TIE: A TRAINING–INVERSION–EXCLUSION FRAMEWORK FOR VISUALLY INTERPRETABLE AND UNCERTAINTY-GUIDED OUT-OF-DISTRIBUTION DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep neural networks often struggle to recognize when an input lies outside their training experience, leading to unreliable and overconfident predictions. Building dependable machine learning systems therefore requires methods that can both estimate predictive *uncertainty* and detect *out-of-distribution (OOD)* samples in a unified manner. In this paper, we propose **TIE: a Training–Inversion–Exclusion** framework for visually interpretable and uncertainty-guided anomaly detection that jointly addresses these challenges through iterative refinement. TIE extends a standard n-class classifier to an $(n+1)$-class model by introducing a garbage class initialized with Gaussian noise to represent outlier inputs. Within each epoch, TIE performs a closed-loop process of *training, inversion, and exclusion*, where highly uncertain inverted samples reconstructed from the just-trained classifier are excluded into the garbage class. Over successive iterations, the inverted samples transition from noisy artifacts into visually coherent class prototypes, providing transparent insight into how the model organizes its learned manifolds. During inference, TIE rejects OOD inputs by either directly mapping them to the garbage class or producing low-confidence, uncertain misclassifications within the in-distribution classes that are easily separable, all without relying on external OOD datasets. A comprehensive threshold-based evaluation using multiple OOD metrics and performance measures such as *AUROC*, *AUPR*, and *FPR@95%TPR* demonstrates that TIE offers a unified framework for robust anomaly detection and uncertainty estimation (UE) achieving near-perfect OOD detection with $\approx$ **0 FPR@95%TPR** when trained on MNIST or FashionMNIST and tested against unseen datasets.

1 INTRODUCTION

The rapid integration of machine learning models into safety-critical domains such as autonomous driving, medical diagnosis, and financial decision-making has made *model reliability* and *robustness* indispensable. Despite remarkable progress in predictive accuracy, modern neural networks remain vulnerable to overconfidence, often producing high-confidence predictions even for inputs far removed from the training distribution Suhail & Sethi (2025). Such behavior compromises safety and trustworthiness, motivating the need for models that can not only recognize when they are uncertain but also identify when an input is fundamentally *out-of-distribution (OOD)*. The complementary tasks of **OOD detection**—discriminating in-distribution from anomalous inputs—and **uncertainty estimation (UE)**—quantifying predictive confidence—are therefore essential for building dependable, interpretable, and fail-safe learning systems.

OOD detection and UE have traditionally been treated as disjoint problems. Most existing methods rely on post-hoc calibration or require access to auxiliary OOD datasets for fine-tuning, which limits scalability and interpretability. Furthermore, post-hoc uncertainty estimates often fail to capture epistemic uncertainty, as they are derived from fixed, discriminative models without revisiting how the underlying decision boundaries are formed. Consequently, there is a need to integrate uncertainty into the learning process while maintaining interpretability without external supervision.

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.

In this paper, we introduce **TIE (Training–Inversion–Exclusion)**, a unified framework that jointly addresses uncertainty estimation and OOD detection through a self-refining training mechanism. TIE performs a cyclic process of *training, inversion, and exclusion*, where inverted reconstructions are assessed using a dynamic uncertainty threshold. Samples exhibiting high uncertainty are excluded into the garbage class, enabling the model to progressively purify its decision boundaries. Over successive iterations, the inverted samples evolve from noisy artifacts into visually coherent class prototypes, offering interpretable insights into the structure of each class manifold. During inference, TIE rejects OOD samples by assigning them either to the garbage class or to any of the in-distribution class with low confidence, making them easily separable by thresholding.

2 Related Work

Network Inversion. Inversion aims to reconstruct inputs that elicit desired outputs from a neural network. Early work on multilayer perceptrons employed gradients to do reconstructions, however these were often noisy in appearance Kindermann & Linden (1990); Jensen et al. (1999); Saad & Wunsch (2007). Later Wong (2017) explored evolutionary optimization and constrained search, while subsequent studies improved visual fidelity through explicit priors such as smoothness constraints or pretrained generative models Mahendran & Vedaldi (2014); Yosinski et al. (2015); Mordvintsev et al. (2015); Nguyen et al. (2016; 2017). The relationship between inversion and adversarial examples became evident, as unconstrained inversion frequently converged to adversarial artifacts Szegedy et al. (2014); Goodfellow et al. (2015). In contrast, adversarially robust networks were shown to yield more human-aligned reconstructions Tsipras et al. (2019); Engstrom et al. (2019); Santurkar et al. (2019). Recently surrogate loss learning for stable inversion Liu et al. (2022), generative modeling conditioned on classifier outputs Suhail & Sethi (2024), and logical formulations using CNF constraints Suhail (2024) for deterministic inversion have been explored.

Out-of-Distribution Detection. OOD detection focuses on identifying test inputs that lie outside the distribution of the training data. Hendrycks & Gimpel (2018) introduced the **Maximum Softmax Probability (MSP)** score as a simple yet effective measure, showing that correctly classified samples tend to yield higher softmax probabilities compared to misclassified or OOD inputs. Building upon this, Liang et al. (2020) proposed **ODIN**, which enhances separability between in- and out-of-distribution samples by applying temperature scaling and small input perturbations, significantly reducing false positives. Subsequently, Liu et al. (2021) reformulated OOD detection through an **energy-based** framework, demonstrating that energy scores aligned with the input's log-likelihood offer superior separation between in- and out-of-distribution data while addressing softmax overconfidence. In parallel, Lee et al. (2018) proposed the **Mahalanobis distance**-based approach, which models class-conditional feature distributions via Gaussian discriminant analysis to compute confidence scores, achieving strong performance on both adversarial and natural OOD detection tasks. Recent works like SCOOD Lu et al. (2023) enhance semantic coherence in OOD detection through uncertainty-aware optimal transport and adaptive cost modeling. Gaussian process-based techniques Chen et al. (2024) model predictive uncertainty using only in-distribution data, while normalizing flow–based models such as PostNet Charpentier et al. (2020) learn posteriors over predictive probabilities for reliable OOD discrimination.

Uncertainty Quantification(UQ). Post-hoc uncertainty quantification methods are widely used due to their compatibility with pretrained classifiers. Monte Carlo Dropout (MC Dropout) Gal & Ghahramani (2016) introduces stochastic inference to approximate Bayesian model averaging, while temperature scaling Guo et al. (2017) improves calibration via a single scalar parameter. Evidential Deep Learning Sensoy et al. (2018) models classification as a Dirichlet evidence estimation problem, while DEUP Jain et al. (2022) predicts generalization error using a learned uncertainty regressor and evidential meta-models Shen et al. (2023) generate Dirichlet parameters directly from classifier embeddings. Bayesian neural networks (BNNs) Neal (1996); Blundell et al. (2015) and variational inference approaches estimate posterior weight distributions, whereas Deep Ensembles Lakshminarayanan et al. (2017) combine multiple independently trained networks for superior calibration and robustness under shift. Domain-specific extensions include test-time augmentation, uncertainty-aware segmentation Jungo et al. (2020), and Bayesian approximations for volumetric imaging Kwon et al. (2020). Autoinverse Ansari et al. (2022) integrates predictive uncertainty into the inversion process, constraining reconstructions toward reliable training regions to improve robustness.

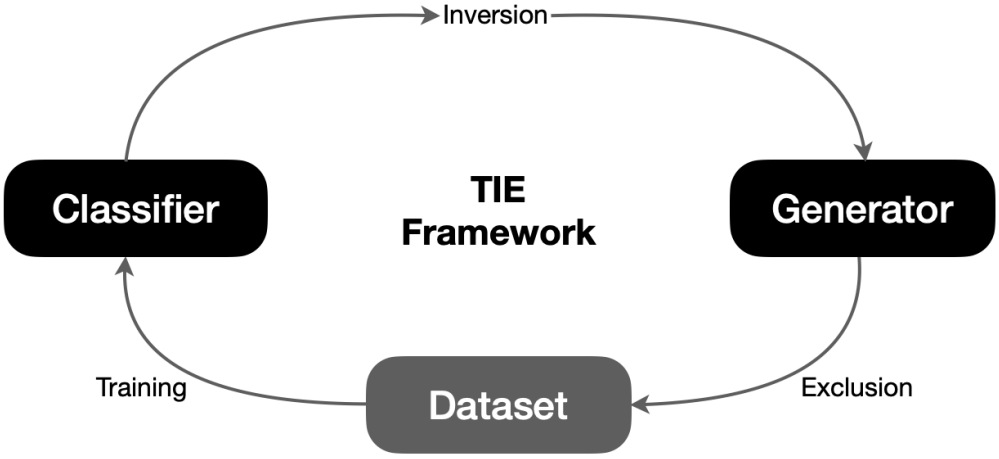

Figure 1: Overview of the proposed **TIE** framework.

3 Preliminaries – Network Inversion

The cornerstone of the **TIE** framework is the *inversion* (Suhail & Sethi, 2024) process that aims to reconstruct inputs that evoke specific outputs. In conventional formulations, inversion is applied *post hoc* on a fully trained classifier to analyze or reconstruct its training distribution. We, in contrast, embed the inversion process *within* the classifier training itself, allowing the inversion dynamics to co-evolve.

Given a classifier $f_\theta : \mathscr{X} \to \Delta^{K-1}$, where $\mathscr{X}$ is the input space and Δ^{K-1} is the $(K-1)$-dimensional probability simplex over class labels we train a conditional generator $\mathscr{G}_\phi : \mathscr{Z} \times \mathbb{R}^K \to \mathscr{X}$, parameterized by ϕ, to invert the classifier's behavior. Instead of conditioning the generator directly on a discrete class label $y \in \{1,\ldots,K\}$, we adopt a soft conditioning strategy based on sampled vectors. Specifically, in addition to the latent input $z \sim \mathscr{N}(0,I)$, we generate a conditioning vector $v \in \mathbb{R}^K$, where each component is independently sampled from a standard normal distribution:$v_k \sim \mathscr{N}(0,1), \quad \text{for } k = 1,\ldots,K$. The vector v is then transformed into a probability distribution $\tilde{y} \in \Delta^{K-1}$ using the softmax function.The generator thus receives the pair $(z,\tilde{y})$ and produces a synthetic input $\hat{x} = \mathscr{G}_\phi(z,\tilde{y})$ intended to be classified as $y = \arg\max_k \tilde{y}_k$.

Inversion aims to recover a plausible input $x \in \mathscr{X}$ that minimizes a label-consistent objective:$\hat{x} = \arg\min_{x\in\mathscr{X}} \mathscr{L}_{\text{inv}}(f_\theta(x),y)$, where $\mathscr{L}_{\text{inv}}$ encourages alignment between the classifier output and the target label, augmented with regularizers for realism or diversity. In our setup, this is realized by optimizing a conditional generator $\mathscr{G}_\phi(z,y)$ to minimize a composite loss from (Suhail & Sethi, 2024):

$$\mathscr{L}_{\text{Inv}} = \alpha \cdot \mathscr{L}_{\text{KL}} + \beta \cdot \mathscr{L}_{\text{CE}} + \gamma \cdot \mathscr{L}_{\text{Cosine}}$$

The **Cross-Entropy Loss** $\mathscr{L}_{\text{CE}}$ aligns generated images with their target classes, while the **KL Divergence** $\mathscr{L}_{\text{KL}}$ ensures that the classifier's output distribution matches the conditioning vector. The **Cosine Similarity Loss** $\mathscr{L}_{\text{Cosine}}$ enforces feature-level diversity across generated samples, mitigating collapse to overly prototypical representations, while the coefficients α,β,γ balance these objectives.

4 The TIE Framework

The proposed **TIE (Training–Inversion–Exclusion)** framework unifies OOD detection and UE into a single, interpretable, and self-refining learning paradigm as shown in Fig 1 and explained in Algorithm 1 provided in the Appendix. TIE is designed as an iterative process that couples classifier learning, network inversion, and uncertainty-guided exclusion, enabling the model to refine its decision boundaries and refine uncertainty over time.

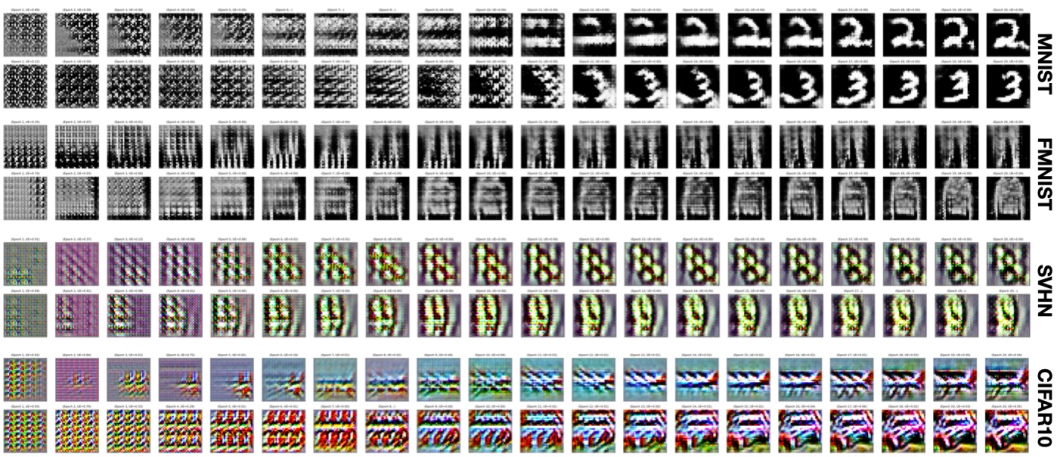

Figure 2: Evolution of inverted samples across epochs for select classes in the TIE framework.

Classifier Extension: For a standard n-class classification task, TIE extends the classifier $f_\theta : \mathscr{X} \to \Delta^n$ into an $(n+1)$-class formulation by introducing an auxiliary garbage class. This additional class acts as a receptacle for anomalous or uncertain samples that do not belong to any of the known in-distribution categories. At initialization, the garbage class is populated with random Gaussian noise, providing a background reference distribution that enables the classifier to learn a preliminary separation between structured and unstructured regions of the input space.

Embedded Inversion: Unlike conventional formulations where inversion is applied on a fully trained classifier, TIE embeds the inversion mechanism directly into the training process. A conditional generator $\mathscr{G}_\phi : \mathscr{Z} \times \mathbb{R}^{n+1} \to \mathscr{X}$ is co-trained with the classifier f_θ, forming a bi-directional loop where both networks evolve together. After each training epoch, $\mathscr{G}_\phi$ reconstructs representative samples corresponding to all output classes, guided by the inversion loss. These reconstructed samples approximate the classifier's current perception of each class manifold, reflecting how confidently the model represents its learned distribution.

Iterative Refinement Cycle: An epoch in TIE includes three stages—**Training**, **Inversion**, and **Exclusion**—that operate in a feedback loop as shown in Figure 1:

1. Training: The classifier f_θ is trained on the combined dataset containing both in-distribution samples and garbage samples using a weighted cross-entropy objective to correct for class imbalance caused by the addition of inverted samples into the garbage class.

2. Inversion: The generator $\mathscr{G}_\phi$ reconstructs input samples conditioned on each class's logits, optimizing the inversion loss $\mathscr{L}_{\text{Inv}}$ to reflect the current state of the classifier's learned manifold.

3. Exclusion: The reconstructed samples are evaluated for confidence and coherence using the classifier's predictive uncertainty. Samples exceeding a dynamic uncertainty threshold τ_t are assigned to the garbage class.

This refinement continues over successive epochs until the model learns to separate structured data from anomalous regions, visually through clearer reconstructions and probabilistically through better uncertainty estimates.

Uncertainty-Guided Exclusion: Uncertainty forms the backbone of the exclusion phase. After each epoch, the predictive uncertainty for all training samples is computed using the classifier's softmax distribution as

$$\text{UE}(\mathbf{p}) = 1 - \frac{\sum_{i=1}^{n+1} \left(p_i - \frac{1}{n+1}\right)^2}{\sum_{i=1}^{n+1} \left(\delta_{i,k} - \frac{1}{n+1}\right)^2},$$

where $k = \arg\max_i p_i$ and $\delta_{i,k}$ denotes the Kronecker delta. This normalized deviation from uniformity quantifies the model's confidence: lower values indicate sharp, confident predictions, while higher values signal uncertainty.

The uncertainty scores of the *training data* are aggregated at each epoch to compute a dynamic exclusion threshold, $\tau_t = \mu_t + \lambda\sigma_t$, where μ_t and σ_t represent the mean and standard deviation of the training uncertainties at epoch t, and λ is a scaling parameter that controls the strictness of the exclusion. As the classifier becomes progressively more confident on in-distribution samples, leading to a natural decline in uncertainty and a correspondingly tighter threshold τ_t. This threshold is then applied to the uncertainties of the *inverted samples* to determine which reconstructions are unreliable. Samples with $u_i > \tau_t$ are considered incoherent or anomalous and are assigned to the garbage class for the next training cycle. This evolving threshold ensures that exclusion criteria adapt automatically to the classifier's calibration, progressively filtering out uncertain reconstructions.

Inference and Evaluation: Once training converges, the TIE framework performs joint **OOD detection** and **UE** during inference. The inclusion of the garbage class enables TIE to perform two-tier OOD detection for both coarse- and fine-grained anomalies while maintaining high in-distribution performance.

1. Direct OOD Detection into Garbage Class. At this first level, the model's $(n+1)$-class formulation allows most OOD inputs to be *directly classified into the garbage class*, enabling coarse anomaly rejection without post-hoc calibration. The goal here is to ensure that the classifier maintains strong predictive accuracy on in-distribution data while simultaneously recognizing clear OOD samples. This stage is evaluated using **predictive accuracy** over both in- and out-of-distribution sets: the model is trained on one dataset and tested on others treated as OOD sources.

2. Threshold-Based Fine-Grained OOD Detection. Some OOD samples lie close to class boundaries and may be misclassified into in-distribution categories with relatively low confidence. To detect such subtle anomalies, TIE employs a threshold-based analysis using multiple **OOD metrics**—including *UE* [4], *ODIN* Liang et al. (2020), *Energy* Liu et al. (2021), and *Mahalanobis distance* Lee et al. (2018). Performance is quantified using **AUROC**, **AUPR**, and **FPR@95%TPR**, which together assess how well the model distinguishes low-confidence OOD predictions from confidently classified in-distribution samples.

5 Qualitative Analysis of TIE

To demonstrate the progressive refinement achieved by the proposed **TIE** framework, we present a qualitative analysis of how the reconstructed samples, associated metrics and internal representations evolve during training.

Visual Evolution of Inverted Samples: Figure 2 shows representative reconstructions for select classes capturing how the network's understanding of class manifolds emerges and stabilizes through training. In early epochs, the inverted samples appear noisy and unstructured, reflecting high uncertainty and weak discriminative capacity. As training progresses through the iterative *TIE* cycle, the classifier and generator co-evolve: the inverted reconstructions become increasingly representative of their true classes.

In later epochs, while the in-distribution classes converge toward distinct, interpretable prototypes, the garbage class accumulates incoherent or ambiguous reconstructions, forming a visual boundary that isolates structured and unstructured regions in the input space. This process mirrors the classifier's decreasing uncertainty and the progressive tightening of the dynamic exclusion threshold.

Quantitative Evolution of Inverted Samples: To complement the visual analysis, we evaluate how key predictive metrics evolve across epochs for the inverted samples of MNIST. Figure 3 presents the averaged values of *entropy*, *confidence*, *margin*, and *uncertainty estimate* for all $(n+1)$-classes. At the onset of training, inverted samples display high entropy and uncertainty, low confidence, and small top-1 margins—consistent with their incoherent appearance. As TIE advances, these trends systematically reverse: entropy and uncertainty decrease, while confidence and margin increase.

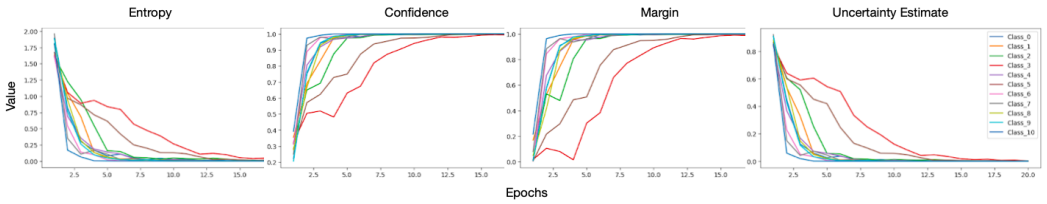

Figure 3: Evolution of averaged metrics associated with the inverted samples across epochs for all classes in MNIST.

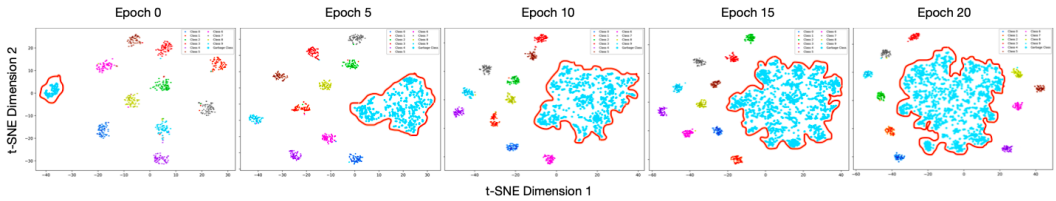

Figure 4: t-SNE visualization of features in the latent space across epochs for all $(n+1)$ classes of MNIST shows compact in-distribution clusters surrounding the expanding garbage class region.

This evolution indicates that the classifier's predictions become sharper and more discriminative as the generator learns to produce increasingly class-consistent inversions. By later epochs, all metrics stabilize, confirming convergence of the inversion process and a consistent separation between in-distribution and garbage samples. These quantitative trends align closely with the qualitative patterns observed in Figure 2, affirming that uncertainty-guided exclusion drives both structural clarity and probabilistic calibration in reconstructed samples.

Feature Evolution in Latent Space: To interpret how TIE organizes the internal representations of the classifier, we visualize the latent feature embeddings using t-SNE projections across epochs. Figure 4 depicts the penultimate-layer embeddings of the classifier for all $(n+1)$ classes on MNIST. As the model undergoes successive *Training–Inversion–Exclusion* cycles, the in-distribution classes progressively form compact and well-separated clusters, while the garbage class expands to occupy a central stage. This expansion visually reflects the exclusion mechanism, wherein uncertain or incoherent inverted samples are reassigned to the garbage class, which effectively acts as a buffer zone in latent space.

6 Quantitative Results and Comparisons

We evaluate the proposed **TIE** framework on six benchmark datasets—MNIST (Deng, 2012), FashionMNIST (Xiao et al., 2017), SVHN, CIFAR-10 (Krizhevsky et al.), CIFAR-100, and TinyImageNet-200—under a one-vs-rest protocol, where the model is trained using TIE Framework solely on one dataset, treated as in-distribution (ID) while the others act as out-of-distribution (OOD) sources.

6.1 Experimental Setup

For MNIST and FashionMNIST, we employ a custom CNN comprising three convolutional and two fully connected layers, each followed by Batch Normalization and ReLU activations. For SVHN and CIFAR datasets, a CNN with five convolutional and two fully connected layers is used to accommodate higher visual complexity. While for CIFAR-100 & TinyImageNet-200 deeper custom made CNNs with residual connections are used. The conditioned generator $\mathscr{G}_\phi$ is initialized with normally distributed weights and constructed using 4–6 transposed convolutional stages to synthesize images of the corresponding dataset resolution.

Table 1: Classification accuracy (%) for ID (diagonal) and OOD detection (off-diagonal) across all dataset pairs. Each cell includes **TIE / No TIE**, where TIE corresponds to the full Training–Inversion–Exclusion cycle, and No TIE to a baseline mode with static garbage class.

Train \ Test	MNIST	FMNIST	SVHN	CIFAR-10	CIFAR-100	TinyImageNet-200
MNIST	**99.1 / 97.5**	89.5 / 23.4	99.1 / 31.7	99.4 / 36.2	98.6 / 28.9	98.3 / 25.6
FMNIST	85.2 / 37.8	**92.6 / 91.3**	96.3 / 39.4	95.7 / 35.7	94.8 / 30.8	93.5 / 27.2
SVHN	93.6 / 40.5	94.9 / 23.1	**89.4 / 87.9**	87.6 / 38.4	86.9 / 36.1	84.2 / 33.7
CIFAR-10	97.8 / 46.9	95.7 / 36.8	88.2 / 25.1	**85.5 / 84.7**	83.6 / 39.2	81.4 / 37.0
CIFAR-100	98.1 / 35.8	96.8 / 22.7	88.9 / 38.3	80.1 / 35.6	**78.1 / 77.5**	83.1 / 30.8
TinyImageNet-200	94.2 / 32.5	91.9 / 28.4	87.4 / 37.6	82.9 / 36.1	81.7 / 24.5	**65.7 / 59.5**

Table 2: FPR@95%TPR (%) across dataset pairs reported as (UE / ODIN) on the first line and (Energy / Mahalanobis) on the second line.

Train \ Test	MNIST	FMNIST	SVHN	CIFAR-10	CIFAR-100	TinyImageNet-200
MNIST	–	7.47 / 5.99 2.86 / **0.83**	45.50 / 33.75 22.47 / **0.04**	11.37 / 6.58 1.99 / **0.00**	14.30 / 9.67 3.43 / **0.00**	12.04 / 7.88 2.54 / **0.01**
FMNIST	66.77 / 45.65 51.21 / **8.81**	–	13.22 / 1.95 **0.90** / 3.98	19.71 / 4.89 1.44 / **0.39**	20.53 / 5.05 1.37 / **0.52**	24.24 / 5.74 2.44 / **1.41**
SVHN	96.24 / 97.82 99.08 / **4.72**	**31.86** / 35.73 41.99 / 37.85	–	15.15 / **12.59** 20.17 / 79.29	16.98 / **15.68** 23.20 / 78.35	16.47 / **14.10** 20.56 / 80.86
CIFAR-10	77.95 / 75.18 77.60 / **12.16**	81.96 / 70.57 72.75 / **53.14**	60.79 / 63.06 **57.58** / 90.56	–	66.32 / **66.19** 67.12 / 94.45	67.74 / **66.63** 67.08 / 94.76
CIFAR-100	72.35 / 68.42 64.38 / **18.3**	75.91 / 70.16 73.27 / **45.8**	78.44 / 73.02 **22.71** / 26.8	66.87 / **19.2** 64.11 / 66.2	–	69.54 / **64.07** 73.85 / 88.3
TinyImageNet-200	71.92 / 74.65 72.44 / **21.8**	68.11 / 77.03 74.10 / **42.6**	65.47 / 60.64 **25.36** / 46.3	64.20 / **20.51** 63.18 / 63.5	76.88 / **63.27** 71.73 / 74.6	–

Both the classifier and generator are optimized using Adam, with learning rates of 1×10^{-4} and 1×10^{-3}, respectively. The slower classifier learning rate allows the generator to co-evolve and stabilize inversion dynamics. Training is performed for 20 epochs; in each epoch, approximately 1000 samples per class are added to the garbage class using a dynamic thresholding factor $\lambda = 0.5$. The composite inversion loss employs weighting coefficients α, β, and γ of 0.1, 1, and 10 for the KLD, cross-entropy, and cosine-similarity terms, respectively. γ is progressively increased (up to 100) to encourage wider input-space exploration.

6.2 OOD Detection into Garbage Class

At the first evaluation level, we assess the model's ability to directly classify out-of-distribution inputs into the garbage class while maintaining strong in-distribution accuracy. Table 1 reports both in-distribution (diagonal) and cross-domain OOD (off-diagonal) accuracies for all dataset pairs. High diagonal scores indicate that TIE retains excellent ID performance, while strong off-diagonal accuracies confirm its capacity to isolate anomalous inputs. These results validate the joint learning of the classifier and generator, which allows the system to simultaneously preserve discriminative structure and enforce exclusion.

Overall, TIE demonstrates excellent in-distribution classification accuracy on simpler datasets such as MNIST and FashionMNIST, achieving over 99% and 92% respectively. However, cross-domain evaluation reveals interesting trends: MNIST performs slightly worse on FashionMNIST owing to their shared grayscale characteristics. Conversely, SVHN struggles when evaluated against MNIST, as both represent digit domains with differing statistical structures. Since all models are trained from scratch without pretrained backbones, the in-distribution performance on complex datasets such as CIFAR-100 and TinyImageNet remains lower, yet even these models exhibit robust OOD discrimination when evaluated against other datasets.

Baseline Comparisons: We compare TIE against a baseline $(n+1)$-class classifier trained with a static garbage class. Removing TIE results in a dramatic collapse in OOD recognition accuracy by **50–70%** as shown in Table 1 with minimal impact on in-distribution performance. TIE's dynamic refinement cycle, powered by inversion and adaptive thresholding, proves essential for carving out meaningful decision boundaries across both ID and OOD regions.

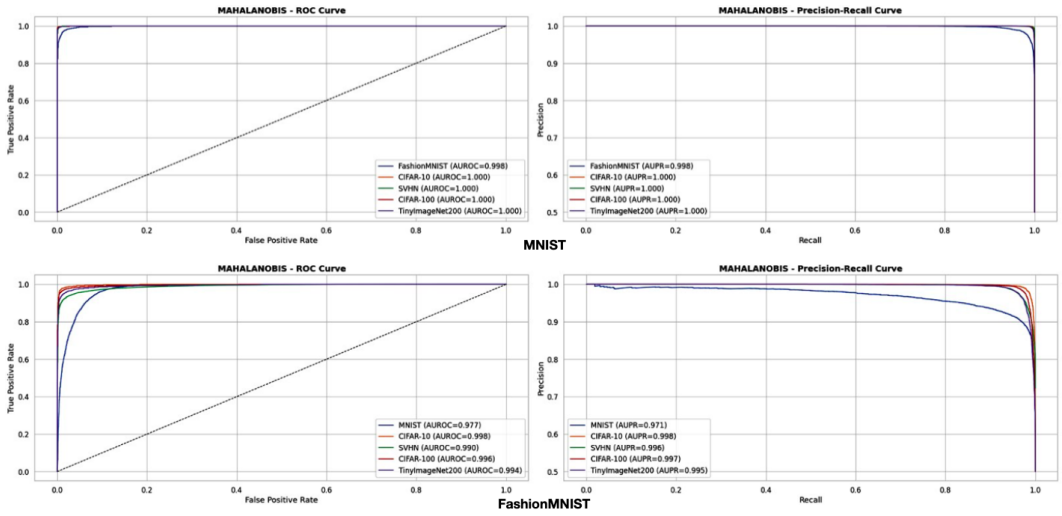

Figure 5: **Threshold-based OOD Detection on MNIST(Top) & FashionMNIST(Bottom).** AUROC and AUPR curves for Mahalanobis distance across multiple OOD datasets.

6.3 Threshold-Based OOD Detection

While most OOD samples are directly assigned to the garbage class during inference, a subset of ambiguous inputs may still be misclassified into in-distribution classes with relatively low confidence and high uncertainty. This evaluation assesses TIE's ability to discriminate such borderline cases. We discard the $(n{+}1)^{\text{th}}$ garbage-class logits and normalize the remaining n outputs.

Metric Comparisons: Table 2 compares *UE*, *ODIN*, *Energy*, and *Mahalanobis Distance* using FPR@95%TPR across all dataset pairs. Since this evaluation considers only OOD samples misclassified into in-distribution classes, AUROC and AUPR are expected to be lower, and FPR@95%TPR higher, than in standard full-dataset OOD evaluation.

Across MNIST and FashionMNIST, Mahalanobis-based thresholding yields strong fine-grained OOD discrimination, as shown in Figure 5. For MNIST, AUROC and AUPR are 0.998 for FashionMNIST and 1 for all other datasets, with an average FPR@95%TPR of 0.00176. For FashionMNIST, the average AUROC, AUPR, and FPR@95%TPR are 0.9911, 0.9916, and 0.0302, respectively.

6.4 Overall FPR after Two-Tier OOD Detection

The threshold-based evaluation in Table 2 considers only OOD samples not directly assigned to the garbage class. We therefore compute the overall false-positive rate of the two-tier detector by combining direct garbage-class rejection with a fixed Mahalanobis-based second stage. Let A_{OOD} denote the percentage of OOD samples assigned to the garbage class (Table 1), and FPR_{Maha} the FPR@95%TPR of Mahalanobis distance on the remaining OOD samples (Table 2). The overall FPR is

$$\text{FPR}_{\text{Overall}} = \left(1 - \frac{A_{\text{OOD}}}{100}\right) \times \text{FPR}_{\text{Maha}}. \tag{1}$$

Table 3 reports the resulting values across all dataset pairs. TIE achieves low overall FPR on simpler dataset pairs, where garbage-class rejection removes most OOD samples, while higher FPRs on CIFAR-10, CIFAR-100, and TinyImageNet reflect the greater difficulty of separating the remaining OOD samples.

Table 3: Overall false-positive rate (%) after the two-tier TIE inference pipeline using Mahalanobis distance consistently in the second stage. Lower is better.

Train	MNIST	FashionMNIST	SVHN	CIFAR-10	CIFAR-100	TinyImageNet
MNIST	–	0.00	0.00	0.00	0.01	0.02
FashionMNIST	1.30	–	0.15	0.02	0.03	0.09
SVHN	0.30	1.93	–	9.83	10.26	12.78
CIFAR-10	0.27	2.29	10.69	–	15.49	17.63
CIFAR-100	0.35	1.47	2.97	13.17	–	14.92
TinyImageNet	1.26	3.45	5.83	10.86	13.65	–

6.5 Comparison with Synthetic OOD Generation Methods

We compare TIE with representative synthetic OOD generation methods, including VOS Du et al. (2022), NPOS Tao et al. (2023), and DREAM-OOD Du et al. (2023), under the CIFAR-10 in-distribution setting. Unlike these approaches, TIE obtains synthetic negatives through uncertainty-guided network inversion and incorporates them into an explicit garbage class during training. As shown in Table 4, for TIE we report the overall FPR of the complete two-tier detector, combining direct garbage-class rejection with Mahalanobis-based detection of the remaining OOD samples. TIE achieves an average FPR of 7.59%, compared with 27.88%, 10.16%, and 7.46% for VOS, NPOS, and DREAM-OOD, respectively. Notably, TIE achieves the lowest FPR on three of the five OOD datasets—Places365, iSUN, and Textures—while remaining competitive in average FPR.

Table 4: Comparison with synthetic OOD generation methods using CIFAR-10 as the in-distribution dataset. FPR (%) is reported; lower is better.

Method	SVHN	Places365	LSUN	iSUN	Textures	Average
VOS	15.69	27.64	30.42	32.68	37.95	27.88
NPOS	**5.61**	4.08	14.13	8.39	18.57	10.16
DREAM-OOD	10.23	3.01	**2.76**	9.05	12.24	**7.46**
TIE (Ours)	10.69	**2.39**	7.48	**8.74**	**8.63**	7.59

6.6 Convergence and Sensitivity Analysis

TIE exhibits stable convergence, with entropy and uncertainty decreasing and confidence and prediction margin increasing until stabilizing after around 15 epochs. Reducing the exclusion threshold factor λ from 0.5 to 0.25 lowers the exclusion threshold and assigns more uncertain reconstructions to the garbage class, whereas increasing it to 0.75 produces a less aggressive exclusion strategy. Empirically, $\lambda = 0.5$ provides a favorable balance between retaining informative reconstructions and excluding uncertain samples. We also vary the number of reconstructed samples added to the garbage class per epoch from 500, to 2000 per class. Using 500 samples provides limited coverage of the input space, while 2000 introduces repetitive samples with only marginal gains. 1000 samples per class provide a trade-off between sample diversity, efficiency, and OOD detection performance.

7 Conclusion and Future Work

In this paper, we introduced **TIE** as a unified framework for visually interpretable and uncertainty-aware out-of-distribution detection by embedding inversion into the classifier training process, coupled with dynamic uncertainty-based exclusion. Across diverse ID–OOD settings, TIE demonstrates effective OOD rejection while preserving in-distribution classification performance, with the evolving inversions providing a visual interpretation of the learned rejection process. Future work can decompose the global garbage class into $n+n$ separate garbage classes—one corresponding to each in-distribution category—for fine-grained OOD detection, reducing dependence on threshold-based post-processing.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and improving the clarity and presentation of the manuscript. They were not used to generate experimental results or replace the authors' scientific judgment. All AI-assisted text was reviewed and revised by the authors, who take full responsibility for the methodology, experiments, results, claims, and final content of the paper.

ETHICS STATEMENT

This work does not involve human participants, personally identifiable information, or the collection of sensitive data. The experiments use standard publicly available image classification benchmarks and focus on out-of-distribution detection and uncertainty estimation. The proposed method is intended to improve the reliability of machine learning systems by enabling models to better identify inputs outside their training distribution.

REPRODUCIBILITY STATEMENT

We have taken several steps to facilitate reproducibility of this work. The TIE framework, including its Training–Inversion–Exclusion procedure and uncertainty-guided exclusion mechanism, is described in Sections 3 and 4 and summarized in Algorithm 1 in the Appendix. The datasets, model architectures, optimization settings, inversion-loss coefficients, exclusion threshold, number of generated samples, and training duration are described in Section 6.1. Additional convergence and sensitivity analyses are provided in Section 6.5. The accompanying code and supplementary materials provide the implementation and experimental configurations necessary to reproduce the reported experiments.

REFERENCES

Navid Ansari, Hans-Peter Seidel, Nima Vahidi Ferdowsi, and Vahid Babaei. Autoinverse: Uncertainty aware inversion of neural networks. In *Advances in Neural Information Processing Systems(NeurIPS)*, 2022.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1613–1622, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/blundell15.html.

Bertrand Charpentier, Daniel Zügner, and Stephan Günnemann. Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Yang Chen, Chih-Li Sung, Arpan Kusari, Xiaoyang Song, and Wenbo Sun. Uncertainty-aware out-of-distribution detection with gaussian processes. *arXiv:2412.20918*, 2024.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Xuefeng Du, Zhaoning Wang, Mu Cai, and Yixuan Li. Vos: Learning what you don't know by virtual outlier synthesis, 2022. URL https://arxiv.org/abs/2202.01197.

Xuefeng Du, Yiyou Sun, Xiaojin Zhu, and Yixuan Li. Dream the impossible: Outlier imagination with diffusion models, 2023. URL https://arxiv.org/abs/2309.13415.

Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. A rotation and a translation suffice: Fooling CNNs with simple transformations, 2019. URL https://openreview.net/forum?id=BJfvknCqFQ.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings*

of The 33rd International Conference on Machine Learning, volume 48 of *Proceedings of Machine Learning Research*, pp. 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/gal16.html.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015. URL https://arxiv.org/abs/1412.6572.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. *CoRR*, abs/1706.04599, 2017. URL http://arxiv.org/abs/1706.04599.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks, 2018. URL https://arxiv.org/abs/1610.02136.

Moksh Jain, Salem Lahlou, Hadi Nekoei, Victor I Butoi, Paul Bertin, Jarrid Rector-Brooks, Maksym Korablyov, and Yoshua Bengio. DEUP: Direct epistemic uncertainty prediction, 2022. URL https://openreview.net/forum?id=Jep2ykGUdS.

C.A. Jensen, R.D. Reed, R.J. Marks, M.A. El-Sharkawi, Jae-Byung Jung, R.T. Miyamoto, G.M. Anderson, and C.J. Eggen. Inversion of feedforward neural networks: algorithms and applications. *Proceedings of the IEEE*, 87(9):1536–1549, 1999. doi: 10.1109/5.784232.

Alain Jungo, Fabian Balsiger, and Mauricio Reyes. Analyzing the quality and challenges of uncertainty estimations for brain tumor segmentation. *Frontiers in Neuroscience*, 14:282, 2020. ISSN 1662-453X. doi: 10.3389/fnins.2020.00282. URL https://doi.org/10.3389/fnins.2020.00282.

J Kindermann and A Linden. Inversion of neural networks by gradient descent. *Parallel Computing*, 14(3):277–286, 1990. ISSN 0167-8191. doi: https://doi.org/10.1016/0167-8191(90)90081-J. URL https://www.sciencedirect.com/science/article/pii/016781919090081J.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). URL http://www.cs.toronto.edu/~kriz/cifar.html.

Yongchan Kwon, Joong-Ho Won, Beom Joon Kim, and Myunghee Cho Paik. Uncertainty quantification using bayesian neural networks in classification: Application to biomedical image segmentation. *Comput. Stat. Data Anal.*, 142(C), February 2020. ISSN 0167-9473. doi: 10.1016/j.csda.2019.106816. URL https://doi.org/10.1016/j.csda.2019.106816.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6405–6416, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks, 2018. URL https://arxiv.org/abs/1807.03888.

Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks, 2020. URL https://arxiv.org/abs/1706.02690.

Ruoshi Liu, Chengzhi Mao, Purva Tendulkar, Hao Wang, and Carl Vondrick. Landscape learning for neural network inversion, 2022. URL https://arxiv.org/abs/2206.09027.

Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. Energy-based out-of-distribution detection, 2021. URL https://arxiv.org/abs/2010.03759.

Fan Lu, Kai Zhu, Wei Zhai, Kecheng Zheng, and Yang Cao. Uncertainty-aware optimal transport for semantically coherent out-of-distribution detection. In *Computer Vision and Pattern Recognition (CVPR)*, 2023.

Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them, 2014. URL https://arxiv.org/abs/1412.0035.

Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks, 2015. URL https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html.

Radford M. Neal. *Bayesian Learning for Neural Networks*, volume 118 of *Lecture Notes in Statistics*. Springer, New York, NY, 1 edition, 1996. ISBN 978-0-387-94724-2. doi: 10.1007/978-1-4612-0745-0.

Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks, 2016. URL https://arxiv.org/abs/1605.09304.

Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. Plug & play generative networks: Conditional iterative generation of images in latent space, 2017. URL https://arxiv.org/abs/1612.00005.

Emad W. Saad and Donald C. Wunsch. Neural network explanation using inversion. *Neural Networks*, 20(1):78–93, 2007. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2006.07.005. URL https://www.sciencedirect.com/science/article/pii/S0893608006001730.

Shibani Santurkar, Dimitris Tsipras, Brandon Tran, Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Image synthesis with a single (robust) classifier, 2019. URL https://arxiv.org/abs/1906.09453.

Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty, 2018. URL https://arxiv.org/abs/1806.01768.

Maohao Shen, Yuheng Bu, Prasanna Sattigeri, Soumya Ghosh, Subhro Das, and Gregory Wornell. Post-hoc uncertainty learning using a dirichlet meta-model. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i8.26167. URL https://doi.org/10.1609/aaai.v37i8.26167.

Pirzada Suhail. Network inversion of binarised neural nets. In *The Second Tiny Papers Track at ICLR 2024*, 2024. URL https://openreview.net/forum?id=zKcB0vb7qd.

Pirzada Suhail and Amit Sethi. Network inversion of convolutional neural nets. In *Muslims in ML Workshop co-located with NeurIPS 2024*, 2024. URL https://openreview.net/forum?id=f9sUu7U1Cp.

Pirzada Suhail and Amit Sethi. Network inversion for generating confidently classified counterfeits, 2025. URL https://arxiv.org/abs/2503.20187.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014. URL https://arxiv.org/abs/1312.6199.

Leitian Tao, Xuefeng Du, Xiaojin Zhu, and Yixuan Li. Non-parametric outlier synthesis, 2023. URL https://arxiv.org/abs/2303.02966.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy, 2019. URL https://arxiv.org/abs/1805.12152.

Eric Wong. Neural network inversion beyond gradient descent. In *WOML NIPS*, 2017. URL https://api.semanticscholar.org/CorpusID:208231247.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. URL https://arxiv.org/abs/1708.07747.

Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization, 2015. URL https://arxiv.org/abs/1506.06579.

A TIE ALGORITHM

Algorithm 1 TIE: Training–Inversion–and–Exclusion

Require: Training data $\mathscr{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N}$; number of classes n; classifier f_θ; generator $\mathscr{G}_\phi$; epochs T; uncertainty threshold factor λ

Ensure: Trained f_θ for unified OOD detection and UE

Learning: TIE Framework
Initialize garbage class $(n{+}1)$ with Gaussian noise $\mathscr{D}_{\text{garbage}} \sim \mathscr{N}(0, I)$
Initialize class weights $\mathbf{w}_0$ for weighted cross-entropy
for $t = 1$ to T **do** ▷ Training–Inversion–Exclusion loop
    Compute class weights: $\mathbf{w}_t \leftarrow$ proportions in $(\mathscr{D}_{\text{train}} \cup \mathscr{D}_{\text{garbage}})$
    Train classifier: update f_θ on $\mathscr{D}_{\text{train}} \cup \mathscr{D}_{\text{garbage}}$ using $\mathscr{L}_{\text{CE}}(\mathbf{w}_t)$
    Learn generator via inversion: for each class $c \in \{1, \ldots, n{+}1\}$, sample $z \sim \mathscr{N}(0, I)$ and soft conditioning vector $\tilde{y}_c$, generate $\hat{x}_c = \mathscr{G}_\phi(z, \tilde{y}_c)$, and optimize ϕ using $\mathscr{L}_{\text{Inv}}$ ▷ Section 3
    Collect inverted samples $\mathscr{D}_{\text{inv}} = \{\hat{x}_c\}_{c=1}^{n+1}$
    Compute dynamic uncertainty threshold:
    for each $x \in \mathscr{D}_{\text{train}}$ **do**
        $u(x) = \text{UE}(f_\theta(x))$ ▷ Eq. 4
    end for
    $\mu_t = \text{mean}(u),\ \sigma_t = \text{std}(u),\ \tau_t = \mu_t + \lambda \sigma_t$
    Exclude high-uncertainty inversions:
    for each $\hat{x} \in \mathscr{D}_{\text{inv}}$ **do**
        if $\text{UE}(f_\theta(\hat{x})) > \tau_t$ **then**
            $\mathscr{D}_{\text{garbage}} \leftarrow \mathscr{D}_{\text{garbage}} \cup \{\hat{x}\}$
        end if
    end for
    Update $\mathbf{w}_{t+1}$ for the next epoch
end for
Inference: Two-Level OOD Detection
For test sample x^*, compute $\mathbf{p} = f_\theta(x^*)$
if $\arg\max_i \mathbf{p}_i = n{+}1$ **then**
    Level 1: Reject x^* as OOD via the garbage class
else
    Discard the garbage-class logit and normalize the remaining n outputs
    Compute Mahalanobis OOD score $s_{\text{Maha}}(x^*)$
    if $s_{\text{Maha}}(x^*)$ satisfies the OOD rejection criterion **then**
        Level 2: Reject x^* as OOD
    else
        Assign in-distribution label $\arg\max_{i \le n} \mathbf{p}_i$
    end if
end if