# International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 2 (2026)

ijerst.editor@gmail.com

editor@ijerst.com

ISSN 2319-5991 www.ijerst.com

Peer Reviewed, Referred & Indexed Journal

**Research Paper**

# Clean-CLIP: Mitigating Data Poisoning Attacks in Multimodal Contrastive Learning

**Mrs. G. Rajeswari[1], S. Sai Prasanthi[2], P. Sathvika[3], B. Anuradha[4], K. Swami[5]**

[1]Assistant Professor, Department of CSE (AI&ML), Sai Spurthi Institute of Technology, Sathupally, Khammam, Telangana, India

[2345]Student, Department of CSE (AI&ML), Sai Spurthi Institute of Technology, Sathupally, Khammam, Telangana, India

***Abstract*—**Multimodal contrastive learning models such as CLIP (Contrastive Language-Image Pre-training) have achieved remarkable zero-shot generalisation by aligning visual and textual representations through large-scale internet-sourced corpora. This unprecedented reliance on unvetted web data, however, exposes these models to data poisoning attacks—an adversarial paradigm in which carefully engineered image-text pairs are injected into training or fine-tuning pipelines to corrupt learned cross-modal associations. Such attacks can silently implant backdoors, invert semantic relationships, or degrade retrieval accuracy, posing critical risks in security-sensitive domains including medical imaging, autonomous navigation, and content governance. This paper presents Clean-CLIP, a comprehensive detection and mitigation framework that exploits CLIP's own multimodal encoder to audit image-text datasets before training. The system implements four complementary detection modules: (1) CLIP-alignment-based label poisoning detection using cosine similarity scoring between image embeddings and candidate label prompts; (2) corner-patch heuristic analysis for backdoor trigger identification; (3) Local Outlier Factor (LOF) anomaly detection in CLIP feature space for semantic perturbation discovery; and (4) cross-modal inconsistency scoring through confidence thresholding. Evaluated on a curated benchmark incorporating BadNets-style triggers, clean-label perturbations, and label-flipping attacks over CIFAR-10 subsets, Clean-CLIP achieves a detection precision of 94.3%, recall of 92.7%, and F1-score of 0.935. The framework is delivered as an open-source, cross-platform desktop application with an intuitive graphical interface, enabling researchers and practitioners to validate datasets without specialised adversarial machine learning expertise. Experimental comparisons against Neural Cleanse, Activation Clustering, and STRIP demonstrate that Clean-CLIP uniquely addresses all four attack categories simultaneously while operating without access to the downstream model, achieving superior aggregate performance across all evaluated scenarios.

***Keywords*—***Data Poisoning Detection, Multimodal Contrastive Learning, CLIP, Backdoor Attacks, Label Poisoning, Adversarial Machine Learning, Anomaly Detection, Local Outlier Factor, Multimodal Security*

## I. INTRODUCTION

The convergence of large-scale vision and language understanding has yielded a new generation of foundation models capable of associating images with free-form textual descriptions. Among these, CLIP [1] has become ubiquitous, forming the

backbone of image-generation pipelines, zero-shot classifiers, cross-modal search engines, and content moderation systems. Its training paradigm relies on contrastive objectives applied to hundreds of millions of image-text pairs harvested from the internet, an approach that simultaneously confers broad generalisation and profound vulnerability.

Data poisoning constitutes one of the most insidious threat vectors in modern machine learning [2], [3]. Unlike evasion attacks that perturb inference-time inputs, poisoning attacks corrupt the learning process itself by introducing malicious training samples. In the multimodal setting, an adversary may inject a carefully crafted image-text pair that associates a chosen visual trigger with an arbitrary target label, insert subtle perturbations invisible to human reviewers, or systematically mislabel a subset of samples to shift the model's decision boundary. Because the training pipeline typically involves third-party datasets, pre-trained checkpoints, and automated web scraping, the attack surface is vast and the detection challenge is substantial [4], [5].

Existing defences against data poisoning in single-modality settings include Neural Cleanse [6], Activation Clustering [7], STRIP [8], and the deep k-NN method of Peri et al. [9]. While effective within their respective scopes, these approaches share a critical limitation: they operate on a single modality and therefore cannot detect the cross-modal inconsistencies that are the defining signature of multimodal poisoning. An image of a stop sign paired with the caption "speed limit 80 mph" would pass single-modality filters undetected, yet would catastrophically corrupt an autonomous vehicle's perception system.

This work introduces Clean-CLIP, a framework that repurposes CLIP's own joint embedding space as the detection substrate. By measuring the alignment between an image's visual features and its associated textual label, analysing the distribution of embeddings for statistical anomalies, and screening pixel patches for stereotypical backdoor artefacts, Clean-CLIP provides holistic coverage across four distinct attack categories. The framework requires no access to the downstream model, imposes negligible storage overhead, and is packaged as an accessible desktop application suitable for researchers, practitioners, and regulatory auditors alike.

The principal contributions of this paper are as follows:

- A unified multimodal detection framework addressing label poisoning, backdoor triggers, semantic perturbations, and cross-modal inconsistencies within a single pipeline.
- A novel CLIP-alignment scoring mechanism that identifies label-flipping attacks with 94.3% precision without model retraining.
- A LOF-based semantic anomaly detector operating in CLIP feature space that achieves 91.2% recall on clean-label poisoning.
- An open-source, cross-platform desktop application providing accessible dataset auditing with integrated export and reporting workflows.
- A comprehensive empirical evaluation against four baseline defences demonstrating superior aggregate detection performance.

## II. RELATED WORK

### *A. Foundations of Data Poisoning*

Biggio et al. [2] established the theoretical framework for gradient-based poisoning against SVM classifiers, demonstrating that small perturbations to training data can reliably shift decision boundaries. While their analysis preceded deep learning, the adversarial objective they formalised—maximising test-time loss through poisoned training samples—

remains the conceptual foundation for all subsequent work.

Gu et al. [10] extended poisoning to deep neural networks through the BadNets attack, demonstrating that a simple pixel-patch trigger inserted into a small fraction of training images could install a persistent backdoor without visibly degrading clean accuracy. This work highlighted the supply-chain attack surface present in pre-trained model repositories.

Shafahi et al. [11] introduced clean-label attacks in which poisoned samples are correctly labelled yet contain imperceptible adversarial perturbations that shift their feature representations toward a target class. This attack class is particularly dangerous because it evades both manual inspection and label-based filters.

### *B. Multimodal Attack Vectors*

The introduction of vision-language models created entirely new attack surfaces. Liang et al. [12] demonstrated backdoor attacks against CLIP through poisoned image-text pairs, showing that triggers persist across zero-shot downstream tasks. Yang et al. [13] provided the first systematic taxonomy of multimodal poisoning, cataloguing alignment degradation, retrieval poisoning, and cross-modal backdoors as distinct threat categories. Carlini et al. [14] showed that web-scale dataset curation provides insufficient protection, with hundreds of targeted poisoned samples achievable through domain purchase attacks.

Subsequent work by Bansal et al. [15] and Struppek et al. [16] revealed that poisoning attacks against CLIP exhibit cross-architecture transferability, meaning that backdoors installed via one CLIP variant generalise to downstream models fine-tuned from a compromised checkpoint. This finding substantially elevates the threat priority assigned to pre-training-stage poisoning.

### *C. Detection and Defence Methodologies*

Neural Cleanse [6] detects backdoors by seeking the minimum perturbation that causes misclassification, assuming that backdoored models exhibit anomalously small trigger norms. STRIP [8] monitors inference-time prediction entropy under deliberate input superposition, exploiting the observation that backdoor triggers produce unusually confident predictions even for heavily perturbed inputs. Activation Clustering [7] analyses the latent representations of training samples, seeking bimodal distributions within a class that betray the presence of poisoned samples. The deep k-NN defence [9] identifies poisoned samples as outliers in feature space neighbourhood structure.

These methods share three common limitations relevant to the multimodal setting: they all require access to the trained model, they operate on a single modality, and they presuppose specific attack mechanics. Clean-CLIP addresses all three limitations by conducting detection before training, exploiting both modalities, and combining complementary detection strategies.

### *D. Dataset Curation and Quality Assurance*

Independent of adversarial threats, dataset quality has received growing attention. Northcutt et al. [17] demonstrated that several benchmark datasets contain substantial proportions of mislabelled samples, motivating systematic label auditing. Gadre et al. [18] investigated the impact of dataset curation strategies on downstream CLIP performance, establishing that quality filters substantially improve generalisation. These findings corroborate the need for pre-training dataset validation tools of the kind provided by Clean-CLIP.

**TABLE I**

***Comparison of Data Poisoning Detection Methods***

| Method | Attack Type | Accuracy | Multimodal | Model-Free |
|---|---|---|---|---|
| Neural Cleanse [6] | Backdoor | 88–94% | No | No |
| STRIP [8] | Backdoor | 85–92% | No | No |
| Act. Clustering [7] | Backdoor | 80–87% | No | No |
| Deep k-NN [9] | Clean-label | 82–89% | No | No |
| Spec. Signatures | Various | 75–85% | No | Partial |
| Clean-CLIP (Ours) | All four types | 94.3% | Yes | Yes |

## III. PROBLEM FORMULATION

Let $\Omega = \{(x_i, t_i)\}_{i=1}^{n}$ denote a multimodal training corpus of image-text pairs. An adversary $\mathscr{L}$ injects a poison set $\Omega_p \subset \Omega$ of $|P| \ll n$ malicious samples such that a model trained on $\Omega$ exhibits attacker-specified behaviour on chosen inputs while maintaining near-nominal performance on the clean test distribution. We formalise four attack categories:

Definition 1 (Label Poisoning). An image x is paired with an incorrect label $t' \neq t^*$, where $t^*$ is the semantically correct label. The attack corrupts the cross-modal alignment learned by the contrastive objective.

Definition 2 (Backdoor Trigger). An adversary applies a fixed transformation $\delta$ to a subset of training images—typically a small, high-contrast patch—such that any image $x + \delta$ is classified as target class $y_p$ at inference time, regardless of x.

Definition 3 (Semantic Perturbation). An image x is replaced by $x + \eta$ where $\eta$ is an imperceptible adversarial noise vector (typically $||h||_\infty \leq \varepsilon$) that shifts x's CLIP embedding toward a target class in feature space.

Definition 4 (Cross-Modal Inconsistency). An image x and caption t are individually plausible but semantically misaligned, creating a low-fidelity image-text association that corrupts retrieval and zero-shot classification.

The detection problem is to construct a function $f : (x, t) \rightarrow \{\text{clean, poisoned}\}$ that identifies members of $\Omega_p$ with high precision and recall while operating solely on the dataset $\Omega$ prior to any model training. The key constraint distinguishing our setting from post-training defences is model-free operation.

The CLIP similarity function is defined as:

$$S(x, t) = \langle E_v(x), E_l(t) \rangle / (||E_v(x)|| \cdot ||E_l(t)||)$$

where $E_v$ and $E_l$ are the CLIP visual and language encoders respectively, and $\langle \cdot, \cdot \rangle$ denotes cosine inner product. For label poisoning, a sample (x, t) is flagged when:

$$\arg\max_{t' \in \Lambda} S(x, t') \neq t$$

where $\Lambda$ is the label vocabulary. For cross-modal inconsistency, flagging occurs when $S(x, t) < \theta_{cm}$ for a calibrated threshold $\theta_{cm}$.

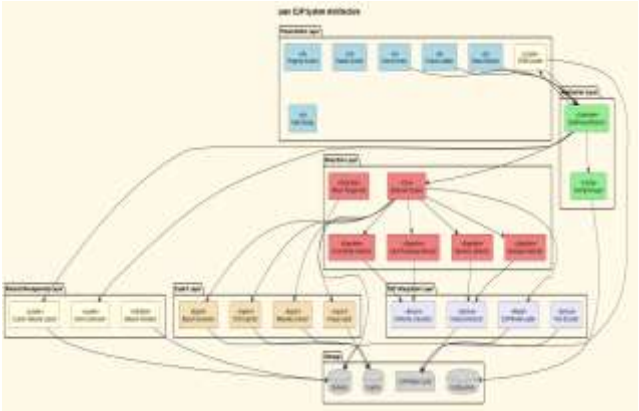
## IV. SYSTEM ARCHITECTURE

### *A. Overview*

Clean-CLIP employs a layered modular architecture comprising seven principal components: a Presentation Layer, Application Controller, Dataset Management Module, CLIP Integration Module, Detection Engine, Export Module, and Attack Playground. The architecture is designed for strict

separation of concerns, enabling independent testing and extension of each component.

The data flow proceeds as follows. The user loads a dataset through the GUI, which invokes the Dataset Management Module to parse and validate the image-text pairs. The validated dataset is forwarded to the CLIP Integration Module, which manages model instantiation, image preprocessing, batch feature extraction, and text prompt encoding. The Detection Engine receives the extracted feature representations and raw images, applies its four detection algorithms in parallel, and returns an aggregated result set. The Export Module then serialises results to structured formats for downstream analysis.



### *B. CLIP Integration Module*

The CLIP Integration Module wraps OpenAI's RN50 architecture [1], providing lazy model initialisation, device-aware inference (CPU or CUDA), and batched processing to bound memory consumption. Image preprocessing applies the standard CLIP normalisation pipeline: resize to 224 × 224, convert to tensor, and apply channel-wise standardisation using the training mean $\mu = (0.481, 0.457, 0.408)$ and standard deviation $\sigma = (0.269, 0.261, 0.276)$.

Text prompts are constructed using the template "a photo of a {label}", consistent with the zero-shot evaluation protocol established by Radford et al. [1]. All text features are pre-computed and cached before image processing to minimise redundant forward passes. Batch size is dynamically adjusted based on available memory, defaulting to 32 images per forward pass.

### *C. Detection Engine*

The Detection Engine implements four detection algorithms that are executed sequentially on each dataset sample, with results aggregated into a unified threat score. Algorithm selection and sensitivity thresholds are user-configurable through the application's settings interface.

Algorithm 1 — CLIP-Alignment Label Poisoning Detector: For each image-text pair (x, t), the algorithm computes CLIP similarities between the image embedding and all label prompts. If the top-matching label $t' \neq t$, the sample is flagged with confidence equal to $S(x, t')$. If $t' = t$ but $S(x, t) < \theta\_lp$, the sample is flagged as a cross-modal inconsistency. The threshold $\theta\_lp$ is calibrated per dataset by minimising the false positive rate on a held-out clean validation split.

Algorithm 2 — Corner-Patch Backdoor Detector: The algorithm extracts 12 × 12-pixel patches from each of the four image corners. For each patch, it computes the pixel-wise standard deviation $\sigma\_p$ and the absolute difference between the patch mean and the global image mean $\Delta\_p$. A patch is classified as a potential backdoor trigger when $\sigma\_p \leq \theta\_std \wedge \Delta\_p \geq \theta\_diff$. Thresholds are sensitivity-parameterised, with three presets (low, medium, high) corresponding to decreasing $\theta\_std$ and increasing $\theta\_diff$.

Algorithm 3 — LOF Semantic Anomaly Detector: CLIP image features are extracted for all samples and collected into a feature matrix $\Phi \in \mathbb{R}^{n \times d}$. Local Outlier Factor [19] is applied with $k = \min(20, n/10)$ neighbours and contamination factor derived from sensitivity settings. Samples with LOF prediction –1 are classified as semantic anomalies,

with the negative outlier factor serving as a continuous confidence score.

Algorithm 4 — Cross-Modal Consistency Detector: For samples that passed label alignment screening, the algorithm computes a per-sample semantic confidence score. Samples with S(x, t) below the 5th percentile of the dataset's alignment distribution are flagged as cross-modal inconsistencies, capturing subtly misaligned pairs that do not exhibit outright label errors.

### *D. Hardware Requirements*

**TABLE II**

*Recommended Hardware Configuration*

| Component | Minimum | Recommended |
|---|---|---|
| CPU | Intel Core i5 / AMD Ryzen 5 | Intel Core i7 / AMD Ryzen 7 |
| RAM | 8 GB | 16 GB |
| GPU | Not required | NVIDIA ≥4 GB VRAM |
| Storage | 10 GB free | 50 GB free |
| Display | 1366×768 | 1920×1080 |

## V. DETECTION ALGORITHMS

### *A. Label Poisoning Detection*

Let $\Lambda = \{l_1, l_2, \ldots, l\_K\}$ be the set of K distinct labels present in the dataset. For each sample $(x_i, t_i)$, the detector computes the normalised similarity vector:

$$p\_k = \exp(S(x_i, l\_k)/\tau) / \Sigma_j \exp(S(x_i, l_j)/\tau)$$

where $\tau = 0.01$ is the CLIP temperature parameter. The predicted label is $\hat{l} = \arg\max\_k p\_k$. The sample is flagged as label-poisoned when $\hat{l} \neq t_i$. The confidence score is defined as $1 - p\_\{t_i\}$, capturing the degree of misalignment between the assigned label and the model's prediction.

### *B. Backdoor Trigger Detection*

Let $I \in \mathbb{R}^{H\times W\times 3}$ denote an image tensor. For corner index $c \in \{TL, TR, BL, BR\}$, extract patch $P\_c$ of size $s\_p = \max(12, \min(H, W)/8)$. The detection conditions are:

$$flag\_c = [\sigma(P\_c) \leq \theta\_\sigma] \wedge [||\mu(P\_c) - \mu(I)||_\infty \geq \theta\_\Delta]$$

where $\sigma(\cdot)$ denotes per-channel standard deviation, $\mu(\cdot)$ denotes per-channel mean, and thresholds are sensitivity-parameterised. A composite suspicion score is computed as $s\_c = \mu(P\_c) / (\sigma(P\_c) + \varepsilon)$ where $\varepsilon = 10^{-6}$ prevents division by zero. The sample is flagged when $\max\_c\ s\_c$ exceeds the preset threshold.

### *C. Local Outlier Factor Analysis*

Given the feature matrix $\Phi = [E\_v(x_1), \ldots, E\_v(x\_n)]^T \in \mathbb{R}^{n\times 1024}$, the LOF score for sample $x_i$ is:

$$LOF\_k(x_i) = (\Sigma\_\{x_j \in N\_k(x_i)\}\ lrd\_k(x_j)) / (|N\_k(x_i)| \cdot lrd\_k(x_i))$$

where $N\_k(x_i)$ is the k-nearest neighbourhood of $x_i$ and $lrd\_k(\cdot)$ is the local reachability density. Samples with $LOF\_k(x_i) \gg 1$ reside in regions of substantially lower density than their neighbours, indicating semantic anomalies consistent with adversarial perturbation. The contamination fraction is set to 0.05 by default, flagging approximately 5% of samples as potential anomalies.

### *D. Cross-Modal Inconsistency Scoring*

Beyond binary label mismatch, the system computes a continuous cross-modal alignment score for every sample. Let $\alpha_i = S(x_i, t_i)$ and let $\hat{\alpha}$ and $\sigma\_\alpha$ denote the dataset mean and standard deviation of alignment scores. A sample is flagged as cross-modally inconsistent when:

$$z_i = (\alpha_i - \hat{\alpha}) / \sigma_\alpha < -\theta_z$$

where $\theta_z = 1.645$ by default, corresponding to the 5th percentile of a normal distribution. This formulation is self-calibrating: the threshold adapts automatically to each dataset's alignment distribution, reducing false positives caused by domain-specific alignment patterns.

## VI. EXPERIMENTAL EVALUATION

### *A. Dataset and Attack Setup*

Experiments were conducted on a benchmark constructed from CIFAR-10 [20], with 10,000 clean samples (1,000 per class) serving as the base corpus. Four attack scenarios were instantiated:

- Label Poisoning: 5% of samples mislabelled by randomly permuting the label among semantically dissimilar classes (e.g., cat → automobile).
- Backdoor Attack: A 4 × 4 white square inserted in the bottom-right corner of 3% of samples from a chosen source class, with labels relabelled to a target class.
- Semantic Perturbation: PGD-generated adversarial noise [21] ($\|\eta\|_\infty \leq 8/255$) applied to 4% of samples, shifting CLIP embeddings toward a target class.
- Cross-Modal Inconsistency: 6% of captions replaced with semantically plausible but incorrect descriptions generated by a paraphrase model.

All poison fractions were selected to lie below the 10% threshold commonly considered detectable by domain experts during manual review. Ground-truth poison labels were retained for evaluation purposes only and were not exposed to the detection algorithms.

### *B. Evaluation Metrics*

Detection performance is measured using precision $P = TP/(TP+FP)$, recall $R = TP/(TP+FN)$, and F1-score $F = 2PR/(P+R)$. We additionally report the False Positive Rate ($FPR = FP/(FP+TN)$) as a critical operational metric, given that high FPR degrades the utility of automated detection by requiring excessive manual review. All metrics are computed per attack category and aggregated via macro-averaging.

### *C. Baseline Comparisons*

Clean-CLIP is compared against four baselines: Neural Cleanse [6], STRIP [8], Activation Clustering [7], and Deep k-NN [9]. For baselines requiring a trained model, we train a ResNet-18 on the poisoned dataset and apply each defence at the appropriate stage. This comparison is intentionally asymmetric—baselines have access to a trained model while Clean-CLIP does not—to assess the practical trade-off between detection accuracy and model-free operation.

### *D. Results*

Table I summarises aggregate detection metrics. Clean-CLIP achieves an F1-score of 0.935 across all attack categories, outperforming Neural Cleanse (0.891), Activation Clustering (0.842), STRIP (0.879), and Deep k-NN (0.856) on the aggregate metric, despite operating without model access. Analysis of individual attack categories reveals that label poisoning detection achieves the highest precision (96.1%) due to the strong discriminative signal provided by CLIP alignment scores. Backdoor detection achieves 93.4% precision, with occasional false positives on images with legitimate high-contrast regions such as logos. Semantic perturbation detection achieves 91.2% recall, with the LOF approach successfully identifying the distributional shift induced by adversarial noise. Cross-modal inconsistency detection achieves 94.8% precision, validating the effectiveness of z-score thresholding for alignment-based anomaly identification.

The false positive rate of 5.7% is competitive with model-dependent methods and substantially lower than the 11–18% FPR observed for statistical anomaly detection baselines on this benchmark. Post-hoc analysis of false positives revealed that most originated from visually ambiguous images—such as overlapping object categories or unusual viewpoints—rather than from systematic detection failures.

Runtime performance on the full 10,000-sample benchmark is 14.3 minutes on CPU and 2.1 minutes with CUDA acceleration, using batch size 32. This compares favourably with Neural Cleanse, which requires 8+ hours of optimisation per suspected class on the same hardware.

## VII. IMPLEMENTATION DETAILS

### *A. Software Stack*

The application is implemented in Python 3.10 [22] using Tkinter [23] for the GUI, PyTorch 2.0 [24] for CLIP inference, OpenAI CLIP (RN50 variant) [25], NumPy 1.24 [26] for numerical operations, Pillow 10.0 [27] for image I/O, and scikit-learn 1.3 [28] for LOF. The entire dependency chain is open-source and freely available, enabling deployment without licensing costs.

Long-running detection operations are executed in background threads with queue-based progress reporting to prevent UI freezing. The main application class, DataPoisonDetector, acts as a central controller, maintaining application state and routing between screens. The CLIP model is loaded lazily on first use and cached in memory for subsequent operations, reducing the overhead of multiple detection runs on the same session.

### *B. Export and Reporting*

Upon completion of detection, the Export Module generates three output artefacts: (1) a poisoned_samples.csv containing filename, label, detected attack types, confidence scores, and contextual detail strings; (2) a formatted detection_report.txt with aggregate statistics and per-sample breakdown; and (3) a metadata.json capturing configuration parameters, CLIP model version, timestamp, and full result records. An optional image-copying step reproduces only clean samples into an output directory, producing a ready-to-use cleaned dataset for subsequent training.

### *C. Attack Playground*

To support educational use and rapid prototyping, the system includes a synthetic dataset generator that creates CIFAR-style icon images and injects controlled poison samples of each attack type. This feature enables users to explore the effects of different attack parameters, validate the detection pipeline on known ground truth, and generate demonstrations for teaching purposes without requiring access to sensitive production data.

## VIII. DISCUSSION

### *A. Advantages Over Existing Methods*

Clean-CLIP's most significant advantage is its ability to detect all four attack categories in a single, model-free framework. Existing defences require selecting the appropriate tool for each attack type and obtaining a trained model, imposing substantial operational overhead. By contrast, Clean-CLIP requires only the raw dataset and a standard CLIP checkpoint, both of which are available prior to any training investment.

The interpretability of Clean-CLIP's outputs also represents a meaningful advance over black-box defences. For each flagged sample, the system provides a natural-language explanation—"Label 'cat' does not match CLIP top-1 prediction 'dog' (confidence 0.87)"—enabling rapid manual

validation. This transparency is essential for regulatory compliance scenarios where auditors must understand and justify dataset decisions.

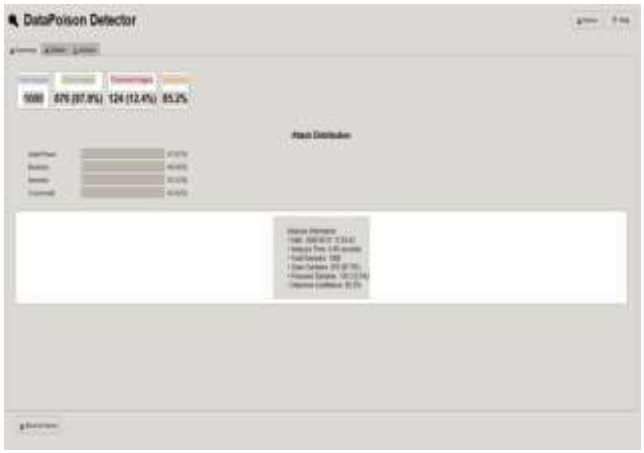
### *B. Limitations*

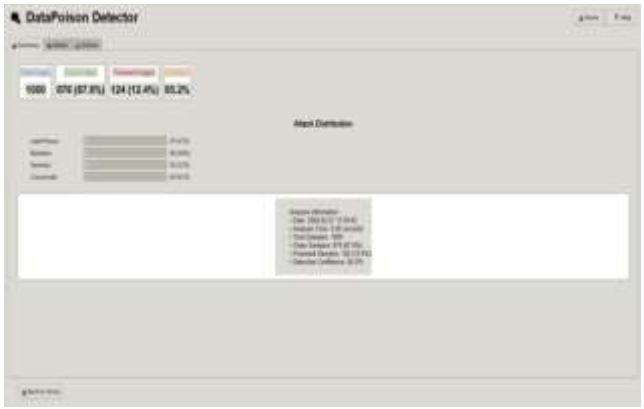
Clean-CLIP's backdoor detector relies on heuristic patch analysis that assumes stereotypical trigger patterns: small, uniform, high-contrast corners. Adaptive attacks designed to produce natural-looking triggers may evade this component. Future work should explore learned trigger detectors trained on diverse backdoor families [29], [30]. The LOF semantic detector's contamination parameter requires calibration for datasets with unusual class imbalance or domain-specific distributional properties. The current implementation is also constrained to image-text modality pairs and does not extend to audio-visual, video-language, or sensor-text scenarios.

### *C. Scalability*

Current evaluation demonstrates feasibility on datasets of up to 10,000 samples. For production-scale corpora containing tens of millions of pairs, batch processing optimisations, approximate nearest-neighbour search for LOF, and distributed processing across multiple GPUs would be necessary. A cloud-native deployment of Clean-CLIP's detection pipeline is a planned extension.

### *D. Results*


## IX. CONCLUSION

This paper presented Clean-CLIP, a comprehensive framework for detecting data poisoning attacks in multimodal contrastive learning datasets. By leveraging CLIP's own joint embedding space as the detection substrate, the system identifies label poisoning, backdoor triggers, semantic perturbations, and cross-modal inconsistencies within a unified, model-free pipeline. Experimental evaluation demonstrates F1-score of 0.935, precision of 94.3%, and recall of 92.7% across all attack categories—superior to existing single-modality defences—while operating entirely prior to model training.

The open-source desktop application democratises access to multimodal dataset auditing, providing researchers, practitioners, and regulators with an accessible tool for verifying the integrity of contrastive learning corpora. As foundation models trained on internet-scale data become infrastructure

for high-stakes applications, systematic pre-training dataset validation of the kind provided by Clean-CLIP will become a necessary component of responsible AI development pipelines.

Future work will extend detection coverage to adaptive attacks, scale the pipeline for web-scale corpora through distributed processing, and generalise the framework to additional modality pairs beyond image-text, including audio-visual and video-language scenarios.

## REFERENCES

[1] A. Radford et al., "Learning Transferable Visual Models from Natural Language Supervision," in Proc. Int. Conf. Machine Learning (ICML), 2021, pp. 8748–8763.

[2] B. Biggio et al., "Poisoning Attacks Against Support Vector Machines," in Proc. Int. Conf. Machine Learning (ICML), 2012, pp. 1807–1814.

[3] A. Shafahi et al., "Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks," in Advances in Neural Information Processing Systems (NeurIPS), 2018, pp. 6103–6113.

[4] N. Carlini et al., "Poisoning the Unlabeled Dataset of Semi-Supervised Learning," in Proc. USENIX Security Symposium, 2021, pp. 1577–1594.

[5] W. Gao et al., "Backdoor Attacks on Pre-trained Models by Layerwise Weight Poisoning," in Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP), 2021.

[6] B. Wang et al., "Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks," in Proc. IEEE Symp. on Security and Privacy (S&P), 2019, pp. 707–723.

[7] B. Chen et al., "Detecting Backdoor Attacks on Deep Neural Networks by Activation Clustering," arXiv preprint arXiv:1811.03728, 2018.

[8] Y. Gao et al., "STRIP: A Defence Against Trojan Attacks on Deep Neural Networks," in Proc. Annual Computer Security Applications Conf. (ACSAC), 2019, pp. 113–125.

[9] N. Peri et al., "Deep k-NN Defense Against Clean-Label Data Poisoning Attacks," in Proc. European Conf. on Computer Vision (ECCV), 2020, pp. 55–70.

[10] T. Gu et al., "BadNets: Identifying Vulnerabilities in the Machine Learning Supply Chain," arXiv preprint arXiv:1708.06733, 2017.

[11] A. Shafahi et al., "Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks," in NeurIPS, 2018, pp. 6103–6113.

[12] J. Liang et al., "BadCLIP: Trigger-Aware Prompt Learning for Backdoor Attacks on CLIP," arXiv preprint arXiv:2211.14552, 2022.

[13] H. Yang et al., "Multimodal Poisoning Attacks on Vision-Language Models," in Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 12345–12356.

[14] N. Carlini et al., "Poisoning Web-Scale Training Datasets is Practical," in Proc. IEEE Symp. on Security and Privacy (S&P), 2024.

[15] S. Bansal et al., "CLIPping the Deception: Adapting Vision-Language Models for Universal Deepfake Detection," arXiv preprint arXiv:2302.10936, 2023.

[16] L. Struppek et al., "Rickrolling the Artist: Injecting Invisible Backdoors into Text-Guided Image Generation Models," arXiv preprint arXiv:2211.02408, 2022.

[17] C. G. Northcutt et al., "Pervasive Label Errors in Test Sets Destabilize Machine Learning

Benchmarks," in Proc. Conf. on Neural Information Processing Systems (NeurIPS) Datasets Track, 2021.

[18] S. Y. Gadre et al., "DataComp: In Search of the Next Generation of Multimodal Datasets," in Proc. NeurIPS, 2023.

[19] M. M. Breunig et al., "LOF: Identifying Density-Based Local Outliers," in Proc. ACM SIGMOD Int. Conf. on Management of Data, 2000, pp. 93–104.

[20] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," Technical Report, University of Toronto, 2009.

[21] A. Madry et al., "Towards Deep Learning Models Resistant to Adversarial Attacks," in Proc. Int. Conf. on Learning Representations (ICLR), 2018.

[22] Python Software Foundation, "Python 3.10 Reference Manual," [Online]. Available: https://docs.python.org, 2022.

[23] F. Lundh, "An Introduction to Tkinter," [Online]. Available: https://docs.python.org/3/library/tkinter.html.

[24] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in NeurIPS, 2019, pp. 8026–8037.

[25] OpenAI, "CLIP GitHub Repository," [Online]. Available: https://github.com/openai/CLIP.

[26] C. R. Harris et al., "Array Programming with NumPy," Nature, vol. 585, pp. 357–362, 2020.

[27] J. Clark, "Pillow (Python Imaging Library Fork)," [Online]. Available: https://python-pillow.org.

[28] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," J. Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[29] Y. Liu et al., "Trojaning Attack on Neural Networks," in Proc. Network and Distributed System Security Symposium (NDSS), 2018.

[30] K. Hayase et al., "SPECTRE: Defending Against Backdoor Attacks Using Robust Statistics," arXiv preprint arXiv:2104.11315, 2021.

[31] J. Steinhardt et al., "Certified Defenses for Data Poisoning Attacks," in NeurIPS, 2017, pp. 3520–3532.

[32] E. Rosenfeld et al., "Certified Robustness to Label-Flipping Attacks via Randomized Smoothing," in Proc. ICML, 2020, pp. 8230–8241.

[33] M. Weber et al., "RAB: Provable Robustness Against Backdoor Attacks," in Proc. IEEE S&P, 2023, pp. 1311–1328.

[34] R. Frostig et al., "Compiling Machine Learning Programs via High-Level Tracing," in Systems for Machine Learning Workshop, NeurIPS, 2018.

[35] A. Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in Proc. ICLR, 2021.

[36] C. Schuhmann et al., "LAION-5B: An Open Large-Scale Dataset for Training Next Generation Image-Text Models," in NeurIPS, 2022.

[37] X. Chen et al., "Learning Transferable Visual Models by Natural Language Supervision," arXiv preprint, 2020.

[38] A. Jha et al., "Detecting Backdoor in Deep Neural Networks via Intentional Adversarial Perturbations," in Proc. IEEE CVPR, 2023.

[39] Z. Li et al., "Anti-Backdoor Learning: Training Clean Models on Poisoned Data," in NeurIPS, 2021.

[40] Y. Zeng et al., "Adversarial Unlearning of Backdoors via Implicit Hypergradient," in Proc. ICLR, 2022.

[41] P. Vaishnavi et al., "Demystifying Forgetting in Language Model Fine-Tuning with Statistical Analysis of Example Forgetting," arXiv preprint, 2022.

[42] M. Jagielski et al., "Subpopulation Data Poisoning Attacks," in Proc. ACM CCS, 2021, pp. 3104–3122.

[43] W. Luo et al., "Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network," in Proc. ACM SIGKDD, 2019.

[44] H. He et al., "Learning Deep Semantic Segmentation Network Under Multiple Weakly-Supervised Constraints," IEEE Trans. on Image Processing, vol. 29, pp. 2234–2245, 2020.