What Your Posts Reveal: A Benchmark and Agentic Framework for User-Level Privacy Leakage on Social Media

Anonymous ACL submission

Abstract

Public social media posts can reveal private information through weak cues scattered across text, images, or metadata. Such leakage is often cumulative and cross-post: cues that appear harmless in isolation may jointly expose a user's home, workplace, or routine. However, current research lacks a unified benchmark for user-level multimodal privacy leakage and an evaluation metric that captures exposure severity beyond binary accuracy.

To address these gaps, we propose SopriBench, a synthetic benchmark guided by leakage patterns abstracted from a private reference corpus of Rednote and Instagram accounts, covering 50 user profiles and 1,569 images with attributes, contextual sensitivity, granularity, leakage type, inference difficulty, and supporting evidence. We further introduce the Privacy Exposure Score (PES), which weights value granularity by contextual sensitivity. Inspired by abductive reasoning, we introduce Argus, a training-free agentic framework for cumulative leakage inference. Argus forms hypotheses from accumulated evidence, verifies supporting evidence, and aggregates cross-post cues into privacy profiles, achieving 0.55 PES, a 25% improvement over the strongest baseline, with the largest gain on cross-post leakage.

1 Introduction

Social media users often share daily posts that appear harmless in isolation: a delayed delivery screenshot, a favorite snack shop, a subway station passed on the way home, or a short daily-life caption. Yet privacy leakage on social media is rarely limited to one explicit identifier in one post (Rusert et al., 2019). A small map contour in a delivery screenshot, a nearby restaurant, a recurring commute station, and a city-level IP region may together narrow a user's home or work location (Pontes et al., 2012; Drakonakis et al., 2019). Recent reports make this risk concrete: ordinary visual details, route histories, or repeated location traces can expose residential areas and daily routines (Claburn, 2019; Franceschi-Bicchierai, 2022; Cluley, 2015).

This reveals the core challenge we study: **public social media privacy leakage is often cumulative and cross-post.** Weak cues that are harmless at the post level can become revealing once connected across posts, modalities, and platform context. Such cues may come from captions, images, or metadata, and can jointly support more specific inferences about a user's home address, workplace, routine, or social relationships. An adversary can connect such cues using public posts and common tools such as web or map search, making user-level leakage fundamentally different from post-level disclosure detection.

However, current research has two key gaps. First, there is no unified benchmark for user-level multimodal privacy leakage on public social media. Prior work has studied post-level self-disclosure, PII recognition, image privacy, and profile inference (Luo et al., 2025; Tömekçe et al., 2024; Liu et al., 2025), but these settings do not fully capture public accounts where captions, images, metadata, and repeated routines accumulate across posts. Real-user datasets are hard to release because user labels are sensitive, and methods are difficult to compare at scale.

Second, existing evaluations usually ask only whether an attribute is predicted correctly, which is too coarse to reflect real exposure severity. A city-level location and an exact residential compound should not receive the same exposure score. Besides, the same attribute can carry different privacy sensitivity depending on the user context. For example, a generic inference that a user is heterosexual may carry limited sensitivity in many contexts, while inferring a minority sexual orientation can be substantially more sensitive for the user.

To address these gaps, we introduce **SopriBench**

to measure the problem and Argus to study the risk. SopriBench is a controllable synthetic benchmark guided by leakage patterns abstracted from a private reference corpus of Rednote (Xiaohongshu, 2026) and Instagram (Instagram, 2026) accounts. It contains 50 synthetic users, 500 multimodal posts, and 1,569 images. SopriBench records the ground-truth value, contextual sensitivity, granularity, leakage type, inference difficulty, and post-level supporting evidence. We further introduce **Privacy Exposure Score** (PES), a privacy exposure metric that accounts for both attribute granularity and sensitivity. This distinguishes coarse exposure from fine-grained exposure.

Beyond benchmarking, cumulative leakage also changes the nature of the inference problem. User-level privacy inference is not a one-shot classification task: relevant clues are scattered, ambiguous, and often meaningful only when combined across posts and modalities. We therefore propose **Argus**, a training-free agentic framework inspired by abductive reasoning (Harman, 1965; Jang et al., 2025). Argus forms hypotheses from accumulated evidence, verifies supporting evidence, and aggregates cross-post cues into a privacy profile.

In experiments, Argus achieves a 25% PES improvement over the strongest baseline, SingleAgent (0.44 to 0.55), with the largest gain on cross-post leakage (+0.17 PES). These gains are associated with explicit hypothesis tracking, evidence verification, and cross-post aggregation of weak cues. Ablations further suggest that verification improves exposure quality: removing it increases binary accuracy but lowers PES.

In summary, our contributions are as follows:

- **We identify and study the cumulative nature of user-level privacy leakage on public social media**: privacy risk can arise from weak cues scattered across posts and modalities rather than from explicit disclosure in a single post.
- **We propose new benchmark SopriBench and metric PES**: SopriBench provides a controllable benchmark for user-level privacy leakage, and PES scores exposure severity by combining value granularity with contextual sensitivity.
- **We introduce an new agentic inference framework**: Argus is a training-free framework that treats user-level privacy inference as an abductive reasoning process. Argus maintains explicit hypotheses, evidence, and aggregates cross-post cues through a graph structure.

2 Problem Setup and Related Work

Related work. Existing inference methods (Staab et al., 2024; Lermen et al., 2026; Du et al., 2025) show that large language models (LLMs) and vision-language models (VLMs) can infer private attributes from benign text or images. However, most methods are closer to direct classification or profile summarization than to an evidence-grounded, multi-post investigation process.

Existing datasets cover related but narrower settings. Self-disclosure (Luo et al., 2025) datasets focus on detecting disclosure in individual posts, and synthetic self-disclosure (Jangra et al., 2025) data remain largely post-level. PII (Shahariar et al., 2026; Tömekçe et al., 2024) and visual privacy (Tsaprazlis et al., 2025) benchmarks evaluate identifiable items or image-level risks, while Holmes (Liu et al., 2025) studies private albums. However, they do not provide a releasable benchmark for public social media where weak cues accumulate across posts. Besides, their evaluation is also mostly binary and does not distinguish coarse guesses from sensitive or fine-grained exposure.

Task formulation. We study user-level privacy inference from public social media profiles. For a user u, we denote the observable public content as

$$\mathcal{P}_u = \{p_i\}_{i=1}^{N_u}, \quad p_i = (t_i, v_i), \tag{1}$$

where p_i is the i-th post, t_i denotes all textual content such as captions, IP region, hashtags, and posting time, and v_i denotes all visual content such as images and avatars. Let $\mathcal{A}$ denote the private-attribute schema. Given $\mathcal{P}_u$ and $\mathcal{A}$, the task is to infer a user-level privacy profile:

$$\{(a_j, \hat{y}_j)\}_{j=1}^{K}, \quad a_j \in \mathcal{A}, \tag{2}$$

where a_j is an attribute and $\hat{y}_j$ is the inferred value.

Threat model. We consider an external adversary who has access only to publicly available social media content. The adversary cannot use private databases, account backends, exploit the platform, or other non-public platform logs. However, the adversary may inspect all public posts from a user, aggregate clues across posts, and call common tools such as OCR, image understanding models, web search, map search, or geolocation tools. This setting reflects practical privacy risks from ordinary public social media exposure while limiting the adversary to publicly observable information and generally available tools.

3 SopriBench Construction

We construct SopriBench in three parts: a private real-user reference corpus for pattern abstraction, a releasable synthetic benchmark for evaluation, and PES for sensitivity-aware exposure scoring. Figure 1 summarizes the construction; detailed construction procedures and data quality control are provided in Appendix Section A.

3.1 Private Real-user Corpus

We use a private reference corpus to understand how privacy-relevant cues appear in public social media posts and to derive aggregate generation patterns for SopriBench. We choose Rednote and Instagram because both support public visual-textual lifestyle sharing. We collect public non-public-figure personal accounts through risk-guided search over six broad categories, then filter for accounts with ordinary life posts and rich contextual cues (see details in Section A.1). The final corpus covers 161 Rednote users, 157 Instagram users, 11,243 posts, and 46,771 images. Annotators then abstract the retained posts into de-identified patterns, including posting scenarios, clue carriers, leakage forms, visual styles, and cross-post relations. The corpus is used only for internal pattern abstraction; no raw posts, images, usernames, profile URLs, or inferred real-user profiles are released or copied into SopriBench.

3.2 Synthetic Data

Guided by the real-user corpus, we construct SopriBench, a controllable multimodal benchmark for user-level privacy inference on social media. SopriBench is designed for controlled evaluation rather than estimating the platform-wide prevalence of privacy leakage. It contains 50 synthetic users, 500 posts, and 1,569 images. For each attribute, SopriBench records the attribute type, ground-truth value, contextual sensitivity, granularity, leakage type, inference difficulty, and supporting evidence. Table 1 summarizes the main dataset statistics.

Some synthetic cues use real public places or address regions so that map and web verification remain executable, but these locations are independent from the private reference corpus and are not copied from, or linked to, any retained real user.

Synthesis process. The synthesis process starts from de-identified patterns abstracted from the private corpus and then has two main generation phases. In the text phase, Gemini 3.1 Pro (DeepMind, 2026) first generates a coherent hidden user profile with 28 privacy-relevant attributes. It then creates a leakage plan that selects which attributes are exposed and decomposes each exposed attribute into clue fragments assigned to specific posts, modalities, and carrier types. Finally, the text phase writes 10 ordinary post scripts per user, where the planned privacy cues appear as incidental details rather than as explicit benchmark prompts. In the image phase, the post scripts are realized as multimodal posts. Generated images are checked against the leakage plan so that planned visual or OCR cues are present, contextually plausible, and not overly salient. Seven annotators then revise the generated images to improve realism, preserve planned cues, and remove accidental identifiers.

Property	Value
Synthetic users / posts / images	50 / 500 / 1,569
Total / Leaked attributes	1393 / 781
Mean leaked attributes per user	15.62
Leakage type (%)	explicit / implicit / cross-post: 8.0 / 28.6 / 63.4
Difficulty (%)	D1 / D2 / D3 / D4: 1.4 / 13.8 / 34.1 / 50.7
Sensitivity (%)	L1 / L2 / L3 / L4 / L5: 9.3 / 41.9 / 26.2 / 10.8 / 11.8
Post category (%)	Info / Ent. / Soc. / Self.: 27.0 / 29.6 / 7.8 / 35.6

Table 1: Dataset statistics. Post categories denote information sharing, entertainment, social interaction, and self-expression.

Quality control. We apply automatic and manual checks to ensure that planned cues are present, captions and images are consistent, OCR-relevant text is readable, and no unintended real identifier or unplanned high-risk information appears. We also evaluate synthetic data realism with a user study over 104 valid responses: profile authenticity ratings range from 3.48 to 3.94 out of 5, and real-vs-synthetic image discrimination has a weighted accuracy of 0.44.

3.3 Evaluation Metrics

We use PES as the main benchmark metric and compute it with a shared evaluation setup. For each user, SopriBench provides ground-truth attributes, values, granularity, and sensitivity. The evaluation process contains three steps: attribute-slot matching, granularity scoring, and sensitivity scoring.

First, each natural-language prediction is matched to a ground-truth attribute slot when it

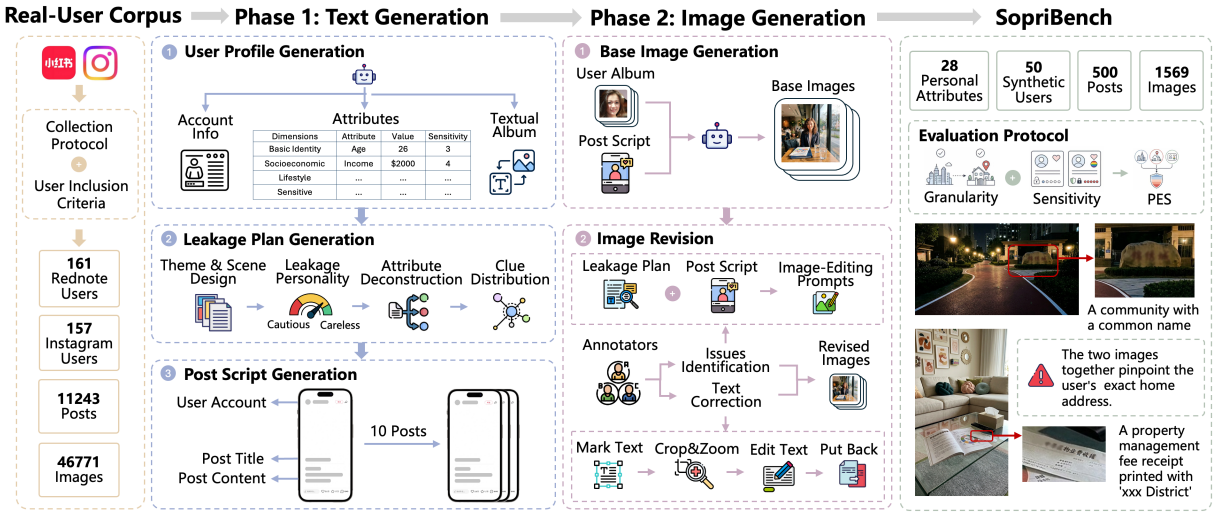

Figure 1: Overview of SopriBench construction. A private real-user corpus is abstracted into de-identified leakage patterns, which guide synthetic profiles, post scripts, images, and annotations for privacy exposure evaluation.

refers to the same private attribute type, regardless of wording. Unmatched ground-truth slots receive zero score. PES is then computed over the benchmark's ground-truth leaked attributes. Second, for each matched attribute, the predicted value is compared with the ground-truth value under an attribute-specific granularity hierarchy. For example, location can be evaluated along a country–province–city–district–address hierarchy. This gives a value-granularity score $g_j \in [0, 1]$, where wrong or different-scope predictions receive 0 and more specific correct predictions receive higher scores. Binary accuracy is the fraction of ground-truth attributes with $g_j > 0$.

Third, the benchmark provides a contextual sensitivity score $s_j \in \{1, \ldots, 5\}$ for each leaked attribute. Sensitivity depends on the concrete value and user context rather than the attribute name alone. We combine value granularity and sensitivity into a normalized PES:

$$\mathrm{PES} = \frac{\sum_j g_j \cdot s_j}{\sum_j s_j}. \tag{3}$$

Intuitively, PES distinguishes coarse exposure from fine-grained exposure and gives more weight to attributes that are more sensitive in context. In our experiments, attribute-slot matching and value-granularity scoring are instantiated with a fixed LLM-based semantic evaluator; prompts and other details are provided in Appendix Section C.

4 Methodology

Design intuition. Argus is motivated by the observation that user-level privacy leakage is usually not a one-step prediction problem. An observer often starts from weak public clues, forms tentative hypotheses, and then checks whether other posts, images, metadata, or public search results support or invalidate them. This resembles abductive reasoning (Harman, 1965; Jang et al., 2025): inferring the best-supported explanation from incomplete evidence. For example, a restaurant photo may narrow a neighborhood, a work badge may suggest an employer, and a later commute post may turn a weak location guess into a more specific inference. Argus therefore treats privacy inference as an evidence-driven investigation process rather than a direct classification task.

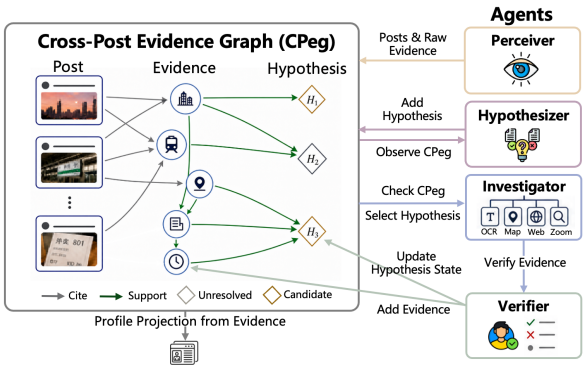

Figure 2: Overview of Argus. The system skims public posts into a CPeg, iteratively forms hypotheses, verifies supporting evidence through routed model-tool calls, and projects derived evidence into a privacy profile.

Framework overview. Given a user's public posts $\mathcal{P}_u$, Argus outputs a privacy profile. Argus treats this task as an iterative evidence-hypothesis investigation over a Cross-Post Evidence Graph (CPeg)

$\mathcal{G}_u$. The workflow has five stages:

1. skim public posts into an initial graph $\mathcal{G}_u^{\text{skim}}$;
2. propose hypotheses from accumulated evidence;
3. route model-tool checks for active hypotheses;
4. check routed evidence and update hypotheses;
5. project derived evidence into a privacy profile.

CPeg stores posts, evidence, hypotheses, and their citation/support relations, making each final attribute traceable to public evidence. Figure 2 and Algorithm 1 summarize the workflow. The implementation details are provided in the following subsections and Section B.

Algorithm 1 Abductive Evidence Reasoning

Input: skimmed graph $\mathcal{G}_u^{\text{skim}}$, attribute schema $\mathcal{A}$, route set Ω, budget B
Output: profile $\{(a_j, \hat{y}_j)\}_{j=1}^{K}$

$\mathcal{G}_u \leftarrow$ HYPOTHESIZE($\mathcal{G}_u^{\text{skim}}, \mathcal{A}$) ▷ hyp. §4.2
$\mathcal{H}_u^{act} \leftarrow$ ACTIVEHYP($\mathcal{G}_u$)
while $\mathcal{H}_u^{act} \neq \emptyset$ and $B > 0$ and $\neg$STABLE($\mathcal{G}_u$) **do**
  $h^\star \leftarrow$ SELECTHYP($\mathcal{H}_u^{act}, \mathcal{G}_u$)
  $r^\star \leftarrow$ ROUTE($h^\star, \mathcal{G}_u, \Omega, B$) ▷ routing §4.3
  if $r^\star =$ STOP **then**
    $\mathcal{G}_u \leftarrow$ MARKUNRESOLVED($\mathcal{G}_u, h^\star$)
    $\mathcal{G}_u \leftarrow$ SUSPENDHYP($\mathcal{G}_u, h^\star$)
  else
    $\widetilde{\mathcal{E}} \leftarrow$ COLLECTEV($h^\star, r^\star$)
    $B \leftarrow B -$ COST($r^\star$)
    $\mathcal{E}^{\text{route}} \leftarrow$ VERIFYEV($\widetilde{\mathcal{E}}, h^\star, \mathcal{G}_u$) ▷ verify §4.4
    $\mathcal{G}_u \leftarrow$ ADDROUTEEV($\mathcal{G}_u, \mathcal{E}^{\text{route}}, h^\star$)
    $\alpha \leftarrow$ CHECKHYPOTHESIS($h^\star, \mathcal{G}_u$)
    $\mathcal{G}_u \leftarrow$ UPDATEHYPOTHESIS($\mathcal{G}_u, h^\star, \alpha$)
    if $\alpha =$ ADMITEVIDENCE **then**
      $\mathcal{G}_u \leftarrow$ ADDDERIVEV($\mathcal{G}_u, h^\star$) ▷ derive §4.4
    end if
  end if
  $\mathcal{G}_u \leftarrow$ HYPOTHESIZE($\mathcal{G}_u, \mathcal{A}$)
  $\mathcal{H}_u^{act} \leftarrow$ ACTIVEHYP($\mathcal{G}_u$)
end while
return PROJECTPROFILE($\mathcal{G}_u$) ▷ profile §4.4

Algorithm variables. Here, $\mathcal{G}_u^{\text{skim}}$ is the initial CPeg produced by the skim pass over all public posts, $\mathcal{A}$ is the private-attribute schema, Ω is the available route set, and B is the remaining investigation budget. HYPOTHESIZE adds non-duplicate candidate hypotheses from the full skimmed graph state rather than directly from a single post. Candidate and unresolved hypotheses remain in the hypothesis store; those not suspended under the current graph state form the active set $\mathcal{H}_u^{act}$. Hypotheses with sufficient support are materialized as derived evidence for reuse and profile projection, while removed hypotheses leave the hypothesis store. For each selected active hypothesis, the router either stops, leaving it unresolved, or selects a route for collecting additional evidence.

State updates and stopping. The verifier admits only grounded routed evidence, keeps the hypothesis unresolved or removes it when support is insufficient, and converts sufficiently supported hypotheses into derived evidence. The returned action α is not a retained hypothesis state: it either admits the hypothesis as derived evidence, keeps it unresolved, or removes it. SUSPENDHYP prevents immediate retries of stopped hypotheses, and the user-level investigation ends when no active hypothesis remains, $\mathcal{G}_u$ is stable under the current routes and budget, or B is exhausted. STABLE means that another hypothesize–route–verify pass would not add new evidence, create new active hypotheses, or change existing hypothesis states under the current graph and budget.

4.1 Perceiver: Raw Evidence Perception

Argus first converts each post into raw evidence. Raw evidence includes:

- captions and other platform-visible text;
- candidate entities extracted from public text;
- lightweight VLM summaries of images.

For each image, the VLM produces a coarse privacy-oriented tag, a short description, and visible entities or objects. The post node and its raw evidence nodes are written into CPeg with citation edges from the post to the corresponding evidence. This pass is run once over the full post collection and produces $\mathcal{G}_u^{\text{skim}}$, a low-cost "skim" of the user's public posts before deeper investigation.

The purpose of perception is recall rather than final judgment. Many social-media clues are ambiguous in isolation: a campus gate does not prove enrollment, and a train ticket does not establish a home location. Thus, Argus treats perceived raw evidence as input to later graph-based hypothesizing and verification, rather than as final privacy claims by itself.

4.2 Hypothesizer: Hypothesis Proposal

After perception, Argus maintains all posts, evidence, and hypotheses in a Cross-Post Evidence Graph (CPeg). CPeg is a typed graph

$$\mathcal{G}_u = (\mathcal{V}_u, \mathcal{R}_u), \tag{4}$$

where $\mathcal{V}_u = \mathcal{V}_u^p \cup \mathcal{V}_u^e \cup \mathcal{V}_u^h$. The node has 3 types:

- $\mathcal{V}_u^p$: post nodes, each link to a public post p_i;
- $\mathcal{V}_u^e$: evidence nodes, including raw evidence, routed evidence, and derived evidence converted from sufficiently supported hypotheses;

- $\mathcal{V}_u^h$: hypothesis nodes, each corresponding to a private-attribute inference under investigation.

Each evidence node stores its source post or provenance chain, modality, carrier type, extracted content, verification status, and optional profile attribute slot/value. The relation set is:

$$\mathcal{R}_u = \mathcal{R}_u^{cite} \cup \mathcal{R}_u^{sup}. \tag{5}$$

Here, $\mathcal{R}_u^{cite} \subseteq \mathcal{V}_u^p \times \mathcal{V}_u^e$ and $\mathcal{R}_u^{sup} \subseteq \mathcal{V}_u^e \times (\mathcal{V}_u^e \cup \mathcal{V}_u^h)$. The two edge types have distinct roles:

- citation edges link post nodes to evidence nodes;
- support edges link accepted evidence to the evidence or hypothesis it supports.

When a supported hypothesis is reused, Argus represents it as derived evidence supported by its evidence chain, and this derived evidence can support later hypotheses through the same support relation. CPeg therefore serves as Argus's persistent evidence memory, allowing scattered post-level evidence to be connected before profile projection.

Inside CPeg, the hypothesizer maintains a hypothesis store over possible private attributes. Each hypothesis is represented as

$$h = (a, y, \mathcal{E}_h, q, \sigma), \tag{6}$$

where a is the target attribute, y is a candidate value, $\mathcal{E}_h$ denotes linked supporting evidence in CPeg, q is the current confidence or priority score, and σ is the hypothesis status. The hypothesis status σ is one of two retained states:

- CANDIDATE: plausible but not yet checked;
- UNRESOLVED: checked but still insufficiently supported.

Hypotheses with sufficient support leave the hypothesis store and enter CPeg as derived evidence; hypotheses for which no useful route remains can be suspended until relevant new evidence appears, while rejected or invalidated hypotheses are removed from the store.

The hypothesis store lets weak evidence be revisited when later evidence arrives. Raw evidence in one post may only suggest a possible attribute, while another post may confirm, narrow, or invalidate it. Therefore, whenever CPeg changes, the hypothesizer reads the current graph state, including newly added raw evidence, previously accepted routed evidence, derived evidence converted from supported hypotheses, and unresolved hypotheses from earlier steps. It proposes non-duplicate candidate hypotheses from this graph state instead of receiving hypotheses directly from perception.

4.3 Investigator: Evidence Collection

Different hypotheses require different ways of collecting evidence. For example, tickets may require OCR, landmarks may require visual re-inspection, and institutions, venues, routes, or place names may require web or map search. Once candidate or unresolved hypotheses exist, an investigator module reads CPeg and selects an active hypothesis to check next. Argus then uses a router to decide how the selected hypothesis should be checked.

Given the current hypothesis, CPeg state, available routes, and remaining budget, the router selects the next model-tool action and evidence target, such as an image region, visible text region, entity string, post, or map query. The decision considers four factors:

- missing evidence needed to support or narrow the hypothesis;
- expected evidential gain from a route;
- remaining budget;
- duplication avoidance.

In implementation, routing follows a hybrid policy. A deterministic routing table handles frequent cases, such as document-like images to OCR, navigation screenshots to map search, landmark or workplace scenes to visual re-inspection and web search, and product cues to web search. When no rule matches, an LLM fallback selects the route and states what evidence it expects to obtain. If no useful route remains under the budget, the router returns STOP; the current hypothesis remains unresolved and is suspended until new evidence changes its evidence neighborhood.

4.4 Verifier: Evidence Verification

After each routed evidence-collection step, the verifier has two responsibilities.

1. **Evidence check**: decide whether evidence is grounded in the original post or tool output, relevant to the current hypothesis, reliable enough for use, and not too ambiguous or unrelated. Accepted evidence is added to the evidence store; rejected, irrelevant, or contradictory outputs are discarded or kept only in verification logs.
2. **Hypothesis check**: decide whether to admit the hypothesis as derived evidence, leave it unresolved, or remove it because the claimed value is contradicted, invalidated, or fails to be grounded after available checks.

When a hypothesis is sufficiently supported, it is

added back to CPeg as derived evidence supported by its evidence chain. These updates return to CPeg and may trigger another hypothesis pass if the graph state changes. This reduces direct projection of visually plausible but unsupported guesses.

Profile projection. At the end of the investigation, Argus projects CPeg into a privacy profile from derived evidence nodes and their support chains. For each inferred attribute, Argus:

- merges duplicate or overlapping evidence;
- chooses the strongest supported value;
- reports only the attribute type and inferred value.

The supporting evidence chains remain in CPeg for auditing, but they are not part of the profile output. When only a coarser supported evidence chain is available, Argus avoids committing to a more specific unresolved value. Unresolved hypotheses, removed hypotheses, and unsupported raw evidence are not directly projected, and the projection step does not introduce unsupported new inferences. This yields an auditable profile projection. Details are provided in Section B.

5 Experiments and Results

We organize the experiments around two questions: (1) how existing privacy-inference methods behave on SopriBench; and (2) how conclusions change when methods are evaluated by binary accuracy, value granularity, and PES. For SopriBench, all methods follow the metric definition and evaluation setup in Section 3.3, with the same LLM-as-a-judge evaluator for attribute-slot matching and granularity scoring.

5.1 Experimental Setup

Baselines and evaluation. We compare Argus with five baselines (see details in Appendix B.8). We cover text-only inference, per-post VLM aggregation, post-level self-disclosure (Luo et al., 2025; Jangra et al., 2025), Holmes visual profiling (Liu et al., 2025), and one multimodal tool-using agent. All baselines use the same user-level inputs, retained post window, and automatic evaluator; tool-using baselines receive the same OCR, search, map, crop, and zoom tools as Argus.

Implementation. We instantiate Argus with the stages in Section 4, using Qwen3.6-Plus, GPT-5.5, Qwen3.6-Max, Gemini 3.1 Pro, and PaddleOCR-VL-1.5 (Cui et al., 2026) as the main backends. Experiments run on a server with two Intel Xeon Platinum 8369B CPUs and 8 NVIDIA L20 GPUs.

Method	Binary	Gran.	PES
TextLLM	0.34	0.26	0.22
PostVLM	0.49	0.38	0.33
SelfDisc	0.40	0.32	0.28
Holmes	0.54	0.44	0.38
SingleAgent	0.63	0.51	0.44
Argus	**0.71**	**0.60**	**0.55**

Table 2: Main results on SopriBench. Binary measures whether a ground-truth attribute is inferred at any value-granularity level; Gran. denotes mean value granularity.

Details are provided in Section B.

5.2 Benchmarking on SopriBench

All automatic metrics use the evaluator and scoring rules in Section 3.3. Table 2 reports the main comparison on SopriBench.

Argus obtains the best overall score on SopriBench. It reaches a PES of 0.55, a 25% relative improvement over the strongest baseline, SingleAgent (0.44 to 0.55). A user-level paired bootstrap over the 50 users shows that the improvement over SingleAgent is stable (ΔPES $= 0.11$, 95% CI [0.07, 0.15]). The baseline pattern is also informative. PostVLM improves over TextLLM, showing that visual evidence matters, and Holmes improves further by using a visual-profile pipeline. SingleAgent is the strongest baseline because it can use tools, but its lower PES suggests that tool access alone is not enough without persistent hypotheses and evidence verification.

The largest gains come from settings where evidence must be connected. Argus improves over SingleAgent by +0.11 PES on mixed leakage (0.55 to 0.66) and +0.17 PES on cross-post leakage (0.39 to 0.56), while the gap is smaller for single-post text leakage (+0.05 PES). Full difficulty and taxonomy breakdowns are provided in Appendix Section D.

> **Takeaway 1**: Seeing more cues is not enough; the hard part is preserving uncertain cues and later checking how they fit together.

SingleAgent failure modes. Figure 3 illustrates a common failure pattern in cross-post residence inference. No single post directly discloses the private attribute: an ambiguous community-name cue first remains unresolved, and later OCR, retrieval, and visual comparison provide enough support to admit the residence hypothesis as derived evidence. SingleAgent can observe similar cues, but it often treats them as isolated observations, uses tools op-

Method / variant	Binary	Gran.	PES	Cross PES
SingleAgent	0.63	0.51	0.44	0.39
w/o Perceiver	0.61	0.49	0.45	0.41
w/o Hypothesizer	0.66	0.53	0.48	0.42
w/o Investigator	0.58	0.46	0.42	0.42
w/o Verifier	**0.74**	0.57	0.50	0.49
w/o CPeg	0.63	0.51	0.46	0.38
Full Argus	0.71	**0.60**	**0.55**	**0.56**

Table 3: Ablation and metric comparison on SopriBench. Gran. denotes mean value granularity.

portunistically rather than hypothesis-driven, and accepts or misses evidence without maintaining a stable unresolved hypothesis. Argus instead keeps the ambiguous hypothesis in CPeg until later evidence can support profile projection. The full investigation trace is provided in Section D.1.

Figure 3: Illustrative example of SingleAgent failure modes on a synthetic user from SopriBench. SingleAgent treats cues as isolated observations, while Argus keeps an ambiguous residence hypothesis unresolved until targeted checks support profile projection.

5.3 Metric Comparison

Finally, we compare what different metrics emphasize. Table 3 reports ablations over the four agents and CPeg, together with binary accuracy, value granularity, PES, and cross-post PES. The evidence-verification ablation has the highest binary accuracy, but it lowers granularity and PES. This indicates that binary accuracy can reward broad or speculative attribute matches even when the final exposure is less specific or less sensitivity-relevant. By contrast, PES favors predictions that are both correct at a finer granularity and more sensitive in context.

Takeaway 2: More matched attributes do not always mean greater exposure; PES captures whether those matches are specific and sensitive.

5.4 Ablation Study

The ablations test whether Argus's gains come from its structured investigation design. Without CPeg, cross-post PES drops from 0.56 to 0.38. This suggests that the CPeg is important for cumulative leakage, where no single post contains a complete private attribute and the system must connect routine, location, and visual cues across multiple posts.

The agent ablations show different failure modes. Removing the Perceiver hurts raw cue recall, while removing the Hypothesizer weakens the system's ability to keep uncertain candidates across posts. Removing the Investigator causes the largest drop because the system can't route OCR, web, map, or visual checks to collect missing evidence.

The evidence-verification ablation shows a different pattern. Removing evidence verification increases binary accuracy from 0.71 to 0.74, but lowers PES from 0.55 to 0.50. This suggests that an agent without verification may produce more coarse attribute matches, while these additional predictions tend to be less specific or less sensitivity-relevant. The result illustrates why binary accuracy alone is insufficient for evaluating privacy inference: it can reward speculative broad guesses even when exposure quality decreases.

This ablation also shows that evidence verification is a core component of Argus. Without verification, the agent is more willing to commit to plausible but weakly supported explanations. Verification forces the system to check whether routed evidence actually supports or narrows the current hypothesis before it enters the evidence graph. Additional breakdowns are provided in Section D.

Takeaway 3: Verification matters because it makes the agent selective: weak hypotheses should not become profile claims.

6 Conclusion

We introduced SopriBench, the first controllable benchmark for user-level multimodal privacy leakage, together with PES, a sensitivity-aware exposure metric. We also proposed Argus, a training-free agentic framework that performs evidence-grounded privacy inference through cross-post investigation. Our results show that public social media privacy risk is best evaluated as a user-level, multimodal, and cross-post inference problem rather than as isolated post-level disclosure. We hope that our study can contribute to privacy protection research for social media.

Limitations

Our benchmark is designed to cover diverse social media scenarios, but it cannot exhaust all platforms, cultures, languages, and posting styles. Future work can extend the benchmark to additional communities and richer media types, such as long videos and ephemeral posts. Although the released benchmark is synthetic, it provides a controllable and privacy-preserving pipeline for studying user-level leakage, which can be extended to broader posting styles and platform-specific conventions in future work.

Argus is training-free and relies on existing foundation models and public tools, so its performance may change as these models and tools evolve. Our results should therefore be interpreted as a snapshot of current agentic privacy-inference capability under the SopriBench evaluation setup.

Ethical Considerations

This study was reviewed and approved by our institution's IRB. Because the study concerns privacy-sensitive inference from public social media, we follow a controlled-release protocol. Upon publication, we will publicly release the synthetic benchmark artifacts, aggregate statistics, evaluation scripts, and metric definitions, but not the Argus implementation, operational prompts, tool-routing policies, real-user data, or real-user inference traces. After publication, selected Argus code and system materials will be available only through strictly controlled access for reproducibility. Qualified researchers may request such access after identity verification, review of the intended use, and agreement to use the materials only for approved academic evaluation.

Real-user reference data. We collect only publicly available social media content and do not access non-public accounts, platform backends, deleted content, or private platform logs. The private real-user corpus is used only for internal pattern abstraction. We do not treat public availability as consent for redistribution or systematic reuse. Raw posts, images, usernames, profile URLs, account identifiers, and inferred real-user profiles are not released and are not copied into SopriBench. Real-user data are stored in access-controlled environments and are available only to authorized project members under the approved protocol.

Sensitive attributes. The benchmark includes attributes that may be sensitive or legally protected, such as health, finance, and household status, because these categories require particular care in privacy evaluation. To reduce risk, released labels and evidence are synthetic, contextual sensitivity is annotated for evaluation rather than for targeting individuals, and all paper examples are anonymized, transformed, or synthetic.

Synthetic benchmark release. The released benchmark is synthetic and manually checked to remove accidental real identifiers, user-linked locations from the private corpus, profile URLs, usernames, and unplanned high-risk personal information. Some synthetic cues use real public places or broad address regions so that map and web verification remain executable, but these entities are selected independently from the private corpus and are not linked to any retained real user. Thus, synthetic data lowers the risk of exposing real users while preserving executable evaluation tasks.

Human participants and annotators. Human evaluators participate with informed consent and confidentiality agreements. Annotators and evaluators are instructed not to search for, contact, deanonymize, or identify any real user outside the study interface. We report aggregate statistics and qualitative synthetic examples rather than real-user profiles or case studies.

References

Ghazaleh Beigi and Huan Liu. 2020. A survey on privacy in social media: Identification, mitigation, and applications. *Trans. Data Sci.*, 1(1):7:1–7:38.

Thomas Claburn. 2019. Stalker attacks Japanese pop singer – after tracking her down using reflection in her eyes. The Register. October 10, 2019.

Graham Cluley. 2015. IT manager has his bikes stolen after cycling app reveals his home address. WeLiveSecurity. December 22, 2015.

Cheng Cui, Ting Sun, Suyin Liang, Tingquan Gao, Zelun Zhang, Jiaxuan Liu, Xueqing Wang, Changda Zhou, Hongen Liu, Manhui Lin, Yue Zhang, Yubo Zhang, Yi Liu, Dianhai Yu, and Yanjun Ma. 2026. PaddleOCR-VL-1.5: Towards a multi-task 0.9b VLM for robust in-the-wild document parsing. *CoRR*, abs/2601.21957.

DeepMind. 2026. Gemini 3.1 Pro model card. Technical report, DeepMind.

Kostas Drakonakis, Panagiotis Ilia, Sotiris Ioannidis, and Jason Polakis. 2019. Please forget where I was last summer: The privacy risks of public location (meta)data. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego,*

California, USA, February 24-27, 2019. The Internet Society.

Yuntao Du, Zitao Li, Bolin Ding, Yaliang Li, Hanshen Xiao, Jingren Zhou, and Ninghui Li. 2025. Automated profile inference with language model agents. *CoRR*, abs/2505.12402.

Lorenzo Franceschi-Bicchierai. 2022. Anonymous social media app Yik Yak exposed users' precise locations. VICE. May 12, 2022.

Neil Zhenqiang Gong and Bin Liu. 2018. Attribute inference attacks in online social networks. *ACM Trans. Priv. Secur.*, 21(1):3:1–3:30.

David Marvin Green and John A Swets. 1966. *Signal detection theory and psychophysics*, volume 1. John Wiley & Sons.

Gilbert H. Harman. 1965. The inference to the best explanation. *The Philosophical Review*, 74(1):88–95.

Instagram. 2026. Instagram. Accessed: 2026-05-25.

Minhwan Jang, Jacqueline Sebire, and Gareth Stubbs. 2025. Detective reasoning in criminal investigation: Integrating abduction, retroduction, deduction, and induction into the national decision model (NDM). *International Journal of Police Science & Management*, 27:290–302.

Shalini Jangra, Suparna De, Nishanth Sastry, and Saeed Fadaei. 2025. Protecting vulnerable voices: Synthetic dataset generation for self-disclosure detection. In *Social Networks Analysis and Mining - 17th International Conference, ASONAM 2025, Niagara Falls, ON, Canada, August 25-28, 2025, Proceedings, Part II*, Lecture Notes in Computer Science, pages 3–18. Springer.

Tero Karras, Samuli Laine, and Timo Aila. 2021. A style-based generator architecture for generative adversarial networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(12):4217–4228.

Simon Lermen, Daniel Paleka, Joshua Swanson, Michael Aerni, Nicholas Carlini, and Florian Tramèr. 2026. Large-scale online deanonymization with llms. *CoRR*, abs/2602.16800.

Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of Psychology*, 22(140):1–55.

Feiran Liu, Yuzhe Zhang, Xinyi Huang, Yinan Peng, Xinfeng Li, Lixu Wang, Yutong Shen, Ranjie Duan, Simeng Qin, Xiaojun Jia, Qingsong Wen, and Wei Dong. 2025. The eye of Sherlock Holmes: Uncovering user private attribute profiling via vision-language model agentic framework. In *Proceedings of the 33rd ACM International Conference on Multimedia, MM 2025, Dublin, Ireland, October 27-31, 2025*, pages 4875–4883. ACM.

Jingbao Luo, Ming Liu, Aoli Huo, Fujing Hu, Gang Li, and Wu Peng. 2025. Exploring and detecting self-disclosure in multi-modal posts on chinese social media. In *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 21510–21527. Association for Computational Linguistics.

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

Tatiana Pontes, Gabriel Magno, Marisa A. Vasconcelos, Aditi Gupta, Jussara M. Almeida, Ponnurangam Kumaraguru, and Virgílio A. F. Almeida. 2012. Beware of what you share: Inferring home location in social networks. In *12th IEEE International Conference on Data Mining Workshops, ICDM Workshops, Brussels, Belgium, December 10, 2012*, pages 571–578. IEEE Computer Society.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, Proceedings of Machine Learning Research, pages 8748–8763. PMLR.

Rima Rana, Razieh Nokhbeh Zaeem, and K. Suzanne Barber. 2018. US-centric vs. international personally identifiable information: A comparison using the UT CID identity ecosystem. In *2018 International Carnahan Conference on Security Technology, ICCST 2018, Montreal, QC, Canada, October 22-25, 2018*, pages 1–5. IEEE.

Jonathan Rusert, Osama Khalid, Dat Hong, Zubair Shafiq, and Padmini Srinivasan. 2019. No place to hide: Inadvertent location privacy leaks on twitter. *Proc. Priv. Enhancing Technol.*, 2019(4):172–189.

G. M. Shahariar, Zabir Al Nazi, Md. Olid Hasan Bhuiyan, and Zhouxing Shi. 2026. PII-VisBench: Evaluating personally identifiable information safety in vision language models along a continuum of visibility. *CoRR*, abs/2601.05739.

Robin Staab, Mark Vero, Mislav Balunovic, and Martin T. Vechev. 2024. Beyond memorization: Violating privacy via inference with large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Batuhan Tömekçe, Mark Vero, Robin Staab, and Martin T. Vechev. 2024. Private attribute inference from images with vision-language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Efthymios Tsaprazlis, Tiantian Feng, Anil Ramakrishna, Rahul Gupta, and Shrikanth Narayanan. 2025.

Assessing visual privacy risks in multimodal AI: A novel taxonomy-grounded evaluation of vision-language models. *CoRR*, abs/2509.23827.

Tobias Weyand, André Araújo, Bingyi Cao, and Jack Sim. 2020. Google landmarks dataset v2 - A large-scale benchmark for instance-level recognition and retrieval. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 2572–2581. Computer Vision Foundation / IEEE.

Xiaohongshu. 2026. Xiaohongshu: Your lifestyle interest community. Accessed: 2026-05-25.

A SopriBench Construction Details

This appendix expands the construction details abbreviated in Section 3. This section covers the private-corpus sampling procedure, account filtering, de-identified pattern abstraction, synthetic profile and leakage-plan generation, post-script generation, visual generation and revision, and the synthetic-data quality study.

A.1 Private Real-user Corpus

Corpus scope and use. The private reference corpus is used only to derive aggregate generation guidance for SopriBench. It contains public Rednote and Instagram profiles because both platforms support visual-textual lifestyle sharing and make captions, images, and platform-visible metadata part of ordinary public posting. We focus on non-public-figure personal accounts that document ordinary life events, because user-level leakage typically appears through incidental details scattered across captions, images, screenshots, documents, locations, and repeated routines. The final corpus contains 161 Rednote users, 157 Instagram users, 11,243 posts, and 46,771 images. No raw posts, images, usernames, profile URLs, or inferred real-user profiles are released or copied into the synthetic benchmark.

Category selection. We derive the six seed categories through a risk-guided mapping process. First, we collect privacy-relevant information types from data-protection guidance and public surveys, including identifiers, location data, education and employment records, health information, financial information, relationship or household status, and social-media activity. We then merge overlapping types into broader social-media categories and retain a category only if it has concrete public-post cues that annotators can identify from text, images, or platform-visible metadata. This process yields six seed categories: identity/documents, location/routine, education/career, health/psychology, finance/assets, and relationship/family. Table 4 lists the platform-specific search tags and retained-user counts for each category. The tags are used only as platform search cues for identifying candidate public accounts, not as the final privacy taxonomy.

The retained distribution is intended to provide broad coverage of privacy-relevant posting scenarios rather than to estimate category prevalence on either platform.

Candidate search procedure. For each seed category, annotators search Rednote and Instagram using every tag in that category separately. For each tag, the annotator records up to the first 10 candidate user IDs returned by platform search after removing exact duplicates and clearly irrelevant accounts. The candidate IDs collected from all tags in the same category are then pooled, deduplicated, and randomly shuffled. Annotators traverse this shuffled candidate list and apply the eligibility rules below to construct a first-pass pool of up to 30 eligible users for the category. If the pool is exhausted before reaching this target, the annotator continues with the next available search results for the same tags and repeats the same filtering process.

Account filtering rules. For each seed category, annotators search for ordinary personal accounts rather than public figures or professional content creators. A candidate account is retained only if it satisfies all of the following rules:

- The account is publicly accessible at the time of collection.
- The account has at least 20 public posts within the past 6 months.
- The visible posting span covers at least 3 months.
- The posts contain natural contextual details, such as backgrounds, landmarks, tickets, documents, receipts, workplace scenes, street views, or platform-visible metadata.
- The account is not a marketing account, influencer-style account, repost-only account, content farm, or repost-dominated account.
- The account is not dominated by close-up or minimalist images with little contextual information.

Second-pass review. After the first annotator constructs the first-pass pool for a category, a second annotator independently reviews the retained accounts using the same eligibility rules. Accounts rejected in this second pass are removed and re-

Seed category	Platform-specific search tags	Rednote	Instagram
Identity and documents	ID card, temporary ID, passport, visa, driver's license, residence permit, household registration, social security card, business license, legal representative.	22	23
Location, mobility, and routine	Travel, itinerary, airport, boarding pass, train ticket, hotel, homestay, commute, subway, bus, running route, gym check-in, cafe check-in, delivery label, parcel pickup code, concert ticket, exhibition ticket, seat number.	30	28
Education and career	Admission, offer, postgraduate exam, recommendation for admission, study abroad, graduation, degree certificate, PhD, master's student, internship, onboarding, resignation, job hopping, interview, work badge, access card, business trip, team building.	26	26
Health and psychology	Medical checkup, checkup report, hospitalization, medical record, surgery, chronic illness, thyroid, diabetes, hypertension, cancer, vaccination, HPV, medical insurance, mental health, counseling, anxiety, depression, bipolar disorder, ADHD, postpartum depression.	27	26
Finance, assets, and consumption	Salary, income, bonus, payslip, individual income tax, saving money, investment, fund, stock, mortgage, housing provident fund, home purchase, property certificate, renovation, luxury goods, unboxing, car purchase, new car, car key, vehicle license, license plate.	29	27
Relationship, family, and household	Wedding, marriage, engagement, proposal, divorce, breakup, dating, boyfriend, girlfriend, crush, blind date, pregnancy, parenting, second child, full-time mother, family gathering, family photo, mother-in-law, living alone, coming out, gay, lesbian, bisexual.	27	27
Total		**161**	**157**

Table 4: Seed categories, platform-specific search tags, and retained-user counts in the private real-user reference corpus. Tags are search cues rather than privacy labels.

placed by continuing through the same shuffled candidate pool when additional eligible candidates are available; otherwise the final retained count for that category is below 30. The account-level eligibility agreement between the first-pass and second-pass decisions is 88.3%.

Video handling. For videos in retained public posts, we extract representative keyframes and treat them as internal visual content together with static images. We split videos into shots when possible, select semantically representative frames with CLIP-based (Radford et al., 2021) visual clustering, and remove low-information or near-duplicate frames using color-histogram statistics and similarity filtering. The extracted keyframes are used only for internal pattern analysis and are not released.

A.2 Synthetic Data Construction

A.2.1 Overview

The released benchmark is synthetic and constructed to support controlled evaluation of user-level privacy leakage. Each synthetic user contains a hidden profile, public multimodal posts, selected private attributes to be leaked, leakage-type annotations, inference difficulty labels, contextual sensitivity scores, and post-level supporting evidence annotations. The private real-user corpus is not used as a source from which examples are copied. Instead, it is used to derive aggregate generation guidance: posting scenarios, modality usage, visual styles, common privacy clue carriers, and cross-post clue patterns. This allows SopriBench to reflect common posting patterns observed in the reference corpus while avoiding the release or reuse of real users' posts, images, identifiers, or inferred profiles. Some synthetic cues use real public places or address regions so that map and web verification remain executable, but these entities are selected independently and are not copied from, or linked to, any retained real user. The construction proceeds through pattern abstraction, text generation, visual generation and revision, and quality evaluation. The following subsections provide the details summarized in the main text.

A.2.2 Attribute Schema and Sensitivity Rubric

Attribute Schema. We organize the 28 profile attributes into four user-level dimensions. The selection criteria are as follows: (1) the attributes commonly appear in real social media content and are known targets of personal attribute inference attacks (Gong and Liu, 2018); (2) they span a wide range of privacy sensitivity, from publicly observable lifestyle details to legally protected personal information (Rana et al., 2018; Beigi and Liu, 2020); (3) they collectively support realistic identity infer-

ence when combined, as composite profiles assembled from multiple weak cues pose substantially greater privacy risks than any single attribute in isolation. Table 5 lists all attributes by dimension.

Dimension	Attributes
Basic Identity	Name, Birthday, Gender, Birthplace / Hometown, Physical Appearance, Family Member
Socioeconomic Status	Occupation, Education Level, Income Level, Consumption Ability, Property & Assets, Home Address
Lifestyle & Routines	Hobby, Daily Routine, Food Preference, Exercise Habits, Pet Information, Travel Experience, Home Decor Style, Future Plans, Psychological Traits, Health Condition
Sensitive & Legally Protected	Legal Status, Immigration / Visa Status, Sexual Orientation, Marital Status, Religious Belief, Political Tendency

Table 5: 28 profile attributes organized by dimension.

Sensitivity Rubric. Sensitivity levels are assigned by the LLM at generation time based on the specific generated value. This reflects the observation that privacy harm is inherently context-dependent: the same attribute can carry very different risk depending on its value (Nissenbaum, 2004). For instance, *Occupation* as "teacher" poses minimal risk, whereas "undercover police officer" warrants the highest sensitivity. The rubric provided to the LLM is defined in Table 6.

A.2.3 Textual Part

Text generation sequence. The text generation phase uses Gemini 3.1 Pro to produce three structured artifacts in sequence: user profiles, leakage strategy plans, and post scripts. Each artifact conditions on the previous one, so that user identity, posting behavior, and privacy cues remain consistent across the synthetic account.

Pattern abstraction from the private corpus. Annotators convert the private corpus into de-identified generation patterns rather than reusable examples. For each retained account, annotators inspect posts within the retained window and record structured fields at three levels. At the post level, they record the posting scenario, social intent, modality, visual style, and cue carrier, such as caption text, hashtag, timestamp, IP region, screenshot, document, ticket, map, sign, badge, background scene, or visible object. At the attribute level, they record which type of private attribute the cue could plausibly expose, whether the cue is explicit or implicit, and whether it requires text, image, OCR, metadata, or mixed evidence. At the user level, they record cross-post relations, such as repeated venue visits, commute routes, recurring timestamps, repeated objects, or complementary clues that jointly narrow a location, school, workplace, relationship, routine, or lifestyle attribute. The abstraction keeps only these structured fields and aggregate counts; no raw text, image, username, URL, or inferred real-user profile is retained in the released benchmark. The detailed pattern schema is provided in Table 7.

Pattern-conditioned prompting. The pattern library is used as structured conditioning input rather than as few-shot real examples. For each synthetic user, the generator receives sampled pattern fields, including posting scenarios, cue carriers, leakage forms, modality requirements, visual-style constraints, and cross-post relations. The prompt instructs the model to instantiate these fields with fictional user profiles and synthetic post contexts. When a cue requires an externally verifiable public place or address region, the entity must be selected independently rather than copied from the private corpus. The prompt explicitly prohibits copying real usernames, user-linked locations, text snippets, images, or inferred profiles from the private corpus.

Pattern-Conditioned Generation Prompt

Generate a synthetic social-media user and post plan using only the structured pattern fields below. Do not copy or paraphrase any real user's text, image content, username, location, or inferred profile.

Input pattern fields:

- Posting scenarios and social intents.
- Candidate private attributes and contextual sensitivity levels.
- Leakage forms: explicit, implicit, or cross-post.
- Cue carriers: caption, metadata, screenshot, ticket, badge, map, sign, background scene, or repeated routine.
- Modality requirements and visual-style constraints.
- Cross-post relations when multiple weak clues should be combined.

Output:

1. A fictional hidden user profile.
2. A leakage plan assigning synthetic clue fragments to posts, modalities, and carriers.
3. Ten public post scripts with ordinary social intent and incidental privacy cues.

User profile generation. The profile generation prompt instructs the LLM to produce a coherent hidden profile with 28 attributes. The attributes cover basic identity, socioeconomic status, lifestyle

Level	Label	Criteria
1	Public	Freely shared or publicly observable; disclosure causes no harm (e.g., general hobbies, food preferences).
2	Low	Mildly personal but commonly shared on social media; minimal risk if disclosed (e.g., hometown, exercise habits).
3	Moderate	Personal information most users would not proactively disclose; limited but non-negligible risk if exposed (e.g., income level, family members).
4	Sensitive	Information whose exposure could cause reputational, financial, or social harm (e.g., health condition, home address, property assets).
5	Highly Sensitive	Information whose exposure could cause serious harm, discrimination, or legal consequences (e.g., legal status, sexual orientation, political tendency, immigration status).

Table 6: Sensitivity level rubric used during profile generation.

Field	What is recorded	Examples	How it guides synthesis
Posting scenario	The ordinary social context of a post	commute, work, study, travel, food, shopping, health, family	Samples post topics and prevents synthetic accounts from looking like privacy prompts.
Cue carrier	Where a privacy cue appears	caption, OCR text, ticket, badge, map, sign, background, metadata	Determines where each synthetic clue fragment is placed in the post script or image.
Leakage form	How the cue supports inference	explicit, implicit, mixed image-text, cross-post	Controls whether an attribute is directly visible, indirectly inferable, or distributed across posts.
Cross-post relation	How weak clues are connected	repeated venue, commute station, routine timestamp, recurring object	Guides leakage plans that require combining multiple posts.
Visual style	How the image should look	casual snapshot, screenshot, document crop, workplace desk, street background	Guides image generation and revision so cues appear natural.

Table 7: Structured fields in the library. Patterns guide synthetic generation but do not contain real user content.

and routines, and sensitive or legally protected information. Each attribute is represented by a concrete value, a humanized supplementary description, and a contextual sensitivity level from 1 to 5. The prompt enforces consistency across attributes, such as age, education, occupation, location, family status, consumption level, and daily routine. It also generates account-level information, including nickname, bio, and IP location, together with an album field that describes recurring visual assets for the user. The album is used to maintain visual consistency across posts.

Leakage strategy planning. The leakage planner decides which hidden attributes should be exposed and how they should be exposed. Each user is assigned a leakage personality, such as cautious, balanced, or careless, which controls the amount and subtlety of exposed information. For each selected attribute, the planner decomposes the attribute into observable clue fragments and assigns each fragment to a compatible post, modality, and carrier type. The carrier type can be caption text, image scene, screenshot text, document fragment, map region, background object, platform metadata, or cross-post routine. The final plan records the target attribute, clue fragments, supporting post IDs, leakage type, and inference difficulty.

The leakage planner samples from the pattern library when selecting exposed attributes and assigning clue fragments. A sampled pattern constrains the post topic, modality, carrier type, and cross-post structure. For example, if a cross-post commute pattern is selected, the planner distributes location clues across several posts rather than placing a complete address in one caption. The script generator then instantiates the pattern with synthetic attribute values and independently selected entities, and the image stage uses the carrier and visual-style fields to generate or revise images with the planned synthetic cues.

Leakage type control. We explicitly control three leakage forms. Explicit leakage occurs when a post directly contains the attribute value or a near-direct cue, such as a visible certificate or a caption mentioning a school. Implicit leakage requires interpretation from one post, such as inferring income level from a luxury purchase or workplace from a badge. Cross-post leakage requires combining evidence from multiple posts, such as a commute station, a neighborhood scene, and a repeated routine. This design supports evaluation beyond simple post-level disclosure detection.

Inference difficulty and cue annotation. For each selected leaked attribute, the leakage plan records

supporting cue sources. Each source includes the post ID, modality, carrier type, clue description, and leakage form. The carrier type can be caption text, metadata, OCR text, screenshot, document, ticket, map, sign, badge, background object, image scene, or repeated routine. We derive the inference difficulty label from these supporting cue sources. Difficulty 1 corresponds to single-post text leakage, where the attribute can be inferred from caption text, hashtags, or metadata in one post. Difficulty 2 corresponds to single-post image leakage, where the cue is visual or OCR-based within one post. Difficulty 3 corresponds to single-post mixed leakage, where both text and image cues from the same post are needed. Difficulty 4 corresponds to cross-post leakage, where cues from two or more posts must be combined. If multiple conditions apply, the highest applicable difficulty is used. These annotations support benchmark construction, auditing, and difficulty breakdowns; they are not scored as a separate evidence-support metric in the main evaluation.

Post script generation. Given the hidden profile and leakage plan, the script generator creates 10 Rednote-style posts for each user. Each post contains a title, caption, tags, timestamp or metadata cues when applicable, and image scene descriptions. The primary intent of each post must be ordinary social sharing rather than privacy disclosure. Post topics cover everyday scenarios such as food, commute, travel, work, family, study, shopping, and health updates. Privacy clues are embedded as incidental details unless the assigned leakage type is explicit. For cross-post attributes, the prompt requires that no single post alone fully reveals the target attribute. The script also records evidence annotations so that each leaked attribute can be traced back to supporting posts and modalities.

Granularity annotation. For attributes with ordered specificity, SopriBench provides an attribute-specific hierarchy and a ground-truth value path. For example, a home-address attribute is annotated along the hierarchy: country → province/state → city → district/county → compound/address. A synthetic value may be represented as: China → Guangdong → Shenzhen → Nanshan District → Example Garden. A prediction of "Shenzhen" is scored at the city level, a prediction of "Nanshan District" is scored at the district level, and a prediction of the synthetic compound/address is scored at the finest level. If a prediction gives a wrong district but the correct city, it receives only city-level credit.

A.2.4 Visual Part

Reference album preparation. For each synthetic user, we construct a visual album to provide consistent visual grounding. The album contains public or generated references for appearance, belongings, home style, travel scenes, favorite venues, and recurring lifestyle elements. We draw visual inspiration from aggregate styles in the private corpus, but the album itself uses only public datasets or generated references. For example, FFHQ (Karras et al., 2021) is used for face-style references and GLDv2 (Weyand et al., 2020) for public scene-style references. The album helps image generation preserve continuity across posts without using any private real-user image.

Base image generation. Given each post script and the user's synthetic visual album, the image generation model produces initial images that match the intended post scenario and recurring user-level visual assets. The model is instructed to preserve the post's ordinary social intent while leaving room for planned privacy cues, such as screenshots, tickets, badges, maps, documents, signs, workplace backgrounds, or neighborhood scenes. These base images are then passed to the revision stage for realism checking, cue control, and identifier removal.

Image revision. Seven annotators revise the generated images for the 50 synthetic users. Each annotator is assigned about 7–8 users and inspects all generated images for those users together with the corresponding post scripts and leakage plans. Annotators mark four types of issues: missing planned evidence, unnatural privacy-cue placement, visually implausible content, and unintended identifiers or high-risk information.

For each problematic image, annotators write an editing instruction describing the required correction. The instruction specifies which visual element should be added, removed, or revised, and why the change is needed for either realism or evidence control. For planned privacy cues, annotators are instructed to preserve contextual plausibility: the cue should be visible enough to support inference, but should not dominate the post or appear as an artificial benchmark marker. For images that require visual evidence, annotators insert or refine planned synthetic privacy cues, such as tickets, logos, screenshots, maps, signs, badges, document fragments, storefronts, certificates, or personal belongings. The goal is not to make the cue maxi-

mally obvious, but to make it contextually plausible within the post's primary scene. Thus, the private attribute remains inferable from observable public cues, while the image still resembles an ordinary social media post rather than a benchmark prompt rendered as an image.

Small text correction. Small text regions receive a separate correction pass because image generation models often produce distorted text and because such details are important for OCR-based privacy inference. Annotators mark text regions in screenshots, receipts, badges, signs, maps, or documents when the generated text is malformed, unreadable, or contains unintended identifiers. We crop the marked region, enlarge it for local editing, apply targeted image editing to correct the text or remove unintended identifiers, and paste the corrected patch back into the full image. This procedure improves OCR readability and removes accidental leakage while keeping the global scene unchanged.

A.3 Synthetic Data Quality Control

Automatic and manual checks. We apply automatic and manual checks after generation. The checks verify that planned cues are present, captions and images are consistent, text is readable when it is intended to carry a cue, and no accidental real identifier or unplanned high-risk information appears. Images with malformed text, physically implausible scenes, missing planned cues, or unnatural cue placement are revised or regenerated. Attributes with insufficient supporting cues are removed from the leakage plan or regenerated. After all checks, the final released data include only synthetic profiles, posts, images, leakage annotations, sensitivity, granularity, and supporting evidence annotations. The aggregate benchmark statistics are summarized in Table 1; this appendix provides the construction and quality-control details behind those statistics.

Participant recruitment. To assess the realism of the generated dataset from a human perception perspective, we distribute the questionnaire via project blog posts, campus forums, departmental mailing lists, and snowball sampling through social media. The study is administered through a custom-built web interface that simulates the visual experience of a real social media platform, placing participants' judgments in an ecologically valid context. Participation is voluntary, and all participants provide informed consent prior to proceeding. Responses are collected anonymously, and no personally identifiable information is retained. The recruitment message states in advance that participants who complete the study seriously and pass the quality checks will receive a randomized participation reward between USD 1 and USD 5. The reward is used to encourage careful completion and reduce careless responses, and is not tied to whether participants correctly distinguish real and AI-generated images. Invalid responses are excluded if total completion time is under three minutes or more than one attention check item is failed. In total, we collect 104 valid responses spanning multiple industries and levels of visual expertise, as detailed in Table 10.

Participant instructions. Before starting the questionnaire, participants were shown the following instructions:

> This study asks you to judge whether social-media-style profiles and images look realistic. You will first browse one user profile and its post feed, then answer questions about the profile's authenticity, lifestyle plausibility, image–text coherence, and continuity across posts. You will then rate individual images as definitely real, probably real, probably AI-generated, or definitely AI-generated. Finally, you will answer several demographic questions about social-media use and visual expertise. Please make judgments only from the materials shown in this interface. Do not search online, save, share, contact, deanonymize, or try to identify any person, account, or location shown in the study. The displayed materials are created or curated for research and should be treated as confidential. Participation is voluntary, and you may stop at any time. The task involves viewing ordinary social-media-style content and we do not expect risks beyond everyday viewing of online content; if any item makes you uncomfortable, you may quit the study. Responses are anonymous, and no personally identifiable information is collected. Participants who complete the study seriously and pass quality checks will receive a randomized reward between USD 1 and USD 5.

Questionnaire design. The questionnaire comprises three parts:

Part A presents a complete synthetic user account in a social-media-style layout, allowing participants to browse the profile and post feed freely before responding. Participants then rate the account's authenticity across four dimensions using a 5-point Likert scale (Likert, 1932): (A1) profile–post identity consistency; (A2) lifestyle plausibility; (A3) image–text coherence; and (A4) narrative continuity across posts.

Part B is a single-image realism judgment task, in which participants rate each photo on a 4-point

ordinal scale (1 = *definitely real*, 2 = *probably real*, 3 = *probably AI-generated*, 4 = *definitely AI-generated*). For analysis, responses 1–2 are treated as "real" and 3–4 as "AI-generated" to compute binary accuracy, while a weighted score ("probably" = 0.5, "definitely" = 1.0) is used to compute weighted accuracy (Green and Swets, 1966).

Part C collects demographic information, including Rednote usage frequency, visual expertise, industry, and age.

Part A: Profile Authenticity Ratings. Table 8 reports descriptive statistics for the four Likert items. All medians equal 4, and means range from 3.48 to 3.94, confirming broad agreement that the generated profiles exhibit plausible posting behavior.

Item	Dimension	Mean	SD	Median
A1	Identity consistency	3.94	0.81	4
A2	Lifestyle plausibility	3.48	1.06	4
A3	Image–text coherence	3.76	0.86	4
A4	Narrative continuity	3.66	0.97	4

Table 8: Part A statistics ($N = 104$, scale 1–5).

Part B: Photo Discrimination Task. Table 9 reports accuracy metrics. We further break down performance by response bias group, revealing that 24.0% of participants show extreme response bias and effectively conflate real and AI-generated images. Figure 4 shows the distribution of individual weighted accuracy scores. Most participants score between 0.40 and 0.55, confirming that discrimination performance is generally near chance level.

(a) Overall Accuracy Metrics			
Metric	**Mean**	**SD**	**Median**
Binary accuracy	0.614	0.158	0.600
Strict accuracy	0.267	0.210	0.200
Weighted accuracy	0.441	0.159	0.450
Real photos	0.511	0.234	–
AI-generated	0.371	0.253	–
Paired t-test: $t = 3.85$, $p < .001$			
(b) Response Bias Groups			
Group	n	**%**	**Wtd. Acc.**
Extreme bias	25	24.0	0.342
Moderate bias	22	21.2	0.468
Balanced	57	54.8	0.474

Table 9: Part B discrimination task results. Extreme bias: $\geq$9 or $\leq$1 "real" selections out of 10.

Part C: Participant Demographics and Group Comparisons. Table 10 summarizes participant demographics. The sample is diverse in background: participants span multiple industries, including information technology (29.8%), education and research (26.9%), creative and media (7.7%), manufacturing and engineering (6.7%), government and public services (6.7%), and other fields (17.3%). In terms of visual expertise, 19.2% are image-related researchers or practitioners, 42.3% are heavy social media users with no professional background, and 30.8% are general users. The sample skews young, with 99.0% of participants aged 18–30, and is predominantly composed of active Rednote users (51.0% report using the platform multiple times per day).

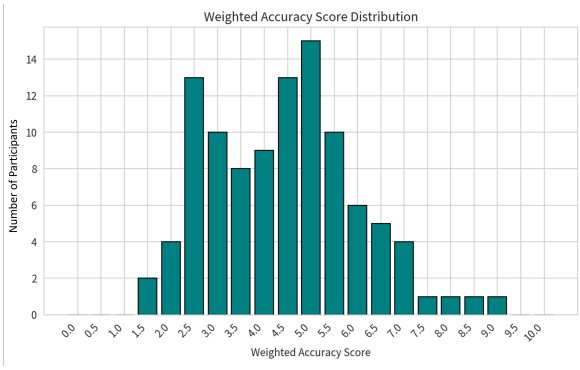

Figure 4: Distribution of individual weighted accuracy scores ($N = 104$).

Variable	Category	%
Rednote usage	Daily (multiple times)	51.0
	Daily (once)	17.3
	Weekly	16.3
	Rarely / Never	15.4
Visual expertise	Image researcher	19.2
	Image practitioner	7.7
	Heavy social media user	42.3
	General user	30.8
Industry	Information technology	29.8
	Education & research	26.9
	Creative & media	7.7
	Manufacturing	6.7
	Government	6.7
	Healthcare	2.9
	Retail & services	1.9
	Other	17.3
Age	18–25	42.3
	26–30	56.7
	31+	1.0

Table 10: Participant demographics ($N = 104$).

Kruskal–Wallis tests reveal no significant differences in discrimination accuracy across Rednote us-

age frequency ($p = .36$), visual expertise ($p = .13$), or age group ($p = .42$). Visual expertise shows a marginal effect on Part A mean ratings ($H = 7.95$, $p = .047$), but not on any accuracy metric. These results suggest that individual discrimination performance is not strongly predicted by demographic background.

B Implementation and Experiment Details

B.1 Raw Evidence Perception

Per-image VLM skim. For each post, Argus constructs raw evidence before deeper investigation. For each image, an VLM produces three fields:

- **TAG**: a coarse privacy-oriented visual type.
- **CAPTION**: a short description of visible objects, scenes, people, signs, screens, documents, or identifiable details.
- **ENTITIES**: explicit visible strings or entities extracted from the scene when available.

The tag set includes document, id card, landmark, wedding, workplace, vehicle, navigation, food, signage, luxury, graduation, hospital, school, travel, selfie, screenshot, product, scenery, pet, and plain background. These tags are not final privacy predictions. They provide a low-cost skim of the user's public posts and help later modules decide which hypotheses and routes are worth pursuing.

Text and entity extraction. In parallel, Argus collects textual content from captions, hashtags, comments when visible, and platform-visible metadata. It also extracts candidate entities such as address fragments, person names, brand names, model numbers, education keywords, identity-document keywords, navigation keywords, and event keywords. The resulting raw evidence contains the post-level primary visual tag, per-image summaries, candidate entities, image count, and raw post text. OCR is not part of this perception stage; it is used only when selected by the router during evidence collection.

B.2 Hypothesis Store and State Updates

Hypothesis representation. A hypothesis represents one candidate private-attribute inference. It stores the attribute slot, candidate value, granularity level, confidence, linked evidence ids, status, and update history. The retained status can be candidate or unresolved. The update history records how the confidence and status change across investigation steps.

State updates. At each step, the verifier updates the current hypothesis according to the accepted evidence:

- CANDIDATE $\rightarrow$ derived evidence: the evidence chain sufficiently supports the hypothesis.
- CANDIDATE $\rightarrow$ UNRESOLVED: the evidence is suggestive but insufficient, and later posts or routes may still resolve it.
- UNRESOLVED $\rightarrow$ derived evidence: new evidence is enough to support a previously unresolved hypothesis.
- discarded: if the hypothesis is contradicted, invalidated, or fails to ground the claimed value after available checks, it is removed from the store rather than assigned another retained status.

Unresolved hypotheses remain in the hypothesis store and can become selectable again when later posts or accepted routed evidence become relevant. Hypotheses converted into derived evidence leave the hypothesis store and become reusable evidence nodes in CPeg. Removed hypotheses do not create derived evidence and are not used for profile projection. This explicit state machine prevents the agent from treating every raw evidence item as a final profile claim.

B.3 Adaptive Model-Tool Routing

Routing inputs. The router receives the current hypothesis, the current CPeg state, candidate entities, region hints, and active attributes. It outputs a route consisting of a model family and a tool family. This design reflects that privacy investigation involves heterogeneous subtasks: some require OCR, some require high-resolution visual inspection, some require map search, and some require multi-step reasoning or evidence verification.

Routing table. The current routing policy uses a deterministic routing table with an LLM fallback when no rule matches. Rules condition on attribute slots, attribute classes, visual tags, entity types, and region hints. Examples include routing navigation screenshots or Chinese address fragments to map search, routing documents to zoom followed by OCR, routing workplace or landmark images to stronger visual inspection followed by web or map verification, and routing brand or luxury cues to web search for price or entity lookup.

Routing criteria. The router scores candidate routes using four criteria: (1) *evidence need*, i.e., what missing evidence would support or narrow the hypothesis; (2) *expected evidential gain*, i.e.,

whether a route can plausibly produce such evidence; (3) *cost and budget*, i.e., whether the route is worth its model/tool cost under the remaining budget; and (4) *duplication avoidance*, i.e., whether the same route has already been attempted for the same hypothesis and source.

Tool families. The canonical tool families are map search, web search, OCR, zoom, webpage fetching, and stop. Map search uses Amap for China-related locations and Google Maps for other locations. Web search is used for institutions, companies, venues, products, and public webpages. Zoom and image cropping are used before OCR or visual re-inspection when small regions contain relevant text or objects.

B.4 Evidence Verification and Graph Updates

Evidence types. Argus uses three evidence types. Raw evidence is directly observed from captions, hashtags, metadata, platform-visible fields, and lightweight visual perception. Router-collected evidence is returned by selected routes such as OCR, web search, map search, image cropping, zoom, webpage fetching, or deeper visual inspection. Derived evidence is a supported hypothesis materialized as evidence after the verifier confirms that its supporting evidence chain is grounded.

Verification. After each routed evidence-collection step, the verifier checks whether the candidate routed evidence is grounded in the original post or tool output, relevant to the target hypothesis, reliable enough for use, and not too ambiguous or unrelated. Accepted routed evidence is admitted into the evidence graph and linked to the hypothesis through support edges. Rejected, irrelevant, unreliable, or contradictory outputs are discarded or kept only in verification logs. The verifier then checks whether to admit the hypothesis as derived evidence, leave it unresolved, or remove it because the claimed value is contradicted, invalidated, or fails to be grounded after available checks. Supported hypotheses are materialized as derived evidence nodes, with support links from the accepted evidence chain that justified them.

Graph structure. The CPeg contains post nodes, evidence nodes, hypothesis nodes, and typed relations. Relations include citation edges from posts to evidence and support edges from evidence to either evidence or hypotheses. When one supported hypothesis is used to support another hypothesis, it is first represented as derived evidence and then linked with a normal support edge. This graph is used both for profile projection and for auditing the evidence trail behind each inferred attribute.

Profile projection. At the end of a user run, Argus projects the derived evidence subgraph into the final privacy profile. For each attribute, Argus merges duplicate or overlapping profile-bearing evidence, selects the best-supported value, and reports the attribute-value pair in the final profile. The evidence ids and support chains remain available in CPeg for audit and qualitative analysis. Unresolved or removed hypotheses do not create derived evidence and are not projected into the final profile.

B.5 Model and Tool Backends

This appendix provides the concrete model and tool backends used in our experiments. Table 11 summarizes the backend assignment for each component, while the execution limits and baseline settings are described below.

Component	Backend
Investigator	GPT-5.5
Perception VLM	Qwen3.6-Plus
Evidence verifier	Qwen3.6-Max
Deep visual specialist	Gemini 3.1 Pro Preview
OCR	PaddleOCR-VL-1.5
Web search	SerpApi
Map search	Amap and Google Maps
Image tools	Local image cropping and zoom

Table 11: Model and tool backends used by Argus.

Model backends. Argus uses GPT-5.5 as the main investigator model. Qwen3.6-Plus is used for lightweight post-level visual perception, while Qwen3.6-Max is used for evidence verification and routing fallback. Gemini 3.1 Pro is used for difficult visual cases that require deeper image understanding, such as landmarks, documents, workplace scenes, or small contextual details. Foundation model calls are served through their respective API backends unless otherwise specified.

Tooling. The tool set includes OCR, web search, map search, webpage fetching, image cropping, and adaptive zoom. We use PaddleOCR-VL-1.5 for OCR on tickets, badges, receipts, screenshots, documents, and small visible text. Web search is performed through SerpApi, and map search uses Amap for China-related locations and Google Maps for other locations. Image cropping and adaptive zoom are used before OCR or visual re-inspection when small regions contain potentially relevant evidence.

B.6 Execution Settings and Compute

Execution limits. Argus processes at most 50 posts per user, matching the retained post window in the dataset. The investigator uses a bounded tool-calling loop with at most 6 iterations for each selected hypothesis and evidence source. STOP leaves the current hypothesis unresolved when no useful route remains, while contradiction or failed grounding removes the hypothesis from the hypothesis store. The user-level loop stops only when no active hypothesis remains, CPeg is stable, or the remaining budget is exhausted. An unresolved hypothesis stopped under the current graph state is suspended rather than immediately retried. It may become active again only when later posts or accepted routed evidence change its evidence neighborhood.

Route budget. The budget B in Algorithm 1 is an implementation-level route budget, not monetary cost. It limits the amount of model-tool investigation that can be spent on one user. Each call to COLLECTEV consumes $\text{COST}(r)$ according to the route family in Table 12. Pure bookkeeping operations, such as hypothesis selection, hypothesis-store updates, graph linking, and profile projection, have zero route cost. Perception is run once for each post before hypothesis investigation and is logged in the runtime statistics, but it is not charged against B. The verifier runs after each investigated route and is included in model-call and token statistics, but B controls investigation breadth rather than total API usage. In our experiments, we set the per-user route budget B to 20 cost units.

Route family	Cost	Examples
Local crop / zoom	0.5	Region selection, patch enlargement
OCR	1	Ticket, badge, receipt, screenshot text
VLM re-inspection	1	Rechecking visible entities or scene details
Webpage fetch / web search	2	Institution, venue, product, public page lookup
Map search / geolocation	2	Address fragment, station, route, landmark lookup
Deep visual specialist	3	Small text, landmark, document, or workplace cases

Table 12: Route-budget cost units used by Argus. Costs are implementation-level control units for bounded investigation, not API price or latency.

Compute environment. Local preprocessing, OCR, image utilities, and evaluation scripts are run on a server with two Intel Xeon Platinum 8369B CPUs, totaling 64 physical cores and 128 logical threads. The server has 8 NVIDIA L20 GPUs. Foundation-model inference is performed through API backends, while local image processing and OCR-related utilities use the local compute environment when applicable.

B.7 Runtime and Cost Analysis

Table 13 reports estimated average per-user runtime and cost statistics. Model calls include LLM and VLM API calls. Tool calls include OCR, web search, and map search; local image cropping and zooming are not counted as tool calls. Token counts include text tokens and provider-side image-token equivalents.

Method	Model calls	Tool calls	Input tok.	Output tok.	Latency
TextLLM	1.0	0.0	7.2K	1.5K	0.5 min
PostVLM	11.0	0.0	43.5K	5.4K	2.6 min
SelfDisc	11.0	0.0	35.8K	4.2K	2.1 min
Holmes	13.2	1.6	50.4K	6.1K	3.4 min
SingleAgent	7.5	5.1	78.6K	7.9K	4.8 min
Argus	46.3	9.2	104.7K	13.2K	8.9 min

Table 13: Average per-user runtime and cost statistics. Token counts include text tokens and image-token equivalents.

B.8 Baseline Settings

All baselines use the same user-level inputs and the same retained post window. TextLLM receives all post text t_i in one user-level prompt and does not use images or tools. PostVLM analyzes posts independently with a VLM and aggregates the per-post outputs into a user profile. SingleAgent receives the same public posts, images, and tools as Argus, but runs everything in one agent context without separate perception, verification, adaptive model-tool routing, or CPeg-based evidence management. SelfDisc follows a post-level self-disclosure pipeline and aggregates detected disclosures. Holmes follows a visual extraction and profile summarization pipeline without Argus's verifier, adaptive routing, or CPeg.

C Evaluation Details

LLM-as-a-judge setup. We use LLM-as-a-judge for two semantic evaluation steps: attribute-slot matching and value-granularity scoring. Attribute-slot matching decides whether a predicted attribute refers to the same private attribute type as the

Baseline	Description
TextLLM	One-pass profile inference from all post text t_i.
PostVLM	Per-post image-text analysis followed by profile aggregation.
SelfDisc	Post-level detection and aggregation.
Holmes	Visual cue extraction and profile summarization.
SingleAgent	One multimodal tool-using agent.

Table 14: Baseline methods.

benchmark slot. Value-granularity scoring then maps the predicted value to the deepest correct level in the benchmark-provided hierarchy. The judge receives the ground-truth slot, ground-truth value, system-predicted attribute and value, and the attribute-specific granularity hierarchy when applicable. It does not assign contextual sensitivity or inference difficulty; these are fixed benchmark annotations. All judge calls use the same evaluator model (GPT-5.5) across methods.

Evaluator Prompt

You are evaluating one privacy-inference system prediction against one benchmark annotation.
Input:

- Ground-truth attribute slot and value.
- System-predicted attribute and value.
- Attribute-specific granularity hierarchy when applicable.

Definitions:

- Attribute-slot match: true only when the prediction refers to the same private attribute type as the ground-truth slot, even if the wording differs. For example, "where the user lives" matches Home Address, while workplace location does not match Home Address.
- Value-granularity score: if the slot matches, compare the predicted value with the ground-truth value along the provided hierarchy. Assign 0 if the value is wrong, unsupported by the hierarchy, or refers to a different scope. Otherwise, assign the score for the deepest correct hierarchy level reached by the prediction.

Tasks:

1. Decide slot_match.
2. If slot_match=false, set granularity_score=0 and matched_level=none.
3. If slot_match=true, identify the deepest correct hierarchy level matched by the predicted value and assign granularity_score.
4. Return a structured JSON object with slot_match, matched_slot, granularity_score, matched_level, and a brief justification.

Do not infer contextual sensitivity or inference difficulty. These are provided by the benchmark.

Value-granularity score. The value-granularity score g_j is computed by an LLM judge using an attribute-specific hierarchy. The judge receives the attribute name, ground-truth value, predicted value, and prediction reasoning. It first determines whether the predicted value matches the ground truth at any level. If the prediction is wrong or refers to a different attribute scope, $g_j = 0$. Otherwise, the score is assigned according to the deepest correct hierarchy level reached by the prediction.

Uncertainty estimates. Because multiple leaked attributes from the same user are correlated, we estimate uncertainty by resampling users rather than individual attributes. For each bootstrap sample, we sample 50 users with replacement and recompute the aggregate metric over the selected users. For paired comparisons, the same resampled users are used for both methods. We report 95% percentile intervals for the paired PES difference.

Human audit of evaluator reliability. To check whether the automatic evaluator is stable enough for the benchmark, we manually audit a stratified sample of evaluator decisions across methods, attribute dimensions, and inference difficulty levels. The audit covers 78 ground-truth leaked attributes (10% of 781). We sample them by attribute dimension and inference difficulty, including 18 identity, 21 socioeconomic, 25 lifestyle, and 14 sensitive attributes, with 5/11/26/36 attributes from D1–D4. For each sampled attribute, we audit the evaluator decisions for the six main methods and the evidence-verification ablation, yielding 546 attribute-system decisions. Two human annotators independently review the ground-truth slot, ground-truth value, model prediction, judge attribute-slot matching decision, and judge value-granularity score. The two annotators agree with each other on 95.6% of attribute-slot matching decisions and 90.8% of value-granularity decisions. After discussion, the remaining disagreements are resolved by adjudication. Compared with the adjudicated human labels, the automatic evaluator agrees on 94.0% of attribute-slot matching decisions and 91.2% of value-granularity decisions.

Disagreements are concentrated in semantically fuzzy attributes such as lifestyle, psychological traits, and broad socioeconomic descriptions, while exact or hierarchical attributes such as home address, school, workplace, and document-like cues show higher agreement. When we recompute PES on the audited subset using adjudicated human labels, the ranking of the six main methods remains unchanged, and the evidence-verification ablation remains below the full Argus. We use the automatic evaluator for the full benchmark and report the human-audit statistics as a reliability check rather than as a separate scoring procedure.

D Detailed Results

Difficulty breakdown. Table 15 reports PES by inference difficulty. The gap between Argus and the baselines becomes larger as inference moves from single-post text cues to image-based, mixed, and cross-post evidence aggregation. For single-post text leakage, Argus improves moderately over SingleAgent, from 0.60 to 0.65. For image leakage, it improves from 0.51 to 0.60. For mixed leakage, the gap increases from 0.55 to 0.66. For cross-post leakage, Argus reaches 0.56, compared with 0.39 for SingleAgent, 0.31 for Holmes, and 0.25 for PostVLM. This suggests that Argus is especially useful when privacy leakage requires connecting weak cues across posts or modalities.

Method	Text	Image	Mixed	Cross-post
TextLLM	0.48	0.17	0.21	0.14
PostVLM	0.53	0.40	0.43	0.25
SelfDisc	0.51	0.31	0.29	0.19
Holmes	0.55	0.45	0.47	0.31
SingleAgent	0.60	0.51	0.55	0.39
Argus	**0.65**	**0.60**	**0.66**	**0.56**

Table 15: PES breakdown by inference difficulty.

Taxonomy breakdown. Table 16 reports PES across the four attribute dimensions. Argus performs consistently best across all dimensions, where correct predictions often require combining visual cues, text snippets, and public contextual information.

Ablation results. Table 3 reports the full ablation results in the main text. Removing the Perceiver lowers PES to 0.45, indicating that a broad initial skim is important for raw cue recall. Removing the Hypothesizer lowers PES to 0.48, showing that persistent hypothesis state matters even when evidence is available. Removing the Investigator lowers PES to 0.42, indicating the importance of routed OCR, web search, map search, and fine-grained visual inspection. Removing the Verifier lowers PES to 0.50 despite increasing binary accuracy, while removing CPeg lowers cross-post PES to 0.38.

Method	Identity	Socioecon.	Lifestyle	Sensitive
TextLLM	0.26	0.20	0.21	0.18
PostVLM	0.36	0.31	0.34	0.29
SelfDisc	0.31	0.27	0.29	0.25
Holmes	0.43	0.36	0.40	0.34
SingleAgent	0.50	0.43	0.46	0.41
Argus	**0.61**	**0.56**	**0.57**	**0.52**

Table 16: PES breakdown by attribute taxonomy. Socioecon. denotes socioeconomic status.

D.1 Qualitative Investigation Trace

Figure 3 illustrates a representative residence inference from a synthetic user in SopriBench. The target private attribute is not directly disclosed by any single post. In one post, an outdoor image contains the name of a residential community, but this cue is ambiguous because communities with the same name can appear in multiple cities. Argus therefore keeps the location hypothesis unresolved rather than projecting it into the profile. In a later post, another image contains small OCR-visible text, including a district name and a property-management fee notice. The router invokes OCR and retrieval; after verification, Argus links the district cue to a specific city and retrieves public images of matching residential communities. Visual comparison with the earlier post supports the same community hypothesis, and the property-management fee notice further suggests that the user is a resident rather than a visitor. After these checks, the supported hypothesis is materialized as derived evidence, and the evidence chain is projected into the final profile through CPeg. By contrast, SingleAgent's error occurs at the cross-post binding step. It can mention the community sign from the first post and the district or property-fee text from the second, but it does not maintain an unresolved residence hypothesis in CPeg, route the later district cue back to check the earlier sign, or convert the supported chain into profile-bearing evidence. Its final profile therefore remains vague or incomplete and misses the resident-status evidence.