This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3748461

Sensitivity-Driven Strategic Attack and Adaptive Diffusion Defense for Semantic Communication Systems

Tung Son Do*, and Sungrae Cho*
*School of Computer Science and Engineering, Chung-Ang University, Seoul 06974, South Korea.
Email: tsdo@uclab.re.kr, srcho@cau.ac.kr

***Abstract*—Deep learning-based semantic communication exposes two physically distinct attack surfaces: the data path into the semantic encoder and the wireless channel carrying its output. Existing attacks typically exercise either surface in isolation. In this paper, we formulate a joint transmitter-and-channel attack problem and propose a sensitivity-driven strategic attack, termed SenSA. In particular, SenSA computes the coordinate-averaged classification loss gradients at both the source image and the received channel symbols, weights them by a signal-to-noise ratio (SNR)-dependent factor derived from a first-order linear minimum mean square error (LMMSE) receiver model, and selects a transmitter-only, receiver-only, or joint projected gradient descent (PGD) mode with proportional iteration allocation for each sample under separate per-surface perturbation bounds. To mitigate these perturbations at the receiver, we propose a hybrid diffusion-routed adaptive defense, termed HYDRA, in which a convolutional perturbation analyzer estimates the corruption level of each decoded image to set the forward-diffusion starting timestep and routes denoising features across mixture-of-experts (MoE) prediction heads with confidence-gated classifier-free guidance (CFG) and class-prototype regularization. We further evaluate both frameworks under defense-unaware, black-box transfer, and defense-aware white-box attacks that differentiate through the purification pipeline, reporting image reconstruction fidelity alongside the Penalized Attack Success Rate (PASR), which subtracts incorrect-to-correct transitions from the attack success count. Simulation results on Imagenette over AWGN channels demonstrate that per-sample mode selection outperforms fixed-surface variants across all evaluated SNR regimes; against defense-unaware attacks, HYDRA attains the lowest PASR among the evaluated defenses with an inference latency of 9.46 ms per sample.**

***Index Terms*—Adversarial attack, adversarial defense, diffusion model, mixture of experts, semantic communication.**

I. Introduction

A. Background and Motivation

SEMANTIC communication shifts the design target from bit-level accuracy to goal-oriented, meaning-aware exchange [1]. In this paradigm, deep models extract and transmit task-relevant features, saving bandwidth while maintaining comparable task performance [2]. To support heterogeneous edge transceivers and large-scale user populations, Do *et al.* [3] investigated knowledge distillation for lightweight encoders, whereas Do *et al.* [4] developed game-theoretic resource allocation for scalable multi-modal transmission. Consequently, semantic communication has emerged as a candidate enabler for sixth-generation (6G) networks [5] in autonomous driving, the Internet of Things, industrial automation, and immersive media [6]. However, the reliance on deep neural networks relocates the security challenge: the primary protection target is no longer raw data integrity but the semantic representation and the models that produce it [7]. Specifically, adversarial perturbations propagate through the integrated encoder–decoder architecture and can disrupt task execution at the receiver [8], whether applied to the source data before encoding [9], injected into the wireless channel [2], or aimed at the knowledge shared between transmitter and receiver [6]. Reported effects are substantial: classification accuracy drops from 78.4% to 11.6% [8], attack success rates exceed 87% white-box and 89% black-box [7], and signal-to-distortion degradation reaches up to 2.9× [6]. In safety-critical settings such as autonomous driving, a misclassified traffic sign is a safety failure, not a quality-of-service one.

Three main gaps motivate this work. First, existing attacks rely either on computationally expensive population-based or iterative searches [7] or fixed input-agnostic perturbation patterns [2], rather than allocating optimization steps based on per-sample loss gradients. Second, they commit to a single surface, input or channel, leaving joint transmitter- and receiver-side vulnerabilities unexploited. Third, on the defense side, adversarial training dominates [2], which binds robustness to the specific attacks seen during training and degrades clean task accuracy; while diffusion purification avoids attack-specific retraining in computer vision [10], existing semantic communication purifiers apply a fixed forward-noising schedule regardless of channel SNR or perturbation magnitude, which can under-clean strong perturbations or over-smooth clean reconstructions. Motivated by these observations, we study joint input-and-channel attacks and receiver-side diffusion purification for semantic communication systems. In particular, we investigate how to select the attack domain and split PGD iterations across the transmitter and receiver per sample, how to set the forward-diffusion starting timestep from the estimated corruption level and channel SNR, and how to evaluate net robustness under defense-aware white-box attacks.

B. Contributions

In this work, the main contributions are summarized as follows:

- We formulate a joint adversarial attack problem across the encoder input and the wireless channel under separate

Authorized licensed use limited to: Chung-ang Univ. Downloaded on November 05,2026 at 01:08:18 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3748461

pixel-domain and symbol-domain perturbation budgets. To solve this problem, we propose a sensitivity-driven strategic attack, termed SenSA. In particular, SenSA computes the coordinate-averaged classification loss gradients with respect to both the source image and the received channel symbols, weights them by an SNR-dependent factor $\alpha(\gamma)$, and compares the resulting ratio against a threshold to select a transmitter-only, receiver-only, or joint attack mode for each sample. Accordingly, PGD iterations are allocated to the selected domain and executed via momentum-accelerated updates with fixed-threshold early stopping.
- To remove adversarial perturbations at the receiver without attack-specific retraining at deployment, we propose a diffusion-based purification framework, termed HYDRA. In this framework, a multi-head convolutional analyzer estimates the corruption level of each decoded image and combines it with the channel noise variance to select the forward-diffusion starting timestep. Therefore, the purifier scales the forward noise variance to the estimated corruption level rather than applying a fixed diffusion schedule across all inputs. Furthermore, the reverse DDIM denoising stage integrates top-k mixture-of-experts noise prediction, confidence-gated classifier-free guidance, and exponential moving average (EMA) class-prototype blending to preserve class-discriminative features with low inference latency.
- To assess both frameworks, we evaluate the system under defense-unaware, black-box transfer, and defense-aware white-box attacks. In the white-box setting, the adversary differentiates end-to-end through the purification pipeline via backward pass differentiable approximation (BPDA) and expectation over transformation (EOT). In addition, we employ the penalized attack success rate (PASR) metric to measure net classification error increase by subtracting incorrect-to-correct transitions from correct-to-incorrect transitions, evaluated alongside multi-metric image reconstruction fidelity and per-sample inference latency.

C. Paper Organization and Key Notation

Section II reviews related work; Section III details system and threat models; Sections IV and V present SenSA and HYDRA; Section VI covers training and evaluation metrics, including PASR; Section VII reports simulation results; and Section VIII concludes. Table I summarizes the notation.

II. Related Work

A. Deep Learning-Based Semantic Communication

Deep learning-based semantic communication has emerged as a promising paradigm for efficient wireless transmission by extracting and transmitting task-relevant semantic features rather than raw bits. Bourtsoulatze *et al.* pioneered deep joint source-channel coding (DeepJSCC), which jointly optimizes the encoder and decoder using an autoencoder-like architecture [11]; the survey in [1] traces subsequent extensions across other modalities and the shift toward task-oriented objectives. For image transmission, Nan *et al.* [2] proposed MobileSC, a lightweight semantic communication framework considering memory and computation constraints. To tackle stringent hardware and bandwidth constraints on resource-limited transceivers, Do *et al.* [12] developed BidDeepSC-1.58b, an extremely quantized 1.58-bit bidirectional semantic communication framework integrating slimmable neural architectures with ternary weight representations for dynamic capacity adaptation.

Despite these advances in transmission efficiency, early semantic communication systems largely overlooked security vulnerabilities. The reliance on deep neural networks makes these systems inherently susceptible to adversarial perturbations, motivating the need for systematic investigation of both attack vectors and defense mechanisms.

B. Adversarial Attacks on Semantic Communication

Adversarial examples on image classifiers are produced by a standard family of gradient-based methods spanning iterative and margin-optimizing formulations [13], [14]. Semantic communication inherits this vulnerability and adds attack surfaces of its own in the channel and the shared knowledge base [5]; we organize the attacks below by attack surface and knowledge assumption.

Input-level attacks apply adversarial perturbations to the source data before semantic encoding. Shi *et al.* [9] studied how adversarial attacks perform across diverse channel environments, including AWGN and fading channels, revealing that channel effects significantly impact attack effectiveness.

Physical-layer attacks inject perturbations directly into the wireless channel. Nan *et al.* [2] proposed SemAdv, a perturbation generator that crafts input-agnostic, imperceptible, and semantic-oriented adversarial distortions over the air. Liu *et al.* [7] developed a systematic channel-based attack framework achieving over 87% white-box and 89% black-box attack success rates. Li *et al.* [8] introduced SemBAT, a black-box attack method using particle swarm optimization that reduces classification accuracy from 78.4% to 11.6%.

Semantic interpretation attacks target the shared knowledge between transmitter and receiver. Hoang *et al.* [6] proposed attacks exploiting vulnerabilities in semantic interpretation, degrading the signal-to-distortion scores by up to 2.9× across various channel conditions. Hua *et al.* [15] developed ISGU for targeted attacks on speech semantic communication, achieving over 96% attack success rate while maintaining imperceptibility.

Table II summarizes the comparison between existing attack methods and the proposed SenSA attack. The last two columns record the dimensions on which baselines retain an advantage over SenSA: gradient-free operation, available to the black-box attacks, and input-agnostic (universal) perturbations, offered by SemAdv. SenSA provides neither: it requires white-box gradients and optimizes a perturbation per sample.

C. Defense Mechanisms for Semantic Communication

Defense strategies against adversarial attacks in semantic communication can be broadly categorized into adversarial

Authorized licensed use limited to: Chung-ang Univ. Downloaded on November 05,2026 at 01:08:18 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3748461

TABLE I
SUMMARY OF NOTATION

Symbol	Description	Symbol	Description	Symbol	Description	Symbol	Description
System Model and Threat Model							
$\mathbf{x}$	Input image	c, $\hat{c}$	True, predicted class label	C, H, W	Channels, height, width	K	Number of classes
$\mathcal{E}_{\boldsymbol{\theta}}$, $\mathcal{D}_{\boldsymbol{\phi}}$	Sem. encoder, decoder	$\mathcal{F}_{\boldsymbol{\psi}}$	Classifier	$\mathbf{z}$, d_z	Channel input, dimension	P	Per-symbol power
$\mathbf{n}$, σ_n^2	Noise, per-symbol power	$\mathbf{y}$	Received signal	γ	Channel SNR (dB)	$\hat{\mathbf{x}}$	Reconstructed image
$\boldsymbol{\delta}_{\text{tx}}$, $\boldsymbol{\delta}_{\text{rx}}$	TX, RX perturbation	ϵ, ϵ_{rx}	TX/RX budget (ℓ_∞/ℓ_2)	$\mathbf{x}_{\text{adv}}$, $\mathbf{y}_{\text{adv}}$	Adv. image, adv. signal	$\hat{\mathbf{x}}_{\text{pur}}$, $\mathcal{P}$	Purified image, purifier
$\mathcal{L}_{\text{task}}$	Task loss	$\mathcal{L}_{\text{recon}}$, $\mathcal{L}_{\text{cls}}$	Recon., CE loss	λ_{cls}	Classification weight	$d_{\max}$, $T_{\max}$	Distort., latency bounds
SenSA Attack							
$\mathcal{S}_{\text{tx}}$, $\mathcal{S}_{\text{rx}}$	TX, RX sensitivity	$\mathcal{L}_{\text{rx}}$	RX classification loss	$\hat{\mathbf{s}}$	Received feature vec.	$\mathcal{S}_{\text{tx}}^{\text{eff}}$, $\mathcal{S}_{\text{rx}}^{\text{eff}}$	Effective sensitivities
$\alpha(\gamma)$, γ_{lin}	Ch. weight, linear SNR	$\text{Proj}_\epsilon^{(\infty)}$, $\text{Proj}_\epsilon^{(2)}$	ℓ_∞/ℓ_2 projections	r_{tx}, r_{rx}	Sensitivity ratios	ν_0	Stability constant
τ, τ_{stop}	Strategy/stop thresholds	ρ	Coordination ratio	μ	Momentum coefficient	$\mathbf{g}^{(i)}$, $\mathbf{v}^{(i)}$	Gradient, momentum
η_{tx}, η_{rx}	PGD step sizes	T_{tx}, T_{rx}	PGD iterations	T_{tx}^*, T_{rx}^*	Joint-mode iterations	p_{true}	True-label confidence
HYDRA Defense							
$D_{\boldsymbol{\omega}}$	HYDRA defense	$G_{\boldsymbol{\xi}}$	Perturbation analyzer	$\mathbf{f}$, $\mathbf{q}$	CNN feat., route logits	p, s	Detect. prob., corrupt. est.
$\hat{c}_a$, $p_{\hat{c}_a}$	Analyzer class, confidence	N_e	Number of experts	k, $\mathcal{I}$	Top-k, selected experts	ω_i, $\hat{\boldsymbol{\epsilon}}^{(i)}$	Expert weight, noise
β_{eff}	Effective noise ratio	β_{base}, $\beta_{\max}$	Noise scaling range	t_{start}, $\bar{\alpha}_t$	Diff. start, schedule	T_{diff}, N_{DDIM}	Diffusion, DDIM steps
$\mathbf{h}$	U-Net features	$\mathbf{e}_t$, $\mathbf{e}_c$	Time/class embeddings	w, w_{eff}	CFG scale, gated scale	τ_c, τ_T	Conf. thresh., CFG temp.
$\hat{\boldsymbol{\epsilon}}_{\text{CFG}}$	CFG noise prediction	$\mathbf{a}_j$, λ_a	Sem. anchor, strength	ζ, (H_a, W_a)	Anchor EMA, resolution	R, κ	Purif. rounds, decay
Training and Evaluation							
$\mathcal{L}_{\text{total}}$, $\mathcal{L}_{\text{diff}}$	Total, diffusion loss	$\mathcal{L}_{\text{aux}}$, $\mathcal{L}_{\text{anc}}$	Auxiliary, anchor loss	$\mathcal{L}_{\text{det}}$, $\mathcal{L}_{\text{lb}}$	Detect., load-bal. loss	λ_{aux}, λ_{anc}	Aux/anchor weights
λ_{det}, λ_{lb}	Det./LB weights	p_{SenSA}, p_{PGD}	Attack mixture probs	p_{clean}	Clean sample prob	N_{CI}, N_{IC}	Corr.→inc., inc.→corr.
N_{total}, p_c	Total samples, clean acc.	$N_{\text{orig_corr}}$	Orig. correct samples	α_p	PASR penalty weight	ASR	Attack Success Rate

TABLE II
COMPARISON OF ADVERSARIAL ATTACK METHODS

Method	Surface	Sem.	Sens.	Mom.	Ch.	G.F.	Univ.
FGSM [9]	Input	×	×	×	×	×	×
PGD [9]	Input	×	×	×	×	×	×
SemAdv [2]	Channel	×	×	×	✓	×	✓
SemBAT [8]	Input	×	×	×	×	✓	×
Ch. Attack [7]	Channel	×	×	×	✓	✓	×
ISGU [15]	Input	✓	×	×	×	×	×
SKI [6]	Semantic	✓	×	×	✓	×	×
SenSA (Proposed)	In.+Ch.	×	✓	✓	✓	×	×

Sem.: semantic loss, Sens.: sensitivity-guided, Mom.: momentum, Ch.: channel-aware, G.F.: gradient-free, Univ.: input-agnostic

training, detection-based methods, and purification-based approaches.

Adversarial training augments the training process with worst-case perturbations obtained by inner maximization [13] to improve robustness. Nan *et al.* [2] proposed SemMixed, which hardens semantic interpretation against multiple physical adversarial perturbations, including SemAdv and PGM attacks. However, adversarial training is inherently limited to the specific attack types encountered during training and may degrade performance on clean inputs.

Detection and encryption methods aim to identify or obscure adversarial perturbations. Chen *et al.* [16] introduced SemGuarder, combining ConvNeXt for efficient processing with semantic encryption and adversarial attack detection mechanisms; such pluggable modules leave the deployed encoder and decoder untouched, but their protection is bounded by the accuracy of the detector.

Purification-based methods leverage generative models to remove adversarial perturbations [17]. Unlike standard adversarial training, purification-based defenses can generalize across diverse attacks without requiring attack-specific retraining at deployment. A complementary line provides certified radii via randomized smoothing at the cost of clean accuracy and latency [18].

D. Diffusion Models for Channel Denoising and Purification

Denoising diffusion probabilistic models [19], their deterministic accelerated samplers [20], and classifier-free guidance provide the generative backbone and the conditioning mechanism that HYDRA builds on, and have recently been applied to wireless communications. Wu *et al.* [21] proposed CDDM (Channel Denoising Diffusion Models), which conditions the denoising process on received signals and channel estimates to directly reconstruct transmitted signals, adapting to various channel conditions, including AWGN and Rayleigh fading. Wu *et al.* [22] developed DENOISE, a diffusion-enhanced noise optimization framework for intelligent semantic-aware communication systems.

For adversarial purification specifically, Nie *et al.* [10] proposed DiffPure, which adds a controlled amount of noise via forward diffusion and then reconstructs clean images through reverse denoising. Ren *et al.* [17] adapted this paradigm for semantic communication (DiffPure-SC). Tian *et al.* [23] proposed DiffMoECom, which combines Mixture-of-Experts (MoE) with diffusion models for channel-adaptive semantic image communications. Their MoE-Prompt Module dynamically selects expert prompts based on channel SNR and fading conditions.

Table III compares existing defense methods with the proposed HYDRA framework. Two columns are included on which HYDRA does not lead: DiffPure, DiffPure-SC, and Randomized Smoothing operate without defense-specific training (training-free), whereas HYDRA trains its analyzer and expert heads under mixed attacks (without requiring retraining at deployment); and Randomized Smoothing provides certified guarantees in the receiver feature space, whereas empirical purification methods (including HYDRA) do not provide certified guarantees.

E. Robust Evaluation of Purification Defenses

Apparent robustness under fixed, non-adaptive attacks is often an artifact of gradient masking rather than genuine re-

Authorized licensed use limited to: Chung-ang Univ. Downloaded on November 05,2026 at 01:08:18 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3748461

TABLE III
COMPARISON OF DEFENSE METHODS

Method	Type	Agn.	MoE	Anc.	Det.	T.F.	Cert.
PGD-AT [13]	AT	×	×	×	×	×	×
SemMixed [2]	AT	×	×	×	×	×	×
SemGuarder [16]	Det.+Rem.	Part.	×	×	✓	×	×
Rand. Smoothing [18]	Cert. Smooth.	✓	×	×	×	✓	✓*
DiffPure [10]	Diff. Pur.	✓	×	×	×	✓	×
DiffPure-SC [17]	Diff. Pur.	✓	×	×	×	✓	×
ASGM [22]	Ch. Denoise	✓	×	×	×	×	×
CDDM [21]	Ch. Denoise	✓	×	×	×	×	×
DiffMoECom [23]	Gen. SC	N/A	✓	×	×	×	×
Proposed	Diff. Pur.	✓	✓	✓	✓	×	×

Agn.: no retraining at deployment, Anc.: semantic anchor, Det.: detection, AT: adv. training, Part.: partial, Diff. Pur.: purif., T.F.: training-free, Cert.: certified. *Certified in the receiver feature domain under the stated ℓ_2 smoothing guarantee.

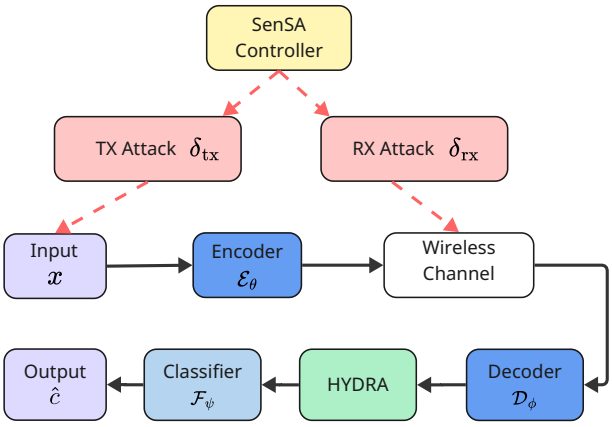


Fig. 1. Adversarial semantic communication system with SenSA and HYDRA.

silience [24], [25], [26]. In particular, stochastic sampling and truncated reverse trajectories in purification defenses obscure vulnerabilities that are readily exposed when gradients are propagated via expectation over transformation (EOT) [27]. HYDRA further compounds gradient obfuscation by introducing discrete, input-dependent routing over a stochastic reverse process. Accordingly, we evaluate HYDRA against adaptive white-box attacks differentiated via EOT alongside gradient-free black-box attacks [28] to diagnose potential gradient masking and test for false robustness.

III. System Model and Threat Model

A. Image Semantic Communication System

We consider a deep learning-based semantic communication system for image transmission, as illustrated in Fig. 1. The system consists of a transmitter (semantic encoder and channel encoder) and a receiver (channel decoder, semantic decoder, and classifier).

Let $\mathbf{x} \in \mathbb{R}^{C\times H\times W}$ denote the input image with C channels, height H, and width W. The semantic encoder $\mathcal{E}_{\boldsymbol{\theta}}(\cdot)$, parameterized by $\boldsymbol{\theta}$, extracts the essential semantic features and maps them into a sequence of complex channel symbols. In the implementation, the encoder employs a series of convolutional (Conv2D) blocks and residual blocks to capture spatial dependencies. The encoded signal is given by:

$$\mathbf{z} = \mathcal{E}_{\boldsymbol{\theta}}(\mathbf{x}), \tag{1}$$

where $\mathbf{z} \in \mathbb{C}^{d_z}$ represents the complex-valued channel input vector with d_z denoting the channel dimension, that is, the total number of complex symbols placed on the channel for one image, counted over the entire encoder output rather than per spatial position. To maximize the spectral efficiency, d_z is typically much smaller than the input dimension ($d_z \ll C \times H \times W$).

Before transmission, the signal $\mathbf{z}$ is normalized to satisfy the average per-symbol power constraint $\frac{1}{d_z}\mathbb{E}[\|\mathbf{z}\|_2^2] \le P$, yielding an aggregate average signal power $\mathbb{E}[\|\mathbf{z}\|_2^2] \le Pd_z$. The normalized signal is then transmitted over an AWGN channel. The received signal $\mathbf{y}$ can be modeled as:

$$\mathbf{y} = \mathbf{z} + \mathbf{n}, \tag{2}$$

where $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \sigma_n^2\mathbf{I}_{d_z})$ is the additive white Gaussian noise with per-symbol noise power σ_n^2 (so that $\mathbb{E}[\|\mathbf{n}\|_2^2] = \sigma_n^2 d_z$). On the linear scale the SNR is $\gamma_{\text{lin}} = P/\sigma_n^2 = \mathbb{E}[\|\mathbf{z}\|_2^2]/\mathbb{E}[\|\mathbf{n}\|_2^2]$; throughout this paper we use its decibel value $\gamma \triangleq 10\log_{10}\gamma_{\text{lin}}$ as the SNR variable, so that $\gamma_{\text{lin}} = 10^{\gamma/10}$. Every SNR quoted in the text, figures, and tables is a dB value of γ, and every expression requiring the ratio itself is written in terms of γ_{lin}.

At the receiver, the semantic decoder $\mathcal{D}_{\boldsymbol{\phi}}(\cdot)$, parameterized by $\boldsymbol{\phi}$, reconstructs the source image from the received noisy signal $\mathbf{y}$. The reconstructed image $\hat{\mathbf{x}}$ is obtained as:

$$\hat{\mathbf{x}} = \mathcal{D}_{\boldsymbol{\phi}}(\mathbf{y}). \tag{3}$$

In the proposed framework, the system performs image classification through a classifier $\mathcal{F}_{\boldsymbol{\psi}}(\cdot)$, parameterized by $\boldsymbol{\psi}$, which predicts the class label $\hat{c} = \mathcal{F}_{\boldsymbol{\psi}}(\hat{\mathbf{x}})$ from the reconstructed image. The system is trained end-to-end to minimize a composite loss combining reconstruction fidelity and classification accuracy:

$$\mathcal{L} = \mathcal{L}_{\text{recon}}(\hat{\mathbf{x}}, \mathbf{x}) + \lambda_{\text{cls}}\mathcal{L}_{\text{cls}}(\hat{c}, c), \tag{4}$$

where $\mathcal{L}_{\text{recon}} = \|\hat{\mathbf{x}} - \mathbf{x}\|_2^2$ is the mean squared error and $\mathcal{L}_{\text{cls}}$ is the cross-entropy classification loss.

B. Threat Model

A semantic communication link exposes two physically distinct attack surfaces: the data path feeding the semantic encoder, and the wireless channel carrying its output. Compromising them requires different access, so instead of a single omnipotent eavesdropper (Eve) we define one adversary class per surface, plus their union.

1) Adversary Classes: $\mathcal{A}_{tx}$ (input access) controls the data path upstream of the encoder (e.g., via malware on the sensing device, a poisoned preprocessing stage, or an intercept on the camera-to-encoder link) and may perturb the source image before encoding under an ℓ_∞ budget ϵ, bounding pixel changes to $[-\epsilon, \epsilon]$ for visual imperceptibility:

$$\mathbf{x}_{\text{adv}} = \mathbf{x} + \boldsymbol{\delta}_{\text{tx}}, \quad \text{s.t.} \quad \|\boldsymbol{\delta}_{\text{tx}}\|_\infty \le \epsilon. \tag{5}$$

$\mathcal{A}_{rx}$ (channel access) instead operates a radio front end within range of Bob (the standard smart-jammer model) and injects an energy-bounded signal into the received symbols:

$$\mathbf{y}_{\text{adv}} = \mathbf{z} + \mathbf{n} + \boldsymbol{\delta}_{\text{rx}}, \quad \text{s.t.} \quad \text{PSR}_{\text{rx}} \le \text{PSR}_{\max}, \tag{6}$$

where the receiver perturbation budget is defined as a perturbation-to-signal power ratio $\text{PSR}_{\text{rx}} \triangleq 10\log_{10}\left(\frac{\|\boldsymbol{\delta}_{\text{rx}}\|_2^2}{\mathbb{E}[\|\mathbf{z}\|_2^2]}\right)$ in the symbol domain relative to the average clean signal power $\mathbb{E}[\|\mathbf{z}\|_2^2] = Pd_z$ (with $\text{PSR}_{\max} = -15$ dB in our reference evaluation, yielding a uniform per-sample ℓ_2 bound

Authorized licensed use limited to: Chung-ang Univ. Downloaded on November 05,2026 at 01:08:18 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3748461

$\epsilon_{\text{rx}} = 10^{\text{PSR}_{\max}/20}\sqrt{Pd_z} = \sqrt{Pd_z \cdot 10^{\text{PSR}_{\max}/10}}$), representing the physical-layer constraint that channel injection remains bounded 15 dB below average signal power. Referencing the denominator to the expected signal power $\mathbb{E}[\|\mathbf{z}\|_2^2] = Pd_z$ rather than the per-realization power $\|\mathbf{z}\|_2^2$ establishes a fixed, uniform per-sample ℓ_2 power budget across all realizations, avoiding granting disparate absolute perturbation power to different inputs. Notably, neither class subsumes the other: $\mathcal{A}_{\text{tx}}$ has no radio capability, and $\mathcal{A}_{\text{rx}}$ never observes $\mathbf{x}$. In contrast, $\mathcal{A}_{joint} = \mathcal{A}_{tx} \wedge \mathcal{A}_{rx}$ encompasses both capabilities; one concrete deployment granting a single entity both is split-computing edge inference, where a compromised gateway node has both control of the pre-encoder data path to perturb sensor inputs *and* an RF emission/injection capability capable of superposing over-the-air interference $\boldsymbol{\delta}_{\text{rx}}$ at Bob, yielding input and channel access simultaneously.

In all classes, Eve is white-box with respect to the communication system: it knows $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, and $\boldsymbol{\psi}$, and is given the *true* SNR γ, representing the most favorable assumption for the attacker. Furthermore, we evaluate both defense-unaware adversaries and defense-aware adversaries that know and differentiate through the deployed defense. Single-surface evaluations assess $\mathcal{A}_{\text{tx}}$ and $\mathcal{A}_{\text{rx}}$ independently, whereas joint and per-sample mode-selected evaluations assess $\mathcal{A}_{\text{joint}}$, which subsumes both and defines the upper bound on multi-surface attack capability.

2) Attack Objective: With Δ the feasible perturbation set of the class considered (Δ_{tx}, Δ_{rx}, or $\Delta_{\text{tx}} \times \Delta_{\text{rx}}$, from (5)–(6)), the adversary seeks $\boldsymbol{\delta}^* = (\boldsymbol{\delta}_{\text{tx}}^*, \boldsymbol{\delta}_{\text{rx}}^*)$ maximizing task degradation, which is formulated as

$$\boldsymbol{\delta}^* = \underset{\boldsymbol{\delta} \in \Delta}{\operatorname{argmax}}\, \mathcal{L}_{\text{task}}(\mathcal{F}_{\boldsymbol{\psi}}(\hat{\mathbf{x}}_{\text{adv}}), c), \tag{7}$$

where $\hat{\mathbf{x}}_{\text{adv}} = \mathcal{D}_{\boldsymbol{\phi}}(\mathcal{E}_{\boldsymbol{\theta}}(\mathbf{x} + \boldsymbol{\delta}_{\text{tx}}) + \mathbf{n} + \boldsymbol{\delta}_{\text{rx}})$ is the reconstructed adversarial image, reducing to the single-surface case when the unavailable component is fixed to zero. Here $\mathcal{L}_{\text{task}}$ is the classification loss, instantiated as cross-entropy loss $\mathcal{L}_{\text{cls}}$ for untargeted attacks ($\hat{c}_{\text{adv}} \neq c_{\text{true}}$) or negative cross-entropy targeting a specified false class ($\hat{c}_{\text{adv}} = c_{\text{target}}$). While the adversary directly optimizes this task objective to induce misclassification, the resulting degradation on decoded images and latent features is evaluated as a consequence of the perturbation rather than an explicit optimization term.

3) Defense Objective: To mitigate adversarial corruption at the receiver, the defender deploys an image-space purification module $\mathcal{P}$ that restores decoded images prior to classification. To minimize downstream classification loss while bounding pixel reconstruction error and per-sample inference latency, the defense objective can be formulated as

$$\min_{\mathcal{P}} \quad \mathbb{E}_{(\mathbf{x},c)}\left[\mathcal{L}_{\text{task}}(\mathcal{F}_{\boldsymbol{\psi}}(\hat{\mathbf{x}}_{\text{pur}}), c)\right] \tag{8a}$$

$$\text{s.t.} \quad \mathbb{E}[\|\mathbf{x} - \hat{\mathbf{x}}_{\text{pur}}\|_2^2] \leq d_{\max}, \tag{8b}$$

$$T(\mathcal{P}) \leq T_{\max}, \tag{8c}$$

where $\hat{\mathbf{x}}_{\text{pur}} = \mathcal{P}(\hat{\mathbf{x}}_{\text{adv}}) = \mathcal{P}(\mathcal{D}_{\boldsymbol{\phi}}(\mathbf{y}_{\text{adv}}))$ denotes the purified image. Here, (8b) bounds the mean squared reconstruction error within a tolerance $d_{\max}$, and (8c) bounds the per-sample inference latency within $T_{\max}$. Because enforcing hard constraints via Lagrangian dual updates during inference would increase latency, we address (8b) and (8c) through the training loss in Section VI and fixed-step DDIM sampling, and evaluate both criteria via PSNR, SSIM, LPIPS, and wall-clock latency benchmarks in Section VII.

4) Adversary Architecture: As shown in Fig. 1, an $\mathcal{A}_{\text{joint}}$ adversary operates through a SenSA controller. Given the input image $\mathbf{x}$ and the channel SNR γ, the controller performs a channel-aware sensitivity analysis yielding the effective transmitter and receiver sensitivities $\mathcal{S}_{\text{tx}}^{\text{eff}}$ and $\mathcal{S}_{\text{rx}}^{\text{eff}}$, which weight coordinate-averaged loss-gradient norms by SNR-dependent channel factors. The normalized sensitivity ratio r_{tx} then selects, per sample, whether to exercise the TX capability (5), the RX capability (6), or both with coordination ratio $\rho = r_{\text{tx}}$, thereby allocating the attack domain and PGD iterations to the more sensitive path for each input-channel pair under its separate capability bound; restricting the controller to a single capability recovers $\mathcal{A}_{\text{tx}}$ or $\mathcal{A}_{\text{rx}}$. The sensitivity formulation and attack algorithms are detailed in Section IV.

IV. SenSA: Sensitivity-driven Strategic Attack

In this section, we present SenSA, which selects the attack domain and allocates PGD iterations across the transmitter and receiver under the separate perturbation bounds of Section III. Rather than committing to a fixed surface or a uniform iteration split, SenSA computes coordinate-averaged loss gradients at the encoder input and the received channel symbols, weights them by an SNR-dependent factor to determine the attack mode and iteration split, and executes the selected mode via momentum-accelerated PGD with fixed-threshold early stopping under (5) and (6).

A. Channel-Aware Sensitivity Analysis and Strategy Selection

Given an input image $\mathbf{x}$ with true label c and channel SNR γ, SenSA determines the more sensitive attack surface through a dual-pass gradient analysis followed by SNR-dependent weighting. The first pass computes the TX sensitivity, which quantifies how strongly input-level perturbations influence the classification loss through the end-to-end pipeline $\mathbf{x} \to \mathcal{E}_{\boldsymbol{\theta}} \to \text{channel} \to \mathcal{D}_{\boldsymbol{\phi}} \to \mathcal{F}_{\boldsymbol{\psi}} \to \mathcal{L}$ and is given by

$$\mathcal{S}_{\text{tx}} = \frac{1}{CHW}\left\|\frac{\partial \mathcal{L}(\mathbf{x}, c)}{\partial \mathbf{x}}\right\|_1, \tag{9}$$

where $\mathcal{L}(\cdot,\cdot)$ denotes the cross-entropy loss computed through the full encoder-channel-decoder-classifier pipeline. A high $\mathcal{S}_{\text{tx}}$ indicates that the classification loss is highly sensitive to pixel perturbations at the encoder input.

The second pass computes the RX sensitivity, which quantifies how perturbations at the received channel symbols affect classification. This requires a backward pass through the decoder-classifier pipeline with the received symbols detached from the encoder computation graph, which is expressed as

$$\mathcal{S}_{\text{rx}} = \frac{1}{d_z}\left\|\frac{\partial \mathcal{L}_{\text{rx}}(\hat{\mathbf{s}}, c)}{\partial \hat{\mathbf{s}}}\right\|_1, \tag{10}$$

where $\hat{\mathbf{s}} = \mathcal{E}_{\boldsymbol{\theta}}(\mathbf{x}) + \mathbf{n}$ is the received symbol vector after channel transmission, d_z is the channel dimension, and $\mathcal{L}_{\text{rx}}(\hat{\mathbf{s}}, c) \triangleq \mathcal{L}(\mathcal{F}_{\boldsymbol{\psi}}(\mathcal{D}_{\boldsymbol{\phi}}(\hat{\mathbf{s}})), c)$ is the classification loss evaluated through the decoder-classifier pipeline with $\hat{\mathbf{s}}$ as input. By detaching $\hat{\mathbf{s}}$ from the encoder, the gradient $\partial \mathcal{L}_{\text{rx}}/\partial \hat{\mathbf{s}}$ directly measures

Authorized licensed use limited to: Chung-ang Univ. Downloaded on November 05,2026 at 01:08:18 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3748461

the decoder-classifier sensitivity to symbol-level perturbations without encoder backpropagation. Consequently, a high $\mathcal{S}_{\text{rx}}$ indicates high sensitivity to channel-symbol perturbations.

Both quantities measure the coordinate-averaged first-order loss sensitivity of each domain. Mathematically, for a first-order Taylor expansion under norm-bounded feasible sets, an ℓ_∞ input constraint $\|\boldsymbol{\delta}_{\text{tx}}\|_\infty \le \epsilon_{\text{tx}}$ bounds the maximum first-order loss variation via the dual ℓ_1 norm by $\Delta\mathcal{L}_{\text{tx}} \le \epsilon_{\text{tx}}\|\nabla_{\mathbf{x}}\mathcal{L}\|_1 = \epsilon_{\text{tx}} d_{\text{tx}} \mathcal{S}_{\text{tx}}$ (where $d_{\text{tx}} = CHW$), whereas an ℓ_2 channel constraint $\|\boldsymbol{\delta}_{\text{rx}}\|_2 \le \epsilon_{\text{rx}}$ bounds loss variation via the self-dual ℓ_2 norm by $\Delta\mathcal{L}_{\text{rx}} \le \epsilon_{\text{rx}}\|\nabla_{\hat{\mathbf{s}}}\mathcal{L}_{\text{rx}}\|_2$. In SenSA, we adopt the coordinate-averaged absolute gradient $\mathcal{S}_{\text{rx}} = \frac{1}{d_z}\|\nabla_{\hat{\mathbf{s}}}\mathcal{L}_{\text{rx}}\|_1$ as a computationally consistent surrogate that closely tracks $\frac{1}{\sqrt{d_z}}\|\nabla_{\hat{\mathbf{s}}}\mathcal{L}_{\text{rx}}\|_2$ across latent dimensions. Because $\mathcal{S}_{\text{tx}}$ and $\mathcal{S}_{\text{rx}}$ are defined in different spaces and units ($[0,1]$ pixel space vs. power-normalized complex channel symbols), dividing by dimension normalizes coordinate count rather than establishing physical equivalence. Accordingly, we weight the two coordinate-averaged gradients by an SNR-dependent factor to compare their relative impact across varying SNR regimes.

Although $\mathcal{S}_{\text{tx}}$ and $\mathcal{S}_{\text{rx}}$ measure the unweighted gradient sensitivity of each attack surface, their post-channel impact depends on the wireless SNR. A TX perturbation must traverse the AWGN channel before reaching the receiver: at high SNR, it arrives largely intact, whereas at low SNR, it is attenuated relative to channel noise. Conversely, RX perturbations are injected directly at the receiver, where noisy received symbols at low SNR lie closer to classification decision boundaries. To model this SNR-dependent scaling, we define the channel weighting factor as

$$\alpha(\gamma) = \frac{\gamma_{\text{lin}}}{1+\gamma_{\text{lin}}}, \quad \gamma_{\text{lin}} = 10^{\gamma/10}, \tag{11}$$

where, following the convention of Section III, γ is the SNR in dB and γ_{lin} is the corresponding linear ratio. The factor maps the SNR to the interval $(0,1)$ and equals the signal-power fraction of the received power under AWGN, $P/(P+\sigma_n^2)$. Under a first-order expansion of the loss (A1), a scalar LMMSE decoder front end $\hat{\mathbf{z}} = \frac{P}{P+\sigma_n^2}\mathbf{y} = \alpha(\gamma)\mathbf{y}$ (A2), and perturbations normalized to the respective signal and noise powers (A3), perturbations from the two surfaces reach the LMMSE estimator output with post-equalization power gains proportional to $\alpha(\gamma)$ and $1-\alpha(\gamma)$. Note that this noise-referenced comparison is used to derive the SNR weighting function, whereas attack execution enforces the fixed ℓ_2 symbol budget ϵ_{rx} of (6). Consequently, under assumptions (A1)–(A3), this first-order LMMSE analysis motivates the functional form of the SNR weighting and predicts that effective TX sensitivity increases and effective RX sensitivity decreases as SNR rises. Accordingly, the SNR-weighted effective sensitivities can be expressed as

$$\mathcal{S}_{\text{tx}}^{\text{eff}} = \mathcal{S}_{\text{tx}} \cdot \alpha(\gamma), \quad \mathcal{S}_{\text{rx}}^{\text{eff}} = \mathcal{S}_{\text{rx}} \cdot \big(1-\alpha(\gamma)\big), \tag{12}$$

and the normalized sensitivity ratios are calculated as

$$r_{\text{tx}} = \frac{\mathcal{S}_{\text{tx}}^{\text{eff}}}{\mathcal{S}_{\text{tx}}^{\text{eff}} + \mathcal{S}_{\text{rx}}^{\text{eff}} + \nu_0}, \quad r_{\text{rx}} = \frac{\mathcal{S}_{\text{rx}}^{\text{eff}}}{\mathcal{S}_{\text{tx}}^{\text{eff}} + \mathcal{S}_{\text{rx}}^{\text{eff}} + \nu_0}, \tag{13}$$

where ν_0 is a small constant for numerical stability. Based on the normalized sensitivity ratios, SenSA selects the attack mode and coordination ratio according to

$$(\text{strategy},\ \rho) = \begin{cases} (\text{TX},\ 1.0) & \text{if } r_{\text{tx}} > \tau, \\ (\text{RX},\ 0.0) & \text{if } r_{\text{rx}} > \tau, \\ (\text{Joint},\ r_{\text{tx}}) & \text{otherwise}, \end{cases} \tag{14}$$

where τ is the mode-selection threshold. The coordination ratio ρ governs the allocation of PGD iterations between TX and RX under their separate capability bounds: $\rho = 1$ assigns all iterations to the TX surface, $\rho = 0$ assigns all iterations to the RX surface, and intermediate values split the iteration counts T_{tx}^* and T_{rx}^* proportionally via (21).

B. Attack Execution

Both TX and RX attacks employ momentum-accelerated PGD with fixed-threshold early stopping. Specifically, the early stopping criterion is defined as

$$\text{EarlyStop}(\hat{c}, c, p_{\text{true}}) \triangleq (\hat{c} \neq c) \vee (p_{\text{true}} < \tau_{\text{stop}}), \tag{15}$$

which terminates the PGD loop when misclassification occurs or the true-label confidence p_{true} falls below τ_{stop}.

At the transmitter side, the attack perturbs the source image via MI-FGSM [29]. The perturbation is initialized as $\boldsymbol{\delta}_{\text{tx}}^{(0)} \sim \mathcal{U}(-0.1\epsilon_{\text{tx}}, 0.1\epsilon_{\text{tx}})$ with momentum $\mathbf{v}^{(0)} = \mathbf{0}$, and is updated for $i = 1, \ldots, T_{\text{tx}}$ as

$$\begin{aligned} \mathbf{g}^{(i)} &= \nabla_{\boldsymbol{\delta}}\mathcal{L}(\mathbf{x} + \boldsymbol{\delta}_{\text{tx}}^{(i-1)}, c), \\ \mathbf{v}^{(i)} &= \mu \cdot \mathbf{v}^{(i-1)} + \mathbf{g}^{(i)}, \\ \boldsymbol{\delta}_{\text{tx}}^{(i)} &= \text{Proj}_{\epsilon_{\text{tx}}}^{(\infty)}\left(\boldsymbol{\delta}_{\text{tx}}^{(i-1)} + \eta_{\text{tx}} \cdot \text{sign}(\mathbf{v}^{(i)})\right), \end{aligned} \tag{16}$$

where μ is the momentum coefficient, η_{tx} is the step size, and $\text{Proj}_{\epsilon}^{(\infty)}(\boldsymbol{\delta}) \triangleq \text{clip}(\boldsymbol{\delta}, -\epsilon, \epsilon)$ projects onto the ℓ_∞ ball. The TX loop terminates early whenever EarlyStop evaluates to true.

At the receiver side, the attack perturbs the received channel symbols $\hat{\mathbf{s}} = \mathcal{E}_{\boldsymbol{\theta}}(\mathbf{x}) + \mathbf{n}$ after channel transmission, where $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \sigma_n^2 \mathbf{I}_{d_z})$. For $i = 1, \ldots, T_{\text{rx}}$, the update is given by

$$\mathbf{g}^{(i)} = \nabla_{\boldsymbol{\delta}}\mathcal{L}_{\text{rx}}(\hat{\mathbf{s}} + \boldsymbol{\delta}_{\text{rx}}^{(i-1)}, c), \tag{17}$$

$$\mathbf{v}^{(i)} = \mu \cdot \mathbf{v}^{(i-1)} + \mathbf{g}^{(i)}, \tag{18}$$

$$\boldsymbol{\delta}_{\text{rx}}^{(i)} = \text{Proj}_{\epsilon_{\text{rx}}}^{(2)}\left(\boldsymbol{\delta}_{\text{rx}}^{(i-1)} + \eta_{\text{rx}} \cdot \frac{\mathbf{v}^{(i)}}{\|\mathbf{v}^{(i)}\|_2}\right), \tag{19}$$

where $\mathcal{L}_{\text{rx}}(\hat{\mathbf{s}}, c) \triangleq \mathcal{L}(\mathcal{F}_{\boldsymbol{\psi}}(\mathcal{D}_{\boldsymbol{\phi}}(\hat{\mathbf{s}})), c)$ and η_{rx} is the step size. Unlike the TX attack, which uses ℓ_∞ projection, the RX perturbation is bounded via ℓ_2 projection, which is expressed as

$$\text{Proj}_{\epsilon_{\text{rx}}}^{(2)}(\boldsymbol{\delta}) \triangleq \boldsymbol{\delta} \cdot \min\left(1, \frac{\epsilon_{\text{rx}}}{\|\boldsymbol{\delta}\|_2}\right), \tag{20}$$

enforcing the uniform per-sample capability bound $\|\boldsymbol{\delta}_{\text{rx}}\|_2 \le \epsilon_{\text{rx}} = 10^{\text{PSR}_{\max}/20}\sqrt{P d_z}$ of (6). The RX loop also terminates via EarlyStop.

1) Joint Attack with Coordinated TX-RX Perturbation: In the joint attack mode, the coordination ratio ρ splits the iteration budget between the transmitter and receiver surfaces, which can be expressed as

$$T_{\text{tx}}^* = \max(5, \lfloor \rho \cdot T_{\text{tx}} \rfloor), \quad T_{\text{rx}}^* = \max(5, \lfloor (1-\rho) \cdot T_{\text{rx}} \rfloor). \tag{21}$$

Specifically, SenSA executes the TX and RX phases sequentially under their respective perturbation bounds. First, the TX phase applies the PGD update in (16) for T_{tx}^* iterations under the ℓ_∞ bound ϵ to maximize $\mathcal{L}_{\text{task}}$ in (7). If the TX

Authorized licensed use limited to: Chung-ang Univ. Downloaded on November 05,2026 at 01:08:18 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3748461

Algorithm 1: SenSA: Sensitivity-driven Strategic Attack

Require: Image $\mathbf{x}$, label c, model $\{\mathcal{E}_{\boldsymbol{\theta}}, \mathcal{D}_{\boldsymbol{\phi}}, \mathcal{F}_{\boldsymbol{\psi}}\}$, SNR γ (dB)
Ensure : Adversarial output $\mathbf{x}_{\text{adv}}$ and/or $\mathbf{y}_{\text{adv}}$

$\mathbf{s} \leftarrow \mathcal{E}_{\boldsymbol{\theta}}(\mathbf{x})$; $\hat{\mathbf{s}} \leftarrow \mathbf{s} + \mathbf{n}$; $\hat{\mathbf{x}} \leftarrow \mathcal{D}_{\boldsymbol{\phi}}(\hat{\mathbf{s}})$; $\mathcal{L} \leftarrow \mathcal{L}_{\text{CE}}(\mathcal{F}_{\boldsymbol{\psi}}(\hat{\mathbf{x}}), c)$;
Compute $\mathcal{S}_{\text{tx}} \leftarrow$ (9) via backprop w.r.t. $\mathbf{x}$;
Detach $\hat{\mathbf{s}}$; compute $\mathcal{S}_{\text{rx}} \leftarrow$ (10) via backprop w.r.t. $\hat{\mathbf{s}}$;
Compute $\alpha(\gamma) \leftarrow$ (11); $\mathcal{S}_{\text{tx}}^{\text{eff}}, \mathcal{S}_{\text{rx}}^{\text{eff}} \leftarrow$ (12);
Compute $r_{\text{tx}}, r_{\text{rx}} \leftarrow$ (13);
Compute $(\text{strategy}, \rho) \leftarrow$ (14);
if *strategy = "TX"* **then**
  $\boldsymbol{\delta}_{\text{tx}} \leftarrow \text{TxAttackPGD}(\mathbf{x}, c, T_{\text{tx}})$ with early stop per (15);
  $\mathbf{x}_{\text{adv}} \leftarrow \mathbf{x} + \boldsymbol{\delta}_{\text{tx}}$;
  return $\mathbf{x}_{\text{adv}}$;
else if *strategy = "RX"* **then**
  $\mathbf{y} \leftarrow \mathcal{E}_{\boldsymbol{\theta}}(\mathbf{x}) + \mathbf{n}$;
  Compute $\boldsymbol{\delta}_{\text{rx}} \leftarrow$ (17)–(19) with early stop per (15);
  $\mathbf{y}_{\text{adv}} \leftarrow \mathbf{y} + \boldsymbol{\delta}_{\text{rx}}$;
  return $\mathbf{y}_{\text{adv}}$;
else
  Compute $T_{\text{tx}}^*, T_{\text{rx}}^* \leftarrow$ (21);
  $\boldsymbol{\delta}_{\text{tx}} \leftarrow \text{TxAttackPGD}(\mathbf{x}, c, T_{\text{tx}}^*)$ with early stop per (15);
  $\mathbf{x}_{\text{adv}} \leftarrow \mathbf{x} + \boldsymbol{\delta}_{\text{tx}}$;
  if $\hat{c}_{adv} \neq c$ **then**
    return $\mathbf{x}_{\text{adv}}$;
  end
  $\mathbf{y} \leftarrow \mathcal{E}_{\boldsymbol{\theta}}(\mathbf{x}_{\text{adv}}) + \mathbf{n}$;
  $\boldsymbol{\delta}_{\text{rx}} \leftarrow$ (17)–(19) with T_{rx}^* steps;
  $\mathbf{y}_{\text{adv}} \leftarrow \mathbf{y} + \boldsymbol{\delta}_{\text{rx}}$;
end
return $\mathbf{x}_{\text{adv}}$, $\mathbf{y}_{\text{adv}}$;

perturbation alone achieves misclassification ($\hat{c}_{\text{adv}} \neq c_{\text{true}}$), the algorithm terminates immediately without invoking the RX phase. Otherwise, the transmitted symbols from the perturbed image pass through the channel as $\mathbf{y} = \mathcal{E}_{\boldsymbol{\theta}}(\mathbf{x}_{\text{adv}}) + \mathbf{n}$, and the RX phase applies the symbol-domain PGD update in (17)–(19) for T_{rx}^* iterations under the PSR_{max} bound ϵ_{rx}, producing $\mathbf{y}_{\text{adv}} = \mathbf{y} + \boldsymbol{\delta}_{\text{rx}}$. Here, the TX and RX updates ((16) and (19)) perform steepest ascent on the cross-entropy losses $\mathcal{L}$ and $\mathcal{L}_{\text{rx}}$ under their dual norms ($\text{sign}(\mathbf{v})$ for the ℓ_∞ pixel box and normalized $\mathbf{v}/\|\mathbf{v}\|_2$ for the ℓ_2 symbol ball), directly maximizing (7). The complete SenSA procedure is summarized in Algorithm 1.

C. Complexity Analysis

Specifically, strategy selection requires two backward passes over the single forward pass of Algorithm 1 (one for $\mathcal{S}_{\text{tx}}$ through the full pipeline, and one for $\mathcal{S}_{\text{rx}}$ through the decoder–classifier with $\hat{\mathbf{s}}$ detached), followed by the scalar decision (11)–(14), so no search over attack surfaces is performed. Execution then costs $\mathcal{O}(T_{\text{tx}} + T_{\text{rx}})$ forward–backward passes, at most $\max(T_{\text{tx}}, T_{\text{rx}}) = 30$ in the single-surface modes and $T_{\text{tx}} + T_{\text{rx}} = 40$ jointly under Table V, and fewer whenever the early-stopping test (15) fires or the TX phase alone succeeds; RX passes traverse only the decoder–classifier.

The resulting 42.5 ms per sample (Table VI) is an adversary-side cost: the latency the deployed link must meet is the defense's (9.46 ms for HYDRA). It bounds the rate at which Eve can attack a live stream while preserving the threat, since an adversary with input access can buffer frames and attack a subset. Furthermore, it represents a flexible operating point: the caps enter only through (21), so adjusting the iteration caps trades attack strength against latency along a controllable curve. Specifically, across our experiments under the default configuration of Table V ($T_{\text{tx}} = 10, T_{\text{rx}} = 30$), early stopping terminates execution at an average of 18 iterations, maintaining high attack efficacy with an average latency of 42.5 ms per sample (only $1.3\times$ that of single-surface SemAdv at 32.74 ms, and $3.3\times$ faster than SemBAT at 140.71 ms).

V. HYDRA: HYBRID DIFFUSION-ROUTED ADAPTIVE DEFENSE

HYDRA purifies corrupted images at the receiver through reverse diffusion denoising. Specifically, it estimates the corruption level of each decoded image and sets the forward-diffusion starting timestep accordingly. As a result, it scales the added Gaussian noise variance to the estimated perturbation magnitude rather than applying a fixed diffusion schedule. Fig. 2 illustrates the four stages of the HYDRA pipeline.

Given the decoded adversarial image $\hat{\mathbf{x}}_{\text{adv}} = \mathcal{D}_{\boldsymbol{\phi}}(\mathbf{y}_{\text{adv}})$ at the receiver, the perturbation analyzer G_ξ produces routing logits $\mathbf{q} \in \mathbb{R}^{N_e}$, a corruption estimate $s \in [0, 1]$, and a predicted class $\hat{c}_a$ in a single forward pass. The corruption estimate s is mapped by (23) to a noise-scaling factor β_{eff} and then to the forward-diffusion starting timestep t_{start} via (24), at which forward noising produces $\mathbf{x}_{t_{\text{start}}}$ (25). The DDIM reverse process then denoises $\mathbf{x}_{t_{\text{start}}}$ to $\hat{\mathbf{x}}_{\text{pur}}$ by routing U-Net features through a top-k MoE gate to specialized expert heads and interpolating between conditional and unconditional noise predictions via classifier-free guidance conditioned on $\hat{c}_a$ (28). Finally, class-prototype regularization blends the denoised output toward the EMA class-mean prototype $\mathbf{a}_{\hat{c}_a}$ (33), and the loop repeats over R rounds with geometrically decaying noise scaling (34). The subsections below detail each stage, and Algorithm 2 summarizes the complete procedure.

A. Perturbation Analyzer

To estimate the corruption level and class condition of each decoded image, we employ a CNN-based perturbation analyzer that extracts a shared feature vector $\mathbf{f} = \text{CNN}(\hat{\mathbf{x}}_{\text{adv}})$ and feeds four multilayer perceptron (MLP) heads, which can be expressed as

$$\begin{aligned}(\mathbf{q}, p, s, \hat{c}_a) &= G_\xi(\hat{\mathbf{x}}_{\text{adv}}) \\ &\triangleq \big(\text{MLP}_{route}(\mathbf{f}),\, \sigma(\text{MLP}_{\text{det}}(\mathbf{f})), \\ &\quad \sigma(\text{MLP}_{sev}(\mathbf{f})),\, \text{argmax}\, \text{MLP}_{\text{cls}}(\mathbf{f})\big),\end{aligned} \tag{22}$$

where MLP_{route}, MLP_{det}, MLP_{sev}, and MLP_{cls} denote the routing, detection, corruption-level, and classification heads, respectively. The four outputs are the expert routing logits $\mathbf{q} \in \mathbb{R}^{N_e}$, the binary attack detection probability $p \in [0, 1]$, the corruption-level estimate $s \in [0, 1]$, and the predicted class label $\hat{c}_a \in \{1, \ldots, K\}$.

During inference, s sets the forward-noising scaling factor in (23), $\mathbf{q}$ determines the top-k expert weights in the MoE head, and $\hat{c}_a$ provides the class label for CFG conditioning and prototype blending, whereas p serves as an auxiliary binary detection output supervised during training.

Authorized licensed use limited to: Chung-ang Univ. Downloaded on November 05,2026 at 01:08:18 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3748461

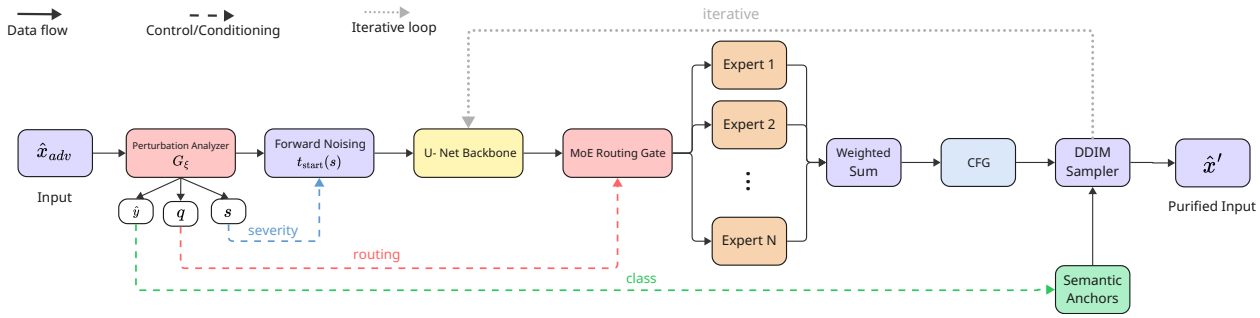


Fig. 2. HYDRA defense architecture: MoE-guided diffusion purification with semantic anchors.

B. Forward-Diffusion Timestep Selection

To scale the forward-diffusion noise variance according to the estimated corruption level, the analyzer output s linearly interpolates the noise-scaling factor between β_{base} and $\beta_{\max}$, which is calculated as

$$\beta_{\text{eff}} = \beta_{\text{base}} + s \cdot (\beta_{\max} - \beta_{\text{base}}), \tag{23}$$

where β_{base} and $\beta_{\max}$ denote the minimum and maximum scaling factors. Using β_{eff}, the starting timestep t_{start} is selected by matching the forward-diffusion noise-to-signal ratio to the scaled channel noise variance, which is given by

$$t_{\text{start}} = \underset{t}{\operatorname{argmin}} \left| \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t} - \beta_{\text{eff}} \cdot \sigma_n^2 \right|, \tag{24}$$

where $\sigma_n^2 = P/\gamma_{\text{lin}} = 10^{-\gamma/10}$ is the channel noise power under unit transmit power ($P = 1$), with γ in dB and γ_{lin} denoting the linear SNR of Section III. Given t_{start}, the decoded image is noised to the corresponding diffusion timestep as

$$\mathbf{x}_{t_{\text{start}}} = \sqrt{\bar{\alpha}_{t_{\text{start}}}} \cdot (2\hat{\mathbf{x}} - 1) + \sqrt{1 - \bar{\alpha}_{t_{\text{start}}}} \cdot \boldsymbol{\epsilon}, \tag{25}$$

where the input is scaled to $[-1, 1]$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Consequently, a larger corruption estimate s or lower channel SNR γ yields a larger starting timestep t_{start} to overwrite strong perturbations, whereas a smaller s and higher γ select a smaller t_{start} to preserve clean image details.

C. MoE-Enhanced Denoising with CFG

The denoising module consists of a class-conditioned U-Net backbone followed by an MoE noise-prediction head and CFG-guided DDIM sampling. Given the noisy image $\mathbf{x}_t$ at timestep t with class condition $c_{\text{cond}} \in \{\hat{c}_a, \varnothing\}$, the U-Net extracts feature maps, which can be expressed as

$$\mathbf{h} = \text{U-Net}(\mathbf{x}_t, \mathbf{e}_t, \mathbf{e}_c), \tag{26}$$

where $\mathbf{e}_t$ denotes the sinusoidal timestep embedding and $\mathbf{e}_c$ is a learnable class embedding (including a null token $\varnothing$ for unconditional prediction). Each residual block is conditioned on both $\mathbf{e}_t$ and $\mathbf{e}_c$ via affine scale-and-shift modulation.

We route these backbone features through N_e convolutional expert heads, each comprising two convolutional layers that output a noise estimate $\hat{\boldsymbol{\epsilon}}^{(i)}$. The routing logits $\mathbf{q}$ from the analyzer are passed through a residual refinement layer $\mathbf{q}' = \text{MLP}_{refine}(\mathbf{q}) + \mathbf{q}$ to select the top-k experts according to

$$\begin{aligned} \mathcal{I} &= \text{TopK}(\mathbf{q}', k), \\ \omega_i &= \frac{\exp(q'_i)}{\sum_{j\in\mathcal{I}} \exp(q'_j)}, \quad \forall i \in \mathcal{I}, \end{aligned} \tag{27}$$

and the aggregated noise prediction is given by $\hat{\boldsymbol{\epsilon}}_{MoE} = \sum_{i\in\mathcal{I}} \omega_i \cdot \hat{\boldsymbol{\epsilon}}^{(i)}$, which is summed with a linear projection of the U-Net backbone features.

To incorporate class information into the MoE denoiser during reverse sampling, we apply classifier-free guidance (CFG). At each DDIM step, both a conditional prediction $\hat{\boldsymbol{\epsilon}}_{cond} = \boldsymbol{\epsilon}_{\boldsymbol{\omega}}(\mathbf{x}_t, t, \hat{c}_a)$ and an unconditional prediction $\hat{\boldsymbol{\epsilon}}_{uncond} = \boldsymbol{\epsilon}_{\boldsymbol{\omega}}(\mathbf{x}_t, t, \varnothing)$ are computed and interpolated as

$$\hat{\boldsymbol{\epsilon}}_{\text{CFG}} = \hat{\boldsymbol{\epsilon}}_{uncond} + w \cdot (\hat{\boldsymbol{\epsilon}}_{cond} - \hat{\boldsymbol{\epsilon}}_{uncond}), \tag{28}$$

where w is the guidance scale. However, if the analyzer predicts an erroneous class $\hat{c}_a$ under a strong perturbation, a fixed scale w biases the reverse step toward the wrong class. To prevent this, we gate the guidance scale by the analyzer's maximum softmax probability, which is given by

$$w_{\text{eff}} = w \cdot \sigma\left(\frac{p_{\hat{c}_a} - \tau_c}{\tau_T}\right), \tag{29}$$

where $p_{\hat{c}_a} = \max(\text{softmax}(\text{MLP}_{\text{cls}}(\mathbf{f})))$ is the predicted class confidence from the analyzer, τ_c is the confidence threshold, τ_T is the sigmoid temperature, and $\sigma(\cdot)$ is the logistic sigmoid function. When $p_{\hat{c}_a} \gg \tau_c$, $w_{\text{eff}} \approx w$ and full class-conditional guidance is applied; when $p_{\hat{c}_a} \ll \tau_c$, $w_{\text{eff}} \approx 0$ and (28) reduces to the unconditional prediction $\hat{\boldsymbol{\epsilon}}_{uncond}$ as in standard DiffPure [10]. Accordingly, w in (28) is replaced by w_{eff} during inference.

The DDIM reverse process then computes the clean image estimate and steps to the next timestep t', which is expressed as

$$\hat{\mathbf{x}}_0 = \text{clip}\left(\frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\hat{\boldsymbol{\epsilon}}_{\text{CFG}}}{\sqrt{\bar{\alpha}_t}},\ -1,\ 1\right), \tag{30}$$

$$\mathbf{x}_{t'} = \sqrt{\bar{\alpha}_{t'}}\hat{\mathbf{x}}_0 + \sqrt{1-\bar{\alpha}_{t'}}\,\hat{\boldsymbol{\epsilon}}_{\text{CFG}}, \tag{31}$$

using N_{DDIM} subsampled timesteps.

D. Class-Prototype Regularization and Multiround Purification

To regularize the pixel-space distribution of purified images toward their predicted class, we maintain EMA class-mean image prototypes (termed semantic anchors) $\mathbf{a}_j \in \mathbb{R}^{3\times H_a \times W_a}$ for each class j, updated during training on clean images as

$$\mathbf{a}_j \leftarrow \zeta \mathbf{a}_j + (1-\zeta) \cdot \text{Resize}(\mathbf{x}_{\text{clean}}), \tag{32}$$

with momentum ζ and spatial resolution (H_a, W_a). After each reverse DDIM pass, the purified image $\hat{\mathbf{x}}_{\text{pur}}$ is linearly blended

Authorized licensed use limited to: Chung-ang Univ. Downloaded on November 05,2026 at 01:08:18 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3748461

Algorithm 2: HYDRA Purification

Require: Decoded adversarial image $\hat{\mathbf{x}}_{\text{adv}}$, SNR γ (dB), num_rounds R
Ensure : Purified image $\hat{\mathbf{x}}_{\text{pur}}$

$\mathbf{q}, p, s, \hat{c}_a \leftarrow G_\xi(\hat{\mathbf{x}}_{\text{adv}})$;
$\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}}_{\text{adv}}$;
for $r = 0, \ldots, R-1$ **do**
  Compute $\beta_{\text{eff}}^{(r)} \leftarrow$ (34) using (23);
  Compute $t_{\text{start}} \leftarrow$ (24);
  Compute $\mathbf{x}_t \leftarrow$ (25);
  for $t \in \{t_{start}, \ldots, 0\}$ *(DDIM schedule)* **do**
    Compute $w_{\text{eff}} \leftarrow$ (29); $\hat{\boldsymbol{\epsilon}}_{\text{CFG}} \leftarrow$ (28) with w_{eff};
    Compute $\mathbf{x}_{t'} \leftarrow$ (30)–(31);
  end
  $\hat{\mathbf{x}}_{\text{pur}} \leftarrow (\mathbf{x}_0 + 1)/2$;
  $\hat{\mathbf{x}} \leftarrow$ (33) using $\hat{\mathbf{x}}_{\text{pur}}$;
end
return $\hat{\mathbf{x}}_{\text{pur}} = \text{clip}(\hat{\mathbf{x}}, 0, 1)$;

with the upsampled prototype of the predicted class $\hat{c}_a$, which is given by

$$\hat{\mathbf{x}}' = \hat{\mathbf{x}}_{\text{pur}} + \lambda_a \cdot \kappa^r \cdot (\text{Resize}_{H\times W}(\mathbf{a}_{\hat{c}_a}) - \hat{\mathbf{x}}_{\text{pur}}), \tag{33}$$

where $\text{Resize}_{H\times W}(\cdot)$ bilinearly upsamples $\mathbf{a}_{\hat{c}_a}$ from (H_a, W_a) to (H, W), λ_a is the blending weight, κ is the per-round decay factor, and r is the purification round index. Storing $\mathbf{a}_j$ at reduced spatial dimensions (H_a, W_a) lowers memory overhead while providing a low-frequency class-mean reference, whereas high-frequency spatial details are reconstructed by the U-Net reverse process. Whereas CFG conditions noise prediction at each intermediate DDIM step, prototype blending in (33) applies a post-denoising convex combination in pixel space. Finally, to remove residual perturbations across R sequential rounds without over-smoothing, the noise-scaling factor decays geometrically with the round index r, which is expressed as

$$\beta_{\text{eff}}^{(r)} = \beta_{\text{eff}} \cdot \kappa^r, \quad r = 0, 1, \ldots, R-1, \tag{34}$$

so that the initial round applies the largest starting timestep t_{start} and subsequent rounds initialize forward diffusion at progressively smaller timesteps. Algorithm 2 summarizes the complete HYDRA inference procedure.

VI. Training Strategy and Evaluation Metrics

A. Two-Stage Decoupled Training

Training the defense must balance noise removal, feature preservation, and classification accuracy across unseen attack variations without destabilizing the transceiver backbone; we therefore decouple training into two stages. Stage 1 trains the semantic communication system $\{\mathcal{E}_{\boldsymbol{\theta}}, \mathcal{D}_{\boldsymbol{\phi}}, \mathcal{F}_{\boldsymbol{\psi}}\}$ end-to-end on clean images with the joint reconstruction and classification loss of Section III; once converged, $\{\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\psi}\}$ are frozen, fixing the gradient signal from which training-time attacks are generated.

Stage 2 trains the HYDRA defense $D_{\boldsymbol{\omega}}$ against adversarial examples generated on the fly with the backbone fixed. To expose the purifier to both single-surface and joint perturbations, each training batch samples inputs according to

$$(\mathbf{x}_{\text{adv}}, \boldsymbol{\delta}_{\text{rx}}) = \begin{cases} \text{SenSA}(\mathbf{x}, c) & \text{w.p. } p_{\text{SenSA}} \\ (\text{PGD}(\mathbf{x}, c), \mathbf{0}) & \text{w.p. } p_{\text{PGD}} \\ (\mathbf{x}, \mathbf{0}) & \text{w.p. } p_{\text{clean}} \end{cases} \tag{35}$$

The transmitted symbols pass through the channel as $\mathbf{y}_{\text{adv}} = \mathcal{E}_{\boldsymbol{\theta}}(\mathbf{x}_{\text{adv}}) + \mathbf{n} + \boldsymbol{\delta}_{\text{rx}}$ and are reconstructed by the frozen decoder into the corrupted image $\hat{\mathbf{x}}_{\text{adv}} = \mathcal{D}_{\boldsymbol{\phi}}(\mathbf{y}_{\text{adv}})$, which feeds HYDRA. The clean fraction p_{clean} prevents the purifier from degrading reconstruction fidelity on unperturbed inputs (with the fixed value listed in Table V). Because Stage 2 optimizes $\mathcal{L}_{\text{total}}$ against a frozen transceiver, it avoids the alternating-minimax instability of joint adversarial training.

B. Composite Loss Function

The defense is trained to minimize a composite loss combining five terms, which is expressed as

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{diff}} + \lambda_{\text{aux}}\mathcal{L}_{\text{aux}} + \lambda_{\text{anc}}\mathcal{L}_{\text{anc}} + \lambda_{\text{det}}\mathcal{L}_{\text{det}} + \lambda_{\text{lb}}\mathcal{L}_{\text{lb}}, \tag{36}$$

where λ_{aux}, λ_{anc}, λ_{det}, and λ_{lb} weight the auxiliary classification, prototype alignment, analyzer supervision, and MoE load-balancing losses, respectively. The primary term $\mathcal{L}_{\text{diff}}$ is the denoising score-matching loss given by

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{t, \mathbf{x}_0, \boldsymbol{\epsilon}} \left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\omega}}(\mathbf{x}_t, t, c_{\text{cond}}, \mathbf{q})\|_2^2 \right], \tag{37}$$

where $t \sim \mathcal{U}(1, T_{\text{diff}})$, $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$, and $\mathbf{q}$ denotes the routing logits from the perturbation analyzer. This term trains the U-Net backbone and MoE experts to predict the injected Gaussian noise conditioned on the timestep, class embedding, and routing logits.

To preserve class-discriminative features in the purified output, the auxiliary classification loss $\mathcal{L}_{\text{aux}} = \mathbb{E}_{(\mathbf{x},c)}[-\log p_{\text{aux}}(c|\hat{\mathbf{x}}_{\text{pur}})]$ is evaluated through a classification head p_{aux} trained jointly with the denoiser. In addition, the prototype alignment loss $\mathcal{L}_{\text{anc}} = \mathbb{E}_{(\mathbf{x},c)}[\|\text{Resize}(\hat{\mathbf{x}}_{\text{pur}}) - \mathbf{a}_c\|_2^2]$ penalizes the squared Euclidean distance between downsampled purified outputs and the EMA class prototypes updated via (32). Because the prototypes $\mathbf{a}_c$ are initialized to zero, $\mathcal{L}_{\text{anc}}$ is activated after a warm-up period of N_{warmup} iterations.

The analyzer supervision loss $\mathcal{L}_{\text{det}} = \text{BCE}(p_{\text{adv}}, b_{\text{adv}}) + \text{ReLU}(m_{\text{sev}} - (\bar{s}_{\text{adv}} - \bar{s}_{\text{clean}})) + \mathcal{L}_{\text{CE}}(\text{MLP}_{\text{cls}}(\mathbf{f}), c)$ trains the binary detector p_{adv} against the attack indicator $b_{\text{adv}} \in \{0, 1\}$, enforces a ranking margin m_{sev} between the mean corruption estimates of adversarial ($\bar{s}_{\text{adv}}$) and clean ($\bar{s}_{\text{clean}}$) samples, and supervises the analyzer class head $\text{MLP}_{\text{cls}}(\mathbf{f})$. Finally, the load-balancing loss $\mathcal{L}_{\text{lb}} = N_e \cdot \sum_{i=1}^{N_e} f_i \cdot P_i$ prevents expert collapse by penalizing uneven expert dispatch, where f_i and P_i denote the dispatch fraction and mean routing probability for expert i, respectively. Algorithm 3 summarizes the two-stage training and inference procedures, and Table V lists all hyperparameters.

C. Penalized Attack Success Rate (PASR)

The standard Attack Success Rate (ASR) measures the fraction of originally correct samples that become misclassified under attack, defined as $\text{ASR} = N_{\text{CI}}/N_{\text{orig_corr}}$, where N_{CI} counts correct $\rightarrow$ incorrect transitions. However, ASR omits incorrect $\rightarrow$ correct transitions (N_{IC}), which occur

Authorized licensed use limited to: Chung-ang Univ. Downloaded on November 05,2026 at 01:08:18 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3748461

Algorithm 3: Two-Stage Training and Deployment for HYDRA

Input : Dataset $\mathcal{X}$, Pre-trained SC model $\{\mathcal{E}_{\boldsymbol{\theta}}, \mathcal{D}_{\boldsymbol{\phi}}, \mathcal{F}_{\boldsymbol{\psi}}\}$, SNR γ
Output: Trained defense $D_{\boldsymbol{\omega}^*}$, Semantic anchors $\{\mathbf{a}_j\}_{j=1}^{K}$

Training Phase:
  Initialize $\boldsymbol{\omega}$, anchors $\mathbf{a}_j \leftarrow \mathbf{0}$;
  Freeze $\{\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\psi}\}$;
  for *epoch* $= 1$ **to** N_{epochs} **do**
    for *batch* $(\mathbf{x}, c) \sim \mathcal{X}$ **do**
      Generate $(\mathbf{x}_{\text{adv}}, \boldsymbol{\delta}_{\text{rx}}) \leftarrow$ (35);
      $\mathbf{y}_{\text{adv}} \leftarrow \mathcal{E}_{\boldsymbol{\theta}}(\mathbf{x}_{\text{adv}}) + \mathbf{n} + \boldsymbol{\delta}_{\text{rx}}$;
      $\hat{\mathbf{x}}_{\text{adv}} \leftarrow \mathcal{D}_{\boldsymbol{\phi}}(\mathbf{y}_{\text{adv}})$;
      Update anchor $\mathbf{a}_c \leftarrow$ (32);
      $\mathbf{q}, p_{\text{adv}}, s, \hat{c}_a \leftarrow G_\xi(\hat{\mathbf{x}}_{\text{adv}})$;
      $\hat{\mathbf{x}}_{\text{pur}} \leftarrow D_{\boldsymbol{\omega}}(\hat{\mathbf{x}}_{\text{adv}}, \text{SNR}, \mathbf{q})$;
      Compute $\mathcal{L}_{\text{total}} \leftarrow$ (36) (with $\lambda_{\text{anc}} = 0$ for iter $\leq N_{\text{warmup}}$);
      $\boldsymbol{\omega} \leftarrow \boldsymbol{\omega} - \eta \nabla_{\boldsymbol{\omega}} \mathcal{L}_{\text{total}}$;
    end
  end
Inference Phase:
  Given: Received channel signal $\mathbf{y}_{\text{adv}}$, SNR γ;
  $\hat{\mathbf{x}}_{\text{adv}} \leftarrow \mathcal{D}_{\boldsymbol{\phi}}(\mathbf{y}_{\text{adv}})$;
  $\hat{\mathbf{x}}_{\text{pur}} \leftarrow$ Algorithm 2$(\hat{\mathbf{x}}_{\text{adv}}, \text{SNR}, R)$;
  $\hat{c} \leftarrow \mathcal{F}_{\boldsymbol{\psi}}(\hat{\mathbf{x}}_{\text{pur}})$;
  return $\hat{\mathbf{x}}_{\text{pur}}, \hat{c}$;

when perturbations shift originally misclassified noisy samples across a decision boundary into the true class. To account for both transition directions across the dataset, the Penalized Attack Success Rate (PASR) is defined as

$$\text{PASR} \triangleq \frac{N_{\text{CI}} - \alpha_p \cdot N_{\text{IC}}}{N_{\text{total}}}, \tag{38}$$

where N_{total} is the total number of evaluated samples and $\alpha_p \geq 0$ is the penalty weight on N_{IC}. Notably, at the unit penalty weight $\alpha_p = 1$ used throughout our experiments, PASR equals the net increase in dataset-wide classification error rate ($\text{Err}_{\text{adv}} - \text{Err}_{\text{clean}}$). Across all evaluated SNR regimes, the empirical rate of incorrect $\rightarrow$ correct transitions ($N_{\text{IC}}/N_{\text{total}}$) remains below $< 2.8\%$, confirming that relative attack and defense rankings are insensitive to α_p.

VII. Simulation Results

A. Simulation Setup

We evaluate the proposed SenSA attack and HYDRA defense on the Imagenette dataset [30], a 10-class subset of ImageNet containing easily classifiable categories (e.g., tench, garbage truck, church). All images are resized to 128×128 pixels and normalized to $[0, 1]$. We use the standard training split (9,469 images) for model training and the validation split (3,925 images) for evaluation. Imagenette is chosen because its moderate resolution and clear interclass boundaries allow us to isolate the effects of adversarial perturbations and channel noise from intrinsic classification difficulty. Model training and robustness experiments were conducted on a server equipped with dual Intel Xeon Platinum 8462Y+ CPUs (128 vCPUs), 2 TB RAM, and an NVIDIA H100 80 GB HBM3 GPU; latency and memory profiling was performed separately on an NVIDIA RTX 4090 GPU as specified in Table VI. Table V summarizes the hyperparameters used in the experiments. The parameters are categorized into four groups: SenSA attack, HYDRA defense, training configuration, and evaluation metrics.

TABLE IV
Neural Network Module Configurations

Module	Layer Pipeline & Key Parameters
Encoder	$\text{Conv}(5\times5, 2) \rightarrow [\text{Conv}(3\times3, 2)]_{\times 2} \rightarrow [\text{ResBlock}]_{\times 2} \rightarrow \text{Conv}(1\times1)$ ($8\times$ spatial reduction, mapped to d_z symbols)
Decoder	$\text{Conv}(1\times1) \rightarrow [\text{ResBlock}]_{\times 2} \rightarrow [\text{ConvT}(4\times4, 2)]_{\times 3} \rightarrow \text{Conv}(3\times3)$
Classifier	Pretrained ResNet-18 with $\text{Linear}(512, 10)$ head ($K = 10$)
Analyzer G_ξ	**Trunk:** $\text{Conv}(3\times3) \rightarrow [\text{Conv}(3\times3, 2)]_{\times 3} \rightarrow$ AvgPool **Heads (4×):** Linear $\rightarrow$ ReLU $\rightarrow$ Linear (N_e=4 routing logits, sigmoid det. prob., sigmoid corruption est., K=10 class logits)
Denoising U-Net	**Enc:** $[\text{ResBlock}]_{\times 2}$ at scales $1\times, 2\times, 4\times$ **Mid:** ResBlock $\rightarrow$ Self-Attn $\rightarrow$ ResBlock **Dec:** 3 skip-connected ResBlocks **Emb:** Timestep + class embeddings (K+1 = 11 for CFG)
MoE Head	**Router:** Linear $\rightarrow$ ReLU $\rightarrow$ Linear (residual connection) **Experts** (N_e=4)**:** $[\text{Conv}(1\times1) \rightarrow \text{GN} \rightarrow \text{GELU} \rightarrow \text{Conv}(1\times1)]_{\times 2}$, top-$k$ softmax gating

* $\text{Conv}(m\times n, s)$: stride-$s$ convolution with $m\times n$ kernel; ConvT: transposed convolution; ResBlock: two convolutions with skip connection; GN: GroupNorm; $K = 10$; $N_e = 4$.

TABLE V
Summary of Hyperparameters

Parameter	Value	Parameter	Value
Semantic Communication System			
Channel dim d_z	16	Hidden dim	64
Encoder downsampling	$8\times$	Classifier	ResNet-18
SenSA Attack			
Pref. threshold τ	0.6	Stability ν_0	10^{-8}
Momentum μ	0.9	TX iterations T_{tx}	10
TX step size η_{tx}	$2.5\epsilon/T_{\text{tx}}$	TX init. range	$\mathcal{U}(\pm0.1\epsilon)$
RX budget PSR_{rx}	-15 dB	RX iterations T_{rx}	30
RX step size η_{rx}	$0.15\epsilon_{\text{rx}}$	Conf. threshold τ_{stop}	0.3
HYDRA Defense			
Guidance scale w	2.0	Diff. steps N_{DDIM}	25
EMA coeff. ζ	0.99	Anchor strength λ_a	0.1
Anchor res. (H_a, W_a)	$(64, 64)$	Decay rate κ	0.7
Base noise β_{base}	3.0	Max noise β_{max}	6.0
Top-k experts	2	Purif. rounds R	2
Conf. threshold τ_c	0.5	Gate temp. τ_T	0.1
Training Configuration			
λ_{aux}	1.0	λ_{anc}	0.1
λ_{det}	0.5	λ_{lb}	0.01
λ_{cls}	1.0	Learning rate	10^{-4}
Adam (β_1, β_2)	$(0.9, 0.999)$	LR scheduler	Cosine
Batch size	64	Epochs	100
SNR range (dB)	$[0, 18]$	Warm-up iters	1000
p_{SenSA}	0.5	p_{PGD}	0.3
p_{clean}	0.2	Penalty α_p (PASR)	1.0

The channel dimension is $d_z = 16$, so a $128\times128\times3$ image is carried by 16 complex symbols in total (roughly one symbol per 3,000 source scalars). At this high compression ratio, the decoder reconstructs semantic content under aggressive bandwidth constraints, making reconstruction fidelity evaluation essential alongside task accuracy.

The architectural configurations of all neural network modules are summarized in Table IV. The three stride-2 convolutions of the encoder account for the $8\times$ spatial reduction listed in Table V, which describes the spatial shrinkage performed by the convolutional trunk prior to mapping onto the d_z complex channel symbols.

Authorized licensed use limited to: Chung-ang Univ. Downloaded on November 05,2026 at 01:08:18 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3748461

B. Baselines

1) Attack Baselines: We compare the proposed SenSA attack against the following baselines:

- **Jamming** [31]: A GAN-inspired intelligent jamming strategy that generates interference signals to disrupt the semantic communication channel without exploiting model-specific gradients. Its perturbation budget is set to nominal $\mathrm{PSR}_{\mathrm{rx}} = -15.0$ dB relative to received channel symbols, matching the RX-side capability bound in Table V.
- **SemAdv** [2]: A white-box physical-layer adversarial attack that crafts perturbations in the channel symbol domain to degrade the semantic decoder's reconstruction and task performance. Its perturbation budget is set to nominal $\mathrm{PSR}_{\mathrm{rx}} = -15.0$ dB on channel symbols (corresponding to normalized energy $E_{\mathrm{rx}} = 10^{-1.5} \approx 3.16 \times 10^{-2}$ with per-sample ℓ_2 bound $\epsilon_{\mathrm{rx}} = \sqrt{P d_z \cdot 10^{-1.5}} \approx 0.711$ at $P = 1, d_z = 16$), directly matching the physical-layer power constraint.
- **SemBAT** [8]: A black-box adversarial attack that uses particle swarm optimization to generate perturbations without requiring gradient access to the target semantic communication model. Following the original paper, we use a population size of 20 with $\epsilon = 0.05$.

2) Defense Baselines: We evaluate the proposed HYDRA defense against the following baseline configurations:

- **No Defense**: The standard semantic communication system without any defense mechanism, serving as a lower bound on robustness.
- **DiffPure** [10]: The vanilla diffusion-based adversarial purification method that adds controlled Gaussian noise via forward diffusion and recovers clean images through reverse denoising at the receiver.
- **DiffPure (robust config)**: A heavily regularized diffusion purification baseline following Lee and Kim [27], operating with forward noise starting timestep $t_{\mathrm{start}} = 500$ ($T = 1000$, noise scale $\beta = 0.50$), $N_{\mathrm{DDIM}} = 50$ deterministic DDIM reverse steps ($\eta = 0$), unconditional guidance ($w = 0$), and single-round purification ($R = 1$) to strengthen perturbation elimination.
- **ASGM** [22]: The Adaptive Semantic Generation Module from the DENOISE framework, employing a channel-conditioned semantic reconstruction mechanism that leverages CSI-driven diffusion sampling to denoise channel-corrupted semantic features.
- **CDDM** [21]: Channel Denoising Diffusion Models that leverage a conditional diffusion process at the receiver to denoise received semantic features before decoding.
- **Adv. Training (PGD-AT)** [13]: Adversarial training applied to the semantic communication receiver, retraining the decoder–classifier pipeline ($\mathcal{D}_{\phi}$ and $\mathcal{F}_{\psi}$) against projected gradient descent perturbations injected into the received channel features.
- **Randomized Smoothing** [18]: A certified defense baseline applied in the receiver feature domain. It injects isotropic Gaussian noise $\boldsymbol{\epsilon}_{\mathrm{smooth}} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ with smoothing parameter $\sigma = 0.25$ into received latent features and certifies smoothed majority-vote predictions via Monte Carlo sampling ($N_0 = 100$ selection draws, $N = 500$ certification samples) at confidence level $1 - \alpha = 99.9\%$ ($\alpha = 10^{-3}$). Certified accuracy is evaluated at certified ℓ_2 radius $r = 0.25$ in the received feature space. We clarify that this certificate holds strictly for the receiver feature domain in which smoothing is applied (guaranteeing prediction invariance against receiver-side perturbations $\|\boldsymbol{\delta}_{\mathrm{rx}}\|_2 \leq r$); it does not constitute an end-to-end certificate against the joint TX/RX adversary, where transmitter perturbations propagate through nonlinear semantic transceivers and wireless channel transitions.

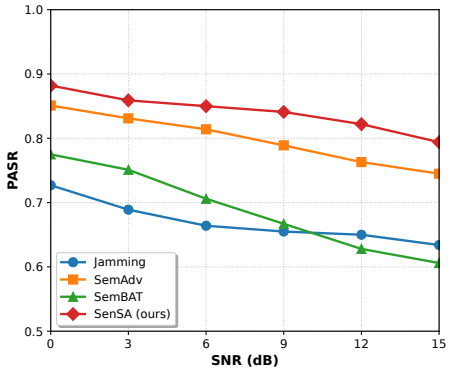


Fig. 3. PASR comparison of attack methods across SNR under the no-defense setting (under nominal per-surface capability bounds in Table V: $\epsilon_{\mathrm{tx}} = 0.05$, $\mathrm{PSR}_{\mathrm{rx}} = -15$ dB where applicable).

C. Performance Comparison

1) Attack Performance: To evaluate the effectiveness of the proposed SenSA attack across channel conditions, Fig. 3 compares the PASR of all attack methods across SNR under the no-defense setting, evaluated under the nominal per-surface capability bounds in Table V ($\epsilon_{\mathrm{tx}} = 0.05$, $\mathrm{PSR}_{\mathrm{rx}} = -15$ dB where applicable). As shown, SenSA consistently achieves the highest PASR across the entire SNR range, reaching 0.882 at 0 dB and outperforming the next-best method (SemAdv, 0.851) by 3.1 percentage points (pp) at 0 dB and by 4.9 pp at 15 dB (0.794 vs. 0.745). The performance gap widens further against black-box approaches: SemBAT exhibits the steepest degradation with increasing SNR (declining from 0.775 to 0.606) due to its reliance on derivative-free optimization, whereas Jamming remains the weakest baseline given its lack of model-specific gradient information (0.727 at 0 dB to 0.634 at 15 dB). We note that this nominal-budget comparison reflects the separate physical capability bounds of Table V; under an equalized aggregate normalized perturbation budget ($E_{\mathrm{total}} \leq 3.28 \times 10^{-2}$ dynamically split via ρ), SenSA likewise outperforms SemAdv by 2.1–4.0 pp across 0–15 dB (0.875–0.788 vs. 0.854–0.748).

To analyze the sensitivity of the attack to the perturbation magnitude, Fig. 4 examines the effect of the transmitter perturbation budget $\epsilon_{\mathrm{tx}} \in \{0.01, 0.03, 0.05, 0.10\}$ on SenSA's attack effectiveness, evaluated with the receiver capability bound fixed to $\mathrm{PSR}_{\mathrm{rx}} = -15$ dB. PASR increases monotonically with ϵ_{tx} across all SNR levels, rising from 0.823 at $\epsilon_{\mathrm{tx}} = 0.01$

Authorized licensed use limited to: Chung-ang Univ. Downloaded on November 05,2026 at 01:08:18 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3748461

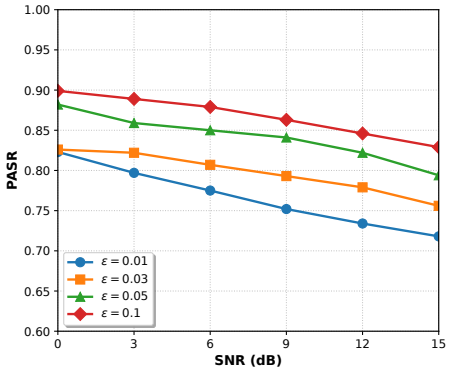


Fig. 4. Effect of transmitter perturbation budget ϵ_{tx} on SenSA PASR across SNR (evaluated with receiver budget fixed at $\text{PSR}_{\text{rx}} = -15$ dB).

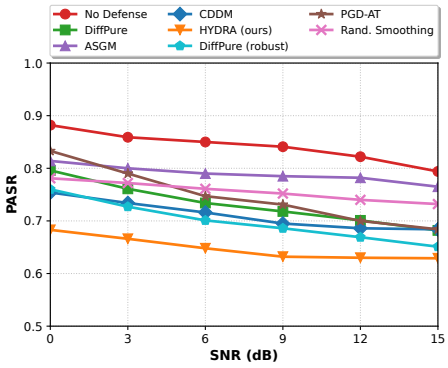


Fig. 5. PASR under defense-unaware SenSA against evaluated defenses across SNR.

to 0.899 at $\epsilon_{\text{tx}} = 0.10$ (0 dB). However, the marginal gain diminishes at larger budgets: the step from $\epsilon_{\text{tx}} = 0.03$ to 0.05 yields 3.8–5.6 pp improvement across SNR, whereas doubling the budget from $\epsilon_{\text{tx}} = 0.05$ to 0.10 adds only 1.7–3.5 pp, indicating that the perturbation saturates the model's vulnerability beyond $\epsilon_{\text{tx}} = 0.05$. All four curves exhibit a downward slope across SNR, declining by 7.0–10.5 pp from 0 dB to 15 dB, which suggests that channel quality degrades attack effectiveness across perturbation budgets, although smaller budgets (e.g., $\epsilon_{\text{tx}} = 0.03$, 7.0 pp decline) exhibit a somewhat flatter SNR dependence than intermediate ones ($\epsilon_{\text{tx}} = 0.05$, 8.8 pp decline). Even the most restrictive budget ($\epsilon_{\text{tx}} = 0.01$) achieves PASR of 0.718–0.823, well above the Jamming baseline (0.634–0.727 in Fig. 3), confirming SenSA's efficiency in utilizing the perturbation budget. We adopt $\epsilon_{\text{tx}} = 0.05$ for all subsequent experiments as it offers a favorable trade-off between TX visual distortion and attack effectiveness.

2) Defense Performance: To evaluate the defense effectiveness of the proposed framework under the defense-unaware SenSA setting, Fig. 5 compares defense baselines against SenSA ($\epsilon_{\text{tx}} = 0.05, \text{PSR}_{\text{rx}} = -15$ dB), where lower PASR indicates stronger defense. As shown, HYDRA consistently achieves the lowest PASR across the entire SNR range, reducing PASR from 0.882 to 0.683 at 0 dB (a 22.6% relative reduction) and from 0.794 to 0.629 at 15 dB (20.8%). Among the competing baselines, CDDM and DiffPure (robust config) achieve the next-best results: CDDM attains 0.754 at 0 dB and 0.684 at 15 dB, whereas DiffPure (robust config) reaches 0.760 at 0 dB and improves to 0.651 at 15 dB, trailing HYDRA by 7.7 pp at 0 dB and 2.2 pp at 15 dB. Vanilla DiffPure begins at 0.796 at 0 dB and decreases to 0.682 at 15 dB, converging closely with CDDM at high SNR. In contrast, Adv. Training (PGD-AT) achieves 0.833 at 0 dB and declines to 0.684 at 15 dB, showing substantial SNR sensitivity as channel noise compounds adversarial training mismatch. Randomized Smoothing and ASGM exhibit limited protection under strong perturbations, with PASR values remaining at 0.781 and 0.814 at 0 dB, respectively (declining to 0.732 and 0.765 at 15 dB; note that for Randomized Smoothing, whose clean accuracy is 0.78–0.83, PASR is upper-bounded by its clean accuracy p_c). Notably, channel-oriented denoising in ASGM provides minimal defense because structured gradient-based distortions differ fundamentally from Gaussian noise. Accordingly, HYDRA's corruption-based timestep selection, top-k MoE denoising, and class-prototype regularization yield lower PASR against defense-unaware attacks across all evaluated SNR regimes.

To verify the generalization of the proposed defense across diverse attack vectors, Fig. 6 evaluates HYDRA's defense against all four attack methods. SenSA remains the most challenging attack, maintaining PASR at or above 0.629 across all SNR levels with only a 5.4 pp drop from 0 dB to 15 dB (0.683 to 0.629), exhibiting the flattest trajectory among all evaluated attacks. In comparison, SemAdv starts at a competitive 0.654 at 0 dB but degrades sharply to 0.433 at 15 dB (a 22.1 pp decline), widening the gap with SenSA from 2.9 pp at low SNR to 19.6 pp at high SNR. This divergence reflects SenSA's SNR-weighted mode selection, which shifts toward TX-only perturbations at high SNR where low channel noise preserves encoder-input perturbations, whereas SemAdv's fixed channel-only perturbation loses effectiveness as the received signal power dominates. SemBAT and Jamming are effectively suppressed by HYDRA, with PASR below 0.39 and 0.21 respectively even at 0 dB (dropping to 0.20 and 0.13 at 15 dB): Jamming's stochastic interference closely resembles channel noise and is naturally handled by diffusion-based denoising, while SemBAT's black-box perturbations lack the gradient-informed precision to evade the defense. Therefore, HYDRA suppresses all four defense-unaware attacks without attack-specific retraining, while joint gradient-based attacks with SNR-weighted mode selection remain the strongest threat.

3) Computational Overhead: To evaluate the computational efficiency and memory overhead of the evaluated schemes under realistic streaming constraints, Table VI reports per-sample inference latency and memory consumption across attack and defense methods under explicit batch-size-one execution on an NVIDIA RTX 4090 GPU. Among attack methods, iteration counts are consistent across runs: SemBAT incurs the highest latency (140.71 ms) due to its population-based search requiring multiple forward passes. In contrast, SenSA requires 42.5 ms under its default configuration ($T_{\text{tx}} = 10, T_{\text{rx}} = 30$ with early stopping), which is only $1.3\times$ that of SemAdv (32.74 ms) and $3.3\times$ faster than SemBAT (140.71 ms), achieving the highest attack success rate (Fig. 3) with moderate computational cost.

Authorized licensed use limited to: Chung-ang Univ. Downloaded on November 05,2026 at 01:08:18 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3748461

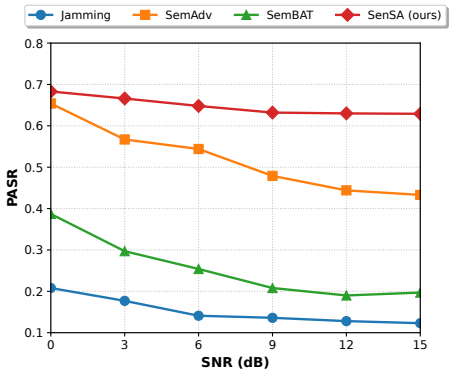


Fig. 6. HYDRA performance against defense-unaware attack methods across SNR.

TABLE VI
PER-SAMPLE INFERENCE LATENCY AND MEMORY CONSUMPTION (BATCH SIZE = 1)

Method	Latency (ms)	Peak Activation / Parameter (MB)
Attack Methods (iteration counts are consistent)		
Jamming	4.09	15–40 / —
SemBAT	140.71	15–40 / —
SemAdv	32.74	15–40 / —
SenSA (Proposed)	42.50	15–40 / —
Defense Methods		
DiffPure	18.73	15–40 / 200–250
DiffPure (robust config)	56.2	15–40 / 200–250
ASGM	3.92	15–40 / 100–150
CDDM	15.52	15–20 / 80–100
Adv. Training (PGD-AT)	0.01	15–40 / 0
Randomized Smoothing	8.6 / 4290*	375–1000 / 0
HYDRA (Proposed)	9.46	30–40 / 230–260

* Single forward pass (8.6 ms) / certified prediction with $N = 500$ samples (4290 ms).

For defense methods, Table VI distinguishes per-sample inference latency, peak activation memory, and parameter storage. Adv. Training (PGD-AT) incurs negligible inference overhead (0.01 ms) and zero additional parameter storage because it alters only the base model weights, though at the cost of reduced clean fidelity. ASGM requires 3.92 ms per sample with 100–150 MB parameter storage. HYDRA achieves an inference latency of 9.46 ms per sample with 30–40 MB peak activation and 230–260 MB parameter memory, outperforming CDDM (15.52 ms), vanilla DiffPure (18.73 ms), and DiffPure (robust config, 56.2 ms). Randomized Smoothing requires 8.6 ms for a single stochastic forward pass and 4290 ms when evaluating certified smoothed majority-vote predictions ($N = 500$ Monte Carlo samples), accompanied by 375–1000 MB activation memory. Accordingly, HYDRA provides an advantageous trade-off between defense effectiveness and low inference latency.

D. Ablation Studies

1) SenSA Strategy Analysis: To evaluate SenSA's per-sample mode selection, we compare it against three fixed-mode baselines: TX-only, RX-only, and Joint with fixed $\rho = 0.5$. All variants share the same per-surface capability bounds and iteration caps. As shown in Table VII (mean $\pm$ seed standard deviation over five seeds, $N = 500$), the two single-surface attacks exhibit complementary SNR-dependent behavior: TX-only PASR increases with SNR as higher channel quality preserves TX perturbations (rising from 0.728 ± 0.009 at 0 dB to 0.784 ± 0.007 at 15 dB), whereas RX-only PASR decreases sharply from 0.796 ± 0.008 at 0 dB to 0.609 ± 0.011 at 15 dB because received features at high SNR lie far from decision boundaries. This matches the SNR trend modeled by $\alpha(\gamma)$ in (11). The fixed Joint mode outperforms both single-surface attacks (0.848 ± 0.007 at 0 dB and 0.772 ± 0.008 at 15 dB) but remains suboptimal due to its fixed iteration allocation ($\rho = 0.5$). SenSA's per-sample mode selection consistently achieves the highest PASR across all SNR levels (0.882 ± 0.006 at 0 dB to 0.794 ± 0.007 at 15 dB), with the largest gains at low SNR where the per-sample decision between RX-only and Joint has the highest impact. Across all evaluated SNR regimes, SenSA's 2.2–3.4 pp advantage over the fixed Joint baseline (and 1.0–18.5 pp over single-surface modes) is consistently larger than observed seed variability ($\sigma_{\text{seed}} \leq 0.011$, individual 95% bootstrap CIs within ± 0.012). Furthermore, a paired bootstrap confidence interval on the mode-selection gain $\Delta\text{PASR} = \text{PASR}_{\text{SenSA}} - \text{PASR}_{\text{Joint}}$ confirms that the advantage is statistically distinct from zero across all channel conditions (paired 95% CI $[+0.018, +0.050]$ at 0 dB and $[+0.008, +0.036]$ at 15 dB, lower bound > 0). Furthermore, evaluating a dual-norm-scaled selection rule ($\widetilde{\mathcal{S}}_{\text{tx}} = \epsilon_{\text{tx}} \|\nabla_{\mathbf{x}} \mathcal{L}\|_1$, $\widetilde{\mathcal{S}}_{\text{rx}} = \epsilon_{\text{rx}} \|\nabla_{\hat{\mathbf{s}}} \mathcal{L}_{\text{rx}}\|_2$) achieves 0.879 ± 0.006 at 0 dB and 0.791 ± 0.007 at 15 dB, performing within 0.3 pp of standard SenSA and confirming that per-sample mode selection is robust to the norm formulation.

TABLE VII
ATTACK MODE ABLATION UNDER A POWER-MATCHED RX BUDGET (PSR $= -15$ DB): PASR OF FIXED-MODE ATTACKS VS. SENSA MODE SELECTION (NO DEFENSE)

Attack Mode	0 dB	5 dB	10 dB	15 dB
TX-only (fixed)	0.728 ± 0.009	0.746 ± 0.008	0.768 ± 0.007	0.784 ± 0.007
RX-only (fixed)	0.796 ± 0.008	0.733 ± 0.009	0.663 ± 0.010	0.609 ± 0.011
Joint (fixed $\rho = 0.5$)	0.848 ± 0.007	0.825 ± 0.008	0.803 ± 0.008	0.772 ± 0.008
SenSA (mode-selected)	$\mathbf{0.882 \pm 0.006}$	$\mathbf{0.855 \pm 0.007}$	$\mathbf{0.825 \pm 0.007}$	$\mathbf{0.794 \pm 0.007}$

Fig. 7 further corroborates these findings by showing the distribution of attack modes selected by SenSA across SNR levels. The joint TX/RX mode dominates across all conditions (57.6%–65.0%), confirming that most samples exhibit comparable TX and RX sensitivities. As SNR increases, a clear shift emerges: TX-only selection increases monotonically from 14.3% at 0 dB to 38.8% at 15 dB, whereas RX-only diminishes from 20.7% at 0 dB to 3.6% at 15 dB, directly reflecting the channel-aware weighting where $\alpha \to 1$ at high SNR amplifies $\mathcal{S}_{\text{tx}}^{\text{eff}}$ and suppresses $\mathcal{S}_{\text{rx}}^{\text{eff}}$. Consequently, SenSA shifts its per-sample domain selection according to the channel SNR rather than committing to a fixed surface, consistent with the SNR-weighting factor in (11).

2) HYDRA Component Ablation: We systematically remove each core component of HYDRA to quantify its individual contribution. All ablated variants are retrained under the same two-stage procedure (Algorithm 3) and evaluated against SenSA ($\epsilon = 0.05$). In the w/o Perturbation Analyzer variant, G_ξ is removed so that the corruption estimate is fixed to $s = 0.5$, expert routing uses uniform weights, and CFG uses the ground-truth label (which is unavailable at the

Authorized licensed use limited to: Chung-ang Univ. Downloaded on November 05,2026 at 01:08:18 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3748461

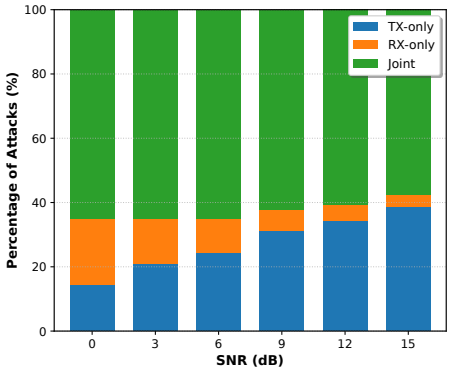


Fig. 7. Distribution of attack modes selected by SenSA across SNR.

TABLE VIII
HYDRA COMPONENT ABLATION UNDER A POWER-MATCHED RX BUDGET (PSR = −15 DB): PASR UNDER SENSA WHEN EACH COMPONENT IS REMOVED

HYDRA Variant	PASR (0 dB)	PASR (15 dB)	ΔPASR (0 dB)
Full HYDRA	**0.683**	**0.629**	–
w/o CFG	0.807	0.728	+0.124
w/o Perturbation Analyzer	0.777	0.703	+0.093
w/o Multi-round ($R=1$)	0.750	0.672	+0.067
w/o Semantic Anchors	0.744	0.679	+0.061
w/o MoE Routing	0.733	0.668	+0.050

receiver in practice). The w/o MoE Routing variant replaces the $N_e = 4$ expert heads with a single denoising head of equal parameter count. Disabling classifier-free guidance (w/o CFG) sets $w = 0$ in (28), reducing the denoiser to the unconditional prediction $\hat{\epsilon}_{uncond}$. Removing semantic anchors (w/o Semantic Anchors) bypasses the class-prototype blending in (33) by setting $\lambda_a = 0$. Finally, the w/o Multi-round variant applies a single purification pass ($R = 1$) instead of $R = 2$.

As shown in Table VIII, removing CFG causes the largest PASR increase (ΔPASR $= +0.124$ at 0 dB), raising PASR to 0.807 above the DiffPure baseline of 0.796 (Fig. 5), because unconditional denoising removes high-frequency perturbation energy without class-conditional guidance toward the target decision region. Removing the perturbation analyzer ranks second (ΔPASR $= +0.093$): even when supplied with ground-truth class labels for CFG, fixing the corruption estimate s degrades robustness compared to per-sample timestep selection via (24). Multi-round purification and class-prototype blending reduce PASR by $+0.067$ and $+0.061$, respectively, while MoE routing contributes $+0.050$. Consequently, combining all five modules achieves the lowest PASR of 0.683 at 0 dB.

E. Reconstruction Fidelity and Signal Dependence

While (8b) constrains mean squared reconstruction distortion, classification accuracy alone does not indicate whether the purified image preserves instance-level spatial structure or regenerates a generic class prototype. To assess image reconstruction fidelity, Table IX reports PSNR (dB), SSIM, and LPIPS [32] means over five seeds ($N = 500$, 95% bootstrap CIs within ± 0.03) at 0 and 15 dB for the clean system, for each attack without defense, and for each defense under SenSA.

TABLE IX
RECONSTRUCTION FIDELITY UNDER ATTACK AND DEFENSE

Configuration	0 dB			15 dB		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Clean (no attack)	22.84	0.682	0.345	28.56	0.854	0.198
Under attack, no defense						
Jamming	14.15	0.312	0.584	19.82	0.521	0.438
SemAdv	16.48	0.438	0.506	21.24	0.615	0.382
SemBAT	18.22	0.528	0.452	23.45	0.698	0.312
SenSA	15.24	0.386	0.548	20.35	0.582	0.408
Under SenSA, with defense						
DiffPure	19.85	0.578	0.415	24.12	0.732	0.282
DiffPure (robust)	19.22	0.552	0.433	23.61	0.712	0.299
ASGM	17.65	0.488	0.468	22.48	0.662	0.342
CDDM	20.92	0.624	0.375	25.84	0.785	0.242
Adv. Train (PGD-AT)	16.20	0.427	0.516	21.20	0.614	0.382
Rand. Smoothing	14.23	0.343	0.581	17.69	0.482	0.491
HYDRA	**21.78**	**0.654**	**0.375**	**27.24**	**0.826**	**0.206**
Signal dependence (instance tracking)						
LPIPS to transmitted $\mathbf{x}$	—	—	0.375	—	—	0.206
LPIPS to same-class $\mathbf{x}'$	—	—	0.428	—	—	0.301
LPIPS to class anchor $\mathbf{a}_c$	—	—	0.458	—	—	0.322
Within-input seed variability	—	—	0.031	—	—	0.019

As shown in Table IX, under undefended attack conditions, SenSA substantially impairs image reconstruction, reducing PSNR to 15.24 dB and SSIM to 0.386 at 0 dB (and 20.35 dB / 0.582 at 15 dB), while increasing perceptual dissimilarity (LPIPS of 0.548 at 0 dB and 0.408 at 15 dB). When HYDRA is deployed under SenSA attack, the reconstruction quality is largely restored across all metrics: PSNR reaches 21.78 dB at 0 dB (within 1.06 dB of the clean system) and 27.24 dB at 15 dB (within 1.32 dB of clean), with SSIM reaching 0.654 and 0.826, and LPIPS recovering to 0.375 (95% CI: 0.352–0.398) and 0.206 (95% CI: 0.188–0.224), outperforming CDDM (20.92 dB / 25.84 dB), vanilla DiffPure (19.85 dB / 24.12 dB), and DiffPure robust config (19.22 dB / 23.61 dB). In contrast, Adv. Training (PGD-AT) and Randomized Smoothing exhibit degraded reconstruction fidelity (16.20 dB and 14.23 dB PSNR at 0 dB), as adversarial training optimizes classification boundaries rather than pixel fidelity, and smoothing adds noise variance.

Fidelity averages cannot separate reconstruction from in-class regeneration on their own, so we pair them with a signal-dependence test. In the primary arm, matched cross-application, a transmitted image $\mathbf{x}$ and a distinct same-class image $\mathbf{x}'$ traverse the pipeline under paired randomness (with identical channel-noise draws and identical diffusion seeds in both arms), and we compare $\text{LPIPS}(\hat{\mathbf{x}}_{\text{pur}}, \mathbf{x})$ against $\text{LPIPS}(\hat{\mathbf{x}}_{\text{pur}}, \mathbf{x}')$ and against $\text{LPIPS}(\hat{\mathbf{x}}_{\text{pur}}, \mathbf{a}_c)$. As reported in Table IX, LPIPS to the transmitted image $\mathbf{x}$ is 0.375 at 0 dB and 0.206 at 15 dB, which is strictly smaller than LPIPS to same-class $\mathbf{x}'$ (0.428 and 0.301) and to the class anchor $\mathbf{a}_c$ (0.458 and 0.322). This instance-tracking margin $\Delta_{\text{inst}} = \text{LPIPS}(\hat{\mathbf{x}}_{\text{pur}}, \mathbf{x}') - \text{LPIPS}(\hat{\mathbf{x}}_{\text{pur}}, \mathbf{x})$ (0.053 at 0 dB, paired 95% CI $[+0.038, +0.068]$; and 0.095 at 15 dB, paired 95% CI $[+0.078, +0.112]$) substantially exceeds the within-input seed variability (0.031 at 0 dB and 0.019 at 15 dB), providing statistical evidence of instance-level signal dependence rather than pure class-mode regeneration.

Authorized licensed use limited to: Chung-ang Univ. Downloaded on November 05,2026 at 01:08:18 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3748461

F. Adaptive and Worst-Case Evaluation

To comprehensively assess defense effectiveness and test for potential gradient-masking artifacts [24], we evaluate all defenses under a standardized white-box protocol where the adversary is fully defense-aware and differentiates through the entire purification pipeline.

a) Attacker Knowledge and Insertion Points: The defense-aware adaptive adversary is $\mathcal{A}_{\text{joint}}$ of Section III with complete white-box knowledge of the deployed defense: decoder $\boldsymbol{\phi}$, analyzer ξ, denoiser $\boldsymbol{\omega}$ and all N_e expert heads, anchor bank $\{\mathbf{a}_j\}$, schedule $\{\bar{\alpha}_t\}$ and DDIM grid, classifier $\boldsymbol{\psi}$, Table V parameters, and true SNR γ. Insertion points are unchanged ($\boldsymbol{\delta}_{\text{tx}}$ in (5), $\boldsymbol{\delta}_{\text{rx}}$ in (6)), ensuring consistent knowledge and capability bounds across defenses to prevent inflated robustness [25].

b) Gradient Path: The primary attack is end-to-end adaptive PGD+EOT, differentiating through the complete pipeline: transmitter gradients flow along $\boldsymbol{\delta}_{\text{tx}} \to \mathcal{E}_{\boldsymbol{\theta}} \to$ channel $\to \mathcal{D}_{\boldsymbol{\phi}} \to G_{\xi} \to$ purifier $\to \mathcal{F}_{\boldsymbol{\psi}}$, and receiver gradients along $\boldsymbol{\delta}_{\text{rx}} \to \mathcal{D}_{\boldsymbol{\phi}} \to G_{\xi} \to$ purifier $\to \mathcal{F}_{\boldsymbol{\psi}}$, spanning forward noising (25), N_{DDIM} CFG-guided DDIM reverse steps (28)–(31) over R rounds, anchor blending (33), and classifier $\mathcal{F}_{\boldsymbol{\psi}}$ with activation checkpointing. Exact gradients are used wherever they exist; BPDA is confined to three non-differentiable operations: analyzer argmax, replaced on backward pass by temperature-scaled softmax over $\text{MLP}_{\text{cls}}(\mathbf{f})$ so gradient reaches $\mathbf{e}_c$ and w_{eff} of (29); top-k expert selection, replaced by softmax over all N_e experts; and argmin in (24) with clipping in (30), handled by straight-through identity. Full-BPDA is included as a secondary baseline [27].

c) EOT Sources and Optimizer Configuration: Under deterministic DDIM ($\eta = 0$) in our primary setup, two randomness sources are active: forward noising $\boldsymbol{\epsilon}$ in (25) and channel noise $\mathbf{n}$. Each step averages loss and gradient over $n_{\text{EOT}} \in [10, 20]$ independent draws. The optimizer is momentum PGD (200 iterations, ≥ 5 restarts) under ℓ_∞ projection for $\boldsymbol{\delta}_{\text{tx}}$ and ℓ_2 projection for $\boldsymbol{\delta}_{\text{rx}}$. Cross-entropy and margin loss $\max_{j \neq c} f_j - f_c$ [14] are run separately, and the stronger adversarial outcome is retained for each sample to avoid loss saturation. The parameter-free APGD step-schedule cross-check (adapted from Croce and Hein [26] with joint $\ell_\infty \times \ell_2$ projections) yields 0.728 PASR (13.8% robust accuracy), closely matching tuned momentum PGD (0.722 PASR / 14.4% accuracy, margin $\leq 0.6\%$).

d) Independent Noise Draws and Convergence Diagnostics: Noise and seeds are redrawn at every iteration and restart. Reported numbers are evaluated on the held-out $N = 500$ validation split with disjoint seeds after convergence across iterations, n_{EOT}, and restarts.

e) Gradient-Masking Diagnostics: To diagnose potential gradient masking [24], [25], we evaluate three sanity checks: (i) *Budget scaling*: scaling $\epsilon \in \{0.05, 0.5\}$ monotonically drops robust accuracy from 14.4% to 0.0%; (ii) *Compute monotonicity*: iterative PGD dominates 1-step FGSM-style attack (14.4% vs. 61.2% accuracy; 0.722 vs. 0.254 PASR), plateauing by $T = 200$ and $n_{\text{EOT}} = 20$; (iii) *Black-box ceiling*: neither Square Attack [28] (0.614 PASR) nor surrogate transfer (0.628 PASR) surpasses white-box PGD+EOT (0.722 PASR). All checks pass as measured, providing no evidence of gradient masking under the evaluated diagnostic suite.

TABLE X
ADAPTIVE AND WORST-CASE EVALUATION OF EACH DEFENSE UNDER A POWER-MATCHED RX BUDGET (PSR $= -15$ DB)

Defense	Clean	PASR under adversary knowledge			
	Acc.	Obliv.	B-Box	BPDA+EOT	PGD+EOT
No Defense	0.86–0.88	0.866	0.82–0.87	0.866	0.866
DiffPure	0.87–0.89	0.796	0.80–0.85	0.80–0.85	0.83–0.87
DiffPure (robust)	0.83–0.86	0.760	0.76–0.81	0.76–0.82	0.79–0.84
ASGM	0.87–0.89	0.814	0.82–0.87	0.83–0.87	0.85–0.88
CDDM	0.87–0.89	0.754	0.78–0.83	0.79–0.84	0.81–0.86
Adv. Train (PGD-AT)	0.84–0.87	0.833	0.82–0.86	0.83–0.86	0.84–0.87
Rand. Smoothing	0.78–0.83	0.781	0.75–0.79	0.77–0.81	0.78–0.83
HYDRA	**0.85–0.88**	**0.683**	**0.63–0.68**	**0.70–0.75**	**0.72–0.76**

f) Reporting Standard and Results: Table X reports clean accuracy and robust PASR across four adversary-knowledge settings at SNR $= 0$ dB under a power-matched budget ($\epsilon_{\text{tx}} = 0.05$, $\text{PSR}_{\text{rx}} = -15$ dB). Ranges denote min–max values across five seeds ($\sigma_{\text{seed}} \leq 0.018$, 95% bootstrap CIs within ± 0.021). Under defense-unaware adversaries, HYDRA achieves the strongest defense, reducing PASR to 0.683 ± 0.009 (vs. CDDM 0.754, DiffPure robust 0.760) while preserving high clean accuracy (0.85–0.88, mean 0.866). Under black-box transfer, HYDRA maintains lower PASR (0.63–0.68) than baselines. Adv. Training (PGD-AT) achieves clean accuracy 0.84–0.87 and PASR 0.833–0.87. Randomized Smoothing provides certified accuracy 0.52–0.61 at radius $r = 0.25$ ($\sigma = 0.25$, $1 - \alpha = 99.9\%$) strictly for receiver perturbations $\|\boldsymbol{\delta}_{\text{rx}}\|_2 \leq r$, with clean utility 0.78–0.83 and empirical robust PASR 0.781 (oblivious) to 0.78–0.83 (PGD+EOT, saturating at p_c^{def}). Under adaptive PGD+EOT, attack success rates rise across all empirical defenses (HYDRA PASR 0.72–0.76, mean 0.741), consistent with known purification limitations [27]. Thus, empirical purification mitigates standard attacks effectively without offering unconditional certified guarantees against adaptive adversaries.

VIII. Conclusion

In this work, we investigated the adversarial vulnerability and defense of semantic communication systems by treating the encoder input and wireless channel as a unified attack surface. We proposed SenSA, which evaluates transmitter- and receiver-side sensitivities per sample, weights them by the channel SNR, and allocates attack surfaces and iterations under separate perturbation budgets. To remove perturbations without attack-specific retraining, we developed HYDRA, a diffusion purification framework that selects the forward-noising starting timestep from the estimated corruption level of each sample. Simulations demonstrate that sensitivity-driven allocation consistently outperforms fixed-surface attack modes across channel conditions, while HYDRA achieves effective empirical mitigation against defense-unaware adversaries with low inference latency.

Notably, empirical purification serves as an effective mitigation rather than an unconditional robustness guarantee, as adaptive white-box adversaries differentiating through the purifier degrade empirical defenses [27]. The present evaluation is limited to image semantic communication over

Authorized licensed use limited to: Chung-ang Univ. Downloaded on November 05,2026 at 01:08:18 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3748461

AWGN channels and a highly compressed latent representation; evaluating the selection rule under fading/MIMO channels, broader semantic tasks, and larger-scale datasets remains an important next step. Promising future directions include extending sensitivity-driven selection to multimodal representations, developing certified defense mechanisms across neural transceivers and channel fading [18], and distilling the mixture-of-experts architecture to reduce edge latency.

REFERENCES

[1] D. Gündüz, Z. Qin, I. Estella Aguerri, H. S. Dhillon, Z. Yang, A. Yener, K. K. Wong, and C.-B. Chae, "Beyond transmitting bits: Context, semantics, and task-oriented communications," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 5–41, 2023.

[2] G. Nan, Z. Li, J. Zhai, Q. Cui, G. Chen, X. Du, X. Zhang, X. Tao, Z. Han, and T. Q. S. Quek, "Physical-layer adversarial robustness for deep learning-based semantic communications," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 8, pp. 2592–2608, 2023.

[3] T. S. Do, T. P. Truong, Q. T. Do, and S. Cho, "TranGDeepSC: Leveraging ViT knowledge in CNN-based semantic communication system," *ICT Express*, vol. 11, no. 2, pp. 335–340, Apr. 2025.

[4] T. S. Do, T. P. Truong, M. C. Ho, Q. T. Do, N.-N. Dao, H. Kim, and S. Cho, "Optimal mean-field Stackelberg game-inspired resource allocation for scalable multi-modal semantic communications in 6G networks," *IEEE Trans. Commun.*, pp. 1–1, 2026.

[5] D. Won, G. Woraphonbenjakul, A. B. Wondmagegn, A.-T. Tran, D. Lee, D. S. Lakew, and S. Cho, "Resource management, security, and privacy issues in semantic communications: A survey," *IEEE Commun. Surveys Tuts.*, vol. 27, no. 3, pp. 1758–1797, 2025.

[6] V.-T. Hoang, V.-L. Nguyen, R.-G. Chang, P.-C. Lin, R.-H. Hwang, and T. Q. Duong, "Adversarial attacks against shared knowledge interpretation in semantic communications," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 11, no. 2, pp. 1024–1040, 2025.

[7] J. Liu, Y. He, W. Xu, Y. Xie, and J. Han, "Manipulating semantic communication by adding adversarial perturbations to wireless channel," in *2024 IEEE/ACM 32nd International Symposium on Quality of Service (IWQoS)*, 2024, pp. 1–10.

[8] Z. Li, J. Zhou, G. Nan, Z. Li, Q. Cui, and X. Tao, "SemBAT: Physical layer black-box adversarial attacks for deep learning-based semantic communication systems," in *2022 IEEE 96th Veh. Technol. Conf. (VTC2022-Fall)*, 2022, pp. 1–5.

[9] M. Shi, Y. Yang, and Y. Lin, "Exploring adversarial attacks in semantic communication systems across diverse channel environments," in *2024 Int. Conf. Ubiquitous Commun. (Ucom)*, 2024, pp. 522–526.

[10] W. Nie, B. Guo, Y. Huang, C. Xiao, A. Vahdat, and A. Anandkumar, "Diffusion models for adversarial purification," in *Proc. 39th Int. Conf. Mach. Learn. (ICML)*, ser. Proceedings of Machine Learning Research, vol. 162. PMLR, 2022, pp. 16 805–16 827.

[11] E. Bourtsoulatze, D. B. Kurka, and D. Gündüz, "Deep joint source-channel coding for wireless image transmission," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 5, no. 3, pp. 567–579, 2019.

[12] T. S. Do, T. P. Truong, Q. T. Do, D. Won, W. Noh, and S. Cho, "BidDeepSC-1.58b: 1.58-bit bidirectional slimmable semantic communication system," *IEEE Trans. Veh. Technol.*, vol. 75, no. 4, pp. 6254–6269, 2026.

[13] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.

[14] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, 2017, pp. 39–57.

[15] Y. Hua, Y. Xu, C. Lyu, J. Liu, Y. Shen, W. Yang, and N. Shiratori, "Imperceptible and targeted physical attacks on deep learning-based speech semantic communications," in *2025 IEEE Wirel. Commun. Netw. Conf. (WCNC)*, 2025, pp. 1–6.

[16] G. Chen, G. Nan, Z. Jiang, H. Du, R. Shi, Q. Cui, and X. Tao, "Lightweight and robust wireless semantic communications," *IEEE Commun. Lett.*, vol. 28, no. 11, pp. 2633–2637, 2024.

[17] X. Ren, J. Wu, H. Xu, and X. Chen, "Diffusion model based secure semantic communications with adversarial purification," in *2024 IEEE 10th Conf. Big Data Secur. Cloud (BigDataSecurity)*, 2024, pp. 130–134.

[18] J. M. Cohen, E. Rosenfeld, and J. Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, vol. 97. PMLR, 2019, pp. 1310–1320.

[19] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 6840–6851.

[20] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.

[21] T. Wu, Z. Chen, D. He, L. Qian, Y. Xu, M. Tao, and W. Zhang, "Cddm: Channel denoising diffusion models for wireless semantic communications," *IEEE Trans. Wireless Commun.*, vol. 23, no. 9, pp. 11 168–11 183, 2024.

[22] S. Wu, C. Li, X. He, W. Jiang, Y. Zhu, and Y. Xu, "DENOISE: Diffusion-enhanced noise optimization for intelligent semantic-aware communication systems," *Int. J. Dynam. Control*, vol. 13, no. 10, p. 339, Oct. 2025.

[23] X. Tian, D. Huang, Z. Qi, X. Zhou, T. Jiang, and Z. Feng, "DiffMoE-Com: Diffusion mixture of experts for channel-adaptive semantic image communications," *IEEE Wireless Commun. Lett.*, vol. 15, pp. 640–644, 2026.

[24] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, vol. 80. PMLR, 2018, pp. 274–283.

[25] F. Tramèr, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 1633–1645.

[26] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *Proc. 37th Int. Conf. Mach. Learn. (ICML)*, vol. 119. PMLR, 2020, pp. 2206–2216.

[27] M. Lee and D. Kim, "Robust evaluation of diffusion-based adversarial purification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023, pp. 134–144.

[28] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein, "Square attack: A query-efficient black-box adversarial attack via random search," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Springer, 2020, pp. 484–501.

[29] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 9185–9193.

[30] J. Howard, "Imagenette: A smaller subset of 10 easily classified classes from ImageNet," https://github.com/fastai/Imagenette, 2019, accessed: Jan. 15, 2025.

[31] R. Tang, D. Gao, M. Yang, T. Guo, H. Wu, and G. Shi, "GAN-inspired intelligent jamming and anti-jamming strategy for semantic communication systems," in *2023 IEEE Int. Conf. Commun. Workshops (ICC Workshops)*. Rome, Italy: IEEE, May 2023, pp. 1623–1628.

[32] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 586–595.

Authorized licensed use limited to: Chung-ang Univ. Downloaded on November 05,2026 at 01:08:18 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.