SEQBENCH: BENCHMARKING CUMULATIVE PRIVACY LEAKAGE IN LONGITUDINAL TEXT

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models infer a writer's attributes from evidence accumulated across posts and linked accounts. We call a target *cumulative* when the history recovers it and no post alone does. We prove that a per-post anonymiser that detects no more than an evidence-based reader and edits only what it detects leaves every cumulative target recoverable, and that, for a reader of the current archive, only revising released posts repairs a released leak. An anonymiser that edits only what it detects must therefore be *sequential*: it reads the history and revises what it has released. SeqBench is built to measure this separation. Its eleven synthetic corpora of up to 1,000 personas spread designed targets over several posts, most across platforms, and a blind leave-one-post-out certifier admits a target as cumulative only if a reader recovers it from its support and from no labelled post alone. On six further corpora from the same generator, labelled one post at a time, which mark no conjunctive route, the only kind per-post anonymisation provably cannot see, our sequential anonymiser edits 406 posts that per-post leaves unedited (86 the other way) and reaches back to 282 of its edited posts after release. Under a same-family attacker both cut strict value recall from 0.347 to 0.049 (per-post) and 0.056 (sequential), with no significant difference ($p = 0.30$); the 90% interval of their type-F1 difference lies within ± 0.031. In a four-seat loop run by one minimal-edit script, the sequential scope edits fewer posts and keeps more utility in every seat ($+0.058$, $p = 0.002$), while round-4 type recall does not differ significantly.

1 INTRODUCTION

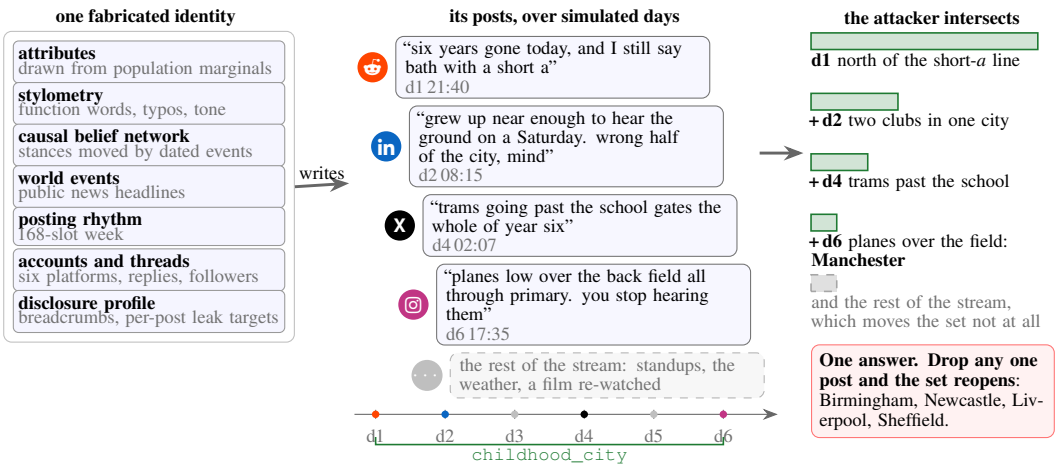

Figure 1: **One cumulative target** (illustrative). Four posts each rule out a different rival city and none names the answer, so removing any one reopens the set.

The code, six of SeqBench's eleven corpora and the evaluation's saved answers are in the supplementary material (Appendix L). The companion submission (Anonymous, 2026) measures deanonymisation, which this paper does not.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

An off-the-shelf language model names a writer's location, age, sex, occupation and income from a handful of unremarkable posts, nearly as accurately as human experts (Staab et al., 2024). No single post need give an attribute away (Figure 1). Evidence accumulates across a user's posts and linked accounts, as answers do in the tracker attack on statistical databases (Denning et al., 1979) and observations do in re-identification (de Montjoye et al., 2013). Yet anonymisers rewrite one fixed text, a post or a finished profile, until the adversary's inference fails (Staab et al., 2025). A per-post rewriter never sees split evidence, a whole-profile rewriter sees it only once every post is public, and to our knowledge none revisits what it has released (Section 2).

We argue that leakage in longitudinal, cross-platform text is cumulative, so an anonymiser must read a user's history and revise what it has released; we call this *sequential anonymisation* in the idealised scope (Definition 3). A target is *cumulative* when the history recovers it and no post alone does (Section 3). A per-post anonymiser that detects no more than an evidence-based reader and edits only what it detects leaves every such target recoverable, however hard it edits what it detects (Proposition 1); to close such a route without the history it must break those hypotheses, e.g. by over-editing (Appendix B). A window narrower than every minimal support of a target fails the same way (Proposition 3), and, for a reader of the archive as it stands, only revising released posts can repair a leak once it is out (Proposition 2).

Testing this needs known supports. None of the six closest resources (Figure 2) labels leakage post by post for one person across platforms, and only SopriBench (Peng et al., 2026) labels cross-post targets, which it reads off its annotation with no rule for testing them against a reader. SeqBench (Section 4) builds targets from partial constraints spread over posts and platforms, and its admission rule, a blind leave-one-post-out certifier, counts a target as cumulative only if a reader recovers the value from its support and from no labelled post alone. Its planner holds each shard back until 2 to 10 of the author's posts have passed, so at $1{,}000$ personas 35.8% of the designed supports are wider than 8 posts; 67.4% of all designed groups cross platforms.

Our scope comparison runs on further corpora from the same generator, kept outside the release and labelled one post at a time, so no label marks a conjunctive route, the only kind on which Proposition 1 separates the scopes; it measures what reaching the history costs. On SeqBench-6seed, six 20-persona corpora, the sequential anonymiser of Section 5 edits 406 posts that per-post anonymisation leaves unedited (86 the other way) and reaches back to 282 of its edited posts after their release, in 89 of the 119 personas who post (Sections 5 and 6.2). Strict value recall, the primary score, falls from 0.347 raw to 0.049 per-post and 0.056 sequential, with no significant difference ($p = 0.30$), and the 90% interval of the type-F1 difference lies within ± 0.031, so on the primary score reaching the history costs no significant leakage; several secondary readings favour per-post (Section 6.2). In a four-seat adversarial loop on SeqBench-1k/150, where one minimal-edit script runs both scopes, the sequential scope edits fewer posts and keeps more utility in every seat (sequential minus per-post $+0.058$, $p = 0.002$), also over each arm's edited posts alone (Table 11).

- We define cumulative leakage and prove that a per-post anonymiser cannot see it and that, for a reader of the current archive, only revising released posts can repair a released leak.
- We give a sequential anonymiser that reads the history and rewrites released posts.
- We release SeqBench, six-platform corpora whose designed supports spread one attribute over several posts, with a certifier that admits a target as cumulative against a reader.
- We show that on per-post-labelled corpora sequential anonymisation costs no significant leakage on the primary score and keeps more utility in a four-seat adversarial loop.

2 Background and Related Work

Composition under sequential release. Releases that are each safe can disclose jointly (Ganta et al., 2008). Sequential-release anonymisation (Wang & Fung, 2006; Xiao & Tao, 2007) and query auditing (Kenthapadi et al., 2005) therefore judge each release against past ones, and continual-observation differential privacy bounds the loss over a stream (Dwork et al., 2010), all on records, counts and query answers. The cumulative class is this composition in free text. Sequential anonymisation applies their rule to it, judging each release against past ones, and also revises past releases, which those methods do not, since their adversary keeps every release (Appendix B). None of them acts on text, and none labels leakage (Appendix A, with memorisation- and norm-side benchmarks).

	scale			releasable	ground-truth	coverage		
	posts	attr.	plat.	(synthetic)	attributes	cross-plat.	longit.	cumulative
TAB	1,268	8	n.r.	○	◑	○	○	○
SynthPAI	7,823	8	1	●	◑	○	○	○
RAT-Bench	100	15	n.r.	●	●	○	○	○
SPIA	675	15	n.r.	○	◑	○	○	○
SopriBench	500	28	1	●	●	○	●	asserted
TWIN	1.7M	n.r.	2	○	○	●	●	○
SeqBench (ours)	142,536	27	6	●	●	●	●	designed + certifier

● has it, or ground truth *drawn* ◑ partial: ground truth *annotated* ○ lacks it, or *none* n.r. unreported

Figure 2: **What the closest resources supply.** SeqBench counts: largest over its corpora (Appendix A).

Inference and its defences. The main defence against LLM attribute inference (Staab et al., 2024) is adversarial: Staab et al. (2025) pit an LLM adversary against an LLM anonymiser that minimally edits until the inference fails, and on attribute inference it beats commercial and rule-based de-identifiers. Dou et al. (2024) instead fine-tune models to abstract self-disclosures. Later work trains smaller local anonymisers on the loop's own signal, AgentStealth (Shao et al., 2025) by contrastive adaptation and online RL, SEAL (Kim et al., 2025) by distillation and GRASP (Ghiasvand & Sehatbakhsh, 2026) by GRPO, while AURA (Li et al., 2026) sets a privacy scope per transcript against web-search re-identification agents. IncogniText (Frikha et al., 2024) misleads the adversary towards a wrong value, RUPTA (Yang et al., 2025) trades re-identification risk against utility, and locally differentially private rewriting gives each document a formal guarantee (Utpala et al., 2023; Igamberdiev & Habernal, 2023). Each rewrites one fixed text.

What anonymisation means here. De-identification removes predefined identifier types and scores span precision and recall (Stubbs & Uzuner, 2015). Anonymisation in the data-protection sense asks whether a person stays identifiable (Lison et al., 2021), and classical disclosure control mainly tests it against re-identification (Lambert, 1993; Sweeney, 2002); TAB (Pilán et al., 2022) applies this sense to text, scoring entity recall over expert-marked direct and quasi-identifiers. Attribute disclosure can occur without identity disclosure, so neither identifier recall nor re-identification risk bounds sensitive-attribute inference (Lambert, 1993; Li et al., 2007). Staab et al. (2025) and successors such as AgentStealth (Shao et al., 2025) and SEAL (Kim et al., 2025) call a rewrite anonymised when an adversary no longer infers the author's attributes, and score the adversary's accuracy. We use *anonymiser* and *anonymisation* in this sense unless we name another. Definition 1 concerns attribute inference alone, with no claim about re-identification, k-anonymity or differential privacy (Appendix A).

Per-document benchmarks. Classical de-identification, TAB and utility-paired frameworks (Loiseau et al., 2025) score one document at a time. The 2014 i2b2/UTHealth corpus gathers several records per patient because identifiers accumulate across them, yet annotates and scores each record alone (Stubbs & Uzuner, 2015). Two 2026 resources move the unit outward without reaching the conjunction. SPIA (Oh et al., 2026) scores the subject rather than the span and finds two-thirds of PII still inferable at 99.7% entity recall. RAT-Bench (Krčo et al., 2026) scores population re-identification risk on texts in isolation. SynthPAI (Yukhymenko et al., 2024) labels inferable attributes post hoc on single-platform Reddit. Closest to us, SopriBench (Peng et al., 2026) targets user-level, cross-post leakage on social media, annotates the supporting evidence and scores exposure on a graded scale; its cross-post targets are *asserted* from the annotation, with no reader test; Section 4.2 gives SeqBench's rule for one. TWIN (Chen et al., 2021) links $5{,}765$ Twitter accounts to their owners' Instagram accounts, the link our threat model grants the reader; with no leak labels, it can drive an attack but cannot score a defence.

3 THREE CLASSES OF LEAKAGE

Write $\mathcal{X}$ for the space of post texts, $\mathcal{K}$ for a finite set of sensitive attributes, each with a finite value set V_A, and $\mathcal{Z} = \{(A, v) : A \in \mathcal{K},\, v \in V_A\}$. A user authors posts $X_1, \ldots, X_T \in \mathcal{X}$; the index pools every platform they write on into one chronological stream, so $X_{1:T}$ is the *linked* cross-platform history. Write $X_{1:t}$ for the history through t and $X_S = (X_i)_{i \in S}$ for a subhistory. The adversary is a reader granted the linked history, identified with a map $\mathcal{D}\colon \mathcal{X}^* \to 2^{\mathcal{Z}}$ returning the attribute–value pairs it infers from the texts it is shown. It is scored against $Z = \{(A, v_A) : A \in \mathcal{K}\}$, where $v_A \in V_A$ is the author's true value on A.

Every result about $\mathcal{D}$ rests on one assumption. We say $\mathcal{D}$ is *evidence-based* if each pair $z \in Z$ it can infer is carried by identifiable phrases, and $\mathcal{D}$ recovers z from a set of texts if and only if that set still contains a supporting collection of them; such a $\mathcal{D}$ is *evidence-monotone*, since a supporting collection present in X_S survives in $X_{S'}$ whenever $S \subseteq S'$: $\mathcal{D}(X_S) \subseteq \mathcal{D}(X_{S'})$, and extra text never retracts an inference. This is the adversary the anonymiser is built against (its detector is asked to quote a phrase per value it names) and the one Section 4.2's certifier instantiates.

Definition 1 (Sequential anonymisation)**.** A *sequential anonymiser* is a family of maps $(\mathcal{A}_t)_{t \le T}$ that at each t turns the history $X_{1:t}$ into a release $Y_{1:t}$. It *solves* the task against $\mathcal{D}$ if $\mathcal{D}(Y_{1:t}) \cap Z = \emptyset$ for every $t \le T$. Staab et al. (2025) rewrite only the finished history (the whole-profile setting); run per post, their rewriter is the special case in which $\mathcal{A}_t$ depends on X_t alone and emits only Y_t.

Definition 2 (Minimal supports and the three leakage classes)**.** Let $z \in Z$ be recoverable at time t, i.e. $z \in \mathcal{D}(X_{1:t})$. The *minimal supports* of z at t are the inclusion-minimal $\sigma \subseteq \{1, \ldots, t\}$ with $z \in \mathcal{D}(X_\sigma)$, written $\Sigma_t(z)$: the distinct routes by which z is recoverable, of which there may be several. Split them by size, $\Sigma_t^{\mathrm{sng}}(z) = \{\sigma \in \Sigma_t(z) : |\sigma| = 1\}$ for the routes carried by a single post and $\Sigma_t^{\mathrm{cnj}}(z) = \{\sigma \in \Sigma_t(z) : |\sigma| \ge 2\}$ for the conjunctive ones. Then z is

- **single-post** if $\Sigma_t^{\mathrm{sng}}(z)$ holds exactly one route and $\Sigma_t^{\mathrm{cnj}}(z) = \emptyset$ (only one post carries it and no others do);
- **redundant** if $\Sigma_t^{\mathrm{sng}}(z)$ holds at least two and $\Sigma_t^{\mathrm{cnj}}(z) = \emptyset$ (several posts each carry it independently);
- **cumulative** if $\Sigma_t^{\mathrm{sng}}(z) = \emptyset$ (recoverable from the history and from no post alone).

The remaining cell, $\Sigma_t^{\mathrm{sng}}(z) \neq \emptyset$ and $\Sigma_t^{\mathrm{cnj}}(z) \neq \emptyset$, is **mixed**: a single-post route and a conjunctive route coexist, so editing the post that carries z alone leaves the conjunction intact. Whenever $\Sigma_t^{\mathrm{cnj}}(z) \neq \emptyset$, $\mathrm{span}_t(z) = \max_{\sigma \in \Sigma_t^{\mathrm{cnj}}(z)} \big(\max \sigma - \min \sigma\big)$ is the widest stretch of history any one conjunctive route demands; the maximum is over routes because a defence must close every one of them, where an attacker need only walk the cheapest.

Proposition 1 (Separation)**.** *Let $\mathcal{D}$ be evidence-based and $\mathcal{A}^{\mathrm{pp}}$ a per-post anonymiser, $Y_i = \mathcal{A}^{\mathrm{pp}}(X_i)$, such that (i) it detects in X_i only pairs in $\mathcal{D}(X_i)$, (ii) it changes X_i only inside phrases supporting pairs it detects there, and (iii) no phrase it changes overlaps a phrase supporting a pair of Z it does not detect there. If $z \in Z$ is cumulative or mixed at t ($\Sigma_t^{\mathrm{cnj}}(z) \neq \emptyset$), then $z \in \mathcal{D}(Y_{1:t})$: Definition 1's objective fails at t, however hard $\mathcal{A}^{\mathrm{pp}}$ edits the evidence it detects.*

The failure is informational: no route in $\Sigma_t^{\mathrm{cnj}}(z)$ is visible in any single post, so a per-post anonymiser cannot see the conjunction. Under the same hypotheses, Proposition 3 shows a width-k window editing only the evidence it sees neutralises such a z only if $\mathrm{span}_t(z) \le k$ when wide routes carry their own evidence, so span is the quantity a benchmark must report alongside the class.

Definition 3 (Edit scope)**.** The *idealised* scope maps $X_{1:t} \mapsto Y_{1:t}$ and may revise the entire history; the *realistic* scope maps $[X_{1:t-1}]\, X_t \mapsto Y_t$ and the *online* scope $[Y_{1:t-1}]\, X_t \mapsto Y_t$, both rewriting only the newest post with the bracketed history read-only. Write $Y_{1:t}^{(t)}$ for the release after step t, and $\mathcal{F}_{\mathrm{ide}}, \mathcal{F}_{\mathrm{real}}, \mathcal{F}_{\mathrm{onl}}$ for the sets of anonymisers of each scope, each viewed as a map from a raw history $X_{1:T}$ to its sequence of releases $(Y_{1:t}^{(t)})_{t \le T}$ in which $Y_{1:t}^{(t)}$ depends on $X_{1:t}$ alone. Elsewhere $Y_{1:t}$ abbreviates $Y_{1:t}^{(t)}$.

We target the *idealised* scope, which revises the whole history and so can repair a leak that newest-post scopes cannot (Proposition 2); proofs are in Appendix C, and the model's four boundaries (a snapshot adversary, phrase-localised leakage, a deterministic $\mathcal{D}$, no utility constraint) in Appendix B.

Per-class predictions. On single-post and redundant targets, where $\Sigma_t^{\mathrm{cnj}}(z) = \emptyset$, Proposition 1 predicts nothing: both scopes see the evidence, both can edit it, and any difference is detector coverage and editing efficiency rather than reach, so a corpus carrying only those two classes cannot test the claim in either direction. A null on it is uninformative, and so is a win. Only a target with a certified conjunctive route, cumulative or mixed, puts the two scopes on opposite sides of an information boundary.

4 SeqBench: construction and certification

A cumulative target is testable only against an answer key that records which posts jointly support an attribute and where they sit in time (Definition 2). Real span-annotated corpora mark identifying text inside one document (Pilán et al., 2022), lack exact per-post ground truth, and would need consented cross-platform linkage (European Parliament and Council of the European Union, 2016), so SeqBench builds its targets. A seeded generator, Mirage, fabricates *de novo* identities and simulates their coordinated activity across six platforms over simulated time (Figure 3).

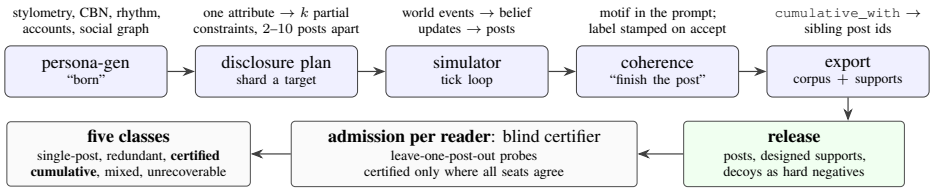

Figure 3: **Generation and release** (certifier: Appendix F).

The generator cannot be a post sampler: leakage accumulates only out of consistent, correlated behaviour. One identity drives every channel of Figure 1, from a stylometric fingerprint (Narayanan et al., 2012) and a homophily follower graph (Narayanan & Shmatikov, 2009) to the disclosure profile that becomes the leak targets, and several channels are generator switches. A *breadcrumb* is the atom of the ground truth: an oblique fact whose *motif* tells the generator how to embed it and whose *leak targets* name the attributes it concedes, each with its reasoning chain.

4.1 Designing a multi-post support

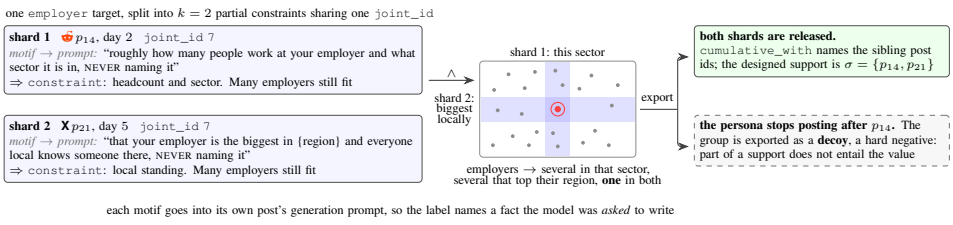

Figure 4: **Sharding one attribute into partial constraints.** Only the conjunction leaves one employer.

The disclosure planner runs before any generation. It splits one attribute into $k \geq 2$ partial constraints, each narrowing the value without determining it (Figure 4), and finishes a started group before it opens another. A shard becomes eligible only once at least g of the persona's own posts have passed since the group's preceding shard, with g drawn per group uniformly from 2 to 10. At 1,000 personas the 4,375 designed supports hold two to four shards (median span 7 posts; Figure 7). A designed support is only a candidate, since its class depends on the reader; a user certifies the supports against the readers they evaluate (Section 4.2).

4.2 CERTIFICATION

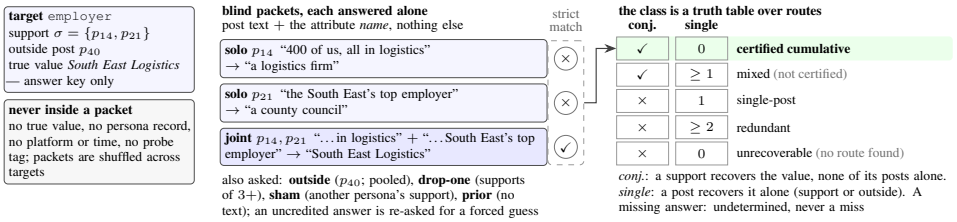

Figure 5: **Inside the certifier** (illustrative). One employer target, its blind probes and its truth table.

The class is a claim about a reader, so the certifier tests it with one. It gives a reader blind packets holding nothing but an attribute's name and post text, and reads the class off the truth table of Figure 5. A packet recovers the value only under the strict matcher, when the answer names exactly one value and that value is the truth. A generous matcher, which raises solo and joint hits alike, has no safe direction and is only a sensitivity. Only groups on the 14 *pool* attributes, each with a best constant guess below 50% (Table 5), can yield a certified target, and a target is certified only if every reader seat that audits it certifies it under the null rule (a refusal is a miss). Appendix F gives the probes and rules.

Corpora. SeqBench holds eleven separately generated corpora of 50 to 1,000 personas (Table 4) from a deliberately confusable population (Appendix E), with unclosed groups kept as decoys (Figure 4). Of the designed groups, 67.4% span two or more platforms, so only linked accounts show their supports whole.

5 SEQUENTIAL ANONYMISATION

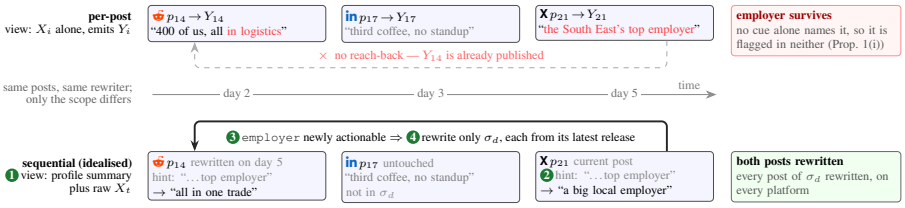

Figure 6: **Only the scope differs** (illustrative, under Proposition 1's hypotheses). Post p_{21} completes the employer route; only the sequential arm sees it and can revise the released p_{14}.

The sequential anonymiser reads a summary of the history, which a per-post anonymiser lacks (Proposition 1), and rewrites released posts, so it can repair a leak found only after its posts are out (Proposition 2). It runs the infer–rewrite–verify loop of Staab et al. (2025) in one pass over the author's timeline, all platforms pooled: a *detector* names the attributes a text exposes and the evidence for each, a rewriter rewrites the whole post with that evidence as hints, and a verifier re-attacks the rewrite. The per-post arm runs the same loop with the profile cleared before every post, so the arms differ in scope alone and the comparison does not measure rewriting power. Per-post is the comparator because a whole-profile rewrite (Staab et al., 2025) protects no post released before the history ends. The loop does not enforce hypotheses (i)–(iii) of Proposition 1, and a whole-post rewrite can also remove evidence the detector missed.

Definition 4 (Inference detail and profile)**.** An *inference detail* is a tuple $d = (z_d, \sigma_d, e_d, \tau_d, c_d)$ reported by the detector: an attribute key from the detector's vocabulary with a free-text value, together z_d, which need not lie in $\mathcal{Z}$; the set σ_d of posts it names as jointly supporting z_d, its *support set*, which always includes the post it is reading; one evidence phrase e_d, a short quote it is asked to copy verbatim from that post (the copy is not checked, and the quote may be missing); a sensitivity tier τ_d; and a confidence c_d. The detector returns at most one detail per attribute at each step. The

profile P_t holds at most one detail per attribute after step t, shared across all of the user's platforms. A new detail on an attribute already held replaces the stored one only if it is stronger (by tier, then confidence, then $|\sigma_d|$). Either way, the stored support set becomes the union of the two support sets. A detail is *actionable* if τ_d and c_d clear its attribute's gate, a minimum tier and confidence, and an attribute is actionable while its stored detail is. A support set is what the detector reported, not a minimal support (Appendix D).

For each new post X_t the anonymiser performs four steps (numbered in Figure 6):

1. **Builds the view**: a compact summary of the profile (attribute, value, sensitivity tier and supporting post ids) plus the *raw* text of X_t. Past posts are not re-sent, so the pass is linear rather than quadratic in the number of posts.
2. **Infers** over that view, returning inference details in the sense of Definition 4, each with a sensitivity tier and a confidence.
3. **Merges** every inference into P_t, actionable or not, so a later post can complete a partial one, and returns the *newly actionable* details.
4. **Rewrites the touched posts** (every released post in a newly actionable detail's support set, and the current post for an attribute already actionable), each whole from its *latest release* with its attributes' stored evidence phrases as hints. A verifier re-checks each rewrite; a post that still leaks is rewritten again, and on a second failure the evidence is redacted without a further check.

In the SeqBench-6seed run of Section 6.2, step 4 reaches back to 282 of the arm's edited posts after their release, in 89 of the 119 personas who post, and the pass costs 3,115 tokens a post against 2,227 for per-post (Appendix N).

6 Evaluation

Metrics. A target is a (persona, attribute) pair with a true value (Tables 2 and 3). An attacker's answer is quote-valid when its quote occurs in the text the attacker saw. We score *type recall*, the share of targets whose attribute a quote-valid answer names, whatever value it gives; type precision is the share of the attributes named that are targets, and F1 combines the two. A null attacker naming every attribute scores type recall 1, so only differences between arms compare. A strict re-score checks the value: a target is recovered when the first quote-valid answer on its attribute names exactly one value, the true one, and *strict value recall* is the recovered share; a generous matcher is a sensitivity. *Chance-corrected (cc) recall* discounts each pool attribute's best constant answer (Appendix G). Utility has three tiers: *surface*, a character n-gram cosine to the original; *semantic*, an embedding cosine; *factual*, the share of content words outside every flagged span that survive.

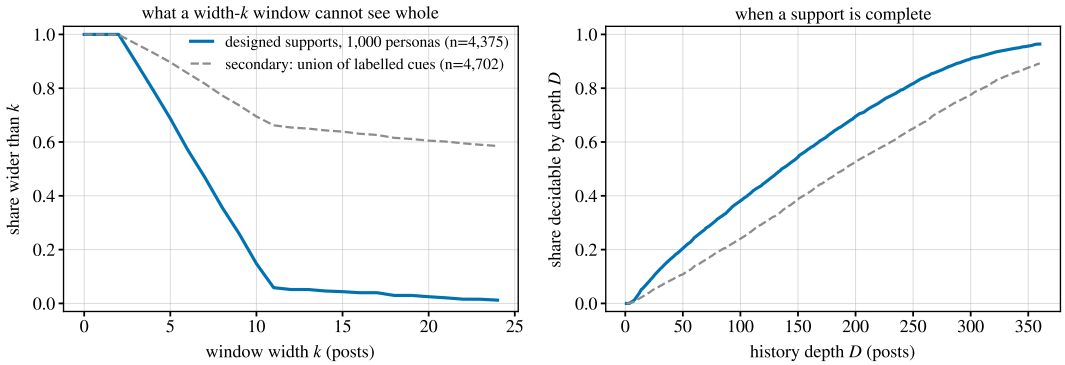

Figure 7: **Span and completion depth** of the designed supports, from the generator's records with no attacker. Dashed: the union of a target's labelled cues (Appendix C).

6.1 How far the evidence spreads

A width-k window never holds a support wider than k whole, so under Proposition 1's hypotheses it cannot neutralise a target whose minimal supports are all wider (Proposition 3). At $1{,}000$ personas 35.8% of SeqBench's designed supports are wider than 8 posts and 4.0% wider than 16; the median support completes at post 136 of its author's timeline, and only 7.9% complete within its first 20 posts (Figure 7; Appendix C). Labels assigned after generation also recur across the history. SeqBench-1k, a $1{,}000$-persona corpus of $15{,}785$ posts generated apart from SeqBench's eleven, was labelled by a detector reading one post at a time; $1{,}482$ of its $4{,}510$ targets carry cues on two or more posts, each cue labelled on its own post, a median of 13 posts apart, and 71% of those spread over more than 8 posts (Appendix C). These are repeated cues, not designed supports, so they bound no window, but they show one attribute returning across the history. On SeqBench-6seed the sequential anonymiser's own detector backs 980 profile entries with two or more posts, a median of 10 posts apart (support sets, not minimal supports; Definition 4).

6.2 Per-post against sequential

On SeqBench-6seed, six 20-persona corpora of 7 simulated days (seeds 43 to 48, $1{,}956$ posts), the loop of Section 5 ran per-post and sequential at standard strength (Appendix D) on DeepSeek-V4-Flash, and DeepSeek-V4-Pro, of the same family, attacked each persona's first 20 posts. SeqBench-6seed's $1{,}003$ leak labels give 609 targets, 60 with no labelled cue in that view. Like all labels on SeqBench-1k and SeqBench-20, these were assigned by a detector reading one post at a time, so no label marks a conjunctive route (Section 3). Both scopes see every labelled cue, so Proposition 1 predicts nothing about them, and a difference here cannot test it in either direction.

Edits and reach-back. Sequential edits 84.2% of posts against 67.8% (pooled): 406 posts only it edits against 86 only per-post edits (McNemar $\chi^2 = 206.8$; persona sign flip $p \leq 5.0 \times 10^{-5}$). Of the 282 edited posts that step 4 reaches back to after their release (Section 5), 60 are among the 406, and 44 of those were never touched in their own step: only a later post sent them to the rewriter, as p_{21} sends p_{14} in Figure 6. These counts show the mechanism at work; on labels that mark no conjunctive route they cannot show that a reach-back closed one.

Leakage. Strict value recall, the primary score, on the 375 targets the strict matcher admits falls from 0.347 raw to 0.049 per-post and 0.056 sequential (Table 7); per-post minus sequential is -0.007 (95% CI ± 0.016, $p = 0.30$, minimum detectable effect 0.022). Type F1 falls from 0.625 to 0.371 (per-post) and 0.365 (sequential), below raw on $6/6$ and $6/6$ seeds ($p = 1.9 \times 10^{-6}$ and 5.2×10^{-5}; Table 1). Between the scopes the paired t is $+0.46$ ($p = 0.66$), and the 90% interval of the difference, $[-0.019, +0.030]$, lies within ± 0.031, the narrowest margin on a 0.001 grid that two one-sided tests accept at 5%. On the primary score and in type F1, reaching the history thus costs no significant leakage on these targets; the generous-matcher readings below differ.

Utility. On SeqBench-6seed, where sequential edits more posts, per-post keeps more utility over all posts on every tier (per-post minus sequential $+0.023$, $+0.007$ and $+0.035$; $p = 8.5 \times 10^{-4}$, 0.008 and 7.3×10^{-4}; Table 1). Over edited posts alone surface fidelity does not differ (-0.005, $p =$

Table 1: **Per-post against sequential on SeqBench-6seed** (DeepSeek-V4-Pro attacker): type leakage over all 609 targets and utility, seed mean $\pm$ 95% CI (t, df 5).

	type leakage ↓			utility ↑		
arm	**prec.**	**recall**	**F1**	**surface**	**semantic**	**factual**
raw	0.635 ± 0.069	0.622 ± 0.036	0.625 ± 0.021	1	1	1
per-post	0.490 ± 0.040	0.299 ± 0.029	0.371 ± 0.031	0.887 ± 0.013	0.914 ± 0.007	0.956 ± 0.007
sequential	0.499 ± 0.079	0.291 ± 0.045	0.365 ± 0.047	0.864 ± 0.011	0.907 ± 0.005	0.921 ± 0.008
per-post − sequential, all posts				$+0.023$	$+0.007$	$+0.035$
p (seeds per-post ahead)				8.5×10^{-4} $(6/6)$	0.008 $(6/6)$	7.3×10^{-4} $(6/6)$
per-post − sequential, ***edited posts only***				-0.005	$\mathbf{-0.016}$	$+0.029$
p (seeds per-post ahead)				0.30 $(2/6)$	0.003 $(0/6)$	0.003 $(6/6)$

0.30), semantic fidelity favours sequential (-0.016, $p = 0.003$) and the factual tier favours per-post ($+0.029$, $p = 0.003$), so sequential's edits remove more unflagged content. On SeqBench-1k/150, a 150-persona sample of SeqBench-1k with 858 targets, a separate minimal-edit script anonymised both scopes in a four-round adversarial loop under each of four attacker seats (Appendix I). There sequential edits fewer posts and keeps more of the loop's all-posts fidelity in every seat (sequential minus per-post $+0.058$, paired t over the four seats on one sample, $p = 0.002$; Table 10; Figure 11), also over each arm's edited posts alone (Table 11), while round-4 type recall does not separate the scopes (per-post minus sequential -0.022, $p = 0.32$; sequential nominally higher in 3 seats of four). These runs partly reflect the detector's vocabulary loss (reproducibility statement).

Other readings. Several readings favour per-post. On SeqBench-6seed itself, under the generous matcher, at uncorrected p, sequential leaks more in value F1 (-0.041, $p = 0.031$) and value precision ($p = 0.003$; Table 9), and a value-aware matcher (Appendix G) puts per-post minus sequential at -0.030 in value-aware recall ($p = 0.085$, per-post lower on 5 of six seeds). The label-derived difference-in-differences in type recall there, per-post minus sequential on multi-cue minus on single-cue targets, does not separate the scopes (-0.076, $p = 0.33$; sequential nominally higher on multi-cue targets; Figure 10). SeqBench-1k/150 and SeqBench-20 differ from SeqBench-6seed in anonymiser implementation and attacker protocol (Table 8), so their readings are not replications. On SeqBench-1k/150, with DeepSeek-V4-Pro attacking once over the 740 targets with a cue in view, per-post concedes fewer by type recall (111 leak only under sequential, 77 only under per-post; persona permutation $p = 0.026$; single-cue $p = 0.024$), while strict value recall, the primary score, on the 476 of those targets the strict matcher scores does not separate the scopes ($p = 0.67$; Appendix I). On 16 personas of SeqBench-20 a cross-family attacker, Claude Sonnet, has strict value recall 0.315 per-post against 0.259 sequential, not a significant difference ($p = 0.25$, crediting the adjacent education level; $p = 0.50$ with SeqBench's pinned vocabulary).

7 Discussion and limitations

Discussion. On targets labelled one post at a time, which both scopes see, reaching the history costs no significant leakage on the primary score, a null that cannot test Proposition 1 (Section 3). Over all posts sequential keeps less utility on SeqBench-6seed, where it edits more, and more in every seat of the loop, where it edits fewer. Per edited post its semantic fidelity is higher on both corpora and its factual tier lower on SeqBench-6seed.

Limitations. The propositions assume a reader that infers from phrases, which leaves out stylometric authorship linkage; they and the strict matcher treat a value as recovered or not, so a release that narrows it to a few candidates counts as no leak (Appendix B). Reach-back repairs a leak only against a reader who reads the history after the rewrite; a reader who saved the release before the rewrite keeps the leak (Appendix B). Once an attribute is actionable, a released post that a later step first names as its evidence is not rewritten. The SeqBench-6seed attacker shares the anonymiser's model family and may share its blind spots. Span is a planner parameter and SeqBench's designed supports span a median of 7 posts, so memory over longer stretches stays untested, and no result isolates cross-platform structure. The certifier probes only labelled posts, so an unlabelled post that gives the value away would make a certified cumulative target mixed without the certificate showing it (Appendix F). We ran no human study of the posts' realism, the labels or the certifier, because the corpora hold too many posts (Table 4) to trace every leak back through every post by hand. Posting hours carry no time zone, so SeqBench cannot test whether they narrow a location (Appendix D).

Conclusion. A defence that edits only what it detects must check each release against what is already out, and revise released posts when it finds a leak only after the posts that carry it are out. Benchmarks should label which posts jointly carry each value and report leakage per class with each target's span, which bounds the history a defence must see when wide routes carry their own evidence (Proposition 3); SeqBench supplies designed supports with their spans, and a certifier to admit its targets.

AI USE STATEMENT

We used generative AI tools extensively and disclose broadly. AI coding assistants (Claude, run in multi-agent workflows under our direction) wrote most of the code, including the code to **implement methods** and to **clean and reformat data**, the scoring and analysis scripts and the figures, and restructured and edited the text of this paper. We used `google/gemma-4-31B-it` to **generate synthetic data sets**: it wrote every post of SeqBench, whose leak labels come from the generator's seeded planner, and the SeqBench-20, SeqBench-6seed and SeqBench-1k corpora are model-written too. DeepSeek-V4-Flash (anonymiser and one attacker seat), DeepSeek-V4-Pro, Claude Sonnet, Qwen3-32B and Gemma-4-31B (attackers) are objects of measurement (Appendix N). **Assisting with translation** is not applicable.

We reviewed the AI-assisted work, much of it by script: one script regenerates every evaluation number from saved outputs, byte for byte, and separately written scorers re-derived the corpus statistics from the released files. Re-derivation of this kind found the detector failure of the reproducibility statement. Citations were checked against bibliographic records. We take responsibility for the final content of this work, including all text, claims and artefacts produced with the aid of generative AI.

ETHICS STATEMENT

The corpus describes no real person, which is why exact per-post leak labels exist at all (Section 4). This work involved no human subjects and collected no personal data. The work is dual-use, but prior work documents the capability it measures, attribute inference, in commercially available models (Staab et al., 2024). Lermen et al. (2026) document LLM deanonymisation. We add labelled targets and a check on their class. Names matching public figures are renamed in the release, a disclosed erratum that changes no label, value or offset; names shared with less prominent people remain as common-name coincidences. Both release gates pass: name collisions, after that erratum, and memorisation, on a 100-post web-search sample (Appendix K gives each gate and the intended and prohibited uses).

REPRODUCIBILITY STATEMENT

A corpus's structure (personas, threads, timeline and leak labels) regenerates from the seed, the flags and the build revision; post text written by the served model need not regenerate byte for byte, so the released files, fixed by their sha256, are the reference. Every reported result was computed on the corpora as generated, which carry the public-figure names the release renames (Appendix L). Every anonymiser and attacker output behind a reported number comes from a model called between June and August 2026, through a provider's API or, for the Qwen3-32B and Gemma-4-31B seats, a self-hosted server that recorded no revision (Appendix N). A served model can change under a fixed name, so the saved outputs are the reproducible object. We release what each evaluation number is computed from (in `figs/`): every attacker answer's attribute, value and quote with its validity flag and the sha256 of the text the attacker saw, the edit sets, the token counts, the four-seat loop's round histories, the statistics and support spans of SeqBench-20, SeqBench-6seed and SeqBench-1k, and the matcher version behind Table 6. A script there rebuilds every evaluation table and number from them with no model call. Those corpora and their anonymised texts are not in the release, so these numbers re-tabulate but cannot be re-scored. We release SeqBench's corpora under CC BY 4.0 and the code under Apache 2.0 or MIT. The code and six of the eleven corpora are in the supplementary material, and the other five follow at camera-ready. Appendices J and L give the details.

One failure we hit is silent and not specific to this work. Our detector filtered inferences against a fixed attribute vocabulary, but the prompt never named that vocabulary. The served model's phrasing drifted to prose labels, the filter discarded most of what the detector found, and downstream a discarded attribute looks the same as one never found. The defence reported success while its per-post and sequential arms edited 5.1% and 21.9% of posts, against 67.8% and 84.1% once fixed (seed means), and no error appeared anywhere. Assert detection recall against the filter instead of assuming it.

REFERENCES

Mário S. Alvim, Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Geoffrey Smith. Measuring information leakage using generalized gain functions. In *2012 IEEE 25th Computer Security Foundations Symposium (CSF)*, pp. 265–279. IEEE, 2012. doi: 10.1109/CSF.2012.26.

Anonymous. Before the attacker acts: Floors and ceilings for evaluating LLM deanonymisation, 2026. Companion submission under review.

Article 29 Data Protection Working Party. Opinion 05/2014 on anonymisation techniques. `https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf`, 2014. 0829/14/EN, WP216. Adopted on 10 April 2014.

Michael Brennan, Sadia Afroz, and Rachel Greenstadt. Adversarial stylometry: Circumventing authorship recognition to preserve privacy and anonymity. *ACM Transactions on Information and System Security (TISSEC)*, 15(3):1–22, 2012.

Yang Cao, Masatoshi Yoshikawa, Yonghui Xiao, and Li Xiong. Quantifying differential privacy under temporal correlations. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pp. 821–832. IEEE, 2017.

Xiaolin Chen, Xuemeng Song, Siwei Cui, Tian Gan, Zhiyong Cheng, and Liqiang Nie. User identity linkage across social media via attentive time-aware user modeling. *IEEE Transactions on Multimedia*, 23:3957–3967, 2021. doi: 10.1109/TMM.2020.3034540.

Yves-Alexandre de Montjoye, César A. Hidalgo, Michel Verleysen, and Vincent D. Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific Reports*, 3:1376, 2013.

Dorothy E. Denning, Peter J. Denning, and Mayer D. Schwartz. The tracker: A threat to statistical database security. *ACM Transactions on Database Systems*, 4(1):76–96, 1979. doi: 10.1145/320064.320069.

Yao Dou, Isadora Krsek, Tarek Naous, Anubha Kabra, Sauvik Das, Alan Ritter, and Wei Xu. Reducing privacy risks in online self-disclosures with language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference (TCC)*, volume 3876 of *Lecture Notes in Computer Science*, pp. 265–284. Springer, 2006.

Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. Differential privacy under continual observation. In *Proceedings of the Forty-second ACM Symposium on Theory of Computing (STOC)*, pp. 715–724. ACM, 2010. doi: 10.1145/1806689.1806787.

European Parliament and Council of the European Union. Regulation (eu) 2016/679 of the european parliament and of the council (general data protection regulation). Official Journal of the European Union L 119, pp. 1–88, 2016. Article 4(1) and Recital 26 (personal data); Article 4(5) (pseudonymisation).

Xi Fang, Weijie Xu, Fiona Anting Tan, Jiani Zhang, Ziqing Hu, Yanjun Qi, Scott Nickleach, Diego Socolinsky, Srinivasan Sengamedu, and Christos Faloutsos. Large language models (LLMs) on tabular data: Prediction, generation, and understanding – a survey. *Transactions on Machine Learning Research (TMLR)*, 2024.

Ahmed Frikha, Nassim Walha, Krishna Kanth Nakka, Ricardo Mendes, Xue Jiang, and Xuebing Zhou. IncogniText: Privacy-enhancing conditional text anonymization via LLM-based private attribute randomization. In *NeurIPS 2024 Workshop on Safe Generative AI*, 2024. arXiv:2407.02956.

Srivatsava Ranjit Ganta, Shiva Prasad Kasiviswanathan, and Adam Smith. Composition attacks and auxiliary information in data privacy. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 265–273. ACM, 2008. doi: 10.1145/1401890.1401926.

Sajjad Ghiasvand and Nader Sehatbakhsh. GRASP: Reinforcing language model anonymizers with group relative policy optimization. *arXiv preprint arXiv:2608.06526*, 2026.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

Timour Igamberdiev and Ivan Habernal. DP-BART for privatized text rewriting under local differential privacy. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13914–13934. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-acl.874.

Krishnaram Kenthapadi, Nina Mishra, and Kobbi Nissim. Simulatable auditing. In *Proceedings of the Twenty-fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, pp. 118–127. ACM, 2005. doi: 10.1145/1065167.1065183.

Kyuyoung Kim, Hyunjun Jeon, and Jinwoo Shin. Self-refining language model anonymizers via adversarial distillation. *arXiv preprint arXiv:2506.01420*, 2025.

Siwon Kim, Sangdoo Yun, Hwaran Lee, Martin Gubri, Sungroh Yoon, and Seong Joon Oh. ProPILE: Probing privacy leakage in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Moshe Koppel, Jonathan Schler, and Shlomo Argamon. Computational methods in authorship attribution. *Journal of the American Society for Information Science and Technology*, 60(1):9–26, 2009.

Nataša Krčo, Zexi Yao, Matthieu Meeus, and Yves-Alexandre de Montjoye. RAT-Bench: A comprehensive benchmark for text anonymization. *arXiv preprint arXiv:2602.12806*, 2026.

Diane Lambert. Measures of disclosure risk and harm. *Journal of Official Statistics*, 9(2):313–331, 1993.

Simon Lermen, Daniel Paleka, Joshua Swanson, Michael Aerni, Nicholas Carlini, and Florian Tramèr. Large-scale online deanonymization with LLMs. In *ICLR 2026 Workshop on Agents in the Wild: Safety, Security, and Beyond*, 2026. URL https://openreview.net/forum?id=cRos9u8shf. arXiv:2602.16800.

Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd International Conference on Data Engineering (ICDE)*, pp. 106–115. IEEE, 2007.

Ziwen Li, Jianing Wen, and Tianshi Li. LLM anonymization against agentic re-identification. *arXiv preprint arXiv:2605.30848*, 2026.

Pierre Lison, Ildikó Pilán, David Sánchez, Montserrat Batet, and Lilja Øvrelid. Anonymisation models for text data: State of the art, challenges and future directions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pp. 4188–4203, 2021.

Gabriel Loiseau, Damien Sileo, Damien Riquet, Maxime Meyer, and Marc Tommasi. Tau-Eval: A unified evaluation framework for useful and private text anonymization. In *Proceedings of EMNLP 2025: System Demonstrations*, 2025.

Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3–es, 2007.

Niloofar Mireshghallah, Hyunwoo Kim, Xuhui Zhou, Yulia Tsvetkov, Maarten Sap, Reza Shokri, and Yejin Choi. Can LLMs keep a secret? testing privacy implications of language models via contextual integrity theory. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

MiroFish. MiroFish: An AI prediction engine for social simulation and forecasting. `https://mirofish.co.uk/`, 2025. Built on the OASIS engine; version 0.1.0.

Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (S&P)*, pp. 111–125. IEEE, 2008. doi: 10.1109/SP.2008.33.

Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. In *2009 IEEE Symposium on Security and Privacy (S&P)*, pp. 173–187. IEEE, 2009.

Arvind Narayanan, Hristo Paskov, Neil Zhenqiang Gong, John Bethencourt, Emil Stefanov, Eui Chul Richard Shin, and Dawn Song. On the feasibility of internet-scale author identification. In *IEEE Symposium on Security and Privacy*, pp. 300–314, 2012.

Helen Nissenbaum. Privacy as contextual integrity. *Washington Law Review*, 79(1):119–157, 2004.

Myeong Seok Oh, Dong-Yun Kim, Hanseok Oh, Chaean Kang, Joeun Kang, Xiaonan Wang, Hyunjung Park, Young Cheol Jung, and Hansaem Kim. Subject-level inference for realistic text anonymization evaluation. *arXiv preprint arXiv:2604.21211*, 2026.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST)*, 2023.

Zifan Peng, Yini Huang, Aiwen Lu, Qiming Ye, Peixian Zhang, Jingyi Zheng, Yule Liu, Xuechao Wang, Xinlei He, and Jiaheng Wei. What your posts reveal: A benchmark and agentic framework for user-level privacy leakage on social media. *arXiv preprint arXiv:2606.06784*, 2026.

Ildikó Pilán, Pierre Lison, Lilja Øvrelid, Anthi Papadopoulou, David Sánchez, and Montserrat Batet. The text anonymization benchmark (TAB): A dedicated corpus and evaluation framework for text anonymization. *Computational Linguistics*, 48(4):1053–1101, 2022.

David E. Pozen. The mosaic theory, national security, and the Freedom of Information Act. *The Yale Law Journal*, 115(3):628–679, 2005.

Francisco Rangel, Paolo Rosso, Moshe Koppel, Efstathios Stamatatos, and Giacomo Inches. Overview of the author profiling task at PAN 2013. In *CLEF 2013 Working Notes*, volume 1179 of *CEUR Workshop Proceedings*, 2013.

Luc Rocher, Julien M. Hendrickx, and Yves-Alexandre de Montjoye. Estimating the success of re-identifications in incomplete datasets using generative models. *Nature Communications*, 10: 3069, 2019.

Chenyang Shao, Tianxing Li, Chenhao Pu, Fengli Xu, and Yong Li. AgentStealth: Reinforcing large language model for anonymizing user-generated text. *arXiv preprint arXiv:2506.22508*, 2025.

Yijia Shao, Tianshi Li, Weiyan Shi, Yanchen Liu, and Diyi Yang. PrivacyLens: Evaluating privacy norm awareness of language models in action. In *Thirty-eighth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS D&B)*, 2024.

Geoffrey Smith. On the foundations of quantitative information flow. In *Foundations of Software Science and Computational Structures (FoSSaCS)*, Lecture Notes in Computer Science, pp. 288–302. Springer, 2009. doi: 10.1007/978-3-642-00596-1_21.

Robin Staab, Mark Vero, Mislav Balunović, and Martin Vechev. Beyond memorization: Violating privacy via inference with large language models. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Robin Staab, Mark Vero, Mislav Balunović, and Martin Vechev. Large language models are advanced anonymizers. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. arXiv:2402.13846.

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

Efstathios Stamatatos et al. Overview of the author identification task at PAN 2015. In *CLEF 2015 Working Notes*, volume 1391 of *CEUR Workshop Proceedings*, 2015.

Amber Stubbs and Özlem Uzuner. Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/UTHealth corpus. *Journal of Biomedical Informatics*, 58:S20–S29, 2015.

Latanya Sweeney. Simple demographics often identify people uniquely. Technical Report Data Privacy Working Paper 3, Carnegie Mellon University, Pittsburgh, PA, 2000.

Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

Saiteja Utpala, Sara Hooker, and Pin-Yu Chen. Locally differentially private document generation using zero shot prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 8442–8457. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.566.

Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, et al. DecodingTrust: A comprehensive assessment of trustworthiness in GPT models. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS D&B)*, 2023.

Ke Wang and Benjamin C. M. Fung. Anonymizing sequential releases. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 414–423. ACM, 2006. doi: 10.1145/1150402.1150449.

Xiaokui Xiao and Yufei Tao. m-invariance: Towards privacy preserving re-publication of dynamic datasets. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, pp. 689–700. ACM, 2007. doi: 10.1145/1247480.1247556.

Tianyu Yang, Xiaodan Zhu, and Iryna Gurevych. Robust utility-preserving text anonymization based on large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL)*, pp. 28922–28941. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1404.

Ziyi Yang, Zaibin Zhang, Zirui Zheng, Yuxian Jiang, Ziyue Gan, Zhiyu Wang, Zijian Ling, Jinsong Chen, Martz Ma, Bowen Dong, Prateek Gupta, Shuyue Hu, Zhenfei Yin, Guohao Li, Xu Jia, Lijun Wang, Bernard Ghanem, Huchuan Lu, Chaochao Lu, Wanli Ouyang, Yu Qiao, Philip Torr, and Jing Shao. OASIS: Open agent social interaction simulations with one million agents. *arXiv preprint arXiv:2411.11581*, 2024.

Hanna Yukhymenko, Robin Staab, Mark Vero, and Martin Vechev. A synthetic dataset for personal attribute inference. In *Thirty-eighth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS D&B)*, 2024. URL `https://openreview.net/forum?id=1nqfIQIQBf`.

A Extended related work

Three notions share one name. De-identification removes predefined identifier types, such as the 18 HIPAA categories of protected health information (Stubbs & Uzuner, 2015). Its benchmarks score precision, recall and F_1 against annotated identifiers, at entity and token level (Stubbs & Uzuner, 2015; Lison et al., 2021). Anonymisation in the data-protection sense requires that the data subject is no longer identifiable (European Parliament and Council of the European Union, 2016). The test counts all the means that a controller or another person is reasonably likely to use. Statistical

disclosure control separates identity disclosure, a link between a person and a record, from attribute disclosure, new knowledge about the person (Lambert, 1993; Li et al., 2007). Classical risk measures score re-identification, and they often omit attribute disclosure without identification (Lambert, 1993). k-anonymity protects against identity disclosure (Sweeney, 2002), but a k-anonymous table can still disclose a sensitive attribute (Machanavajjhala et al., 2007; Li et al., 2007). ℓ-diversity and t-closeness add protection against this attribute disclosure (Machanavajjhala et al., 2007; Li et al., 2007). Differential privacy bounds how much one person's record changes the distribution of the output (Dwork et al., 2006). It does not promise that a person's secrets stay secret (Dwork & Roth, 2014, Section 2.3.2). A 2014 opinion under the former Data Protection Directive tests anonymisation techniques against three risks: singling out, linkability and inference (Article 29 Data Protection Working Party, 2014). The opinion defines inference as the deduction, with significant probability, of an attribute value from the values of other attributes. Anonymisation in the data-protection sense thus names attribute disclosure as a risk. Its usual measures, such as k or a re-identification risk, do not bound it.

How text benchmarks use the term. TAB defines text anonymisation by identity disclosure and presents itself as a step beyond de-identification (Pilán et al., 2022). Expert annotators decide for each entity whether to mask it to conceal the person's identity. TAB scores entity recall against these masking decisions, and its metrics cover identifiers only. It annotates confidential attributes and calls the obfuscation of author attributes a different task from text anonymisation. Staab et al. (2025) use the term more broadly. They state the goal of text anonymisation as the removal of all information that could lead to the inference of pre-specified author attributes. They score it by the accuracy of an LLM adversary on the attribute values rather than by a re-identification risk. They also note that the literature sometimes calls entity redaction de-identification rather than anonymisation. RAT-Bench joins the two views. It converts the attributes that an attacker infers correctly into a re-identification risk for the U.S. population (Krčo et al., 2026). Usage also varies. Stubbs & Uzuner (2015) give anonymisation as another name for de-identification, and Lison et al. (2021) note that some authors use the two terms interchangeably.

Which notion SeqBench measures. Definition 1 asks only that the reader infer no true value of a sensitive attribute from any prefix of the release. This objective relates to the inference risk of the 2014 opinion but differs from it. It checks a fixed attribute set against one reader, it has no identifiability condition, and it says nothing about singling out or linkability. The threat model even grants the reader the linked cross-platform history. An anonymiser that solves the task can still release text that lets a reader identify the author. A pass on SeqBench therefore does not show that a release is anonymous in the data-protection sense. Our term *sensitive attribute* corresponds to what Staab et al. (2025) call a personal attribute. In statistical disclosure control, age and location are key variables, useful for identification but not usually sensitive (Lambert, 1993). There, a sensitive attribute is one such as a disease or a debt (Lambert, 1993; Li et al., 2007). Staab et al. (2025) themselves call attributes such as location, gender and socioeconomic status quasi-identifying.

Terms. Tables 2 and 3 fix the terms the paper uses for arms, groups, classes and scores. *Sequential* names one arm; Definition 1's sequential anonymiser is the general family of maps, of which both arms are members; its whole-profile setting is a single map $\mathcal{A}_T$ applied once to the finished history.

Adjacent privacy benchmarks measure a different axis. DecodingTrust (Wang et al., 2023) scores privacy as one dimension of a trustworthiness suite, ProPILE (Kim et al., 2023) probes whether a model's own weights link a person's PII, ConfAIde (Mireshghallah et al., 2024) shows that models misjudge when a disclosure is appropriate under contextual integrity (Nissenbaum, 2004), and PrivacyLens (Shao et al., 2024) scores whether an agent violates a contextual norm when it acts. All four are memorisation- or norm-side, where SeqBench measures content-side inference from what a person wrote.

Aggregation before language models. A handful of demographic fields often single a person out (Sweeney, 2000); sparse behavioural signatures recover named individuals from de-identified releases (Narayanan & Shmatikov, 2008; 2009); later work quantified how few observations that takes (de Montjoye et al., 2013; Rocher et al., 2019), and privacy loss compounds through temporal correlation when data is released over time (Cao et al., 2017). The syntactic family (Sweeney, 2002;

Table 2: **Glossary: the arms and the loop.** Each term has this meaning throughout the paper.

Term	Meaning
per-post	the loop of Staab et al. (2025) run on one post at a time: the profile is cleared before every post, so each post is anonymised from itself alone (Appendix D)
history-aware	any scope whose detector reads the author's history through the profile. An adjective, not an arm
sequential (idealised)	the arm that reads the history and may rewrite already-released posts of a detail's support set, each from its latest release (Appendix D)
whole-profile	the setting of Staab et al. (2025) on a finished profile (Definition 1): detection over the complete raw timeline, then one rewrite of each supporting post; it needs the future
standard strength	SeqBench-6seed's strength: one detector pass per post, no refinement, no paraphrase, no k-anonymity rewrite
tier, gate, actionable	an inference's sensitivity tier (low, medium, high) and confidence; each attribute's gate is a minimum tier and confidence, and a detail is actionable while it clears its gate; each time it newly clears it, the loop rewrites its support set
detector, rewriter, verifier	the loop's three roles, all one checkpoint (Appendix N)
profile P_t	the running set of inference details (Definition 4); the detector reads its compact summary, not the released text
edit set	the posts an arm rewrites, whose released text differs from the original

Table 3: **Glossary: groups, classes and scores.**

Term	Meaning
designed group (support)	a non-decoy (persona, `joint_id`) shard group with at least two released posts on a value-scorable attribute (Table 4)
decoy	a group whose support never closed, kept as a hard negative with no answer
target	a (persona, attribute) pair; its class comes from all of its groups
value-scorable	the persona record holds a true value for the attribute
pool	the 14 attributes the pinned strict constant-guess table keeps (Table 5); each has a best constant guess below 50%
guessable	a target on an excluded attribute, or one a seat's no-text prior guess recovers; undetermined when the prior is missing
conjunctive route	a minimal support of two or more posts (Definition 2); the certifier finds one where a support recovers the value and none of its posts does alone. Certified cumulative and mixed targets have one
certified (cumulative)	a non-guessable target of the class certified cumulative under the agree rule and the null rule. Mixed is not certified
agree rule	a target is certified only if every reader seat that audited it certifies it; a missing verdict leaves it undetermined
null rule, forced rule	a refusal is a miss; an uncredited answer is re-asked once for a forced guess (Appendix F)
undetermined, disputed, unaudited	a needed verdict is missing (never a miss); the seats class the target differently; no seat audited it
span	of a designed support: last minus first position in the author's chronological timeline, all platforms pooled
window (width k)	$k+1$ consecutive posts (Proposition 3); what a detector or an attacker reads is its *view*
decidable depth	the timeline position of the post that completes a support
r_0	the hit rate of an attribute's best single constant answer under the strict matcher
chance-corrected recall	$(H - \sum r_0)/(n - \sum r_0)$ over n pool targets with H strict hits

Machanavajjhala et al., 2007; Li et al., 2007) and differential privacy (Dwork et al., 2006; Dwork & Roth, 2014) were built for structured records; the next two paragraphs give their sequential-release forms and the differentially private rewriting of text. Synthetic generation (Fang et al., 2024) and agent societies (Park et al., 2023; Yang et al., 2024; MiroFish, 2025) supply the closest machinery to our generator but aim at prediction rather than labelled leakage, and adversarial stylometry (Sta-

matatos, 2009; Koppel et al., 2009; Brennan et al., 2012; Stamatatos et al., 2015; Narayanan et al., 2012) contributes a channel that survives entity removal and that a generator switch weakens on its own (Appendix J).

Composition and sequential release. Releases that each pass a privacy test can disclose jointly. A tracker combines permitted statistical queries to answer a refused query about one person (Denning et al., 1979), and a composition attack intersects independently anonymised releases of overlapping populations (Ganta et al., 2008). Sequential-release anonymisation prepares each new release so that joining it with the earlier ones does not breach the privacy requirement (Wang & Fung, 2006), and m-invariance keeps the set of sensitive values in an individual's group fixed across re-publications of a changing table (Xiao & Tao, 2007). Query auditing decides whether to answer each new query given the earlier ones; simulatable auditing decides from the earlier queries and answers alone, because a denial that depends on the data can itself leak (Kenthapadi et al., 2005). Differential privacy composes by construction (Dwork & Roth, 2014), and under continual observation it keeps a statistic private while the statistic is re-released at every step of a stream (Dwork et al., 2010). Legal scholarship on national-security secrecy calls the effect the mosaic theory: items that are harmless alone can reveal a secret together (Pozen, 2005). The cumulative class of Definition 2 is this effect in free text, with two differences. The released unit is a post rather than a record or a query answer, so the attributes are not columns and a defence must first find the evidence it would edit. And no existing resource labels which posts jointly carry a value; the certifier of Section 4.2 is SeqBench's rule for that label.

Differentially private rewriting. DP-Prompt paraphrases a document with a language model, and the sampling temperature over clipped logits sets its local privacy budget (Utpala et al., 2023); DP-BART adds calibrated noise to a BART encoder's representation of the document before decoding it (Igamberdiev & Habernal, 2023). Both give local differential privacy to one document. A person who writes T posts, each rewritten at budget ε, is protected at level $T\varepsilon$ under basic composition (Dwork & Roth, 2014), and the adjacency of whole documents already forces a high level of noise for a single document (Igamberdiev & Habernal, 2023). The guarantee bounds how much the release changes what a reader believes; it does not name an attribute that must stay unrecoverable, which is what Definition 1 asks.

Attribute inference and leakage measures. Author profiling inferred age and gender from text as a shared task before large language models (Rangel et al., 2013); Staab et al. (2024) showed that a pretrained LLM infers a wider set of attributes from ordinary posts without task-specific training. Quantitative information flow compares an adversary's chance of guessing a secret in one try before and after it sees a system's output (Smith, 2009), and gain functions generalise what counts as a correct guess (Alvim et al., 2012). Our chance-corrected recall makes the same comparison on the attacker's hits: the pinned best constant guess plays the prior, and the score is the attacker's gain over it as a share of the gain still possible (Appendix G). The strict and generous matchers are two gain functions in this sense.

Each resource counts posts in its own unit: documents for TAB and SPIA, comments for SynthPAI, records for RAT-Bench, tweets and captions for TWIN. The *attr.* column counts each resource's own label categories. TAB counts entity types, RAT-Bench counts direct and indirect identifiers, and SPIA counts PII categories. SynthPAI, SopriBench and SeqBench count inferable personal attributes.

B Boundaries of the model

The scopes of Definition 3 differ in what revision can repair.

Proposition 2 (What revision buys)**.** *(a) $\mathcal{F}_{\mathrm{onl}} \subseteq \mathcal{F}_{\mathrm{real}} \subseteq \mathcal{F}_{\mathrm{ide}}$, and both inclusions are strict when $T \geq 2$; neither fact involves $\mathcal{D}$. (b) Let $\mathcal{D}$ be evidence-based, and let an anonymiser leak $z \in Z$ at step s: $z \in \mathcal{D}(Y_{1:s}^{(s)})$. If it is realistic (online ones included), $z \in \mathcal{D}(Y_{1:t}^{(t)})$ at every $t \geq s$, whatever it releases later. For every t with $s < t \leq T$, some idealised anonymiser agrees with it before step t and has $\mathcal{D}(Y_{1:t}^{(t)}) \cap Z = \emptyset$.*

Proposition 2 concerns repairing a leak, not feasibility. With no utility constraint, releasing the empty text for every post solves Definition 1 against an evidence-based $\mathcal{D}$ in every scope, the per-post one included, because an empty text contains no phrase, phrases being non-empty strings (Appendix C). Such an anonymiser generally violates hypothesis (ii) of Proposition 1, so that proposition does not apply to it. Reach-back buys the power to repair a leak after it has been released, such as one the detector found only at a later step.

The four boundaries below are properties of the model, so no implementation choice in the anonymiser removes them.

A snapshot adversary. $Y_{1:t}$ denotes the archive as it currently stands. The idealised scope rewrites earlier posts, so the released history depends on when a reader reads it. Retroactive re-generalisation protects against a reader who fetches the history after an edit, not against one who scraped an earlier state. Released text cannot be unpublished, and the objective of Definition 1 does not claim otherwise. A model of logged adversaries would index the objective by observation time and quantify over observation schedules. The realistic and online scopes of Definition 3 are the degenerate case in which every state is assumed logged.

Phrase-localised leakage only. Evidence-basedness assumes each recoverable attribute is carried by identifiable spans; the detector's quoted phrase (e_d of Definition 4) only points the whole-post rewriter at them. Distributional channels fall outside: stylometric authorship linkage is a property of the whole text rather than of any excisable phrase, so Propositions 1 and 3 are silent about it. The companion submission (Anonymous, 2026) measures it.

A deterministic adversary. $\mathcal{D}$ is set-valued, so it either recovers an attribute or does not. Confidence-weighted recovery and calibration are outside the model. So is the probability mass an adversary places on the true value, and so is any partial leakage of a single attribute. The evaluation's primary score is discrete in the same way: under the strict matcher a target is recovered or not (Appendix G), with no credit for probability placed on the truth.

No utility constraint. Nothing in Section 3 bounds the fidelity of Y. As the empty-text anonymiser above shows, the propositions cannot by themselves distinguish a minimal edit from a destructive one. We therefore report text fidelity alongside every leakage number (Section 6).

C Window width and proofs

Proposition 3 (Window width)**.** *Let $\mathcal{D}$ be evidence-based and consider the width-k sliding-window anonymiser, which at step t reads the posts $W_t = \{t-k, \dots, t\} \cap \{1, \dots, t\}$ and may change only those. Assume it meets hypotheses (i)–(iii) of Proposition 1 at every step with X_{W_t} as its view: it detects only pairs in $\mathcal{D}(X_{W_t})$, changes text only inside phrases supporting pairs it detects there, and no phrase it changes overlaps a phrase supporting a pair of Z it does not detect there. It* neutralises *z at t if $z \notin \mathcal{D}(Y_{1:t})$, where $Y_{1:t}$ is its release at t. Let $z \in Z$ be recoverable at t.*

(a) If every $\sigma \in \Sigma_t(z)$ has $\max\sigma - \min\sigma > k$, then $z \in \mathcal{D}(Y_{1:t})$. So the window neutralises z only if some minimal support of z fits inside $k+1$ consecutive posts, and a width-0 (per-post) anonymiser neutralises no cumulative z.

(b) Assume further that at every step where it detects z, each phrase it changes that overlaps a phrase supporting z belongs to a supporting collection of z within its view. If some $\sigma \in \Sigma_t^{\mathrm{cnj}}(z)$ with $\max\sigma - \min\sigma > k$ has a supporting collection of z within its posts no phrase of which overlaps a phrase of any supporting collection of z within $k+1$ consecutive posts, then $z \in \mathcal{D}(Y_{1:t})$. So if every conjunctive route of z wider than k has such a collection, the window neutralises z only if $\mathrm{span}_t(z) \le k$.

Every designed support in the ground truth, one (persona, `joint_id`) group, records the posts that carry it. Mapping them to positions in the author's own chronological timeline, all platforms pooled, gives the support's *span*, its last minus its first position, and its *decidable depth*, the position of the post that completes it (Figure 7). A support can be edited as a whole only once all its posts have ap-

peared, so the share of supports that a history of a given depth can decide grows monotonically with that depth, as the right panel plots. Regenerate them with `figs/make_window_curves.py`.

Conventions for the proofs. Evidence-basedness is read in this form. A *phrase* is a non-empty string, and a set of texts *contains* a finite set C of phrases if each phrase of C is a substring of one of the texts. $\mathcal{D}$ is applied through this containment to any texts, raw or released, so an edit can also add evidence. A set C of phrases is *sufficient* for a pair $w \in \mathcal{Z}$ if $\mathcal{D}$ recovers w from every set of texts that contains C, and a set C' *reduces* C if each phrase of C' is a substring of a phrase of C. The *supporting collections* of w are the sufficient sets C that every sufficient set reducing C contains: no phrase of C can be dropped, shortened or split without losing sufficiency. This fixes one family per pair, determined by $\mathcal{D}$ alone, and Propositions 1 and 3 are read against it. The choice matters. Under the family of all sufficient sets, every phrase q would support every pair that has a sufficient set C, because $C \cup \{q\}$ is sufficient too, and hypothesis (iii) would forbid every edit while such a pair of Z goes undetected. The choice loses nothing. The sufficient sets that reduce a given sufficient set C_0 are finitely many; take one that is minimal under reduction and drop each of its phrases that is a substring of another. The result is a supporting collection that reduces C_0, so every set of texts that contains C_0 contains it. A $\mathcal{D}$ that is evidence-based for some family of collections (Section 3) is therefore evidence-based for this one: it recovers $z \in Z$ from a set of texts if and only if the set contains a supporting collection of z. Every pair it infers is carried by phrases, so it recovers nothing from empty texts, and every supporting collection is non-empty.

A *phrase of* a post is an occurrence of a phrase in it, a run of consecutive character positions. It *supports* w if its string belongs to a supporting collection of w, and two phrases of a post *overlap* if they share a character position. A supporting collection lies *within* a set of posts if each of its phrases occurs in one of them, and its phrases there are one chosen occurrence of each. Hypotheses (ii) and (iii) and the added hypotheses of Proposition 3(b) speak of phrases of the raw posts. An anonymiser changes a post only inside some of its phrases if every character it deletes or replaces lies in one of them and any text it inserts goes inside one of them. Every other character is then carried over unchanged and in order, across steps as well, so a phrase of a raw post that overlaps none of the changed phrases still occurs, as the same string, in the release. Two facts about minimal supports are used. *(M1)* If $\sigma \in \Sigma_t(z)$, then $z \notin \mathcal{D}(X_{\sigma'})$ for every $\sigma' \subsetneq \sigma$; this is the definition of inclusion-minimality. *(M2)* If $z \in \mathcal{D}(X_S)$ for some $S \subseteq \{1, \dots, t\}$, then some $\sigma \in \Sigma_t(z)$ has $\sigma \subseteq S$: S is finite, so it has an inclusion-minimal subset σ with $z \in \mathcal{D}(X_\sigma)$, and minimality refers only to the subsets of σ.

Proposition 1 (Separation).

Proof. Fix a route $\sigma \in \Sigma_t^{\mathrm{cnj}}(z)$. For each $i \in \sigma$ the singleton $\{i\}$ is a proper subset of σ, because $|\sigma| \geq 2$, so (M1) gives $z \notin \mathcal{D}(X_i)$. This step covers both classes. If z is cumulative, $z \notin \mathcal{D}(X_i)$ holds for every $i \leq t$. If z is mixed, it has a single-post route $\{j\}$, but $j \notin \sigma$, since $\{j\} \subsetneq \sigma$ with $z \in \mathcal{D}(X_j)$ would contradict (M1); so whatever $\mathcal{A}^{\mathrm{pp}}$ does to X_j leaves Y_σ untouched. By (i), $\mathcal{A}^{\mathrm{pp}}$ detects z in no X_i with $i \in \sigma$. Since $z \in \mathcal{D}(X_\sigma)$ and $\mathcal{D}$ is evidence-based, some supporting collection C of z lies within σ. Each phrase of C is a phrase of some X_i with $i \in \sigma$, and it supports z, a pair of Z that $\mathcal{A}^{\mathrm{pp}}$ does not detect in X_i. By (ii), $\mathcal{A}^{\mathrm{pp}}$ changes X_i only inside phrases supporting pairs it detects there, and by (iii) none of these overlaps a phrase of C. So each phrase of C occurs unchanged in Y_i, Y_σ contains C and $z \in \mathcal{D}(Y_\sigma)$, and evidence-monotonicity with $\sigma \subseteq \{1, \dots, t\}$ gives $z \in \mathcal{D}(Y_{1:t})$. No step uses how $\mathcal{A}^{\mathrm{pp}}$ rewrites the phrases it does change, so the conclusion holds however hard it edits the evidence it detects. □

Proposition 2 (What revision buys).

Proof. (a) An online anonymiser is realistic: its Y_t is a function of $(Y_{1:t-1}^{(t-1)}, X_t)$, and $Y_{1:t-1}^{(t-1)}$ depends on $X_{1:t-1}$ alone by induction on t, so Y_t is a function of $X_{1:t}$. A realistic anonymiser never revises a released post, so its release after step t is $Y_{1:t}^{(t)} = (Y_{1:t-1}^{(t-1)}, Y_t)$, and by induction on t this release depends on $X_{1:t}$ alone. So it is an idealised anonymiser that never revises a released post. This gives both inclusions. For strictness when $T \geq 2$, fix two distinct texts $x \neq x'$. The realistic anonymiser with $Y_1 = x$ and $Y_t = X_1$ for $t \geq 2$ is not online: holding X_2 fixed, its Y_2 changes with X_1 while Y_1 does not, and an online Y_2 is a function of (Y_1, X_2). The idealised anonymiser that, whatever

the input, releases $Y_1^{(1)} = x$ and, for $t \geq 2$, $Y_{1:t}^{(t)} = (x', x, \ldots, x)$ revises post 1 at step 2, which no realistic anonymiser does. Neither argument involves $\mathcal{D}$.

(b) For a realistic anonymiser, online ones included, $Y_{1:s}^{(s)}$ is a prefix of $Y_{1:t}^{(t)}$ for every $t \geq s$, so evidence-monotonicity gives $z \in \mathcal{D}(Y_{1:s}^{(s)}) \subseteq \mathcal{D}(Y_{1:t}^{(t)})$. For the idealised claim, fix t with $s < t \leq T$. Consider the anonymiser that uses the given maps at every step before t and releases the empty text for every post at step t and after. By (a), the given maps are idealised, so this anonymiser is idealised too, and it agrees with the given one before step t. Its release at t consists of empty texts, which contain no phrase because phrases are non-empty, so it contains no supporting collection of any pair of Z, and hence $\mathcal{D}(Y_{1:t}^{(t)}) \cap Z = \emptyset$. The witness is destructive by design: it shows what the idealised scope permits, not what an anonymiser should do (Appendix B, no utility constraint). □

Proposition 3 (Window width).

Proof. (a) Suppose $z \in \mathcal{D}(X_{W_s})$ for some step $s \leq t$. By (M2), some $\sigma \in \Sigma_t(z)$ has $\sigma \subseteq W_s$, and then $\max \sigma - \min \sigma \leq \max W_s - \min W_s \leq k$, which contradicts the assumption. So $z \notin \mathcal{D}(X_{W_s})$ at every step, and by (i) the anonymiser never detects z. By (ii) and (iii), no phrase it changes at any step overlaps a phrase supporting z. Now z is recoverable at t, so we can take some $\sigma \in \Sigma_t(z)$ and, by evidence-basedness, a supporting collection C of z within σ. No changed phrase overlaps a phrase of C, so each phrase of C occurs unchanged in the release, Y_σ contains C and $z \in \mathcal{D}(Y_\sigma) \subseteq \mathcal{D}(Y_{1:t})$. The statement about neutralising is the contrapositive. For $k = 0$, every minimal support of a cumulative z has at least two posts and hence $\max \sigma - \min \sigma \geq 1 > 0$.

(b) Let C be the collection that the hypothesis gives for σ, and let ψ be a phrase that the anonymiser changes at some step s. Suppose ψ overlaps a phrase of C, which supports z. If the anonymiser does not detect z at step s, this contradicts (iii). If it does, the added hypothesis makes ψ a phrase of a supporting collection of z within W_s. W_s lies within $k+1$ consecutive posts, so by the hypothesis on C no phrase of C overlaps ψ, again a contradiction. So no changed phrase overlaps a phrase of C, each phrase of C occurs unchanged in the release, Y_σ contains C, and $z \in \mathcal{D}(Y_\sigma) \subseteq \mathcal{D}(Y_{1:t})$. For the last sentence, if $\mathrm{span}_t(z) > k$, some $\sigma \in \Sigma_t^{\mathrm{cnj}}(z)$ has $\max \sigma - \min \sigma = \mathrm{span}_t(z) > k$. That route has such a collection, so $z \in \mathcal{D}(Y_{1:t})$. □

Neither added hypothesis of (b) can be dropped. Take $k = 1$ and a cumulative z whose only minimal supports are $\{1, 2\}$ and $\{2, 30\}$. At step 2 the window sees $\{1, 2\}$ and detects z. If one phrase of post 2 belongs to a supporting collection of each route, changing it for $\{1, 2\}$ also breaks $\{2, 30\}$. If the two routes use non-overlapping phrases of post 2, (ii) still lets the anonymiser change every phrase supporting z in post 2, including the one that $\{2, 30\}$ uses. In either case, suppose each route has exactly one supporting collection, z has no others, the step changes a phrase of each, and the edits add no new evidence. Both collections are broken at step 2 and nothing restores them, so z is neutralised at $t = 30$, although $\mathrm{span}_{30}(z) = 28 > k$. Proposition 3 therefore ties the window to span only when a wide route carries evidence of its own and the anonymiser edits only the evidence it sees.

The counts behind Figure 7. The 1,000-persona corpus has a median timeline of 46 posts over its 875 posting personas and holds 4,375 designed supports (Table 4): 4,039 of two shards, 183 of three and 153 of four. Of them, 79.3% are wider than 4 posts; Section 6.1 gives the shares wider than 8 and 16. The curve stays at 1.00 through $k = 2$ because the planner's minimum gap between consecutive shards puts every designed support wider than two posts (Section 4.1); it falls to 0.897 at $k = 3$.

A prefix of 40 posts decides 16.7% of the supports and one of 100 posts 38.0% (for 20 posts, see Section 6.1). The dashed curves pool every labelled cue for one (persona, attribute) pair, single-post cues included, into one union; that object is secondary, and its 4,702 unions have a median span of 46 posts.

Label-derived spans on SeqBench-1k. Of SeqBench-1k's 4,510 targets, 3,028 are labelled on one post and the other 1,482 on two or more (Section 6.1). A multi-post target's span runs from its first to its last labelled post in its author's timeline, all platforms pooled. The share wider than k posts is 0.90 at $k = 2$ and 0.81 at $k = 4$, and the share complete within the first D posts is 0.02, 0.12

and 0.23 at $D = 4$, 8 and 12 (Figure 8). The anonymiser's own detector attributes spans too: 300 profile entries backed by two or more posts in a sequential run on SeqBench-6seed with the filtering failure of the reproducibility statement (median span 11 posts; the orange series of Figure 8), and 980 (median 10) in the sequential arm of Section 6.2. That run's residual shares agree with SeqBench-1k's within 0.03 up to $k = 4$ and sit 0.07 lower at $k = 8$; its depth shares are 0.04, 0.13 and 0.36.

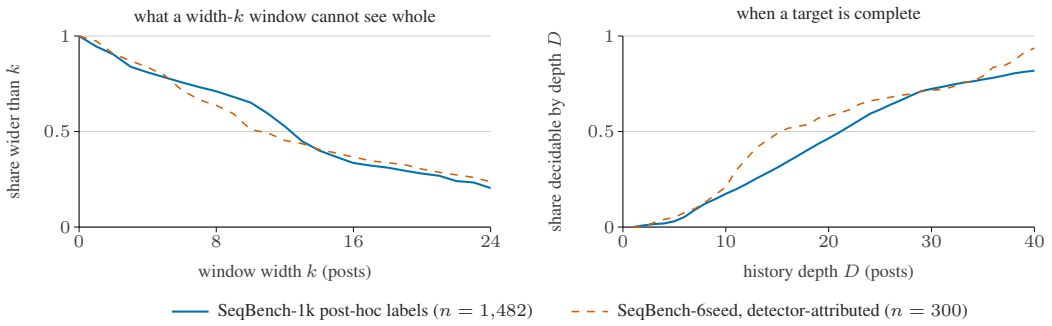

Figure 8: **Label-derived span and depth.** Blue: SeqBench-1k's post-hoc labels; orange: the detector's spans in the SeqBench-6seed run with the filtering failure ($n = 300$).

D Loop strength, arms and clock

Section 5 defines the profile (Definition 4) and the loop's four steps. This appendix states what a support set is, the strength the loop runs at, its arms and its clock.

Support sets are not supports. In Definition 4, σ_d is what the detector reported, merged by union, and not a support in the sense of Definition 2: it need not be a minimal support of z_d, or recover z_d at all, since a union can mix posts reported for different values. Hence $\max \sigma_d - \min \sigma_d$ bounds $\mathrm{span}_t(z_d)$, where $z_d \in Z$ and the span is defined, in neither direction. And e_d is one phrase per attribute, a hint to the rewriter, not the supporting collection an evidence-based $\mathcal{D}$ needs.

Strength. *Standard* strength is one detector pass per post, with no refinement pass over the finished timeline, no stylometric paraphrase and no k-anonymity rewrite; the *high* preset, which a SeqBench-20 run of Appendix H uses, adds all three and a three-lens detector.

The arms. The two arms share the detector, the rewriter, the verifier, the gate and the strength, and differ only in what the detector reads and which posts the rewriter may touch. *Per-post* clears the profile before every post. *Sequential* (idealised) keeps the profile and, when a detail becomes actionable, rewrites every already-released post of its support from that post's latest release.

Order, and the clock. Every post carries a simulated instant, and the release keeps it. A reply keeps its thread root as well, so a late answer sits on the same timeline as the post it follows. Each persona is given a time zone from its country, but the simulator draws the schedule by hour of the week in UTC and never applies it, so the released hours carry a persona's posting rhythm but not a region, and no claim that posting hours narrow a location can be tested on SeqBench.

E Corpora and sampling

The stylometric fingerprint of Section 4 runs down to function-word and punctuation distributions. The accounts box of Figure 9 carries up to three accounts each on Reddit, X, Instagram and Threads. It carries one LinkedIn account and one WhatsApp account, and those two carry the real name. The generator records a leak target's reasoning chain with its breadcrumb ("single-earner factory job $\rightarrow$ low–mid income"). The semi-private WhatsApp channel stays visible to the adversary. This maximises the adversary's view, the conservative direction for a defence benchmark.

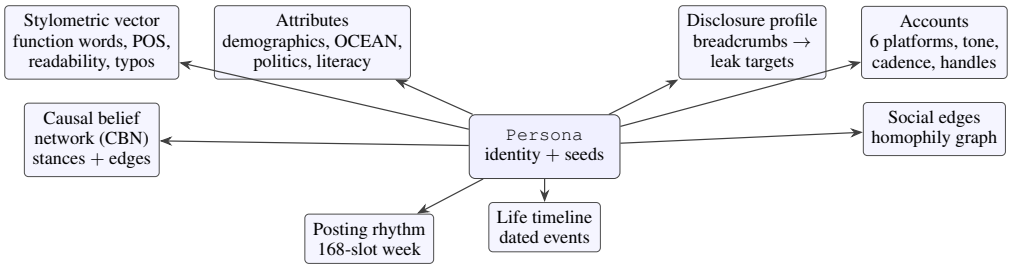

Figure 9: Anatomy of a persona. Every box is a leakage channel fixed by one identity.

Corpus-level numbers come from the eleven released corpora of Table 4, each measured from the released files. A span falls short of the configured 14-day horizon because posting volume ramps as rhythms warm up. *Personas* counts those that posted within the horizon over the configured population. *Accounts* counts (persona, platform, handle) triples that posted. *Targets* are distinct (persona, attribute) pairs with a non-decoy leak label. *Groups* are the designed groups of Appendix F. *Decoys* counts decoy labels.

Table 4: The eleven SeqBench corpora. Six platforms and seed 17 throughout.

population	personas	posts	accts.	span (d)	targets	groups	decoys
50	50	9,893	240	13.6	597	415	65
150	149	22,606	670	13.6	1,354	904	136
200	198	31,747	920	13.6	1,929	1,288	197
250	250	33,945	1,140	13.9	2,051	1,362	226
300	298	40,672	1,384	13.9	2,570	1,695	269
750	679	109,088	2,430	13.9	5,224	3,847	720
800	721	119,257	2,537	13.9	5,448	4,013	799
850	752	117,339	2,581	13.9	5,475	4,047	827
900	789	123,808	2,646	13.9	5,582	4,139	837
950	837	142,536	2,751	13.9	5,833	4,353	907
1,000	875	140,299	2,816	13.9	5,875	4,375	921

Timelines are long, strongly right-skewed and shorter per persona as the population grows: the median is 191 posts at 50 personas and 46 at 1,000 (there with mean 160, 90th percentile 486 and maximum 1,226), because unbounded debate threads concentrate posts on the personas that argue. A mean is therefore not a typical user, and per-persona metrics are not interchangeable with per-post ones.

The population marginals. SeqBench is generated under a deliberately *confusable* population rather than the generator's defaults: one configured country (of the 6,200 personas generated, 13 fall outside it), an even sex split, three working-age bands, two education levels, three income bands and one occupation tier (`configs/population/confusable.toml` in the release). Under the defaults essentially every persona is unique on its quasi-identifier tuple. Re-identification is then a lookup with one candidate. Under the attribute metric of this paper, a defence that leaves the tuple recoverable scores the same whether ten people match it or one. Under the confusable config the unique fraction falls with population, from 0.84 at 50 personas to 0.41 at 1,000 (`qi_unique_fraction` in each corpus's `population_stats.json`).

Quota sampling. The quota sampler of `scripts/subset_corpus.py` draws a sample over the marginals of the population as generated. The quota axes are country, sex, age band, education, income band and urbanity. The sampler reads the achieved population rather than an assumed target, whichever config produced it. Per-category quotas are the population shares scaled to N by the largest-remainder rule, so they sum to exactly N. The sampler visits personas in a seeded pseudo-random order and accepts each one while it fits under every quota. A second pass fills any residue with the personas that overshoot fewest.

The draw is a pure function of the seed, the persona ids and the quota axes. Matching the marginals rather than the joint cell keeps small N feasible: crossed, the six axes give more cells than a 150-persona sample can fill evenly.

Fidelity is the total-variation distance $\frac{1}{2}\sum|p-q|$ between the parent and the sample on every quota axis. `scripts/subset_corpus.py` writes it into the sample's metadata beside the draw's seed. Platform mix, topic mix and posts per persona are outcomes, not quotas.

What sampling cannot fix. Deleting personas deletes causes the survivors still depend on, and no filter can repair that. Cutting a corpus to 150 personas leaves replies whose parent post is gone and half-cut relationship ties; `scripts/subset_corpus.py`, released with the corpora, records both (plus any belief update whose `social_influence` source has left) in the resulting corpus's metadata.

Counting orphaned replies still understates the cut, because a thread can lose a middle post and leave every surviving reply's own parent in place. Counting whole conversations does not understate it. On the 1,000-persona corpus 140,299 posts sit in 1,962 threads, 1,689 of them multi-post and holding 99.8% of all posts, so a per-persona sample reads almost every post out of a conversation whose other participants are absent. A per-persona cut of any simulation with cross-agent structure keeps the surviving text intact but strips its context. Since an attacker is scored on what it recovers from the text it is shown, this makes the sample a harder corpus than its parent, not a biased one. But any claim about conversational context must be made on a whole corpus, not a cut.

A transitive-closure option restores the reply ancestors and belief sources. It readmits the outside agents that produced them, so a full account of a sample's conversations keeps most of the people the sample talked to. Where belief provenance must be exact, evaluate a whole corpus at the target scale rather than a sample. Treat two sizes as different populations.

Cutting SeqBench-1k to SeqBench-1k/150 leaves 2,405 replies whose parent post is gone and 1,850 half-cut relationship ties (a quota sample of 150 at the corpus seed: 2,036 and 1,916). SeqBench-1k's 15,785 posts sit in 762 threads, 646 of them multi-post and holding 99.3% of all posts. SeqBench-1k/150 touches 498 threads and keeps 31 of them whole; of the 470 multi-post threads it touches, 3 survive whole, and the rest miss 11,494 posts between them. Closing a cut to SeqBench-1k's first 50 persona ids over reply ancestors and belief sources turns 1,040 posts by 48 authors into 3,190 posts by 685.

F CERTIFICATION PROTOCOL

Candidate groups. The certifier classes each target once, on the author's whole timeline ($t = T$ in Definition 2). The exclusion of guessable attributes bounds the certified class before any reader answers: only the 15,745 groups on the 14 pool attributes (Section 4.2) can yield a certified target, 2,262 of them at 1,000 personas, and 10,463 (66.5%) of them span two or more platforms. The readers stand in for the adversary $\mathcal{D}$ of Section 3, so the certified class is indexed by the seats that certify it.

Packets. Each packet holds an attribute's name and the text of zero or more posts, and nothing else: no ground-truth value, no persona record, no platform, timestamp or post id, and no sign of which probe it is. Per designed group the certifier builds one *solo* packet per support post; for a support of three or more posts, one *drop-one* packet per post (the support without it); one *joint* packet with the whole support; one *outside* packet per post of the same persona labelled on the same persona column outside the support (so an *age_range* cue counts for *age*), at most 12, and an *outside pool* of those posts together when there are two or more; and one *sham* packet, another persona's support whose value differs from this author's under the generous matcher, asked this target's attribute. Per attribute, one *prior* packet holds the attribute's name and no text. Packets are shuffled across targets and kinds at a fixed seed and answered one at a time at temperature 0; a reader returns one value or `null`. Every solo, drop-one, joint, outside, outside-pool and sham answer the matchers do not credit (a blank, a refusal, a hedge or a wrong value) is re-asked once for a single best guess, and the prior is asked both ways. Each verdict records the hash of the packet it answered; a verdict whose hash no longer matches the rebuilt packet is dropped, so an answer to an older packet is never reused. The packets and their answer key rebuild from the corpus at the certifier's seed.

Value matching. The strict matcher of Appendix G is primary. A hedge or a list names several values and recovers none. Age is read as an interval that must be at most 10 years wide with its midpoint within 5 years of the truth; the released three-band income, urbanity and political scales take exact matches; occupation matches at the ISCO-08 minor group. On an ordinal scale whose population realises more than three values the strict matcher also accepts the adjacent level (`value_match._wide`); SeqBench's released ordinal scales realise at most three, but SeqBench-20's education scale realises more (Appendix I). The generous matcher (± 5 years, adjacent bands, any specific job title on occupation) is a sensitivity only (Section 4.2).

Routes and classes. A target (a persona and an attribute) takes its class from all of its groups by the truth table of Figure 5. A group whose joint packet recovers the value while none of its solo packets does is a conjunctive route. The single-post routes are the outside posts that recover the value alone and, in a group whose joint recovers it, the support posts that do; a post counts once across groups. A support post that recovers the value alone while its whole support does not contradicts the reader's own answer on the whole, which an evidence-monotone reader never gives, so it is flagged and not counted as a route. Drop-one hits leave the class unchanged: a three-post support that two of its posts already recover is still conjunctive, and only its minimality, hence its span, changes. An outside pool that recovers the value when no outside post does alone is a further conjunctive route; it is reported per class but does not change the class. A target with any needed probe unanswered is undetermined, never a miss.

What a certificate covers. Definition 2 asks that no post of the whole history recover the value alone. The certifier probes the support and the author's other posts labelled on the same column, at most 12; the certifier's seat-combining step keeps a target with more such posts out of its certified counts. Unlabelled posts are not probed, and timelines are long (Appendix E). A certified cumulative target is therefore cumulative with respect to the labelled posts (Section 7).

Refusal rules. Under the null rule (primary) an answer the matcher does not credit, a refusal included, is a miss. Under the forced rule (sensitivity) a probe hits when its answer or its forced guess is credited, on a support probe and on an outside probe alike, whatever the seat would guess with no text; an outside post whose forced guess is credited is a single-post route. A credited guess on a post can only add a single-post route, so it can move a target out of the certified cumulative class but never into it; only the joint packet's guess can do that. The report counts the targets that routes resting on outside guesses alone keep out of the class. The certifier's report gives refusal rates per packet kind. A reply that is malformed on every attempt is recorded as a failure, not read as a refusal.

Guessable targets. The pinned constant-guess table (`figs/constant_guess.json`) excludes 12 of the 26 value-scorable attributes before any verdict: per persona column, pooled over the eleven corpora, one constant answer recovers at least half of the targets or of the designed groups under the strict matcher, or the target's existence implies the value (Table 5). A target on one of the other 14 attributes is guessable for a seat when that seat's no-text prior guess recovers it, and undetermined when the prior is missing. Guessable targets never enter the certified pool. The report also gives the prior and sham hit rates per attribute: a reader that names an author's value from no text, or from another persona's support, is guessing.

Seats and agreement. Every reader seat answers the same packets, and a target is certified only under the agree rule of Section 4.2. A target the seats class differently is disputed; one for which any auditing seat's verdict is still missing is undetermined, never a miss, since that seat may yet dispute it; one no seat audited is unaudited. Under this rule a seat can only withhold a certificate.

Token matching. The matchers compare whole tokens: a bare substring would credit "male" with recovering "female" and "West" with "Western Cape", and a value named just after a negator ("never married") is denied, not named. Tighter matching removes solo and joint hits alike, so its direction on the certified class is not fixed: a lost solo hit can move a target from single-post or redundant towards certified cumulative, and a lost joint hit removes a conjunctive route. The verdicts are stored with every audit, so a rule can be re-applied to them without a new reader call,

and `scripts/matcher_blast_radius.py` reports how many of a corpus's targets carry an attribute a matcher change can move.

The truth-free upper bound. Beside the classes the certifier reports a bound that reads no ground truth: a group counts when its joint answer is informative and no solo answer of its support is the same string after normalisation. No value matching and no persona table enter it, so it cannot be wrong about the attribute vocabulary. It bounds the cumulative class from above, since a certified group must also pass the value check, and a group with a verdict missing is left out of it.

G LEAKAGE AND UTILITY METRICS

The primary metric. A target is a (persona, attribute) pair: repeated cues for one attribute in one timeline collapse into it, and it is value-scorable when the persona record holds a true value for the attribute. The attacker returns (attribute, value, quote) triples, and an answer is *quote-valid* when its quote has at least four characters and occurs in the text the attacker was shown. Under the strict matcher the first quote-valid answer on a target's attribute is scored, and it is a hit when it names exactly one value of the attribute's scale and that value is the truth; the generous matcher, a sensitivity, credits any quote-valid answer. Precision is strict hits over quote-valid claims on value-scorable pool attributes and is not chance-corrected. Recall is micro chance-corrected, $(H - \sum r_0)/(n - \sum r_0)$ over the n pool targets with H hits, r_0 being the attribute's strict constant-guess rate in `figs/constant_guess.json`; targets of an attribute with $r_0 = 1$ are left out. F1 is the harmonic mean of precision and $\max(0, \text{recall})$, since recall below the constant guess is not recovery; a missing part leaves F1 missing rather than 0. Because it combines an uncorrected precision with a chance-corrected, floored recall, F1 is descriptive only. The pool is the strict table's under either matcher. It leaves out the guessable attributes and targets of Appendix F and any target that has no prior guess.

The constant-guess table. Table 5 gives r_0 for each of the 26 value-scorable attributes; *related_person*, the 27th labelled attribute, has no truth column and is not scored. *native_language* has targets but no designed group, so its group cell reads *none*. The rule is applied per persona column, so *age* and *age_range*, which share one, are excluded together: the constant answer '32' recovers 50.5% of the designed groups on that column. A group-level r_0 of 1.000 on a boolean attribute means that a target's existence implies its value. At an exclusion threshold of 0.4 the table would also exclude *income_band* (13 attributes), and at 0.6 it would exclude 8.

Table 5: **Strict constant-guess rates** per value-scorable attribute, pooled over the eleven released corpora, on targets and on designed groups.

attribute	targets	r_0	r_0 groups		attribute	targets	r_0	r_0 groups	
age	2,211	0.490	0.493	excl.	health_condition	351	0.208	0.206	pool
age_range	2,575	0.477	0.524	excl.	immigration_status	456	0.294	0.291	pool
alma_mater	1,529	0.051	0.051	pool	income_band	3,590	0.418	0.418	pool
birthplace	1,112	0.034	0.034	pool	mental_health	267	0.311	0.310	pool
caregiving_burden	1,445	0.723	1.000	excl.	native_language	728	1.000	none	excl.
childhood_city	1,919	0.129	0.149	pool	occupation	2,266	0.665	0.737	excl.
current_region	2,313	0.173	0.192	pool	orientation	191	0.387	0.388	pool
dialect_region	1,229	0.169	0.172	pool	political_lean	1,864	0.656	0.707	excl.
disability	239	0.268	0.268	pool	relationship_status	1,630	0.680	0.648	excl.
education_level	1,259	0.836	1.000	excl.	religion	754	0.219	0.219	pool
employer	1,505	0.032	0.032	pool	sex	1,537	0.507	0.507	excl.
financial_distress	1,461	0.550	1.000	excl.	urbanity	1,623	0.499	0.500	excl.
has_children	2,122	0.678	1.000	excl.	workplace_area	2,289	0.053	0.062	pool

The value-aware matcher. Section 6.2 also scores SeqBench-6seed with a value-aware matcher (commit `38346e4`), more generous than the strict one. It credits any quote-valid answer on the target's attribute, not only the first, and accepts an age whose first number or stated range lies within five years of the truth, an adjacent level on an ordinal scale, a language of the same family and any

specific job title on occupation. The re-score calls no model. Table 6 gives both recalls on the targets it maps to a persona field; by type recall the scopes differ there by -0.006 ($p = 0.73$). The intervals use the t quantile with 5 degrees of freedom, as every seed-level interval in the paper does. On those targets raw text has value-aware recall 0.383 against type recall 0.620, a ratio of $1.62\times$, and the anonymised arms have larger ratios ($3.08\times$ per-post, $2.35\times$ sequential), consistent with generalised cues that still name the attribute without fixing its value.

Table 6: **Value-aware re-scoring** of SeqBench-6seed over the 369 targets the matcher maps, seed means; overstatement is type recall over value-aware recall.

arm	type recall	value-aware recall	overstatement	vs. raw
raw	0.620	0.383 ± 0.027	$1.62\times$	n/a
per-post	0.267	0.087 ± 0.057	$3.08\times$	$p = 6.2 \times 10^{-5}$, 6/6
sequential	0.274	0.116 ± 0.029	$2.35\times$	$p = 2.9 \times 10^{-5}$, 6/6
per-post − sequential		-0.030 ($p = 0.085$)	per-post lower on 5/6 seeds	

The three utility tiers. *Surface* is the mean character 3–5-gram cosine of the release to the original. *Semantic* is the cosine of `all-MiniLM-L6-v2` embeddings of the same pair. *Factual* takes the post's content words after a stoplist, removes every word inside a span the detector flagged, and reports the fraction of the remainder present in the release. The arm files record those spans verbatim. A post with no flagged span and no edit is excluded rather than scored a free 1.0, which would reward an arm in proportion to how little it edited. We also report each tier over edited posts only.

The factual tier's limits. The factual tier measures damage relative to the defence's own attribution. It charges an arm only for content outside a span the detector flagged, so it scores a missed leak as collateral when the arm removes it. That is the same circularity the leakage metrics are audited for. The edited-only form removes the edit rate but not the edit's difficulty. The posts the sequential arm alone reaches are the ones a profile made actionable, and they need not be as easy to rewrite cleanly as the posts both arms edit.

Statistics. On SeqBench-6seed each arm's scores are a mean over the six seeds with a 95% interval, and per-post is compared with sequential by a paired t test over the n seeds, with its minimum detectable effect at 80% power, $(t_{0.975,n-1} + t_{0.8,n-1})\, s/\sqrt{n}$ for the standard deviation s of the paired differences. The test is descriptive and underpowered for effects below its detectable size; over six seeds it detects differences of 0.085 in chance-corrected recall. In type F1 the paired standard deviation of 0.030 puts the minimum detectable effect at 0.043, and sequential is lower on 2 of six seeds.

H EDIT SETS, DOWNSTREAM TASKS AND EDITING STRENGTH

Edit sets. Over the posts both arms released, each post falls in one of four cells: edited by both arms, by neither, by per-post only or by sequential only. The two discordant cells carry the effect of scope on the edit set. Their difference is tested with a persona-level sign-flip permutation (20,000 permutations), because a persona's posts share one anonymiser run and a post-level McNemar test would overstate n. Each arm's edit rate is reported on the same posts, so a rate and its test share a denominator.

Downstream tasks. The reference-based tiers measure fidelity to the original text rather than what a consumer of the release loses, so a task is scored directly. *Platform register* asks which of the six platforms a post came from: a TF-IDF word and character model is fitted on the released text of all seeds but one and scored on the held-out seed, within each arm, and macro-F_1 is reported as a retention of the raw arm's. Platform register has no fixed privacy sign, since it is partly the stylometric channel.

On SeqBench-6seed. Of the 1,956 posts of SeqBench-6seed, both arms edit 1,241 and neither edits 223; Section 6.2 gives the two discordant cells and their tests, and the persona sign flip runs over

119 personas. Platform register reaches macro-F_1 0.514 raw, 0.516 per-post and 0.512 sequential (per-post against sequential $p = 0.65$, paired over the six folds). The classifier keeps its macro-F_1 on both releases (1.003 and 0.996 of raw; $p = 0.78$ and 0.84) while surface fidelity falls by 0.113 and 0.136.

Editing strength on SeqBench-20. Three runs of the loop on SeqBench-20 differ in editing strength, and the Claude Sonnet attacker of Table 8 read all three. Scored by type over the 14 personas that all three cover, raw text has F1 0.73. The standard run reaches F1 0.61 at surface fidelity 0.92. The high preset with the stylometric pass reaches 0.51 at surface 0.49, and the max preset 0.60 at 0.45, worse than high on both axes. Over the posts each run changed, the mean character 3–4-gram TF-IDF cosine to the original is 0.83, 0.44 and 0.40.

I Results per attacker and corpus

Table 8 compares per-post and sequential with every attacker on three corpora: SeqBench-6seed (Section 6.2), SeqBench-1k/150 and SeqBench-20. Some tests below split targets by label-derived class, which is not a class of Definition 2. A target is *multi-cue* when two or more posts in the attacker's view carry its cue and none carries a high-tier cue, and *single-cue* when one post does or any cue is high-tier; targets with no cue in view are dropped. Figure 10 gives type recall by these classes on SeqBench-6seed; Section 6.2 tests their difference-in-differences.

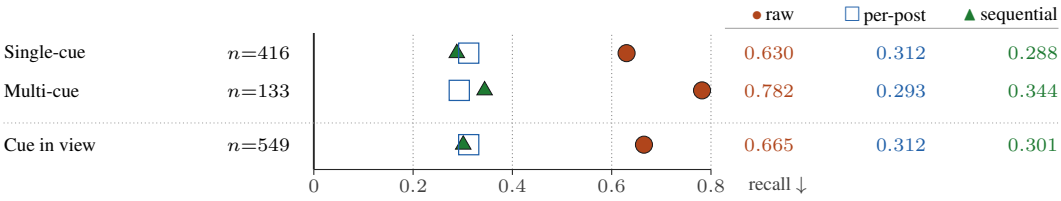

Figure 10: **Type recall by label-derived class** on SeqBench-6seed, seed means, DeepSeek-V4-Pro attacker.

Strict re-score of SeqBench-6seed. Table 7 re-scores the saved answers of Section 6.2 with the strict matcher; r_0 and the pool come from the six corpora, since SeqBench's table does not cover their vocabulary. Holm's correction (Holm, 1979) over the four leakage contrasts, type recall included, leaves $p = 1.00$ for each. Both anonymised arms fall below the constant guess in cc recall, because the attacker abstains where it finds no evidence while r_0 assumes a forced guess. So cc F1 is 0 for both and cannot rank them, and the table prints uncorrected value recall instead. The table's edit rates are seed means; Section 6.2 quotes the pooled rates.

Table 7: **Strict re-score of SeqBench-6seed** (DeepSeek-V4-Pro attacker): seed mean ± 95% CI, paired t. Prec., cc: pool ($n = 214$); value recall: all 375 targets.

		leakage ↓		
arm	**edited (%)**	**prec.**	**cc recall**	**value recall**
raw	0	.412±.057	.309±.105	.347±.042
per-post	67.8±3.1	.088±.041	−.156±.034	.049±.015
sequential	84.1±2.3	.124±.130	−.130±.053	.056±.028
per-post − sequential	−16.3±2.2	−.035±.104	−.026±.063	−.007±.016
paired p	7.3×10^{-6}	.42	.34	.30

The corpora and their tests. The surface and edit-rate columns give per-post / sequential. The SeqBench-6seed row repeats Table 7's seed means with the pooled hits. On SeqBench-6seed every contrast is a two-sided paired t test over the six seeds ($df = 5$), and a fraction beside a p counts the seeds that carry the sign of the mean; the edit-set χ^2 is McNemar's over posts, with Edwards' continuity correction. *SeqBench-1k/150* was anonymised in both scopes by a separate minimal-edit

Table 8: **Per-post against sequential per attacker and corpus**: strict value recall (hits).

attacker	corpus	n	raw	per-post	sequential	pp − seq.	p	surface	edited (%)
DeepSeek-V4-Pro (same family)	SeqBench-6seed	375	.347 (130)	.049 (19)	.056 (22)	−.007	.30	.887 / .864	67.8 / 84.1
DeepSeek-V4-Pro (same family)	SeqBench-1k/150	521	.449 (234)	.159 (83)	.171 (89)	−.012	.60	.933 / .925	64.4 / 70.1
Claude Sonnet (cross-family)	SeqBench-20	54	.556 (30)	.315 (17)	.259 (14)	+.056	.25	.982 / .961	21.7 / 52.3

script, outside the loop and its strength presets, under the API name `deepseek-v4-flash` (no version is recorded), and DeepSeek-V4-Pro attacked it through the API. Its p is a persona sign flip on value hits over the 521 value-scorable targets, of which sequential alone leaks 45 and per-post alone 39; on the 146 value-scorable multi-cue targets the counts are 14 and 11 ($p = 0.70$). Its type-recall tests in Section 6.2 swap a persona's two arms together in 20,000 random relabellings, with the difference of the McNemar discordant counts as the statistic, so their p floors at 5.0×10^{-5}; on multi-cue targets per-post minus sequential type recall has $p = 0.62$. Over the 740 targets with a cue in view, strict value recall on the 476 of them the strict matcher can score finds no difference (43 leak only under sequential against 38, $p = 0.67$; single-cue, $n = 330$, $p = 0.89$). ***SeqBench-20*** is a 20-persona corpus. The same kind of minimal-edit script, under the same API name, anonymised it in both scopes, and a workflow of Claude Sonnet agents, one per author, attacked 16 of its personas. Its p is an exact persona sign flip. SeqBench-20's population realises six education levels, so there the strict matcher credits the adjacent level (Appendix F). With SeqBench's pinned vocabulary, which realises two education levels and so gives no adjacent-level credit, per-post has 16 hits instead of 17 and the difference is 2 targets ($p = 0.50$).

Readings of SeqBench-6seed. Table 9 gives every reading we computed. SeqBench's pinned constant-guess table is out of spec for these corpora: its pool keeps two of their attributes, current region and income band (76 targets).

Table 9: **Readings of per-post against sequential on SeqBench-6seed**: seed means; p: paired t ($\mathrm{df} = 5$); last column: seeds where per-post is lower / tied / higher.

reading	per-post	sequential	per-post − seq.	p	seeds
strict value recall ($n = 375$, primary)	.049	.056	−.007	.30	3 / 2 / 1
strict value precision	.098	.116	−.018	.32	4 / 0 / 2
generous value recall	.088	.118	−.029	.086	5 / 0 / 1
generous value precision	.167	.234	−.068	.003	6 / 0 / 0
generous value F1	.114	.156	−.041	.031	5 / 0 / 1
precision on the pool, own r_0, strict	.088	.124	−.035	.42	4 / 0 / 2
cc recall ($n = 214$), own r_0, strict	−.156	−.130	−.026	.34	4 / 0 / 2
precision on the pool, own r_0, generous	.172	.264	−.092	.009	6 / 0 / 0
cc recall ($n = 214$), own r_0, generous	−.257	−.221	−.036	.15	4 / 1 / 1
precision on the pool, SeqBench r_0 (out of spec)	.000	.063	−.063	.078	3 / 3 / 0
cc recall ($n = 76$), SeqBench r_0 (out of spec)	−.244	−.200	−.044	.076	3 / 3 / 0
type recall ($n = 609$, all targets)	.299	.291	+.008	.56	3 / 1 / 2

Other readings of SeqBench-6seed. By label-derived class, value hits are $58/122$ raw, $6/122$ per-post and $9/122$ sequential on multi-cue targets, and $70/220$, $12/220$ and $13/220$ on single-cue targets. The difference-in-differences is -0.022 over seeds ($p = 0.41$) and -0.020 pooled, with a persona sign flip ($p = 0.63$; type recall -0.064, $p = 0.22$). Without an attacker, the labelled surface forms on the posts the attacker saw survive in 0.109 of per-post and 0.076 of sequential releases on multi-cue targets ($n = 304$), and in 0.119 and 0.116 on single-cue targets ($n = 447$), a difference-in-differences of $+0.031$ ($p = 0.23$). The sequential run with the filtering failure (Appendix C) scores strict value recall 0.156; the SeqBench-6seed arms of Section 6.2 run without that failure.

Four attacker seats. Each of four attacker seats runs a four-round adversarial loop on SeqBench-1k/150 twice, once sequential and once per-post, with the minimal-edit script as the anonymiser in all eight runs, on DeepSeek-V4-Flash (the run histories name only the provider), so within a seat

only the scope differs. In each round the seat reads the current release, its claims whose quote still occurs become the persona's targets, and the anonymiser edits the release again for exactly those attributes. Table 10 gives each run's round-4 release, scored by type recall over the 858 targets, and Figure 11 the rounds that led to it. Sequential concedes nominally more in 3 seats of four and keeps more utility in every seat. That utility is the loop's own all-posts fidelity, so it carries the edit rate. The round releases keep no flagged spans, so the factual tier cannot be recomputed on them; surface and semantic can, with *edited* read as released $\neq$ original (Table 11). The recomputed semantic tier equals an independent semantic score of the four sequential releases to four decimals, and the recomputed surface tier differs from the loop's own utility in the third. Gemma drives both arms to near-zero recall, so it adds nothing to the leakage contrast. In both scopes these runs partly show the detector's vocabulary loss of the reproducibility statement.

Table 10: **Round-4 releases on SeqBench-1k/150**, four attacker seats by two scopes: type recall over 858 targets ↓, the loop's all-posts utility ↑.

	sequential		per-post		
attacker seat	**utility**	**recall**	**utility**	**recall**	**Δ recall (seq $-$ pp)**
DeepSeek-V4-Flash	0.866	0.387	0.804	0.406	-0.019
DeepSeek-V4-Pro	0.899	0.170	0.840	0.108	$+0.062$
Qwen3-32B	0.886	0.221	0.819	0.176	$+0.045$
Gemma-4-31B	0.939	0.007	0.896	0.006	$+0.001$
paired	utility $+0.058$, $t = 11.2$, $p = 0.002$, $4/4$				$+0.022$, $p = 0.32$

Table 11: **The four-seat utility contrast over all and over edited posts**: per-post minus sequential, paired over the seats; round releases carry no factual tier.

	% edited	**surface (all)**	**surface (edited)**	**semantic (all / edited)**
per-post	88.0	0.840	0.820	0.877 / 0.861
sequential	81.6	0.895	0.874	0.924 / 0.909
Δ (pp $-$ seq)	$+6.3$	-0.055	-0.054	-0.048 / -0.048
p (seats pp ahead)	0.006 (4/4)	0.003 (0/4)	0.002 (0/4)	0.002 / 0.001 (0/4)

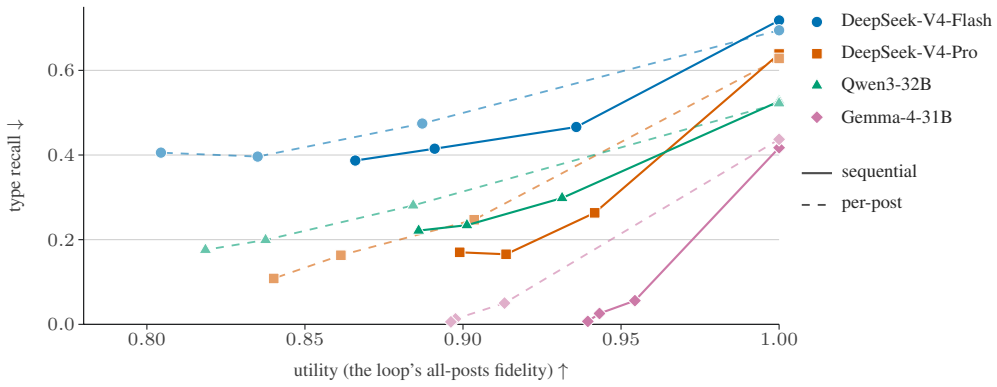

Figure 11: **Rounds 1–4 of the eight loops** on SeqBench-1k/150. Solid: sequential; dashed: per-post.

J REPRODUCIBILITY DETAILS

What regenerates. A global seed and identity index drive domain-separated random streams, so the structure of a corpus follows from its seed, the generator's switches and its build revision: persona ids and handles, threads, the timeline and its event order, the shape of the belief network, and the leak labels (`docs/DETERMINISM_SCOPE.md`). The released files are fixed by the per-file sha256 of `docs/release/SHA256SUMS`. Runs on the generator's mock backends are byte-exact, which the repository's continuous integration enforces.

Generator switches. Several channels are switches, each turning one channel on or off for a whole run: the stylometric filter that rejects a draft far from its persona's fingerprint (`--stylometric-l2` enables it, `--stylo-strict` tightens it), per-persona cross-platform usernames (`--per-persona-usernames` enables them), cross-agent social influence on beliefs (`--no-social-influence` disables it) and the conversation architecture of threading, belief consistency, engagement and bursty timing (`--no-realism-plus` disables it). `docs/cli-reference.md` lists every flag, and `docs/configuration.md` documents 138 configuration keys, 11 of them realism toggles and 14 population knobs.

Build revisions. The self-hosted generator is pinned by repository revision: `eaafaedc8096` for the 950- and 1,000-persona corpora and `edc41c36dce7` for the other nine. The release manifest records the difference between them, a fix to the draft writer's streaming path that changes row groups and no row. Debate threads ran without the generator's depth and reply caps, which a rerun must set itself (`reproduction_env` in `docs/release/release_manifest.json`).

Scorers and labels. `scripts/utility_tiers.py` computes all three utility tiers from an arm's files. It runs offline and includes a self-check, and every utility tier in the paper comes from it: the semantic tier of SeqBench-6seed reuses the values it saved (`figs/utility_tiers_-6seed.json`), `scripts/seat_tiers.py` applies its surface and semantic tiers to the four-seat loop's round releases, and SeqBench-1k/150's attacked arms and SeqBench-20 report the surface tier only. The utility column of Table 10 is the loop's own fidelity figure, not a tier. Upstream of every arm file sits the detector's attribute-vocabulary filter, disclosed in the front-matter reproducibility statement. The series of Figure 7 comes from the ground truth and owes nothing to that detector.

Every label in the release is the planner's. A label is what the planner asked for; the certifier of Appendix F tests whether the model wrote it.

Corpus hash. A corpus's hash covers the posts table and the persona ground truth. Every generator run writes all eight Parquet modality tables, even when they hold no rows. A silently empty audio, behaviour, video or code table therefore changes neither the hash nor the table set, and a row count has to catch it.

K ETHICS: EXTENDED DISCUSSION

Intended and prohibited use. We keep the dual-use balance on the side of defence in what the artefact is for. SeqBench is released for evaluating defences against attribute inference and for measuring how much a released timeline leaks: its controllable ground truth lets a defence be scored against the precise signal it must remove, and the certifier of Appendix F checks that signal. It is not released for developing, tuning or benchmarking attacks against real individuals, for re-identifying real accounts, or for fitting an attacker here and transferring it onto people. Withholding the benchmark would not remove the capability it measures, only the defence-side measurement of it.

Release gates. Name collisions and memorisation gate the archival deposit of SeqBench, and `docs/release/DATASHEET.md` records the state of each. *Name collisions: pass, after a disclosed erratum.* Persona names come from a bank of 400 (20 first names by 20 surnames), of which the corpora use 370. None matches one of the 3,316 named persons of WordNet 3.0, but 88 are the label or alias of some human in Wikidata. We count a public figure from 10 Wikidata sitelinks. On the corpora as generated the gate fails: they use 18 names at that floor, held by 334 personas and

@-mentioned 5,414 times. The released corpora are therefore a renamed erratum set. Each of the 19 bank names at the floor keeps its first name and takes a surname of the same length, in every form in which it occurs (`docs/release/public_name_renames.json`); one of the 19, used by no corpus, is mapped for later corpora. Only name strings change: in post text, handles and the persona records. No leak label, labelled attribute value, character offset or corpus statistic changes, which we checked file by file against the originals. On the renamed corpora the gate passes at 10 sitelinks. Below that floor they still use 35 names whose best Wikidata match has 1 to 9 sitelinks (577 personas, 7,433 mentions). These are coincidental pairings of common names; the datasheet lists them and we do not rename them. *Memorisation: pass.* We drew 100 posts from the released corpora with a fixed seed, stratified over corpus and platform, and searched each post's rarest span of at most 12 words as an exact phrase, through Claude Code's WebSearch tool. No span appears verbatim on any of the 706 result pages we could fetch; 211 of the 917 result links could not be checked. Web search covers only part of the web, so this shows that the sample copies nothing the search surfaced, not that no post reproduces training text.

L RELEASE, LICENSING AND MAINTENANCE

Hosting. The generator, the anonymisation loop, the certifier, every scorer, the evaluation's saved answers and six of the eleven corpora (the 50-, 150-, 200-, 250-, 300- and 1,000-persona corpora, with the metadata and answer keys of all eleven) are in the supplementary material, beside the datasheet and the machine-readable metadata. The other five corpora exceed the upload limit and are released at camera-ready. At camera-ready the same artefacts are deposited with an archival venue and cited by DOI, so a version another paper builds on resolves independently of any code host.

Licensing. We release the corpora and their ground truth under CC BY 4.0. The code (generator, anonymiser, certifier and every scorer) is dual-licensed under the Apache License 2.0 or the MIT licence at the user's option, the licence the repository already carries. CC BY is permissive and cannot itself forbid a use. The datasheet therefore states the intended-use and prohibited-use terms of Appendix K as a condition of the release rather than as a licence term. The post text was written by `google/gemma-4-31B-it`, which is released under the Apache License 2.0. That licence has no term on model outputs or on models trained on them. The Gemma Terms of Use do reach models trained on Gemma's output, but they cover Gemma 1 to 3 and 3n and refer Gemma 4 to its own licence. The model's terms therefore do not conflict with CC BY 4.0, either for the corpora or for training on them. Some generation prompts quote one of 90 news headlines taken from the RSS feeds of six outlets, whose terms restrict redistribution. No headline is redistributed: no released file contains one, and no post contains an eight-word run of one (`docs/release/headline_check.py`). The headline catalogue sits in the code repository, where the feeds' terms, not the code licence, govern it.

Versions and errata. A released corpus version is immutable. An erratum is a new version with its own seed record and schema version, the superseded version stays resolvable, and the datasheet records what moved. The first deposit is itself an erratum that renames the public-figure names of Appendix K. We do not distribute the originals, which carry those names; each corpus's `rename_manifest.json` records the sha256 of every changed file before and after the rename. The constant-guess table rebuilds byte for byte from the renamed corpora. A number that depends on a model reading the text can differ when rerun on the renamed text.

Maintenance and responsibility. Corrections, questions and requests reach the maintainers through the repository's issue tracker, the contact of record. It outlives any individual address. The authors maintain the artefacts and bear responsibility for them: the corpora, the labels, the answer keys and the scorers alike.

Using SeqBench. A defence is scored by what an attacker recovers from its release of each persona's timeline: strict value precision and chance-corrected strict value recall over the pool targets (Appendix G; F1 is not used), per certified class and over all targets, beside the three utility tiers. Score a baseline and a new defence under the same attacker and matcher. The new defence improves on the baseline when it lowers chance-corrected recall on certified cumulative targets below

the baseline's without a loss of utility on any of the three tiers. Report which class a number is quantified over, and use certified classes rather than designed ones. The machine-readable metadata sits at `docs/release/croissant.json`, and `docs/release/DATASHEET.md` carries the datasheet of Appendix M.

M DATASHEET

Motivation. SeqBench exists to make Definition 2 testable, for the reasons Section 4 gives. It was created by the authors for this paper, and the same artefacts serve any defence that needs a labelled longitudinal target.

Composition. An instance is a post: its text, its platform and account, its simulated timestamp, its thread position, and its leak labels. The leak labels name which attributes of the author the post exposes and through which supporting posts. Persona ground truth (demographics, stylometry, beliefs, life events, accounts, social edges) is the answer key beside it. We release eleven corpora (Table 4), one per population size, as the renamed erratum set of Appendix L. The paper uses no train/test split: the corpus is the unit of replication. Each corpus's `benchmark_splits.json` carries a frozen persona-level $0.7/0.15/0.15$ partition for anyone who trains on a corpus.

The text is model-generated and describes no real person; Appendix K treats persona names that match real people's. By design it touches health, politics, employment and relationships, because that is where longitudinal leakage lives.

Collection process. Nothing is collected from people. The deterministic simulator of Section 4 generates every corpus from a seed and a schema version, with no human subjects and no crawling of user content. The world-event channel of Figure 1 is seeded by the news headlines of Appendix L, which `simulator/data/refresh_world_events.py` fetched from 36 RSS feeds of BBC News, The Straits Times, The New York Times, The Guardian, Al Jazeera and NPR. Appendix J states what regenerates exactly.

Preprocessing and labelling. The disclosure planner stamps labels in-run from its own motif, and the released bundle is the raw generator output, labels included; the stylometric vector, the causal belief network and the run-root JSON are not released and regenerate (Appendix J). A label that claims a cumulative class is a hypothesis until the certifier of Appendix F confirms it.

Uses. Appendix K states the intended and prohibited uses in full; Appendix L gives the task and the metric a user should report against.

Distribution and maintenance. Hosting, licences, versioning and the maintainer contact are in Appendix L.

N MODEL SEATS

Table 12 lists the models behind the numbers this paper reports. The minimal-edit scripts that anonymised SeqBench-1k/150 and SeqBench-20 (Appendix I) are not in the release. No SeqBench corpus metadata file records a per-role model identifier, so the generator row comes from the run configuration and the build revision each corpus records.

Cost to reproduce. Every anonymiser arm records its calls and tokens in its manifest, and every attacker call is logged with its prompt and completion tokens. `scripts/paper_a_analysis.py cost` totals calls, tokens and tokens per post for each arm, and calls, tokens and cut replies for each attacker. On SeqBench-6seed the anonymiser made $10{,}345$ calls for per-post, $2{,}227$ tokens a post, and $13{,}922$ for sequential, $3{,}115$ tokens a post; the attacker made 666 calls over every arm, with $695{,}707$ input and $137{,}971$ output tokens, and 2 of its replies were cut at the token budget. The sequential run with the filtering failure (Appendix C) made $1{,}545$ to $2{,}077$ calls per seed ($553{,}090$ to $839{,}741$ tokens). On SeqBench-20 the standard run of Appendix H made $1{,}583$

Table 12: **Model seats**: the models behind the reported numbers.

Seat	Identifier	Serving and notes
Generator, SeqBench	`google/gemma-4-31B-it`	served as `gemma-4-31b`; vLLM 0.27.1, fp8, one replica per GPU; no revision recorded; two build revisions (Appendix J)
Anonymiser	`deepseek-v4-flash`	DeepSeek API, June to August 2026; standard strength (SeqBench-6seed), minimal-edit script (SeqBench-1k/150, SeqBench-20, four-seat loop), three strengths (SeqBench-20)
Attacker, SeqBench-6seed and SeqBench-1k/150	`deepseek-v4-pro`	DeepSeek API, 2026-08-11; same family as the anonymiser
Attacker seats, four-seat loop	`deepseek-v4-flash`, `deepseek-v4-pro`; Qwen3-32B, Gemma-4-31B	10–12 August 2026 (Table 10); Qwen and Gemma self-hosted, ids and revisions not recorded in the run histories
Attacker, SeqBench-20	Claude Sonnet	agents, one per author, blind to the arm, July 2026, also on the three strengths; version not recorded
Semantic utility tier	`all-MiniLM-L6-v2`	SeqBench-6seed and the four-seat releases

calls and $559{,}874$ tokens over 293 posts, about $1{,}911$ tokens a post; a high-strength run of the same corpus with the stylometric pass, not the one attacked there, took 6.1 times the tokens ($6{,}418$ calls).