# LVForge: PE Malware Detection

## Transformer + Deep Metric Learning

Ly Ngoc Vu     Dang Thi Phuc

Industrial University of Ho Chi Minh City

## Problem & Motivation

- Windows PE malware detection needs **high recall** at **low false-positive rate**.
- Standard accuracy alone is insufficient for deployment.
- Class imbalance (benign:malware $\approx 1 : 19$) makes threshold behavior critical.

**Goal:** build an operationally robust detector using a shared Transformer and compare objective functions.

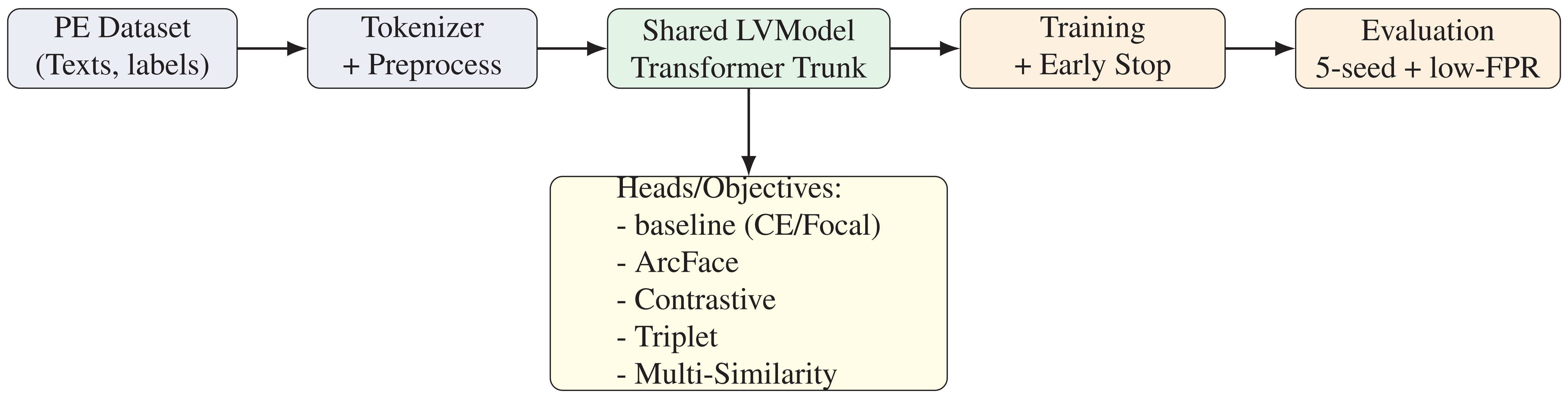
## Contributions

- Unified Flax/JAX pipeline for baseline + DML variants.
- Controlled comparison of `baseline`, `arcface`, `contrastive`, `triplet`, `multi_similarity`.
- Multi-seed evaluation with operational metric: $\mathrm{TPR@FPR} = 10^{-2}$.
- Architecture-level extension of LVModel with metric-learning heads.

## Data & Setup

- Dataset size: **34,370** PE samples.
- Input: text-like PE features, labels {benign, malware}.
- Backbone config: $d_{model} = 256$, heads$= 8$, FFN$= 512$, layers$= 2$, max seq len$= 380$.
- Training: batch$= 128$, epochs$= 5$, LR$= 2 \times 10^{-4}$, dropout$= 0.1$.

## Unified Pipeline




## LVModel Architecture Details

**Base LVModel:**

- Input shape: $(B, T, d)$; token + positional embedding: $\mathbf{H}_0 = E_{tok}(\mathbf{X}) + E_{pos}(1{:}T)$.
- MHA uses one combined QKV projection: $\mathrm{QKV} = W_{qkv}\mathbf{H}$, reshaped to $Q, K, V$ by heads.
- Scaled dot-product attention: $\mathrm{softmax}(QK^\top/\sqrt{d_h})$, then output projection.
- **Pre-norm** encoder layer: $x \leftarrow x + \mathrm{MHA}(\mathrm{LN}(x))$, $x \leftarrow x + \mathrm{FFN}(\mathrm{LN}(x))$.
- LVModel head: mean-pool $\rightarrow$ dense+tanh $\rightarrow$ dropout $\rightarrow$ LN $\rightarrow$ classifier logits.

**Metric extensions:**

- Shared trunk $\rightarrow$ projection to $d_{emb} = 256$ $\rightarrow$ LayerNorm + $\ell_2$-norm embedding.
- $n = 0.5, s = 64$).
- Contrastive/Triplet/MS: linear logits + auxiliary metric losses.

## Main Results (5-seed mean)

| Variant | Accuracy | F1 | ROC-AUC | PR-AUC | TPR@FPR=1e-2 |
|---|---|---|---|---|---|
| baseline | 0.9924 | 0.9960 | 0.9983 | 0.9999 | 0.9754 |
| arcface | 0.7979 | 0.7934 | 0.9704 | 0.9984 | 0.0000 |
| contrastive | 0.9931 | 0.9964 | 0.9971 | 0.9997 | 0.9533 |
| triplet | 0.9932 | 0.9964 | 0.9969 | 0.9998 | 0.9351 |
| multi_similarity | **0.9946** | **0.9972** | 0.9978 | 0.9999 | **0.9851** |

**Runtime (single run):** baseline 107.1s, arcface 93.6s, contrastive 130.8s, triplet 132.6s, multi_similarity 128.3s.

## Discussion & Takeaways

- DML improves performance, but **objective selection matters**.
- **Multi-Similarity** is the best overall operating point.
- Baseline remains a strong competitor.
- ArcFace is unstable at strict low-FPR thresholds in this setup.

**Deployment recommendation:** Use Multi-Similarity as primary model, base-

## Reproducibility

Code and paper assets are in: `/root/LVForge/docs/paper/`
Main paper: `IEEE-conference-template-062824/IEEE-conference-template-062824.tex`
All variants executed via: `scripts/run_all.py` with recorded logs and aggregated JSON metrics.