Beyond the Description: Structured Metadata and Label Semantics for CVE-to-ATT&CK Mapping

Cédric Bonhomme
Computer Incident Response Center Luxembourg (CIRCL)
cedric.bonhomme@circl.lu [57B7 A70D]

Alexandre Dulaunoy
Computer Incident Response Center Luxembourg (CIRCL)
alexandre.dulaunoy@circl.lu [44E6 CBCD]

October 2026

Abstract

In prior work we trained a multi-label classifier that suggests MITRE ATT&CK techniques from CVE descriptions and found it label-quality bound: LLM-assisted label expansion brought no reliable gain [3]. This paper asks the complementary question: if more examples do not help, does more information per example, and per label? We extend the curated gold set with the structured metadata every CVE carries (CVSS vector, CWE class, affected products) and with the candidate prior derived from the CWE→CAPEC→ATT&CK chain, measure each signal under a multi-seed, same-session paired protocol, and replace the closed-vocabulary head with a bi-encoder that scores descriptions against the official technique texts. Most answers are negative. The derivation chain covers 3.3% of analyst-chosen techniques and degrades ranking at every fusion strength; the CVSS vector is null, products and derived candidates mildly harmful, and concatenating all signals lands below the baseline. The one conditional positive is the curated CWE: $+0.030$ recall@5, concentrated entirely on the KEV-sourced rows and not recoverable from a predicted CWE. The bi-encoder trades a small ranking cost for the largest consistent rare-technique gain of the study (macro-F1 $+0.037$, $+21\%$), but a five-fold label holdout shows its zero-shot promise does not survive fine-tuning: held-out techniques rank below a generic embedder (recall@5 0.12 vs. 0.24). Bucket-aware variants degrade the union task. The rare-technique tail is data-bound at both ends: in-vocabulary it yields to architecture; below the vocabulary floor, nothing we measure replaces curated examples. Datasets, models and code are released, and the bi-encoder is deployed in Vulnerability-Lookup for technique-to-CVE and CVE-to-CVE retrieval.

1 Introduction

A published CVE record tells a defender how bad a vulnerability is (the CVSS score), what kind of flaw it is (the CWE weakness class), and what is affected (the CPE product enumeration) — but not what an adversary will *do* with it. That last question is answered in the vocabulary of MITRE ATT&CK [21], and bridging the two worlds — mapping CVEs to the techniques their exploitation enables — is what connects vulnerability management to detection engineering and threat-informed defense. The Center for Threat-Informed Defense (CTID) defined a methodology for this mapping [4] and curated it for roughly a thousand CVEs; no manual effort scales to the tens of thousands of CVEs published each year. In prior work [3] we trained a multi-label classifier on the curated gold set and released it as a candidate-suggestion tool for analysts; the motivation in full lives there.

This paper starts from that paper's verdict. Its gold-size scaling curve showed every metric rising monotonically with the number of curated rows, while LLM-assisted label expansion at 0.39 labeler agreement added nothing at any mixture size and eroded the rare-technique tail at scale: the classifier is *label-quality bound, not data bound*. We use the term operationally, to distinguish the constraint from mere dataset size: weakly-agreeing labels improved nothing at any mixture size, whereas every measured gain in that paper traced to curated, analyst-grade rows — the binding quantity is the quality of the supervision obtainable, not the count of examples. Two consequences drive the present work. First, if more examples of that quality cannot be had, each existing example must be exploited harder — and the description is not the only inference-time signal a CVE carries. Every record arrives with analyst-produced structured metadata (the CVSS vector, the CWE class, the affected products) whose marginal value for this task has, to our knowledge, never been measured under a controlled protocol. Second, the binding constraint is the rare-technique tail (macro-F1 ≈ 0.18 against recall@5 ≈ 0.67), and a per-label classification head is structurally unable to fix it: it cannot learn a technique from a handful of examples, and it cannot score a technique it has

never seen at all. The label side also carries unused information — every ATT&CK technique has an official name and description — which a label-semantics architecture can exploit and a classification head cannot.

We therefore run the follow-up as an audit, not an advocacy exercise: each candidate signal and each architectural variant is measured against a same-session description-only baseline, five paired seeds each, with stratified breakdowns wherever coverage or provenance could confound the result. Most of the answers are negative, and we report them at full strength — the audit's value lies in closing paths as much as in the two interventions that survive it.

Contributions:

1. A controlled ablation of structured vulnerability metadata — CVSS vector components, CWE (gold and predicted), affected-product CPE — as auxiliary inputs to a CVE→ATT&CK classifier, under a multi-seed protocol with paired deltas. Verdict: no signal pays unconditionally, and the combined-signal arm falls below the baseline — at ∼1,000 gold examples serialization is signal selection, not accumulation. The only conditional positive is CWE (pooled $+0.030$ recall@5 over ten paired runs); a stratified replication with CWE-presence cross-strata locates the entire gain on the curated KEV stratum — rows whose gold CWE is merely *present* but NVD-grade gain nothing — and shows its magnitude attenuating under retraining; a cascade arm substituting a predicted CWE (27% agreement) recovers less than half the KEV gain and none consistently, attributing the gain substantially to curation quality rather than headroom (section 6.1).

2. A measured rejection of the deterministic CWE→CAPEC→ATT&CK derivation chain even in its weakest role, as an inference-time re-ranking prior: its candidate sets cover only 3.3% of the analyst-chosen techniques (parent level, test split), so every fusion strength degrades ranking — the chain's labels are not noise around the right answer but are systematically pointed at *different* techniques than analyst judgment (section 6.2).

3. A label-semantics bi-encoder that scores CVEs against official technique descriptions, evaluated on the rare-technique tail, including techniques below the vocabulary floor of the released classifier — a zero-shot setting a classification head cannot address at all. On the shared 53-technique vocabulary it trades a small consistent recall@5 cost (-0.024) for the largest consistent macro-F1 gain of any arm in this paper ($+0.037$, a 21% relative improvement on the binding constraint); under open-vocabulary ranking over all 222 active parent techniques it retains $2.3\times$ the ranking quality of a generic zero-shot sentence embedder. The zero-shot promise, however, does not survive fine-tuning: a five-fold label-holdout evaluation shows held-out techniques ranking *worse* than under the generic embedder (recall@5 0.12 vs. 0.24, negative in four of five folds) — label semantics regularizes the trained rare classes but does not extend below the vocabulary floor (section 6.3).

4. A negative result on bucket-aware training: predicting the CTID role structure (exploitation / primary / secondary impact) as separate per-technique heads, or adding a free tactic-level auxiliary loss, both consistently degrade the flattened-union task at this data size — although the role-conditioned heads themselves reach usable ranking quality (0.78 recall@5 for the exploitation technique), establishing role prediction as a capability with a measurable union-accuracy price (section 6.4).

5. An extended public dataset (v2 of `CIRCL/vulnerability-attack-techniques`) carrying the joined metadata columns, a released bi-encoder checkpoint [10], and the full trainer logs behind every table. Section 8 describes how the bi-encoder is served to Vulnerability-Lookup for the two retrieval directions a classification head cannot offer.

2 Background and Related Work

2.1 Prior work: the gold-set classifier and the expansion null result

The gold set of [3] joins the two CTID curation efforts — the 2021 "Mapping ATT&CK to CVE for Impact" [4] mappings (788 CVEs) and the later KEV-catalog mappings [6, 13] (392, plus 27 in both) — into 1,207 CVEs with analyst-assigned techniques, published as `CIRCL/vulnerability-attack-techniques` [11]. Sub-techniques are collapsed to parents and techniques with fewer than five training examples are dropped, leaving the 53-technique vocabulary of the released model (nominal across seeds; section 5); a `roberta-base` [19] encoder with a sigmoid head is trained with per-label-weighted BCE. The paper's protocol corrections matter for everything that follows: best-checkpoint selection on a gold-only validation carve-out (never the test split), the test split evaluated once per trained model, and five seeds with paired deltas — an earlier apparent LLM-expansion gain was traced precisely to violating these (test-split checkpoint selection amplified by single-seed evaluation noise). The numbers of record for the released classifier are recall@5 0.673 ± 0.019 and macro-F1 0.177 ± 0.014. The expansion experiment added up to 984 LLM-labeled CVEs at 0.39 labeler–gold agreement: no reliable gain at any expansion size, and measurable erosion of the rare-technique tail at scale.

2.2 Automated CVE-to-ATT&CK mapping

Existing automation falls into three families. CVE2ATT&CK [17] is the supervised neighbor: BERT-family encoders fine-tuned on annotated CVE descriptions, description-only inputs. SMET [1] is the unsupervised end: CVE text is mapped to techniques by semantic similarity in an embedding space, with no task supervision — our

zero-shot sentence-embedding reference (section 6.3) reproduces this setting. CVE2CAPEC [16] and BRON [18] are deterministic: they traverse the published CWE→CAPEC→ATT&CK cross-references, an approach whose output we measure against analyst judgment in section 6.2. (TRAM [5] addresses the adjacent problem of mapping threat *reports*, not CVE records.) To our knowledge, no prior work measures the marginal value of structured CVE metadata for this task under a controlled, seed-replicated protocol, and no supervised label-semantics model for this task has been evaluated on techniques outside its training vocabulary.

2.3 Structured vulnerability metadata

The CVSS base vector [14, 15] is an analyst-produced, machine-readable summary whose components align with the CTID methodology's structure by design: the exploitability metrics (attack vector, privileges required, user interaction) describe the preconditions the *exploitation technique* bucket captures, and the C/I/A impact metrics describe what the *primary/secondary impact* buckets encode. CWE [22] is a hierarchical weakness taxonomy; we use parent-level classes throughout, matching the granularity of the CIRCL CWE classifier [7] from the VulnTrain model family [2], which the cascade arm of section 6.1 uses to substitute predictions where no curated assignment exists. CPE identifies affected vendors and products; product identity carries an ecosystem prior (a WordPress plugin and a network appliance invite different post-exploitation behavior), which is the hypothesis the products arm tests.

2.4 Label-semantics and few-/zero-shot multi-label classification

A per-label classification head learns one output row per label, so rare labels learn from only their own examples and unseen labels cannot be scored at all. Label-semantics architectures address both by representing each label as *text* and scoring input–label pairs in a shared representation space: bi-encoders embed input and label independently and score by similarity (Sentence-BERT [20] is the canonical recipe; cross-encoders re-rank pairs jointly at higher cost). Parameters are shared across all labels, rare labels borrow strength from the label text, and a new label can in principle be added by writing its description. SMET [1] instantiates the unsupervised extreme of this spectrum for our task. What has been missing is the measurement in between: whether task supervision on a small gold set *preserves* the zero-shot property while improving in-vocabulary performance — the question section 6.3 answers, negatively.

3 Candidate Signals and Dataset Extension

3.1 Signals

Four candidate signals, each available at inference time for an arbitrary CVE:

- **CVSS vector components.** The base-metric vector from the CVE record (highest available version; CNA container preferred over ADP within a version). The exploitability metrics should correlate with the exploitation technique and the C/I/A metrics with the impact techniques (section 3.3 argues why this correlation is signal, not leakage). Coverage: 72% overall, 100% on the KEV subset; the version mix (v2/v3.x/v4.0) is handled by the version-neutral verbalization of section 4.1.
- **CWE, gold and predicted.** The weakness class — arguably a compressed expert summary of the flaw's mechanism — in two variants: the curated assignment from the CVE record (84% coverage), and the top-1 parent-level prediction of the CIRCL CWE classifier [7] run on title+description (100% coverage by construction, 27.3% ancestor-level agreement with gold). The predicted variant makes the pipeline self-contained where no assignment exists, and the gold/predicted contrast isolates signal quality (section 6.1).
- **Affected products.** Vendor/product pairs from the CVE record merged with CPE-derived ones (67%/93% coverage) — the ecosystem prior.
- **Derived candidates.** The CVE2CAPEC candidate set (`techniques_derived`), derived through CWE→CAPEC→ATT&CK. Kept strictly out of the training labels, as in [3]; tested here both as an appended input signal and as an inference-time re-ranking prior.

3.2 Dataset v2

The generation pipeline now fetches the raw CVE record of every gold CVE from Vulnerability-Lookup and extracts: the CVSS vector string (highest available version; the CNA container preferred over ADP within a version), the CWE assignments (CNA and ADP `problemTypes`), and the affected vendor/product pairs; CPE lists are joined from `CIRCL/vulnerability-scores`. The extension is strictly additive — identical rows, splits, and v1 columns, verified against the published v1 — so every prior result remains reproducible on v2.

Among the 869 vectors: 677 CVSS v3.1, 173 v3.0, 18 v4.0, 1 v2.0. Coverage differs sharply between the two label sources, so all metadata-ablation results are additionally reported stratified by `label_sources` (section 5).

3.3 Coverage and leakage considerations

Three considerations shape how the metadata results must be read. First, *coverage is confounded with the label source* (table 1): the KEV subset has full CVSS/CWE coverage via the CISA ADP container while the 2021 subset does not, so any per-signal gain could be a subset effect in disguise — hence the stratified reporting and the CWE-presence cross-strata of section 6.1. Second, *CVSS-as-input is not label leakage* even though the CTID impact buckets correlate with the C/I/A impact metrics by construction: the vector is produced

Table 1: Metadata coverage per label source (dataset v2). The CISA ADP Vulnrichment container gives the KEV subset full CVSS/CWE coverage; the 2021 subset's gaps are the source-truth limit for old CVEs.

Subset	CVEs	cvss_vector	cwes	affected_products	cpes
all	1,207	72.0%	84.3%	67.4%	93.2%
ctid_kev	392	100%	100%	79.8%	79.1%
ctid_cve	788	57.1%	76.0%	62.2%	100%
both	27	100%	100%	40.7%	100%

by CNA/ADP analysts from the vulnerability itself, not from the ATT&CK mapping, and is available at inference time for any CVE — it is a legitimate input feature whose correlation with the labels is precisely why it is a candidate signal. Third, the *cascade arm carries a mild feature-level leakage concern in the optimistic direction*: the CWE classifier used to produce `cwes_predicted` is trained on NVD-derived vulnerability data that predates our experiments and therefore likely includes the descriptions and gold CWE assignments of our test CVEs. Its predictions on test rows may thus be optimistically accurate relative to a truly unseen CVE. Since the cascade result is *negative* (section 6.1), the leakage can only have flattered the arm; the conclusion that predicted CWEs do not recover the gold-CWE gain would survive, and likely strengthen, on a leakage-free replication.

4 Methods

4.1 Metadata as text: serialization ablation

Each enabled signal is verbalized and appended to the title+description input as one line under a constant section header. CVSS vectors are rendered *version-neutrally* — v2, v3.x and v4.0 metrics map to the same phrases (e.g. `AV:N` → "attack vector network") — so the version mix of table 1 does not leak into the text representation; temporal and environmental metrics are skipped. CWEs use their id+name display strings; products merge the record's vendor/product pairs with CPE-derived ones (deduplicated); the derived signal lists the CVE2CAPEC candidate technique IDs. Missing metadata renders as an explicit `unknown` under the same header, so absence is itself a signal and the text structure is constant across rows. With no signals enabled the input is byte-identical to the v1 text, making the `desc` arm an exact re-run of the prior paper's configuration. The enabled signals are recorded in the trained model's configuration and read back by the evaluation code, so train and test inputs cannot diverge. The grid: `desc` / +CVSS / +CWE / +products / +derived / all, five seeds each, under the protocol of section 5.

4.2 Re-ranking with the derivation-chain prior

Late fusion of the released classifier's logits with the CVE2CAPEC candidate set (`techniques_derived`, collapsed to parent techniques): $s'_t = s_t + \alpha \cdot \mathbf{1}[t \in \text{derived}]$, with the mask variant as the $\alpha \to \infty$ limit. No retraining; α tuned on the validation carve-out reconstructed with the trainer's split parameters (fraction 0.1, seed 42); test evaluated once. Implemented as `--prior`/`--prior-alpha` on the released validator. Results in section 6.2.

4.3 Label-semantics architectures

A per-label classification head allocates its output parameters per technique, so it cannot score a technique it was never trained on — the vocabulary floor is structural. The label-semantics alternative replaces the head with the *official technique text*: a single shared `roberta-base` encoder embeds both the CVE input (identical serialization to the classifier arms) and each technique's name and description from the enterprise ATT&CK STIX bundle (citation markers and markup stripped, truncated to 256 tokens), mean-pooled and L_2-normalized in the style of SBERT [20]. Concretely, for a text tokenized to T positions with last-layer hidden states $h_1, \ldots, h_T \in \mathbb{R}^{768}$ and attention mask $m_i \in \{0, 1\}$,

$$u = \frac{\bar{u}}{\|\bar{u}\|_2}, \qquad \bar{u} = \frac{\sum_{i=1}^{T} m_i \, h_i}{\sum_{i=1}^{T} m_i}. \tag{1}$$

The mask-weighted mean turns a text of any length into one vector of fixed size — the encoder's hidden width, inherited unchanged because no projection layer is added — while excluding padding positions and deliberately not using the `<s>` vector, which `roberta-base` was never trained to summarize with. The unit norm makes the dot product of two such vectors their cosine, which is what the affine below was learned on and what the retrieval of section 8 indexes. Equation (1) is the one function every consumer must reproduce exactly; the trainer, the validator, the gateway and the reference implementation in the documentation implement it identically, and listing 1 gives it in full. The score is a learnable affine over the cosine, $s_t = a \cdot \cos(u, v_t) + b$ (initialized a=10, b= − 5), trained end-to-end with the same per-label-weighted BCE loss, validation carve-out, and per-seed vocabulary as the classification head; CVE inputs are truncated to 512 tokens, and technique texts are re-encoded with gradients at every step. Because the label representation is text rather than a learned output row, the trained model can rank *any* technique with an official description. Everything needed to reproduce the score is shipped with the released checkpoint: the training-time technique texts as a JSON file next to the weights, and the vocabulary, the truncation lengths and the learned a and b in the model configuration. In the released seed-42 checkpoint a and b ended training at 10.00 and −5.00, unchanged from their initialization to two decimals.

An informal picture of the difference: the classification head is a receptionist who has memorized which of 53 desks each kind of visitor is sent to, but has never read what any desk does; the bi-encoder is a receptionist who has read every desk's job description and seats visitors by matching what they say against it. The first is sharper for the regular visitors it was trained on. The second can handle a desk created yesterday, and, because visitors and desks share one description space, can also say which visitors resemble each other — the property that makes technique→CVE and CVE→CVE retrieval possible with the same weights.

Three evaluations follow. (i) *Head-to-head*: the standard 53-technique vocabulary, paired per seed against a fresh description-only classification head. (ii) *Open-vocabulary*: ranking over all 222 active enterprise parent techniques, reported overall and restricted to in-vocabulary and below-floor gold — the natural below-floor stratum is tiny (n=9 test rows), so it is indicative only. (iii) *Label-holdout zero-shot*: the 53 vocabulary techniques are dealt into five folds round-robin by descending train frequency; each fold retrains the bi-encoder with those techniques withheld entirely — neither a labeled example nor the technique text is seen — and evaluates open-vocabulary, scoring only the held-out gold (30–53 affected test rows per fold). This is the powered version of the below-floor stratum. A cross-encoder re-ranker over the top-k candidates is left to future work.

4.4 Bucket-aware training

The CTID methodology assigns each technique to a CVE in one of three roles — exploitation technique, primary impact, secondary impact — which the flattened union target discards. Two arms test whether that structure carries trainable signal. *Bucket multi-task*: the classification head is widened to four per-technique blocks (the three named buckets plus an uncategorized slot for the ∼27% of training occurrences the source mapping leaves unbucketed), trained with BCE over all slots. The union task — the max over buckets — drives checkpoint selection and every reported metric, so the union target is identical to the baseline's and the paired comparison is exact; per-bucket ranking quality is additionally reported as the new capability (predicting not just *which* technique but *in what role*). *Tactic auxiliary*: an auxiliary tactic-level multi-label head on the [CLS] representation, with targets derived for free from the gold techniques via the STIX kill-chain phases — including below-floor gold techniques the technique head cannot express. The loss is $\mathrm{BCE}(\text{techniques}) + \lambda \cdot \mathrm{BCE}(\text{tactics})$ with $\lambda = 0.5$ fixed a priori (untuned); the auxiliary head is dropped at save time, so the checkpoint remains a standard classifier. The tactic vocabulary is derived from the bundle (15 tactics in the current enterprise release).

5 Experimental Protocol

The protocol is inherited unchanged from [3]: best-checkpoint selection on a gold-only validation carve-out (fraction 0.1, split with the training seed), the test split evaluated once per trained model, five fresh seeds (42–46) per arm, paired per-seed deltas with the $|\Delta| > 2 \cdot \mathrm{SEM}$ consistency heuristic, and the label vocabulary frozen across arms. One precision the archived logs force: the five-example cutoff is applied to the gold train split *after* the seed-dependent validation carve-out, so the vocabulary holds 53 techniques for seed 42 — the released configuration and this paper's nominal "53-technique" vocabulary — but 54–56 for seeds 43–46, one to three techniques sitting exactly at the cutoff, and the test rows with an in-vocabulary technique number 118 (119 for seed 45). Within a seed every arm of every grid shares the carve-out and hence the vocabulary, so all paired deltas are like for like; the vocabulary is frozen across arms, not across seeds. Three additions. First, stratified breakdowns by `label_sources` and gold-CWE presence (the coverage confound of section 3.3). Second, the open-vocabulary and label-holdout evaluations of section 4.3. Third, same-session baselines: the description-only baseline moved by up to 0.018 recall@5 between identically-seeded training sessions (replicated five times across this paper's grids), which exceeds typical seed spread, so every paired comparison retrains its baseline in the same session rather than reusing an earlier grid's.

One protocol lesson emerged the hard way and is reported so others can avoid it: *checkpoint selection can degenerate silently when the selection metric interacts with the architecture*. The bi-encoder's affine calibration was initially configured so that its maximum attainable logit was zero; every thresholded prediction was therefore empty and macro-F1 — the selection metric shared with the classifier arms — was exactly 0.0 at every epoch. With a constant selection metric, `load_best_model_at_end` keeps the first epoch, and the first sweep silently evaluated near-untrained models (test recall@5 0.378 while the validation series in the trainer logs climbed to ∼0.52). No error is raised anywhere in this failure mode; the paired comparison is simply invalid. The sweep was re-run selecting the bi-encoder's checkpoints on validation recall@5 (a ranking model warrants a ranking metric, independent of calibration), and the per-epoch validation series in the released trainer logs allow the check to be repeated. The broken sweep's logs are archived alongside the valid ones.

6 Results

6.1 Metadata ablation

Table 2 reports the six-arm grid (five seeds per arm, roberta-base, identical hyper-parameters; only the input serialization of section 4.1 varies).

CWE is the only signal that pays, and in this grid it pays everywhere: the paired deltas vs. the baseline clear the consistency threshold on all four metrics (recall@5 +0.046, recall@3 +0.025, micro-F1 +0.018, macro-F1 +0.017). It is also the *most stable* arm — the smallest seed spread on every metric (e.g. ±0.009 recall@5 vs. ±0.017 for the baseline): the weakness class does not just add information, it anchors

Table 2: Marginal contribution of each metadata signal (five seeds, mean ± std). * marks arms whose paired per-seed Δ vs. the description-only baseline satisfies $|\Delta| > 2 \cdot \mathrm{SEM}$ on that metric — the consistency heuristic of [3], necessary but not sufficient. Best value per column in bold.

Input	recall@5	recall@3	micro-F1	macro-F1
description only (baseline)	0.664 ± 0.017	0.521 ± 0.011	0.408 ± 0.011	0.182 ± 0.018
+ CVSS vector	0.666 ± 0.026	0.529 ± 0.032	0.405 ± 0.018	0.181 ± 0.020
+ CWE (gold)	$\mathbf{0.710 \pm 0.009}^*$	$\mathbf{0.546 \pm 0.013}^*$	$\mathbf{0.427 \pm 0.011}^*$	$\mathbf{0.199 \pm 0.008}^*$
+ affected products	0.646 ± 0.016	0.517 ± 0.021	0.396 ± 0.012	$0.166 \pm 0.011^*$
+ derived candidates	0.642 ± 0.021	0.523 ± 0.023	0.398 ± 0.013	0.168 ± 0.009
all signals	0.659 ± 0.027	$0.506 \pm 0.023^*$	$0.390 \pm 0.009^*$	0.176 ± 0.015

training against seed-level variance in what the model learns from free text. The macro-F1 gain matters most for the use case: CWE lifts the technique tail, not just the frequent heads.

The CVSS vector is a null result (+0.001 recall@5). Despite the construct-level correlation between the C/I/A impact vector and the CTID impact-oriented techniques (section 3.1), a verbalized vector adds nothing the description did not already say — consistent with severity being largely recoverable from the description, as CIRCL's deployed severity classifiers demonstrate in the opposite direction. Affected products and the derived candidate IDs are mildly harmful (products: macro-F1 −0.016, consistent); the derived arm confirms from the input side what section 6.2 shows from the score side.

The second headline is the *all* arm: concatenating every signal destroys the CWE gain and lands below the baseline (recall@3 −0.015, micro-F1 −0.018, both consistent). With 512-token inputs and ∼1,000 training examples, serialization is signal *selection*, not signal accumulation — three noisy fields dilute the one informative field rather than averaging out.

Stratified replication. Because the grid of table 2 retained no checkpoints, the stratified analysis retrains the full grid — same seeds, same hyper-parameters, per-run save directories — and evaluates each run once on test with per-`label_sources` breakdowns (table 3). This makes it a *replication* as much as a stratification, and the replication carries its own lesson: under GPU-level nondeterminism alone (identical seeds), the overall CWE gain attenuates from +0.046 to +0.014 recall@5, and only macro-F1 (+0.012) still clears the $2 \cdot \mathrm{SEM}$ threshold. Seed spread understates true run-to-run variance, and the consistency heuristic is — as stated — necessary but not sufficient. Across both grids the qualitative ordering is untouched: CWE is the only arm that is ever consistently positive and never consistently negative on any metric, products the only arm consistently harmful in both.

The stratification answers the coverage question, and the answer is the confounded one: the CWE gain lives entirely on the `ctid_kev` rows (+0.053 recall@5, consistent), where gold-CWE coverage is 100%, and is absent on the `ctid_cve` rows (−0.003), where coverage is 76%. A second evaluation pass over the same 30 models (no retraining) adds gold-CWE-*presence* cross-strata, and rules out the coverage explanation directly: even on the 64 `ctid_cve` rows that *do* carry a gold CWE, the CWE arm gains nothing (+0.004 recall@5, null on every metric — table 3, third column). The effect is a property of the stratum, not of token availability. Two stratum-level explanations remain. First, headroom: KEV rows are markedly harder for the description-only model (0.614 vs. 0.700 recall@5 on the CWE-covered `ctid_cve` rows — KEV descriptions of actively exploited vulnerabilities are terse, and the KEV-era technique distribution differs from the 2021 mapping), so a structured signal has more room to help. Second, signal quality: KEV CWE assignments are curated alongside CISA's KEV enrichment process, whereas the 2021 rows carry ordinary NVD assignments. The stratum is small (n=36, hence the wide baseline spread), so we treat the location of the gain as established but its magnitude as indicative. Pooling the two grids (ten paired runs per arm) gives the honest overall effect size: recall@5 +0.030, micro-F1 +0.011, macro-F1 +0.015 (all > 2·SEM); recall@3 +0.010 does not clear the threshold.

Cascade arm: predicted CWE. With coverage ruled out, a cascade arm separates the two remaining explanations. We run the CIRCL CWE classifier [7] (parent-level, 303 classes) over every row's title and description, add its top-1 prediction to the dataset as `cwes_predicted` (v2.1; 100% coverage by construction, 27.3% ancestor-level top-1 agreement with the gold CWE), and train a third fresh grid — description-only, +CWE (gold), +CWE (predicted), five seeds each, all arms retrained in the same session so the paired deltas are unaffected by the session-level variance documented above. The predicted CWE is verbalized under the same `CWE:` header as the gold arm, so signal *quality* is the only difference between the two CWE arms. If the KEV gain is headroom, an NVD-grade predicted CWE should still buy most of it; if it is curation quality, the gain should collapse.

It collapses (table 4). The gold-CWE KEV gain replicates a third time, stronger than in either previous grid (+0.082 recall@5, micro-F1 +0.034, macro-F1 +0.014, all consistent) — the one finding of the ablation that survives every retraining. The predicted CWE recovers less than half of it (+0.038 recall@5) and clears the consistency threshold on

Table 3: Stratified replication grid (five fresh seeds, recall@5). The baseline row is absolute; the remaining rows are paired per-seed deltas vs. the baseline within each stratum. `ctid_cve`∩CWE restricts to the `ctid_cve` rows whose gold CWE is present. * marks $|\Delta| > 2 \cdot \text{SEM}$. Macro-F1 is not comparable across strata (labels absent from a stratum contribute zero) and is therefore omitted here.

Input	overall (n=118)	`ctid_cve` (n=81)	`ctid_cve`∩CWE (n=64)	`ctid_kev` (n=36)
description only	0.682 ± 0.018	0.715 ± 0.007	0.700 ± 0.020	0.614 ± 0.068
+ CVSS vector	-0.020^*	-0.027^*	-0.037^*	-0.003
+ CWE (gold)	$+0.014$	-0.003	$+0.004$	$\mathbf{+0.053}^*$
+ affected products	-0.041^*	-0.048^*	-0.050^*	-0.028
+ derived candidates	-0.014	-0.025	-0.022	$+0.013$
all signals	-0.010	-0.029^*	-0.023	$+0.023$

Table 4: Cascade grid (five fresh seeds, recall@5). Baseline row absolute, remaining rows paired per-seed deltas within each stratum. * marks $|\Delta| > 2 \cdot \text{SEM}$.

Input	overall (n=118)	`ctid_cve` (n=81)	`ctid_kev` (n=36)
description only	0.664 ± 0.010	0.701 ± 0.015	0.587 ± 0.042
+ CWE (gold)	$+0.026$	$+0.003$	$\mathbf{+0.082}^*$
+ CWE (predicted)	-0.006	-0.027	$+0.038$

no metric in no stratum; overall it is exactly null (-0.006 recall@5, $+0.001$ micro-F1). The KEV gain is therefore driven substantially by curation quality, not by headroom alone — a 27%-accurate NVD-grade prediction cannot substitute for the curated assignment, and the positive-but-inconsistent KEV trend is at most a residual headroom component. Operationally the cascade is closed: at current CWE-classifier accuracy, feeding predicted CWEs to the technique classifier costs nothing but buys nothing, so the honest deployment configuration remains description-only, with gold CWE appended when a curated assignment exists. As a side observation, the description-only baseline moved again between sessions (0.682 vs. 0.664 overall recall@5, identical seeds and code), a third replication of the run-to-run variance noted above.

Description length. A recurring triage hypothesis is that very short descriptions, under 256 characters, are too thin to map. The per-row predictions of the five same-session description-only heads (exported by the validator and committed with the paper) answer it with the predeclared audit of the repository script: recall@5 by character band of the description alone, and a two-sided permutation test of the short–long contrast in which the length indicator is shuffled only within `label_sources` strata, since 17 of the 36 KEV rows are short against 26 of the 81 CTID-CVE rows. Table 5 gives the per-seed result. Short descriptions are not harder: the adjusted short-minus-long recall@5 is positive in four seeds of five and significant in none, and the 128–255 band has the highest hit@5 of the four bands in every seed (0.886–0.943). Averaged over seeds, recall@5 is 0.633 under 128 characters (n=9, too few to read), 0.702 at 128–255 (n=35), 0.661 at 256–511 (n=54) and 0.642 at 512 and above (n=20). The design has the power of 44 short against 74 long rows, so it rules out a large penalty, not a small one; it also tests association, since short rows may differ in technique frequency and source. What it does establish is that character count below 256 is not, on this test set, a marker of difficulty, and therefore not by itself a trigger for analyst routing.

Table 5: Description-length audit on the five same-session description-only heads (gold test split; 44 rows under 256 characters, 74 at or above, 75 for seed 45). Δ is the row-count-weighted mean of the within-`label_sources` short-minus-long recall@5; p from 10,000 stratified permutations. Script and inputs in the repository.

Seed	Δ recall@5 (short $-$ long)	p
42	-0.015	0.83
43	$+0.047$	0.50
44	$+0.029$	0.70
45	$+0.120$	0.12
46	$+0.047$	0.52
mean	$+0.046$	

6.2 Derivation-chain re-ranking

We fuse the released checkpoint's logits with the CVE2CAPEC candidate set as $s'_t = s_t + \alpha \cdot \mathbf{1}[t \in \text{derived}]$, with *mask* (candidates rank strictly first) as the $\alpha \to \infty$ limit. α is tuned on the validation carve-out, never on test. The validation sweep is monotonically negative — MRR 0.598 with no prior, 0.590 at α=0.25, 0.287 at α=4 — so the tuned choice is α=0; we report the mildest boost and the mask limit on test once, for the record (table 6). Inference is deterministic (a fixed checkpoint; no seed variance is involved).

Table 6: Derived-prior re-ranking on the gold test split (118 examples, the released model's 53-technique vocabulary). The candidate sets cover 3.3% of the gold techniques (parent level) — the hard ceiling of any mask-style filter.

Configuration	recall@1	recall@3	recall@5	recall@10	MRR
classifier (no prior)	0.209	0.518	0.644	0.858	0.616
+ boost (α=0.25)	0.208	0.509	0.644	0.848	0.608
+ mask	0.065	0.261	0.494	0.756	0.340

The mechanism is coverage, not weighting. At the parent-technique level the derived candidate sets contain on average 3.3% of the gold techniques on the test split (6.7% on train), and the most frequent derived parents (T1134, T1562, T1574 — the table-expansion artifacts quantified in [3]) are entirely disjoint from the most frequent gold parents (T1190, T1059, T1203, T1068). The chain's output is not noise scattered around the analyst answer that a learned re-weighting could denoise; it points systematically at *different* techniques. This closes the last proposed use of the derivation chain for this task: its labels serve neither as training targets [3] nor as an inference-time prior.

6.3 Label semantics and the rare-technique tail

Head-to-head on the shared vocabulary. Both arms retrained fresh, five paired seeds (table 7). The bi-encoder is competitive with the classification head, and the trade is informative: a small, consistent recall@5 cost (-0.024) against the largest consistent macro-F1 gain of any arm in this paper ($+0.037$, from 0.176 to 0.212 — a 21% relative improvement on the metric the prior paper identified as the binding constraint). MRR is statistically indistinguishable. Anchoring every technique to its official description evidently regularizes the rare in-vocabulary classes, at a slight cost in head-class ranking sharpness.

Stratified, the recall@5 deficit is not uniform: it concentrates entirely on the 17 test rows without a gold CWE (-0.220^* — rows on which the classification head happens to do best, 0.778 recall@5), while on the 101 CWE-covered rows the bi-encoder matches or exceeds the head ($+0.009$ recall@5, $+0.020^*$ micro-F1, $+0.042^*$ macro-F1) and on the KEV stratum it is ahead on every metric ($+0.027^*$ macro-F1). The CWE-absent group is small; we report the location of the deficit without interpreting it further.

Open vocabulary. The setting the classification head cannot enter at all: ranking over all 222 active enterprise parent techniques (table 8). Expanding the candidate set $53 \rightarrow 222$ costs the bi-encoder 0.13 recall@5 ($0.643 \rightarrow 0.515$), but it retains $2.3\times$ the ranking quality of the generic zero-shot sentence-embedding reference (all-MiniLM-L6-v2 scoring the same technique texts): fine-tuning on $\sim$1,000 gold examples teaches the encoder what a CVE-to-technique match looks like well beyond generic semantic similarity. On the natural below-floor tail, however, the fine-tuned model shows no advantage over the generic embedder (0.136 ± 0.049 vs. 0.150 recall@5) — an n=9 stratum, so indicative only. Whether fine-tuning transfers *any* ranking ability to techniques it never saw is exactly the question the label-holdout folds answer with power.

Label-holdout zero-shot. The powered answer is negative, and more sharply than a mere null (table 9). Each fold's model remains healthy on its trained labels (0.544 ± 0.037 recall@5 on in-vocabulary gold, open candidates), yet on the held-out gold the fine-tuned encoder averages 0.123 ± 0.144 recall@5 — *below* the generic MiniLM embedder scoring the identical fold gold (0.237 ± 0.117; paired per-fold $\Delta = -0.114$, negative in four of five folds, under the $2 \cdot \mathrm{SEM}$ threshold only because fold-to-fold spread is huge). Fine-tuning on the head vocabulary does not merely fail to transfer; it specializes the embedding space toward the trained technique texts and away from unseen ones, degrading zero-shot ranking below an off-the-shelf baseline. The fold spread is strongly bimodal (three folds near zero, two with partial transfer) and partially tracks the MiniLM difficulty of the same fold, i.e. some held-out technique sets are intrinsically easier to match from description text.

The E4 verdict is therefore two-sided. In-vocabulary, label semantics is the honest architectural improvement this paper finds: the largest consistent macro-F1 gain of any arm, bought with a small head-class ranking cost. Below the vocabulary floor, the zero-shot promise of the architecture does not survive fine-tuning, and the fallback that actually ranks unseen techniques best — a generic sentence embedder at 0.24 recall@5 over 222 candidates — is far from operationally usable. The tail below the floor remains data-bound: no architecture evaluated here reaches it, and nothing in these experiments suggests one can substitute for curated examples. The bi-encoder is released as `CIRCL/vulnerability-attack-technique-biencoder` [10] — seed-42 configuration, shipping the training-time technique texts and scoring calibration alongside the weights, with the zero-shot caveat stated on its model card. The practical guidance is stated there and holds here: the released classification head remains the default for top-5 candidate suggestion; the bi-encoder is the variant of choice when rare-technique coverage matters more than head-class ranking sharpness.

6.4 Bucket-aware training

Both bucket-aware arms lose to the flattened-union baseline, and the bucket multi-task arm loses badly: every metric is consistently negative overall (table 10), with macro-F1 — the metric the structure was hoped to help — taking the largest

Table 7: Classification head vs. bi-encoder on the shared per-seed vocabulary (nominally 53 techniques; gold test split, n=118, 119 for seed 45; five paired seeds). * marks $|\Delta| > 2 \cdot \text{SEM}$.

Model	recall@5	MRR	micro-F1	macro-F1
classification head	0.667 ± 0.015	0.632 ± 0.013	0.409 ± 0.006	0.176 ± 0.016
bi-encoder	0.643 ± 0.019	0.626 ± 0.018	0.419 ± 0.018	0.212 ± 0.011
paired Δ	-0.024^*	-0.006	$+0.010$	$+\mathbf{0.037}^*$

Table 8: Open-vocabulary ranking over all 222 active parent techniques (recall@5; bi-encoder mean $\pm$ std over five seeds, MiniLM deterministic). Each column restricts scoring to the named gold subset.

Model	overall (n=121)	in-vocab gold (n=118)	below-floor gold (n=9)
bi-encoder (fine-tuned)	0.515 ± 0.020	0.520 ± 0.022	0.136 ± 0.049
MiniLM (zero-shot)	0.227	0.228	0.150

hit (-0.072, from 0.182 to 0.109). The mechanism is supervision starvation: the gold set carries ∼2,500 positive technique occurrences in training, and splitting them across four role-specific blocks leaves each block a third of the signal the union head concentrates — at ∼1,000 examples the union task needs all of it. The tactic-auxiliary arm is milder but still measurably harmful at the pre-registered λ=0.5 (MRR -0.061, recall@3 -0.065, both consistent): the "free" hierarchical gradient competes with the technique loss rather than regularizing it. A vanishing λ trivially recovers the baseline, so the honest statement is not that an auxiliary tactic loss cannot be neutral, but that it shows no evidence of benefit and measured harm at the one weight we pre-registered.

The structure itself is learnable — it just cannot be had for free. Scored against role-restricted gold, the multi-task model's own bucket heads rank usefully (table 11): 0.781 recall@5 for the exploitation technique, 0.682 for primary impact, with the expected difficulty ordering (secondary impact, the most indirect judgment, is hardest). These numbers are not comparable to the union rows — role-restricted gold sets are smaller — but they establish that role-conditioned ranking, a capability the CTID methodology defines and the union model cannot express, is within reach; it currently costs union accuracy that a deployment almost certainly values more.

With E5 the constructive sweep of this paper is complete, and its pattern is uniform: at ∼1,000 curated examples, every *free* training signal we tried — LLM-expanded labels [3], derivation-chain candidates, concatenated metadata, role structure, tactic hierarchy — either does nothing or hurts. The two interventions that paid were curated per-example information (the gold CWE, on the stratum where it is curated) and label-semantic regularization (the bi-encoder's macro-F1 gain). The description-only baseline also moved between sessions a fifth time (0.668 here), keeping the same-session retraining rule load-bearing.

7 Discussion and Limitations

Why free signals fail here. The audit's uniform pattern — every free signal null or harmful, one curated signal conditionally positive — admits a consistent reading. At ∼1,000 training examples the encoder cannot learn to *ignore* an appended signal that is noisy or redundant: every serialized token competes for the same small supervision budget, so concatenation is signal selection, not accumulation (the `all` arm landing below the baseline is the direct demonstration). The CWE result sharpens this: the same token format under the same header helps when the assignment is KEV-grade curated and does nothing when it is NVD-grade — at 27% agreement, predicted — so what the model extracts from "CWE:" lines is not taxonomy structure but analyst signal quality. The same budget argument explains the E5 failures from the target side: splitting supervision across role-specific heads, or spending part of the gradient on a tactic auxiliary, starves the union task that deployment actually uses.

Run-to-run variance is a first-class result. The description-only baseline moved five times in this series between training sessions that were identically seeded and identically coded (0.664–0.682 overall recall@5) — more than typical seed spread within a session. Practically: single-grid consistency checks overstate certainty (our first CWE grid cleared every check and then attenuated under mere retraining), so effects should be replicated across sessions and compared only against same-session baselines. The checkpoint-selection degeneracy of section 5 belongs to the same family: at this scale, protocol failures do not announce themselves, and the discipline (paired deltas, fresh baselines, archived logs, per-epoch validation series) is what makes small effects believable at all.

Selection bias and external validity. The gold set overrepresents exploited and enriched CVEs — the KEV subset by construction, the 2021 subset by curation choice — and the one positive metadata finding lives precisely on the enriched

Table 9: Five-fold label-holdout zero-shot evaluation (recall@5 on held-out gold, 222 open candidates). Each fold's bi-encoder is trained with the fold's techniques withheld entirely; MiniLM scores the identical restricted gold.

Model (n test rows)	fold 0 (53)	fold 1 (51)	fold 2 (42)	fold 3 (30)	fold 4 (35)	mean $\pm$ std
bi-encoder (holdout)	0.057	0.010	0.321	0.000	0.229	0.123 ± 0.144
MiniLM (zero-shot)	0.211	0.078	0.369	0.333	0.195	0.237 ± 0.117

Table 10: Bucket-aware arms vs. the flattened-union baseline (gold test split, n=118, 119 for seed 45; five paired seeds; union task). Baseline row absolute, remaining rows paired per-seed deltas. * marks $|\Delta| > 2 \cdot \mathrm{SEM}$.

Arm	recall@5	recall@3	MRR	micro-F1	macro-F1
union baseline	0.668 ± 0.018	0.539 ± 0.008	0.642 ± 0.020	0.413 ± 0.008	0.182 ± 0.017
bucket multi-task	-0.076^*	-0.083^*	-0.044^*	-0.065^*	-0.072^*
tactic auxiliary	-0.018	-0.065^*	-0.061^*	-0.036^*	-0.030^*

stratum. We read this operationally: the CWE gain applies to the actively-exploited population that triage prioritizes, and nothing in our data says it transfers to the long tail of unexploited CVEs with NVD-grade metadata (the measured 2021-stratum null says it does not). The KEV stratum is also small (n=36), so we treat gain *locations* as established and their magnitudes as indicative.

The cascade could reopen. The gold-vs-predicted gap on KEV (+0.082 vs. +0.038 recall@5) prices the cost of CWE prediction error at 27% ancestor agreement. A substantially better CWE classifier would shrink that gap, and the cascade — costing nothing at inference — would be worth remeasuring; the negative verdict is a statement about current prediction quality, not about the architecture.

Task ceiling and intended use. Technique mapping is judgment: the CTID methodology leaves room for analyst disagreement, most visibly in the secondary-impact role — consistently the hardest bucket in table 11 — and the union target inherits that subjectivity. Absolute numbers should be read against this unmeasured ceiling, and the models remain what the prior paper released them as: generators of *candidate* techniques for analyst review, not authoritative mappings.

Short-input handling. A short description should not be padded, summarized or discarded in production: none of those operations creates the exploit-mechanism evidence that is missing, and the enrichment measured in this paper does not either — the CVSS vector is null and affected products are mildly harmful (section 6.1); only a curated CWE helped, and only on the KEV stratum. The defensible intervention is an explicit *insufficient-context* flag routing the row to analyst review with the curated CWE attached where one exists — but the length audit of section 6.1 found no penalty for descriptions under 256 characters on this test set, so character count is not that flag's criterion; whatever is, it has to be selected on validation data by measured difficulty and reported on test once.

Limitations. All experiments use one base encoder (`roberta-base`; the prior paper's ModernBERT check did not change its verdict) and English descriptions. Techniques are parent-level throughout. The test split is small (121 rows, of which 118 — 119 for seed 45 — carry an in-vocabulary technique and are evaluated), which is why strata as small as the CWE-absent group (n=17) are reported but not interpreted. The tactic-auxiliary weight was fixed a priori rather than swept, so E5 rules out one point, not the curve; and the tactic vocabulary is tied to the current ATT&CK release. Finally, the below-floor zero-shot verdict rests on five hold-out folds with large fold-to-fold spread — its direction (fine-tuning does not transfer, and tends to hurt) is consistent, but per-fold magnitudes vary widely. The deployment discussion of section 8 is an architectural analysis with a cost model, not a latency measurement: encoder latency, scan latency, throughput under concurrent requests and the behaviour of a cold memory-mapped store on specified hardware remain to be benchmarked. The description-length audit of section 6.1 has the power of 44 short against 74 long test rows: it rules out a large penalty for descriptions under 256 characters, not a small one, and its under-128-character band holds nine rows.

8 Deployment: Retrieval in Vulnerability-Lookup

The released classifier is served to Vulnerability-Lookup [12] through ML-Gateway [9], an HTTP service that keeps every machine-learning dependency out of the platform: the pages send a description and render the answer, and this is how the per-CVE technique suggestion is produced. The bi-encoder is served the same way on the public instance (`https://vulnerability.circl.lu`), for the two directions the head cannot provide — technique→CVE and CVE→CVE — following the receptionist picture of section 4.3: once every

Table 11: Per-role ranking quality of the bucket multi-task model (mean $\pm$ std over five seeds), each row scored only on test examples with gold in that role. Not comparable to union metrics (smaller gold sets); descriptive.

Role	n	recall@5	recall@3	MRR
exploitation technique	55	0.781 ± 0.017	0.705 ± 0.047	0.638 ± 0.023
primary impact	70	0.682 ± 0.053	0.558 ± 0.026	0.508 ± 0.033
secondary impact	32	0.527 ± 0.015	0.379 ± 0.038	0.381 ± 0.022
uncategorized	40	0.696 ± 0.023	0.501 ± 0.051	0.598 ± 0.029

description and every technique text is a point in one space, the corpus can be searched from either side with the same weights. Figure 1 shows the division of labor.

The gateway owns the encoder, the vectors and the search. At start-up it loads the released checkpoint once and encodes the shipped technique texts, keeping one vector per technique in memory; these are recomputed at every start rather than stored, so they always match the served weights. Its index endpoint embeds a batch of {identifier, description} pairs exactly as in section 4.3 (512-token truncation, mean-pooling, L_2-normalization) and upserts the vectors; the description itself is not kept. From then on the encoder runs only for a new record or a free-text query: a search over the whole corpus is one product between a query vector and the stored matrix, with no model call. Two retrieval endpoints answer the pages. The technique search ranks indexed vulnerabilities for one technique by the training-time score $\sigma(a\cos+b)$, so a vulnerability's score for a technique is exactly what the measured CVE→technique direction gives it; techniques outside the trained vocabulary are embedded from their ATT&CK text and flagged as such, since section 6.3 shows they rank noticeably worse. The related search ranks by plain cosine around an indexed identifier or a free description. The store itself is one row-major $N \times 768$ float16 matrix — one row per indexed vulnerability, the row's 768 values being the unit vector of section 4.3, 1.5 KB each regardless of the description's length — with the identifiers kept in row order beside it, so the vector payload is $1{,}536N$ bytes (about 0.77 GB at $N = 500{,}000$, before identifiers and process overhead). An exact query is one matrix–vector product, $768N$ multiply–accumulates over a sequential read of that payload: linear in N and, once the matrix is resident, bound by memory bandwidth rather than arithmetic. We do not turn that model into a latency figure without a reproducible hardware benchmark. The result is exact — every stored vector is scored, none is skipped by an approximate index. A memory map avoids a private copy per worker but not the page faults of a cold first scan, so cold and warm results must be reported separately. Vectors are only comparable within one model revision; the store records the revision it was built with and every response carries it, so a model refresh is a rebuild rather than a silent drift. The index endpoint is the one write into that state and is treated accordingly: it requires a bearer token shared with the platform, refuses every call while no token is configured, and refuses growth past a configurable ceiling on the number of indexed identifiers; the gateway itself is reachable only from the platform, since the read endpoints remain an unauthenticated compute cost.

Two latencies and a scaling boundary. A query by an indexed identifier or by technique ID costs only the scan, since both vectors are precomputed. A free-text query must first run the 512-token encoder, and that call, not the scan, dominates. The encoder is also what the index endpoint consumes, so indexing load reaches free-text queries: on the public instance on 17 September 2026, while the store was being reseeded, free-text fallback searches exceeded the proxy's ten-second limit whereas searches by stored identifier answered at once. Indexing itself is off the user-facing path and batched, so its measures are vectors per second and backlog age, not request latency.

The served kernel already applies the optimizations that preserve the ranking exactly: technique vectors are cached at start-up; the memory-mapped matrix is scanned in blocks of 65,536 rows, each cast to float32 before the product, so float16 is a storage format and not an accumulation precision; the top k are found by partial selection rather than a full sort; and because $a > 0$ and the sigmoid is monotone, $\sigma(a\cos+b)$ is applied only to the returned items. Two refinements remain future work: evaluating queued queries as one matrix–matrix product, and a release check of top-k agreement against a float32 reference, especially around ties.

The exact scan is the right choice while warm and cold p50/p95/p99 latency, throughput under concurrent queries and RAM use meet the deployment target at the actual corpus size; it is not an unconditional claim that a vector database is unnecessary. Beyond that boundary the candidates are hardware-tuned blocked scans, a tested approximate-nearest-neighbour index, or 8-bit quantization of the payload, each to be measured for recall@k and rank agreement against the float32 exact baseline. A benchmark that supports any of these decisions reports corpus size, hardware, thread count, storage and accumulation dtypes, cache state, encoder and scan latency separately, throughput under concurrency, and retrieval recall. Until it exists, the design supports functional deployment and a capacity model, not a performance conclusion.

The platform's one contribution to the live path is an *embedding indexer*: a single process that subscribes to the publish/subscribe channel on which each of the platform's feeders announces every record it stores. Its role is to keep the gateway's index current between seeds without touching the

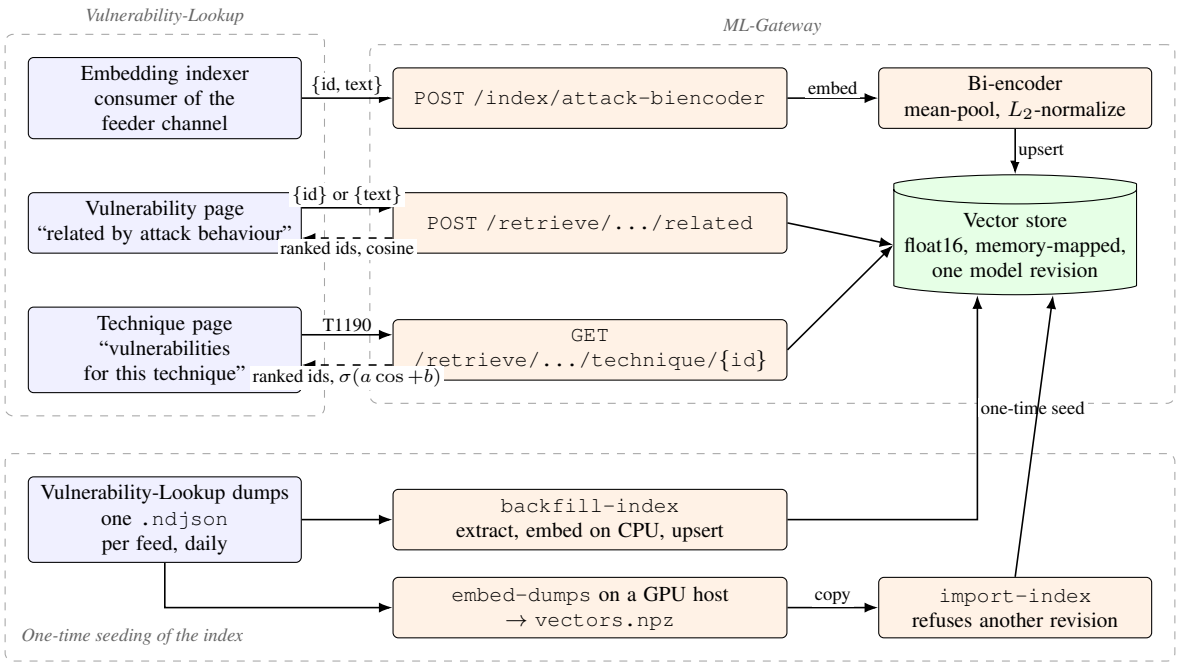


Figure 1: Serving the bi-encoder to Vulnerability-Lookup. The platform holds no model and no vector: an indexer process subscribed to the channel on which every feeder publishes sends each new or changed description to the gateway, and the two pages query the gateway by technique or by vulnerability. The gateway owns the encoder, a memory-mapped float16 vector store shared by its worker processes and pinned to one model revision, and the seeding commands that fill the store once from the public dumps.

feeders. It reduces each published record to its public identifier and English description with the same function the gateway's backfill uses, buffers, and sends batches to the index endpoint: 16 items at start (or whatever has accumulated after five seconds), with a read timeout of eight seconds per item and at least a minute; a call that times out halves the batch, down to a single item, and four consecutive full batches that succeed grow it back by four, up to 64. The adaptation exists because the gateway embeds a batch synchronously on a CPU it shares with free-text queries and with any reseed in progress, and a fixed batch that cannot finish inside a fixed timeout is re-sent unchanged forever — the failure observed on 17 September 2026, before the change. The indexer is built to fail softly rather than to guarantee delivery: a gateway that is down, answers a server error or refuses the token is retried with exponential backoff capped at five minutes, the backlog is bounded at 20,000 descriptions beyond which the oldest are dropped with a warning, and publish/subscribe itself is fire-and-forget, so records published while the indexer restarts are missed. Both gaps are closed by re-running the seed with its skip-existing option, which is why the seed, not the indexer, is the source of truth for the index. The indexer holds no model, no vector and no machine-learning dependency, and is off by default behind a configuration switch, so an instance without a gateway index runs exactly as before. It presents the shared token on every index call; a rejected token is logged and treated as an outage rather than retried.

The store is seeded once from the NDJSON dumps that Vulnerability-Lookup publishes daily (one file per feed; the CVE corpus alone is several gigabytes) by one of two paths: `backfill-index` extracts, embeds on the gateway host's CPU and upserts — a few hours for several hundred thousand descriptions — or `embed-dumps` runs the same extraction on a GPU host and writes one archive of identifiers, float16 vectors and the model revision — no descriptions and no technique vectors, so the archive, like the store, is a derived artifact that the dumps can regenerate at any time — which `import-index` loads, refusing an archive built with another model revision. The extraction rule (public identifier plus the English description, per feed layout) is the same function on both sides, so the seed and the live path agree on what a vulnerability's text is. The commands and their options are documented in the ML-Gateway repository [9].

What this deployment does and does not claim follows section 6.3. Only the CVE→technique direction has measured accuracy, and for it the classification head remains the default of the per-CVE suggestion. The two retrieval directions reuse the same trained similarity but have no reported numbers; the interface presents them as similarity aids, labelled with model and revision, not as classifications. Measuring them is a natural follow-up: the gold set already provides technique→CVE relevance by inversion, and a CVE→CVE judgment set would need to be built.

As an illustration rather than a measurement, fig. 2 shows the two notions of relatedness side by side on the public instance for CVE-2026-89207, a Siemens web-interface vulnerability that lets an unauthenticated remote attacker force the device into protection mode and cut its remote connec-

tivity. The platform's own "related vulnerabilities" tab lists three records linked by explicit cross-reference, led by the GitHub mirror of the same advisory. The bi-encoder tab lists the ten nearest descriptions (table 12): the mirror first at cosine 0.99, a near-identical text and therefore a sanity check rather than a finding, followed by nine vulnerabilities from six other vendors that share no product and no cross-reference with the source, every one of which describes an unauthenticated network attacker degrading or disabling an embedded or network device. The footer of the panel also shows the live path of this section at work: the record had been published the day before, after the index was seeded, so the search ran on its description through the fallback rather than on a stored row — with the same function and therefore the same result.

9 Conclusion

We asked whether a label-quality-bound CVE-to-ATT&CK classifier can be bought out of its bind with more information per example and per label, rather than more examples. Mostly, it cannot. The derivation-chain prior fails at its last proposed role; the CVSS vector is null; products and derived candidates are mildly harmful; concatenating everything is worse than nothing; the CTID role structure and the tactic hierarchy both cost union accuracy when used as training signal. Two interventions survive the audit, and each is precise about what it buys: verbalizing a *curated* CWE helps exactly the enriched, actively-exploited stratum that operational triage cares about (and its predicted substitute buys nothing), and a label-semantics bi-encoder converts a small head-class ranking cost into the largest consistent rare-technique gain of the series — while its apparent zero-shot promise does not survive fine-tuning, leaving no measured route to the below-floor tail except curation.

The deployment configuration that follows: a description-only, flattened-union classifier, with the gold CWE appended when a curated assignment exists; the bi-encoder as the variant of choice when rare-technique coverage matters more than top-5 sharpness, and as the engine of the technique→CVE and CVE→CVE retrieval of section 8. The intervention with measured payoff remains growing the curated gold set — every scaling curve in both papers still rises at the current size. The negative results are released with the same care as the positive ones — full trainer logs, seeds, and scripts — because at this data scale, knowing which paths are closed is most of what a follow-up needs.

Reproducibility and Artifacts

- Code: VulnTrain (github.com/vulnerability-lookup/VulnTrain). The experiments start from release v3.2.0, and all experiment code (metadata serialization, prior fusion, bi-encoder, label-holdout, bucket/tactic arms, stratified and open-vocabulary evaluation, per-row prediction export) is contained in release v3.3.0, tagged for this paper. Every trained arm is an entry-point invocation with an explicit seed; the sweep scripts (`run_*.sh`) and the aggregation script are in the paper repository.
- Dataset: `CIRCL/vulnerability-attack-techniques` v2.1 (revision `319c3e3`), DOI 10.57967/hf/9621 [11] — a strictly additive extension of v1 (identical rows and splits; metadata columns and `cwes_predicted` added).
- Models: the released classifier of the prior paper (`...-classification-roberta-base`, DOI 10.57967/hf/9623 [8]) remains the deployment configuration; the bi-encoder of section 6.3 is released as `...-attack-technique-biencoder` (DOI 10.57967/hf/9968 [10]); the cascade uses the public CIRCL CWE classifier [7]. The remaining per-arm experiment checkpoints (ablation grids, non-released bi-encoder seeds, holdout folds) are not individually published — they are reproducible from the released code, dataset revision, seeds, and logs.
- Serving: the retrieval endpoints, vector store and seeding commands of section 8 are in ML-Gateway [9]; the embedding indexer, proxy endpoints and pages in Vulnerability-Lookup [12]. A self-contained reference implementation of the scoring function (encoder, pooling, normalization, affine, and the two truncation lengths) that depends only on `transformers`, and reproduces the validator's numbers, is in the VulnTrain documentation (attack-biencoder-retrieval).
- Trainer logs behind every table — including the invalid first bi-encoder sweep (section 5) — are under `trainer-logs/` in the paper repository, one directory per experiment.

Listing 1 is the embedding function of eq. (1) as served by ML-Gateway, stripped of batching and device handling. The same function, applied with `max_length` 512 to a description and `technique_max_length` from the model configuration to a technique text, followed by $\sigma(a\,u^\top v + b)$ with a and b read from the same configuration, reproduce every bi-encoder score in this paper.

```python
def embed(texts, tokenizer, model, max_length):
    """Unit vectors, shape (len(texts), 768)."""
    batch = tokenizer(
        texts, padding=True, truncation=True,
        max_length=max_length, return_tensors="pt")
    with torch.no_grad():
        out = model(**batch)
    hidden = out.last_hidden_state      # (n, T, 768)
    mask = batch["attention_mask"]
    mask = mask.unsqueeze(-1).to(hidden.dtype)
    pooled = (hidden * mask).sum(1)
    pooled = pooled / mask.sum(1).clamp(min=1e-9)
    return torch.nn.functional.normalize(pooled, dim=1)
```

Listing 1: Reference embedding function (mask-weighted mean pooling, L_2 normalization). `model` is the released checkpoint loaded with `AutoModel`; `tokenizer` its tokenizer.

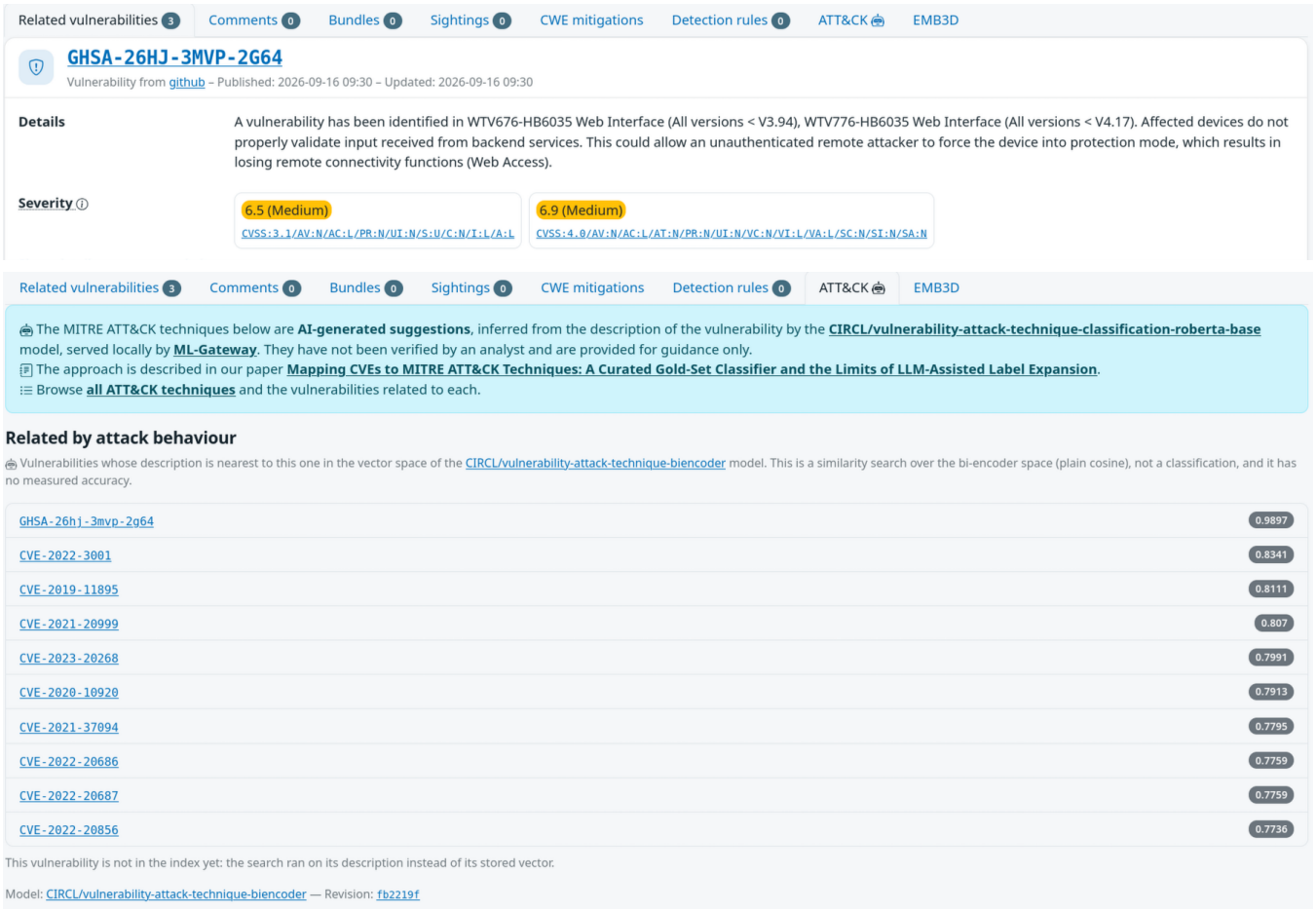


Figure 2: CVE-2026-89207 on `https://vulnerability.circl.lu`. Top: the "Related vulnerabilities" tab, populated by explicit cross-references (three records, the GitHub mirror of the advisory shown). Bottom: the ATT&CK tab, with the classification head's suggestion notice above and the bi-encoder's "Related by attack behaviour" list below, each result being an indexed vulnerability and its cosine to the source. The footer reports that the source record was not yet indexed and that the search used its description instead of a stored vector; model and revision are shown on every answer.

Acknowledgements

This work was carried out in the context of the AIPITCH project (AI-Powered Innovative Toolkit for Cybersecurity Hubs), which aims to create advanced artificial-intelligence-based tools supporting key operational services in cyber defence, including technologies for early threat detection, automatic malware classification, and the improvement of analytical processes through the integration of Large Language Models. The project is led by the NASK National Research Institute (Poland); the international consortium includes CIRCL (Computer Incident Response Center Luxembourg, Luxembourg), The Shadowserver Foundation (Netherlands), NCBJ (National Centre for Nuclear Research, Poland), and ABI Lab (Centre of Research and Innovation for Banks, Italy).

Funding

This work was co-funded by CIRCL and by the European Union through the AIPITCH project. The European Union contribution was provided by the European Cybersecurity Industrial, Technology and Research Competence Centre under grant agreement No 101190545 (call DIGITAL-ECCC-2024-DEPLOY-CYBER-06). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Cybersecurity Industrial, Technology and Research Competence Centre. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] Basel Abdeen, Ehab Al-Shaer, Anoop Singhal, Latifur Khan, and Kevin W. Hamlen. SMET: Semantic mapping of CVE to ATT&CK and its application to cybersecurity. In *Data and Applications Security and Privacy*

Table 12: The ten nearest neighbours of CVE-2026-89207 in fig. 2, with the vendor and the behaviour each description reports. Descriptions retrieved from the platform; the cosine is the value displayed.

Neighbour	Vendor	Reported behaviour	Cosine
GHSA-26hj-3mvp-2g64	(mirror)	same advisory, same text	0.990
CVE-2022-3001	Milesight	improper input handling in a camera VMS web interface, remote attacker	0.834
CVE-2019-11895	Bosch	denial of service of a smart-home controller via its JSON-RPC interface	0.811
CVE-2021-20999	Weidmüller	internal-only port reachable externally on IoT gateways; device disrupted	0.807
CVE-2023-20268	Cisco	unauthenticated adjacent attacker exhausts an access point's resources	0.799
CVE-2020-10920	C-MORE	unauthenticated remote attack on an HMI touch-panel firmware	0.791
CVE-2021-37094	Huawei	improper input validation leading to system denial of service	0.780
CVE-2022-20686	Cisco	unauthenticated remote attacker crashes the LLDP service of a telephone adapter	0.776
CVE-2022-20687	Cisco	same advisory family as the previous row	0.776
CVE-2022-20856	Cisco	unauthenticated remote denial of service of a wireless controller (CAPWAP)	0.774

XXXVII (DBSec 2023), Lecture Notes in Computer Science, pages 243–260. Springer, 2023.

[2] Cédric Bonhomme and Alexandre Dulaunoy. VLAI: A RoBERTa-based model for automated vulnerability severity classification. *arXiv preprint arXiv:2507.03607*, 2025.

[3] Cédric Bonhomme and Alexandre Dulaunoy. Mapping CVEs to MITRE ATT&CK techniques: A curated gold-set classifier and the limits of LLM-assisted label expansion. *arXiv preprint arXiv:2607.25572*, 2026.

[4] Center for Threat-Informed Defense. Mapping ATT&CK to CVE for impact. https://github.com/center-for-threat-informed-defense/attack_to_cve, 2021. MITRE Engenuity Center for Threat-Informed Defense.

[5] Center for Threat-Informed Defense. TRAM: Threat report ATT&CK mapper. https://github.com/center-for-threat-informed-defense/tram, 2023.

[6] Center for Threat-Informed Defense. ATT&CK mappings explorer. https://center-for-threat-informed-defense.github.io/mappings-explorer/, 2024. Includes CVE (KEV) to ATT&CK mappings.

[7] Computer Incident Response Center Luxembourg. cwe-parent-vulnerability-classification-roberta-base. Hugging Face, https://huggingface.co/CIRCL/cwe-parent-vulnerability-classification-roberta-base, 2026. Fine-tuned RoBERTa parent-level CWE classifier, trained with VulnTrain.

[8] Computer Incident Response Center Luxembourg. vulnerability-attack-technique-classification-roberta-base (Revision 3b98a1d). Hugging Face, https://huggingface.co/CIRCL/vulnerability-attack-technique-classification-roberta-base, 2026. DOI: 10.57967/hf/9623.

[9] Computer Incident Response Center Luxembourg and Cédric Bonhomme. ML-Gateway: HTTP gateway serving the CIRCL vulnerability models. GitHub, https://github.com/vulnerability-lookup/ML-Gateway, 2026.

[10] Computer Incident Response Center Luxembourg and Cédric Bonhomme. vulnerability-attack-technique-biencoder (Revision fb2219f). Hugging Face, https://huggingface.co/CIRCL/vulnerability-attack-technique-biencoder, 2026. DOI: 10.57967/hf/9968.

[11] Computer Incident Response Center Luxembourg and Cédric Bonhomme. vulnerability-attack-techniques (Revision c2b5da0). Hugging Face, https://huggingface.co/datasets/CIRCL/vulnerability-attack-techniques, 2026. DOI: 10.57967/hf/9621.

[12] Computer Incident Response Center Luxembourg and Cédric Bonhomme. Vulnerability-Lookup: a platform for correlating vulnerabilities and coordinating disclosure. GitHub, https://github.com/vulnerability-lookup/vulnerability-lookup, 2026.

[13] Cybersecurity and Infrastructure Security Agency (CISA). Known exploited vulnerabilities catalog. https://www.cisa.gov/known-exploited-vulnerabilities-catalog, 2026.

[14] FIRST.Org, Inc. Common vulnerability scoring system version 3.1: Specification document, 2019. Accessed 2026-08-06.

[15] FIRST.Org, Inc. Common vulnerability scoring system version 4.0: Specification document, 2023. Accessed 2026-08-06.

[16] Galeax. CVE2CAPEC: Mapping CVE to CWE, CAPEC and MITRE ATT&CK. https://github.com/Galeax/CVE2CAPEC, 2024.

[17] Octavian Grigorescu, Andreea Nica, Mihai Dascalu, and Razvan Rughiniş. CVE2ATT&CK: BERT-based mapping of CVEs to MITRE ATT&CK techniques. *Algorithms*, 15(9):314, 2022.

[18] Erik Hemberg, Jonathan Kelly, Michal Shlapentokh-Rothman, Bryn Reinstadler, Katherine Xu, Nick Rutar, and Una-May O'Reilly. Linking threat tactics, techniques, and patterns with defensive weaknesses, vulnerabilities and affected platform configurations for cyber hunting. *arXiv preprint arXiv:2010.00533*, 2020.

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

[20] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

[21] Blake E. Strom, Andy Applebaum, Doug P. Miller, Kathryn C. Nickels, Adam G. Pennington, and Cody B. Thomas. MITRE ATT&CK: Design and philosophy. Technical Report MP180360R1, The MITRE Corporation, 2020. https://attack.mitre.org.

[22] The MITRE Corporation. CWE: Common weakness enumeration. https://cwe.mitre.org, 2026.