ATFEATUREPURE: LEARNING A CLASS-CONDITIONAL VECTOR FIELD IN DECISION-RELEVANT FEATURE SPACE FOR ADVERSARIAL DEFENSE

Anonymous authors
Paper under double-blind review

ABSTRACT

Adversarial training (AT) provides a reliable foundation for adversarial robustness, yet adversarial perturbations can still substantially corrupt the internal representations that determine model predictions. Rather than recovering the entire clean input with a high-dimensional generative purifier, we investigate whether AT can be strengthened by directly correcting these decision-relevant representations. We introduce **ATFeaturePure**, which retains AT as the robust learning foundation and augments the classifier with a lightweight class-conditional vector field operating on the compact pre-logit feature. The key challenge is how to learn correction dynamics that actually improve classification robustness. To address this, we develop a local theoretical analysis that identifies the behavior required for effective feature correction and translate these insights into a learning mechanism that jointly trains the classifier and vector field while encouraging controlled local sensitivity, reliable class guidance, correction of adversarial features toward clean representations, and decision-consistent predictions. Extensive experiments across three datasets, two backbones, and multiple strong attacks show that ATFeaturePure consistently improves robustness over recent AT and purification baselines, including against adaptive attacks, while maintaining favorable training efficiency. These results show that theory-guided correction of decision-relevant features can effectively strengthen AT without requiring full input reconstruction.

1 INTRODUCTION

Deep neural networks (DNNs) remain vulnerable to adversarial examples (AEs), where small, carefully crafted perturbations can substantially alter model predictions. Among existing defenses, adversarial training (AT) remains one of the most reliable approaches against strong attacks (Goodfellow et al., 2015; Madry et al., 2018; Athalye et al., 2018; Kim et al., 2023; Losch et al., 2024; Zhang et al., 2025c). By explicitly training on adversarially perturbed examples, AT substantially improves empirical robustness. Nevertheless, two limitations remain prominent. First, robust accuracy commonly remains well below clean accuracy (Tsipras et al., 2019). Second, generating strong AEs during training makes standard AT computationally expensive (Shafahi et al., 2019; Wong et al., 2020; Rice et al., 2020).

Adversarial purification provides a complementary direction by removing adversarial perturbations from the input before classification. Recent score- and diffusion-based methods (Yoon et al., 2021; Nie et al., 2022; Li et al., 2025; Park et al., 2025) have demonstrated strong recovery capability under conventional attacks. However, their robustness can deteriorate substantially under stronger adaptive attacks that account for the purification procedure (Lee & Kim, 2023; Wang et al., 2024). Moreover, diffusion-based purification typically relies on a large generative model and iterative denoising in the input space.

Since AT and purification are complementary, a straightforward way to combine these directions is to place such an input-space purifier before an adversarially trained classifier. Doing so, however, adds an expensive generative procedure to an already costly AT pipeline. This motivates us to reconsider where to perform purification. For classification, recovering the entire clean input in pixel space may be unnecessary if we can instead correct the high-level representation used to make the final decision.

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

Indeed, prior work has shown that small adversarial perturbations in pixel space can induce substantial corruption in internal feature representations (Xie et al., 2019). At the same time, the pooled hidden representation immediately preceding the classification head is substantially lower-dimensional than the input image and concentrates information directly relevant to the final prediction. This suggests a different strategy: rather than coupling AT with an image-space generative purifier, can we retain AT as the robust learning foundation while directly rectifying adversarially perturbed representations in this compact, decision-relevant feature space?

We pursue this idea with **ATFeaturePure**, which augments an adversarially trained classifier with a lightweight, class-conditional vector field that operates on the hidden feature immediately before the final classification head. Starting from a potentially perturbed feature, the vector field performs a small number of iterative corrections conditioned on its class guidance. In this way, purification acts directly on the representation that determines the classifier's prediction, avoiding the need to reconstruct a complete image with a large image-space generative model.

The central question then becomes how to learn such a feature-space vector field. We address this question through a local theoretical analysis before specifying the complete training objective. The analysis identifies a favorable regime in which the class-conditional vector field contracts perturbed features toward class-relevant neighborhoods while introducing only limited distortion to clean features. Together with local stability of the backbone and classification head, these properties yield sufficient conditions for preserving a positive classification margin after purification; when the raw classifier is already locally robust, they further characterize when the practical pseudo-label purifier preserves this robustness and can tighten a regional margin lower bound. These results identify desirable properties of the purification dynamics and motivate our training design.

Guided by these theoretical conditions, we construct the learning objective accordingly. AT of the raw classifier encourages positive raw margins and reliable class guidance; class-conditional flow matching supplies inward-pointing targets toward clean feature endpoints along its training paths; feature alignment anchors purified adversarial features to their paired clean representations; purified classification encourages positive post-purification margins; and parameter norm regularization encourages controlled local sensitivity of the classifier and vector field. Together, these components yield an end-to-end training procedure for learning both the robust classifier and its feature-space purifier.

We evaluate ATFeaturePure on CIFAR-10, CIFAR-100, and Tiny ImageNet using ResNet-18 and WideResNet-28-10 under multiple strong attack families, including multi-restart PGD (Madry et al., 2018), AutoAttack (Croce & Hein, 2020b), and the purification-aware DiffHammer attack (Wang et al., 2024). Across 25 evaluations spanning different datasets, architectures, and attacks, ATFeaturePure outperforms the strongest baseline among PGD-AT (Madry et al., 2018), DAT (Li et al., 2024), and DHAT (Zhang et al., 2025a) by 11.73–32.04 percentage points in robust accuracy, with an average gain of 20.35 percentage points. The robustness advantage also persists against diffusion-based ADBM (Li et al., 2025) under adaptive attacks. At the same time, operating in hidden space keeps the purifier compact: ATFeaturePure requires substantially less total training computation than DAT and DHAT across all directly comparable settings and substantially less than ADBM even before accounting for its score-model pretraining cost, while remaining competitive with PGD-AT in training cost on the CIFAR benchmarks.

Contributions.

- We introduce ATFeaturePure, a defense framework that retains AT as the robust learning foundation while performing purification directly on the compact, decision-relevant pre-logit feature. A lightweight class-conditional vector field iteratively corrects perturbed representations without requiring image reconstruction.
- We develop a local theoretical characterization of favorable feature purification behavior, connecting contractive class-conditional dynamics, preservation of clean features, and controlled local sensitivity to margin preservation and practical pseudo-label guidance.
- We develop a theory-motivated learning mechanism for the class-conditional vector field that combines AT, class-conditional flow matching, feature alignment, purified classification, and parameter regularization to learn the robust classifier and its feature-space purifier jointly.

2 LITERATURE REVIEW

Adversarial training. AT improves robustness by optimizing models against AEs and is commonly formulated as robust optimization over norm-bounded perturbations (Goodfellow et al., 2015; Madry et al., 2018). Subsequent methods have sought a better balance between clean and robust accuracy through alternative objectives or training mechanisms, including TRADES (Zhang et al., 2019) and MART (Wang et al., 2020). More recently, DAT (Li et al., 2024) uses an Adversarial Amplitude Generator to mix frequency-domain amplitudes while preserving phase information for dual AT, whereas DHAT (Zhang et al., 2025a) mitigates biased feature activation by aligning adversarial logits with debiased high-confidence logits. Our ATFeaturePure retains AT as the foundation for learning a robust classifier, but augments it with a learned purification mechanism acting directly on the classifier's hidden representation.

Adversarial purification. Adversarial purification aims to remove adversarial perturbations before classification. Early approaches reconstructed adversarial inputs or projected them toward the clean-data distribution using autoencoders, density models, or generative models (Meng & Chen, 2017; Song et al., 2018; Samangouei et al., 2018). More recent methods employ score-based and diffusion models for purification (Yoon et al., 2021; Nie et al., 2022). In particular, ADBM (Li et al., 2025) constructs an adversarial diffusion bridge from diffused adversarial inputs toward their clean counterparts. CLIPure (Zhang et al., 2025b) moves purification into CLIP's multimodal latent space. A recent preprint, FlowPure (Collaert et al., 2025), instead uses conditional flow matching to learn adversarial purification directly in input space. However, reliably evaluating stochastic purification remains challenging. Lee & Kim (2023) show that inadequate gradient computation can substantially overestimate the robustness of diffusion-based purification, while DiffHammer (Wang et al., 2024) identifies gradient dilemmas in conventional adaptive attacks and develops a stronger attack for evaluating purification defenses. Most existing purification methods operate in the high-dimensional input space, leaving the purification of compact, decision-relevant representations largely unexplored.

Feature purification with AT. Motivated by empirical observations of adversarial corruption in internal features, Xie et al. (2019) introduced denoising blocks at multiple intermediate layers and trained them jointly with the classifier using AT. Their ImageNet training used 30-step PGD before each model update and synchronized training on 128 NVIDIA V100 GPUs, reflecting substantial training cost. Our ATFeaturePure instead targets the compact, decision-relevant feature immediately before the classification head of a standard classifier, where a lightweight class-conditional vector field performs purification. Our local analysis further identifies desirable properties of these purification dynamics and provides the theoretical motivation for learning the defense.

3 METHODOLOGY

Our ATFeaturePure learns a lightweight class-conditional vector field on the decision-relevant feature immediately before the classification head. We first define the purification map, then characterize local properties under which feature correction is favorable, and finally use these properties to motivate the training objective and practical inference rule. Figure 1 illustrates the training and inference pipelines of ATFeaturePure.

3.1 PRELIMINARIES AND FEATURE-SPACE PURIFICATION

Let $(\mathbf{x}, y) \sim \mathcal{D}$, and decompose the raw classifier as

$$f_\theta(\mathbf{x}) = g_\theta(h_\theta(\mathbf{x})), \qquad \psi(\mathbf{x}) = \arg\max_c f_\theta(\mathbf{x})_c, \tag{1}$$

where $h_\theta : \mathbb{R}^{d_\mathbf{x}} \to \mathbb{R}^{d_h}$ is the feature encoder and $g_\theta : \mathbb{R}^{d_h} \to \mathbb{R}^C$ is the affine classification head. We consider untargeted ℓ_∞ perturbations $\mathbf{x}' \in \mathcal{B}_\infty(\mathbf{x}, \varepsilon) := \{\mathbf{x}' : \|\mathbf{x}' - \mathbf{x}\|_\infty \le \varepsilon\}$. Additional notation is summarized in Appendix A.

Class-conditional vector field. We introduce a vector field

$$v_\phi : [0,1] \times \mathbb{R}^{d_h} \times \{1, \dots, C\} \to \mathbb{R}^{d_h}, \tag{2}$$

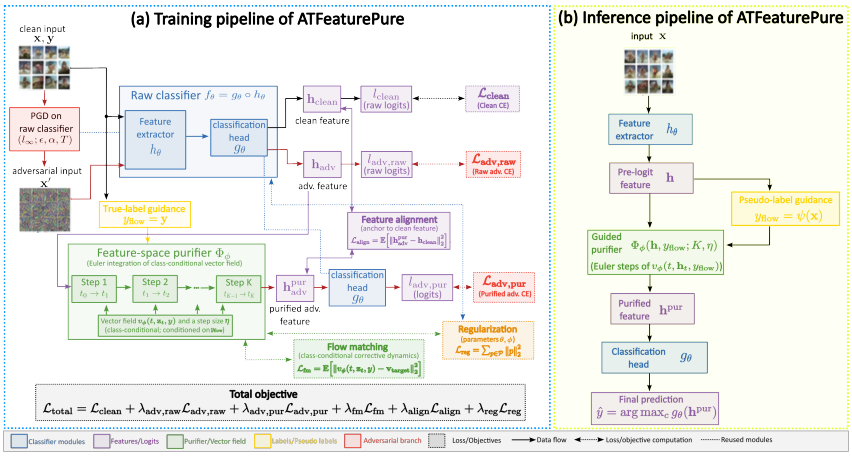


Figure 1: **Overview of ATFeaturePure. (a) Training.** The objective combines clean, raw adversarial, and purified adversarial classification with flow matching, feature alignment, and parameter regularization under true-label guidance. **(b) Inference.** The raw prediction provides pseudo-label guidance for feature purification before final classification.

which takes a time t, feature $\mathbf{z}$, guidance label y_{flow}, and outputs a velocity in feature space. Conceptually, purification follows

$$\frac{d\mathbf{z}_t}{dt} = v_\phi(t, \mathbf{z}_t, y_{\text{flow}}), \qquad \mathbf{z}_0 = h_\theta(\mathbf{x}). \tag{3}$$

In practice, we use $K \geq 1$ explicit Euler steps,

$$\mathbf{z}^{(k+1)} = \mathbf{z}^{(k)} + \eta v_\phi(t_k, \mathbf{z}^{(k)}, y_{\text{flow}}), \qquad k = 0, \ldots, K-1, \tag{4}$$

with $\mathbf{z}^{(0)} = h_\theta(\mathbf{x})$. We denote the resulting map by $\Phi_\phi(\mathbf{z}, y_{\text{flow}}; K, \eta)$, suppressing the time schedule for brevity, and define

$$F_{\theta,\phi}(\mathbf{x}, y_{\text{flow}}) := g_\theta(\Phi_\phi(h_\theta(\mathbf{x}), y_{\text{flow}}; K, \eta)) \,. \tag{5}$$

Training and the theoretical analysis use the true label y as guidance. At validation and test time, where y is unavailable, we use the raw prediction $y_{\text{flow}} = \psi(\mathbf{x})$. Accordingly, the deployed ATFeaturePure classifier is

$$f_{\text{ATFP}}(\mathbf{x}) := F_{\theta,\phi}(\mathbf{x}, \psi(\mathbf{x})) = g_\theta(\Phi_\phi(h_\theta(\mathbf{x}), \psi(\mathbf{x}); K, \eta)) \,. \tag{6}$$

3.2 Local Theory of Feature Purification

We next identify local properties that make feature purification favorable. The analysis is stated for true-label guidance; Corollary 3.5 connects it to the pseudo-label rule used in practice.

For class y, define the feature-space logit margin

$$m_y(\mathbf{z}) := g_\theta(\mathbf{z})_y - \max_{c \neq y} g_\theta(\mathbf{z})_c, \tag{7}$$

and, for a clean sample $(\mathbf{x}, y)$, write $\gamma(\mathbf{x}) := m_y(h_\theta(\mathbf{x})) > 0$. A positive $m_y(\mathbf{z})$ means that the head predicts class y, while its magnitude measures the logit separation between y and its strongest competitor; thus $\gamma(\mathbf{x})$ is the clean class-y margin of $\mathbf{x}$.

Assumption 3.1 (Local Lipschitzness)**.** There exist $L_h, L_g > 0$ such that, over the local input region and all feature points reached by the purification trajectories under consideration,

$$\|h_\theta(\mathbf{x}) - h_\theta(\mathbf{x}')\|_2 \leq L_h \|\mathbf{x} - \mathbf{x}'\|_\infty \,, \tag{8}$$

$$\|g_\theta(\mathbf{z}) - g_\theta(\mathbf{z}')\|_2 \leq L_g \|\mathbf{z} - \mathbf{z}'\|_2 \,. \tag{9}$$

Assumption 3.2 (Contractive class-conditional vector field)**.** For each class y, there exist an analytical reference point $\mathbf{c}_y \in \mathbb{R}^{d_h}$ and constants $\kappa, M, R > 0$ such that, for all $t \in [0, 1]$ and $\|\mathbf{z} - \mathbf{c}_y\|_2 \leq R$,

$$\langle \mathbf{z} - \mathbf{c}_y, v_\phi(t, \mathbf{z}, y) \rangle \leq -\kappa \|\mathbf{z} - \mathbf{c}_y\|_2^2 \,, \tag{10}$$

$$\|v_\phi(t, \mathbf{z}, y)\|_2 \leq M \|\mathbf{z} - \mathbf{c}_y\|_2 \,. \tag{11}$$

The Euler step size is chosen such that

$$0 \le 1 - 2\eta\kappa + \eta^2 M^2 < 1, \qquad \rho := \sqrt{1 - 2\eta\kappa + \eta^2 M^2} < 1. \tag{12}$$

The reference $\mathbf{c}_y$ is used only for the local analysis and is not estimated by the implementation.

Assumption 3.3 (Clean-feature preservation under purification)**.** For a clean sample $(\mathbf{x}, y)$,

$$\|\Phi_\phi(h_\theta(\mathbf{x}), y; K, \eta) - h_\theta(\mathbf{x})\|_2 \le \delta_{\text{clean}}(\mathbf{x}) \tag{13}$$

for some $\delta_{\text{clean}}(\mathbf{x}) \ge 0$.

To characterize the head around the class reference, define the regional margin profile

$$\Gamma_y(r) := \inf_{\|\mathbf{z} - \mathbf{c}_y\|_2 \le r} m_y(\mathbf{z}), \qquad 0 \le r \le R. \tag{14}$$

Thus, $\Gamma_y(r)$ is the worst class-y margin within the radius-r neighborhood around $\mathbf{c}_y$; $\Gamma_y(r) > 0$ means that the entire region remains class-y consistent. By set inclusion, $\Gamma_y(r)$ is nonincreasing as r grows.

Theorem 3.4 (Local robustness and clean-margin preservation)**.** *Suppose Assumptions 3.1–3.3 hold for $(\mathbf{x}, y)$, and define*

$$r_0(\mathbf{x}, \varepsilon) := L_h \varepsilon + \|h_\theta(\mathbf{x}) - \mathbf{c}_y\|_2, \qquad r_K := \rho^K r_0(\mathbf{x}, \varepsilon). \tag{15}$$

If $r_0(\mathbf{x}, \varepsilon) \le R$, then every $\mathbf{x}' \in \mathcal{B}_\infty(\mathbf{x}, \varepsilon)$ satisfies

$$\|\Phi_\phi(h_\theta(\mathbf{x}'), y; K, \eta) - \mathbf{c}_y\|_2 \le r_K, \qquad m_y(\Phi_\phi(h_\theta(\mathbf{x}'), y; K, \eta)) \ge \Gamma_y(r_K). \tag{16}$$

Hence, if $\Gamma_y(r_K) > 0$, the true-label-guided defended classifier is locally robust on $\mathcal{B}_\infty(\mathbf{x}, \varepsilon)$. Moreover,

$$m_y(\Phi_\phi(h_\theta(\mathbf{x}), y; K, \eta)) \ge \gamma(\mathbf{x}) - 2L_g \delta_{\text{clean}}(\mathbf{x}), \tag{17}$$

so the clean prediction is preserved whenever $2L_g \delta_{\text{clean}}(\mathbf{x}) < \gamma(\mathbf{x})$.

Theorem 3.4 identifies two complementary design targets: perturbed features should be contracted into a smaller class-relevant region, while purification should not substantially distort an already correct clean feature. The intermediate results and complete proof are given in Appendix B.

Corollary 3.5 (Conditional robustness under pseudo-label guidance)**.** *Under Theorem 3.4, suppose the raw classifier is locally robust on $\mathcal{B}_\infty(\mathbf{x}, \varepsilon)$, i.e., $\psi(\mathbf{x}') = y$ for every $\mathbf{x}'$ in the ball. Then the practical pseudo-label purifier applies the same class-y field as the theorem and is locally robust whenever $\Gamma_y(r_K) > 0$. In addition,*

$$\Gamma_y(r_K) \ge \Gamma_y(r_0), \tag{18}$$

because $r_K \le r_0$. Thus contraction preserves the regional margin lower bound and tightens it whenever $\Gamma_y(r_K) > \Gamma_y(r_0)$.

For fixed local conditions with $\rho < 1$, increasing K progressively shrinks the feature-region radius r_K, yielding a monotonically nondecreasing regional margin lower bound $\Gamma_y(r_K)$. Since K also interacts with the Euler step size η and the learned vector field in practice, we select (K, η) jointly on held-out validation data.

The theorem and corollary identify four properties that guide our learning design: (i) a robust raw classifier that supplies reliable class guidance, (ii) class-conditioned corrective dynamics toward clean feature regions, (iii) preservation of decision-relevant clean representations together with positive post-purification margins, and (iv) controlled local sensitivity. We next translate these properties into the training objective.

3.3 Theory-Motivated Learning of ATFeaturePure

For each minibatch $(\mathbf{x}, y)$, we generate AEs against the raw classifier as

$$\mathbf{x}_{\text{adv}} = \text{PGD}(f_\theta, \mathbf{x}, y; \varepsilon, \alpha, T), \tag{19}$$

where α and T denote the PGD step size and number of iterations, respectively. We then compute

$$\begin{aligned} \mathbf{h}_{\text{clean}} &= h_\theta(\mathbf{x}), & \mathbf{h}_{\text{adv}} &= h_\theta(\mathbf{x}_{\text{adv}}), \\ \mathbf{h}_{\text{adv}}^{\text{pur}} &= \Phi_\phi(\mathbf{h}_{\text{adv}}, y; K_{\text{train}}, \eta_{\text{train}}). \end{aligned} \tag{20}$$

Local stability encouraged by AT. Corollary 3.5 shows why the raw classifier is not merely an auxiliary branch: when its prediction remains correct in the local perturbation ball, the pseudo-label used in practice coincides with the true class that conditions the theoretical field. We therefore retain AT on the raw classifier. Let

$$\mathcal{L}_{\mathrm{AT}}(\theta; \mathbf{x}, y) := \max_{\mathbf{x}' \in \mathcal{B}_\infty(\mathbf{x}, \varepsilon)} \mathcal{L}_{\mathrm{CE}}(f_\theta(\mathbf{x}'), y), \tag{21}$$

which is approximated by PGD during training. The following elementary relation makes the margin connection explicit.

Lemma 3.6 (Cross-entropy control of the class margin)**.** *For any feature* $\mathbf{z}$*, if* $\mathcal{L}_{\mathrm{CE}}(g_\theta(\mathbf{z}), y) \le \tau < \log 2$*, then*

$$m_y(\mathbf{z}) \ge -\log(e^\tau - 1) > 0. \tag{22}$$

The proof is in Appendix C. Consequently, sufficiently small robust cross-entropy yields a positive raw margin throughout the perturbation ball, placing the raw classifier in the favorable regime of Corollary 3.5. This motivates

$$\mathcal{L}_{\mathrm{clean}} = \mathcal{L}_{\mathrm{CE}}(g_\theta(\mathbf{h}_{\mathrm{clean}}), y), \qquad \mathcal{L}_{\mathrm{adv,raw}} = \mathcal{L}_{\mathrm{CE}}(g_\theta(\mathbf{h}_{\mathrm{adv}}), y). \tag{23}$$

Corrective class-conditioned dynamics from flow matching. Assumption 3.2 favors a field with inward corrective directions in a class-relevant neighborhood. We learn such directions from clean feature distributions using flow matching (Lipman et al., 2023). For $t \sim \mathrm{Uniform}(0, 1)$ and $\mathbf{z}_0 \sim \mathcal{N}(0, I)$, define

$$\begin{aligned} \mathbf{z}_t &= (1-t)\mathbf{z}_0 + t\mathbf{h}_{\mathrm{clean}}, & \mathbf{v}_{\mathrm{target}} &= \mathbf{h}_{\mathrm{clean}} - \mathbf{z}_0, \\ \mathcal{L}_{\mathrm{fm}} &= \mathbb{E}\left[\left\| v_\phi(t, \mathbf{z}_t, y) - \mathbf{v}_{\mathrm{target}} \right\|_2^2 \right]. \end{aligned} \tag{24}$$

For $t < 1$, the interpolation gives

$$\mathbf{v}_{\mathrm{target}} = \frac{\mathbf{h}_{\mathrm{clean}} - \mathbf{z}_t}{1 - t}, \tag{25}$$

so the target velocity is parallel to the direction from the current interpolated feature $\mathbf{z}_t$ toward its clean endpoint. Conditioning these targets on y encourages the learned field to capture class-conditioned corrective directions consistent with the contraction behavior identified by the local theory. Appendix C gives the full geometric derivation and clarifies the relation to Assumption 3.2.

Clean-feature anchoring and post-purification margin. Assumption 3.3 identifies the clean representation as an anchor that purification should not move away from unnecessarily. In training, we use the paired clean feature to anchor the corrected adversarial representation,

$$\mathcal{L}_{\mathrm{align}} = \mathbb{E}\left[\left\| \mathbf{h}_{\mathrm{adv}}^{\mathrm{pur}} - \mathbf{h}_{\mathrm{clean}} \right\|_2^2 \right]. \tag{26}$$

This is a practical surrogate rather than a direct enforcement of Assumption 3.3 on a purified clean input. Nevertheless, the same Lipschitz argument behind Eq. (17) shows that if $\left\| \mathbf{h}_{\mathrm{adv}}^{\mathrm{pur}} - \mathbf{h}_{\mathrm{clean}} \right\|_2 \le \delta$, then its class-$y$ margin is at least $\gamma(\mathbf{x}) - 2L_g\delta$; see Appendix C. We additionally optimize

$$\mathcal{L}_{\mathrm{adv,pur}} = \mathcal{L}_{\mathrm{CE}}(g_\theta(\mathbf{h}_{\mathrm{adv}}^{\mathrm{pur}}), y), \tag{27}$$

which directly encourages a positive margin after purification through Lemma 3.6. Thus alignment supplies the clean feature target while purified classification encourages the corrected representation to remain decision-consistent.

Controlled local sensitivity through regularization. The sufficient conditions depend explicitly on L_h through r_0, on M through ρ, and on L_g through clean-margin preservation. The contraction factor ρ in Eq. (12) depends jointly on the vector-field magnitude bound M and the Euler step size η. Although reducing η can help satisfy $\rho < 1$, relying on an excessively small step size also weakens each corrective update because $\left\| \mathbf{z}^{(k+1)} - \mathbf{z}^{(k)} \right\|_2 = \eta \left\| v_\phi(t_k, \mathbf{z}^{(k)}, y) \right\|_2$. We therefore also regularize model parameters to encourage controlled sensitivity and vector-field magnitude rather than relying solely on shrinking η.

For a matrix-valued weight W, let $\|W\|_2$ denote its spectral (operator) norm and $\|W\|_F$ its Frobenius norm. The squared ℓ_2 penalty on its entries is $\|W\|_F^2$, and $\|W\|_2 \le \|W\|_F$ (Trefethen & Bau, 1997). Since standard network Lipschitz upper bounds depend on such layer operator norms, discouraging large Frobenius norms provides a tractable surrogate for controlling these bounds. We use $\mathcal{L}_{\text{reg}} = \sum_{p\in\mathcal{P}} \|p\|_2^2$; the full connection and parameter set $\mathcal{P}$ are detailed in Appendix C.4.

Combining these components gives

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{clean}} + \lambda_{\text{adv,raw}}\mathcal{L}_{\text{adv,raw}} + \lambda_{\text{adv,pur}}\mathcal{L}_{\text{adv,pur}} + \lambda_{\text{fm}}\mathcal{L}_{\text{fm}} + \lambda_{\text{align}}\mathcal{L}_{\text{align}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}. \quad (28)$$

The classifier and vector field are optimized jointly. Loss weights, optimizer settings, and complete training details are given in Appendix E.

3.4 Lightweight Purifier Architecture and Inference

Purifier architecture. Because the vector field operates on the pooled pre-logit feature rather than a high-dimensional image, we design lightweight MLP-based parameterizations for the ATFeaturePure purifier. We use a two-layer MLP (Haykin, 2009) on CIFAR-10 and a SimpleResMLP on CIFAR-100 and Tiny ImageNet. Full layer specifications and embedding dimensions are provided in Appendix D.

Inference and hyperparameter selection. Corollary 3.5 motivates using the raw prediction as class guidance in the deployed classifier of Eq. (6). As motivated by the local analysis above, we select (K, η) jointly on the held-out validation split by maximizing the sum of clean and PGD robust accuracy. The candidate grids, checkpoint-selection procedure, and additional inference details are given in Appendix E.2; the test set is reserved only for final evaluation.

4 Experiments

4.1 Experimental settings

Benchmarks and baselines. Following recent work (Li et al., 2024; 2025; Zhang et al., 2025a), we evaluate ResNet-18 (Res-18) (He et al., 2016) on CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009), as well as Tiny ImageNet (Deng et al., 2009). We additionally evaluate WideResNet-28-10 (WRN28-10) (Zagoruyko & Komodakis, 2016) on CIFAR-10 and CIFAR-100. We compare against PGD-AT (Madry et al., 2018), Dual Adversarial Training (DAT) (Li et al., 2024), Debiased High-Confidence Adversarial Training (DHAT) (Zhang et al., 2025a), and the purification-based Adversarial Diffusion Bridge Model (ADBM) (Li et al., 2025). Unless otherwise specified, we follow the original papers for baseline training and inference settings.

Model selection and training. For each dataset, we use 95% of the original training set for training and 5% for validation, reserving the official test set only for final benchmarking. Checkpoint and (K, η) selection follow Section 3.4 and Appendix E.2. For ATFeaturePure, training and validation AEs use $T = 10$ PGD steps, following the standard setting in (Madry et al., 2018; Li et al., 2024; Zhang et al., 2025a). Appendix F provides detailed optimization schedules, regularization coefficients, and baseline protocol details.

Empirical robustness evaluation. Following recent work (Li et al., 2024; Zhang et al., 2025a), we evaluate final robustness under an ℓ_∞ budget of $\frac{8}{255}$. For fixed-step PGD attacks, we use step size $\alpha = \frac{2}{255}$. We conduct all robustness evaluations on the full official test set of each dataset, without subsampling or filtering. We measure all reported ATFeaturePure robust accuracies on the complete deployed pipeline in Eq. (6), including pseudo-label guidance, feature purification, and final classification, rather than on the raw classifier or the purification component in isolation. We use multi-restart PGD (Madry et al., 2018) (PGD-20 $\times$ 3 and PGD-100 $\times$ 10), AutoAttack (AA) (Croce & Hein, 2020b), and DiffHammer (Wang et al., 2024). We use PGD restarts only for final test-set evaluation. Our AA configuration uses all non-targeted attacks and is stronger than the default composition in our reproduced experiments. For DiffHammer, we report both its PGD and APGD modes. Following Lee & Kim (2023), we additionally use expectation over transformation (EOT, 20 iterations) in the adaptive attack against the stochastic ADBM purifier to account for purification randomness. Appendix F provides complete attack configurations and gradient computation details.

Reporting protocol. We report mean ± standard deviation of clean and robust accuracy over four runs. For each seed, all methods use the same train/validation split and randomized experimental pipeline; Appendix F lists the exact seeds.

4.2 Experimental results

4.2.1 Robustness against AT baselines.

Tables 1 and 2 show that ATFeaturePure achieves the highest robust accuracy across every evaluated combination of dataset, architecture, and attack. Compared with the strongest competing AT baseline in each column, gains range from 11.73 to 32.04 percentage points across 25 comparisons involving PGD, AA, and DiffHammer, with an average improvement of 20.35 percentage points. The consistency across attack families with different optimization mechanisms indicates that the gain is not tied to a particular attack configuration.

Across datasets and backbones. With ResNet-18, ATFeaturePure improves the strongest baseline by up to 26.39 points on CIFAR-10 and 32.04 points on CIFAR-100, while also achieving the highest clean accuracy on both datasets. On Tiny ImageNet, the robustness advantage remains substantial across all attacks, with gains of 15.84–25.67 points, while DAT has 1.94 points higher clean accuracy. Table 2 shows the same pattern with WRN28-10: ATFeaturePure improves the strongest baseline by 11.73–24.63 points across the evaluated attacks, while its clean accuracy remains within 2.38 points of PGD-AT on CIFAR-10 and 1.15 points on CIFAR-100, and exceeds DAT and DHAT. These results show that the robustness advantage persists across dataset scale, image resolution, number of classes, and backbone capacity.

4.2.2 Comparison with purification-based defense.

Table 3 compares ATFeaturePure with diffusion-based ADBM. Although ADBM has 7.23 points higher clean accuracy, ATFeaturePure is substantially more robust under every evaluation: the gains are 23.85 and 29.45 points under PGD-20×3 and PGD-100×10, 12.66 points under AA without EOT, and 30.97 points with EOT. Under DiffHammer, which explicitly targets purification defenses, gains increase further to 41.81 and 39.00 points for its PGD and APGD modes.

4.3 Model Size and Training Efficiency

Table 4 reports learnable parameters and the total computation required to obtain a trained defense; for ATFeaturePure, this cost also includes validation-based selection of (K, η). Relative to PGD-AT, ATFeaturePure uses about 28.5% less training computation on CIFAR-10, incurs only a modest increase on CIFAR-100, and is more expensive on Tiny ImageNet. Thus, its training cost remains competitive with PGD-AT on the CIFAR benchmarks, even with the additional feature-purification component.

The efficiency advantage is larger relative to recent stronger defenses. Across directly comparable settings, ATFeaturePure reduces total training computation by 61.59%–82.50% relative to DAT and 52.21%–68.24% relative to DHAT. On CIFAR-10 with WRN28-10, it also reduces the reported ADBM training computation by 73.77%, even before accounting for the pretraining cost of ADBM's score-SDE model (Song et al., 2021). The parameter overhead remains moderate: ATFeaturePure uses 12.23–15.46M learnable parameters with ResNet-18 and 38.13–41.05M with WRN28-10, remaining substantially smaller than DHAT and ADBM in the directly comparable settings.

5 Conclusion

We introduced **ATFeaturePure**, which retains AT as the robust learning foundation while shifting purification from the input space to a compact pre-logit feature. A lightweight class-conditional vector field corrects perturbed representations without reconstructing the input. Our local analysis characterizes favorable purification behavior through contractive dynamics, preservation of clean features, and controlled local sensitivity, providing theoretical motivation for the learning design. Across three datasets, two backbones, and multiple strong attacks, ATFeaturePure consistently improves robust accuracy over recent AT baselines and remains substantially more robust than

Table 1: ResNet-18 clean and robust accuracy (mean ± std over four runs).

Dataset	Method	Clean	PGD-20 × 3	PGD-100 × 10	AA	DiffHammer	
						PGD	APGD
CIFAR-10	PGD-AT	83.68 ± 0.94	50.86 ± 0.28	50.44 ± 0.34	47.85 ± 0.13	48.62 ± 0.08	48.59 ± 0.11
	DAT	81.04 ± 0.84	56.23 ± 0.27	56.08 ± 0.29	50.85 ± 0.13	51.38 ± 0.19	51.41 ± 0.19
	DHAT	78.52 ± 0.77	51.27 ± 0.19	51.08 ± 0.21	44.71 ± 0.69	45.56 ± 0.64	45.54 ± 0.65
	Ours	83.97 ± 0.50	82.62 ± 1.61 (+26.39)	81.30 ± 1.67 (+25.22)	62.98 ± 1.10 (+12.13)	70.25 ± 0.97 (+18.87)	67.45 ± 1.28 (+16.04)
CIFAR-100	PGD-AT	56.83 ± 0.85	27.99 ± 0.17	27.86 ± 0.18	24.64 ± 0.14	25.49 ± 0.03	25.46 ± 0.08
	DAT	57.06 ± 0.56	32.15 ± 0.30	32.04 ± 0.24	26.42 ± 0.20	27.02 ± 0.21	27.04 ± 0.22
	DHAT	51.83 ± 0.54	28.06 ± 0.29	27.95 ± 0.32	22.51 ± 0.51	23.43 ± 0.49	23.39 ± 0.49
	Ours	60.93 ± 0.40	64.19 ± 1.29 (+32.04)	63.70 ± 1.26 (+31.66)	41.71 ± 0.92 (+15.29)	48.52 ± 0.73 (+21.50)	46.64 ± 0.93 (+19.60)
Tiny ImageNet	PGD-AT	50.23 ± 0.26	23.70 ± 0.13	23.56 ± 0.09	19.25 ± 0.11	20.30 ± 0.18	20.28 ± 0.18
	DAT	52.28 ± 0.44	26.23 ± 0.41	26.16 ± 0.41	18.95 ± 0.34	20.06 ± 0.38	20.01 ± 0.39
	Ours	50.34 ± 0.30	51.90 ± 1.02 (+25.67)	51.59 ± 1.02 (+25.43)	35.09 ± 0.48 (+15.84)	39.07 ± 0.42 (+18.77)	38.26 ± 0.46 (+17.98)

Table 2: WRN28-10 clean and robust accuracy (mean ± std over four runs).

Dataset	Method	Clean	PGD-20 × 3	PGD-100 × 10	AA	DiffHammer	
						PGD	APGD
CIFAR-10	PGD-AT	86.49 ± 0.45	53.26 ± 0.12	52.88 ± 0.12	50.96 ± 0.36	51.54 ± 0.38	51.50 ± 0.36
	DAT	83.14 ± 0.79	57.89 ± 0.20	57.68 ± 0.21	53.18 ± 0.48	53.67 ± 0.45	53.68 ± 0.45
	DHAT	82.19 ± 0.91	53.09 ± 0.80	52.95 ± 0.81	46.02 ± 0.65	46.85 ± 0.65	46.85 ± 0.65
	Ours	84.11 ± 0.94	81.75 ± 4.55 (+23.86)	80.01 ± 4.98 (+22.33)	64.91 ± 0.97 (+11.73)	71.44 ± 1.39 (+17.77)	69.44 ± 1.42 (+15.76)
CIFAR-100	PGD-AT	62.08 ± 0.36	30.85 ± 0.39	30.56 ± 0.35	27.26 ± 0.25	28.27 ± 0.27	28.26 ± 0.27
	DAT	56.73 ± 0.49	31.21 ± 0.67	31.07 ± 0.75	26.60 ± 0.49	27.28 ± 0.43	27.23 ± 0.44
	DHAT	56.78 ± 0.82	29.87 ± 1.95	29.71 ± 1.97	24.35 ± 1.46	25.20 ± 1.59	25.13 ± 1.54
	Ours	60.93 ± 0.43	55.84 ± 1.64 (+24.63)	54.82 ± 1.86 (+23.75)	40.65 ± 0.57 (+13.39)	45.33 ± 0.69 (+17.06)	44.33 ± 0.68 (+16.07)

Table 3: Comparison of ATFeaturePure and ADBM on CIFAR-10 with WRN28-10 (mean ± std over four runs).

Method	Clean	PGD-20 × 3	PGD-100 × 10	AA (wo EOT)	AA (w EOT)	DiffHammer	
						PGD	APGD
ADBM	91.34 ± 0.30	57.90 ± 0.18	50.56 ± 0.29	52.25 ± 0.23	34.02 ± 0.43	29.63 ± 0.46	30.44 ± 0.40
Ours	84.11 ± 0.94	81.75 ± 4.55 (+23.85)	80.01 ± 4.98 (+29.45)	64.91 ± 0.97 (+12.66)	64.99 ± 1.87 (+30.97)	71.44 ± 1.39 (+41.81)	69.44 ± 1.42 (+39.00)

Table 4: Learnable parameters (M) and total training computation (PFLOPs) for ResNet-18 and WRN28-10 defenses. A dash denotes an unevaluated setting.

(a) ResNet-18

Method	CIFAR-10		CIFAR-100		Tiny ImageNet	
	Params (M)	Cost (PFLOPs)	Params (M)	Cost (PFLOPs)	Params (M)	Cost (PFLOPs)
PGD-AT	11.17	94.98	11.22	94.99	11.27	506.59
DAT	15.04	387.99	15.11	388.05	24.65	2069.31
DHAT	22.35	213.70	22.44	213.74		–
Ours	12.23	67.94	14.18	102.14	15.46	794.76

(b) WRN28-10

Method	CIFAR-10		CIFAR-100	
	Params (M)	Cost (PFLOPs)	Params (M)	Cost (PFLOPs)
PGD-AT	36.48	896.71	36.54	896.73
DAT	40.35	3661.72	40.43	3661.79
DHAT	76.24	2017.60	76.36	2017.64
Ours	38.13	640.74	41.05	961.45
ADBM	106.64	$2442.74 + C_{\text{score}}$		–

C_{score} denotes the computational cost of pretraining the score-SDE model required by ADBM, which is not included in the reported 2442.74 PFLOPs.

diffusion-based ADBM under adaptive attacks, while requiring markedly less training computation than DAT, DHAT, and ADBM in directly comparable settings. These results support decision-relevant feature purification as an effective and efficient way to strengthen AT.

ADDITIONAL STATEMENTS

(This section is **required/recommended** and does not count toward the page limit.)

AI USE STATEMENT

Generative AI tools were used to assist with language polishing, manuscript drafting and editing, and presentation and formatting. The research ideas, methodology, experimental design and execution, mathematical results, literature review and citation selection, and scientific conclusions were developed and finalized by the authors. All AI-assisted text and presentation materials were reviewed and revised by the authors. The authors take full responsibility for the final content of this work.

ETHICS STATEMENT

This work does not involve human subjects, personal or private data, or deployment in high-risk application domains. The study concerns adversarial robustness and uses standard benchmark datasets and adversarial attacks for defensive evaluation. We are not aware of additional ethical concerns specific to this work beyond the general dual-use considerations associated with adversarial machine learning research.

REPRODUCIBILITY STATEMENT

We provide methodological definitions and the training objective in the main text, complete theoretical assumptions, statements, and proofs in the main text and appendix, and detailed model architectures, training and inference settings, attack configurations, hyperparameters, evaluation protocols, and random seeds in the appendix. These details support independent reproduction of both the theoretical and empirical results. We did not include code with this submission; upon acceptance, we will release the implementation and add the corresponding link to the final version.

REFERENCES

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *ECCV*, pp. 484–501. Springer, 2020.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, pp. 274–283. PMLR, 2018.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. *NeurIPS*, 30, 2017.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57. Ieee, 2017.

Elias Collaert, Abel Rodríguez, Sander Joos, Lieven Desmet, and Vera Rimmer. Flowpure: Continuous normalizing flows for adversarial purification. *arXiv preprint arXiv:2505.13280*, 2025.

Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *ICML*, pp. 2196–2205. PMLR, 2020a.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, pp. 2206–2216. PMLR, 2020b.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pp. 248–255. IEEE, 2009.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *ICLR*, 2015.

Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J Cree. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110(2):393–416, 2021.

Simon Haykin. *Neural networks and learning machines, 3/E*. Pearson education india, 2009.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

Hoki Kim, Jinseong Park, Yujin Choi, and Jaewook Lee. Fantastic robustness measures: The secrets of robust generalization. *NeurIPS*, 36:48793–48818, 2023.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Minjong Lee and Dongwoo Kim. Robust evaluation of diffusion-based adversarial purification. In *ICCV*, pp. 134–144, 2023.

Fengpeng Li, Kemou Li, Haiwei Wu, Jinyu Tian, and Jiantao Zhou. Dat: Improving adversarial robustness via generative amplitude mix-up in frequency domain. *NeurIPS*, 37:127099–127128, 2024.

Xiao Li, Wenxuan Sun, Huanran Chen, Qiongxiu Li, Yining Liu, Yingzhe He, Jie Shi, and Xiaolin Hu. Adbm: Adversarial diffusion bridge model for reliable adversarial purification. *ICLR*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023.

Max Losch, Mohamed Omran, David Stutz, Mario Fritz, and Bernt Schiele. On adversarial training without perturbing all examples. In *ICLR*, 2024.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *ICLR*, 2018.

Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the ACM SIGSAC conference on computer and communications security (CCS)*, pp. 135–147, 2017.

Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. *ICML*, 2022.

Mincheol Park, Cheonjun Park, Seungseop Lim, Mijin Koo, Hyunwuk Lee, Won Woo Ro, and Suhyun Kim. Adversarial purification via super-resolution and diffusion. In *ICCV*, pp. 4605–4615, 2025.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, volume 32, 2018.

Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *ICML*, pp. 8093–8104. PMLR, 2020.

Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *ICLR*, 2018.

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *NeurIPS*, 32, 2019.

Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *ICLR*, 2018.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

Lloyd N. Trefethen and David Bau, III. *Numerical Linear Algebra*. SIAM, 1997.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *ICLR*, 2019.

Kaibo Wang, Xiaowen Fu, Yuxuan Han, and Yang Xiang. Diffhammer: Rethinking the robustness of diffusion-based adversarial purification. *NeurIPS*, 37:89535–89562, 2024.

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *ICLR*, 2020.

Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *ICLR*, 2020.

Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *CVPR*, pp. 501–509, 2019.

Jongmin Yoon, Sung Ju Hwang, and Juho Lee. Adversarial purification with score-based generative models. In *ICML*, pp. 12062–12072. PMLR, 2021.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *BMVC*, 2016.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, pp. 7472–7482. PMLR, 2019.

Kejia Zhang, Juanjuan Weng, Shaozi Li, and Zhiming Luo. Towards adversarial robustness via debiased high-confidence logit alignment. In *ICCV*, pp. 2783–2792, 2025a.

Mingkun Zhang, Keping Bi, Wei Chen, Jiafeng Guo, and Xueqi Cheng. Clipure: Purification in latent space via clip for adversarially robust zero-shot classification. In *ICLR*, 2025b.

Yi Zhang, Yuhang Chen, Zhen Chen, Wenjie Ruan, Xiaowei Huang, Siddartha Khastgir, and Xingyu Zhao. Adversarial training for probabilistic robustness. In *ICCV*, pp. 1675–1685, 2025c.

A Notation and Additional Setup

This appendix supplements Section 3. The main paper keeps only the notation needed to follow the method and central theoretical result; Table 5 collects the symbols used throughout the analysis and training objective.

Table 5: Notation used in the methodology and theoretical analysis.

Symbol	Meaning
$\mathbf{x}, y$	input and ground-truth class drawn from $\mathcal{D}$
$\mathbf{x}'$	perturbed input in $\mathcal{B}_\infty(\mathbf{x}, \varepsilon)$
α, T	PGD step size and number of attack iterations used to generate training AEs
θ	trainable parameters of the raw classifier f_θ
ϕ	trainable parameters of the class-conditional vector field v_ϕ
f_θ	raw classifier, decomposed as $g_\theta \circ h_\theta$
h_θ	feature encoder/backbone
g_θ	affine classification head
$\psi(\mathbf{x})$	raw predicted class $\arg\max_c f_\theta(\mathbf{x})_c$
v_ϕ	class-conditional feature-space vector field
y_{flow}	guidance class supplied to the vector field
Φ_ϕ	K-step Euler purification map
η, K	purification step size and number of Euler steps; selected at inference
$\eta_{\text{train}}, K_{\text{train}}$	fixed training-time purification step size and number of Euler steps
$\mathbf{c}_y$	analytical class-y reference used only in the local theory
L_h, L_g	local Lipschitz constants of the encoder and head
κ, M, R	inwardness, magnitude, and neighborhood constants in Assumption 3.2
ρ	one-step contraction factor in Eq. (12)
$\delta_{\text{clean}}(\mathbf{x})$	clean-feature distortion bound in Assumption 3.3
$m_y(\mathbf{z})$	class-y logit margin at feature $\mathbf{z}$
$\gamma(\mathbf{x})$	clean class-y margin $m_y(h_\theta(\mathbf{x}))$
$\Gamma_y(r)$	worst class-y margin over the radius-r feature ball around $\mathbf{c}_y$
$\mathbf{h}_{\text{clean}}, \mathbf{h}_{\text{adv}}$	clean and raw-adversarial features used in training
$\mathbf{h}_{\text{adv}}^{\text{pur}}$	purified adversarial feature

Training AEs are generated as in Eq. (19), with perturbation radius ε, PGD step size α, and T attack iterations; the values used in each experiment are reported in the experimental settings. The generic purification map in Eq. (4) permits any prescribed time sequence for which the local conditions hold. The theoretical statements assume Assumption 3.2 uniformly over the time values used by the discretization. The class reference $\mathbf{c}_y$ is an analytical device for describing a favorable local geometry and is not a learned prototype or an additional inference component.

B Complete Theoretical Results and Proofs

This section gives the intermediate results and complete proofs supporting Section 3.2. We first establish the contraction behavior of the Euler map, then propagate input perturbations through the encoder and purifier, and finally derive the margin statements used in Theorem 3.4 and Corollary 3.5.

B.1 Single-Step Contraction

Lemma B.1 (Single-step contraction around the class reference)**.** *Under Assumption 3.2, if* $\|\mathbf{z} - \mathbf{c}_y\|_2 \le R$*, then one Euler step*

$$\mathbf{z}^+ = \mathbf{z} + \eta v_\phi(t, \mathbf{z}, y) \tag{29}$$

satisfies

$$\left\|\mathbf{z}^+ - \mathbf{c}_y\right\|_2 \le \rho \left\|\mathbf{z} - \mathbf{c}_y\right\|_2, \tag{30}$$

where ρ *is defined in Eq.* (12)*.*

Proof. Fix y and a feature $\mathbf{z}$ with $\|\mathbf{z} - \mathbf{c}_y\|_2 \le R$. Expanding the squared distance after one Euler step gives

$$\left\|\mathbf{z}^+ - \mathbf{c}_y\right\|_2^2 = \|\mathbf{z} - \mathbf{c}_y + \eta v_\phi(t, \mathbf{z}, y)\|_2^2 \tag{31}$$

$$= \|\mathbf{z} - \mathbf{c}_y\|_2^2 + 2\eta \langle \mathbf{z} - \mathbf{c}_y, v_\phi(t, \mathbf{z}, y)\rangle + \eta^2 \|v_\phi(t, \mathbf{z}, y)\|_2^2 . \tag{32}$$

Assumption 3.2 provides the two bounds

$$\langle \mathbf{z} - \mathbf{c}_y, v_\phi(t, \mathbf{z}, y)\rangle \le -\kappa \|\mathbf{z} - \mathbf{c}_y\|_2^2 , \tag{33}$$

$$\|v_\phi(t, \mathbf{z}, y)\|_2^2 \le M^2 \|\mathbf{z} - \mathbf{c}_y\|_2^2 . \tag{34}$$

Substitution into Eq. (32) yields

$$\left\|\mathbf{z}^+ - \mathbf{c}_y\right\|_2^2 \le \left(1 - 2\eta\kappa + \eta^2 M^2\right) \|\mathbf{z} - \mathbf{c}_y\|_2^2 \tag{35}$$

$$= \rho^2 \|\mathbf{z} - \mathbf{c}_y\|_2^2 . \tag{36}$$

Both sides are nonnegative, so taking square roots gives the claimed contraction bound. Since $\rho < 1$, the next iterate also remains inside the radius-R neighborhood whenever the current iterate does. □

B.2 Multi-Step Contraction

Lemma B.2 (Multi-step contraction)**.** *Let $\mathbf{z}^{(0)}$ satisfy $\left\|\mathbf{z}^{(0)} - \mathbf{c}_y\right\|_2 \le R$. Under Assumption 3.2, the K Euler steps in Eq.* (4) *satisfy*

$$\left\|\mathbf{z}^{(K)} - \mathbf{c}_y\right\|_2 \le \rho^K \left\|\mathbf{z}^{(0)} - \mathbf{c}_y\right\|_2 . \tag{37}$$

Proof. Lemma B.1 gives

$$\left\|\mathbf{z}^{(1)} - \mathbf{c}_y\right\|_2 \le \rho \left\|\mathbf{z}^{(0)} - \mathbf{c}_y\right\|_2 \le \rho R < R. \tag{38}$$

Hence Assumption 3.2 remains applicable at the second step. Suppose inductively that

$$\left\|\mathbf{z}^{(k)} - \mathbf{c}_y\right\|_2 \le \rho^k \left\|\mathbf{z}^{(0)} - \mathbf{c}_y\right\|_2 \le R. \tag{39}$$

Applying Lemma B.1 once more gives

$$\left\|\mathbf{z}^{(k+1)} - \mathbf{c}_y\right\|_2 \le \rho \left\|\mathbf{z}^{(k)} - \mathbf{c}_y\right\|_2 \le \rho^{k+1} \left\|\mathbf{z}^{(0)} - \mathbf{c}_y\right\|_2 . \tag{40}$$

Thus the bound holds for every $k = 0, \dots, K$ by induction, and in particular for $k = K$. □

B.3 Perturbed Feature Radius After Purification

Lemma B.3 (Perturbed feature radius after purification)**.** *For a sample $(\mathbf{x}, y)$, define*

$$r_0(\mathbf{x}, \varepsilon) = L_h \varepsilon + \|h_\theta(\mathbf{x}) - \mathbf{c}_y\|_2 . \tag{41}$$

Under Assumptions 3.1 and 3.2, if $r_0(\mathbf{x}, \varepsilon) \le R$, then every $\mathbf{x}' \in \mathcal{B}_\infty(\mathbf{x}, \varepsilon)$ satisfies

$$\|\Phi_\phi(h_\theta(\mathbf{x}'), y; K, \eta) - \mathbf{c}_y\|_2 \le \rho^K r_0(\mathbf{x}, \varepsilon). \tag{42}$$

Proof. Take any $\mathbf{x}' \in \mathcal{B}_\infty(\mathbf{x}, \varepsilon)$. Assumption 3.1 gives

$$\|h_\theta(\mathbf{x}') - h_\theta(\mathbf{x})\|_2 \le L_h \|\mathbf{x}' - \mathbf{x}\|_\infty \le L_h \varepsilon. \tag{43}$$

By the triangle inequality,

$$\|h_\theta(\mathbf{x}') - \mathbf{c}_y\|_2 \le \|h_\theta(\mathbf{x}') - h_\theta(\mathbf{x})\|_2 + \|h_\theta(\mathbf{x}) - \mathbf{c}_y\|_2 \tag{44}$$

$$\le L_h \varepsilon + \|h_\theta(\mathbf{x}) - \mathbf{c}_y\|_2 \tag{45}$$

$$= r_0(\mathbf{x}, \varepsilon) \le R. \tag{46}$$

Therefore the entire purification trajectory starts inside the region where Assumption 3.2 applies. Lemma B.2 then yields

$$\|\Phi_\phi(h_\theta(\mathbf{x}'), y; K, \eta) - \mathbf{c}_y\|_2 \le \rho^K \|h_\theta(\mathbf{x}') - \mathbf{c}_y\|_2 \tag{47}$$

$$\le \rho^K r_0(\mathbf{x}, \varepsilon), \tag{48}$$

as claimed. □

B.4 Clean-Feature Margin Preservation

Lemma B.4 (Purified clean-margin lower bound)**.** *Under Assumptions 3.1 and 3.3, a clean sample with $\gamma(\mathbf{x}) = m_y(h_\theta(\mathbf{x})) > 0$ satisfies*

$$m_y(\Phi_\phi(h_\theta(\mathbf{x}), y; K, \eta)) \geq \gamma(\mathbf{x}) - 2L_g\delta_{\text{clean}}(\mathbf{x}). \tag{49}$$

Proof. Set $\mathbf{z}^\star = h_\theta(\mathbf{x})$ and $\tilde{\mathbf{z}}^\star = \Phi_\phi(\mathbf{z}^\star, y; K, \eta)$. By Assumption 3.3,

$$\|\tilde{\mathbf{z}}^\star - \mathbf{z}^\star\|_2 \leq \delta_{\text{clean}}(\mathbf{x}). \tag{50}$$

The head Lipschitz condition in Assumption 3.1 then gives

$$\|g_\theta(\tilde{\mathbf{z}}^\star) - g_\theta(\mathbf{z}^\star)\|_2 \leq L_g\delta_{\text{clean}}(\mathbf{x}). \tag{51}$$

For any class c, the absolute change of its logit is bounded by the Euclidean norm of the complete logit difference, so

$$|g_\theta(\tilde{\mathbf{z}}^\star)_c - g_\theta(\mathbf{z}^\star)_c| \leq L_g\delta_{\text{clean}}(\mathbf{x}). \tag{52}$$

Consequently, the true-class logit can decrease by at most $L_g\delta_{\text{clean}}(\mathbf{x})$ and the largest competing logit can increase by at most the same amount. Therefore

$$m_y(\tilde{\mathbf{z}}^\star) \geq m_y(\mathbf{z}^\star) - 2L_g\delta_{\text{clean}}(\mathbf{x}) \tag{53}$$

$$= \gamma(\mathbf{x}) - 2L_g\delta_{\text{clean}}(\mathbf{x}), \tag{54}$$

which proves the lemma. □

B.5 Proof of Theorem 3.4

Proof. Let $\mathbf{x}' \in \mathcal{B}_\infty(\mathbf{x}, \varepsilon)$ be arbitrary. By Lemma B.3,

$$\|\Phi_\phi(h_\theta(\mathbf{x}'), y; K, \eta) - \mathbf{c}_y\|_2 \leq r_K = \rho^K r_0(\mathbf{x}, \varepsilon). \tag{55}$$

Because $r_0(\mathbf{x}, \varepsilon) \leq R$ and $\rho < 1$, we also have $r_K \leq R$, so the regional margin profile in Eq. (14) is defined at r_K. By definition of the infimum,

$$m_y(\Phi_\phi(h_\theta(\mathbf{x}'), y; K, \eta)) \geq \Gamma_y(r_K). \tag{56}$$

If $\Gamma_y(r_K) > 0$, then the class-y logit is strictly larger than every competing logit, and hence

$$\arg\max_c F_{\theta,\phi}(\mathbf{x}', y)_c = y. \tag{57}$$

Since $\mathbf{x}'$ was arbitrary, the true-label-guided defended classifier is locally robust over the full perturbation ball.

For the clean input, Lemma B.4 gives Eq. (17). If $2L_g\delta_{\text{clean}}(\mathbf{x}) < \gamma(\mathbf{x})$, the resulting lower bound is positive and the clean class prediction is preserved. □

B.6 Proof of Corollary 3.5

Proof. The raw local-robustness premise gives $\psi(\mathbf{x}') = y$ for every $\mathbf{x}' \in \mathcal{B}_\infty(\mathbf{x}, \varepsilon)$. Therefore the practical purifier and the true-label-guided purifier coincide pointwise on the perturbation ball:

$$\Phi_\phi(h_\theta(\mathbf{x}'), \psi(\mathbf{x}'); K, \eta) = \Phi_\phi(h_\theta(\mathbf{x}'), y; K, \eta). \tag{58}$$

The robustness conclusion of Theorem 3.4 therefore applies directly whenever $\Gamma_y(r_K) > 0$.

For the regional margin bound, Eq. (12) gives $0 \leq \rho < 1$, and hence $r_K = \rho^K r_0 \leq r_0$. Thus

$$\{\mathbf{z} : \|\mathbf{z} - \mathbf{c}_y\|_2 \leq r_K\} \subseteq \{\mathbf{z} : \|\mathbf{z} - \mathbf{c}_y\|_2 \leq r_0\}. \tag{59}$$

Taking the infimum of the same margin function over the smaller set cannot decrease its value, which yields

$$\Gamma_y(r_K) \geq \Gamma_y(r_0). \tag{60}$$

If $\Gamma_y(r_K) > \Gamma_y(r_0)$, the regional margin lower bound is strictly tighter after contraction. This establishes the regional margin comparison in the corollary. □

C THEORY-TO-LEARNING CONNECTIONS

The theoretical conditions in Section 3.2 identify the local behaviors that motivate the practical objective in Section 3.3. This section provides the detailed analytical connections between those conditions and the corresponding learning components.

C.1 ADVERSARIAL CROSS-ENTROPY AND RAW-CLASSIFIER GUIDANCE

Lemma 3.6 links the AT objective to the positive raw margin required for reliable pseudo-label guidance.

Proof of Lemma 3.6. Let $\boldsymbol{\ell} = g_\theta(\mathbf{z})$. The cross-entropy loss for class y can be written as

$$\mathcal{L}_{\mathrm{CE}}(\boldsymbol{\ell}, y) = -\log \frac{e^{\boldsymbol{\ell}_y}}{\sum_j e^{\boldsymbol{\ell}_j}} \tag{61}$$

$$= \log\left(1 + \sum_{c \neq y} e^{\boldsymbol{\ell}_c - \boldsymbol{\ell}_y}\right). \tag{62}$$

If this quantity is at most τ, then

$$\sum_{c \neq y} e^{\boldsymbol{\ell}_c - \boldsymbol{\ell}_y} \leq e^\tau - 1. \tag{63}$$

In particular,

$$e^{\max_{c \neq y}(\boldsymbol{\ell}_c - \boldsymbol{\ell}_y)} \leq e^\tau - 1. \tag{64}$$

Taking logarithms and rearranging gives

$$\boldsymbol{\ell}_y - \max_{c \neq y} \boldsymbol{\ell}_c \geq -\log(e^\tau - 1). \tag{65}$$

When $\tau < \log 2$, we have $e^\tau - 1 < 1$, so the right-hand side is strictly positive. □

If the robust cross-entropy in Eq. (21) is bounded by the same $\tau < \log 2$, then the premise of Lemma 3.6 holds for every $\mathbf{x}' \in \mathcal{B}_\infty(\mathbf{x}, \varepsilon)$. Hence the raw classifier has a positive class-y margin throughout the ball and supplies the correct pseudo-label there. Since the inner maximization is approximated by PGD during training, this margin relation provides the analytical motivation for the raw adversarial loss.

C.2 INWARD STRUCTURE OF THE FLOW-MATCHING TARGET

For the straight-line conditional flow used in Eq. (24),

$$\mathbf{z}_t = (1-t)\mathbf{z}_0 + t\mathbf{h}_{\mathrm{clean}}, \qquad \mathbf{v}_{\mathrm{target}} = \mathbf{h}_{\mathrm{clean}} - \mathbf{z}_0. \tag{66}$$

Rearranging the first expression gives

$$\mathbf{h}_{\mathrm{clean}} - \mathbf{z}_t = (1-t)(\mathbf{h}_{\mathrm{clean}} - \mathbf{z}_0) = (1-t)\mathbf{v}_{\mathrm{target}}. \tag{67}$$

Hence, for $t < 1$,

$$\mathbf{v}_{\mathrm{target}} = \frac{\mathbf{h}_{\mathrm{clean}} - \mathbf{z}_t}{1-t}. \tag{68}$$

The numerator is the displacement from the current interpolated feature $\mathbf{z}_t$ to the clean endpoint, and the scalar $1/(1-t)$ is positive. Therefore $\mathbf{v}_{\mathrm{target}}$ points in exactly that direction. Equivalently,

$$\langle \mathbf{z}_t - \mathbf{h}_{\mathrm{clean}}, \mathbf{v}_{\mathrm{target}} \rangle = -\frac{\|\mathbf{z}_t - \mathbf{h}_{\mathrm{clean}}\|_2^2}{1-t} \leq 0, \tag{69}$$

which expresses the same inward geometry through a negative directional inner product. Because the vector field is conditioned on y, samples from the same class contribute corrective targets toward their respective clean features under a shared class condition. This provides the practical geometric link to the inward-pointing behavior in Assumption 3.2: the flow-matching objective uses sample-level clean endpoints, whereas the assumption describes a local class-reference geometry around $\mathbf{c}_y$.

C.3 Alignment and the Purified Margin

The alignment loss does not directly evaluate $\Phi_\phi(h_\theta(\mathbf{x}), y; K, \eta)$ on a clean starting feature, so it should not be interpreted as an exact enforcement of Assumption 3.3. Instead, it uses the clean feature as the endpoint anchor for the purified adversarial representation. The following bound explains why this is useful for the margin condition in Theorem 3.4.

Lemma C.1 (Margin bound from clean-feature alignment)**.** *Under the head Lipschitz condition of Assumption 3.1, let* $\mathbf{h}_{\text{clean}} = h_\theta(\mathbf{x})$ *with clean margin* $\gamma(\mathbf{x}) > 0$*. For any feature* $\tilde{\mathbf{h}}$ *satisfying*

$$\left\|\tilde{\mathbf{h}} - \mathbf{h}_{\text{clean}}\right\|_2 \leq \delta, \tag{70}$$

we have

$$m_y(\tilde{\mathbf{h}}) \geq \gamma(\mathbf{x}) - 2L_g\delta. \tag{71}$$

Proof. The proof is the same coordinate-wise Lipschitz argument used in Lemma B.4. Specifically,

$$\left\|g_\theta(\tilde{\mathbf{h}}) - g_\theta(\mathbf{h}_{\text{clean}})\right\|_2 \leq L_g\delta, \tag{72}$$

so the true-class logit can decrease by at most $L_g\delta$ and the largest competing logit can increase by at most $L_g\delta$. Subtracting the two bounds gives the result. □

Applying Lemma C.1 to $\tilde{\mathbf{h}} = \mathbf{h}_{\text{adv}}^{\text{pur}}$ shows that reducing the sample-level alignment error directly improves a lower bound on the purified class margin relative to its paired clean feature. The additional purified classification loss $\mathcal{L}_{\text{adv,pur}}$ acts on the same corrected feature and, through Lemma 3.6, provides a complementary direct pressure toward positive post-purification margins.

C.4 Parameter Norms as a Sensitivity Surrogate

The theoretical bounds contain three sensitivity-related quantities: L_h controls how far an input perturbation can move the initial feature, L_g controls how feature displacement changes logits, and M enters the contraction factor ρ. These quantities are local and are not directly optimized in our implementation. We therefore use parameter-norm regularization as a scalable surrogate.

The role of M should be considered jointly with the Euler step size. From Eq. (12), contraction requires

$$0 \leq 1 - 2\eta\kappa + \eta^2 M^2 < 1. \tag{73}$$

For fixed local geometry (κ, M), changing η changes this contraction factor. However, making η arbitrarily small is not by itself a desirable solution in the finite-step purifier because each Euler displacement satisfies

$$\left\|\mathbf{z}^{(k+1)} - \mathbf{z}^{(k)}\right\|_2 = \eta \left\|v_\phi(t_k, \mathbf{z}^{(k)}, y)\right\|_2. \tag{74}$$

Thus an excessively small η also weakens the corrective motion of each step, while $\rho \to 1$ as $\eta \to 0$. Controlling the vector-field magnitude and sensitivity therefore provides a complementary way to favor the contraction regime without relying solely on shrinking the step size.

For a feed-forward composition of Lipschitz nonlinearities and linear maps $W_1, \ldots, W_L$, a standard global upper bound on the network Lipschitz constant contains the product of the corresponding operator norms. For each matrix-valued linear map W, let $\|W\|_2$ denote its spectral (operator) norm and $\|W\|_F$ its Frobenius norm. These satisfy

$$\|W\|_2 \leq \|W\|_F. \tag{75}$$

Moreover, the squared ℓ_2 penalty on the entries of such a weight matrix is exactly its squared Frobenius norm, $\sum_{i,j} W_{ij}^2 = \|W\|_F^2$. Therefore, penalizing the entrywise squared norm provides a tractable way to discourage large Frobenius norms and, in turn, loose operator-norm upper bounds (Gouk et al., 2021). Applied to the classifier, this targets norm-based sensitivity bounds associated with L_h and L_g; applied to the vector field, it discourages large network amplification and velocity scale, which is the practical quantity associated with the magnitude term M in the contraction

condition. Such norm control also has a generalization-side motivation, since layer spectral norms enter margin-based generalization bounds for neural networks (Bartlett et al., 2017). We implement

$$\mathcal{L}_{\text{reg}} = \sum_{p \in \mathcal{P}} \|p\|_2^2 , \tag{76}$$

where $\mathcal{P}$ contains trainable weights of the classifier and vector field, excluding biases and one-dimensional normalization parameters. For matrix weights this term is the squared Frobenius norm; for other parameter tensors it is the analogous entrywise squared ℓ_2 penalty. Thus, the same scalable penalty is applied to both parts of the defended model according to the sensitivity quantities identified by the local analysis.

D Purifier Architecture Details

Figure 2 illustrates both vector-field parameterizations. We provide the corresponding implementation details below.

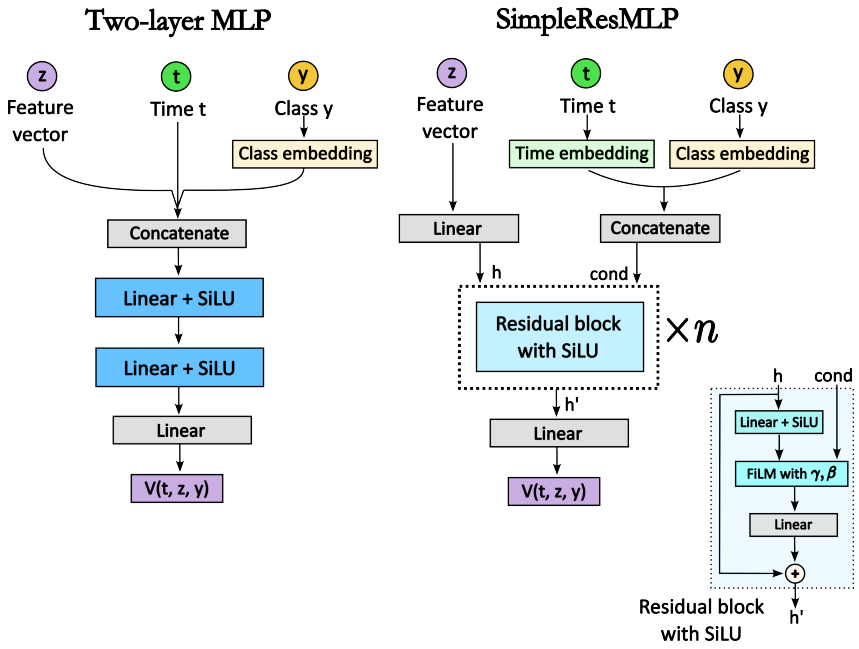


Figure 2: **Architectures of the class-conditional vector field. Left:** Two-layer MLP, which conditions the feature on time and class information through concatenation. **Right:** SimpleResMLP, which uses FiLM-conditioned residual blocks. Both lightweight networks parameterize the feature-space velocity $v_\phi(t, \mathbf{z}, y_{\text{flow}})$; detailed layer specifications and embedding dimensions are given below.

Two-layer MLP. For CIFAR-10, the feature $\mathbf{z}$, scalar time t, and a learned label embedding are concatenated and passed through two Linear+SiLU layers followed by a final linear projection,

$$v_\phi(t, \mathbf{z}, y_{\text{flow}}) = \text{MLP}([\mathbf{z} \,\|\, t \,\|\, e(y_{\text{flow}})]). \tag{77}$$

The output has the same dimension as the input feature.

SimpleResMLP. For CIFAR-100 and TinyImageNet, we first project $\mathbf{z}$ into a hidden space, apply n residual blocks, and project the result back to the feature dimension:

$$\mathbf{z} \xrightarrow{\text{Linear}} \mathbf{h} \xrightarrow{\text{ResBlocks}(\cdot\,; e(y_{\text{flow}}), \text{emb}(t))} \mathbf{h}' \xrightarrow{\text{Linear}} v_\phi(t, \mathbf{z}, y_{\text{flow}}). \tag{78}$$

Each residual block is conditioned with FiLM (Perez et al., 2018). The conditioning signal is formed from the concatenation of a learned class embedding and a time embedding. We use a

label-embedding dimension of 512 and a time-embedding dimension of 128. The number of residual blocks is $n = 2$ for CIFAR-100 and $n = 3$ for TinyImageNet.

Both parameterizations are intentionally small relative to image-space generative backbones because they operate only on pooled pre-logit features. Their measured parameter and computation costs are reported with the experimental efficiency results.

E Additional Training and Inference Details

E.1 Training Objective and Optimization

For completeness, the full objective is Eq. (28). Its fixed coefficients are

$$\lambda_{\mathrm{adv,raw}} = 1.0, \qquad \lambda_{\mathrm{adv,pur}} = 0.5, \qquad \lambda_{\mathrm{fm}} = 0.01, \qquad \lambda_{\mathrm{align}} = 0.1. \tag{79}$$

The value of λ_{reg} depends on the classifier, purifier, and dataset and is reported with the experimental settings. In addition to the explicit parameter penalty $\lambda_{\mathrm{reg}}\mathcal{L}_{\mathrm{reg}}$ motivated above, we update the classifier and vector field jointly using Adam with weight decay 5×10^{-4}; both forms of regularization are used in all experiments. For all experiments, the training-time purifier uses $K_{\mathrm{train}} = 2$ Euler steps. We set $\eta_{\mathrm{train}} = 0.05$ on CIFAR-10 and CIFAR-100, and $\eta_{\mathrm{train}} = 0.1$ on Tiny ImageNet.

During training, AEs are generated against the raw classifier using PGD with perturbation radius ε, step size α, and T attack iterations as defined in Eq. (19). The clean and adversarial features are then computed by the shared encoder, and the adversarial feature is purified using true-label guidance before evaluating $\mathcal{L}_{\mathrm{adv,pur}}$ and $\mathcal{L}_{\mathrm{align}}$. This ordering avoids differentiating through the purification iterations during every PGD step while still training the purifier on adversarial features produced by the current raw classifier.

E.2 Purification Hyperparameter and Checkpoint Selection

We select the inference pair (K, η) on a held-out validation split. Here K controls how many corrective Euler updates are accumulated, while η affects both the displacement of each update and the contraction factor in Eq. (12). Because these effects depend on the learned vector field and cannot be determined from η alone, we choose the operating point empirically on validation data by searching

$$K \in \mathcal{K}, \qquad \eta \in \mathcal{H}, \tag{80}$$

where the dataset-dependent candidate sets are

- CIFAR-10: $\mathcal{K} = \{1, 2, 3, 4, 5, 6, 8, 10\}$ and $\mathcal{H} = \{0.05, 0.1, 0.15, 0.2, 0.25, 0.3\}$;
- CIFAR-100: $\mathcal{K} = \{2, 3, 4, 6, 8, 10, 12, 14, 16\}$ and $\mathcal{H} = \{0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4\}$;
- TinyImageNet: $\mathcal{K} = \{2, 3, 4, 6, 8, 10, 12, 14, 16\}$ and $\mathcal{H} = \{0.1, 0.2, 0.3, 0.4, 0.5\}$.

The step-size candidates are multiples of the base value used during training.

The original training data are divided into a training portion and a held-out validation portion. After each epoch, the full defended pipeline is evaluated on the validation split under the same PGD setting used for training, and a checkpoint is saved whenever validation clean accuracy or validation robust accuracy improves. After training is complete, we use the last saved checkpoint and select (K, η) on the validation split by maximizing the sum of clean and PGD robust accuracy. The checkpoint and selected purification parameters are then fixed for final benchmarking. The test set is never used for training, checkpoint selection, or selection of purification hyperparameters.

F Additional Experimental Details

This section records implementation and evaluation details moved from the main paper for compactness.

F.1 Training Settings

We use ResNet-18 and WRN28-10 as described in Section 4.1. ATFeaturePure is trained for 100, 150, and 150 epochs on CIFAR-10, CIFAR-100, and Tiny ImageNet, respectively, with batch size 128. The initial learning rate is 10^{-3} and is decayed by a factor of 0.2 at epochs 60 and 80 on CIFAR-10 and at epochs 120 and 140 on CIFAR-100 and Tiny ImageNet. The optimizer and training-time purification settings are given in Appendix E.

The coefficient λ_{reg} is 2×10^{-5}, 2×10^{-5}, and 1.58×10^{-5} for ResNet-18 on CIFAR-10, CIFAR-100, and Tiny ImageNet, respectively, and 6.6×10^{-6} and 6×10^{-6} for WRN28-10 on CIFAR-10 and CIFAR-100. For ATFeaturePure, AEs used in training and validation are generated with $T=10$ PGD steps, matching the number of PGD steps used for the AT-based baselines in our experiments. Unless otherwise specified, the baselines follow the training and inference hyperparameters recommended in their original papers, and training, validation, and hyperparameter selection use each method's default AE generation protocol.

F.2 Attack Configurations

For PGD, PGD-20×3 denotes 20 iterations with 3 restarts and PGD-100×10 denotes 100 iterations with 10 restarts; these restarts are used only for final test-set evaluation.

For AutoAttack (Croce & Hein, 2020b), we use all available non-targeted components: APGD-CE, APGD-DLR, FAB (Croce & Hein, 2020a), and Square (Andriushchenko et al., 2020). In our reproduced experiments, this configuration yields consistently lower robust accuracy for both the baselines and ATFeaturePure than the default composition, so we use it as the stronger evaluation protocol. Following (Wang et al., 2024), DiffHammer uses the C&W (Carlini & Wagner, 2017), CE, and DLR objectives, and we report both its PGD and APGD modes. Following (Lee & Kim, 2023), for both ATFeaturePure and ADBM, attack gradients are computed by direct back-propagation through the complete purification process rather than using an adjoint-based numerical solver.

F.3 Random Seeds

The four-run results use seeds 19205881, 48046373, 1279194201, and 27837429. For each seed, all compared methods use the same train/validation split and randomized experimental pipeline.

F.4 Computing resources.

Experiments were conducted on three GPU servers equipped with $4\times$ RTX 2080 Ti, $4\times$ RTX 3090, and $2\times$ RTX 6000 Ada GPUs, respectively. The complete experimental campaign, including baseline reproduction, multi-seed runs, hyperparameter selection, and adaptive attacks, spanned approximately five months.