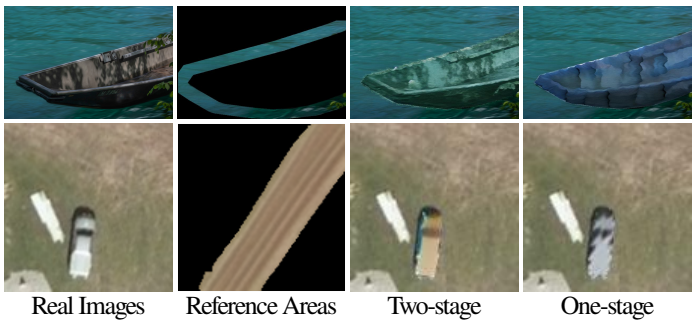


Figure 15: **Comparing our pipeline with one-stage variants**.

C.5 Ablation Studies

Two-stage vs. One-stage. We evaluate the effectiveness of our proposed two-stage pipeline against a one-stage variant that jointly fine-tunes the ControlNet using the structure preservation loss L_{struct}, style loss L_{s}, adversarial loss L_{adv}, and background supervision loss L_{b} under the image-level setting, as illustrated in fig. 15. The two-stage pipeline achieves more accurate stylization of vehicles according to the visual characteristics of the reference area. This improvement arises because, early in the training, the pipeline often fails to reconstruct the vehicle at its original location via eq. (2),

Table 9: **Effectiveness of adversarial loss.**

Dataset	Method	Faster-RCNN		ViTDet	
		image-level	scene-level	image-level	scene-level
COCO	Normal	85.6	85.6	91.4	91.4
COCO	W/O L_{adv}	67.2	65.1	73.6	72.1
COCO	Ours	15.0	16.6	19.2	12.5
LINZ	Normal	98.3	98.3	97.8	97.8
LINZ	W/O L_{adv}	92.4	77.2	85.2	52.9
LINZ	Ours	18.3	27.5	13.7	11.1

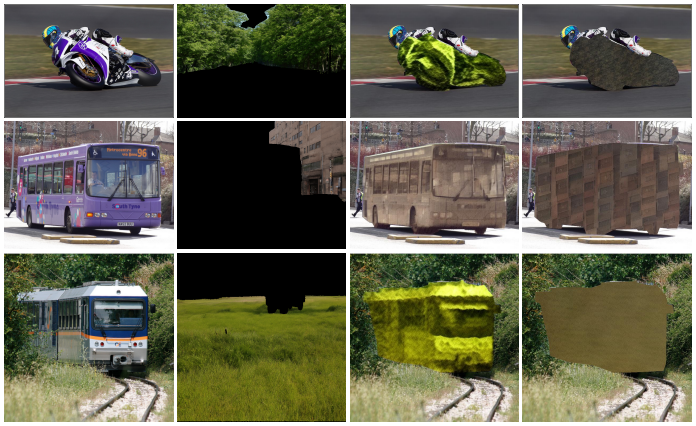


Figure 16: **Effectiveness of structure preservation loss.**

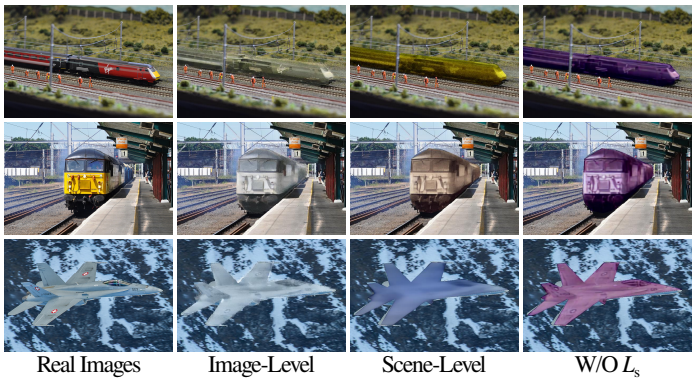


Figure 17: **Effectiveness of style loss.**

producing arbitrary content instead. Consequently, the adversarial loss L_{adv} becomes misleading, since detectors already assign low confidence to non-vehicle regions, which does not reflect true adversarial success. This inaccurate supervision interferes with the optimization of other objectives and leads to unstable training. By first ensuring reliable vehicle reconstruction before applying adversarial loss, the two-stage design stabilizes optimization and improves camouflage quality.

Effectiveness of structure preservation loss. We evaluate the effectiveness of structure preservation loss L_{struct} for the scene-level strategy on the COCO dataset, as shown in fig. 16. When the structure-preservation loss is removed, the model produces heavily distorted shapes that no longer resemble the underlying object. Although the stylization remains roughly consistent with the reference area, the resulting camouflages become unrealistic, demonstrating that L_{struct} is essential for preserving vehicle structure while adapting appearance.

Table 10: **Color-distribution similarity between edited vehicles and their reference areas.** We measure Sliced Wasserstein Distance (SWD) in HSL space. Lower is better.

Method	COCO		LINZ	
	image-level	scene-level	image-level	scene-level
AdvCAM Duan et al. (2020)	0.120	0.201	0.098	0.184
Diff-PGD Xue et al. (2023)	*0.116*	0.184	0.081	0.181
CtrlCamo	0.121	*0.107*	*0.074*	*0.109*

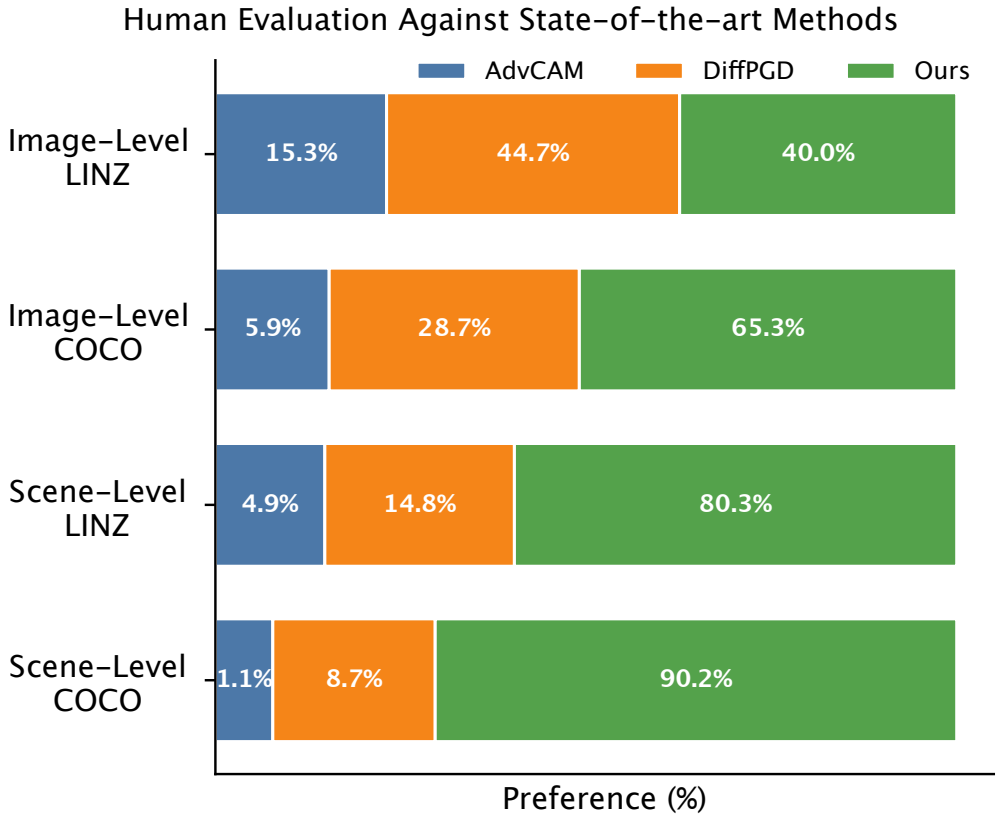


Figure 18: **Human Evaluation of stealthiness comparing with state-of-the-art methods.**

Effectiveness of style loss. We evaluate the effectiveness of style loss L_s on the COCO dataset, as illustrated in fig. 17. When the style loss is removed, the pipeline tends to learn a similar stylization for all vehicles, often producing colors or textures that appear unnatural and may draw human attention. In contrast, by guiding the model to transfer each vehicle's appearance either to its surrounding region in the image level setting or to an appropriate visual concept present in the scene in the scene level setting, the style loss encourages camouflages that remain consistent with the environment.

Effectiveness of adversarial loss. We evaluate the effectiveness of the adversarial loss L_{adv} introduced in section 3.4. Removing this term is equivalent to performing only the No-Box Attack stage described in section 3.3. Admittedly, style-based editing alone reduces detector confidence by altering the vehicle appearance, as shown in table 9. However, our stylization procedures are not designed to introduce patterns that deviate significantly from the distribution of the dataset. Consequently, incorporating the adversarial loss further decreases detector confidence by explicitly optimizing for adversarial behavior.

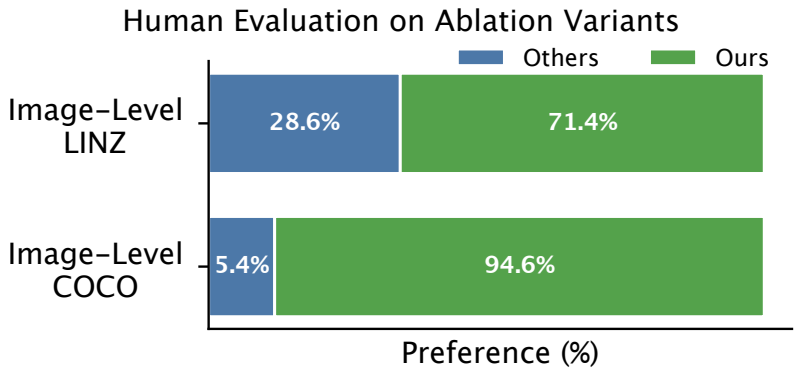


Figure 19: **Human Evaluation of stealthiness across ablation variants.**

D Perceptual Stealthiness Evaluation

In this section, we provide additional evaluations of perceptual stealthiness. In section D.1, we first quantify human color perceptual similarity to reference images using Sliced Wasserstein Distance in HSL space. In section D.2, we then compare CtrlCamo with state-of-the-art methods through a human study based on perceived consistency with the reference area. Finally, section D.3 evaluates the perceptual effect of our design choices through a separate human study on ablation variants.

D.1 Quantitative Reference-Area Color Similarity

We first quantitatively evaluate the similarity between edited vehicle regions and their reference areas using Sliced Wasserstein Distance (SWD) in HSL space. A lower distance indicates closer alignment between their color distributions. As shown in table 10, CtrlCamo achieves the lowest SWD in three of the four settings, including both scene-level settings and image-level stylization on LINZ. The only exception is the COCO image-level setting, where its distance is comparable to the competing methods. This is consistent with the ground-view limitation discussed in section 5, where perspective variation can make local reference selection less reliable. Overall, these results indicate stronger reference-area color consistency in most settings.

D.2 Human Evaluation Against State-of-the-art Methods

We provide additional details for the user study described in section 4.2, which evaluates human-perceived consistency with the reference area against state-of-the-art methods. The questionnaire is divided into two parts corresponding to the two camouflage strategies: image-level and scene-level. The order of these two parts is randomized for each participant. The question pool contains 15 images from the LINZ dataset and 15 images from COCO. There are three images in each scene type. For each image, camouflaged results are generated using both image-level and scene-level strategies for all compared methods. The presentation order of the three methods is randomized for each question. In each part, participants are first presented with a guideline explaining the evaluation task, followed by 15 questions randomly sampled from the pool. Participants are asked to select the method whose stylization best matched the reference areas. The guidelines for both parts are shown in fig. 27 and fig. 30. Example questions for both datasets and both strategies are shown in fig. 28, fig. 31, fig. 29, and fig. 32. The aggregated preference results are shown in fig. 18. On the LINZ dataset under the image-level strategy, our method achieves a preference rate comparable to Diff-PGD (Xue et al., 2023). Across the remaining settings, including image-level on COCO and scene-level strategies on both datasets, our method receives substantially higher preference rates than competing approaches. These results indicate stronger human-perceived consistency between the edited vehicles and their reference areas.

D.3 Human Evaluation Against Ablation Variants

We conduct a separate human study to analyze the effect of different design choices on appearance naturalness. Direct comparison of this property with previous methods is challenging because their outputs may differ in multiple aspects beyond appearance style, such as structural preservation and adversarial effectiveness (see table 1 and table 2). These factors can influence human perception of style realism and therefore confound the evaluation of perceptual stealthiness (Li et al., 2023). Therefore, we focus on ablation variants of our image-level strategy by selectively removing key loss terms. This setting ensures that all edited vehicles maintain comparable structural consistency and adversarial effectiveness, so that the primary variation lies in the resulting appearance style. For the COCO dataset, we compare against variants that remove the style loss or the background reconstruction loss. For the LINZ dataset, we compare against variants that remove the style loss and use the single-stage generation paradigm. We use the same image pool as in section D.2. Participants are first provided with a guideline explaining the evaluation task, followed by 10 questions randomly sampled from the image pool. In each question, participants are shown the real image together with three edited vehicles and asked to select the version that appears most natural within the scene context. The order of the three edited variants is randomized for each question. Example questions are illustrated in fig. 34 and fig. 33.The aggregated results are summarized in fig. 19. Our method achieves substantially higher preference rates than other ablation variants on both datasets, receiving 71.4% preference on LINZ and 94.6% on COCO. These results indicate that the complete formulation of our method produces camouflage patterns that better match the surrounding environment and are consistently perceived as more natural and stealthy by human evaluators.

E Transferability

In this section, we provide additional transferability analyses of our method. In section E.1, we evaluate whether scene-level stylization transfers across different locations within the same scene category. In section E.3, we further conduct a projector-based study to evaluate whether the adversarial effect of digitally edited images transfers under a different imaging process involving projection and camera recapture.

E.1 Cross-Location Transferability

To support our claim in section 1 that scene-level stylization can be applied consistently across an entire scene, we evaluate whether the learned camouflage generalizes to different locations within the same scene type. We conduct experiments on the COCO dataset targeting Faster R-CNN. Specifically, for each scene type, we select two vehicles and composite them into five additional backgrounds using PhotoShop by replacing only the background while preserving the vehicle appearance. We use the same compositing pipeline for both clean and camouflaged vehicles, without additional blending or appearance harmonization. The new scenes introduce diverse environmental conditions, including different weather (e.g., cloudy, rainy, and foggy) and seasonal variations (e.g., summer and winter). Representative examples are shown in fig. 20. As illustrated, Faster R-CNN achieves an AP_{50} of 99.5% on the corresponding clean images, whereas the camouflaged versions reduce performance to 38.2%, corresponding to a 61.3% drop. These results indicate that the learned scene-level camouflage remains adversarial even when the vehicle is placed at different locations within the same scene category, demonstrating strong cross-location transferability under diverse environmental conditions.

E.2 Cross-View Transferability

Beyond changes in scene location, viewpoint variation provides another important axis of transferability. Since our method edits images independently, it is unclear whether the resulting adversarial appearance remains effective when the same object is observed from unseen viewpoints. We therefore conduct a simulation study to evaluate cross-view transferability.

We use the Audi E-tron dataset (Zhou et al., 2024), which contains 20,000 rendered views of the same vehicle generated in CARLA (Dosovitskiy et al., 2017) under different weather conditions, together with the vehicle mesh and corresponding camera parameters. We randomly



Figure 20: **Transferability of scene-level adversarial camouflage across locations in COCO dataset.** For each scene type, we composite the vehicle into five additional backgrounds to create synthetic scenes while keeping the vehicle appearance unchanged. In each scene group, the first row shows detection results on clean vehicles, and the second row shows results on camouflaged vehicles. "Original" denotes the original COCO scene, whereas "Synthetic" refers to the newly composed scenes with replaced backgrounds.

split the dataset into 15,000 training and 5,000 testing views. For each training view, we independently generate an adversarial edit through a white-box attack against YOLOv3 (Redmon & Farhadi, 2018), without using geometric correspondence or information from other viewpoints. We adopt YOLOv3 following the setup commonly used in prior texture-based camouflage stud-



(a) (b) (c)

Figure 21: **Projector-based setup for the LINZ dataset.** (a) Device configuration with the projector and camera aligned to minimize parallax. (b) A real image is projected onto a whiteboard with a 3D-printed car model placed at the corresponding location. (c) Alternative view showing attachment of the printed model to the board.



(a) (b) (c) (d)

Figure 22: **Projector-based setup for the COCO dataset.** (a) Hardware configuration. (b) 3D-printed scene reconstructed from two COCO images via single-view reconstruction. (c-d) Real images projected onto the corresponding 3D-printed models.

ies (Lyu et al., 2024; Zhou et al., 2024). We then optimize a shared texture on the fixed vehicle mesh using a PyTorch-based differentiable Neural Renderer and the camera utilities following (Zhou et al., 2024). The mesh geometry remains fixed, and only texture values on predefined exterior faces are optimized. We use a per-face texture resolution of 6 and optimize the shared texture for up to 1,500 steps using Adam with a learning rate of 0.01 and batch size 32. The texture is optimized using a masked L_1 reconstruction loss between the rendered vehicle and the independently edited training views. At test time, we render the fitted texture from held-out viewpoints using the corresponding camera parameters and evaluate the rendered images with the same YOLOv3 detector. YOLOv3 achieves an AP50 of 72.0% on clean views and 40.3% on views rendered with the fitted adversarial texture, corresponding to a 31.7 percentage-point decrease. These results show that the shared texture distilled from independently generated 2D edits retains substantial adversarial effectiveness on unseen viewpoints, indicating meaningful cross-view transferability.

E.3 Transferability under Imaging Variations

Digital adversarial effects may be sensitive to changes introduced by the image formation process, such as illumination variation, color shifts, and camera capture artifacts. We therefore conduct a projector-based study to examine whether the adversarial effect produced by our digital editing pipeline persists under such imaging variations.

Following (Liu et al., 2025), we construct projector-based setups for both LINZ and COCO, targeting Faster R-CNN. We use an Epson EpiqVision^{TM} Mini EF12 projector and an iPhone 16 Pro Max for image capture. The camera is positioned close to the projector's optical axis and the intended

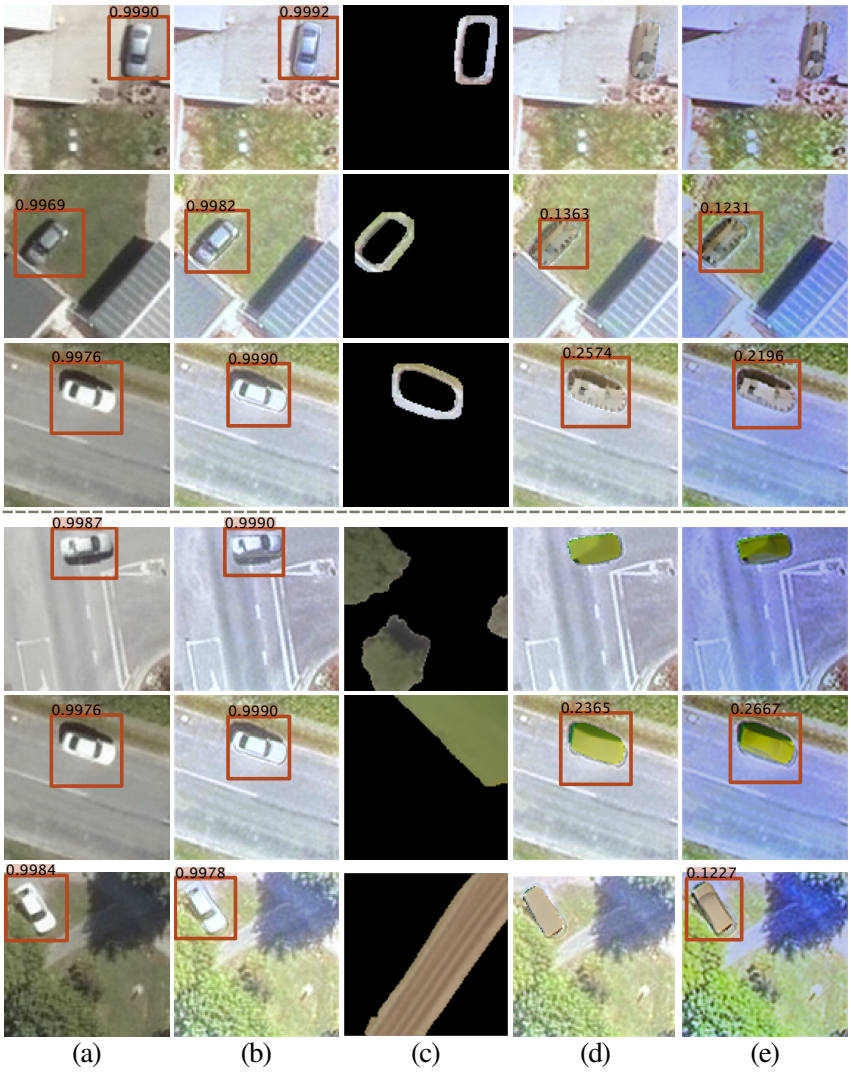


Figure 23: **Projector-based experiment for the LINZ dataset.** The first three rows present examples produced using the image-level strategy, and the last three rows illustrate the scene-level strategy. (a) Real images in digital space. (b) Photos captured from real images. (c) Reference areas used for style guidance. (d) Camouflaged images generated from the captured photos. (e) Photos taken after projecting the camouflaged images back onto the 3D car models.

viewing direction to reduce parallax and perspective differences. All experiments are conducted under near-dark illumination to increase projection contrast. For LINZ, we place a 3D-printed sedan model at the corresponding vehicle location on a whiteboard, as shown in fig. 21. For COCO, we reconstruct scene geometry from selected images using monocular depth estimation (Lin et al., 2025) and 3D-print the reconstructed scene, as shown in fig. 22. The printed models use matte PLA filament and have a maximum length of approximately 180 mm.

For the LINZ dataset, we evaluate five images for the scene-level strategy and four for the image-level strategy. For the COCO dataset, we evaluate three images for each strategy. Visualizations are shown in fig. 23 and fig. 24. We first project and recapture the original images to establish the effect of the imaging process itself. Despite moderate illumination and color changes, detector confidence remains similar to that of the original digital images, as shown in fig. 23 (a) and (b). We then apply our digital editing pipeline to the recaptured images, generate image-level and scene-level edited results, as shown in fig. 23 (d). Finally, we project these results onto the same setups for a second capture, as shown in fig. 23 (e). Because the number of test images is small, reporting AP_{50} is not

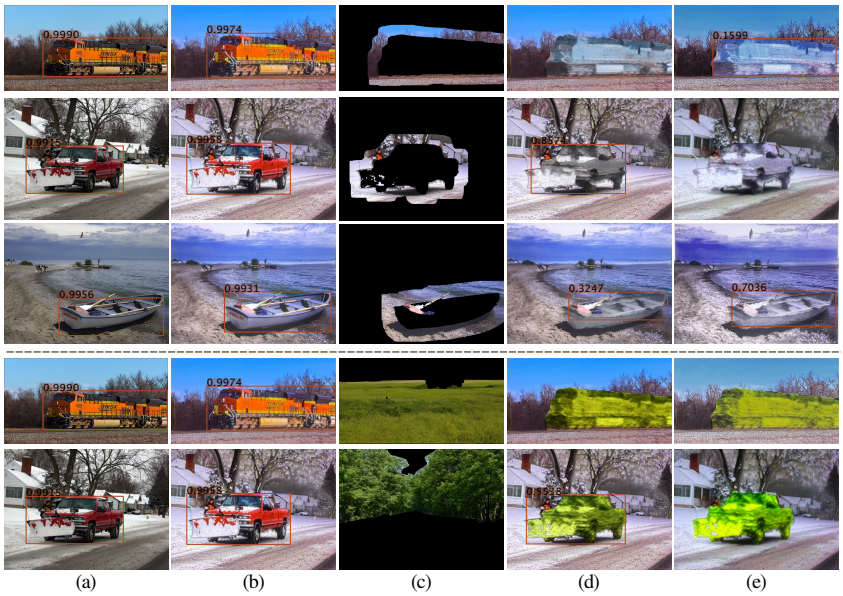


Figure 24: **Projector-based experiment for the COCO dataset.** The first three rows present examples produced using the image-level strategy, and the last three rows illustrate the scene-level strategy. (a) Real images in digital space. (b) Photos captured from real images projected on the 3D-printed scenes. (c) Reference areas used for style guidance. (d) Camouflaged images generated from the captured photos. (e) Photos taken after projecting the camouflaged images back onto the 3D-printed scenes.

reliable. With limited samples, a single prediction can significantly influence the precision–recall curve, and the resulting value does not reflect stable detector performance. Therefore, we instead report the *attack success rate*, which depends only on the detector confidence associated with the ground-truth object. To determine a reliable confidence threshold, we evaluate the detector on the validation set and select the confidence value that yields the best F1 score. The resulting threshold is 0.948 for LINZ and 0.806 for COCO. A detection is considered positive only if its confidence exceeds the threshold and its bounding box has IoU greater than 0.5 with the ground-truth vehicle. Under this criterion, all LINZ samples produce confidence scores below 0.55 for scene-level, and below 0.22 for image-level camouflage. Both values are far below the 0.948 threshold. For COCO samples, scene-level camouflaged vehicles are not detected, while image-level predictions have confidence scores below 0.71, also below the 0.806 threshold.

F LIMITATIONS

First, our framework relies on the L channel in LAB space as a coarse approximation for the shading component, which limits its ability to preserve shading independently from stylization. Consequently, the camouflaged vehicles do not perfectly retain their original shading patterns, as observed in fig. 14, fig. 11, and fig. 12. Since the L channel also carries texture information, the joint optimization of the structure preservation and style losses may interact during training, introducing subtle shifts in the vehicle's final appearance. These effects are visible in fig. 14 and fig. 25, where the edited vehicles exhibit perceptible differences relative to their reference areas. Nonetheless, the accompanying hue histograms show strong alignment between the hue distributions of the camouflaged vehicles and their reference areas. Since hue specifies the angular position on the color wheel, representing the underlying color class independent of its value or saturation, this alignment indicates that our method reliably transfers the intended color component of the style, even when other perceptual attributes may be affected by the interactions of loss terms.

Additionally, the image-level strategy is less effective for ground-view images. Because vehicles in ground-view scenes exhibit strong perspective distortion, the dilated region surrounding the vehicle

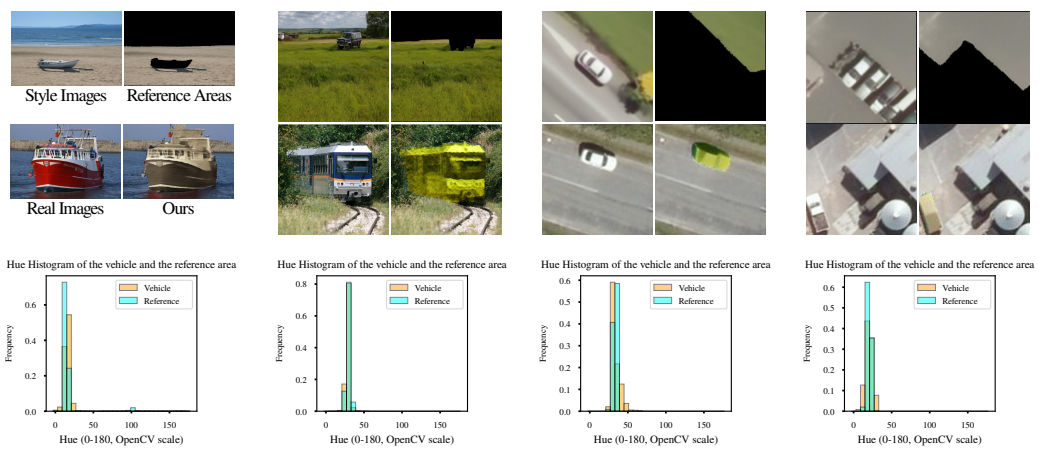


Figure 25: **Comparison of camouflaged vehicles and their corresponding style reference areas in HSV hue space.**

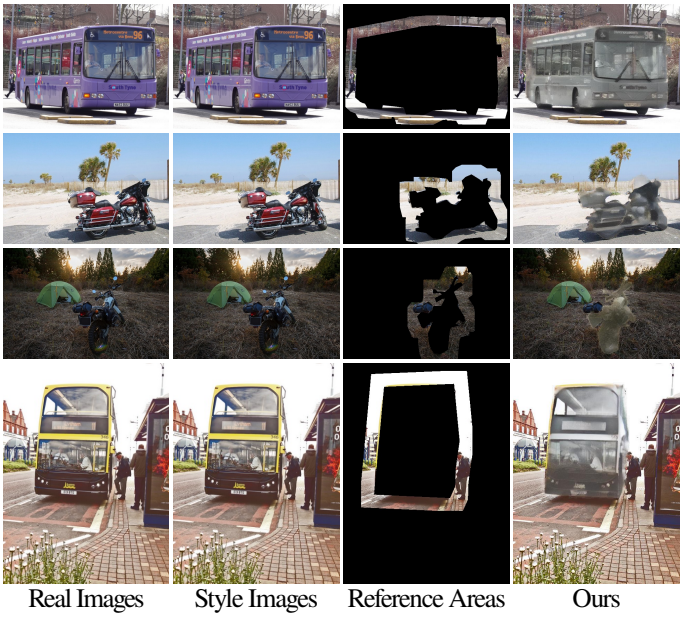


Figure 26: **Failure cases of the image-level adversarial camouflage editing strategy on the COCO dataset.**

mask frequently extends into areas that are spatially distant or semantically unrelated to the target vehicle. As a result, the extracted reference regions provide inconsistent or misleading style guidance. Failure cases produced after the No-Box Attack stage are shown in fig. 26, where the transferred vehicle's appearance does not align well with its immediate context. These observations suggest that perspective-aware or spatially adaptive reference selection could further improve performance in ground-view scenarios, which we leave for future work.

Finally, the projector-based study in section E.3 is intended to evaluate robustness to projection and camera recapture under controlled imaging conditions, rather than full physical-world deployment. It does not model the challenges of applying a coherent camouflage pattern to a real 3D vehicle across arbitrary viewpoints, illumination conditions, and surface materials. Extending our method toward physically realizable multi-view camouflage remains future work.

Task Description

In this questionnaire, you will be shown a natural image containing a vehicle and three edited versions of the image, where only the vehicle's appearance is modified.

Your task is to select the edited image whose vehicle most closely resembles **the style of the surrounding environment**.

There are **15** questions, with **2** examples shown first to help you understand the task. The questionnaire takes around 5-10 minutes to complete.

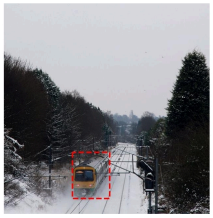
Question Format

The original image (left) and the image (middle) are identical, except for a bounding box that indicates the vehicle location. The image (right) shows the surrounding area of the vehicle whose style the edited vehicle is intended to resemble.

Image

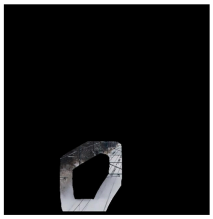


Image



bounding box annotated

Reference image



surrounding area

We would like you to select the edited vehicle that best matches the style of its surrounding area.



A



B



C

The guideline will remain available on the left sidebar once you start the questionnaire.

Start questionnaire

Figure 27: **Image-level user study guideline.**

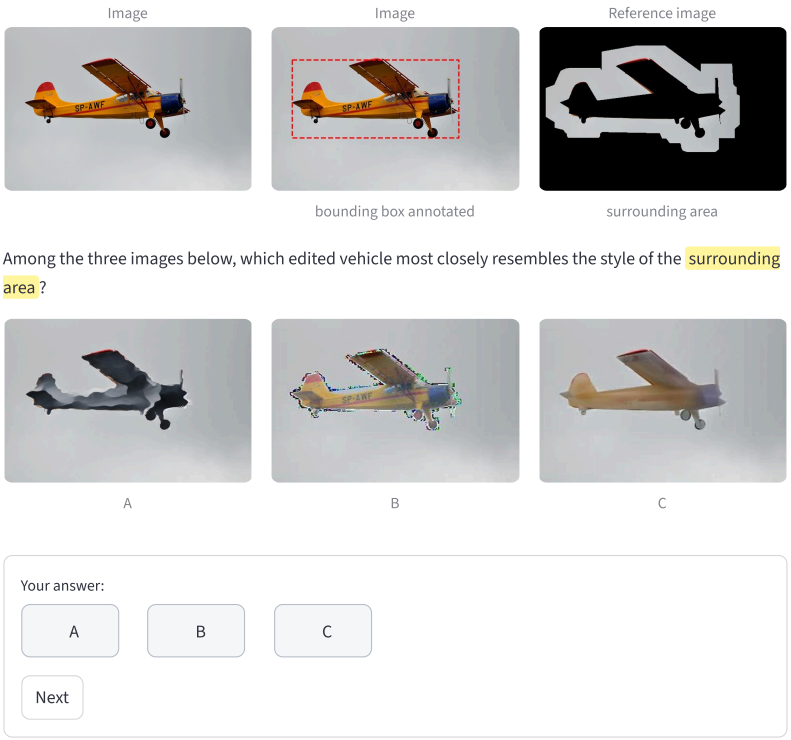


Figure 28: **Image-level example question on the COCO dataset.**

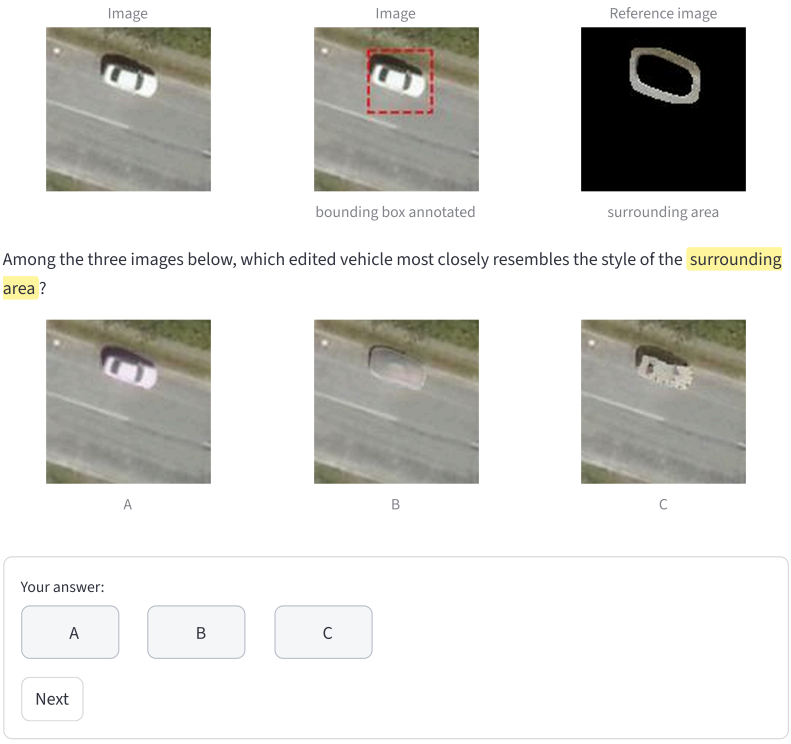


Figure 29: **Image-level example question on the LINZ dataset.**

Task Description

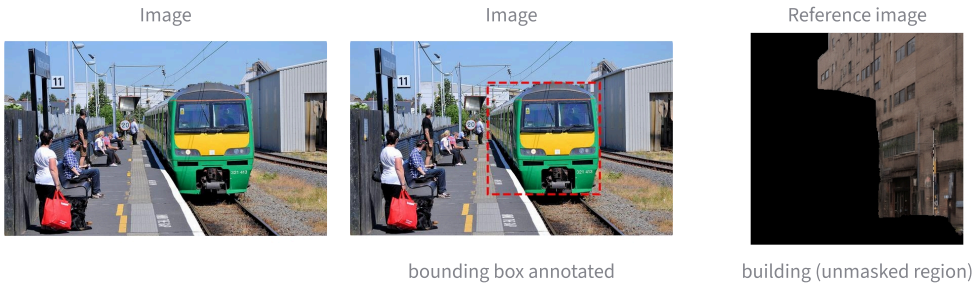
In this questionnaire, you will be shown a natural image containing a vehicle and three edited versions of the image, where only the vehicle's appearance is modified.

Your task is to select the edited image whose vehicle most closely resembles **the style of the reference object**.

There are **15** questions, with **2** examples shown first to help you understand the task. The questionnaire takes around 5-10 minutes to complete.

Question Format

The original image (left) and the image (middle) are identical, except for a bounding box that indicates the vehicle location. The reference image (right) shows the object (in this case, building) whose style the edited vehicle is intended to resemble.



We would like you to select the edited vehicle that best matches the style of the reference object.



The guideline will remain available on the left sidebar once you start the questionnaire.

Start questionnaire

Figure 30: **Scene-level user study guideline.**

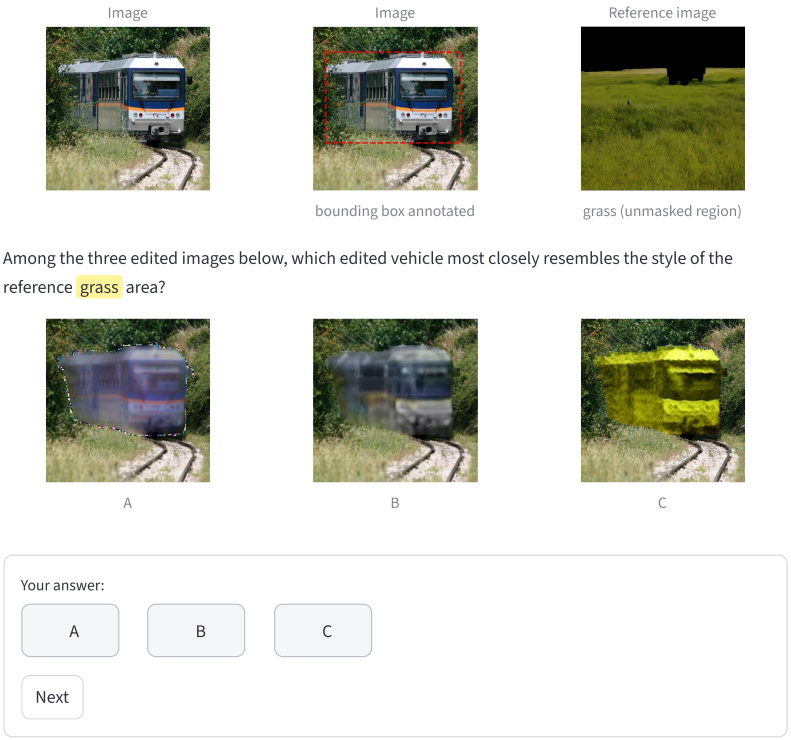


Figure 31: **Scene-level example question on the COCO dataset.**

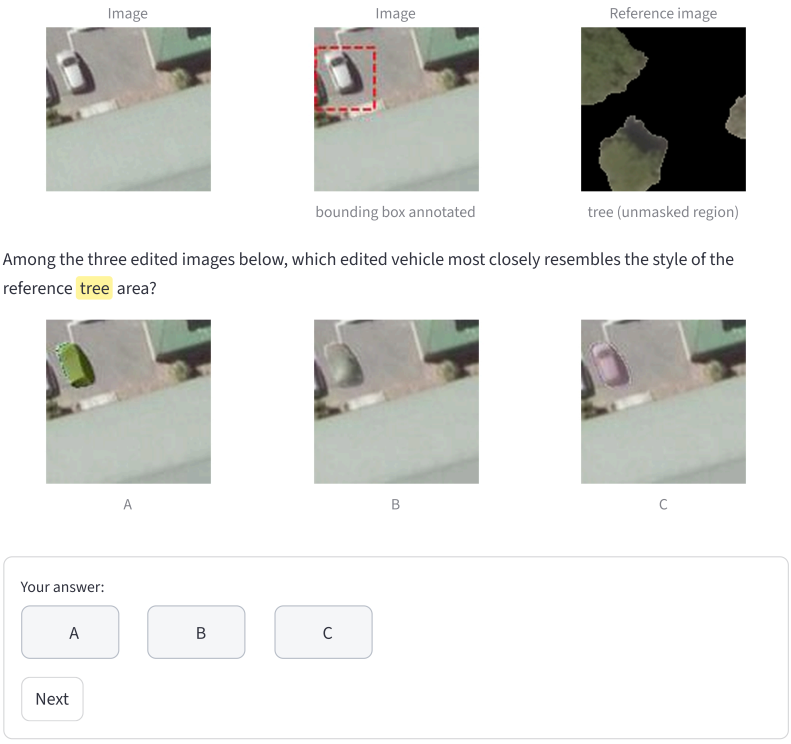


Figure 32: **Scene-level example question on the LINZ dataset.**

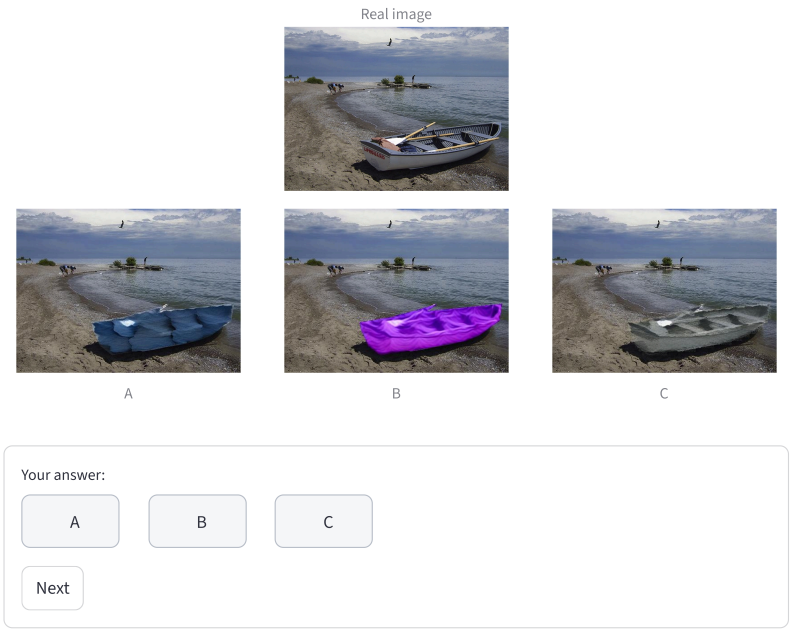


Figure 33: **Example question on the COCO dataset.**

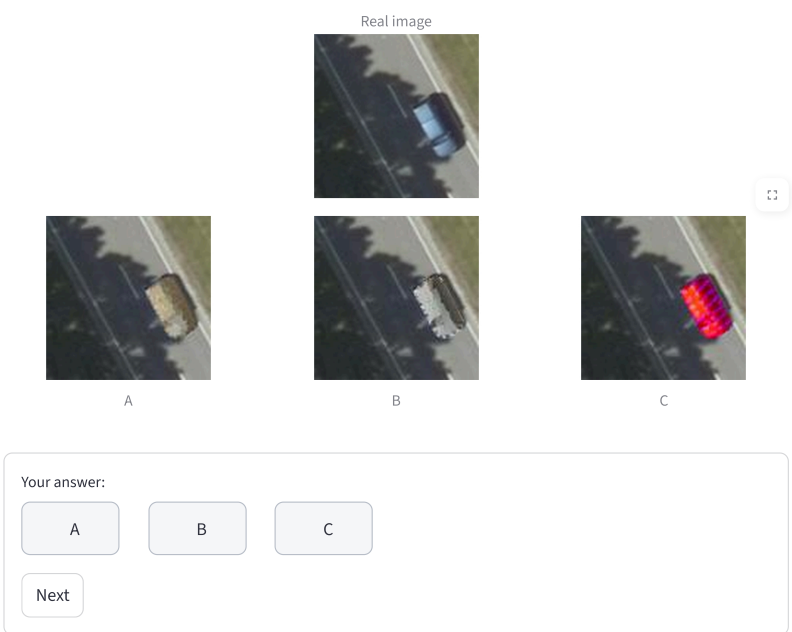


Figure 34: **Example question on the LINZ dataset.**