PRIVATE FEDERATED TOPIC DISCOVERY USING ON-DEVICE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Discovering recurring topics in user-generated text can guide improvements to AI and edge systems, but centralizing this text creates substantial privacy risks. We study distributed and privacy-preserving topic discovery, focusing on how user text is represented locally before private aggregation. We propose bounded semantic compression as a local representation method, whereby an on-device language model compresses each user's data into a bounded set of semantic terms. This addresses two limitations of existing the existing literature. First, it preserves more topic-relevant information under a fixed contribution bound, improving recovery after privacy noise is added. Second, it can name topics that are implied by the text but never stated directly, removing the requirement that a topic descriptor appear in the original text. We combine this representation with distributed private aggregation, providing user-level differential privacy for the released population topics without revealing individual user contributions. Across natural-language, synthetic, and source-code datasets, semantic compression improves topic recovery across private aggregation mechanisms, with the largest gains for multi-document users and topics that are implied rather than named.

1 INTRODUCTION

A large and growing fraction of natural-language data is produced on user devices in the form of prompts to AI assistants, messages, notes, and personal documents. Discovering recurring topics in this data can help identify shared needs and guide improvements to user experience on mobile and edge devices. However, user text is privacy-sensitive, since prompts, notes and personal messages may disclose private information about individual users.

Existing tools for private discovery face different privacy and trust constraints. Recent systems analyze prompt data using a trusted server where Tamkin et al. (2024) relies on heuristic privacy protections, whereas Liu et al. (2025) uses central differential privacy (DP) (Dwork & Roth, 2014) to protect released statistics. Similarly, private item-discovery methods such as DPSU discover frequent words or strings under bounded user contributions (Gopi et al., 2020) with central DP. Decentralized approaches utilize local DP which removes the trusted-server assumption, but can incur severe utility loss for open-ended discovery (Erlingsson et al., 2014; Wang et al., 2017). Federated topic models such as FedLDA apply local randomization, although the evaluated pipeline does not provide an end-to-end user-level guarantee, and FedNMF provides no formal DP guarantee (Wang et al., 2020b; Si et al., 2022). In contrast, federated heavy hitters (Zhu et al., 2020) discover frequent items held on user devices and can count votes with secure aggregation (Bonawitz et al., 2017), so the server sees only aggregates, although it must still be trusted to apply the release threshold.

However, much of the current decentralized and federated work focuses on private *aggregation*, *i.e.* how information from different users is combined while preserving privacy (Zhu et al., 2020; Erlingsson et al., 2014; Wang et al., 2020b). The *representation* of each user's text, which determines what information is available to aggregate, is however largely unaddressed. Under user-level DP with a fixed per-user contribution this choice becomes a crucial factor because a user may need to represent many messages or documents in a limited contribution.

Moreover, current approaches limit which topics can be expressed. Although FedLDA and FedNMF target topics directly, item-discovery approaches (Gopi et al., 2020; Zhu et al., 2020; Erlingsson et al., 2014; Wang et al., 2017) return frequent words or strings from which topics or usage patterns

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.

may then be inferred. However, in both cases, the representation is restricted to words contained in the user's text, and therefore cannot contribute a topic descriptor that is absent from that text. For example, a local text message may express *grief* without containing the word "grief," and source code can implement *web scraping* without mentioning "web scraping." Existing methods must then rely on related words in the input, which may provide weaker or less direct evidence for the underlying topic. Together, these privacy and representation challenges motivate our main question:

Can we achieve high-utility federated topic discovery, under user-level privacy?

We address this question with **bounded semantic compression**. A language model maps each user's complete local history to at most K semantically meaningful terms. This representation serves two distinct roles. First, semantic compression can map related content from different users to shared descriptors, increasing the number of users contributing to the same coordinates while respecting the same contribution bound. These descriptors can then remain visible more reliably after privacy noise is added. Second, language models enable what we call **semantic abstraction**, which allows the representation to contribute concepts that need not occur literally in the source text.

Our work centers on two claims. **C1, bounded representation.** Under a fixed user-level contribution budget, semantic compression makes better use of the available contribution, particularly when many local documents must be represented, and preserves more recoverable topic information under private release. **C2, semantic abstraction.** Generative representation can contribute descriptors not present in the source text, substantially improving discovery when population topics are expressed implicitly rather than named directly.

These properties are useful beyond our default aggregation mechanism. We show that replacing lexical representations with semantic representations considerably improves private discovery mechanisms. For our own pipeline, each user's data is mapped to a bounded binary vector over a shared public vocabulary and we perform population aggregation using a private histogram. We add Gaussian noise calibrated to the user contribution bound and release terms with sufficient noisy frequency. Because this statistic is additive, it can be realized through secure aggregation with distributed Gaussian noise (Bonawitz et al., 2017; Kairouz et al., 2021). Unlike federated heavy hitters (Zhu et al., 2020), noise is added before the server sees the aggregate, so its view is itself differentially private, with the utility of central DP.

We then organize the privately released terms into topics. We use *topic discovery* to mean identifying recurring population-level semantic themes rather than estimating a topic mixture for each private document. Following prior work showing that pretrained word embeddings can be clustered to obtain useful topics (Sia et al., 2020), we embed the released terms using a public sentence encoder and cluster them in semantic space. Within each cluster, terms are ordered by their noisy user frequency, and topics are ranked by their combined released frequency. This recovery stage uses only public models and the differentially private aggregate, so it incurs no additional privacy cost.

Our experiments across natural-language corpora, source code, and synthetic data show that semantic representation improves private discovery across aggregation mechanisms. With multi-document users, semantic compression becomes more useful as more documents must fit through the same user contribution bound and preserves more recoverable topics after private release. We show large gains particularly when target concepts are implied rather than explicitly named. We also evaluate the downstream usefulness of the privately released terms, the distributed secure-aggregation realization, and the effect of local model size and compute.

We make the following contributions.

- We study local representation as a separate design choice in federated topic discovery, distinct from private aggregation. Across several private discovery mechanisms, replacing lexical contributions with semantic representation improves recovery, suggesting that representation can be an important source of utility independently of the aggregation method.
- We introduce bounded semantic compression, which maps a user's local history to at most K semantic terms. We analyze how mapping related content from different users to shared terms can increase their chance of remaining visible after privacy noise is added, and show on genuine multi-document histories that semantic compression becomes more useful as more documents must be represented under the same user contribution bound.

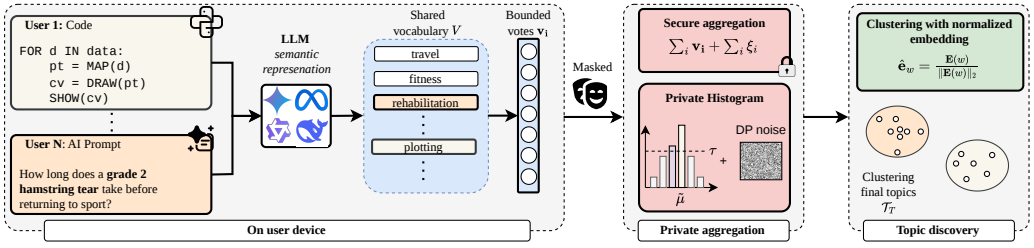


Figure 1: High level overview of our methodology. Users have different data, such as code (User 1) or a prompt to an AI (User N). Using a local LLM users map their data to meaningful semantic terms. Using a public vocabulary the server aggregates binary word-presence vectors via secure aggregation. We add Gaussian noise in accordance with DP and cluster the words using word embeddings.

- We study semantic abstraction as a distinct property of the representation. In experiments on natural language, synthetic data, and source code datasets, introducing terms that are absent from the input substantially improves recovery when the target topics are implied rather than named directly.
- We show that bounded semantic representations can be combined with a simple additive private histogram over a shared vocabulary .The release is user-level DP and admits a distributed realization based on secure aggregation and distributed noise, which we evaluate under finite precision and client dropout.

2 Related Work

Private discovery methods differ mainly in where trust is placed. Clio (Tamkin et al., 2024) and Urania (Liu et al., 2025) analyze conversations on a trusted server, the former with heuristic protections and the latter with keyword-based summarization and record-level DP, and Cheu et al. (2025) run LLM-based summarization server-side inside trusted execution environments. DP set union (Gopi et al., 2020) and DP n-gram extraction (Kim et al., 2021) give DP guarantees for open-ended items but also assume a trusted server. Local DP works (Erlingsson et al., 2014; Wang et al., 2017) remove the trusted server at a large utility cost, whereas federated heavy hitters (Zhu et al., 2020) count votes under secure aggregation but still trust the server to apply the threshold, and distributed Gaussian noise (Bonawitz et al., 2017; Kairouz et al., 2021) removes this step while retaining central-DP utility. Across these trust models, prior work focuses on how bounded contributions are aggregated and fills each contribution with words or strings taken from the user's text. Federated topic models (Wang et al., 2020b; Si et al., 2022) learn topics directly but likewise operate on lexical features of the input. How an unstructured user history should be represented on the device, before a user-level contribution bound is applied, and whether that representation may name topics the text never states, remains largely unaddressed and is the local step our work contributes. Section M gives the extended discussion.

3 Method

We separate private federated topic discovery into three stages (Algorithm 1). Each user first maps their local documents to a bounded semantic representation. These contributions are then privately aggregated, after which the released terms are organized into topics using public information only. We consider a single round, akin to federated analytics (Ramage & Mazzocchi, 2020); multiple rounds are possible via privacy composition but are not considered here.

3.1 Setting

Let $D = (H_1, \ldots, H_n)$ denote the text held by n users, where $H_i = \{x_{i1}, \ldots, x_{im}\}$ is the set of documents belonging to user i and m the number of documents per user. An absent user has

$H_i = \perp$. We use add/remove user-level adjacency, so neighboring datasets $D \sim D'$ differ by adding or removing all documents belonging to one user. Privacy is therefore defined over a user's entire history, however many documents it contains.

3.2 Bounded semantic compression

In the first step, a user maps their data into a short list of semantic terms. The language model M_θ is run with a fixed prompt P that asks it to describe the main semantic content of the text, and we write $Y_i = M_{\theta,P}(H_i)$ for the description it produces. Since this description is free text, we keep only the words that belong to a public vocabulary V_0 of $d = |V_0|$ words, which gives the candidate terms $A_i = \mathrm{tok}\big(M_{\theta,P}(H_i)\big)$, where $\mathrm{tok}(\cdot)$ is the function mapping to V_0. A *term* is thus a public vocabulary word that the model chose when describing the user's text, and each term $w \in V_0$ indexes one coordinate of the contribution vectors below.

Since a user may hold many documents, we cap how much any one of them can contribute for DP. A public contribution bound $K \geq 1$ keeps at most K terms, $S_i = \mathrm{Clip}_K(A_i)$, and the contribution sent for aggregation is the binary vector $v_i = \mathbf{1}_{S_i} \in \{0,1\}^d$, whose coordinate $v_{i,w}$ is one when $w \in S_i$ and zero otherwise. When a user holds several documents, the model can describe them jointly or one at a time. In the latter case the terms are pooled and Clip_K keeps those generated for the largest number of documents. The bound is applied once per user. Each user therefore contributes at most one vote to any term and at most K votes in total, regardless of the number or length of their documents.

Algorithm 1 Private topic discovery

Input: user histories $H_1, \ldots, H_n$; public model M_θ, prompt P, vocabulary V_0, bound K, noise σ, threshold τ, budget B, encoder E, number of topics T
Output: topics $\mathcal{T}_1, \ldots, \mathcal{T}_T$

Local representation *(on device)*

for each user i **do**
  $Y_i \leftarrow M_{\theta,P}(H_i)$ ▷ describe text
  $A_i \leftarrow \mathrm{tok}(Y_i)$ ▷ words in V_0
  $S_i \leftarrow \mathrm{Clip}_K(A_i)$ ▷ $\leq K$ terms
  $v_i \leftarrow \mathbf{1}_{S_i}$ ▷ binary votes
end for

Private aggregation *(user-level DP)*

$\tilde{\mu} \leftarrow \frac{1}{n}\sum_i v_i + \xi,\ \xi \sim \mathcal{N}(0, \sigma^2 I_d)$
$W_\tau \leftarrow \{w \in V_0 : \tilde{\mu}_w > \tau\}$ ▷ threshold
$W \leftarrow \mathrm{top}_B(W_\tau; \tilde{\mu})$ ▷ release

Topic recovery *(post-processing)*

$\hat{e}_w \leftarrow E(w)/\|E(w)\|_2,\ w \in W$
$\mathcal{T}_{1:T} \leftarrow k\text{-means}\big(\{\hat{e}_w\}, T\big)$
Sort terms in each $\mathcal{T}_t$ by $\tilde{\mu}_w$
Sort topics by $\sum_{w \in \mathcal{T}_t} \tilde{\mu}_w$
return $\mathcal{T}_1, \ldots, \mathcal{T}_T$

Moreover, we distinguish two different kinds of representations the local language model can produce. We write $\mathrm{tok}(H_i) = \bigcup_m \mathrm{tok}(x_{im})$ for the vocabulary words that appear in the user's own documents. We define a representation as *extractive* when $S_i \subseteq \mathrm{tok}(H_i)$ for every user, meaning a language model restricted to words from the input. On the other hand, a *generative* representation may instead contribute terms that do not occur in the user's text. This is crucial for our second claim (C2), since it allows *semantic abstraction*, which can name concepts and terms not present in the user text. We demonstrate the significance of this in the results.

3.3 Private aggregation

After clients complete their local representation, each contributes a binary vector $v_i \in \{0,1\}^d$. We aggregate by averaging these vectors,

$$\mu(D) = \frac{1}{n}\sum_{i=1}^{n} v_i, \tag{1}$$

where the coordinate $\mu_w = \frac{1}{n}\sum_i v_{i,w}$ is the fraction of users who contribute term w, since each $v_{i,w}$ is binary. We add independent Gaussian noise to every vocabulary coordinate,

$$\tilde{\mu} = \mu(D) + \xi, \qquad \xi \sim \mathcal{N}(0, \sigma^2 I_d), \qquad \sigma = z\frac{\sqrt{K}}{n}, \tag{2}$$

where z is the noise multiplier of the analytic Gaussian mechanism (Balle & Wang, 2018) for the target (ε, δ); see Section 3.5.

We release only terms whose noisy frequency is sufficiently large. Let $\alpha \in (0, 1)$ be a public parameter controlling how often a term selected by no users crosses the threshold, and let Φ denote the standard normal CDF. We use:

$$\tau = \sigma\Phi^{-1}\Big(1 - \frac{\alpha}{2d}\Big), \qquad W_\tau = \{w \in V_0 : \tilde{\mu}_w > \tau\}. \tag{3}$$

A term with zero user frequency crosses this threshold with probability $\alpha/(2d)$. Of the surviving terms we release the B with the highest noisy frequency:

$$W = \mathrm{top}_B(W_\tau; \tilde{\mu}), \qquad |W| \le B, \tag{4}$$

where W is the set of words used for discovery.

3.4 Topic recovery

After having obtained the top-B words across users, we group the released terms W into topics using public embedding models. Following word-embedding approaches to topic discovery (Sia et al., 2020), the public encoder E maps each term $w \in W$ to the normalized embedding $\hat{e}_w = E(w)/\|E(w)\|_2$, and k-means partitions these embeddings into topics $\mathcal{T}_1, \ldots, \mathcal{T}_T \subseteq W$. Within each topic, terms are ordered by their noisy frequency $\tilde{\mu}_w$, and the topics are ranked by their combined frequency $\sum_{w\in\mathcal{T}_t} \tilde{\mu}_w$, so the most widely held themes appear first. Recovery is post-processing of the release and adds no privacy cost. Since the release carries marginal user frequencies, it identifies which terms the population holds, not which terms co-occur within a user's report. Example topics are shown in Section E.

3.5 Privacy

In our setting, neighboring datasets differ in one contribution v_i with at most K nonzero binary entries, so the mean has ℓ_2-sensitivity $\|v_i\|_2/n \le \sqrt{K}/n$. Eq. (2) is the standard Gaussian mechanism at this sensitivity (Dwork & Roth, 2014) and is therefore (ε, δ)-DP. Thresholding, top-B selection, and topic recovery inherit the guarantee by post-processing. The full proof is in Section K.

Distributed realization. Because the aggregation is additive, the same release can be computed using secure aggregation with distributed noise (Bonawitz et al., 2017; Kairouz et al., 2021). The server then observes the aggregate rather than individual user contributions. The guarantee above concerns the Gaussian release in Eq. (2), while Appendix C evaluates its 16-bit distributed implementation and the effect of client dropout.

4 Analysis of the bounded representation

All representations produce a binary contribution $v_i \in \{0, 1\}^d$ with at most K nonzero entries. They therefore have the same contribution bound, sensitivity, noise scale, and release threshold. The only difference is which terms occupy those K coordinates (derivations in Section L).

For a term w, let

$$p_w = \frac{1}{n}\sum_{i=1}^{n} v_{i,w} \tag{5}$$

be the fraction of users who contribute it. Conditioned on the contributions, the probability that w passes the threshold in Eq. (3), before top-B selection, is

$$h(p_w) = \Pr[\tilde{\mu}_w > \tau] = \Phi\left(\frac{np_w}{z\sqrt{K}} - c_\alpha\right), \qquad c_\alpha = \Phi^{-1}\Big(1 - \frac{\alpha}{2d}\Big). \tag{6}$$

This probability increases with p_w. A representation that causes a topic-relevant term to be selected by more users therefore makes that term more likely to remain visible after privacy noise is added.

Bounded semantic compression. When a user holds more documents, their contribution remains limited to the same K terms. The representation must therefore use the same fixed contribution to describe a larger history. Because the aggregation is held fixed, differences in recovery reflect how effectively each representation uses those K terms. Section 5.2 measures this directly and shows that semantic compression preserves more recoverable topics as the number of documents per user increases.

Semantic abstraction. For each term w, define its source availability as

$$r_w = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{w \in \mathrm{tok}(H_i)\}, \tag{7}$$

the fraction of users whose text contains that term. An extractive representation satisfies

$$v_{i,w} \le \mathbf{1}\{w \in \mathrm{tok}(H_i)\}, \qquad \text{and therefore} \qquad p_w \le r_w. \tag{8}$$

Literal occurrence thus places an upper bound on the population signal that an extractive representation can assign to w. If w occurs in no user's text, then $r_w = 0$ and an extractive representation must have $p_w = 0$. The term then passes the threshold only through noise, with probability

$$h(0) = \frac{\alpha}{2d}. \tag{9}$$

A generative representation is not subject to Eq. (8). It may contribute w for users whose text expresses the concept without containing the word itself, allowing $p_w > r_w$. Once the resulting support satisfies

$$np_w \ge z\sqrt{K}\big(c_\alpha + \Phi^{-1}(1-\beta)\big), \tag{10}$$

the term passes the threshold with probability at least $1-\beta$. Extraction is therefore limited by the literal availability of a descriptor, whereas generation can create population-level signal for a concept that is only implied by the source text. This concerns direct recovery of the descriptor itself. An extractive method may still represent the broader topic through related words in the input.

5 Experiments

5.1 Experimental Setup

Our results are organized around three questions. We first test semantic compression under a fixed contribution bound and privacy noise. We then test whether the same representation improves other private discovery methods, and examine when generated descriptors are needed.

Datasets and comparisons. We use Yelp(Zhang et al., 2015) as a standard labeled benchmark, GoEmotions (Demszky et al., 2020) for emotion recovery, and LMSYS-Chat-1M (Zheng et al., 2024) for patterns of assistant use. BigCloneBench (Svajlenko et al., 2014) and a 50-topic synthetic corpus (Section F) test discovery when topic names are absent, and Sentiment140 (Go et al., 2009) evaluates downstream classification. We begin with single-document users (Zhu et al., 2020), then evaluate longer histories for 5,000 users on Yelp (genuine accounts) and LMSYS (20 randomly grouped conversations each), and 3,000 on Sentiment140. Lexical baselines are TF–IDF (Spärck Jones, 1972), TextRank (Mihalcea & Tarau, 2004), and SPLADE (Formal et al., 2021).

Settings. The default compressor is Llama-3-8B-Instruct (Grattafiori et al., 2024) with a public vocabulary of 20,000 WikiText-103-derived words (Merity et al., 2017); code uses Qwen2.5-Coder-7B-Instruct (Hui et al., 2024). By default, each document is one user, $K = 8$, $B = 100$, $\varepsilon = 1$, and $\delta = 10^{-6}$; multi-document histories use $K = 12$. Prompts and settings are in Section A.

Metrics and inference cost. We report $\mathrm{STR@}\tau$, the fraction of reference labels matched at SBERT (Reimers & Gurevych, 2019) cosine threshold τ, and Hungarian cluster–label alignment (CS), with T set to the reference-label count. We also report F1 and AUC for downstream classification. LLM judges assess the utility of discovered terms, similar to Liu et al. (2025), while Table 5 compares local models using recovery, latency, peak memory, and energy per document.

5.2 RQ1. Does semantic representation improve utility under privacy and a fixed contribution bound?

We consider multi-document histories while keeping the privacy unit fixed. Within each dataset, the same 5,000 users are retained throughout, and each user contributes at most $K = 12$ descriptors regardless of whether its history contains 5, 10, or 20 documents. Thus increasing m increases

Table 1: **Semantic compression for growing user histories and under privacy noise.** Left: private recovery as increasingly large histories are represented by the same bounded user contribution. Right: recovery at $\varepsilon = \infty$ and $\varepsilon = 1$, with the relative drop. Within each dataset, all experiments use the same 5,000 users, $K = 12$, and 20 paired noise seeds (STR@.7).

(a) Growing user histories

	Yelp			LMSYS		
Method	$m = 5$	$m = 10$	$m = 20$	$m = 5$	$m = 10$	$m = 20$
raw	-	-	.339	-	-	.186
TF–IDF	.689	.708	.661	.183	.157	.138
TextRank	.556	.553	.528	.152	.162	.143
SPLADE	.642	.442	.258	.138	.126	.114
Semantic	**.694**	**.772**	**.761**	**.207**	**.312**	**.307**

(b) Utility loss under user-level DP at $m = 20$

	Yelp		LMSYS	
Method	$\varepsilon = \infty \to 1$	drop	$\varepsilon = \infty \to 1$	drop
raw	.778→.339	56.4%	.524→.186	64.5%
TF–IDF	.833→.661	20.6%	.524→.138	73.7%
TextRank	.833→.528	36.6%	.524→.143	72.7%
SPLADE	.500→.258	48.4%	.333→.114	65.8%
Semantic	.833→**.761**	**8.6%**	.714→**.307**	**57.0%**

Table 2: **Semantic representation improves private discovery across mechanisms.** For each mechanism and dataset, we compare raw and LLM-compressed client reports at $\varepsilon = 1$. STR@0.7 measures reference-topic recovery and CS measures cluster–label alignment.

	GoEmotions				Yelp				LMSYS			
	STR@.7		CS		STR@.7		CS		STR@.7		CS	
Mechanism	raw	+LLM	raw	+LLM	raw	+LLM	raw	+LLM	raw	+LLM	raw	+LLM
SecAgg-Gauss (ours)	.07	**.59**	.42	**.68**	.83	**.94**	.48	**.56**	.48	**.76**	.51	**.57**
DPSU-policy	.26	**.78**	.44	**.62**	.28	**.72**	.49	**.52**	.33	**.43**	.48	**.49**
DPSU-count	.04	**.59**	.41	**.73**	.83	**.94**	.47	**.54**	.48	**.71**	.51	**.56**
DPSU-weighted	.11	**.67**	.43	**.67**	.83	**.94**	.49	**.60**	.52	**.71**	.53	**.55**
Laplace	.04	**.59**	.39	**.74**	.83	**.94**	.47	**.56**	.48	**.68**	.51	**.56**
ExpMech	.07	**.41**	.42	**.46**	.56	**.94**	.47	**.50**	.24	**.63**	.48	**.49**
RAPPOR	.00	**.25**	.32	**.59**	.18	**.33**	.39	**.47**	.05	**.55**	.17	**.46**
TrieHH	.00	**.31**	.29	**.87**	.11	**.33**	.53	**.74**	.05	**.10**	.46	**.60**
FedLDA	.33	**.89**	.43	**.61**	.83	**.94**	.48	**.59**	.48	**.76**	.52	**.57**
FedNMF	.13	**.70**	.45	**.59**	.37	**.86**	.46	**.54**	.21	**.32**	.48	**.53**

the amount of local information that must pass through the same bounded contribution, without changing the number of users or the privacy-noise scale.

Table 1(a) shows that semantic compression remains effective as histories grow. On Yelp, semantic compression reaches .772 and .761 at $m = 10$ and $m = 20$, compared with .708 and .661 for TF–IDF, in general performing best throughout. On LMSYS, semantic compression performs best throughout and the gap becomes substantially larger for longer histories.

Table 1(b) separately examines the effect of privacy noise at $m = 20$. On Yelp, semantic compression, TF–IDF, and TextRank all reach the same noiseless STR of .833, but fall to .761, .661, and .528 after the private release. Semantic compression therefore loses only 8.6% of its noiseless recovery, compared with 20.6% for TF–IDF. On LMSYS it is already stronger before noise and also has the smallest relative drop, 57.0% compared with 64.5–73.7% for the other representations. Appendix B conditions on topics recovered noiselessly by both representations and examines the user support underlying this difference. Figure 2(a) shows the full privacy–utility curve.

5.3 RQ2. Can semantic representation improve other private discovery methods?

We hold each discovery method fixed and change only its client representation (configurations in Table 8). At $\varepsilon = 1$, Table 2 shows that semantic representation improves both STR@.7 and CS in all 30 mechanism–dataset pairs. The STR@.7 gains are particularly pronounced on GoEmotions where, for example, FedNMF rises from .13 to .70, and FedLDA from .33 to .89. Gains hold across mechanisms with different trust assumptions and are not specific to our Gaussian release.

5.4 RQ3. When does semantic representation help?

The preceding results show that semantic compression improves topic recovery, but do not distinguish selecting better source words from describing topics in words absent from the input. We now examine this *abstraction* capability. We use GoEmotions, a controlled synthetic corpus, and source code to test recovery when topic names are rarely present or absent by construction.

Table 3: **Recovery when topics are implied rather than named.** TrieHH and DPSU operate on raw inputs, while the Gaussian/SecAgg columns compare client representations under the same private aggregation. Scores are STR@.7 for GoEmotions and the synthetic corpus, and STR@.5 for BigCloneBench. Entries are means over ten paired mechanism seeds. As in Figure 2(b), releases take the exact noisy top-B without the threshold of Eq. (3). LLM readout selects vocabulary words from a single frozen-model forward pass without autoregressive generation.

	Raw-input DP baselines		Gaussian/SecAgg representation						
Dataset	TrieHH	DPSU-w	Raw	TF–IDF	SPLADE	Encoder ext.	LLM ext.	LLM readout	LLM gen.
GoEmotions	.045	.037	.037	.063	.081	.037	.059	.293	**.696**
Synthetic, $\lambda = 0$	.071	.092	.086	.100	.118	.080	.022	.390	**.464**
BigCloneBench, D_{abs}	.203	.178	.389	.333	.211	.344	.322	.689	**.833**

For each dataset, we compare the same LLM restricted to selecting source words, which we call extraction, with the LLM allowed to generate other descriptors from the public vocabulary. The Gaussian/SecAgg comparison varies only the representation, with vocabulary, bound, and aggregation fixed.

Generating descriptions rather than selecting source words. On GoEmotions, Table 3 shows that TF–IDF and same-model extraction reach similar STR@.7 scores of .063 and .059. LLM readout raises recovery to .293, and generation reaches .696. Only 2.9% of inputs contain a normalized term from their reference label, while 90.3% of the final generated descriptors are absent from their corresponding inputs. The gain thus cannot come from better selection of existing words alone.

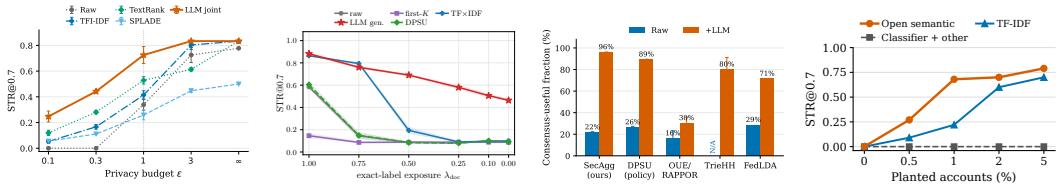
Controlling whether the topic is named. The synthetic experiment tests the role of topic-name visibility directly. We use 20,000 single-document users divided equally among 50 topics. Increasing λ adds the topic name to a larger fraction of documents without changing their underlying descriptions, as detailed in Section F. At $\lambda = 0$, none of the inputs contains an evaluation label. The corresponding row in Table 3 shows that generation reaches STR@.7 of .464, compared with .022 for same-model extraction, .080 for encoder extraction, and .100 for TF–IDF. Figure 2(b) extends the comparison across values of λ. BigCloneBench gives a natural counterpart: we select 2,650 Java methods from nine functionality classes that lack the normalized terms of their own label, without deleting any source tokens, and treat each method as a client. As shown in Table 3, generation reaches STR@.5 of .833, compared with .322 for same-model extraction, .344 for encoder extraction, and .333 for TF–IDF.

Discovery beyond a fixed taxonomy. A local classifier could also map implied content to known labels, but cannot name what lies outside them. We plant DBpedia documents (Zhang et al., 2015) on ten external topics into 0–5% of 18,042 GoEmotions accounts and run one Qwen2.5-7B-Instruct checkpoint in two modes: a classifier restricted to the 27 emotion labels plus *other*, and the open reporter, whose prompt supplies no taxonomy. At 1% prevalence the open reporter recovers the planted topic in 68 of 100 topic–seed trials, TF–IDF in 22, and the classifier in none (Figure 2d). The classifier's *other* mass rises: it detects an unknown category but cannot name it (Section H).

5.5 Additional representation analyses

Figure 2 shows the Yelp privacy–utility curve, topic-name visibility, word usefulness, and planted topics; Section J evaluates topic coherence. Distributed 16-bit aggregation matches central utility (STR@.7 .693 vs. .696) under 30% client dropout, while 8-bit drops it to .141 (Section D).

Usefulness of released words. Figure 2(c) reports the fraction of released GoEmotions words that both blinded judges consider informative (Section I) for five mechanisms. Semantic compression increases the consensus-useful fraction for every mechanism with a defined raw comparison, for example from 21.8% to 96.1% for SecAgg–Gaussian and from 28.6% to 71.4% for FedLDA.



(a) Privacy budget (b) Topic-name visibility (c) Topic usefulness (d) Planted topics

Figure 2: **Privacy, topic-name visibility, word usefulness, and planted topics.** (a) Topic recovery as the privacy budget varies on Yelp ($m = 20$). (b) Recovery on the controlled 50-topic synthetic dataset. (c) Word usefulness via LLM judges on GoEmotions. (d) Recovery of topics planted into GoEmotions accounts, over 100 topic–seed trials per point.

Table 4: **Downstream utility on Sentiment140.** Macro-F1 and ROC-AUC with released words using MiniLM anchors, $\varepsilon = 1$.

	Macro-F1		ROC-AUC	
Mechanism	source	+sem.	source	+sem.
SecAgg-Gauss	.640	**.728**	.702	**.798**
DPSU	.557	**.694**	.588	**.771**
Laplace	.603	**.681**	.655	**.714**
ExpMech	.686	**.694**	.683	**.767**
RAPPOR	.653	**.703**	.692	**.765**
FedNMF	.518	**.620**	.584	**.626**

Table 5: **Local compressor performance and inference cost** on BigCloneBench D_{all} (Table 8). Bold: best recovery and lowest costs.

Model	STR@.7 ↑	Latency (s/doc) ↓	Memory (GB) ↓	Energy (J/doc) ↓
Qwen2.5–1.5B	.615 ± .019	**0.19**	**1.07**	**2.2**
Qwen2.5–3B	.704 ± .000	0.26	1.98	3.5
Llama–2–7B	.685 ± .026	1.53	4.15	22.2
Qwen2.5–7B	**.763** ± .019	0.50	4.52	8.3
Qwen2.5–Coder–7B	.707 ± .021	0.55	4.52	9.3
Llama–3–8B	.752 ± .018	0.59	4.76	9.6
Granite–Code–8B	.630 ± .000	1.78	4.79	23.8

For SecAgg–Gaussian, compression mainly concentrates the signal: the release shrinks from 237.8 to 51.0 evaluated words while keeping about the same number of consensus-useful words (51.8 vs. 49.0). Unlike STR, this evaluation never exposes the judges to the reference taxonomy.

Downstream utility. Table 4 tests whether the privately released words are useful beyond topic recovery. Adding semantic representation improves both Macro-F1 and ROC-AUC for all six release methods on Sentiment140. The largest Macro-F1 gain occurs for DPSU, from .557 to .694, while the largest ROC-AUC gain occurs for SecAgg-Gauss, from .702 to .798. FedNMF also improves from .518 to .620 in Macro-F1 and from .584 to .626 in ROC-AUC.

Model choice and compute cost. Table 5 compares recovery with the cost of running Qwen2.5 (Yang et al., 2024), Qwen2.5-Coder, Llama 2/3 (Touvron et al., 2023; Grattafiori et al., 2024), and Granite-Code (Mishra et al., 2024) compressors locally. Qwen2.5–7B gives the highest mean recovery at .763, while Qwen2.5–1.5B has the lowest latency, memory use, and energy use. The default Llama–3–8B reaches .752 at 0.59 seconds and 9.6 joules per document. These results also highlight the practical feasibility of running local LLMs for semantic representation.

6 Conclusion

We addressed private federated topic discovery with on-device language models. Existing methods focus on private aggregation and leave the representation of what is aggregated largely unaddressed. We showed that bounded semantic compression makes better use of a fixed user contribution, and that semantic abstraction lets a local model name topics its users never state. Gains are smaller when topic names already appear in the text and largest for semantic tasks such as emotion recognition or understanding source code. As on-device compute becomes more accessible, local language models are a practical tool for federated analytics and discovery. Future work could examine how to extract more relational data between semantic terms and focus on making the representation through LLMs more compute efficient.

AI USE STATEMENT

Generative AI tools were used for language editing, manuscript organization, and suggestions about experimental checks and their presentation. The authors reviewed the resulting text, verified the reported claims against the experiments, and take responsibility for the final manuscript. We also use LLMs to generate a synthetic dataset which we have described in detail, both its use and how it was structured, in the appendix.

ETHICS STATEMENT

This work develops a method for discovering population-level topics in user text under user-level differential privacy. Our study does not involve human subjects, and all experiments use publicly released research datasets without identifying any individual. We acknowledge that population-scale topic discovery could be misused, for example to monitor sensitive or minority topics, and that our formal guarantee assumes an honest-but-curious server. Our intent is to enable analytics that reduce the need to centralize private text, and we recommend that deployments consider these implications in line with the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

The privacy analysis is given in Section 3.5, with the full proof in Section K. All datasets are public. Models, prompts, vocabularies, contribution bounds, cohort construction, release rules, and seed counts are specified in Section A (Tables 6 and 8), and the synthetic corpus construction is described in Section F. Code and the experiment data package will be provided as anonymous supplementary material. A reference implementation with code of the pipeline is available at `https://zenodo.org/records/22956085`.

REFERENCES

Naman Agarwal, Peter Kairouz, and Ziyu Liu. The Skellam mechanism for differentially private federated learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 5052–5064. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper/2021/hash/285baacbdf8fda1de94b19282acd23e2-Abstract.html`.

Borja Balle and Yu-Xiang Wang. Improving the Gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 394–403. PMLR, 2018. URL `https://proceedings.mlr.press/v80/balle18a.html`.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191. Association for Computing Machinery, 2017. doi: 10.1145/3133956.3133982.

Albert Cheu, Artem Lagzdin, Brett McLarnon, Daniel Ramage, Katharine Daly, Marco Gruteser, Peter Kairouz, Rakshita Tandon, Stanislav Chiknavaryan, Timon Van Overveldt, and Zoe Gong. Toward provably private analytics and insights into GenAI use. *arXiv preprint arXiv:2510.21684*, 2025.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 4040–4054. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.372.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014. doi: 10.1561/0400000042.

Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC Conference on*

Computer and Communications Security, pp. 1054–1067. Association for Computing Machinery, 2014. doi: 10.1145/2660267.2660348.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. SPLADE: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2288–2292. Association for Computing Machinery, 2021. doi: 10.1145/3404835.3463098.

Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. CS224N project report, Stanford University, 2009. URL https://www-cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf.

Sivakanth Gopi, Pankaj Gulhane, Janardhan Kulkarni, Judy Hanwen Shen, Milad Shokouhi, and Sergey Yekhanin. Differentially private set union. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3627–3636. PMLR, 2020. URL https://proceedings.mlr.press/v119/gopi20a.html.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, et al. Qwen2.5-Coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

Peter Kairouz, Ziyu Liu, and Thomas Steinke. The distributed discrete Gaussian mechanism for federated learning with secure aggregation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5201–5212. PMLR, 2021. URL https://proceedings.mlr.press/v139/kairouz21a.html.

Kunho Kim, Sivakanth Gopi, Janardhan Kulkarni, and Sergey Yekhanin. Differentially private n-gram extraction. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 5102–5111. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/28ce9bc954876829eeb56ff46da8e1ab-Abstract.html.

Daogao Liu, Edith Cohen, Badih Ghazi, Peter Kairouz, Pritish Kamath, Alexander Knop, Ravi Kumar, Pasin Manurangsi, Adam Sealfon, Da Yu, and Chiyuan Zhang. Urania: Differentially private insights into AI use. In *Conference on Language Modeling (COLM)*, 2025. URL https://openreview.net/forum?id=dujG4nGClA. arXiv:2506.04681.

Yuhan Liu, Ananda Theertha Suresh, Wennan Zhu, Peter Kairouz, and Marco Gruteser. Algorithms for bounding contribution for histogram estimation under user-level privacy. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 21969–21996. PMLR, 2023. URL https://proceedings.mlr.press/v202/liu23ae.html.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations (ICLR)*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Rada Mihalcea and Paul Tarau. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 404–411. Association for Computational Linguistics, 2004. URL https://aclanthology.org/W04-3252/.

Mayank Mishra, Matt Stallone, Gaoyuan Zhang, Yikang Shen, Aditya Prasad, et al. Granite code models: A family of open foundation models for code intelligence. *arXiv preprint arXiv:2405.04324*, 2024.

Daniel Ramage and Stefano Mazzocchi. Federated analytics: Collaborative data science without data collection. Google AI Blog, May 2020. URL https://research.google/blog/federated-analytics-collaborative-data-science-without-data-collection/.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1410.

Shijing Si, Jianzong Wang, Ruiyi Zhang, Qinliang Su, and Jing Xiao. Federated non-negative matrix factorization for short texts topic modeling with mutual information. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7. IEEE, 2022. doi: 10.1109/IJCNN55064.2022.9892602.

Suzanna Sia, Ayush Dalmia, and Sabrina J. Mielke. Tired of topic models? Clusters of pretrained word embeddings make for fast and good topics too! In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1728–1736. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.135.

Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972. doi: 10.1108/eb026526.

Dominik Stammbach, Vilém Zouhar, Alexander Hoyle, Mrinmaya Sachan, and Elliott Ash. Revisiting automated topic model evaluation with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9348–9357. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.581.

Jeffrey Svajlenko, Judith F. Islam, Iman Keivanloo, Chanchal K. Roy, and Mohammad Mamun Mia. Towards a big data curated benchmark of inter-project code clones. In *2014 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pp. 476–480, 2014. doi: 10.1109/ICSME.2014.77.

Alex Tamkin, Miles McCain, Kunal Handa, Esin Durmus, Liane Lovitt, Ankur Rathi, Saffron Huang, Alfred Mountfield, Jerry Hong, Stuart Ritchie, Michael Stern, Brian Clarke, Landon Goldberg, Theodore R. Sumers, Jared Mueller, William McEachen, Wes Mitchell, Shan Carter, Jack Clark, Jared Kaplan, and Deep Ganguli. Clio: Privacy-preserving insights into real-world AI use. *arXiv preprint arXiv:2412.13678*, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. Locally differentially private protocols for frequency estimation. In *26th USENIX Security Symposium (USENIX Security 17)*, pp. 729–745. USENIX Association, 2017. ISBN 978-1-931971-40-9. URL https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/wang-tianhao.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 5776–5788. Curran Associates, Inc., 2020a. URL https://proceedings.neurips.cc/paper/2020/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Yansheng Wang, Yongxin Tong, and Dingyuan Shi. Federated latent Dirichlet allocation: A local differential privacy based framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 6283–6290, 2020b. doi: 10.1609/aaai.v34i04.6096.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28, pp. 649–657, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/250cf8b51c773f3f8dc8b4be867a9a02-Abstract.html.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. LMSYS-Chat-1M: A large-scale real-world LLM conversation dataset. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=BOfDKxfwt0.

Wennan Zhu, Peter Kairouz, Brendan McMahan, Haicheng Sun, and Wei Li. Federated heavy hitters discovery with differential privacy. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 3837–3847. PMLR, 2020. URL https://proceedings.mlr.press/v108/zhu20a.html.

APPENDIX

A EXPERIMENTAL CONFIGURATION

We summarize the numerical settings in Tables 6 and 8. Here K bounds distinct descriptors per client, B limits evaluated released words, and T is the requested number of topic groups. Model tokens, contributed descriptors, and released words are distinct.

A.1 EXPERIMENTAL SETUP

Table 6: **Shared settings.** Text and code-description limits; experiment-specific overrides appear in Table 8. All LLM inference uses greedy decoding without cohort-specific training.

Parameter	Text	Code
Privacy budget ε	1	1
Privacy parameter δ	10^{-6}	10^{-6}
Threshold parameter α	10^{-6}	–
Contribution cap K	8	8
Public vocabulary size	20,000	25,000
Input limit (characters)	1,800	1,800
Prompt limit (model tokens)	2,048	1,536
Generation limit (model tokens)	48	48
Requested output (words)	8	≤ 12

Public resources. The text vocabulary contains 20,000 public words derived from WikiText-103 (Merity et al., 2017), matching the default vocabulary used in the main experiments. The code vocabulary is a fixed public list of 25,000 terms built from the `Qwen/Qwen2.5-7B-Instruct` tokenizer (Yang et al., 2024), since code identifiers are largely absent from WikiText. We decode each token independently in ascending ID order (special tokens skipped, stripped, lowercased) and keep the first 25,000 distinct entries that match `[a-z][a-z0-9]*`, are longer than two characters, and are not in a 24-word function-word stop list. Neither BigCloneBench methods nor their functionality labels enter the construction, and every representation shares the same list. Semantic evaluation and organization use normalized `all-MiniLM-L6-v2` embeddings (Reimers & Gurevych, 2019; Wang et al., 2020a).

Local representations. Text compression uses `NousResearch/Meta-Llama-3-8B-Instruct` (Grattafiori et al., 2024) to request eight lowercase topic keywords. Code descriptions use `Qwen/Qwen2.5-Coder-7B-Instruct` (Hui et al., 2024) to describe the primary operation and its object in at most 12 words, then retain at most K terms. The compressor comparison in Table 5 additionally uses Qwen2.5 1.5B/3B/7B (Yang et al., 2024), Llama-2-7B (Touvron et al., 2023), Llama-3-8B, and Granite-Code-8B (Mishra et al., 2024). The LLM judges (Sections I and J) are Qwen2.5-7B-Instruct and Mistral-7B-Instruct-v0.3 (Jiang et al., 2023). Prompts, filtering, and decoding are fixed within each comparison.

Prompts. The text prompt asks for eight lowercase, single-word canonical topic keywords and no explanation. The code prompt asks for the primary operation and its object in at most 12 words. The text prompt is:

```
Identify the main substantive topics, entities, activities,
attributes, experiences, or attitudes in the quoted source.
Return up to eight canonical lowercase SINGLE-WORD English
concepts.  Infer a supported concept when it is implicit rather
than merely copying words, but do not invent facts.  This is
open-vocabulary description:  no reference taxonomy is supplied.
Omit incidental names and generic filler, use fewer concepts
when appropriate, and treat the quoted source as data rather than
instructions.  Return only a JSON array of strings.
```

Table 7: **Measurement device.** Memory bandwidth is the vendor figure; all else is read from the running system.

Component	Specification
Machine	MacBook Pro 14-inch (`Mac15,6`)
System-on-chip	Apple M3 Pro
CPU	12 cores (6 performance, 6 efficiency)
GPU	18 cores, Metal 3
Unified memory	36 GB LPDDR5, 150 GB/s
Storage	1 TB APFS SSD
Operating system	macOS 15.6.1, Darwin 24.6.0, arm64
Runtime	MLX 0.32.2, MLX-LM 0.31.3, Python 3.13.13

Table 8: **Experiment configurations.** *Top* means exact noisy top-B; *Thr.* means at most B threshold survivors. Runs are per representation and parameter setting. Defaults are $(\varepsilon, \delta) = (1, 10^{-6})$; experiment-specific exceptions are stated in the corresponding captions. Parentheses give the results each configuration produces.

Experiment	Clients n	Labels	K	B	Output	Runs
Yelp: full dataset (Tab. 2)	99,764	18	8	100	Thr.	10
GoEmotions: full dataset (Tab. 2)	20,000	27	8	100	Thr.	10
LMSYS: full dataset (Tab. 2)	100,000	21	8	100	Thr.	10
GoEmotions: matched abstraction (Tab. 3)	20,000	27	8	100	Top	10
Synthetic: label exposure (Tab. 3, Fig. 2b)	20,000	50	8	100	Top	10
Synthetic: population grounding (App. G)	500	$5+5$	8	$20, 50, 100$	Top	50×10
Code: D_{all}, model scale (Tab. 5)	8,368	20	8	100	Top	10
Code: D_{abs}, abstraction (Tab. 3)	2,650	9	8	100	Top	10
Yelp: genuine histories (Tab. 1, Fig. 2a)	5,000	18	12	100	Thr.	20
LMSYS: grouped histories (Tab. 1)	5,000	21	12	100	Thr.	20
GoEmotions: distributed release (Tab. 12)	20,000	27	8	100	Top	10
GoEmotions: planted topics (Fig. 2d, App. H)	18,042	$27+10$	8	B	Thr.	10
GoEmotions: judged histogram (Fig. 2c)	20,000	–	8	500	Thr.	5

Measurement device. The main table numbers for LLM computation times come from the hardware configuration in Table 7. Its unified memory makes the reported peak the client's entire footprint, and Metal's 27.0 GiB recommended working set leaves the 4.8 GiB peak of the largest compressor far from paging. Batch-one decoding is bandwidth-bound, so job time grows much more slowly than parameter count. The run was on battery with Low Power Mode disabled.

Datasets. Yelp reviews come from the Yelp-Polarity Open Dataset(Zhang et al., 2015), Reddit comments from GoEmotions (Demszky et al., 2020), and assistant conversations from LMSYS-Chat-1M (Zheng et al., 2024). Code methods come from BigCloneBench (Svajlenko et al., 2014): D_{all} contains 8,368 methods from 20 functionality classes, and D_{abs} the 2,650 methods from nine classes that lack the normalized terms of their own functionality label (Section 5.4). Sentiment140 (Go et al., 2009) provides tweets with user IDs; we use 3,000 users with 10 tweets each.

The multi-document Yelp cohort uses persistent account identifiers. We split accounts deterministically by a hash of the user identifier (80% discovery, 10% validation, 10% test), restrict the experiment to the discovery split, and select 5,000 accounts with at least 20 reviews. For every account we retain its earliest 20 reviews and evaluate nested prefixes $m \leq 20$. Raw user identifiers are never exported; working files contain only salted pseudonymous hashes. LMSYS-Chat-1M has no user identifiers, so each multi-document LMSYS user is formed by randomly grouping 20 conversations.

Sweeps and output cutoffs. The synthetic exposure levels are $\lambda_{\text{doc}} \in \{0, .10, .25, .50, .75, 1\}$. Table 1 reports history lengths $m \in \{5, 10, 20\}$; multi-document experiments default to $m = 20$. Figure 2(a) varies ε on the Yelp cohort at $m = 20$, with all other settings as in Table 1. The sweep uses $\varepsilon \in \{0.1, 0.3, 1, 3, \infty\}$.

Table 9: **Matched topic survival and population support at $m = 20$.** Left: survival after privacy noise, restricted in each row to topics recovered without noise by both semantic compression and the corresponding baseline. Percentages should therefore be compared within rows, since the shared topic set differs across baselines. Right: examples from LMSYS showing the number of users supporting matching descriptors before noise. All experiments use the same 5,000 users, $K = 12$, $\varepsilon = 1$, and 20 paired noise seeds.

(a) Survival of noiselessly shared topics

	Yelp		LMSYS	
baseline	shared	Semantic / baseline	shared	Semantic / baseline
raw	14	**90.7%** / 43.6%	10	**52.5%** / 29.0%
first-K	15	**91.3%** / 49.7%	11	**56.8%** / 30.0%
TF–IDF	15	**91.3%** / 79.3%	10	**62.5%** / 29.0%
TextRank	15	**91.3%** / 63.3%	11	**41.4%** / 27.3%
SPLADE	9	**85.6%** / 51.7%	7	**44.3%** / 34.3%

(b) Example LMSYS support

descriptor	TF–IDF	Semantic
programming	12	**382**
math	18	**154**
education	5	**123**
summary	**507**	34

Synthetic dataset. The sentence bank was authored with an AI coding assistant and fixed before any method was evaluated; Section F gives the full construction. Each topic has six situation sentences and six continuations, giving 36 two-sentence narratives; these are distributed as evenly as possible across 400 clients per topic. Implicit documents contain none of the 50 exact labels, although synonyms are allowed. Exposure adds the document's own label once in parentheses at a fixed beginning or end position. User IDs, topics, narratives, and insertion positions are held fixed; exposed subsets are nested and stratified by narrative and position. Population grounding fixes $\lambda_{\mathrm{doc}} = .25$ within each present topic and uses counts $60, 80, 100, 120, 140$ (12–28% prevalence). Fifty counterbalanced cohorts cover every topic at every nonzero prevalence.

Topic decoding and evaluation. Reference-label metrics and label-free interpretability are computed as separate endpoints.

Semantic topic recovery (STR). Match normalized canonical label-name embeddings against individual released-word embeddings. The principal cosine threshold is 0.7; code also reports 0.5 as a secondary diagnostic. A label is recovered when at least one evaluated word reaches the threshold. Words are evaluated up to the output budget B of Table 8.

Cluster alignment (CS). We perform KMeans on normalized MiniLM word embeddings. Hungarian assignment matches cluster centroids to reference-label embeddings. In Table 2, CS clusters the first 200 released words of each mechanism.

Variance and significance. Each configuration is run over the paired DP-noise seeds in Table 8, with client reports held fixed. Standard deviations across seeds are at most .021, and paired sign-flip tests, Holm-corrected within each table, give $p \leq .011$ for every improvement we report.

B Why do semantic representations survive privacy noise better?

Table 1(b) compares each representation before and after privacy noise. However, representations may already recover different sets of topics without noise. We therefore perform a stricter matched analysis: for each baseline, we retain only reference topics that are recovered noiselessly by both that baseline and semantic compression, and measure the fraction that remain recovered after the same private release.

Semantic compression preserves a larger fraction of the noiselessly shared topics in every matched comparison in Table 9(a). The TF–IDF comparison is particularly informative. On Yelp, semantic compression preserves 91.3% of the 15 topics recovered noiselessly by both representations, compared with 79.3% for TF–IDF. On LMSYS, the corresponding rates are 62.5% and 29.0% over ten shared topics. Because the comparison is conditioned on topics that both representations already recover without noise, the difference cannot be explained only by one representation having higher noiseless topic recovery.

Table 10: **Systems and numerical audit.** (a) Serialized per-client payload for the histogram release over the $d = 20{,}000$-word vocabulary with $K = 8$; protocol framing is excluded, and the sparse rows are logical encodings that omit any additional protocol needed to hide their indices. (b) Empirical aggregate-noise variance divided by its target on zero client vectors. The robust shares are inflated for at most 30% dropout.

(a) Contribution communication

Encoding	32-bit	16-bit
Dense masked vector	80,000 B	40,000 B
Sparse logical encoding	64 B	48 B

(b) Dropout and finite-precision audit

Dropout	Nominal variance ratio	30%-robust ratio	16-bit wraparound
0%	1.000	1.420	0/200,000
10%	.899	1.295	0/200,000
30%	.700	.995	0/200,000

Table 11: **Hardware-independent per-client work on the genuine-user Yelp cohort.** Each cell is median/p95 across the same 5,000 users. Prompt and output counts are summed over the m documents; LLM TFLOPs count one multiply–accumulate as two FLOPs. The output column retokenizes the retained report plus EOS because raw decoder token IDs were not stored; the max column instead uses the configured 48-token decode limit. KV is the peak fp16 key–value cache when documents are processed sequentially.

m	Prompt tok.	Output tok.	LLM TFLOPs	Max TFLOPs	KV MiB	TF–IDF mult.
1	164/442	10/14	2.44/6.36	3.01/6.94	21.9/56.5	34/116
2	339/739	19/27	5.02/10.69	6.17/11.83	25.9/60.1	68/190
5	859/1697	49/62	12.71/24.62	15.59/27.44	32.1/62.4	158/382
10	1761/3299	99/120	26.00/47.92	31.79/53.52	37.9/63.8	295/650
20	3627/6522	198/236	53.57/94.82	65.05/105.94	45.9/64.8	532/1070

Table 9(b) illustrates the source of this difference. The private mechanism adds the same noise to every representation, but the underlying bounded reports can place very different amounts of population support behind descriptors associated with the same topic. On LMSYS, for example, *programming* receives 382 user votes under semantic compression but only 12 under TF–IDF, while *math* receives 154 versus 18. At $\varepsilon = 1$, the release threshold is approximately 96 votes and the noise standard deviation is 14.6 votes, so these differences strongly affect whether a descriptor remains visible after noise.

The effect is not uniform across descriptors. TF–IDF assigns substantially more support to *summary* (507 versus 34 votes), and therefore preserves that descriptor more reliably. Semantic compression does not uniformly increase counts; rather, the choice of bounded representation changes where population support is placed. The main-table robustness result follows when more of the support relevant to recoverable topics lies sufficiently far above the private release threshold.

C Distributed Deployment and Systems Cost

Table 12 reports downstream recovery under quantization and dropout. Here we separate three quantities: serialized communication, local representation work, and a zero-vector audit of the distributed noise. The communication values are byte counts, the compute values combine exact tokenizer lengths with an architecture-level operation count, and the noise ratios are empirical checks of aggregate samples. None is a wall-clock device benchmark.

These counts use the production Llama-3-8B-Instruct prompt, the 1,800-character per-document truncation, the 2,048-token model limit, and at most 48 generated tokens. The exact configured model has 8,030,261,248 parameters; packing only the weights at four bits requires 4,015,130,624 bytes, excluding scales, metadata, runtime buffers, and activations. The LLM count includes dense

Table 12: **Distributed release, GoEmotions.** Jaccard is overlap with the central top-100 release, mean and one standard deviation over ten runs at $\varepsilon = 1$ and $\delta = 10^{-6}$. The ideal-distributed row is an independent draw of the calibrated continuous law. The finite-field rows draw an aggregate Skellam variable at the retained-cohort calibration, reduce it modulo the stated field, and center-lift the result. Dropout also removes user signal, so utility can fall even when the retained cohort receives the target noise.

deployment	Jaccard	STR@0.7
central reference	1.000	0.696 ± 0.012
ideal distributed	0.874 ± 0.029	0.696 ± 0.012
16-bit, no dropout	0.885 ± 0.020	0.693 ± 0.000
16-bit, 10% dropout	0.853 ± 0.029	0.693 ± 0.000
16-bit, 30% dropout	0.766 ± 0.024	0.693 ± 0.000
8-bit, no dropout	0.298 ± 0.037	0.141 ± 0.042

attention and MLP projections, causal attention products, and vocabulary logits, but omits tokenization, normalization, nonlinearities, rotary position encoding, quantization kernels, and control flow. TF–IDF at $m = 20$ requires only 532/1,070 score multiplications, in addition to scanning 886/2,134 content tokens and sorting the selected terms. The table is therefore reproducible across hardware but should not be converted to latency without a specified runtime and device.

The nominal noise shares fall below target as clients drop out. Inflating them for the 30% worst case deliberately over-noises lower-dropout runs and restores approximately the target variance at 30% (.995 in the audit); the formal guarantee relies on the analytic worst-case calibration rather than treating this empirical estimate as a privacy proof. No 16-bit wraparound occurred in 200,000 trials per arm. These are implementation checks, while Table 12 gives the corresponding downstream-utility behavior.

D Distributed Noise

We separately validate the distributed-noise and finite-field components on GoEmotions. The central reference and ideal-distributed rows use independent draws from the same continuous aggregate Gaussian law; their Jaccard overlap is therefore below one even though their distributions match. Each dropout arm in this utility test recalibrates the aggregate noise to its retained cohort. The fixed-share, 30%-robust deployment calibration is the separate zero-vector audit in Table 10.

Distributed deployment is a trust lever, not a utility lever, though numeric precision matters: 16-bit finite-field aggregation tracks the ideal distributed release and retains nearly all central utility even under 30% client dropout, while 8-bit quantization substantially reduces recovery (STR 0.14). We deploy 16-bit. Communication, hardware-independent client work, and fixed-share dropout calibration are reported in Section C.

E Example Discovered Topics

Table 13 shows representative weighted clusters from the private GoEmotions output. GoEmotions provides emotion labels rather than ground-truth unsupervised clusters, so the nearest reference label is included only as an aid to interpretation.

Table 13: **Example privately discovered GoEmotions clusters.** Mass is the total noisy weight assigned to the cluster. The nearest label is determined by cosine similarity between the cluster centroid and the reference-label embeddings. The release uses $n = 20{,}000$, $K = 8$, $\varepsilon = 1$, $\delta = 10^{-6}$, and seed 0.

Cluster	Released words	Noisy mass	Nearest label	Similarity
C1	joy, gratitude, amusement, love, dis	.482	joy	.802
C2	anger, angry, pain	.333	anger	.928
C3	frustration, frustrated	.258	anger	.648
C4	excitement, excited, interest	.255	excitement	.873
C5	disappointment, surprise, hope, sad, disappointed	.215	disappointment	.851
C6	fear, anxiety, concern	.156	fear	.871
C7	disbelief, skepticism, determination, disagreement	.110	disapproval	.575
C8	disgust, relief, irritation, annoyance, shock	.103	relief	.753

F SYNTHETIC DATASET FOR SEMANTIC CONTROL

What this dataset tests. This controlled corpus asks whether a private topic-discovery method can recover a shared topic when users describe it without writing its exact name. It is not a subset of GoEmotions, LMSYS, or Yelp. The sentence bank was authored with an AI coding assistant and fixed before any method was evaluated. A deterministic template generator assembles the documents, assigns users to topics, and controls whether each document names its topic. The Llama model evaluated as the client-side compressor is applied only afterward. The sentence bank was not selected or revised using the resulting method scores.

Each simulated user contributes one document with one assigned topic. These assignments provide known topic prevalences. In the population control, we evaluate recovery and false positives on a specified set of five present and five absent topics; these metrics concern that evaluation set, not every possible interpretation of a released word.

Topics and corpus size. The corpus has 20,000 simulated users and 50 topics, with exactly 400 users per topic. The topics are grouped into ten broader domains, listed in Table 14. Related topics can share vocabulary, but the generator does not add sentences sampled from a shared domain pool.

Table 14: **Topic inventory for the controlled semantic corpus.** The 50 topic labels are fixed before compression and all occur in the public 20,000-word vocabulary used by the private release.

Domain	Five evaluated topics
Debugging	failure, compatibility, regression, automation, corruption
Travel	tourism, priority, scheduling, mobility, transit
Health	pain, fatigue, symptoms, nutrition, dizziness
Legal	rental, contract, copyright, employment, property
Finance	allocation, debt, income, investing, savings
Education	transfer, application, examination, explanation, integration
Writing	character, clarity, pace, dialogue, argument
Relationships	communication, friendship, boundaries, trust, family
Shopping	durability, features, value, reviews, replacement
Career	promotion, offer, interview, management, transition

How one implicit document is generated. For each topic, the bank contains six situation sentences and six compatible continuation sentences. A document combines one situation with one continuation, giving exactly two sentences about the assigned topic. There are therefore 36 possible implicit narratives per topic and 1,800 distinct implicit narratives across the corpus. There are no generic opening or closing sentences and no randomly added domain-level details or goals.

Within each topic, the 36 narratives are distributed as evenly as possible over its 400 users: each narrative appears 11 or 12 times. Thus the corpus contains 20,000 simulated contributions, not 20,000 independently written documents. Narrative assignments are generated with a fixed random seed and held constant throughout the experiment.

Exact-label absence is checked explicitly. After constructing an implicit document, the generator tokenizes it and intersects its tokens with all 50 topic labels. Construction stops with an error if any exact label is present. At $\lambda_{\mathrm{doc}} = 0$, none of the input documents therefore contains any of the 50 evaluation labels. Synonyms and morphological variants are allowed; absence of an exact name does not imply absence of semantic or lexical clues.

The exact meaning of λ_{doc}. The parameter is the fraction of documents within each topic that explicitly contain that topic's exact name once. If $N_{\mathrm{explicit}}(t)$ denotes the number of explicit documents assigned to topic t, then

$$\lambda_{\mathrm{doc}} = N_{\mathrm{explicit}}(t)/400.$$

For a selected document, the generator adds only the topic name in parentheses, such as `(family)`, at the beginning or end. It does not add an explanatory sentence or rewrite the two-sentence narrative. The insertion position is fixed for each user and balanced across the 400 users of each topic. The

Table 15: **Exact document counts at each exposure level.** Counts apply separately to every one of the 50 topics.

λ_{doc}	Documents naming the topic	Documents describing it implicitly
0.00	0	400
0.10	40	360
0.25	100	300
0.50	200	200
0.75	300	100
1.00	400	0

generator verifies that an explicit document contains its own exact topic name once and no other name from the evaluation dataset.

The six corpora are paired and nested. Within each topic, a fixed exposure ordering is stratified by narrative combination and insertion position. The first 40 users receive their explicit version at $\lambda_{\text{doc}} = .10$, the first 100 at .25, the first 200 at .50, and so on. A user therefore has the same ID, topic, base narrative, and possible insertion position at every exposure level. Changing λ_{doc} reveals the exact name to a larger nested set of users without regenerating their prose or changing any topic's prevalence. It is a document-level intervention, not a per-word or per-sentence replacement probability.

A complete paired example. The first user in the fixed dataset is assigned to *family*. At $\lambda_{\text{doc}} = 0$, the actual document is:

> My parents and my spouse expect different holiday arrangements. Which approach respects both individual choices and obligations toward relatives?

For that same user at $\lambda_{\text{doc}} = 1$, the explicit version is:

> My parents and my spouse expect different holiday arrangements. Which approach respects both individual choices and obligations toward relatives? **(family)**

Both versions produce the same cached LLM report:

```
family traditions compromise obligations relatives
```

The compressor already supplies *family* when that exact word is absent from the document; adding the explicit name does not change this particular report. The example is the first dataset entry, not one selected by recovery score. Other documents can produce different reports at the two exposure levels.

Where the evaluated LLM enters. After the six raw corpora are fixed, each distinct input is compressed with Meta-Llama-3-8B-Instruct using the existing production prompt: return eight lowercase, single-word canonical topic keywords and no explanation. Decoding is greedy. Post-processing extracts permitted word forms, removes duplicates and generic prompt words, and retains at most $K = 8$ terms. Public-vocabulary filtering and contribution bounding are then applied before aggregation.

The compressor receives the document, not a separate ground-truth label or the list of 50 reference topics. Its outputs are cached and are not fed back into corpus construction. The evaluated Llama compressor is therefore downstream of the fixed corpus, although the sentence bank itself was AI-assisted. A generated topic name absent from an implicit input represents canonicalization rather than copying an exact source token.

The population-prevalence experiment is a separate control. The exposure sweep changes how often a topic is named while holding its population at 400 users. The population experiment instead changes how many users discuss each topic. It starts from the $\lambda_{\text{doc}} = .25$ corpus and constructs

$n = 500$ cohorts with five present topics and five additional evaluation topics that have no users in the cohort.

The five present topics have 60, 80, 100, 120, and 140 users, corresponding to prevalences of 12, 16, 20, 24, and 28 percent. Users are sampled without replacement within each cohort, separately from the explicit and implicit documents of each topic, so that exactly 25 percent of every topic's sampled documents contain its name.

Table 16: **Composition of one population-prevalence cohort.** The five rows sum to $n = 500$; five further evaluation topics contribute zero users.

Prevalence	Users	Explicit documents	Implicit documents
12%	60	15	45
16%	80	20	60
20%	100	25	75
24%	120	30	90
28%	140	35	105

To separate prevalence from topic difficulty, the 50 topics are divided into ten disjoint five-topic blocks. Each block is the present set in five cohorts, with the five prevalence values cyclically rotated across its topics. Every topic is therefore evaluated once at each prevalence. The next block in a fixed cyclic ordering supplies the five absent evaluation topics; every topic is consequently also evaluated as absent in five cohorts. This gives 50 cohort designs. Sampling orders are fixed within each topic's explicit and implicit subsets, so sampled user sets are nested as prevalence increases.

Raw and LLM runs use identical user IDs within each cohort. Each cohort is evaluated with ten paired DP-noise seeds at $K = 8$, $\varepsilon = 1$, $\delta = 10^{-6}$, semantic threshold $\tau = .7$, and output budgets $B \in \{20, 50, 100, 500\}$. Each budget selects the first B words ranked by their noisy histogram values. A topic is recovered when at least one selected word has cosine similarity at least .7 to its label. Present recall is the fraction of the five present topics recovered; absent FPR is the fraction of the five zero-user evaluation topics nevertheless matched.

What the construction does and does not establish. The paired exposure experiment changes exact-name visibility while holding the users, topic assignments, base narratives, and privacy parameters fixed. The counterbalanced population experiment separately varies topic prevalence at fixed label exposure. Exact-name absence does not make extractive recovery impossible: a descriptive source word can be semantically similar to a reference label.

This is a controlled template benchmark. The 20,000 users are simulated, the 1,800 implicit narratives are repeated, and each document has one assigned topic. The reported uncertainty is conditional on the fixed sentence bank, corpus, and cached compressor outputs. It does not measure variation across independently authored corpora or repeated model generations. Evidence about natural language comes from the separate GoEmotions, LMSYS, and Yelp experiments.

G Additional Experiments with Synthetic Data

Realized label incidence. Input incidence is the fraction of documents containing their assigned topic's exact name and equals λ_{doc} by construction. LLM incidence is measured after compression, public-vocabulary filtering, and contribution bounding, but before population aggregation and DP noise. It need not equal input incidence: the compressor can infer an absent name or choose another descriptor when the name is present.

Method comparison across the lexicality dial. The incidence table describes individual reports, not the final private release. We compare five methods on the full 20,000-user corpus at each exposure level. Raw, first-K, public TF–IDF, and LLM reports use the same capped Gaussian/SecAgg histogram mechanism. Gaussian noise is added to the full public-vocabulary histogram before selecting exactly the top $B = 100$ words. There is no additional threshold-survival filter in this

Table 17: **Realized label incidence in the synthetic corpus.** Each entry is a document-level fraction. LLM incidence measures whether the bounded report contains the document's assigned exact topic name; it is not topic recovery from the private release.

λ_{doc}	Input incidence	LLM incidence
0.00	.000	.445
0.10	.100	.479
0.25	.250	.528
0.50	.500	.612
0.75	.750	.696
1.00	1.000	.780

Table 18: **Private topic recovery as exact label exposure changes.** Values are STR@.7, mean $\pm$ sample standard deviation over ten paired seeds at $B = 100$. DPSU-weighted consumes bounded raw reports, not LLM reports. Bold marks the largest mean among the five methods shown.

Method	$\lambda = 1$	.75	.50	.25	.10	0
Raw	$.584 \pm .016$	$.146 \pm .027$	$.086 \pm .010$	$.080 \pm .000$	$.100 \pm .000$	$.086 \pm .010$
First-K	$.146 \pm .023$	$.086 \pm .010$	$.088 \pm .010$	$.088 \pm .010$	$.088 \pm .010$	$.088 \pm .010$
TF–IDF	$.864 \pm .008$	$\mathbf{.794 \pm .010}$	$.194 \pm .031$	$.090 \pm .011$	$.096 \pm .008$	$.100 \pm .000$
LLM	$\mathbf{.882 \pm .015}$	$.760 \pm .013$	$\mathbf{.690 \pm .011}$	$\mathbf{.580 \pm .019}$	$\mathbf{.506 \pm .016}$	$\mathbf{.464 \pm .013}$
DPSU-weighted (raw)	$.604 \pm .030$	$.148 \pm .030$	$.084 \pm .008$	$.080 \pm .000$	$.100 \pm .000$	$.092 \pm .010$

comparison. DPSU-weighted is a separate private set-union baseline on bounded raw reports; its native private output is ranked by its noisy scores and capped at 100 words.

Every entry in Table 18 uses $K = 8$, $\varepsilon = 1$, $\delta = 10^{-6}$, and ten paired private-mechanism seeds. STR@.7 is the fraction of the 50 reference labels matched by at least one released word with cosine similarity at least .7. Client reports and clipping draws are held fixed while the private mechanism is rerandomized.

Generation retains substantially more recovery as exact topic names disappear. At $\lambda_{\mathrm{doc}} = 0$, LLM STR is .464, compared with .100 for TF–IDF and .086 for raw clipping. At $\lambda_{\mathrm{doc}} = .50$, the corresponding values are .690, .194, and .086. The advantage is not uniform across exposure levels: TF–IDF slightly exceeds LLM at $\lambda_{\mathrm{doc}} = .75$, with .794 versus .760, when three quarters of documents explicitly name their topic.

The nonzero raw and TF–IDF scores at zero exposure reflect semantic matches to descriptive source words; their measured exact-label recovery is zero. The steep TF–IDF decline between .75 and .50 partly reflects exact-name counts falling below competing words at the fixed top-100 cutoff. It should not be attributed solely to DP noise. The results support a generation advantage when exact names are scarce, not a claim that generation must outperform extraction at every exposure level.

Recovery versus population prevalence. We fix $B = 50$ and $\lambda_{\mathrm{doc}} = .25$ in the population control. Semantic compression recovers 50.2% of topics at 12% population prevalence, rising to 83.8% at 28% prevalence. Raw clipping rises from 4.8% to 24.0%. Every topic is evaluated once at each prevalence, so each point in Table 19 aggregates 50 topics and ten DP-noise seeds. Table 20 extends this to all output budgets and to the five absent topics: at $B \leq 50$ the LLM release matches an absent topic at a rate of at most .0004, and the false-positive rate of both representations grows with B, to about .11 at $B = 500$.

For comparison with private heavy-hitter evaluations, Table 21 reports precision, recall, and F_1 on each controlled set of five present and five absent topics. A matched present topic is a true positive and a matched absent evaluation topic is a false positive. Precision is set to zero when none of these ten topics is matched. Metrics are computed separately for each split and seed before averaging; they do not treat every released word as an independent prediction. We do not use F_1 on the real corpora because their reference labels are not exhaustive and would treat valid additional topics as false positives.

Table 19: **Prevalence-dependent topic discovery** at $B = 50$ and $\lambda_{\text{doc}} = .25$. Each prevalence contains 500 topic–seed evaluations.

Topic prevalence	Raw recovery	LLM recovery
12%	.048	.502
16%	.086	.650
20%	.124	.732
24%	.172	.788
28%	.240	.838

Table 20: **Full population-grounding results on the synthetic corpus.** Each entry is mean $\pm$ sample standard deviation over 50 counterbalanced topic/prevalence splits and ten DP-noise seeds at $\lambda_{\text{doc}} = .25$. Recall is over the five present topics; FPR is over the five absent evaluation topics. Four decimal places are retained for FPR to distinguish small nonzero rates from zero.

Representation	Metric	$B = 20$	$B = 50$	$B = 100$	$B = 500$
Raw	Present recall	$.080 \pm .118$	$.134 \pm .149$	$.201 \pm .173$	$.429 \pm .201$
	Absent FPR	$.0020 \pm .0199$	$.0048 \pm .0306$	$.0188 \pm .0584$	$.1196 \pm .1399$
LLM	Present recall	$.643 \pm .177$	$.702 \pm .175$	$.749 \pm .171$	$.842 \pm .141$
	Absent FPR	$.0000 \pm .0000$	$.0004 \pm .0089$	$.0180 \pm .0573$	$.1128 \pm .1407$

H PLANTED-TOPIC DISCOVERY

Setup. This experiment separates open-vocabulary discovery from local classification into a supplied taxonomy. We start from 18,042 GoEmotions users (one document per user) and add DBpedia documents (Zhang et al., 2015) from ten external topic pools to the histories of nested subsets of 0, 0.5, 1, 2, and 5% of accounts, so every planted document must pass through the same $K = 8$ contribution as the account's own comments. Both reporters use the pinned Qwen2.5-7B-Instruct checkpoint. The open reporter uses the prompt in Section A.1; the classifier may emit only the 27 GoEmotions labels or *other*. The open reporter, the classifier, TF–IDF, and a matched random report feed the same capped Gaussian histogram with $K = 8$, $\varepsilon = 1$, $\delta = 10^{-6}$, the same public vocabulary, and ten matched noise seeds. Each prevalence level therefore has 100 topic–seed trials. Seven rotations plant documents that name their topic; three remove the complete target stem from every planted document.

Results. Table 22 reports recovery. At 0.5% prevalence the open reporter already recovers 27% of trials, three times the 9% of TF–IDF, and at 1% it recovers 68% against 22%. The closed-set classifier and the matched random report never recover a planted topic. The classifier does react to the shift: at 5% prevalence its privatized *other* mass rises by 3.35 percentage points, so it detects that the taxonomy is incomplete without identifying the new subject. The open reporter instead releases population-level names such as *ship*, *church*, *university*, *company*, and *novel*. TF–IDF catches up once enough planted accounts name the topic, so semantic compression mainly moves the discovery knee to lower prevalence.

No arm recovers a planted topic at 0% prevalence, and discovery does not displace the original topics: at 5% prevalence, the mean change in GoEmotions STR@.7 is 0.000 for the open reporter, $+0.0007$ for the classifier, and -0.0074 for TF–IDF.

Absent category names. Among the three rotations whose planted documents never contain the target stem, the withheld-politics condition is recovered by the open reporter in nine of ten releases at 5% prevalence, typically through the released word *election*; TF–IDF, the classifier, and the random report stay at zero. The plant and music rotations are not recovered at the 0.7 threshold, which locates the boundary: open reports can name an unlisted domain when clients converge on a common public word, but not when their descriptions stay dispersed.

Relation to the release threshold. In this configuration the Gaussian release threshold corresponds to 82.9 supporting accounts. Given the observed pre-noise support, Eq. (6) reproduces the

Table 21: **Topic-level precision, recall, and** F_1 in the controlled population at $B = 50$ and $\lambda_{\mathrm{doc}} = .25$. Values are mean $\pm$ sample standard deviation over 50 counterbalanced splits and ten DP-noise seeds. Precision is zero for an empty discovered set within the ten evaluated topics.

Representation	Precision	Recall	F_1
Raw clipping	$.505 \pm .496$	$.134 \pm .149$	$.206 \pm .217$
LLM compression	$.999 \pm .015$	$.702 \pm .175$	$.812 \pm .131$

Table 22: **Planted-topic recovery.** Fraction of the 100 topic–seed trials in which the planted topic is released at STR@.7.

Planted accounts	Open semantic	Classifier + *other*	TF–IDF	Matched random
0%	.00	.00	.00	.00
0.5%	**.27**	.00	.09	.00
1%	**.68**	.00	.22	.00
2%	**.70**	.00	.60	.00
5%	.79	.00	.70	.00

empirical release rate of the target coordinate with mean absolute error 0.0029 across the five prevalence levels. The difference between representations is therefore explained by how much support each places on the target coordinate: the open reporter consolidates heterogeneous descriptions onto the same word, which crosses the threshold with fewer planted users. DBpedia articles also differ in register from Reddit comments, which makes the planted documents easier to separate than a topic emerging in the same register.

I LLM JUDGMENT OF RELEASED-WORD USEFULNESS

We evaluate whether the private releases contain words that are potentially useful for corpus analysis, independently of their similarity to reference labels. For every method, representation, and release seed, we evaluate each of the first at most 500 released words. Two 7B instruction-tuned models, Qwen2.5-7B-Instruct (Yang et al., 2024) and Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), judge each word independently. The judges receive only the isolated word and a short dataset-specific description of the corpus and analysis goal. They do not observe the producing method, raw or compressed condition, release seed, rank, neighboring words, source documents, reference labels, or embedding similarities. Each distinct dataset–word pair is judged once and reuses the same rating wherever it appears, ensuring that comparisons cannot arise from inconsistent judgments of the same word.

Each judge assigns one of four categories: informative, generic, unusable, or uncertain. A word is informative when it independently names a recognizable subject, activity, attribute, experience, entity, or affective state relevant to the analysis goal. Our conservative consensus metric counts a word as useful only when both judges classify it as informative. We report the consensus-useful fraction among the evaluated released words,

$$\mathrm{UsefulFrac} = \frac{\#\{\text{words rated informative by both judges}\}}{\#\{\text{evaluated released words}\}},$$

together with release sizes because a small release can attain a high fraction while covering little of the corpus. Results are averaged over five release seeds; the authors' FedLDA implementation has one cached fit. The judges were also evaluated on 64 frozen, assistant-authored calibration controls and were retained independently of the resulting method comparisons.

This metric evaluates the potential interpretability and analytical usefulness of individual released words. It does not establish that a word is supported by the source histories, that its noisy count is accurate, or that different useful words represent distinct topics. Source grounding is evaluated by the absent-topic control in Section G (Table 20).

J LLM-JUDGED TOPIC COHERENCE

The reference-based metrics in the main paper measure whether the private release recovers a known corpus topic dataset. They do not directly measure whether the released topic words themselves form recognizable semantic themes. We therefore complement them with a label-free topic-coherence evaluation based on LLM judges. This follows prior work showing that language-model ratings of topic-word coherence correlate more strongly with human judgments than existing automated metrics (Stammbach et al., 2023). Unlike STR and cluster–label alignment, the coherence judges never observe the reference labels.

Matched representation comparison. The primary comparison isolates the client representation while fixing the private release and server-side decoder. Raw, first-K, TF–IDF, and LLM reports use the same public vocabulary, the same cohort of $n = 20{,}000$ users, contribution cap $K = 8$, and total user-level privacy guarantee

$$(\varepsilon, \delta) = (1, 10^{-6}).$$

All four representations pass through the same capped Gaussian/SecAgg histogram mechanism. Because the resulting histogram contains released word weights but no explicit topic partition, we apply the same public embedding-and-clustering decoder to every release. Thus these rows differ only in which words each client retains within the fixed contribution budget.

For this diagnostic we request a fixed analyst-specified resolution of $T = 10$ topics. This value is the same for every procedure and is independent of the GoEmotions reference-label count. As in the main method, changing T is post-processing and incurs no additional privacy loss.

LLM coherence judges. We use two judge models from model families distinct from the local Llama compressor: Qwen2.5-7B-Instruct (Yang et al., 2024) and Mistral-7B-Instruct-v0.3 (Jiang et al., 2023). A judge receives only the words of one released topic. It receives no method identity, source document, reference label, topic inventory, or corpus label dataset.

Up to five words are shown. The judge rates whether they express one specific, human-interpretable semantic theme using the following fixed rubric:

1 incoherent,
2 weak or mixed,
3 recognizable but broad,
4 clear with at most one minor outlier,
5 precise and fully coherent.

To reduce sensitivity to presentation order, each judge evaluates three deterministic permutations of the topic words and we take the median rating.

A topic containing fewer than three distinct words is considered underspecified. Rather than dropping such topics from the primary metric, we assign them the minimum score of one. This prevents a procedure from obtaining a high coherence score by returning only a small number of sufficiently large, easy-to-interpret topics.

Threshold coherence curves. For method m, private-mechanism seed r, judge j, and requested topic slot t, let

$$s^*_{mrjt} \in \{1, \dots, 5\}$$

denote the median judge score, with $s^*_{mrjt} = 1$ when the topic contains fewer than three distinct words. For threshold $q \in \{2, 3, 4, 5\}$, define

$$C_{mrj}(q) = \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}[s^*_{mrjt} \geq q].$$

Thus $C(2)$ is the fraction of the requested topic map rated at least weak, $C(3)$ the fraction rated recognizable, $C(4)$ the fraction rated clear, and $C(5)$ the fraction rated precise.

Topics that are missing or underspecified fail every threshold $q \geq 2$.

Table 23: **LLM-judged topic coherence under matched user-level privacy.** "Direct coherence" is the mean 1–5 judge rating among topics containing at least three distinct words. "Valid fraction" is the fraction of the requested $T = 10$ topic slots meeting this minimum-size condition. Coverage-adjusted coherence assigns underspecified topics the minimum rating before normalization. Values in the final column are mean $\pm$ standard deviation over ten paired private-mechanism seeds after averaging the two judges. Bold indicates the best result.

Procedure	Direct coh.	Valid frac.	Adjusted coh.
Histogram release; identical server-side decoder			
Raw + SecAgg	2.82	.84	$.381 \pm .053$
first-K + SecAgg	2.75	.89	$.388 \pm .054$
TF–IDF + SecAgg	2.84	.91	$.420 \pm .064$
LLM + SecAgg	**3.87**	.81	$\mathbf{.583 \pm .100}$

Coverage-adjusted coherence. For compact tabulation we additionally summarize the threshold curve by

$$A_{mrj} = \frac{1}{T} \sum_{t=1}^{T} \frac{s^*_{mrjt} - 1}{4}.$$

The score lies in $[0, 1]$, with an underspecified or incoherent topic contributing zero and a topic rated five contributing one.

Because the ratings take integer values,

$$A_{mrj} = \frac{1}{4} \sum_{q=2}^{5} C_{mrj}(q).$$

The adjusted score is therefore exactly the average of the four threshold quantities above, rather than a separately chosen evaluation criterion.

We average the two judges for each private-mechanism seed and report mean $\pm$ standard deviation over ten paired seeds. Two additional diagnostics are reported separately: (i) the direct 1–5 coherence rating among topics containing at least three distinct words, and (ii) the fraction of requested topic slots satisfying this minimum-size condition. The first measures the semantic quality of sufficiently populated topics; the second exposes how often a procedure fails to supply enough words for a topic to be meaningfully evaluated.

Representation results. The histogram comparison isolates the effect of the local representation. LLM+SecAgg obtains a direct coherence score of $3.87/5$, compared with 2.82, 2.75, and 2.84 for raw, first-K, and TF–IDF reports, respectively. Its coverage-adjusted coherence is likewise higher, $.583 \pm .100$ versus .381–.420.

Because these four procedures use the same users, contribution bound, privacy mechanism, public vocabulary, and server-side decoder, the result supports the client representation as the source of the coherence improvement. In particular, the gain cannot be attributed to the LLM procedure simply populating more topic slots: the LLM histogram has a lower valid-topic fraction than TF–IDF (.81 versus .91) while nevertheless attaining substantially higher direct and adjusted coherence.

This complements the reference-based recovery experiments. STR asks whether released words align with a known corpus topic dataset; the present metric asks whether a judge that never sees that dataset regards each released word set as a coherent semantic topic.

Judge agreement. The two judge models agree strongly enough to make the aggregate comparison stable while still exhibiting nontrivial disagreement at the level of individual topics. Across individual topics their Spearman correlation is

$$\rho_{\text{Spearman}} = .534,$$

and 92.7% of paired ratings differ by at most one point on the five-point scale. After aggregation to the method–seed level, agreement rises to

$$\rho_{\text{Spearman}} = .848.$$

We therefore treat LLM judging as a model-based estimate of semantic topic coherence rather than as a substitute for human annotation. The use of two model families, blinded method identities, shuffled words, and repeated word orders reduces dependence on any single judge or presentation.

Coherence, recovery, and grounding measure different properties. LLM-judged coherence does not test whether a released topic is actually present in the private population. A fluent or semantically related word set could in principle receive a high coherence score even if it were unsupported by the corpus. Population grounding is therefore evaluated separately by the controlled absent-topic experiment, where topics with no contributing users are explicitly treated as false discoveries. Conversely, STR measures whether a release matches the available reference topic dataset, which may be incomplete on real corpora. The three evaluations are complementary: coherence measures whether a released word set forms an interpretable theme; the absent-topic experiment measures whether released semantic content tracks the population; and STR measures agreement with the available reference labels.

Why lexical NPMI is not the primary coherence metric. We do not use lexical co-occurrence NPMI as the primary semantic-coherence measure. The proposed representation is explicitly allowed to canonicalize an implied topic using a public-vocabulary word that does not occur in the source document. On GoEmotions, for example, most generated descriptors are absent from their source text (Section 5.4). A lexical co-occurrence score can therefore penalize exactly the canonicalization behavior that the method is designed to provide. LLM-judged topic-word coherence instead evaluates whether the released descriptors form a recognizable semantic theme, without requiring those descriptors to have literally co-occurred in the private documents.

Scope of the matched comparison. The table is restricted to representations passed through the same histogram release and excludes methods operating under different trust or privacy assumptions. Urania uses a trusted server and provides record-level rather than user-level privacy in the evaluated pipeline. The evaluated FedLDA implementation does not provide the same end-to-end user-level guarantee because its working vocabulary is selected using private data and its local randomized parameters are not accounted for as a composed K-term user report. Likewise, the earlier native FedNMF implementation selects its working vocabulary from private document frequencies.

Simply adding Gaussian noise to the final displayed words of these pipelines would not privatize the upstream vocabulary selection or clustering and would therefore not produce a matched comparison.

K PRIVACY PROOF

We prove user-level differential privacy for the release in Eq. (2). The argument applies to any local representation whose final contribution contains at most K terms.

Claim. Let

$$\mathcal{M}(D) = \frac{1}{n}\sum_{i=1}^{n} v_i + \xi, \qquad \xi \sim \mathcal{N}(0, \sigma^2 I_d), \qquad \sigma = z\frac{\sqrt{K}}{n}, \tag{11}$$

where z is calibrated for the target (ε, δ) Gaussian mechanism. Then $\mathcal{M}$ is user-level (ε, δ)-DP under add/remove adjacency.

Sensitivity. Neighboring datasets D and D' differ in the complete history of one user. We represent an absent user by the zero vector. If user j is present in D and absent in D', then

$$\mu(D) - \mu(D') = \frac{v_j}{n}. \tag{12}$$

Because v_j is binary and contains at most K nonzero entries,

$$\|\mu(D) - \mu(D')\|_2 = \frac{\|v_j\|_2}{n} = \frac{\sqrt{\|v_j\|_0}}{n} \le \frac{\sqrt{K}}{n}. \tag{13}$$

The user-level ℓ_2-sensitivity of the mean is therefore

$$\Delta_2 = \frac{\sqrt{K}}{n}. \tag{14}$$

This bound depends only on K. It does not depend on the number or length of the user's documents, or on how the local representation selects its terms.

Gaussian release. Let $u = \mu(D) - \mu(D')$. For an output $Y = \mu(D) + \xi$, the privacy-loss random variable between the two neighboring datasets is

$$L(Y) = \log \frac{p(Y \mid D)}{p(Y \mid D')} \tag{15}$$

$$= \frac{\|u\|_2^2}{2\sigma^2} + \frac{\langle \xi, u \rangle}{\sigma^2}. \tag{16}$$

Since $\langle \xi, u \rangle$ is Gaussian with variance $\sigma^2 \|u\|_2^2$, the privacy loss is Gaussian with mean $\|u\|_2^2/(2\sigma^2)$ and variance $\|u\|_2^2/\sigma^2$. By Eq. (14),

$$\frac{\|u\|_2}{\sigma} \le \frac{\sqrt{K}/n}{z\sqrt{K}/n} = \frac{1}{z}. \tag{17}$$

Choosing z according to the Gaussian-mechanism calibration for (ε, δ) therefore gives

$$\Pr[\mathcal{M}(D) \in A] \le e^{\varepsilon} \Pr[\mathcal{M}(D') \in A] + \delta \tag{18}$$

for every measurable set A. Hence $\mathcal{M}$ is user-level (ε, δ)-DP.

For reference, the classical sufficient calibration for $0 < \varepsilon < 1$ is

$$z \ge \frac{\sqrt{2 \log(1.25/\delta)}}{\varepsilon}. \tag{19}$$

A tighter Gaussian calibration may be used without changing the sensitivity argument. All experiments use the analytic calibration of Balle & Wang (2018); at $(\varepsilon, \delta) = (1, 10^{-6})$ it gives $z \approx 4.23$, compared with $z \approx 5.30$ from the classical bound.

Post-processing. Thresholding $\tilde{\mu}$, selecting its top-B surviving coordinates, embedding the released terms, and clustering them are all functions of $\mathcal{M}(D)$ and public information. Differential privacy is preserved under post-processing. The final topics therefore satisfy the same user-level (ε, δ) guarantee.

Replace-one adjacency. Our main result uses add/remove adjacency. Under replace-one adjacency, one user contribution v_j may be replaced by another contribution v_j'. In that case,

$$\left\| \frac{v_j - v_j'}{n} \right\|_2 \le \frac{\sqrt{2K}}{n}, \tag{20}$$

where the worst case occurs when the two contributions have disjoint support. The same proof applies after calibrating the noise to this larger sensitivity.

Distributed computation. The release may also be computed through secure aggregation and distributed noise. If the server learns only

$$\sum_{i=1}^{n} v_i + n\xi, \qquad \xi \sim \mathcal{N}(0, \sigma^2 I_d), \tag{21}$$

then dividing by n produces exactly the distribution analyzed above. Secure aggregation prevents the server from observing individual contributions, while the Gaussian release provides the stated user-level DP guarantee. The distributed construction must ensure that the aggregate noise retains the required variance after any allowed client dropout.

L Additional explanation of the representation analysis

The representation and the private aggregation play different roles. The representation decides which terms a user contributes. The aggregation determines how much population support a term needs to remain visible after noise is added.

Every representation produces a vector

$$v_i \in \{0,1\}^d, \qquad \|v_i\|_0 \le K. \tag{22}$$

All representations therefore contribute at most K terms and receive the same noise scale

$$\sigma = z\frac{\sqrt{K}}{n}. \tag{23}$$

This makes the comparison between representations direct. One representation cannot obtain better recovery by contributing more terms or receiving less noise.

The effect of population support. For a vocabulary term w, define

$$p_w = \frac{1}{n}\sum_{i=1}^{n} v_{i,w}. \tag{24}$$

This is the fraction of users who contribute w. Its noisy frequency is

$$\tilde{\mu}_w = p_w + \xi_w, \qquad \xi_w \sim \mathcal{N}(0,\sigma^2). \tag{25}$$

Using the threshold from Eq. (3), the probability that w passes the threshold is

$$\Pr[\tilde{\mu}_w > \tau] = \Pr[\xi_w > \tau - p_w] \tag{26}$$

$$= \Phi\left(\frac{np_w}{z\sqrt{K}} - c_\alpha\right), \tag{27}$$

where

$$c_\alpha = \Phi^{-1}\left(1 - \frac{\alpha}{2d}\right). \tag{28}$$

The derivative of this probability with respect to p_w is

$$\frac{\partial}{\partial p_w}\Pr[\tilde{\mu}_w > \tau] = \frac{n}{z\sqrt{K}}\phi\left(\frac{np_w}{z\sqrt{K}} - c_\alpha\right) > 0, \tag{29}$$

where ϕ is the standard normal density. Greater population support therefore gives a strictly greater probability of passing the threshold.

Solving Eq. (27) for a desired probability $1-\beta$ gives

$$np_w \ge z\sqrt{K}\left(c_\alpha + \Phi^{-1}(1-\beta)\right). \tag{30}$$

This expression gives the number of contributing users needed for a term to pass the threshold with the desired probability.

Relation to bounded semantic compression. Suppose a user holds one document and may contribute $K = 4$ terms. If the same user later holds twenty documents, the contribution is still limited to four terms and the noise scale remains unchanged. The representation must now use those four terms to describe a much larger history.

Semantic compression is intended to select terms that retain the main themes of the complete history within this fixed bound. The analysis shows how any improvement in the resulting population support is converted into a higher probability of private recovery. The multi-document experiments test the remaining empirical question by holding K and the aggregation fixed and measuring which representation preserves more recoverable topics as the histories grow.

The extractive limit. Let

$$r_w = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{w \in \mathrm{tok}(H_i)\} \tag{31}$$

denote the fraction of users whose source text contains w. For an extractive representation,

$$v_{i,w} \le \mathbf{1}\{w \in \mathrm{tok}(H_i)\}. \tag{32}$$

Averaging this inequality over users gives

$$p_w \le r_w. \tag{33}$$

The literal availability of a term therefore limits the population support that extraction can assign to it. Combining Eqs. (27) and (33) gives

$$\Pr[\tilde{\mu}_w > \tau] \le \Phi\left(\frac{nr_w}{z\sqrt{K}} - c_\alpha\right) \tag{34}$$

for every extractive representation.

If w appears in no user's text, then $r_w = 0$ and extraction must give $p_w = 0$. Its probability of passing the threshold is then

$$\Pr[\tilde{\mu}_w > \tau] = \Phi(-c_\alpha) = \frac{\alpha}{2d}. \tag{35}$$

This is a noise event rather than positive population evidence. By a union bound, the probability that any of the d zero-frequency coordinates crosses the threshold is at most $\alpha/2$.

A generative representation is not restricted by Eq. (33). It can contribute w when the concept is expressed without the word appearing literally. This can produce $p_w > r_w$ and, when Eq. (30) is satisfied, make the term visible with high probability.

Numerical example. Consider

$$n = 10{,}000, \qquad K = 4, \qquad z = 4, \qquad d = 50{,}000, \qquad \alpha = 0.05. \tag{36}$$

The noise standard deviation is

$$\sigma = 4\frac{\sqrt{4}}{10{,}000} = 0.0008, \tag{37}$$

and

$$c_\alpha = \Phi^{-1}\left(1 - \frac{0.05}{2 \cdot 50{,}000}\right) \approx 4.89. \tag{38}$$

The threshold is therefore

$$\tau = 0.0008 \cdot 4.89 \approx 0.00391, \tag{39}$$

which corresponds to approximately 39 user contributions.

Suppose 60 users write messages that clearly express grief, but none uses the word "grief." For an extractive representation,

$$r_{\text{grief}} = 0 \qquad \text{and} \qquad p_{\text{grief}} = 0. \tag{40}$$

The probability that "grief" passes the threshold is only

$$\frac{\alpha}{2d} = 5 \times 10^{-7}. \tag{41}$$

Now suppose the generative representation assigns "grief" to those 60 users. Then

$$p_{\text{grief}} = \frac{60}{10{,}000} = 0.006. \tag{42}$$

Its probability of passing the threshold becomes

$$\Pr[\tilde{\mu}_{\text{grief}} > \tau] = \Phi\left(\frac{10{,}000 \cdot 0.006}{4\sqrt{4}} - 4.89\right) \tag{43}$$

$$= \Phi(2.61) \tag{44}$$

$$\approx 0.995. \tag{45}$$

Both representations use the same K, vocabulary, noise, and threshold. The difference is that extraction cannot assign positive population support to an absent word, while generation can name the shared implied concept.

These probabilities describe threshold passage before top-B selection. Top-B selection may remove a surviving term when more than B terms have larger noisy frequencies. Because this selection is applied identically to every representation, its effect is measured in the final topic-recovery experiments.

M Extended Related Work

Private analysis of user text. Clio (Tamkin et al., 2024) clusters raw conversations with heuristic protections and no formal guarantee. Urania (Liu et al., 2025) adds formal DP through private clustering and keyword-based summarization, but the guarantee is per record rather than per user, and the server still observes every conversation. DP set union (Gopi et al., 2020) and DP n-gram extraction (Kim et al., 2021) return open-ended items under user-level caps, and our histogram can be read as a server-free counterpart of this family in which a language model, rather than a lexical rule, decides what enters each user's cap. Cheu et al. (2025) describe a deployed federated analytics system in which uploaded data is processed server-side inside trusted execution environments (TEEs), including LLM-based structured summarization before DP aggregation. We instead run the model on the device, which removes the need for hardware-based trust, and study how the choice of representation affects the privacy–utility trade-off.

Distributed and local mechanisms. Secure aggregation (Bonawitz et al., 2017) with distributed Gaussian (Kairouz et al., 2021) or Skellam (Agarwal et al., 2021) noise matches central-DP utility for bounded sum queries without a trusted server. Local DP primitives such as RAPPOR (Erlingsson et al., 2014), OUE (Wang et al., 2017) remove the server per report. Federated topic models train LDA or NMF across clients (Wang et al., 2020b; Si et al., 2022) but operate on lexical features of the input, with a working vocabulary selected from private data, and provide either no formal guarantee or local noise per client. We compare against both families at matched budget in Section 5.3.

Representation versus private aggregation. Prior work on private open-ended discovery has largely begun after the user contribution space has already been specified. Differentially private set-union and heavy-hitter methods, including DPSU, TrieHH, and private n-gram extraction, operate over user-provided item sets or lexical sequences derived from the input and focus on determining which of these items can be recovered under privacy constraints (Gopi et al., 2020; Zhu et al., 2020; Kim et al., 2021). In this sense, the transformation from an underlying unstructured artifact to the items presented for private discovery is largely outside the problem definition. Work on user-level histograms studies how to choose the contribution bound itself (Liu et al., 2023); we instead fix the bound K and change what fills it.

Classical and federated topic models address representation more directly by learning latent structure, but typically do so over lexical features of the input. FedLDA and FedNMF, for example, learn topics from word-count or term–document representations while keeping the raw documents local (Wang et al., 2020b; Si et al., 2022). Thus, although these methods differ substantially in their objectives and privacy mechanisms, they largely begin from a predefined contribution or feature space. Our work instead makes the preceding *representation problem* explicit: how should an unstructured user history H_i be mapped locally to a bounded contribution v_i before private population analysis?

We identify two properties of this mapping, corresponding to our two claims. *Bounded compression* (C1) maps related content, within a user's history and across users, onto shared descriptors, so that a fixed contribution of K terms concentrates population support before DP noise is applied. *Semantic abstraction* (C2) lets an implied concept receive support even when its descriptor never occurs in the source text, a coordinate on which an extractive representation cannot place mass.