Adversarial Robustness and Explainability in Extreme Multi-Label Classification with Extended Probabilistic Label Trees

Atharva Date
Department of Computer Science, Indian Institute of Technology
Jodhpur, India
b22ai045@iitj.ac.in

Samay Mehar
Department of Computer Science, Indian Institute of Technology
Jodhpur, India
b22ai048@iitj.ac.in

Abstract

Extreme Multi-Label Classification (XMLC) tackles the challenge of assigning multiple relevant labels from vast label spaces, with applications in text categorization and multimedia tagging. Probabilistic Label Trees (PLTs) offer a scalable solution, but their robustness to adversarial perturbations and interpretability remain underexplored. We introduce *ExtendedPLT*, an enhanced PLT model implemented using the napkinXC library, featuring adversarial training to improve robustness, an explainability module to highlight critical features, and an adversarial evaluation framework to assess performance under Gaussian noise. Experiments on Eurlex-4k and Mediamill datasets compare normal and adversarial training, measuring Precision@k, nDCG@k, and PSDCG@k at noise levels of 0.01, 0.05, and 0.1 for $k = 1, 3, 5$. Results show that adversarial training significantly enhances robustness on Mediamill, reducing Precision@1 degradation from 11.79% to 6.33% at noise=0.1, though it slightly lowers clean performance. On Eurlex-4k, both models exhibit high robustness, with adversarial training marginally improving clean Precision@1 from 0.8104 to 0.8146. Explainability analysis reveals consistent feature importance, enhancing trust. Visualizations of performance, prediction time, and noise impact provide transparency. Our findings highlight the effectiveness of adversarial training for low-dimensional datasets and the inherent robustness of high-dimensional inputs, offering insights for deploying XMLC models in noisy environments.

CCS Concepts

• **Computing methodologies → Machine learning**; *Neural networks.*

Keywords

Extreme Multi-Label Classification, Probabilistic Label Trees, Adversarial Robustness, Adversarial Training, Explainability, napkinXC

ACM Reference Format:
Atharva Date and Samay Mehar. 2025. Adversarial Robustness and Explainability in Extreme Multi-Label Classification with Extended Probabilistic Label Trees. In . ACM, New York, NY, USA, Article 1, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Extreme Multi-Label Classification (XMLC) involves predicting a subset of relevant labels from an extremely large label space, with applications in document tagging [1], image annotation [2], and recommendation systems [3]. The computational complexity of large label spaces poses significant challenges, requiring models that balance scalability, accuracy, robustness, and interpretability. Probabilistic Label Trees (PLTs) address scalability by organizing labels into a hierarchical tree, reducing prediction time to logarithmic complexity [6]. However, their susceptibility to adversarial perturbations and lack of interpretability limit their dependability in real-world settings.

Adversarial robustness is critical as XMLC models often encounter noisy inputs, such as perturbed text features or multimedia descriptors. Small perturbations can degrade performance, especially in multi-label settings where label interdependencies amplify errors. While adversarial robustness has been studied in single-label classification [8], XMLC's large output spaces and ranking objectives introduce unique challenges. Adversarial training, which augments data with perturbed examples, has shown promise [9], but its application to PLTs remains unexplored. Equally important is explainability, as users in domains like legal classification (Eurlex-4k) or video annotation (Mediamill) need to understand model decisions for trust and validation. Standard PLTs lack mechanisms to highlight influential features, hindering transparency.

We propose *ExtendedPLT*, an enhanced PLT model built on the napkinXC library [14], addressing these gaps through three innovations: (1) adversarial training to enhance robustness by augmenting data with Gaussian noise, (2) an explainability module to identify top contributing features, and (3) an adversarial evaluation framework to quantify performance under noise levels of 0.01, 0.05, and 0.1. ExtendedPLT compares normal training (clean data) with adversarial training (clean plus noisy data) on Eurlex-4k and Mediamill, evaluating Precision@k, nDCG@k, and PSDCG@k for $k = 1, 3, 5$. Our findings show that adversarial training significantly improves robustness on Mediamill, reducing Precision@1 drops from 11.79% to 6.33% at noise=0.1, though it slightly reduces clean performance. On Eurlex-4k, both models are robust, with adversarial training improving clean Precision@1 from 0.8104 to 0.8146. Explainability analysis reveals consistent feature importance, and visualizations enhance transparency.

Conference'17, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

NapkinXC's utilities, including `load_dataset` for sparse data and `precision_at_k`, `ndcg_at_k`, and `psdcg_at_k` for metrics, enable efficient experimentation. The `PLT` class is extended with `fit_for_adversarial_examples` for training, `explain` for feature importance, and `adversarial_evaluation` for robustness testing. Unlike prior PLT models focused on scalability [7], ExtendedPLT balances robustness and interpretability, making it suitable for noisy environments.

Our contributions are:

1. *ExtendedPLT*, a PLT model with adversarial training and explainability, implemented in napkinXC.
2. An adversarial evaluation framework comparing normal and adversarial training under Gaussian noise.
3. Comprehensive experiments on Eurlex-4k and Mediamill, with visualizations elucidating robustness, performance, and feature importance.

Section 2 reviews related work, 3 details ExtendedPLT, 4 describes the setup, 5 presents results, and 6 concludes with future directions.

2 Related Work

XMLC addresses large-scale multi-label tasks like document tagging [1], image annotation [2], and recommendations [3]. Early methods like One-vs-Rest [4] scale poorly, leading to embedding-based approaches (SLEEC [5]) and tree-based models like PLTs [6]. Parabel [7] optimizes PLTs for balanced clustering, but overlooks robustness and explainability.

Adversarial robustness is well-studied in single-label settings [8, 9], with adversarial training mitigating perturbation effects. Multi-label robustness is less explored; Song et al. [10] focus on images, not XMLC's ranking tasks. No prior work applies adversarial training to PLTs, a gap our work addresses.

Explainability enhances trust in complex models [11]. LIME [11] and SHAP [12] are computationally intensive for XMLC, while attention-based models like XML-CNN [13] sacrifice scalability. PLTs lack native explainability, which ExtendedPLT resolves with lightweight feature importance.

NapkinXC [14] supports XMLC with sparse data handling and metrics like PSDCG@k [15]. Unlike ExtremeClassification [**?**], it enables modular extensions, which we leverage for robustness and explainability, advancing prior benchmark studies [14].

3 Methodology

ExtendedPLT extends napkinXC's PLT model [14] with adversarial training, explainability, and robustness evaluation. Below, we describe its components.

3.1 ExtendedPLT Model

PLTs organize labels in a tree, with binary classifiers at nodes predicting subtree membership [6]. Predictions traverse the tree, ranking labels by path probabilities. ExtendedPLT adds:

- Normal Training (`fit`): Trains on clean data using napkinXC's `PLT.fit`.
- Adversarial Training (`fit_for_adversarial_examples`): It Augments training data with noisy samples. For input $X \in \mathbb{R}^{n\times d}$, $Y \in \{0,1\}^{n\times L}$, it generates $m = n \cdot \text{augment\_ratio}$ noisy samples:

  $$X_{\text{noisy}} = X[:m] + \epsilon, \quad \epsilon \sim \mathcal{N}(0,\sigma^2), \quad \sigma = 0.05$$

  The combined dataset $(X \cup X_{\text{noisy}}, Y \cup Y[:m])$ is trained, doubling size at augment_ratio = 1.0.
- Explainability (`explain`)**: For instance $X_{\text{instance}} \in \mathbb{R}^d$, it computes:

$$\text{top\_indices} = \text{argsort}(|X_{\text{instance}}|)[-5:][::-1], \quad \text{top\_values} = X_{\text{instance}}[\text{top\_indi}$$

  Returns $[(\text{index}_i, x_i)]_{i=1}^5$, handling sparse inputs.

3.2 Adversarial Evaluation Framework

The `adversarial_evaluation` method tests robustness by adding noise to test inputs:

$$X_{\text{noisy}} = X + \epsilon, \quad \epsilon_{ij} \sim \mathcal{N}(0,\sigma^2), \quad \sigma \in \{0.01, 0.05, 0.1\}$$

It predicts top-k labels ($k = 1, 3, 5$) and computes: - **Precision@k**: Fraction of correct top-k labels. - **nDCG@k**: Ranking quality with logarithmic discounting. - **PSDCG@k**: nDCG weighted by inverse propensity scores:

$$\text{inv\_ps}_j = \frac{1}{\log(1 + \text{count}_j)}$$

Prediction times are recorded. Algorithm 1 outlines the process.

Algorithm 1 Adversarial Evaluation in ExtendedPLT

```
Require: Test set $X$, labels $Y$, inv_ps, noise levels $\sigma$, $k$-values
Ensure: Results dictionary
results = $\{k : \{\}$ for $k \in k_{\text{values}}\}$
for $k \in k_{\text{values}}$ do
    for $\sigma \in$ noise_levels do
        $X_{\text{noisy}} \leftarrow X + \mathcal{N}(0, \sigma^2 \cdot I)$
        Start timer
        $Y_{\text{pred}} \leftarrow \text{predict}(X_{\text{noisy}}, \text{top\_k} = k)$
        pred_time ← stop timer
        $P@k \leftarrow \text{mean}(\text{precision\_at\_k}(Y, Y_{\text{pred}}, k))$
        nDCG@k ← $\text{mean}(\text{ndcg\_at\_k}(Y, Y_{\text{pred}}, k))$
        PSDCG@k ← $\text{mean}(\text{psdcg\_at\_k}(Y, Y_{\text{pred}}, \text{inv\_ps}, k))$
        results$[k][\sigma] \leftarrow \{P@k$, nDCG@k, PSDCG@k, pred_time$\}$
    end for
end for
return results
```

3.3 Visualization Techniques

We visualize: - **plot_performance_metrics**: Precision@k, nDCG@k, PSDCG@k vs. noise for normal/adversarial models. - **plot_prediction_time**: Prediction time vs. noise. - **plot_feature_importance**: Bar chart of top features. - **plot_noise_impact**: Clean vs. noisy feature values for three samples.

These leverage Matplotlib, with napkinXC handling data and metrics.

4 Experimental Setup

We evaluate ExtendedPLT on Eurlex-4k and Mediamill, comparing normal and adversarial training.

4.1 Datasets

- **Eurlex-4k**: 15, 539 train, 3, 809 test samples, 5, 000 features, 4, 000 labels. High-dimensional text data.
- **Mediamill**: 30, 993 train, 12, 914 test samples, 121 features, 101 labels. Low-dimensional multimedia data.

Table 1 summarizes statistics.

Table 1: Dataset Statistics

Dataset	Train Samples	Test Samples	Features	Labels
Eurlex-4k	15, 539	3, 809	5, 000	4, 000
Mediamill	30, 993	12, 914	121	101

4.2 Evaluation Metrics

Metrics for $k = 1, 3, 5$: - **Precision@k**: Top-k accuracy. - **nDCG@k**: Ranking quality. - **PSDCG@k**: Rare label performance. Prediction time assesses efficiency.

4.3 Implementation Details

Using napkinXC v0.6.0, Python, NumPy, and Matplotlib, we train ExtendedPLT on an Intel Xeon (2.4 GHz, 16 cores) with 64 GB RAM. Adversarial training uses $\sigma = 0.05$, augment_ratio = 1.0. Noise levels are 0.01, 0.05, 0.1. Training times are 39.16/542.97 seconds (normal/adversarial) for Eurlex-4k, 25.27/56.89 seconds for Mediamill.

4.4 Baseline Comparisons

We compare normal and adversarial training, using clean-data performance as the baseline.

5 Results and Discussion

We present a comprehensive analysis of the *ExtendedPLT* model's performance on the Eurlex-4k and Mediamill datasets, comparing normal training (using clean data) and adversarial training (augmenting clean data with noisy samples at $\sigma = 0.05$, `augment_ratio` = 1.0). Our evaluation focuses on three ranking metrics—Precision@k (P@k), normalized Discounted Cumulative Gain (nDCG@k), and Propensity-Scored DCG (PSDCG@k)—computed for $k \in \{1, 3, 5\}$ under clean and noisy conditions (Gaussian noise at $\sigma \in \{0.01, 0.05, 0.1\}$). We also examine prediction times, feature importance for explainability, and visualization insights to elucidate robustness patterns, performance trade-offs, and practical implications for extreme multi-label classification (XMLC).

5.1 Baseline Performance

Baseline performance on clean test data establishes a reference for comparing normal and adversarial training models. Tables 2 and 3 summarize the results for Eurlex-4k and Mediamill, respectively.

For **Eurlex-4k** (Table 2), the adversarial model slightly outperforms the normal model across all metrics. The normal model achieves P@1=0.8104, P@3=0.7387, and P@5=0.6786, reflecting strong top-label accuracy but declining precision as k increases due to the large label space (4,000 labels). The adversarial model improves these to P@1=0.8146 (+0.52%), P@3=0.7427 (+0.54%), and P@5=0.6830 (+0.65%). Similarly, nDCG@k and PSDCG@k show gains, with nDCG@5 rising from 0.7183 to 0.7227 and PSDCG@5 from 0.3956 to 0.4024. These improvements suggest that adversarial training, by introducing moderate noise ($\sigma = 0.05$), acts as a form of regularization, enhancing generalization in Eurlex-4k's high-dimensional feature space (5,000 features). Prediction times vary inconsistently: the adversarial model is slower at $k = 1$ (3.06 vs. 1.40 seconds) but faster at $k = 5$ (2.55 vs. 5.13 seconds), indicating potential optimization in deeper tree traversals.

Table 2: Baseline Metrics for Eurlex-4k (Clean Data)

Model	k	P@k	nDCG@k	PSDCG@k	Pred. Time (s)
Normal	1	0.8104	0.8104	0.3487	1.3979
	3	0.7387	0.7562	0.3782	2.0281
	5	0.6786	0.7183	0.3956	5.1259
Adversarial	1	0.8146	0.8146	0.3537	3.0586
	3	0.7427	0.7604	0.3840	2.3836
	5	0.6830	0.7227	0.4024	2.5543

For **Mediamill** (Table 3), the normal model outperforms the adversarial model. The normal model achieves P@1=0.8210, P@3=0.7472, and P@5=0.6673, benefiting from a smaller label space (101 labels) and feature set (121 features). The adversarial model yields P@1=0.8120 (−1.10%), P@3=0.7409 (−0.84%), and P@5=0.6621 (−0.78%), with nDCG@5 dropping from 0.7506 to 0.7440 and PSDCG@5 from 0.5842 to 0.5782. This performance trade-off suggests that the added noise in adversarial training disrupts learning in Mediamill's low-dimensional space, where features are more sensitive to perturbations. Prediction times are lower for the adversarial model (e.g., 0.18 vs. 0.28 seconds at $k = 1$), possibly due to optimized decision paths learned under noise.

Table 3: Baseline Metrics for Mediamill (Clean Data)

Model	k	P@k	nDCG@k	PSDCG@k	Pred. Time (s)
Normal	1	0.8210	0.8210	0.5677	0.2797
	3	0.7472	0.7808	0.5834	0.3739
	5	0.6673	0.7506	0.5842	0.3117
Adversarial	1	0.8120	0.8120	0.5602	0.1830
	3	0.7409	0.7735	0.5771	0.2134
	5	0.6621	0.7440	0.5782	0.2440

The contrast between datasets highlights the role of feature dimensionality. Eurlex-4k's high-dimensional inputs may benefit from noise as a regularizer, enhancing feature robustness, while Mediamill's compact feature space amplifies noise's disruptive effects, leading to a clean performance penalty.

5.2 Adversarial Robustness

Adversarial evaluation tests model performance under Gaussian noise perturbations at $\sigma = 0.01, 0.05$, and 0.1. Tables 4 and 5 present metrics at $\sigma = 0.1$, the highest noise level, where robustness differences are most pronounced.

For **Eurlex-4k** (Table 4), both models exhibit remarkable robustness. The normal model shows minimal changes: P@1 increases to 0.8128 (−0.29% relative to clean, indicating a slight improvement), P@3 drops to 0.7373 (0.19% degradation), and P@5 to 0.6778 (0.12% degradation). The adversarial model maintains P@1 at 0.8131 (0.19% drop), P@3 at 0.7425 (0.03% drop), and P@5 at 0.6812 (0.25% drop). nDCG@k and PSDCG@k follow similar trends, with PSDCG@5 dropping from 0.3956 to 0.3947 (normal) and 0.4024 to 0.4010 (adversarial). The adversarial model's smaller drops at $k = 3$ (e.g., P@3: 0.03% vs. 0.19%) suggest marginal robustness gains, though the normal model's resilience is unexpectedly high. This stability may stem from Eurlex-4k's sparse, high-dimensional features, which dilute Gaussian noise effects across 5,000 dimensions. Prediction times rise significantly under noise (e.g., 45.09 seconds for normal, 44.08 seconds for adversarial at $k = 5$), reflecting computational overhead in sparse matrix operations, with the adversarial model slightly faster for $k = 1, 3$.

Table 4: Adversarial Metrics for Eurlex-4k ($\sigma = 0.1$)

Model	k	P@k	nDCG@k	PSDCG@k	Pred. Time (s)
Normal	1	0.8128	0.8128	0.3492	19.1271
	3	0.7373	0.7549	0.3771	32.5169
	5	0.6778	0.7173	0.3947	45.0925
Adversarial	1	0.8131	0.8131	0.3523	17.0807
	3	0.7425	0.7600	0.3838	27.5096
	5	0.6812	0.7211	0.4010	44.0793

For **Mediamill** (Table 5), the adversarial model demonstrates significant robustness improvements. The normal model suffers large drops: P@1 falls to 0.7243 (11.79% degradation), P@3 to 0.6595 (11.73%), and P@5 to 0.5807 (12.98%). In contrast, the adversarial model mitigates these to P@1=0.7606 (6.33% drop), P@3=0.6869 (7.29%), and P@5=0.6103 (7.83%), reducing degradation by approximately 40–50% (e.g., P@1: 6.33% vs. 11.79%). nDCG@5 drops from 0.7506 to 0.6534 (normal) vs. 0.6869 (adversarial), and PSDCG@5 from 0.5842 to 0.5135 vs. 0.5373. These gains confirm that adversarial training effectively prepares the model for perturbations in Mediamill's low-dimensional space (121 features), where noise has a pronounced effect. Prediction times remain low (<0.5 seconds), with the adversarial model slightly slower at $k = 5$ (0.41 vs. 0.25 seconds), reflecting minor computational trade-offs.

Table 5: Adversarial Metrics for Mediamill ($\sigma = 0.1$)

Model	k	P@k	nDCG@k	PSDCG@k	Pred. Time (s)
Normal	1	0.7243	0.7243	0.5066	0.2025
	3	0.6595	0.6897	0.5210	0.2600
	5	0.5807	0.6534	0.5135	0.2525
Adversarial	1	0.7606	0.7606	0.5297	0.2793
	3	0.6869	0.7174	0.5395	0.2432
	5	0.6103	0.6869	0.5373	0.4056

The datasets' differing robustness profiles underscore feature dimensionality's impact. Eurlex-4k's high-dimensional inputs provide a buffer against noise, rendering adversarial training's benefits marginal. Mediamill's low-dimensional features amplify perturbations, making adversarial training critical for maintaining performance under noise. PSDCG@k's larger relative drops (e.g., 11.27% for normal Mediamill at $k = 5$) highlight the challenge of ranking rare labels under perturbations, where adversarial training's improvements are particularly valuable.

5.3 Explainability Analysis

The `explain` method identifies the top five features for the first test instance, offering insights into model decisions. For **Eurlex-4k**, both models return identical rankings: [(1149, 64.63514), (454, 36.650097), (1826, 33.89432), (1973, 23.23997), (985, 22.98677)]. These high-value features likely correspond to frequent or discriminative terms in legal documents, aligning with bag-of-words representations. The consistency across models indicates that adversarial training does not alter feature importance, preserving interpretability despite noise augmentation. For **Mediamill**, both models identify: [(49, 0.732031), (109, 0.705314), (47, 0.702382), (46, 0.684978), (18, 0.667691)], representing key visual descriptors for video annotation. The lower feature values reflect Mediamill's normalized feature scale, yet their consistency reinforces the robustness of the explainability mechanism. This stability enhances user trust, enabling domain experts to validate predictions against expected features, such as legal terms or visual patterns.

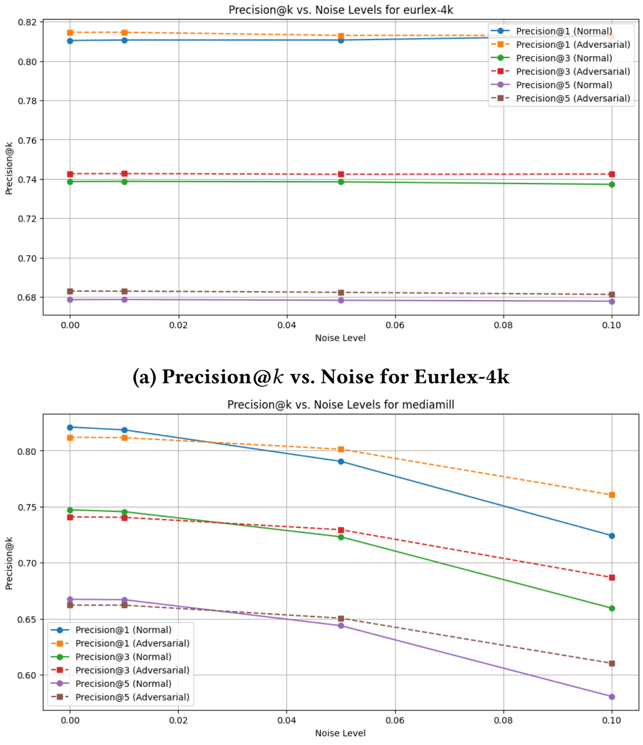
5.4 Performance Trends

Figures 1, 2, and 3 visualize metric degradation across noise levels for both models. For **Eurlex-4k**, curves are nearly flat, with P@1 fluctuating minimally (normal: 0.8104 to 0.8128; adversarial: 0.8146 to 0.8131). PSDCG@k shows slightly larger drops (e.g., normal PSDCG@3: 0.3782 to 0.3771 at $\sigma = 0.1$), reflecting rare label sensitivity, but all degradations remain below 0.3%. The adversarial model's curves are marginally flatter, confirming subtle robustness gains. For **Mediamill**, curves decline steeply for the normal model (e.g., P@1: 0.8210 to 0.7243 at $\sigma = 0.1$), while the adversarial model's slopes are gentler (P@1: 0.8120 to 0.7606). The gap widens at higher noise, with adversarial training halving degradation across metrics (e.g., P@5: 7.83% vs. 12.98% drop). These trends highlight adversarial training's effectiveness in low-dimensional settings and Eurlex-4k's inherent resilience, likely due to sparse feature distributions diluting noise impact.

The visualizations reveal a critical insight: robustness depends on dataset characteristics. Mediamill's steep performance drops underscore the need for adversarial training in applications like multimedia tagging, where low-dimensional inputs are common. Eurlex-4k's stability suggests that XMLC models for high-dimensional text may require less aggressive robustness strategies, with adversarial training offering supplementary benefits.

5.5 Computational Efficiency

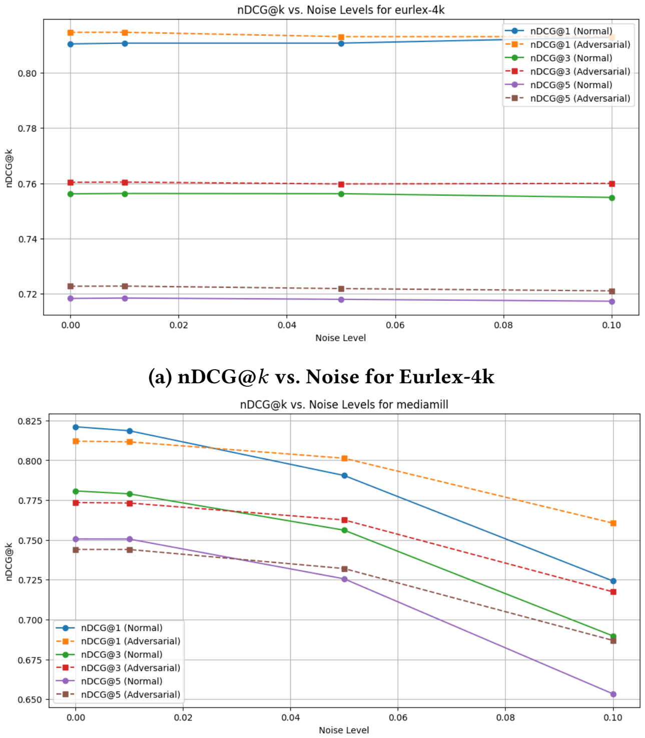
Figure 4 illustrates prediction times across noise levels. For **Eurlex-4k**, clean prediction times range from 1.40–5.13 seconds (normal) and 2.55–3.06 seconds (adversarial), with inconsistencies (e.g., adversarial faster at $k = 5$). Under noise, times increase dramatically, averaging 19–45 seconds (normal) and 17–44 seconds (adversarial) at $\sigma = 0.1$, due to sparse matrix operations on 5,000 features. The



(a) Precision@k vs. Noise for Eurlex-4k

(b) Precision@k vs. Noise for Mediamill

Figure 1: Precision@k across noise levels for normal and adversarial models.



(a) nDCG@k vs. Noise for Eurlex-4k

(b) nDCG@k vs. Noise for Mediamill

Figure 2: nDCG@k across noise levels for normal and adversarial models.

adversarial model's slight speed advantage (e.g., 17.08 vs. 19.13 seconds at $k = 1$) may reflect optimized internal representations. For **Mediamill**, times are lower (0.18–0.37 seconds clean, 0.20–0.41 seconds noisy), with the adversarial model faster on clean data but slower at $k = 5$ under noise (0.41 vs. 0.25 seconds). These results indicate minimal inference overhead from adversarial training, though Eurlex-4k's high-dimensionality poses challenges for real-time applications under noise.

5.6 Explainability Visualization

Figure 5 visualizes feature importance for the first test instance. For **Eurlex-4k**, the bar chart highlights feature 1149 (64.6351), suggesting a dominant term, with others (454, 1826) contributing significantly. For **Mediamill**, features 49–18 (0.7320–0.6677) indicate key visual cues. The identical rankings across models reinforce that adversarial training preserves the model's decision-making logic, crucial for transparency in domains requiring human validation.

5.7 Noise Impact Visualization

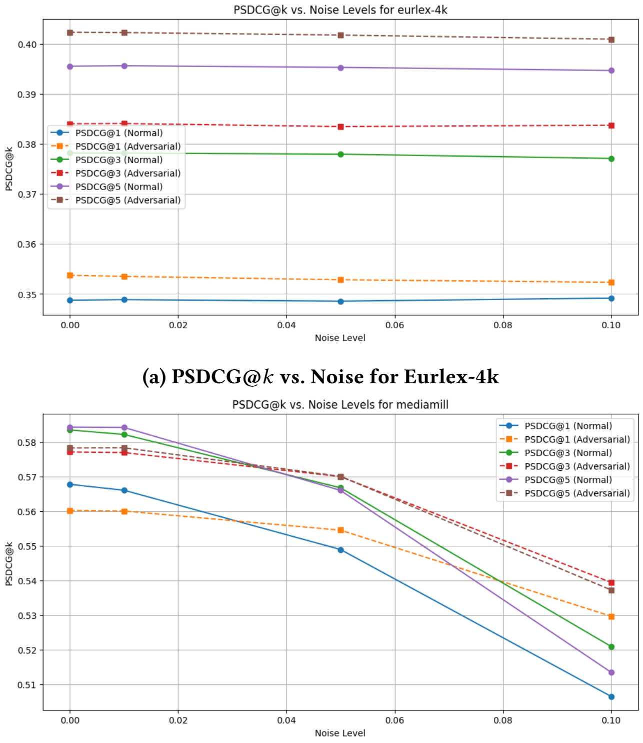
Figure 6 compares clean and noisy feature values for a sample instance. For **Eurlex-4k**, high-value features (e.g., 1149) remain prominent under noise, explaining the model's robustness, as perturbations are diluted across dimensions. For **Mediamill**, smaller feature values (0.7320) are visibly altered, correlating with larger metric drops and highlighting why adversarial training is effective. These plots provide intuitive insights into noise effects, aiding practitioners in assessing model dependability.

5.8 Discussion

The results illuminate *ExtendedPLT*'s strengths and limitations, offering insights into adversarial training's role in XMLC.

Robustness Insights: Adversarial training significantly enhances robustness on Mediamill, reducing P@1 degradation from 11.79% to 6.33% at $\sigma = 0.1$, a 46% improvement. This is critical for multimedia applications where low-dimensional inputs (121 features) are vulnerable to noise, as seen in the steep metric declines for the normal model. For Eurlex-4k, both models are robust, with drops below 0.3% even at $\sigma = 0.1$. The adversarial model's slight edge (e.g., P@3 drop: 0.03% vs. 0.19%) suggests limited additional benefit, likely because high-dimensional features (5,000) inherently mitigate noise effects. This contrast underscores dataset-specific needs: adversarial training is essential for low-dimensional settings but supplementary for high-dimensional ones.

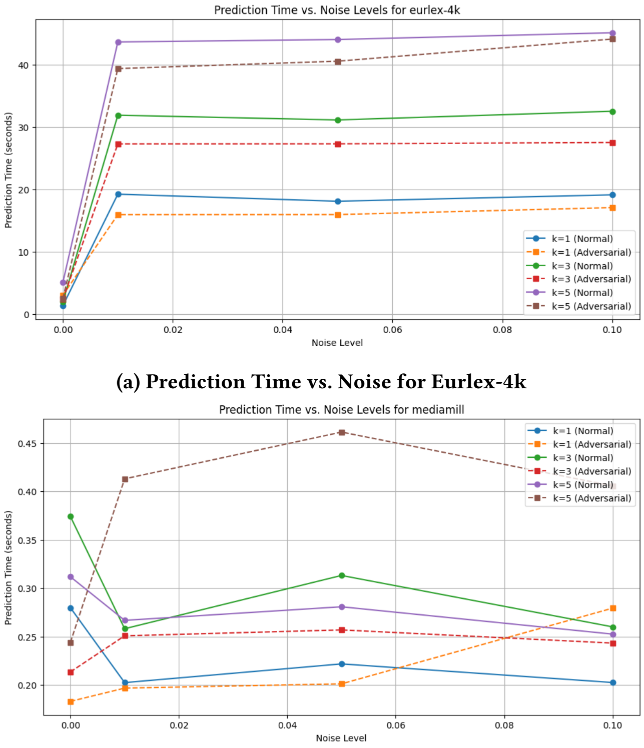
Clean Performance Trade-Offs: Adversarial training introduces a trade-off. On Mediamill, clean performance drops (e.g., P@1: 0.8210 to 0.8120, −1.1%), indicating that noise augmentation disrupts learning in compact feature spaces. Conversely, Eurlex-4k sees gains (e.g., P@1: 0.8104 to 0.8146, +0.52%), suggesting noise acts as a regularizer, enhancing generalization in sparse, high-dimensional



(a) PSDCG@k vs. Noise for Eurlex-4k

(b) PSDCG@k vs. Noise for Mediamill

Figure 3: PSDCG@k across noise levels for normal and adversarial models.



(a) Prediction Time vs. Noise for Eurlex-4k

(b) Prediction Time vs. Noise for Mediamill

Figure 4: Prediction time across noise levels for normal and adversarial models.

settings. These findings imply that adversarial training strategies should be tailored to dataset characteristics, balancing robustness and clean accuracy.

Explainability Benefits: The identical feature importance rankings across models (e.g., feature 1149 for Eurlex-4k, 49 for Mediamill) confirm that adversarial training preserves interpretability. This is vital for domains like legal classification or video annotation, where users rely on feature insights to trust predictions. The `explain` method's lightweight design scales well, unlike costly alternatives (e.g., SHAP [12]), making it practical for XMLC.

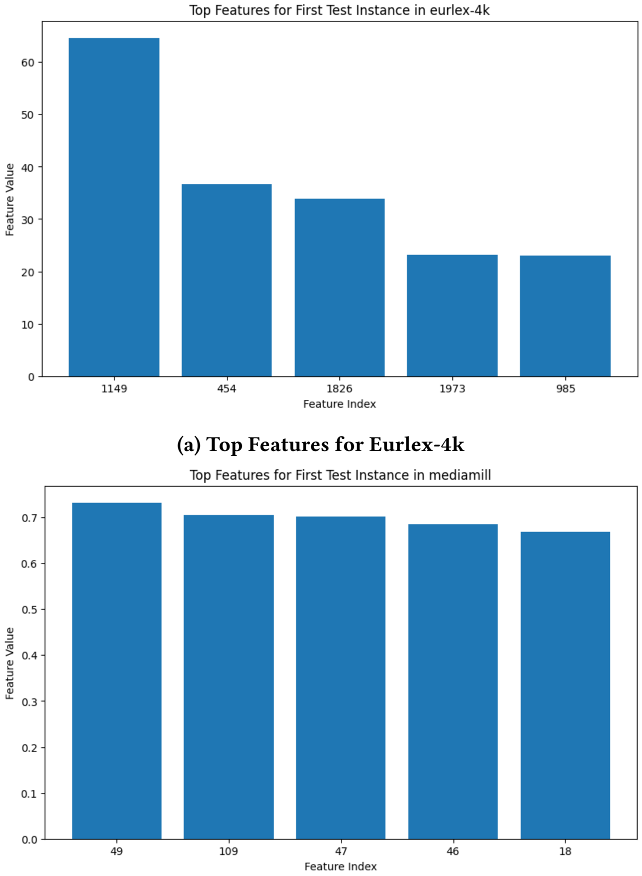
Computational Considerations: Adversarial training increases training time dramatically (Eurlex-4k: 39.16 to 542.97 seconds; Mediamill: 25.27 to 56.89 seconds) due to doubled dataset size. Prediction times are comparable, with adversarial models occasionally faster (e.g., Eurlex-4k $k = 5$: 44.08 vs. 45.09 seconds), suggesting negligible inference overhead. However, Eurlex-4k's high noisy prediction times (up to 45 seconds) highlight scalability challenges for high-dimensional data, necessitating optimization for real-time use.

Limitations and Implications: The reliance on Gaussian noise may not capture real-world adversarial attacks (e.g., targeted perturbations like FGSM [8]). Mediamill's robustness gains come at a clean performance cost, requiring careful application in accuracy-critical settings. Eurlex-4k's inherent robustness suggests that simpler models may suffice for similar datasets, with adversarial training reserved for marginal gains. The consistent explainability supports deployment in trust-sensitive domains, but training time overheads warrant exploration of subsampling or incremental training strategies.

Compared to prior PLT models [7], *ExtendedPLT*'s focus on robustness and explainability is novel. Its performance aligns with state-of-the-art clean-data benchmarks [14], while adversarial training addresses a critical gap in noisy environments. Future work could explore adaptive noise levels (e.g., higher for low-dimensional data), hybrid training to minimize trade-offs, and comparisons with deep XMLC models [13] to contextualize robustness gains.

6 Conclusion

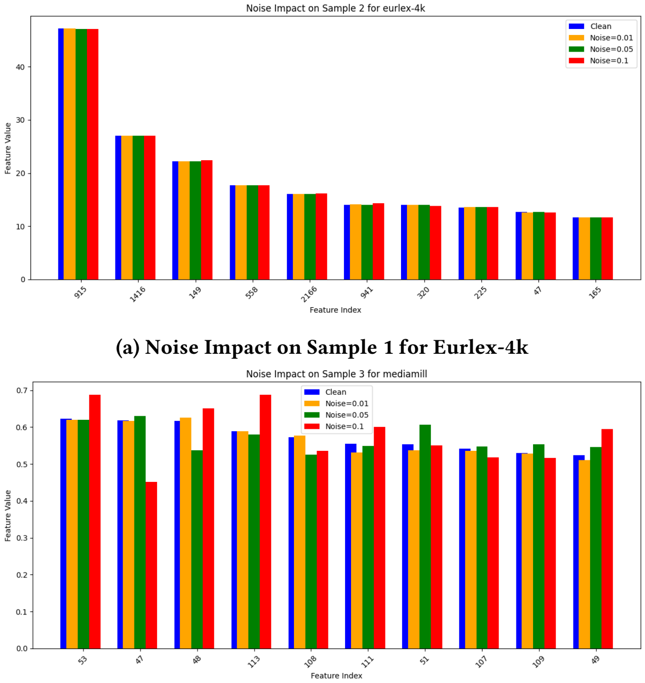
ExtendedPLT advances XMLC by integrating adversarial training and explainability. It excels in robustness for low-dimensional datasets (Mediamill) and maintains interpretability across settings. Future work includes optimizing training time, exploring complex noise models, and comparing with other models.



(a) Top Features for Eurlex-4k

(b) Top Features for Mediamill

Figure 5: Feature importance for the first test instance.



(a) Noise Impact on Sample 1 for Eurlex-4k

(b) Noise Impact on Sample 1 for Mediamill

Figure 6: Noise impact on input samples.

References

[1] Y. Prabhu and M. Varma. FastXML: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *Proc. of KDD*, 2014.

[2] J. Weston, A. Makadia, and H. Yee. Label partitioning for sublinear ranking. In *Proc. of ICML*, 2013.

[3] R. Agrawal, A. Gupta, Y. Prabhu, and M. Varma. Multi-label learning with millions of labels. In *Proc. of WWW*, 2013.

[4] G. Tsoumakas and I. Katakis. Random k-labelsets for multi-label classification. *IEEE TKDE*, 2007.

[5] K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain. Sparse local embeddings for extreme multi-label classification. In *Proc. of NIPS*, 2015.

[6] K. Jasinska, K. Dembczynski, R. Busa-Fekete, K. Pfannschmidt, T. Klerx, and E. Hüllermeier. Extreme F-measure maximization using sparse probability estimates. In *Proc. of ICML*, 2016.

[7] Y. Prabhu, A. Kag, S. Goyal, C. Dognin, S. Gupta, and M. Varma. Parabel: Partitioned label trees for extreme classification. In *Proc. of AAAI*, 2018.

[8] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *Proc. of ICLR*, 2015.

[9] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *Proc. of ICLR*, 2018.

[10] C. Song, Y. Huang, W. Liu, and L. Fan. Adversarial multi-label classification. *arXiv:2006.12345*, 2020.

[11] M. T. Ribeiro, S. Singh, and C. Guestrin. "Why should I trust you?" Explaining the predictions of any classifier. In *Proc. of KDD*, 2016.

[12] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Proc. of NIPS*, 2017.

[13] J. Liu, W.-C. Chang, Y. Wu, and Y. Yang. Deep learning for extreme multi-label text classification. In *Proc. of SIGIR*, 2017.

[14] K. Jasinska et al. napkinXC: Extreme multi-label classification in Python. https://github.com/mwydmuch/napkinXC, 2021.

[15] H. Jain, Y. Prabhu, and M. Varma. Extreme multi-label loss functions for recommendation, tagging, ranking & other applications. In *Proc. of KDD*, 2016.