Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

PhishJEPA – a Transferable Phishing Email-Related Representation Learning

Tandin Wangchuk

tandinwangchuk.cst@rub.edu.bt

Sophia University https://orcid.org/0000-0002-5167-7228

Karma Wangchuk

Sophia University

Tad Gonsalves

Sophia University

Research Article

Keywords: Phishing email detection, Joint Embedding Predictive Architecture (JEPA), Self-supervised learning, Cross-corpus generalization, Distribution shift, Isotonic calibaration

Posted Date: November 2nd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11120622/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

PhishJEPA – a Transferable Phishing Email-Related Representation Learning

Tandin Wangchuk[1,2*], Karma Wangchuk[1,2†] and Tad Gonsalves[1†]

[1*]Department of Information and Communication Sciences, Faculty of Science and Technology, Sophia University, 7-1 Kioi-cho, Chiyoda-ku, 102-8554, Tokyo, Japan.

[2]Computing Technologies Department, College of Science and Technology, Rinchending, Phuentsholing, 21001, Chukha, Bhutan.

*Corresponding author(s). E-mail(s): tandin@eagle.sophia.ac.jp;
Contributing authors: karma@eagle.sophia.ac.jp; t-gonsal@sophia.ac.jp;
†These authors contributed equally to this work.

Abstract

Phishing is a primary initial vector for data breaches and results in high global costs. While deep learning models achieve near-perfect in-domain detection, cross-corpus distribution shifts severely degrade performance owing to overfitting on dataset-specific artifacts instead of learning semantic intent. To address this, we introduce PhishJEPA, a self-supervised Joint Embedding Predictive Architecture using Exponential Moving Average student-teacher updates and Variance-Invariance-Covariance Regularization to model email subject-body semantic relationships in latent space. We systematically evaluate representation transferability across four unseen benchmark corpora in two stages over multiple random seeds. In Stage 1 (feature probing), JEPA representations demonstrate statistically robust improvement in cross-corpus separability over vanilla DistilBERT under a linear probe (+0.0124 pooled AUROC, $\boldsymbol{p \approx 9 \times 10^{-11}}$), whereas multilayer perceptron probing ($\boldsymbol{p \approx 0.93}$)and unsupervised domain adaptation (-0.0039 AUROC, $\boldsymbol{p \approx 1 \times 10^{-9}}$) offer no gains. In Stage 2 (downstream classification), freezing the JEPA backbone and training a lightweight linear head achieves a cross-corpus performance (0.9394 AUROC / 0.9157 AUPRC) comparable to full end-to-end DistilBERT fine-tuning (0.9496 AUROC), while reducing the training time by 60% and peak-allocated GPU memory by 43%. Crucially, we identify an operational capacity tradeoff: higher-capacity MLP heads overfit in-domain (0.9980 AUROC) and suffer larger cross-corpus performance drops (7.24%-19.20%) than linear heads (5.19%-10.63%), proving far

less threshold stability. Finally, post-processing per-domain isotonic calibration reduces the fixed false-positive rates by up to 16.7% across training sources without degrading ranking performance. These findings indicate that JEPA learns transferable semantic representations, but reliable deployment under distribution shifts depends critically on a simple classifier design and post-hoc calibration.

Keywords: Phishing email detection, Joint Embedding Predictive Architecture (JEPA), Self-supervised learning, Cross-corpus generalization, Distribution shift, Isotonic calibaration

1 Introduction

Phishing continues to be a prevalent cybersecurity threat [1, 2] and a common initial attack vector for gaining access to targeted environments [2–4]. The APWG [5] reported an increase in phishing from 853,244 in Q4 2025 to 971,181 in Q1 2026, a 13.8% increase in a quarter. Similarly, Comcast Business [2] indicated a similar trend, detecting and blocking 4.7 billion phishing events. Phishing attacks are often delivered via email and can lead to serious security incidents. IBM [3] found that global data breaches incurred an average cost of USD 4.99 million, an increase of 12% from the previous year. Business Email Compromise (BEC) alone account for USD 3 billion in the reported US losses in 2025 [1]. As deceptive communication, disguised as legitimate emails, becomes increasingly sophisticated, organization-level defenses require robust automated detection.

Combating phishing attacks has transitioned from static lists [6] and rule-based [7] filters to authentication protocols (such as the Sender Policy Framework (SPF), DomainKeys Identified Mail (DKIM), and DMARC) and machine-learning approaches [8, 9]. However, static lists suffer from high false-negative rates on novel domains [6]. Furthermore, authentication protocols are frequently misconfigured by domain owners [10], and dynamic Secure Email Gateway (SEG) sandboxing introduces noticeable execution latency [11]. Consequently, machine learning and deep natural language processing (NLP) [12], including BERT [13], DistilBERT [14], and hybrid transformer models [15], have been widely adopted to infer semantic intent, such as credential theft or artificial urgency. Moreover, deep-learning approaches [15–17] address the challenges of slow manual list updates [6] and the requirements of domain experts in handcrafted feature engineering. The dependency of models on large, labelled datasets creates another challenge that requires laborious dataset labelling and is prone to human oversights [18]. As an alternative to training models on labelled datasets, self-supervised learning is a common pretraining method used for training most LLMs on raw corpora [19].

Typical evaluation practices focus on a single benchmark dataset and achieve nearly perfect accuracy under the same distribution, such as in [16, 17, 20]. However, during real-world deployment, models are often sensitive to changes in data distribution, including new or modified phishing emails [21–23]. Models must understand the nuances and intent of phishing emails and not just the keywords associated

with attacks and dataset-specific characteristics. To bridge the gap between the in-domain benchmark performance and real-world deployment reliability, we propose PhishJEPA, a self-supervised representation learning framework built on the Joint Embedding Predictive Architecture (JEPA) [24]. Drawing inspiration from world-model paradigms that predict abstract target representations in latent space rather than reconstructing surface-level pixels or tokens, PhishJEPA models the latent semantic alignment between an email's subject and body. Using the dual-encoder setup, the student encoder processes an email subject to predict the latent representation of the email body produced by the teacher encoder. To prevent trivial representation collapse without relying on contrastive negative pairs, training was optimized via Variance-Invariance-Covariance Regularization (VICReg) [25] while updating the teacher parameters through an Exponential Moving Average.

Furthermore, rather than positioning self-supervision solely as a tool for chasing incremental ranking metrics, this study systematically investigated the operational trade-offs of parameter frozen world model representations under cross-corpus distribution shifts. By freezing the pretrained JEPA backbone and fitting lightweight downstream classifiers, PhishJEPA provides a low-latency, computationally efficient screening layer suitable for integration into a multistage Secure Email Gateway (SEG) or Integrated Cloud Email Security (ICES) pipelines.

The present study addresses the main research question: Can a JEPA-pretrained subject–body representation support phishing detection that generalizes across email corpora under a distribution shift, and how does it compare with supervised alternatives? We decomposed this inquiry into four target sub-questions:

- **RQ1.1**: Does JEPA pretraining yield more transferable phishing representations than an identical backbone trained without self-supervision, as measured by cross-corpus performance under a fixed linear probe?
- **RQ1.2**: Can a frozen JEPA encoder with a lightweight classifier match or approach the cross-corpus performance of end-to-end fine-tuning and strong classical baselines?
- **RQ1.3**: How does classifier capacity (linear vs. MLP) trade cross-corpus ranking quality against operating point stability under a distribution shift?
- **RQ1.4**: Can per-domain calibration recover a reliable fixed-threshold operating point under a distribution shift without degrading the ranking performance?

The scope of PhishJEPA is explicitly defined as the detection of semantic intent and subject-body text discordance in human-authored emails. Multimodal signals (such as embedded URLs, image-based email, and header authentication) and generative adversarial or LLM-generated phishing vectors were excluded from this study and are considered for future work.

The key contributions of this study are summarized as follows:

1. We introduced **PhishJEPA**, applying a Joint Embedding Predictive Architecture with VICReg and EMA student-teacher updates to model email intent in latent space via subject-body semantic alignment without token-level reconstruction.

2. We established a cross-corpus evaluation framework spanning four unseen benchmark datasets (Enron, TREC_05, TREC_06, TREC_07, and CEAS_08) using paired statistical t-tests across multiple random seeds, showing JEPA linear probing outperforms raw DistilBERT (+0.0124 pooled AUROC, $p \approx 9 \times 10^{-11}$)
3. We identify a critical capacity-deployability trade-off: while higher-capacity MLP heads overfit in-domain (0.9980 AUROC) and suffer severe cross-corpus degradation (7.24%-19.20%), lightweight linear heads exhibit threshold stability, matching end-to-end fine-tuning (0.9394 vs. 0.9496 AUROC) while reducing training execution time by 60% (6.39 min vs. 15.69 min) and peak GPU memory by 43% (1.22 GB vs. 2.14 GB).
4. We demonstrate that post-processing per-domain isotonic calibration successfully recovers fixed-threshold operational stability at $\tau = 0.5$, reducing false-positive rates by up to 16.7% across training sources without altering ranking order.

The remainder of this paper is organized as follows. Section 2 discusses related work on email security and deep learning, followed by details of the PhishJEPA architecture, self-supervised loss objectives, and evaluation protocol in Section 3. Section 4 presents the empirical benchmark results, followed by discussion of the results, deployment implications, and limitations in Section 5, and the study's conclusion in Section 6.

2 Related work

The following section provides a chronological order of phishing email prevention approaches proposed in the literature.

The earliest approaches to combat phishing attacks relied on static rules and reputation. These methods use domain and URL blacklists or whitelists, keyword matching, and standard anti-spam filters, such as SpamAssassin. The filtering system checks whether a URL, IP address, or keyword is on a whitelist (allowed) or blacklist (disallowed) [26] which is effective at detecting known malicious URLs, IP addresses, and keywords. However, they have high false-negative rates for new domains and are vulnerable to newly created malicious links. Hence, lists must be updated quickly [6]. This has led to other approaches to combat phishing, as explained in the following paragraph.

Subsequently, authentication-based and signature-based approaches were explored as alternatives to static-list and rule-based approaches. Email security schemes such as the Sender Policy Framework (SPF) [27], DomainKeys Identified Mail (DKIM) [28], and Domain-based Message Authentication, Reporting, and Conformance (DMARC) [29] have been designed to authenticate senders and prevent direct domain spoofing [10]. The greatest weakness, according to a study by Hureau et al. [10], is that these schemes are poorly understood by domain owners and are not configured according to standards. Furthermore, the authors highlighted that genuine emails may be discarded by servers because of incorrect settings or stringent anti-spoofing mechanisms. Similarly, signature-based file scanning methods were used for email attachments [30], and basic computer vision was employed for logo matching, as in [31]. Signature-based file scanning is a standard in antivirus and email security software that helps detect

known malicious attachments, such as EXE files and macro-enabled documents. However, they cannot be used to detect unknown malicious attachments or malware that are not in the signature database [30]. Similarly, logo detection has limited capability if the trusted list is small.

Then came the era of machine learning and behavioral analysis. Traditional machine learning techniques, as discussed in [8, 32, 33], have been proposed for phishing detection. Furthermore, computer vision techniques, such as in [34, 35] were used for web User Interface (UI) clone detection. Machine learning methods, namely logistic regression, decision trees, naïve Bayes, and random forest, are lightweight and transparent for detecting phishing. However, these methods require domain experts for engineering handcrafted features and laborious effort for labelling datasets [18]. Simultaneously, secure email gateways (SEGs) based on signature and machine learning methods were employed to filter the emails using behavioral analysis. Dynamic sandboxing was used in the SEGs to test attachments in isolated environments and segregate malicious emails, as in a previous study [36]. SEGs are implemented to analyze sender history, communication anomalies, and payload execution, aiding in the early detection of Business Email Compromise (BEC). However, the drawbacks of these email gateways include high latency owing to sandbox delays, vulnerability to link wrapping, and deferred redirects [11].

The latest approaches to phishing email detection use deep learning, including NLP models [12](such as BERT, DistilBERT, Transformer architectures), and graph-based relational graph analysis [15]. These techniques analyze email content and context to infer intent, such as urgency or credential theft. Several studies [15, 16, 20, 23, 37, 38] have demonstrated that contextual models outperform shallow feature models. Atewneh [16] explored and compared several deep-learning architectures, such as CNNs, RNNS, LSTMs, and BERT, for suitable phishing email detection. The study found that the BERT+LSTM hybrid was the best, with an accuracy of 99.61%, precision of 99.87%, recall of 99.23%, and F1-score of 99.55%. The reported results were based on in-domain performance. Another study [15] used DistilBERT to extract features rather than manually extracting them and constructed an input to be fed into a Graph Attention Network (GAT) for learning. The hybrid scheme achieved an in-distribution test accuracy of 99.39% on the CEAS_08 benchmark dataset and reported a cross-validation (CV) accuracy of 99.10% on Nazario (imbalanced dataset). However, the CV indicated by the authors appeared to be a single-source training and evaluation of their model on the Nazario dataset. Although these studies achieved near-perfect performance metrics when trained on a single dataset, they suffered from severe performance degradation when evaluated across various datasets [21, 23, 39, 40]. In other words, most of these models failed to transfer representations when evaluated using unseen corpora.

Other industry solutions for phishing emails are API-based email security integrations, including the Integrated Cloud Email Security (ICES) solution. ICES integrates directly with cloud email platforms rather than relying solely on traditional secure email gateways (SEGs). Despite these advances, handling multistage attacks, image-only emails, and the associated computational costs continues to be an area of active engineering work [41].

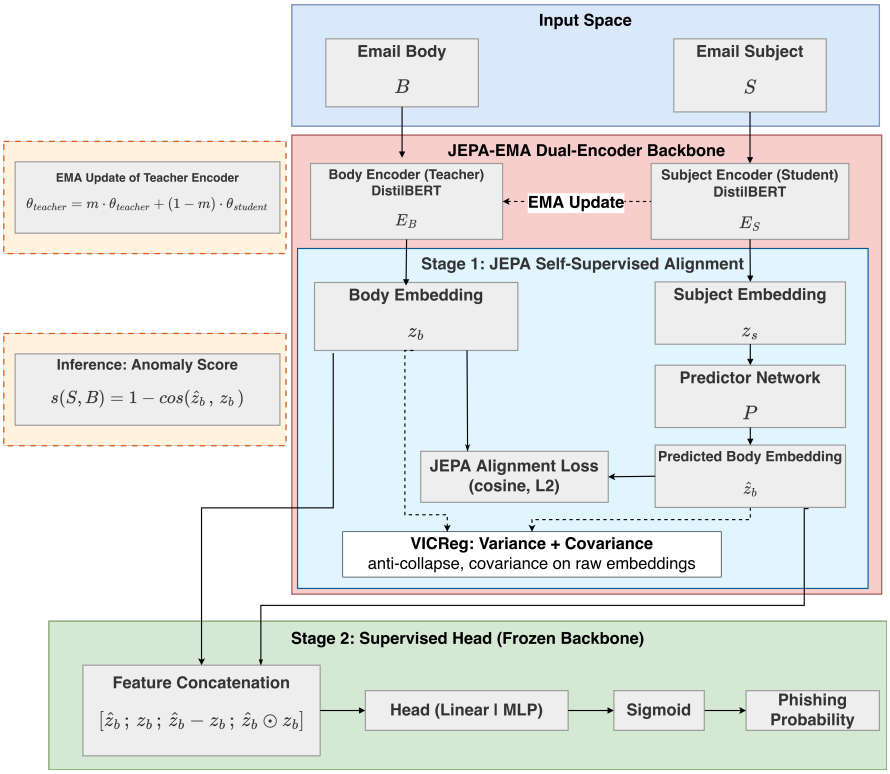


Fig. 1: PhishJEPA: Phishing email detection based on Joint Embedding Predictive Architecture with VICReg training optimization and EMA

This study presents a JEPA-based phishing detection framework that uses email subjects to predict email bodies in the latent space during self-supervised training. The framework was evaluated using linear and MLP probes across five unseen corpora. The probing results were later supported by the downstream classification of emails as phishing or non-phishing using a linear and MLP head.

3 Methodology

3.1 Proposed Framework

In JEPA [24], prediction is performed in the latent space rather than at the token level, which leaves out unnecessary details, thereby reducing the computation. In the proposed framework, as shown in Fig. 1, the teacher encoder encodes the tokenized **email body** to produce z_b embeddings, and the student encoder encodes the tokenized **subject** into z_s embeddings in parallel using separate encoders. The latent context embedding (encoded email subject) is then used to predict the semantic representation of the email body $\widehat{z}_b$ using a separate predictor network in the latent space. The training objective is to calculate the following loss,

$$L \;=\; cos_weight \;*\; L_{\cos} \;+\; var_weight \;*\; L_{var} + cov_weight \;*\; L_{cov}$$

Cosine loss (invariance loss) is the core predictive loss calculated using cosine distance between the student's guess (predicted embedding $\widehat{z}_b$) and the teacher's target (ground truth embedding z_b),

$$L_{\cos} = \; 1 \; - \; \mathbb{E}\left[\langle z_b \;\; , \;\; \widehat{z}_b \rangle\right]$$

The aim was to bring the predicted latent embeddings closer to the target embeddings. If the embeddings are aligned ($\theta \;=\; 0$), the cosine similarity is one, and the cosine loss is zero. However, if the embeddings are random, the cosine similarity is zero, and the cosine loss is one.

Representation collapse is a common problem in self-supervised learning, in which a neural network learns a trivial solution during training without learning meaningful features from data. Variance-Invariance-Covariance Regularization (VICReg) [25] is one of the solutions introduced to address this challenge. Variance loss, which is a part of VICReg, is the anti-collapse constraint that is calculated as

$$L_{var} = \; \sum \max(0, \; 1 \; - \; \sigma(z))$$

This forces each dimension to have a nonzero variance, which prevents $z \;\; = \;\; constant$ representation, thereby preventing norm collapse.

Covariance Loss decorrelates and penalizes correlated dimensions to increase information representation. This is calculated as follows:

$$L_{cov} = \; \sum_{i \;\neq\; j} Cov \; (z_i, \; z_j)^2$$

Cosine Loss, or invariance loss, is the core objective, whereas variance loss and covariance loss act as regularizer.

The student network is updated through backpropagation, whereas the teacher network is updated using an Exponential Moving Average (EMA) of the student parameters smoothly over time, as follows:

$$\theta_{teacher} \;\; = \;\; m \;\; * \theta_{teacher} \;\; + \;\; (1 \;\; - \;\; m) * \theta_{student}$$

Where:

- $\theta_{teacher}$: teacher weights
- $\theta_{student}$: student weights
- m: momentum (e.g., 0.99–0.999)

This provides a stable and slow-evolving target representation that improves convergence and prevents the collapse of the temporal representation. VICReg and EMA provide a means of preventing representation collapse in the proposed framework.

In Stage 1, with self-supervised training on the unlabeled Enron corpus, we assessed the performance of our model on benign-only data using anomaly score distributions rather than classification metrics. Specifically, the mean, standard deviation, and percentile statistics of the cosine-based anomaly scores are reported. These metrics provide information regarding learned normal data and an estimate of the score threshold that separates typical observations from rare observations.

In the intermediate stage, the model was trained through unsupervised domain adaptation using an auxiliary contemporary dataset built from [42]. This was performed to adapt the learned semantic space to new email distributions without labels. The same setup as that in Stage 1 was used during this stage, where modern benign emails were passed through encoders and optionally used to compute the loss. The student weights were updated through gradients, and the teacher weights were updated via the EMA. The embedding space shifted to reflect modern language, corporate tone variations, and formatting patterns; however, the semantic structure remained unaltered.

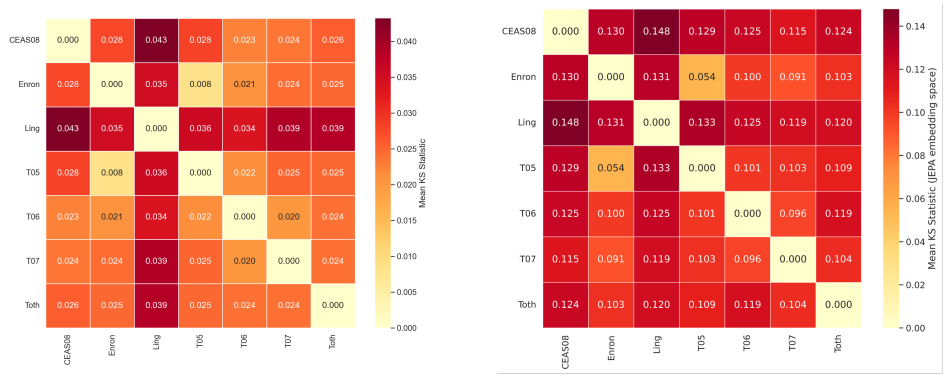
In Stage 2, we trained a lightweight classification head on top of the frozen JEPA embeddings through supervised learning using the TREC_05, TREC_06, and TREC_07 datasets. During this stage, the model learns to classify emails as phishing or benign using fixed semantic representations. The discordance and product signals were constructed from the predicted embedding and target embedding during the forward pass. These feature embeddings were then concatenated and passed through an MLP classifier trained using binary cross-entropy. This detects inconsistencies, understands intent, and distinguishes benign signals from phishing signals. The classification head then used the concatenated features to classify the prediction as 1 for phishing and 0 for non-phishing. The gradients flowed only through the MLP classification head, whereas the encoders and predictors were frozen. Freezing the encoders prevents leakage, preserves semantic space, and enables domain adaptation, thereby improving robustness under a distribution shift.

3.2 Experiment Setup

Two stages of experiments were conducted using an NVIDIA GeForce RTX 3090 GPU. In Stage 1, the pretrained JEPA embeddings were assessed with linear and MLP probes across four unseen corpora, and performance metrics, such as AUROC, AUPRC, and FPR@95% TPR, were computed over five seeds. The JEPA backbone was then used for comparisons between the probes and vanilla DistilBERT. The backbone was also used for domain adaptation to an auxiliary modern email corpus. This experiment was conducted to evaluate the separability of the representations. In Stage 2, the JEPA backbone is used to fine-tune a linear or MLP classifier for downstream classification. During fine-tuning, BCE with class weights was used. The fine-tuned heads were evaluated across the same four unseen corpora using the same metrics over five seeds and were compared with the following baselines. Baseline A was the backbone used for anomaly detection. Baseline B was the vanilla DistilBERT feed with only the email subject, whereas Baseline C was the vanilla DistilBERT feed with only the email body. Baseline D was a dual-stream classifier built cleanly from vanilla DistilBERT. Baseline E was a shallow statistical n-gram tracking baseline using TF-IDF. Baseline Fa

Table 1: Summary of 8 benchmark email datasets used in the evaluation

Datasets	Phishing	Non-phishing	Total	Year	Citation
AVN_Basic	18515	15565	34080	2010–2020	[43]
Ling	458	2401	2859	Early 2000	[32, 44]
Enron	13976	15791	29767	1999–2002	[44, 45]
TREC_05	21494	30334	51828	2005	[44, 46]
TREC_06	3989	12392	16381	2006	[44, 47]
TREC_07	28931	24321	53252	2007	[44, 48]
toth_email	2125	950	3075	2025	[42]
CEAS_08	21842	17312	39154	2008	[44]



(a) (b)

Fig. 2: Drift among the datasets (a) Linguistic drift heatmap (Mean KS), and (b) Representation drift heatmap (JEPA embedding, Mean KS)

was a classification baseline based on FastText, and Baseline Fb was a full end-to-end supervised fine-tuned classifier. A t-test was performed for comparison. Finally, calibration of the metrics was conducted to test the deployability of the model.

3.3 Datasets

The benchmark datasets (refer to Table 1) available in the public domain were used in this study. Before the experiment, a drift analysis was performed to determine whether there was a drift between the datasets. The drift analysis included linguistic drift (Figure 2a), representation drift (Figure 2b), and email length analysis (Figure 3).

The Ling dataset had the largest drift among the datasets for both linguistic (0.034–0.043) and representation (0.119–0.148). Enron and TREC_05 exhibited the lowest drift for both linguistic (0.008) and representation (0.054). Although both linguistic and representation drift were comparatively small (0.08 – 0.148), representation drift (0.054 – 0.148) was slightly larger than linguistic drift (0.008 – 0.043). In terms of email length, most of the corpus lay between approximately 5 and 8.5 on the log

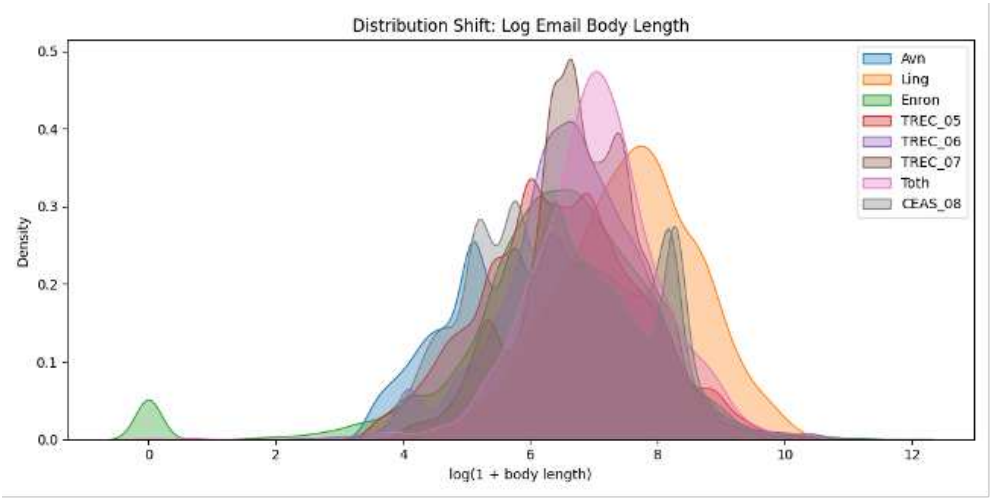


Fig. 3: Email length distribution among the datasets

scale, and the corpora overlapped substantially. Ling had the rightmost distribution and tended to have longer emails than other corpora. AVN had the broadest distribution and contained a mixture of both long and short emails. Enron had the widest spread and contained the leftmost spikes at 0. This indicates the presence of short or nearly empty emails. TREC_05 and TREC_07 exhibited similar email-length distributions. TREC_07 had the narrowest and most concentrated distribution. The TREC corpora exhibited remarkable consistency. Toth had a narrower distribution and was more concentrated than Ling but was slightly left-shifted relative to the TREC corpora. CEAS_08 exhibited a multipeak distribution and was left-shifted relative to those of Ling and Toth.

4 Results

4.1 Self-supervised training, Domain Adaptation and Fine-tuning

Self-supervised learning ran smoothly on the unlabeled Enron corpus. Table 2 shows that the subject-body consistency scores were mostly concentrated around a mean score of 0.5695, with an average cosine similarity score of 0.43. Only 5% of the emails scored above 0.8440, indicating an absence of collapse. Overall, the score distribution showed meaningful variance and moderate alignment without collapse, with a smooth training loss curve (Figure 4).

In addition, no collapse occurred during the domain adaptation process using the auxiliary corpus, as shown in Table 3. The same loss objective was pursued during domain adaptation, and a lower learning rate was assigned to prevent catastrophic forgetting.

Table 2: Summary statistics of the score distribution

Metric	Value
Total Sample	46010
Mean Score	0.5696
Standard Deviation	0.1847
50th Percentile	0.5880
75th Percentile	0.7015
90th Percentile	0.7922
95th Percentile	0.8440
99th Percentile	0.9287

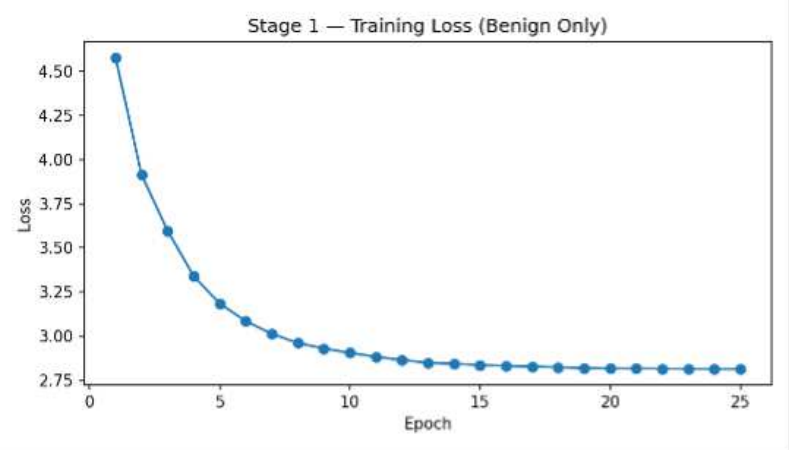
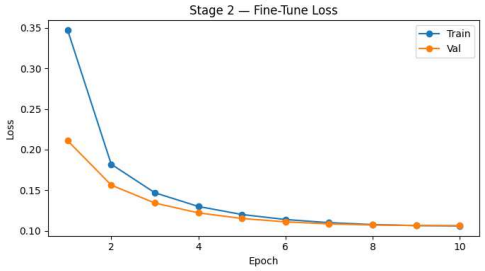


Fig. 4: Loss curve for self-supervised training

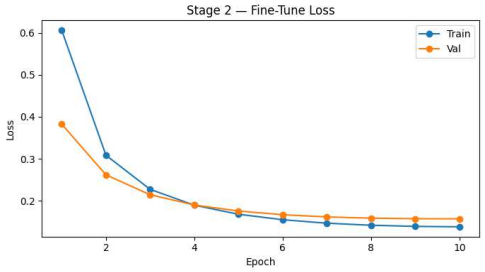
Fine-tuning the linear head on three corpora, TREC_05, TREC_0 6, and TREC_07, is depicted in Figures 5a, 5b and 5c. The training and validation loss curves converged for TREC_05 and TREC_07 and slightly diverged at five epochs for TREC_06. The training and validation loss curves diverged significantly when the MLP head was fine-tuned, as shown in Figure 5d, indicating that this approach overfits.

4.2 Probing of SSL-trained JEPA backbone

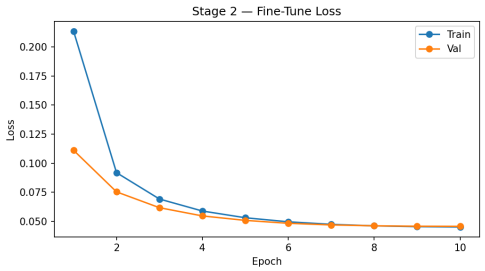
An SSL-trained JEPA backbone was evaluated for representation learning across four unseen corpora, each trained on one of three source corpora (TREC_07, TREC_06, and TREC_05) over five random seeds. The scores for the AUROC evaluation metrics were computed for each seed. For comparisons between the linear and MLP probes and between the JEPA backbone and vanilla DistilBERT, t-tests were conducted on the mean score differences, as shown in Tables 4, 5, and 6. From Figures 6a, 6b, and 6c the linear probe on the JEPA backbone shows a clearer separability of representations than the vanilla DistilBERT, except when TREC_05 was used as a training source and



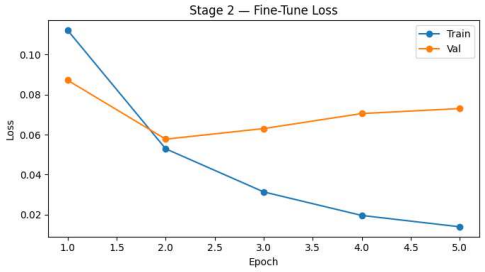
(a)



(b)



(c)



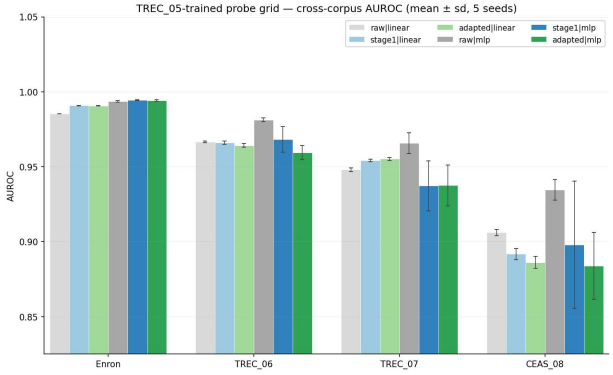
(d)

Fig. 5: Loss curves during Stage 2 fine-tuning of the JEPA backbone: (a) `TREC_05` with linear head, (b) `TREC_06` with linear head, (c) `TREC_07` with linear head, and (d) With MLP head

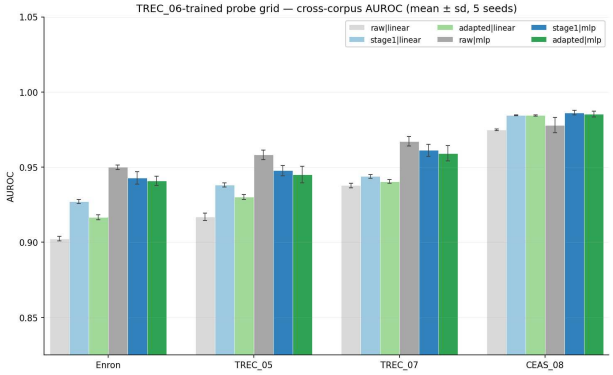
Table 3: Training metrics across epochs

Epoch	Loss	Mean Std (healthy ∼1.0)	Step Speed (it/s)
1	4.805060	0.9148	26.52
2	4.763422	0.9169	29.41
3	4.749636	0.9165	29.06
4	4.715651	0.9174	29.49
5	4.704952	0.9190	29.49
6	4.695112	0.9198	29.46
7	4.679479	0.9206	31.84
8	4.662361	0.9217	29.33
9	4.658292	0.9225	29.41
10	4.640884	0.9235	29.24
11	4.621585	0.9239	29.23
12	4.600123	0.9254	29.17
13	4.600638	0.9268	29.09
14	4.593065	0.9272	29.28
15	4.573267	0.9275	28.99

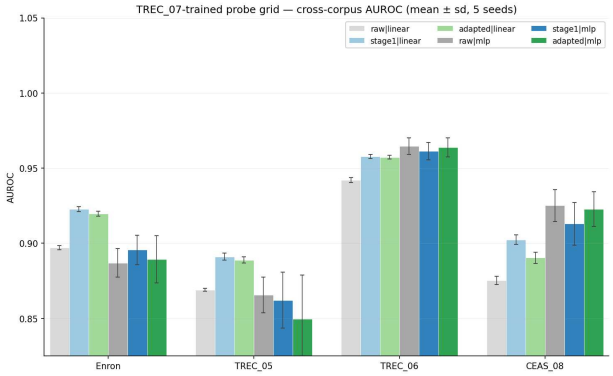
evaluated using TREC_06 and CEAS_08. This is supported by Table 4, which shows statistically significant results with a pooled AUROC of +0.0124 ($p \approx 9 \times 10^{-11}$), positive for 85% of the seeds. With the MLP probe, the separability of representations was unclear compared to the linear probe, as the tests showed no statistical significance ($p \approx 0.93$). The backbone was also used for domain adaptation on an auxiliary modern email corpus, and a t-test was conducted between the linear probe on the backbone and the adapted backbone. The domain adaptation offered no benefits but rather a small statistically significant degradation (pooled -0.0039 AUROC and ($p \approx 1 \times 10^{-11}$)) as shown in Table 7.



(a)



(b)



(c)

Fig. 6: Comparison of Stage 1 probing across five corpora with cross-corpus AUROC (mean ± SD, 5 seeds): (a) `TREC_07`-trained probe grid, (b) `TREC_06`-trained probe grid, and (c) `TREC_05`-trained probe grid

4.3 Fine-tuning classification head and testing

The SSL-pretrained JEPA backbone with a predictor and complete architecture was subsequently used to fine-tune the linear and MLP classifiers for downstream phishing email classification. The backbone was frozen, and a supervised head was trained. Training was repeated for each training corpus for both the linear and MLP heads for all five seeds. Three evaluation scores, AUROC, AUPRC, and FPR@95% TPR, were computed over five random seeds to evaluate and compare the performance efficacy. Similar to backbone probing, the classification heads were trained on three source corpora and cross-evaluated on five unseen corpora (Enron, TREC_05, TREC_06, TREC_07, CEAS_08, and Ling). T-tests were conducted to compare MLP and Linear classification heads, as shown in Tables 7, 8, and 9. There was no clear distinction between the linear and MLP performances. The performance of the heads varied across the source and evaluation corpora. When trained on TREC_05 and TREC_07, most evaluation corpora favored the linear head for the AUROC, AUPRC, and FPR@95% metrics. However, when trained on TREC_06, most evaluation corpora favored the MLP head for AUROC and FPR@95% and the linear head for AUPRC. When the aggregate AUROC and AUPRC scores were compared with those of the in-domain test, the scores decreased for both heads, as expected. However, the MLP head had a higher drop (7.24% to 19.2%) than the linear head (5.19% to 10.63%), despite having higher in-domain test scores. The aggregate scores across the five unseen and three source corpora were AUROC 0.9394, AUPRC 0.9157, and FPR@95% 0.3040 for the linear head and AUROC 0.9246, AUPRC 0.8722, and FPR@95% 0.3223 for the MLP head.

4.4 Baseline comparison

Seven baselines were used to compare the linear and MLP classification heads. The heads combined with the SSL-pretrained JEPA backbone trained on the three source corpora were evaluated using four unseen corpora. Baseline A was the SSL-trained JEPA backbone without a classification head. Baseline B was DistilBERT, trained as an email subject-only classifier, and Baseline C was an email body-only classifier. Baseline D is a standard dual-stream classifier built cleanly using vanilla DistilBERT. Baseline E was a shallow statistical n-gram tracking baseline built using TF-IDF. Baseline Fa is a FastText classifier, and Baseline Fb is a full end-to-end fine-tuning classifier. The results are presented in Figure 7 and Table 12. Irrespective of the training sources, classification heads, or evaluation corpora, Baseline Fa (FastText) performed the worst among the baselines. Similarly, Baseline A performed worse with the MLP head than with the linear head across the four unseen corpora. The end-to-end fine-tuned DistilBERT performed better than all the baselines and the JEPA backbone with a classification head. The SSL-pretrained JEPA backbone combined with linear classification achieved a performance comparable to that of end-to-end fine-tuning, apart from faster training and lower memory utilization than the fine-tuning of DistilBERT. Table 13 presents a comparison of the training execution times and peak GPU memory utilization. Compared with end-to-end DistilBERT fine-tuning,

Table 4: Cross-corpus representation probing AUROC comparison between self-supervised `stage1|linear` and baseline `vanilla|linear`

Train	Eval	n	Δ_{mean}	Δ_{sd}	t-value	p-value	Sign	Verdict
TREC_07	Enron	5	0.025676	0.001617	35.497181	0.000004	all+	robust +
	TREC_05	5	0.022137	0.002425	20.413090	0.000034	all+	robust +
	TREC_06	5	0.015780	0.000700	50.417747	0.000001	all+	robust +
	CEAS_08	5	0.027119	0.001034	58.659641	0.000001	all+	robust +
TREC_06	Enron	5	0.024727	0.001478	37.404681	0.000003	all+	robust +
	TREC_05	5	0.021231	0.001896	25.036767	0.000015	all+	robust +
	TREC_07	5	0.006096	0.000446	30.568838	0.000007	all+	robust +
	CEAS_08	5	0.009535	0.000406	52.453943	0.000001	all+	robust +
TREC_05	Enron	5	0.005439	0.000128	94.677372	0.000000	all+	robust +
	TREC_06	5	-0.000479	0.000674	-1.587889	0.187506	mix	n.s.
	TREC_07	5	0.006113	0.001254	10.897078	0.0000403	all+	robust +
	CEAS_08	5	-0.014329	0.002248	-14.249976	0.000141	all-	robust -
Pooled	Enron	15	0.018614	0.009722	–	0.000003	all+	robust +
	TREC_05	10	0.021684	0.002107	–	0.000000	all+	robust +
	TREC_06	10	0.007651	0.008594	–	0.020203	mix	robust +
	CEAS_08	15	0.007442	0.017633	–	0.124417	mix	n.s.
	TREC_07	10	0.006104	0.000888	–	0.000000	all+	robust +
	ALL	60	0.012420	0.012226	–	0.000000	85%pos	robust +

Table 5: Cross-corpus representation probing AUROC comparison between MLP and linear probes on the self-supervised backbone (`stage1|mlp` vs. `stage1|linear`)

Train	Eval	n	Δ_{mean}	Δ_{sd}	t-value	p-value	Sign	Verdict
TREC_07	Enron	5	-0.027112	0.009551	-6.347667	0.003155	all-	robust -
	TREC_05	5	-0.028969	0.017568	-3.687170	0.021071	mix	robust -
	TREC_06	5	0.003482	0.004735	1.644342	0.175451	mix	n.s.
	CEAS_08	5	0.010578	0.015126	1.563792	0.192910	mix	n.s.
TREC_06	Enron	5	0.015715	0.003821	9.197215	0.000776	all+	robust +
	TREC_05	5	0.009556	0.002521	8.476206	0.001062	all+	robust +
	TREC_07	5	0.017357	0.003542	10.957569	0.000394	all+	robust +
	CEAS_08	5	0.001793	0.001615	2.482852	0.068003	mix	n.s.
TREC_05	Enron	5	0.003622	0.000302	26.826853	0.000011	all+	robust +
	TREC_06	5	0.002075	0.008192	0.566412	0.601382	mix	n.s.
	TREC_07	5	-0.016845	0.016057	-2.345823	0.078874	mix	n.s.
	CEAS_08	5	0.006201	0.040486	0.342480	0.749229	mix	n.s.
Pooled	Enron	15	-0.002592	0.019454	–	0.613915	mix	n.s.
	TREC_05	10	-0.009706	0.023500	–	0.223910	mix	n.s.
	TREC_06	10	0.002779	0.006351	–	0.199879	mix	n.s.
	CEAS_08	15	0.006191	0.023414	–	0.323181	mix	n.s.
	TREC_07	10	0.000256	0.021097	–	0.970208	mix	n.s.
	ALL	60	-0.000212	0.020171	–	0.935340	68%pos	n.s.

Table 6: Cross-corpus representation probing AUROC comparison between self-supervised `stage1|mlp` and baseline `vanilla|mlp`

Train	Eval	n	Δ_{mean}	Δ_{sd}	t-value	p-value	Sign	Verdict
TREC_07	Enron	5	0.008617	0.017662	1.090994	0.336589	mix	n.s.
	TREC_05	5	-0.003496	0.022314	-0.350297	0.743784	mix	n.s.
	TREC_06	5	-0.003396	0.008303	-0.914560	0.412166	mix	n.s.
	CEAS_08	5	-0.012149	0.012477	-2.177416	0.095019	mix	n.s.
TREC_06	Enron	5	-0.007028	0.005141	-3.056824	0.037775	mix	robust -
	TREC_05	5	-0.010453	0.002174	-10.753301	0.000424	all-	robust -
	TREC_07	5	-0.006007	0.002720	-4.937623	0.007830	all-	robust -
	CEAS_08	5	0.008362	0.004073	4.590541	0.010103	all+	robust +
TREC_05	Enron	5	0.000801	0.000256	6.989716	0.002204	all+	robust +
	TREC_06	5	-0.013123	0.007421	-3.954129	0.016760	all-	robust -
	TREC_07	5	-0.028504	0.012474	-5.109471	0.006937	all-	robust -
	CEAS_08	5	-0.036673	0.040504	-2.024539	0.112912	mix	n.s.
Pooled	Enron	15	0.000797	0.011849	–	0.798320	mix	n.s.
	TREC_05	10	-0.006974	0.015389	–	0.185631	mix	n.s.
	TREC_06	10	-0.008259	0.009022	–	0.017740	mix	robust -
	CEAS_08	15	-0.013487	0.029683	–	0.100281	mix	n.s.
	TREC_07	10	-0.017256	0.014596	–	0.004636	all-	robust -
	ALL	60	-0.008587	0.019108	–	0.000946	35%pos	robust -

Table 7: Cross-corpus representation probing AUROC comparison between domain-adapted linear probe and base self-supervised linear probe (`adapted|linear` vs. `stage1|linear`)

Train	Eval	n	Δ_{mean}	Δ_{sd}	t-value	p-value	Sign	Verdict
TREC_07	Enron	5	-0.002973	0.000501	-13.269587	0.000186	all-	robust -
	TREC_05	5	-0.002229	0.000764	-6.518978	0.002859	all-	robust -
	TREC_06	5	-0.000476	0.000215	-4.945059	0.007789	all-	robust -
	CEAS_08	5	-0.012025	0.000739	-36.370190	0.000003	all-	robust -
TREC_06	Enron	5	-0.010414	0.000386	-60.381792	0.000000	all-	robust -
	TREC_05	5	-0.008081	0.000248	-72.772844	0.000000	all-	robust -
	TREC_07	5	-0.003502	0.000102	-77.005494	0.000000	all-	robust -
	CEAS_08	5	-0.000123	0.000110	-2.510367	0.066031	all-	n.s.
TREC_05	Enron	5	-0.000142	0.000040	-7.961364	0.001348	all-	robust -
	TREC_06	5	-0.001894	0.000258	-16.399692	0.000081	all-	robust -
	TREC_07	5	0.001017	0.000563	4.036035	0.015655	all+	robust +
	CEAS_08	5	-0.005535	0.001413	-8.757254	0.000937	all-	robust -
Pooled	Enron	15	-0.004510	0.004497	–	0.001652	all-	robust -
	TREC_05	10	-0.005155	0.003131	–	0.000559	all-	robust -
	TREC_06	10	-0.001185	0.000780	–	0.000968	all-	robust -
	CEAS_08	15	-0.005894	0.005108	–	0.000530	all-	robust -
	TREC_07	10	-0.001243	0.002412	–	0.137755	mix	n.s.
	ALL	60	-0.003865	0.004157	–	1E-09	8%pos	robust -

Table 8: Comparison of in-domain performance, aggregate cross-corpus validation (CV) results, and percentage performance drop between linear and MLP classification heads

	Linear		MLP	
Source	AUROC	AUPRC	AUROC	AUPRC
In-domain test				
TREC_05	0.9951	0.9954	0.9985	0.9979
TREC_06	0.9948	0.9846	0.9963	0.9877
TREC_07	0.9981	0.9982	0.9991	0.9991
Average	*0.9960*	*0.9927*	*0.9980*	*0.9949*
Aggregate CV result				
TREC_05	0.9432	0.9167	0.9261	0.8692
TREC_06	0.9549	0.9387	0.9607	0.9403
TREC_07	0.9200	0.8919	0.8871	0.8071
Average	*0.9394*	*0.9157*	*0.9246*	*0.8722*
Drop (In percentage)				
TREC_05	5.19%	7.87%	7.24%	12.87%
TREC_06	3.99%	4.59%	3.56%	4.74%
TREC_07	7.81%	10.63%	11.20%	19.20%
Average	*5.66%*	*7.70%*	*7.33%*	*12.27%*

the JEPA backbone combined with a linear classification head reduced the training execution time by 60% and peak-allocated GPU memory by 43%.

4.5 Calibration effects at fixed $\tau = 0.5$

Using Isotonic regression, when calibrated at fixed threshold $\tau = 0.5$ on the linear head, keeping AUROC, AUPRC, and FPR@95% constant, the fixed FPR decreased by 0.1056 (10.6%) when trained on TREC_05, by 0.0945 (9.5%) when trained on TREC_06, and by 0.1667 (16.7%) when trained on TREC_07. Regarding the fixed F1 score for the linear head at fixed tau=0.5, the score increased by 0.0404 (4%) when trained on TREC-05, 0.0143 (1.4%) when trained on TREC_06, and 0.0357 (3.6%) when trained on TREC_07. When calibrated at a fixed tau on the MLP head, the F1 score increased by 0.0395 (4%) for the training corpus TREC_05, 0.0143 (1.4%) for TREC_06, and 0.0234 (2.3%) for TREC_07. However, the FPR decreased by 0.0147 (1.5%) for the training corpus TREC_05, 0.0323(3.2%) for the training corpus TREC_06, and 0.1506 (15%) for TREC_07. Overall, with the linear head, the calibration gains were 0.0301 in F1 score and -0.1223 in FPR, whereas with the MLP head, the gain was 0.0257 for F1 score and -0.0958 for FPR. The F1-optimized remained unchanged on both heads when calibrated. The calibration results are shown in Figure 8.

Table 9: Cross-corpus Stage 2 downstream classification AUROC comparison between MLP and linear heads (`stage2|mlp` vs. `stage2|linear`)

Train	Corpus	Δ_{mean}	Δ_{sd}	SE	t-value	p-value	Verdict	Winner
TREC_07	Enron	-0.0528	0.0090	0.0040	-13.1500	0.0001	robust - (all-)	linear
	TREC_05	-0.0495	0.0157	0.0070	-7.0500	0.0011	robust - (all-)	linear
	TREC_06	0.0021	0.0024	0.0011	1.9600	0.0608	n.s. (mix)	mlp
	CEAS_08	0.0173	0.0350	0.0157	1.1100	0.1651	n.s. (mix)	mlp
	Ling	-0.0812	0.0588	0.0263	-3.0900	0.0183	robust - (all-)	linear
TREC_06	Enron	0.0083	0.0032	0.0014	5.8100	0.0022	robust + (all+)	mlp
	TREC_05	0.0084	0.0011	0.0005	17.7100	0.0000	robust + (all+)	mlp
	TREC_07	0.0143	0.0049	0.0022	6.5100	0.0014	robust + (all+)	mlp
	CEAS_08	-0.0057	0.0074	0.0033	-1.7300	0.0794	n.s. (mix)	linear
	Ling	0.0038	0.0094	0.0042	0.9000	0.2099	n.s. (mix)	mlp
TREC_05	Ling	-0.0224	0.0345	0.0154	-1.4600	0.1096	n.s. (mix)	linear
	Enron	0.0019	0.0012	0.0006	3.3600	0.0141	robust + (mix)	mlp
	TREC_06	-0.0143	0.0064	0.0029	-4.9800	0.0038	robust - (all-)	linear
	TREC_07	-0.0254	0.0107	0.0048	-5.3300	0.0030	robust - (all-)	linear
	CEAS_08	-0.0256	0.0288	0.0129	-1.9800	0.0592	n.s. (mix)	linear

Table 10: Cross-corpus Stage 2 downstream classification AUPRC comparison between MLP and linear heads (`stage2|mlp` vs. `stage2|linear`)

Train	Corpus	Δ_{mean}	Δ_{sd}	SE	t-value	p-value	Verdict	Winner
TREC_07	Ling	-0.2373	0.0868	0.0388	-6.1100	0.0036	robust - (all-)	linear
	Enron	-0.0847	0.0157	0.0070	-12.0700	0.0003	robust - (all-)	linear
	TREC_05	-0.0967	0.0350	0.0156	-6.1800	0.0035	robust - (all-)	linear
	TREC_06	-0.0118	0.0194	0.0087	-1.3600	0.2458	n.s. (mix)	linear
	CEAS_08	0.0067	0.0133	0.0059	1.1300	0.3227	n.s. (mix)	mlp
TREC_06	Ling	-0.0010	0.0324	0.0145	-0.0700	0.9487	n.s. (mix)	linear
	Enron	-0.0062	0.0083	0.0037	-1.6600	0.1726	n.s. (mix)	linear
	TREC_05	0.0054	0.0034	0.0015	3.4900	0.0252	robust + (all+)	mlp
	TREC_07	0.0122	0.0049	0.0022	5.5500	0.0052	robust + (all+)	mlp
	CEAS_08	-0.0022	0.0053	0.0024	-0.9300	0.4073	n.s. (mix)	linear
TREC_05	Ling	-0.1503	0.0652	0.0292	-5.1600	0.0067	robust - (all-)	linear
	Enron	0.0019	0.0010	0.0005	4.1400	0.0144	robust + (all+)	mlp
	TREC_06	-0.0382	0.0250	0.0112	-3.4100	0.0271	robust - (all-)	linear
	TREC_07	-0.0323	0.0149	0.0067	-4.8400	0.0084	robust - (all-)	linear
	CEAS_08	-0.0187	0.0177	0.0079	-2.3700	0.0771	n.s. (mix)	linear

5 Discussion

We trained JEPA on email subject-body representations using self-supervised learning on the Enron email corpus. Subsequently, the self-supervised JEPA representation was tested using linear and MLP probes without a predictor network or downstream

Table 11: Cross-corpus Stage 2 downstream classification FPR@95% TPR comparison between MLP and linear heads (`stage2|mlp` vs. `stage2|linear`), rounded to 4 decimal places

Train	Corpus	Δ_{mean}	Δ_{sd}	SE	t-value	p-value	Verdict	Winner
TREC_07	Ling	0.1896	0.1605	0.0718	2.6400	0.0575	n.s. (mix)	linear
	Enron	0.0739	0.0213	0.0095	7.7400	0.0015	robust + (all+)	linear
	TREC_05	0.0666	0.0618	0.0277	2.4100	0.0736	n.s. (mix)	linear
	TREC_06	-0.0318	0.0097	0.0043	-7.3100	0.0019	robust - (all-)	mlp
	CEAS_08	-0.1394	0.2167	0.0969	-1.4400	0.2238	n.s. (mix)	mlp
TREC_06	Ling	0.0048	0.0519	0.0232	0.2100	0.8453	n.s. (mix)	linear
	Enron	-0.1132	0.0094	0.0042	-27.0700	0.0000	robust - (all-)	mlp
	TREC_05	-0.0485	0.0199	0.0089	-5.4600	0.0055	robust - (all-)	mlp
	TREC_07	-0.0652	0.0228	0.0102	-6.4000	0.0031	robust - (all-)	mlp
	CEAS_08	0.0559	0.0460	0.0206	2.7200	0.0531	n.s. (all+)	linear
TREC_05	Ling	0.0376	0.0963	0.0431	0.8700	0.4321	n.s. (mix)	linear
	Enron	-0.0121	0.0024	0.0011	-11.3000	0.0003	robust - (all-)	mlp
	TREC_06	0.0506	0.0214	0.0096	5.2900	0.0061	robust + (all+)	linear
	TREC_07	0.0761	0.0455	0.0203	3.7400	0.0201	robust + (all+)	linear
	CEAS_08	0.1290	0.1310	0.0586	2.2000	0.0925	n.s. (mix)	linear

classification tasks across the four unseen corpora. We observed no collapse of subject-body representations with self-supervised learning using VICReg during Stage 1. We used this self-supervised and pretrained JEPA backbone to answer our research sub-questions.

RQ1.1 Does JEPA pretraining yield more transferable phishing representations than an identical backbone trained without self-supervision, as measured by cross-corpus performance under a fixed linear/MLP probe?

To answer research question RQ1.1, we used a linear probe and compared it with vanilla DistilBERT as a comparison backbone. We observed an improvement in AUROC (+0.012) with pooled seeds, supported by a p-value of $p \approx 9 \times 10^{-11}$ and positive in 85% of the seeds. This was not observed with MLP probing, where the pooled AUROC decreased by 0.009 and was positive in only 35% of the seeds. The direct comparison between the MLP and linear probes on the JEPA backbone was not statistically significant ($p \approx 0.94$), and the AUROC loss was 0.000212 for the pooled seeds. The MLP probe could have performed better than the linear probe if the training corpus was larger and additional representations were used beyond the subject and body.

We also tested an adapted self-trained JEPA backbone trained on an auxiliary modern email corpus. The intention was to adapt the JEPA backbone to a modern email corpus to enhance representation. Domain adaptation did not provide any benefits and instead resulted in a small but statistically significant deterioration of the backbone. Under linear probing, the adapted JEPA backbone achieved a 0.003865 lower AUROC for pooled seeds with $p \approx 1 \times 10^{-9}$ and only 8% positive seeds. Overall,

Table 12: Cross-corpus baseline evaluation metrics (AUPRC, AUROC, FPR@95% TPR) comparing Stage 2 with a linear head against baselines across all training source configurations (TREC_07, TREC_06, TREC_05, and POOLED)

		AUPRC						AUROC						FPR@95% TPR					
Train	Model	Enron	TREC_05	TREC_06	CEAS_08	Ling	Avg.	Enron	TREC_05	TREC_06	CEAS_08	Ling	Avg.	Enron	TREC_05	TREC_06	CEAS_08	Ling	Avg.
TREC_07	B	0.8220	0.7423	0.7629	0.9333	0.4425	0.7406	0.8391	0.8222	0.8956	0.9116	0.7939	0.8525	83.9%	82.2%	89.6%	91.2%	79.4%	85.3%
	C	0.9095	0.8572	0.9536	0.9159	0.8450	0.8962	0.9136	0.8806	0.9834	0.8389	0.9348	0.9103	91.4%	88.1%	98.3%	83.9%	93.5%	91.0%
	D	0.9019	0.8562	0.9392	0.9340	0.6437	0.8550	0.9102	0.8853	0.9793	0.8849	0.8980	0.9115	39.3%	52.7%	9.5%	71.7%	42.0%	43.0%
	E	0.9410	0.8917	0.9001	0.8789	0.8328	0.8889	0.9402	0.9058	0.9547	0.7901	0.9272	0.9036	32.5%	47.7%	23.0%	90.8%	44.1%	47.6%
	Fa	0.3646	0.2895	0.1647	0.4409	0.1070	0.2733	0.3341	0.2582	0.2821	0.3414	0.2745	0.2981	97.2%	100.0%	100.0%	100.0%	98.6%	99.2%
	Fb	0.9311	0.8695	0.9377	0.9432	0.8586	0.9080	0.9368	0.8994	0.9774	0.9126	0.9463	0.9345	30.9%	43.2%	11.9%	48.9%	31.1%	33.2%
	Stage2—Lin	0.9337	0.8729	0.9226	0.9228	0.8075	0.8919	0.9350	0.9035	0.9709	0.8660	0.9243	0.9200	32.4%	42.7%	13.9%	74.1%	38.7%	40.4%

		AUPRC						AUROC						FPR@95% TPR					
Train	Model	Enron	TREC_05	TREC_07	CEAS_08	Ling	Avg.	Enron	TREC_05	TREC_07	CEAS_08	Ling	Avg.	Enron	TREC_05	TREC_07	CEAS_08	Ling	Avg.
TREC_06	B	0.8489	0.8053	0.9329	0.9713	0.7733	0.8663	0.9553	0.8444	0.8881	0.8558	0.8920	0.8871	63.8%	61.3%	26.5%	10.7%	25.3%	37.5%
	C	0.9482	0.9392	0.9565	0.9677	0.9033	0.9430	0.9491	0.9519	0.9543	0.9622	0.9739	0.9583	27.6%	24.5%	19.1%	16.6%	13.0%	20.2%
	D	0.9561	0.9398	0.9656	0.9793	0.9311	0.9544	0.9569	0.9552	0.9634	0.9769	0.9847	0.9674	23.8%	22.5%	16.0%	10.2%	8.1%	16.1%
	E	0.9092	0.8736	0.9390	0.9397	0.9301	0.9183	0.9249	0.9079	0.9338	0.9562	0.9762	0.9398	30.8%	34.1%	22.9%	11.2%	12.9%	22.4%
	Fa	0.3657	0.4612	0.7825	0.7416	0.1300	0.4962	0.3239	0.5531	0.8059	0.7352	0.4002	0.5637	99.2%	96.3%	51.9%	96.3%	99.1%	88.5%
	Fb	0.9728	0.9557	0.9813	0.9836	0.9416	0.9670	0.9727	0.9656	0.9765	0.9824	0.9744	0.9743	14.9%	18.6%	13.3%	7.6%	17.2%	14.3%
	Stage2—Lin	0.9433	0.9335	0.9577	0.9797	0.8792	0.9387	0.9395	0.9472	0.9505	0.9787	0.9586	0.9549	35.7%	25.3%	23.3%	7.8%	19.2%	22.3%

		AUPRC						AUROC						FPR@95% TPR					
Train	Model	Enron	TREC_06	TREC_07	CEAS_08	Ling	Avg.	Enron	TREC_06	TREC_07	CEAS_08	Ling	Avg.	Enron	TREC_06	TREC_07	CEAS_08	Ling	Avg.
TREC_05	B	0.9538	0.7243	0.9076	0.8851	0.7098	0.8361	0.9553	0.8444	0.8881	0.8558	0.8920	0.8702	22.9%	76.7%	55.4%	44.8%	52.9%	50.5%
	C	0.9928	0.9474	0.9520	0.9121	0.5610	0.8731	0.9938	0.9762	0.9370	0.8686	0.8190	0.9189	2.2%	11.1%	36.8%	55.0%	67.5%	34.5%
	D	0.9940	0.9461	0.9561	0.9262	0.6956	0.9036	0.9948	0.9766	0.9472	0.8922	0.8994	0.9420	1.6%	11.8%	27.2%	47.4%	44.6%	26.5%
	E	0.9961	0.8835	0.9433	0.8798	0.8560	0.9117	0.9966	0.9389	0.9243	0.8365	0.9433	0.9279	1.3%	33.7%	44.9%	64.6%	33.7%	35.6%
	Fa	0.2927	0.1768	0.4612	0.4343	0.1160	0.2962	0.0809	0.3247	0.3902	0.3144	0.3186	0.2858	99.6%	97.9%	94.9%	96.9%	99.4%	97.7%
	Fb	0.9980	0.9462	0.9559	0.9018	0.8351	0.9274	0.9983	0.9769	0.9436	0.8475	0.9337	0.9400	0.6%	12.2%	31.6%	66.8%	40.8%	30.4%
	Stage2—Lin	0.9921	0.9400	0.9727	0.9333	0.7453	0.9167	0.9928	0.9746	0.9643	0.9047	0.8798	0.9432	3.0%	13.4%	18.9%	47.1%	60.4%	28.6%

		AUPRC						AUROC						FPR@95% TPR					
Train	Model	Enron	TREC_06	TREC_07	CEAS_08	Ling	Avg.	Enron	TREC_06	TREC_07	CEAS_08	Ling	Avg.	Enron	TREC_06	TREC_07	CEAS_08	Ling	Avg.
POOLED	B	0.8749	0.7573	0.8678	0.9299	0.6419	0.8541	0.8796	0.8516	0.8798	0.8865	0.8521	0.8699	56.9%	73.4%	57.2%	48.9%	52.5%	57.8%
	C	0.9502	0.9146	0.9541	0.9319	0.7698	0.9041	0.9522	0.9363	0.9582	0.8899	0.9092	0.9292	40.4%	41.2%	51.4%	51.8%	58.0%	48.6%
	D	0.9507	0.9140	0.9536	0.9465	0.7568	0.9043	0.9540	0.9390	0.9633	0.9180	0.9274	0.9403	21.6%	29.0%	17.6%	43.1%	31.6%	28.6%
	E	0.9487	0.8829	0.9275	0.8995	0.8730	0.9063	0.9539	0.9175	0.9376	0.8609	0.9489	0.9238	21.5%	38.5%	30.3%	55.5%	30.2%	35.2%
	Fa	0.3410	0.3092	0.4695	0.5389	0.1177	0.3553	0.2463	0.3787	0.4927	0.4636	0.3311	0.3825	98.7%	98.0%	82.3%	97.7%	99.0%	95.1%
	Fb	0.9673	0.9238	0.9583	0.9428	0.8784	0.9341	0.9693	0.9473	0.9658	0.9142	0.9514	0.9496	15.4%	24.6%	18.9%	41.1%	29.7%	26.0%
	Stage2—Lin	0.9564	0.9155	0.9510	0.9453	0.8107	0.9157	0.9558	0.9418	0.9619	0.9165	0.9209	0.9394	23.7%	27.1%	18.7%	43.0%	39.4%	30.4%

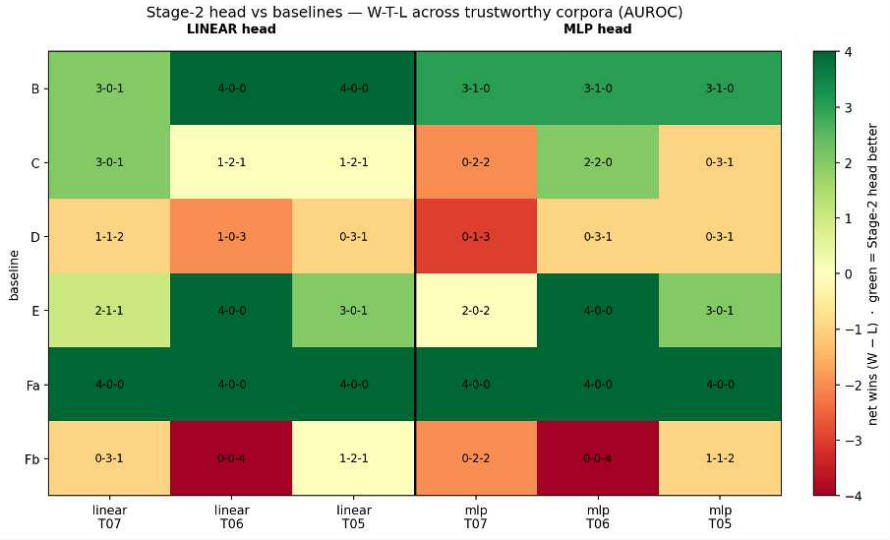


Fig. 7: Comparison of linear and MLP head classifications with baselines over three training corpora

Table 13: Comparison of training execution time (in minutes) and peak GPU memory consumption (in GB) between Stage 2 JEPA classifiers and full end-to-end Vanilla DistilBERT fine-tuning

	Training time (in minutes)				Memory (in GB)	
Model \ Source	TREC_05	TREC_06	TREC_07	Average	Allocated	Reserved
JEPA+MLP	8.18	2.22	7.32	5.91	1.19	1.32
JEPA+Linear	7.47	2.68	9.03	6.39	1.22	1.37
Vanilla DistilBERT	20.30	6.48	20.30	15.69	2.14	2.62

the self-supervised pretrained JEPA backbone demonstrated better transferable phishing presentations than the vanilla DistilBERT under a linear probe without domain adaptation, as measured by cross-corpus performance, which answered our research question, RQ1.1.

RQ1.2 Can a frozen JEPA encoder with a lightweight classifier match or approach the cross-corpus performance of end-to-end fine-tuning and strong classical baselines?

The pretrained JEPA backbone was frozen, and a supervised linear and MLP classifier head, along with the predictor and complete architectural components, were trained. This was repeated for all three training corpora and five seeds evaluated against five unseen evaluation corpora. The t-tests revealed no clear distinction between the linear and MLP heads. When trained on TREC_05 and TREC_07, most evaluation corpora favored the linear head for the AUROC, AUPRC, and FPR@95% metrics. However, when trained on TREC_06, most evaluation corpora favored the

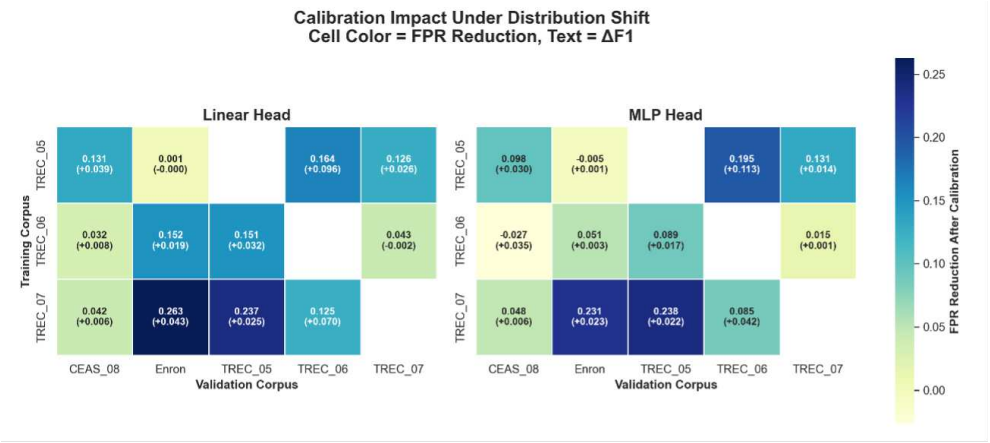


Fig. 8: Linear head and MLP head calibration impact across four unseen corpora

MLP head for AUROC and FPR@95% and the linear head for AUPRC. Performance varied across the training and evaluation corpora; however, linear-head performance stood out in two of the three source corpora in the cross-corpus validation.

The Stage 2 model was subjected to baseline comparisons. The frozen JEPA backbone with the trained classification head was compared with six baselines: anomaly detection, subject-only phishing classifier, body-only phishing classifier, standard dual-stream classifier built cleanly using vanilla DistilBERT, shallow statistical n-gram tracking baseline, FastText classifier, and full end-to-end BERT fine-tuning classifier. The frozen JEPA with a linear classifier head achieved a performance comparable to that of the end-to-end BERT fine-tuning classifier while training faster and matching the standard dual-stream classifier.

RQ1.3 How does classifier capacity (linear vs. MLP) trade cross-corpus ranking quality against operating point stability under a distribution shift?

The in-domain test results showed near-perfect AUROC and AUPRC scores, with the MLP head ranking being slightly higher than the linear head ranking. The average in-domain test scores were AUROC 0.9960 and AUPRC 0.9927 for the linear head and AUROC 0.9980 and AUPRC 0.9949 for the MLP head. During the cross-validation tests, the rank scores varied within the training sources and across the four unforeseen corpora. As expected, the CV rank scores were lower than the in-domain test rank scores. The average CV rank scores for the linear head were AUROC 0.9394 and AUPRC 0.9157, whereas those for the MLP head were AUROC 0.9246 and AUPRC 0.8872, respectively. The decrease in rank scores was higher for the MLP head (ranging from 7.24% to 19.2%) than for the linear head (5.19% to 10.63%). As expected, the scores dropped when evaluated with unseen corpora but remained above 90% for both rank scores. The FPR@95% TPR was 0.3040, which means that when a true-positive rate of 95% was considered, false positives were detected 30% of the time, which was slightly higher for the phishing detection model.

RQ1.4 Can per-domain calibration recover a reliable fixed-threshold operating point under a distribution shift without degrading the ranking performance?

Calibration with Isotonic Regression pulled the threshold back toward 0.5, without affecting the ranking metrics. During calibration, keeping the ranks (AUROC, AUPRC, and FPR@95) constant, the linear head gained a slightly higher fixed F1 score (0.0301 vs. 0.0257) and higher FPR loss (-0.1223 vs. -0.0958) than that of the MLP head. The aggregate tau value decreased from 0.6784 (raw value) to 0.4556 (calibrated value) for the linear head and from 0.7537 to 0.4455 for the MLP head. Although the performance varied across the source corpora, the linear head outperformed the MLP head on all three source corpora and the aggregate. Contrary to expectations, the MLP head underperformed in end-to-end fine-tuning and calibration, despite its higher capacity. Overall, the per-domain calibration recovered a fixed and reliable threshold operating point under a distribution shift without degrading the ranking performance.

In this study, we used only subject-body representations for phishing-related details. Subject–body discordance is one exploitable regularity among several. For our self-supervised training, we used heuristic-based multi-approaches to address representation collapse. However, these can be replaced by better modern approaches. The pretraining corpus used was an unlabeled, relatively small, and outdated Enron email corpus. The training corpus was slightly outdated, and the auxiliary corpus for domain adaptation was not purely a modern email corpus.

Because all corpora predate LLMs, PhishJEPA does not guarantee to work on LLM-generated phishing and generative adversarial networks, which have not been tested. The framework also lacks the explainability of which representations are retained. Similarly, the FPR@95% TPR was high for the phishing detection model used for real-time deployment.

6 Conclusion

In this study, we implemented **PhishJEPA**, a self-supervised framework leveraging a Joint Embedding Predictive Architecture (JEPA), VICReg regularizer, and Exponential Moving Average (EMA) updates to model email subject-body semantic relationships. Under Stage 1 feature probing across four unseen evaluation corpora, pretrained JEPA representations demonstrated statistically significant improvements in cross-corpus separability over vanilla DistilBERT (+0.0124 pooled AUROC, $p \approx 9 \times 10^{-11}$). The unsupervised domain adaptation of the JEPA backbone on the auxiliary corpus induced a subtle representation drift (-0.0039 AUROC, $p \approx 1 \times 10^{-11}$), indicating that poorly curated adaptation can degrade the pretrained semantic alignment.

For Stage 2 downstream classification, freezing the JEPA backbone and training a lightweight linear head achieved a cross-corpus performance (0.9394 AUROC / 0.9157 AUPRC) comparable to that of full end-to-end DistilBERT fine-tuning (0.9496 AUROC / 0.9341 AUPRC). Crucially, linear heads reduced the average training execution time by 60% (6.39 min vs. 15.69 min) and peak GPU memory allocation by 43% (1.22 GB vs. 2.14 GB). Although higher-capacity MLP heads achieved higher in-domain scores (0.9980 AUROC), they suffered substantially larger performance drops

(7.24%–19.20%) under a distribution shift than linear heads (5.19%–10.63%), rendering them less threshold stable. Finally, post-processing per-domain isotonic calibration at a fixed operating threshold ($\tau = 0.5$) successfully restored operational stability, reducing false-positive rates by an average of 12.2% (up to 16.7%) without altering the ranking performance.

In summary, while JEPA effectively captures transferable semantic representations, achieving reliable real-world phishing detection under severe distribution shift depends on combining lightweight classifier design with post-hoc operating threshold calibration.

Future studies should use not only email subject-body relationships but also extend to other features and multimodal data. These studies could also use heuristics-free approaches to effectively address collapses. A good blend of historical and modern email corpora can be used for SSL pretraining. It is recommended to use transformers/encoders other than DistilBERT to improve latent representations and computational resource utilization.

Acknowledgements. We acknowledge the support received for this study from Sophia University and the Project for Human Resource Development Scholarship (JDS).

Author contributions. **Tandin Wangchuk**: Conceptualization, Methodology, Experimentation, Data curation, Writing – original draft, formatting. **Karma Wangchuk**: Writing – review & editing. **Tad Gonsalves**: Supervision, Validation.

Data availability. The datasets used in the current study are publicly available [44] from Zenodo and other public repositories.

Declarations.

Funding. Authors received no specific funding for this study.

Conflict of interest / Competing interests. The authors declare that there are no conflicts of interest regarding the publication of this study.

Consent for publication. All authors consent to the publication of this study.

Ethics approval and consent to participate. Not applicable.

Clinical Trial Number. Not applicable.

References

[1] Investigation (FBI), F.B.: 2025 Internet Crime Complaint Center (IC3) Annual Report (2025). https://www.ic3.gov/AnnualReport/Reports Accessed 2026-07-20

[2] Business, C.: 2025 Comcast Business Cybersecurity Threat Report. Technical report, Comcast Business (2025). https://business.comcast.com/resources/enterprise/reports/2025-comcast-business-cybersecurity-threat-report Accessed 2026-08-31

[3] IBM: Cost of a Data Breach Report 2026. Technical report, IBM (2026). https://www.ibm.com/reports/data-breach Accessed 2026-08-30

[4] Business, V.: 2026 Data Breach Investigations Report. Technical report, Verison Business (2026). https://www.verizon.com/business/resources/reports/dbir/ Accessed 2026-09-04

[5] APWG: Phishing Activity Trends Report - 1st Quarter 2026. Technical report, APWG (May 2026). https://apwg.org/trendreports Accessed 2026-08-30

[6] Sheng, S., Wardman, B., Warner, G., Cranor, L., Hong, J., Zhang, C.: An Empirical Analysis of Phishing Blacklists. Carnegie Mellon University (2009)

[7] Prakash, P., Kumar, M., Kompella, R.R., Gupta, M.: PhishNet: Predictive Blacklisting to Detect Phishing Attacks. In: 2010 Proceedings IEEE INFOCOM, pp. 1–5 (2010). https://doi.org/10.1109/INFCOM.2010.5462216 . ISSN: 0743-166X. https://ieeexplore.ieee.org/document/5462216 Accessed 2026-09-07

[8] Basnet, R., Mukkamala, S., Sung, A.: Detection of Phishing Attacks: A Machine Learning Approach. In: Studies in Fuzziness and Soft Computing vol. 226, pp. 373–383. Springer, ??? (2008). https://doi.org/10.1007/978-3-540-77465-5_19 . Journal Abbreviation: Studies in Fuzziness and Soft Computing

[9] Yasin, A., Abuhasan, A.: An Intelligent Classification Model for Phishing Email Detection. International Journal of Network Security & Its Applications **8**(4), 55–72 (2016) https://doi.org/10.5121/ijnsa.2016.8405 . Accessed 2026-09-05

[10] Hureau, O., Bayer, J., Duda, A., Korczyński, M.: Spoofed Emails: An Analysis of the Issues Hindering a Larger Deployment of DMARC. In: Richter, P., Bajpai, V., Carisimo, E. (eds.) Passive and Active Measurement vol. 14537, pp. 232–261. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-56249-5_10 . Series Title: Lecture Notes in Computer Science. https://link.springer.com/10.1007/978-3-031-56249-5_10 Accessed 2026-09-07

[11] Security, A.: What Is Email Sandboxing: How It Detects Advanced Email-Borne Threats That Signature-Based Filters Miss (2026). https://www.adaptivesecurity.com/blog/email-sandboxing-explained Accessed 2026-09-12

[12] Benavides-Astudillo, E., Fuertes, W., Sanchez-Gordon, S., Nuñez-Agurto, D., Rodríguez-Galán, G.: A Phishing-Attack-Detection Model Using Natural Language Processing and Deep Learning. Applied Sciences **13**(9), 5275 (2023) https://doi.org/10.3390/app13095275 . Accessed 2026-09-02

[13] Ibrahim, M., Elhafiz, R.: Phishing Email Detection Using BERT and RoBERTa. Computation **14**(2), 46 (2026) https://doi.org/10.3390/computation14020046 . Accessed 2026-09-02

[14] Songailaitė, M., Kankevičiūtė, E., Zhyhun, B., Mandravickaitė, J.: BERT-Based Models for Phishing Detection. (2023)

[15] M. Safran, A. Musleh: PhishingGNN: Phishing Email Detection Using Graph Attention Networks and Transformer-Based Feature Extraction. IEEE Access **13**, 131390–131399 (2025) https://doi.org/10.1109/ACCESS.2025.3592135

[16] Atawneh, S., Aljehani, H.: Phishing Email Detection Model Using Deep Learning. Electronics **12**(20), 4261 (2023) https://doi.org/10.3390/electronics12204261 . Accessed 2026-09-05

[17] Stevanović, N.: Character and Word Embeddings for Phishing Email Detection. Computing and Informatics **41**(5), 1337–1357 (2022) https://doi.org/10.31577/cai_2022_5_1337 . Accessed 2026-09-05

[18] Yeboah, P.N., Kayes, A.S.M., Rahayu, W., Pardede, E., Mahbub, S.: A Framework for Phishing and Web Attack Detection Using Ensemble Features of Self-supervised Pre-trained Models (2025). https://doi.org/10.36227/techrxiv.173603362.21995515/v1 . https://www.techrxiv.org/doi/full/10.36227/techrxiv.173603362.21995515/v1 Accessed 2026-09-01

[19] Kotei, E., Thirunavukarasu, R.: A Systematic Review of Transformer-Based Pre-Trained Language Models through Self-Supervised Learning. Information **14**(3), 187 (2023) https://doi.org/10.3390/info14030187 . Accessed 2026-09-12

[20] Fang, Y., Zhang, C., Huang, C., Liu, L., Yang, Y.: Phishing Email Detection Using Improved RCNN Model With Multilevel Vectors and Attention Mechanism. IEEE Access **7**, 56329–56340 (2019) https://doi.org/10.1109/ACCESS.2019.2913705 . Accessed 2026-09-05

[21] Bhuiyan, I., Bhuiyan, T.: The Fragility of Phishing Detection Models: Evidence from Cross-Corpus Transfer, Prevalence Shift, Artifact Learning, and Evasion Risk. Big Data and Cognitive Computing **10**(7), 211 (2026) https://doi.org/10.3390/bdcc10070211 . Accessed 2026-09-04

[22] Maliha, F., Jahin, G.T.S., Habiba, U., Chy, A.N., Patwary, M.J.A.: Explainable Comprehensive Analysis of Cross-Dataset Phishing Detection Models. In: 2026 IEEE 2nd International Conference on Quantum Photonics, Artificial Intelligence & Networking (QPAIN), pp. 1–6 (2026). https://doi.org/10.1109/QPAIN69676.2026.11546228 . https://ieeexplore.ieee.org/document/11546228 Accessed 2026-09-11

[23] Pajola, L., Caripoti, E., Banzer, S., Pizzi, S., Conti, M., Apruzzese, G.: E-PhishGEN: Unlocking Novel Research in Phishing Email Detection. In: Proceedings of the 18th ACM Workshop on Artificial Intelligence And Security. AISec '25, pp. 64–76. Association for Computing Machinery, New York, NY, USA (2026). https://doi.org/10.1145/3733799.3762967 .

https://dl.acm.org/doi/10.1145/3733799.3762967 Accessed 2026-09-05

[24] LeCun, Y.: A Path Towards Autonomous Machine Intelligence Version 0.9.2, 2022-06-27 (2022)

[25] Bardes, A., Ponce, J., LeCun, Y.: VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning. arXiv. arXiv:2105.04906 [cs] (2022). https://doi.org/10.48550/arXiv.2105.04906 . http://arxiv.org/abs/2105.04906 Accessed 2026-04-23

[26] Vijayalakshmi, M., Mercy Shalinie, S., Yang, M.H., U., R.M.: Web phishing detection techniques: a survey on the state-of-the-art, taxonomy and future directions. IET Networks **9**(5), 235–246 (2020) https://doi.org/10.1049/iet-net.2020.0078 . Accessed 2026-09-06

[27] Kitterman, S.: Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1. Request for Comments RFC 7208, Internet Engineering Task Force (April 2014). https://doi.org/10.17487/RFC7208 . Num Pages: 64. https://datatracker.ietf.org/doc/rfc7208 Accessed 2026-09-07

[28] Crocker, D., Hansen, T., Kucherawy, M.: DomainKeys Identified Mail (DKIM) Signatures. Request for Comments RFC 6376, Internet Engineering Task Force (September 2011). https://doi.org/10.17487/RFC6376 . Num Pages: 76. https://datatracker.ietf.org/doc/rfc6376 Accessed 2026-09-07

[29] Kucherawy, M., Zwicky, E.: Domain-based Message Authentication, Reporting, and Conformance (DMARC). Request for Comments RFC 7489, Internet Engineering Task Force (March 2015). https://doi.org/10.17487/RFC7489 . Num Pages: 73. https://datatracker.ietf.org/doc/rfc7489 Accessed 2026-09-07

[30] Maniriho, P., Mahmood, A.N., Chowdhury, M.J.M.: A systematic literature review on Windows malware detection: Techniques, research issues, and future directions. Journal of Systems and Software **209**, 111921 (2024) https://doi.org/10.1016/j.jss.2023.111921 . Accessed 2026-09-07

[31] Afroz, S., Greenstadt, R.: PhishZoo: Detecting Phishing Websites by Looking at Them. In: 2011 IEEE Fifth International Conference on Semantic Computing, pp. 368–375 (2011). https://doi.org/10.1109/ICSC.2011.52 . https://ieeexplore.ieee.org/document/6061361 Accessed 2026-09-08

[32] Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C.D., Stamatopoulos, P.: A Memory-Based Approach to Anti-Spam Filtering for Mailing Lists. Information Retrieval **6**(1), 49–73 (2003) https://doi.org/10.1023/A:1022948414856 . Accessed 2026-04-23

[33] Fette, I., Sadeh, N., Tomasic, A.: Learning to detect phishing emails. In: Proceedings of the 16th International Conference on World Wide

Web. WWW '07, pp. 649–656. Association for Computing Machinery, New York, NY, USA (2007). https://doi.org/10.1145/1242572.1242660 . https://dl.acm.org/doi/10.1145/1242572.1242660 Accessed 2026-09-07

[34] Abdelnabi, S., Krombholz, K., Fritz, M.: VisualPhishNet: Zero-Day Phishing Website Detection by Visual Similarity. In: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security. CCS '20, pp. 1681–1698. Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3372297.3417233 . https://dl.acm.org/doi/10.1145/3372297.3417233 Accessed 2026-09-07

[35] Medvet, E., Kirda, E., Kruegel, C.: Visual-similarity-based phishing detection. In: Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks. SecureComm '08, pp. 1–6. Association for Computing Machinery, New York, NY, USA (2008). https://doi.org/10.1145/1460877.1460905 . https://dl.acm.org/doi/10.1145/1460877.1460905 Accessed 2026-09-07

[36] Rekeraho, A., Balan, T., Cotfas, D.T., Cotfas, P.A., Acheampong, R., Musuroi, C.: Sandbox Integrated Gateway for the Discovery of Cybersecurity Vulnerabilities. In: 2022 International Symposium on Electronics and Telecommunications (ISETC), pp. 1–4 (2022). https://doi.org/10.1109/ISETC56213.2022.10010327 . ISSN: 2475-7861. https://ieeexplore.ieee.org/abstract/document/10010327 Accessed 2026-09-08

[37] Tarapiah, S., Abbas, L., Mardawi, O., Atalla, S., Himeur, Y., Mansoor, W.: Evaluating the Effectiveness of Large Language Models (LLMs) Versus Machine Learning (ML) in Identifying and Detecting Phishing Email Attempts. Algorithms **18**(10), 599 (2025) https://doi.org/10.3390/a18100599 . Accessed 2025-11-18

[38] Uddin, M.A., Mahiuddin, M., Sarker, I.H.: An explainable transformer-based model for phishing email detection: A large language model approach. Computer Networks **277**, 112061 (2026) https://doi.org/10.1016/j.comnet.2026.112061 . Accessed 2026-09-09

[39] Alajaji, A.: Classical-First Selective Cascades for Resource-Constrained Phishing Email Detection. Electronics **15**(14), 3051 (2026) https://doi.org/10.3390/electronics15143051 . Accessed 2026-09-02

[40] Asliyuksek, H., Tonkal, O., Kocaoglu, R.: A Comparative Evaluation of a Multimodal Approach for Spam Email Classification Using DistilBERT and Structural Features. Electronics **14**(19), 3855 (2025) https://doi.org/10.3390/electronics14193855 . Accessed 2026-09-02

[41] Nardin, A.D., Zottin, S., Piciarelli, C., Foresti, G.L.: Deep Learning-Based Intrusion Detection Systems for Phishing Email Detection: A Short Survey (2025)

[42] Toth, R., Bisztray, T., Gruschka, N.: The Phish, The Spam, and The Valid: Generating Feature-Rich Emails for Benchmarking LLMs. arXiv. arXiv:2511.21448 [cs.CR] (2026). https://doi.org/10.48550/arXiv.2511.21448 . http://arxiv.org/abs/2511.21448 Accessed 2026-09-02

[43] Bluefox, A.: AVN Phishing Email Classification Dataset. https://www.kaggle.com/datasets/avnbluefox/avn-phishing-email-classification-dataset Accessed 2026-04-23

[44] Champa, A.I., Rabbi, M.F., Zibran, M.F.: Curated Datasets and Feature Analysis for Phishing Email Detection with Machine Learning. In: 2024 IEEE 3rd International Conference on Computing and Machine Intelligence (ICMI), pp. 1–7 (2024). https://doi.org/10.1109/ICMI60790.2024.10585821 . https://ieeexplore.ieee.org/document/10585821 Accessed 2026-04-23

[45] Klimt, B., Yang, Y.: Introducing the Enron Corpus. (2004). https://www.semanticscholar.org/paper/Introducing-the-Enron-Corpus-Klimt-Yang/135d89a35623359aa3af7ce6f95b0078c6acc43a Accessed 2026-09-11

[46] Cormack, G., Lynam, T.: TREC 2005 Spam Track Overview. In: Proceedings of the Fourteenth Text REtrieval Conference (TREC 2005) (2005)

[47] Cormack, G.: TREC 2006 Spam Track Overview. In: Proceedings of the Fifteenth Text REtrieval Conference (TREC 2005) (2006)

[48] Cormack, G.V.: TREC 2007 Spam Track Overview. In: Proceedings of the Sixteenth Text REtrieval Conference (TREC 2007) (2007)