Anomaly Detection using Source-Reversed Flow Matching

Amine Youcef Khodja **Jéremie Pantin** **Gaël Dias** **Fabrice Maurel**

Université Caen Normandie, ENSICAEN, CNRS,
Normandie Univ, GREYC UMR 6072, F-14000 Caen, France
firstname.lastname@unicaen.fr

Correspondence: amine.youcef-khodja@unicaen.fr

Abstract

Anomaly detection in textual data remains an underexplored problem, hindered by fragmented benchmarks, scarce labeled data, and methods that fail to model the distribution of normal text in a continuous and generative manner. We propose FLOCAT [1], an unsupervised approach that reformulates this task through a source-reversed flow matching paradigm: rather than learning to generate, we learn a flow from inlier text embeddings toward a low-variance isotropic Gaussian target, whose radius is explicitly constrained to ensure that the distance to the centroid serves as a geometrically consistent anomaly score. Built on a Diffusion Transformer (DiT) backbone, FLOCAT operates at both token and sentence level through a level-agnostic interpolation mechanism, enabling fine-grained signal interpretability. We evaluate on 10 datasets spanning text classification, spam detection, sentiment analysis, human- vs. machine-generated text and depression screening, consistently achieving state-of-the-art AUC-ROC across all benchmarks and text representations.

1 Introduction

Anomaly detection (Chandola et al., 2009; Ruff et al., 2019; Boutalbi et al., 2023) identifies observations that deviate from a notion of normality. Yet anomaly detection in text remains less developed than anomaly detection in vision or tabular data. Textual data raises two specific problems: semantic normality lies in high-dimensional contextual embeddings, whose geometry varies across encoders and domains, and useful systems need both document-level scores and token-level evidence because anomalous signals can be local.

Existing approaches address only part of this setting. One-class classification methods and reconstruction-based models score fixed representations through distances, densities, or reconstruction errors. Recent neural methods add self-supervised or contrastive objectives, and LLM-based approaches rely on prompting or embedding extraction (Ruff et al., 2019; Manolache et al., 2021; Das et al., 2023; Yang et al., 2025; Dong et al., 2024). While effective on specific benchmarks, none learns a continuous transport model of the inlier embedding distribution that yields a geometrically grounded anomaly score.

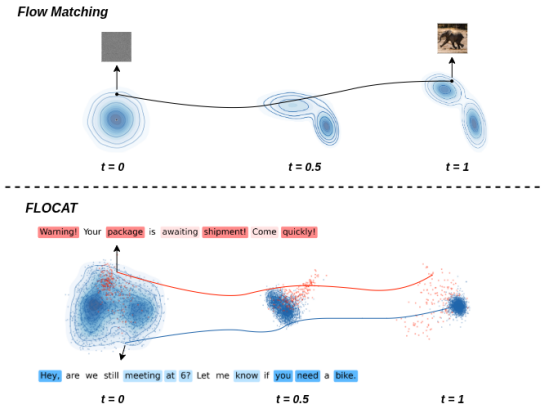


Figure 1: *Top*: standard flow matching learns a generative flow from Gaussian noise toward the data distribution. *Bottom*: FLOCAT inverts this direction and learns a flow from inlier text embeddings toward a low-variance isotropic Gaussian target $p_1 = \mathcal{N}(\boldsymbol{\mu}, \sigma^2\mathbf{I})$. At $t = 1$, inlier representations converge tightly around $\boldsymbol{\mu}$ while anomalous representations remain scattered, yielding a geometrically consistent anomaly score.

Flow matching (FM) (Lipman et al., 2023; Liu et al., 2022) offers a natural tool for this gap. In its standard generative form, FM learns a time-dependent velocity field that transports a simple source distribution toward the data distribution. Recent anomaly detection methods reverse this direction in tabular data (Li et al., 2025b) and images (Li et al., 2025a): they contract normal samples toward a fixed target and score deviations from this transport. FLOCAT extends this principle to text, as

[1] The code is available at https://github.com/AmineYK/Anomaly-Detection-Using-Source-Reversed-Flow-Matching.

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 52746–52767
October 24-29, 2026 ©2026 Association for Computational Linguistics

illustrated in Figure 1. Text makes this direction non-trivial to apply directly. Inputs are variable-length sequences of contextual embeddings, detection must work at sentence and token granularities, and fine-grained explanations require token-level signals aligned with the anomaly score.

In this work, we propose FLOCAT, an unsupervised approach for anomaly detection in text built on source-reversed flow matching. While this reversal paradigm has been explored for tabular data (Li et al., 2025b) and images (Li et al., 2025a), applying it to text raises specific challenges that require dedicated solutions. For handling text granularity, FLOCAT uses a level-agnostic Diffusion Transformer (DiT) (Peebles and Xie, 2023): token residuals inject token-specific information into the DiT input, learned velocity pooling maps token states back to a sentence-level velocity, and the same forward pass handles sentence-level and token-level inputs. Token-level velocities yield scalar projection attribution scores, providing fine-grained interpretability without additional supervision. We evaluate FLOCAT on 10 datasets and five encoder backbones, achieving state-of-the-art AUC-ROC for each encoder setting on the main benchmarks. We also evaluate on two application scenarios rarely addressed in unsupervised settings: human- vs. machine-generated text detection and depression screening, where normality is less sharply defined than in standard topic-based benchmarks.

2 Related Work

2.1 Anomaly Detection in Text

Anomaly detection has been widely studied in fraud detection, medical diagnosis, cybersecurity, and other domains (Chandola et al., 2009; Pang et al., 2021; Ruff et al., 2021). Classical methods such as OC-SVM (Schölkopf et al., 2001), Isolation Forest (Liu et al., 2008), and LOF (Breunig et al., 2000) score samples through geometric or density criteria, while autoencoders (Hinton and Salakhutdinov, 2006) and Deep SVDD (Ruff et al., 2018) use reconstruction or hypersphere objectives. In text, these methods are often used in two-stage pipelines with pre-trained representations such as SBERT (Reimers and Gurevych, 2019).

Text-specific detectors refine this idea with architectures and objectives adapted to language. CVDD (Ruff et al., 2019) learns static context vectors with self-attention; DATE (Manolache et al., 2021) combines token and sequence level self-supervision; FATE (Das et al., 2023) uses labelled anomalies in a deviation-learning objective; and RoSAE (Pantin and Marsala, 2024) and RSRAE (Lai et al., 2020) constrain latent spaces through robust subspace recovery. LLM-based approaches form a parallel line, using zero or few-shot prompting (Yang et al., 2025; Dong et al., 2024), domain-specific fine-tuning (Gu et al., 2024; Li et al., 2024a), or embeddings extracted from large models (Li et al., 2024b; Cao et al., 2025). These methods improve semantic coverage, but most still score fixed representations, or decouple document-level detection from token-level evidence.

2.2 Flow Matching

Generative modelling has a long connection to anomaly detection through the hypothesis that models trained on normal data assign low likelihood to anomalous inputs (Nalisnick et al., 2019). Normalising flows (Rezende and Mohamed, 2015; Kingma and Dhariwal, 2018; Albergo and Vanden-Eijnden, 2022) implement this idea through invertible mappings and have been used for image and time-series anomaly detection: CFlow-AD (Gudovskiy et al., 2022) and MSFlow (Zhou et al., 2024) exploit multi-scale likelihoods for industrial inspection, while GANF (Dai and Chen, 2022) and SGFM (He et al., 2024) target multivariate time series. FlowSVDD (Sendera et al., 2021) further combines normalising flows with the hypersphere objective of Deep SVDD. However, strict invertibility constrains model design and scalability (Li et al., 2025b,a).

Flow Matching (FM) (Lipman et al., 2023; Liu et al., 2022) relaxes these constraints by learning a time-dependent vector field that transports a source distribution toward a target along deterministic probability paths. This approach has recently entered anomaly detection through time-reversed or contractive formulations: TCCM (Li et al., 2025b) learns a one-step contraction field for tabular data, and WT-Flow (Li et al., 2025a) formalises time-reversed FM for image anomaly detection. These works show that a learned transport toward a concentrated target can produce effective anomaly scores, but they do not address variable-length text embeddings or token-level attribution.

2.3 Flow Matching for Text Representation

Adapting FM to language is non-trivial since text is discrete while FM is defined on continuous spaces.

One line of work operates in embedding space: Diffusion-LM (Li et al., 2022) denoises Gaussian vectors into word embeddings, and FlowSeq (Hu et al., 2024) learns trajectories between text embeddings and Gaussian noise for few-step generation. A second line moves to token or probability space; Discrete Flow Matching (Gat et al., 2024) defines probability paths over categorical data for language and code generation. All these works remain generative: they synthesise or edit sequences and do not study FM as a representation-learning operator for scoring. The central problem in anomaly detection is not generation, but a scoring geometry that remains meaningful across sentence and token granularities.

3 FLOCAT: FLOw Matching toward a CompAct Target Distribution

3.1 Setup and Notation

Let $\mathcal{D} = \{x^i\}_{i=1}^N$ be a training corpus of N normal (inlier) documents, with no anomaly labels available during training. Each document x^i is mapped to a continuous representation by a pretrained encoder: either a single sentence embedding $\mathbf{z}^i \in \mathbb{R}^d$, or a sequence of S token embeddings $\mathbf{Z}^i = (\mathbf{z}^{i,1}, \ldots, \mathbf{z}^{i,S}) \in \mathbb{R}^{S\times d}$. In both cases, the sentence-level representation $\mathbf{z}^i$ is used as the flow source, obtained directly at the sentence level or via mean pooling of $\mathbf{Z}^i$ at the token level. When token embeddings are used, this source representation is $\mathbf{z}^i = \frac{1}{S}\sum_{k=1}^S \mathbf{z}^{i,k}$. Throughout the paper, non-bold x^i denotes a document, while bold $\mathbf{x}_t$ denotes a flow state. The set $\{\mathbf{z}^i\}_{i=1}^N$ defines the source distribution p_0.

We define $p_1 = \mathcal{N}(\boldsymbol{\mu}, \sigma^2\mathbf{I})$, the target distribution, as an isotropic Gaussian, where the centroid $\boldsymbol{\mu}$ and variance σ^2 are set from the source distribution:

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^N \mathbf{z}^i \tag{1}$$

$$\sigma^2 = \alpha_c \cdot \frac{1}{Nd}\sum_{i=1}^N \left\|\mathbf{z}^i - \boldsymbol{\mu}\right\|^2 \tag{2}$$

where $\alpha_c \in (0,1)$ is the variance ratio controlling the tightness of the target relative to the source spread. This anchors p_1 in the same region of $\mathbb{R}^d$ as p_0, ensuring scale consistency.

3.2 Flow Matching Background

Flow Matching (FM) (Lipman et al., 2023) is a simulation-free framework for learning continuous generative models via regression on a vector field. Given a source p_0 and a target p_1, FM learns a time-dependent vector field $v_\theta : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ that generates a probability path $\{p_t\}_{t\in[0,1]}$ interpolating between p_0 and p_1 by solving the flow ODE:

$$\frac{d\mathbf{x}_t}{dt} = v_\theta(\mathbf{x}_t, t), \quad \mathbf{x}_0 \sim p_0 \tag{3}$$

Rather than maximising likelihood directly, FM regresses v_θ against a conditional vector field on sample pairs, yielding a tractable and stable objective. For anomaly detection, standard FM still leaves the endpoint pairing problem open: the transport must expose deviations from normal data, not only model the data distribution.

Conditional Flow Matching. Given a pair $(\mathbf{x}_0, \mathbf{x}_1)$ with $\mathbf{x}_0 \sim p_0$ and $\mathbf{x}_1 \sim p_1$, the intermediate state at time $t \in [0,1]$ follows linear interpolation:

$$\mathbf{x}_t = (1-t)\,\mathbf{x}_0 + t\,\mathbf{x}_1 \tag{4}$$

The corresponding conditional vector field that generates this path is:

$$u_t(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_1) = \mathbf{x}_1 - \mathbf{x}_0 \tag{5}$$

The Conditional Flow Matching (CFM) objective minimises the expected squared error between the learned field and this target (Lipman et al., 2023; Esser et al., 2024):

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{t,\,\mathbf{x}_0\sim p_0,\,\mathbf{x}_1\sim p_1}\left\|v_\theta(\mathbf{x}_t, t) - u_t(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_1)\right\|^2 \tag{6}$$

Lipman et al. (Lipman et al., 2023) show that this conditional objective matches the marginal FM objective, so CFM gives a stable training rule. In our unsupervised setup, training provides only inlier documents, so the transport must concentrate normal embeddings around a fixed target and make anomalous text deviate from the learned field.

One-Class Flow Matching. The one-class setting changes the endpoint construction of CFM. Each inlier embedding defines the source endpoint, $\mathbf{x}_0^{\mathrm{sent},i} = \mathbf{z}^i$, while a sample from the low-variance target defines the terminal endpoint, $\mathbf{x}_1^i \sim p_1$. The learned field therefore transports normal text representations from the empirical inlier distribution

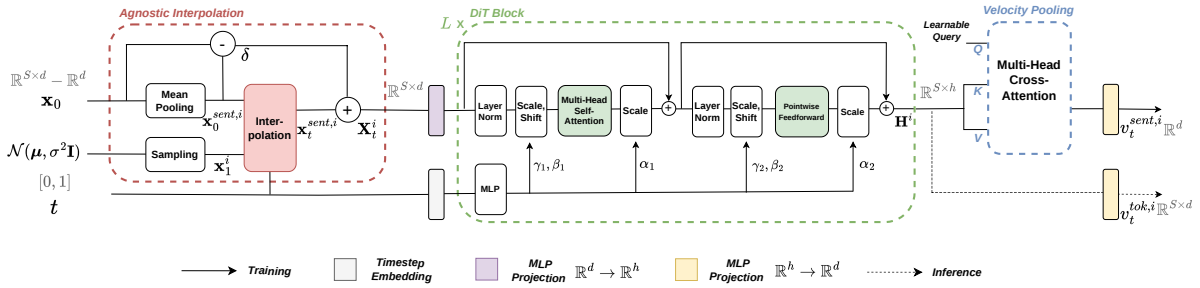


Figure 2: Overview of the FLOCAT framework. The agnostic interpolation (Eq. 4) and residual broadcast (Eq. 7) compute $\mathbf{x}_t^{\text{sent},i}$ and broadcast it to each token via residual injection. The DiT backbone processes the resulting sequence through adaLN-conditioned blocks and aggregates token hidden states into a sentence-level velocity $v_t^{\text{sent},i}$ via learned velocity pooling (Eq. 8). Token-level velocities $v_t^{\text{tok},i}$ are used for anomaly attribution at inference.

toward a low-variance Gaussian anchored at the inlier centroid. This source-reversed transport turns normality into a geometric criterion: inliers follow a coherent path to p_1, whereas anomalies induce inconsistent velocities and land far from the target centroid. This principle extends flow-based anomaly detectors from tabular and image domains (Li et al., 2025b,a) to text representations.

3.3 FLOCAT

Figure 2 provides an overview of FLOCAT. An agnostic interpolation module prepares the input at both granularity levels; a DiT-based backbone processes the resulting sequence and a velocity pooling module aggregates token hidden states into a sentence-level velocity, yielding: $v_\theta(\mathbf{X}_t, t) \rightarrow (v_\theta^{\text{sent}}, v_\theta^{\text{tok}})$.

Agnostic interpolation and residual broadcast. The flow operates at the sentence level regardless of input granularity. Given $\mathbf{x}_0^{\text{sent},i} = \mathbf{z}^i$ and a sample $\mathbf{x}_1^i \sim p_1$, the interpolated sentence-level state follows Eq. 4. When token embeddings are provided, the token-level input to the DiT is constructed by injecting the per-token residual $\boldsymbol{\delta}^{i,k} = \mathbf{z}^{i,k} - \mathbf{z}^i$:

$$\mathbf{x}_t^{i,k} = \mathbf{x}_t^{\text{sent},i} + \boldsymbol{\delta}^{i,k} \tag{7}$$

We denote the resulting Diffusion Transformer (DiT) input sequence by $\mathbf{X}_t^i = (\mathbf{x}_t^{i,1}, \ldots, \mathbf{x}_t^{i,S})$. $\boldsymbol{\delta}^{i,k}$ encodes both the direction and magnitude of the deviation of token k from the sentence centroid, preserving token-specific structure while keeping the interpolation trajectory defined at the sentence level. At sentence-level, $\boldsymbol{\delta}^{i,1} = \mathbf{0}$ vanishes identically and Eq. 7 reduces to $\mathbf{x}_t^{i,1} = \mathbf{x}_t^{\text{sent},i}$, so the same forward pass handles both granularities without modification.

Diffusion Transformer. The token sequence $\mathbf{X}_t^i$ is projected to a hidden dimension h and processed by L stacked DiT blocks (Peebles and Xie, 2023). A timestep embedding $\tau(t)$ conditions each block through adaptive layer normalisation (adaLN) (Peebles and Xie, 2023), which produces shift, scale, and gate parameters for the self-attention and feed-forward branches. Self-attention propagates token residual information across the sequence, while timestep conditioning makes the predicted velocity depend on the position along the interpolation path. We denote the final hidden sequence by $\mathbf{H}^i = \mathbf{H}_L^i \in \mathbb{R}^{S\times h}$. For sentence-level inputs ($S = 1$), the sequence reduces to a single token and the same backbone applies unchanged.

Velocity pooling. Since the FM loss and the anomaly score operate at the sentence level (Eq. 9, Eq. 14), the token-level hidden states produced by the DiT must be aggregated back into a single sentence-level velocity. A learned cross-attention module achieves this using a trainable query $\mathbf{q} \in \mathbb{R}^h$ over the token hidden states $\mathbf{H}^i \in \mathbb{R}^{S\times h}$:

$$\mathbf{h}^{\text{sent},i} = \text{MHCA}(\mathbf{q}, \mathbf{H}^i, \mathbf{H}^i) \tag{8}$$

This is passed through an adaLN-modulated linear projection to produce $v_t^{\text{sent},i} \in \mathbb{R}^d$. Token-level velocities $v_t^{\text{tok},i} = (v_t^{i,1}, \ldots, v_t^{i,S}) \in \mathbb{R}^{S\times d}$ are obtained by applying the same projection to each token hidden state independently, and are used for anomaly attribution at inference (Section 3.5).

Flow Matching Loss. We minimise the squared error between the predicted sentence-level velocity and the conditional target field $u^i = \mathbf{x}_1^i - \mathbf{x}_0^{\text{sent},i}$:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{t,\, \mathbf{x}_0^{\text{sent},i}\sim p_0,\, \mathbf{x}_1^i \sim p_1} \left\| v_t^{\text{sent},i} - u^i \right\|^2 \tag{9}$$

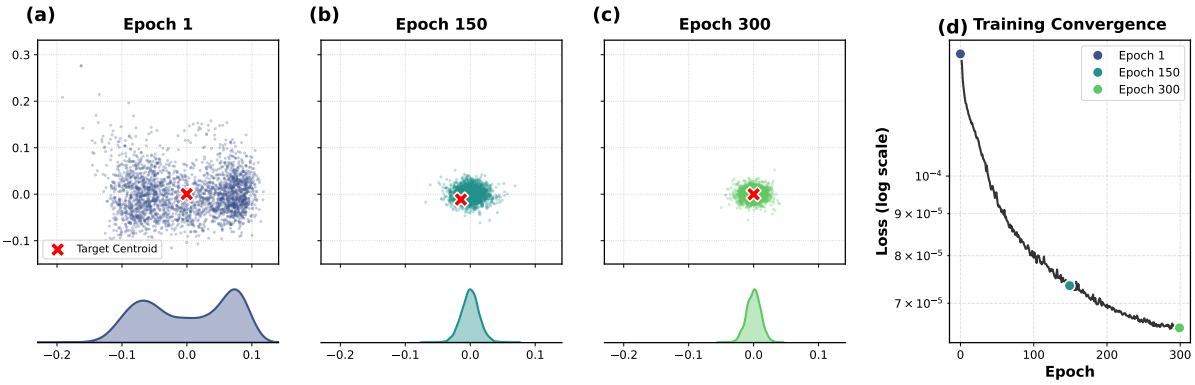


Figure 3: Evolution of transported inlier representations $\hat{\mathbf{x}}_1$ at epochs 1, 150, and 300. The low-variance constraint progressively concentrates inliers around $\boldsymbol{\mu}$, reducing the effective radius of the hypersphere.

Low-Variance Loss. $\mathcal{L}_{\text{FM}}$ matches local velocities along sampled paths but does not control the geometry of the transported endpoints: small velocity errors can leave inliers dispersed around the target. Following hypersphere-based one-class objectives (Ruff et al., 2018), we impose a low-variance loss directly in the target space. Accessing this space requires integrating the learned velocity field; for the linear CFM path, a single Euler step at $t = 0$ provides a computationally cheap estimate of the transported endpoint:

$$\hat{\mathbf{x}}_1^i = \mathbf{x}_0^{\text{sent},i} + v_\theta^{\text{sent}}(\mathbf{X}_0^i,\, t{=}0) \tag{10}$$

The low-variance loss $\mathcal{L}_{\text{LOVE}}$ [2] fits these endpoints inside a soft hypersphere of learnable radius R centred at the fixed inlier centroid $\boldsymbol{\mu}$:

$$\mathcal{L}_{\text{LOVE}} = R^2 + \frac{1}{B}\sum_{i=1}^{B} \ell_i \tag{11}$$

R is optimised jointly with θ, and we set $\ell_i = \max\left(0, \left\|\hat{\mathbf{x}}_1^i - \boldsymbol{\mu}\right\|^2 - R^2\right)$. As shown in Figure 5, the radius term shrinks the sphere, while the hinge term penalises inlier endpoints outside it. The fixed centroid $\boldsymbol{\mu}$ and the FM objective prevent the degenerate solution in which the learned field collapses all inputs without preserving the conditional transport structure. Figure 3 illustrates the progressive compaction of the target space during training; an ablation of $\mathcal{L}_{\text{LOVE}}$ is provided in Appendix K.

FLOCAT Loss. Training optimises the velocity-field parameters θ and the radius R with a two-term objective:

$$\mathcal{L}_{\text{FLOCAT}}(\theta, R) = \mathcal{L}_{\text{FM}}(\theta) + \lambda\, \mathcal{L}_{\text{LOVE}}(\theta, R) \tag{12}$$

[2] LOVE : LOw-VariancE

The FM term preserves the conditional transport field, while the low-variance term contracts transported inliers around the fixed target centroid. The coefficient $\lambda > 0$ controls this trade-off. The complete training procedure is summarised in Algorithm 1.

3.4 Anomaly Scoring

At inference, the flow ODE (Eq. 3) cannot be solved analytically and is approximated numerically. We use the Euler method (Li et al., 2025b,a) with T discretisation steps:

$$\phi(\mathbf{x}_0^{\text{sent},i}) \approx \mathbf{x}_0^{\text{sent},i} + \frac{1}{T}\sum_{m=0}^{T-1} v_\theta^{\text{sent}}\left(\mathbf{X}_{m/T}^i,\, \frac{m}{T}\right) \tag{13}$$

The anomaly score is defined as the squared distance between the transported representation and the target centroid:

$$s(x^i) = \left\|\phi(\mathbf{x}_0^{\text{sent},i}) - \boldsymbol{\mu}\right\|^2 \tag{14}$$

This score has a direct probabilistic interpretation as the NLL of the transported point under p_1, up to constants (Nalisnick et al., 2018; Ruff et al., 2018). Inliers are mapped close to $\boldsymbol{\mu}$ and receive low scores, while anomalies deviate from the learned transport and receive high scores. Figure 4 and 6 provides qualitative evidence of this behaviour: inlier trajectories converge smoothly toward $\boldsymbol{\mu}$, while anomaly trajectories are irregular and diverge.

3.5 Transport-Based Token Attribution

Beyond document-level anomaly scoring, FLOCAT provides token-level attribution scores without additional supervision. We define the attribution score of token k as the scalar projection of its velocity onto the sentence-level velocity. Starting from the

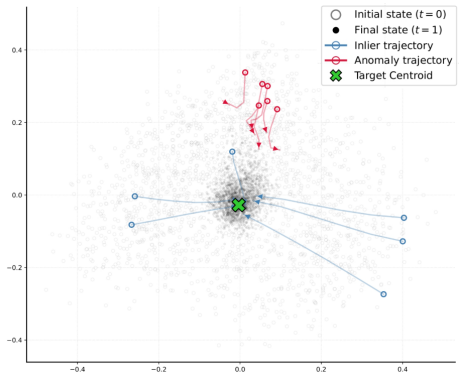


Figure 4: Flow trajectories of inlier and anomaly samples. Inlier trajectories converge toward $\boldsymbol{\mu}$; anomaly trajectories diverge, producing high anomaly scores.

cosine similarity between $v_t^{i,k}$ and $v_t^{\text{sent},i}$:

$$\cos^{i,k} = \frac{v_t^{i,k} \cdot v_t^{\text{sent},i}}{\left\| v_t^{i,k} \right\|_2 \left\| v_t^{\text{sent},i} \right\|_2} \tag{15}$$

which captures directional alignment but ignores magnitude, we recover both direction and magnitude by multiplying by the token velocity norm:

$$a^{i,k} = \cos^{i,k} \cdot \left\| v_t^{i,k} \right\|_2 = \frac{v_t^{i,k} \cdot v_t^{\text{sent},i}}{\left\| v_t^{\text{sent},i} \right\|_2} \tag{16}$$

This is the standard scalar projection of $v_t^{i,k}$ onto the direction of $v_t^{\text{sent},i}$, and measures how much token k effectively contributes to the global document transport. Scores are then normalised over the sequence to obtain

$$\tilde{a}^{i,k} = \frac{a^{i,k} - \min_j a^{i,j}}{\max_j a^{i,j} - \min_j a^{i,j} + \varepsilon} \in [0, 1], \tag{17}$$

where $\varepsilon > 0$ avoids division by zero. For a normal document, tokens with high $\tilde{a}^{i,k}$ contribute most to the transport toward $\boldsymbol{\mu}$: these are the tokens most characteristic of the inlier class. For an anomalous document, tokens with low $\tilde{a}^{i,k}$ resist or fail to align with the sentence-level transport: these are the primary drivers of the anomaly decision. This signal emerges directly from the learned velocity field, requiring no post-hoc analysis.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate FLOCAT on ten datasets covering a broad range of text domains and tasks. The main benchmarks consist of eight datasets spanning three NLP tasks: text classification (20Newsgroups (Lang, 1995), Reuters (Lewis, 1997), AGNews (Zhang et al., 2015), DBpedia14 (Zhang et al., 2015)), spam detection (SMSSpam (Almeida et al., 2011), Enron (Klimt and Yang, 2004)), and sentiment analysis (IMDB (Maas et al., 2011), SST-2 (Socher et al., 2013)). We additionally evaluate on two application datasets: M4 (Wang et al., 2024b) for human- vs. machine-generated text detection, and DAIC-WoZ (Gratch et al., 2014) for depression screening.

Evaluation protocol. Following standard practice in unsupervised text anomaly detection (Ruff et al., 2019; Manolache et al., 2021), each dataset contains multiple categories; one category is designated as the inlier class and all remaining categories are treated as anomalies. The model is trained exclusively on inlier documents. For the test set, we sample 10% of the anomaly pool to construct a balanced evaluation set, ensuring that no anomaly information leaks into training. Category assignments follow prior work. All results are reported as mean and standard deviation over 5 independent runs, where the anomaly test sample is redrawn at each run to avoid bias toward easy anomaly scenarios. The primary evaluation metric is AUC-ROC.

Baselines. We compare against five baselines at two levels of input granularity. Granularity refers to the level at which each method processes its input: sentence or token, not to the scoring level, which is sentence-level in all cases. At the sentence level: FATE (Das et al., 2023), RSRAE (Lai et al., 2020), and TCCM (Li et al., 2025b). At the token level: CVDD (Ruff et al., 2019), DATE (Manolache et al., 2021), and RSRAE and TCCM applied at the token level. TCCM is adapted to text by replacing its tabular source distribution with pre-trained text embeddings, as indicated by † in the tables. DATE and FATE are run using their official implementations; all other baselines are reimplemented by us. We further include two instruction-tuned LLMs in few-shot mode (3 inlier examples) following Yang et al. (2025): Mistral-7B-Instruct (Jiang et al., 2023) and Qwen2.5-7B-Instruct (Hui et al., 2024).

Embeddings. We evaluate all methods under multiple pre-trained encoders spanning different model families, sizes, and training objectives. At the sentence level: MPNet (Song et al., 2020), a strong all-round encoder widely used in anomaly detection (Novoa-Paradela et al., 2024; Siino,

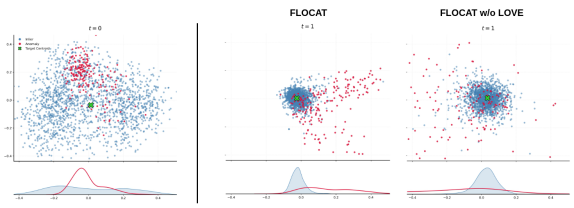


Figure 5: Transported inlier representations $\hat{\mathbf{x}}_1$ in the flow target space for FLOCAT and FLOCAT w/o LOVE. The low-variance constraint concentrates inliers into a tight region around $\boldsymbol{\mu}$, whereas without it the representations remain dispersed.

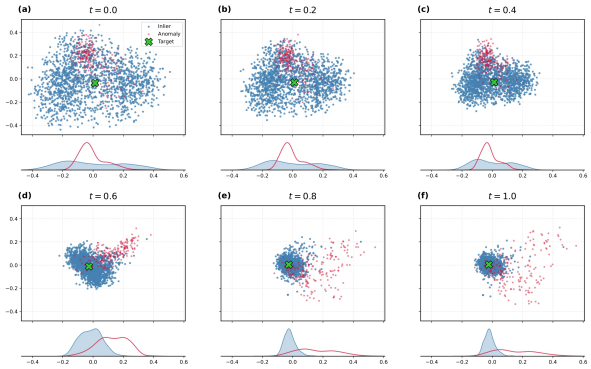


Figure 6: Transported representations across Euler steps ($t = 0 \rightarrow 1$). Inliers (blue) converge toward $\boldsymbol{\mu}$ while anomalies (red) remain scattered, producing high scores $s(x^i)$ at $t = 1$.

2024); ST5 (Ni et al., 2022), fine-tuned from T5 (Raffel et al., 2020) and adopted for semantic similarity tasks (Yano et al., 2024); and E5-Mistral (Wang et al., 2024a), an instruction-tuned model built on Mistral-7B representing the frontier of large-scale text embeddings. At the token level: RoBERTa (Liu et al., 2019), the standard contextual encoder for text anomaly detection (Manolache et al., 2021; Das et al., 2023); and Qwen (Bai et al., 2023), a large autoregressive model increasingly used as a feature extractor for downstream NLP tasks.

4.2 Main Results

Table 1 reports AUC-ROC on the eight main benchmarks, averaged over all inlier topic configurations. FLOCAT consistently achieves state-of-the-art AUC-ROC results across all encoder backbones and dataset groups. At the sentence level, our method outperforms all baselines on the vast majority of benchmarks, with particularly strong gains on spam detection and text classification, where the inlier distribution is well-defined at the sentence level. The only datasets where sentence-level methods collectively struggle are SST-2 and IMDB: sentiment texts induce fine-grained token-level distinctions that are diluted when averaging over the full sequence, making sentence-level representations inherently less discriminative for this task. Switching to our token-level variant directly addresses this limitation, surpassing the baselines on SST-2 and IMDB. On SMSSpam, token-level FLOCAT also outperforms its sentence-level counterpart, confirming that the discriminative signal in this task is local and distributed across individual tokens rather than global. TCCM remains the strongest competitor, further validating flow matching for text anomaly detection; LLM-based baselines fall short across all settings despite their significantly higher inference cost.

4.3 Applications

Beyond standard benchmarks, we evaluate FLOCAT on two tasks where token-level interpretability matters most: human- vs. machine-generated text detection and depression screening. Both tasks are commonly framed as binary classification, yet we argue they are more naturally cast as anomaly detection problems: the notion of normality is well-defined (machine-generated text; non-depressed speech), labelled anomalies are scarce, and the decision should be explainable at the token level (Agarwal et al., 2024). Table 2 reports AUC-ROC on both datasets. To assess the quality of the attribution scores, we compare FLOCAT scalar projection scores ($\tilde{a}^{i,k}$) against LIME (Ribeiro et al., 2016) attributions computed on the same anomaly score $s(x^i)$; the top words identified by each method are shown in Tables 3 and 4.

4.3.1 Discussion

Tables 3 and 4 show that FLOCAT scalar projection scores recover attribution patterns broadly consistent with LIME, despite requiring no perturbations. On M4/PeerRead, both methods identify structured, domain-specific tokens as characteristic of machine-generated text (*formula*, *parametric*, *generative*), while flagging evaluative and opinionated vocabulary as anomalous (*creative*, *interesting*, *wellwritten*). On DAIC-WoZ, both highlight socially grounded tokens for non-depressed speech (*parents*, *economic*, *opportunities*) and distress-related vocabulary for depressed speech (*homeless*, *hard*, *selfish*). The key practical difference

Level	Emb.	Method	Text Classification				Spam		Sentiment Analysis		Avg.
			Reuters	20News	AGNews	DBpedia	SMSSpam	Enron	SST-2	IMDB	
Token	RoBERTa	DATE	85.80±9.5	73.57±6.8	79.67±3.5	69.84±9.6	67.33±19.3	62.45±14.7	53.71±5.1	52.05±2.0	68.05
		CVDD	84.90±7.6	69.56±3.2	73.70±1.1	94.55±0.4	75.81±10.7	78.20±0.6	49.54±2.8	56.48±0.5	72.84
		RSRAE	83.83±4.9	71.50±2.1	83.58±1.4	93.07±0.2	45.40±1.3	82.15±1.5	53.12±5.2	60.00±0.2	71.58
		TCCM†	75.67±7.9	72.13±2.2	87.13±1.2	92.30±0.2	46.29±1.3	82.44±1.8	53.78±5.4	59.01±1.2	71.09
		FLOCAT	**94.83±3.3**	**76.82±4.2**	**90.54±0.8**	**96.78±0.2**	**76.68±1.3**	**87.66±1.4**	**61.49±1.3**	**60.19±0.4**	**80.62**
	Qwen	DATE	–	–	–	–	–	–	–	–	–
		CVDD	88.41±7.4	75.89±9.7	84.56±1.4	90.75±0.4	68.10±1.8	82.15±1.9	49.61±2.4	56.95±0.9	74.55
		RSRAE	76.24±9.1	68.20±2.5	66.17±0.9	69.31±0.7	66.43±1.2	65.16±1.1	48.12±4.1	53.41±0.4	64.13
		TCCM†	83.80±8.9	72.48±2.5	84.01±0.8	89.25±0.6	69.20±0.2	85.70±1.9	52.20±2.4	57.17±0.2	74.23
		FLOCAT	**90.86±4.0**	**76.30±2.9**	**85.77±1.4**	**91.54±0.4**	**72.58±0.1**	**92.38±0.0**	**54.10±1.4**	**59.43±1.9**	**77.87**
Sentence	MPNet	FATE	90.33±5.2	91.17±1.7	92.91±1.4	96.88±0.1	55.79±2.8	65.43±25.1	50.25±2.4	47.51±3.5	73.78
		RSRAE	98.11±1.3	92.81±0.9	93.68±0.5	98.57±0.1	57.96±1.7	95.05±0.5	**61.20±1.9**	50.51±0.8	80.99
		TCCM†	98.28±2.0	93.45±0.9	94.03±0.5	98.45±0.1	57.45±1.0	93.81±0.7	60.38±1.3	48.58±0.4	80.55
		FLOCAT	**98.50±1.5**	**93.89±0.9**	**94.56±0.3**	**98.83±0.1**	**66.30±0.7**	**95.65±0.5**	59.91±1.6	**56.09±0.5**	**82.97**
	E5-Mistral	FATE	–	–	–	–	–	–	–	–	–
		RSRAE	96.46±2.1	88.91±0.9	93.04±0.5	99.47±0.1	57.74±1.7	91.11±1.3	**64.33±1.6**	**69.52±0.5**	82.57
		TCCM†	95.10±3.4	89.30±0.9	93.33±0.4	99.35±0.1	60.73±2.0	92.97±1.3	61.30±1.5	63.46±0.5	81.94
		FLOCAT	**97.11±1.8**	**89.79±1.8**	**93.84±0.4**	**99.50±0.1**	**65.37±12.1**	**95.01±1.1**	61.16±1.4	68.64±1.9	**83.80**
	ST5	FATE	–	–	–	–	–	–	–	–	–
		RSRAE	96.41±2.8	84.65±1.3	90.86±0.7	98.91±0.1	62.81±1.1	90.61±0.8	**71.05±2.0**	62.61±0.5	82.24
		TCCM†	96.04±3.5	86.13±1.3	91.53±0.7	98.67±0.1	63.54±1.1	92.50±0.8	70.19±2.1	61.62±0.6	82.53
		FLOCAT	**96.69±2.8**	**88.73±2.2**	**91.90±0.5**	**98.99±0.1**	**72.25±2.4**	**94.10±0.6**	65.93±2.8	**63.97±1.4**	**84.07**
LLM-based											
	–	Qwen2.5-7B-Instruct	80.94±5.9	73.87±2.0	82.30±0.5	92.62±0.6	75.37±0.9	90.68±0.4	64.71±2.8	67.14±1.2	78.45
	–	Mistral-7B-Instruct	75.03±5.4	67.31±1.6	79.89±0.6	89.54±0.4	74.19±0.3	83.50±0.5	65.26±4.2	61.32±1.6	74.51

Table 1: AUC-ROC on main benchmarks. Mean ± std over 5 runs. **Bold**: best. Underline: second best. †: methods adapted from non-text domains. –: original implementations do not support this encoder.

Level	Emb.	Method	H-. vs. M-.	Dep. Scr.
			M4	DAIC-WoZ
Token	RoBERTa	DATE	59.02±5.0	43.51±2.0
		CVDD	45.41±1.0	52.81±0.0
		RSRAE	63.18±1.0	36.36±0.0
		TCCM†	65.18±1.0	38.96±0.0
		FLOCAT	**68.92±2.0**	**60.61±3.0**

Table 2: AUC-ROC on application datasets. Mean ± std over 5 runs. †: methods adapted from non-text domains.

Inlier (machine-generated)		Anomaly (human-written)	
FLOCAT	LIME	FLOCAT	LIME
formula	sentences	outperforms	lack
physics	generative	wellwritten	creative
parametric	completion	proposes	lower
policies	network	different	inference
component	paper	representation	model
level	firstly	additional	convincing
programming	describe	interesting	usual
objective	areas	learn	learned
series	this	generative	great
prior	very	learning	interesting

Table 3: Top attributed words on M4/PeerRead. FLOCAT scores are scalar projections $\tilde{a}^{i,k}$; LIME scores are attribution weights on $s(x^i)$.

is computational cost: LIME requires ∼300 forward passes per document to estimate attribution via local perturbations, whereas FLOCAT attribution scores $\tilde{a}^{i,k}$ are a direct by-product of a single inference pass, adding zero overhead to the anomaly scoring pipeline.

4.3.2 Faithfulness Evaluation

To quantitatively assess the attribution scores $\tilde{a}^{i,k}$, we conduct a masking-based faithfulness evaluation on FLOCAT. For each document, we mask the top-K tokens identified by each attribution method and measure the resulting AUC-ROC drop Δ relative to the unmasked baseline. We compare against LIME (Ribeiro et al., 2016) with 150 and 300 perturbation samples per document, and against random masking as a lower bound. Table 5 reports results on M4 and DAIC-WoZ.

FLOCAT attribution strongly outperforms random masking across all configurations, confirming that $\tilde{a}^{i,k}$ identifies informative tokens. It surpasses LIME (150) in most configurations and remains close to LIME (300), while requiring a single forward pass compared to 150–300 for LIME.

5 Conclusion

FLOCAT demonstrates that source-reversed flow matching is a principled and effective paradigm

Inlier (non-depressed)		Anomaly (depressed)	
FLOCAT	**LIME**	**FLOCAT**	**LIME**
parents	florida	money	somebody
nineteen	angeles	stuff	state
angeles	shit	work	selfish
weather	opposite	job	bring
environment	smoking	different	quiet
five	achieved	hard	anxiety
opportunities	conservative	love	laughter
traffic	economic	homeless	becoming
business	serious	sure	except
studied	tendency	three	problems

Table 4: Top attributed words on DAIC-WoZ. FLOCAT scores are scalar projections $\tilde{a}^{i,k}$; LIME scores are attribution weights on $s(x^i)$.

Dataset	Method	Top-10	Top-20	Top-1%	Top-5%	Passes
M4	LIME (300)	**5.12**$_{\pm 0.45}$	**8.45**$_{\pm 0.62}$	**12.10**$_{\pm 0.85}$	**18.12**$_{\pm 0.95}$	300
	LIME (150)	4.45$_{\pm 0.58}$	7.10$_{\pm 0.75}$	10.15$_{\pm 0.92}$	16.25$_{\pm 1.20}$	150
	FLOCAT	4.40$_{\pm 0.51}$	7.25$_{\pm 0.68}$	10.42$_{\pm 0.88}$	16.42$_{\pm 1.12}$	**1**
	Random	0.85$_{\pm 1.10}$	1.40$_{\pm 1.25}$	2.15$_{\pm 1.40}$	9.15$_{\pm 1.55}$	–
DAIC-WoZ	LIME (300)	**2.10**$_{\pm 0.55}$	**3.15**$_{\pm 0.72}$	**3.20**$_{\pm 0.81}$	**4.10**$_{\pm 1.30}$	300
	LIME (150)	1.75$_{\pm 0.65}$	2.55$_{\pm 0.88}$	2.65$_{\pm 0.95}$	3.80$_{\pm 1.50}$	150
	FLOCAT	1.82$_{\pm 0.62}$	2.60$_{\pm 0.85}$	2.55$_{\pm 0.92}$	3.85$_{\pm 1.45}$	**1**
	Random	0.30$_{\pm 0.95}$	0.55$_{\pm 1.20}$	0.60$_{\pm 1.35}$	1.50$_{\pm 1.85}$	–

Table 5: Faithfulness evaluation: AUC-ROC drop Δ (%) after masking the top-K tokens identified by each method. **Bold**: best. Underline: second best. Last column: number of forward passes per document.

for unsupervised text anomaly detection. By transporting inlier embeddings toward a low-variance Gaussian target, the model turns geometric deviation into an anomaly signal that generalises across tasks, encoders, and input granularities. Beyond detection, the scalar projection attribution scores show that meaningful linguistic and clinical signals, distinguishing machine from human writing or healthy from depressive speech, emerge without any domain-specific supervision as a direct byproduct of the learned transport.

Acknowledgements

This work was performed using the computing resources of CRIANN (Normandy, France).

Limitations

FLOCAT has several limitations. First, it relies on frozen pre-trained encoders; joint fine-tuning could yield richer representations at higher computational cost. Second, mean pooling to obtain the sentence-level source embedding may not be optimal for documents with highly uneven token importance. Third, token-level attribution is currently evaluated through a faithfulness protocol based on token masking; a more direct evaluation against datasets with ground-truth token-level anomaly annotations, such as BEA-2019, FCE, or BLiMP, would provide stronger evidence for interpretability and remains future work.

Ethical Considerations

FLOCAT targets applications such as spam filtering, content moderation, and depression screening, which carry risks alongside their benefits. Systems trained on a fixed notion of normality may inadvertently penalise underrepresented linguistic communities, raising fairness concerns. In depression screening specifically, false positives could stigmatise individuals while false negatives may leave vulnerable patients undetected; any deployment must be framed as a decision-support tool under qualified clinical supervision, with clear uncertainty communication and strict privacy compliance. Finally, token-level highlighting could be misused for surveillance; responsible deployment and appropriate governance are essential.

References

Navneet Agarwal, Kirill Milintsevich, Lucie Metivier, Maud Rotharmel, Gaël Dias, and Sonia Dollfus. 2024. Analyzing symptom-based depression level estimation through the prism of psychiatric expertise. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 974–983.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631.

Michael S Albergo and Eric Vanden-Eijnden. 2022. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*.

Tiago A Almeida, José María G Hidalgo, and Akebo Yamakami. 2011. Contributions to the study of sms spam filtering: new collection and results. In *Proceedings of the 11th ACM symposium on Document engineering*, pages 259–262.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei

Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Karima Boutalbi, Faiza Loukil, Hervé Verjus, David Telisson, and Kavé Salamatian. 2023. Machine learning for text anomaly detection: A systematic review. In *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 1319–1324. IEEE.

Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: Identifying density-based local outliers. In *ACM SIGMOD*, pages 93–104.

Yang Cao, Sikun Yang, Chen Li, Haolong Xiang, Lianyong Qi, Bo Liu, Rongsheng Li, and Ming Liu. 2025. Tad-bench: A comprehensive benchmark for embedding-based text anomaly detection. *arXiv preprint arXiv:2501.11960*.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):1–58.

Enyan Dai and Jie Chen. 2022. Graph-augmented normalizing flows for anomaly detection of multiple time series. In *ICLR*.

Anindya S. Das, A. Ajay, Sriparna Saha, and Monowar Bhuyan. 2023. Few-shot anomaly detection in text with deviation learning. In *ICONIP*, volume 14448 of *LNCS*, pages 425–437.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Manqing Dong, Hao Huang, and Longbing Cao. 2024. Can llms serve as time series anomaly detectors? *arXiv preprint arXiv:2408.03475*.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, and 1 others. 2024. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. 2024. Discrete flow matching. *Advances in Neural Information Processing Systems*, 37:133345–133385.

Jonathan Gratch, Ron Artstein, Gale M Lucas, Giota Stratou, Stefan Scherer, Angela Nazarian, Rachel Wood, Jill Boberg, David DeVault, Stacy Marsella, and 1 others. 2014. The distress analysis interview corpus of human and computer interviews. In *Lrec*, volume 14, pages 3123–3128. Reykjavik.

Zhaopeng Gu, Bingke Zhu, Guibo Zhu, Yingying Chen, Ming Tang, and Jinqiao Wang. 2024. Anomalygpt: Detecting industrial anomalies using large vision-language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 1932–1940.

Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. 2022. Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 98–107.

Yongping He, Tijin Yan, Yufeng Zhan, Zihang Feng, and Yuanqing Xia. 2024. Sgfm: Conditional flow matching for time series anomaly detection with state space models. *IEEE Internet of Things Journal*, 11(22):36979–36990.

Geoffrey E. Hinton and Ruslan R. Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

Vincent Hu, Di Wu, Yuki Asano, Pascal Mettes, Basura Fernando, Björn Ommer, and Cees Snoek. 2024. Flow matching for conditional text generation in a few sampling steps. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 380–392.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, and 1 others. 2024. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, and 1 others. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Diederik P. Kingma and Prafulla Dhariwal. 2018. Glow: Generative flow with invertible 1×1 convolutions. In *NeurIPS*.

Bryan Klimt and Yiming Yang. 2004. The enron corpus: A new dataset for email classification research. In *European conference on machine learning*, pages 217–226. Springer.

Chieh-Hsin Lai, Dongmian Zou, and Gilad Lerman. 2020. Robust subspace recovery layer for unsupervised anomaly detection. In *International Conference on Learning Representations (ICLR)*.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Machine learning proceedings 1995*, pages 331–339. Elsevier.

David Lewis. 1997. Reuters-21578 text categorization test collection. *Distribution 1.0, AT&T Labs-Research*.

Aodong Li, Yunhan Zhao, Chen Qiu, Marius Kloft, Padhraic Smyth, Maja Rudolph, and Stephan Mandt. 2024a. Anomaly detection of tabular data using llms. *arXiv preprint arXiv:2406.16308*.

Liangwei Li, Lin Liu, Hanzhe Liang, Juanxiu Liu, Jing Zhang, Ruqian Hao, Xiaohui Du, Yong Liu, and Pan Li. 2025a. Time-reversed flow matching with worst transport in high-dimensional latent space for image anomaly detection. *arXiv:2508.05461*.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. 2022. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35:4328–4343.

Yuangang Li, Jiaqi Li, Zhuo Xiao, Tiankai Yang, Yi Nian, Xiyang Hu, and Yue Zhao. 2024b. Nlp-adbench: Nlp anomaly detection benchmark. *arXiv preprint arXiv:2412.04784*.

Zhong Li, Jiayang Shi, and Matthijs van Leeuwen. 2025b. Scalable, explainable and provably robust anomaly detection with one-step flow matching. In *NeurIPS*.

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

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation forest. In *IEEE ICDM*, pages 413–422.

Xingchao Liu, Chengyue Gong, and Qiang Liu. 2022. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv:2209.03003*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Andrei Manolache, Florin Brad, and Elena Burceanu. 2021. DATE: Detecting anomalies in text via self-supervision of transformers. In *NAACL-HLT*, pages 267–277.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. 2018. Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136*.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. 2019. Do deep generative models know what they don't know? In *ICLR*.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the association for computational linguistics: ACL 2022*, pages 1864–1874.

David Novoa-Paradela, Oscar Fontenla-Romero, and Bertha Guijarro-Berdinas. 2024. Explained anomaly detection in text reviews: Can subjective scenarios be correctly evaluated? *Engineering Applications of Artificial Intelligence*, 133:108065.

Guansong Pang, Chunhua Shen, Longbing Cao, and Anton van den Hengel. 2021. Deep learning for anomaly detection: A review. *ACM Computing Surveys*, 54(2):1–38.

L. Pantin and C. Marsala. 2024. A robust autoencoder ensemble-based approach for anomaly detection in text. In *COLING*.

William Peebles and Saining Xie. 2023. Scalable diffusion models with transformers. In *ICCV*, pages 4195–4205.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *EMNLP*, pages 3982–3992.

Danilo Jimenez Rezende and Shakir Mohamed. 2015. Variational inference with normalizing flows. In *ICML*, pages 1530–1538.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Lukas Ruff, Jacob R. Kauffmann, Robert A. Vandermeulen, and 1 others. 2021. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 109(5):756–795.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep one-class classification. In *ICML*, pages 4393–4402.

Lukas Ruff, Yury Zemlyanskiy, Robert Vandermeulen, Thomas Schnake, and Marius Kloft. 2019. Self-attentive, multi-context one-class classification for unsupervised anomaly detection on text. In *ACL*, pages 4061–4071.

Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. 2001. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471.

Marcin Sendera and 1 others. 2021. Flow-based SVDD for anomaly detection. *arXiv:2108.04907*.

Marco Siino. 2024. All-mpnet at semeval-2024 task 1: Application of mpnet for evaluating semantic textual relatedness. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 379–384.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33:16857–16867.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024a. Improving text embeddings with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11897–11916.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, and 1 others. 2024b. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1369–1407.

Tiankai Yang, Yi Nian, Li Li, Ruiyao Xu, Yuangang Li, Jiaqi Li, Zhuo Xiao, Xiyang Hu, Ryan A Rossi, Kaize Ding, and 1 others. 2025. Ad-llm: Benchmarking large language models for anomaly detection. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1524–1547.

Chihiro Yano, Akihiko Fukuchi, Shoko Fukasawa, Hideyuki Tachibana, and Yotaro Watanabe. 2024. Multilingual sentence-t5: scalable sentence encoders for multilingual applications. In *Proceedings of the 2024 joint international conference on computational linguistics, language resources and evaluation (LREC-COLING 2024)*, pages 11849–11858.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Yixuan Zhou, Xing Xu, Jingkuan Song, Fumin Shen, and Heng Tao Shen. 2024. Msflow: Multiscale flow-based framework for unsupervised anomaly detection. *IEEE transactions on neural networks and learning systems*.

A Training Procedure

Algorithm 1 FLOCAT Training Step

Require: Inlier batch $\{\mathbf{z}^i\}_{i=1}^{B}$ (sentence or token embeddings), centroid $\boldsymbol{\mu}$, variance σ^2, learnable radius R, hyperparameter λ

Ensure: Updated parameters θ and radius R

// Agnostic Interpolation

$\mathbf{x}_0^{\text{sent},i} \leftarrow \mathbf{z}^i$ **if** $S = 1$ **else** $\frac{1}{S}\sum_{k=1}^{S} \mathbf{z}^{i,k}$

$\mathbf{x}_1^i \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2\mathbf{I}), \quad t \sim \mathcal{U}[0,1]$

$\mathbf{x}_t^{\text{sent},i} \leftarrow (1-t)\,\mathbf{x}_0^{\text{sent},i} + t\,\mathbf{x}_1^i, \quad u^i \leftarrow \mathbf{x}_1^i - \mathbf{x}_0^{\text{sent},i}$ ▷ Eq. 4, 5

$\mathbf{x}_t^{i,k} \leftarrow \mathbf{x}_t^{\text{sent},i} + \boldsymbol{\delta}^{i,k}, \quad \boldsymbol{\delta}^{i,k} = \mathbf{z}^{i,k} - \mathbf{z}^i$ ▷ Eq. 7

// FLOCAT Forward Pass

$\mathbf{X}_t^i = (\mathbf{x}_t^{i,1}, \ldots, \mathbf{x}_t^{i,k}, \ldots, \mathbf{x}_t^{i,S})$

$v_t^{\text{sent,i}} \leftarrow v_\theta^{\text{sent}}(\mathbf{X}_t^i, t)$ ▷ Sentence velocities

// Training Losses

$\mathcal{L}_{\text{FM}} \leftarrow \left\| v_t^{\text{sent},i} - u^i \right\|^2$ ▷ Eq. 9

$\hat{\mathbf{x}}_1^i \leftarrow \mathbf{x}_0^{\text{sent},i} + v_\theta^{\text{sent}}(\mathbf{X}_t^i, t)$ ▷ One-step Euler, Eq. 10

$\mathcal{L}_{\text{LOVE}} \leftarrow R^2 + \frac{1}{B}\sum_{i=1}^{B} \max\Big(0, \left\|\hat{\mathbf{x}}_1^i - \boldsymbol{\mu}\right\|^2 - R^2\Big)$ ▷ Eq. 11

// Update

Update θ, R by gradient descent on $\mathcal{L}_{\text{FM}} + \lambda\,\mathcal{L}_{\text{LOVE}}$ ▷ Eq. 12

B Dataset Categories

Table 6 details the category structure of each dataset used in our experiments. For each dataset, one category is designated as the inlier class during training, while all remaining categories constitute the anomaly pool from which 10% is sampled to form the test set. For binary datasets (SMSSpam, Enron), the inlier class is fixed by convention; for multi-class datasets, we follow the category assignments established in prior work (Ruff et al., 2019; Manolache et al., 2021).

All datasets are loaded from HuggingFace and we use the predefined splits without modification. Regarding the M4 dataset, we follow the construction methodology of the original paper (Wang et al., 2024b), focusing on the five English domains (Wikipedia, WikiHow, Reddit ELI5, arXiv, and PeerRead). For each domain, the machine-generated text consists of an ensemble of outputs from six different LLMs: Davinci003, GPT-4, Cohere, Dolly-v2, BLOOMz, and ChatGPT.

Dataset	Categories	size
20Newsgroups	computer, recreation, science, miscellaneous, politics, religion	11 004
Reuters	earn, trade, acq, money-fx, crude, ship, interest	9 603
AGNews	World, Sports, Business, Science/Technology	120 000
DBpedia14	Company, Educational Institution, Artist, Athlete, Office Holder, Mean of Transportation, Building, Natural Place, Village, Animal, Plant, Album, Film, Written Work	560 000
SMSSpam	ham (inlier), spam (anomaly)	4 459
Enron	ham (inlier), spam (anomaly)	31 716
IMDB	positive, negative	25 000
SST-2	positive, negative	67 349
M4	wikipedia, arxiv, wikihow, reddit, peerread	87 178
DAIC-WoZ	non-depressed (inlier), depressed (anomaly)	187

Table 6: Category structure and training set size of each dataset. Each experiment designates one category as the inlier class; remaining categories form the anomaly pool.

C Implementation Details

Our Method

All experiments are implemented in PyTorch (2.0.0) using the Adam optimiser (Kingma and Ba, 2014) with a learning rate of $\eta = 10^{-3}$ and weight decay 10^{-5}. The learning rate follows a warmup-cosine schedule (Loshchilov and Hutter,

2016; Devlin et al., 2019): it increases linearly during the first E_{warm} epochs to allow the model to reach a stable region of the parameter space before committing to large updates, then decays following a cosine annealing for smooth convergence toward a minimum.

All experiments are run on a single NVIDIA A100 80GB GPU. Hyperparameters were selected via Optuna (Akiba et al., 2019) on a held-out validation set. We use two configurations depending on the dataset group, summarised in Table 7. Text classification and spam datasets use a lighter architecture, while sentiment analysis and application datasets (M4, DAIC-WoZ) use a deeper configuration to handle the increased semantic complexity. For token-level inputs, we set a maximum sequence length of $S_{\max} = 128$ tokens for text classification and spam datasets, and $S_{\max} = 256$ tokens for sentiment analysis and application datasets; sentence-level inputs are not subject to this constraint as they consist of a single embedding vector.

Hyperparameter	Classification & Spam	Sentiment & Application
Hidden dim h	128	256
Depth L	4	8
Attention heads (DiT)	4	8
Attention heads (Velocity Pooling)	2	2
Learning rate η	10^{-3}	10^{-3}
Weight decay	10^{-5}	10^{-5}
λ ($\mathcal{L}_{\text{LOVE}}$)	10^{-3}	10^{-3}
Epochs E	300	300
Warmup epochs E_{warm}	120	120
Batch size	128	256
Max sequence length $S_{\max}$	128	256
Variance Ratio α_c	0.5	0.5

Table 7: Hyperparameters of our model per dataset group.

The radius R (Eq. 11) is initialised to 0.5, a scale commensurate with the expected spread of layer-normalised transformer embeddings, and optimised jointly with θ. The centroid $\boldsymbol{\mu}$ (Eq. 2) is computed once on the full training set before training begins and kept fixed thereafter. At inference, the flow map $\phi(\mathbf{x}_0^{\text{sent}})$ (Eq. 13) is approximated using the Euler method with T integration steps; a study on the effect of T is provided in Appendix F.

Baselines

For all baselines, we use the hyperparameters reported in the original papers, summarised below.

CVDD (Ruff et al., 2019) uses a self-attention layer with $d_a = 150$ and $r = 10$ attention heads. Optimisation is performed with Adam (Kingma and Ba, 2014), batch size 64, with a two-phase learning rate schedule: $\eta = 0.01$ for the first 40 epochs, followed by $\eta = 0.001$ for 60 additional epochs. Context weights are set with logarithmic annealing ($\alpha = 0$).

DATE (Manolache et al., 2021) sets the RTD loss weight to $\lambda_{\text{RTD}} = 50$ and the RMD loss weight to $\mu_{\text{RMD}} = 100$.

FATE (Das et al., 2023) uses the Adam optimiser with $\eta = 10^{-6}$, batch size 16, self-attention dimension $r_a = 150$, and $m = 5$ deviation reference samples.

TCCM (Li et al., 2025b) uses a time embedding of dimension 128, concatenated with the input vector and passed through an MLP to predict the flow vector. Optimisation uses Adam with $\eta = 5 \times 10^{-3}$.

RSRAE (Lai et al., 2020) is optimised with Adam, $\eta = 2.5 \times 10^{-4}$, for 200 epochs with batch size 128.

D Computational Efficiency

We compare FLOCAT against all baselines along three dimensions: number of trainable parameters, inference time, and training time. These comparisons are reported in Tables 8, 9, and 10 respectively.

Parameter count. Table 8 shows that FLOCAT occupies a middle ground: it is heavier than TCCM and RSRAE, which use simple MLP or attention-based architectures on fixed-size inputs, but significantly lighter than DATE and FATE. This difference is explained by the additional requirements of processing variable-length text sequences at both sentence and token granularities simultaneously, which necessitate self-attention across the sequence (Eq. 3, 9), adaLN-conditioned velocity prediction, and velocity pooling (Eq. 8).

Model	Trainable parameters
CVDD	0.05M
RSRAE	0.07M
TCCM	0.56M
FLOCAT	**1.53M**
DATE	7.31M
FATE	110.12M

Table 8: Number of trainable parameters per method.

Inference time. Table 9 shows that FLOCAT operates in the same order of magnitude as all

lightweight baselines at inference. LLM-based methods are 3 to 4 orders of magnitude slower due to sequential per-document prompting with generation.

Dataset	TCCM	RSRAE	FLOCAT	DATE	FATE	CVDD	Qwen-7B	Mistral-7B
SMS Spam	0.04	0.04	0.05	0.05	0.06	0.06	1161.65	1676.20
Reuters	0.08	0.08	0.09	0.10	0.11	0.11	415.33	692.85
20Newsgroups	0.10	0.11	0.12	0.13	0.14	0.15	2484.99	3560.06
IMDB	0.25	0.26	0.28	0.30	0.33	0.35	3236.53	4317.35
Enron	0.30	0.32	0.34	0.36	0.39	0.42	1384.04	2090.29
SST-2	0.55	0.57	0.60	0.64	0.68	0.72	1560.76	2203.86
AGNews	1.10	1.15	1.20	1.28	1.35	1.45	2141.15	3367.73
DBpedia14	4.80	4.95	5.15	5.40	5.65	5.90	4200.00	5800.00

Table 9: Inference time (seconds, full test set).

Training time. Table 10 shows that FLOCAT trains slower than TCCM and RSRAE, but faster than DATE and FATE, and comparably to CVDD on several datasets. TCCM and RSRAE are faster precisely because they are restricted to a single granularity and do not produce token-level attribution signals. The DiT backbone choice is thus primarily motivated by the scientific challenges specific to anomaly detection on textual data, rather than by performance considerations alone.

Dataset	TCCM	RSRAE	CVDD	FLOCAT	DATE	FATE
20Newsgroups	15.20	42.78	115.40	339.69	452.10	662.13
AGNews	28.50	53.43	165.20	496.73	692.67	836.39
Enron	12.40	42.04	142.10	356.82	517.44	1509.49
IMDB	21.10	42.03	125.80	327.21	486.12	1201.68
Reuters	8.90	40.04	54.30	65.50	92.84	119.73
SMS Spam	11.30	41.27	215.60	684.11	857.12	1319.59
SST-2	32.60	47.94	280.50	905.15	1212.37	2769.15
DBpedia14	115.00	411.77	465.30	520.00	910.23	1523.72

Table 10: Training time (seconds).

E Robustness to Training Set Contamination

In unsupervised anomaly detection, training sets are rarely perfectly clean in practice. We evaluate the robustness of FLOCAT by injecting a fraction $\mu \in \{5\%, 10\%, 20\%\}$ of anomalies into the training set and measuring AUC-ROC degradation relative to the clean setting ($\mu = 0\%$). We compare against CVDD and TCCM as representative baselines, across two encoders (RoBERTa and MPNet) and three datasets (Reuters, 20Newsgroups, AGNews).

Table 11 reports AUC-ROC and the absolute drop Δ at each contamination level. FLOCAT consistently shows the smallest degradation across all settings, confirming that the learned transport is less sensitive to the presence of outliers in the training set.

We attribute this robustness to the $\mathcal{L}_{\text{LOVE}}$ constraint (Eq. 11): by enforcing a compact hypersphere around the inlier centroid $\boldsymbol{\mu}$ with a learnable radius R, the hinge loss ℓ_i penalises only endpoints that fall outside the sphere. Contaminating samples, whose transported representations deviate from the inlier cluster, therefore exert limited influence on the learned transport field.

			$\mu = 0\%$	$\mu = 5\%$		$\mu = 10\%$		$\mu = 20\%$	
Dataset	**Emb.**	**Model**	AUC	AUC	Δ	AUC	Δ	AUC	Δ
Reuters	*RoBERTa*	CVDD	$84.90_{\pm 0.3}$	$77.83_{\pm 0.5}$	-7.07	$72.26_{\pm 0.6}$	-12.64	$68.24_{\pm 0.8}$	-16.66
		FLOCAT	$94.83_{\pm 0.2}$	$88.12_{\pm 0.4}$	**-6.71**	$84.55_{\pm 0.4}$	**-10.28**	$79.31_{\pm 0.5}$	**-15.52**
	MPNet	TCCM	$98.28_{\pm 0.1}$	$91.03_{\pm 0.3}$	-7.25	$88.59_{\pm 0.3}$	-9.69	$81.23_{\pm 0.5}$	-17.05
		FLOCAT	$\mathbf{98.50_{\pm 0.1}}$	$\mathbf{92.27_{\pm 0.2}}$	**-6.23**	$\mathbf{89.11_{\pm 0.3}}$	**-9.39**	$\mathbf{84.35_{\pm 0.4}}$	**-14.15**
20Newsgroups	*RoBERTa*	CVDD	$69.56_{\pm 0.8}$	$65.28_{\pm 0.9}$	-4.28	$61.92_{\pm 1.1}$	-7.64	$60.17_{\pm 1.2}$	-9.39
		FLOCAT	$76.82_{\pm 0.6}$	$74.15_{\pm 0.7}$	**-2.67**	$71.04_{\pm 0.9}$	**-5.78**	$68.41_{\pm 1.0}$	**-8.41**
	MPNet	TCCM	$93.45_{\pm 0.3}$	$91.83_{\pm 0.4}$	-1.62	$89.15_{\pm 0.5}$	-4.30	$84.64_{\pm 0.6}$	-8.81
		FLOCAT	$\mathbf{93.89_{\pm 0.2}}$	$\mathbf{92.32_{\pm 0.3}}$	**-1.57**	$\mathbf{90.26_{\pm 0.4}}$	**-3.63**	$\mathbf{86.27_{\pm 0.5}}$	**-7.62**
AGNews	*RoBERTa*	CVDD	$73.70_{\pm 0.5}$	$70.44_{\pm 0.6}$	-3.26	$68.05_{\pm 0.7}$	-5.65	$66.16_{\pm 0.9}$	-7.54
		FLOCAT	$90.54_{\pm 0.3}$	$88.41_{\pm 0.4}$	**-2.13**	$87.12_{\pm 0.4}$	**-3.42**	$84.06_{\pm 0.6}$	**-6.48**
	MPNet	TCCM	$94.03_{\pm 0.2}$	$92.25_{\pm 0.3}$	-1.78	$91.25_{\pm 0.3}$	-2.78	$86.55_{\pm 0.5}$	-7.48
		FLOCAT	$\mathbf{94.56_{\pm 0.2}}$	$\mathbf{93.75_{\pm 0.2}}$	**-0.81**	$\mathbf{91.93_{\pm 0.3}}$	**-2.63**	$\mathbf{88.81_{\pm 0.4}}$	**-5.75**

Table 11: AUC-ROC (%) and absolute drop Δ under training set contamination ($\mu \in \{0, 5, 10, 20\}\%$). **Bold**: smallest $|\Delta|$ at each contamination level.

F Number of Euler Integration Steps

The Euler method is used at two distinct stages of FLOCAT, each with a different role and therefore a different requirement on the number of steps T.

Inference steps T_{inf}. At inference, the flow $\phi(\mathbf{x}_0^{\text{sent}})$ (Eq. 13) must accurately transport a test sample to compute the anomaly score $s(x^i)$. Figure 7 reports AUC-ROC and inference time per sample as a function of T on 20Newsgroups (RoBERTa embeddings), averaged across all six inlier configurations. AUC-ROC improves from $T = 1$ to $T = 5$, then plateaus. Inference time grows approximately linearly. We set $T_{\text{inf}} = 5$ in all experiments as the optimal trade-off.

Training steps T_{train}. At training, the one-step Euler estimate (Eq. 10) approximates $\hat{\mathbf{x}}_1^i$ solely to evaluate the $\mathcal{L}_{\text{LOVE}}$ constraint. This estimate does not need to be precise: it only needs to place the endpoint sufficiently close to the target space for the compactness constraint to operate. Table 12 reports AUC-ROC and training time for $T_{\text{train}} \in \{1, 3, 5\}$, with $T_{\text{inf}} = 5$ fixed.

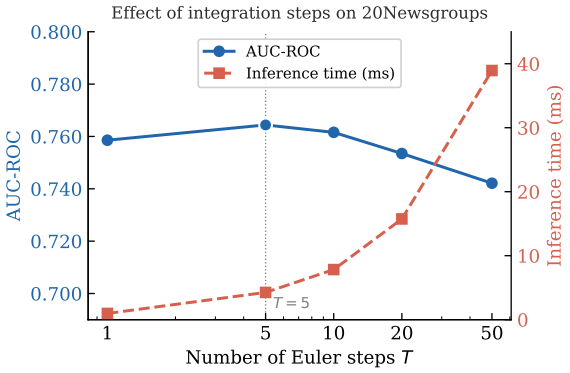


Figure 7: AUC-ROC and inference time (ms per sample) as a function of the number of Euler steps T on 20Newsgroups. Performance peaks at $T = 5$ and remains stable up to $T = 10$.

Dataset	Metric	$T_{\text{train}} = 1$	$T_{\text{train}} = 3$	$T_{\text{train}} = 5$
Reuters	AUC	**99.12**±0.05	99.09±0.06	**99.12**±0.05
	Training time (s)	**25.46**±1.12	41.59±1.45	59.14±1.88
20Newsgroups	AUC	**92.75**±0.15	92.67±0.18	92.76±0.16
	Training time (s)	**51.43**±2.10	83.93±2.45	125.30±3.12
Enron	AUC	95.65±0.10	95.62±0.12	**95.67**±0.11
	Training time (s)	**154.53**±3.50	251.88±4.20	370.87±5.15
IMDB	AUC	56.09±1.25	56.05±1.30	**56.11**±1.28
	Training time (s)	**176.01**±4.10	286.90±5.30	422.42±6.10
SST-2	AUC	59.91±0.95	59.88±0.98	**59.93**±0.96
	Training time (s)	**620.40**±8.20	1011.25±12.5	1488.96±15.8
SMS Spam	AUC	66.30±0.75	66.27±0.78	**66.32**±0.76
	Training time (s)	**141.18**±3.10	230.12±4.25	338.83±5.40
AG News	AUC	94.56±0.20	94.52±0.22	**94.58**±0.21
	Training time (s)	**451.05**±6.50	735.21±9.10	1082.52±12.4
DBPedia	AUC	98.83±0.08	98.80±0.09	**98.85**±0.08
	Training time (s)	**380.00**±5.40	619.40±8.20	912.00±11.5

Table 12: AUC-ROC (%) and training time (s) as a function of T_{train}, with $T_{\text{inf}} = 5$ fixed. **Bold**: best AUC or lowest training time.

AUC-ROC remains virtually identical across all values of T_{train}, while training time increases substantially. A single Euler step is therefore sufficient for the $\mathcal{L}_{\text{LOVE}}$ constraint to operate correctly, with no cost to final performance.

G Sensitivity Analysis: Variance Ratio α_c

The variance ratio α_c controls the tightness of the Gaussian target $p_1 = \mathcal{N}(\boldsymbol{\mu}, \sigma^2\mathbf{I})$ relative to the source distribution spread (Eq. 2). Table 13 reports AUC-ROC for $\alpha_c \in \{0.1, 0.5, 1.0\}$, with $\lambda = 10^{-3}$ fixed.

Dataset	$\alpha_c = 0.1$	$\alpha_c = 0.5$ (ours)	$\alpha_c = 1.0$
Reuters	99.01±0.08	**99.19**±0.05	98.70±0.12
20Newsgroups	92.86±0.18	**93.19**±0.14	92.76±0.20
AG News	94.23±0.22	**94.56**±0.18	94.11±0.25
DBPedia	98.50±0.09	**98.83**±0.07	98.38±0.11
SMS Spam	**66.97**±0.82	66.10±0.70	65.85±0.88
Enron	95.32±0.15	**95.65**±0.12	95.20±0.18
SST-2	59.58±0.98	**59.91**±0.85	59.46±1.05
IMDB	55.76±1.40	**56.09**±1.25	55.64±1.45

Table 13: AUC-ROC (%) for varying α_c, with $\lambda = 10^{-3}$ fixed. **Bold**: best.

Setting $\alpha_c = 1.0$ yields the worst results: the target distribution is as spread as the source, providing no compactness gain. Conversely, very low values of α_c reduce the target variance excessively, removing the dispersion necessary for $\mathcal{L}_{\text{LOVE}}$ to converge properly. The value $\alpha_c = 0.5$ achieves the best trade-off on the large majority of datasets.

H Sensitivity Analysis: Trade-off Coefficient λ

The coefficient λ controls the relative weight of $\mathcal{L}_{\text{LOVE}}$ with respect to $\mathcal{L}_{\text{FM}}$ in the total objective $\mathcal{L}_{\text{FLOCAT}}$ (Eq. 12). Table 14 reports AUC-ROC for $\lambda \in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$, with $\alpha_c = 0.5$ fixed.

Dataset	$\lambda = 10^{-1}$	$\lambda = 10^{-2}$	$\lambda = 10^{-3}$ (ours)	$\lambda = 10^{-4}$
Reuters	98.75±0.12	99.38±0.07	**99.52**±0.04	99.19±0.08
20Newsgroups	91.87±0.25	93.15±0.18	**93.31**±0.15	92.56±0.22
AG News	93.12±0.30	94.40±0.22	**94.56**±0.19	93.81±0.28
DBPedia	98.39±0.15	97.67±0.10	**98.83**±0.08	98.08±0.14
SMS Spam	64.86±0.95	66.14±0.82	**66.30**±0.75	65.55±0.90
Enron	94.21±0.22	95.49±0.15	**95.65**±0.12	94.90±0.20
SST-2	58.97±1.15	58.75±1.02	**59.91**±0.90	59.16±1.10
IMDB	54.65±1.55	55.93±1.35	**56.09**±1.28	55.34±1.48

Table 14: AUC-ROC (%) for varying λ, with $\alpha_c = 0.5$ fixed. **Bold**: best.

At $\lambda = 10^{-1}$, $\mathcal{L}_{\text{LOVE}}$ dominates the objective and forces the target space too aggressively, preventing $\mathcal{L}_{\text{FM}}$ from converging properly. At $\lambda = 10^{-4}$, the compactness constraint is too weak to shape the endpoint geometry meaningfully. The value $\lambda = 10^{-3}$ achieves the best balance between transport quality and compactness regularisation across all datasets.

I Per-Token Residual Injection

Figure 8 illustrates the per-token residual injection mechanism described in Section 3.3. Each token embedding $\mathbf{z}^{i,k}$ is related to the sentence barycentre $\mathbf{z}^i$ by a vector $\boldsymbol{\delta}^{i,k} = \mathbf{z}^{i,k} - \mathbf{z}^i$ that encodes both the direction and magnitude of the token's deviation from the sentence mean. At interpolation time t, this residual is added back to the sentence-level interpolated state $\mathbf{x}_t^{\text{sent,i}}$ to reconstruct a token-level input to the DiT, preserving token-specific structure while keeping the interpolation trajectory defined at the sentence level.

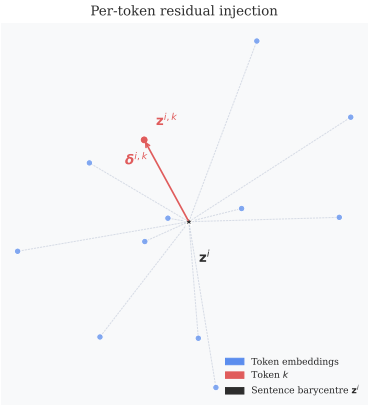


Figure 8: Per-token residual injection. Each token embedding (blue) is offset from the sentence barycentre $\mathbf{z}^i$ (star) by a vector $\boldsymbol{\delta}^{i,k}$ encoding both direction and magnitude. This vector is added to the interpolated sentence state $\mathbf{x}_t^{\text{sent}}$ to construct the token-level DiT input.

J Robustness to Centroid Choice

A natural question arising from our formulation is whether the choice of the target centroid $\boldsymbol{\mu}$ affects the quality of the learned transport. In Section 3.1, we set $\boldsymbol{\mu}$ to the barycentre of the source distribution for scale consistency; however, the flow matching framework is in principle agnostic to the absolute position of the target, as the vector field v_θ is learned to bridge whatever source and target distributions are provided.

To verify this empirically, we train FLOCAT with a fixed target centroid set to $\mathbf{6} \in \mathbb{R}^d$, a point far from the source distribution, and visualise the resulting transport trajectories.

As shown in Figure 9, the model successfully learns to transport inlier embeddings toward this alternative centroid, producing trajectories that are qualitatively identical to those obtained with our default choice of $\boldsymbol{\mu}$. This confirms that the learned flow adapts to any target position, and that our choice of the barycentre is a principled but not critical design decision: it ensures scale consistency and numerical stability without constraining the expressiveness of the transport.

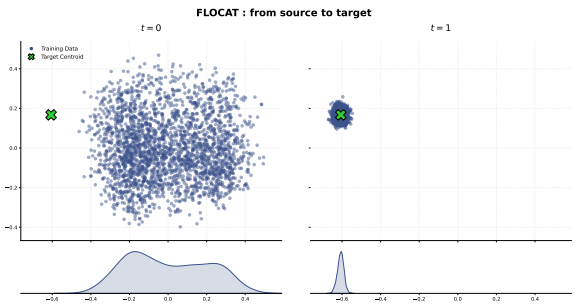


Figure 9: Illustration of centroid-agnostic transport. At $t = 0$, the target centroid $\boldsymbol{\mu}$ is set to a fixed point far from the source distribution, intentionally distant from its barycentre. At $t = 1$, the learned flow has successfully transported all inlier representations toward $\boldsymbol{\mu}$, forming a low-variance cluster around it despite the large initial displacement.

K Effect of the Low-Variance Loss

Table 15 reports the AUC-ROC of FLOCAT with and without the low-variance loss $\mathcal{L}_{\text{love}}$ across all datasets and encoder backbones. Removing $\mathcal{L}_{\text{love}}$ degrades performance, confirming that the low-variance constraint on the target space is a necessary component of the approach.

Emb.	Method	Reuters	20News	AGNews	DBpedia	SMSSpam	Enron	SST-2	IMDB	Avg.
RoBERTa	FLOCAT	**94.83**±3.3	**76.82**±4.2	**90.54**±0.8	**96.78**±0.2	**76.68**±1.3	87.66±1.4	**61.49**±1.3	**60.19**±0.4	**80.62**
	FLOCAT w/o $\mathcal{L}_{\text{love}}$	88.23±4.7	76.06±1.9	86.53±3.1	96.79±0.5	74.20±0.4	**90.20**±1.1	56.66±0.7	52.28±3.7	77.62
Qwen	FLOCAT	**90.86**±4.0	**76.30**±2.9	**85.77**±1.4	**91.54**±0.4	**72.58**±0.1	**92.38**±0.0	**54.10**±1.4	**59.43**±1.9	**77.87**
	FLOCAT w/o $\mathcal{L}_{\text{love}}$	90.02±5.4	74.25±2.0	83.95±4.2	90.22±0.3	71.82±1.3	90.91±0.8	54.05±0.1	51.64±0.2	75.86
MPNet	FLOCAT	**98.50**±1.5	**93.89**±0.9	**94.56**±0.3	**98.83**±0.1	**66.30**±0.7	**95.65**±0.5	**59.91**±1.6	**56.09**±0.5	**82.97**
	FLOCAT w/o $\mathcal{L}_{\text{love}}$	96.70±2.3	88.17±1.0	93.49±0.5	97.12±0.1	64.34±1.1	93.68±1.0	58.30±1.8	55.86±0.3	80.96
E5-Mistral	FLOCAT	**97.11**±1.8	**89.79**±1.8	**93.84**±0.4	**99.50**±0.1	65.37±12.1	**95.01**±1.1	**61.16**±1.4	**68.64**±1.9	**83.80**
	FLOCAT w/o $\mathcal{L}_{\text{love}}$	94.29±2.6	89.52±1.1	91.77±0.6	96.00±0.2	**65.55**±1.2	93.99±1.2	64.20±1.0	60.99±0.7	82.04
ST5	FLOCAT	**96.69**±2.8	**88.73**±2.2	**91.90**±0.5	**98.99**±0.1	**72.25**±2.4	**94.10**±0.6	**65.93**±2.8	**63.97**±1.4	**84.07**
	FLOCAT w/o $\mathcal{L}_{\text{love}}$	95.14±1.4	87.96±0.8	90.40±0.4	96.04±0.1	63.95±0.2	92.22±0.7	58.82±1.5	54.98±0.5	80.19

Table 15: Ablation of $\mathcal{L}_{\text{love}}$. AUC-ROC (%) averaged over 5 runs.

L Anomaly Score as Negative Log-Likelihood

We show that the anomaly score $s(x^i) = \|\phi(\mathbf{x}_0^{\text{sent}}) - \boldsymbol{\mu}\|^2$ is equivalent to the negative log-likelihood (NLL) of the transported point under the target distribution p_1, up to constants independent of the input.

For a general Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the NLL of a point $\mathbf{x}$ is:

$$-\log p(\mathbf{x}) = \frac{1}{2}\log\det(2\pi\boldsymbol{\Sigma}) + \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) \tag{18}$$

Since $p_1 = \mathcal{N}(\boldsymbol{\mu}, \sigma^2\mathbf{I})$ is isotropic, $\boldsymbol{\Sigma}^{-1} = \frac{1}{\sigma^2}\mathbf{I}$ and $\log\det(2\pi\boldsymbol{\Sigma}) = d\log(2\pi\sigma^2)$, so Eq. 18 becomes:

$$-\log p_1(\mathbf{x}) = \underbrace{\frac{d}{2}\log(2\pi\sigma^2)}_{\text{const.}} + \frac{1}{2\sigma^2}\|\mathbf{x} - \boldsymbol{\mu}\|^2 \tag{19}$$

The first term is constant with respect to $\mathbf{x}$ and does not affect the ranking of anomaly scores. The second term is a positive scaling of the squared Euclidean distance, which also preserves the ranking. Therefore, minimising the NLL under p_1 is equivalent to minimising $\|\mathbf{x} - \boldsymbol{\mu}\|^2$, and the anomaly score $s(x^i) = \|\phi(\mathbf{x}_0^{\text{sent}}) - \boldsymbol{\mu}\|^2$ is the NLL of the transported point under p_1 up to additive and multiplicative constants.

M Token Attribution Score: Detailed Analysis

This appendix provides a detailed analysis of the per-token attribution score $\tilde{a}^{i,k}$ (Eq. 16) across Euler integration steps $t \in \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. For each example, we show two tokens: one whose score evolves toward high attribution at $t = 1$, and one whose score evolves toward neutrality, illustrating how the flow progressively assigns attribution as the transport converges. Shading intensity reflects the attribution: dark (highly attributed), medium (moderately attributed), transparent (neutral).

Human- vs. Machine-Generated Text

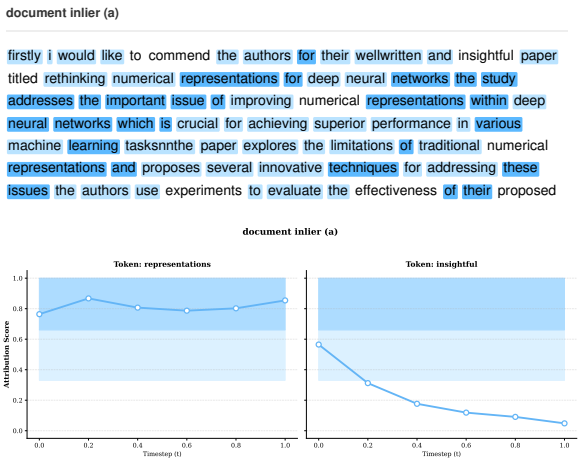


Figure 10: *Left*: the token *representations* increases until $t = 1$, where the model identifies it as a highly attributed token characteristic of machine-generated academic writing. *Right*: the token *insightful* begins as a highly attributed token at $t = 0$ but its score decreases over integration steps, converging to a neutral token at $t = 1$.

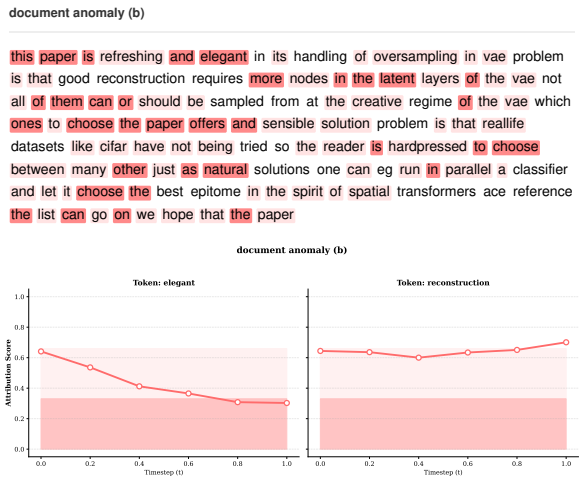


Figure 11: *Left*: The token *elegant* exhibits a decreasing score, converging to a low value at $t = 1$, confirming its role as a primary anomaly driver. Its low score reflects the subjective critical language characteristic of human reviewers. *Right*: the token *reconstruction* shows a slight score increase, converging toward a neutral attribution at $t = 1$, as the model recognises it as a common technical term shared across both human and machine writing.

Depression Screening (DAIC-WoZ)

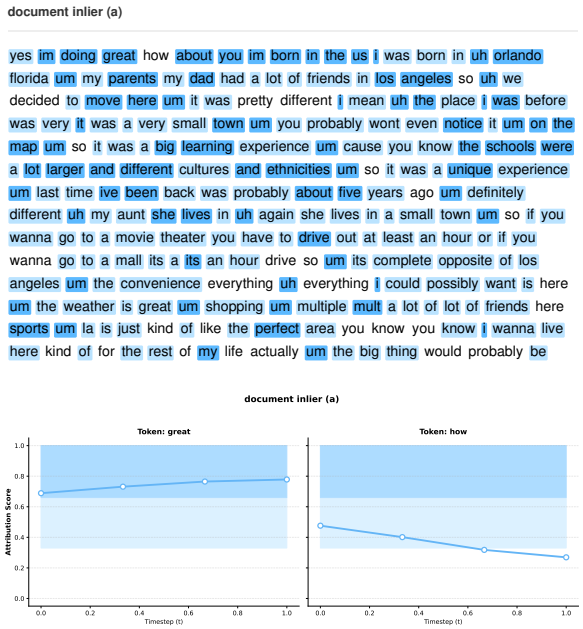


Figure 12: *Left*: the token *great* exhibits an increasing score, becoming highly attributed at $t = 1$ and reflecting the positive affect characteristic of non-depressed speech. *Right*: the token *how* begins as a highly attributed token at $t = 0$ but its score decreases over integration steps, reaching the neutral range at $t = 1$, as the model learns that this function word is not discriminative of normality.

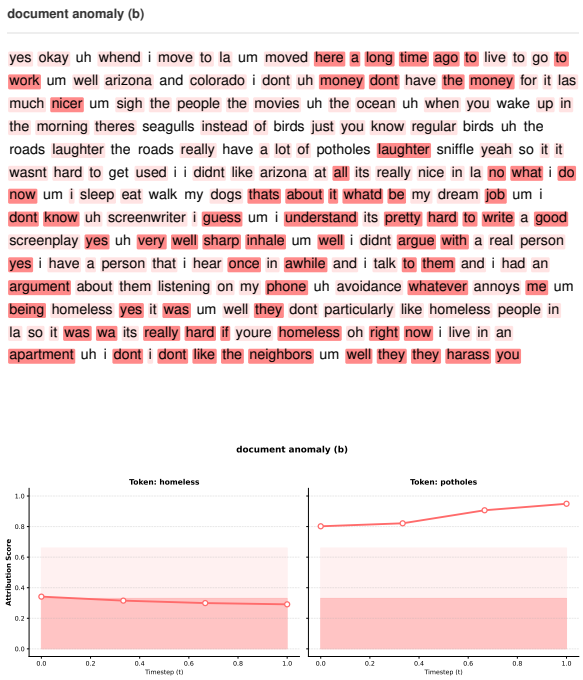
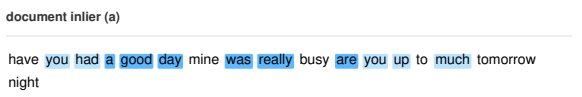


Figure 13: *Left*: the token *homeless* shows a decreasing score over integration steps, converging to a low value at $t = 1$ and confirming its role as a primary anomaly driver, directly reflecting the material hardship and social vulnerability characteristic of depressive speech. *Right*: The token *potholes*, initially neutral, shows an increasing score converging toward neutrality at $t = 1$, confirming that not all tokens in a depressed transcript contribute equally to the anomaly decision.

SMS Spam Detection

To further validate the token-level interpretability of FLOCAT, we present qualitative examples on the SMSSpam dataset, where ham messages constitute the inlier class and spam messages are anomalies. For each example, we show the highlighted message followed by the attribution score evolution across integration steps for two representative tokens.



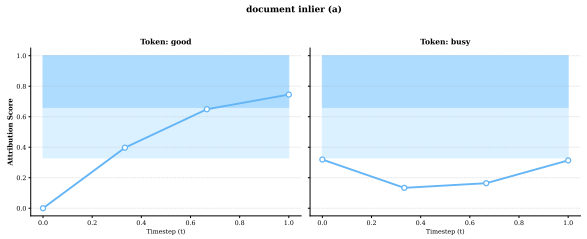


Figure 14: The token *good* starts near zero at $t = 0$ and its score rises to 0.7 at $t = 1$, showing that the model progressively identifies it as a primary marker of normal, non-spam communication. The second token *busy* converges toward a neutral attribution, confirming that the model discriminates between content-bearing and neutral tokens even within inlier messages.

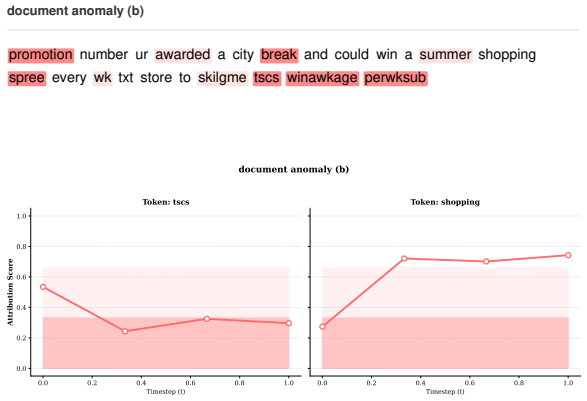


Figure 15: The token *tscs* exhibits a decreasing score, converging to a low value at $t = 1$ and flagged as a primary anomaly driver. Notably, the highlighted message also reveals two additional highly attributed tokens: *winawkage* and *perwksub*, which are not valid English words and are characteristic of obfuscation techniques commonly used in spam to bypass keyword-based filters. This demonstrates that FLOCAT captures anomaly signals at the lexical level without any explicit linguistic knowledge.

N LLM Zero-Shot Prompting Protocol

We follow the prompting protocol of Yang et al. (2025) to evaluate Mistral-7B-Instruct and Qwen2.5-7B-Instruct in few-shot anomaly detection (3 inlier examples). Both models are run with greedy decoding (temperature $= 0$) and a maximum output length of 256 tokens. For each test document, the model receives a structured prompt describing the normal category, optional in-context examples, and the text to score. The model is instructed to respond exclusively in JSON format with two fields: a short reasoning chain and a continuous anomaly confidence score $\hat{s} \in [0, 1]$, which is used directly as the anomaly score for AUC-ROC computation. The full prompt template is below.

```
You are an intelligent and professional
assistant that detects anomalies in text data.

## Task:
Following the rules below, determine whether
the given text sample is an anomaly. Provide a
brief explanation of your reasoning and assign
an anomaly confidence score between 0 and 1.

## Normal Category:
{categorie}

[Normal Text Examples from the training set.]


## Rules:
1. Anomaly Definition: A text sample is an
anomaly if it does not belong to the normal
category listed above.
2. Scoring: Assign a score in [0,1]. Use higher
scores (close to 1) when highly confident the
text IS an anomaly; lower scores (close to 0)
when it belongs to the normal category.
3. Step-by-step Reasoning (Chain of Thought):
   Step 1. Read the text carefully.
   Step 2. Compare it to the normal category.
   Step 3. Determine alignment: if yes →
not an anomaly (score ≈ 0); if no → anomaly
(score ≈ 1).
   Step 4. Assign the anomaly confidence
score.
4. Additional Notes: If uncertain, assume it
is not an anomaly.
5. Response Format: Respond ONLY in valid
JSON with exactly two keys: "reason" (one to
three sentences) and "anomaly_score" (float in
[0,1]).
   Example: {"reason": "The text is about
sports.", "anomaly_score": 0.05}

Text sample: "{text}"

Response in JSON format:
```

Prompt template used for LLM few-shot baselines.

O Detailed Results per Dataset

Emb.	Method	mis.	rel.	pol.	rec.	sci.	com.	Avg.
RoBERTa	DATE	74.46±4.8	**81.54±3.1**	76.51±2.5	69.56±13.6	61.15±3.2	78.17±13.8	73.56
	CVDD	**75.98±3.8**	71.35±2.7	67.39±2.9	62.15±4.5	59.25±3.9	**81.23±1.5**	69.56
	RSRAE	61.46±3.9	78.63±2.0	74.96±1.7	71.04±2.0	65.22±2.3	77.67±0.8	71.50
	TCCM†	58.85±4.4	79.22±2.1	77.38±1.4	71.49±1.9	66.54±2.6	79.29±1.0	72.13
	FLOCAT	75.32±5.3	81.46±2.5	**82.44±1.6**	**76.54±1.9**	**69.16±4.4**	76.01±9.8	**76.82**
Qwen	DATE	–	–	–	–	–	–	–
	CVDD	87.59±7.9	74.90±14.3	63.43±11.3	76.19±11.3	**80.10±11.3**	73.12±1.7	75.89
	RSRAE	72.65±1.9	65.58±1.4	62.03±1.9	59.17±6.5	74.67±1.2	75.07±2.0	68.20
	TCCM†	80.83±1.5	74.20±2.8	60.89±1.7	71.40±2.7	72.47±3.7	**75.10±2.5**	72.48
	FLOCAT	**88.56±0.7**	**76.56±0.7**	**67.27±1.5**	**79.65±3.8**	70.71±3.4	75.06±7.3	**76.30**
MPNet	FATE	93.78±1.9	95.12±0.9	82.17±2.1	94.46±2.3	91.27±1.2	90.21±1.9	91.17
	RSRAE	95.65±0.6	94.82±0.7	81.41±1.8	97.02±0.6	**96.41±0.4**	91.52±1.6	92.81
	TCCM†	96.16±0.6	**95.87±0.7**	82.93±1.9	**97.25±0.5**	95.91±0.7	**92.56±1.4**	93.45
	FLOCAT	**96.35±0.5**	95.84±0.6	**85.18±1.4**	97.24±0.5	96.22±0.7	92.54±1.4	**93.89**
ST5	FATE	–	–	–	–	–	–	–
	RSRAE	90.17±1.1	80.74±0.5	74.34±1.1	89.70±1.7	89.09±0.8	83.85±2.6	84.65
	TCCM†	92.98±1.0	84.87±0.5	76.01±1.5	88.23±2.0	89.45±0.7	85.26±2.1	86.13
	FLOCAT	**93.33±3.7**	**87.50±1.1**	**79.76±1.4**	**93.10±1.0**	**89.89±3.8**	**88.81±2.4**	**88.73**
E5-Mistral	FATE	–	–	–	–	–	–	–
	RSRAE	94.08±1.1	89.81±0.7	78.73±0.2	91.02±1.1	91.12±0.9	88.72±1.6	88.91
	TCCM†	94.58±1.1	90.19±0.9	**79.25±0.6**	90.95±1.2	91.18±0.6	89.64±1.1	89.30
	FLOCAT	**95.32±0.9**	**91.22±1.0**	75.59±5.9	**93.27±0.8**	**92.49±0.6**	**90.84±1.4**	**89.79**
LLM-based	Qwen2.5-7B-Instruct	64.13±4.9	84.28±1.0	78.99±2.4	69.95±1.3	66.43±0.5	79.44±1.5	73.87
LLM-based	Mistral-7B-Instruct	63.57±1.8	70.17±1.2	58.12±0.7	69.48±0.7	64.51±3.2	77.99±1.0	67.31

Table 16: AUC-ROC on 20Newsgroups per category and embedding.

Emb.	Method	earn	trade	acq	money-fx	crude	ship	interest	Avg.
RoBERTa	DATE	94.66±4.3	84.42±15.6	88.13±6.9	91.06±6.5	81.12±12.0	73.54±12.8	87.65±8.5	85.80
	CVDD	74.44±4.4	88.31±9.4	96.11±1.2	88.28±12.8	93.90±4.0	81.25±8.5	72.00±13.1	84.90
	RSRAE	40.94±1.2	95.02±3.6	**96.38±0.9**	90.24±9.3	91.93±4.2	82.50±9.2	89.81±5.6	83.83
	TCCM†	48.28±1.8	79.98±8.5	95.21±1.0	76.16±10.4	80.62±8.9	81.15±14.9	68.31±9.8	75.67
	FLOCAT	**95.09±0.6**	**96.84±2.3**	96.10±1.2	**93.25±7.5**	**94.87±2.8**	**93.33±4.5**	**94.31±4.2**	**94.83**
Qwen	DATE	–	–	–	–	–	–	–	–
	CVDD	94.35±1.8	86.58±1.3	93.28±1.6	**92.86±17.1**	91.89±0.2	**90.75±23.1**	69.16±6.6	88.41
	RSRAE	65.00±3.2	88.83±7.9	81.08±2.0	72.22±10.2	82.22±7.2	84.38±11.2	59.92±22.0	76.24
	TCCM†	94.06±0.8	**89.09±9.0**	94.09±0.6	75.19±12.3	83.04±8.0	84.58±11.7	66.54±19.6	83.80
	FLOCAT	**96.02±0.8**	86.44±9.4	**96.80±0.4**	86.77±8.0	**91.22±1.2**	88.89±3.8	**89.87±4.4**	**90.86**
MPNet	FATE	92.96±0.5	91.43±5.3	93.89±1.3	88.97±10.2	86.19±6.5	89.56±7.2	89.28±5.5	90.33
	RSRAE	98.28±0.2	96.31±0.5	98.19±0.3	96.98±6.0	98.98±0.8	98.33±0.8	**99.69±0.3**	98.11
	TCCM†	99.20±0.3	**99.13±1.2**	97.62±0.2	96.03±7.7	**99.64±0.3**	97.29±3.1	99.08±1.5	98.28
	FLOCAT	**99.25±0.2**	98.87±1.1	98.54±0.3	**96.83±6.3**	99.27±0.5	**97.50±1.1**	99.23±0.8	**98.50**
ST5	FATE	–	–	–	–	–	–	–	–
	RSRAE	**97.49±2.6**	96.15±4.0	96.38±0.6	93.92±5.3	98.93±0.7	93.75±5.9	97.00±4.4	96.41
	TCCM†	96.15±4.0	96.15±4.0	**96.66±0.6**	93.07±7.0	98.44±1.4	93.33±6.4	96.46±5.2	96.04
	FLOCAT	96.49±3.2	**96.49±3.2**	95.65±0.8	**94.92±6.4**	**99.07±1.1**	**95.00±3.3**	**97.46±4.7**	**96.69**
E5-Mistral	FATE	–	–	–	–	–	–	–	–
	RSRAE	**99.48±0.2**	**99.46±0.6**	97.76±0.7	93.58±7.3	95.17±0.3	92.57±3.6	**97.21±2.0**	96.46
	TCCM†	99.45±0.3	98.65±1.6	**98.21±0.5**	92.59±8.5	**97.50±0.1**	85.68±10.5	93.65±2.3	95.10
	FLOCAT	**99.48±0.3**	98.92±1.4	97.97±0.9	**95.93±4.3**	96.67±1.1	**93.96±1.8**	96.83±2.6	**97.11**
LLM-based	Qwen2.5-7B-Instruct	91.11±0.5	90.62±5.7	75.66±0.5	82.05±11.6	88.94±0.2	67.01±5.7	71.22±17.0	80.94
LLM-based	Mistral-7B-Instruct	76.38±1.6	82.79±5.3	74.97±1.2	82.76±3.1	87.80±3.5	45.14±19.1	75.38±4.0	75.03

Table 17: AUC-ROC on Reuters per category and embedding.

Emb.	Method	wikipedia	arxiv	wikihow	reddit	peerread	Avg.
RoBERTa	DATE	50.57±9.2	57.37±7.9	51.20±2.4	47.55±2.4	88.42±2.9	59.02
	CVDD	47.15±0.6	40.57±1.5	48.51±1.2	56.09±1.1	79.35±1.4	54.33
	RSRAE	61.59±1.1	58.76±0.6	**63.20±0.4**	54.09±1.7	78.25±1.2	63.18
	TCCM†	61.60±0.3	60.94±0.7	61.88±0.9	57.43±1.7	84.04±2.2	65.18
	FLOCAT	**62.74±2.3**	**61.92±6.6**	62.66±0.6	**60.00±1.1**	**92.12±1.1**	**67.89**

Table 18: AUC-ROC on M4 per category (RoBERTa embeddings).

Emb.	Method	Company	Educ. Inst.	Artist	Athlete	Off. Holder	Mean. Transp.	Building	Nat. Place	Village	Animal	Plant	Album	Film	Written W.	Avg.
RoBERTa	DATE	64.63±0.4	67.66±16.6	71.33±17.5	71.24±17.8	64.93±7.4	71.83±1.3	73.17±11.9	64.90±8.4	86.94±14.3	50.33±12.3	71.07±19.2	88.93±0.6	58.23±6.1	72.55±1.1	69.84
	CVDD	**89.52±0.6**	96.02±0.6	86.27±1.0	94.98±0.5	92.32±0.6	96.37±0.2	92.88±0.6	96.82±0.3	99.53±0.3	96.83±0.3	98.64±0.2	**99.05±0.1**	**97.68±0.1**	86.79±0.8	94.55
	RSRAE	82.67±0.2	94.12±0.2	82.98±0.6	94.33±0.2	91.34±0.1	97.11±0.2	93.02±0.1	95.47±0.1	99.30±0.1	97.93±0.3	97.66±0.1	95.95±0.1	90.14±0.3	91.00±0.7	93.07
	TCCM†	85.31±0.1	92.40±0.2	83.50±0.4	92.78±0.2	89.84±0.2	96.12±0.1	92.99±0.2	93.49±0.1	99.00±0.1	97.45±0.2	97.19±0.2	95.22±0.1	88.61±0.1	88.28±0.8	92.30
	FLOCAT	88.64±0.5	**96.36±0.1**	**92.63±0.3**	**99.15±0.1**	**97.55±0.2**	**99.13±0.1**	**96.09±0.1**	**99.12±0.1**	**99.81±0.0**	**98.92±0.1**	**99.14±0.0**	97.53±0.2	97.51±0.1	**93.35±0.6**	**96.78**
Qwen	DATE	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	CVDD	**83.80±1.2**	88.58±0.4	82.96±0.1	95.45±0.5	**91.61±0.7**	90.95±0.2	87.44±0.6	90.46±0.0	97.86±0.0	**96.91±0.1**	**98.73±0.1**	95.10±0.2	**90.51±0.7**	80.09±0.5	90.75
	RSRAE	65.93±1.4	73.68±0.5	57.60±0.7	68.66±0.7	68.68±0.6	72.69±1.0	72.93±0.9	60.82±0.8	79.70±0.5	72.26±1.0	74.71±0.2	73.13±0.3	64.79±0.5	64.75±0.5	69.31
	TCCM†	73.90±0.3	87.52±0.3	83.24±1.1	92.93±0.4	90.03±0.7	93.03±0.5	86.79±1.0	**91.99±0.6**	**98.24±0.2**	95.29±0.7	94.92±0.2	93.28±0.3	89.37±0.4	78.98±1.4	89.25
	FLOCAT	83.48±0.4	**89.64±0.9**	**83.97±1.2**	**95.59±0.0**	91.52±0.8	**95.78±0.2**	**88.80±0.4**	91.61±0.3	98.09±0.4	96.68±0.1	97.98±0.1	**96.52±0.1**	90.00±0.6	**81.95±0.4**	**91.54**
MPNet	FATE	90.58±0.1	98.41±0.1	**99.52±0.1**	95.56±0.1	97.32±0.1	98.46±0.1	94.83±0.1	97.50±0.1	97.36±0.1	96.84±0.1	98.51±0.1	97.35±0.1	97.40±0.1	96.62±0.1	96.88
	RSRAE	94.32±0.4	97.64±0.1	96.31±0.2	99.80±0.0	99.57±0.0	**99.91±0.1**	97.01±0.1	**99.64±0.1**	99.68±0.1	**99.90±0.0**	**99.81±0.0**	**98.99±0.1**	**99.32±0.1**	**98.11±0.2**	98.57
	TCCM†	92.37±0.3	**99.31±0.1**	98.45±0.2	99.86±0.0	99.53±0.1	99.84±0.1	97.80±0.2	99.36±0.1	99.83±0.0	99.71±0.1	99.53±0.1	98.70±0.2	99.13±0.1	94.90±0.2	98.45
	FLOCAT	**96.58±0.5**	99.20±0.0	98.73±0.2	**99.90±0.0**	**99.58±0.1**	99.83±0.1	**98.67±0.3**	99.48±0.1	**99.88±0.0**	99.60±0.1	99.65±0.0	98.89±0.2	99.07±0.1	94.50±0.3	**98.83**
ST5	FATE	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	RSRAE	93.88±0.3	**99.41±0.1**	97.83±0.4	99.92±0.0	99.47±0.1	**99.93±0.0**	**98.47±0.1**	**99.64±0.1**	**99.95±0.0**	**99.52±0.1**	**99.69±0.0**	**99.40±0.3**	**99.68±0.1**	97.89±0.1	98.91
	TCCM†	93.63±0.1	99.24±0.1	98.29±0.2	99.90±0.0	**99.59±0.0**	99.90±0.0	97.39±0.3	99.02±0.1	99.94±0.0	99.38±0.2	99.66±0.0	99.26±0.3	99.65±0.0	96.58±0.1	98.67
	FLOCAT	**95.32±0.4**	98.79±0.1	**98.34±0.1**	**99.93±0.0**	99.56±0.0	99.86±0.1	98.72±0.3	98.85±0.1	99.90±0.0	99.26±0.1	99.58±0.0	98.95±0.4	99.50±0.1	**99.33±0.2**	**98.99**
E5-Mistral	FATE	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	RSRAE	97.50±0.2	99.60±0.1	98.40±0.1	**99.93±0.0**	99.64±0.1	**99.97±0.0**	**99.30±0.2**	**99.92±0.0**	**99.96±0.0**	**99.91±0.0**	**99.91±0.0**	**99.66±0.1**	99.50±0.1	**99.44±0.0**	99.47
	TCCM†	97.24±0.2	**99.63±0.1**	98.30±0.2	99.87±0.0	99.63±0.1	99.96±0.0	98.71±0.4	99.81±0.1	**99.96±0.0**	99.83±0.1	99.78±0.0	99.59±0.1	**99.53±0.1**	99.10±0.2	99.35
	FLOCAT	**99.06±0.2**	99.60±0.1	**98.73±0.2**	99.90±0.0	**99.68±0.1**	99.92±0.0	99.13±0.2	99.72±0.0	**99.96±0.0**	99.63±0.1	99.74±0.0	99.45±0.2	99.33±0.1	99.09±0.1	**99.50**
LLM-based	Qwen2.5-7B-Instruct	89.92±0.4	96.29±0.4	93.26±0.7	95.13±0.1	96.89±0.4	84.29±0.6	90.30±1.0	90.38±1.0	99.22±0.1	95.19±0.6	96.39±0.1	97.20±0.4	97.64±0.1	74.63±2.4	92.62
LLM-based	Mistral-7B-Instruct	85.30±0.6	92.65±0.1	85.98±0.7	92.28±0.2	95.16±0.2	87.05±0.2	89.44±0.2	89.70±1.1	96.47±0.1	81.34±1.3	91.83±0.1	93.02±0.2	96.89±0.3	76.47±0.9	89.54

Table 19: AUC-ROC on DBpedia14 per category and embedding.

Emb.	Method	World	Sports	Business	Sci-Tech	Avg.
RoBERTa	DATE	67.36±10.2	90.89±0.5	81.33±1.8	79.12±1.4	79.68
	CVDD	78.58±0.8	62.03±0.3	74.35±1.5	79.85±1.8	73.70
	RSRAE	85.98±0.9	89.89±1.5	80.33±1.8	78.12±1.4	83.58
	TCCM†	89.51±0.8	93.79±1.1	83.85±1.5	81.37±1.4	87.13
	FLOCAT	**90.93±0.5**	**96.90±0.5**	**87.83±1.3**	**86.49±0.8**	**90.54**
Qwen	DATE	–	–	–	–	–
	CVDD	**86.99±1.4**	96.74±1.4	78.35±1.4	76.17±1.4	84.56
	RSRAE	71.39±1.3	73.27±0.8	58.52±0.5	61.49±1.1	66.17
	TCCM†	83.94±1.0	95.00±0.3	79.96±0.8	77.13±1.2	84.01
	FLOCAT	85.21±1.4	**97.30±1.4**	**82.24±1.4**	**78.34±1.4**	**85.77**
MPNet	FATE	91.18±1.8	96.67±1.3	91.95±1.1	91.84±1.6	92.91
	RSRAE	92.57±0.4	98.74±0.2	90.81±0.8	**92.59±0.6**	93.68
	TCCM†	94.24±0.2	98.66±0.3	91.74±0.9	91.49±0.7	94.03
	FLOCAT	**94.94±0.1**	**98.77±0.3**	**92.78±0.2**	91.77±0.6	**94.56**
ST5	FATE	–	–	–	–	–
	RSRAE	87.64±0.5	98.29±0.5	90.63±0.4	**86.88±1.4**	90.86
	TCCM†	90.81±0.5	**98.33±0.4**	90.76±0.7	86.22±1.1	91.53
	FLOCAT	**91.73±0.3**	98.24±0.5	**91.00±0.7**	86.62±0.7	**91.90**
E5-Mistral	FATE	–	–	–	–	–
	RSRAE	90.16±0.4	98.32±0.6	92.15±0.3	**91.53±0.6**	93.04
	TCCM†	92.62±0.2	98.38±0.6	91.99±0.4	90.31±0.4	93.33
	FLOCAT	**93.43±0.2**	**98.52±0.4**	**92.61±0.7**	90.79±0.5	**93.84**
LLM-based	Qwen2.5-7B-Instruct	73.12±0.7	96.08±0.4	80.65±0.3	79.34±0.9	82.30
LLM-based	Mistral-7B-Instruct	67.79±0.4	91.57±0.4	85.26±1.5	74.95±0.3	79.89

Table 20: AUC-ROC on AGNews per category and embedding.

Emb.	Method	SMSSpam	Enron
RoBERTa	DATE	67.33±19.3	62.45±14.7
	CVDD	75.81±10.7	78.20±0.6
	RSRAE	45.40±1.3	82.15±1.5
	TCCM†	46.29±1.3	82.44±1.8
	FLOCAT	**76.68±1.3**	**87.66±1.4**
Qwen	DATE	–	–
	CVDD	68.10±1.8	82.15±1.9
	RSRAE	66.43±1.2	65.16±1.1
	TCCM†	69.20±0.2	85.70±1.9
	FLOCAT	**72.58±0.1**	**92.38±0.0**
MPNet	FATE	55.79±2.8	65.43±25.1
	RSRAE	57.96±1.7	95.05±0.5
	TCCM†	57.45±1.0	93.81±0.7
	FLOCAT	**66.30±0.7**	**95.65±0.5**
ST5	FATE	–	–
	RSRAE	62.81±1.1	90.61±0.8
	TCCM†	63.54±1.1	92.50±0.8
	FLOCAT	**72.25±2.4**	**94.10±0.6**
E5-Mistral	FATE	–	–
	RSRAE	57.74±1.7	91.11±1.3
	TCCM†	60.73±2.0	92.97±1.3
	FLOCAT	**65.37±12.1**	**95.01±1.1**
LLM-based	Qwen2.5-7B-Instruct	75.37±0.9	90.68 ±0.4
LLM-based	Mistral-7B-Instruct	74.19±0.3	83.50 ±0.5

Table 21: AUC-ROC on SMSSpam and Enron (single inlier class: ham).

Emb.	Method	IMDB pos.	IMDB neg.	IMDB Avg.	SST-2 pos.	SST-2 neg.	SST-2 Avg.
RoBERTa	DATE	50.06±1.9	54.04±2.2	52.05	49.38±4.4	58.04±5.9	53.71
	CVDD	46.60±0.5	66.36±0.4	56.48	54.83±3.1	44.24±2.5	49.54
	RSRAE	51.66±0.2	**68.35±0.2**	60.00	52.47±7.5	53.77±2.9	53.12
	TCCM†	51.96±1.3	66.07±1.1	59.01	53.67±8.0	53.89±2.8	53.78
	FLOCAT	**52.80±0.8**	67.58±0.1	**60.19**	**63.32±0.8**	**59.67±1.8**	**61.49**
Qwen	DATE	–	–	–	–	–	–
	CVDD	46.61±0.2	67.29±1.6	56.95	45.59±2.4	**53.64±2.4**	49.61
	RSRAE	44.40±0.7	62.42±0.2	53.41	45.42±0.8	50.82±7.4	48.12
	TCCM†	49.07±0.2	65.27±0.2	57.17	**56.81±2.4**	47.59±2.5	52.20
	FLOCAT	**50.33±2.3**	**68.53±1.5**	**59.43**	55.60±1.4	52.60±1.4	**54.10**
MPNet	FATE	47.40±0.8	47.62±6.3	47.51	50.09±4.8	50.40±0.0	50.25
	RSRAE	**55.89±0.6**	45.13±1.0	50.51	**68.66±2.0**	53.74±1.9	**61.20**
	TCCM†	51.88±0.5	45.28±0.4	48.58	68.41±1.0	52.34±1.5	60.38
	FLOCAT	46.86±0.7	**65.32±0.4**	**56.09**	67.90±1.0	**51.92±2.2**	59.91
ST5	FATE	–	–	–	–	–	–
	RSRAE	50.91±0.8	74.32±0.3	62.61	75.16±2.8	**66.93±1.3**	**71.05**
	TCCM†	50.15±0.8	73.09±0.4	61.62	**76.09±3.6**	64.28±0.6	70.19
	FLOCAT	**53.26±1.1**	**74.67±1.6**	**63.97**	70.74±4.4	61.13±1.1	65.94
E5-Mistral	FATE	–	–	–	–	–	–
	RSRAE	**64.67±0.6**	**74.37±0.5**	**69.52**	**63.59±1.5**	**65.07±1.7**	**64.33**
	TCCM†	57.50±0.4	69.42±0.6	63.46	61.81±1.8	60.80±1.2	61.30
	FLOCAT	64.26±2.2	73.01±1.7	68.64	60.96±1.5	61.35±1.2	61.16
LLM-based	Qwen2.5-7B-Instruct	76.16±1.3	58.11±1.0	67.14	69.54±1.3	59.87±4.3	64.71
LLM-based	Mistral-7B-Instruct	81.66±1.9	40.98±1.2	61.32	71.68±1.0	58.84±7.5	65.26

Table 22: AUC-ROC on IMDB and SST-2 per category and embedding.

P Notation Summary

Symbol	Description
Data	
x^i	i-th document
$\mathbf{z}^i \in \mathbb{R}^d$	sentence-level embedding of x^i
$\mathbf{Z}^i \in \mathbb{R}^{S\times d}$	token-level embeddings of x^i
$\mathbf{z}^{i,k} \in \mathbb{R}^d$	k-th token embedding of x^i
S	number of tokens in a document
d	embedding dimension
N	number of inlier training documents
B	batch size during training
Distributions	
p_0	source distribution (inlier embeddings)
$p_1 = \mathcal{N}(\boldsymbol{\mu}, \sigma^2\mathbf{I})$	low-variance Gaussian target distribution
$\boldsymbol{\mu} \in \mathbb{R}^d$	target centroid (inlier barycentre)
σ^2	target variance
$\alpha_c \in (0,1)$	variance ratio controlling target tightness
Flow Matching	
$\mathbf{x}_t$	flow state at time $t \in [0,1]$
$\mathbf{x}_t^{\text{sent},i}$	sentence-level interpolated state
$\mathbf{x}_t^{i,k}$	token-level DiT input at time t
$\mathbf{X}_t^i = (\mathbf{x}_t^{i,1}, \ldots, \mathbf{x}_t^{i,S})$	DiT input sequence at time t
$\boldsymbol{\delta}^{i,k} = \mathbf{z}^{i,k} - \mathbf{z}^i$	per-token residual
$u^i = \mathbf{x}_1^i - \mathbf{x}_0^{\text{sent},i}$	conditional vector field (constant in t for linear CFM)
T	number of Euler integration steps at inference
$\phi(\mathbf{x}_0^{\text{sent},i})$	transported representation at $t = 1$
Model	
h	DiT hidden dimension
L	number of DiT blocks
$\tau(t)$	timestep embedding
$\mathbf{H}^i \in \mathbb{R}^{S\times h}$	DiT output hidden states
$\mathbf{q} \in \mathbb{R}^h$	trainable query for velocity pooling
$\mathbf{h}^{\text{sent},i}$	cross-attention output before linear projection
R	learnable hypersphere radius
λ	weight of $\mathcal{L}_{\text{LOVE}}$
Velocity field	
v_θ	full backbone: $v_\theta(\mathbf{X}_t, t) \to (v_\theta^{\text{sent}}, v_\theta^{\text{tok}})$
v_θ^{sent}	sentence-level velocity function (used for training & scoring)
v_θ^{tok}	token-level velocity function (used for attribution)
$v_t^{\text{sent},i} \in \mathbb{R}^d$	sentence-level velocity output for document i at time t
$v_t^{\text{tok},i} \in \mathbb{R}^{S\times d}$	token-level velocity outputs for document i at time t
$v_t^{i,k} \in \mathbb{R}^d$	velocity of token k for document i at time t
$\hat{\mathbf{x}}_1^i$	one-step Euler estimate of transported endpoint
Losses	
$\mathcal{L}_{\text{FM}}$	flow matching loss
$\ell_i = \max(0, \|\hat{\mathbf{x}}_1^i - \boldsymbol{\mu}\|^2 - R^2)$	per-sample hinge loss
$\mathcal{L}_{\text{LOVE}}$	low-variance loss
$\mathcal{L}_{\text{FLOCAT}}$	total training loss
Scoring & Attribution	
$s(x^i)$	document-level anomaly score
$\cos^{i,k}$	cosine similarity between $v_t^{i,k}$ and $v_t^{\text{sent},i}$
$a^{i,k}$	raw token attribution score (scalar projection)
$\tilde{a}^{i,k} \in [0,1]$	normalized token attribution score
ε	small constant for numerical stability

Table 23: Summary of notation

Q Other Details

Ai-Assistant was used for spelling and typos.