Beyond the Login: Continuous Behavioral Authorship Verification for LLM Agent Sessions

Muskan Sharma
Dept. of Information Technology
Delhi Technological University
muskankalonia@gmail.com

Seba Susan
Dept. of Information Technology
Delhi Technological University
sebasusan@dtu.ac.in

Abstract

LLM agents now operate as a privileged application layer, with access to mail, files and MCP-connected tools, yet they have no built-in way to verify that the person typing into an already-signed-in session is still the legitimate user. This identity gap aligns with OWASP LLM06 (Excessive Agency), Agentic ASI03 (Identity and Privilege Abuse), and NIST guidance on AI risk management, but whether it is closable from user text alone is largely open. We build a profile-query authorship verification benchmark on WildChat-1M and PAN-20 regrouped per author, with author-disjoint 60/20/20 splits across three seeds. We train contrastive bi-encoders with triplet margin loss on DistilBERT, BERT, MiniLM-L6 and ELECTRA, under 100% random imposters or a 50/30/20 mix of random, semi-hard and hard imposters mined by TF-IDF. Models trained on random imposters look strong (ROC-AUC above 0.95) but drop by around 10 points on WildChat against topic-similar imposters, suggesting they rely on topic and vocabulary rather than writing style only. Our contribution is to expose and quantify this failure mode and to provide a reproducible hardness-stratified benchmark; applying established hardness-aware training as a controlled mitigation closes most of the gap, lifting TAR@FAR=1% on WildChat Hard from around 6–7% to 20–22% for BERT and DistilBERT with median messages-to-detect of one, and the same direction holds under a secondary SBERT miner. We release the benchmark construction and evaluation code[1].

1 Introduction

LLM agents are quietly becoming one of the most privileged pieces of software on a user's machine. With protocols like Model Context Protocol (Anthropic, 2024) an agent can be connected to tools that send the user's mail, push commits, edit files in the user's workspace and run a shell command, all on the basis of a few words typed into a chat box. While these capabilities substantially improve user productivity, they also introduce a persistent authentication problem: once an agent session has been authenticated, subsequent actions are typically authorized solely on the basis of continued session access. Consequently, any individual with access to an already authenticated interface—whether through device sharing, session exposure, or workstation compromise—may issue commands with the same privileges as the legitimate user. Existing agent architectures generally lack mechanisms for continuously verifying that the active operator remains consistent with the user who originally authenticated the session. Behavioral authorship verification refers to the task of determining whether two text segments originate from the same author based on patterns in their writing behavior across interaction histories, rather than relying solely on static stylistic features of isolated documents.

Many surrounding layers already perform some form of identity or authorization check: the operating system authenticates the user at login or unlock, Multi-Factor Authentication (MFA) may be required for sensitive services, and browser or OAuth tokens authorize access to external applications. The LLM agent layer, however, usually has no equivalent check once a session is already open. The security community has identified related risks: the OWASP Top 10 for LLM Applications (OWASP Foundation, 2025) lists LLM06: Excessive Agency, the OWASP Top 10 for Agentic Applications (OWASP Foundation, 2026) identifies ASI03: Identity and Privilege Abuse, MITRE ATT&CK lists Impersonation (MITRE, 2023) as an established adversary technique, and NIST's AI Risk Management Framework (National Institute of Standards and Technology, 2023) emphasizes governance and accountability for AI systems.

[1]Our benchmark-construction scripts and evaluation code are available at https://github.com/Muskankalonia/beyond-the-login-continuous-authorship-verification-LLM.

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 1039–1054
November 6-10, 2026 ©2026 Association for Computational Linguistics

These frameworks motivate the risk, but they do not answer whether the problem is technically solvable at the agent layer using only the signal the agent reliably has: the text the user has typed. This paper investigates that question and, additionally, examines the amount of interaction context required for a verifier to make reliable authorship verification decisions.

We model continuous authorship verification as a representation-learning problem over interaction text. Given a reference *profile* consisting of previously observed user messages and an incoming *query* message, a text encoder maps both inputs into a shared embedding space. Profile-level and query-level representations are then compared using cosine similarity to estimate whether the incoming message is consistent with the enrolled user identity. We instantiate this idea with a contrastive bi-encoder trained using triplet margin loss. The model performs very well when the imposter corresponds to a uniformly random alternative user: on the random test regime, ROC-AUC rises above 0.9 even for MiniLM-L6, a 22.71M-parameter encoder. At first glance, the problem appears close to solved.

But a uniformly random imposter is not the hardest adversary an LLM agent may face. A real imposter may share a workplace, family, or social circle with the user; they may know the same projects, codebases, or clients; and the topics they discuss may overlap substantially with those of the legitimate user. When we replace the random imposter with another real user mined for topical similarity using TF-IDF over character and word n-grams, performance drops sharply: on WildChat, ROC-AUC falls by about 10 points on the hardest 5% of imposters for several encoders. The model that looked excellent on random imposters becomes much less reliable against topic-similar imposters, suggesting that it relies partly on topic and vocabulary rather than writing style alone. This topic-vs-style gap is the empirical core of the paper.

Because the failure mode is security-relevant, we evaluate at security operating points rather than relying only on ROC-AUC. We report **TAR@FAR=1%** (fraction of genuine users accepted at a fixed 1% imposter false-accept rate), **FAR@FRR=1%** (the symmetric metric), EER, the robustness gap between random and hard imposters, and streaming metrics over the first three query messages (catch@3, median messages-to-detect). The full metric list and security interpretation is in Section 6. Our key finding is that replacing purely random training negatives with a 50/30/20 mix of random, semi-hard and hard imposters substantially reduces the random-to-hard robustness gap, especially on WildChat: TAR@FAR=1% on WildChat Hard rises from 6-7% under random-only training to 20-22% under hardness-aware training for BERT and DistilBERT, while the median messages-to-detect remains one. We build this benchmark from WildChat-1M (Zhao et al., 2024) as the primary dataset and PAN-20 (Bevendorff et al., 2020) regrouped at the author level as a secondary, with author-disjoint 60/20/20 splits across three seeds and a 70/30 profile-query split within each author. We release the benchmark construction scripts and evaluation code.

Contributions. We do not propose a new negative-sampling algorithm; hard-negative mining is well established in metric learning and dense retrieval (Schroff et al., 2015; Xiong et al., 2021; Robinson et al., 2021). Instead, we (i) identify a security-relevant evaluation gap: random-imposter tests overestimate robustness by letting continuous authorship verifiers exploit topic and vocabulary shortcuts; (ii) release a reproducible, hardness-stratified profile-query benchmark with author-disjoint splits and security-oriented metrics (TAR@FAR, FAR@FRR, EER, catch@3, median messages-to-detect); and (iii) show that hardness-aware training, used as a controlled mitigation, narrows but does not close the gap across four encoders, two datasets, strict operating points, and two independent hardness miners.

2 Threat Model

Figure 1 situates the verifier within an LLM agent session: an enrolled profile and an incoming message are encoded by a shared bi-encoder, their cosine score feeds a tiered action gate, and the gate's behaviour depends on the consequence class of the agent action being requested.

The adversary. We consider an adversary who has physical or session-level access to an LLM agent already enrolled for a legitimate user. The legitimate user has signed in at some point in the past, the agent may have access to the user's email, files, calendar, and MCP-connected tools, and the adversary is now using that same open session. The adversary's goal is to make the agent perform an action the legitimate user did not intend, such as sending a sensitive email, exfiltrating files, posting

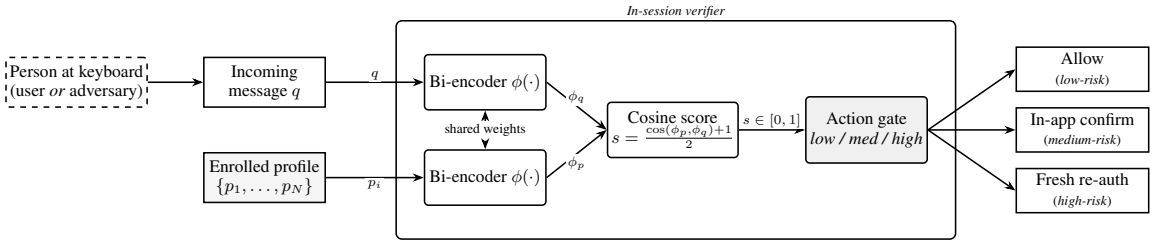


Figure 1: System overview. A user interacts with an already-authenticated LLM agent session; the verifier has no prior knowledge of whether the active user is the legitimate account holder or an adversary. The in-session verifier (rounded box) encodes the enrolled profile $\{p_1, \ldots, p_N\}$ and the incoming message q *independently* with the same shared-weight bi-encoder ϕ (there is no joint cross-encoder), scores their pooled cosine in $[0, 1]$, and feeds the score to a tiered action gate (Section 8): *low-risk* actions are allowed silently, *medium-risk* actions require in-app confirmation when the score is below threshold, and *high-risk* actions require a fresh OS-level re-authentication.

a message in the user's voice, or running a destructive shell command. The adversary types into the agent and can approve tool calls when the agent requests authorization. This adversary maps to OWASP LLM06 (Excessive Agency) and OWASP Agentic Applications ASI03 (Identity and Privilege Abuse), as discussed in Section 1.

What the adversary knows. We give the adversary a realistic but nontrivial amount of knowledge. They may know the user's general topic distribution because they share a workplace, family, or social circle; they may know the projects, codebases, or clients the user works with; and they may attempt to mimic the user's writing style or register if they have seen the user write before. The adversary has access to neither the verifier's internal weights nor the enrolled profile stored on the device, which matches the on-device deployment setting we recommend later.

What the defender sees. The defender is the LLM agent. It holds an enrolled profile of the user's past messages and sees the new incoming text in the current session. The verifier produces a score and a decision using a fixed threshold selected on validation data, and does not get OS-level signals such as keystroke dynamics, mouse movement, or window focus; it only sees the text the user has typed. We treat the verifier as one layer in a defense-in-depth stack, not a replacement for OS-level authentication; the concrete policy mapping is in Section 8.

Out of scope. We do not study adversaries with direct access to the verifier weights, adversaries who have stolen the enrolled profile from the device (the on-device deployment assumption is already broken in that case), or adversaries who use another LLM to rewrite their messages in the user's style using examples of the user's writing.

3 Related Work

Authorship verification. Authorship verification has a long history in stylometry and the PAN evaluation campaigns have driven much of the modern work on it (Stamatatos, 2009; Bevendorff et al., 2020). The classical recipe combines character and word n-grams with shallow classifiers, and character n-grams in particular are known to mix style and topic signals in ways that can conflate style with topic/content signal (Sapkota et al., 2015), which is one of the things we explicitly stress-test against in this paper. More recent work has moved to neural encoders: Boenninghoff et al. (2019) train hierarchical recurrent Siamese LSTM-based encoders for authorship verification on social media, Barlas and Stamatatos (2020) compare pre-trained language models against cross-domain authorship attribution, and Tyo et al. (2022) survey the broader state of the art across authorship and verification. Our contribution relative to this literature is twofold: we restructure PAN-20 from its native pairwise format into per-author profiles so that the task matches the deployment setting where an agent has an enrolled user history rather than two unrelated texts, and we stress-test the encoders against topic-similar imposters mined by an identity-agnostic similarity function instead of reporting only on the random or natural test distribution.

Contrastive learning and hard-negative mining. Siamese and bi-encoder objectives are standard in sentence representation learning, including triplet-network SBERT-style training (Reimers and Gurevych, 2019) and contrastive sentence-embedding methods such as SimCSE (Gao et al., 2021); they also build on the broader metric-learning tradition exemplified by triplet loss in FaceNet (Schroff et al., 2015). Hard-negative min-

ing is well studied in dense retrieval (Xiong et al., 2021) and in general contrastive and metric learning (Robinson et al., 2021; Schroff et al., 2015), where easy random negatives can quickly become uninformative and semi-hard or hard negatives are often used to learn more discriminative representations. We apply the same idea to authorship verification specifically and evaluate the effect not just on threshold-free metrics like ROC-AUC but on the security operating points (TAR@FAR=1%, FAR@FRR=1%, catch@3 and median messages-to-detect) that a real agent deployment would care about, which is where most of the gains from hardness-aware training actually land.

Concurrent work. In concurrent work, Wang et al. (2026) study dialogue attribution in unauthenticated human-LLM sessions, introducing WildAuth, a benchmark derived from real-world ChatGPT logs, and an uncertainty-aware multi-aspect attribution model. Our work and theirs are complementary but differ in three places: we frame the problem as an agent-layer security primitive with operating-point metrics (TAR@FAR=1%, FAR@FRR=1%, catch@3) rather than attribution accuracy alone; we add a hardness-stratified evaluation against topic-similar imposters so that the random-versus-hard robustness gap is observable; and we study a hardness-aware training regime alongside the random-trained baseline, which lets us measure whether the topic-vs-style shortcut is fixable through training rather than just diagnosed.

4 Dataset

We use two complementary datasets. WildChat-1M (Zhao et al., 2024) is our primary dataset because it consists of real user-ChatGPT interactions and most closely approximates the agent-chat setting. PAN-20 (Bevendorff et al., 2020) is our secondary dataset because it is a widely used authorship-verification benchmark; using both allows us to test whether our findings hold across short conversational text and long fanfiction text.

WildChat-1M. We use the AllenAI release, filter to English-language users using the provided language field (mixing languages would let a verifier to exploit language ID rather than writing style), and use `hashed_ip` as a proxy user identifier, following the dataset metadata. This is imperfect (shared-IP households merge identities, roaming users split them), which we discuss in the limitations. The model only ever sees raw user text. After filtering to English users with at least 10 messages, we obtain **16,467 eligible users**.

Table 1: Dataset statistics after all filters. Train/val/test counts are identical across seeds 13, 21, 42 because the stratified shuffle preserves bucket sizes.

	PAN-20	**WildChat-1M**
Eligible authors/users	8,483	16,467
Mean texts per author	16.9	47.7
profile (70%)	11.4	33.0
query (30%)	5.5	14.7
Mean chars/text	∼21k	∼1.2k
Train/Val/Test	5090/1697/1696	9880/3294/3293

PAN-20. The official PAN-20 authorship-verification training release is in pairwise format, which does not match our deployment setting; we regroup it by author identifier across all pairs and treat each author as a single identity. After applying the same ≥ 10-text filter we obtain **8,483 eligible authors**. PAN authors write fanfiction across multiple fandoms; we do not use fandom as a model input or a split key. PAN texts are much longer than WildChat messages (mean ∼21k characters), so we char-truncate at 2,000 before tokenization (defensive, not tuned).

Preprocessing and counts. We collapse repeated whitespace during model preprocessing but do not lowercase, strip punctuation, or remove special characters because punctuation habits and capitalization are exactly the stylometric signals the encoder should learn. After all filters, PAN gives a mean of 16.9 texts/author (11.4 profile, 5.5 query); WildChat gives 47.7 messages/user (33.0/14.7). The 60/20/20 author-disjoint split yields 5,090/1,697/1,696 on PAN and 9,880/3,294/3,293 on WildChat, identical across seeds (Table 1).

5 Methodology

Profile-query construction. Each identity's text pool is split 70%/30% into profile (the enrolled history the agent has) and query (the new incoming text). For PAN we use a stable per-(seed, author) hash to shuffle and take the first 70% as profile; for WildChat we sort messages chronologically by timestamp and take the first 70% as profile and the last 30% as query, matching the realistic deployment. We evaluate at two prefix lengths, k=1 (first held-out query message) and k=3 (first three averaged). Each author contributes one positive (profile vs. own query) and one negative (profile vs. another

author's query) pair at every prefix length, so the evaluation set is balanced.

Author-disjoint stratified splits. We split authors 60/20/20 into train, validation and test with no overlap. The split is stratified by text count: PAN authors into 10–12, 13–16, 17–20, 21+ buckets; WildChat users into 10–12, 13–16, 17–20, 21–49, 50–99, 100–499, 500+. We deliberately do not split on topic or fandom: holding fandom shared but authors disjoint puts the verifier in the realistic situation (same topic, different writer), which is the only situation in which the random and hard test regimes have room to differ. The three seeds are 13, 21, 42; every reported number is the mean over them with sample standard deviation.

Hardness mining. A hard imposter is a different identity whose text looks similar to the anchor's under an identity-agnostic similarity function. For each identity A in a split we rank every other identity B by $\cos(\phi(\text{profile}_A), \phi(\text{query}_B))$. The top 5% forms the hard bucket, ranks 5-30% the semi-hard bucket, and the random bucket is sampled uniformly from other identities without similarity weighting. For each triplet we sample a negative identity from the requested bucket and then a uniformly random text from that identity's pool, rather than the single most similar text, which would concentrate gradients on a few identities. The *primary miner* is a TF-IDF cosine model combining character 3-5-grams and word 1-2-grams ($\min_df{=}1$, word features capped at 200k, char at 300k), trained separately within each split on that split's profile and query texts. The *secondary miner* is the Sentence-BERT (SBERT) model `sentence-transformers/all-MiniLM-L6-v2` (Reimers and Gurevych, 2019), used as a fixed encoder with mean pooling and L2 normalization. Hardness indexes are cached per (dataset, seed, split).

Primary miner, step by step. Concretely, within each split the TF-IDF miner (1) fits a character 3-5-gram and word 1-2-gram TF-IDF vectorizer on that split's profile and query texts ($\min_df{=}1$, word vocabulary capped at 200k features and character vocabulary capped at 300k); (2) L2-normalizes the resulting vectors; (3) for each anchor identity A, computes the cosine similarity between A's profile vector and the query vector of every other identity B's query vector; and (4) ranks the other identities by this similarity, assigning the top 5% to the hard bucket and those ranked 5-30% to the semi-hard bucket. The vectorizer is fit separately per split, so no text from another split informs a split's hardness ranking, and the resulting indexes are cached per (dataset, seed, split). The exact configuration and implementation are included in the released code.

Contrastive bi-encoder and triplet loss. A single transformer backbone with shared weights encodes the anchor, positive and negative texts independently. We load each encoder from `AutoModel` rather than `AutoModelForSequenceClassification`: the latter adds a task-specific classification head, whereas we need the encoder representation directly. For every text we mean-pool the last hidden state weighted by the attention mask and L2-normalize the pooled vector; we add no projection head, no custom special tokens and no prefix prompts, and use the stock tokenizer with max_length $= 256$. We train with triplet margin loss in Euclidean space with margin 0.3. Because embeddings are L2-normalized, squared Euclidean distance is monotonic in cosine similarity ($\|a - b\|^2 = 2 - 2\cos(a, b)$), so the loss encourages anchor-positive cosine to exceed anchor-negative cosine, aligning training with the cosine-based evaluation score. We do not use in-batch negatives, which keeps the fraction of hard negatives a property of the sampler rather than the batch. Triplets are resampled every epoch with a per-epoch seed derived from the run seed.

Two training regimes. We compare two ways of choosing the training negatives: (i) **random-only**, with 100% random negatives; and (ii) **mixed**, with 50% random, 30% semi-hard and 20% hard negatives. The mixed proportions match the mixed test composition by design. The two regimes differ only in negative composition; everything else is held fixed, which lets us attribute any difference to the negative composition.

Test protocol and scoring. Each profile text is encoded independently, embeddings are averaged and re-L2-normalized to give one profile embedding per author, and the same is done for the query at each prefix length. The verification score is $(\cos{+}1)/2$, which is monotonic in cosine and preserves rank-based metrics. We score with fixed cosine rather than a learned pair scorer because cosine is symmetric, parameter-free, and generalizes to authors unseen during training, which matches our author-disjoint protocol; we discuss learned symmetric scorers as a future ablation in Section 9. The decision threshold is selected once on mixed-regime validation predictions by maximising F1

and applied across all four test regimes: **random** (100% random imposters), **semi-hard** (100% from the 5-30% bucket), **hard** (100% from the top 5% bucket), and **mixed** (50/30/20, the realistic deployment composition). The positives are identical across regimes, so any difference is attributable to imposter hardness. To check that our findings are not an artifact of the TF-IDF definition of "hard", we also run a secondary SBERT-miner ablation on DistilBERT and BERT in three configurations (random-only with SBERT test imposters; mixed TF-IDF train with SBERT test imposters; mixed SBERT on both sides).

6 Experimental Setup

Models. We evaluate four pre-trained transformer encoders in the TF-IDF hardness regime: DistilBERT-base-uncased (Sanh et al., 2019) (66M parameters), BERT-base-uncased (Devlin et al., 2019) (110M), MiniLM-L6 (`sentence-transformers/all-MiniLM-L6-v2`, 22.71M; an SBERT-pretrained MiniLM checkpoint (Reimers and Gurevych, 2019)), and ELECTRA-base-discriminator (Clark et al., 2020) (110M). The four encoders span the compact-to-base size range that fits within our 8 GB VRAM constraint. The SBERT cross-miner ablation is restricted to DistilBERT and BERT.

Hyperparameters. Triplets per identity per epoch are fixed at 16, giving 81,440 and 158,080 triplets per training epoch on PAN ($5{,}090 \times 16$) and WildChat ($9{,}880 \times 16$). Texts are char-truncated to 2,000 chars and tokenized at max_length $= 256$ with truncation. Train batch size is 16 for DistilBERT and MiniLM, and 8 with gradient accumulation 2 for BERT and ELECTRA (effective batch 16, fits within 8 GB VRAM); eval batch size is 32 for DistilBERT/MiniLM and 16 for BERT/ELECTRA. Optimizer is AdamW with lr $= 2 \times 10^{-5}$, weight decay 0.01, no warmup, gradient clipping at L2-norm 1.0, and FP16 mixed precision. We train for 3 epochs and select the best epoch by validation ROC-AUC, using the *mixed* validation regime for both training regimes so that both pick their best checkpoint against the same difficulty distribution.

Metrics. We frame the task as a security problem, so alongside threshold-free summaries we report operating-point metrics. We treat genuine (same-author) pairs as the positive class throughout, so True Accept Rate (TAR) equals the true-positive rate (TPR) and False Accept Rate (FAR) equals the false-positive rate (FPR); the reverse convention, with imposters as positive, maps these to TNR and FNR. ROC-AUC (area under the receiver-operating-characteristic curve) summarizes ranking quality across all thresholds and is invariant to class balance; PR-AUC (area under the precision-recall curve) summarizes the precision-recall trade-off and, unlike ROC-AUC, depends on class prevalence. Because our evaluation set is balanced by construction (one genuine and one imposter pair per author), the two are numerically close here and we report both as complementary rank summaries; a deployment with its own genuine/imposter prevalence should recompute PR-AUC at that operating prevalence. **F1** is computed at the validation-selected threshold, held fixed across test regimes and prefix lengths. **EER** (Equal Error Rate) is the operating point where false-accept and false-reject rates meet. **TAR@FAR=1%** answers "if we are only willing to let 1 imposter in 100 slip through, what fraction of legitimate users do we accept?"; **FAR@FRR=1%** answers the symmetric question (what fraction of imposters we accept when only 1 in 100 genuine users is rejected). For streaming detection over the first query messages we report **catch@3** (fraction of imposters whose averaged 3-message score falls below threshold), the genuine survival rate at 3 (fraction of genuine users still accepted after three messages), and **median messages-to-detect**: among imposters that are eventually caught, the median first prefix $k \in \{1, 3\}$ at which the score falls below the fixed threshold. Because we evaluate only two prefix lengths, messages-to-detect is a coarse two-point measure (its value is 1 or 3); we therefore report it alongside catch@3 rather than on its own, and also report its P90.

Hardware and reproducibility. All contrastive experiments were run on a single workstation with an RTX 4060 Ti (8 GB VRAM), Intel i9-13900K, and 31 GB RAM; we use FP16 mixed precision to fit BERT and ELECTRA. Each (dataset, model, regime, seed) contrastive cell takes 2-3 hours including evaluation on all four test regimes. We do not sweep hyperparameters: the per-model-class settings above are held fixed across all cells. We release the source code, preprocessing and split-generation scripts, configs, evaluation code, and result JSONs; raw text files are not redistributed because both datasets contain user-sensitive content, but splits can be reconstructed deterministically from the released seeds.

7 Results

All aggregate numbers in the main tables are means over seeds 13, 21 and 42; sample standard deviations are reported in the appendix. Table 2 gives the headline Hard-regime results, Table 3 gives the random-to-hard robustness gap per encoder, and Table 4 gives the SBERT cross-miner ablation on DistilBERT and BERT. The full per-test-regime breakdown for the contrastive and SBERT-ablation runs is in Appendix A.

Table 3: Random-to-Hard ROC-AUC **robustness gap** per encoder (%, lower is better). Δ = Mix$-$RO; negative means the gap shrank under hardness-aware training. WildChat shows the largest reductions, consistent with WildChat having the largest random-only gap.

Dataset	Encoder	RO	Mix	Δ
PAN-20	DistilBERT	3.42	2.58	-0.84
	BERT	2.91	2.43	-0.48
	MiniLM	3.61	2.81	-0.80
	ELECTRA	2.92	2.15	-0.76
WildChat	DistilBERT	9.54	6.78	-2.76
	BERT	9.85	6.51	-3.33
	MiniLM	7.55	5.85	-1.70
	ELECTRA	10.63	8.23	-2.41

Table 4: SBERT cross-miner ablation on DistilBERT and BERT, Hard regime only (%). The topic-vs-style gap and the hardness-aware fix both replicate when the test imposters are mined by SBERT instead of TF-IDF, so the finding is not an artifact of the specific definition of "hard".

		Hard regime	
Setting	**Model**	ROC-AUC	TAR@FAR=1%
TF-IDF eval (Mix orig)			
PAN	DistilBERT	93.9	46.1
PAN	BERT	94.5	50.9
WildChat	DistilBERT	89.6	20.3
WildChat	BERT	89.8	22.3
SBERT eval, RO training			
PAN	DistilBERT	94.0	46.0
PAN	BERT	94.5	49.8
WildChat	DistilBERT	83.5	7.4
WildChat	BERT	83.1	5.2
SBERT eval, TF-IDF Mix training (cross-miner)			
PAN	DistilBERT	93.6	45.4
PAN	BERT	94.3	47.9
WildChat	DistilBERT	85.8	18.6
WildChat	BERT	85.9	18.6
SBERT eval, SBERT Mix training (same-miner)			
PAN	DistilBERT	94.4	48.5
PAN	BERT	94.8	49.7
WildChat	DistilBERT	86.6	20.3
WildChat	BERT	86.4	18.3

Reference baselines. The top block of Table 2 reports three training-free families. On PAN, TF-IDF cosine is surprisingly strong on Hard (ROC-AUC 95.4, TAR@FAR=1% 82.3), outperforming every fine-tuned encoder; this matches prior observations that n-gram methods remain competitive on long, content-rich authorship tasks (Tyo et al., 2022). On WildChat (texts roughly 18$\times$ shorter), TF-IDF degrades sharply at the strict operating point (TAR@FAR=1% 1.4), and the strongest training-free verifier is the frozen `all-MiniLM-L6-v2` checkpoint (ROC-AUC 82.9, TAR@FAR=1% 5.7), which already benefits from sentence-similarity pretraining; SBERT-mpnet zero-shot is slightly below it. Frozen DistilBERT, BERT, and ELECTRA are weak on WildChat (TAR@FAR=1% 3.2-3.5), so off-the-shelf transformer embeddings without sentence-level pretraining are poor zero-shot verifiers here. BERT and DistilBERT Mix (TAR@FAR=1% 22.3 and 20.3) reach 3.5–3.9$\times$ the best training-free WildChat number.

The encoders look strong on random imposters but struggle on topic-similar ones. Across all four encoders the random-imposter regime looks strong by ROC-AUC (0.95-0.97 on both datasets), though the stricter TAR@FAR=1% metric is much lower on WildChat random (45-63%). The hard regime is a different story. Under random-only training, ROC-AUC drops from 0.96-0.97 random to 0.85-0.89 hard on WildChat (9.85 points for BERT, 9.54 for DistilBERT, 10.63 for ELECTRA, 7.55 for MiniLM), and TAR@FAR=1% on WildChat Hard collapses to 6.2% (BERT), 7.3% (DistilBERT), 4.3% (ELECTRA) and 21.7% (MiniLM). On PAN the drops are smaller (2.9-3.6 ROC-AUC points), consistent with PAN texts being long and content-rich enough that the encoder has more style signal to fall back on, but the gap is present on every encoder.

Hardness-aware training substantially reduces the WildChat gap. The 50/30/20 mix shrinks the WildChat random-to-hard ROC-AUC gap by 22-34% across encoders (Table 3); the PR-AUC gap roughly halves on BERT (13.84 $\rightarrow$ 7.56) and DistilBERT (12.88 $\rightarrow$ 7.91). The biggest absolute gain lands on TAR@FAR=1% on WildChat Hard, the metric a security team would deploy on: 6.2% $\rightarrow$ 22.3% for BERT (+16.1 points) and 7.3% $\rightarrow$ 20.3% for DistilBERT (+12.9 points), with the corresponding WildChat Mixed-test TAR@FAR=1% climbing 18.9% $\rightarrow$ 42.2% (BERT) and 22.3% $\rightarrow$ 41.9% (DistilBERT). PAN gains are smaller (+1.5

Table 2: Main results on the **Hard** test regime (top 5% similarity imposters), means over seeds 13/21/42 (%). ROC-AUC, TAR@FAR=1%, and EER are computed over pooled k=1 and k=3 predictions; catch@3 uses the k=3 prefix. Reference baselines (top block) are evaluated without contrastive fine-tuning; their single score per metric is shown under both RO and Mix columns via `\multicolumn`. "SBERT-mpnet" uses `sentence-transformers/all-mpnet-base-v2`; "Frozen MiniLM" uses `sentence-transformers/all-MiniLM-L6-v2`, a sentence-similarity-pretrained checkpoint. RO = random-only training, Mix = 50/30/20 random/semi-hard/hard training. For each RO/Mix pair we **bold** the better value (higher for ROC-AUC, TAR@FAR=1% and catch@3; lower for EER). Hardness-aware training delivers the largest gains on WildChat, particularly on TAR@FAR=1%. See Appendix A for the full per-regime breakdown with sample standard deviations.

		ROC-AUC		**TAR@FAR=1%**		**catch@3**		**EER**	
Dataset	**Encoder**	RO	Mix	RO	Mix	RO	Mix	RO	Mix
Reference baselines (no contrastive fine-tuning)									
PAN-20	TF-IDF cosine	95.4		82.3		81.4		11.0	
PAN-20	SBERT-mpnet	68.8		16.4		4.5		37.6	
PAN-20	Frozen DistilBERT	73.5		27.1		5.9		36.2	
PAN-20	Frozen BERT	73.3		29.7		3.6		37.2	
PAN-20	Frozen MiniLM	69.9		19.7		4.4		37.4	
PAN-20	Frozen ELECTRA	74.1		21.8		17.3		33.4	
WildChat	TF-IDF cosine	74.5		1.4		54.4		28.2	
WildChat	SBERT-mpnet	81.9		4.5		65.6		21.6	
WildChat	Frozen DistilBERT	73.9		3.4		41.9		30.1	
WildChat	Frozen BERT	72.8		3.5		39.8		31.0	
WildChat	Frozen MiniLM	82.9		5.7		67.7		20.7	
WildChat	Frozen ELECTRA	62.4		3.2		18.5		40.4	
Contrastive fine-tuning									
PAN-20	DistilBERT	93.4	**93.9**	43.9	**46.1**	74.9	**77.4**	14.4	**13.9**
	BERT	94.2	**94.5**	48.8	**50.9**	76.7	**79.1**	13.3	**13.2**
	MiniLM	92.0	**92.2**	39.1	**40.7**	73.4	**73.7**	16.0	16.0
	ELECTRA	94.2	**94.7**	47.1	**50.6**	75.7	**78.6**	13.2	**12.9**
WildChat	DistilBERT	86.7	**89.6**	7.3	**20.3**	65.2	**69.1**	19.9	**18.3**
	BERT	86.4	**89.8**	6.2	**22.3**	65.0	**69.6**	20.1	**18.3**
	MiniLM	89.0	**90.6**	21.7	**23.7**	67.5	**69.8**	18.8	**17.4**
	ELECTRA	85.1	**87.6**	4.3	**9.9**	62.4	**65.7**	21.3	**19.6**

to +3.5 TAR@FAR=1% on Hard), expected given the small PAN random-only gap. Hardness-aware training matters most where the gap is large.

Streaming detection is fast when it fires. On WildChat Hard, catch@3 reaches 69.6% for BERT and 69.1% for DistilBERT under mixed training, up from 65.0% and 65.2% under random-only training. The median messages-to-detect is one across all four encoders, both datasets, and all test regimes in the main contrastive runs, so in the cases where the verifier flags the imposter it usually does so on the first message. On the realistic WildChat mixed-test distribution, catch@3 sits at 85.1% for BERT and 85.0% for DistilBERT under mixed training.

The SBERT cross-miner ablation reproduces the direction. Swapping the test imposters from TF-IDF to SBERT-mined drops the WildChat Hard ROC-AUC by about 4 points for the TF-IDF-mixed model (BERT 89.76 → 85.95, DistilBERT 89.60 → 85.75). Training on SBERT-mined negatives gives a modest additional ROC-AUC gain (BERT 86.43, DistilBERT 86.60), while the larger effect is visible at the strict operating point. Under SBERT evaluation, WildChat Hard TAR@FAR=1% rises from 5.2% to 18.3% for BERT and from 7.4% to 20.3% for DistilBERT, so the topic-vs-style gap and the hardness-aware fix are not artifacts of the TF-IDF definition of "hard". On PAN, the SBERT cross-miner numbers stay close to the TF-IDF numbers, consistent with PAN being a less hardness-sensitive dataset.

8 Deployment Recommendations

The results in this paper indicate that the verifier is not strong enough to be a deployment's only line of defense, but it is useful as one layer when combined with the controls OWASP recommends for LLM06 (OWASP Foundation, 2025). We translate the operating-point numbers into concrete agent-side policies that a deployment could implement without changing the underlying model.

Action categories. Following OWASP LLM06's emphasis on minimizing agency and requiring human approval for high-impact actions, we organize agent actions by consequence into three deployment tiers. *Low-risk* actions are non-sensitive reads

and short responses that do not change persistent state (answering a factual question, fetching a public URL), where a wrong verdict is recoverable in both directions. *Medium-risk* actions change persistent state in visible and recoverable ways (drafting an email but not sending it, creating a calendar event, editing a file in the user's workspace, committing code that has not been pushed). *High-risk* actions carry incident-level risk (sending email, transferring funds, running shell commands, modifying access tokens, posting publicly in the user's voice, or calling an MCP tool that changes an external system).

Suggested policy. For low-risk actions, the verifier is run silently and its score is logged for telemetry; no blocking is performed. For medium-risk actions, the verifier is used as a risk signal, with lightweight step-up confirmation required when the score falls below a threshold. For high-risk actions, the verifier is not sufficient on its own; fresh OS-level authentication (biometric, password, or hardware token), re-enrollment of the agent session, or a delay sufficient for session expiry and screen-lock mechanisms to trigger is required. The verifier's role at this tier is to flag the session and raise the friction the legitimate user has to clear, not to make the trust call by itself. Because the verifier scores every incoming message, a rolling three-message monitor can escalate the next medium-risk action to high-risk treatment if the score drops below threshold mid-session. At a strict 1% false-accept budget on WildChat Hard, the verifier still rejects roughly 80% of legitimate attempts, too friction-heavy to gate medium-risk actions on its own; adaptive thresholds, longer profiles, and behavioral biometrics are the natural levers.

Profile updating and drift. Our benchmark assumes a fixed enrolled profile and therefore does not specify a deployment refresh schedule. In practice, profiles may need periodic updates as writing style changes, but indiscriminately adding accepted messages is unsafe: a single false accept can gradually poison the enrolled representation. A safer policy would refresh cached profile embeddings only with messages that pass a high-confidence check or follow fresh OS-level re-authentication, while retaining a stable historical component and bounding the recent-message window; the globally trained bi-encoder weights remain fixed. We leave refresh frequency and poisoning-resistant update policies to future work. Online cost remains low: with profile embeddings cached at enrollment, each new message requires only one forward pass and one cosine comparison.

9 Conclusion

We study whether a text encoder can serve as an agent-layer identity check for LLM-driven desktop agents, using a profile–query authorship verification benchmark from WildChat-1M and a regrouped PAN-20. Fine-tuned contrastive bi-encoders look strong against random imposters (ROC-AUC above 0.95) but rely heavily on topic and vocabulary, so WildChat ROC-AUC drops 8–11 points and TAR@FAR=1% collapses into the single digits once the imposter shares the user's topic. A 50/30/20 mix of random, semi-hard and hard imposters substantially reduces this gap, with the largest gains on TAR@FAR=1% (WildChat Hard 6–7% $\rightarrow$ 20–22% for BERT and DistilBERT) and the direction holding under a secondary SBERT miner. The absolute numbers are still not strong enough for the verifier to be a deployment's only line of defense, so we recommend the tiered policy in Section 8.

We consider several follow-up points. First, a stronger adversary model where the imposter uses an LLM to actively rewrite messages in the user's style; our current imposters are real topic-similar users, not adversarial mimics. Second, longer enrollment profiles plus behavioral signals (keystroke timing, action patterns) to push TAR@FAR=1% high enough for medium-risk gating. Third, scaling to larger encoders and multilingual settings beyond English. Fourth, per-user calibration or lightweight adaptation on top of the globally trained encoder, which may capture user-specific writing cues that a shared model misses. Fifth, replacing the fixed cosine score with a symmetric learned scorer over embedding pairs, for example over the features $[\,|a-b|;\, a \odot b\,]$, which could model richer interactions provided it preserves symmetry and does not learn corpus-specific pair shortcuts. We release the benchmark construction scripts and evaluation code so these can be reproduced from the same setup.

Limitations

In order to identify users in WildChat we use the `hashed_ip` field, but this is imperfect because it suffers from the shared IP problem and a single user moving between networks can contribute multiple identities. We accept these caveats since the dataset

is large enough that we expect noise to dilute when aggregated at scale, but a real deployment environment would need a more reliable identity signal than IP addresses.

Our results do not say anything about multilingual authorship verification, the case where an imposter uses a different language than the user, or the case where the same author uses different languages while communicating with the LLM. In WildChat we filter the dataset to English-language users only so that the verifier does not win the task just by detecting language ID instead of writing style.

The hard imposters in our datasets are real authors or normal users and not humans with malicious intent. A real adversary trying to impersonate a user will probably do something similar by matching topics or vocabulary, but we do not test the stronger case where the adversary actively tries to replicate the user's style.

All experiments were run on a single 8 GB GPU, which limits us to models under approximately 200M parameters at our sequence length. We evaluate on four encoders (DistilBERT, BERT-base, MiniLM-L6 and ELECTRA-base) and the SBERT cross-miner ablation is run on two of these. We cannot verify whether the robustness gap closes at scale given the hardware constraint.

WildChat was collected in the early days when LLMs were just being released and most conversations are about coding help or short drafting tasks. It does not contain the chats where users ask an LLM to perform operations via MCP servers with access to delete files, read emails or access personal accounts, so a real deployment threat model is sharper than what we directly study. Related to this, we do not study the risk scores of imposter messages to determine how dangerous an accepted imposter would be; that is planned for follow-up work.

These constraints mean our findings should be read within the evaluated setting (English-only text, non-adaptive topic-similar imposters, encoders below 200M parameters) and not assumed to transfer directly to multilingual users, style-mimicking adversaries, or substantially larger encoders.

Ethics Statement

Authorship verification is a double-edged sword: the techniques we study to refuse imposter sessions inside LLM agents can also be used to link anonymous online writing back to real people with known writings, which would be disastrous for users like whistleblowers, journalists and activists who depend on the right of anonymous speech. We did not invent the underlying techniques, but we want to be explicit that any deployment of this verifier outside the LLM-agent setting should be scrutinized carefully.

Beyond this, we recommend that deployments run on a user's on-premises device and the enrolled profiles themselves should not be sent to a central server or stored online. Moving user profiles into a remote backend database, even for legitimate verification purposes, creates aggregated datasets of how individuals write, which can become a target for both insider misuse and external attacks. Keeping the verifier on the user's device with the profile is a comparatively safer default because the writing style stays with the user.

PAN-20 is distributed under a research license and WildChat-1M is distributed under the license terms listed with the AllenAI release; we follow the terms of the exact dataset artifact used. Both datasets may contain sensitive content related to users in their texts, so we do not redistribute the texts and do not release any model checkpoint fine-tuned on individual user data. During training or evaluation the model never sees metadata fields such as `hashed_ip`, country or `conversation_id`; it only sees the user texts.

When the verifier is deployed inside an LLM agent, the cost of being wrong by accepting an imposter and by rejecting a legitimate user is not the same: a false accept can let the imposter take a consequential action, while a false reject merely requires the legitimate user to re-authenticate. We therefore operate in a strict setting where the verifier keeps out imposters even at the cost of rejecting some legitimate users, and we treat the verifier as one layer on top of OS-level login and per-action approval, not a replacement for them.

References

Anthropic. 2024. Model context protocol. https://modelcontextprotocol.io. Open protocol specification.

Georgios Barlas and Efstathios Stamatatos. 2020. Cross-domain authorship attribution using pre-trained language models. In *Artificial Intelligence Applications and Innovations (AIAI)*, pages 255–266.

Janek Bevendorff, Bilal Ghanem, Anastasia Giachanou,

Mike Kestemont, Enrique Manjavacas, Ilia Markov, Maximilian Mayerl, Martin Potthast, Francisco Rangel, Paolo Rosso, Günther Specht, Efstathios Stamatatos, Benno Stein, Matti Wiegmann, and Eva Zangerle. 2020. Overview of PAN 2020: Authorship verification, celebrity profiling, profiling fake news spreaders on Twitter, and style change detection. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Eleventh International Conference of the CLEF Association (CLEF 2020)*.

Benedikt Boenninghoff, Steffen Hessler, Dorothea Kolossa, and Robert M. Nickel. 2019. Similarity learning for authorship verification in social media. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2457–2461.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations (ICLR)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 4171–4186.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6894–6910.

MITRE. 2023. Social engineering: Impersonation, sub-technique T1684.001. https://attack.mitre.org/techniques/T1684/001/. MITRE ATT&CK Enterprise Matrix; previously T1656.

National Institute of Standards and Technology. 2023. Artificial intelligence risk management framework (AI RMF 1.0). Technical Report NIST AI 100-1, NIST.

OWASP Foundation. 2025. OWASP top 10 for LLM applications 2025. https://genai.owasp.org/llm-top-10/. LLM06: Excessive Agency.

OWASP Foundation. 2026. OWASP top 10 for agentic applications. https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026/. ASI03: Identity and Privilege Abuse.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3982–3992.

Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2021. Contrastive learning with hard negative samples. In *International Conference on Learning Representations (ICLR)*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Upendra Sapkota, Steven Bethard, Manuel Montes-y Gómez, and Thamar Solorio. 2015. Not all character N-grams are created equal: A study in authorship attribution. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 93–102.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823.

Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556.

Jacob Tyo, Bhuwan Dhingra, and Zachary C. Lipton. 2022. On the state of the art in authorship attribution and authorship verification. *arXiv preprint arXiv:2209.06869*.

Wenxuan Wang, Zirui Liu, Haoxuan Kou, Xuefeng Liu, and Jiaxing Shen. 2026. The chatbot knows it's you: Dialogue attribution in unauthenticated human-LLM sessions. In *Proceedings of the ACM Web Conference (WWW)*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations (ICLR)*.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. WildChat: 1M ChatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations (ICLR)*.

A Full Per-Encoder Results (TF-IDF training)

This appendix reports the main metrics broken down by dataset, encoder, test regime and training regime. All numbers are mean ± sample standard deviation over seeds 13, 21 and 42, in percent. RO = random-only training; Mix = 50/30/20 hardness-aware training. Tables 5–8 cover PAN-20, and Tables 9–12 cover WildChat-1M. The compact random→hard robustness-gap summary is reported in Table 3 in the main paper.

How to read these tables. Each table has four *test* blocks: *Random* (uniformly sampled imposters), *Semi-hard* (imposters from the 5–30% TF-IDF similarity bucket), *Hard* (top 5% TF-IDF similarity), and *Mixed* (the 50/30/20 deployment composition). Positives are identical across the four regimes, so any change in a row is attributable to imposter difficulty rather than to the genuine-user distribution. The RO and Mix columns hold the test set and the encoder fixed and vary only the training negative composition; the Δ column is Mix $-$ RO. A negative Δ on EER, FAR@FRR1% or on the robustness gap is an improvement; a positive Δ on ROC-AUC, PR-AUC, F1, TAR@FAR1% or catch@3 is an improvement.

What to look for. Four patterns recur across encoders. (i) Many random-regime metrics move little between RO and Mix, sometimes slightly downward, which is expected: hardness-aware training trades some easy-imposter headroom for hard-imposter robustness. (ii) The largest absolute gains land on TAR@FAR1% in the Hard and Mixed regimes on WildChat, especially for BERT and DistilBERT, because this metric reads off the tail of the score distribution where the topic-vs-style shortcut hurts the most. (iii) The TAR@FAR1% gain grows monotonically with imposter hardness rather than being confined to a single slice, confirming that the effect is a broad robustness improvement rather than a shift on the hardest test alone: on WildChat BERT the Mix$-$RO gain rises from $+8.39$ points at Random to $+12.43$ at Semi-hard to $+16.15$ at Hard. (iv) Standard deviations are largest on TAR@FAR1% and FAR@FRR1% for a tail-reading reason: a fixed 1% FAR budget on a few thousand test pairs is sensitive to a handful of score swaps near threshold, whereas ROC-AUC averages over the full ranking and is correspondingly stable.

Per-encoder pattern. On PAN-20 the four encoders rank similarly: BERT and ELECTRA lead on ROC-AUC and TAR@FAR1%, DistilBERT trails by a small margin, and MiniLM is consistently last but within roughly 2 ROC-AUC points. PAN's small random-only gap (Table 3) leaves less room for hardness-aware training, so the Mix gains are modest. On WildChat the ranking shifts: MiniLM has the smallest random-only gap (7.55 ROC-AUC points) and also shows the smallest absolute Mix improvement, while BERT and DistilBERT absorb the largest TAR@FAR1% gains ($+16.15$ and $+12.94$ points respectively on Hard). ELECTRA is the weakest WildChat encoder in both regimes and benefits less from hardness mining than BERT or DistilBERT ($+5.60$ TAR@FAR1% on Hard), suggesting that its representations are less well suited to short conversational text under this setup.

Table 5: PAN-20, DistilBERT-base-uncased (66M).

Test	Metric	RO	Mix	Δ
Random	ROC-AUC	96.78±0.43	96.53±0.56	−0.25
Random	PR-AUC	96.89±0.41	96.60±0.50	−0.28
Random	F1	90.39±0.80	89.80±1.10	−0.58
Random	EER	9.60±0.83	10.16±1.09	+0.56
Random	FAR@FRR1%	40.85±3.82	39.87±5.93	−0.98
Random	TAR@FAR1%	62.78±2.80	59.85±2.14	−2.94
Random	catch@3	88.88±2.26	87.70±1.20	−1.18
Semi-hard	ROC-AUC	95.47±0.24	95.54±0.41	+0.07
Semi-hard	PR-AUC	95.67±0.11	95.65±0.34	−0.02
Semi-hard	F1	88.39±0.28	88.49±0.61	+0.10
Semi-hard	EER	11.60±0.40	11.49±0.71	−0.11
Semi-hard	FAR@FRR1%	52.16±4.21	48.44±5.42	−3.72
Semi-hard	TAR@FAR1%	56.08±1.90	55.62±2.45	−0.46
Semi-hard	catch@3	83.37±1.47	84.55±2.71	+1.18
Hard	ROC-AUC	93.36±0.24	93.95±0.22	+0.59
Hard	PR-AUC	93.60±0.19	94.14±0.12	+0.54
Hard	F1	85.39±0.37	86.33±0.33	+0.94
Hard	EER	14.43±0.24	13.87±0.30	−0.56
Hard	FAR@FRR1%	62.72±3.06	56.99±4.35	−5.73
Hard	TAR@FAR1%	43.87±1.10	46.08±3.27	+2.21
Hard	catch@3	74.94±1.68	77.44±1.47	+2.50
Mixed	ROC-AUC	95.67±0.44	95.65±0.69	−0.02
Mixed	PR-AUC	95.85±0.43	95.77±0.68	−0.08
Mixed	F1	88.64±0.47	88.75±0.75	+0.12
Mixed	EER	11.47±0.54	11.40±0.93	−0.07
Mixed	FAR@FRR1%	49.38±4.46	46.08±6.21	−3.30
Mixed	TAR@FAR1%	55.67±3.11	55.40±5.19	−0.27
Mixed	catch@3	84.00±1.81	84.67±1.78	+0.67

Table 6: PAN-20, BERT-base-uncased (110M).

Test	Metric	RO	Mix	Δ
Random	ROC-AUC	97.10±0.39	96.97±0.24	−0.13
Random	PR-AUC	97.22±0.34	97.03±0.25	−0.19
Random	F1	91.02±0.74	90.56±0.63	−0.46
Random	EER	9.00±0.74	9.34±0.42	+0.33
Random	FAR@FRR1%	38.77±4.49	34.94±2.80	−3.82
Random	TAR@FAR1%	63.74±2.35	62.15±4.60	−1.59
Random	catch@3	89.39±2.55	89.64±1.05	+0.26
Semi-hard	ROC-AUC	95.99±0.36	96.17±0.57	+0.18
Semi-hard	PR-AUC	96.18±0.25	96.33±0.54	+0.14
Semi-hard	F1	89.09±0.65	89.28±1.03	+0.19
Semi-hard	EER	10.90±0.58	10.71±1.01	−0.19
Semi-hard	FAR@FRR1%	48.44±4.93	42.75±3.88	−5.69
Semi-hard	TAR@FAR1%	59.22±1.97	61.17±3.56	+1.96
Semi-hard	catch@3	84.14±1.99	86.50±0.48	+2.36
Hard	ROC-AUC	94.18±0.39	94.54±0.43	+0.35
Hard	PR-AUC	94.38±0.49	94.69±0.50	+0.30
Hard	F1	86.47±0.47	86.96±0.53	+0.49
Hard	EER	13.35±0.57	13.17±0.45	−0.18
Hard	FAR@FRR1%	58.79±2.87	52.34±2.63	−6.45
Hard	TAR@FAR1%	48.78±3.44	50.87±4.93	+2.09
Hard	catch@3	76.67±1.66	79.07±1.84	+2.40
Mixed	ROC-AUC	96.10±0.46	96.00±0.64	−0.10
Mixed	PR-AUC	96.26±0.44	96.06±0.64	−0.20
Mixed	F1	89.25±0.83	89.16±1.17	−0.09
Mixed	EER	10.82±0.78	10.94±1.00	+0.12
Mixed	FAR@FRR1%	46.06±4.79	41.32±3.29	−4.74
Mixed	TAR@FAR1%	59.06±2.88	56.49±3.68	−2.57
Mixed	catch@3	84.65±1.92	85.61±0.81	+0.96

Table 7: PAN-20, MiniLM-L6 (22.71M).

Test	Metric	RO	Mix	Δ
Random	ROC-AUC	95.64±0.60	95.03±0.44	−0.60
Random	PR-AUC	95.77±0.58	95.16±0.45	−0.60
Random	F1	88.75±1.00	87.74±0.50	−1.02
Random	EER	11.22±1.01	12.34±0.67	+1.12
Random	FAR@FRR1%	46.31±5.22	50.59±2.50	+4.27
Random	TAR@FAR1%	53.98±3.60	51.40±1.53	−2.58
Random	catch@3	86.79±1.56	84.00±1.10	−2.79
Semi-hard	ROC-AUC	94.17±0.58	94.09±0.26	−0.08
Semi-hard	PR-AUC	94.41±0.37	94.32±0.17	−0.09
Semi-hard	F1	86.77±0.81	86.57±0.50	−0.19
Semi-hard	EER	13.26±0.86	13.71±0.53	+0.45
Semi-hard	FAR@FRR1%	55.90±6.78	56.72±1.85	+0.83
Semi-hard	TAR@FAR1%	48.48±1.81	48.87±0.61	+0.39
Semi-hard	catch@3	81.53±0.48	81.03±1.07	−0.49
Hard	ROC-AUC	92.03±0.60	92.22±0.26	+0.20
Hard	PR-AUC	92.30±0.65	92.52±0.36	+0.23
Hard	F1	84.18±0.61	84.31±0.24	+0.13
Hard	EER	15.97±0.71	16.03±0.19	+0.06
Hard	FAR@FRR1%	65.70±4.41	65.35±0.21	−0.35
Hard	TAR@FAR1%	39.14±2.01	40.66±3.45	+1.52
Hard	catch@3	73.45±0.70	73.74±0.53	+0.29
Mixed	ROC-AUC	94.39±0.89	94.14±0.54	−0.25
Mixed	PR-AUC	94.56±0.87	94.30±0.52	−0.26
Mixed	F1	87.05±1.10	86.85±1.01	−0.20
Mixed	EER	13.01±1.20	13.35±0.91	+0.33
Mixed	FAR@FRR1%	53.39±7.06	56.08±1.62	+2.69
Mixed	TAR@FAR1%	48.09±6.58	47.40±2.07	−0.70
Mixed	catch@3	82.04±1.62	81.27±2.08	−0.77

Table 8: PAN-20, ELECTRA-base-discriminator (110M).

Test	Metric	RO	Mix	Δ
Random	ROC-AUC	97.10±0.13	96.85±0.57	−0.25
Random	PR-AUC	97.19±0.16	96.96±0.49	−0.22
Random	F1	91.06±0.18	90.60±1.06	−0.46
Random	EER	8.95±0.09	9.49±0.94	+0.54
Random	FAR@FRR1%	36.01±4.06	38.91±7.86	+2.90
Random	TAR@FAR1%	63.37±1.99	62.73±3.91	−0.65
Random	catch@3	88.74±1.20	88.09±0.93	−0.65
Semi-hard	ROC-AUC	96.03±0.10	95.95±0.75	−0.08
Semi-hard	PR-AUC	96.25±0.19	96.12±0.65	−0.13
Semi-hard	F1	89.16±0.12	88.98±1.06	−0.18
Semi-hard	EER	10.59±0.10	11.03±1.18	+0.43
Semi-hard	FAR@FRR1%	45.57±5.01	45.74±7.44	+0.18
Semi-hard	TAR@FAR1%	58.91±2.88	58.24±3.87	−0.68
Semi-hard	catch@3	83.37±1.84	83.71±1.15	+0.33
Hard	ROC-AUC	94.18±0.33	94.70±0.59	+0.52
Hard	PR-AUC	94.36±0.40	94.84±0.59	+0.49
Hard	F1	86.56±0.19	87.26±0.55	+0.70
Hard	EER	13.24±0.61	12.88±0.75	−0.35
Hard	FAR@FRR1%	54.83±3.47	53.16±6.55	−1.66
Hard	TAR@FAR1%	47.07±2.39	50.58±2.05	+3.51
Hard	catch@3	75.69±1.76	78.62±1.89	+2.93
Mixed	ROC-AUC	96.02±0.32	95.91±0.85	−0.10
Mixed	PR-AUC	96.09±0.34	96.05±0.79	−0.04
Mixed	F1	89.22±0.29	88.94±1.38	−0.27
Mixed	EER	10.83±0.33	11.01±1.44	+0.18
Mixed	FAR@FRR1%	43.24±5.29	44.08±7.74	+0.85
Mixed	TAR@FAR1%	54.50±2.02	57.11±3.21	+2.61
Mixed	catch@3	83.61±0.89	83.67±1.37	+0.06

Table 9: WildChat-1M, DistilBERT-base-uncased (66M).

Test	Metric	RO	Mix	Δ
Random	ROC-AUC	96.28±0.22	96.38±0.12	+0.10
Random	PR-AUC	96.05±0.33	96.45±0.15	+0.40
Random	F1	90.24±0.18	89.94±0.22	−0.30
Random	EER	9.80±0.16	10.07±0.24	+0.27
Random	FAR@FRR1%	43.88±1.57	44.94±1.19	+1.06
Random	TAR@FAR1%	54.47±6.56	59.31±1.26	+4.84
Random	catch@3	88.62±1.54	88.82±0.82	+0.20
Semi-hard	ROC-AUC	95.05±0.27	95.60±0.24	+0.55
Semi-hard	PR-AUC	94.53±0.38	95.57±0.36	+1.05
Semi-hard	F1	88.74±0.28	89.07±0.25	+0.33
Semi-hard	EER	11.34±0.39	11.04±0.28	−0.30
Semi-hard	FAR@FRR1%	50.70±0.75	48.25±1.14	−2.45
Semi-hard	TAR@FAR1%	41.26±5.08	50.18±4.65	+8.92
Semi-hard	catch@3	85.06±2.32	86.88±0.90	+1.82
Hard	ROC-AUC	86.74±1.02	89.60±0.85	+2.86
Hard	PR-AUC	83.17±1.37	88.54±1.26	+5.37
Hard	F1	81.40±0.66	82.38±0.19	+0.98
Hard	EER	19.89±0.92	18.26±0.89	−1.63
Hard	FAR@FRR1%	73.89±0.41	69.37±1.21	−4.52
Hard	TAR@FAR1%	7.31±0.95	20.25±6.91	+12.94
Hard	catch@3	65.21±2.99	69.13±1.58	+3.92
Mixed	ROC-AUC	94.14±0.33	94.89±0.23	+0.74
Mixed	PR-AUC	92.86±0.54	94.58±0.38	+1.72
Mixed	F1	88.15±0.22	88.28±0.09	+0.13
Mixed	EER	12.09±0.35	12.01±0.18	−0.08
Mixed	FAR@FRR1%	51.20±0.36	49.94±1.23	−1.26
Mixed	TAR@FAR1%	22.28±3.31	41.85±2.90	+19.57
Mixed	catch@3	83.59±1.94	84.96±1.39	+1.37

Table 10: WildChat-1M, BERT-base-uncased (110M).

Test	Metric	RO	Mix	Δ
Random	ROC-AUC	96.25±0.21	96.28±0.13	+0.03
Random	PR-AUC	95.97±0.36	96.33±0.17	+0.37
Random	F1	90.12±0.47	89.78±0.26	−0.34
Random	EER	9.90±0.41	10.16±0.22	+0.26
Random	FAR@FRR1%	42.78±0.98	43.06±1.74	+0.28
Random	TAR@FAR1%	50.63±3.76	59.02±0.29	+8.39
Random	catch@3	88.61±0.18	88.87±1.75	+0.25
Semi-hard	ROC-AUC	94.97±0.28	95.54±0.25	+0.56
Semi-hard	PR-AUC	94.31±0.48	95.48±0.30	+1.17
Semi-hard	F1	88.66±0.42	88.86±0.52	+0.20
Semi-hard	EER	11.44±0.36	11.25±0.54	−0.19
Semi-hard	FAR@FRR1%	49.45±0.91	45.41±2.20	−4.03
Semi-hard	TAR@FAR1%	36.94±4.33	49.37±1.61	+12.43
Semi-hard	catch@3	84.97±0.20	86.57±1.77	+1.60
Hard	ROC-AUC	86.40±0.72	89.76±0.91	+3.36
Hard	PR-AUC	82.13±1.14	88.78±1.26	+6.65
Hard	F1	81.26±0.57	82.51±0.44	+1.25
Hard	EER	20.11±0.65	18.28±0.94	−1.83
Hard	FAR@FRR1%	72.59±0.64	66.36±1.63	−6.24
Hard	TAR@FAR1%	6.17±0.55	22.32±3.94	+16.15
Hard	catch@3	65.04±1.14	69.62±2.56	+4.59
Mixed	ROC-AUC	94.02±0.22	94.90±0.12	+0.88
Mixed	PR-AUC	92.48±0.34	94.58±0.33	+2.11
Mixed	F1	87.94±0.30	88.20±0.23	+0.25
Mixed	EER	12.32±0.28	12.04±0.27	−0.28
Mixed	FAR@FRR1%	50.27±0.64	47.77±2.38	−2.50
Mixed	TAR@FAR1%	18.90±1.75	42.16±3.68	+23.26
Mixed	catch@3	83.31±0.67	85.09±1.95	+1.78

Table 11: WildChat-1M, MiniLM-L6 (22.71M).

Test	Metric	RO	Mix	Δ
Random	ROC-AUC	96.52±0.18	96.41±0.05	−0.12
Random	PR-AUC	96.63±0.22	96.53±0.07	−0.11
Random	F1	90.29±0.28	89.84±0.12	−0.45
Random	EER	9.62±0.20	10.07±0.04	+0.45
Random	FAR@FRR1%	47.49±1.39	44.24±1.15	−3.26
Random	TAR@FAR1%	62.86±1.50	60.36±0.52	−2.50
Random	catch@3	89.80±0.29	88.79±1.43	−1.00
Semi-hard	ROC-AUC	95.58±0.16	95.86±0.03	+0.28
Semi-hard	PR-AUC	95.72±0.23	95.96±0.05	+0.24
Semi-hard	F1	89.09±0.28	89.16±0.19	+0.06
Semi-hard	EER	10.89±0.38	10.81±0.15	−0.08
Semi-hard	FAR@FRR1%	54.47±2.15	46.42±1.93	−8.05
Semi-hard	TAR@FAR1%	54.04±2.91	54.53±1.94	+0.49
Semi-hard	catch@3	86.73±0.44	87.06±1.29	+0.33
Hard	ROC-AUC	88.98±0.62	90.56±0.32	+1.58
Hard	PR-AUC	88.17±0.71	89.91±0.66	+1.74
Hard	F1	81.86±0.50	82.71±0.40	+0.84
Hard	EER	18.83±0.89	17.37±0.36	−1.47
Hard	FAR@FRR1%	76.80±2.01	67.62±1.85	−9.19
Hard	TAR@FAR1%	21.67±2.14	23.75±6.48	+2.08
Hard	catch@3	67.52±0.80	69.84±2.60	+2.32
Mixed	ROC-AUC	94.80±0.28	95.16±0.17	+0.36
Mixed	PR-AUC	94.61±0.46	95.00±0.32	+0.38
Mixed	F1	88.36±0.29	88.41±0.17	+0.05
Mixed	EER	11.81±0.28	11.74±0.34	−0.07
Mixed	FAR@FRR1%	54.98±1.88	48.80±0.29	−6.17
Mixed	TAR@FAR1%	44.40±4.21	46.77±1.12	+2.36
Mixed	catch@3	85.30±0.38	85.35±2.06	+0.05

Table 12: WildChat-1M, ELECTRA-base-discriminator (110M).

Test	Metric	RO	Mix	Δ
Random	ROC-AUC	95.71±0.47	95.79±0.32	+0.09
Random	PR-AUC	95.23±0.69	95.65±0.45	+0.43
Random	F1	89.38±0.71	89.28±0.40	−0.10
Random	EER	10.76±0.66	10.73±0.37	−0.02
Random	FAR@FRR1%	44.62±3.01	47.90±1.06	+3.28
Random	TAR@FAR1%	45.23±6.05	52.85±4.46	+7.62
Random	catch@3	86.89±1.01	87.05±1.32	+0.16
Semi-hard	ROC-AUC	94.36±0.37	94.80±0.39	+0.44
Semi-hard	PR-AUC	93.32±0.38	94.36±0.66	+1.04
Semi-hard	F1	87.87±0.67	88.19±0.38	+0.32
Semi-hard	EER	12.16±0.59	11.95±0.42	−0.21
Semi-hard	FAR@FRR1%	50.51±1.85	50.73±1.54	+0.22
Semi-hard	TAR@FAR1%	29.91±5.00	39.03±6.91	+9.12
Semi-hard	catch@3	82.98±1.45	84.22±1.40	+1.23
Hard	ROC-AUC	85.08±0.82	87.57±1.66	+2.49
Hard	PR-AUC	79.62±1.44	84.63±3.00	+5.02
Hard	F1	80.35±0.56	81.34±0.48	+0.99
Hard	EER	21.29±0.77	19.57±1.47	−1.72
Hard	FAR@FRR1%	72.92±0.19	72.03±0.47	−0.89
Hard	TAR@FAR1%	4.31±0.52	9.91±3.87	+5.60
Hard	catch@3	62.39±0.95	65.65±1.25	+3.26
Mixed	ROC-AUC	93.30±0.36	93.97±0.38	+0.67
Mixed	PR-AUC	91.22±0.74	92.92±0.81	+1.70
Mixed	F1	87.16±0.46	87.51±0.28	+0.34
Mixed	EER	13.07±0.45	12.69±0.39	−0.38
Mixed	FAR@FRR1%	51.27±0.90	53.19±0.40	+1.92
Mixed	TAR@FAR1%	15.05±4.43	25.98±6.86	+10.93
Mixed	catch@3	81.42±0.29	83.10±1.85	+1.68

B SBERT Cross-Miner Ablation, Full Detailed Results

Table 13 gives the cross-miner summary on the Hard regime, Table 14 reports the random→hard robustness gap under SBERT mining, and Tables 15–18 report the full per-regime breakdowns. "SBERT eval, RO training" means random-only training evaluated against SBERT-mined imposters; "SBERT eval, MIX TF-IDF" means mixed TF-IDF training evaluated against SBERT-mined imposters; and "SBERT eval, MIX SBERT" is the same-miner upper bound.

Why this ablation matters. A natural worry about the TF-IDF hardness story in Section A is that the encoder may have learned to defeat *TF-IDF-mined imposters specifically*, not topic-similar imposters in general. If swapping the test miner from TF-IDF to SBERT collapsed the Mix gains, the topic-vs-style fix would be miner-specific rather than a general property of hardness-aware training. The cross-miner test ("MIX TF-IDF" column under SBERT eval) checks this directly: train on TF-IDF negatives, evaluate against SBERT-mined imposters the model has never seen during training. The same-miner column ("MIX SBERT") is the matched-miner comparison: train and evaluate against imposters selected by the same miner.

What the cross-miner numbers show. On PAN the three SBERT-eval columns are close to each other (93.6–94.8 ROC-AUC for both encoders), consistent with PAN's small random-only gap leaving little room for miner-specific behavior. On WildChat the picture is more informative: under random-only training, switching from TF-IDF to SBERT imposters lowers Hard ROC-AUC by a further 3.3 points (BERT $86.40 \rightarrow 83.07$, DistilBERT $86.74 \rightarrow 83.46$), showing that the topic-vs-style failure mode transfers across miners. Cross-miner-trained models (TF-IDF train, SBERT test) recover part of the loss (BERT to 85.95, DistilBERT to 85.75), and the matched-miner setting adds a smaller additional ROC-AUC gain (BERT 86.43, DistilBERT 86.60). This is consistent with hardness-aware training improving robustness beyond the specific TF-IDF miner. The TAR@FAR1% numbers tell a stronger version of the same story: WildChat Hard TAR@FAR1% under SBERT evaluation moves from 5.2% to 18.6% to 18.3% for BERT, and from 7.4% to 18.6% to 20.3% for DistilBERT.

Table 13: Cross-miner summary on the Hard regime for ROC-AUC and TAR@FAR=1%. Full per-regime details are in Tables 15–18.

		TF-IDF eval	SBERT eval		
Dataset	**Model**	MIX orig	RO	MIX TF-IDF	MIX SBERT
Hard regime ROC-AUC (%)					
PAN	DistilBERT	93.95	93.97	93.59	94.45
PAN	BERT	94.54	94.48	94.31	94.78
WildChat	DistilBERT	89.60	83.46	85.75	86.60
WildChat	BERT	89.76	83.07	85.95	86.43
Hard regime TAR@FAR=1% (%)					
PAN	DistilBERT	46.08	45.95	45.35	48.48
PAN	BERT	50.87	49.84	47.88	49.74
WildChat	DistilBERT	20.25	7.41	18.55	20.25
WildChat	BERT	22.32	5.21	18.64	18.33

Table 14: SBERT ablation random→hard **robustness gap**. RO = random-only training; Mix = SBERT-mined mixed training. Negative Δ means the gap shrank.

		ROC-AUC gap			Other Δ	
Dataset	**Encoder**	RO	Mix	Δ	PR-AUC	F1
PAN-20	DistilBERT	2.82±0.70	2.30±0.16	−0.52	−0.51	−0.69
	BERT	2.69±0.43	2.16±0.47	−0.53	−0.37	−1.08
WildChat	DistilBERT	12.84±0.65	9.91±0.68	−2.93	−5.26	−1.33
	BERT	13.16±0.82	9.91±0.63	−3.25	−6.42	−1.06

Table 15: SBERT ablation: PAN-20, DistilBERT.

Test	**Metric**	**RO**	**Mix**	**Δ**
Random	ROC-AUC	96.79±0.51	96.75±0.34	−0.04
Random	PR-AUC	96.88±0.48	96.86±0.31	−0.02
Random	EER	9.48±0.88	9.66±0.79	+0.18
Random	FAR@FRR1%	39.82±6.81	39.74±1.15	−0.08
Semi-hard	ROC-AUC	95.57±0.61	95.58±0.45	+0.01
Semi-hard	PR-AUC	95.73±0.44	95.62±0.37	−0.11
Semi-hard	EER	11.51±1.04	11.28±0.75	−0.23
Semi-hard	FAR@FRR1%	48.78±8.31	47.28±1.55	−1.50
Hard	ROC-AUC	93.97±0.73	94.45±0.45	+0.48
Hard	PR-AUC	94.10±0.67	94.59±0.48	+0.49
Hard	F1	86.48±1.03	87.12±0.59	+0.63
Hard	EER	13.66±0.92	13.02±0.54	−0.64
Hard	FAR@FRR1%	57.10±7.89	54.28±1.70	−2.82
Hard	TAR@FAR1%	45.95±4.20	48.48±4.84	+2.53
Hard	catch@3	78.40±3.20	79.99±2.58	+1.59
Mixed	ROC-AUC	95.80±0.37	95.83±0.44	+0.03
Mixed	PR-AUC	95.89±0.32	95.83±0.49	−0.05
Mixed	EER	11.16±0.47	11.00±0.61	−0.17
Mixed	FAR@FRR1%	46.94±7.80	45.11±1.69	−1.84
Mixed	catch@3	85.57±1.09	85.53±1.76	−0.04
Mixed	TAR@FAR1%	56.38±0.92	53.73±4.75	−2.64

Table 16: SBERT ablation: PAN-20, BERT.

Test	**Metric**	**RO**	**Mix**	**Δ**
Random	ROC-AUC	97.17±0.45	96.95±0.64	−0.23
Random	PR-AUC	97.30±0.41	97.06±0.63	−0.23
Random	EER	8.94±0.70	9.38±1.13	+0.43
Random	FAR@FRR1%	38.47±6.05	37.11±5.35	−1.37
Semi-hard	ROC-AUC	95.87±0.54	95.77±0.67	−0.10
Semi-hard	PR-AUC	96.00±0.44	95.79±0.58	−0.21
Semi-hard	EER	10.83±1.08	11.08±1.12	+0.25
Semi-hard	FAR@FRR1%	47.15±6.90	43.94±4.96	−3.21
Hard	ROC-AUC	94.48±0.58	94.78±0.76	+0.30
Hard	PR-AUC	94.69±0.59	94.83±0.84	+0.14
Hard	F1	87.09±0.74	87.53±0.81	+0.43
Hard	EER	12.93±0.85	12.58±0.96	−0.35
Hard	FAR@FRR1%	55.77±6.01	50.29±3.95	−5.47
Hard	TAR@FAR1%	49.84±3.46	49.74±6.33	−0.10
Hard	catch@3	79.64±1.65	82.61±1.76	+2.97
Mixed	ROC-AUC	96.07±0.39	95.89±0.58	−0.18
Mixed	PR-AUC	96.06±0.51	95.86±0.55	−0.20
Mixed	EER	10.72±0.52	10.93±0.89	+0.21
Mixed	FAR@FRR1%	45.11±5.86	41.72±4.29	−3.39
Mixed	catch@3	85.87±1.09	86.81±1.57	+0.94
Mixed	TAR@FAR1%	55.59±5.87	53.01±4.19	−2.58

Table 17: SBERT ablation: WildChat-1M, DistilBERT.

Test	**Metric**	**RO**	**Mix**	**Δ**
Random	ROC-AUC	96.30±0.20	96.51±0.11	+0.21
Random	PR-AUC	96.12±0.33	96.62±0.19	+0.50
Random	EER	9.90±0.28	9.85±0.08	−0.05
Random	FAR@FRR1%	42.30±2.59	44.96±1.03	+2.67
Semi-hard	ROC-AUC	94.88±0.41	95.51±0.20	+0.64
Semi-hard	PR-AUC	94.86±0.50	95.81±0.23	+0.95
Semi-hard	EER	12.23±0.46	11.67±0.35	−0.56
Semi-hard	FAR@FRR1%	50.90±2.08	50.72±1.81	−0.18
Hard	ROC-AUC	83.46±0.83	86.60±0.79	+3.14
Hard	PR-AUC	80.33±1.52	86.09±1.30	+5.76
Hard	F1	77.66±0.67	78.80±0.32	+1.14
Hard	EER	23.33±0.51	21.59±0.60	−1.75
Hard	FAR@FRR1%	79.51±0.75	79.04±0.80	−0.47
Hard	TAR@FAR1%	7.41±1.65	20.25±4.30	+12.84
Hard	catch@3	53.27±3.46	57.96±2.43	+4.69
Mixed	ROC-AUC	93.40±0.23	94.34±0.20	+0.94
Mixed	PR-AUC	92.41±0.50	94.15±0.40	+1.74
Mixed	EER	13.60±0.24	13.02±0.35	−0.59
Mixed	FAR@FRR1%	51.56±0.26	53.65±1.97	+2.09
Mixed	catch@3	80.04±2.04	82.47±1.66	+2.43
Mixed	TAR@FAR1%	26.13±5.12	43.66±5.92	+17.53

Table 18: SBERT ablation: WildChat-1M, BERT.

Test	**Metric**	**RO**	**Mix**	**Δ**
Random	ROC-AUC	96.23±0.39	96.35±0.21	+0.12
Random	PR-AUC	95.97±0.59	96.43±0.24	+0.47
Random	EER	9.78±0.44	10.11±0.56	+0.32
Random	FAR@FRR1%	43.77±2.09	45.13±2.64	+1.36
Semi-hard	ROC-AUC	94.81±0.70	95.34±0.14	+0.53
Semi-hard	PR-AUC	94.66±1.01	95.62±0.20	+0.96
Semi-hard	EER	12.20±0.64	12.08±0.16	−0.12
Semi-hard	FAR@FRR1%	52.02±0.71	50.54±3.54	−1.49
Hard	ROC-AUC	83.07±1.21	86.43±0.78	+3.37
Hard	PR-AUC	78.88±1.97	85.76±1.00	+6.89
Hard	F1	77.77±0.52	78.61±0.58	+0.84
Hard	EER	23.16±1.08	21.61±0.82	−1.54
Hard	FAR@FRR1%	79.70±0.84	78.24±1.93	−1.46
Hard	TAR@FAR1%	5.21±1.11	18.33±1.05	+13.12
Hard	catch@3	54.05±1.95	55.39±1.22	+1.34
Mixed	ROC-AUC	93.33±0.51	94.12±0.29	+0.79
Mixed	PR-AUC	92.01±1.11	93.88±0.30	+1.87
Mixed	EER	13.60±0.31	13.26±0.53	−0.34
Mixed	FAR@FRR1%	52.98±0.68	53.08±2.72	+0.11
Mixed	catch@3	80.01±0.86	80.57±0.92	+0.57
Mixed	TAR@FAR1%	21.86±9.03	40.30±3.01	+18.43