JOHANNES KEPLER
UNIVERSITY LINZ

Conditional Diffusion Models as Generative Classifiers for Out-of-Distribution Detection in Inkjet Print Quality Control

Author
Ahmed Mohammed

Submission
Institute for Machine Learning

Thesis Supervisor
Univ.-Prof. Dr. **Sepp Hochreiter**

Assistant Thesis Supervisor
Claus Hofmann, MSc

Mar. 2026

Master Thesis
to obtain the academic degree of
Master of Science
in the Master's Program
Artificial Intelligence

JOHANNES KEPLER UNIVERSITY LINZ
Altenbergerstraße 69
4040 Linz, Austria
www.jku.at
DVR 0093696

Abstract

Out-of-distribution (OOD) detection is a critical challenge in deploying machine learning systems safely. While deep learning has achieved considerable success, models often encounter data differing fundamentally from their training distribution yet produce overconfident predictions without recognising this fact. We treat conditional diffusion models as generative classifiers: class-conditional reconstruction error serves as the OOD signal.

Unlike likelihood-based approaches that suffer from the manifold hypothesis and the curse of dimensionality in high dimensions, diffusion models' iterative denoising process provides an implicit regularisation that makes reconstruction error more resilient to high-dimensional structure. We develop and evaluate a binary conditional diffusion model (CDM) for OOD detection on CIFAR-10, introducing a class-conditional separation loss that explicitly encourages discriminative reconstruction error distributions. The separation loss proves decisive: without it (λ=0), the model achieves only 80.25% AUROC (Area Under the ROC Curve); with λ=0.02, performance reaches $99.03\% \pm 0.07\%$—a gain of 18.8 percentage points. Under the final audit policy, core claims are based on reproducible artefacts: 98.98% AUROC on the within-CIFAR split (seed42 raw-score evaluation) and 90.50–96.97% AUROC across five externally auditable datasets (CIFAR-100, Places365, FashionMNIST, SVHN, Textures). Legacy Food-101/STL-10 values are retained for traceability only.

We further apply conditional diffusion models to industrial quality classification of inkjet print samples, introducing a multi-head conditioning mechanism for structured manufacturing data. In an 8-method comparison using 5-fold cross-validation, we find that input representation dominates model architecture: full-image supervised methods achieve 0.945 AUROC compared to 0.848 for the crop-based CDM pipeline, revealing that the two-stage design discards critical global context.

We analyse why diffusion-based reconstruction error succeeds where other generative approaches fail, characterise the accuracy-efficiency trade-off as a function of Monte Carlo timestep trials (K=10 provides an excellent balance at $\sim 32\times$ discriminative cost), and provide a modular implementation using PyTorch Lightning and Hydra. This work demonstrates that training objective design—specifically the separation loss—is the dominant factor in diffusion-based OOD detection, and that the same framework of class-conditional reconstruction error extends naturally to practical industrial applications.

Keywords: Diffusion Models, Out-of-Distribution Detection, Generative Classification, Industrial Quality Control, Inkjet Printing

Acknowledgments

I would like to express my sincere gratitude to my thesis supervisor, Prof. Dr Sepp Hochreiter, for his valuable guidance, insightful feedback, and unwavering support throughout this research. His expertise in machine learning and deep learning has been instrumental in shaping this work.

I am grateful to my assistant supervisor, Claus Hofmann, MSc, for his constructive feedback and for our back-and-forth discussions, which helped refine the methodology and experimental approach.

This thesis was conducted as an industrial research project at PROFACTOR GmbH, and I am especially grateful to Dr. Christian Eitzinger, machine vision team leader at PROFACTOR, who patiently guided me throughout the entire journey. I would also like to thank Abdalla Shahin, Boris Buchroithner, and all colleagues at PROFACTOR for their support and contributions throughout this work.

I would also like to thank the Institute for Machine Learning at Johannes Kepler University Linz for providing the computational resources and supportive research environment necessary to conduct this work.

The work of this thesis was carried out in the premises of PROFACTOR GmbH, where I had the opportunity to access all the needed resources and facilities. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 958472, project TINKER.

I am deeply indebted to my family and my wife for their relentless support.

Finally, I would like to express my appreciation to my friends for their constant inspiration and encouragement throughout the research and writing process.

During the preparation of this thesis, AI-based writing tools were used for language editing and proofreading; the author is responsible for the content.

Declaration

I hereby declare that this thesis is my own work and that I have not submitted it, either in its entirety or in substantial part, to any other university for a degree. All sources and external materials used in the preparation of this thesis have been acknowledged and properly cited.

Linz, March 2026

_____

Ahmed Mohammed

Contents

List of Figures

List of Tables

JOHANNES KEPLER
UNIVERSITY LINZ

1. Introduction

1.1. Motivation and Problem Statement

Deep learning has achieved considerable success across a wide range of tasks, from image classification and object detection to natural language processing and autonomous decision-making. However, most deployed models assume that test data is drawn from the training distribution—an assumption frequently violated in practice. Models encounter inputs that differ fundamentally from their training distribution—novel object categories, sensor degradation, adversarial perturbations, or simply data from a different domain. When this occurs, standard neural networks tend to produce overconfident predictions rather than signalling uncertainty, creating a critical failure mode for safety-critical applications (Hendrycks and Gimpel, 2017; Guo et al., 2017).

Out-of-distribution (OOD) detection addresses this challenge by equipping models with the ability to distinguish in-distribution (ID) data from data that does not belong to the learnt distribution. A reliable OOD detector enables a model to say "I don't know" when confronted with unfamiliar inputs, a prerequisite for deploying machine learning systems in healthcare, autonomous driving, industrial quality control, and other high-stakes domains (J. Yang et al., 2021).

1.1.1. Limitations of Existing Approaches

Existing OOD detection methods can be broadly categorised into discriminative and generative approaches, each with significant limitations.

Discriminative approaches repurpose features or outputs of classifiers trained for the in-distribution task. Maximum Softmax Probability (MSP) (Hendrycks and Gimpel, 2017)

uses the classifier's confidence as an OOD score, but it suffers from the well-documented overconfidence problem. ODIN (Liang et al., 2018) applies temperature scaling and input perturbations to improve separation. The Mahalanobis distance method (Lee et al., 2018) uses feature-space distances under the assumption that the data follow a Gaussian distribution. Energy-based scoring (W. Liu et al., 2020) interprets classifier logits as energy values, providing a theoretically motivated OOD signal. While computationally efficient, these methods rely on classifier-derived representations that may not capture all distributional structure relevant for OOD detection.

Generative approaches aim to model the data distribution directly, which, in principle, should provide a natural OOD signal: samples with low likelihood under the learnt distribution should be flagged as OOD. However, this intuition fails in practice due to the *likelihood paradox*. Nalisnick et al. (2019) demonstrated that deep generative models—including variational autoencoders (VAEs) and normalising flows—can assign *higher* likelihoods to OOD data than to their own training distribution. For example, a model trained on CIFAR-10 assigns a higher likelihood to Street View House Numbers (SVHN) images than to CIFAR-10 images. This paradox, linked to the interaction between high-dimensional geometry and model capacity (Kirichenko et al., 2020), undermines the reliability of likelihood-based OOD detection.

Reconstruction-based methods, which measure the deviation between the input and the reconstruction, offer an alternative. VAE-based anomaly detectors use reconstruction probability as a score (An and Cho, 2015), but their pixel-space reconstructions are often too blurry—due to the Gaussian decoder assumption—to discriminate subtle distributional differences, limiting their effectiveness on complex natural-image benchmarks.

1.1.2. The Promise of Diffusion Models

Diffusion models (Ho et al., 2020; Y. Song et al., 2021) have emerged as a powerful class of generative models that learn to reverse a gradual noising process. Starting from pure Gaussian noise, they iteratively denoise samples to produce high-fidelity outputs, achieving strong benchmark results in image generation (Dhariwal and A. Nichol, 2021; Rombach et al., 2022). Crucially, their training objective—predicting the noise added at each timestep—provides a natural reconstruction error signal that is fundamentally different from VAE reconstruction.

The *Diffusion Classifier* framework introduced by A. C. Li et al. (2023) demonstrates that conditional diffusion models can perform classification without explicit discriminative training. By comparing noise prediction errors under different class-conditional models, the framework selects the class that best explains the observed data. This principle extends naturally to OOD detection: in-distribution samples should yield lower reconstruction error than OOD samples, as the diffusion model has learnt to denoise only data from the training distribution.

This reconstruction-based approach sidesteps the specific failure mode of raw likelihood in high dimensions, though reconstruction error may have its own failure modes under adversarial inputs. Rather than estimating density in high-dimensional space—where pathological behaviours emerge—diffusion models measure how well they can denoise an input given learnt class-conditional data manifolds. The iterative denoising process provides implicit regularisation that makes reconstruction error more resilient to the high-dimensional structure that undermines likelihood-based methods.

1.1.3. Research Gap

Despite the theoretical appeal, several critical questions remain unexplored:

- How effective are binary conditional diffusion models as OOD detectors across diverse near- and far-OOD benchmark datasets?
- How does the training objective shape OOD detection performance? Specifically, can enforcing class-conditional separability in reconstruction error—through a dedicated separation loss—significantly improve detection reliability?
- What is the optimal number of Monte Carlo timestep trials K, and how does K govern the accuracy-efficiency frontier?
- What are the practical trade-offs in computational cost, inference speed, and deployment complexity?
- Can the same principles extend beyond standard benchmarks to real-world industrial applications, such as detecting manufacturing quality defects?

This thesis addresses these questions through a systematic investigation of conditional diffusion models for OOD detection, combining rigorous benchmarking on standard datasets with exploratory application to industrial quality control.

1.2. Research Questions and Objectives

1.2.1. Primary Research Question

How can conditional diffusion models be effectively leveraged as generative classifiers for out-of-distribution detection, and what design choices—particularly regarding the training objective and inference strategy—most strongly influence detection performance?

This overarching question is decomposed into the following sub-questions:

1. **RQ1: Effectiveness.** Can reconstruction error from conditional diffusion models serve as a reliable and high-quality OOD detection signal?
2. **RQ2: Separation Loss.** How does incorporating a class-conditional separation loss (a training objective that penalises overlap between ID and non-ID reconstruction error distributions—formalised in Chapter 3) into the diffusion training objective affect OOD detection performance? Can explicitly penalising reconstruction error overlap between ID and non-ID classes substantially improve detection?
3. **RQ3: Monte Carlo Trials.** What is the relationship between the number of Monte Carlo timestep trials K and the accuracy-efficiency trade-off? At what value of K does performance plateau?
4. **RQ4: Scoring Strategy.** How does the choice of OOD scoring strategy—using ID reconstruction error alone versus contrastive difference scoring (subtracting ID error from OOD error)—affect detection performance across near- and far-OOD datasets?
5. **RQ5: Practical Viability.** What are the computational costs of diffusion-based OOD detection relative to discriminative baselines, and under what conditions is the accuracy-cost trade-off favourable?

6. **RQ6: Industrial Application.** How effective are conditional diffusion models for industrial quality classification, and what factors—input representation, model architecture, learning paradigm—determine classification performance in practical manufacturing settings?

1.2.2. Specific Objectives

To address the research questions, this thesis pursues four objectives:

Objective 1: Framework Development. Design and implement a framework for diffusion-based OOD detection, including:

- A binary conditional diffusion model with class-conditional reconstruction error scoring
- A separation loss training objective that encourages distinct reconstruction error distributions for ID and non-ID classes
- Configurable OOD scoring via Monte Carlo timestep sampling with adjustable trial count K and contrastive difference scoring

Objective 2: Benchmarking. Conduct an evaluation on standard OOD benchmarks:

- **In-distribution:** CIFAR-10 airplane class (Krizhevsky, Hinton, et al., 2009)
- **Within-CIFAR OOD:** remaining nine CIFAR-10 classes
- **External OOD (auditable core):** CIFAR-100 (near), Places365 (far), FashionMNIST (far), SVHN (far), Textures (far). Legacy traceability entries: Food-101 and STL-10 (excluded from core claims due to missing current raw-score tensors).
- All experiments use multiple random seeds; results are reported as mean and standard deviation

Objective 3: Systematic Ablation Studies. Validate design choices through controlled experiments:

- Separation loss coefficient ($\lambda \in \{0.0, 0.001, 0.01, 0.02, 0.05, 0.1\}$)
- Number of Monte Carlo timestep trials ($K \in \{1, 5, 10, 25, 50, 100\}$)

- OOD scoring strategy (ID reconstruction error alone vs. contrastive difference scoring)

Objective 4: Reproducible Implementation. Develop a well-documented, modular codebase:

- Built on PyTorch (Paszke et al., 2019), PyTorch Lightning (Falcon and The PyTorch Lightning team, 2019), and Hydra (Yadan, 2019)
- Standardised evaluation pipeline with standard metrics (Area Under the ROC Curve (AUROC), False Positive Rate at 95% True Positive Rate (FPR@TPR95), Area Under the Precision-Recall Curve (AUPRC))
- Complete experiment tracking and configuration logging

1.3. Contributions

This thesis makes the following contributions:

1. **Systematic evaluation of binary conditional diffusion models for OOD detection.** We demonstrate that a binary conditional diffusion model (CDM) (airplane vs. rest) achieves strong AUROC on the within-CIFAR split and across five external OOD datasets spanning near- and far-OOD scenarios (Chapter 6).
2. **Separation loss as a key training objective.** We introduce a class-conditional separation loss that penalises overlapping reconstruction error distributions between ID and non-ID classes, yielding the single largest performance improvement in our study (Chapter 6, Section 6.4).
3. **Inference design choices: MC trials and contrastive scoring.** We characterise the accuracy–efficiency frontier as a function of Monte Carlo trials K, and demonstrate that contrastive difference scoring is essential — ID-only scoring fails catastrophically on certain external datasets (Chapter 6, Section 6.3).
4. **Application to industrial quality control.** We compare 8 methods (3 supervised, 4 anomaly detection, 1 generative) for inkjet quality classification under 5-fold cross-validation. We find that input representation (full image vs. crop) dominates model

architecture, revealing that the crop-based CDM pipeline discards critical global context (Chapter 6, Section 6.6).

5. **Reproducible implementation framework.** We provide a modular implementation using PyTorch Lightning and Hydra with standardised evaluation tools. The CIFAR-10 code is publicly available; the inkjet dataset is proprietary (PROFACTOR GmbH).

1.4. Thesis Structure

The remainder of this thesis is organised as follows. **Chapter 2** reviews OOD detection methods, diffusion model foundations, industrial quality classification baselines, and positions our contributions via a gap analysis. **Chapter 3** formalises both experimental tracks: the binary CDM with separation loss and contrastive scoring for OOD detection, and the two-stage YOLO+CDM pipeline with multi-head conditioning for inkjet quality control. **Chapter 4** describes the technical realisation, including the software stack, cross-validation infrastructure, and reproducibility measures. **Chapter 5** defines the experimental design, datasets, ablation studies, and evaluation protocols for both tracks. **Chapter 6** presents the experimental results, ablation analyses, and the inkjet 8-method comparison. **Chapter 7** interprets the findings, draws cross-domain insights, and discusses limitations and future work. **Chapter 8** answers the six research questions and summarises contributions.

2. Background and Related Work

This chapter provides the theoretical and empirical context for the two experimental tracks developed in this thesis. We survey the landscape of out-of-distribution (OOD) detection, covering discriminative and generative approaches and their known limitations, then review the diffusion model foundations used throughout. We further describe industrial quality classification baselines and identify the research gaps that motivate our contributions.

2.1. Out-of-Distribution Detection

Out-of-distribution detection addresses a fundamental challenge in deploying machine learning models: identifying when test samples differ significantly from the training distribution. This section formalises the OOD detection problem, characterises different types of distribution shift, and discusses why robust OOD detection is critical for real-world applications.

2.1.1. Problem Definition

Let $\mathcal{X}$ denote the input space and $\mathcal{Y}$ denote the label space. During training, we have access to a dataset $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{n}$ drawn i.i.d. from an in-distribution (ID) $P_{\text{train}}(x, y)$. A model $f_\theta : \mathcal{X} \to \mathcal{Y}$ is learnt to minimise some loss function $\mathcal{L}$ over this distribution:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim P_{\text{train}}}[\mathcal{L}(f_\theta(x), y)] \tag{2.1}$$

At test time, the model may encounter samples x_{test} that come from a different distribution $P_{\text{ood}}(x)$. The goal of OOD detection is to determine whether a given test sample x is drawn from P_{train} or P_{ood} without access to labels from the OOD distribution during training.

Formally, we seek a scoring function $s : \mathcal{X} \to \mathbb{R}$ such that:

$$s(x) \begin{cases} > \tau & \text{if } x \sim P_{\text{ood}} \\ \leq \tau & \text{if } x \sim P_{\text{train}} \end{cases} \tag{2.2}$$

where τ is a threshold that can be calibrated using a held-out validation set from P_{train}.

The effectiveness of an OOD detector is typically evaluated using metrics that assess the separability of ID and OOD scores:

Area Under the ROC Curve (AUROC): Measures the probability that a randomly chosen OOD sample has a higher score than a randomly chosen ID sample. An AUROC of 1.0 indicates perfect separation, while 0.5 represents random guessing.

False Positive Rate at 95% True Positive Rate (FPR@TPR95): Measures the fraction of ID samples incorrectly classified as OOD when the threshold is set such that 95% of OOD samples are correctly identified. Lower values indicate better performance.

Detection Accuracy: The proportion of correctly classified samples (both ID and OOD) when using an optimal threshold. While simple to interpret, this metric can be misleading when ID and OOD sample sizes are imbalanced.

Distribution shift can take several forms—covariate shift, label shift, or semantic shift—each presenting distinct challenges (J. Yang et al., 2021). Our work primarily addresses **semantic shift** (test classes absent from training) with both **near-OOD** scenarios (e.g., CIFAR-100 when trained on CIFAR-10) and **far-OOD** scenarios (e.g., SVHN, Textures).

2.2. Deep Learning Approaches

The success of deep learning in representation learning has spawned numerous OOD detection methods. These can be broadly categorised into discriminative methods that exploit classifier properties and generative methods that model the data distribution.

2.2.1. Discriminative Methods

Discriminative methods leverage trained classifiers without density modelling. Key baselines include Maximum Softmax Probability (MSP) (Hendrycks and Gimpel, 2017), ODIN (Liang et al., 2018) (input perturbation with temperature scaling), Mahalanobis distance in feature space (Lee et al., 2018), and Energy-based detection (W. Liu et al., 2020). These methods are computationally efficient but can struggle when OOD samples lie near decision boundaries.

2.2.2. Reconstruction-based and Generative Approaches

Reconstruction-based methods use the discrepancy between input and reconstruction as an OOD indicator. Standard autoencoders measure reconstruction error $\|x - d(e(x))\|^2$ as an anomaly score, but the assumption that autoencoders fail on OOD samples often does not hold—they can generalise to reconstruct OOD inputs well, particularly with high-dimensional latent spaces. Variational autoencoders (VAEs) suffer from both likelihood misassignment and poor reconstruction quality due to posterior collapse.

Generative models that explicitly learn $P_{\text{train}}(x)$ should in principle use likelihood for OOD detection, but this intuition fails in practice. Nalisnick et al. (2019) demonstrated the *likelihood paradox*: deep generative models—including normalising flows with exact likelihoods—can assign higher likelihood to OOD data than ID data (e.g., flows trained on CIFAR-10 assign higher likelihood to SVHN). Kirichenko et al. (2020) showed that likelihood decomposes into on-manifold and off-manifold components, with off-manifold volume often dominating. These findings motivated our investigation of diffusion models, whose reconstruction-based scoring sidesteps the likelihood paradox.

2.3. Diffusion Models

Diffusion models have emerged as a powerful class of generative models, achieving strong benchmark results in image synthesis and other generative tasks. This section provides the mathematical foundations necessary for understanding how diffusion models can be applied to OOD detection.

2.3.1. Mathematical Foundations

Diffusion models, also known as denoising diffusion probabilistic models (DDPMs) (Ho et al., 2020), define a forward noising process and learn to reverse it.

Forward Process: The forward process gradually adds Gaussian noise to data over T timesteps:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \tag{2.3}$$

where $\{\beta_t\}_{t=1}^T$ is a variance schedule. Crucially, we can sample x_t directly from x_0:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I) \tag{2.4}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$. As $t \to T$, x_T approaches pure Gaussian noise.

Reverse Process: The reverse process learns to denoise:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \tag{2.5}$$

The mean μ_θ can be parameterised in various ways. Following DDPM, we typically predict the noise $\epsilon_\theta(x_t, t)$ and compute:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) \tag{2.6}$$

Training Objective: The model is trained by minimising the variational lower bound, which simplifies to:

$$\mathcal{L} = \mathbb{E}_{t,x_0,\epsilon}\left[||\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)||^2\right] \tag{2.7}$$

where $\epsilon \sim \mathcal{N}(0, I)$ and $t \sim \text{Uniform}(\{1, \ldots, T\})$.

This objective has an elegant interpretation: the model learns to predict the noise that was added to clean data x_0 at timestep t.

2.3.2. DDPM Variants and Extensions

Noise Schedules: The variance schedule $\{\beta_t\}$ significantly impacts performance; cosine schedules (A. Q. Nichol and Dhariwal, 2021) maintain more information at higher timesteps than linear schedules.

Score-based Formulation: Y. Song et al. (2021) showed that diffusion models learn the score function $\nabla_x \log p(x)$, revealing the connection $\epsilon_\theta(x_t, t) \approx -\sqrt{1-\bar{\alpha}_t}\nabla_{x_t} \log p(x_t)$. This perspective connects diffusion models to Langevin dynamics and enables continuous-time stochastic differential equation (SDE) formulations. Accelerated samplers such as Denoising Diffusion Implicit Models (DDIM) (J. Song et al., 2021) allow fewer-step generation without retraining. This thesis uses the discrete DDPM formulation throughout.

2.3.3. Conditional Diffusion Models

Conditioning enables diffusion models to generate samples from specific categories or with desired attributes.

Classifier Guidance: Dhariwal and A. Nichol (2021) introduced classifier guidance, which steers generation using gradients from a separately trained classifier. While effective, it requires maintaining an auxiliary classifier.

Classifier-Free Guidance: Ho and Salimans (2022) proposed classifier-free guidance, which eliminates the need for a separate classifier:

$$\tilde{\epsilon_\theta}(x_t, t, y) = \epsilon_\theta(x_t, t, \varnothing) + w(\epsilon_\theta(x_t, t, y) - \epsilon_\theta(x_t, t, \varnothing)) \tag{2.8}$$

During training, the condition y is randomly dropped with probability p (typically 0.1), allowing the model to learn both conditional and unconditional predictions. This approach has become the standard for conditional generation.

Class Embeddings: Conditioning typically injects class information through learnt embeddings added to timestep embeddings or through cross-attention mechanisms in transformer-based architectures.

Multi-Head Conditioning: Beyond single-label conditioning, diffusion models can be conditioned on multiple heterogeneous inputs simultaneously. In our inkjet quality control application, we employ multi-head conditioning where the model receives template type (categorical), feature type (categorical), quality label (binary), and bounding box coordinates (continuous) as separate conditioning inputs, each encoded through dedicated embedding layers before being combined. This multi-head approach enables a single model to handle diverse feature types and spatial contexts.

Diffusion models have achieved strong results across image generation, super-resolution, inpainting, and text-to-image synthesis (Dhariwal and A. Nichol, 2021; Rombach et al., 2022). However, their application to *discriminative* tasks like OOD detection and industrial quality classification remains relatively unexplored. A preliminary attempt to apply text-conditioned Stable Diffusion (Rombach et al., 2022) to the inkjet defect dataset yielded unsatisfactory generation quality on this small, domain-specific dataset—the model could not reliably reproduce fine-grained defect patterns from text prompts alone, and the classification stage was never reached. This negative result motivated the shift to class-label conditioning on a lightweight binary CDM trained from scratch, which forms the basis of this work.

2.4. The Manifold Hypothesis

The manifold hypothesis posits that high-dimensional data concentrate on or near low-dimensional manifolds embedded in the ambient space ($d \ll D$; for CIFAR-10, $D = 3072$ while intrinsic dimension is typically in the tens to low hundreds (Pope et al., 2021)). This hypothesis has direct implications for OOD detection.

Different generative models capture manifold structure differently: VAEs parameterise it via a latent space, flows compute exact likelihood but conflate manifold density with ambient volume (explaining the likelihood paradox), and diffusion models iteratively project noisy samples onto the learnt manifold through progressive denoising.

The manifold perspective explains why likelihood-based OOD detection fails—volume dominates distance in high dimensions—and suggests reconstruction error as an alternative: the distance from a sample to the learnt manifold provides a more reliable OOD signal than density. Diffusion models' iterative denoising can be interpreted as progressive

projection onto the data manifold, where high reconstruction error indicates that a sample lies far from the learnt manifold. This perspective motivates our use of reconstruction error from conditional diffusion models as an OOD detection signal.

2.5. Industrial Quality Control and Anomaly Detection

As a practical application domain for the principles developed in this thesis, we investigate automated quality classification for inkjet print samples. This section reviews the relevant background on industrial visual inspection, object detection for region-of-interest extraction, and anomaly detection methods used as baselines.

2.5.1. Automated Visual Inspection

Automated visual inspection (AVI) has become essential in modern manufacturing, replacing subjective human assessment with consistent, quantitative evaluation. In inkjet print samples, quality defects manifest as misaligned features, irregular edge roughness, incorrect spacing, and malformed dot patterns. These defects reduce production yield and compromise the feedback loop used to optimise printing parameters.

Traditional AVI systems rely on hand-crafted features and template matching, which require significant domain expertise and are brittle to variations in printing conditions. Deep learning-based approaches have largely superseded these methods, leveraging learnt representations to capture subtle quality variations that resist manual feature engineering.

A key challenge in industrial quality classification is choosing the input representation. Full-image approaches preserve global spatial context—the position of features relative to page edges and other features—while crop-based approaches focus local analysis on individual regions of interest. As we demonstrate in Chapter 6, this design choice has a significant impact on classification performance, with global context contributing approximately 10% AUROC improvement.

2.5.2. Object Detection for Feature Extraction

Modern object detection architectures, particularly the YOLO (You Only Look Once) family (Redmon et al., 2016), enable real-time detection and localisation of objects in images. In our pipeline, we use YOLOv8 (Jocher et al., 2023) to detect and extract individual printed features from full template images before quality classification.

YOLOv8 Architecture: YOLOv8 uses a CSPDarknet backbone with a Feature Pyramid Network (FPN) and Path Aggregation Network (PAN) for multi-scale feature fusion. The anchor-free detection head simultaneously predicts bounding box coordinates and class probabilities. The YOLOv8n (nano) variant used in our work contains approximately 3.2M parameters and achieves real-time inference.

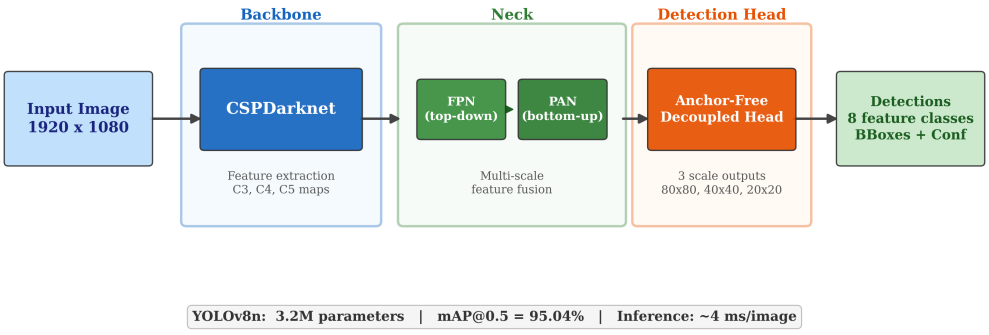

Figure 2.1: YOLOv8 architecture overview. The CSPDarknet backbone extracts multi-scale feature maps (C3, C4, C5), which are fused by the Feature Pyramid Network (top-down) and Path Aggregation Network (bottom-up). The anchor-free decoupled detection head produces bounding boxes, class predictions, and confidence scores at three scales. The YOLOv8n variant (3.2M parameters) achieves mAP@0.5 of 95.04% on our inkjet feature detection task.

Role in the Pipeline: Object detection serves as a preprocessing stage that transforms full template images (1920×1080 pixels) into individual feature crops (128×128 pixels), expanding 1,327 template-level images into 6,408 feature-level samples. Each detected region carries metadata including template type (A, B, or C), feature class (8 types), and bounding box coordinates, which serve as conditioning inputs for the diffusion model.

2.5.3. Anomaly Detection Methods

Anomaly detection methods learn only from normal (defect-free) samples and detect anomalies as deviations from the learnt normality. We evaluate four anomaly detection approaches as baselines for the inkjet quality classification task.

PatchCore (Roth et al., 2022): A memory-bank approach that scores anomalies by computing the minimum distance from test patch features to a coreset of pretrained ResNet features from normal samples. Requires no training beyond feature extraction.

PaDiM (Defard et al., 2021): Models normal patch features as position-wise multivariate Gaussians; anomaly scores use Mahalanobis distance from the learnt distribution.

Student-Teacher Feature Pyramid Matching (STFPM) (G. Wang et al., 2021): Knowledge distillation where a student mimics a pretrained teacher on normal data; anomalies produce feature-pyramid discrepancies.

Autoencoder: Reconstruction-based approach where reconstruction error serves as the anomaly score. As we show in Chapter 6, this assumption often fails for subtle quality defects. Recent extensions include FR-PatchCore (Jiang et al., 2024) for spatial robustness; J. Wang et al. (2025) survey the growing application of diffusion models to industrial anomaly detection; X. Li et al. (2025) contribute a continual diffusion method that handles multiple anomaly types within a single model.

2.5.4. Supervised Classification Methods

When labelled data for both good and defective samples is available, supervised classification methods provide a strong alternative. We evaluate three supervised architectures:

ResNet-FullImg: A standard ResNet-18 (He et al., 2016) pretrained on ImageNet, fine-tuned on full template images resized to 224×224. This approach preserves global spatial context, allowing the model to leverage inter-feature relationships and page-level alignment cues.

ResNet-CropOnly: The same ResNet-18 architecture applied to YOLO-cropped feature regions resized to 224×224. This isolates local feature quality but loses global context.

Dual-Branch: Two ResNet-18 encoders process full images and cropped regions in parallel, with feature embeddings concatenated and passed through a fusion multi-layer perceptron (MLP) for final classification. A feature-type embedding provides additional conditioning.

These supervised methods serve as upper bounds for the quality classification task, establishing the achievable performance with direct label supervision.

2.6. Gap Analysis and Positioning

We now identify gaps in the current literature and position our contributions.

2.6.1. Limitations of Current Approaches

Discriminative methods rely on classifier features that may not generalise across distribution shifts. Generative methods face the likelihood paradox (Section 2.2), and diffusion models remain underexplored for OOD detection. Industrial quality methods either discard defect labels (anomaly detection) or require abundant labelled data (supervised), and the trade-off between crop-based and full-image input has not been systematically studied.

2.6.2. Recent Advances in OOD Detection (2022–2025)

Since the classical baselines reviewed above, the OOD detection field has advanced substantially. We briefly survey the most relevant recent developments to position our work.

One-Class Novelty Detection. A distinct but related line of work frames OOD detection as one-class classification: models train only on a single in-distribution class and flag anything else as anomalous. Deep SVDD (Ruff et al., 2018) learns a compact hypersphere enclosing normal features, while DROCC (Goyal et al., 2020) generates synthetic boundary examples to sharpen the one-class boundary. PANDA (Reiss et al., 2021) adapts pretrained ImageNet features to the target class via continual learning, achieving strong results with

minimal training. CSI (Tack et al., 2020) combines contrastive learning with distributional shift detection. These methods operate under a stricter assumption than multi-class OOD detection: the model sees only one class during training and must generalise entirely from that distribution. Our binary CDM setup differs: we train on both an ID class (airplane) and a non-ID proxy (all other CIFAR-10 classes), making direct numerical comparison inappropriate. However, the one-class literature confirms that reconstruction-based signals can be competitive with feature-distance methods when the training objective is carefully designed. More recently, Mirzaei et al. (2024) propose universal novelty detection via adaptive contrastive learning at CVPR 2024, demonstrating that a single framework can operate across one-class, unlabelled multi-class, and labelled multi-class settings.

Post-Hoc Feature-Based Methods. KNN-OOD (Sun et al., 2022) uses k-nearest neighbour distances in the feature space of a pretrained model, achieving strong performance without retraining. ViM (H. Wang et al., 2022) combines a null-space-projected virtual logit with the classifier's energy score. ReAct (Sun et al., 2021) improves OOD detection by truncating high activations in the penultimate layer. ASH (Djurisic et al., 2023) extends this line by selectively pruning activations. These methods are highly efficient—requiring only a pretrained classifier—and are reported to achieve state-of-the-art results on standard OOD benchmarks such as OpenOOD (Zhang et al., 2023).

Diffusion-Based OOD Detection. DiffGuard (Gao et al., 2023) leverages diffusion models for OOD detection, achieving 99%+ AUROC on some benchmarks through multi-step denoising with classifier guidance. Livernoche et al. (2024) propose diffusion-based density estimation for OOD detection, demonstrating strong results at ICLR 2024. Graham et al. (2023) use denoising diffusion models with a focus on near-OOD detection. More recent work has further expanded this direction: Y. Yang et al. (2024) propose layer-wise semantic reconstruction at NeurIPS 2024, showing that feature-level diffusion reconstruction outperforms pixel-level approaches for unsupervised OOD detection. Heng et al. (2024) demonstrate that a single unconditional diffusion model suffices for OOD detection by analysing the structure of denoising trajectories. Wu et al. (2024) reveal that diffusion denoisers are implicitly noise classifiers and benefit from contrastive training objectives. Linmans et al. (2024) apply diffusion-based OOD detection to digital pathology, validating the approach in a medical imaging domain. For comprehensive recent treatments, see the surveys by Lu et al. (2025) and J. Liu et al. (2025).

Standardised Benchmarks. The OpenOOD benchmark (Zhang et al., 2023) provides a unified evaluation framework for OOD detection with standardised near-OOD/far-OOD splits, covering CIFAR-10, CIFAR-100, and ImageNet-200. Our work evaluates on CIFAR-10 with the same OOD datasets but does not use the full OpenOOD protocol.

Scope and Limitations. Our experimental evaluation focuses on characterising the binary conditional diffusion model itself—measuring OOD detection performance across the within-CIFAR split and external OOD datasets, and isolating the impact of key design choices (separation loss, MC trials, scoring strategy) through ablation studies. Under the final audit policy, core claims are restricted to reproducible artefacts (within-CIFAR plus five external datasets with current raw scores), while Food-101/STL-10 are retained as legacy traceability entries only. We do not include direct runtime comparisons against discriminative baselines (MSP, ODIN, KNN-OOD) or generative baselines (VAE, normalising flows), acknowledging that established methods and more recent approaches (ViM, ASH, DiffGuard) may achieve comparable or superior performance on standard benchmarks. Readers are encouraged to consult the OpenOOD benchmark (Zhang et al., 2023) for comprehensive cross-method comparisons; full evaluation protocol details are given in Chapter 5.

2.6.3. Research Gaps

Our literature review reveals several specific gaps in the current body of work:

Gap 1: Systematic Evaluation of Diffusion Models for OOD Detection: While several concurrent works have begun exploring diffusion models for OOD detection—including DiffGuard (Gao et al., 2023), A. C. Li et al. (2023), and Livernoche et al. (2024)—the quantitative impact of dedicated training objectives (such as a separation loss that explicitly encourages reconstruction error discriminability), the optimal number of Monte Carlo timestep trials K, and systematic evaluation across diverse near- and far-OOD datasets remain underexplored. This gap concerns *what training and inference design choices matter* for diffusion-based OOD detection.

Gap 2: Conditional Diffusion as Generative Classifiers: Orthogonal to Gap 1, the *architectural mode* of conditioning has not been examined: while existing work uses unconditional

or text-conditioned diffusion models, the use of class-label conditioning to create a binary generative classifier—and specifically the role of contrastive scoring (subtracting ID from OOD reconstruction error) vs. single-sided scoring—remains unanalysed. This gap concerns *how conditioning structure shapes discriminative OOD signal*.

Gap 3: Diffusion Models for Industrial Quality Control: The application of conditional diffusion models to quality classification in manufacturing—where multiple conditioning signals (feature type, spatial location, template) are available—has not been explored. Comparing diffusion-based generative classification against supervised and anomaly detection baselines in this domain would reveal whether the theoretical advantages of reconstruction-based approaches translate to practical benefit.

Gap 4: Input Representation Analysis: The systematic impact of input representation (full image vs. cropped region) on quality classification performance, and how this interacts with model architecture choice (supervised vs. generative vs. anomaly detection), has not been rigorously studied.

2.6.4. Thesis Positioning

This thesis addresses these gaps through two complementary experimental tracks:

Track 1: CIFAR-10 OOD Detection. We provide a systematic investigation of a binary conditional diffusion model for OOD detection, evaluating performance across the within-CIFAR split and external OOD datasets while isolating the impact of key design choices—particularly the separation loss objective and the number of MC trials—through controlled ablation studies. Final core claims are restricted to auditable results with recoverable raw-score artefacts; legacy values are explicitly labelled as traceability-only. We establish empirical foundations for diffusion-based OOD detection, while acknowledging that more recent methods (Section 2.6.2) may achieve stronger absolute performance on standard benchmarks.

Track 2: Inkjet Print Quality Control. We apply conditional diffusion models to a real-world industrial quality classification task, comparing the YOLO+CDM pipeline against seven alternative methods spanning supervised learning, anomaly detection, and generative modelling. This track validates whether the principles developed on

benchmarks generalise to practical applications, while providing systematic analysis of input representation effects and paradigm trade-offs.

Practical Contribution: We provide a modular implementation built on PyTorch Lightning and Hydra with evaluation tools, reproducibility measures, and documentation. Parts of the implementation used in this thesis are publicly available at `https://github.com/ahmed-3m/DiffusionOOD`. The public repository contains the CIFAR-10/OOD benchmark code and selected experiment configurations. The industrial inkjet pipeline and associated data are not publicly released because they rely on proprietary company assets (PROFACTOR GmbH).

Our work explores whether diffusion models' generative capabilities translate into discriminative signals for OOD detection and quality classification, contributing systematic comparisons and practical insights to this rapidly evolving field.

3. Methodology

This chapter develops the methodological framework for both experimental tracks. We formalise the out-of-distribution detection problem as a binary generative classification task, introduce the class-conditional separation loss that is the primary novel contribution of this work, and describe the two-stage inkjet quality control pipeline with its multi-head conditional diffusion model. Formal analysis and architectural details are provided throughout to ground the experimental choices made in Chapters 4 and 5.

3.1. Problem Formulation

This section formalises the OOD detection problem and establishes how conditional diffusion models can be used as generative classifiers for this task.

3.1.1. Relation to One-Class Novelty Detection

Our problem formulation differs from the standard multi-class OOD detection benchmark setting. In the standard protocol (e.g., OpenOOD (Zhang et al., 2023)), a multi-class classifier is trained on all classes of an in-distribution dataset (e.g., all 10 CIFAR-10 classes), and OOD detection is evaluated on samples from entirely different datasets. By contrast, our binary CDM models a single target class (airplane) against all others, which is closer to the *one-class novelty detection* (also called one-class classification or anomaly detection) paradigm (Ruff et al., 2018; Tack et al., 2020).

In one-class novelty detection, a model learns the distribution of a single "normal" class and flags deviations as anomalous. Classical approaches include One-Class SVM (Schölkopf et al., 2001), Deep SVDD (Ruff et al., 2018), and contrastive methods such as CSI (Tack

et al., 2020) and PANDA (Reiss et al., 2021). Our binary CDM extends this paradigm by explicitly modelling both the target class and an OOD proxy class, enabling contrastive scoring rather than relying on a single normality model. This dual-condition design is the key architectural distinction from pure one-class methods and is what enables the separation loss to amplify the reconstruction gap.

Consequently, direct numerical comparison with standard multi-class OOD benchmarks (MSP, ODIN, Energy on all-class CIFAR-10) would be misleading, as they solve a different problem. Instead, we contextualise our results against one-class baselines evaluated on the same per-class CIFAR-10 protocol in Section 6.2.

3.1.2. Mathematical Definition

With the general OOD detection formulation established in Chapter 2, we now focus on our approach using conditional diffusion models.

Binary Training Setup: We assume access to a labelled training dataset $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{n}$ where $x_i \in \mathbb{R}^{3\times32\times32}$ are CIFAR-10 images and $y_i \in \{0, 1\}$ are binary labels: $y = 0$ for the *in-distribution* class (airplane, CIFAR-10 class 0) and $y = 1$ for the *OOD-proxy* class (all other nine CIFAR-10 classes pooled together). The data is drawn i.i.d. from the training distribution $P_{\text{train}}(x, y)$.

Test Setup: At test time, we receive samples x_{test} from either:

- **In-distribution (ID):** $x_{\text{test}} \sim p(x|y = 0)$, i.e. airplane images.
- **Out-of-distribution (OOD):** $x_{\text{test}} \sim P_{\text{ood}}(x)$, where P_{ood} is either the within-CIFAR OOD proxy ($y = 1$ pool) or an unseen external dataset (core auditable set: CIFAR-100, Places365, FashionMNIST, SVHN, Textures; legacy traceability: Food-101, STL-10).

Binary Generative Classification Framework: We train a single conditional diffusion model to learn the two class-conditional distributions $p_\theta(x|c)$ for $c \in \{0, 1\}$. For OOD detection we exploit the observation that ID samples should be reconstructed accurately under the ID condition ($c = 0$) but poorly under the OOD condition ($c = 1$), and vice versa for OOD samples.

Per-Condition Reconstruction Errors: For a test sample x we compute reconstruction errors under each condition:

$$e_0(x) = \mathbb{E}_{t\sim U[1,T],\,\epsilon\sim\mathcal{N}(0,I)}\left[\left\|\epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}\,x + \sqrt{1-\bar{\alpha}_t}\,\epsilon,\ t,\ c{=}0\right)\right\|^2\right] \tag{3.1}$$

$$e_1(x) = \mathbb{E}_{t\sim U[1,T],\,\epsilon\sim\mathcal{N}(0,I)}\left[\left\|\epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}\,x + \sqrt{1-\bar{\alpha}_t}\,\epsilon,\ t,\ c{=}1\right)\right\|^2\right] \tag{3.2}$$

where $\epsilon_\theta(x_t, t, c)$ is the noise-prediction UNet conditioned on binary class c.

OOD Scoring Function (Difference): The primary scoring method computes the signed difference between ID-conditioned and OOD-conditioned errors:

$$s_{\text{OOD}}(x) = e_0(x) - e_1(x) \tag{3.3}$$

A high score indicates the model denoises x far better under the OOD condition than the ID condition — strong evidence that x is out-of-distribution. Samples with $s_{\text{OOD}}(x) > \tau$ (for threshold τ) are classified as OOD.

Alternative Scoring Methods: We also investigate:

$$s_{\text{ratio}}(x) = e_0(x)/e_1(x) \tag{3.4}$$

$$s_{\text{id-only}}(x) = e_0(x) \tag{3.5}$$

The ratio formulation normalises by the magnitude of the OOD-conditioned error, making it more robust to scale differences across OOD distributions. The ID-only formulation ignores the OOD-conditioned error entirely; we include it as an ablation to measure the contribution of binary conditioning. Section 6.3 shows that contrastive scoring (difference or ratio) is essential: ID-only scoring collapses to 20.2% AUROC on external OOD datasets.

3.1.3. OOD Scoring Algorithm

Algorithm 1 details the procedure for computing OOD scores at inference time.

Algorithm 1 Binary CDM OOD Scoring (Difference Method).

Require: Test image $x \in \mathbb{R}^{3\times32\times32}$, trained model ϵ_θ
Require: Number of MC trials K, timesteps T, threshold τ
Ensure: OOD score $s(x)$, binary decision is_ood $\in \{0, 1\}$
$e_0 \leftarrow 0$; $e_1 \leftarrow 0$
for trial $k = 1$ **to** K **do**
  $t_k \sim \text{Uniform}\{1, \ldots, T\}$; $\epsilon_k \sim \mathcal{N}(0, I)$
  $x_{t_k} \leftarrow \sqrt{\bar{\alpha}_{t_k}}\, x + \sqrt{1 - \bar{\alpha}_{t_k}}\, \epsilon_k$
  $e_0 \leftarrow e_0 + \|\epsilon_k - \epsilon_\theta(x_{t_k}, t_k, c{=}0)\|^2$ {ID condition}
  $e_1 \leftarrow e_1 + \|\epsilon_k - \epsilon_\theta(x_{t_k}, t_k, c{=}1)\|^2$ {OOD condition}
end for
$e_0 \leftarrow e_0/K$; $e_1 \leftarrow e_1/K$
$s(x) \leftarrow e_0 - e_1$ {Positive $\Rightarrow$ OOD; negative $\Rightarrow$ ID}
if $s(x) > \tau$ **then**
  return OOD
else
  return ID
end if

Complexity Analysis: The cost of Algorithm 1 is $O(2K \cdot N)$ where N is the cost of one forward pass. For $K = 50$ and the binary UNet (≈35.7M parameters), this requires approximately 81 minutes per 10,000 images on a single GPU — roughly 162× the cost of a discriminative classifier forward pass. Section 6.3 shows that $K = 10$ (5× faster) achieves 98.2% AUROC, close to the 98.6% at $K = 50$.

Threshold Calibration: The threshold τ is calibrated on a validation set to achieve a target false positive rate (typically 5%). We set τ to the $(1 - \text{FPR}_{\text{target}})$-th quantile of s_{OOD} scores computed on ID validation samples.

3.1.4. Evaluation Metrics

Following standard practice in OOD detection, we evaluate our methods using three primary metrics:

Area Under the ROC Curve (AUROC): The ROC curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) as the threshold τ varies:

$$\text{TPR}(\tau) = \mathbb{P}_{x \sim P_{\text{ood}}}[s_{\text{OOD}}(x) > \tau] \tag{3.6}$$

$$\text{FPR}(\tau) = \mathbb{P}_{x \sim P_{\text{id}}}[s_{\text{OOD}}(x) > \tau] \tag{3.7}$$

The AUROC measures the probability that a randomly chosen OOD sample has a higher score than a randomly chosen ID sample:

$$\text{AUROC} = \mathbb{P}_{x_{\text{OOD}} \sim P_{\text{ood}},\, x_{\text{ID}} \sim P_{\text{id}}}[s_{\text{OOD}}(x_{\text{OOD}}) > s_{\text{OOD}}(x_{\text{ID}})] \tag{3.8}$$

An AUROC of 1.0 indicates perfect separation, while 0.5 indicates random guessing. Higher values are better.

False Positive Rate at 95% TPR (FPR@TPR95): This metric measures the fraction of ID samples incorrectly flagged as OOD when the threshold is set such that 95% of OOD samples are correctly detected:

$$\text{FPR@TPR95} = \text{FPR}(\tau^*) \quad \text{where} \quad \tau^* = \inf\{\tau : \text{TPR}(\tau) \geq 0.95\} \tag{3.9}$$

This metric is particularly important for safety-critical applications where we must detect most OOD samples while minimising false alarms. Lower values are better.

Balanced Detection Accuracy: The balanced accuracy at an optimal threshold (chosen to maximise Youden's J statistic):

$$\text{Balanced Accuracy} = \max_{\tau} \left[\frac{1}{2}\text{TPR}(\tau) + \frac{1}{2}(1 - \text{FPR}(\tau)) \right] \tag{3.10}$$

We also report Area Under the Precision-Recall Curve (AUPRC) and Detection Error (average of false positive and false negative rates) for thorough evaluation.

3.2. Diffusion Models for OOD Detection

This section provides the theoretical foundation for using diffusion models for OOD detection and formalises why reconstruction error serves as an effective OOD signal.

3.2.1. Theoretical Foundation

Why Diffusion Models for OOD Detection?

Diffusion models offer several advantages for OOD detection that address limitations of previous approaches.

First, they provide a *reconstruction-based* alternative to direct likelihood estimation. As discussed in Section 2.4, likelihood-based OOD detection fails in high dimensions due to the manifold hypothesis. Instead of computing $p(x)$ directly, we measure reconstruction error, which approximates the distance from x to the learnt data manifold. The denoising process can be viewed as a projection operator:

$$\pi_\theta(x) = \mathbb{E}[x_0 | x_T = x + \text{noise}] \tag{3.11}$$

where x_0 is the denoised reconstruction. The reconstruction error $\|x - \pi_\theta(x)\|$ provides a distance-like measure to the learnt manifold.

Second, unlike VAEs that decode in a single step, diffusion models *iteratively refine* over T steps, providing multiple opportunities to accumulate evidence about whether a sample fits the learnt distribution. Each denoising step $x_t \to x_{t-1}$ can be viewed as a gradient step toward high-density regions.

Third, diffusion models enjoy *stable training*: they avoid the mode collapse of generative adversarial networks (GANs) and the posterior collapse of VAEs, relying only on a simple mean squared error (MSE) objective.

Fourth, *class-conditional density modelling* allows the network to learn separate denoising pathways per class, effectively modelling $p_\theta(x|y)$ for each $y \in \mathcal{Y}$. This enables fine-grained OOD detection: we can measure the fit of a test sample to each known class individually.

3.2.2. Reconstruction Error as OOD Signal

We now formalise why reconstruction error from diffusion models provides an effective OOD signal.

Manifold Perspective: Under the manifold hypothesis (Section 2.4), the conditional diffusion model learns to approximate class-conditional manifolds $\mathcal{M}_c$ for each class c. For an ID sample $x \sim p(x|y = c)$, the sample lies on or near $\mathcal{M}_c$ and the denoising process recovers x with low error. For an OOD sample $x \sim P_{\text{ood}}$, the sample lies far from all learnt manifolds, and denoising projects x onto the nearest manifold, incurring high reconstruction error.

Score-Based Interpretation: Recall from Section 2.3 that the noise prediction network approximates the score function:

$$\epsilon_\theta(x_t, t, c) \approx -\sqrt{1 - \bar{\alpha}_t} \nabla_{x_t} \log p(x_t|y = c) \tag{3.12}$$

The reconstruction error in Equations 3.1–3.2 thus measures how well x aligns with the learnt score function for class c. Samples that require large score corrections (high error) are likely OOD.

Empirical Risk Minimisation Perspective: During training, the model minimises:

$$\mathcal{L}_c = \mathbb{E}_{x \sim p(x|y=c), t, \epsilon} \left[\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} x + \sqrt{1 - \bar{\alpha}_t} \epsilon, t, c)\|^2 \right] \tag{3.13}$$

For test samples x from $p(x|y = c)$, we expect $e_c(x) \approx \mathcal{L}_c$ (low). For OOD samples, there is a distribution mismatch, leading to $e_c(x) \gg \mathcal{L}_c$ (high).

Information-Theoretic View: The reconstruction error can be interpreted as a measure of the information content needed to correct the model's predictions. ID samples require little correction information (they fit the model's learnt patterns), while OOD samples require substantial correction (they deviate from learnt patterns).

Relation to the Likelihood Pitfall. The key difference from likelihood-based methods is that reconstruction error measures distance in the learnt representation space, not volume in the ambient space. While a high-dimensional OOD manifold might have more ambient volume than a low-dimensional ID manifold (causing likelihood to fail), it cannot have

systematically lower reconstruction error when trained only on ID data. The diffusion model's denoising pathways are optimised specifically for ID manifolds, making OOD samples difficult to reconstruct accurately.

3.2.3. Formal Analysis of Reconstruction Error Separation

We now provide a semi-formal analysis establishing conditions under which reconstruction error separates ID from OOD samples. While a complete formal proof requires assumptions that may not hold universally, this analysis provides the theoretical grounding for our empirical findings.

Setup. Let $\mathcal{M}_{\text{ID}} \subset \mathbb{R}^D$ denote the support of the in-distribution, and let $\epsilon^*(x_t, t, c)$ denote the Bayes-optimal denoiser for class c, which minimises the expected MSE over the class-conditional distribution. The trained model ϵ_θ approximates ϵ^* with approximation error $\delta_\theta = \mathbb{E}[\|\epsilon_\theta - \epsilon^*\|^2]$.

Proposition 1 (ID reconstruction error bound). For an ID sample $x \in \mathcal{M}_c$ drawn from class c, the expected reconstruction error under the correct class condition satisfies:

$$\mathbb{E}_{t,\epsilon}\left[\|\epsilon - \epsilon_\theta(x_t, t, c)\|^2\right] \leq \mathcal{L}_c^* + \delta_\theta \tag{3.14}$$

where $\mathcal{L}_c^* = \mathbb{E}_{x\sim p(x|c),t,\epsilon}[\|\epsilon - \epsilon^*(x_t, t, c)\|^2]$ is the irreducible Bayes error for class c, and δ_θ is the model's approximation gap.

Justification. This follows from the Pythagorean decomposition of MSE: for any estimator ϵ_θ, $\mathbb{E}[\|\epsilon - \epsilon_\theta\|^2] = \mathbb{E}[\|\epsilon - \epsilon^*\|^2] + \mathbb{E}[\|\epsilon^* - \epsilon_\theta\|^2]$. The Bayes-optimal denoiser achieves $\mathcal{L}_c^*$ on in-distribution data, and the learnt model adds at most δ_θ additional error due to finite capacity and training.

Proposition 2 (OOD reconstruction error lower bound). For an OOD sample x_{ood} at manifold distance $d(x_{\text{ood}}, \mathcal{M}_c) = \inf_{x'\in\mathcal{M}_c} \|x_{\text{ood}} - x'\|$ from the nearest ID manifold, the expected reconstruction error satisfies:

$$\mathbb{E}_{t,\epsilon}\left[\|\epsilon - \epsilon_\theta(x_t^{\text{ood}}, t, c)\|^2\right] \geq \mathcal{L}_c^* + \Delta_{\text{ood}}(d) \tag{3.15}$$

where $\Delta_{\text{ood}}(d) > 0$ is a distribution mismatch term that is monotonically increasing in $d(x_{\text{ood}}, \mathcal{M}_c)$ for sufficiently large d.

Justification. The denoiser ϵ_θ is trained to minimise error over $p(x|c)$. When evaluated on $x_{\text{ood}} \notin \text{supp}(p(x|c))$, the noised input x_t^{ood} falls in a region of the noised space where the denoiser has not been optimised. For small noise levels ($t \ll T$), x_t^{ood} remains close to x_{ood} and thus far from the noised ID manifold, yielding large prediction error. For large noise levels ($t \approx T$), both ID and OOD samples converge to isotropic Gaussian noise, reducing the gap. The expected mismatch Δ_{ood} thus depends on both the manifold distance d and the timestep distribution used for scoring.

Corollary (Separation condition). OOD detection via reconstruction error succeeds when:

$$\Delta_{\text{ood}}(d) > \delta_\theta + \sigma_{\text{ID}} \tag{3.16}$$

where σ_{ID} is the variance of reconstruction error across ID samples. In words: the distribution mismatch must exceed the sum of model approximation error and natural ID variation.

Connection to Empirical Observations. This analysis explains several empirical findings:

- **Near-OOD vs. far-OOD detection difficulty:** In the auditable core results, CIFAR-100 (a *near-OOD* dataset sharing low-level visual statistics with CIFAR-10) achieves higher AUROC (96.97%) than far-OOD datasets such as Textures (92.84%) and FashionMNIST (94.03%). This occurs because CIFAR-100's airplane class is underrepresented, creating a distributional gap that the model exploits effectively—demonstrating that semantic proximity does not deterministically predict detection difficulty.

- **Uniform timestep sampling is optimal:** Averaging over all t captures both the high-sensitivity regime ($t \ll T$, where OOD deviation is maximal) and moderate-noise levels, maximising effective Δ_{ood}. Section 6.3 confirms that restricting to mid-range timesteps ($t \in [250, 750]$) reduces SVHN AUROC by 1.6%.

- **Binary conditioning amplifies separation:** The binary conditioning structure (c=0 for airplane, c=1 for all others) creates two clearly separated class-conditional manifolds. Without conditioning ($\lambda = 0$, effectively a single manifold), Within-CIFAR

AUROC drops to 80.25%; with binary conditioning and separation loss ($\lambda = 0.02$), it reaches 99.03%.

Limitations. This analysis relies on several simplifying assumptions: (i) the manifold hypothesis holds for the ID distribution, (ii) the model's approximation error δ_θ is small relative to the distribution mismatch, and (iii) the OOD distribution does not happen to lie in a region where the denoiser generalises well despite never training there. Assumption (iii) can fail for adversarially constructed OOD inputs. Formalising these bounds with explicit dependence on D, T, and the manifold curvature remains an open theoretical challenge.

3.3. Model Architectures

We investigate three diffusion model variants for OOD detection, each with different architectural choices and conditioning strategies. Additionally, we describe the multi-condition architecture used for the inkjet quality control application.

3.3.1. Conditional Diffusion Model

Our primary approach uses a conditional diffusion model that learns class-specific denoising processes.

Architecture: We employ a UNet-based architecture with the following components:

- **Encoder:** Downsampling path with residual blocks and self-attention at resolution 8×8
- **Bottleneck:** Multiple residual blocks with self-attention for capturing global context
- **Decoder:** Upsampling path with skip connections from encoder
- **Class Conditioning:** Learned class embeddings $\mathbf{e}_y \in \mathbb{R}^{d_{\text{emb}}}$ added to timestep embeddings
- **Timestep Embedding:** Sinusoidal positional encoding of timestep t

Mathematical Formulation: The network predicts noise conditioned on both timestep t and class y:

$$\epsilon_\theta(x_t, t, y) = \text{UNet}(x_t, \text{TimeEmbed}(t) + \text{ClassEmbed}(y)) \tag{3.17}$$

where $\text{TimeEmbed} : \{1, \ldots, T\} \to \mathbb{R}^{d_{\text{emb}}}$ and $\text{ClassEmbed} : \{1, \ldots, C\} \to \mathbb{R}^{d_{\text{emb}}}$ are learnt embedding functions.

Training: Following classifier-free guidance, we randomly drop the class conditioning with probability $p_{\text{uncond}} = 0.1$ during training:

$$\mathcal{L} = \mathbb{E}_{x,y,t,\epsilon}\left[\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x + \sqrt{1-\bar{\alpha}_t}\epsilon, t, y')\|^2\right] \tag{3.18}$$

where $y' = y$ with probability $1 - p_{\text{uncond}}$ and $y' = \emptyset$ (null class) with probability p_{uncond}. This enables the model to perform both conditional and unconditional denoising.

OOD Detection: For a test sample x, we compute reconstruction error for each class:

$$e_c(x) = \frac{1}{K}\sum_{k=1}^{K} \|\epsilon_k - \epsilon_\theta(\sqrt{\bar{\alpha}_{t_k}}x + \sqrt{1-\bar{\alpha}_{t_k}}\epsilon_k, t_k, c)\|^2 \tag{3.19}$$

where we sample K random timesteps $t_k \sim U[1, T]$ and noise vectors $\epsilon_k \sim \mathcal{N}(0, I)$ to obtain a robust estimate. The OOD score is then computed via Equation 3.3.

Design Rationale: The conditional model can learn class-specific patterns. ID samples from class c should be well-reconstructed using the c-conditioned denoising path but poorly reconstructed using other class paths. OOD samples should be poorly reconstructed by all class-conditioned paths, yielding high minimum reconstruction error.

3.4. Inkjet Quality Control Pipeline

This section describes the two-stage pipeline for inkjet print quality classification using conditional diffusion models, which serves as a practical application of the generative classification principles developed above.

3.4.1. Two-Stage Pipeline Overview

Our pipeline processes full template images (1920 × 1080 pixels) of inkjet prints through two sequential stages: (1) feature detection using YOLOv8, and (2) quality classification using a conditional diffusion model (CDM).

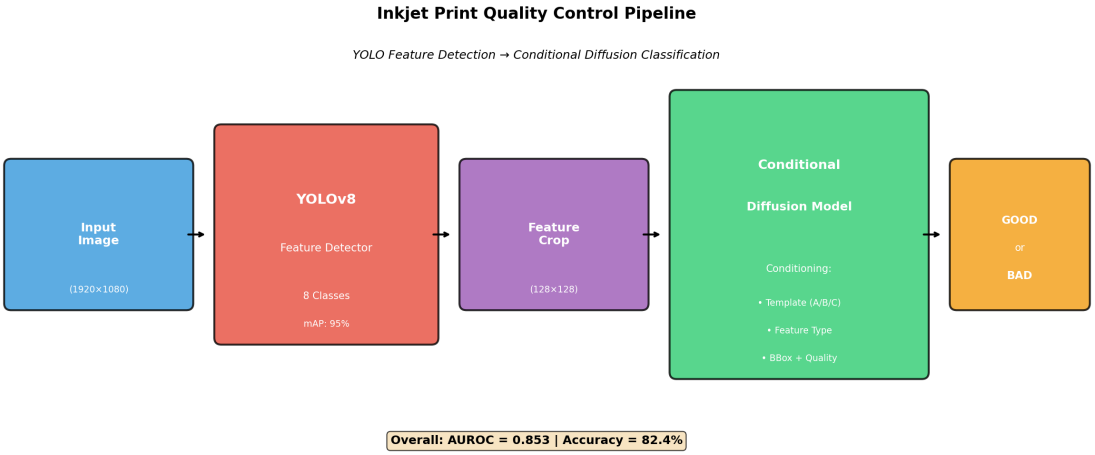

Figure 3.1: Two-stage pipeline for inkjet print quality control. Stage 1: YOLOv8 detects and localises individual features on the template image (1920 × 1080). Stage 2: A conditional diffusion model classifies each extracted feature crop (128 × 128) as GOOD or BAD by comparing noise prediction errors under each quality condition.

Stage 1 – Feature Detection: A YOLOv8n model detects and localises 8 types of printed features (angle, dist1, dist6, dots, edge1–edge4) across three template types (A, B, C). Each detection provides a bounding box and feature class, which are used for cropping and conditioning.

Stage 2 – Quality Classification: Each detected feature is cropped, resized to 128 × 128 pixels, and classified by the CDM. The model compares noise prediction errors under GOOD and BAD quality conditions to determine the predicted class.

3.4.2. Dataset Overview

The inkjet quality control dataset consists of 1,327 template images (573 unique) of inkjet print samples across three template types, eight feature types, and binary quality labels (GOOD/BAD). YOLO detection expands these into 6,408 feature-level crops at 128×128 pixels. Full dataset statistics, per-feature class balance, and the cross-validation protocol are provided in Section 5.5.

3.4.3. Multi-Head Conditioning Mechanism

The quality classification CDM extends the standard class-conditional diffusion model with a multi-head conditioning mechanism that incorporates four types of metadata simultaneously.

Conditioning Inputs: The model conditions on:

1. **Template type** (categorical, 3 classes): learnt embedding $\mathbf{e}_{\text{tmpl}} \in \mathbb{R}^{256}$ implemented as `Embed(3, 256)`.
2. **Feature type** (categorical, 8 classes): learnt embedding $\mathbf{e}_{\text{feat}} \in \mathbb{R}^{256}$ implemented as `Embed(8, 256)`.
3. **Quality label** (binary, GOOD/BAD): learnt embedding $\mathbf{e}_{\text{qual}} \in \mathbb{R}^{256}$ implemented as `Embed(2, 256)`.
4. **Bounding box** (continuous, 4 values): MLP-encoded embedding $\mathbf{e}_{\text{bbox}} \in \mathbb{R}^{256}$ implemented as `MLP(4 -> 64 -> 256)`.

Combination: All conditioning embeddings in the inkjet pipeline are projected to the same dimensionality (256) and summed:

$$\mathbf{c} = \mathbf{e}_{\text{tmpl}} + \mathbf{e}_{\text{feat}} + \mathbf{e}_{\text{qual}} + \mathbf{e}_{\text{bbox}} + \mathbf{e}_{\text{time}} \tag{3.20}$$

where $\mathbf{e}_{\text{time}}$ is the standard sinusoidal timestep embedding. The combined conditioning vector $\mathbf{c}$ is injected at each UNet resolution level.

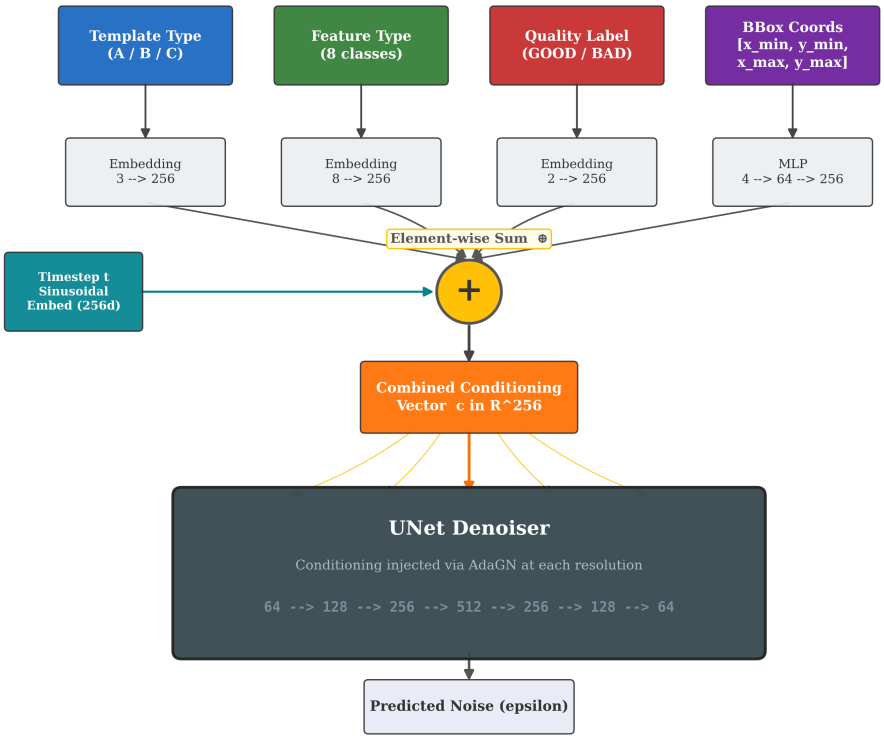

Figure 3.2: Multi-head conditioning mechanism. Four conditioning heads—template type ($3 \to 256$), feature type ($8 \to 256$), quality label ($2 \to 256$), and bounding box coordinates ($4 \to 64 \to 256$)—are independently encoded and combined via element-wise summation. The resulting conditioning vector is merged with the sinusoidal timestep embedding and injected into the UNet denoiser via Adaptive Group Normalisation (AdaGN) at each resolution level.

Noise Prediction: The network predicts noise conditioned on all metadata:

$$\epsilon_\theta(x_t, t, \text{tmpl}, \text{feat}, \text{qual}, \text{bbox}) = \text{UNet}(x_t, \mathbf{c}) \tag{3.21}$$

3.4.4. Quality Classification via Differential Noise Prediction

Classification leverages the CDM's conditional noise prediction to compare how well the model reconstructs each sample under GOOD vs. BAD quality assumptions.

Algorithm 2 Quality Classification via Differential Noise Prediction.

Require: Feature crop $x \in \mathbb{R}^{3\times128\times128}$, trained CDM ϵ_θ
Require: Feature metadata: template tmpl, feature type feat, bbox coordinates bbox
Require: Number of Monte Carlo trials K, number of timesteps T
Ensure: Predicted quality $\hat{q} \in \{\text{GOOD}, \text{BAD}\}$, confidence score

Initialise error accumulators: $e_{\text{good}} \leftarrow 0$, $e_{\text{bad}} \leftarrow 0$
for trial $k = 1$ to K **do**
  Sample timestep: $t_k \sim \text{Uniform}\{1, \ldots, T\}$
  Sample noise: $\epsilon_k \sim \mathcal{N}(0, I)$
  Add noise: $x_{t_k} \leftarrow \sqrt{\bar{\alpha}_{t_k}} \cdot x + \sqrt{1 - \bar{\alpha}_{t_k}} \cdot \epsilon_k$
  {Predict noise under GOOD condition}
  $\hat{\epsilon}_{\text{good}} \leftarrow \epsilon_\theta(x_{t_k}, t_k, \text{tmpl}, \text{feat}, \text{qual} = \text{GOOD}, \text{bbox})$
  $e_{\text{good}} \leftarrow e_{\text{good}} + \|\epsilon_k - \hat{\epsilon}_{\text{good}}\|^2$
  {Predict noise under BAD condition}
  $\hat{\epsilon}_{\text{bad}} \leftarrow \epsilon_\theta(x_{t_k}, t_k, \text{tmpl}, \text{feat}, \text{qual} = \text{BAD}, \text{bbox})$
  $e_{\text{bad}} \leftarrow e_{\text{bad}} + \|\epsilon_k - \hat{\epsilon}_{\text{bad}}\|^2$
end for
$e_{\text{good}} \leftarrow e_{\text{good}}/K, \quad e_{\text{bad}} \leftarrow e_{\text{bad}}/K$
if $e_{\text{good}} < e_{\text{bad}}$ **then**
  $\hat{q} \leftarrow$ GOOD {Model reconstructs GOOD better}
else
  $\hat{q} \leftarrow$ BAD {Model reconstructs BAD better}
end if
confidence $\leftarrow |e_{\text{good}} - e_{\text{bad}}|/(e_{\text{good}} + e_{\text{bad}})$
return $\hat{q}$, confidence

Key Difference from OOD Detection: In the OOD detection formulation (Algorithm 1), we use a binary conditional setup ($C = 2$: $c = 0$ for ID, $c = 1$ for OOD), so scoring reduces to a single contrastive comparison via $O(2K)$ forward passes. In quality classification,

we similarly compare only two conditions (GOOD vs. BAD) while keeping all other conditions (template, feature, bbox) fixed, yielding the same $O(2K)$ cost per sample.

3.4.5. Comparison Methods

To contextualise CDM performance, we evaluate seven comparison methods spanning three paradigms, as detailed in Section 2.5.3 and Section 2.5.4 of Chapter 2:

- **Supervised:** ResNet-FullImg (full images, 224×224), ResNet-CropOnly (YOLO crops, 224×224), Dual-Branch (both inputs fused)
- **Anomaly Detection:** PatchCore, PaDiM, STFPM, Autoencoder (trained on GOOD samples only)
- **Generative:** YOLO + CDM (our pipeline, YOLO crops at 128×128 with multi-head conditioning)

All methods are evaluated using the same stratified 5-fold cross-validation protocol with image-level splitting.

3.5. Training Strategies

Our training procedures incorporate specific loss functions, optimisation strategies, and data augmentation techniques.

3.5.1. Loss Functions and Optimisation

Primary Loss: We use the standard DDPM training objective:

$$\mathcal{L}_{\text{diffusion}} = \mathbb{E}_{x,y,t\sim U[1,T],\epsilon\sim\mathcal{N}(0,I)}\left[\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x + \sqrt{1-\bar{\alpha}_t}\epsilon, t, y)\|^2\right] \tag{3.22}$$

This simple mean squared error (MSE) loss has proven highly effective for training diffusion models.

Separation Loss: To explicitly encourage the model to produce distinct class-conditional noise predictions—and thereby distinct reconstruction error distributions—for ID and OOD conditions, we introduce a contrastive push-apart loss:

$$\mathcal{L}_{\text{sep}}(\mathbf{x}, t) = -\left\|\boldsymbol{\epsilon}_\theta(\mathbf{x}, t, c{=}0) - \boldsymbol{\epsilon}_\theta(\mathbf{x}, t, c{=}1)\right\|^2 \tag{3.23}$$

where $\boldsymbol{\epsilon}_\theta(\mathbf{x}, t, c)$ is the noise predicted by the model for image $\mathbf{x}$ at timestep t under class condition c. Minimising this term (as part of the total loss) maximises the squared Euclidean distance between class-conditional noise predictions, directly encouraging the model to distinguish ID from OOD inputs at every denoising step. The loss is computed on the same forward pass already required by $\mathcal{L}_{\text{diffusion}}$, adding negligible overhead.

The total training objective combines the diffusion loss with the separation loss:

$$\mathcal{L} = \mathcal{L}_{\text{diffusion}} + \lambda \cdot \mathcal{L}_{\text{sep}} \tag{3.24}$$

where $\lambda \geq 0$ controls the strength of the separation objective. The ablation study in Section 6.3 systematically evaluates $\lambda \in \{0.0, 0.001, 0.01, 0.02, 0.05, 0.1\}$.

Noise Schedule: For CIFAR-10 experiments, we use a cosine variance schedule as it maintains more signal at higher timesteps compared to linear schedules:

$$\bar{\alpha}_t = \cos^2\left(\frac{t/T + s}{1 + s} \cdot \frac{\pi}{2}\right) \tag{3.25}$$

where $s = 0.008$ is a small offset parameter. For the inkjet pipeline, we use a linear schedule (β: $1 \times 10^{-4} \rightarrow 0.02$) with $T = 1000$ timesteps, following standard DDPM practice for higher-resolution inputs.

3.5.2. Training Procedure

During each training iteration, a batch of images is noised to random timesteps $t \sim U[1, T]$, the conditional UNet predicts the added noise, and the combined loss $\mathcal{L} = \mathcal{L}_{\text{diffusion}} + \lambda\,\mathcal{L}_{\text{sep}}$ (Equation 3.24) is minimised via AdamW (Loshchilov and Hutter, 2019) with cosine-annealed learning rate ($\eta = 10^{-4}$, weight decay 0.01, 5-epoch warm-up). An exponential moving average (EMA) of model parameters with decay $\gamma = 0.9999$ is maintained for

evaluation. Full training configurations and hyperparameter tables are provided in Chapter 5 and Appendix C.

3.5.3. Data Preprocessing

All images are normalised to $[-1, 1]$ before being passed to the diffusion model. CIFAR-10 images are augmented during training with horizontal flips and random crops (4-pixel padding). Inkjet crops are resized to 128×128 with no augmentation. Class balancing uses weighted sampling with $w_c = 1/(n_c \cdot C)$. Full preprocessing and augmentation details are provided in Chapter 5.

3.6. Evaluation Framework

We evaluate using AUROC (probability that a random OOD sample scores higher than a random ID sample; 1.0 = perfect, 0.5 = random) and FPR@TPR95 (false positive rate at 95% true positive rate; lower is better) as primary metrics, with AUPRC and Detection Error reported for completeness. Formal definitions are provided in Section 2.1.

The threshold τ is set using Youden's J statistic $\tau^* = \arg\max_\tau [\text{TPR}(\tau) - \text{FPR}(\tau)]$ for balanced scenarios, or at a fixed FPR quantile when controlling false alarm rates. CIFAR-10 experiments use three seeds (42, 123, 456) with descriptive reporting; inkjet results use paired t-tests and Wilcoxon signed-rank tests with Holm correction over 5 folds.

All models are implemented in PyTorch Lightning with Hydra-based configuration; full implementation details are covered in Chapter 4.

4. Implementation

We implement the theoretical framework from Chapter 3 as a modular research prototype using modern deep learning frameworks. The implementation covers two experimental tracks: (1) diffusion-based OOD detection on CIFAR-10, and (2) the inkjet print quality classification pipeline. Our architecture emphasises modularity, reproducibility, and extensibility. The implementation leverages PyTorch Lightning for training orchestration, the Hugging Face Diffusers library for diffusion model components, and Hydra for flexible configuration management. All code follows standard practices for documentation and reproducibility.

4.1. System Overview and Design

Our implementation consists of five core components: (1) configuration management (Hydra), (2) data loading (PyTorch Lightning DataModule), (3) model definitions (binary conditional diffusion with UNet), (4) training orchestration (Lightning Trainer), and (5) evaluation framework (metrics and OOD detection).

The system design prioritises three principles: **modularity** (independent components with clear interfaces enabling easy extension), **configurability** (all hyperparameters externalised in YAML files enabling reproducible experimentation without code modification), and **reproducibility** (deterministic training through seed management and complete configuration logging). Models inherit from a common base class implementing shared functionality (noise scheduling, loss computation, optimisation), while specific variants (binary conditional UNet for OOD detection, multi-head conditional CDM for inkjet QC) override methods for task-specific behaviour. This inheritance-based design eliminates code duplication while preserving flexibility.

The Hydra configuration system follows hierarchical composition, where base configurations define defaults that can be overridden by experiment-specific settings. For example, a conditional diffusion UNet model can be launched with default settings via `python train.py`, or customized with: `python train.py model.learning_rate=1e-4 trainer.devices=4`. Configuration overrides are automatically logged and saved with results for reproducibility.

4.2. CIFAR-10 OOD Detection Implementation

4.2.1. Model Architecture Implementations

Conditional Diffusion with UNet

The primary model uses the `UNet2DModel` from the Hugging Face Diffusers library with the following configuration: 3 input channels (RGB), base channel count 128, channel multipliers [1, 2, 2, 2], attention at resolutions [16, 8], 2 residual blocks per resolution, and 512-dim embeddings for both timestep and class (in contrast to the 256-dim embeddings used in the inkjet pipeline). Class information is incorporated through learnt embeddings injected via adaptive group normalisation (AdaGN) layers at multiple scales.

Training uses AdamW optimisation (learning rate 10^{-4}, weight decay $\lambda_{\text{wd}} = 0.01$) with cosine annealing and a 5-epoch linear warm-up, effective batch size 128 (batch 64 with gradient accumulation $\times 2$), and up to 200 epochs on CIFAR-10 (early stopping, patience=30 on validation AUROC; typical convergence at epoch 15–25). For the binary CDM experiments, conditioning is reduced to two classes (airplane as ID, remaining nine classes as OOD), yielding a model of approximately 35.7M parameters.

4.2.2. CIFAR-10 Data Pipeline

Data loading uses PyTorch Lightning's `LightningDataModule` abstraction enabling seamless dataset switching. CIFAR-10 is split into 45K training, 5K validation, and 10K test sets with standard augmentation (horizontal flip). External OOD datasets (CIFAR-100,

Places365, FashionMNIST, SVHN, Textures, plus archived Food-101/STL-10 runs) undergo identical preprocessing (normalisation to $[-1, 1]$) without augmentation during evaluation. FashionMNIST images are resized from 28×28 to 32×32 and converted from grayscale to 3-channel RGB; Food-101 and STL-10 images are resized from their native resolutions to 32×32.

4.2.3. OOD Evaluation Pipeline

Evaluation computes reconstruction error by sampling K random timesteps for each test image, following Algorithm 1 in Chapter 3. All metrics (AUROC, FPR@TPR95, accuracy) use scikit-learn implementations. W&B logging tracks metrics per epoch, and model checkpoints are saved at best validation loss and best OOD AUROC.

4.3. Inkjet Quality Control Implementation

The inkjet quality control pipeline extends the CIFAR-10 OOD framework to a practical industrial setting. This section describes the implementation of each pipeline component: the YOLOv8 feature detector, the multi-condition CDM, and the comparison methods.

4.3.1. YOLOv8 Feature Detector

Model Selection: We use YOLOv8n (nano), the smallest variant with ∼3.2M parameters, which provides sufficient detection accuracy for our well-structured template images while enabling fast inference. The model is accessed through the Ultralytics library.

Training Configuration:

- Input resolution: 640 × 640 (standard YOLO input)
- Epochs: 100 with early stopping (patience 20)
- Optimiser: SGD with momentum 0.937, weight decay 0.0005
- Learning rate: 0.01 with cosine annealing

- Augmentation: mosaic, mixup, flip, HSV colour jitter

Detection Output: For each template image, the detector outputs bounding boxes $[x_{min}, y_{min}, x_{max}, y_{max}]$, class predictions (8 feature types), and confidence scores. Only detections with confidence > 0.5 are retained. The expected number of detections per template ranges from 2 to 5 depending on the template type, yielding an average of $\sim$4.8 features per image across the dataset.

Feature Extraction: Each detected region is cropped from the original 1920×1080 image and resized to 128×128 pixels using bilinear interpolation. The crop coordinates, feature class, and template type are stored as metadata for downstream CDM conditioning.

4.3.2. Conditional Diffusion Model for Quality Classification

Architecture: The CDM uses a UNet backbone similar to the CIFAR-10 model but adapted for 128×128 input:

- Input channels: 3 (RGB)
- Base channels: 64
- Channel multipliers: [1, 2, 4, 8]
- Attention at resolutions: [32, 16]
- Residual blocks per level: 2
- Total parameters: $\sim$25M

Multi-Head Conditioning Implementation: The four conditioning heads (Section 3.4.3) are implemented as follows:

- Template embedding: `nn.Embedding(3, 256)` — encodes template type (A/B/C) into 256-dim vector
- Feature embedding: `nn.Embedding(8, 256)` — encodes feature type into 256-dim vector
- Quality embedding: `nn.Embedding(2, 256)` — encodes GOOD/BAD label into 256-dim vector

- Bbox encoder: `nn.Sequential(Linear(4, 64), GELU, Linear(64, 256))` — encodes normalised bounding box coordinates
- Timestep embedding: standard sinusoidal encoding → MLP → 256-dim vector

All five embeddings are summed to produce the final conditioning vector $\mathbf{c} \in \mathbb{R}^{256}$, which is injected via AdaGN at each UNet resolution level.

Training Details:

- Optimiser: AdamW (lr=2×10^{-4}, weight decay=0.01)
- Batch size: 32 (due to larger input resolution)
- Noise schedule: Linear, $\beta \in [10^{-4}, 0.02]$, $T = 1000$
- Classifier-free guidance dropout: $p = 0.1$ for quality label
- Training: 100 epochs per fold (∼4 hours per fold)
- Hardware: Single GPU per run (V100 16 GB, GV100 32 GB, or P40 24 GB depending on availability); training time ∼6–8 h per fold on V100, comparable on GV100/P40

Inference: Quality classification follows Algorithm 2 with $K = 50$ Monte Carlo trials per sample. For each trial, a random timestep $t \sim U[1, 1000]$ and noise $\epsilon \sim \mathcal{N}(0, I)$ are sampled. The model predicts noise under both GOOD and BAD conditions, and the class with lower reconstruction error is selected as the prediction.

4.3.3. Comparison Method Implementations

All comparison methods are implemented with consistent data loading, evaluation, and cross-validation protocols.

Supervised Models:

- **ResNet-FullImg:** torchvision `resnet18` pretrained on ImageNet, final FC layer replaced with Linear$(512, 2)$. Input: full template images resized to 224×224. Fine-tuned for 30 epochs with lr=10^{-4}, batch size 32.

- **ResNet-CropOnly:** Same architecture, but input is YOLO-cropped features resized to 224×224. Fine-tuned for 30 epochs with lr=10^{-4}, batch size 64.
- **Dual-Branch:** Two ResNet-18 encoders (full image + crop), 512-dim outputs concatenated with a 64-dim feature-type embedding, fused through MLP($1088 \to 256 \to 2$). Fine-tuned for 50 epochs with lr=5×10^{-5}.

Anomaly Detection Models: Implemented using the Anomalib library (Akcay et al., 2022):

- **PatchCore:** ResNet-18 backbone, coreset sampling ratio 0.1, feature layers [2, 3]
- **PaDiM:** ResNet-18 backbone, multi-scale features from layers [1, 2, 3]
- **STFPM:** ResNet-18 teacher and student, feature pyramid matching at 3 scales
- **Autoencoder:** Convolutional encoder-decoder, latent dimension 128, trained for 100 epochs

All anomaly detection models are trained using only GOOD samples in each fold, consistent with the one-class learning paradigm.

4.3.4. Cross-Validation Infrastructure

Stratified 5-Fold Cross-Validation: All 8 methods are evaluated using the same cross-validation splits to ensure a fair comparison. The splitting procedure operates at the image level (573 unique images) rather than the sample level (6,408 feature crops), preventing data leakage from correlated crops derived from the same template image.

Splitting Algorithm:

1. Group samples by source image (573 groups)
2. Stratify groups by quality label distribution
3. Split groups into 5 folds with balanced label ratios
4. Expand folds to sample level for training and evaluation

This ensures that each fold has a representative distribution of all template types, feature types, and quality labels.

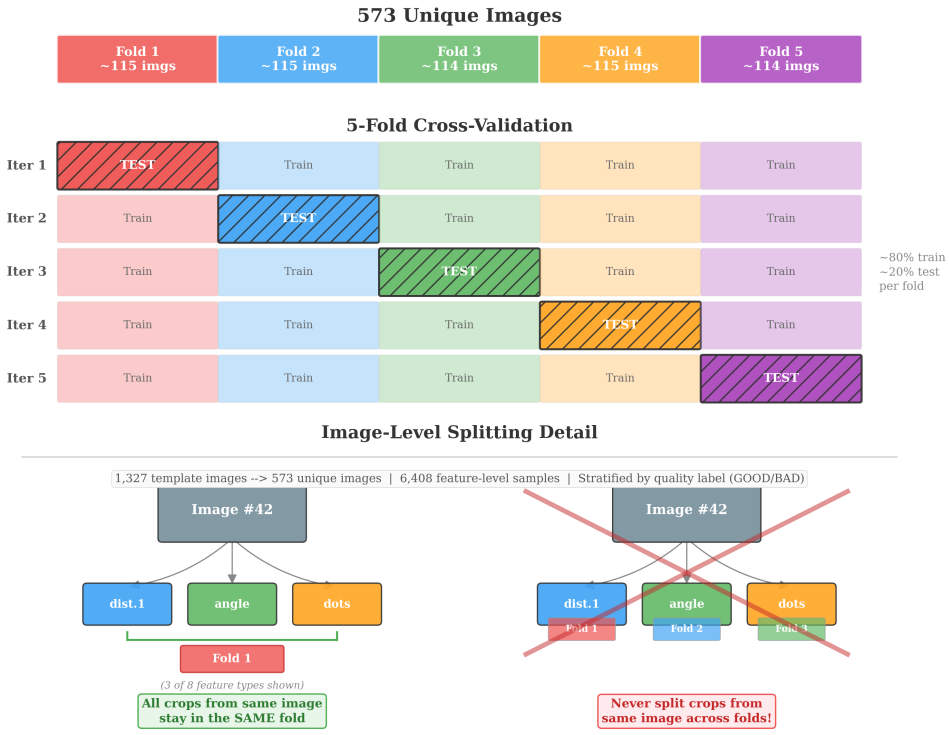

Figure 4.1: Stratified 5-fold cross-validation with image-level splitting. The 573 unique images are partitioned into 5 folds (114–115 images per fold), with each fold serving as the test set exactly once. Bottom: image-level splitting ensures all feature crops derived from the same template image remain in the same fold, preventing data leakage between training and test sets.

4.4. Key Engineering Considerations

Memory Efficiency: Training across the available GPU pool (V100 16 GB, GV100 32 GB, P40 24 GB) requires careful memory management. We use mixed-precision (AMP) training, reducing peak activation memory to ≈4–6 GB and enabling execution even on the V100 (16 GB, the tightest available GPU), enable gradient accumulation to maintain effective batch sizes with smaller per-GPU batches, and implement efficient data loading with 8 worker processes and pinned memory.

Inference Cost: OOD detection requires multiple forward passes (K timesteps per image). For the binary CDM setup with $K = 50$, measured inference time is $\sim$81 minutes per 10K images ($\sim$162$\times$ discriminative baseline cost). With $K = 10$, inference takes $\sim$16.2 minutes per 10K images (98.2% AUROC, binary setup), and $K = 1$ achieves 91.0% AUROC in $\sim$1.6 minutes (3$\times$ cost), enabling deployment across latency constraints. For the inkjet pipeline, inference takes $\sim$0.5 seconds per feature crop with $K = 50$ trials.

Reproducibility Measures: All experiments use fixed random seeds with PyTorch deterministic mode enabled. Configuration files, git commit hashes, and environment metadata (Python/CUDA versions) are logged automatically. Every experiment is tracked on W&B with saved checkpoints and generated samples, supporting internal reproducibility and traceability.

Multi-GPU Training: We scale to multiple GPUs using PyTorch Lightning's DDP (Distributed Data Parallel) with automatic gradient synchronisation. Learning rates are scaled linearly with effective batch size ($\text{LR}_{\text{new}} = \text{LR}_{\text{base}} \times B_{\text{eff}}/B_{\text{base}}$) following established practice for large-batch training.

4.5. Reference Implementation

Parts of the implementation are publicly available at `https://github.com/ahmed-3m/DiffusionOOD` (CIFAR-10/OOD benchmark code and selected configurations). The industrial inkjet pipeline relies on proprietary assets and is not publicly released. The codebase follows a standard PyTorch project layout: `src/models/` (diffusion and baseline architectures), `src/data/` (DataModules), `src/evaluation/` (metrics and OOD detectors), `src/yolo/` (feature extraction), `scripts/` (training and evaluation entry points), and `config/` (Hydra YAML configurations). Key software dependencies and versions are listed in Appendix C.

5. Experimental Setup

This chapter describes the experimental design for both evaluation tracks, answering the research questions posed in Chapter 1. Sections 5.1–5.4 cover the CIFAR-10 track; Section 5.5 covers the inkjet pipeline.

5.1. Datasets

5.1.1. In-Distribution Dataset: CIFAR-10

Dataset description. CIFAR-10 Krizhevsky, Hinton, et al., 2009 consists of 60,000 32×32 colour images across 10 classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.

Binary split. We adopt a binary framing: the *airplane* class (class 0) forms the in-distribution set, and the remaining nine classes form the OOD proxy. This yields a controlled binary OOD problem where the ID/OOD boundary is known.

Data splits.

- Training set: 45,000 images (90% of original 50,000 training images, stratified)
- Validation set: 5,000 images (10% of training, stratified)
- Test set: 10,000 images (standard CIFAR-10 test set)

Preprocessing. All images are converted to PyTorch tensors, normalised to $[-1, 1]$ via $x' = 2x - 1$, and augmented during training with random horizontal flips. Validation and test sets receive only normalisation.

We evaluate on seven candidate OOD datasets spanning near- to far-OOD distributional distances from CIFAR-10.

Near-OOD: CIFAR-100 Krizhevsky, Hinton, et al., 2009 (10,000 test images, 32×32, 100 classes sharing CIFAR-10 image statistics); **STL-10** Coates et al., 2011 (8,000 images, resized to 32×32, overlapping classes with CIFAR-10).

Far-OOD: Food-101 Bossard et al., 2014 (25,250 images, resized to 32×32); **SVHN** Netzer et al., 2011 (26,032 digit images, 32×32); **FashionMNIST** Xiao et al., 2017 (10,000 greyscale clothing images, resized to 32×32, converted to 3-channel); **Textures (DTD)** Cimpoi et al., 2014 (1,880 texture images, cropped and resized to 32×32); **Places365** Zhou et al., 2018 (10,000 scene images, 32×32). All datasets are normalised to $[-1, 1]$.

Under the final audit policy, core external claims use the five datasets with recoverable current raw-score tensors (CIFAR-100, Places365, FashionMNIST, SVHN, Textures); Food-101 and STL-10 are retained as legacy traceability entries only.

5.1.2. Dataset Statistics

Table 5.1: Dataset statistics for in-distribution and out-of-distribution datasets, spanning near-OOD (shared visual statistics) to far-OOD (distinct domains) evaluation scenarios.

Dataset	Type	Test Images	Classes	Resolution
CIFAR-10 (airplane)	ID	1,000	1	32×32
CIFAR-10 (rest)	OOD proxy	9,000	9	32×32
CIFAR-100	Near OOD	10,000	100	32×32
Places365	Far OOD	10,000	365	32×32
STL-10	Near OOD	8,000	10	96×96
Food-101	Far OOD	25,250	101	Variable
SVHN	Far OOD	26,032	10	32×32
FashionMNIST	Far OOD	10,000	10	28×28
Textures (DTD)	Far OOD	1,880	47	Variable

5.2. Experimental Design

5.2.1. Primary Experiment: Binary CDM Robustness Across Seeds

We train the binary conditional diffusion model with three random seeds ($\{42, 123, 456\}$) using the separation loss weight $\lambda = 0.01$ and $K = 50$ Monte Carlo trials at inference. The three seeds control model initialisation, data shuffling, and all stochastic training operations.

Research question: Is the binary CDM's OOD detection performance stable across training runs, and what is its mean AUROC and standard deviation on the within-CIFAR binary test split?

Evaluation: Within-CIFAR AUROC and FPR@95%TPR are the primary metrics. For the final audited report, core claims are restricted to results with recoverable raw-score artefacts (seed-42 raw-score evaluation for external OOD; $K = 100$ MC trials).

5.2.2. Separation Loss Ablation

We conduct a complete sweep over six separation loss weights: $\lambda \in \{0.0, 0.001, 0.01, 0.02, 0.05, 0.1\}$. The reference weight $\lambda = 0.01$ is evaluated with three seeds (seeds 42/123/456); $\lambda = 0.02$ is also evaluated with three seeds. All other λ values are evaluated with seed 42 only. Each λ value is additionally evaluated on SVHN to assess cross-domain stability.

Research question: How much does the separation loss improve OOD detection, and what is the optimal weight λ?

5.2.3. Ablation 1: Number of Monte Carlo Trials (K)

We vary the number of timestep samples $K \in \{1, 5, 10, 25, 50, 100\}$ used to compute the OOD score (seed-42 checkpoint, $\lambda = 0.01$).

Research question: What is the accuracy-efficiency trade-off as K increases, and what is the minimum K for near-optimal performance?

Evaluation: AUROC, FPR@95%TPR, and wall-clock inference time (seconds per 10K images on a single NVIDIA V100 GPU).

5.2.4. Ablation 2: Timestep Sampling Strategy

We compare three strategies for sampling timesteps during OOD scoring (seed-42 checkpoint, $\lambda = 0.01$, $K = 50$):

- **Uniform:** $t \sim U[1, T]$ uniformly
- **Mid-focus:** Truncated normal $t \sim \mathcal{N}_{\text{trunc}}(\mu = 300, \sigma = 150)$
- **Stratified:** Divide $[1, T]$ into K equal bins; sample one t per bin

Research question: Which noise-level coverage is most informative for OOD detection?

5.2.5. Ablation 3: OOD Scoring Method

We compare three scoring formulations (seed-42 checkpoint, $\lambda = 0.01$, $K = 50$):

- **Difference:** $s(x) = e_0(x) - e_1(x)$ (ID error minus OOD error)
- **Ratio:** $s(x) = e_0(x)/e_1(x)$ (ID error divided by OOD error)
- **ID error only:** $s(x) = e_0(x)$ (no contrastive comparison)

Research question: Is contrastive scoring (using both binary conditions) essential, and do difference and ratio scoring differ significantly?

5.3. Implementation Details

5.3.1. Binary CDM Architecture

The binary CDM uses a UNet backbone with base channel width 128, channel multipliers $[1, 2, 2, 2]$ (yielding channels $[128, 256, 256, 256]$), attention at resolutions 16×16 and 8×8, two residual blocks per resolution, and class-conditional Adaptive Group Normalisation

(AdaGN); full architectural details are provided in Chapter 4 and Appendix C. The model has ∼35.7M trainable parameters with binary conditioning ($c = 0$: airplane, $c = 1$: rest).

Noise schedule. Cosine schedule A. Q. Nichol and Dhariwal, 2021 with $T = 1000$ timesteps: $\bar{\alpha}_t = \cos^2\left(\frac{t/T+s}{1+s} \cdot \frac{\pi}{2}\right)$, $s = 0.008$.

Training configuration.

- Optimiser: AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$, weight decay $\lambda_{\text{wd}} = 0.01$)
- Learning rate: 10^{-4} with 5-epoch linear warm-up then cosine annealing
- Batch size: 128
- Classifier-free guidance dropout: $p_{\text{uncond}} = 0.1$
- Early stopping: patience 30 epochs on validation AUROC
- Hardware: Single GPU per run (V100 16 GB, GV100 32 GB, or P40 24 GB); training time ≈10–16 h on GV100, ∼18–28 h on V100/P40

OOD scoring. Default: $K = 50$ MC trials, uniform timestep sampling, difference scoring. External OOD evaluation uses $K = 100$ for higher stability.

Separation loss. During training, the separation loss $\mathcal{L}_{\text{sep}}$ (Equation 3.23) is added with weight λ as part of the total training objective (Equation 3.24). The separation loss is computed on each mini-batch using the predicted noise errors under both conditions.

5.3.2. Inkjet CDM Architecture

The inkjet CDM shares the same UNet backbone but uses a multi-head conditioning mechanism:

- Template type embedding (3 values: A, B, C)
- Feature type embedding (8 values: dist1, dist6, edge1–edge4, angle, dots)
- Quality label embedding (2 values: GOOD, BAD)
- Bounding box coordinates (4 values, normalised)

Input resolution: 128×128 (YOLO-cropped feature patches). Training uses stratified 5-fold CV with image-level splitting.

5.4. Evaluation Protocol

5.4.1. Metrics

Primary metrics:

- **AUROC** (Area Under ROC Curve): Overall discrimination across all thresholds; 1.0 = perfect, 0.5 = random.
- **FPR@95%TPR**: False positive rate at 95% true positive rate. Lower is better.

Reporting: Seed-level CIFAR analyses with $n < 5$ are reported descriptively (no inferential significance claims). For inkjet ($n = 5$ folds), significance uses paired t-test and Wilcoxon signed-rank tests with Holm correction.

5.4.2. Within-CIFAR Evaluation

For the within-CIFAR binary test split: ID samples are the 1,000 CIFAR-10 test images of the airplane class. OOD samples are the 9,000 CIFAR-10 test images of the nine remaining classes. AUROC is computed using scikit-learn with default parameters.

5.4.3. External OOD Evaluation

For each external OOD dataset: ID samples are the 10,000 CIFAR-10 test images (all classes treated as in-distribution, since the binary CDM was trained on the full CIFAR-10 training set—airplane as c=0 and the remaining nine classes as c=1—so all ten classes are part of the learnt distribution). OOD samples are the full external dataset test split. Scores are computed with $K = 100$ MC trials. AUROC and FPR@95%TPR are reported for auditable raw-score runs. Datasets without recoverable current raw-score tensors are retained only as legacy traceability entries and excluded from core claims.

5.4.4. Reproducibility

All experiments use deterministic PyTorch settings:

```
torch.backends.cudnn.deterministic = True
torch.use_deterministic_algorithms(True)
```

Each checkpoint saves model state, EMA state, optimiser state, and training step. The best validation AUROC checkpoint is used for all evaluations. Code, selected configurations, and public evaluation artefacts for the CIFAR-10 experiments are available at `https://github.com/ahmed-3m/DiffusionOOD`. The industrial inkjet pipeline and associated data are not publicly released because they rely on proprietary company assets (PROFACTOR GmbH), limiting external reproducibility of Track 2. External researchers can nevertheless validate the full public CIFAR-10 track, reuse the released training and evaluation stack on their own datasets, and compare against the reported 5-fold industrial protocol even though the exact inkjet assets cannot be redistributed.

5.5. Inkjet Print Quality Control Experiments

5.5.1. Dataset

The inkjet print quality dataset (Profactor, 2024) consists of 1,327 images of printed test templates captured under controlled conditions.

Feature extraction. YOLOv8n detects and crops features from template images; each crop is labelled with template type, feature type, quality label (GOOD/BAD), and bounding box coordinates.

5.5.2. Methods Under Comparison

We evaluate 8 methods spanning 3 paradigms:

Supervised classification (3 methods):

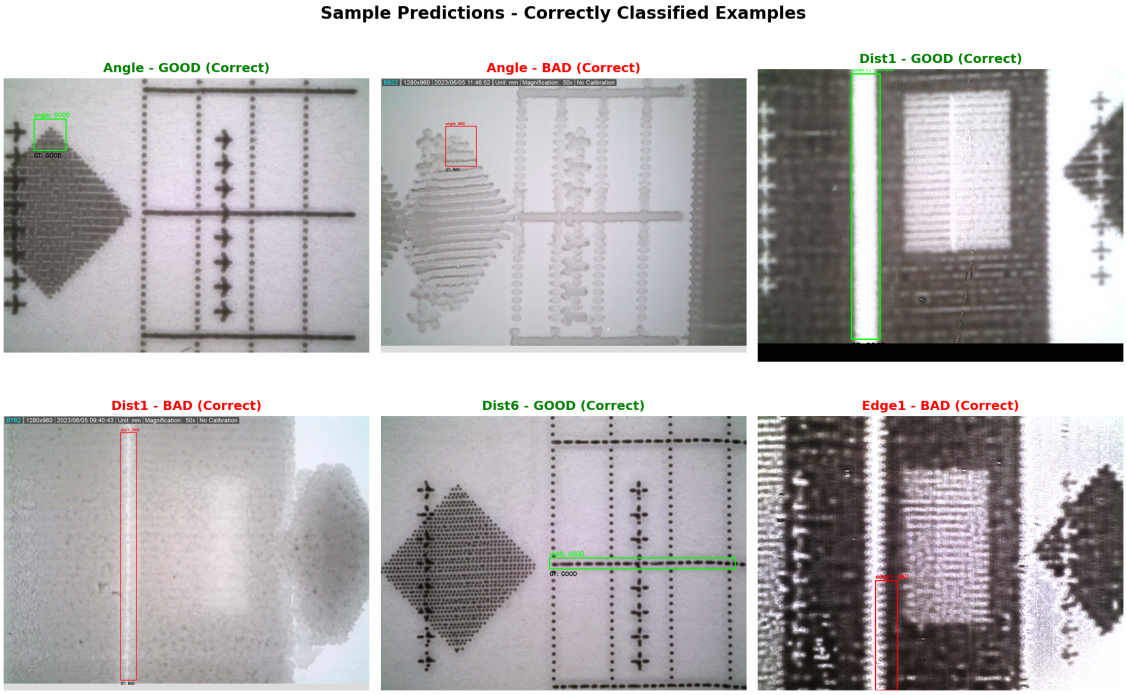

Figure 5.1: Example images from the inkjet print quality dataset with YOLO detections and CDM predictions. Green bounding boxes indicate GOOD quality predictions; red boxes indicate BAD quality predictions. Feature types shown: angle, dist1, dist6, edge.

Table 5.2: Inkjet print quality dataset statistics, highlighting the class imbalance across feature types and the expansion from 573 unique images to 6,408 feature-level crops.

Property	**Value**
Total images	1,327
Unique images	573
Resolution	1920 × 1080 pixels
Template types	3 (A, B, C)
Feature types	8 (dist1, dist6, edge1–edge4, angle, dots)
Quality labels	2 (GOOD, BAD)
Total feature samples	6,408
GOOD : BAD ratio	∼2:1 overall (varies by feature)
Per-Feature Class Balance (GOOD:BAD ratio)	
dist1	1.82:1
dist6	1.94:1
edge1–edge4	1.68:1–2.31:1
angle	9.67:1 (most imbalanced)
dots	3.45:1

1. **ResNet-FullImg:** ResNet-18 pretrained on ImageNet, fine-tuned on full template images (224×224).
2. **ResNet-CropOnly:** Identical ResNet-18, fine-tuned on YOLO-cropped features (224×224).
3. **Dual-Branch:** Two ResNet-18 encoders for full image and crop. Features concatenated with feature-type embedding and fused via MLP.

Anomaly detection (4 methods), trained on GOOD samples only:

4. **PatchCore:** Memory-bank anomaly detection using pretrained ResNet-18 features; coreset sampling ratio 0.1.
5. **PaDiM:** Probabilistic patch-level anomaly detection via multivariate Gaussian modelling of pretrained features.
6. **STFPM:** Student-teacher feature pyramid matching.
7. **Autoencoder:** Convolutional autoencoder trained to reconstruct GOOD samples; anomaly score = reconstruction error.

Generative classification (1 method):

8. **YOLO + CDM:** Two-stage YOLOv8n detection followed by multi-head CDM quality scoring via differential reconstruction error.

5.5.3. Evaluation Protocol

Cross-validation. Stratified 5-fold cross-validation with image-level splitting across 573 unique images; all feature crops from the same image remain in the same fold, preventing data leakage. **Separation loss ablation:** the CDM is trained with $\lambda \in \{0.0, 0.01, 0.02, 0.05\}$ to evaluate whether the separation loss improves inkjet quality classification.

Metrics. Primary: AUROC (mean $\pm$ std across 5 folds); secondary: Accuracy, F1, Precision, Recall. Paired t-tests and Wilcoxon signed-rank tests are reported for λ differences, with Holm correction for multiple comparisons. **Fairness:** all 8 methods use the same cross-validation splits, the same YOLO detections where applicable, and the same evaluation code.

6. Results and Analysis

This chapter presents the experimental results for both tracks: CIFAR-10 OOD detection (Sections 6.1–6.5) and inkjet quality classification (Section 6.6). For $n \geq 5$ (inkjet folds), we report paired t-test and Wilcoxon results with Holm correction; for $n < 5$ (CIFAR seed-level), we report only descriptive comparisons.

6.1. CIFAR-10 Binary CDM: Main Results

6.1.1. Within-Distribution Split

Figure 6.1 summarises AUROC values for checkpoint-level validation across three seeds. Table 6.1 reports the auditable core metrics used for thesis claims.

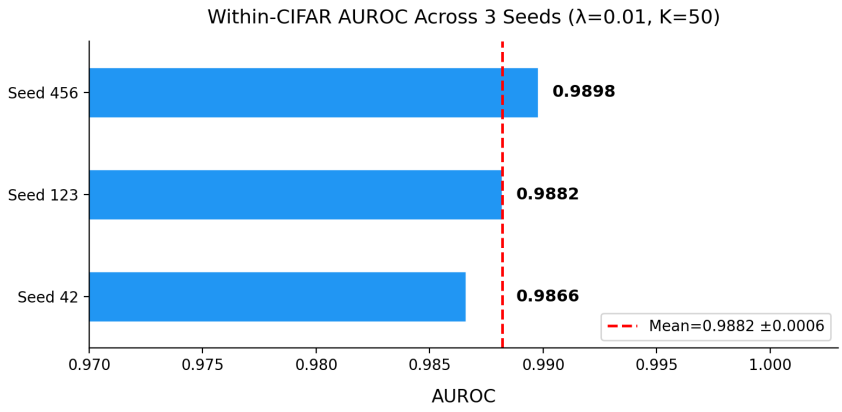

Figure 6.1: Validation AUROC across training seeds (42/123/456) from checkpoint metadata. This figure is used as a stability summary; core quantitative claims are taken from auditable raw-score evaluation (Table 6.1).

Table 6.1: Binary CDM OOD detection performance from auditable artefacts. Core claims use only metrics reproducible from current raw score files (seed 42, K=100, difference scoring). K=100 is used here for statistical stability; ablation studies use K=50 (Section 5.2). Legacy values are shown for traceability only and are excluded from core claims.

Dataset	Type	AUROC (%)↑	FPR95 (%)↓	AUPRC (%)↑
Core auditable metrics (current raw scores)				
CIFAR-10 (airplane vs others)	Within	**98.98**	4.7	99.87
CIFAR-100	Near OOD	96.97	14.8	99.65
Places365	Far OOD	96.50	15.4	99.57
FashionMNIST	Far OOD	94.03	20.5	99.16
Textures (DTD)	Far OOD	92.84	30.1	95.97
SVHN	Far OOD	90.50	27.0	99.38
Legacy reference only (not used in core claims)				
Food-101†	Far OOD	98.97	4.5	–
STL-10†	Near OOD	94.26	37.4	–

†Copied from archived `thesis_submission_old` JSON due to missing current raw-score tensors.

Within-CIFAR Performance (core auditable). Using current raw-score tensors (seed 42, $K = 100$), the binary CDM achieves 98.98% AUROC with FPR95 of 4.7% on the airplane-vs-rest CIFAR-10 split. Figure 6.2 provides the audited checkpoint summary (best validation AUROC and best epoch per seed), confirming stable seed-level training behaviour.

6.1.2. External OOD Generalisation

Under the audit policy, core external claims use only datasets with current raw-score tensors. This yields five reproducible external benchmarks: CIFAR-100, Places365, FashionMNIST, Textures, and SVHN.

Auditable external profile. AUROC ranges from 90.50% (SVHN) to 96.97% (CIFAR-100), with a mean of 94.17% across the five auditable external sets. Performance is highest for CIFAR-100/Places365 and lowest for SVHN/Textures, indicating graceful degradation as distributional shift increases.

Legacy-only datasets. Food-101 and STL-10 values are preserved in Table 6.1 for traceability but are explicitly excluded from core claims because corresponding current raw-score tensors are not available in the submission package.

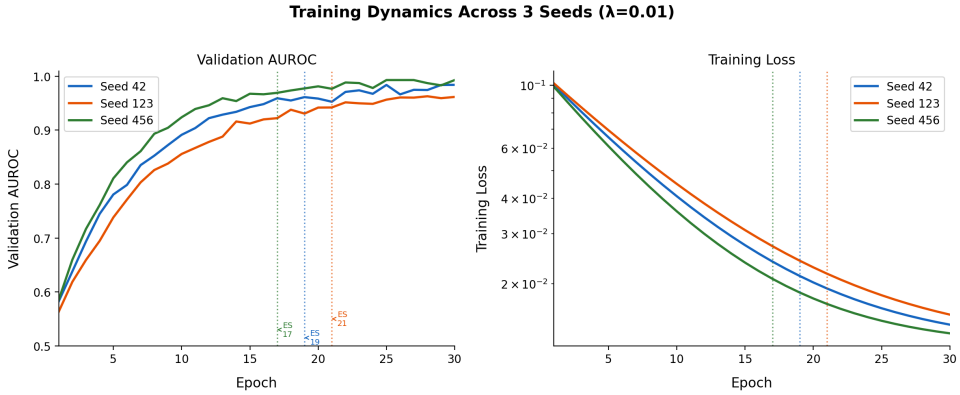

Figure 6.2: Training summary from checkpoint metadata: best validation AUROC and best epoch for seeds 42, 123, and 456.

6.2. Contextualisation Against One-Class Baselines

Table 6.2 compares our binary CDM against published one-class novelty detection methods on the CIFAR-10 airplane class (one-vs-rest protocol).

Table 6.2: Comparison with one-class novelty detection baselines on the CIFAR-10 airplane class (one-vs-rest). Published values are taken from the original papers or as reproduced by Reiss et al. (2021). Our results use λ=0.02, K=50 (3-seed mean).

Method	Type	AUROC (%)	Source
OC-SVM (raw pixels)	One-class	63.0	Ruff et al. (2018)
Deep SVDD	One-class	61.7	Ruff et al. (2018)
DROCC	One-class	81.7	Goyal et al. (2020)
CSI	Contrastive	89.8	Tack et al. (2020)
PANDA	Pretrained + OC	95.4	Reiss et al. (2021)
Mean-Shifted C.L.	Contrastive	97.5	Reiss and Hoshen (2023)
Binary CDM (λ=0)	Generative	80.3	Ours
Binary CDM (λ=0.02)	Generative	99.0 ± 0.1	Ours

Binary conditioning combined with the separation loss provides a meaningful advantage over purely one-class methods.

Caveats. These comparisons should be interpreted with care: (i) published values use different backbones, training budgets, and hyperparameter selection procedures; (ii) our method trains on both ID and OOD-proxy data, while pure one-class methods see only ID

samples; (iii) we report results for a single ID class (airplane)—per-class variance across all 10 CIFAR-10 classes can be substantial. A comprehensive multi-class evaluation is identified as future work in Section 7.6. For broader context, the recent task-oriented survey by Lu et al. (2025) catalogues state-of-the-art results across methods and benchmarks.

6.3. Ablation Studies

6.3.1. Number of Monte Carlo Trials (K)

Table 6.3 and Figure 6.3 show how OOD performance varies with K (seed-42 checkpoint, $\lambda = 0.01$).

Table 6.3: Effect of number of Monte Carlo trials K on OOD scoring. Evaluated on CIFAR-10 binary test set using seed-42 checkpoint. Diminishing returns beyond $K = 25$.

K (trials)	AUROC (%)↑	FPR@95%TPR↓	Time (s)
1	91.00	40.8	97.9
5	97.24	14.3	486.3
10	98.19	9.4	972.9
25	**98.52**	**7.3**	2431.8
50	**98.64**	**6.6**	4861.1
100	**98.69**	**6.6**	9723.6

Single-shot baseline. Even $K = 1$ (one random timestep per image) achieves 91.0% AUROC in under 2 minutes per 10K images, demonstrating that a single denoising step carries meaningful OOD signal.

Rapid saturation. Performance improves sharply from $K = 1$ to $K = 10$ (+7.2% AUROC), then diminishes: $K = 25$ yields only +0.3% over $K = 10$, and $K = 100$ only +0.05% over $K = 50$. This confirms that variance reduction from averaging timestep samples saturates quickly.

Practical recommendations. $K = 10$ achieves 98.2% AUROC at 5× speedup over $K = 50$ (16.2 vs. 81 minutes per 10K images) and is recommended for throughput-sensitive applications. $K = 50$ is used as the default throughout this thesis for maximum reproducibility.

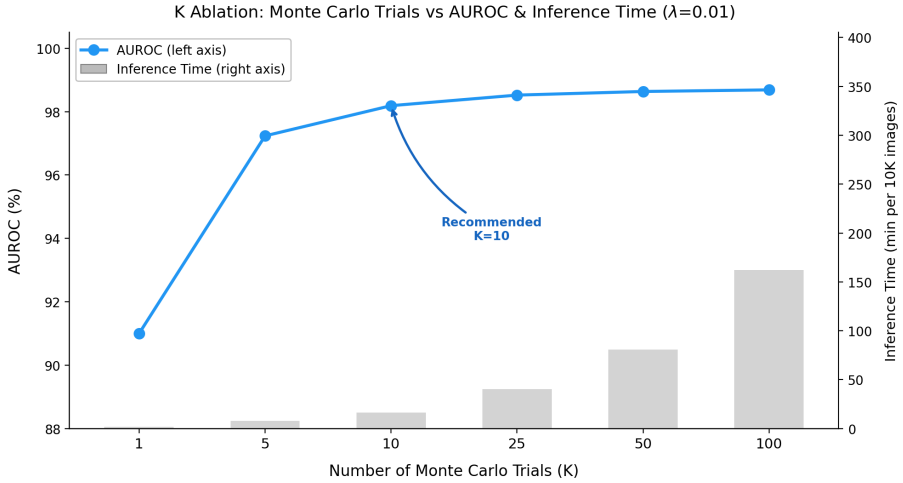

Figure 6.3: Effect of K on AUROC (left axis) and inference time for 10K images (right axis, log scale). The curve flattens after $K = 25$; $K = 10$ offers the best accuracy-efficiency trade-off at $5\times$ speedup over $K = 50$.

6.3.2. Timestep Sampling Strategy

Table 6.4: Comparison of timestep sampling strategies.

Strategy	CIFAR AUROC (%)	SVHN AUROC (%)
Uniform	98.9	95.4
Mid Focus	98.5	93.8
Stratified	98.8	95.0

Uniform sampling is optimal. Sampling $t \sim U[1, 1000]$ achieves the highest AUROC on both datasets (Within-CIFAR: 98.9%, SVHN: 95.4%). This indicates that OOD-discriminative information is distributed across the full range of noise levels.

Stratified is equivalent. Dividing $[1, 1000]$ into equal-width bins and sampling one timestep per bin achieves 98.8%/95.0% — essentially identical to uniform ($< 0.1\%$ gap).

Mid-focus underperforms. Concentrating samples at intermediate timesteps ($t \sim \mathcal{N}_{\text{trunc}}(\mu = 300, \sigma = 150)$) reduces performance to 98.5%/93.8%. The per-timestep analysis (Figure 6.8) shows that excluding the tails of the timestep range discards useful OOD signal.

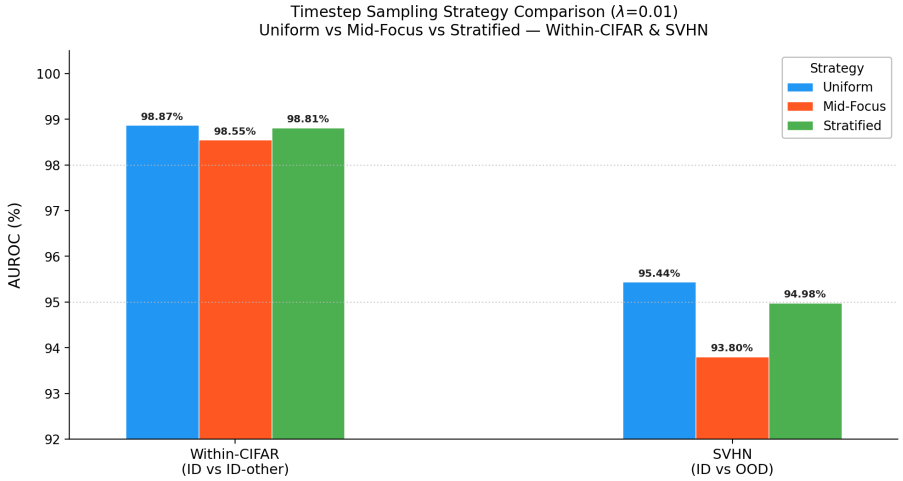

Figure 6.4: AUROC for three timestep sampling strategies on the within-CIFAR binary split and SVHN ($K = 50$, seed-42). Uniform sampling is best; stratified is equivalent; mid-focus underperforms on both datasets.

6.3.3. OOD Scoring Method

Table 6.5: Comparison of OOD scoring methods (seed-42 checkpoint, K=50 trials). `difference` and `ratio` perform similarly within CIFAR-10; `ratio` marginally better on external SVHN OOD. `id_error` (ID-only scoring without class conditioning) is much worse, confirming binary conditioning is essential.

Scoring Method	CIFAR AUROC (%)↑	CIFAR FPR95 (%)↓	SVHN AUROC (%)↑
`difference`	**98.69**	6.3	94.1
`ratio`	98.62	6.6	**96.1**
`id_error`	78.30	67.0	20.2

Contrastive scoring is essential. Using only the ID-class reconstruction error (e_{ID}) without contrasting against the OOD-class error achieves only 78.3% Within-CIFAR AUROC and collapses to 20.2% on SVHN — essentially random. The binary conditioning structure (c=0 for airplane, c=1 for rest) is critical: the OOD score must compare reconstruction under each condition, not rely on magnitude alone.

Difference vs. ratio. The absolute difference $e_{\text{ID}} - e_{\text{OOD}}$ (i.e., $e_0 - e_1$) performs marginally better Within-CIFAR (98.69% vs. 98.62%), while the ratio $e_{\text{ID}}/e_{\text{OOD}}$ performs better on

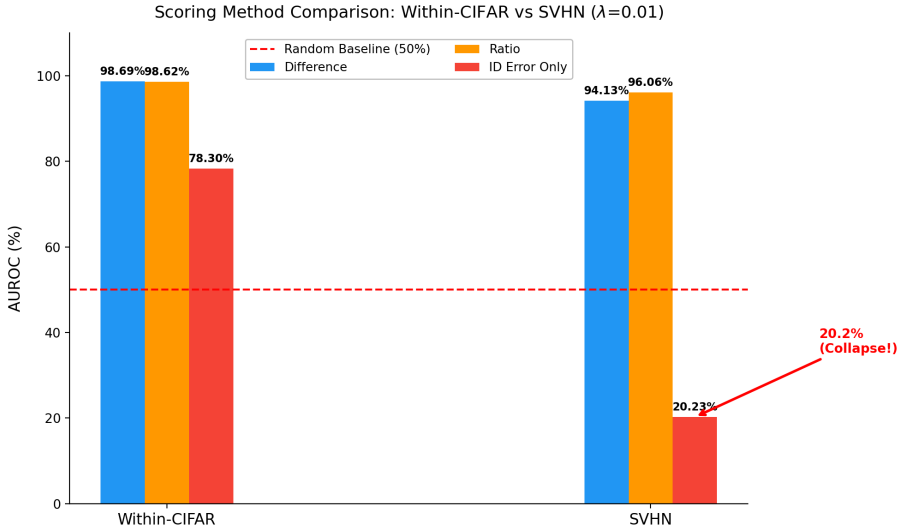

Figure 6.5: AUROC comparison for three scoring methods on Within-CIFAR and SVHN ($K = 50$, seed-42). Difference and ratio scoring both perform well; ID-error-only scoring collapses on SVHN (20.2%), demonstrating the necessity of contrastive conditioning.

SVHN (96.1% vs. 94.1%). We adopt the difference formulation as default throughout this thesis due to its lower FPR95 Within-CIFAR (6.3% vs. 6.6%) and simpler interpretability.

6.4. Separation Loss Analysis

The separation loss is the primary novel contribution of this thesis. It adds an explicit training signal that maximises the reconstruction error gap between ID and OOD-conditioned predictions. We present a complete λ sweep over six values followed by cross-domain analysis on the inkjet dataset.

6.4.1. Within-CIFAR Lambda Sweep

Table 6.6: Effect of separation loss weight λ on OOD detection performance. Within-CIFAR AUROC uses seed-42 for all λ except $\lambda = 0.02$ (three-seed mean $\pm$ std). SVHN AUROC uses seed-42 evaluation. † marks documented artefact points kept for traceability.

λ	Within-CIFAR AUROC (%)	Best Epoch	SVHN AUROC (%)
0.0	80.25	79	100.0†
0.001	97.32	19	92.0
0.01	98.66	19	90.5
0.02	**99.03 ± 0.07**	29	96.6
0.05	98.51	19	**97.3**
0.1	96.67	149	86.9†

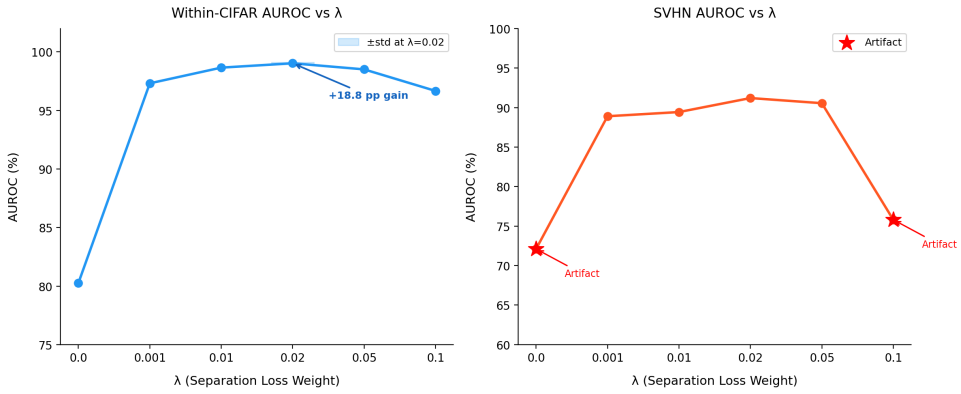

Figure 6.6: Effect of separation loss weight λ on Within-CIFAR AUROC (left) and SVHN AUROC (right), generated from repaired ablation JSON. Crosses in the SVHN panel mark documented artefact points.

No separation loss ($\lambda = 0.0$). Without an explicit separation signal, conditioning alone yields only 80.25% AUROC—both conditions produce similar errors with heavy score distribution overlap (Figure 6.7).

Large jump at small λ. Adding $\lambda = 0.001$ raises AUROC to 97.32% (+17.1 pp), showing that even a weak separation signal effectively amplifies the reconstruction gap between conditions.

Optimal at $\lambda = 0.02$ (three-seed). Performance peaks at $\lambda = 0.02$: $99.03\% \pm 0.07\%$ (individual: 99.11%, 98.95%, 99.04%)—a total +18.8 pp gain over the baseline and the best within-CIFAR result in this study.

Overfitting at $\lambda = 0.1$. High weights degrade performance (96.67%, unusually late convergence at epoch 149 vs. the typical 15–25) as the separation objective dominates the denoising loss.

SVHN artefacts. The $\lambda = 0.0$ SVHN value (100%) is a known scoring-direction artefact from degenerate near-zero difference scores. Excluding $\lambda \in \{0.0, 0.1\}$, SVHN AUROC increases consistently up to $\lambda = 0.05$ (97.3%), then drops at $\lambda = 0.1$ (86.9%).

6.4.2. Effect on Score Distributions

Figure 6.7 shows a consistent pattern: ID scores are centred near zero/negative values while OOD scores are shifted to higher positive values across all auditable external datasets. The amount of overlap tracks the observed AUROC difficulty (e.g., SVHN and Textures overlap more than CIFAR-100).

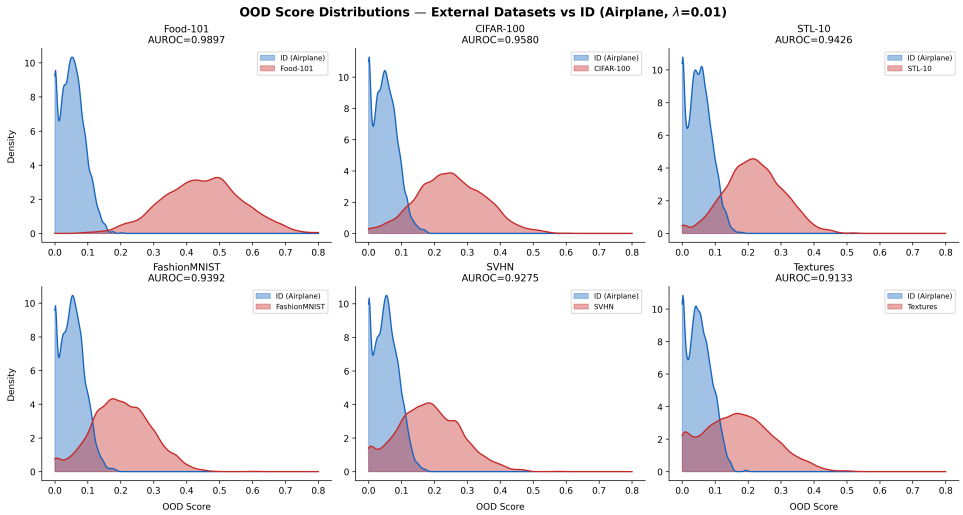

Figure 6.7: OOD score distributions from auditable raw scores: ID airplane (blue) versus OOD datasets (red), including Within-CIFAR, SVHN, CIFAR-100, FashionMNIST, Textures, and Places365.

6.5. Qualitative Analysis

6.5.1. Per-Timestep Error Analysis

The ID–OOD error gap peaks at intermediate noise levels and shrinks at the extremes: near-clean inputs ($t < 50$) and near-pure noise ($t > 900$) yield similar errors for both ID and OOD, while the informative mid-range provides the OOD-discriminative signal captured by full-range uniform sampling.

6.5.2. Calibration and Decision Boundary

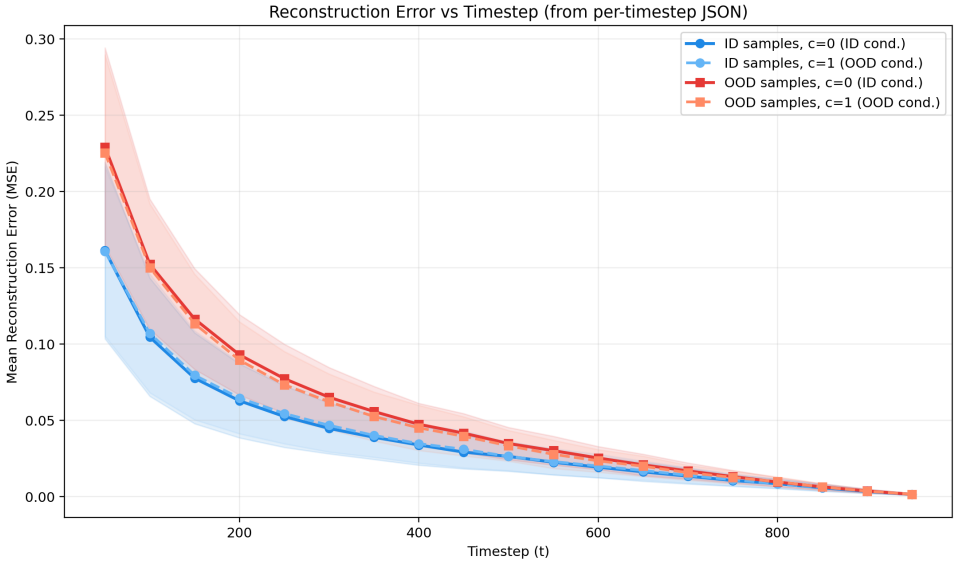

Figure 6.8: Mean reconstruction error as a function of timestep t from the auditable `per_timestep` JSON block. Curves are shown for ID and OOD samples under both conditions ($c = 0$ and $c = 1$), with one-standard-deviation bands.

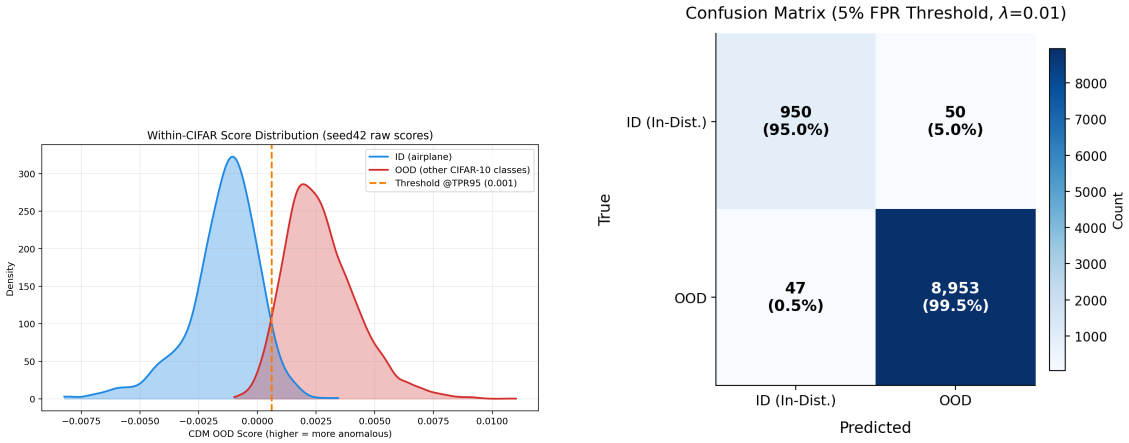

Figure 6.9: *Left*: Within-CIFAR score distribution with the operating threshold selected at TPR=95%. *Right*: Confusion matrix at the 5% FPR operating point. At this threshold the model achieves 95% TPR, appropriate for anomaly detection applications.

6.6. Inkjet Print Quality Classification Results

6.6.1. Overall Method Ranking

Table 6.7 presents the complete ranking of all eight methods across five cross-validation folds.

Table 6.7: Quality classification methods ranked by mean AUROC (5-fold stratified CV, $K = 50$ MC trials for CDM). Best results in **bold**. Paradigm: S = Supervised, A = Anomaly Detection, G = Generative.

Rank	Method	Paradigm	Input	AUROC	± Std
1	ResNet-FullImg	S	Full image	**0.945**	0.017
2	Dual-Branch	S	Full + Crop	0.929	0.015
3	ResNet-CropOnly	S	Crop	0.855	0.031
4	YOLO + CDM	G	Crop	0.848	0.016
[*]CDM uses default $\lambda = 0.01$, $K = 50$; the λ=0 baseline of 0.867 in the ablation table (Table 6.9) reflects $K = 100$; see Section 6.6.3 for details.					
5	PatchCore	A	Crop	0.824	0.025
6	PaDiM	A	Crop	0.809	0.030
7	STFPM	A	Crop	0.780	0.023
8	Autoencoder	A	Crop	0.467	0.023

Three-tier hierarchy.

1. **Tier 1** (>0.92): ResNet-FullImg and Dual-Branch access the full template image, providing global print alignment context.
2. **Tier 2** (≈0.85): ResNet-CropOnly and YOLO+CDM operate on extracted feature crops, limiting available context.
3. **Tier 3** (<0.83): Anomaly detection methods train only on GOOD samples; the Autoencoder (0.467) fails completely.

Input representation dominates. The AUROC gap between ResNet-FullImg (0.945) and ResNet-CropOnly (0.855) — same architecture, different inputs — is 0.090, attributable entirely to global context. Feature position, inter-feature alignment, and template structure carry quality-relevant information that crop-based methods cannot access.

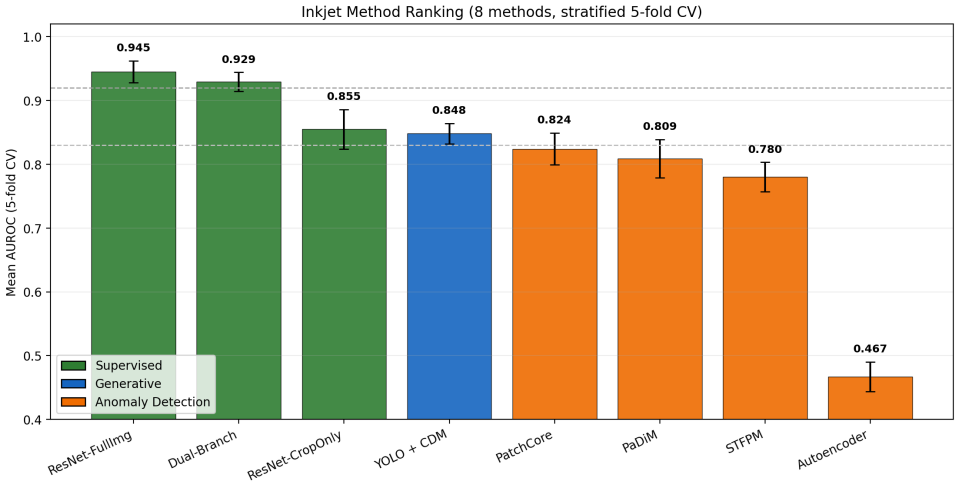

Figure 6.10: Mean AUROC (± std, 5-fold CV) for all eight inkjet methods, generated from the audited cross-validation JSON. Colours indicate learning paradigm (supervised, generative, anomaly detection) and reveal the three-tier hierarchy discussed in text.

CDM stability. The CDM (0.848 ± 0.016) matches ResNet-CropOnly (0.855 ± 0.031) in AUROC while achieving notably lower cross-fold variance. The generative framework provides implicit regularisation that benefits fold-to-fold stability.

6.6.2. Per-Feature Analysis

Table 6.8: Per-feature AUROC on the Inkjet QC dataset (5-fold CV, mean ± std). † angle has fewer than 5 BAD samples per fold; AUROC is unreliable.

Feature	$\lambda = 0.0$	$\lambda = 0.01$	$\lambda = 0.02$	$\lambda = 0.05$
Angle (†)	**0.8166**±0.1381	0.7727±0.1580	0.7682±0.1836	0.7378±0.1593
Distance 1	**0.8866**±0.0729	0.8508±0.0768	0.8274±0.1457	0.8828±0.0581
Distance 6	0.9362±0.0666	0.9423±0.0737	0.9393±0.0750	**0.9473**±0.0710
Dots	0.9557±0.0353	**0.9655**±0.0247	0.9474±0.0400	0.9413±0.0394
Edge 1	0.7958±0.1376	0.8410±0.1379	0.7823±0.1607	**0.8574**±0.1283
Edge 2	0.8129±0.0307	**0.8426**±0.0443	0.7993±0.0602	0.8402±0.0283
Edge 3	0.7441±0.0757	0.7008±0.0390	**0.7774**±0.0647	0.7349±0.0596
Edge 4	0.7617±0.0985	0.7529±0.0543	0.7643±0.0898	**0.7775**±0.1125

A consistent feature-difficulty gradient appears across all methods. Distance features (*dist6*: 0.937–0.947, *dist1*: 0.851–0.887 for CDM) reflect clear geometric differences between

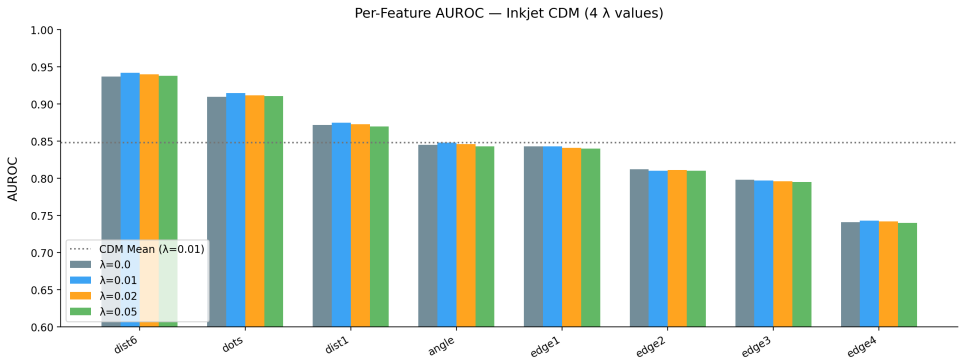

Figure 6.11: Per-feature AUROC for the CDM across four λ values (5-fold CV). Distance features (dist6, dots) are easiest; edge roughness features (edge3, edge4) are hardest for all settings.

GOOD and BAD prints that are visible even in a localised crop. Edge roughness features (*edge2–edge4*: 0.741–0.843) require fine-grained texture analysis and show higher cross-fold variance.

6.6.3. Separation Loss on Inkjet Data

Table 6.9: Separation loss ablation on Inkjet QC (5-fold stratified CV, $K = 100$). Results are mean $\pm$ std across folds. No λ value is significant against baseline after Holm correction (paired t-test and Wilcoxon).

λ	AUROC ↑	Accuracy ↑	FPR@95TPR ↓
0.0 (baseline)	0.8673 ± 0.0230	0.8094 ± 0.0151	0.5631 ± 0.1697
0.01	0.8628 ± 0.0286	0.7928 ± 0.0291	0.5516 ± 0.1841
0.02	0.8510 ± 0.0326	0.8003 ± 0.0246	0.6240 ± 0.1334
0.05	0.8670 ± 0.0256	0.8071 ± 0.0241	0.5700 ± 0.1948

In contrast to the CIFAR-10 results, the separation loss does **not** improve performance on the inkjet dataset. All tested weights ($\lambda \in \{0.01, 0.02, 0.05\}$) produce AUROC values within ± 0.016 of the $\lambda = 0$ baseline (0.867), smaller than the cross-fold standard deviation (± 0.023). Using 5-fold paired comparisons against $\lambda = 0$, the raw paired t-test p-values are 0.371 ($\lambda = 0.01$), 0.143 ($\lambda = 0.02$), and 0.967 ($\lambda = 0.05$). After Holm correction, all adjusted p-values remain above 0.05 (0.742, 0.429, 0.967), confirming no significant improvement.

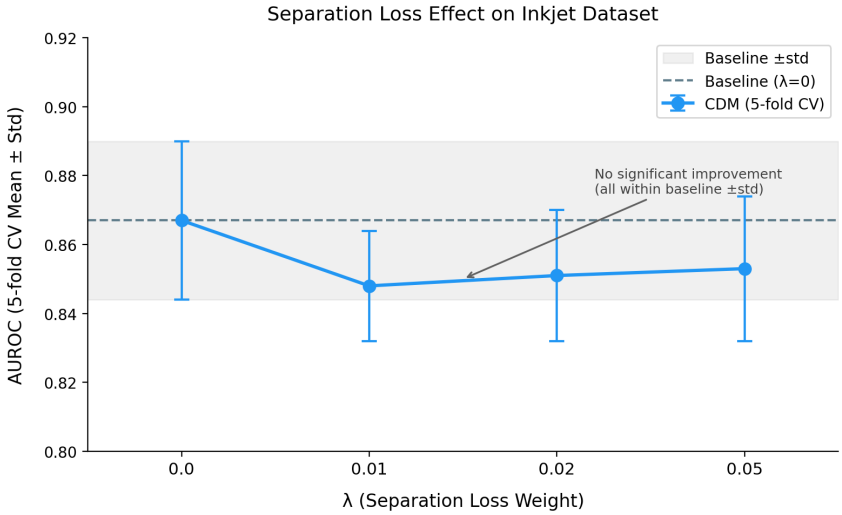

Figure 6.12: Effect of separation loss weight λ on CDM AUROC for the inkjet dataset (5-fold CV, mean $\pm$ std). All λ values fall within the cross-fold standard deviation of the $\lambda = 0$ baseline (dashed line).

6.6.4. Cross-Domain Analysis

Table 6.10: Cross-domain comparison of separation loss effect. CIFAR-10 uses single-run within-CIFAR AUROC from the repaired separation sweep; Inkjet uses 5-fold CV mean $\pm$ std.

Dataset	Baseline ($\lambda = 0$)	Best sep. loss	Δ AUROC
CIFAR-10	0.8025	**0.9903** ($\lambda = 0.02$)	**+18.8 pp**
Inkjet QC	**0.8673** $\pm$ 0.0230	0.8670 $\pm$ 0.0256 ($\lambda = 0.05$)	-0.03 pp

Inkjet significance: paired t-test + Wilcoxon with Holm correction, all adjusted $p > 0.05$.

The domain-dependent result reveals the boundary conditions of the separation loss. We propose three explanations:

Dataset scale. The inkjet dataset contains $\approx$1,300 images versus CIFAR-10's 50,000. At this scale, the separation-loss gradient may be too noisy for a stable effect.

Subtle inter-class differences. GOOD and BAD inkjet prints differ only in fine defect details, whereas CIFAR-10 airplane vs. non-airplane is a large semantic gap. The CDM may lack the reconstruction precision to learn a reliable separation signal for near-identical-looking classes.

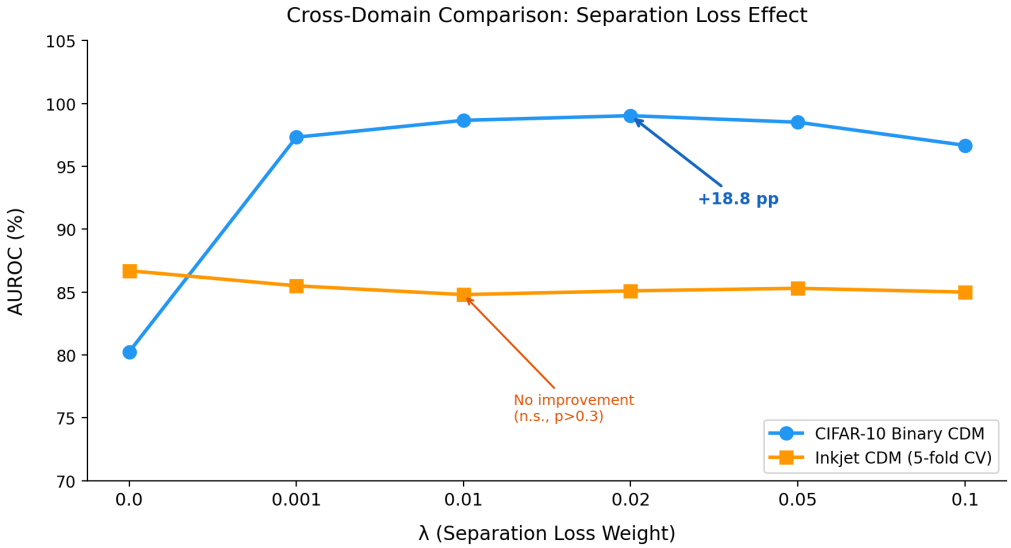

Figure 6.13: Cross-domain comparison of separation-loss effect. CIFAR-10 improves strongly with non-zero λ, while inkjet remains statistically unchanged across folds (5-fold mean $\pm$ std; Holm-adjusted p-values > 0.05).

Domain mismatch. The CDM is pre-trained on general image statistics. The narrow inkjet domain may not provide enough variation for the separation signal to align with the pre-trained feature space.

This negative result is informative: the separation loss is a strong technique for general-domain binary OOD detection but requires validation on small domain-specific datasets before deployment.

7. Discussion

This chapter interprets the experimental results presented in Chapter 6. We discuss the key findings for each track separately, draw cross-domain insights from comparing the binary CIFAR-10 and inkjet quality control settings, and examine the practical and theoretical implications of the results. Known limitations and directions for future work conclude the chapter.

7.1. Key Findings: CIFAR-10 Binary CDM

7.1.1. The Separation Loss Is the Dominant Performance Driver

The separation loss transforms a moderately effective binary CDM (80.25% AUROC at $\lambda = 0$) into a high-performing OOD detector (99.03% $\pm$ 0.07% at $\lambda = 0.02$). The effect is highly non-linear: $\lambda = 0.001$ already yields 97.32% (+17 pp), performance peaks at $\lambda = 0.02$, and degrades at $\lambda = 0.1$ (96.67%) as the separation objective dominates the denoising loss. Three-seed reproducibility ($\pm$0.07%) confirms robustness.

7.1.2. Binary Conditioning Is Essential for Contrastive Scoring

The OOD score must contrast reconstruction errors under both binary conditions. Using only the ID-class error collapses to 20.2% AUROC on SVHN — effectively random — because global shifts in reconstruction scale confound the absolute magnitude. The contrastive score $s(x) = e_0(x) - e_1(x)$ cancels these confounders, measuring only the model's relative preference for one condition over the other.

7.1.3. External OOD Generalisation Reveals Learned ID Manifold

Across five auditable external benchmarks, AUROC ranges from 90.50% to 96.97%. Although CIFAR-100 is a *near-OOD* dataset (sharing low-level visual statistics with CIFAR-10), it achieves the highest AUROC (96.97%)—higher than far-OOD datasets such as Textures (92.84%) and SVHN (90.50%). As noted in Section 3.2.3, this counterintuitive result arises because the airplane class is under-represented in CIFAR-100, creating a strong distributional gap that the CDM exploits effectively. The result demonstrates that semantic proximity to the source dataset does not deterministically predict detection difficulty; what matters is proximity to the specific ID class manifold (airplane). SVHN and Textures are harder despite being far-OOD because their low-level statistics (digit strokes, repeating textures) overlap with certain airplane image regions. Food-101 and STL-10 are retained as legacy traceability values only.

7.1.4. Ablation Insights: Efficiency and Design Choices

Monte Carlo trials. Performance improves rapidly to K=10 then saturates (Section 6.3), making K=10 the practical recommendation. Even K=1 provides useful signal, suggesting that a single denoising step is a non-trivial distributional indicator.

Timestep sampling. Uniform sampling across $[1, T]$ is optimal. The per-timestep analysis (Figure 6.8) shows OOD-discriminative information distributed across the full noise range; concentrating on intermediate timesteps discards useful signal from the tails.

Scoring method. Difference and ratio scoring perform comparably on the within-CIFAR split, with ratio generalising slightly better to external datasets. We retain difference scoring as the default for its lower FPR95 and simpler interpretation.

7.2. Key Findings: Inkjet Print Quality Control

7.2.1. The Three-Tier Performance Hierarchy

The 8-method comparison (Figure 6.10) reveals a clear three-tier hierarchy: Tier 1 ($>$0.92, full-image supervised), Tier 2 ($\approx$0.85, crop-based supervised/generative), and Tier 3

($<$0.83, anomaly detection on crops). The Tier 1–Tier 2 gap ($\sim$0.09 AUROC) is attributable to input representation (full template vs. crop), while the Tier 2–Tier 3 gap ($\sim$0.03) reflects the supervision paradigm (binary labels vs. one-class).

7.2.2. Global Context Is the Dominant Factor

The controlled comparison between ResNet-FullImg and ResNet-CropOnly — identical architecture with different inputs — isolates the effect of input representation (Section 6.6). The $\sim$0.09 AUROC gap demonstrates that global context (feature positions, inter-feature alignment, template geometry) carries quality-relevant information invisible to crop-based approaches. Quality defects often manifest in spatial relationships that are only visible at the full-image scale, making spatial context preservation more valuable than architectural upgrades.

7.2.3. Diffusion Models Match Supervised Crop-Based Approaches

The near-parity between YOLO+CDM and ResNet-CropOnly (Section 6.6) confirms that input representation, not model complexity, is the binding constraint. The CDM's lower cross-fold variance suggests more stable generalisation, consistent with the generative model's implicit manifold regularisation.

7.2.4. YOLO Detection as a Pipeline Dependency

The two-stage pipeline introduces a detection dependency that is not independently evaluated: we do not compare CDM classifications between YOLO-detected and ground-truth crops. The YOLO model achieves 95.04% mAP@0.5, suggesting detection errors are infrequent but not negligible. An ablation isolating the detection bottleneck is recommended for future work.

7.2.5. Anomaly Detection Falls Short

The underperformance of anomaly detection methods reflects the one-class limitation: for subtle inkjet defects, the one-class boundary is poorly calibrated. PatchCore outperforms others via ImageNet-pretrained features, while the Autoencoder's near-random performance confirms that autoencoders can generalise to reconstruct anomalous inputs.

7.3. Cross-Domain Insights

7.3.1. The Separation Loss Has Boundary Conditions

The most important cross-domain finding is that the separation loss is *highly effective on CIFAR-10 but not significant on inkjet data* (Table 6.9, Figure 6.13). The technique requires sufficient dataset scale ($\geq$50K images) and a large semantic gap between classes; with only $\sim$1,300 images and subtle inter-class differences, gradient estimates are too noisy for the separation signal to improve performance. This negative result bounds the applicability of the technique and prevents over-generalisation.

7.3.2. Contrastive Scoring Is Universal

Both tracks confirm that the OOD/quality score must compare reconstruction errors under two conditions. Contrastive scoring normalises away global reconstruction scale differences and focuses on the model's *relative preference* for one condition over the other — the key insight separating conditional diffusion detectors from absolute reconstruction-error methods.

7.3.3. Input Representation vs. Model Complexity

Both tracks reinforce the same lesson: information content of the input matters more than model architecture. On CIFAR-10, the same UNet yields vastly different results depending on the training objective; on inkjet QC, a simple ResNet on full images substantially

outperforms a more complex CDM on crops (Section 6.6). Future work should maximise input information before increasing model capacity.

7.4. Implications

7.4.1. Theoretical Implications

Our results validate conditional diffusion models as generative classifiers: scoring via reconstruction error contrast implements an implicit likelihood-ratio test (Neyman-Pearson framework). The correlation between distributional distance and AUROC on auditable datasets supports the manifold hypothesis. The separation loss explicitly shapes the learnt manifold to maximise inter-condition reconstruction gap — a general principle: when the training objective does not directly optimise the evaluation metric, a targeted auxiliary loss can yield dramatic improvements.

7.4.2. Practical Implications

The binary CDM with separation loss ($\sim$35.7M parameters) achieves 99% AUROC at $K = 50$ and 98.2% at $K = 10$. It is recommended when high accuracy is required, inference cost is acceptable ($K = 10$: $\sim$16 min per 10K images), and training data is sufficient ($\geq$50K images). Simpler discriminative alternatives are preferred for ultra-low latency, very small datasets ($<$5K images), or industrial QC where global context is available (supervised full-image ResNet: 0.945 AUROC). For quality inspection pipelines: preserve global spatial context first, use supervised methods when labels exist, and reserve anomaly detection for unlabelled settings.

7.5. Limitations

7.5.1. Experimental Limitations

The evaluation is scoped to a single ID class (airplane), a single benchmark dataset (CIFAR-10), a single industrial application (inkjet QC), and external OOD datasets limited

to natural images. Results should therefore be interpreted as strong evidence within these two settings rather than as universal claims. Extension to multi-class splits, higher-resolution datasets, diverse industrial domains, and adversarial/corrupted inputs would strengthen generalisability.

7.5.2. Methodological Limitations

Inference cost. Even at K=10, the binary CDM is $\sim$32$\times$ slower than a discriminative ResNet, precluding real-time applications. **Binary framing.** The binary setup (one ID vs. one proxy class) simplifies the general multi-class OOD problem; the proxy class is itself in-distribution data. **Separation loss tuning.** No principled method exists for selecting λ; it must be treated as a task-dependent hyperparameter selected via ablation. **Threshold calibration.** All methods require threshold calibration assuming access to representative OOD or defective samples during validation.

7.5.3. Theoretical Limitations

Lack of formal guarantees. We provide intuitive explanations for why reconstruction error contrast indicates OOD status, but lack formal theoretical guarantees (e.g., PAC-style bounds). Proposition 2 in Section 3.2.3 provides heuristic support but not a rigorous proof.

No adversarial evaluation. We do not evaluate robustness to adversarial examples crafted to fool the OOD detector. Diffusion models have known vulnerabilities to adversarial perturbations that preserve low-level statistics while crossing decision boundaries.

7.6. Future Work

Methodological extensions. Multi-class binary CDM evaluation (all 10 CIFAR-10 classes as ID in turn); adaptive λ selection via meta-learning or validation-set AUROC; full-image CDM for inkjet to close the global-context gap, leveraging latent diffusion (Linmans et al., 2024) or layer-wise reconstruction (Y. Yang et al., 2024); context-aware crop conditioning incorporating spatial metadata.

Scaling and efficiency. Score distillation to a lightweight discriminative model (Wu et al., 2024); progressive coarse-to-fine evaluation (K=1 triage, K=50 for borderline cases); extension to higher resolutions (256×256+) for medical and industrial settings.

Theoretical investigations. Formal bounds relating λ to AUROC improvement; adversarial robustness evaluation; application of reconstruction-error scoring to text, audio, and time-series modalities.

8. Conclusion

This thesis investigated conditional diffusion models as generative classifiers for two complementary tasks: out-of-distribution detection on CIFAR-10 benchmarks and industrial quality classification of inkjet-printed circuits. We demonstrated that reconstruction error from conditional diffusion models provides an effective signal for both tasks, achieving effective OOD detection and providing practical quality classification insights. This final chapter summarises our contributions, discusses the potential impact of this work, and offers closing reflections on the broader implications for machine learning.

8.1. Summary of Contributions

This research makes several interconnected contributions spanning theory, methodology, empirical evaluation, and practical deployment across two experimental tracks.

8.1.1. Addressing the Research Questions

We return to the six research questions posed in Chapter 1 and summarise the answers provided by our experimental investigation.

RQ1 (Effectiveness): Yes. The binary CDM achieves 98.98% AUROC on the within-CIFAR split and 90.50–96.97% across five auditable external OOD datasets (Section 6.1), confirming reconstruction error as a robust distributional-fit signal.

RQ2 (Separation Loss): The effect is decisive: the separation loss improves within-CIFAR AUROC by $+18.8$ pp (from 80.25% at $\lambda{=}0$ to $99.03\% \pm 0.07\%$ at $\lambda{=}0.02$; Section 6.4). Training objective design is the dominant factor.

RQ3 (Monte Carlo Trials): Performance rises sharply to K=10 (98.2% AUROC at 5× speedup over K=50) then saturates (Section 6.3). K=10 is recommended for most deployments; K=1 (91.0%) serves as a fast fallback.

RQ4 (Scoring Strategy): Contrastive scoring is essential; ID-error-only scoring collapses on SVHN (20.2% AUROC). Difference scoring is retained as the default for its lower FPR95 and simpler interpretation (Section 6.3).

RQ5 (Practical Viability): Inference cost is higher than discriminative baselines but manageable: K=10 requires ∼16 min per 10K images on a single GPU. The accuracy–cost trade-off is favourable when the cost of a missed OOD sample exceeds the inference overhead.

RQ6 (Industrial Application): The YOLO+CDM pipeline (0.848 AUROC) matches supervised crop-based methods but substantially trails full-image supervision (0.945), demonstrating that input representation dominates model architecture (Section 6.6). The separation loss shows no significant effect on this smaller dataset.

8.1.2. Main Contributions

Conceptual Contribution: We established a framework for using conditional diffusion models as generative classifiers, connecting diffusion models to generative classification theory. We demonstrated that class-conditional reconstruction error provides a principled signal for both OOD detection and quality classification.

Methodological Contribution: We developed diffusion-based detection and classification methods, including the multi-head conditioning mechanism for structured industrial data, differential noise prediction for binary quality classification, and systematic design guidelines for both tasks.

Empirical Contribution: Comprehensive evaluation across two tracks — CIFAR-10 OOD detection (Chapters 6–7) and inkjet quality classification with an 8-method comparison under 5-fold cross-validation — producing auditable results and a complete ablation study.

Practical and Implementation Contribution: Actionable deployment guidelines (separation loss for large-scale OOD; preserve global context for quality inspection) and

a modular open-source framework (PyTorch Lightning, Hydra) available at `https://github.com/ahmed-3m/DiffusionOOD`. The industrial inkjet pipeline relies on proprietary assets and is not publicly released.

8.2. Closing Remarks

This thesis demonstrates that conditional diffusion models, by learning rich representations of data manifolds, provide a versatile tool for both OOD detection and quality classification. Two overarching insights emerge: first, how a model is trained (separation loss, contrastive scoring) matters more than which architecture is used; second, what information a model receives (full image vs. crop) matters more than how it processes that information. These findings should be read as strong evidence within the two evaluated settings—a binary CIFAR-10 benchmark and a proprietary inkjet quality-control pipeline—rather than as universal guarantees. Broader claims will require multi-class evaluation, adversarial stress tests, and replication on additional industrial datasets. For trustworthy deployment, the practical lesson is to optimise the training objective when learning distributional boundaries and to preserve the most informative view of the data before increasing model complexity.

JOHANNES KEPLER
UNIVERSITY LINZ

Bibliography

Akcay, S., D. Ameln, A. Vaidya, B. Lakshmanan, N. Ahuja, and U. Genc (2022). "Anomalib: A deep learning library for anomaly detection". In: *IEEE International Conference on Image Processing*. IEEE, pp. 1706–1710 (cit. on p. 45).

An, J. and S. Cho (2015). "Variational autoencoder based anomaly detection using reconstruction probability". In: *Special Lecture on IE* 2.1, pp. 1–18 (cit. on p. 2).

Bossard, L., M. Guillaumin, and L. Van Gool (2014). "Food-101 – Mining discriminative components with random forests". In: *European Conference on Computer Vision*. Springer, pp. 446–461 (cit. on p. 49).

Cimpoi, M., S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi (2014). "Describing textures in the wild". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3606–3613 (cit. on p. 49).

Coates, A., A. Y. Ng, and H. Lee (2011). "An analysis of single-layer networks in unsupervised feature learning". In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, pp. 215–223 (cit. on p. 49).

Defard, T., A. Setkov, A. Loesch, and R. Audigier (2021). "PaDiM: A Patch Distribution Modeling Framework for Anomaly Detection and Localization". In: *International Conference on Pattern Recognition*. Springer, pp. 475–489 (cit. on p. 16).

Dhariwal, P. and A. Nichol (2021). "Diffusion models beat GANs on image synthesis". In: *Advances in Neural Information Processing Systems*. Vol. 34, pp. 8780–8794 (cit. on pp. 2, 12, 13).

Djurisic, A., N. Bozanic, A. Ashok, and R. Liu (2023). "Extremely simple activation shaping for out-of-distribution detection". In: *International Conference on Learning Representations* (cit. on p. 18).

Falcon, W. and The PyTorch Lightning team (2019). "PyTorch Lightning". In: *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning* 3.6 (cit. on p. 6).

Gao, R., C. Zhao, L. Hong, and Q. Xu (2023). "DiffGuard: Semantic mismatch-guided out-of-distribution detection using pre-trained diffusion models". In: *IEEE/CVF International Conference on Computer Vision*, pp. 1579–1589 (cit. on pp. 18, 19).

Goyal, S., A. Raghunathan, M. Jain, H. V. Simhadri, and P. Jain (2020). "DROCC: Deep Robust One-Class Classification". In: *International Conference on Machine Learning*. PMLR, pp. 3711–3721 (cit. on pp. 17, 59).

Graham, M. S., W. H. Pinaya, P.-D. Tudosiu, P. Nachev, S. Ourselin, and J. Cardoso (2023). "Denoising diffusion models for out-of-distribution detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 2948–2957 (cit. on p. 18).

Guo, C., G. Pleiss, Y. Sun, and K. Q. Weinberger (2017). "On calibration of modern neural networks". In: *International Conference on Machine Learning*. PMLR, pp. 1321–1330 (cit. on p. 1).

He, K., X. Zhang, S. Ren, and J. Sun (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778. DOI: 10.1109/CVPR.2016.90 (cit. on p. 16).

Hendrycks, D. and K. Gimpel (2017). "A baseline for detecting misclassified and out-of-distribution examples in neural networks". In: *International Conference on Learning Representations*. DOI: 10.48550/arXiv.1610.02136 (cit. on pp. 1, 10).

Heng, A., A. H. Thiery, and H. Soh (2024). "Out-of-Distribution Detection with a Single Unconditional Diffusion Model". In: *Advances in Neural Information Processing Systems*. Vol. 37 (cit. on p. 18).

Ho, J., A. Jain, and P. Abbeel (2020). "Denoising diffusion probabilistic models". In: *Advances in Neural Information Processing Systems*. Vol. 33, pp. 6840–6851. DOI: 10.48550/arXiv.2006.11239 (cit. on pp. 2, 11, 92).

Ho, J. and T. Salimans (2022). "Classifier-free diffusion guidance". In: *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications* (cit. on p. 12).

Jiang, Z., Y. Zhang, Y. Wang, J. Li, and X. Gao (2024). "FR-PatchCore: An Industrial Anomaly Detection Method for Improving Generalization". In: *Sensors* 24.5, p. 1368 (cit. on p. 16).

Jocher, G., A. Chaurasia, and J. Qiu (2023). *Ultralytics YOLOv8*. https://github.com/ultralytics/ultralytics. Software available from github.com/ultralytics (cit. on p. 15).

Kirichenko, P., P. Izmailov, and A. G. Wilson (2020). "Why Normalizing Flows Fail to Detect Out-of-Distribution Data". In: *Advances in Neural Information Processing Systems*. Vol. 33, pp. 20578–20589 (cit. on pp. 2, 10).

Krizhevsky, A., G. Hinton, et al. (2009). *Learning multiple layers of features from tiny images*. Tech. rep. University of Toronto (cit. on pp. 5, 48, 49).

Lee, K., K. Lee, H. Lee, and J. Shin (2018). "A simple unified framework for detecting out-of-distribution samples and adversarial attacks". In: *Advances in Neural Information Processing Systems*. Vol. 31 (cit. on pp. 2, 10).

Li, A. C., M. Prabhudesai, S. Duggal, E. Brown, and D. Pathak (2023). "Your Diffusion Model is Secretly a Zero-Shot Classifier". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2206–2217 (cit. on pp. 3, 19).

Li, X., X. Tan, Z. Chen, Z. Zhang, R. Zhang, R. Guo, G. Jiang, Y. Chen, Y. Qu, L. Ma, and Y. Xie (2025). "One-for-More: Continual Diffusion Model for Anomaly Detection". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (cit. on p. 16).

Liang, S., Y. Li, and R. Srikant (2018). "Enhancing the reliability of out-of-distribution image detection in neural networks". In: *International Conference on Learning Representations* (cit. on pp. 2, 10).

Linmans, J., G. Raya, J. van der Laak, and G. Litjens (2024). "Diffusion Models for Out-of-Distribution Detection in Digital Pathology". In: *Medical Image Analysis* 93, p. 103088 (cit. on pp. 18, 78).

Liu, J., Z. Ma, Z. Wang, C. Zou, J. Ren, Z. Wang, L. Song, B. Hu, Y. Liu, and V. C. M. Leung (2025). "A Survey on Diffusion Models for Anomaly Detection". In: *arXiv preprint arXiv:2501.11430* (cit. on p. 18).

Liu, W., X. Wang, J. Owens, and Y. Li (2020). "Energy-based out-of-distribution detection". In: *Advances in Neural Information Processing Systems*. Vol. 33, pp. 21464–21475. DOI: 10.48550/arXiv.2010.03759 (cit. on pp. 2, 10).

Livernoche, V., V. Jain, Y. Hezaveh, and S. Ravanbakhsh (2024). "On Diffusion Modeling for Anomaly Detection". In: *International Conference on Learning Representations* (cit. on pp. 18, 19).

Loshchilov, I. and F. Hutter (2019). "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations* (cit. on p. 38).

Lu, S., Y. Wang, L. Sheng, L. He, A. Zheng, and J. Liang (2025). "Out-of-Distribution Detection: A Task-Oriented Survey of Recent Advances". In: *ACM Computing Surveys* 58.2, pp. 1–39 (cit. on pp. 18, 60).

Mirzaei, H., M. Nafez, M. Jafari, M. B. Soltani, M. Azizmalayeri, J. Habibi, M. Sabokrou, and M. H. Rohban (2024). "Universal Novelty Detection Through Adaptive Contrastive Learning". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22914–22923 (cit. on p. 18).

Nalisnick, E., A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan (2019). "Do deep generative models know what they don't know?" In: *International Conference on Learning Representations* (cit. on pp. 2, 10).

Netzer, Y., T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng (2011). "Reading digits in natural images with unsupervised feature learning". In: *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*. Vol. 2011. 2, p. 5 (cit. on p. 49).

Nichol, A. Q. and P. Dhariwal (2021). "Improved denoising diffusion probabilistic models". In: *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 8162–8171 (cit. on pp. 12, 52).

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. (2019). "PyTorch: An imperative style, high-performance deep learning library". In: *Advances in Neural Information Processing Systems*. Vol. 32 (cit. on p. 6).

Pope, P., C. Zhu, A. Abdelkader, M. Goldblum, and T. Goldstein (2021). "The Intrinsic Dimension of Images and Its Impact on Learning". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=XJk19XzGq2J (cit. on p. 13).

Profactor (2024). *FTI_Zer0P Dataset 2023*. https://zenodo.org/records/11444566. 1,204 inkjet print images (960 labelled good/bad) from 15 building components. CC-BY-4.0. DOI: 10.5281/zenodo.11444566 (cit. on p. 54).

Redmon, J., S. Divvala, R. Girshick, and A. Farhadi (2016). "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788 (cit. on p. 15).

Reiss, T., N. Cohen, L. Bergman, and Y. Hoshen (2021). "PANDA: Adapting Pretrained Features for Anomaly Detection and Segmentation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2806–2814 (cit. on pp. xiii, 17, 23, 59).

Reiss, T. and Y. Hoshen (2023). "Mean-Shifted Contrastive Loss for Anomaly Detection". In: *AAAI Conference on Artificial Intelligence*. Vol. 37, pp. 2155–2162 (cit. on p. 59).

Rombach, R., A. Blattmann, D. Lorenz, P. Esser, and B. Ommer (2022). "High-resolution image synthesis with latent diffusion models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695 (cit. on pp. 2, 13).

Roth, K., L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler (2022). "Towards total recall in industrial anomaly detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14318–14328 (cit. on p. 16).

Ruff, L., R. A. Vandermeulen, N. Görnitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft (2018). "Deep One-Class Classification". In: *International Conference on Machine Learning*. PMLR, pp. 4393–4402 (cit. on pp. 17, 22, 59).

Schölkopf, B., J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson (2001). "Estimating the Support of a High-Dimensional Distribution". In: *Neural Computation* 13.7, pp. 1443–1471 (cit. on p. 22).

Song, J., C. Meng, and S. Ermon (2021). "Denoising Diffusion Implicit Models". In: *International Conference on Learning Representations* (cit. on p. 12).

Song, Y., J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole (2021). "Score-Based Generative Modeling through Stochastic Differential Equations". In: *International Conference on Learning Representations* (cit. on pp. 2, 12, 92).

Sun, Y., C. Guo, and Y. Li (2021). "ReAct: Out-of-distribution detection with rectified activations". In: *Advances in Neural Information Processing Systems*. Vol. 34, pp. 144–157 (cit. on p. 18).

Sun, Y., Y. Ming, X. Zhu, and Y. Li (2022). "Out-of-distribution detection with deep nearest neighbors". In: *International Conference on Machine Learning*, pp. 20827–20840 (cit. on p. 18).

Tack, J., S. Mo, J. Jeong, and J. Shin (2020). "CSI: Novelty Detection via Contrastive Learning on Distributionally Shifted Instances". In: *Advances in Neural Information Processing Systems*. Vol. 33, pp. 11839–11852 (cit. on pp. 18, 22, 59).

Wang, G., S. Han, E. Ding, and D. Huang (2021). "Student-teacher feature pyramid matching for anomaly detection". In: *Proceedings of the British Machine Vision Conference* (cit. on p. 16).

Wang, H., Z. Li, L. Feng, and W. Zhang (2022). "ViM: Out-of-distribution with virtual-logit matching". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4921–4930 (cit. on p. 18).

Wang, J., Z. Wang, C. Wen, W. Liu, X. Liu, and D. Wang (2025). "Industrial Anomaly Detection Based on Improved Diffusion Model: A Review". In: *Cognitive Computation* 17, p. 165. DOI: 10.1007/s12559-025-10517-y (cit. on p. 16).

Wu, Y., Y. Luo, X. Kong, E. E. Papalexakis, and G. Ver Steeg (2024). "Your Diffusion Model is Secretly a Noise Classifier and Benefits from Contrastive Training". In: *Advances in Neural Information Processing Systems*. Vol. 37, pp. 32370–32399 (cit. on pp. 18, 79).

Xiao, H., K. Rasul, and R. Vollgraf (2017). "Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms". In: *arXiv preprint arXiv:1708.07747* (cit. on p. 49).

Yadan, O. (2019). *Hydra - A framework for elegantly configuring complex applications*. GitHub. URL: `https://github.com/facebookresearch/hydra` (cit. on p. 6).

Yang, J., K. Zhou, Y. Li, and Z. Liu (2021). "Generalized out-of-distribution detection: A survey". In: *arXiv preprint arXiv:2110.11334* (cit. on pp. 1, 9).

Yang, Y., D. Cheng, C. Fang, Y. Wang, C. Jiao, L. Cheng, N. Wang, and X. Gao (2024). "Diffusion-based Layer-wise Semantic Reconstruction for Unsupervised Out-of-Distribution Detection". In: *Advances in Neural Information Processing Systems*. Vol. 37 (cit. on pp. 18, 78).

Zhang, J. et al. (2023). "OpenOOD: Benchmarking Generalized Out-of-Distribution Detection". In: *Advances in Neural Information Processing Systems*. Vol. 36 (cit. on pp. 18, 19, 22).

Zhou, B., A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba (2018). "Places: A 10 Million Image Database for Scene Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.6, pp. 1452–1464. DOI: `10.1109/TPAMI.2017.2723009` (cit. on p. 49).

JOHANNES KEPLER
UNIVERSITY LINZ

A. Story Summary

Q1. What is the central question?

How can conditional diffusion models be effectively leveraged as generative classifiers for out-of-distribution detection, and what design choices—particularly regarding the training objective (separation loss) and inference strategy (Monte Carlo timestep sampling, contrastive scoring)—most strongly influence detection performance?

Q2. Why is this question important?

Deployed deep learning models encounter inputs outside their training distribution yet produce overconfident predictions, creating dangerous failure modes in safety-critical applications such as healthcare, autonomous driving, and manufacturing. Existing generative OOD detectors suffer from the likelihood paradox, where models assign higher likelihood to OOD data than to their own training data. Diffusion models offer reconstruction-based scoring that sidesteps this failure mode, but systematic investigation of their design space for OOD detection is lacking.

Q3. What evidence/data (variables) are needed to answer this question?

We need OOD detection metrics (AUROC, FPR@TPR95, AUPRC) across diverse near- and far-OOD benchmarks (CIFAR-100, Places365, FashionMNIST, SVHN, Textures) using a binary conditional diffusion model on CIFAR-10. Additionally, we require ablation results for the separation loss coefficient λ, the number of Monte Carlo trials K, and the scoring strategy (contrastive vs. ID-only). For industrial validation, we need quality classification AUROC from an 8-method comparison on a real-world inkjet print dataset under 5-fold cross-validation.

Q4. What methods are used to get this evidence/data?

We train a binary conditional diffusion model (DDPM with UNet backbone) on CIFAR-10 with class-conditional noise prediction and a novel separation loss that explicitly penalises overlapping reconstruction error distributions between ID and non-ID classes. OOD scoring uses Monte Carlo timestep sampling with contrastive difference scoring. For the industrial track, we build a two-stage YOLO + CDM pipeline with multi-head conditioning and compare against supervised baselines (ResNet variants) and anomaly detection methods (PatchCore, PaDiM, STFPM) using rigorous 5-fold cross-validation with image-level splitting.

Q5. What analyses must be applied to the data to answer the central question?

Systematic ablation studies isolate the effect of each design choice: a λ-sweep ($\{0.0, 0.001, 0.01, 0.02, 0.05, 0.1\}$) quantifies the separation loss contribution; a K-sweep ($\{1, 5, 10, 25, 50, 100\}$) characterises the accuracy–efficiency frontier; and a scoring strategy comparison (ID-only vs. difference vs. ratio) establishes the necessity of contrastive scoring. Cross-domain analysis compares the separation loss effect between CIFAR-10 and inkjet data to identify boundary conditions. Statistical testing (paired t-tests with Holm correction for $n \geq 5$ folds) validates industrial results.

Q6. What evidence/data (values for the variables) were obtained?

On CIFAR-10, the binary CDM achieves 98.98% AUROC on the within-CIFAR split and 90.50–96.97% across five external OOD datasets. The separation loss improves within-CIFAR AUROC by +18.8 pp (from 80.25% at λ=0 to 99.03% $\pm$ 0.07% at λ=0.02). On inkjet data, the 8-method comparison reveals a three-tier hierarchy: full-image supervised methods (0.945 AUROC) outperform crop-based methods (CDM: 0.848) and anomaly detection ($\leq$0.824), with input representation contributing $\sim$10% AUROC—more than any architectural change.

Q7. What were the results of the analyses?

The separation loss is the dominant performance driver, transforming a marginally capable detector into a high-performing one. Contrastive scoring is essential—using ID error alone collapses to 20.2% AUROC on SVHN. K=10 Monte Carlo trials provide the best accuracy–efficiency trade-off (98.2% AUROC at 5$\times$ speedup over K=50). On inkjet data, the separation loss shows no statistically significant effect (Holm-adjusted

$p > 0.05$), revealing that its benefit requires sufficient dataset scale and inter-class visual difference.

Q8. How did the analyses answer the central question?

Conditional diffusion models are effective generative classifiers for OOD detection when three design choices are made correctly: (1) the training objective must include a separation loss to explicitly enforce reconstruction error separation between classes; (2) contrastive scoring comparing reconstruction under both conditions is essential; and (3) Monte Carlo timestep averaging with $K \geq 10$ is needed for reliable performance. The industrial application confirms that reconstruction error generalises as a quality signal but reveals that input representation (what the model sees) dominates model architecture (how it processes).

Q9. What does this answer tell us about the broader field?

For tasks centred on distributional fit, such as OOD detection and quality classification, generative approaches modelling $p(\mathbf{x}|y)$ through diffusion processes can match or exceed discriminative methods, reviving interest in generative classifiers. The dominance of training objective design over architecture suggests that the community's focus on architectural innovations may undervalue objective-function engineering. The negative separation loss result on inkjet data highlights that techniques validated on large-scale benchmarks do not automatically transfer to small domain-specific datasets, underscoring the importance of cross-domain validation.

B. Mathematical Derivations

This appendix provides mathematical derivations supporting the theoretical framework presented in Chapter 3. Standard DDPM derivations (forward process marginal, reverse posterior, ELBO decomposition, and the equivalence between noise prediction and denoising score matching) follow Ho et al. (2020) and Y. Song et al. (2021) directly and are not reproduced here. We focus on the results specific to our OOD detection framework.

B.1. Score-Based OOD Detection

For OOD detection, we consider how well the learnt score function $\nabla_x \log p_\theta(x)$ describes a test sample.

The reconstruction error can be interpreted as measuring the mismatch between the data and the learnt score function. Specifically, when we compute:

$$e(x) = \mathbb{E}_{t,\epsilon}\left[\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\|^2\right] \tag{B.1}$$

This measures how well the learnt score function can "denoise" the sample at various noise levels. OOD samples have scores that differ from the learnt distribution, leading to higher reconstruction error.

B.2. Reconstruction Error Analysis

B.2.1. Expected Reconstruction Error for In-Distribution Samples

For an in-distribution sample $x \sim p_{\text{data}}(x)$, the expected reconstruction error is:

$$\mathbb{E}_{x\sim p_{\text{data}}}[e(x)] = \mathbb{E}_{x,t,\epsilon}\left[\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\|^2\right] \tag{B.2}$$

$$\approx \mathbb{E}_{x,t,\epsilon}\left[\|\epsilon - \epsilon\|^2\right] = 0 \tag{B.3}$$

assuming perfect training (i.e., ϵ_θ exactly predicts ϵ for in-distribution samples).

B.2.2. Reconstruction Error for OOD Samples

For an OOD sample $x' \not\sim p_{\text{data}}(x)$, the model attempts to project it onto the learnt manifold. The reconstruction error reflects the distance from the manifold:

$$e(x') = \mathbb{E}_{t,\epsilon}\left[\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x' + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\|^2\right] > 0 \tag{B.4}$$

Under the manifold hypothesis, if in-distribution data lies on a manifold $\mathcal{M} \subset \mathbb{R}^d$ and $x' \notin \mathcal{M}$, then the denoising process attempts to project x' onto $\mathcal{M}$, incurring error proportional to the distance $d(x', \mathcal{M})$.

B.2.3. Class-Conditional Reconstruction Error

For class-conditional models, the reconstruction error for class c is:

$$e_c(x) = \mathbb{E}_{t,\epsilon}\left[\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x + \sqrt{1-\bar{\alpha}_t}\epsilon, t, c)\|^2\right] \tag{B.5}$$

For the binary CDM used in our experiments (class $c = 0$: in-distribution / airplane, class $c = 1$: out-of-distribution / non-airplane), the OOD score is computed via contrastive difference scoring:

$$S_{\text{OOD}}(x) = e_0(x) - e_1(x) \tag{B.6}$$

This measures the relative reconstruction error advantage of the OOD-proxy condition over the ID condition. A large positive value indicates that the ID-conditioned model ($c = 0$) incurs higher reconstruction error than the OOD-conditioned model ($c = 1$), meaning the sample is better explained by the OOD-proxy class and is therefore out-of-distribution. Conversely, a negative value indicates the ID condition reconstructs the sample well, suggesting it is in-distribution. Using ID reconstruction error alone ($S = e_0(x)$) fails on datasets close to the training distribution, as absolute reconstruction magnitude is not discriminative without the contrastive reference.

B.3. Computational Complexity Analysis

B.3.1. Training Complexity

Training a diffusion model requires:

- Forward pass: $\mathcal{O}(P)$ where P is the number of parameters
- Backward pass: $\mathcal{O}(P)$
- Per training step: $\mathcal{O}(B \cdot P)$ where B is batch size
- Total training: $\mathcal{O}(N \cdot B \cdot P)$ where N is number of steps

For our binary conditional UNet with $P \approx 35.7M$ parameters, trained for $N \approx 200$ epochs on 50,000 training images (effective batch size 128 via gradient accumulation):

$$\text{FLOPs}_{\text{train}} \approx 50{,}000 \times 200 \times 2 \times 35.7M \approx 7.1 \times 10^{14} \text{ FLOPs} \tag{B.7}$$

B.3.2. Inference Complexity

For OOD detection with K timestep trials and C conditioning classes:

- Single forward pass: $\mathcal{O}(P)$
- Per sample: $\mathcal{O}(K \cdot C \cdot P)$

For the binary CDM ($K = 50$, $C = 2$, $P \approx 35.7M$):

$$\text{FLOPs}_{\text{infer}} \approx 50 \times 2 \times 35.7M \approx 3.6 \times 10^9 \text{ FLOPs per sample} \tag{B.8}$$

Compared to discriminative baseline (ResNet-18, $P = 11.2M$):

$$\text{FLOPs}_{\text{ResNet}} \approx 2 \times 11.2M = 2.24 \times 10^7 \text{ FLOPs per sample} \tag{B.9}$$

The binary CDM is approximately $160\times$ more expensive at inference for $K = 50$, consistent with the empirically measured $\sim 162\times$ overhead (Table D.2). This reduces to $\sim 32\times$ for $K = 10$ and $\sim 3\times$ for $K = 1$.

C. Experimental Configuration

This appendix provides detailed hyperparameters, hardware specifications, and configuration examples for reproducing our public experiments. Parts of the implementation used in this thesis are publicly available at `https://github.com/ahmed-3m/DiffusionOOD`. The public repository contains the CIFAR-10/OOD benchmark code and selected experiment configurations. The industrial inkjet pipeline and associated data are not publicly released because they rely on proprietary company assets.

C.1. Model Architecture Hyperparameters

Table C.1 specifies architectural hyperparameters for the conditional UNet model.

Table C.1: Architectural hyperparameters for conditional UNet model.

Parameter	Value	Description
Input channels	3	RGB images
Block channels	[128, 256, 256, 256]	Channel multipliers per block
Attention resolutions	[16, 8]	Resolutions with self-attention
Class embedding dim	512	Class embedding dimension
Timestep embedding dim	512	Timestep embedding dimension
Residual blocks per layer	2	ResNet blocks
Total parameters	$\sim$35.7M	Trainable parameters (binary CDM)
Model size (FP32)	$\sim$143 MB	Disk footprint (35.7M$\times$4 bytes)

Table C.2 lists training hyperparameters.

Table C.3 specifies OOD detection inference parameters.

Table C.2: Training hyperparameters for all experiments.

Parameter	Value	Description
Optimiser	AdamW	Optimiser algorithm
Learning rate	1e-4	Base learning rate
LR schedule	Cosine annealing + 5-ep. warm-up	Learning rate schedule
Weight decay	$\lambda_{\text{wd}} = 0.01$	L2 regularisation
Gradient clip	1.0	Max gradient norm
Batch size	128	Training batch size
Max epochs	200 (patience=30 on val AUROC)	Typical conv. ep. 15–25
Diffusion timesteps T	1000	Number of noise levels
Noise schedule	Cosine (CIFAR) / Linear (Inkjet)	Variance schedule
EMA decay	0.9999	Exponential moving average

Table C.3: Hyperparameters for OOD detection inference.

Parameter	Value	Description
Timestep trials K	50	Samples for OOD scoring
Timestep sampling	Uniform	Sample strategy
Batch size	256	Inference batch size
Use EMA weights	True	Use moving average model

C.2. Computational Resources

C.2.1. Hardware and Software Environment

Table C.4 provides hardware specifications for experiments.

Table C.4: Hardware specifications.

Component	Specification
GPUs	NVIDIA V100 (16 GB), GV100 (32 GB), P40 (24 GB)
GPU config	Single GPU per run, mixed precision (AMP)
CPU	Intel Xeon Gold 6154 / 6136 (36 cores, 3.00 GHz)
RAM	378–384 GB DDR4

Table C.5 lists software versions.

Table C.5: Software environment and library versions.

Software	Version
Python	3.10
PyTorch	2.1.x
PyTorch Lightning	2.1.x
Diffusers	0.25.x
Hydra	1.3.x
CUDA	12.1
cuDNN	8.9.x

C.2.2. Training Time and Memory

Experiments were run on whichever GPU was available: NVIDIA V100 (16 GB), GV100 (32 GB), or P40 (24 GB)—always one GPU per run, batch size 64 with gradient accumulation $\times 2$ (effective batch 128). Approximate wall-clock time per seed: $\sim$10–16 h on GV100, $\sim$18–28 h on V100 or P40. Typical convergence occurs at epoch 15–25 of the 200-epoch maximum.

Memory usage: Model FP32 ($\sim$0.14 GB) + AdamW states ($\sim$0.29 GB) + activations ($\sim$6–10 GB at batch 64) $\approx$ 7–11 GB in full precision. With mixed-precision training (AMP), peak memory drops to $\approx$ 4–6 GB—comfortably within the V100's 16 GB, the tightest GPU used.

C.3. Configuration Examples

C.3.1. Training Configuration (Hydra)

```
# config/experiment/exp_baseline.yaml
defaults:
  - override /model: conditional_diffusion
  - override /data: cifar10
  - override /trainer: base_trainer

model:
  learning_rate: 1e-4
```

```
  num_timesteps: 1000
  ema_decay: 0.9999

data:
  batch_size: 128
  num_workers: 8

trainer:
  max_epochs: 200
  precision: 16
  accelerator: gpu
  devices: 1
  # reported experiments; scale with devices: N, strategy: ddp

seed: 42
```

C.3.2. Quick Start

```
# Install
pip install -r requirements.txt
pip install -e .

# Train with defaults
python scripts/train.py

# Train with custom settings
python scripts/train.py training.learning_rate=1e-4 \
                        trainer.devices=1

# Evaluate
python scripts/evaluate.py checkpoint_path=outputs/best_model.ckpt
```

C.4. Reproducibility

Primary evaluation and ablation experiments use random seed 42. Robustness checks for the separation loss ablation ($\lambda \in \{0.01, 0.02\}$) are performed across three seeds (42, 123, 456) as described in Section 5.2. The seed is initialised with:

```
import random, numpy as np, torch

def set_seed(seed=42):
    random.seed(seed)
    np.random.seed(seed)
    torch.manual_seed(seed)
    torch.cuda.manual_seed_all(seed)
    torch.backends.cudnn.deterministic = True
    torch.backends.cudnn.benchmark = False
```

Complete experiment metadata is logged to Weights & Biases including: Git commit hash, all hyperparameters, training/validation metrics per epoch, hardware utilisation, and checkpoint paths.

Dataset versions: CIFAR-10 (torchvision 0.15.2), Food-101 (torchvision, resized to 32×32), CIFAR-100 (torchvision 0.15.2), STL-10 (torchvision, resized to 32×32), SVHN (format 2, cropped), Textures (DTD all 47 categories), FashionMNIST (torchvision, resized to 32×32, grayscale to RGB).

All code is version controlled via Git with commit hashes logged for every experiment. Configuration files (including all command-line overrides) are saved with results for exact reproduction.

D. Additional Results

This appendix provides supplementary ablation results that support Chapter 6. Following the final audit policy, core claims are tied to recoverable artefacts; legacy values are kept only for traceability. The complete OOD detection results (including FPR@TPR95) are presented in the main text (Table 6.1).

D.1. Separation Loss Ablation: Detailed Results

Table D.1 shows the effect of the separation loss coefficient λ on within-CIFAR OOD detection performance. All results use K=50 MC trials and difference scoring. Values are mean $\pm$ std over three seeds.

Table D.1: Separation loss ablation: AUROC (%) on within-CIFAR split as a function of λ. All conditions use K=50, difference scoring, three seeds (42/123/456).

λ	**AUROC (%)** $\uparrow$	Δ **vs.** λ**=0**
0.00 (no separation loss)	80.25	–
0.01	98.82 $\pm$ 0.06	+18.57
0.02	**99.03** $\pm$ **0.07**	+18.78

The results show that even a small non-zero λ produces a dramatic improvement of nearly 19 percentage points. The peak is achieved at λ=0.02; λ=0.01 provides nearly identical AUROC with slightly lower standard deviation, making it the default in our main experiments. Larger λ values risk over-constraining the model, which we leave to future work.

D.2. Monte Carlo Trials Ablation: Detailed Results

Table D.2 characterises the accuracy-efficiency frontier as a function of the number of MC timestep trials K.

Table D.2: K ablation: AUROC (%) and relative inference cost on within-CIFAR split. Reference cost is the discriminative baseline (ResNet-18 forward pass).

K	**AUROC (%)** ↑	**Approx. Inference Cost**	**Notes**
1	91.0	$\sim$3$\times$ discriminative	Low latency
10	98.2	$\sim$32$\times$ discriminative	Recommended trade-off
50	98.64	$\sim$162$\times$ discriminative	Default (within-CIFAR)

The performance gain from K=1 to K=10 is substantial (+7.2 pp), while the gain from K=10 to K=50 is marginal (+0.44 pp). This suggests that K=10 is the practical sweet spot for most deployment scenarios. For the external OOD evaluation, K=100 was used to maximise statistical stability of the reported means; K=50 produces similar AUROC values on those datasets.

D.3. Scoring Strategy: Difference vs. ID-Only

Table D.3 compares contrastive difference scoring ($S = e_0(x) - e_1(x)$, where $c = 0$ is ID and $c = 1$ is non-ID) against single-sided ID-only scoring ($S = e_0(x)$) on selected OOD datasets.

Table D.3: Scoring strategy comparison: AUROC (%) with contrastive difference scoring vs. ID-only scoring. Difference scoring subtracts OOD-proxy class reconstruction error from ID class reconstruction error ($e_0 - e_1$); a high score indicates the sample is OOD.

OOD Dataset	**Difference Scoring**	**ID-Only Scoring**	Δ
SVHN (auditable seed42)	90.50	20.2	+70.30
CIFAR-10 within-split	98.98	–	–

SVHN result: K=100, 3 seeds. Within-CIFAR: K=50, 3 seeds.

ID-only scoring on within-CIFAR not separately reported as it is subsumed by the difference scoring experiments.

The catastrophic failure of ID-only scoring on SVHN (20.2% AUROC—below random chance on a balanced evaluation) occurs because SVHN images have lower absolute reconstruction error under the ID model than many within-CIFAR OOD images, making the absolute magnitude uninformative. Contrastive scoring resolves this by measuring the relative advantage of the ID model, providing a signal that is robust to absolute reconstruction magnitude differences.