# Generating Naturalistic Adversarial Patch Using Generative AI for Robustness Evaluation of Traffic Sign Recognition Systems

1st Muhammad Rizky Sya'ban
*School of Electrical Engineering and Informatics*
*Bandung Institute of Technology*
Bandung, Indonesia
mrizkysyaban99@gmail.com

2nd Rinaldi
*School of Electrical Engineering and Informatics*
*Bandung Institute of Technology*
Bandung, Indonesia
rinaldi@staff.stei.itb.ac.id

3rd Budi Rahardjo
*School of Electrical Engineering and Informatis*
*Bandung Institute of Technology*
Bandung, Indonesia
br@staff.stei.itb.ac.id

***Abstract*—Traffic Sign Recognition Systems (TSRS) are a crucial component of autonomous vehicles, yet they are highly vulnerable to adversarial attacks. Existing patch-based attacks are often unrealistic and lack robustness in dynamic, real-world conditions. This paper proposes a method to generate naturalistic and effective adversarial patches for evaluating TSRS robustness. Our approach utilizes a Generative Adversarial Network (GAN) augmented with a transformation module that simulates realistic patch placement and lighting, targeting the modern YOLOv8 object detection model. Experiments were conducted using a custom Indonesian traffic sign dataset, with seed patches sourced from the Quick, Draw! dataset to mimic common vandalism. The results show that the generated patches successfully degrade the performance of the target YOLOv8x model, achieving an Attack Success Rate (ASR) of 8.02% on the mAP50-95 metric in a white-box scenario. Furthermore, the attacks exhibit high transferability, reaching an ASR of 9.85% against the YOLO12x model in a black-box scenario. However, a subjective survey involving 24 participants revealed a fundamental trade-off: the most effective patches were consistently rated as the least natural, with the average naturalness score for adversarial patches (32.6%) being lower than that of the original seed patches (43.1%). This research underscores the challenge of balancing attack effectiveness with visual realism and contributes a framework for generating more realistic attacks to test TSRS security.**

***Keywords—Naturalistic Adversarial Patch, TSR, GAN, YOLO, Gen AI***

## I. INTRODUCTION

In recent decades, the advancement of artificial intelligence, particularly deep learning, has significantly impacted real-world technologies. A key driver of this transformation is the widespread adoption of deep neural networks (DNNs), which have become the state-of-the-art method for various visual data processing applications, including object recognition, image classification, and segmentation. The proficiency of DNNs in analyzing visual data has made them a cornerstone in the development of AI-based technologies, most notably in autonomous vehicles.

A reliable perception system is essential for autonomous vehicles to ensure accurate and safe decision-making. A critical component of this system is the ability to recognize traffic signs in real-time, which is paramount for navigation and safety. Modern Traffic Sign Recognition Systems (TSRS) leverage sophisticated DNN architectures like Convolutional Neural Networks (CNNs) to overcome the challenges of sign recognition in complex traffic environments.

However, despite their impressive performance, DNNs have been found to be vulnerable to minor perturbations in their input, a phenomenon known as adversarial examples. Szegedy et al. first demonstrated that minimal, often human-imperceptible, changes to an input can cause a model to make incorrect predictions[1]. This vulnerability is a major concern in safety-critical applications like autonomous driving, where a small disturbance to a traffic sign image could lead to catastrophic failures, such as ignoring a stop sign or misinterpreting a speed limit, thereby endangering passengers and other road users. To mitigate this, adversarial training (training a model on specifically crafted adversarial examples) has become a vital technique for enhancing model robustness.

Early adversarial examples modified every pixel of an image, making them difficult to replicate in the physical world. To address this, Brown et al. introduced patch-based adversarial examples, where the perturbation is confined to a small, localized area, or "patch," of the input image[2]. This approach is more practical for real-world application, as patches can be physically created as stickers and applied to objects. However, these initial patches often lacked the robustness to withstand variations in the physical environment, such as changes in camera angle, distance, and lighting, limiting their effectiveness outside of controlled digital settings.

Subsequent research sought to create more robust physical attacks. Eykholt et al. developed Robust Physical Perturbations (RP2), which produced graffiti-like adversarial patterns that successfully deceived TSRS by accounting for camera angles and distances[3]. Nevertheless, later benchmarks revealed that many existing adversarial patch attacks, including RP2, were not as effective in realistic scenarios as initially claimed. A further limitation is that most adversarial patches appear conspicuous and unnatural, making them easily identifiable by humans. To overcome this, researchers have explored using Generative Adversarial Networks (GANs) to generate more realistic and natural-looking patches. For instance, [4] introduced the Perceptual-Sensitive GAN (PS-GAN), which uses an attention mechanism to create patches that are better integrated with the image context while maintaining strong attack capabilities.

Motivated by these challenges and limitations, this research aims to develop a method for generating contextual and naturalistic adversarial patches using Generative AI that can be physically realized. This study focuses on designing a generative model to produce patches that are seamlessly integrated with the input image's context and are difficult for

human observers to detect. The generated patches will be evaluated under simulated real-world conditions to ensure their attack efficacy and visual realism across various physical environments. Ultimately, this research is expected to contribute to enhancing the robustness of traffic sign recognition systems against adversarial attacks in autonomous vehicle applications.

## II. Related Works

The study of adversarial attacks has evolved from purely digital manipulations to creating physically realizable threats, particularly for vision systems in autonomous vehicles. However, a significant challenge has been the development of effective real-world attacks and the methods to evaluate them, as physical testing is costly and simple synthetic data often fails to capture real-world complexities. Early research into physical threats focused on patch-based attacks, which proved to be a practical approach. For instance, [5] developed a location-independent patch attack that achieved over 90% success in deceiving digital Traffic Sign Recognition Systems (TSRS) and demonstrated a notable 72.2% success rate when physically printed and applied to signs.

While effective, these patches often appeared artificial, prompting a new direction of research focused on creating more naturalistic and visually inconspicuous attacks. To address this, generative models became a key tool. [4] introduced the Perceptual-Sensitive GAN (PS-GAN), which utilized a visual attention mechanism to generate patches that were visually harmonious with the target image while maintaining high attack efficacy. Their method demonstrated strong performance and high transferability in both digital and real-world tests. More recently, [6] proposed AdvDenoise, which employed a denoising diffusion model to generate robust and universal patches with greater efficiency, achieving an 82.49% attack success rate in the CARLA simulator.

Alongside the development of more sophisticated attacks, the research community has also focused on creating more realistic evaluation benchmarks. [7] introduced ImageNet-Patch, a dataset designed to benchmark model robustness against patches that undergo physical transformations like rotation and translation. Critically, the work by [8] with their Realistic Adversarial Patch (REAP) benchmark revealed a crucial insight. By applying patches to real-world images with accurate geometric and lighting transformations, they demonstrated that the effectiveness of many adversarial attacks was significantly overestimated by simpler simulations. This finding highlights a persistent "sim-to-real gap" and underscores the necessity for developing attack generation methods that explicitly account for realistic physical conditions, which is a central motivation for our work.

## III. Proposed Method

Previous research has shown that existing adversarial patches often fail in real-world scenarios due to three primary shortcomings: 1) the use of unnatural, noise-like patterns that are easily spotted by humans; 2) digital placement that ignores the physical boundaries of the target object; and 3) inconsistent lighting that makes the patch appear digitally pasted onto the image. To address these issues, we propose a hybrid framework that integrates the generative power of a Generative Adversarial Network (GAN) with realistic, physically-informed constraints. Our approach adapts the Perceptual-Sensitive GAN (PS-GAN) architecture and incorporates simulation principles from the Realistic Adversarial Patch (REAP) benchmark to generate patches that are both effective and naturalistic.

The proposed solution, illustrated in Fig. 1, is designed to generate adversarial patches that are effective against modern object detection models and are visually coherent with their environment. The process begins with a seed patch, a simple hand-drawn sketch, which is fed into a Generator (G). The Generator's task is to transform this seed into an adversarial patch optimized for an attack. Unlike the original PS-GAN, which targeted image classifiers, our framework is designed to attack a more complex Target Model (F) that is YOLOv8 object detector. The generated adversarial patch is not applied directly; instead, it is processed by a novel Transformation Module (T). This module realistically places the patch onto a ground-truth traffic sign image by simulating geometric perspective, conforming to the sign's physical boundaries, and adjusting its lighting to match the scene. The resulting adversarial image is then used in a dual-objective training process. First, it is fed to the Target Model (F) to calculate an adversarial loss, which pushes the Generator to create more effective attacks. Second, it is passed to a Discriminator (D), which is trained to distinguish between real traffic sign images and those containing our generated patches, compelling the Generator to produce more visually realistic and seamlessly integrated patches. This entire process creates a balanced training dynamic that optimizes for both attack potency and visual stealth.

### *A. Module Architecture*

Our framework consists of four primary modules: the Generator, the Discriminator, the Transformation Module, and the Target Model.

**Generator.** The Generator employs a U-Net architecture, inspired by PS-GAN and image-to-image translation work. It features a symmetric encoder-decoder structure with skip connections between corresponding layers, which allows low-level spatial information from the input patch to be preserved in the final output. To enhance generalization and prevent mode collapse, dropout layers are utilized in the decoder and remain active during both training and inference.

**Discriminator.** The Discriminator is a multi-layered convolutional neural network tasked with classifying its input image as either real or containing a fake patch. Each convolutional layer uses a LeakyReLU activation function and batch normalization to ensure training stability, concluding with a sigmoid activation layer that outputs the probability of visual realism.

**Transformation Module**. This module is crucial for bridging the sim-to-real gap by digitally simulating the physical application of a patch onto a traffic sign. The process involves three sequential steps as shown in Fig. 2:

- **Masking**. A pre-trained Mask R-CNN model first segments the traffic sign from the background image. We then apply contour detection and a convex hull algorithm to the resulting mask to produce a clean, geometrically precise shape (e.g., circle, octagon) that accurately represents the sign's physical boundaries.

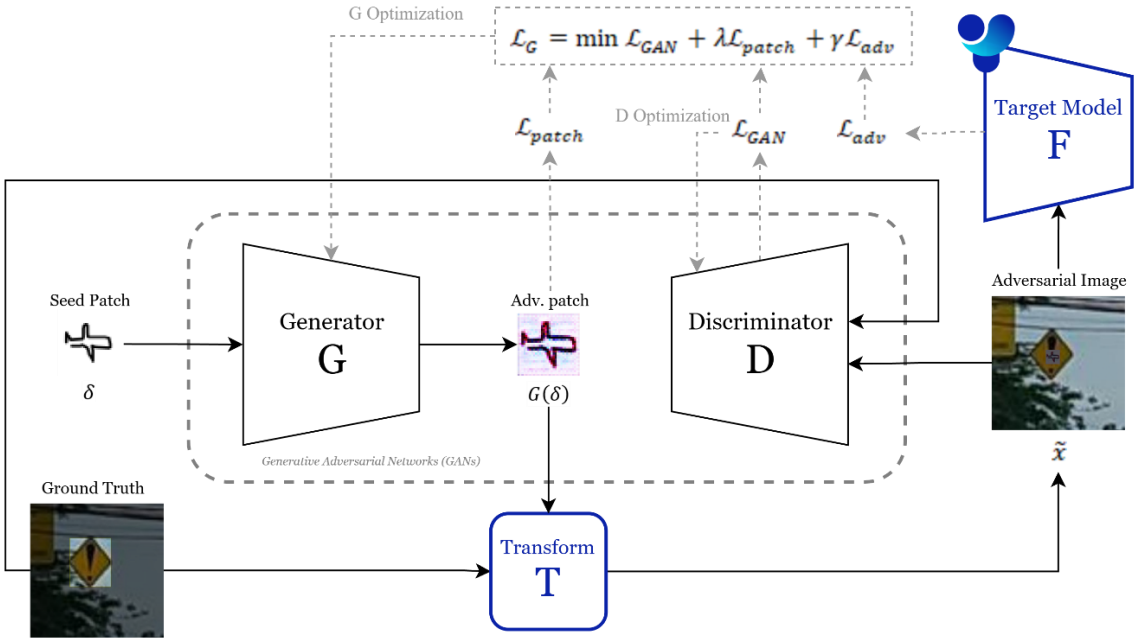


Fig 1. The proposed framework for generating naturalistic adversarial patches

This refined mask provides keypoints for subsequent patch placement.

- **Attentive Region Identification**. To maximize the attack's effectiveness, the patch is placed on the most visually important region of the sign. We use High-Resolution Class Activation Mapping (HiResCAM), a faithful interpretability method, on the target YOLOv8 model to generate a heatmap that identifies the areas most critical to its prediction. The patch is then centered on the peak activation point within this heatmap.
- **Relighting**. To ensure the patch appears naturally integrated with the sign, its lighting is adjusted to match the ambient conditions. Following the percentile method from the REAP benchmark, we analyze the pixel intensity distribution within the masked sign area to calculate a base brightness (β) and a contrast range (α). The patch's pixel values (P) are then linearly transformed according to (1), which aligns its color histogram with that of the target sign.

$$P_{relighted} = \alpha P + \beta \quad (1)$$

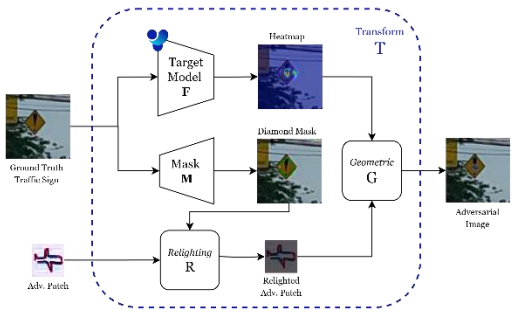


Fig. 2. Transform Module Architecture

**Target Model.** The target for our adversarial attack is YOLOv8, a state-of-the-art one-stage object detector. To create a realistic and consistent evaluation scenario, the YOLOv8 model was specifically fine-tuned on the custom Indonesian traffic sign dataset used throughout the experiments.

### *B. Problem Formulation*

The generation of naturalistic adversarial patches is formulated as an optimization problem involving a Generator (G) and a Discriminator (D). The Generator aims to minimize a composite loss function as shown in (2), while the Discriminator works to maximize its ability to identify generated patches as expressed in (3). In these equations, λ and γ are hyperparameters that balance the trade-off between visual realism and attack strength.

$$\mathcal{L}_G = \min \mathcal{L}_{GAN} + \lambda \mathcal{L}_{patch} + \gamma \mathcal{L}_{adv} \quad (2)$$

$$\mathcal{L}_D = \max \mathcal{L}_{GAN} \quad (3)$$

**Visual Fidelity ($\mathcal{L}_{GAN}$).** The foundation of the framework is the standard generative adversarial loss, which is defined in (4). This loss trains the Discriminator D to distinguish authentic images from the adversarial images ($\tilde{x}$) containing patches from the Generator G, while simultaneously training G to produce patches that D cannot distinguish from reality.

$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}_x[\log D(\delta, x)] + \mathbb{E}_x[\log(1 - D(\delta, \tilde{x})] \quad (4)$$

**Natural Consistency ($\mathcal{L}_{patch}$)**: To ensure the generated patch retains a natural, scribble-like appearance, a patch loss is defined in (5) as the $\ell_2$ *norm* between the generated adversarial patch $G(\delta)$ and the original seed patch $\delta$. This regularizer in (5) prevents the generator from producing overly complex, artificial patterns that would be easily detectable by a human observer.

$$\mathcal{L}_{patch} = \mathbb{E}_\delta \| G(\delta) - \delta \|_2 \quad (5)$$

**Attacking Ability ($\mathcal{L}_{adv}$)** . The adversarial loss is designed to maximize the error of the target model F. The formula shown in (6) where the objective is to maximize the class confidence score F for predictions j that have a high Intersection over Union (IoU) with the ground truth. This effectively suppresses correct detections It is calculated as the maximum objectness score for the ground-truth class within the predictions, thereby training the generator to produce patches that cause the target model to either misclassify the traffic sign or fail to detect it altogether.

$$\mathcal{L}_{adv} = \frac{1}{N}\sum_{i=1}^{N} \max_{j,\ IoU>\tau} F_{cls}^{j}(\tilde{x}_i) \quad (6)$$

### C. Threat Model

The attack scenario for this research is defined by the following threat model:

- **Adversary's Knowledge**. We assume a white-box attack, where the adversary has complete access to the target model's architecture, parameters, and loss function. This allows for the direct calculation of gradients to efficiently optimize the adversarial patch.
- **Adversary's Goal.** The goal is an untargeted attack. The objective is to cause the TSRS to fail in its detection or classification task, rather than forcing it to predict a specific incorrect class. This reflects a more general and practical real-world attack scenario.
- **Adversary's Capabilities**. The adversary is capable of physically placing a patch within the legitimate boundaries of a traffic sign. This is simulated through the realistic geometric and lighting transformations in our framework. The adversary cannot, however, modify the internal weights or architecture of the target model.

## IV. Implementations

This section details the datasets, environment, and specific implementation parameters used to conduct the research, ensuring the experiments are reproducible.

### A. Datasets

Two distinct datasets were utilized: one for the traffic sign images to be attacked and another to provide the initial patterns for the adversarial patches**.**

#### 1) Indonesian Traffic Sign Dataset

To ensure real-world relevance for the target environment, a custom dataset of Indonesian traffic signs was prepared.

- The initial data was sourced from a public repository by Adhy Wiranto, which contained 2100 images across 21 classes collected from various sources like Google Maps and smartphone captures
- A data cleaning process was performed to remove three classes corresponding to traffic lights (red, yellow, and green traffic light) as they were not static signs, resulting in a dataset of 18 classes.
- The data was augmented to improve model generalization and simulate real-world variations. Augmentations included rotation (±15°), shearing (±10°), exposure changes (±10%), and blur (up to 2.5 pixels).
- After augmentation, the final dataset consisted of 4294 images, split into training (3780 images), validation (338 images), and test (176 images) sets.

#### 2) Seed Patch Dataset

The initial patterns for the adversarial patches were sourced from the Quick, Draw! dataset.

- This dataset was chosen because its vast collection of simple, hand-drawn sketches resembles the type of vandalism (graffiti or stickers) commonly found on public signs, providing a naturalistic foundation for the adversarial patches.
- A subset of 11 classes with simple shapes relevant to the traffic sign context was selected, including "airplane," "circle," "star," and "car"
- Since the data is stored in a vector format, a conversion function was implemented to transform the sketches into 56×56 raster images suitable for input to the generator model.

### B. Experimental Environment

All experiments were conducted on the DGX server at the ITB AI Center. The specific hardware and software environment is detailed below:

- Ubuntu 18.04.6 LTS.
- 8x Tesla V100-SXM2-32GB
- Python 3.11.3 managed with Miniconda

### C. Implementation Details

The training process followed the conceptual framework outlined in the Proposed Method.

- The Generator and Discriminator were trained using different optimizers to maintain stability; the Generator used the Adam optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$, while the Discriminator used SGD with a momentum of 0.9.
- The models were trained for 250 epochs with a batch size of 16. The learning rates were initially set and then reduced by 10% every 20 epochs.
- For the patch generation, the patch-to-image area ratio was set to 0.1, with the generator taking a 32×32 patch as input and applying it to a 256×256 image.

## V. Experiments

This section presents the experimental evaluation of the proposed framework. The experiments were designed to assess the effectiveness, transferability, and visual realism of the generated adversarial patches. All evaluations were conducted under controlled digital settings, simulating realistic physical conditions through the transformation module described in Section III.

### A. Hyperparameter Tuning

Due to the known training instability of Generative Adversarial Networks (GANs), a manual and iterative hyperparameter tuning process was conducted. This approach was chosen over automated methods like grid search to allow for careful observation of the complex training dynamics between the generator and discriminator. The process focused on finding an optimal balance by adjusting five key

parameters: the number of generator optimization steps (k), the coefficient for patch loss (λ), the coefficient for adversarial loss (γ), and the learning rates for both the generator and the discriminator. The final optimal configuration was determined to be λ=0.0025, γ=2.0, k=6, a generator learning rate of 0.001, and a discriminator learning rate of 0.00002.

### *B. Attack Effectiveness*

This experiment measured the direct impact of the adversarial patches on the model they were trained against, YOLOv8x, in a white-box scenario. The model's performance was measured on three versions of the test set: the original clean data, the data with unmodified seed patches applied, and the data with the final generated adversarial patches. The seed patch test was crucial to validate that performance degradation was due to the patch's adversarial nature, not merely visual occlusion.

As shown in TABLE I, the YOLOv8x model achieved an mAP50-95 of 0.885 on clean data. When the adversarial patches were applied, the performance dropped to 0.814. This degradation resulted in an Attack Success Rate (ASR) of 8.02% on the mAP50-95 metric. In contrast, the unmodified seed patches only caused a marginal performance drop to 0.86, demonstrating that the optimized adversarial nature of the generated patches was the primary cause of the model's failure.

TABLE I. The Effectiveness of Adversarial Patch Attacks on YOLOv8x

| Data | Precision | Recall | mAP50 | mAP50-95 |
|---|---|---|---|---|
| *Clean* | 0.988 | 0.986 | 0.993 | 0.885 |
| *Seed patch* | 0.955 | 0.936 | 0.98 | 0.86 |
| *Adversarial patch* | 0.82 | 0.889 | 0.943 | 0.814 |
| | | | **ASR**: 5.04% | **ASR**: 8.02% |

### *C. Attack Transferability*

To assess the real-world viability of the patches in black-box scenarios, their ability to deceive models they were not trained on was tested. The attacks were transferred to other YOLO variants with different capacities (YOLOv8n, YOLOv8m) and different architectures (YOLOv11x, YOLOv12x).

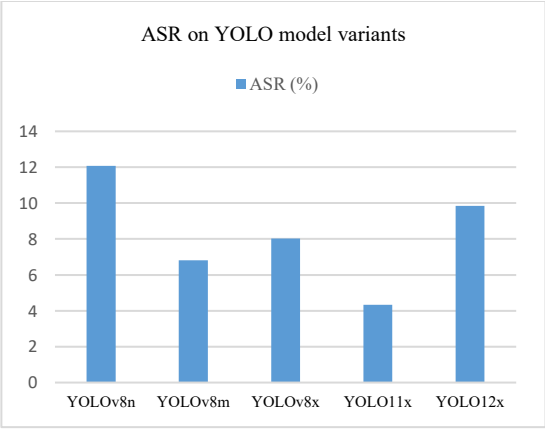


Fig. 3. Comparison of ASR for adversarial patches generated across YOLO model variants

The results, detailed in TABLE II, TABLE III, TABLE IV, and TABLE V, and summarized in Fig. 3, show strong transferability. The attack was highly effective against the smaller YOLOv8n model, achieving an ASR of 12.08%. It also successfully transferred to models with fundamentally different architectures, reaching an ASR of 9.85% against YOLOv12x. This indicates that the patches exploit a more general vulnerability in object detection models rather than a weakness specific to the YOLOv8x architecture, highlighting their practical threat potential.

TABLE II. The Effectiveness of Adversarial Patch Attacks on YOLOv8n

| Data | Precision | Recall | mAP50 | mAP50-95 |
|---|---|---|---|---|
| *Clean* | 0.981 | 0.977 | 0.992 | 0.861 |
| *Seed patch* | 0.928 | 0.892 | 0.951 | 0.807 |
| *Adversarial patch* | 0.881 | 0.849 | 0.905 | 0.757 |
| | | | **ASR**: 8.77% | **ASR**: 12.08% |

TABLE III. The Effectiveness of Adversarial Patch Attacks on YOLOv8m

| Data | Precision | Recall | mAP50 | mAP50-95 |
|---|---|---|---|---|
| *Clean* | 0.989 | 0.975 | 0.989 | 0.88 |
| *Seed patch* | 0.964 | 0.920 | 0.975 | 0.856 |
| *Adversarial patch* | 0.903 | 0.859 | 0.954 | 0.820 |
| | | | **ASR**: 3.54% | **ASR**: 6.82% |

TABLE IV. The Effectiveness of Adversarial Patch Attacks on YOLO11x

| Data | Precision | Recall | mAP50 | mAP50-95 |
|---|---|---|---|---|
| *Clean* | 0.978 | 0.972 | 0.993 | 0.877 |
| *Seed patch* | 0.918 | 0.946 | 0.976 | 0.856 |
| *Adversarial patch* | 0.876 | 0.911 | 0.961 | 0.839 |
| | | | **ASR**: 3.22% | **ASR**: 4.33% |

TABLE V. The Effectiveness of Adversarial Patch Attacks on YOLO12x

| Data | Precision | Recall | mAP50 | mAP50-95 |
|---|---|---|---|---|
| *Clean* | 0.976 | 0.991 | 0.991 | 0.873 |
| *Seed patch* | 0.82 | 0.938 | 0.969 | 0.834 |
| *Adversarial patch* | 0.834 | 0.836 | 0.926 | 0.787 |
| | | | **ASR**: 6.56% | **ASR**: 9.85% |

### *D. Attack Naturalness*

To quantify the perceived visual naturalness of the patches, a subjective survey was conducted with 24 participants, following a methodology adapted from related studies. The survey was designed to determine if the generated patches were inconspicuous and resembled common vandalism. Participants were asked to rate and rank images with the original seed patches and the final adversarial patches.

The results revealed a fundamental trade-off between attack effectiveness and visual realism. The average naturalness score for the generated adversarial patches (32.6%) was significantly lower than that of the original seed

patches (43.1%). Furthermore, as illustrated in Fig. 4, there was a clear inverse correlation: patches that were most effective at degrading the model's performance (lowest mAP50-95) were consistently ranked as the least natural by human observers. This finding underscores the core challenge in generating attacks that are both potent and visually stealthy.

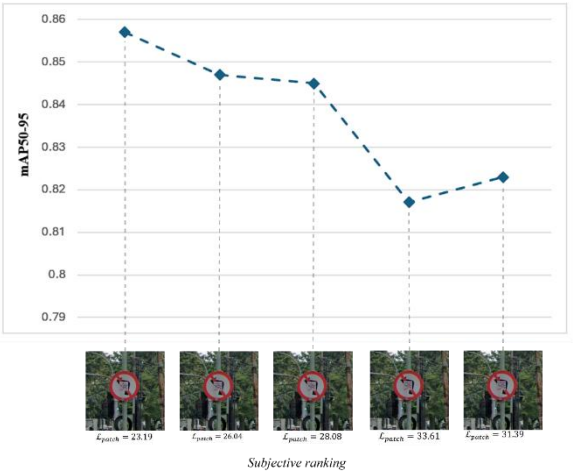


Fig. 4. Correlation Between Subjective Survey Rankings and Attack Effectiveness

## VI. Discussions

The experimental results provide several key insights into the effectiveness, transferability, and inherent limitations of generating naturalistic adversarial patches. This section analyzes these findings in detail.

**Attack Effectiveness and Behavior**. The quantitative results confirm that the proposed method can significantly degrade the performance of a target object detection model. In the white-box scenario against YOLOv8x, the attack achieved an 8.02% ASR on the mAP50-95 metric. A deeper analysis reveals the attack's dual impact on the Traffic Sign Recognition System (TSRS). The sharp drop in precision from 0.988 to 0.82 indicates a surge in False Positives, meaning the model correctly located traffic signs but assigned them the wrong class label. Simultaneously, the decrease in recall from 0.986 to 0.889 points to an increase in False Negatives, where the model failed to detect some traffic signs altogether. This demonstrates that the patch successfully induces both misclassification and detection failures.

Crucially, the comparison with the unmodified seed patches validates that this performance degradation is a direct result of the adversarial optimization, not merely visual occlusion. The seed patches caused only a marginal 2.82% drop in mAP50-95, whereas the optimized adversarial patches caused a nearly threefold greater reduction of 8.02%. This significant difference empirically proves that the adversarial patterns are the dominant factor behind the model's failure.

**Attack Transferability and Architectural Robustness.** The investigation into attack transferability revealed a complex relationship between model architecture and vulnerability. Generally, the patches demonstrated the ability to affect models they were not trained on, but the degree of success varied significantly**.**

When attacking models of the same family, the smaller YOLOv8n proved highly susceptible, with an ASR of 12.08%. However, this high rate was partly attributed to the model's inherent fragility to any visual disruption, as even the non-adversarial seed patch caused a significant performance drop. This aligns with the understanding that models with smaller capacity have less robust feature representations**.**

Transferability to models with different architectures depended heavily on their specific feature extraction and processing mechanisms. The YOLOv11x model, which incorporates advanced attention modules to focus on important image regions, showed high resilience with a very low ASR of 4.33%. This suggests its architecture is effective at filtering out the irrelevant noise introduced by the patch. Conversely, the YOLOv12x, which uses a transformer-based architecture, was surprisingly vulnerable, with an ASR of 9.85%. A possible explanation is that its global attention mechanism, while powerful, may be sensitive to the salient, high-frequency patterns of the adversarial patch, misinterpreting them as important signals.

These findings suggest that while transferability is possible, its success is governed by the similarity of feature representations between models and the specific defense mechanisms, like attention, built into the target's architecture.

**The Trade-off between Naturalness and Efficacy.** A fundamental finding of this research is the quantifiable trade-off between a patch's adversarial strength and its visual naturalness. This conflict arises because the optimization process that minimizes adversarial loss ($\mathcal{L}_{adv}$) inherently pushes the generator to create high-frequency visual artifacts that are highly effective at disrupting convolutional filters but appear unnatural to the human eye. The patch loss ($\mathcal{L}_{patch}$) acts as a regularizer to counteract this tendency.

This dynamic is visually represented in the training loss graphs, as seen in Fig. 5 While the adversarial loss steadily decreases, indicating the patch is becoming more potent, the patch loss slowly increases after an initial drop. This empirically shows that to enhance its attacking ability, the generator is forced to create patterns that deviate further from the original seed patch, thereby increasing distortion and reducing naturalness. The subjective survey results quantitatively confirmed this, as the most effective patches consistently received the lowest naturalness rankings from human participants.

Fig. 5. Patch ratio variations after application to traffic signs

**Limitations of Digital Simulation and GAN Training.** While the transformation module was designed to enhance realism, it has inherent limitations in replicating complex

physical phenomena. The percentile-based relighting is a linear approximation and cannot model non-linear light interactions like specular glare or dynamic shadows. Furthermore, the module's reliance on an automated segmentation model, without ground-truth segmentation masks, sometimes led to imperfect masking. This inaccuracy caused inconsistencies in the final patch-to-sign ratio, as illustrated in, where some patches appear disproportionately large or small. These factors contribute to a "sim-to-real gap," suggesting that the attack's measured effectiveness in a digital environment may be attenuated in a true physical implementation.

Finally, the training process itself highlighted the challenge of mode collapse in GANs. During hyperparameter tuning, certain suboptimal configurations caused the generator to converge on a few effective but monotonous patterns, ceasing to produce diverse outputs. This phenomenon limits the variety of potential attacks and makes them easier to defend against, underscoring the critical importance of careful, iterative tuning to achieve a stable and productive generator**.**

## VII. Conclusions

The experimental results provide several key insights into the effectiveness, transferability, and inherent limitations of generating naturalistic adversarial patches. This section analyzes these findings in detail. This research successfully developed and evaluated a framework for generating naturalistic adversarial patches to test the robustness of Traffic Sign Recognition Systems (TSRS). Our method integrates a Generative Adversarial Network (GAN) with a realistic transformation module that simulates physical placement and lighting conditions, producing patches that are inherently more robust for real-world scenarios. The experiments demonstrate that the generated patches are effective at deceiving modern object detectors. In a white-box scenario, the attack degraded the performance of the target YOLOv8x model with an Attack Success Rate (ASR) of 8.02% on the mAP50-95 metric. More importantly, the patches exhibited strong transferability in black-box scenarios, achieving an even higher ASR of 9.85% against the architecturally different YOLOv12x model, proving the exploited vulnerability is general rather than model-specific.

However, the study also uncovered a fundamental trade-off between attack effectiveness and visual realism. A subjective survey revealed that the most potent adversarial patches were consistently rated as the least natural by human observers. The average naturalness score for the final adversarial patches (32.6%) was notably lower than for the original seed patches (43.1%), highlighting the challenge of creating attacks that are both powerful and imperceptible.

Future work should focus on bridging the "sim-to-real gap" through physical validation of the printed patches. Further refinement could be achieved by employing more sophisticated 3D rendering engines for simulation and exploring advanced generative models, such as Denoising Diffusion Probabilistic Models (DDPMs), to improve training stability and output diversity. Finally, investigating alternative perceptual loss functions may help to better balance the crucial trade-off between adversarial strength and naturalness.

## References


[1] C. Szegedy *et al.*, “Intriguing properties of neural networks,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, ICLR, Dec. 2014. [Online]. Available: http://arxiv.org/abs/1312.6199

[2] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, “Adversarial Patch,” in *Proceeding of the 31st Conference on Neural Information Processing System (NIPS)*, Curran Associates Inc., Dec. 2017. [Online]. Available: http://arxiv.org/abs/1712.09665

[3] K. Eykholt *et al.*, “Robust Physical-World Attacks on Deep Learning Visual Classification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, IEEE, 2018, pp. 1625–1634. [Online]. Available: https://iotsecurity.eecs.umich.edu/#roadsigns

[4] A. Liu *et al.*, “Perceptual-Sensitive GAN for Generating Adversarial Patches,” in *Proceeding of the AAAI Conference on Artificial Intelligence*, Jul. 2019, pp. 1028–1035. doi: 10.1609/aaai.v33i01.33011028.

[5] B. Ye, H. Yin, J. Yan, and W. Ge, “Patch-Based attack on traffic sign recognition,” in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, Institute of Electrical and Electronics Engineers Inc., Sep. 2021, pp. 164–171. doi: 10.1109/ITSC48978.2021.9564956.

[6] J. Li, Z. Wang, and J. Li, “AdvDenoise: Fast Generation Framework of Universal and Robust Adversarial Patches Using Denoise,” in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2024. Accessed: Jan. 06, 2025. [Online]. Available: https://github.com/advdenoise/

[7] M. Pintor *et al.*, “ImageNet-Patch: A dataset for benchmarking machine learning robustness against adversarial patches,” *Pattern Recognit*, vol. 134, p. 109064, Feb. 2023, doi: 10.1016/J.PATCOG.2022.109064.

[8] N. Hingun, C. Sitawarin, J. Li, and D. Wagner, “REAP: A Large-Scale Realistic Adversarial Patch Benchmark,” in *Proceedings of the International Conference on Computer Vision (ICCV)*, IEEE, Aug. 2023. [Online]. Available: http://arxiv.org/abs/2212.05680