FACTORIZEPATCH: FACTORIZING THE PRINTER-CAMERA PIPELINE FOR ROBUST PHYSICAL ADVERSARIAL PATCHES

Anonymous authors
Paper under double-blind review

ABSTRACT

Adversarial patches optimized in the digital domain often lose effectiveness once printed and recaptured, because printing and camera imaging distort their colors, textures, and high-frequency structures. Prior work improves physical robustness through generic geometric and photometric transformations, printer-aware modeling, camera-aware simulation, or a single monolithic digital-to-physical mapping. Single-stage methods, however, cover only part of the print-capture path, while monolithic mappings obscure where degradation originates and offer little control over difficult imaging conditions. We present **FactorizePatch**, which *factorizes* the print-and-capture process into sequential yet independent differentiable stages (print appearance, scene deployment, and camera formation) instead of collapsing it into one such mapping, and places the full pipeline inside targeted universal patch optimization. Printing and imaging therefore remain explicit and independently controllable. To support attack-relevant printer modeling, we build a dedicated print-scan dataset covering adversarial and print-sensitive patterns; on held-out pairs our print proxy reconstructs printed appearance more accurately than prior proxies, reducing MAE by up to 63.5%. Within the camera stage, FactorizePatch samples multiple degradation strengths across optics/motion, sensor, ISP, and full-camera branches and adaptively reweights the branches by their current attack losses, directing optimization toward the patch's most vulnerable imaging condition while preserving coverage of compound degradations. Across CLIP and ResNet victims under simulated and real print-and-capture settings, FactorizePatch narrows the digital-to-physical gap to as little as 0.87% and achieves state-of-the-art targeted physical robustness. Code and data will be publicly available.

1 INTRODUCTION

Adversarial patches are a practical physical threat to modern visual recognition: a fixed, image-agnostic pattern can force an attacker-specified prediction whenever it enters a model's field of view (Brown et al., 2017). Unlike imperceptible perturbations crafted per image, patches can be printed, deployed, and recaptured by a camera, exposing real-world systems such as autonomous perception, visual search, and open-vocabulary recognition (Goodfellow et al., 2015; Kurakin et al., 2017; Sharif et al., 2016; Komkov & Petiushko, 2021; Wei et al., 2024a). As vision-language models such as CLIP are increasingly used as general-purpose encoders and zero-shot classifiers (Radford et al., 2021), this threat now extends to shared image-text representation

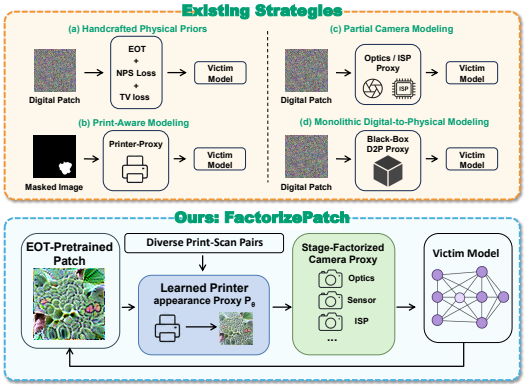


Figure 1: Existing attacks test patterns only *after* physical capture; FactorizePatch instead backpropagates through a differentiable print-camera simulation.

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.

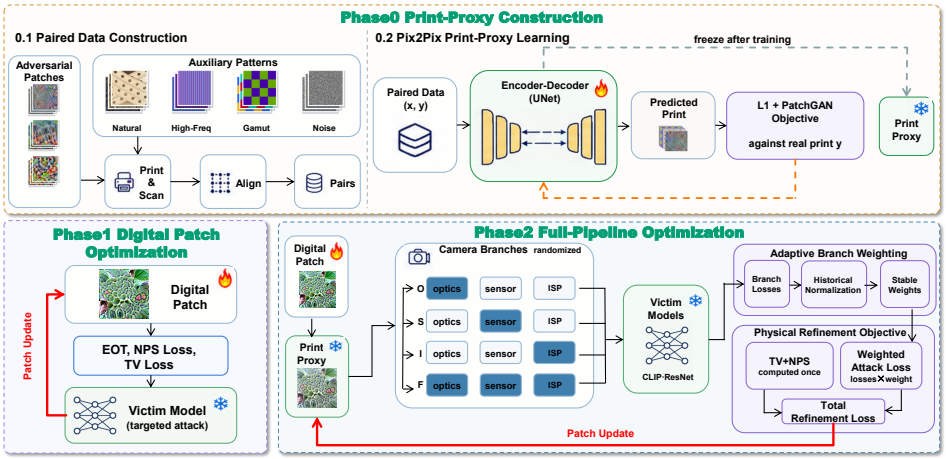


Figure 2: Overview of FactorizePatch. Phase 0 calibrates a frozen encoder-decoder print proxy from paired digital and print-scan patterns. Phase I obtains a digitally robust patch by targeted optimization with EOT, NPS, and TV regularization. Phase II passes the patch through the frozen proxy and four camera branches covering optics, sensor, ISP, and the full imaging pipeline, whose attack losses are aggregated by EMA-standardized adaptive weights. Only the digital patch receives gradients.

spaces (Zhou et al., 2023). Understanding when and why such patches survive physical realization is therefore directly relevant to the safety of deployed vision systems.

The difficulty is that patches which excel digitally often collapse after printing and capture. Existing defenses against this collapse fall into a few families. Expectation over Transformation (EOT) covers common geometric and photometric variation (Athalye et al., 2018); printability-aware attacks discourage colors or patterns that reproduce poorly (Eykholt et al., 2018; Thys et al., 2019); print-scan methods learn printing-induced color and texture distortion and fold it into optimization (Purnekar et al., 2025; Mitkovski et al., 2020); camera-aware attacks address capture-dependent effects (Wei et al., 2024b; Phan et al., 2021); and unified digital-to-physical (D2P) mappings approximate the final printed-and-captured appearance with a single learned transform (Jan et al., 2019).

Physically, however, a digital patch degrades in a *sequential two-stage* process. It is first *printed*, which introduces gamut compression, color shift, ink spreading, and attenuation of high-frequency texture. It is then *imaged*, where scene geometry, camera optics, sensor noise, and image signal processing (ISP) alter it further. Printer-only or camera-only modeling covers only part of this path, while a monolithic D2P mapping collapses both stages into one stage-opaque transform, making it impossible to tell whether attack loss originates from printing, imaging, or their composition. Compounding this, existing print-modeling datasets are built from natural or face images and generic color charts (Jan et al., 2019; Mitkovski et al., 2020; Purnekar et al., 2025; Tominaga, 1993), and thus poorly cover the high-frequency noise, non-natural textures, and gamut-critical structure that characterize physical adversarial patches.

We address these limitations with **FactorizePatch**, whose name states its central design choice. Prior D2P methods approximate the entire print-and-capture process with a single monolithic, stage-opaque transform; FactorizePatch instead *factorizes* this forward model into sequential yet independent differentiable stages and optimizes the patch through their composition. To the best of our knowledge, it is the first framework to model the printer and camera as *separable* physical stages within one differentiable optimization path, following the true deployment order *Digital Patch* $\rightarrow$ *Printer* $\rightarrow$ *Scene* $\rightarrow$ *Camera* $\rightarrow$ *Victim* (Figure 1). This factorization is the core technical differentiator of our approach. Because every stage is explicit and independently controllable, attack degradation can be attributed to printing, to imaging, or to their composition, an analysis that a monolithic, stage-opaque D2P mapping structurally cannot support. To make the printer stage attack-relevant, we construct a dedicated print-scan dataset of adversarial patterns, high-frequency noise, statistical and natural textures, color-response charts, and critical-gamut samples. Leveraging the same factorization, the framework supports printer-only, camera-only, and joint configurations, enabling a controlled study of how each physical stage contributes to attack degradation.

A complete pipeline alone, however, does not guarantee balanced robustness across heterogeneous camera degradations. If all degradation parameters are drawn from one mixed distribution, abundant easy conditions dominate the average attack loss and dilute the hard cases, and large loss swings across degradation types produce oscillatory, unstable training. FactorizePatch therefore incorporates *vulnerability-aware degradation balancing*. This controller organizes camera degradations into optics/motion, sensor, ISP, and full-camera branches, samples multiple severities within each branch across training, and adaptively reweights the branches according to their current attack losses, steering optimization toward the condition under which the patch is presently weakest. By structuring and dynamically balancing heterogeneous degradations, it keeps hard conditions from being overwhelmed by easy samples and strengthens robustness under both isolated and compound physical degradations, most visibly in the hardest imaging regimes.

We evaluate FactorizePatch under digital, simulated-physical, and real print-and-capture settings across multiple victim models, capture devices, and illumination conditions. The results show that FactorizePatch narrows the digital-to-physical gap and achieves state-of-the-art targeted physical attack performance. Our contributions are threefold:

- We propose the first framework to *factorize* the digital-to-physical forward model into sequential yet independent print, scene, and camera stages within a single differentiable optimization path, in explicit contrast to prior monolithic, stage-opaque D2P mappings, supported by a purpose-built, attack-oriented print-scan dataset.
- We introduce vulnerability-aware degradation balancing, which pairs within-branch multi-severity sampling with loss-adaptive cross-branch weighting to focus optimization on the patch's currently weakest camera condition, preventing hard degradations from being diluted and improving robustness in the worst imaging conditions.
- We demonstrate state-of-the-art attack performance, including a digital-to-physical gap as small as 0.87%, across digital, simulated, and real-world settings spanning multiple models, devices, and illumination levels.

2 Related Work

2.0.1 Patch-based Physical Adversarial Attacks

Brown et al. (2017) introduced universal targeted adversarial patches that remain effective under varying image conditions, including physical printing and camera capture. Expectation over Transformation (Athalye et al., 2018) further established a general optimization framework over transformation distributions and has become a standard strategy for physical attacks. Subsequent works extended patch attacks to more challenging physical scenarios, including non-rigid clothing deformation (Xu et al., 2020; Wu et al., 2020), 3D vehicle camouflage (Suryanto et al., 2022; 2023; Wang et al., 2021), and realistic traffic-sign benchmarks (Hingun et al., 2023).

Physical attacks on vision-language models remain relatively underexplored. AdvCLIP (Zhou et al., 2023) shows that universal patches can disrupt the shared representation space of multimodal contrastive models, but its evaluation is mainly conducted in the digital domain. Although recent studies have investigated transformation-aware physical attacks (Wei et al., 2023), most methods still rely on generic transformations or task-specific deformation models, without explicitly modeling the sequential degradation introduced by printing and camera imaging.

2.0.2 Differentiable Physical Modeling

Learned physical proxies provide an alternative to manually modeling real-world transformations (Isola et al., 2017). Jan et al. (2019) use an image-to-image translation network to approximate the mapping from digital images to printed and captured observations, while print-scan simulators further learn physical transformations from paired data (Mitkovski et al., 2020; Purnekar et al., 2025). However, these unified mappings treat printing and imaging as an opaque transformation and are not specifically optimized for the adversarial, high-frequency, and gamut-sensitive patterns of physical patches.

Recent camera-aware studies show that imaging pipelines can significantly affect adversarial signals. Phan et al. (2021) analyze camera optics and ISP effects, while Wei et al. (2024b) improve cross-

camera robustness through differentiable camera modeling. Nevertheless, existing methods do not jointly model printing and complete camera formation as a physically ordered optimization pipeline.

FactorizePatch differs from prior methods by explicitly separating printing and camera imaging as sequential, independently controllable stages within a single differentiable optimization pipeline.

3 FACTORIZEPATCH

FactorizePatch generates a targeted universal adversarial patch that remains effective after printing and camera capture. As illustrated in Figure 2, we model the physical process in its actual order: the digital patch first undergoes printing-induced appearance changes, is then transformed and deployed in a scene, and is finally processed by camera optics, sensor response, and image signal processing. On top of this full physical path, the degradation-balancing controller exposes the patch to different degradation strengths and adaptively emphasizes its currently weakest imaging condition.

3.1 THREAT MODEL

We assume a white-box attacker who controls the digital design of a printed universal patch and seeks a fixed target class t. The finalized patch is physically printed and subsequently imaged by unknown downstream cameras, viewpoints, and illumination. The victim model F is frozen at attack time. Because the attacker does not control the deployment camera, the patch must be robust to a distribution of imaging conditions rather than to any single device.

Let $x \in [0,1]^{3\times H\times W}$ denote a source image, $p \in [0,1]^{3\times h\times w}$ a trainable RGB patch. The full-pipeline captured image is formulated as

$$\hat{x} = \mathcal{C}_\phi\left(\mathcal{A}_\tau\left(x, \mathcal{P}_\theta(p)\right)\right), \tag{1}$$

where $\mathcal{P}_\theta$ is the learned print-appearance proxy, $\mathcal{A}_\tau$ transforms the printed patch and embeds it into the scene, and $\mathcal{C}_\phi$ represents complete camera formation. The physical parameters τ and camera parameters ϕ are varied during optimization. The print proxy, camera simulator, and victim model remain frozen, while their differentiability allows the targeted loss to update the original digital patch. The complete optimization procedure is summarized in Algorithm 1.

3.2 FACTORIZED FULL-PIPELINE SIMULATION

3.2.1 PATCH-ORIENTED PRINT PROXY.

Adversarial patches contain saturated colors, gamut-boundary values, sharp transitions, and high-frequency textures that are underrepresented in natural-image datasets. We therefore build a patch-oriented calibration corpus containing adversarial patterns, high-frequency structures, natural textures, color/grayscale charts, and gamut-boundary samples. Each digital pattern d_i is printed, scanned, and geometrically registered to obtain an aligned pair (d_i, r_i).

Let Π denote physical printing and $\mathcal{R}$ the fixed scanner observation operator. The paired target satisfies $r_i = \mathcal{R}(\Pi(d_i))$, so $\mathcal{P}_\theta$ learns the effective mapping $\mathcal{R}\circ\Pi$. A scanner performs dense planar sampling through controlled mechanical scanning, preserving physical scale and color while avoiding the perspective distortion and spatially varying illumination inherent to camera capture. Its fixed response is therefore absorbed into $\mathcal{P}_\theta$. In contrast, $\mathcal{C}_\phi$ models the variable 3D-to-2D perspective formation, optics, sensor, and ISP effects of downstream cameras. Thus, the two stages capture distinct calibration and deployment effects without duplicating camera formation. We instantiate $\mathcal{P}_\theta$ as a conditional image translation generator and optimize

$$\mathcal{L}_P = \mathcal{L}_{\text{adv}} + \lambda_{\text{rec}}\mathbb{E}_{(d_i,r_i)}\left[\|r_i - \mathcal{P}_\theta(d_i)\|_1\right]. \tag{2}$$

The adversarial term promotes realistic textures, while the reconstruction term preserves pattern and color fidelity. After calibration, the print proxy is frozen but remains differentiable with respect to the input patch. The calibrated proxy is specific to the current printing process; a different printer or print setting requires new paired calibration data and retraining only $\mathcal{P}_\theta$, while the remaining factorized pipeline is unchanged.

3.2.2 SCENE DEPLOYMENT.

The predicted printed patch $\bar{p} = \mathcal{P}_\theta(p)$ is geometrically and photometrically transformed before being embedded into the source image:

$$u = \mathcal{A}_\tau(x, \bar{p}). \tag{3}$$

The transformation parameters τ control patch scale, pose, and illumination during deployment.

3.2.3 STAGE-FACTORIZED CAMERA FORMATION.

The composed scene is subsequently processed by

$$\mathcal{C}_\phi(u) = \mathcal{I}_{\phi_I}\left(\mathcal{S}_{\phi_S}\left(\mathcal{L}_{\phi_L}(u)\right)\right), \tag{4}$$

where $\mathcal{L}_{\phi_L}$, $\mathcal{S}_{\phi_S}$, and $\mathcal{I}_{\phi_I}$ denote optical formation, sensor response, and image signal processing, respectively. The parameters ϕ_L, ϕ_S, and ϕ_I control each imaging stage. Unlike patch-only camera augmentation, $\mathcal{C}_\phi$ operates on the full scene, applying consistent optical, sensor, exposure, noise, and ISP effects to both the patch and background. No Redmi-, OPPO-, or Sony-specific measurements are used to calibrate the lens or sensor modules, and we do not retrain the CAP ISP on these devices; they are used only for our physical attack evaluation.

Together, Eqs. 3 and 4 approximate the full path from printable design to captured image while preserving the physical deployment order and stage-wise controllability.

3.3 VULNERABILITY-AWARE DEGRADATION BALANCING

An unstructured joint degradation distribution may allow easy conditions to dominate optimization. We therefore balance complementary camera-conditioned branches under the same scene/EOT realization: $\mathcal{B} = \{\mathrm{O}, \mathrm{S}, \mathrm{I}, \mathrm{F}\}$, where O, S, I, and F represent optics/motion-focused, sensor-focused, ISP-focused, and fully randomized conditions, respectively. Every branch retains the complete optics–sensor–ISP pipeline. Under a fixed lens, O varies rendering depth and motion blur, S and I adjust sensor and ISP configurations, and F combines all variations. Other configurations remain nominal, while stochastic sensor formation is active in all branches.

3.3.1 WITHIN-BRANCH SEVERITY RANDOMIZATION.

Training proceeds in blocks, where $k \in \{1, \ldots, Q\}$ indexes blocks of K batches. At the start of block k, each branch independently samples one degradation configuration $\xi_b^{(k)} \sim \mathcal{Q}_b$, choosing light, medium, or heavy severity with probabilities $0.3/0.5/0.2$ and then a continuous strength. The configuration is held fixed for the $K = 20$ batches of the block, while source images and shared EOT vary. For batch n,

$$\hat{x}_{b,n}^{(k)} = \mathcal{D}_b\left(x_n, \mathcal{P}_\theta(p); \xi_b^{(k)}\right), \quad \ell_{b,n}^{(k)} = \mathcal{L}_{\mathrm{atk}}\Big(F(\hat{x}_{b,n}^{(k)}), t\Big). \tag{5}$$

The controller aggregates these into the block mean

$$\bar{\ell}_b^{(k)} = \frac{1}{K}\sum_{n=1}^{K} \ell_{b,n}^{(k)}, \tag{6}$$

which exposes the patch to varied degradation strengths across blocks while providing a stable per-branch difficulty estimate within a block.

3.3.2 CROSS-BRANCH VULNERABILITY WEIGHTING.

We compare each block mean with that branch's historical mean μ_b and variance v_b:

$$\begin{aligned} r_b^{(k)} &= \mathrm{sg}\left(\frac{\bar{\ell}_b^{(k)} - \mu_b^{(k-1)}}{\sqrt{\max(v_b^{(k-1)}, \epsilon)}}\right), \\ \bar{w}_b^{(k+1)} &= \frac{1-\alpha}{|\mathcal{B}|} + \alpha\frac{\exp(r_b^{(k)}/T)}{\sum_{j\in\mathcal{B}}\exp(r_j^{(k)}/T)}, \\ \boldsymbol{w}^{(k+1)} &= \Pi_{\Delta_{[w_{\min}, w_{\max}]}}\left(\bar{\boldsymbol{w}}^{(k+1)}\right). \end{aligned} \tag{7}$$

Algorithm 1 Full-Pipeline Optimization of FactorizePatch

Require: Source distribution $\mathcal{D}$, target t, frozen victim F, print pairs $\mathcal{D}_P$, camera $\mathcal{C}$, branches $\mathcal{B}$, digital budget K_0, block size K, and block budget Q

Ensure: Physically robust patch p^*

Initialize $p \in [0,1]^{3\times h\times w}$
Learn $\mathcal{P}_\theta$ from $\mathcal{D}_P$ using Eq. 2; freeze θ
for $i = 1$ to K_0 **do**
  Sample $x \sim \mathcal{D}$ and $\tau_d \sim \mathcal{T}_d$
  Compute the digital objective in Eq. 9
  Update p with Adam and project onto $[0,1]$
end for
Set $p^{(0)} \leftarrow p$; initialize uniform branch weights and EMA histories
for $k = 1$ to Q **do**
  Sample and cache $\xi_b^{(k)} \sim \mathcal{Q}_b$ for all $b \in \mathcal{B}$
  for $n = 1$ to K **do**
    Sample a source batch and shared EOT
    for each branch $b \in \mathcal{B}$ **do**
      Render $\hat{x}_{b,n}^{(k)}$ and compute $\ell_{b,n}^{(k)}$
    end for
    Update only p using Eq. 10 and project onto $[0,1]$
  end for
  Compute $\{\bar{\ell}_b^{(k)}\}$; update weights (Eq. 7) and EMA statistics
end for
return $p^* \leftarrow p$

Here, sg denotes stop-gradient and Π_Δ denotes projection onto a bounded simplex. Branches whose losses increase relative to their own history receive larger weights. The parameter α balances adaptive and uniform weighting, while $[w_{\min}, w_{\max}]$ prevents branch collapse or domination. At the end of each block, the next branch weights are computed before μ_b and v_b are updated using exponential moving averages with decay β.

Within block k, the fixed-weight attack objective is

$$\mathcal{L}_{\text{bal}}^{(k)} = \sum_{b\in\mathcal{B}} w_b^{(k)} \ell_{b,n}^{(k)}. \tag{8}$$

The losses and weights provide a diagnostic view of stage-conditioned attack difficulty and prevent easy conditions from diluting harder ones.

3.4 Progressive Patch Optimization

3.4.1 Digital Patch Construction.

Directly optimizing a random patch through the complete physical pipeline requires it to acquire the target semantics and adapt to compound degradation simultaneously. We therefore first construct a stable digital initialization:

$$p^{(0)} = \arg\min_{p\in[0,1]} \quad \mathbb{E}_{x,\tau_d}\left[\mathcal{L}_{\text{atk}}\left(F(\mathcal{A}_{\tau_d}(x,p)),t\right)\right] + \lambda_{\text{tv}}\mathcal{L}_{\text{TV}}(p) + \lambda_{\text{nps}}\mathcal{L}_{\text{NPS}}(p). \tag{9}$$

Here, τ_d is sampled using EOT with a weak-to-strong schedule.

3.4.2 Full-pipeline Physical Refinement.

Starting from $p^{(0)}$, we refine the patch with severity-randomized, adaptively weighted branches:

$$p^* = \arg\min_{p\in[0,1]} \mathcal{L}_{\text{bal}} + \lambda_{\text{tv}}\mathcal{L}_{\text{TV}}(p) + \lambda_{\text{nps}}\mathcal{L}_{\text{NPS}}(\mathcal{P}_\theta(p))\,. \tag{10}$$

TV regularizes the digital patch, whereas NPS is evaluated on its predicted printed appearance.

During refinement, gradients traverse the frozen victim, full-scene camera, scene composition, and print proxy, but only the digital patch is updated. At deployment, only the final p^* is printed; all surrogate modules are training-only.

4 Experiments

4.1 Experimental Setup

4.1.1 Datasets.

We use the Mini-ImageNet (Vinyals et al., 2016) with 100 classes, 48,000 training images, and 12,000 validation images. We further collect an adversarial patch-oriented print-scan dataset containing 2,000 aligned digital-physical pairs. The dataset covers high-frequency textures, saturated

Victim	Method	Digital Evaluation		Cross-Device Physical Evaluation		
		T-ASR ↑	Margin ↑	Avg. ↑	D2P Gap ↓	Avg. Margin ↑
CLIP	Digital-only	**100.00**	**0.395827**	0.00	100.00	-0.0669
	EOT-only	90.75	0.055944	24.83	65.92	-0.0221
	EOT + Printability	97.81	0.078837	56.00	41.81	0.0072
	P&S	97.99	0.077129	41.54	56.45	0.0190
	D2P Adaptation	76.35	0.029608	63.00	13.35	0.0107
	FactorizePatch	90.71	0.053521	**79.06**	**11.65**	**0.0318**
ResNet	Digital-only	99.93	0.988694	0.00	99.93	-6.2132
	EOT-only	99.91	0.987852	20.67	79.24	-3.1453
	EOT + Printability	**99.95**	0.989177	58.44	41.51	3.7632
	P&S	**99.95**	0.989171	50.57	49.38	5.0443
	D2P Adaptation	95.64	0.909582	71.33	24.31	1.9413
	FactorizePatch	99.93	0.988602	**99.06**	**0.87**	**11.8059**

Table 1: Digital and cross-device physical targeted attack performance. Physical results are averaged equally over Redmi, OPPO, and Sony. T-ASR, Target Top-1, and D2P gap are percentages; margins are comparable only within each victim.

Victim	Method	Redmi	OPPO	Sony	Avg. ↑	Worst ↑	Avg. Margin ↑
CLIP	Pretrain	83.17	42.17	42.67	56.00	42.17	0.0072
	No-P/C	75.83	51.67	58.00	61.83	51.67	0.0142
	Printer-only	77.33	50.50	45.83	57.89	45.83	0.0146
	Camera-only	86.50	46.67	65.67	66.28	46.67	0.0244
	Printer+Camera	**87.00**	**66.00**	**84.17**	**79.06**	**66.00**	**0.0318**
ResNet	Pretrain	66.33	35.00	74.00	58.44	35.00	3.7632
	No-P/C	64.50	59.83	65.17	63.17	59.83	3.4685
	Printer-only	67.67	79.33	91.83	79.61	67.67	6.2257
	Camera-only	85.17	78.83	100.00	88.00	78.83	7.6383
	Printer+Camera	**98.50**	**98.67**	**100.00**	**99.06**	**98.50**	**11.8059**

Table 2: Component analysis under cross-device physical settings. Each device–source–target condition uses 600 frames; device results average the 50 settings equally, and Average/Worst are computed across devices.

and gamut-boundary colors, and sharp transitions, which are prone to detail loss, gamut compression, and ink diffusion during printing. We split the pairs into 1,600/200/200 for training, validation, and testing, with aligned 128×128 crops.

4.1.2 Evaluation Metrics.

Digital performance is evaluated by targeted attack success rate (T-ASR) and the mean margin between target and true scores on initially correct non-target samples. Physical performance is evaluated by frame-level Target Top-1, cross-device average, and worst-device performance. The D2P gap, defined as digital T-ASR minus physical Target Top-1, measures attack degradation during physical transfer.

4.1.3 Implementation Details.

We evaluate OpenAI CLIP ViT-L/14 and a Mini-ImageNet ResNet-50 (Radford et al., 2021; He et al., 2016) over 10 source classes and 5 disjoint target classes. Each target uses a separately optimized universal patch and is evaluated across all sources, yielding 50 equally weighted source–target settings. Baselines include Digital-only, EOT-only, EOT+Printability, Print-and-Scan (P&S), and Digital-to-Physical (D2P), using the same patch resolution and victim-specific evaluation protocol.

We optimize 128×128 patches with Adam (Kingma & Ba, 2015) for 10 refinement epochs using batch size 16 and learning rates of 10^{-4} for CLIP and 10^{-3} for ResNet-50, with TV/NPS weights

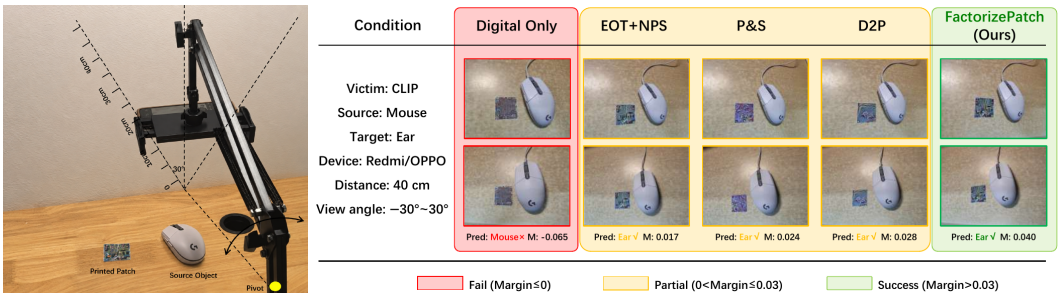


Figure 3: Qualitative physical comparison on CLIP. All patches target *ear* from the source class *mouse*, captured on a Redmi device at about 40 cm with viewing angles near $\pm 30^\circ$. Pred. denotes the top-1 prediction and $M = s(\text{target}) - s(\text{true})$.

Method	MAE ↓	PSNR ↑	SSIM ↑	HF-MAE ↓
P&S	0.10540	16.63 dB	0.4370	0.05190
D2P adaptation	0.18232	13.38 dB	0.5269	0.04002
FactorizePatch	**0.06662**	**18.88 dB**	**0.5538**	**0.03858**

Table 3: Print-appearance reconstruction on 200 held-out print-scan pairs. All methods map aligned digital patches to registered real scans. HF-MAE is the Laplacian-domain MAE over the 31 high-frequency pairs.

of 0.05/0.1. The print proxy is Pix2Pix (Isola et al., 2017), trained for 100 epochs and frozen. Camera formation uses the fixed DeepLens `GeoLens` configuration `rf24mm_f1.8.json` (Yang et al., 2024), our manually parameterized sensor simulator, and the frozen CAP ISP (Wei et al., 2024b), trained on 2,270 COCO-derived pairs generated with fast-openISP. The controller samples light/medium/heavy degradations with probabilities $0.3/0.5/0.2$ and updates configurations and weights every 20 batches.

Each source class is represented by a corresponding physical object, with a $5\,\text{cm} \times 5\,\text{cm}$ patch placed beside it on a tabletop against the wall background. The patch and object remain fixed during each capture sweep, with their midpoint as the scene center. Under the frontal view, the patch occupies approximately 4%, 9%, and 18% of the square model input at the far, middle, and near distances, respectively. For each device and source–target setting, we sample 200 frames at each of three positions over a $\pm 30^\circ$ sweep, yielding 600 frames that form one condition-level result; the 50 settings and three devices are then equally weighted.

4.2 Main Results

Table 1 shows that FactorizePatch achieves the strongest cross-device physical performance for both victims while retaining high digital T-ASR. It reaches 79.06% Target Top-1 on CLIP and 99.06% on ResNet-50, improving over D2P Adaptation by 16.06 and 27.73 percentage points, respectively. On ResNet-50, the D2P gap is reduced to only 0.87%. Representative captures are shown in Figure 3.

4.3 Full-Pipeline Component Analysis

4.3.1 Complementarity of Print and Camera Modeling.

Table 2 confirms that printer and camera modeling are complementary. The joint Printer+Camera configuration consistently outperforms No-P/C and either component alone, improving average Target Top-1 by 17.23 points on CLIP and 35.89 points on ResNet-50 over No-P/C. The gains are also preserved on the worst-performing device.

4.3.2 Cross-device Consistency.

FactorizePatch improves physical performance across all three capture devices rather than benefiting from a single favorable camera. In particular, its worst-device Target Top-1 reaches 66.00% on CLIP and 98.50% on ResNet-50, demonstrating consistent cross-device robustness. Notably, these gains

Victim	Method	Strong ↑	Medium ↑	Weak ↑	Avg. ↑	Worst ↑
CLIP	P+C (single path)	37.92	31.42	38.83	36.06	31.42
	Uniform	26.50	**85.08**	82.75	64.78	26.50
	Adaptive (ours)	**53.83**	75.58	**94.08**	**74.50**	**53.83**
ResNet-50	P+C (single path)	**57.00**	**72.08**	72.33	67.14	**57.00**
	Uniform	25.92	58.92	77.00	53.94	25.92
	Adaptive (ours)	50.58	67.92	**87.58**	**68.69**	50.58

Table 4: Ablation study of the degradation-balancing controller under three illumination settings. Values denote frame-level Target Top-1 rates (%), averaged over OPPO and Redmi. P+C uses a single physical path, Uniform uses equal O/S/I/F weights, and Adaptive balances branch losses.

are obtained without device-specific calibration of the lens, sensor, or ISP for the Redmi, OPPO, or Sony devices used in evaluation. This suggests that the learned patch is not overfitted to a particular capture pipeline, but instead retains its attack effectiveness under heterogeneous downstream imaging conditions. The small gap between average and worst-device performance, especially for ResNet-50, further indicates stable transfer across cameras.

4.3.3 Print-Proxy Calibration.

Table 3 shows that our attack-oriented proxy achieves the best reconstruction quality across all four metrics, reducing MAE by 36.8% over P&S and 63.5% over D2P adaptation. It also achieves the lowest HF-MAE, indicating better preservation of fine adversarial structures. We attribute this improvement to calibration data covering adversarial, gamut-boundary, and high-frequency patterns (Figure 4).

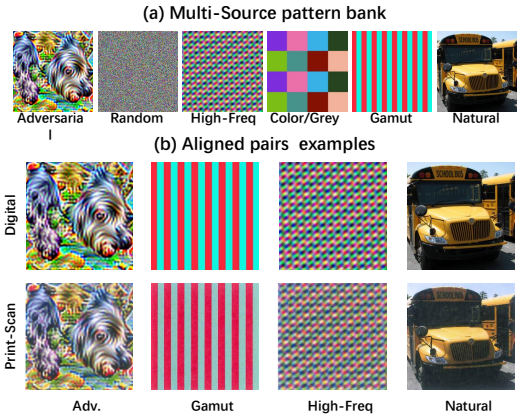


Figure 4: Representative digital/print-scan pairs covering adversarial patches, gamut-boundary patterns, and high-frequency structures.

4.4 Degradation-Balancing Ablation

We compare adaptive degradation balancing with a single-path print-camera pipeline and a multi-branch variant using fixed uniform weights. As shown in Table 4, adaptive balancing achieves the highest average Target Top-1 rate on both victim models. Compared with Uniform, it improves the three-illumination average by 9.72% on CLIP and 14.75% on ResNet-50, while improving the worst-illumination performance by 27.33% and 24.66%, respectively. These results show that separating camera degradations into O/S/I/F branches alone is insufficient; dynamically emphasizing vulnerable branches improves robustness under difficult imaging conditions.

5 Conclusion

This paper presents FactorizePatch, a factorized differentiable framework for targeted physical adversarial patch optimization. By modeling the complete transfer process as print appearance, scene deployment, optics, sensor, and ISP stages, FactorizePatch preserves the real print-and-capture order while enabling stage-wise control and diagnosis. An attack-oriented print-scan dataset calibrates print-sensitive colors, edges, and high-frequency patterns, while vulnerability-aware balancing adaptively weights the imaging branches according to attack difficulty. Experiments on CLIP and ResNet-50 under diverse devices and physical conditions demonstrate strong targeted transfer, reducing the digital-to-physical gap to 0.87%. Adaptive balancing improves the three-illumination average Target Top-1 by 9.72% and 14.75% over uniform weighting on CLIP and ResNet-50. These results show that factorized full-pipeline modeling enables robust physical adversarial patch optimization.

AI USE STATEMENT

OpenAI ChatGPT was used during the preparation of this manuscript for language polishing, improving clarity and organization, and limited assistance with LaTeX formatting. The tool was not used to generate experimental data, experimental results, figures, or scientific conclusions. All technical content, methodological design, experimental procedures, quantitative results, and interpretations were developed and verified by the authors. The authors reviewed and revised all AI-assisted text.

ETHICS STATEMENT

This work investigates the robustness of visual recognition systems against targeted physical adversarial patches by modeling the physical transformations introduced by printing and camera imaging. All experiments were conducted in controlled research environments, including digital evaluation and controlled print-and-capture experiments across multiple vision models, imaging devices, and illumination conditions. The study aims to improve the understanding of how physical imaging processes affect adversarial robustness and to support the security evaluation and development of more robust camera-based AI systems.

REFERENCES

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 284–293. PMLR, 2018.

Gary Bradski. The opencv library. *Dr. Dobb's Journal of Software Tools*, 2000.

Tom B. Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch, 2017. URL https://arxiv.org/abs/1712.09665.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3606–3613, June 2014. doi: 10.1109/CVPR.2014.461.

Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1625–1634, 2018.

Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 (6):381–395, 1981. doi: 10.1145/358669.358692.

Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco J. Madrid-Cuevas, and Manuel J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014. doi: 10.1016/j.patcog.2014.01.005.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

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*, pp. 770–778, 2016.

Nabeel Hingun, Chawin Sitawarin, Jerry Li, and David Wagner. REAP: A large-scale realistic adversarial patch benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4640–4651, 2023.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1125–1134, 2017.

Steve T. K. Jan, Joseph Messou, Yen-Chen Lin, Jia-Bin Huang, and Gang Wang. Connecting the digital and physical world: Improving the robustness of adversarial attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 962–969, 2019. doi: 10.1609/aaai.v33i01.3301962.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Stepan Komkov and Aleksandr Petiushko. AdvHat: Real-world adversarial attack on ArcFace face ID system. In *2020 25th International Conference on Pattern Recognition*, pp. 819–826, 2021. doi: 10.1109/ICPR48806.2021.9412236.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *International Conference on Learning Representations Workshop*, 2017.

Aleksandar Mitkovski, Johannes Merkle, Christian Rathgeb, Benjamin Tams, Kevin Bernardo, Nathania E. Haryanto, and Christoph Busch. Simulation of print–scan transformations for face images based on conditional adversarial networks. In *2020 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 77–86, 2020.

Buu Phan, Fahim Mannan, and Felix Heide. Adversarial imaging pipelines. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16051–16061, 2021.

Nischay Purnekar, Benedetta Tondi, Jana Dittmann, and Mauro Barni. Print and scan simulation for adversarial attacks on printed images. *APSIPA Transactions on Signal and Information Processing*, 14(3):1–32, 2025. doi: 10.1561/116.20250019.

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

Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1528–1540, 2016. doi: 10.1145/2976749.2978392.

Naufal Suryanto, Yongsu Kim, Hyoeun Kang, Harashta Tatimma Larasati, Youngyeo Yun, Thi-Thu-Huong Le, Hunmin Yang, Se-Yoon Oh, and Howon Kim. DTA: Physical camouflage attacks using differentiable transformation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15305–15314, 2022.

Naufal Suryanto, Yongsu Kim, Harashta Tatimma Larasati, Hyoeun Kang, Thi-Thu-Huong Le, Yoonyoung Hong, Hunmin Yang, Se-Yoon Oh, and Howon Kim. ACTIVE: Towards highly transferable 3d physical camouflage for universal and robust vehicle evasion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4305–4314, 2023.

Simen Thys, Wiebe Van Ranst, and Toon Goedeme. Fooling automated surveillance cameras: Adversarial patches to attack person detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

Shoji Tominaga. A neural network approach to color reproduction in color printers. In *IS&T and SID's Color Imaging Conference: Transforms & Transportability of Color*, pp. 173–177. Society for Imaging Science and Technology, 1993. URL https://library.imaging.org/admin/apis/public/api/ist/website/downloadArticle/cic/1/1/art00043.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Jiakai Wang, Aishan Liu, Zixin Yin, Shunchang Liu, Shiyu Tang, and Xianglong Liu. Dual attention suppression attack: Generate adversarial camouflage in physical world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8565–8574, 2021.

Hui Wei, Hao Tang, Xuemei Jia, Zhixiang Wang, Hanxun Yu, Zhubo Li, Shin'ichi Satoh, Luc Van Gool, and Zheng Wang. Physical adversarial attack meets computer vision: A decade survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9797–9817, 2024a. doi: 10.1109/TPAMI.2024.3430860.

Hui Wei, Zhixiang Wang, Kewei Zhang, Jiaqi Hou, Yuanwei Liu, Hao Tang, and Zheng Wang. Revisiting adversarial patches for designing camera-agnostic attacks against person detection. In *Advances in Neural Information Processing Systems*, volume 37, pp. 8047–8064, 2024b. doi: 10.52202/079017-0259.

Xingxing Wei, Yao Huang, Yitong Sun, and Jie Yu. Unified adversarial patch for cross-modal attacks in the physical world. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4445–4454, 2023.

Zuxuan Wu, Ser-Nam Lim, Larry S. Davis, and Tom Goldstein. Making an invisibility cloak: Real world adversarial attacks on object detectors. In *Computer Vision–ECCV 2020*, pp. 1–17. Springer, 2020. doi: 10.1007/978-3-030-58548-8_1.

Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Adversarial T-Shirt! evading person detectors in a physical world. In *Computer Vision–ECCV 2020*, pp. 665–681. Springer, 2020. doi: 10.1007/978-3-030-58558-7_39.

Chenglin Yang, Adam Kortylewski, Cihang Xie, Yinzhi Cao, and Alan Yuille. PatchAttack: A black-box texture-based attack with reinforcement learning. In *Computer Vision – ECCV 2020*, pp. 681–698. Springer, 2020. doi: 10.1007/978-3-030-58574-7_41.

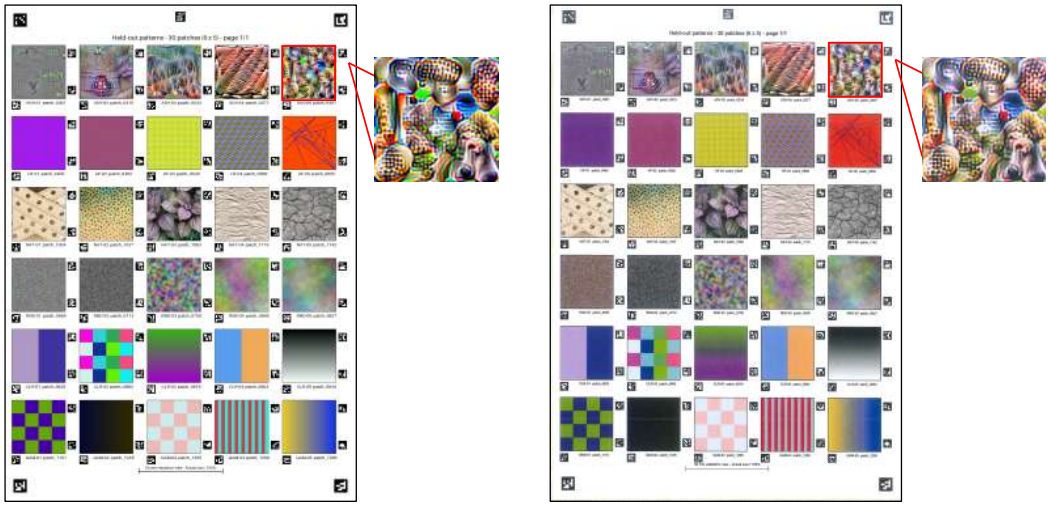
Xinge Yang, Qiang Fu, and Wolfgang Heidrich. Curriculum learning for ab initio deep learned refractive optics. *Nature Communications*, 15(1):6572, 2024.

Ziqi Zhou, Shengshan Hu, Minghui Li, Hangtao Zhang, Yechao Zhang, and Hai Jin. AdvCLIP: Downstream-agnostic adversarial examples in multimodal contrastive learning. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 6311–6320, 2023. doi: 10.1145/3581783.3612454.

APPENDIX

Overview. This appendix provides additional implementation details, experimental protocols, and supporting analyses that are not fully expanded in the main paper:

- Sec. A clarifies the fixed-scanner calibration procedure and evaluates print–scan repeatability.
- Sec. B details the capture devices, controlled acquisition geometry, angular-sweep and frame-sampling protocol, and the additional repeatability evaluation across independent prints, patch placements, and capture sessions.
- Sec. C provides additional camera-simulation details, including the stage-focused branches, degradation-severity parameterization, and parameter-sensitivity analysis of the balancing controller.
- Sec. D describes the construction, geometric registration, and quality control of the attack-oriented print–scan dataset.
- Sec. E evaluates the stability of physical refinement across different training seeds.
- Sec. F details the implementation of the comparison methods and the fairness controls used in evaluation.



(a) Digital pattern sheet (b) Registered print-scan observation

Figure 5: Construction of aligned digital and print–scan pairs using a fixed flatbed scanner. The enlarged crops show the appearance change after physical printing and digitization.

A Print–Scan Calibration

A.1 Role of the Fixed Scanner

Constructing paired supervision for the print proxy requires converting each physical printout back into a digital image, since the printed pattern cannot be directly aligned with its digital source or used as a pixel-level training target. We therefore use a flatbed scanner as a controlled and repeatable digitization interface, avoiding the viewpoint, distance, perspective, and ambient-light variations introduced by camera capture.

Following the notation in the main paper, let Π denote the fixed fabrication process and $\mathcal{R}$ the fixed scanner observation operator. For each digital pattern d_i, the registered calibration target is

$$r_i = \mathcal{R}\left(\Pi(d_i)\right). \tag{11}$$

Because the same scanner and acquisition configuration are used throughout calibration, $\mathcal{R}$ is shared across all pairs and absorbed into the effective print mapping. Accordingly, the proxy learns $\mathcal{P}_\theta \approx \mathcal{R} \circ \Pi$, rather than a scanner-independent printer response.

Figure 5 shows that printing and digitization introduce visible appearance deviations, motivating the learned effective print mapping.

The scanner is used only for offline calibration and is distinct from the unknown cameras used during deployment. The remaining concern is whether random repeated-scan or independent-print fluctuations dominate the calibration targets, which we evaluate below.

A.2 Print–Scan Repeatability

We independently print three copies of the same evaluation page, each containing 30 patterns from six categories. The three copies are placed simultaneously on the scanner bed and captured together in each acquisition. Without repositioning the copies, we repeat the whole-bed scan five consecutive times. Thus, each scan index corresponds to one joint acquisition of all three print copies under the same scanner configuration.

Let r_{ijk} denote the aligned observation of pattern i from print copy j in scan k, where $j \in \{1, 2, 3\}$ and $k \in \{1, \ldots, 5\}$. We define

$$\bar{r}_{ij} = \frac{1}{5}\sum_{k=1}^{5} r_{ijk}, \qquad \bar{r}_i = \frac{1}{3}\sum_{j=1}^{3} \bar{r}_{ij}. \tag{12}$$

Source of variation	MSE ($\times 10^{-3}$) $\downarrow$	Rel. to shift	95% CI ($\times 10^{-3}$)
Repeated-scan variability	2.75 ± 3.64	12.0%	$[1.61,\ 4.13]$
Independent-print variability	1.22 ± 1.36	5.3%	$[0.78,\ 1.75]$
Systematic digital-to-print–scan shift	22.95 ± 22.45	100.0%	$[15.38,\ 31.20]$

Table 5: Random variability versus the systematic appearance shift in the fixed print–scan calibration chain. MSE is computed over normalized RGB values after translation-only alignment and aggregated over 30 patterns. Confidence intervals are obtained from 10,000 pattern-level bootstrap resamples. "Rel. to shift" is the ratio of mean MSE values rather than an additive variance contribution.

	Target Top-1 ASR (%) ↑				Cross-session statistics	
Method	Upper-left	Lower-left	Upper-right	Lower-right	Mean ASR $\pm$ std	Mean margin $\pm$ std
EOT+NPS	56.00	57.00	71.00	55.00	59.75 ± 7.54	0.01157 ± 0.00855
D2P Adaptation	70.50	64.50	63.50	68.00	66.62 ± 3.22	0.00474 ± 0.00502
FactorizePatch	**87.50**	**77.50**	**81.00**	**76.50**	$\mathbf{80.62 \pm 4.97}$	$\mathbf{0.01716 \pm 0.00777}$

Table 6: Physical targeted attack performance across four independent print–placement sessions. Each session uses a separately printed copy of the corresponding optimized digital patch, one of four placements relative to the source object, and an independently initialized capture session. Each session contains 600 evaluated frames. The four placement columns report Target Top-1 ASR. Mean ASR and its sample standard deviation are computed over the four sessions. Margin denotes $s(\mathrm{Ear}) - s(\mathrm{Mouse})$; the margin is first averaged over the 600 frames within each session, and the reported mean and sample standard deviation are then computed over the four session-level values.

The three squared-error quantities are

$$V_{\mathrm{scan}}(i) = \frac{1}{3(5-1)} \sum_{j=1}^{3} \sum_{k=1}^{5} \mathrm{MSE}(r_{ijk}, \bar{r}_{ij}), \tag{13}$$

$$V_{\mathrm{print}}(i) = \max\left(0,\ \frac{1}{2} \sum_{j=1}^{3} \mathrm{MSE}(\bar{r}_{ij}, \bar{r}_i) - \frac{V_{\mathrm{scan}}(i)}{5}\right), \tag{14}$$

$$V_{\mathrm{shift}}(i) = \mathrm{MSE}(d_i, \bar{r}_i). \tag{15}$$

Here, V_{scan} measures within-print acquisition variability, V_{print} estimates independent-print variability after correcting for residual scan noise in the five-scan averages, and V_{shift} measures the systematic digital-to-print–scan appearance change.

As shown in Table 5, the systematic digital-to-print–scan shift is substantially larger than both repeated-scan and independent-print variability. This indicates that the dominant calibration signal arises from the fixed print–scan appearance change rather than random acquisition fluctuations.

B Physical Capture Protocol

B.1 Capture Devices

Physical evaluation uses three capture devices: a Redmi K20 Pro smartphone, an OPPO K9s smartphone (model PERM10), and a Sony NEX-7 mirrorless camera. The shorthand labels *Redmi*, *OPPO*, and *Sony* used in the main paper refer to these three devices, respectively. All devices are evaluated using the same physical patch size, scene-centered acquisition geometry, nominal distances, angular range, recording duration, and frame-sampling protocol. Each capture device is mounted using a support appropriate to its form factor while following the same marked acquisition trajectory.

B.2 Standard Acquisition Protocol

Figure 6 shows the physical setup and reference markings used for video acquisition. Within each capture session, the printed patch and source object are placed at fixed positions on the tabletop, and their midpoint is defined as the scene center. A central baseline and the nominal $-30°$ and $+30°$

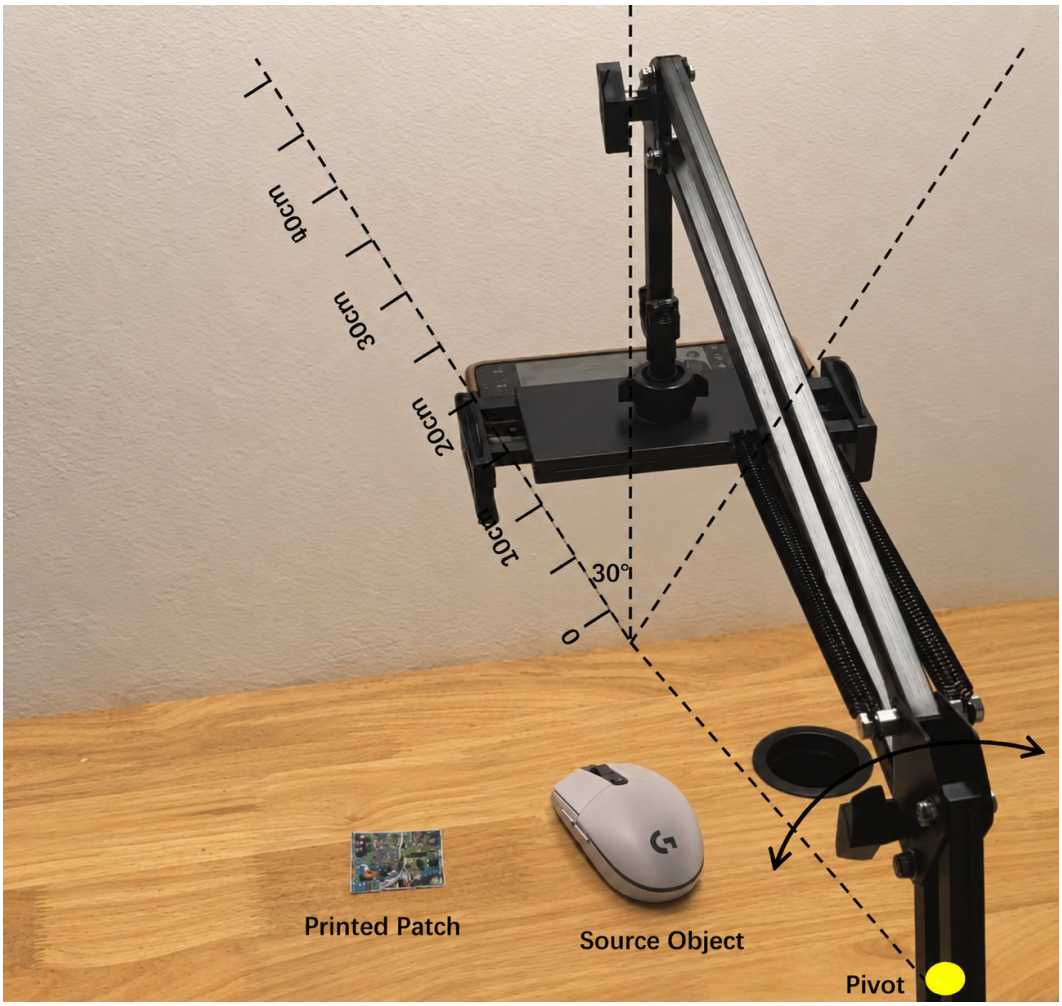


Figure 6: Physical capture setup and scene-centered angular sweep. Within each capture session, the printed patch and source object remain fixed on the tabletop. Their midpoint defines the scene center, relative to which the -30°, 0°, and $+30^\circ$ viewing directions and the nominal distances of 20, 30, and 40 cm are marked. The capture device is mounted on an articulated support, whose mechanical pivot realizes the guided camera motion.

viewing references are marked on the wall, together with the three nominal scene-centered distances of 20, 30, and 40 cm.

The capture device is mounted on an articulated support appropriate to its form factor. At each nominal distance, the support is positioned according to the corresponding distance mark, and its mechanical pivot is fixed. Before each angular sweep, the capture device is visually aligned with the wall-mounted central baseline such that its optical axis is directed toward the scene center. The support is then rotated slowly from the nominal -30° reference to the nominal $+30^\circ$ reference. During the sweep, the wall markings serve as visual guides, and the device orientation is manually adjusted to remain directed toward the scene center. This alignment procedure is repeated at every distance and for every capture session.

The wall-mounted reference markings provide repeatable visual guidance for the scene-centered acquisition geometry, whereas the articulated support provides its mechanical realization. Within each capture session, the printed patch and source object remain fixed, and only the capture-device position and viewing direction vary according to the marked acquisition trajectory.

One approximately 10 s angular sweep is recorded at each of the three nominal distances. We uniformly sample 200 frames from each sweep, yielding 600 frames per device–source–target condi-

tion. These frames form one condition-level result, after which the 50 source–target settings are equally weighted. The same protocol is used for all methods and devices.

B.3 Independent Print and Placement Repeatability

We additionally conduct an end-to-end physical repeatability experiment to evaluate whether the observed attack performance persists across independent printing, spatial deployment, and capture sessions. For each method, we start from its optimized digital patch and produce a separately printed physical copy for each session. In four separate sessions, a newly printed copy is placed at the upper-left, lower-left, upper-right, or lower-right position relative to the source object. Each placement condition therefore uses a separate printout and an independently initialized physical capture session. All three methods are evaluated under the same capture-device and source–target settings.

For each placement condition, the midpoint between the current printed patch and the source object is used as the scene center for camera alignment. Within each session, the printed patch and source object remain fixed, and the complete acquisition protocol is repeated. Specifically, one approximately 10 s angular sweep is recorded at each of the nominal distances of 20, 30, and 40 cm. We uniformly sample 200 frames from each sweep, resulting in 600 evaluated frames per session. The four sessions therefore provide 2,400 evaluated frames per method for this repeatability analysis.

Since the printed copy and patch position vary jointly across sessions, this experiment evaluates their combined end-to-end physical effect rather than isolating print and placement variability separately.

Table 6 shows that FactorizePatch achieves the highest Target Top-1 ASR under all four independent print–placement conditions. Its mean session-level ASR is 80.62%, exceeding D2P Adaptation and EOT+NPS by 14.00 and 20.87 percentage points, respectively. FactorizePatch also achieves the highest mean target margin of 0.01716.

Although attack performance varies across independently printed and spatially redeployed sessions, FactorizePatch remains consistently stronger than both comparison methods under every evaluated placement. These results indicate that the observed improvement is not restricted to a single printout, patch position, or capture session.

C Camera Simulation Details

Our differentiable camera model executes the complete optics/motion–sensor–ISPNet pipeline. To expose stage-specific vulnerabilities while preserving the interactions among camera stages, we construct four stage-focused branches: optics/motion (O), sensor formation (S), image signal processing (I), and full-camera formation (F). All four branches retain the complete camera pipeline. A stage described as *nominal* remains active at its reference configuration rather than being bypassed.

Stochastic sensor noise is retained throughout the pipeline because it represents acquisition-level randomness present in every physical capture. Disabling it in the O or I branch would produce unrealistically deterministic observations and confound branch comparisons with the presence or absence of sensor noise. Therefore, the O and I branches keep the sensor configuration at its nominal setting while resampling stochastic noise, whereas the S branch additionally randomizes the sensor configuration and noise parameters.

C.1 Stage-Focused Camera Branches

Table 7 summarizes the randomized focus and retained configurations of the four branches.

C.2 Marginal Contribution of Camera-Stage Branches

To determine which camera stage provides the most distinctive optimization signal, we individually remove the Optics, Sensor, and ISP branches and measure the resulting validation-loss change, gradient response, and compensation provided by the remaining branches.

Figure 7 shows that removing the ISP branch produces the largest increase in validation loss and induces the strongest gradient response. Despite this increased optimization effort, the remaining

Branch	Pipeline	Randomized focus	Other configurations
O	Optics/motion → Sensor → ISPNet	Rendering depth and motion blur	Fixed lens prescription; nominal sensor and ISP conditions; stochastic sensor noise retained
S	Optics/motion → Sensor → ISPNet	Sensor configuration and noise	Base rendering depth, no motion blur, and nominal ISP condition
I	Optics/motion → Sensor → ISPNet	Six-dimensional ISPNet condition	Base rendering depth and no motion blur; nominal sensor configuration with stochastic noise retained
F	Optics/motion → Sensor → ISPNet	Optics/motion, sensor, and ISP	All supported camera factors jointly randomized

Table 7: Stage-focused camera branches. All branches execute the complete camera pipeline; nominal stages remain active at their reference configurations.

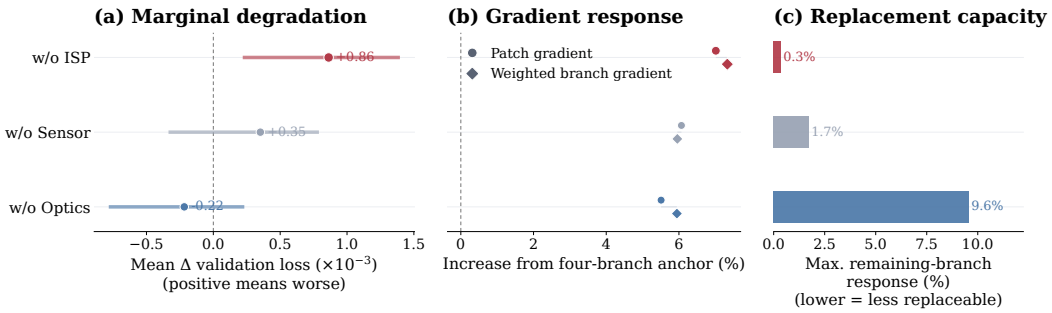


Figure 7: Marginal contribution of the camera-stage branches. (a) Validation-loss change after removing each branch. (b) Relative gradient response induced by branch removal. (c) Maximum compensation provided by the remaining branches.

branches provide only limited compensation for the missing ISP objective. By contrast, the effect of removing the Optics branch is largely absorbed by the other branches.

These results indicate that the ISP branch supplies the least replaceable optimization signal and makes the largest marginal contribution among the O/S/I branches. They also demonstrate the diagnostic value of the factorized design: branch-wise losses, gradients, and compensation reveal which imaging stage currently constitutes the main robustness bottleneck.

C.3 Severity Parameterization

Let $s \in [0, 1]$ denote the normalized degradation severity. The optics/motion stage is represented by $\mathcal{L}_d \circ \mathcal{M}_{k,\theta}$, where d is the signed rendering depth, k is the motion-blur kernel size, and θ is the blur direction. The sensor and ISP stages are denoted by $\mathcal{S}_{\phi_S}$ and $\mathcal{I}_{\mathbf{c}}$, respectively. Table 8 reports the representative settings used for fixed evaluation.

For motion blur, the direction is sampled as $\theta \sim \mathcal{U}(0, \pi)$, and the maximum supported kernel size is 9. The target sensor configuration $\tilde{\phi}_S$ is sampled from the predefined sensor parameter ranges. The sensor parameter vector contains read noise, shot noise, dark current, fixed-pattern noise, quantum efficiency, saturation, gamma, nonlinearity, analog and digital gain, black level, crosstalk, and temperature. The vector $\mathbf{c}$ is the six-dimensional normalized condition supplied to the frozen ISPNet.

During training, the normalized severity is sampled continuously within three severity buckets:

$$s \sim \begin{cases} \mathcal{U}(0, \frac{1}{3}), & \text{light}, \\ \mathcal{U}(\frac{1}{3}, \frac{2}{3}), & \text{medium}, \\ \mathcal{U}(\frac{2}{3}, 1), & \text{heavy}. \end{cases} \tag{16}$$

The light, medium, and heavy buckets are selected with probabilities 0.3, 0.5, and 0.2, respectively. For fixed evaluation, we use the corresponding bucket midpoints $\frac{1}{6}$, $\frac{1}{2}$, and $\frac{5}{6}$ reported in Table 8.

Stage	Nominal	Light	Medium	Heavy
Optics/motion	$d = -2000\,\mathrm{mm}$ $k = 1$	$s = \frac{1}{6},\ \epsilon_d = 0.033$ $k = 3$	$s = \frac{1}{2},\ \epsilon_d = 0.10$ $k = 5$	$s = \frac{5}{6},\ \epsilon_d = 0.167$ $k = 7$
Sensor	$\phi_S(0)$	$\phi_S(\frac{1}{6})$	$\phi_S(\frac{1}{2})$	$\phi_S(\frac{5}{6})$
ISPNet	$\mathbf{c} = 0.5\mathbf{1}_6$	$\mathbf{c} \in [0.417, 0.583]^6$	$\mathbf{c} \in [0.25, 0.75]^6$	$\mathbf{c} \in [0.083, 0.917]^6$

Table 8: Severity parameterization of the camera stages. Light, medium, and heavy correspond to the representative severity values $\frac{1}{6}$, $\frac{1}{2}$, and $\frac{5}{6}$ used for fixed evaluation. For motion blur, $\theta \sim \mathcal{U}(0, \pi)$.

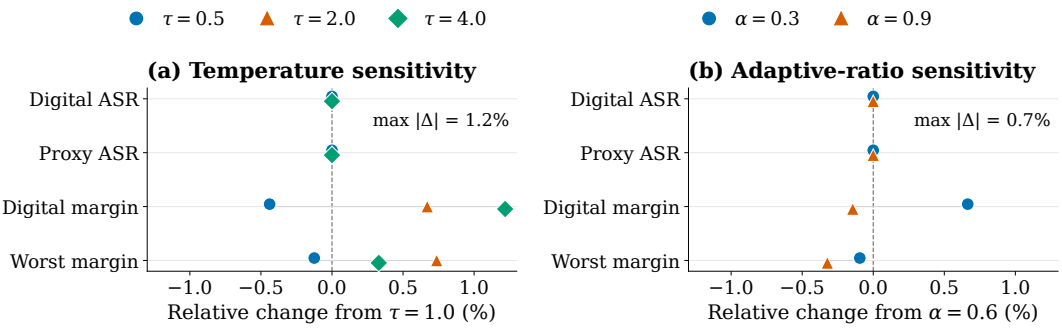


Figure 8: Parameter sensitivity of the balancing controller. Attack outcomes under different values of (a) temperature T and (b) adaptive ratio α.

C.4 Parameter Sensitivity

We further evaluate the sensitivity of the balancing controller to the temperature T, which controls the concentration of the adaptive branch weights, and the adaptive ratio α, which balances adaptive weighting against the uniform component. As shown in Figure 8, performance remains stable over the evaluated ranges of both T and α. Digital and proxy T-ASR show maximum relative variations of only 1.2% and 0.7%, respectively, while the corresponding target margins also remain broadly consistent. These results indicate that the improvements obtained by vulnerability-aware balancing are not restricted to a narrowly selected hyperparameter configuration and do not require careful parameter tuning.

D Print–Scan Dataset Construction

D.1 Dataset Composition

We construct an attack-oriented print–scan dataset containing 2,000 aligned digital–physical pairs across six pattern families. As summarized in Table 9, the dataset combines adversarial patterns, print-sensitive structures, and natural-image content.

The adversarial-pattern subset is composed primarily of patches generated in-house using multiple attack methods against different victim models. It includes patches locally optimized for CLIP and ResNet-50 Radford et al. (2021); He et al. (2016), together with ImageNet patches from AdvPatch-Dict Yang et al. (2020). The natural-texture subset includes DTD texture crops Cimpoi et al. (2014) and class-balanced center crops from the Mini-ImageNet training split Vinyals et al. (2016).

The dataset is split at the level of the original registered pairs into 1,600 training pairs, 200 validation pairs, and 200 held-out test pairs. Training-time paired cropping is performed only after this split to prevent overlap between the training, validation, and test sets.

Pattern family	Pairs	Representative content and source
Adversarial patterns	403	Primarily in-house patches generated using multiple attack methods against different victim models, including patches locally optimized for CLIP and ResNet-50, together with Adv-PatchDict ImageNet patches
High-frequency patterns	260	Procedural stripes, checkerboards, dot grids, halftones, multi-frequency patterns, and fine lines
Statistical textures	170	Procedural uniform, Gaussian, colored, low-frequency, blockwise, and multi-scale random textures
Natural textures	841	DTD texture crops and class-balanced center crops from the Mini-ImageNet training split
Color/grayscale charts	170	Procedural color blocks, grayscale ramps, gradients, neutral tones, and complementary color pairs
Gamut-boundary patterns	156	Procedural saturated colors, near-black and near-white tones, extreme color pairs, and stepped gradients
Total	**2,000**	–

Table 9: Composition of the attack-oriented print–scan dataset. The adversarial-pattern category consists primarily of in-house patches generated using multiple attack methods against different victim models. The natural-texture category includes both DTD texture crops and Mini-ImageNet natural-image crops.

D.2 Page Layout and Acquisition

The 2,000 digital patterns are arranged on 100 A4 sheets, with 20 patterns per page in a 4×5 grid. Each pattern occupies a 32×32 mm physical region. All pages are printed at actual size (100%) using the color-printing device identified as `Matt-08FB (Xerox EX-i 80 Server 1.2)`.

The printed sheets are digitized using the scanner unit of a `RICOH Aficio MP 9002 PCL 6` multifunction device. The same printing device, scanner, and acquisition configuration are used for all samples.

The page layout uses the OpenCV `DICT_5X5_1000` ArUco dictionary Bradski (2000); Garrido-Jurado et al. (2014). Each page contains a page-identity marker for recovering its logical page index, while each pattern is associated with two local markers for geometric registration and extraction. A fixed manifest stores the page index, grid position, physical pattern boundary, and expected marker coordinates for every pattern.

D.3 ArUco Registration and Patch Extraction

ArUco markers are detected using the OpenCV `ArucoDetector`, with `CORNER_REFINE_SUBPIX` enabled for subpixel corner localization. For pattern i, the eight corners provided by its two local markers are matched to their physical coordinates in the manifest to estimate a local planar mapping

$$H_i : (x_{\text{scan}}, y_{\text{scan}}) \rightarrow (x_{\text{page}}, y_{\text{page}}). \tag{17}$$

The local transformation is estimated using RANSAC Fischler & Bolles (1981), and its reprojection RMSE is recorded for quality auditing.

The theoretical 32×32 mm pattern boundary is obtained from the manifest. We apply a uniform inward offset of 0.2 mm on all four sides to avoid contamination from white margins, printed borders, and subpixel registration errors. The resulting 31.6×31.6 mm valid region is mapped back to the scanned image and rectified to a 256×256 RGB image.

The registered digital and print–scan images are stored as aligned 256×256 pairs. During Pix2Pix training Isola et al. (2017), spatially corresponding 128×128 regions are synchronously cropped from the digital input and its print–scan target. The same crop coordinates are applied to both images, preserving pixel-level correspondence while increasing the number and spatial diversity of local training observations.

D.4 Registration Quality

Table 10 summarizes the registration quality of the complete dataset. The pipeline successfully recovers the page identity for all 100 scanned sheets and extracts all 2,000 pattern pairs. The mean and maximum local reprojection RMSE values are 0.02979 mm and 0.04084 mm, respectively.

Audit item	Result	Rate
Page-identity recovery	100/100	100%
Patch extraction	2,000/2,000	100%
Mean reprojection RMSE	0.02979 mm	–
Maximum reprojection RMSE	0.04084 mm	–

Table 10: Quality audit of the ArUco-based registration and patch-extraction pipeline.

Training seed	Validation loss ↓	Target ASR (%) ↑
20260601	−0.079887	89.015
20260602	−0.079821	89.015
20260603	−0.079654	89.015
20260604	−0.079429	89.015
Mean ± sample std	-0.079698 ± 0.000204	89.015 ± 0.000
Range	0.000458	0.000

Table 11: Training-seed stability at the matched post-convergence checkpoint using CLIP ViT-L/14 as the victim model. All runs use the same initial patch, optimization configuration, and fixed validation protocol.

The same physical boundary, content inset, output resolution, interpolation method, and color representation are used for all samples. Page-level extraction overlays and dataset-level contact sheets are additionally inspected to exclude crops containing white margins, neighboring patterns, text labels, or residual ArUco markers.

E Training-Seed Stability

We evaluate the sensitivity of physical refinement to training stochasticity using CLIP ViT-L/14 as the victim model and four training seeds: 20260601, 20260602, 20260603, and 20260604. All runs use an identical initial patch, the same optimization configuration, and a fixed validation seed of 21260604. The target prompt is *a photo of ear*. This setting evaluates randomness arising from the training-data order and stochastic EOT and camera-degradation sampling.

All four runs reached a stable convergence plateau by global batch 6000. We therefore compare them at this matched post-convergence checkpoint. Table 11 reports the validation performance, while Figure 9 compares the visual consistency of the optimized patches.

All four seeds achieve the same Target ASR of 89.015%. The validation-loss sample standard deviation is 0.000204, and the range is 0.000458, corresponding to 0.575% of the absolute mean-loss magnitude. Across the five latest common checkpoints, the average across-seed Target-ASR sample standard deviation is 0.181 percentage points, with a maximum range of 0.758 percentage points.

Figure 9 shows that the optimized patches preserve a consistent dominant visual structure across training seeds. Their mean pixel-wise deviations from the cross-seed mean patch range from 0.0036 to 0.0069, corresponding to 0.36%–0.69% of the normalized RGB range. Pairwise patch MAE ranges from 0.0053 to 0.0107. Although the maximum deviation reaches 0.0435 at a small number of local positions, the 95th-percentile deviation remains between 0.0072 and 0.0150. Seed 20260604 exhibits relatively larger local deviations but retains the same dominant visual structure as the other runs.

Taken together, the small performance variation and high visual consistency indicate that physical refinement is stable with respect to the evaluated training randomness. Different training seeds introduce only minor local texture changes without causing substantial variation in attack performance.

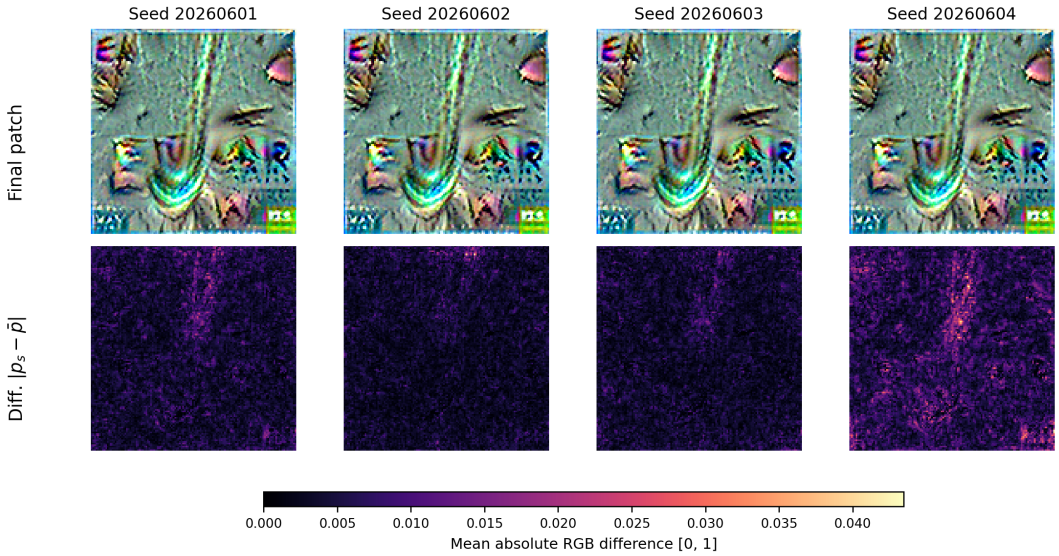


Figure 9: Cross-seed consistency of the optimized ear patches at the matched post-convergence checkpoint. The top row shows the patches obtained using four training seeds. Let $\bar{p} = \frac{1}{4}\sum_{s=1}^{4} p_s$ denote the pixel-wise cross-seed mean patch. The bottom row visualizes the per-pixel mean absolute RGB deviation $\delta_s(u, v) = \frac{1}{3}\sum_{c\in\{R,G,B\}} \left| p_s^{(c)}(u, v) - \bar{p}^{(c)}(u, v) \right|$. All deviation maps share the same color scale.

F Comparison Methods

We compare FactorizePatch with five baselines spanning digital optimization, generic transformation robustness, printability regularization, print–scan modeling, and monolithic digital-to-physical adaptation. The implementation details and fairness controls are described below.

Digital-only. This baseline directly optimizes the adversarial patch against the frozen victim model in the digital domain. No geometric or photometric transformation, printability regularization, learned print mapping, or camera model is applied during optimization. It measures the extent to which a patch optimized solely for digital inputs transfers to the physical environment.

EOT-only. This baseline uses the same targeted attack objective while applying standard geometric and photometric transformations to the patched image during optimization Athalye et al. (2018). It does not include total variation (TV) or non-printability score (NPS) regularization and does not use a learned printer or camera model. This baseline isolates the contribution of generic transformation-based robustness.

EOT+Printability. This baseline augments the same EOT pipeline with TV and NPS regularization Eykholt et al. (2018); Thys et al. (2019). TV regularization suppresses excessive local variation, while NPS encourages the optimized colors to remain close to a predefined printable-color set. This configuration represents a conventional physical-patch optimization strategy based on generic transformation robustness and printable-color constraints.

P&S. For P&S, we use the pretrained print–scan model released by the original authors Purnekar et al. (2025). The released model is inserted into our targeted patch-optimization loop without retraining its parameters. During optimization, the digital patch is first processed by the pretrained print–scan simulator and is then evaluated using the same frozen victim model and targeted attack objective as the other methods. P&S models the appearance changes introduced by the print–scan process, but does not explicitly model the subsequent camera-imaging stage used in our physical evaluation.

D2P Adaptation. At the time of our reproduction, the model weights and training data released for D2P were no longer accessible through its public repository. We therefore reimplement D2P according to the architecture, objective, and training procedure described in the original paper Jan et al. (2019).

Following the monolithic digital-to-physical formulation of D2P and matching the training-set scale used in the original study, we construct a dataset of 200 paired digital and print–camera images. The original D2P work randomly selected 200 training images from the ImageNet validation set and obtained their corresponding physical observations by printing and re-photographing each image Jan et al. (2019).

After training, the D2P mapping is frozen and inserted into the targeted patch-optimization loop. We replace the original task-specific attack objective with the same targeted CLIP or ResNet-50 loss used in our experiments, while retaining the single-stage physical mapping adopted by D2P. Thus, D2P Adaptation represents the combined print-and-capture process through one monolithic transformation, whereas FactorizePatch decomposes the physical path into sequential and independently controllable print appearance, scene deployment, and camera-formation stages.

Fairness controls. All comparison methods use the same target class, frozen victim model, victim-specific source-image set, patch resolution, initial patch, and optimization budget. They are evaluated using the same physical patch size, capture devices, scene layout, marked acquisition geometry, angular sweep, frame-sampling procedure, and physical evaluation set. For methods containing a learned physical mapping, the mapping is frozen during adversarial patch optimization while gradients are propagated through it to update the patch. Therefore, the primary differences among the compared methods lie in the transformations, regularization terms, and physical modeling strategies used during patch optimization.